EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

A Review on Smart Helmet for Accident Detection using IOT

H.C. Impana[1,*], M. Hamsaveni[2] and H.T. Chethana[2]

[1]Mtech Student Department of Computer science and Engineering Vidya Vardhaka college of Engineering Mysore, India
[2]Assistant Professor Department of Computer science and Engineering Vidya Vardhaka college of Engineering Mysore, India

Abstract

As we know that accidents are increasing day by day, we can also notice that many laws and regulations are posed by government in order to avoid this accidents. Accidents can be defined as the unplanned event or the mistake that may occur resulting in injury and sometimes it also leads to death. The accidents in case of two wheelers are more compared to other vehicles. This may be avoided by wearing helmets and riding vehicles without consuming alcohol. This survey is on smart helmet for accidence avoidance and also examining various related techniques. This research also helps us to understand IOT technology which is being emerged now a days .From the literature survey we find that the method proposed using microcontroller RF transmitter and other sensors is cost effective but we find the system proposed using Raspberry pi module, Pi camera, Pressure Sensor, GPS system which uses image processing algorithms is most efficient since the image processing is included so that we can easily detect the use of helmet from the rider. Smart helmet system helps to provide safety and security to the two wheeler riders.

Keywords: Accidents, smart helmet, IOT, Laws and Regulation.

Received on 11 April 2020, accepted on 08 May 2020, published on 14 May 2020

doi: 10.4108/eai.13-7-2018.164559

1. Introduction

Internet of things are currently being used in many fields such as wearable's, home automations, smart appliances, smart agriculture etc where there is a mutual communication between devices and people over a network. The work of the IOT devices is to sense the data and send the data to server by this huge amount of data can be generated. By the generated data we can draw the conclusion by processing and analysing the data obtained. This gives the advantage in real time data reporting from environment. Now a days motorbike accidents are increasing day by day and we can notice numerous loss in lives. We can avoid this by using smart helmet. From the survey we can know that in India 4 people die every hour because they do not wear helmet. In 2017, more than 48,746 two wheeler user died in road accidents, Incidental 78.3% of them did not wear a helmet. To go through or to solve this, there are two important conditions that should be checked before the bike starts by the smart helmet. First most condition is that we should check whether the rider is using the helmet and not just keeping it. Second to check whether the user has consumed alcoholic substance or not by his breath, this can be verified by using sensors. Third if a person meets with an accident, the sensor check the condition of person and bike and send information of location to nearby hospital. If the person has no major injurious then the button is pressed which is present in the bike this indicate that the person condition is good.

*Corresponding author. Email:impanachannegowda@gmail.com

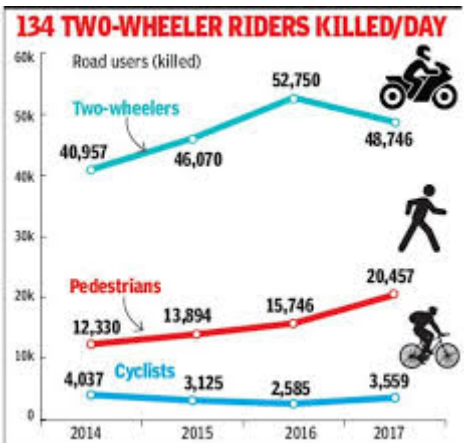


Figure 1. Representation of rate of accidents in two wheeler

The figure 1 gives the picture of rate of accident in two wheelers. The comparison is done between two wheelers, pedestrians, cyclists and the rate of accident is more as represented in graph.

2. Literature survey

In this survey we are discussing various smart helmets with various approaches and methodologies.

Jesudoos A et.al[1] proposed a mechanism, where sensors such as IR sensor, vibration sensor and gas sensor, mems are used. The gas sensor is used to detect the amount of liquor he had consumed by checking the breath of a person wearing the helmet. The bar control of the vehicle is handled by MEMS. Accident is detected by vibration sensor. Load of the vehicle is recognized by load checker. The Sensors are interfaced with the PIC microcontroller. The gas sensor will detect if a user consumed alcohol and display on the LED display. If an accident occurs the vibration sensor, sense the accident and send information through GPS to the hospital .If there is any rash driving is done by the rider the MEME sensor detect the amount of the person from his bank account. To check whether the rider is wearing the helmet or not IR sensor is used. In this system exactness and accuracy are high and ambulance is booked automatically based on ten location.

K.M. Mehata et.al[2] proposed a techniques which provide safety to the workers or to identify any fall of the workers in working area. The proposed system has two components. One is the wearable device built using sensors and electronic elements. Another component is the cell phone. The communication between the two components is provided by GSM module. These devices also monitor the health and safety of the worker is continuously. This system ensures good fall detection and alert the register person to give medical attention.

Divyasudha N et.al[3] proposed a system consists of micro controller, position sensor, Alcohol sensor, piezoelectric sensor, RF Transmitter, IOT Modem, GPS receiver, Power supply & Solar panel to avoid the accidents and check the alcohol consumption. In this system two condition is checked that is whether the rider is wearing the helmet or not and to check whether he had consumed alcohol or not if this is not followed by the rider the bike will not start and it is indicated by beep sound. If any accident Occur it is informed to predefined number and police station using IOT modem. This system is cost efficient compare to other kind of helmets.

Manish Uniyalet.al[4] proposed a system with two units that is helmet unit and two wheeler unit. RF receiver of the matching frequency gives the helmet position data to the two wheeler section. The microcontroller placed on the TW section will have information of the helmet position which is continuously checked. There are various other sensors such as accelerometer (tilt angle measurement), Hall-effect sensor (speed measurement), GPS module (location pointer) placed on the TW vehicle. The sensors collect the data and send the data to the microcontroller then if there is a internet connection then it is sent to the server. The speed of the vehicle can be accessed by the people at any instant by this method. In this system people can access the speed of the vehicle. Parents can see that is their child have worn helmet or not.

ShoebAhmed Shabbeer et.al [5] proposed the smart helmet method which detect and report the accidents. In this method they use microcontroller interfaced with accelerometer and GSM module. The notification and report of the accident is provided using cloud infrastructures. In this method if the level of the acceleration exceeds than the threshold or if any accident occurs the information is sent to the emergency authority server which then sends the message to the assigned emergency contact through GPS module. The result of this system was able to identify accidents is of 94.82% and sends the correct coordinates 96.72% of time.

P.Rojaet.al[6] has proposed a system consisting a 6 units as follow, that is remover sensor, IR sensor, Air quality sensor, Arduinouno microcontroller, GPRS, GSM. This helmet provides the alert about the harmful gases in the mining areas to the workers and also proved information to the server if helmet is removed. Here this data transmission is done using IOT technology .

C.J Bheret.al[7] has proposed a system of smart mining helmet that detects three types of hazards that is harmfull gases, remove of helmet and if any collision. Here they uses many sensors such as IR sensors, gas sensors, accelerometer.

SreenithyChandran et.al [8] has proposed a system of smart helmet named konnect. Here they use integrated network of sensors, WiFi enabled processors, cloudcomputing infrastructures to detect and prevent the accidents. This system also provide the information to the provided contact by text message if the speed is increased than the threshold level.

Mohammed Khaja Areebuddin Aatifet.al[9] proposed a technique consisting of arduinouno, Bluetooth module, push button and 9V battery. Here the smart helmet integrated with Bluetooth is connected to the cell phones and push button is used if any emergency occur.

Archana.Det.al[10] proposed a system to reduce accidents, here the system consist of a sensor which sense the human

touch when he plug in the bike key. After he wear the helmet the sensor automatically lock the helmet and he can only remove is when bike is stopped.

Ahyoung Lee et.al [11] proposed a system based on three sensors: acceleration sensor, ultrasonic sensor, and carbon monoxide sensor, and also based on an Arduino MCU (Micro Controller Unit) with a Bluetooth module to provide safety to the workers.

Agung Rahmat Budiman et.al [12] proposed a system of smart helmet which is integrated with several functionalities. Warning notification is given if a rider is not wearing helmet and if he come with unsafe conditions and if helmet is not correctly locked so that to provide safety to the rider. In this system warning to the rider is generated in the form of notification to notify him in the unsafe condition. In the functionality test it is 100% success rate in 4 smart helmet features and 98.3% success rate in the communication test between the 2 modules.

Sayan Tapadaret.al[13] also proposed a prototype which detects the rate of alcohol consumed by the rider and detecting the accidents using IOT module and sensors. Here they are trying to use Support Vector Machines to predict if the values of the sensors correspond to an accident or not, by training the device using real-time simulation. This system gives satisfactory results. The accuracy and precision is also high.

Prashant Ahujaet.al[14] proposed smart helmet system using GSM and GPRS module. As we all know that the arrival of ambulance to the location may be late this prototype helps to inform the concerned person first about the accident and he may take the steps. In this system we can notice the feature such as high accuracy, cost efficient and giving information about the accident within minute.

Mingi Jeong et.al[15] proposed a system consisting sensors such as thermal camera, visible light camera, drone camera, oxygen remaining sensor, inertia sensor, smartwatch, HMD and command center system to avoid the accidents.This framework allows IOT services to be easily integrated and efficiently managed and able to notify the information in real time.

S.R.Kurkuteet.al[16] proposed a system consisting Raspberry pi module, Pi camera, pressured sensor, inbuilt wifi and GPS system. Image processing algorithms are used to capture the face of the biker. It can be applied in real time and it is cost efficient and effective and also used in any type of vehical.

Kabilan M et.al[17] proposed a system using vibration sensors. When the rider wear the helmet consisting of the vibrator sensor with a frequency if the frequency crosses the threshold then the message is sent to the emergency responses using GPS module. This system helps to detect and report the accident and can save the life. Dr. D

Vivekananda Reddy et.al [18] proposed a system consisting of two sections ie helmet section and bike section. In the helmet section there is a alcohol gas sensor to check if a person is drunk and it also contain IR sensor, Alcohol sensor, LCD display to display the information. In Bike section there is a vibration sensor which sense the accident and sends the information using GSM network and GPS module.

Kimaya Bholaram Mhatreet.al[19] proposed a system which consists of helmet module and bike module. It consists of IR sensors, MQS alcohol sensor, vibration sensor, GSM module, GPS module, Arduino, Intercom system. The workflow of the system is as follow, when the bike starts if the rider has consumed the alcohol and if it is greater than the threshold then the bike don't start else the bike will start and if the vibration sensor limit is greater than the threshold message is sent to the registered number about the accident. This system is cost effective and provide better security to the biker.

3. Comparison

In this section we briefly discuss existing literature review on smart helmet and also discuss various methods applied along with the limitations and accuracy.

Table 1. Provides the comparison of the survey on Smart helmet using IOT.

Authors	Methodology	Limitations	Accuracy
Jesudoss A et.al[1]	Uses Sensors that are interfaced with PIC through the wires. Sensors such as gas sensor, load sensor, vibration sensor, IR sensor and mems sensors are used.	Exactness and accuracy is high.	90%
K.M.Mehata et.al [2]	This method consists of 2 modules Such as health monitoring and safety monitoring of workers. It uses heart beat sensors, temperature sensors, tri-axis accelerometer.	Ensure good fallen detection of the workers in working place.	67%

DivyasudhaN et.al[3]	The system consists of micro controller, position sensor, alcohol sensor, piezoelectric sensor, RF transmitter, IOT modem, GPS receiver, power supply and solar panel.	Cost effective.	85%
Manish Uniyal et.al[4]	There are 2 units namely helmet unit and two wheeler section which Uses helmet sensor switch, microcontroller unit, RF encoder, RF transmitter, accelerometer module, GPS module, speed sensor.	Tilt angle of the vehicle is also detected using accelerometer module. This helps us to know whether the has fallen or not.	92%
Shoeb Ahmed et.al[5]	Microcontroller interfaced with accelerometer and GSM module.The notification and report of the accident is provided using cloud infrastructures	System canfunction as remote immobilizer in case if vehicle is stolen.	94.82%
P.Roja et.al[6]	It consists of data processing unit(arduinouno),air quality sensors, infrared sensor, GSM modem, alerting unit, liquid crystal display to detect the danger in mining area.	The helmet should be properly weared. It works with proper power supply.	88%
C. J. Behr et.al[7]	Composed of Air Quality Sensor, Helmet Removal Sensor, Collision Sensor, Wireless Transmission, Data Processing Unit, Alerting Unit to detect hazards in industries.	Distance of workers want to be limited from interface.	90%
Sreenithy Chandran et.al[8]	Sensors, Wi-Fi enabled processor, and cloud computing infrastructures are utilised for building the system.	Depends on the response of authorized person	82%
Archana.D et.al[10]	Uses ultrasonic sensors, arduinouno , microcontroller, DC motor,LED	Proper power supply should be provided.	78%
MingiJeong et.al[15]	The smart helmet system consists of Bio & Framework Subsystem(BFS), Multimedia Processing Subsystem (MPS), and Communication Subsystem (CPS).	Power must be on.	81%
AgungRahmatBudiman et.al [12]	Consists of bike module, helmet module, external module.	No alcohol detection	78%
SayanTapadar et.al[13]	Consists of several sensors with accelerometer connected to cell phones with API's.	There may be wrong detection some times	83%

PrashantAhuja et.al [14]	Consists of IR sensor, vibration sensor, tilt sensor, NC sensor, microcontroller interface, GSM, GPRS connected to mobile.	The tilt sensor may fail to detect.	79%
S.R Kurkute et.al[16]	Consists of Raspberry pi module, Pi camera, Pressure Sensor,GPS system and uses image processing algorithms.	-	98%
Kabilan M et.al[17]	Consists of vibration sensor, GSM module, GPS module.	Network issues	86%
Dr. D. Vivekananda Reddy[18]	Consists of Helmet section and bike section consisting of sensors.	Power supply is important.	76%
KimayaBholaramMhatre[19]	Consists of Helmet module and Bike module which consists of IR sensor, MQS alcohol sensor,vibration sensor, GSM module,GPS module,Arduino,Intercom.	Need of 3.3V voltage supply for RF module.	81%

From the comparison and survey we can come across the methodology limitations and accuracy. Here we find the method proposed using microcontroller RF transmitter and other sensors is cost effective but we find the system proposed using Raspberry pi module, Pi camera, Pressure Sensor, GPS system and uses image processing algorithms is most efficient as the image processing is included so that we can easily detect the use of helmet from the rider.

4. Applications of Smart Helmet

1. We can use smart helmets in real life it acts as real time application.
2. The Smart helmets can be used as the key as without the helmet we cannot start the vehicle.
3. Smart helmets can be used to warn triple riding, alcohol consumption, using mobile phone and also rash riding.
4. We can also use smart helmet in mining areas and also in construction area to provide safety to the workers.

5. Conclusion

The survey demonstrates Smart helmet for accident avoidance. The helmet should be designed in order to reduce number of accidents in two wheelers this can be done by designing the device using IOT technology. Some sensor like IR sensor, alcohol sensor, GPS modules etc can be used to design a cost effective and user friendly smart helmet. The result should be accurate and should be useful to the government and society. This smart helmet can also be changed to seat belt system in case of four wheelers and can be implemented in future.

References

[1]JesudossA, Vybhavi R, Anusha B “Design of Smart Helmet for Accident Avoidance” International Conference on Communication and Signal Processing, April 4-6, 2019,India.

[2]K.M.Mehata, S.K.Shankar, Karthikeyan N, Nandhinee K, Robin Hedwig P “IoT Based Safety and Health Monitoring for Construction Workers.Helmet System with Data Log System” International Conference.

[3]DivyasudhaN,ArulmozhivarmanP,RajkumarE.R “Analysisof Smart helmets and Designing an IoT based smart helmet: A cost effective solution for Riders” @IEEE.

[4]Manish Uniyal, Manu Srivastava, HimanshuRawat, VivekKumarSrivastava “IOT based Smart Helmet System with Data Log System” International Conference on Advances in Computing, Communication Control and Networking.

[5] Shoeb Ahmed Shabbeer, MerinMeleet “Smart Helmet for Accident Detection and Notification”2nd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions 2017.

[6] P.Roja, D.Srihari “IOT Based Smart Helmet for AirQuality Used for the Mining Industry”@IJSCRT 2018.

[7] C. J. Behr, A. Kumar and G.P. Hancke“ASmart Helmet for Air Quality and HazardousEvent Detection for the Mining Industry”@IEEE2016.

[8] SreenithyChandran, SnehaChandrasekar, Edna Elizabeth N “Konnect: An Internet of Things(IoT) based Smart Helmet for Accident Detection and Notification.

[9]MohammedKhajaAreebuddinAatif,AinapurapuManoj“Smart-HelmetBasedOnIoTTechnology”@IJRASET 2017.

[10]Archana.D,Boomija.G,Manisha.J,Kalaiselvi.V.KG “Mission On! Innovations in Bike Systems to Provide a Safe Ride Based

on IOT"@IEEE 2017.
[11] AhyoungLee, JunYoung Moon, Se Dong Min,Nak-Jun Sung, and Min Hong4"Safety Analysis System using Smart Helmet"@CSREA.
[12]AgungRahmatBudiman,DodiWisaksonoSudiharto, Tri Brotoharsono "The Prototype of Smart Helmet with Safety Riding Notification for Motorcycle Rider" 2018 3rd International Conference on Information Technology,Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia.
[13]SayanTapadar, ShinjiniRay,Arnab Kumar Saha, Robin Karlose, Dr. HimadriNathSaha "Accident and Alcohol Detection in Bluetooth enabled Smart Helmets for Motorbikes" @IEEE2018.
[14]PrashantAhuja,Prof. KetanBhavsar"Microcontroller based Smart Helmet using GSM & GPRS" @IEEE2018.
[15]MingiJeong, Hyesun Lee, MyungnamBae, Dong-BeomShin,Sun-HwaLim,Kang Bok Lee "Development and Application of the Smart Helmet for Disaster and Safety"@2018IEEE.
[16]S.R.Kurkute, N.R.Ahirrao, R.G.Ankad, V.B Khatal "IOT based smart system for the Helmet detection" SUSCOM-2019.
[17]KabilanM,Monish S, DrS.Siamala Devi "Accident detection system based on IOT-Smart Helmet" IJARIIT 2019.
[18]Dr.D Vivekananda Reddy,V.Suresh,T.Hemalatha "Smart Helmet and Bike management system" Journal of Gujarat Research Society 2019.
[19]Kimaya Bholaram Mhatre, Raj Maruthi N and wadeka, Aditya Prasanna Patil Rushikesh vijaysinde, Prof.Pralnyakamble "Smart Helmet with Intercom feature" SSRN 2020.

Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers

Conference Paper · August 2014

DOI: 10.1109/SIBGRAPI.2014.28

3 authors:

Romuere Silva
Universidade Federal do Piauí
60 PUBLICATIONS **617** CITATIONS
SEE PROFILE

Kelson R. T. Aires
Universidade Federal do Piauí
45 PUBLICATIONS **471** CITATIONS
SEE PROFILE

Rodrigo Veras
Universidade Federal do Piauí
89 PUBLICATIONS **621** CITATIONS
SEE PROFILE

Multimed Tools Appl
DOI 10.1007/s11042-017-4482-7

Detection of helmets on motorcyclists

Romuere R. V. e Silva[1] · **Kelson R. T. Aires**[2] · **Rodrigo de M. S. Veras**[2]

Received: 21 March 2016 / Revised: 2 February 2017 / Accepted: 6 February 2017

Abstract The use of motorcycle accidents has rapidly increased. Although the helmet is the main safety equipment of motorcyclists, many drivers do not use it. This paper proposed a method for motorcycle detection and classification and a system for the detection of motorcyclists without helmets. For vehicle classification, we have employed the wavelet transform (WT) as the descriptor and the random forest as the classifier. For helmet detection, the circular Hough transform (CHT) and the histogram of oriented gradients (HOG) descriptor were applied to extract the image attributes, and the multilayer perceptron (MLP) classifier was used to classify the objects. The results for vehicle classification achieved an accuracy rate of 97.78 %. The algorithm step in the helmet detection accomplished an accuracy rate of 91.37 %. The results were obtained with the author's database.

Keywords Helmet detection · Descriptors · Classifiers

1 Introduction

Due to the large number of vehicles in circulation, studies of intelligent traffic systems have increased. The majority of studies focus on the detection, recognition, tracking and counting of vehicles and on the estimation of traffic parameters.

Motorcycles are a predominant method of transport in many countries. The main advantages of motorcycles are their low price and operation cost compared with other vehicles. According to DENATRAN(National Department of Traffic), Brazil maintained a fleet of 20.281,986 motorcycles in 2013 [11].

✉ Romuere R. V. e Silva
romuere@ufpi.edu.br

[1] Information Systems, Federal University of Piauí, Picos, Brazil

[2] Computer Science Department, Federal University of Piauí, Teresina, Brazil

The number of accidents involving motorcycles has increased during the last decade. According to the DNIT, a total of 34,635 motorcycles were involved in accidents in Brazil in 2011 [12]. According to a study of traffic accidents [32], 14,666 traffic fatalities occurred in 2011.

This global problem is prevalent. According to statistics from 2012 [26], more than 17 million motorcycles were officially registered in Thailand in 2010. This large number of vehicles increased the potential for a large number of traffic accidents. According to a report on road safety published by the World Health Organization (WHO) in 2013 [35], the estimated rate of deaths per 100.000 people in Thailand was 38.1. According to the report, Thailand has the third highest traffic mortality rate in the world. In Brazil, the traffic mortality rate was 22.5; in the United States (US), the rate was 11.4. The report noted that approximately 25 % of traffic fatalities in Brazil involved motorcyclists.

The main safety equipment used by motorcyclists is the helmet. Although helmet use is mandatory, some motorcyclists do not use them or incorrectly use them. According to the US Department of Transportation's National Highway Traffic Safety Administration (NHTSA), only 66 % of motorcyclists used a helmet in accordance with the law [22].

The US Centers for Disease Control and Prevention presented a Morbidity and Mortality Weekly Report (MMWR) in 2012, in which the results indicated that 12 % of fatally injured motorcyclists were not using a helmet in states with universal helmet laws. In the states that adopt partial helmet laws , the number increased to 64 %. A total of 79 % of motorcyclists use helmets in states without helmet laws [21].

The motivation of this work is to improve surveillance on the main roads in locations where the use of helmets is mandatory. These data reveal the need for increased enforcement of traffic laws, particularly for offenses for which there are no automatic detection methods. The increase in the number of motorcyclists using helmets causes a decrease in the number of accidents with victims, which is high in those countries.

This study aimed to develop a computational vision methodology to detect motorcyclists without helmets. A two-stage strategy was developed, namely, the detection of motorcycles and the detection of helmet use. The main contribution of this work is related to a general solution proposed to detect motorcyclists without helmets, not the development of new algorithms for each stage of the system.

The proposed methodology has been designed to be applied in American and Asian countries such as Brazil, USA, India, Thailand and others. In these countries the number of motorcyclists and accidents are high, as shown in the statistics. Moreover, in some of those countries there are laws that require helmet use by motorcyclists.

This paper is organized into 5 Sections. Section 2 reviews studies that have been published in this area. Section 3 details the proposed methodology. In Section 4, the results obtained during all stages of the system are detailed, including the metrics for the evaluation of the results. Section 5 shows discussions about the results. Section 6 presents the conclusions and recommendations for future studies.

2 Related studies

In recent years, several studies were performed to analyze traffic on public roads, including the detection, classification and counting of vehicles and helmet detection. The detection and segmentation of vehicles on public roads can be considered as the first step to develop any study related to vehicular traffic. For this reason, some relevant studies are discussed in this section.

A system of vehicle segmentation and classification was proposed by Messelodi et al. [20]. Eight three-dimensional models were created to classify the objects. These models were calculated based on the size of the vehicle, which varies depending on the type of vehicle. The created models were as follows: cycle (motorcycles and bicycles), small car, car, minibus, closed truck, open truck, bus and pedestrian. Models are generated for each captured vehicle, which are compared with the models created for each class of vehicle. The model that is closer to the model of the captured vehicle defines the vehicle class. A disadvantage of this study is that a single model is employed for motorcycles and bicycles. In addition, only geometric information is utilized to classify the vehicles, which has been proven to be insufficient for describing these types of objects. Another disadvantage of this method is that some parameters, such as the camera height and angle and the lens focal distance, should always have the same values. If these parameters are changed, new models should be created, as no vehicle would correspond to the models. Some public roads that did not have a location, such as described in this study, to position the camera, were unable to operate the system.

Leelasantitham and Wongseree [15]proposed a technique that consists of detecting moving vehicles using traffic engineering methods. The vehicles were divided into five groups. The first group consisted of bicycles, motorcycles and tricycles. The second group consisted of cars and vans. Minibuses and small-sized trucks comprise the third group. The fourth group included medium-sized trucks and buses. The fifth group was composed of large-sized trucks and trailers. The features used for the classification were the length and the width of the image. The database consists of only 76 images, which undermines the validity of the results. Note that the system is very sensitive to the road on which the images are captured, that is, all parameters, such as the length and the width of the image, should be modified if the same system is applied to a different road.

In another study, Zengqiang et al. [17] implemented a system that identifies and segments vehicles if they are partly occluded. The system implements a tracking algorithm, which will continue to identify the detected vehicle even if an occlusion occurs. A tracking algorithm is presented based on the features of the contour points.

Takahashi et al. [30] introduced a computational vision system for the detection of bicycles, pedestrians and motorcycles. The system detects both moving objects and the pedalling movement (cyclists) using the Gabor filter [1]. The histogram of oriented gradients (HOG) descriptor [9] is applied to extract the characteristics of the image. The support vector machine (SVM) classifier [8] is employed in two stages. In the first stage, the objects are divided into two classes: two-wheeled vehicles and pedestrians. In the second stage, the two-wheeled vehicles are classified as motorcycles and bicycles. The vertical movement of the scene is calculated using the Gabor filter. If the vertical movement is pedalling, the vehicle is classified as a bicycle. Other types of vehicles, such as cars, vans and buses, travel on a public road.

Sonoda et al. [28] proposed a system for detecting moving objects at an intersection. The objective of the system is to alert a driver to prevent accidents at the intersection. The Adaptive Gaussian Mixture Model (AGMM) [36] was employed for the calculation of the scene background for detecting moving objects. After the detection of the objects, a tracking algorithm is employed. The tracking algorithm is utilized as the moving objects appear in different positions during the sequence of video frames. The Lucas-Kanade tracker algorithm [16] was employed as the tracking algorithm. The tracking of objects determines whether a vehicle and a person will collide. If a pedestrian and a vehicle simultaneously traverse the intersection, the system will issue a warning. The possibility of false positives in this system is high as a pedestrian can change routes after the system issues the warning.

Chen et al. [5] presented a system for the detection, tracking and classification of vehicles. The system counts the vehicles and classifies them in four categories: cars, vans, buses and motorcycles (including bicycles). The Kalman filter is employed as the tracking algorithm. The HOG descriptor and the SVM classifier are employed to extract the image features and to classify the objects, respectively. The hit rate of the system decreases when weather and lighting conditions change.

Previous studies have focused on the detection of vehicles, that is, segmentation and classification. For the proposed system, this stage is essential, as helmet use can only be detected after a motorcycle on the road is detected. Next, some studies related to the detection of helmets are addressed.

Wen et al. [34] proposed a detection method for circular arcs based on the modified circular Hough transform (CHT) [13]. A threshold value is manually defined to calculate the edges of the image. The circle Hough transform is subsequently applied. The transform searches for circular regions, such as a helmet. The method was applied to surveillance systems in automated teller machines (ATM). According to the authors, the majority of ATM criminals wore a helmet to prevent recognition of their faces. The main limitation of this study is the sole use of geometric resources to identify a helmet in the image. Geometric characteristics are not sufficient for locating the helmet; the helmet can be confused with a human head, as their shapes are similar.

Chiu et al. [6] proposed a computational vision system with the objective of detecting and tracking motorcycles that are partially occluded by another vehicle. A vehicle counting system is proposed. A helmet detection system is employed to detect a motorcycle. The system assumes that the helmet region has a shape that resembles a circle. To detect the helmet, the edges of the image are calculated over its possible region, that is, the region where the motorcycle is located. The number of edge points that resemble a circle are subsequently counted. If this number is greater or equal to a predefined value during the calibration of the system, the region will correspond to a helmet. If the system detects a helmet, a motorcycle is assumed to exist in the same location. In its calibration stage, the system requires some parameters to be inputted by the system operator, such as helmet radius, camera angle and height. If any condition, such as camera height or the road on which the system is in operation, changes, all parameters should be altered.

Chiverton [7] described and tested an automatic tracking and classification system of motorcyclists with and without helmets. The system employs the SVM classifier, which is trained with histograms from the image data in the head region of the motorcyclists computed by the HOG descriptor. The HOG descriptor is executed with the Sobel operator [27] to calculate the edges with a neighborhood of 3×3 pixels. The AGMM algorithm is used to calculate the background. In the extraction of the histograms, static photographs and individual frames from video data are utilized. The method obtained a total hit rate of 83 % for the classification of motorcycles and a total hit rate of 85 % for the detection of helmets.

The most recent study of the detection of helmet use was proposed by Waranusast et al. [33]. The system extracts moving objects from videos using the AGMM algorithm. After the objects are extracted, the system classifies them as motorcycles or other objects. For this purpose, three features are employed: the area of the rectangle that contains the image, the ratio between the width and the height of the rectangle and the standard deviation of the H band in the hue-saturation-value (HSV) color space around a rectangle at the center of the object. The next step involved the use of the k-nearest neighbors (KNN) classifier with the calculated features. The primary advantage of this study is the counting of passengers, which is performed by the number of heads that appear on the image. In the final

step, a classification is performed using geometric information of the head region and color information. These features are reapplied by the KNN classifier to classify the images of motorcyclists with helmets and without helmets. According to the authors, the motorcycle detection stage obtained a hit rate of 95 %. The passenger counting stage yielded a total of 83.82 % hits. In the helmet detection stage, the hit rate was 89 %. Note that the images of the head region were manually cut in the latter stage. The images were perpendicularly captured by the camera, that is, the images show the side view of motorcycles, which is a flaw of the system for the detection of traffic violations as the vehicle registration plate is difficult to capture in that position. This method facilitates the identification of more than one person on the motorcycle in the images. If the images had been taken from another angle, one of the persons on the motorcycle would most likely be superimposed on another image, which would generate an occlusion.

3 Proposed methodology

This study proposed a computational vision methodology for the detection of helmet use by motorcyclists on public roads. The study is divided into two stages: vehicle segmentation and classification, and the detection of helmet use.

The stage of vehicle segmentation and classification has the following objectives: determining which objects are moving in the scene and classifying these objects. Similar to the majority of computational vision systems, the proposed system requires a calibration stage. In the calibration stage, parameters that are required for the operation of the system are adjusted. In the calibration stage of the proposed system, a cross line (CL) is defined. This line will be marked by the system operator and should cross the public road where the system will be responsible for capturing the vehicles. The moving objects that cross the CL are subsequently extracted from the video frame. The next step involves extracting the features of the segmented objects. The wavelet transform (WT) was employed [10, 18]. The vectors are the input parameters of the classifier. The random forest classifier was employed to classify the vehicles [4]. The images are grouped into two classes: motorcycle or non-motorcycle. This classification is adopted as it is sufficient for assessing whether an object is a motorcycle in the proposed system.

The second stage consists of the detection of helmet use. To reduce the computational cost and to increase the precision, a region of interest (RoI) was defined. The HOG descriptor was employed in this stage. The descriptor obtains different vectors for an image of a motorcyclist with a helmet and without a helmet. The extraction of features for the detection of helmet use is considered as a critical step in this study, as helmet detection is the main objective of the proposed system. The MLP classification algorithm was used to classify the images into two classes: with helmet or without helmet.

The diagram of the proposed system shows all of the stages and substages of the problem, as illustrated in Fig. 1. The diagram includes all stages from the capture of images to the detection of a helmet.

3.1 Vehicle segmentation and classification

The stage of vehicle segmentation and classification comprises three stages: detection of the background, segmentation of moving objects and vehicle classification. Figure 2 illustrates the main steps of vehicle segmentation.

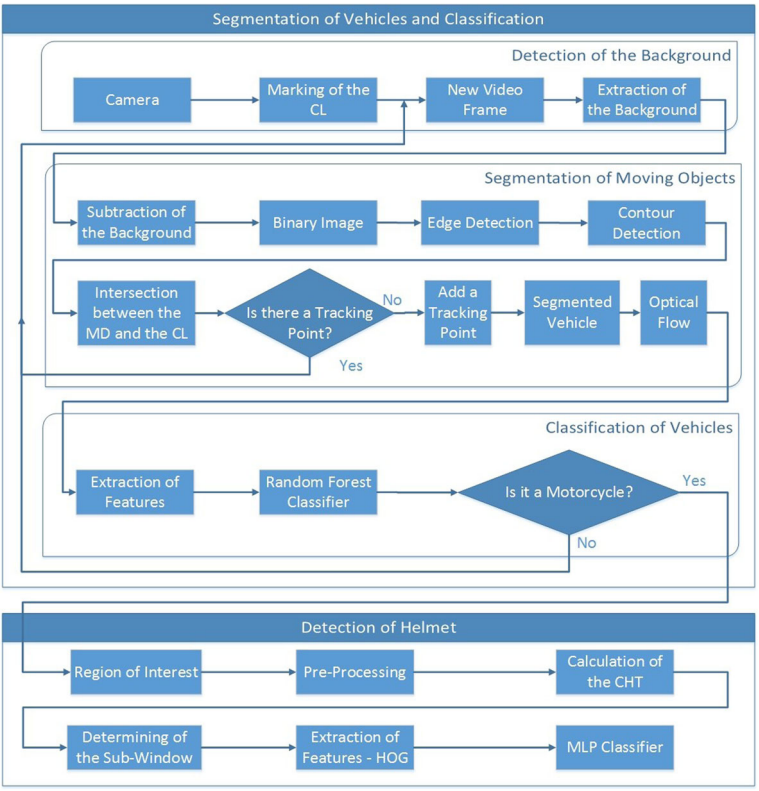


Fig. 1 Diagram of all steps of the proposed system, where MD is the main diagonal, CL is the cross line, WT is the wavelet transform, CHT is the circular Hough transform, HOG is the histogram of oriented gradients descriptor and MLP is the multilayer perceptron classifier

3.1.1 Detection of the background

The detection of the background is critical for the development of this study. The main objective of this step is to obtain an image that will be used to detect moving objects. The pixels that belong to the scene background and moving pixels can be extracted from an image of the background and the current frame. All moving objects are potential vehicles. The calculation of the background image was performed using the AGMM algorithm [36]. Figure 2a shows a background calculation using the AGMM algorithm.

Fig. 2 Vehicle Segmentation: (**a**) Current frame; (**b**) Background; (**c**) Subtraction of the background image from the current frame after a binarization operation; and (**d**) Example of segmented vehicle

This background varies with time. Changes in the road or weather conditions can cause changes in the background image. The most common factors of change are parked cars, changes of illumination, shadows that change position during the day and changes in lighting conditions, such as cloudy days and sunny days.

3.1.2 Segmentation of moving objects

The segmentation of moving objects on the scene facilitates the evaluation of the objects of interest in the image. The method of background subtraction receives all points that were altered in a scene. The subtraction between the current frame (Fig. 2a) and the background (Fig. 2b) after binarization is shown in Fig. 2c.

For the segmentation of moving objects in the proposed system, a CL (Fig. 3), which is marked by the user in the calibration stage of the system, must be defined. The CL is only defined once-when the algorithm begins operating. The CL should be marked approximately perpendicular to the road, where the vehicles circulate, which ensures that any vehicle that uses the road will cross the CL. When a vehicle crosses the CL, the process of segmentation of moving objects begins and the image frame is captured. Figure 2d shows an example of a segmented vehicle.

The next step involves the use of the Otsu threshold [24] for the image that results from the subtraction. This threshold is applied to obtain a binary image, as shown in Fig. 4a. The Sobel algorithm [27] is applied to the binary image for edge detection (Fig. 4b). A morphological closing operator is employed to remove some noises from the image, such as small regions that are not vehicles. The next step is the detection of the object contour. This detection is performed based on the image edges. The algorithm proposed by Suzuki and Be [29] was utilized in this stage. This process is shown in Fig. 4c and d. The maximum points of the contour are used to cut the captured vehicle from the frame and obtain the image of the vehicle. Figure 4e shows an example of a captured vehicle.

The vehicle will continue to cross the CL in other frames of the video. Therefore, a tracking algorithm that ensures that each vehicle is only counted once is needed. For each detected object (vehicle), the point of intersection between the main diagonal of the object (which is calculated from its shape) and the CL is computed. This point is designated as the tracking point of the object. To reduce the computational processing time, only the objects that are detected and not marked are analyzed, that is, after the object is marked, it will not be analyzed in subsequent frames.

Fig. 3 Example of the determination of the CL

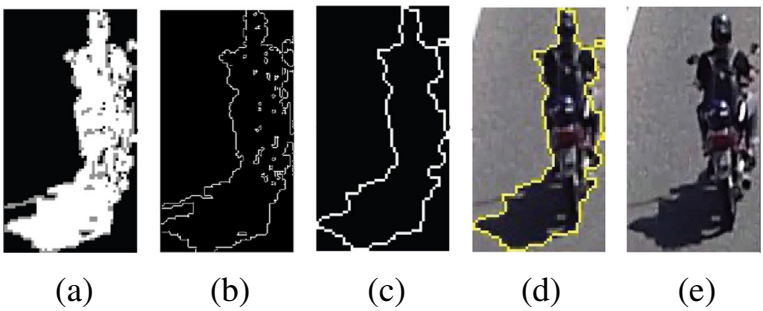


Fig. 4 Stages of vehicle segmentation: (**a**) Binary image computed from the Otsu algorithm; (**b**) Image edges calculated by the Sobel algorithm; (**c**) Detection of shape calculated by the algorithm of Suzuki and Be; (**d**) Shape of the object shown in the original image; and (**e**) Captured vehicle

The next step consists of calculating the optical flow of the detected object using the algorithm proposed by [3]. The optical flow is only calculated for the tracking points; its goal is to identify the same point on another frame. The counter is incremented with the detection of a new vehicle.

3.1.3 Vehicle classification

In the proposed system, the task of classifying the vehicles consists of differentiating the segmented objects into two classes: motorcycles and non-motorcycles.

The features of the images are extracted using the WT and the image that was segmented in the previous stage. The WT was performed using one decomposition level, to get this parameter we tested values between 1 and 10. A feature vector is obtained for each generated image. This vector is used by the random forest classifier to determine which class the vehicle is associated.

One of the requirements for a feature vector to be used in this problem is that it exhibits the same size for all images, regardless of their dimensions. To extract features with the WT, the images had to be resized as the number of features that is returned by the transform is sensitive to the image size. Therefore, the size of all image must be identical. The images were resized to obtain a single size for all images. The best result was obtained for images with 50×200 pixels.

After generating a feature vector for each image, the random forest classifier was employed. The classifier was employed with an unlimited maximum depth of trees. The random forest was tested with 10, 20, 40, 80 and 100 trees. The best result was obtained with 80 trees.

3.2 Detection of the helmet

The stage of detection of helmet use by motorcyclists was divided into three main stages: determining the RoI, extraction of the attributes and image classification.

3.2.1 Determining the RoI

Determining the RoI is an important step of the proposed system. The use of this region enables the reduction of the area in which the search will be performed, which implies less

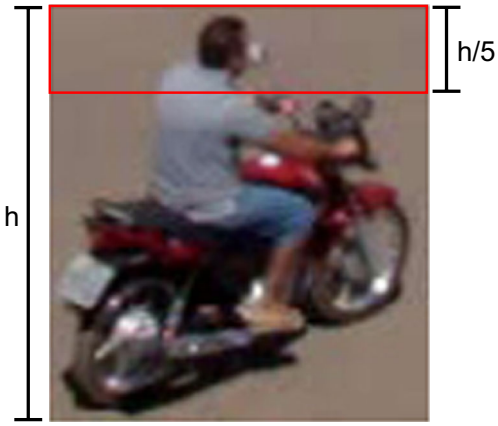


Fig. 5 Example of the RoI calculation. The region bounded by the rectangle corresponds to the RoI

processing time and a greater precision of the results compared with the complete image. The RoI is a region of the captured image in the vehicle segmentation stage. As the proposed system is interested in the detection of motorcyclists without helmets, the head region of the motorcyclist must be located completely inside the RoI.

The upper part of the image (1/5 of the image) was employed for the definition of the RoI, as shown in Fig. 5. This value was empirically selected via tests with the images obtained in the vehicle segmentation stage. The head region is typically located in the upper 1/5 of the image.

The size of the RoI was tested across the image database. In all images of motorcyclists, the head region is located within the selected RoI. Other sizes were tested but did not produce satisfactory results. Figure 6 shows some RoIs that were identified using the image database.

3.2.2 Extraction of features

Prior to calculating the descriptor, pre-processing is performed to obtain a sub-window that corresponds to the head region of the motorcyclist.

First, a grayscale image was calculated. Second, an average filter with a 5×5 neighborhood was applied to reduce the noise of the image. Third, the Otsu threshold is calculated.

Fig. 6 Examples of RoIs computed from images of motorcyclists that were captured in the object segmentation stage

This threshold is applied to the grayscale image to obtain a binary image (black and white). Figure 7c shows the results of this processing. The last step is the application of the Sobel algorithm to the binary image, which results in the image edges (one is more likely to obtain a circle using the image edges). The results of the application of the Sobel operator are shown in Fig. 7d.

After pre-processing, the CHT was employed to calculate the possible circles in the image. The CHT was utilized as the shapes of a helmet and the human head resemble a circle. The calculation of the CHT using images with edges is computationally faster compared with the grayscale images; it is also more precise, as only the pixels of the edges are used.

The calculation of the CHT was performed using a binary image with the image edges. The results of this processing are the possible circles in the image. In the RoI, circular shapes correspond to the head region or the helmet of the motorcyclist. One of the necessary parameters for the CHT is the radii of the desired circles. In this study, the radius was 50 % of the height of the RoI. This size was selected as a helmet will not be larger than the RoI given that it is completely inside of the RoI.

A search for the circle with the largest CHT accumulator (circle with the most points) is performed based on the obtained circles. Figure 7e shows the best circle in the RoI.

A strategy that uses only geometric information, such as the use of the CHT, will not return acceptable results, as the shape of the helmet resembles the shape of the head. Thus, more information is necessary to distinguish heads from helmets.

A sub-window of the RoI is computed to reduce the detection area of the helmet. The use of a sub-window enables the calculated descriptors to better detail the helmet region, as only that region will be processed. A sub-window will correspond to the square that circumscribes the obtained circumference (Fig. 7f). This sub-window will be employed in the extraction of features. After the sub-window is computed, the HOG descriptor is calculated. Figure 7 shows the steps for the calculation of the sub-window.

A hybrid descriptor that combines the CHT and the HOG descriptors was employed for the extraction of features. The hybrid approach incorporates different information from more than one descriptor.

After the sub-window computation, the HOG descriptor is calculated. Figure 8 shows the steps for the calculation of the sub-window.

3.2.3 Classification of sub-windows

The MLP classifier was employed in this stage. The image classification task consists of differentiating the segmented objects into two classes: with helmet and without helmet.

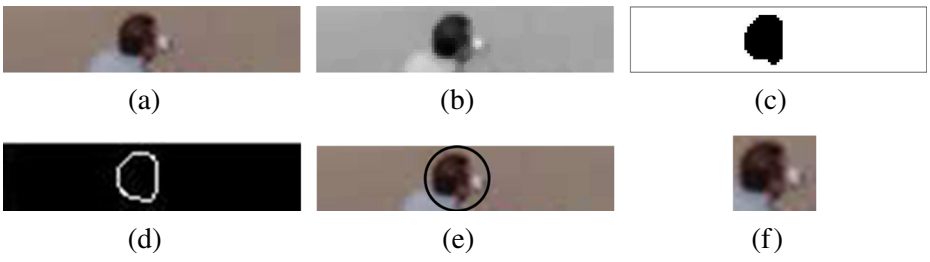


Fig. 7 Steps for the detection of motorcyclists without helmets: (**a**) RoI; (**b**) Grayscale image; (**c**) Binary image; (**d**) Calculated edges; (**e**) Best circle obtained by the CHT and drawn on the RoI; and (**f**) Sub-window

Fig. 8 Examples of computed sub-windows

A feature vector is obtained for each generated sub-window. Figure 8 shows examples of the sub-windows that are used in the image classification stage.

The HOG descriptor was employed for the extraction of features, which were arranged in nine blocks; each block was partitioned into nine cells. Each cell generated a feature. Thus, a vector was generated with 81 features. An extensive variation of blocks and cells was used. Based on the obtained results, the selected combination was determined to be the best selection.

The MLP classifier was utilized with a hidden layer. Other values of hidden layers and a model without this layer were tested. The results obtained with the remaining configurations were not better than the results obtained with one layer. Another parameter that is very sensitive in the MLP classifier is the number of neurons. In this study, 50 neurons were employed. Although other values were also tested, this value returned the best results.

4 Results

In this section, the results are presented and discussed. In addition, a comparative analysis is performed with other algorithms to describe and classify the images. The results are divided into two groups:

– results of the vehicle classifier, and
– results of the helmet detector.

Information about the image databases, generated from the segmentation of vehicles and the methodology employed for the classification of the results, are also presented in this section.

4.1 Image database

The image databases used for the tests of the algorithms were obtained from the application of vehicle segmentation in videos captured on public roads. Two image databases were obtained: *database1* and *database2*.[1] The first database was employed in the classification

[1]The batabases are available for public at http://github.com/romuere/databases/.

of the vehicles, and the second database was employed in the detection of the helmets. The videos were obtained from a charge-coupled device (CCD) video camera that was installed on public roads. The videos were recorded during the day and at night, and under different lighting conditions (cloudy and sunny days). The videos have a resolution of 1280 × 720 pixels and 30 frames/s. The total duration of all videos is 150 minutes.

All results were performed on a machine with AMD Phenom II processor at 2.8 GHz with 8 GB RAM. Regarding to the computational time of the vehicle segmentation, we were capable of executing this task in real-time using 15 frames/s (half of the original video). The generated databases showed that this frame ratio is suitable for the proposed system.

Database1 was obtained from 110 minutes of video and is composed of a total of 3,245 images, which are divided as follows:

– 2,576 images of non-motorcycles, and
– 669 images of motorcycles.

The images of *database1* were employed in the vehicle classification stage, as these images were captured from a location farther from the road compared with *database2*. Therefore, the quality of the images after segmentation were not sufficient to be used in the detection of helmet use. In some images, helmet use was not distinct. Figure 9 shows examples of images from *database1*.

Database2 was obtained from 40 minutes of video. These videos were captured at a location in which the number of motorcyclists without helmets was balanced relative to the number of motorcyclists with helmets. The images were captured at a shorter distance from the road compared with *database1*. Figure 10 shows examples of images from *database2*. *Database2* is distributed as follows:

– 151 images of motorcyclists with helmets, and
– 104 images of motorcyclists without helmets.

Fig. 9 Example of images from *database1*

Fig. 10 Example of images from *database2*

4.2 Methods of evaluation of results

Some known metrics from the literature were used to evaluate the performance of the classification algorithms. This section shows the results of these evaluation metrics.

The confusion matrix is a table that shows the classification results. The matrix is composed of four values: true positive (TP), false positive (FP), false negative (FN) and true negative (TN).

From these rates, the values of specificity (S), negative predictive value (NPV), precision (P), recall (R), accuracy (A), F-Measure (FM) and Kappa coefficient (K) are calculated [25].

As shown in Table 1, the level of accuracy of the Kappa coefficient was classified according to the level of accuracy established by Landis and Kock [14].

Another result evaluation method involves Receiver Operating Characteristic (ROC) curves. The ROC curves show the true positive rates versus the false positive rates.

The curve is constructed by varying the threshold of the classifier and observing the results that are generated from the modification. The ROC curve can be used to evaluate the efficiency in terms of hit rate of the machine learning algorithms. The main information that is extracted from a ROC curve is the area under the curve (AUC); the larger the area is, the better the classifier.

Table 1 Level of accuracy of a classification according to the Kappa coefficient value

Kappa Coefficient (K)	Quality
$K < 0.2$	Poor
$0.2 \geq K < 0.4$	Reasonable
$0.4 \geq K < 0.6$	Good
$0.6 \geq K < 0.8$	Very Good
$K \geq 0.8$	Excellent

4.3 Vehicle classification

The statistical method k-fold cross-validation ($k = 10$) was employed to generate the results in the classification stage. In the 10-fold cross-validation, the set of original data is randomly partitioned in 10 data subsets with the same size. From the ten subsets, nine subsets are used as training data, and only one subset is selected as the validation set to test the model. The cross-validation process is repeated ten times. Each of the ten subsets is only used once to validate the data. The mean of the ten generated results is computed to produce a single estimate. The advantage of this method is that all subsets are employed for testing and training, and each subset is only used once for testing.

The proposed system used the Wavelet transform as feature descriptor and the random forest as classifier. An accuracy of 0.9778 was obtained, that is, of the 3,245 vehicles, only 72 were misclassified. The Kappa coefficient classified the results as "Excellent" according to Table 1, which shows the level of accuracy. The FM also returned a satisfactory result (0.9754), which reflects the P and R rates, as the FM is calculated based on these rates. The values of S (0.9930) and NPV (0.9793) reflect that the proportion of true negatives was satisfactory.

4.3.1 Comparative analysis

In addition to the results of the proposed system, which were previously presented, a comparative analysis was performed using other descriptors and classifiers. This analysis was performed to justify the selection of the algorithms.

In addition to the WT and the random forest classifier, the HOG, local binary pattern (LBP) [23] and Speeded Up Robust Features (SURF) [31] descriptors and the SVM, radial basis function neural network (RBFN), MLP and naive Bayes [19] classifiers were also tested.

To obtain the best set of parameters for each descriptor we made a parameter estimation for each one. The HOG descriptor have two parameters: number of blocks and number of cells. It was employed with nine blocks; each block was partitioned in nine cells. To obtain these values we used an interval of 1 to 30 for both parameters. For the LBP descriptor, the image was divided into nine windows. The neighborhood for the computation of the patterns was 3×3. Each window corresponded to a histogram of the computed values, that is, nine histograms were obtained at the end. We test values of window and neighborhood in the interval of 2 to 10. For the SURF descriptor, the concept of visual dictionaries was employed [2] and 10 % of the database was utilized to create the dictionary. Although other values were tested, the selected value returned the best results.

The SVM was performed using a linear kernel function. The results using this kernel outperformed in relation to polynomial, radial basis and sigmoid tangent kernels. The data

Table 2 Results of the SVM classifier for vehicle classification

Descritor	S	NPV	P	R	FM	K	A
TW	0.9565	0.9466	0.8255	0.7922	0.8819	0.7601	0.9226
HOG	0.9752	0.9602	0.8983	0.8445	0.9282	0.8383	0.9482
LBP	0.9864	0.9837	0.9471	0.9372	0.9651	0.9272	0.9763
SURF	0.9860	0.9788	0.9446	0.9178	0.9614	0.9134	0.9719

of the RBFN classifier was normalized prior to training. The MLP classifier was used with a hidden layer of 50 neurons. Other values in 2 to 100 interval were also tested, but this value returned the best results. The naive Bayes classifier was applied using class estimators.

Table 2 shows the results obtained for the SVM classifier. The classifier obtained a maximum hit rate of 0.9763 using the LBP descriptor. The results were similar to the results of the proposed system, which was 0.9778. Similar results were obtained for the Kappa coefficient, which also obtained "Excellent" results (0.9272).

The results for the RBFN classifier are shown in Table 3. Despite not achieving the best results, this classifier achieved minimal variation between the hit rates, that is, the classifications yielded similar results regarding the accuracy of the different descriptors. The best result for this classifier was 0.9676 with the SURF descriptor, and the worst result was 0.9510 with the HOG descriptor.

The results using the MLP algorithm are shown in Table 4. The LBP descriptor obtained the lowest hit rate for the classification of vehicles (0.7938). All vehicles were classified as non-motorcycle, which indicates that the MLP classifier was not able to learn from the data generated by the LBP classifier. This table shows the importance of the remaining measures for the evaluation of the results in addition to the accuracy. Although the obtained rate was almost eighty percent, the classifier did not differentiate between the classes. The Kappa coefficient classified the result as "Bad".

The results of the naive Bayes classifier are shown in Table 5. This classifier obtained an accuracy of 0.9713 with the SURF descriptor and a specificity of 0.9946; this rate indicates that almost all objects of the non-motorcycle class were correctly classified.

Table 6 shows the results of Random Forest classifier. This classifier obtained an accuracy of 0.9778 with the WT descriptor. This accuracy value is the highest value obtained among all descriptor/classifier pairs evaluated (Tables 2–6). Therefore, we propose that the vehicle classification step of our system be carried out by random forest classifier using the wavelet transform.

In addition to the evaluation metrics, ROC curves were generated for each combination of descriptor and classifier to evaluate the efficiency of each performed classification. Figure 11 shows bests ROC curves (greatest areas) generated in the classification. The largest area

Table 3 Results of the RBFN classifier for vehicle classification

Descritor	S	NPV	P	R	FM	K	A
TW	0.9786	0.9741	0.9163	0.8998	0.9443	0.8844	0.9624
HOG	0.9639	0.9741	0.8664	0.9013	0.9171	0.8525	0.9510
LBP	0.9616	0.9845	0.8641	0.9417	0.9204	0.8742	0.9575
SURF	0.9825	0.9768	0.9312	0.9103	0.9535	0.9003	0.9676

Table 4 Results of the MLP classifier for vehicle classification

Descritor	S	NPV	P	R	FM	K	A
TW	0.9732	0.9698	0.8954	0.8834	0.9312	0.8609	0.9547
HOG	0.9798	0.9844	0.9236	0.9402	0.9531	0.9140	0.9716
LBP	1	0.7938	–	0	–	0	0.7938
SURF	0.9717	0.9889	0.8978	0.9581	0.9411	0.9072	0.9689

of all vehicle classifications is obtained by the random forest classifier using the SURF descriptor, with an area of 0.9973.

4.4 Detection of helmet use

The results obtained for the detection of helmet use are discussed in this section. In this stage, the images from *database2* were employed. The same metrics used for vehicle classification were also used here to evaluate the obtained results. The proposed method calculates a sub-window of the RoI, and the HOG descriptor is used to extract the features. Once the features are extracted, the classification is performed with the MLP network. The obtained result was an accuracy of 0.9137. From a total of 255 images, 22 images were incorrectly classified. Similar to the vehicle classification stage, the Kappa coefficient classified the result as "Excellent" according to the table that lists the accuracy levels ($K \geq 0.8$). The FM value was 0.9281, which reflects the P (0.9161) and R (0.9404) rates.

Figure 12 shows the ROC curve that was generated for the classifier. The AUC was 0.9556.

4.4.1 Comparative analysis

In addition to the results of the proposed system, a comparative analysis was performed using other descriptors and classifiers.

The descriptors tested here were applied to the sub-window in the pre-processing stage. The following descriptors and combinations were tested: WT, HOG, LBP, WT+LBP, WT+HOG, HOG+LBP and WT+HOG+LBP. The hybrid forms of the descriptors were assembled by combining the feature vectors. The descriptors were applied with the same parameters of the vehicle classification stage.

The combinations (hybrid descriptors) combine features of more than one descriptor in a single vector and strengthen the description of the objects in the scene. One of the requirements for a descriptor to be used is that the feature vector calculated for each image has the same size, regardless of the image size.

Table 5 Results of the Naive Bayes classifier for vehicle classification

Descritor	S	NPV	P	R	FM	K	A
TW	0.8323	0.9614	0.5744	0.8714	0.7191	0.5907	0.8404
HOG	0.8171	0.9850	0.5749	0.9522	0.7261	0.6190	0.8450
LBP	0.9433	0.9798	0.8091	0.9253	0.8864	0.8248	0.9396
SURF	0.9946	0.9701	0.9768	0.8819	0.9734	0.9092	0.9713

Table 6 Results of the Random Forest classifier for vehicle classification

Descritor	S	NPV	P	R	FM	K	A
TW	0.9930	0.9793	0.9716	0.9193	0.9754	0.9308	**0.9778**
HOG	0.9724	0.9793	0.8966	0.9208	0.9361	0.8844	0.9618
LBP	0.9515	0.9943	0.8397	0.9791	0.9105	0.8767	0.9572
SURF	0.9992	0.9636	0.9965	0.8550	0.9798	0.9016	0.9695

The bold means the best result

The SURF descriptor was not employed at this stage of the system as it did not return points of interest in some images.

All classifiers, namely SVM, RBFN, MLP, naive Bayes, random forest and KNN, were tested, and the calculated descriptors were employed as their inputs. A total of 42 combinations of descriptors and classifiers (seven descriptors and six classifiers) were generated. To classify the images, the same strategy used for the vehicles classification was also employed here (10-fold cross-validation). The classifiers were used with the same parameters of the vehicle classification stage. The KNN classifier with 5 neighbors was added to the tests of the detection of helmet use, in this classifier we tested a number of neighbors between 1 and 20.

Table 7 shows the results obtained for the SVM classifier. The best results were obtained using the hybrid combination of the HOG and LBP descriptors (0.8784).

Table 8 shows the results generated from the RBFN classifier. The hybrid combination WT+HOG+LBP obtained the best hit rate (0.8863) for this classifier, which shows that the hybrid combinations can improve the results of the individual descriptors in some cases.

The results of Table 9 present the results of the proposed method (MLP + HOG) in addition to the results of the other combinations of descriptors. Only the Kappa coefficient, which was classified as "Excellent", was associated with the proposed system.

The results of the naive Bayes classifier are presented in Table 10. Note that this classifier obtained the worst classification according to the Kappa coefficient and was classified as "Bad".

Table 11 shows the results of the random forest classifier, which obtained an accuracy of 0.8980 as the best classification. This classification was the second best classification among all results, which only ranked below the proposed system.

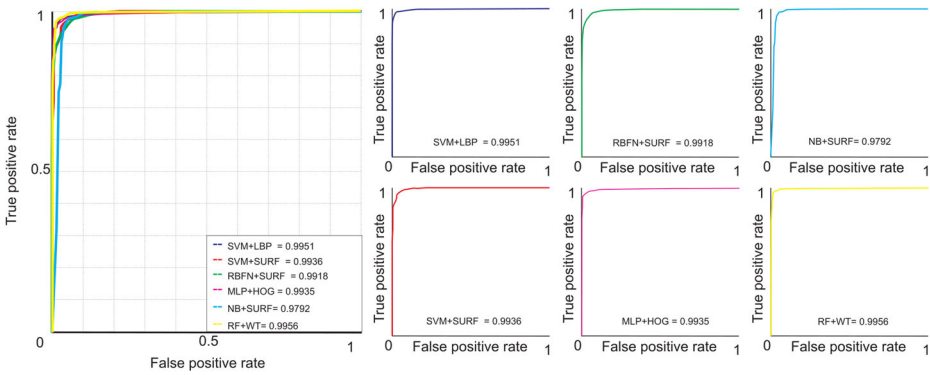


Fig. 11 Bests ROC curve for the vehicle classification, left aggregate and right individual, respectively

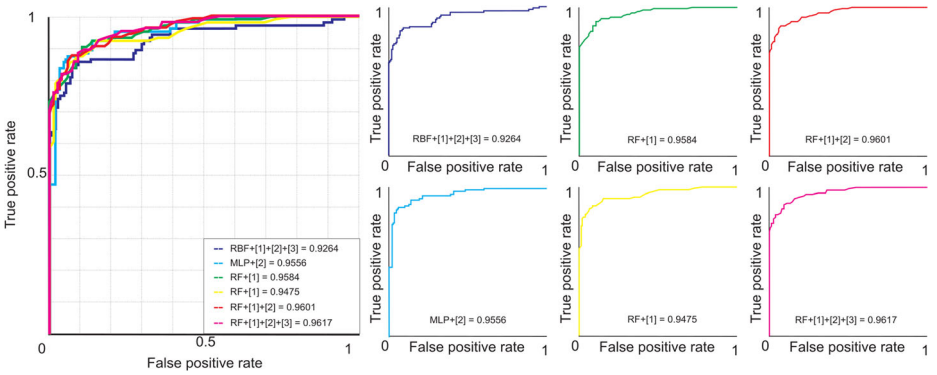


Fig. 12 Bests ROC curves in detection of helmet use, left aggregate and right individual, respectively

The results obtained with the KNN classifier are shown in Table 12.

Figure 12 shows the bests ROC curves and their areas. The largest area of 0.9584 was obtained for the random forest classifier. The curve of the proposed method was obtained from an area under the curve of 0.9558.

5 Discussion

5.1 Analysis of vehicles classification results

The results presented in Tables 2 to 6 demonstrate that the random forest classifier using the wavelet transform obtained the best performance in vehicle classification. Generally, we can say that the random forest classifier yielded the best mean accuracy for the classification, that is, the sum of the accuracies divided by the number of descriptors. The random forest is not a single classifier but a set of classifiers. As previously discussed, the classifier is formed by a set of decision trees. At the end of the classification, each tree returns a result. The final result is the combination of the results of the various decision trees. This same classifier obtained the largest area under the ROC curve (0.9973), which demonstrates its robustness.

Regarding the Kappa coefficient, the majority of the results were "Excellent" ($K \geqslant 0.8$) according to the accuracy table of the coefficients. The WT with the SVM classifier obtained

Table 7 Results of the detection of helmet use for the SVM classifier

Descritor	S	NPV	P	R	FM	K	A
TW-[1]	0.8173	0.8586	0.8782	0.9073	0.8925	0.7301	0.8706
HOG-[2]	0.81731	0.8763	0.8797	0.9205	0.8997	0.7456	0.8784
LBP-[3]	0.7115	0.8043	0.8159	0.8808	0.8471	0.6032	0.8118
[1]+[3]	0.8173	0.8586	0.8782	0.9073	0.8925	0.7301	0.8706
[1]+[2]	0.8173	0.8586	0.8782	0.9073	0.8925	0.7301	0.8706
[2]+[3]	0.81731	0.8763	0.8797	0.9205	0.8997	0.7456	0.8784
[1]+[2]+[3]	0.8173	0.8586	0.8782	0.9073	0.8925	0.7301	0.8706

Table 8 Results of the detection of helmet use for the RBFN classifier

Descritor	S	NPV	P	R	FM	K	A
TW-[1]	0.8269	0.8113	0.8792	0.8675	0.8733	0.6924	0.8510
HOG-[2]	0.7885	0.8200	0.8581	0.8808	0.8693	0.6733	0.8431
LBP[3]	0.7692	0.6504	0.8182	0.7152	0.7633	0.4711	0.7373
[1]+[3]	0.7500	0.8041	0.8354	0.8742	0.8544	0.6308	0.8235
[1]+[2]	0.7885	0.8367	0.8599	0.8940	0.8766	0.6887	0.8510
[2]+[3]	0.8365	0.6797	0.8661	0.7285	0.7914	0.5454	0.7725
[1]+[2]+[3]	0.8462	0.8713	0.8961	0.9139	0.9049	0.7635	0.8863

"Very Good" results. The worst coefficients were obtained by the following combinations: naive bayes with WT, which obtained "Good" results, and MLP with the LBP descriptor, which obtained "Poor" results, as its coefficient was less than 0.2. The analysis of the results obtained by the coefficient shows that it was adequate for the evaluation of the results, as it has the capacity to completely represent the confusion matrix, which is even observed in cases of classes with unbalanced numbers of elements. This finding is better observed with the classification results of the MLP classifier and the LBP descriptor. In this case, the classifier was not able to separate the two classes and classified all objects as "non-motorcycle". Although an accuracy of 0.7938 was attained, the result of the coefficient was 0, which demonstrates that this classifier could not have been used in this problem.

As previously stated, the FM will produce acceptable results only when the P and R rates are balanced, that is, if P is a satisfactory rate and R is not a satisfactory rate, the result of the FM will not be satisfactory. The classification results showed that the FM rates were satisfactory; a rate of 0.9798 was achieved with the use of the LBP descriptor and the random forest classifier. The FM of the proposed system was 0.9754, which is very close to the best results. The worst FM rates were obtained with the naive Bayes classifier. Rates of 0.7191 and 0.7261 were obtained with the WT and the HOG descriptor, respectively. No results were obtained for the FM, the MLP classifier and the LBP descriptor, as a value cannot be divided by 0 (Equation 7).

The proposed method obtained the best result regarding the accuracy (0.9778) and the Kappa coefficient (0.9308). In the remaining evaluation metrics, the proposed method obtained satisfactory rates that always exceeded 0.9.

Table 9 Results of the detection of helmet use for the MLP classifier

Descritor	S	NPV	P	R	FM	K	A
TW-[1]	0.8558	0.8091	0.8966	0.8609	0.8784	0.7103	0.8588
HOG-[2]	0.8750	0.9100	0.9161	0.9404	0.9281	0.8203	0.9137
LBP[3]	0.6923	0.6207	0.7698	0.7086	0.7379	0.3938	0.7020
[1]+[3]	0.8462	0.8627	0.8954	0.9073	0.9013	0.7557	0.8824
[1]+[2]	0.8462	0.8073	0.8904	0.8609	0.8754	0.7018	0.8549
[2]+[3]	0.7981	0.7905	0.8600	0.8543	0.8571	0.6514	0.8314
[1]+[2]+[3]	0.8558	0.8476	0.9000	0.8940	0.8970	0.7487	0.8784

Table 10 Results of the detection of helmet use for the Naive Bayes classifier

Descritor	S	NPV	P	R	FM	K	A
TW-[1]	0.8269	0.7963	0.8776	0.8543	0.8658	0.6772	0.8431
HOG-[2]	0.7981	0.8737	0.8688	0.9205	0.8939	0.7284	0.8706
LBP-[3]	0.1827	0.5429	0.6136	0.8940	0.7278	0.0856	0.6039
[1]+[3]	0.8269	0.8037	0.8784	0.8609	0.8696	0.6848	0.8471
[1]+[2]	0.8269	0.8037	0.8784	0.8609	0.8696	0.6848	0.8471
[2]+[3]	0.5096	0.8281	0.7330	0.9272	0.8187	0.4646	0.7569
[1]+[2]+[3]	0.8077	0.8077	0.8675	0.8675	0.8675	0.6752	0.8431

5.2 Analysis of detection of helmet use results

The analysis of the results shows that the best results were obtained using the HOG descriptor and the MLP classifier. The accuracy of 0.9137 and the Kappa coefficient classified the solution as "Excellent". This result was the only result with the maximum classification regarding the coefficient. The FM was 0.9281; this value was also the highest among all obtained results. Regarding the remaining hit rates, the proposed method did not obtain the best rates but obtained values similar to the best results. The efficacy of HOG descriptor is explained by the difference of textures between heads and helmets. In order to visualize the class separability of this descriptor, we compute the Principal Component Analysis (PCA) and plot its 3 first components as we can see in Fig. 13.

The random forest classifier was shown to be a good classification algorithm, as this classifier obtained the best hit rates for the majority of the results. As previously stated, this classifier operates like a system of multiple classifiers; for this reason, it frequently yields better results. This finding can be proven based on the areas of the ROC curves; the largest area of 0.9617 was attained with the combination WT+HOG+LBP. The KNN classifier had the lowest hit rate with regards to accuracy (0.5569) using the LBP descriptor. The Kappa coefficient classified the results as "Bad". The worst result regarding the Kappa coefficient was 0.0856, which was obtained by the naive Bayes classifier with the LBP descriptor.

The results with the lowest accuracy rate were obtained using the LBP descriptor (0.6039 and 0.5569). The results obtained with the LBP descriptor conclude that descriptors that use

Table 11 Results of the detection of helmet use for the Random Forest classifier

Descritor	S	NPV	P	R	FM	K	A
TW-[1]	0.8173	0.9043	0.8820	0.9404	0.9103	0.7692	0.8920
HOG-[2]	0.8173	0.9239	0.8834	0.9536	0.9172	0.7850	0.8980
LBP-[3]	0.6827	0.8554	0.8081	0.9205	0.8607	0.6228	0.8235
[1]+[3]	0.8173	0.8947	0.8812	0.9338	0.9068	0.7613	0.8863
[1]+[2]	0.8173	0.9239	0.8834	0.9536	0.9172	0.7850	0.8980
[2]+[3]	0.7788	0.9205	0.8623	0.9536	0.9057	0.7505	0.88234
[1]+[2]+[3]	0.8173	0.9239	0.8834	0.9536	0.9172	0.7850	0.8980

Table 12 Results of the detection of helmet use for the KNN classifier

Descritor	S	NPV	P	R	FM	K	A
TW-[1]	0.9327	0.7132	0.9412	0.7417	0.8296	0.6436	0.8196
HOG-[2]	0.9327	0.7519	0.9444	0.7881	0.8592	0.6948	0.8471
LBP-[3]	0.75	0.4727	0.7111	0.4238	0.5311	0.1593	0.5569
[1]+[3]	0.9327	0.7519	0.9444	0.7881	0.8592	0.6948	0.8471
[1]+[2]	0.9327	0.7348	0.9431	0.7682	0.8467	0.6727	0.8353
[2]+[3]	0.8654	0.7258	0.8931	0.7748	0.8298	0.6216	0.8118
[1]+[2]+[3]	0.9327	0.7638	0.9453	0.8013	0.8674	0.7096	0.8549

only texture information are not adequate for the extraction of features in certain pattern recognition problems.

5.3 Final considerations

Some difficulties were encountered during the assembly of the image database based on vehicle segmentation. In the preliminary tests, several images were generated when a cloud passed over the CL due to the low pre-determined value for the threshold of the background detector. This problem was solved by increasing the threshold of the background detector.

Another difficulty was the presence of trees in the scenario. When the CL was marked close to a large tree, several images were generated. This situation occurred every time the wind speed increased to the point that the leaves moved. This problem was analyzed by disregarding small movements. A morphological closing operator was utilized, which caused small regions to be excluded from the image.

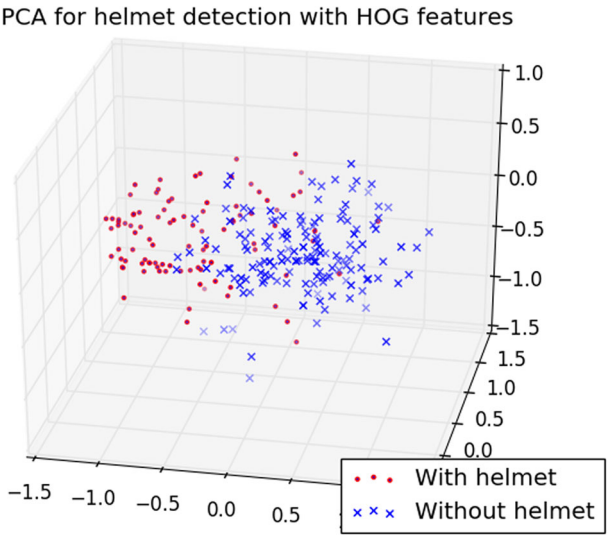


Fig. 13 Principal Component Analysis using HOG descriptor for helmet detection

The helmet detection stage can be improved. Images with better quality are necessary to conduct the processing.

The main shortcoming of this study is the detection of helmet use when both the driver and a passenger ride a motorcycle. This stage of the problem was not analyzed. To solve this problem, some processing is required to calculate the number of passengers on the vehicle.

Regarding the studies related to helmet detection that are cited, this study proved to be more robust by analyzing all stages of the problem. The comparison of this study with the proposal presented by Chiverton [7] reveals that many more images are used in this study. The higher number of images helps to validate the performed tests and the analysis of the results. The study by Chiverton attains 85 % of the total hits in the stage of the detection of helmet use.

The study by Waranusast et al. [33] presents a method to detect more than one person on a motorcycle. A limitation of this study is that the mode of capture of the images is inappropriate for the detection of the registration plate of the vehicle, which renders the system pointless. Although the method proposed in this study does not detect more than one person on a motorcycle, it produces better hit rates.

6 Conclusions and additional studies

This study addresses the detection of motorcyclists without helmets on public roads. A computational vision system was proposed and classified as follows:

– vehicle segmentation and classification, and
– detection of helmet use.

In the stage of vehicle segmentation and classification, algorithms for the background calculation and tracking of objects, descriptors and classifiers that exhibited reasonable hit rates and low processing times were selected from the literature achieving an accuracy of 0.9778.

In the stage of detection of helmet use, algorithms for the extraction of features in images and classification algorithms were employed. The proposed system also obtained satisfactory hit rates. The MLP classifier that incorporated the HOG descriptor obtained the best results, with an accuracy of 0.9137.

6.1 Future studies

The results are promising but can be improved. An important step for improving the results is the stage of image capturing, which should produce better quality images. The images of *database1* were not employed in the stage of helmet detection due to their low quality. Future studies should focus on the detection and recognition of the registration plate of the vehicle. A better quality image is necessary to recognize the characters on the plate.

Hybrid descriptors were not employed in the vehicle segmentation stage. They will be employed to improve the results.

The use of algorithms for feature selection can be evaluated to increase the obtained hit rates. The descriptors returned a large number of features, which frequently hinders the classification of objects. In these feature vectors, the possibility of features that are

insignificant or duplicates is high. Therefore, a thorough analysis of the literature in search of feature selection algorithms is necessary.

Another future study is the detection of passengers on motorcycles. A motorcycle has the capacity to carry a driver and a passenger. The proposed system does not detect more than one helmet in the scene, that is, it does not detect a motorcyclist with a passenger. This functionality can be extended to the detection of more than one passenger, as two or more passengers constitutes a traffic violation.

References

1. Adelson EH, Bergen JR (1985) Spatiotemporal energy models for the perception of motion. J Opt Soc Am A 2(2):284–299
2. Agurto C, Murray V, Barriga E, Murillo S, Pattichis M, Davis H, Russel S, Abramoff M, Soliz P (2010) Multiscale am-fm methods for diabetic retinopathy lesion detection. IEEE Trans Med Imag 29:502–512
3. Bouguet J-Y (2000) Pyramidal implementation of the lucas kanade feature tracker. Intel Corporation, Microprocessor Research Labs
4. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
5. Chen Z, Ellis T, Velastin SA (2012) Vehicle detection, tracking and classification in urban traffic. In: 15th International IEEE conference on intelligent transportation systems, pp 951–956
6. Chiu C-C, Min-Yu K, Chen H-T (2007) Motorcycle detection and tracking system with occlusion segmentation. In: Eighth International workshop on image analysis for multimedia interactive services. USA
7. Chiverton J (2012) Helmet presence classification with motorcycle detection and tracking. Intell Transp Syst 6(3):259–269
8. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
9. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE Computer society conference on computer vision and pattern recognition, pp 886–893
10. Daubechies I (1992) Ten lectures on wavelets. Society for industrial and applied mathematics, 1 edn
11. DENATRAN (2013) Frota por tipo de veículo. Technical report, Departamento Nacional de Infraestrutura de Transportes, Ministério dos Transportes. http://www.denatran.gov.br/frota2013.htm
12. DNIT (2011) Número de veículos envolvidos por finalidade do veículo. Technical report, Departamento Nacional de Infraestrutura de Transportes, Ministério dos Transportes
13. Duda RO, Hart PE (1972) Use of the hough transformation to detect lines and curves in pictures. Commun Assoc Comput Mach 15(1):11–15
14. Landis J, Koch G (1977) The measurement of observer agreement for categorical data. Biometrics 33(1):159–174
15. Leelasantitham A, Wongseree W (2008) Detection and classification of moving thai vehicles based on traffic engineering knowledge. In: International conference on intelligent transportation system telecommunications, pp 439–442
16. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th international joint conference on artificial intelligence. Morgan Kaufmann Publishers Inc, San Francisco, pp 674–679
17. Ma Z, Cunzhi P, Ke H, Qiandong C (2009) Research on segmentation of overlapped vehicles based on feature points on contour. In: International conference on future biomedical information engineering, pp 552–555
18. Mallat SG (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans Pattern Anal Mach Intell 11(7):674–693
19. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York
20. Messelodi S, Modena C, Zanin M (2005) A computer vision system for the detection and classification of vehicles at urban road intersections. Pattern Anal Appl 8:17–31
21. MMWR (2012) Helmet use among motorcyclists who died in crashes and economic cost savings associated with state motorcycle helmet laws - united states, 2008-2010. Technical Report 23, Modidity and Mortality Weekly Report - MMWR

22. NHTSA (2011) National highway traffic safety administration. Traffic safety facts: research note motorcycle helmet use in 2011 - overall results
23. Ojala T, Pietikinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recog 26(1):51–59
24. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
25. Powers DMW (2007) Evaluation: from Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Technical Report SIE-07-001, School of Informatics and Engineering, Flinders University. Adelaide
26. Statistical Forecasting Bureau (2012) Key statistics of thailand. Technical report, National Statistical Office, Ministry of Information and Communication Technology. Bangkok
27. Sobel IE (1970) Camara models and machine perception. Ph.d dissertation, Stanford University, Palo Alto
28. Sonoda S, Tan JK, Kim H, Ishikawa S, Morie T (2011) Moving objects detection at an intersection by sequential background extraction. In: International conference on control, automation and systems (ICCAS), pp 1752 –1755
29. Suzuki S, Be K (1985) Topological structural analysis of digitized binary images by border following. Comput Vis Graph Image Process 30(1):32–46
30. Takahashi K, Kuriya Y, Morie T (2010) Bicycle detection using pedaling movement by spatiotemporal gabor filtering. In: IEEE Region 10 conference 2010 (TENCON 2010), pp 918–922
31. Tuytelaars T, Gool LV, Bay H, Less A (2008) Speeded-up robust features (surf). In: Computer vision image understand, pp 346–359
32. Waiselfisz JJ, da violência M (2013) Acidentes de trânsito e motocicleta. Technical report, Centro Brasileiro de Estudos Latino-Americanos
33. Waranusast R, Bundon N, Timtong V, Tangnoi C, Pattanathaburt P (2013) Machine vision techniques for motorcycle safety helmet detection. In: 2013 28th International conference of image and vision computing New Zealand (IVCNZ), pp 35–40
34. Wen C-Y, Chiu S-H, Liaw J-J, Chuan-Pin L (2003) The safety helmet detection for atm's surveillance system via the modified hough transform. In: IEEE 37th Annual international carnahan conference on security technology, pp 364–369
35. World Health Organization (2013) Global status report on road safety 2013: supporting a decade of action. Technical report, World Health Organization. Geneva
36. Zivkovic Z (2004) Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th international conference on pattern recognition, vol 2, pp 28–31

Romuere R. V. e Silva received the B.S. (2011) and M.Sc (2014) degrees in Computer Science from Federal University of Piauí. Currently, he is Professor at Federal University of Piauí and Ph.D student in Teleinformatics Engineering at Federal University of Ceará.

Kelson R. T. Aires received the B.S. (1999) degree in Electric Engineering, M.Sc and Ph.D (2001, 2009) degrees in Computer Engineering at Federal University of Rio Grande do Norte. Currently, he is Professor at Federal University of Piauí.

Rodrigo de M. S. Veras received the B.S. (2005) degree in Computer Science at Federal University of Piauí and M.Sc (2007) degree in Computer Science at Federal University of Ceará. He obtained his Ph.D (2014) degree in Teleinformatics Engineering at Federal University of Ceará. Currently, he is Professor at Federal University of Piauí.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 12 | Dec 2020 www.irjet.net p-ISSN: 2395-0072

Helmet Detection using ML & IoT

M. V. D. Prasad[1], S.V.N.P VAMSI KRISHNA[2], M.SANTOSH KUMAR[3], P. Sri HARSHA[4]

[1]Assoc.Professor, Department of ECE, KLEF-Deemed to be University, Guntur, A.P, India.
[2,3,4]B.tech Student, Department of ECE, KLEF-Deemed to be University, Guntur, A.P, India.

Abstract— This paper is about detecting motorbike riders without a helmet with the assistance of machine learning and IoT. Motorcycle accidents are increasing day by day in many countries. The helmet is that the primary safety equipment of Bike riders, however many drivers don't use it. The primary objective of a helmet is to protect the driver's or pillion rider head just in case of an accident or fall from bike.

We came up with an approach that first collected a dataset of the time-image of road traffic where we've got differing kinds of photos like with helmet, without a helmet and also the rider is wearing a helmet and another person not wearing a helmet so differentiates the 2 wheelers from other vehicles on road. It then checks whether the rider and pillion rider is wearing a helmet or not using an open-source computer vision and machine-learning software called OpenCV. If anybody of the riders is found not wearing the helmet, their vehicle number plate is processed using optical character recognition (OCR).

Key words—Helmet Detection; Machine Learning; OpenCV; OCR.

1. INTRODUCTION

In the larger part of nations, the two-wheeler is a mainstream method for transport. It is because of the low price and low maintenance cost as contrasted and another vehicle. In any case, there is less security and high danger engaged with engine bicycles. It is profoundly alluring for bike riders to utilize a protective cap to scale back the danger. In any case, the high risk and less safety is involved with bikes. In the past 10 years, it was noticed an growth in the number of bike mishaps.

With regards to insights given by transport service around 28 bike riders kick the bucket every day on Indian streets in 2016 on account of not wearing head protectors. Additionally, it is demonstrated that one among six bicycle riders kicked the bucket because of not wearing a head protector. To downsize the included danger, it is profoundly attractive for bicycle riders to utilize a protective cap.

There are existing strategies that use particular sensors in the ergonomics of the motorbike to check the presence of a head protector. However, it is difficult to persuade each client to the establishment of sensors on the bicycles. [1-3].

As of now, all significant urban communities previously sent huge video observation organizations to keep a vigilance on a wide assortment of street dangers. In this way utilizing such previously existing framework will be a cost-effective arrangement, anyway, these frameworks require an enormous number of people whose presence isn't supportable for significant stretches of time.

Ongoing investigations have demonstrated that human reconnaissance is inadequate, and not exact as the hour of observing recordings builds, human mistakes likewise increments and it is a repetitive cycle. Mechanization of this repetitive process is far required for solid and hearty observing of these infringements just as it likewise decreases the prerequisite of HR required. Be that as it may, to embrace such programmed arrangements certain moves should be addressed.

2. Related work & problems to be addressed

2.1) significant measure of data

Gathering in a restricted time might be a testing task. As such applications include errands like information assortment, highlight extraction, characterization, and following, in which a significant and prominent measure of data should be handled quickly in a brief span to accomplish our objective of continuous execution [1] [2].

2.2) Different situations:

In genuine situations, the dynamic articles ordinarily block each other because of which basic item may just be incompletely noticeable. Recognizable proof and isolation get hard for these somewhat noticeable articles [3].

2.3) Direction of vehicle movement:

As vehicles are 3-dimensional items they generally have various appearances from the changed point of vision or survivable. Notably, the exactness of classifiers relies upon highlights we utilized which thus relies totally upon point somewhat. The best model is to consider the appearance of a bicycle rider from the front view and side view.

2.4) Temporal Changes in Conditions:

Over time, there are numerous adjustments in climate conditions like light, shadows, fog, and so on there could be minute or prompt changes which expands trouble of undertakings like foundation displaying of pictures.

2.5) Quality of Video:

Generally, The CCTV cameras catch low goal recordings and are affected by conditions, for example, low light, terrible climate, fog. Because of such restrictions, undertakings, for example, distinguishing proof, grouping, and the following turn out to be much more troublesome.

As supported in [1], effective structures for observation application have helpful properties like continuous execution, adjusting, dynamic to abrupt changes, and unsurprising. Remembering all difficulties and wanted impacts, we proposed a technique for programmed discovery of bicycle riders without cap utilizing information from existing surveillance cameras, which works progressively mounted on streets. C. Chiu et al. expressed a way to deal with identify protective caps in reconnaissance recordings.

This cycle crops the moving item and afterward distinguishes bikes and heads utilizing likelihood. This framework couldn't deal with minute varieties on account of commotion and brightening impacts. [4] In [2] two stages were utilized for head protector recognition. In the principal stage, moving articles were resolved where a cross-line was determined. It at that point orders if it is a motorbike.

In the subsequent advance, a procedure was utilized to improve productivity. An SVM classifier was utilized to arrange moving items into two classes. Three characterization families were utilized viz. mathematical, intermittent, and tree-based. Recordings were caught at 25 fps and the picture size was 1280x720 to beat conditions, for example, low light, awful climate J. Chiverton et al. utilized edge histogram highlights to distinguish bike drivers. This technique performed well regardless of whether there's low light in reconnaissance recordings because of the utilization of edge histograms close to the head.

In any case, as the edge histograms utilized round hough changes to analyze and arrange caps, it prompts a ton of misclassification among motorcyclists with a Helmet as items like head protectors were ordered while the distinctive Helmet was not classified. [5]

3. LITERATURE SURVEY

3.1 Paper-1

This paper, they implemented the advanced deep learning model motorcyclist using CNN. They used an advanced deep learning method YOLOv2 which combines both object detection and classification of objects with in a single architecture. YOLOv2 implements two different stages successively in order to improve the helmet detection accuracy. At the first stage, the YOLOv2 detects different objects in the given input test image. Then it crops image and cropped images of detected persons in image are provided as input to the second YOLOv2 stage which was previously trained on our predefined dataset of persons wearing helmet images. The proposed approach involved several steps like person detection from input images.

It detects classes in given image and segregate them such as a person, car, motorbike, etc., in addition to other classes that are predefined in the dataset. We utilize a person detection class after segregating of all other classes in order to identify helmeted motorcycle rider. All other segregated classes from the primary YOLOv2 model will be discarded to reduce the lag in the architecture. Intermediate processing is implemented where all the other classes except person are removed. Also, the detected person's bounding box is cropped automatically and that image is stored for further analysis and process. Helmet detection is third step for this step, again used YOLOv2 model which is trained with predefined dataset of helmeted images of Motorcyclists.

The cropped images of the detected person that are obtained in second stage are provided as input here to the YOLOv2 model. This paper successfully decreases the number of helmets being undetected. The helmets are detected accurately in crowded areas and also in the images with a single motorcyclist. In order to increase the helmet detection accuracy used two-stage of YOLOv2 models. The algorithm can successfully distinguish between cap and helmet despite both having the same features. The algorithm has many negatives it requires large datasets and the weather conditions may affect the accuracy of the detection i.e the effect of weather conditions is not addressed in the paper.

3.2PAPER-2

This paper proposed a architecture for the automatic detection of Bike riders driving without helmets which is observed from surveillance videos. In this proposed approach, adaptive background subtraction is implemented on video frames to get proper images from moving objects that are observed in surveillance video.

Later convolutional neural network (CNN) is utilized to select two wheelers among other moving objects. The steps involved are Background Modelling and Moving Object Detection, CNN for Object Classification and detection, Recognition of Motorcyclists from Moving Objects, Recognition of Motorcyclists without Helmet.

This paper addressed issues like illumination effects and the occlusion of objects. The background modeling and moving object detection are very accurate. Adaptive background subtraction which may rise invariant various challenges such as illumination, poor quality of the video is clearly discussed in the paper.

The experiments on real time surveillance videos successfully detected 92.87% bike riders without helmet with a low false rate of 0.5% on average and thus shows how efficient is the proposed approach yet the helmets are not classified as objects in this paper.

3.3 PAPER-3

This paper is completely based on the safety of workers in industrial areas where the wearing of helmets is mandatory. It used the Faster R-CNN algorithm to inspect the wearing of a safety helmet. The experimental outputs show that compared with the Faster R-CNN algorithm, the mean average precision of the Improved Faster R-CNN is improved and the real-time automatic detection of the wearing of safety helmets is realized.

This paper used techniques of Used R-CNN which is more accurate than CNN. All, images are normalized before being input into the feature extraction framework which improves the accuracy of classification of the helmet in traffic. By applying this method to the on-side operation of substations, the interference of light, distance, and other factors can be overcome and the wearing situation of multiple people can be identified at the same time.

The accuracy obtained by this method is up to 94.3%. In addition to this, all positives a few drawbacks involved in this paper also. Not addressed scenario at different backgrounds like a large crowd, large gatherings, and more blockings in background. In order to train the system with the pictures, we need vast data set of different backgrounds and different conditions of weather lighting and object blocking. They implemented R CNN which is advanced than CNN which will increase the accuracy of detecting riders with helmets and without helmets but, still the detection speed is slow.

3.4 PAPER-4

This paper proposes motorcycle riders are detected using the YOLOv3 model which is an incremental version of the YOLO model. In the second stage, a CNN based architecture has been utilized for helmet detection of motorcycle riders. The steps involved are Motorcycle rider detection using CNN where implemented YOLO3. The second stage is Helmet detection using CNN. Then, the proposed lightweight convolutional neural network detects the wearing of a helmet or no helmet for all motorcycle riders. In this proposed architecture they used YOLO3 which is advanced CNN used that of YOLO 2.

This architecture performs comparably well with other CNN based helmet detection. The accuracy of results and detection of the rider without a helmet is up to 96.23 which is almost perfect results for the project. This paper discussed the children also as the heads of children is very small and sometimes children are in their parent's hands so this paper also worked on detecting the heads of small children and highlights of the fact that children security on two-wheelers.

This paper has few drawbacks like not addressed the problems risen by the complex scenarios of bad weather for detection of helmetless motorcyclists not addressed. They have not discussed the low video quality, less clarity of video, and resolution of video which may affect the detection of moving objects in input video. The main drawback is they not discussed the Classification of helmet-like objects (cap, monkey caps, Turbans).

4. PROPOSED FRAMEWORK

The proposed work mainly involves two main steps: i) The deep neural network which is employed for identification of single and multiple riders on a motorcycle using the YOLOv3 model and ii) We implemented another deep neural network in this which is implemented for motorcycle rider's helmet detection. For these two steps, traffic surveillance video is the main input to the YOLO3 model, and the individual video frames are obtained by cropping images and taking screenshots from input videos are utilized as the input to the CNN to detect motorcycle riders and pillion riders with or without a helmet.

4.1 Motorcycle riders detection using CNN

A convolutional neural network (CNN) is a variety of feed-forward neural network that implements the back propagation algorithm. It trains it high-level features from the primary data like images. The current accomplishment of convolutional neural networks is in their ability to extract inter-dependent information from the input images i.e centralization of the pixels which are highly sensitive compared to other pixels. The convolutional neural network training consists of different convolution layers, relu layers max-pooling layers, fully connected layers, and a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer these layers are liable for the detection, classification, and evaluating of objects in images. In the preliminary layers we obtain the edge information of the input images familiar to some of the algorithms but, In the penultimate layers, we start obtaining texture and ridge information which helps us in evaluating sensitive information useful for the classification of objects in images into different classes based on their sizes and category(moving or not).

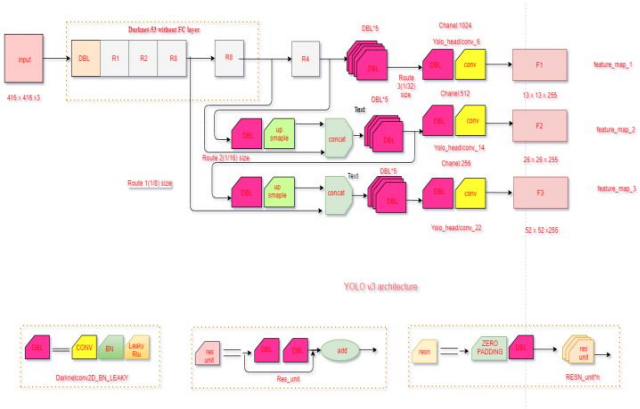


Fig 1 Architecture of YOLO

4.2 Helmet detection using CNN

The proposed CNN architecture is able to detect and classify helmet in images. The top portion of the identified motorcycle rider mainly the head portion is cropped and forwarded as input for the proposed CNN architecture of helmet detection. This proposed network involves in five convolutional layers.

The primary layer takes the cropped input image and passes the image through successive five convolutional layers where each layer converts the image using specific functions, algorithms and sends to the consecutive layers step by step. These layers act as filters to extract features with sufficient discriminating attributes to differentiate targeted object from other objects.

After successfully passing through five convolutional layers, two fully connected layers are further added. Depending on the extracted properties from the layers, softmax classifier which classifies the object to distinguish into different classes with probability distribution as a Helmet or other objects like cap. The CNN predicts bounding boxes along with class probabilities [4] for accurate prediction of helmet. In the detection process the input image is divided into an N × N grids. This grid is responsible for object detection of any kind of objects falls into that grid's cell. Each bounding box consists of 4 measures: px, py, w, h where (px, py) coordinates represent the center of the box relative to the bounds of the grid cell. The width (w) and height (h) are predicted relative to the whole input image.

One bounding box is allotted per object based on highest Intersection over Union (IOU) which represents a fraction between 0 and 1. IOU = AreaofIntersection/AreaofUnion (1) Area of Intersection is the overlaying area between predicted bounding box of object and the ground truth.

Area of Union is the total area of both predicted bounding box and ground truth. The IOU value is predefined as threshold for object detection and it is predefined as IOU threshold = 0.5 which means that a detection with a IOU greater than 0.5 is a truly positive but practically, IOU should be near to 1 to show the perfect matching. In the proposed network to add non-linearity, Rectified Linear Unit (ReLU) activation function is used after each convolutional layer, followed by additional max-pooling layers that are added after convolutional layers for dimension reduction of the feature maps.

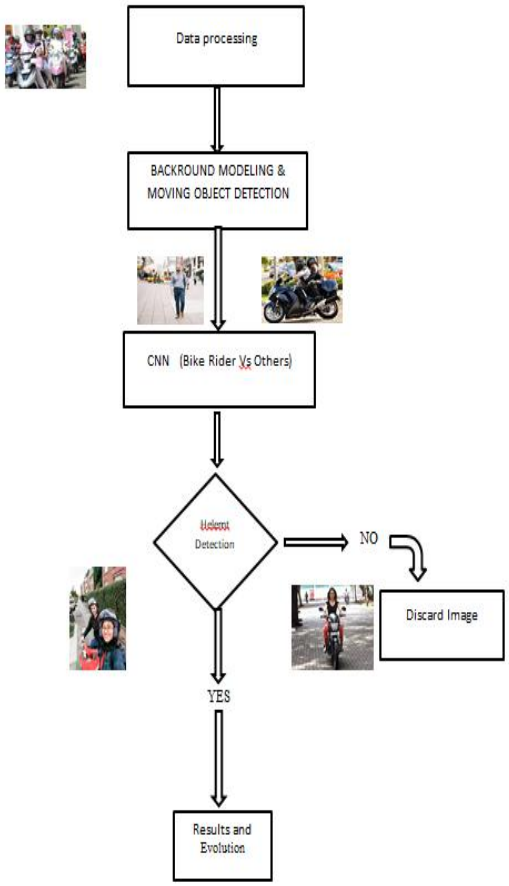


Fig 2 Flow chart of implemented method

Fig.3 Detection of the Helmet

Fig.4 Detection of Without Helmet

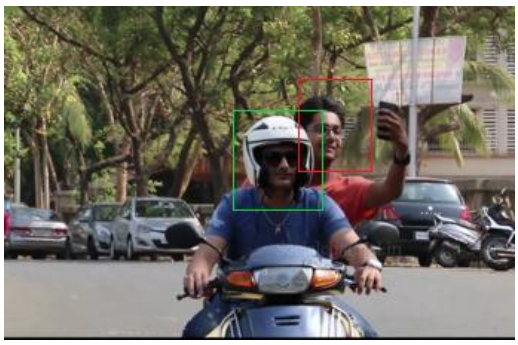


Fig.5 Detection of combined riders with & without Helmet

5. Results and Discussion

The experiments are implemented on non-GPU computer. The experimental platform is Intel core i7 7th generation 4.7 GHz CPU with 16 GB RAM. To implement this proposed model, we opted Python 3.6 as programming language, OpenCV 3.0 as computer vision library, a neural network framework Darknet and various libraries have been used in evolving the accurate results of proposed method. For detection of motorcycle riders, one hour traffic video is recorded using surveillance cameras. Pre uploaded internet images of motorcyclists are also taken for testing purpose to train the model. Altogether among 180 motorcyclists, 174 motorcyclists are successfully identified in the classification of motorcycle riders that shows accuracy of 96.23% which is acceptable in comparison with other existing methods. Helmet dataset is made by considering internet motor-cyclists images and real-time traffic helmet images that are obtained from the cropping the images from video. The recording taken for this purpose is a 45 minutes video. The total frame of first 25 minutes video is used for training the network. Another 10 minutes video is used to validation of the network and remaining 10 minutes video is used for testing purpose of the network. YOLOv3 architecture is used for detection and differentiation of person and motorcycle from others. On the second stage for helmet detection, lightweight CNN architecture has been proposed. The lightweight CNN architecture for non-GPU computer inspired from YOLO-LITE [3]. Due to smaller input image size for helmet detection, less data passes through the network which increases the speed of the network. As batch normalization is not essential for a small network, therefore no intermediate calculation is required for it at each layer of the network. Hence, there is no additional time loss in the feed-forward process.

Technique	Accuracy(%)	Performance of detecting Helmet(%)	Performance of detecting NoHelmet(%)
YOLO3	96.23	99.28	99.21
YOLO2	94.12	98.88	98.91
RCNN	94	91.81	91.78
HOG SVM	92.8	81.84	81.84

Table 1 PERFORMANCE (%) OF THE CLASSIFICATION OF 'HELMET AND NO HELMET' DIFFERENT TECHNIQUES

The above table shows the difference between different techniques used before to detection of helmets, and the accuracy of detection of the helmet. The highest accuracy is observed in the YOLO3 model which was implemented for this project and other methods show their respective accuracies and prove that YOLO3 is produced the best outputs. YOLO3 showed better results in detection of a helmet that is up to 99.28% of the provided images and also effectively detects no helmet also up to 99.21% accurately which increases the reliability of the YOLO3 to the detection of a helmet in this project.

Technique	Low false rate
YOLO3	0.001 – 0.1
YOLO2	0.001
RCNN	0.001
HOG SVM	<0.5

TABLE 2 LOW FALSE RATE OF DIFFERENT TECHNUQUES

The Table 2 shows the Low false rate of different Techniques and shows that the YOLO3 has less low false rate than other techniques which increases the accuracy of detecting riders wearing a helmet and not wearing the helmet.

6. Conclusion and Future work

The software of the helmet detection has been thoroughly tested and implemented we have very good exercise in high level language and have realized the ingenuity and patience with this job has to be done. In our project we provided the YOLO based Helmet detection and also made a detailed study about CNN.

We used jupyter notebook to implement the program and we successfully implemented the program. Our project was tested successfully tested in python. We also made study of applications and future scope of the project.

Our project can be linked with the traffic cameras and with some modifications it can be used to detect helmets in the real time system. Further more we can merge the algorithm of automated license plate detection and make a system which generates challans for those who don't wear helmets.

Reference papers

[1] Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pp. 555–560, March 2008.

[2] B. Duan, W. Liu, P. Fu, C. Yang, X. Wen, and H. Yuan, "Real-time onroad vehicle and motorcycle detection using a single camera," in Procs. of the IEEE Int. Conf. on Industrial Technology (ICIT), 10-13 Feb 2009, pp. 1–6.

[3] C.-C. Chiu, M.-Y. Ku, and H.-T. Chen, "Motorcycle detection and tracking system with occlusion segmentation," in Int. Workshop on Image Analysis for Multimedia Interactive Services, Santorini, June 2007, pp. 32–32

[4]C.-C. Chiu, M.-Y. Ku, and H.-T. Chen, "Motorcycle detection and tracking system with occlusion segmentation," in Proc. Int. Workshop on Image Analysis for Multimedia Interactive Services, Santorini, Greece,6–8 June 2007, pp. 32–32

[5]J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," IET Intelligent Transport Systems (ITS), vol. 6, no. 3, pp. 259–269, 2012.

[6] W. Rattapoom, B. Nannaphat, T. Vasan, T. Chainarong, and P. Pattanawadee, 'Machine vision techniques for motorcycle safety helmet detection,' in Procedings of International Conference on Image and Vision Computing, pp. 35–40, 2013.

[7] J. Li, H. Liu, T. Wang, M. Jiang, S. Wang, K. Li, and X. Zhao, 'Safety helmet wearing detection based on image processing and machine learning' In Proceedings of IEEE International Conference on Advanced Computational Intelligence (ICACI), pp. 201-205, 2017.

[8] R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras, and A. Soares, 'Automatic detection of motorcyclists without helmet,' in Proceedings of Latin American Computing Conference (CLEI), pp. 1–7, 2013.

[9] R. V. Silva, T. Aires, and V. Rodrigo, 'Helmet detection on motorcyclists using image descriptors and classifiers,' in Proceedings of Graphics, Patterns and Images (SIBGRAPI), pp. 141–148, 2014.

[10] G. Ross, D. Jeff, D. Trevor, and M. Jitendra, 'Rich feature hierarchies for accurate object detection and semantic segmentation,' in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587, 2014.

[11] J. Canny, 'A computational approach to edge detection,' IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[12] A. Hirota, N. H. Tiep, L. Van Khanh, and N. Oka, 'Classifying Helmeted and Non-helmeted Motorcyclists', Chapter: Springer International Publishing, pp. 81–86,2017.

[13] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, 'Microsoft COCO: common objects in context,' CoRR, vol. abs/1405.0312, 2014.

[19] J. E. Espinosa, A. S. Velastin, and J. W. Branch. 'Motorcycle detection and classification in urban Scenarios using a model based on Faster R-CNN.', pp. 16-22, 2018.

[20] C. Vishnu, D. Singh, C. K. Mohan, and S. Babu, 'Detection of motorcyclists without helmet in videos using convolutional neural network' In Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN), pp. 3036-3041, 2017.

[21] J. Mistry, K. A. Misraa, M. Agarwal, A. Vyas, V. M. Chudasama, and K. P. Upla, 'An automatic detection of helmeted and non-helmeted motorcyclist with license plate extraction using convolutional neural network' In Proceedings of IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1-6, 2017.

[22] J. Redmon, 'Darknet: Open source neural networks in c,' http://pjreddie.com/darknet/, 2013–20

[23] Z. Kaihua, L. Qingshan, Wu, and Y. Ming-Hsuan, "Robust Visual Tracking via Convolutional Networks without Training," IEEE Trans. Image Processing, vol. 25, no. 4, pp. 1779–1792, 2016. [14] G. Ross, D. Jeff, D. Trevor, and M. Jitendra, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Colombus, Ohio, 24–27 June 2014, pp. 580–587.

[24] R. E. Kalman, "A new approach to linear filtering and prediction problems," Journal of Basic Engineering, vol. 82, no. 1, pp. 35–45, 1960.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[26] D. Navneet and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), San Diego, California, 20–26 June 2005, pp. 886– 893.

[27] Z. Guo, D. Zhang, and L. Zhang, "A completed modeling of local binary pattern operator for texture classification," IEEE Trans. Image Processing, vol. 19, no. 6, pp. 1657–1663, 2010.

[28] C. Cortes and V. Vapnik, "Support vector networks," Machine Learning (Springer), vol. 20, no. 3, pp. 273–297, 1993. [20] D. Singh, D. Roy, and C. K. Mohan, "Dip-svm: distribution preserving kernel support vector machine for big data," IEEE Trans. on Big Data, 2017. [Online]. Available:

[29] D. Singh and C. K. Mohan, "Distributed quadratic programming solver for kernel SVM using genetic algorithm," in Proc. IEEE Congress on Evolutionary Computation, Vancouver, July 24–29 2016, pp. 152–159. [22] Y. Lecun, L. bottou, Y. bengio, and P. haffner, "Gradient-based learning applied to document recognition," Proceedings of IEEE, vol. 86, no. 1, pp. 1–6, 1990.

[30] Z. Zoran, "Improved adaptive gaussian mixture model for background subtraction," in Proc. Int. Conf. Pattern Recognition (ICPR), Cambridge, England, UK, 23–26 August 2004, pp. 28–31.

[31] S. Chris and G. Eric, "Adaptive background mixture models for realtime tracking," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Collins, CO, USA, 23–25 June 1999, pp. 246– 252.

[32] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[33] V. der Maaten, Laurens, Hinton, and Geoffrey, "Visualizing data using t-sne," Journal of Machine Learning Research (JMLR), vol. 9, no. 1, pp. 2579–260

BIOGRAPHIES

Dr.M.V.D PRASAD
Assoc.Professor, Department of ECE, KLEF-Deemed to be University Guntur, A.P., India

S.V.N.P VAMSI KRISHNA, Student
B.tech Department of ECE, KLEF-Deemed to be University Guntur, A.P., India

M.SANTOSH KUMAR, Student
B.tech Department of ECE, KLEF-Deemed to be University Guntur, A.P., India

P.SRI HARSHA, Student
B.tech Department of ECE, KLEF-Deemed to be University Guntur, A.P., India

Automatic Helmet Detection on Public Roads

Article *in* International Journal of Engineering Trends and Technology · May 2016

DOI: 10.14445/22315381/IJETT-V35P241

4 authors, including:

Shilpa Gite
Symbiosis International University
68 PUBLICATIONS **261** CITATIONS

SEE PROFILE

Automatic Helmet Detection on Public Roads

Maharsh Desai #1 , Shubham Khandelwal #2 , Lokneesh Singh #3 , Prof. Shilpa Gite #4

Student #1 ,Student #2 ,Student #3 , Assistant Professor #4 ,Department of CS/IT, Symbiosis Institute of Technology Symbiosis Institute of Technology, Pune, India.

ABSTRACT

Bike riding is a lot of fun, but accidents happen. People choose motorbikes over car as it is much cheaper to run, easier to park and flexible in traffic. In India more than 37 million people are using two wheelers. Since usage is high, accident percentage of two wheelers are also high compared to four wheelers. Motorcycles have high rate of fatal accidents than four wheelers. The impacts of these accidents are more dangerous when the driver involves in a high speed accident without wearing helmet. It is highly dangerous and can cause severe deaths. So wearing a helmet can reduce these number of accidents and may save the life. This paper aims for avoidance of accidents and develop helmet detection system. We intend to use background subtraction and optical character recognition for fall detection and for helmet detection we use background subtraction and Hough transform descriptor.

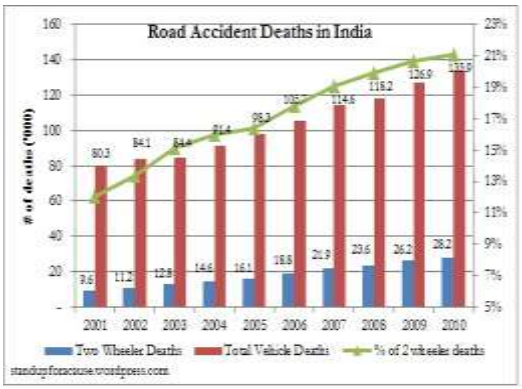
Keywords: *Helmet detection system, fatal, impact, Hough transform descriptor, background subtraction*

I. INTRODUCTION

The project aims to provide total safety for bike riders. Recently helmets have been made compulsory, but still people drive without helmets. Pune City has approx. 35 lakh two-wheeler riders, which includes 500-600 accidents every year out of which 300-400 are fatal. Pune ranks first in the city when it comes to two wheelers riders. In the last few years, there has been rapid increase in number of road accidents. Due to rise in road accidents, it has now become necessary to generate a system to limit accidental deaths.

India accounts for more than 200,000 deaths because of road accidents, according to the Global Road Safety Report,2015 “the report states that the Indian road safety laws do not meet the best practice requirements for four out of five risk factors: enforcing speed limits, prevention of drunk driving, safety of children and use of helmets. According to a report by a Tamil Nadu police, there were a total of 15563 injuries in 14504 accidents. The state also topped the list of most accidents among all states for previous ten years from 2002 to 2102.

Even for seat-belts, where the Motor Vehicles Act, 1988, is in consonance with the WHO standards, the enforcement is poor and India has a pathetic score of four out 10. With respect to vehicle safety, India meets only two out of the seven vehicle safety standards by the World Health Organization (WHO). Two wheelers account for 25% of total road crash deaths. Nearly 75%motorcycle riders involved in accidents continued to wear helmets, crash records show. The main cause of these fatalities is people riding two wheelers under the influence of alcohol results and violation of traffic rules which later on results in serious accidents. “The likelihood of survival of fatalities wearing helmets are high as compared to those not wearing helmets”.



II.CONTRIBUTIONS

The main aim of this study is to propose and develop a system for automatic detection of helmets on public roads. The moto of this project is accident prevention by using the methods of alcohol detection, helmet authentication, fall detection etc.

Helmet Detection:

More than one third who died in road accidents could have survived if they had worn a helmet. Studies shows that usage of helmet can save accident death

by 30 to 40%. The rate at which number of two wheelers in India is rising is 20 times the rate at which human population is growing. The risk of death is 2.5 times more among riders not wearing a helmet compared with those wearing a helmet.

Basically we can detect helmet combining two methods that is background subtraction and Hough transform descriptor. We will extract the background image and then on a particular segment we will apply Hough

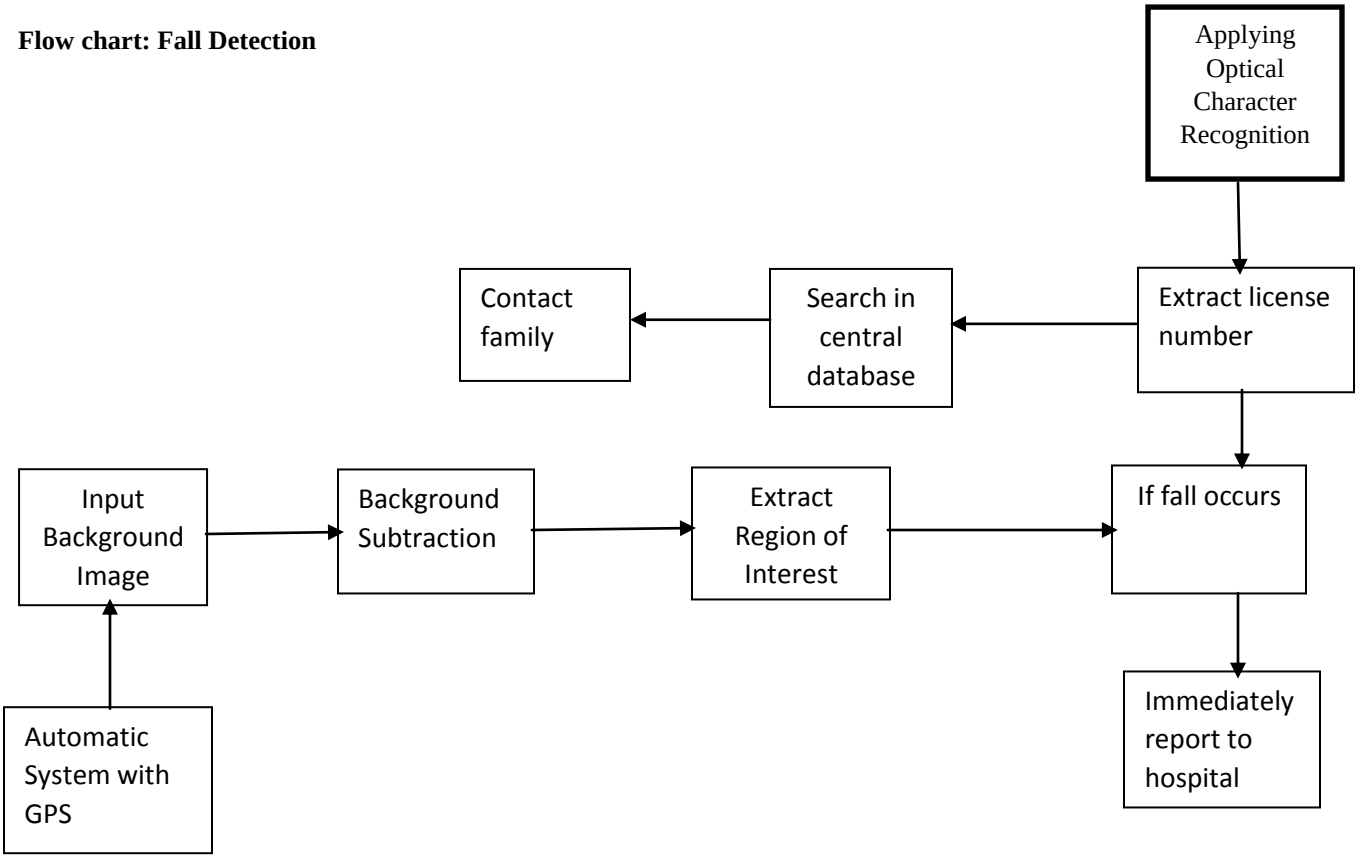
Transform Descriptor. Hough Transform descriptor is basically used for detecting

regular curves such as lines, circles etc. We will detect circles here and find whether a person is wearing helmet or not. If a person is not wearing helmet, use back ground subtraction method for extracting the license plate and use optical recognition method, to get license number of the vehicle. Then look into the database and send the ticket to the matched person.

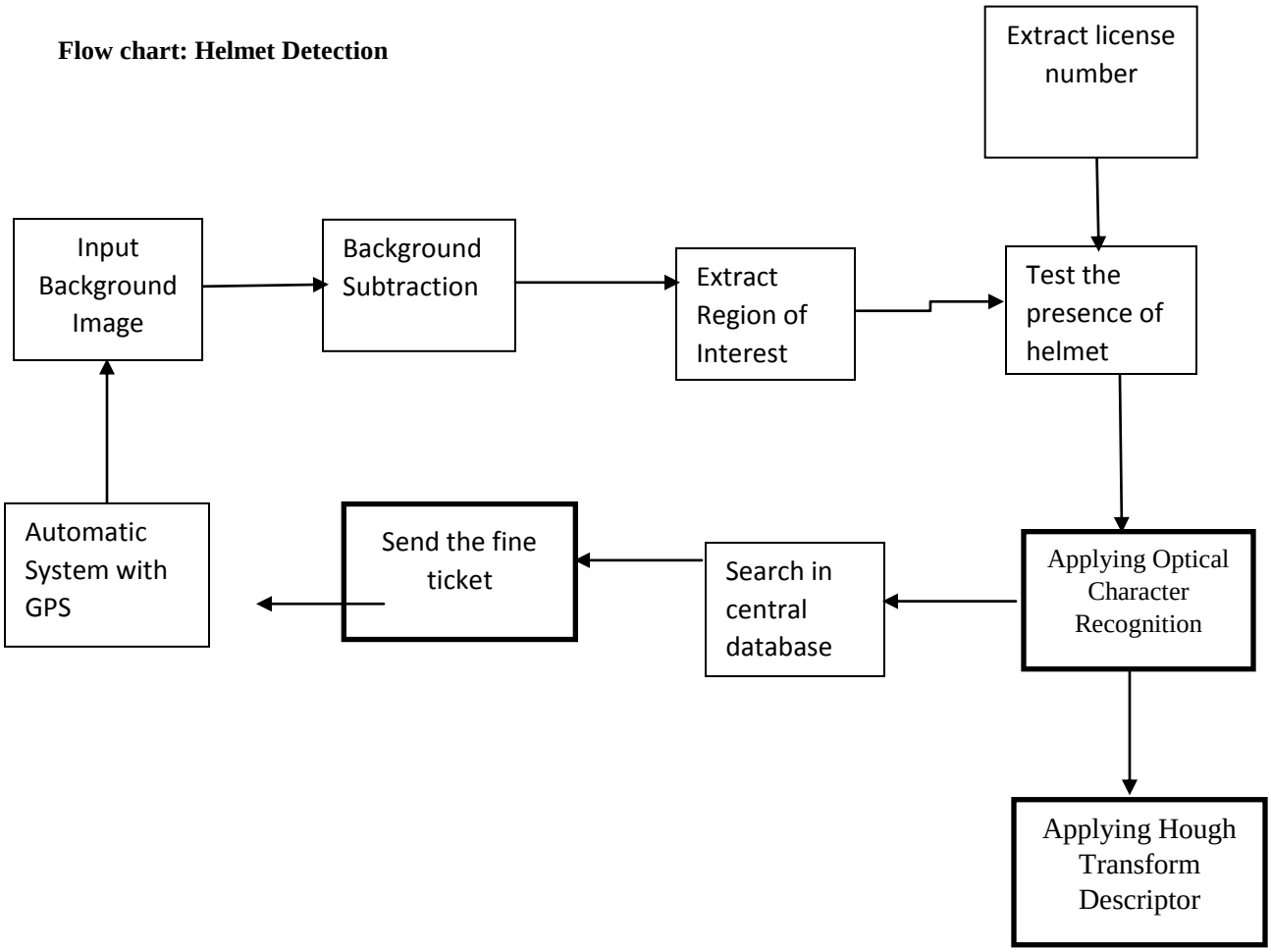
Fall detection:

Fall detection indicates accident has occurred. If any fall is detected, then a message should be sent automatically to his family. We can detect fall detection by using optical recognition technique. First of all, we will use background subtraction where image background is subtracted. An automatic system which is linked with GPS device to get information such as location, time. If any fall is detected by the system, the system will extract license number plate from image frame and will search in central database and will find owner's information. The system will immediately report to the nearby hospital informing them about the accident. System will also contact family informing them about accident.

Flow chart: Fall Detection



Flow chart: Helmet Detection



I .RELATED WORKS

Our work is closely related with the study of helmet detection methods.

A. Vision based method:

It is one of the most popular techniques for traffic surveillance due to low hardware cost.

B. Background subtraction:

This is one of the method where image background is extracted for further processing. It is the best approach for detecting objects from videos taken by static cameras. There are many techniques and both expert and new comers can be confused about limitations and benefits of it. This method based on static background hypothesis not applicable in real environments.

C. Object detection:

It is process of finding instance of real world objects such as faces. We can use Local Binary Pattern,Histogram of Oriented Gradients and Hough transform descriptors

D. Local Binary Pattern:

It is used for face recognition in computer vision. In this method [5] image is divided into several small segments and from which features are extracted. It consists of binary patterns and describe surrounding of pixels.

The features from segment are joint into single feature histogram. This method provides good result in term of speed.

E. Histogram of Oriented Gradients Descriptor:

provides better performance than other existing feature sets. It is used to extract human feature from visible spectrum images. It has been determined that when LBP combined with HOG descriptorit improves, detection, performance considerably on some data sets.

F. Hough transform descriptor:

It is a technique and can be used to isolate features of particular shape with an image. It requires some features in parametric form. It is mostcommonly used for detectingregular curves such as lines, circles, ellipses etc.

IV .RESULTS

To ensure bike rider's safety, we have designed this project. Many projects have been designed so far but they all are concentrated more on four wheelers. Very less importance was given to motorbikes. Today accidents caused by motorbikes are more than cars. Thus in this project safety of bike rider is major concern.

The project consists of 3 parts:

1. Helmet Authentication; to ensure that the bike rider is wearing a helmet.
2. Alcohol detection; to ensure that the bike rider has not consumed any type of alcohol.
3. Fall detection; in case of accident, to inform bike rider's family about the accident.

V. Conclusion

In this paper we have proposed an approach which would detect fall & helmet detection of a two-wheeler driver run-time. Our system would inform nearby hospitals, family members & law enforcement agencies in case of emergency. Hence it ensures safety of the drivers while driving.

Automatic accident detection and reporting system is the motivation of this project. To prevent road accidents, our approach is very useful. Thus safety of bike riders is ensured.

In future we intend to use more advanced safety measures like to check alcohol consumption, lane change detection, collision detection, traffic information, e-toll collection, license renewal etc. We also think of applying deep neural network techniques & make transportation more intelligent.

VI. References

1. IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834, - ISSN: 2278-8735.Volume 10, Issue 4, Ver. II (Jul - Aug .2015), PP 55-59 www.iosrjournals.org
2. CLEI ELECTRONIC JOURNAL, VOLUME 16, NUMBER 3, PAPER 04, DECEMBER 2013
3. ISSN 2319-2518, No.2, Vol 4, April 2015
4. CLEI ELECTRONIC JOURNAL, VOLUME 16, NUMBER 3, PAPER 04, DECEMBER 2013
5. Face Recognition using Local Binary Patterns (LBP) By Md. Abdur Rahim, Md. Najmul Hossain, Tanzillah Wahid Pabna University of Science and Technology, Bangladesh
6. M. Fathy and M. Y. Siyal, Senior Member, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 47, NO. 4, NOVEMBER 1998

Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time

Conference Paper · July 2016

DOI: 10.1109/IJCNN.2016.7727586

3 authors:

Kunal Dahiya
Indian Institute of Technology Delhi
14 PUBLICATIONS **142** CITATIONS
SEE PROFILE

Dinesh Singh
RIKEN
24 PUBLICATIONS **458** CITATIONS
SEE PROFILE

Krishna Mohan Chalavadi
Indian Institute of Technology Hyderabad
73 PUBLICATIONS **1,191** CITATIONS
SEE PROFILE

Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time

Kunal Dahiya, Dinesh Singh, C. Krishna Mohan
Visual Learning and Intelligence Group (VIGIL),
Department of Computer Science and Engineering,
Indian Institute of Technology, Hyderabad, India
Email: {cs11b15m000001, cs14resch11003, ckm}@iith.ac.in

***Abstract*—In this paper, we propose an approach for automatic detection of bike-riders without helmet using surveillance videos in real time. The proposed approach first detects bike riders from surveillance video using background subtraction and object segmentation. Then it determines whether bike-rider is using a helmet or not using visual features and binary classifier. Also, we present a consolidation approach for violation reporting which helps in improving reliability of the proposed approach. In order to evaluate our approach, we have provided a performance comparison of three widely used feature representations namely histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), and local binary patterns (LBP) for classification. The experimental results show detection accuracy of 93.80% on the real world surveillance data. It has also been shown that proposed approach is computationally less expensive and performs in real-time with a processing time of 11.58 ms per frame.**

I. Introduction

Two-wheeler is a very popular mode of transportation in almost every country. However, there is a high risk involved because of less protection. To reduce the involved risk, it is highly desirable for bike-riders to use helmet. Observing the usefulness of helmet, Governments have made it a punishable offense to ride a bike without helmet and have adopted manual strategies to catch the violators. However, the existing video surveillance based methods are passive and require significant human assistance. In general, such systems are infeasible due to involvement of humans, whose efficiency decreases over long duration [1]. Automation of this process is highly desirable for reliable and robust monitoring of these violations as well as it also significantly reduces the amount of human resources needed. Also, many countries are adopting systems involving surveillance cameras at public places. So, the solution for detecting violators using the existing infrastructure is also cost-effective.

However, in order to adopt such automatic solutions certain challenges need to be addressed: 1) *Real-time Implementation:* Processing significant amount of information in a time constraint manner is a challenging task. As such applications involve tasks like segmentation, feature extraction, classification and tracking, in which a significant amount of information need to be processed in short duration to achieve the goal of real-time implementation [1] [2]. 2) *Occlusion:* In real life scenarios, the dynamic objects usually occlude each other due to which object of interest may only be partially visible. Segmentation and classification become difficult for these partially visible objects [3]. 3) *Direction of Motion:* 3-dimensional objects in general have different appearance from different angles. It is well known that accuracy of classifiers depends on features used which in turn depends on angle to some extent. A reasonable example is to consider appearance of a bike-rider from front view and side view. 4) *Temporal Changes in Conditions:* Over time, there are many changes in environment conditions such as illumination, shadows, etc. There may be subtle or immediate changes which increase complexity of tasks like background modelling. 5) *Quality of Video Feed:* Generally, CCTV cameras capture low resolution video. Also, conditions such as low light, bad weather complicate it further. Due to such limitations, tasks such as segmentation, classification and tracking become even more difficult. As stated in [1], successful framework for surveillance application should have useful properties such as *real-time performance, fine tuning, robust to sudden changes and predictive*. Keeping these challenges and desired properties in mind, we propose a method for automatic detection of bike-riders without helmet using feed from existing security cameras, which works in real time.

The remainder of this paper is organized as follows : Section II reviews the related work with their strengths and shortcomings. The proposed approach is presented in Section III. Section IV provides all the experimental details, results and their analysis. The last section summarizes the paper.

II. Existing Work

Automatic detection of bike-riders without helmet falls under broad category of anomaly detection in surveillance videos. As explained in [4], effective automatic surveillance system generally involve following tasks: environment modeling, detection, tracking and classification of moving objects. In [5], Chiverton proposed an approach which uses geometrical shape of helmet and illumination variance at different portions of the helmet. It uses circle arc detection method based on the Hough transform. The major limitation of this approach is that it tries to locate helmet in the full frame which is computationally expensive and also it may often confuse other similar shaped objects as helmet. Also, it oversees the fact that helmet is relevant only in case of bike-rider. In [6], Chen *et al.* proposed an efficient approach to detect and track

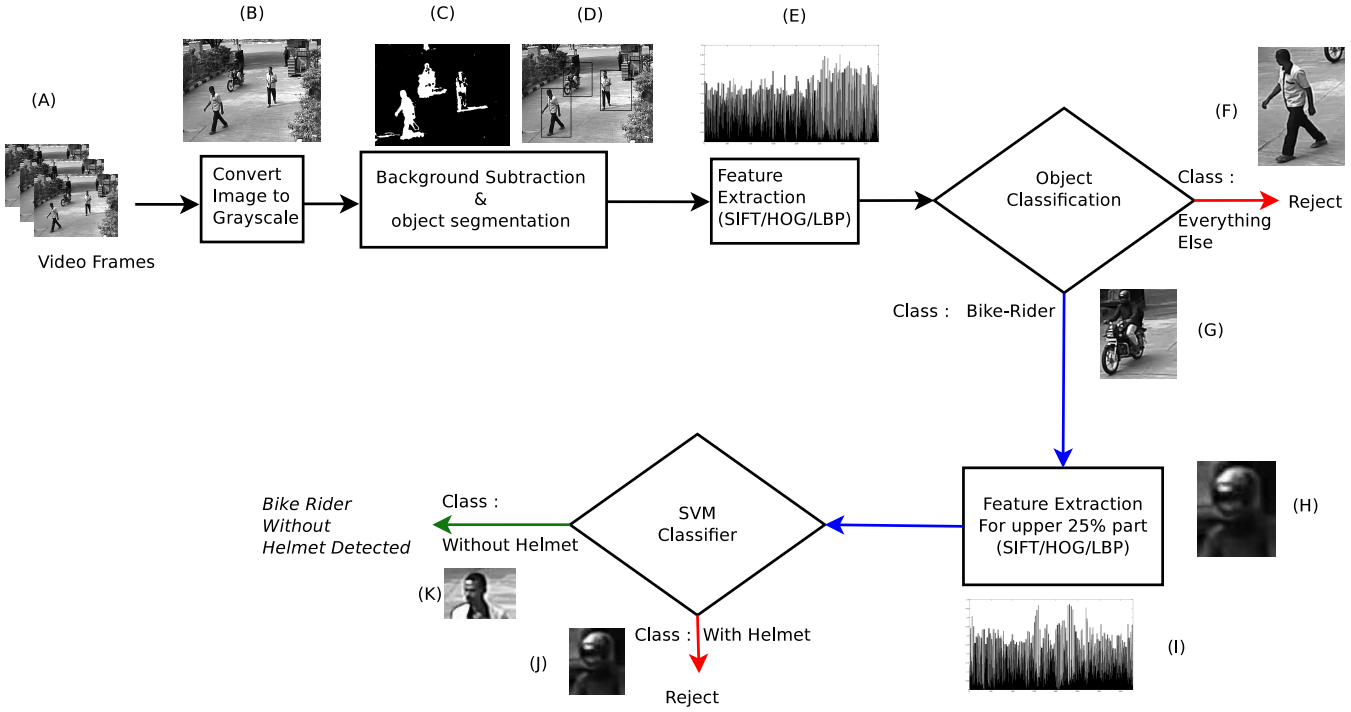


Fig. 1. Proposed approach for detection of bike-riders without helmet. A) Input frame sequence, B) A sample frame, C) Foreground mask for sample frame, D) Bounding box around foreground objects, E) Sample features of objects from D, F) Object classification as non-bike rider, G) Object classification as bike-rider, H) Localized head of the bike-rider, I) Sample Features of objects from H, J) Bike-rider classified as 'with helmet' class and, K) Bike-rider classified as 'without helmet' class.

vehicles in urban traffic. It uses Gaussian mixture model along with a strategy to refine foreground blob in order to extract foreground. It tracks a vehicle using Kalman filter and refine classification using majority voting. In [2], Duan *et al.* suggest a robust approach for tracking of vehicles in real-time from single camera. In order to accelerate the computation, it used integrated memory array processor (IMAP). However, it is not an efficient solution due to its requirement of dedicated hardware. In [7] [8], Silva *et al.* proposed an approach which starts with detection of bike-riders. Then it locates the head of bike-riders by applying Hough transform and then classifies it as head or helmet. However, Hough transform for locating head of bike-rider can be computationally expensive. Also, in [8] experiments are performed on static images only. Broadly, there are two major limitations in the existing work discussed above. Firstly, suggested approaches are either computationally very expensive [5] [7] or passive in nature [2] [8] which are not suitable for real time performance. Secondly, the correlation between the frames is underutilized for final decisions [5] [7], as the results from consecutive frames can be combined in order to raise more reliable alarms for violations. The proposed approach overcome above discussed limitations by providing an efficient solution which is suitable for real-time application.

III. Proposed Work

This section presents the proposed approach for real-time detection of bike-riders without helmet which works in two phases. In the first phase, we detect a bike-rider in the video frame. In the second phase, we locate the head of the bike-rider and detect whether the rider is using a helmet or not. In order to reduce false predictions, we consolidate the results from consecutive frames for final prediction. The block diagram in Fig. 1 shows the various steps of proposed framework such as background subtraction, feature extraction, object classification using sample frames.

As helmet is relevant only in case of moving bike-riders, so processing full frame becomes computational overhead which does not add any value to detection rate. In order to proceed further, we apply background subtraction on gray-scale frames, with an intention to distinguish between moving and static objects. Next, we present steps involved in background modeling.

Background Modeling: Initially, the background subtraction method in [9] is used to separate the objects in motion such as bike, humans, cars from static objects such as trees, roads and buildings. However, there are certain challenges when dealing with data from single fixed camera. Environment conditions like illumination variance over the day, shadows, shaking tree branches and other sudden changes make it difficult to recover and update background from continuous stream of frames. In case of complex and variable situations, single Gaussian is not sufficient to completely model these variations [10]. Due to this reason, for each pixel, it is necessary to use variable number of Gaussian models. Here K, number of Gaussian components for each pixel is kept in between 3

and 5, which is determined empirically. Variable number of Gaussian components enables the background model to easily adjust it's parameters according to situation. However, some errors may still occur due to presence of highly occluded objects and merged shadows. Let us consider $I^1, I^2....I^t$ be the intensity of a pixel for past t, consecutive frames. Then at time t probability of observing intensity value for a pixel is given by:

$$P(I^t) = \sum_{j=1}^{K} w_j^t \times \eta(I^t, \mu_j^t, \sigma_j^t), \tag{1}$$

where, w_j^t is weight and $\eta(\cdot, \cdot, \cdot)$ is j^{th} Gaussian probability density function with mean μ_j^t and σ_j^t as variance at time t. For each pixel, the Gaussian components with low variance and high weight correspond to background class and others with high variance correspond to foreground class. At time t, the pixel intensity I^t is checked against all Gaussian components. If j^{th} component satisfies the condition :

$$\left|\mu_j^t - I^t\right| < e_j \sigma_j^t, \tag{2}$$

then j^{th} component is considered to be a match. Also, the current pixel is classified as background or foreground according to the class of j^{th} Gaussian model. The weight update rule is given by :

$$w_j^t = (1-\alpha) w_j^{t-1} + \alpha(M_j^t), \tag{3}$$

$$M_j^t = \begin{cases} 0, \text{ for matched model} \\ 1, \text{ otherwise ,} \end{cases} \tag{4}$$

where, α is learning rate which determines how frequently parameters are adjusted. Here, e_j is a threshold which has significant impact when different regions have different lightning. Generally the value of e_j is kept around 3, as $\mu^t \pm 3\sigma_j^t$ accounts for approximately 99% of data [9]. Also, other parameters of matched models are updated as:

$$\mu^t = (1-\rho)\mu^{t-1} + \rho I^t, \tag{5}$$

$$(\sigma^2)^{(t)} = (1-\rho)(\sigma^2)^{(t-1)} + \rho(I^t - \mu^t)^2. \tag{6}$$

Here, $\rho = \eta(I^t|\mu_j, \sigma_j)$. When there is no matched component, a new Gaussian model is created with current pixel value as mean, low prior weight and high variance. This newly created model replaces the least probable component or added as a new component if maximum number of components is reached or not, respectively. Background model is approximated using on-line clustering method proposed in [9]. Subtracting background mask from current frame results in foreground mask. In order to segment foreground mask as objects, image processing operations such as noise filter, morphological operation are used. Gaussian filter is applied to Foreground mask to reduce noise and then transformed into binary image using clustering based thresholding [11]. Morphological operations specifically close operation are used to further process the foreground mask to achieve better distinction between objects. Next, this processed frame is segmented into parts based on object boundaries. Background subtraction method retrieves only moving objects and ignore non-useful details such as static objects. Still there may be many moving objects which are not of our interest such as humans, cars etc. These objects are filtered based on their area. Let $\mathbf{B}_j$ be the j^{th} object with area a_j then B_j will be selected if $T_l < a_j < T_h$. Here T_l and T_h are threshold for minimum and maximum area, respectively. The method assumes that for a fixed camera, area of closing boundary of bikes is well differentiated from objects with very large area such as bus or very small area such as noise. The objective behind this is to only consider objects which are more likely to fall in bike-riders category. It helps in reducing the complexity of further steps.

A. Phase-I: Detection Bike-riders

This phase involves detection of bike-riders in a frame. This step uses objects $\mathbf{B}'_j s$, the potential bike-riders returned by background modeling step and classify them as 'bike-rider' vs 'others', based on their visual features. This phase involves two steps : feature extraction and classification.

1) Feature Extraction : Object classification requires some suitable representation of visual features. In literature, HOG, SIFT and LBP are proven to be efficient for object detection. For this purpose, we analyze following features :

- Histogram of Oriented Gradients [12] : HOG descriptors are proven to be very efficient in object detection. These descriptors capture local shapes through gradients. We used 9 bins, 8×8 pixels per cell and 2×2 cells per block. The resulting feature vector is $\mathbf{h}$, where $\mathbf{h} \in \mathbb{R}^n$, and n is 3780.
- Scale Invariant Feature Transform [13] : This approach tries to capture key-points in the image. For each key-point, it extracts feature vectors. Scale, rotation and illumination invariance of these descriptors provide robustness in varying conditions. We used bag of words technique to create a vocabulary $\mathbf{V}$ of size 5000. Then mapping SIFT descriptors to $\mathbf{V}$ results in feature vector $\mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^n$, and n is 5000. Feature vector $\mathbf{s}$ is used to determine similarity between images.
- Local Binary Patterns : These features capture texture information in the frame. For each pixel, a binary number is assigned by thresholding the pixels in the circular neighborhood [14] gives feature vector $\mathbf{l} \in \mathbb{R}^n$, where n is 26.

Fig. 2 visualizes the patterns of phase-I classification in 2-D space using t-SNE [15]. The distribution of the HOG feature vectors show that the two classes i.e 'bike-riders' (Positive class shown in blue crosses) and 'others' (Negative class shown in red dots) fall in almost distinct regions with only few exceptions. This shows that the feature vectors efficiently represent the activity and contains discriminative information, which further gives hope for good classification accuracy.

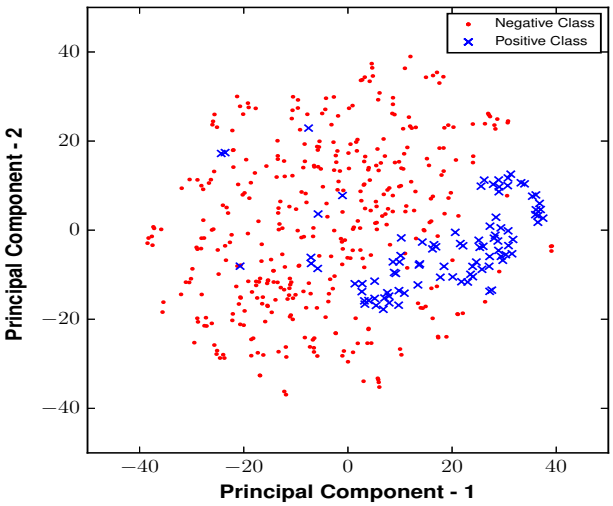


Fig. 2. Visualization of HOG feature vectors for 'bike-rider vs others' classification using t-SNE [15]. Blue cross represent bike-rider class and Red dot represent non bike-rider class [Best viewed in color]

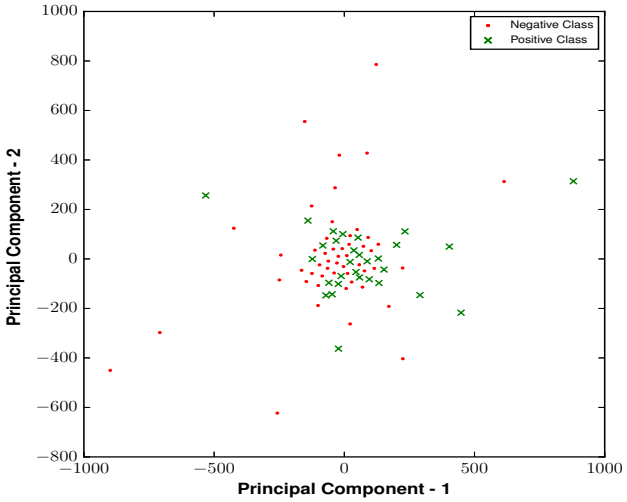


Fig. 3. Visualization of HOG feature vectors for 'helmet vs non-helmet' classification using t-SNE [15]. Red dots indicate helmet class and Green cross indicate non-helmet class [Best viewed in color]

2) Classification: After feature extraction, next step is to classify them as 'bike-riders' vs 'other' objects. Thus, this requires a binary classifier. Any binary classifier can be used here, however we choose SVM due to its robustness in classification performance even when trained from less number of feature vectors. Also, we use different kernels such as linear, sigmoid (MLP), radial basis function (RBF) to arrive at best hyper-plane.

B. Phase-II: Detection of Bike-riders Without Helmet

After the bike-riders are detected in the previous phase, the next step is to determine if bike rider is using a helmet or not. Usual face detection algorithms would not be sufficient for this phase due to following reasons : i) Low resolution poses a great challenge to capture facial details such as eyes, nose, mouth. ii) Angle of movement of bike may be at obtuse angles. In such cases, face may not be visible at all. So proposed framework detects region around head and then proceed to determine whether bike-rider is using helmet or not. In order to locate the head of bike-rider, proposed framework uses the fact that appropriate location of helmet will probably be in upper areas of bike rider. Consider $\mathbf{O}_{1/4}$ be upper one fourth part of object, and $\mathbf{B}_{1/4}$ be upper one fourth part of same object in binary, taken from background modeling step. For a moving bike, pixels in head region will have intensity of 1 i.e. white in $\mathbf{B}_{1/4}$. So, $\mathbf{B}_{1/4} \wedge \mathbf{O}_{1/4}$ gives region only around head. This step is very efficient which is reflected in our classification results for phase-II. Also, proposed approach is computationally less expensive than circular Hough transform which is used in related literature [7] [8] [16], as time complexity of logical "and" operation is $O(n)$ which is lower than $O(n^2)$ of circular Hough Transfrom [17].

1) Feature Extraction: Identified region around head of bike-rider is used to determine if bike-rider is using the helmet or not. To achieve this, similar features as used in phase-I i.e. HOG, SIFT and LBP are used. Fig. 3 visualizes the patterns for phase-II in 2-D using t-SNE [15]. The distribution of the HOG feature vectors show that the two classes i.e 'non-helmet' (Positive class shown in blue cross) and 'helmet' (Negative class shown in red dot) fall in overlapping regions which shows the complexity of representation. However, Table II shows that the generated feature vectors contain significant discriminative information in order to achieve good classification accuracy.

2) Classification: The method needs to determine if biker is violating the law i.e. not using helmet. For this purpose, we consider two classes : i) Bike-rider not using helmet (Positive Result), and ii) Biker using helmet (Negative Result). The support vector machine (SVM) is used to classify using extracted features from previous step. To analyze the classification results and identify the best solution, different combination of features and kernels are used. Results along with analysis is included in *Result* section.

C. Consolidation of Results

From earlier phases, we obtain local results i.e. whether bike rider is using helmet or not, in a frame. However, till now the correlation between continuous frames is neglected. So, in order to reduce false alarms, we consolidate local results. Consider y_i be label for i^{th} frame which is either +1 or -1. If for past n frames, $\frac{1}{n}\sum_{i=1}^{n}(y_i = 1) > T_f$, then framework triggers violation alarm. Here T_f, is threshold value which is determined empirically. In our case, the value of $T_f = 0.8$ and $n = 4$ were used. A combination of independent local results from frames is used for final global decision i.e. biker is using or not using helmet.

Fig. 4. Sample frames from dataset

IV. Experiments and Results

For purpose of related experiments, standalone Linux machine with specifications Intel Xeon(R) CPU E5620@ 2.40GHz x 8 was used. In our experiments, we used OpenCV 3.0 and scikit-learn 0.16 [18].

A. Dataset Used

As there is no public data set available for this purpose, we collected our own data from the surveillance system at Indian Institute of Technology Hyderabad. Here, we collected 2 hour surveillance data with frame rate of 30 fps. We used 1^{st} hour video for training the model and remaining for testing. Training video contain 42 bikes, 13 cars and 40 humans. Whereas, testing video contain 63 bikes, 25 cars and 66 humans.

B. Results and Discussion

In this section, we present experimental results and discuss the suitability of the best performing representation and model over the others. Table. I presents results for bike-rider detection using different features viz; HOG, SIFT, LBP and kernels viz; linear, sigmoid (MLP), radial basis function (RBF). In order to validate the performance of each combination of representation and model, we conducted experiments using 5-fold cross validation. The experimental results in Table I show that average performance of classification using SIFT and LBP is almost similar. Also, the performance of classification using HOG with MLP and RBF kernels is similar to the performance of SIFT and LBP. However, HOG with *linear* kernel performs better than all other combinations, because feature vector for this representation is sparse in nature which is a suitable for linear kernel. Table I displays the accuracy of detecting a bike-rider in a frame.

Table II presents results for detection of bike-rider with or without helmet using different features viz; HOG, SIFT, LBP and kernels viz; linear, MLP, RBF. In order to validate the performance of each combination of representation and model, we conducted experiments using 5-fold cross validation. From Table II we can observe that average performance of classification using SIFT and LBP is almost similar. Also, the performance of classification using HOG with MLP and RBF kernel is similar to the performance of SIFT and LBP.

TABLE I
Performance of phase-I classification (%) of Detection of Bike-rider

Feature	Kernel	Fold1	Fold2	Fold3	Fold4	Fold5	Avg.
HOG	*Linear*	97.93	99.59	98.35	99.38	99.17	**98.88**
	MLP	80.99	80.99	84.30	84.71	83.47	82.89
	RBF	80.99	80.99	84.30	84.71	83.47	82.89
SIFT	*Linear*	80.79	84.30	83.68	83.47	82.23	**82.89**
	MLP	80.79	84.30	83.68	83.47	82.23	82.89
	RBF	80.79	84.30	83.68	83.47	82.23	82.89
LBP	*Linear*	82.64	84.71	81.61	82.44	83.06	**82.89**
	MLP	82.64	84.71	81.61	82.44	83.06	82.89
	RBF	82.64	84.71	81.61	82.44	83.06	82.89

However, HOG with linear kernel performs better than all other combinations.

TABLE II
Performance of phase-II classification (%) of 'bike-rider with helmet' vs 'bike-rider without helmet'

Feature	Kernel	Fold1	Fold2	Fold3	Fold4	Fold5	Avg.
HOG	*Linear*	90.12	95.06	93.83	95.00	95.00	**93.80**
	MLP	62.96	67.90	70.37	61.25	60.00	64.50
	RBF	62.96	67.90	70.37	61.25	60.00	64.50
SIFT	*Linear*	67.90	60.49	66.67	62.50	65.00	**64.51**
	MLP	67.90	60.49	66.67	62.50	65.00	64.51
	RBF	67.90	60.49	66.67	62.50	65.00	64.51
LBP	*Linear*	64.20	60.49	64.20	67.50	66.25	**64.53**
	MLP	64.20	60.49	64.20	67.50	66.25	64.53
	RBF	64.20	60.49	64.20	67.50	66.25	64.53

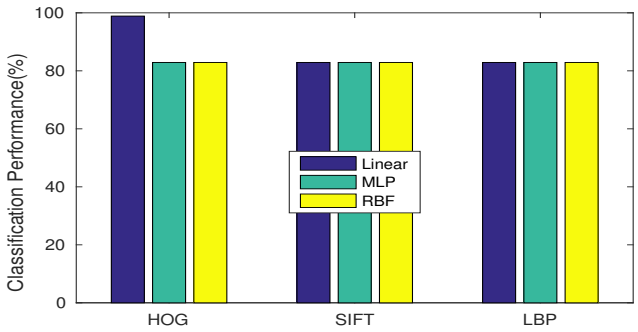


Fig. 5. Performance comparison of classification (%) of 'bike-riders' vs. 'others' in phase-I for different features and kernels.

From the results presented in Table I & Table II, it can be observed that using HOG descriptors helps in achieving best performance. Fig. 7 & Fig. 8 presents ROC curves for performance of classifiers in detection of bike-riders and detection of bike-riders with or without helmet, respectively. Fig. 7 clearly shows that the accuracy is above 95% with a low false alarm rate less than 1% and area under curve (AUC) is 0.9726. Similarly, Fig. 8 clearly shows that the accuracy is above 90% with a low false alarm rate less than 1% and AUC is 0.9328.

C. Computational Complexity

To test the performance, a surveillance video of around one hour at 30 fps i.e. 107500 frames was used. The pro-

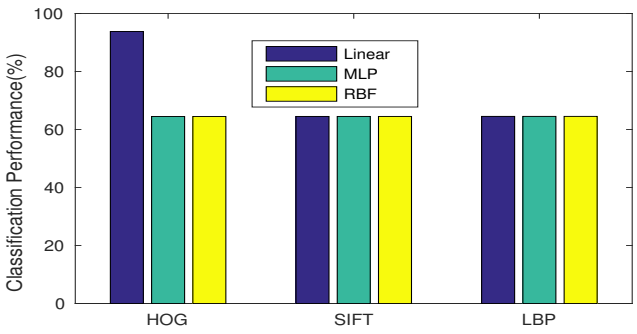


Fig. 6. Performance of phase-II classification (%) of 'bike-rider with helmet' vs 'bike-rider without helmet' for different features and kernels.

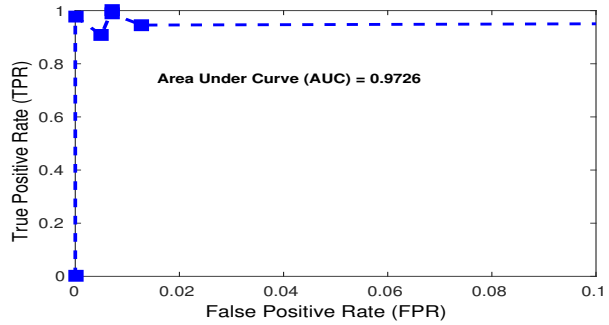


Fig. 7. ROC curve for classification of 'bike-riders' vs. 'others' in phase-I showing high area under the curve

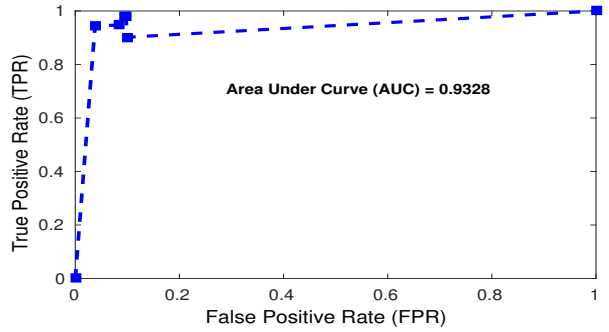


Fig. 8. ROC curve for classification of 'bike-rider with helmet' vs. 'bike-rider without helmet' in phase-II showing high area under the curve

posed framework processed the full data in 1245.52 secs i.e. 11.58 ms per frame. However, frame generation time is 33.33 ms, so the proposed framework is able to process and return desired results in real-time.

Result included in section IV(B) shows that accuracy of proposed approach is either better or comparable to related work presented in [5] [7] [16] [8].

V. Conclusion

In this paper, we propose a framework for real-time detection of traffic rule violators who ride bike without using helmet. Proposed framework will also assist the traffic police for detecting such violators in odd environmental conditions viz; hot sun, etc. Experimental results demonstrate the accuracy of 98.88% and 93.80% for detection of bike-rider and detection of violators, respectively. Average time taken to process a frame is 11.58 ms, which is suitable for real time use. Also, proposed framework automatically adapts to new scenarios if required, with slight tuning. This framework can be extended to detect and report number plates of violators.

References

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 555–560, March 2008.

[2] B. Duan, W. Liu, P. Fu, C. Yang, X. Wen, and H. Yuan, "Real-time on-road vehicle and motorcycle detection using a single camera," in *Procs. of the IEEE Int. Conf. on Industrial Technology (ICIT)*, 10-13 Feb 2009, pp. 1–6.

[3] C.-C. Chiu, M.-Y. Ku, and H.-T. Chen, "Motorcycle detection and tracking system with occlusion segmentation," in *Int. Workshop on Image Analysis for Multimedia Interactive Services*, Santorini, June 2007, pp. 32–32.

[4] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, Aug 2004.

[5] J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," *Intelligent Transport Systems (IET)*, vol. 6, no. 3, pp. 259–269, September 2012.

[6] Z. Chen, T. Ellis, and S. Velastin, "Vehicle detection, tracking and classification in urban traffic," in *Procs. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITS)*, Anchorage, AK, Sept 2012, pp. 951–956.

[7] R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras, and A. Soares, "Automatic detection of motorcyclists without helmet," in *Computing Conf. (CLEI), XXXIX Latin American*, Oct 2013, pp. 1–7.

[8] R. Rodrigues Veloso e Silva, K. Teixeira Aires, and R. De Melo Souza Veras, "Helmet detection on motorcyclists using image descriptors and classifiers," in *Procs. of the Graphics, Patterns and Images (SIBGRAPI)*, Aug 2014, pp. 141–148.

[9] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proc. of the Int. Conf. on Pattern Recognition (ICPR)*, vol. 2, Aug.23-26 2004, pp. 28–31.

[10] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 1999, pp. 246–252.

[11] "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62–66, Jan 1979.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Procs. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2005, pp. 886–893.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] Z. Guo, D. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, June 2010.

[15] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[16] R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, "Machine vision techniques for motorcycle safety helmet detection," in *Int. Conf. of Image and Vision Computing New Zealand (IVCNZ)*, Nov 2013, pp. 35–40.

[17] D. Ioannou, W. Huda, and A. F. Laine, "Circle recognition through a 2d hough transform and radius histogramming," *Image and vision computing*, vol. 17, no. 1, pp. 15–26, 1999.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Visvesvaraya Technological University, Belagavi.

PROJECT REPORT
on

"INTELLIGENT SURVEILLANCE SYSTEM FOR RIDERS WITHOUT HELMET AND TRIPLE RIDING DETECTION ON TWO WHEELERS"

Project Report submitted in partial fulfillment of the requirement for the award of the degree of
Bachelor of Engineering
in
Electronics and Communication Engineering
For the academic year 2019-2020

Submitted by

1CR16EC020	APOORVA SAUMYA
1CR16EC042	GAYATHRI V
1CR16EC153	SARTHAK KALE

Under the guidance of

Internal
PROF. K VENKATESWARAN
Associate Professor
Department of ECE
CMRIT, Bangalore

Department of Electronics and Communication Engineering
CMR Institute of Technology, Bengaluru – 560 037

DEPARTMENT OF ELECTRONICS & COMMUNICATION

CERTIFICATE

This is to certify that the internship work entitled **"Intelligent Surveillance System For Riders Without Helmet And Triple Riding Detection On Two Wheelers"** is carried out by **Apoorva Saumya, Gayathri V, Sarthak kale** bearing USN 1CR16EC020, 1CR16EC042, 1CR16EC153 bonafide students of **CMR INSTITUTE OF TECHNOLOGY** in partial fulfillment for the award of **Bachelor of Engineering** in **Electronics and Communication Engineering** from **Visvesvaraya Technological University, Belagavi** during the academic year **2019-2020**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The Internship Report has been approved and satisfies the academic requirements with respect to internship work prescribed for the said degree.

Signature of Guide	Signature of HOD	Signature of Principal
……	…….	…….
K. Venkateswaran Asst. Professor, Dept. of ECE, CMRIT, Bangalore	**Dr. R. Elumalai** Head of the Department Dept. of ECE, CMRIT, Bangalore	**Dr. Sanjay Jain** Principal, CMRIT, Bangalore

External Viva:

Internal Evaluator	External Evaluator
1 …….	…….
2 …….	……

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose consistent guidance and encouragement crowned our efforts with success.

We consider it as our privilege to express the gratitude to all those who guided in the completion of the project.

We express my gratitude to Principal, **Dr. Sanjay Jain,** for having provided me the golden opportunity to undertake this project work in their esteemed organization.

We sincerely thank **Dr. R. Elumalai**, HOD**,** Department of Electronics and Communication Engineering, CMR Institute of Technology for the immense support given to me.

We express our gratitude to our project guide **Prof. K. Venkateswaran**, Associate Professor, Department of Electronics and Communication Engineering, CMR Institute of Technology for their support, guidance and suggestions throughout the project work.

Last but not the least, heartfelt thanks to our parents and friends for their support.

Above all, we thank the Lord Almighty for His grace on us to succeed in this endeavor.

TABLE OF CONTENTS

ABSTRACT

The rate at which the number of two wheelers in India is rising is 20 times the rate at which human population is growing. The risk of death is 2.5 times more among riders not wearing a helmet compared with those wearing a helmet. The existing video surveillance-based system is effective but it requires significant human assistance whose efficiency decreases with time and human biasing also comes into the picture. This project aims to solve this problem by automating the process of detecting the riders who are riding without helmets. The system takes a video of traffic on public roads as the input and detects moving objects in the scene. A machine learning classifier is applied to the moving object to identify if the moving object is a two-wheeler. If it is a two-wheeler, then another classifier is used to detect whether the rider is wearing a helmet. So, wearing headgear is critical to decrease the danger of injuries in the event that mishap happens. This work proposes a system for location of individual or different riders taking a trip on bikes with no helmets. Inside the proposed approach, from the beginning stage, bike riders are recognized with the use of YOLOv3 model which is a consistent type of YOLO model, the forefront methodology for object distinguishing helps as such in distinguishing the riders with and without helmet. The vertical projection of binary image is used for counting the number of riders if it exceeds two.

Chapter 1

INTRODUCTION

Two-wheeler is a very popular mode of transportation in almost every country. However, there is a high risk involved because of less protection. To reduce the involved risk, it is highly desirable for bike-riders to use a helmet.

Two-Wheelers account for the greatest number of road accidents. Though careless and rash driving is the main cause of these accidents, head injuries form a single largest reason for the road accident deaths. Study shows that more than one- third who died in road accidents could have survived if they would have worn a helmet, the usage of helmet can save accident deaths by 30 to 40%.

Number of road accidents due to bike riders without helmets has been alarming. A Delhi Police annual report (released in 2017) revealed that of the total number of fatal accidents in the city in 2016, 35-40 percent of the deaths were due to riders “not wearing helmets" or “poor quality helmets”. It is compulsory for two-wheeler riders to wear safety helmets under Section 129 of the Motor Vehicles Act, 1988. The rule also says that a helmet should have a thickness of 20-25 mm, with quality foam. It should also have an ISI mark and follow the Bureau of Indian Standards.

But unfortunately, no one seems to follow these rules, at least not for the pillion riders. These days video Surveillance based systems have turned into a significant gear to remain a track on any very crook or hostile to law movement in current human advancement. There are existing methods which use specialized sensors in the ergonomics of the motorbike to check the presence of a helmet. But it is impossible to convince every user to install sensors on the already existing bikes. Also, the accuracy and integrity of these sensors is questionable. Apart from this, systems that use video processing have very high computational costs. The technologies that were used to build the system were very expensive hence making it an economically non-viable choice.

Considering this present, there's an expanding request to build up a solid and convenient proficient system for identifying helmet utilization of motorbike riders that doesn't accept an individual's spectator. A promising strategy for accomplishing this mechanized recognition of motorbike helmet use is AI. AI has been applied to an assortment of street wellbeing related discovery undertakings, and has accomplished high precision for the general recognition. To date several researchers have tried to tackle the problem of detection of motorcyclists without helmets by using different methods but have not been able to accurately identify motorcyclists without helmets under challenging conditions such as occlusion, illumination, poor quality of video, varying weather conditions, etc. One major reason for the poor performance of existing methods is the use of less discriminative representation for object classification as well as the consideration of irrelevant objects against the objective of detection of motorcyclists without helmets. Also, the existing approaches make use of handcrafted features only. Deep networks have gained much attention with state-of-the-art results in complicated tasks such as image classification, object recognition, tracking detection and segmentation due to their ability to learn features directly from raw data without resorting to manual tweaking.

Nowadays video surveillance-based systems have become an essential equipment to keep a track on any kind of criminal or anti law activity in modern civilization.

Over the past decades, some artificial intelligent techniques like computer vision and machine learning with growing progress have been widely applied in intelligent surveillance in power substations. It can not only avoid time consuming labor intensive tasks, but also point out the power equipment fault and worker illegal operation in time and accurately against accidents.

However, the existing video surveillance-based methods are passive and require significant human assistance. In general, such systems are infeasible due to involvement of humans, whose efficiency decreases over long duration. Automation of this process is highly desirable for reliable and robust monitoring of these violations as well as it also significantly reduces the amount of human resources needed. Also, many countries are adopting systems involving surveillance cameras at public places. So, the solution for detecting violators using the existing infrastructure is also cost-effective.

However effective automatic surveillance systems generally involve following tasks: environment modelling, detection, tracking and classification of moving objects.

In order to adopt such automatic solutions certain challenges, need to be addressed:

1) Real-time Implementation: Processing significant amounts of information in a time Constraint manner is a challenging task. As such applications involve tasks like segmentation, feature extraction, classification and tracking, in which a significant amount of information needs to be processed in short duration to achieve the goal of real-time implementation.

2) Occlusion: In real life scenarios, the dynamic objects usually occlude each other due to which object of interest may only be partially visible. Segmentation and classification become difficult for these partially visible objects.

3) Direction of Motion: 3-dimensional objects in general have different appearance from different angles. It is well known that accuracy of classifiers depends on features used which in turn depends on angle to some extent. A reasonable example is to consider the appearance of a bike-rider from front view and side view.

4) Temporal Changes in Conditions: Over time, there are many changes in environment conditions such as illumination, shadows, etc. There may be subtle or immediate changes which increase complexity of tasks like background modelling.

5) Quality of Video Feed: Generally, CCTV cameras capture low resolution video.Also, conditions such as low light, bad weather complicate it further. Due to such limitations, tasks such as segmentation, classification and tracking become even more difficult. As stated in , a successful framework for surveillance application should have useful properties such as real-time performance, fine tuning, robust to sudden changes and predictive. Keeping these challenges and desired properties in mind, we propose a method for automatic detection of bike-riders without helmets using feed from existing security cameras, which works in real time.

The prevailing video statement primarily based on strategies is passive and wants important human help. Automation of this procedure is exceedingly appealing for energetic commentary of this infringement and additionally it likewise altogether lessens the measure of human resource required. Available strategies utilize particular sensors inside the ergonomics of the motorbike to see the existence of a helmet. Be that as it may, it's unrealistic to persuade each individual to place sensors on the effectively present motorbikes. Likewise, the exactness and honesty of those sensors may be flawed. Aside from this, structures that use video processing have very excessive computational prices. The technology that had been wont to build the device is very high-priced consequently building that with inexpensively infeasible preference.

1.1 Challenges Faced

1.1.1 Real-time Implementation: Processing critical measure of data in a period imperative way is a test undertaking. All things considered applications include assignments like segmentation, feature extraction, classification and tracking, in which a lot of data should be prepared in a brief term to accomplish the objective of ongoing usage.

1.1.2 Occlusion: In real life scenarios, the dynamic objects usually occlude each other due to which object of interest may only be partially visible.

1.1.3 Temporal Changes in Conditions: Over time, there are numerous progressions in environmental conditions, for example, light, shadows, and so forth.

1.1.4 Quality of Video Feed: Generally, CCTV cameras catch low resolution video input. Likewise, environmental conditions such as low light, hazy climate may deteriorate it further.

1.2 YOLO (You Only Look Once)

YOLO is an effective real-time object recognition algorithm. Image classification is one of the many exciting applications of convolutional neural networks. Aside from simple image classification, there are plenty of fascinating problems in computer vision, with object detection being one of the most interesting. It is commonly associated with self-driving cars where systems blend computer vision, LIDAR and other technologies to generate a multidimensional representation of the road with all its participants. Object detection is also commonly used in video surveillance, especially in crowd monitoring to prevent terrorist attacks, count people for general statistics or analyse customer experience with walking paths within shopping centres.
YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. First, YOLO is extremely fast. Since we frame detection as a regression problem we don’t need a complex pipeline. We simply run our neural network on a new image at test time to predict detections.

Two types of YOLO algorithm

1. Algorithms based on classification: They are implemented in two stages. First, they select regions of interest in an image. Second, they classify these regions using convolutional neural networks. This solution can be slow because we have to run predictions for every selected region. A widely known example of this type of algorithm is the Region-based convolutional neural network (RCNN) and its cousins Fast-RCNN, Faster-RCNN and the latest addition to the family: Mask-R CNN. Another example is Retina Net.

2. Algorithms are based on regression: instead of selecting interesting parts of an image, they predict classes and bounding boxes for the whole image in one run of the algorithm. The two best known examples from this group are the YOLO (You Only Look Once) family algorithms and SSD (Single Shot Multibox Detector). They are commonly used for real-time object detection as; in general, they trade a bit of accuracy for large improvements in speed.

How does YOLO work?

1. Objects are detected by a combination of **object locator** and an **object recognizer.**
2. YOLO approaches the object detection problem in a completely different way. It forwards the whole image **only once** through the network.
3. First, it divides the image into a 13×13 grid of cells. The size of these 169 cells vary depending on the size of the input.
4. For each bounding box, the network also predicts the confidence that the bounding box actually encloses an object, and the probability of the enclosed object being a particular class.
5. Most of these bounding boxes are eliminated because their confidence is low or because they are enclosing the same object as another bounding box with a very high confidence score. This technique is called non-maximum suppression.

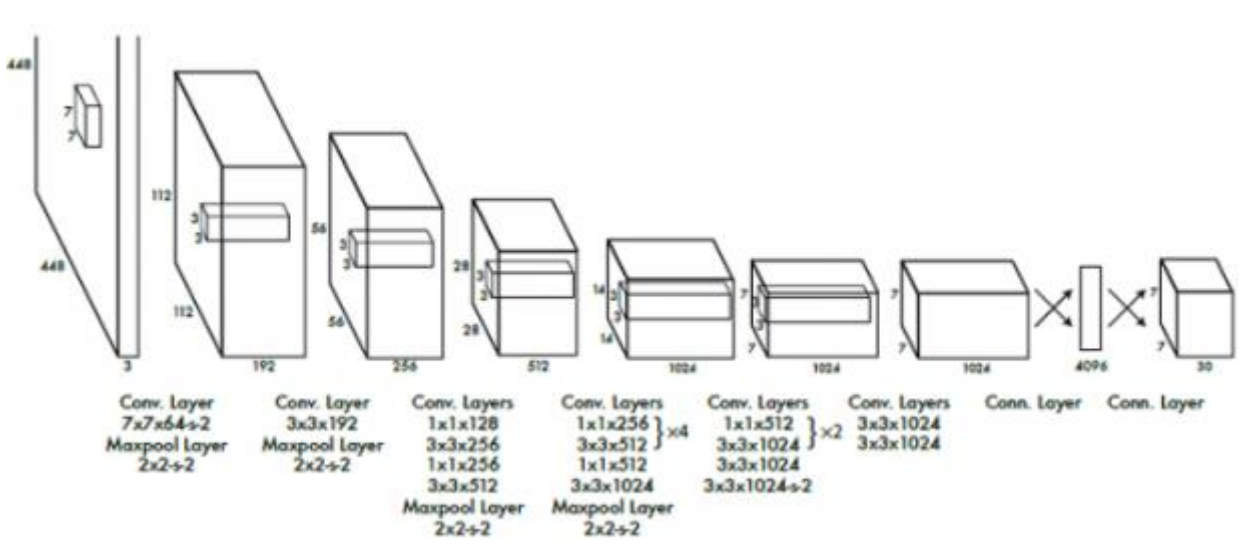


Figure 1 Working of YOLO algorithm

1.3 MACHINE LEARNING

The basic process of machine learning is to give training data to a learning algorithm. The learning algorithm then generates a new set of rules, based on inferences from the data. This is in essence generating a new algorithm, formally referred to as the machine learning model.

Instead of programming the computer every step of the way, this approach gives the computer instructions that allow it to learn from data without new step-by-step instructions by the programmer. This means computers can be used for new, complicated tasks that could not be manually programmed Machine learning, a particular approach to AI and the driving force behind recent developments. Instead of programming the computer every step of the way, machine learning makes use of learning algorithms that make inferences from data to learn new tasks.

As machine learning is used more often in products and services, there are some significant considerations when it comes to users' trust in the Internet. Several issues must be considered when addressing AI, including, socio-economic impacts; issues of transparency, bias, and accountability; new uses for data, considerations of security and safety, ethical issues; and, how AI facilitates the creation of new ecosystems.

At the same time, in this complex field, there are specific challenges facing AI, which include: a lack of transparency and interpretability in decision-making; issues of data quality and potential bias; safety and security implications; considerations regarding accountability; and, its potentially disruptive impacts on social and economic structures. Machine Learning is simply making a computer perform a task without explicitly programming it. In today's world every system that does well has a machine learning algorithm at its heart. Take for example Google Search engine, Amazon Product recommendations, LinkedIn, Facebook etc, all these systems have machine learning algorithms embedded in their systems in one form or the other. They are efficiently utilising data collected from various channels which helps them get a bigger picture of what they are doing and what they should do.

1.4 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

Advantages:

1. easy to learn and use
2. python is broadly adopted and supported
3. python is not a toy language
4. open source with vibrant community
5. extensive support libraries

Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning and AI, Python's simplicity allows developers to write reliable systems. Python code is understandable by humans, which makes it easier to build models for machine learning.

Python is a widely used high-level programming language for general-purpose programming. Apart from being an open source programming language, python is a great object-oriented, interpreted, and interactive programming language. Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high-level dynamic data types, and dynamic typing. There are interfaces to many systems calls and libraries, as well as to various windowing systems. New built-in modules are easily written in C or C++ (or other languages, depending on the chosen implementation). Python is also usable as an extension language for applications written in other languages that need easy-to-use scripting or automation interfaces.

Python is widely considered as the preferred language for teaching and learning Ml (Machine Learning). Few simple reasons are:

- It’s simple to learn. As compared to c, c++ and Java the syntax is simpler and Python also consists of a lot of code libraries for ease of use.
- Though it is slower than some of the other languages, the data handling capacity is great.
- Open Source! – Python along with R is gaining momentum and popularity in the Analytics domain since both of these languages are open source.
- Capability of interacting with almost all the third party languages and platforms.

OpenCV is used for all sorts of image and video analysis, like facial recognition and detection, license plate reading, photo editing, advanced robotic vision, optical character recognition, and a whole lot more. All this under one library of Python making python fan favourite.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in the background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

Chapter 2

LITERATURE SURVEY

Automatic identification of bicycle riders without headgear falls under general class of anomaly recognition in video recordings. Effective detection system framework include following errands: environmental modeling, detection, tracking and classification of moving objects. Chiverton proposed an approach which utilizes geometrical state of headgear and illumination difference at various bits of the headgear. It utilizes circle arc discovery strategy in view of the Hough transform. The major constraint of this approach is that it tries to find headgear in the full frame which is computationally costly and furthermore it might frequently confound other comparable modeled objects as headgear. Additionally, it manages the reality that headgear is applicable just if there should arise an occurrence of bicycle riders. Chen et al. proposed an effective way to distinguish and track vehicles in urban rush hour gridlock. It utilizes Gaussian mixture model along with a system to refine foreground blob keeping in mind the end goal to remove foreground. It tracks a vehicle utilizing Kalman filter and refines classification utilizing dominant part voting. There is a proposed circular arc detection method based on the modified Hough transform. This transformation has been applied to detect a helmet by an ATM surveillance system.

Comprehensively, there are two noteworthy constraints in the current work discussed above. Firstly, recommended approaches are either computationally extremely costly or passive in nature which are not reasonable for ongoing execution. Furthermore, the relationship between the frames is underutilized for final decisions , as the outcomes from back to back frames can be joined to raise more reliable cautions for infringement. The proposed approach overcomes the above examined constraints by giving an effective solution which is suitable for real time application.

Automatic identity of motorbike-riders without helmets in surveillance videos comes underneath the specific category of anomaly detection. As described in [15], powerful automated surveillance generally consists of the subsequent responsibilities: modeling, identity, tracking and type of transferring items within the environment. Chiverton recommended a technique in [16] that uses the geometric form of the helmet and the variant of lights at various parts of the helmet. This uses a way for detecting circle arcs primarily based on the transformation of the Hough. By this strategy the exactness was exceptionally high anyway the quantity of test pictures taken was extremely less so it wasn't a lot dependable.

The greatest weakness of this methodology is that it endeavors to find the helmet in the entire picture, which is computationally expensive, and it can also confuse more fashionable gadgets like helmets. It additionally supervises the fact that helmets are best appropriate only for cyclists.

Doughmala et al.[8] provides a half and full helmet spotting recognition through Haar with abilities like nose, ear, mouth, left eye, appropriate eye and roundabout Hough rebuild to run over helmet presence. In any case, all through this paper it is chipped away to fix resolution of the pictures.
In Dahiya et al. [2], detection of two-wheeler riders without helmet has been developed using real time videos and applied (HOG) Histogram of Oriented Gradients, (SIFT) Scale invariant feature transform, (LBP) Local binary pattern. Through this technique, the detection accuracy was 93.80% however the time interval needed was not speedy at the rate of 11.58 per frame [2]. It affords the method for actual-time detection of motorbike-

riders having no helmets which works in two levels. In the first section, a motorcycle-rider within the video frame is detected. In the second phase, the head of the motorbike-rider is detected and checks if the rider is with or without a helmet, so as to scale back false predictions. Even being less expensive than other previous works, this takes a lot of time implementing the pre-processing techniques as HOG, SIFT, SVM which makes the process slow.

[12] talks about a few systems which are fundamentally the same as the one proposed during this paper, distinguishes bike riders without helmets and catches the sum plate of the considerable number of guilty parties on a COCO database. It characterizes engine bikes and helmets utilizing YOLO and in this manner the innovation utilized for license plate recognition is Open ALPR. Both of those technologies charge monthly fees and hence aren't economically feasible. In rush hour gridlock video, division of motorbike riders utilizing a foundation deduction strategy followed classification using Support Vector Machine (SVM) is proposed by Chiverton et al. [16]. Li et al [3] have applied Histogram of Oriented Gradients (HOG) based absolutely including extraction finished SVM for security helmet discovery. Although operating with nice results, the accuracy and speed of detection was very slow. The classifier additionally gave wrong effects occasionally due to the one of a kind depth of light which had to be corrected.

A local Binary Pattern based cross breed descriptor for features extraction is proposed by Silva et al. [7]. They additionally utilized HOG and Hough Transform descriptors for robotized distinguishing proof of helmet-less motorcyclists. But comparative slower results were obtained. Over the previous decades, some counterfeit canny strategies like PC vision and AI with developing advancement are broadly applied in smart observation in power substations. It can't just maintain a strategic distance from tedious work escalated assignments, yet in addition implies the office gear issue and specialist unlawful activity in time and precisely against mishaps [3].

In rush hour traffic recordings as caught by observation cameras, rapid vehicles don't show up obviously in outlines. So location of rapid bikes in continuously observation video might be difficult work. Furthermore, various shapes, hues and size of bikes additionally make the identification procedure generally complex [1]. In this way a solid system is required to distinguish the defaulters.Above-mentioned researches mainly focus on the power equipment fault detection and state recognition. Aside from equipment safety, intelligent surveillance systems still need to monitor operator safety work. The real time safety helmet wearing detection for perambulatory workers, as a most common safe operation situation in power substation, is a considerably important task related to worker safety. Thus it is necessary to develop a system for automatic detection of safety helmet wearing power substation. Unfortunately, the related work is little and mostly has been made in the detection of motorcyclists with or without helmets.

Wen et al. [18] suggested a circle arc detection method based on Hough transform. They applied it to detect the presence of a helmet on the surveillance system of Automatic Teller Machine. But the drawback of this work was it has used only the geometric features to detect the presence of a helmet. Geometric features are not enough to detect the presence of a helmet; many times the head can be mistaken with the helmet. In Chiu et al. It has used a computer vision based system which aims to detect and segment motorcycles partly occluded by another vehicle. Helmet detection system was used in which the presence of a helmet simplifies that there is a motorcycle. In this paper to detect the helmet edges were computed of the possible helmet region.

Chiverton et al. [16] described and tested a system for automatic classification of motorcycles with and without helmets. It has used (SVM) Support Vector Machine which is trained on (HOG) Histogram of

Oriented Gradients which is derived from the head region of the static images and individual image frame from video data. By this method the accuracy rate was high but the number of testing images taken were very less.

Silva et al. proposed a system for detection of helmet which first starts with moving object segmentation using descriptors then detection of helmet tracing the (ROI) Region of interest which is the head region and then classifies between helmet and non- helmet.But the disadvantage was that it uses circle Hough transform to classify between helmet and non-helmet which also results in misclassification between head and helmet as both has similar shape.[4].

Dahiya et al. proposed a system for detection of two-wheeler riders without helmet using real time videos and has applied (HOG) Histogram of Oriented Gradients, (SIFT) Scale invariant feature transform, (LBP) Local binary pattern. By this method the detection accuracy was 93.80% but the time interval required was very slow at the rate of 11.58 ms per frame [5].

Doughmala et al.[8] presents a half and full helmet wearing detection by Haar with features like nose, ear, mouth, left eye, right eye and circular Hough transform to detect helmet presence. But during this time period project it's worked on fixed resolution images.

The algorithm(YOLO) is used to extract the foreground objects in the video which is then extracted as frames.The location where the helmet can be found is found by the bounding boxes. This area is extracted and the helmet is detected using a machine learning classifier, here YOLO. The driver of the vehicle(two wheeler) is involved in a high-speed accident without wearing a helmet. It is highly dangerous and can cause death. Wearing a helmet can reduce shock from the impact and may save a life. The aim of this research work is to identify the two wheelers riders' so that they can be penalised and to also detect traffic violations such as triple riding .Motorcycles being an obvious choice and a convenient transportation mode, it has a significant contribution to road accident casualties and injuries. Despite the Government traffic regulation, people still avoid using a helmet. The proposed system is an effort to create awareness in society by endorsing the use of helmets and leading people to safety. This project proposes effective enforcement of the use of a helmet by implementing helmet detection for a rider. A system very similar to the one proposed in this paper which identifies bike riders without helmets and captures the number plate of all the offenders on a COCO database. It classifies motor bikes and helmets using YOLO and the technology used for license plate recognition is Open ALPR. Both of these technologies charge monthly fees and hence are not economically feasible.

Based on the YOLO V3 full-regression deep neural network architecture, this paper utilizes the advantage of Densenet in model parameters and technical cost to replace the backbone of the YOLO V3 network for feature extraction, thus forming the so-called YOLO-Densebackbone convolutional neural network. The test results show that the improved model can effectively deal with situations where the helmet is stained, partially occluded, or there are many targets with a low image resolution. In the test set, compared with the traditional YOLO V3, the improved algorithm detection accuracy increased by 2.44% with the same detection rate. The establishment of this model has important practical significance for improving helmet detection and ensuring safe construction.

Over the past years, multiple approaches have been proposed to solve the problem of helmet detection. The authors in [7] use a background subtraction method to detect and differentiate between moving vehicles. And they used Support Vector Machines (SVM) to classify helmets and human heads without helmets. Silva et al.

in [9] proposed a hybrid descriptor model based on geometric shape and texture features to detect motorcyclists without helmets automatically. They used a Hough transform with SVM to detect the head of the motorcyclist. Additionally, they extend their work in [10] by multilayer perceptron models for classification of various objects.

Below are the few major papers, that were referred to during the making of this project:

1. Automated Helmet Detection for Multiple Motorcycle Riders using CNN
(M. Dasgupta, O. Bandyopadhyay and S. Chatterji)

- **Abstract**: Automated detection of traffic rule violators is an essential component of any smart traffic system. In a country like India with a high density of population in all big cities, motorcycles are one of the main modes of transport. Use of a helmet can reduce the risk of head and severe brain injury of the motorcyclists in most of the motorcycle accident cases. Today violation of most of the traffic and safety rules are detected by analysing the traffic videos captured by surveillance cameras. This paper proposes a framework for detection of single or multiple riders traveling on a motorcycle without wearing helmets. In the proposed approach, at first stage, motorcycle riders are detected using the YOLOv3 model which is an incremental version of YOLO model, the state-of-the-art method for object detection. In the second stage, a Convolutional Neural Network (CNN) based architecture has

  been proposed for helmet detection of motorcycle riders. The proposed model is evaluated on traffic videos and the obtained results are promising in comparison with other CNN based approaches

- **Conclusion:** The proposed approach is to detect single or multiple riders basically all riders of a motorcycle without wearing helmets from traffic surveillance videos. First YOLOv3 model has been used for motorcyclist detection. Then, the proposed lightweight convolutional neural network detects wearing of helmet or no helmet for all motorcycle riders. The proposed model works quite well for helmet detection in different scenarios with accuracy of 96.23%. Results of helmet detection for motorcycle riders using proposed approach 500 1000 1500 2000 2500 3000 3500 4000 4500 Iteration 0.4 0.5 0.6 0.7 0.8 0.9 Precision AP=Average Precision Average precision of helmet detection on iteration basis methods and can be extended in future to detect more complicated cases of multiple riders including child riders. Further this work can be extended to even more complex scenarios of bad weather for detection of helmetless motorcyclists.

2. Automatic detection of bike riders without helmets using surveillance videos in real-time
(K. Dahiya, D. Singh and C.K .Mohan)

- **Abstract:** This paper presents a framework for automatic detection of bike-riders without helmets using surveillance videos in real time. The proposed approach first detects bike riders from surveillance video using background subtraction and object segmentation. Then it determines whether the bike-rider is using a helmet or not using visual features and binary classifiers. Also, we present a consolidation approach for violation reporting which helps in improving reliability of the proposed

approach. In order to evaluate our approach, we have provided a performance comparison of three various feature representations for classification. The experimental results show detection accuracy of 93.80% on the real world surveillance data. It has also been shown that the proposed approach is

computationally less expensive and performs in real-time with a processing time of 11.58 ms per frame.

- **Conclusion:** In this paper, we propose a framework for real-time detection of traffic rule violators who ride bikes without using a helmet. Proposed framework will also assist the traffic police for detecting such violators in odd environmental conditions viz; hot sun, etc. Experimental results demonstrate the accuracy of 98.88% and 93.80% for detection of bike-riders and detection of violators, respectively. Average time taken to process a frame is 11.58 ms, which is suitable for real time use. Also, the proposed framework automatically adapts to new scenarios if required, with slight tuning. This framework can be extended to detect and report number plates of violators.

3. Safety helmet wearing detection based on image processing and machine learning
(J. Li et al)

- **Abstract:** Safety helmet wearing detection is very essential in power substation. This paper proposed an innovative and practical safety helmet wearing detection method based on image processing and machine learning. At first, the ViBe background modelling algorithm is exploited to detect motion objects under a view of a fixed surveillant camera in power substation. After obtaining the motion region of interest, the Histogram of Oriented Gradient (HOG) feature is extracted to describe the inner human. And then, based on the result of HOG feature extraction, the Support Vector Machine (SVM) is trained to classify pedestrians. Finally, the safety helmet detection will be implemented by color feature recognition. Compelling experimental results demonstrated the correctness and effectiveness of our proposed method.

- **Conclusion:** In this paper, we have investigated a practical and novel method of safety helmets wearing detection in power substations which can real-time monitor the people whether wearing safety helmets or not. The image processing and machine learning techniques are employed in surveillance systems of power substation. Firstly, the ViBe background modelling algorithm was used to segment the moving objects under the view of the monitoring camera. This trick could filter a lot of static objects. Moveover, the histogram of oriented gradient (HOG) feature extraction and support vector machine (SVM) classifier training were implemented to achieve human location per frame. Finally, we utilized color features to recognize the safety helmet wearing situations. The overall method is verified by an amount of experiments on the surveillance video of power substation. The faster and excellent pedestrian detection algorithm and more accurate safety helmet detection strategy will be considered into our detection system frameworks.

4. YOLOv3: An Incremental Improvement
(Redmon J, Farhadi A)

- **Abstract:** We present some updates to YOLO! We made a bunch of little design changes to make it better. We also trained this new network that's pretty swell. It's a little bigger than last time but more

accurate. It's still fast though, don't worry. At 320 × 320 YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as an SSD but three times faster. When we look at the old .5 IOU mAP detection metric

YOLOv3 is quite good. It achieves 57.9 AP50 in 51 ms on a Titan X, compared to 57.5 AP50 in 198 ms by RetinaNet, similar performance but 3.8× faster.

5. Detection of Motorcyclists without Helmet in Videos using Convolutional Neural Network
(C. Vishnu, Dinesh Singh, C. Krishna Mohan and Sobhan Babu)

- **Abstract:** In order to ensure the safety measures, the detection of traffic rule violators is a highly desirable but challenging task due to various difficulties such as occlusion, illumination, poor quality of surveillance video, varying weather conditions, etc. In this paper, we present a framework for automatic detection of motorcyclists driving without helmets in surveillance videos. In the proposed approach, first we use adaptive background subtraction on video frames to get moving objects. Later convolutional neural networks (CNN) is used to select motorcyclists among the moving objects. Again, we apply CNN on the upper one fourth part for further recognition of motorcyclists driving without a helmet. The performance of the proposed approach is evaluated on two datasets, IITH Helmet 1 contains sparse traffic and IITH Helmet 2 contains dense traffic, respectively. The experiments on real videos successfully detect 92.87% violators with a low false alarm rate of 0.5% on an average and thus shows the efficacy of the proposed approach.

- **Conclusion:** The proposed framework for automatic detection of motorcyclists driving without helmets makes use of adaptive background subtraction which is invariant to various challenges such as illumination, poor quality of video, etc. The use of deep learning for automatic learning of discriminative representations for classification tasks improves the detection rate and reduces the false

  alarms resulting in a more reliable system. The experiments on real videos successfully detect ≈ 92.87% violators with a low false alarm rate of ≈ 0.50% on two real video datasets and thus shows the efficiency of the proposed approach.

Chapter 3

PROPOSED TECHNIQUES

This segment presents the proposed approach for continuous recognition of no. of bike-riders and bike riders without helmets utilizing YOLO.

3.1 You Look Only Once

YOLO is a smart convolutional neural network (CNN) for performing object detection in actual-time. The technique uses one neural network on the entire image, later splits the photograph into different areas and predicts bounding containers along with possibilities for each region. The biggest advantage of using YOLO is its pace which could be very speedy and may process 45 frames according to second.

Beside simple image characterization, there are numerous captivating troubles in PC vision, with item identity being one some of the first fascinating. It's often recognized with self-riding vehicles where systems blend PC imaginative and prescient, LIDAR and one of a kind advances to get a multidimensional portrayal of the street with each one of its members. Item discovery is commonly used in video commentary, for instance, swarm controlling, visitors light, in shopping facilities and so on.

YOLO trains on various full pictures and legitimately expands discovery execution. This particular model has loads of greater advantages over standard strategies for object popularity. In the first vicinity, YOLO is extremely brief. Since area is outlined as a relapse issue, the system need not hassle with a luxurious pipeline. The neural system is run on a substitution photograph at test time to foresee discoveries.

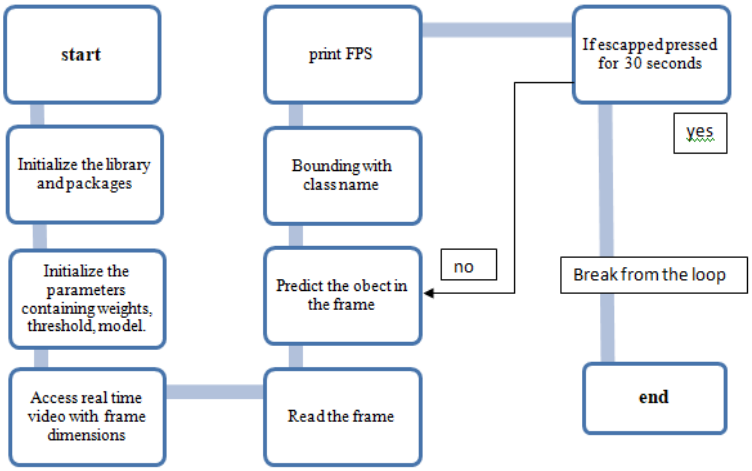


Figure 2. Workflow of YOLO scheme

Contrasted with other nearby proposition association structures (brief RCNN) which carry out place on extraordinary district hints and along those strains land up performing forecast on several occasions for one of a kind regions in a image, Yolo layout is an increasing number just like FCNN (fully convolutional neural gadget) and passes the picture (nxn) as soon as thru the FCNN and yield is (mxm) expectation. This design is parting the records image in mxm lattice and for every matrix age 2 bouncing boxes and sophistication probabilities for the ones leaping bins.

Calculations that depend on relapse, as opposed to choosing fascinating pieces of an image, they anticipate classes and leaping confines for the whole image one run of the calculation. The two most popular models from this gathering are the YOLO family calculations and SSD (Single Shot Multibox Detector). They're normally utilized for ongoing item identification as for the most part, they exchange a hint of exactness for monster enhancements in pace.

3.2 MACHINE LEARNING

The simple system of device studying is to give schooling facts to a learning algorithm. The learning algorithm then generates a brand new set of rules, primarily based on inferences from the facts. This is in essence producing a new algorithm, officially referred to as the machine mastering model.
Instead of programming the computer each step of the manner, this approach offers the device commands that allow it to study from facts without new step-with the aid-of-step commands via the programmer. This approach computer systems may be used for brand new, complex tasks that could not be manually programmed.

Instead of programming the pc every step of the way, machine learning makes use of getting to know algorithms that make inferences from facts to research new obligations.
As devices getting to know are used extra regularly in products and services, there are some vast issues when it comes to customers' agreement with the Internet. Several troubles need to be considered while addressing AI, including, socio-economic effects; troubles of transparency, bias, and accountability; new makes use of for information, considerations of protection and safety, ethical issues; and, how AI enables the advent of latest ecosystems.

At the same time, in this complicated field, there are specific demanding situations facing AI, which encompass: a loss of transparency and interpretability in selection-making; problems of information satisfactory and capability bias; protection and safety implications; concerns regarding responsib ility; and, its doubtlessly disruptive effects on social and monetary structures. Here Machine learning is used with YOLO for the detection specifically of heads, vehicles - two wheelers.

3.3 Google Collab

Collaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Google is quite aggressive in AI research. Over many years, Google developed an AI framework called TensorFlow and a development tool called Collaboratory. Today TensorFlow is open-sourced and since 2017, Google made Collaboratory free for public use. Collaboratory is now known as Google Colab or simply Colab.

Another attractive feature that Google offers to the developers is the use of GPU. Colab supports GPU and it is totally free. The reasons for making it free for the public could be to make its software a standard in the academics for teaching machine learning and data science. It may also have a long term perspective of building a customer base for Google Cloud APIs which are sold per-use basis.
Irrespective of the reasons, the introduction of Colab has eased the learning and development of machine learning applications.

Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. When you create your own Colab notebooks, they are stored in your Google Drive account. With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code. Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPU and CPU, regardless of the power of your machine.

Chapter 4

METHODOLOGY

In this paper, YOLOv3 calculates an attempt to do an image grouping to investigate the info dataset about motorcyclists with a helmet or without a helmet. Additionally, a profound learning technique for picture identification to attempt to discover a biker by not having helmet discovery from the video picture.

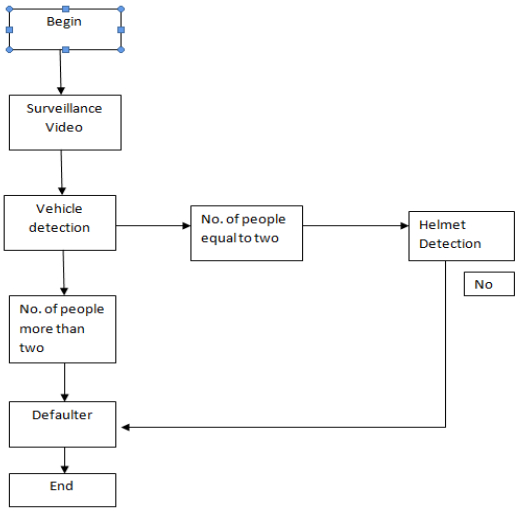
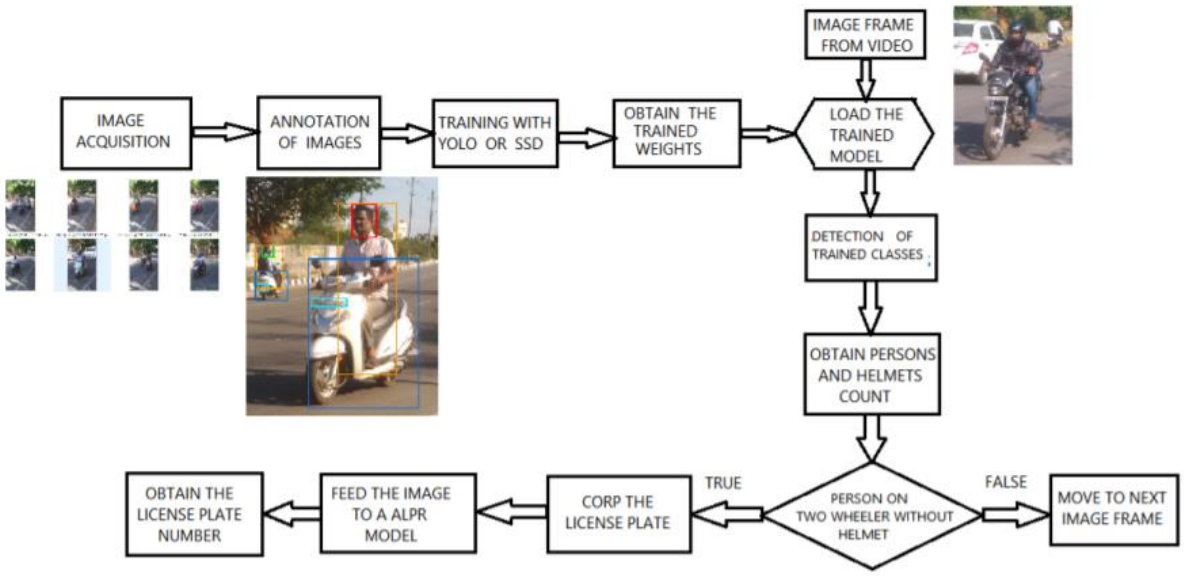


Figure 3. flow diagram of methodology

This exploration has upheld the five-advanced procedure: video and image gathering, image classification, vehicle detection and grouping, image detection examination, and interpretation of result.



1. Video and Image Gathering

The information datasets were gathered from the assets dataset of IIT Hyderabad. There are 3 unique recordings utilized. Video Dataset for Helmet Detection in Sparse Traffic from IITH Campus, in Crowded Traffic from Hyderabad City CCTV Network and Hyderabad City Video Dataset for Accident Detection from Hyderabad City CCTV Network. Images were also collected from different sources in different angles.
For training our custom object detection model, a lot of images of objects are needed which are supposed to be trained, nearly a few thousand because more number of images means more accuracy.

2. Image classification

The pictures were split into two classes in the wake of gathering 1000 pictures for the examination dataset, one for preparing information and another for test information to be utilized in grouping tests. The system has utilized a 10-crease cross approval analysis for the assessment, where a set up for various test information for 10 percent of the general picture. Preparing systems are prepared with the guide of the Python TensorFlow library; at that point exactness is measured and pick two appropriate models for use in picture recognition. Further it is processed with google collab to train the data.

For data preparation we need to use some tool to mark objects in the image. YOLO has its own format for training data.
Yolo format is:

```
<object-class> <x> <y> <width> <height>

<x_center> = ((X_end + X_start) / 2) / image_width

<y_center> = ((Y_end + Y_start) / 2) / image_height

<width> = (X_end - X_start) / image_width

<height> = (Y_end - Y_start) / image_height
```

Figure 4. Yolo format

This will help us in making the bounding boxes around the objects that we want as follows:

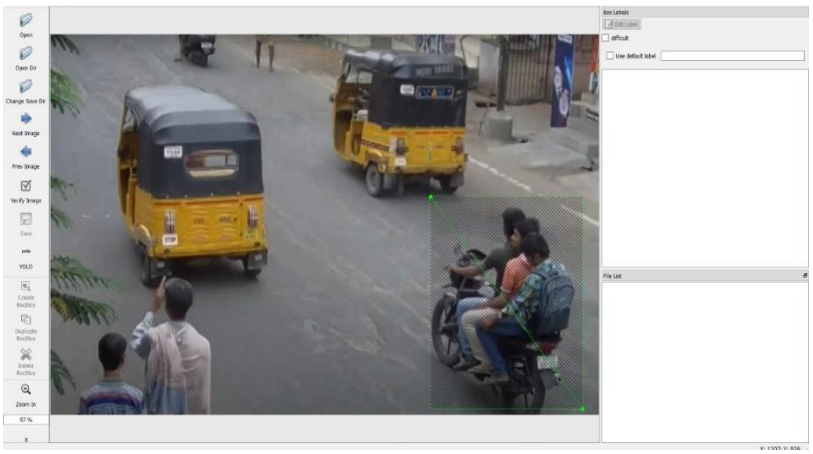


Figure 5. Labelimg tool to make bounding boxes

Setting up a platform for training data:

After having the data set, a platform is needed to train the models. There are a lot of techniques that can help in training which includes the GPU as well as CPU methods.
In this particular experiment, Cloud is used to train which is 15 times faster on GPU than CPU.
Google Colab is used in this project. Python scripts are uploaded which includes the configuration (.cfg) and object (.obj) files and the result is a trained dataset of different objects.

obj.names obj.data

```
1 classes= 1
2 train  = data/train.txt
3 valid  = data/test.txt
4 names = data/obj.names
5 backup = /mydrive/yolov3
6
```

Figure 6. obj.data file format

train.txt

```
1 data/obj/49.jpg
2 data/obj/26.jpg
3 data/obj/118.jpg
4 data/obj/157.jpg
5 data/obj/30.jpg
6 data/obj/20.jpg
7 data/obj/38.jpg
8 data/obj/74.jpg
9 data/obj/29.jpg
10 data/obj/92.jpg
11 data/obj/155.jpg
12 data/obj/131.jpg
```

Figure 7. train.txt that is generated

2) Configure yolov3.cfg file

```
[ ]  # Make a copy of yolov3.cfg
     !cp cfg/yolov3.cfg cfg/yolov3_training.cfg
```

```
[ ]  # Change lines in yolov3.cfg file
     !sed -i 's/batch=1/batch=64/' cfg/yolov3_training.cfg
     !sed -i 's/subdivisions=1/subdivisions=16/' cfg/yolov3_training.cfg
     !sed -i 's/max_batches = 500200/max_batches = 6000/' cfg/yolov3_training.cfg
     !sed -i '610 s@classes=80@classes=3@' cfg/yolov3_training.cfg
     !sed -i '696 s@classes=80@classes=3@' cfg/yolov3_training.cfg
     !sed -i '783 s@classes=80@classes=3@' cfg/yolov3_training.cfg
     !sed -i '603 s@filters=255@filters=24@' cfg/yolov3_training.cfg
     !sed -i '689 s@filters=255@filters=24@' cfg/yolov3_training.cfg
     !sed -i '776 s@filters=255@filters=24@' cfg/yolov3_training.cfg
```

3) Create .names and .data files

```
[ ]  !echo -e 'Wearing Mask\n2nd item\n3rd item' > data/obj.names
     !echo -e 'classes= 3\ntrain  = data/train.txt\nvalid  = data/test.txt\nnames = data/obj.names\nbackup = /mydrive/yolov3' > data/obj.data
```

Figure 8. Google collab notebook

3. Vehicle detection and grouping:

A. Vehicle detection

From the beginning YOLOv3 [9] designing was used for two wheelers and individual revelation. YOLOv3 model is a continuous upgrading type of YOLO obtained by J. Redmon et al [11]. The model is in a circumstance to recognize a colossal game plan of classes, among them only two classes riders and person's head are taken for disclosure. The bouncing boxes are pulled in to confine the things. The framework predicts 4 headings; bx, by are the inside bearings and bw, bh are width, height independently of the ricocheting box of estimate. The covering zone among vehicles and individuals is taken from the bounding boxes to spot whether the individual is a bike rider or not.

B. Grouping (no. of people on bike)

At that point the Euclidean Distance between the center directions of two jumping boxes of an individual are determined and in this manner the cruiser. On the off chance that the space is inside the jumping box of the bike, at that point it may be inferred that the individual is the rider of that vehicle. Utilizing this procedure, all the number of riders on a motorbike is checked. Number of people is distinguished utilizing the directions from the jumping box. First the bike is identified and inside certain arrangements focuses if the quantity of people is surpassing three then violation comes into picture. For recognizing people and vehicles, the system is using a pre-prepared model.

Opencv implementation:

The darknet implementation for detecting objects takes a lot of time to detect the object. Therefore, simple OpenCV code is implemented for it. It is much faster than darknet and can also be used for finding specific classes and finding the coordinates of detected objects.

4. Image detection experiment

In this progression, 3 recordings were gathered and were used to attempt to do a picture recognition test utilizing the YOLOv3 calculation that browsed the past advance. All recordings tried and determined the

exactness of the biker with or without the helmet and number of people recognized on the bike inside the video. Likewise tallying the quantity of undetected motorcyclists to remember for the mistake percent.

5. Interpretation of the result

In the last advance, the performance is compared with two preceding stages and made the conclusion. The exactness of the investigations will show the exhibition of the procedure as far as in terms of image classification and image detection.

The OpenCV Libraries are used alongside the detection system which contains the predefined functions and data members used for processing images like background subtraction, morphological operations, feature extraction and classification.

Chapter 5

SOFTWARE

The code for the object detection is given as follows:

```
import cv2
import numpy as np

net = cv2.dnn.readNet('yolov3_training_last.weights', 'yolov3_testing.cfg')

classes = []
with open("classes.txt", "r") as f:
    classes = f.read().splitlines()

cap = cv2.VideoCapture('test1.mp4')
font = cv2.FONT_HERSHEY_PLAIN
colors = np.random.uniform(0, 255, size=(100, 3))

while True:
    _, img = cap.read()
    height, width, _ = img.shape

    blob = cv2.dnn.blobFromImage(img, 1/255, (416, 416), (0,0,0), swapRB=True, crop=False)
    net.setInput(blob)
    output_layers_names = net.getUnconnectedOutLayersNames()
    layerOutputs = net.forward(output_layers_names)

    boxes = []
    confidences = []
    class_ids = []

    for output in layerOutputs:
        for detection in output:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.2:
                center_x = int(detection[0]*width)
                center_y = int(detection[1]*height)
                w = int(detection[2]*width)
                h = int(detection[3]*height)

                x = int(center_x - w/2)
                y = int(center_y - h/2)

                boxes.append([x, y, w, h])
                confidences.append((float(confidence)))
                class_ids.append(class_id)

    indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.2, 0.4)

    if len(indexes)>0:
        for i in indexes.flatten():
            x, y, w, h = boxes[i]
            label = str(classes[class_ids[i]])
            confidence = str(round(confidences[i],2))
            color = colors[i]
            cv2.rectangle(img, (x,y), (x+w, y+h), color, 2)
            cv2.putText(img, label + " " + confidence, (x, y+20), font, 2, (255,255,255), 2)

    cv2.imshow('Image', img)
    key = cv2.waitKey(1)
    if key==27:
        break

cap.release()
cv2.destroyAllWindows()
```

Chapter 6

DATA ANALYSIS AND RESULT

From prior stages, nearby outcomes for example regardless of whether a two-wheeler rider is using a helmet or not, at some stage in that aspect. Be that as it may, till now the association between consistent casings is dismissed. Along these lines, as to downsize bogus alerts, then merge nearby outcomes. This included first, the detection of a bike and afterwards, the individual. After the identification of the bike just the location of the helmet was done on the rider utilizing YOLO. The heads with and without helmets were separated and exhibited in various shaded bounding boxes.

In corresponding with the helmet location program, moreover the rider counter program is executed which utilizes the projection activities and lessen tasks to check the quantity of riders on the vehicle.

After calibration of the code, the outcomes acquired are shown in Fig.9, Fig.10 and Fig.11.
Objects are detected by a mixture of object locator and an object recognizer. YOLOv3 approaches the thing identification issue in a totally unique manner. It advances the whole picture only one event through the system.

In the first place, it isolates the photo into a 13×13 lattice of cells. The elements of those 169 cells shift depending on the components of the information. For each bouncing compartment, the system likewise predicts the pride that the jumping holder genuinely encases an object, and therefore the possibility of the encased thing being a specific class.

A large portion of those jumping boxes are disposed of due to their certainty which is low or in light of the fact that they're encasing a proportional item as another bouncing box with a very high certainty score. This system is named non-maximum suppression.

As it is visible from the video snippets, the algorithm used here gives a very accurate output percentage of helmet on bikers ranging from 70% - approx 90% and triple riders showcasing a very good percentage.

Figure 9 Annotation

Figure 10

Figure 11 Detecting Helmet and Non helmet bike riders

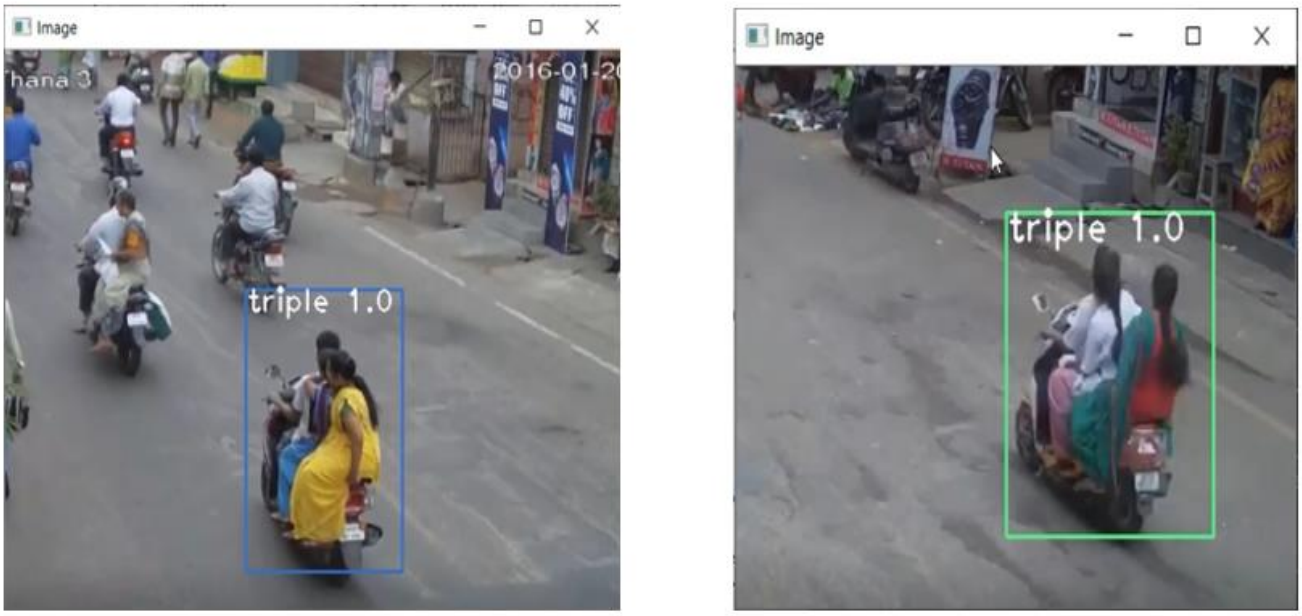


Figure 12. Detecting Triple rider

Figure 13 Detecting with different angle

Chapter 7

CONCLUSION

In this project, a system is proposed for ceaseless identification of traffic rule violators who ride motorbikes without using helmets and also defaulters, who triple ride on the vehicle. A PC vision framework that is isolated into modules like moving items division, moving articles arrangement and helmet use identification will help the traffic specialists to require activity contrary to managing violators. Proposed framework additionally will help the traffic police for such violators in odd ecological conditions like scorching sun, and so on. This framework is regularly stretched out to recognize and report number plates of violators by consolidating this method with programmed vehicle place acknowledgment frameworks by synchronizing various view cameras.

The annotated images are given as input to the YOLOv3 model to train for the custom classes. The weights generated after training are used to load the model. Once this is done, an image is given as input. The model detects all the five classes trained. From this we obtain the information regarding the person riding the motorbike. If the person is not wearing a helmet, then we can easily extract the other class information of the rider. This can further be used to extract the license plate.

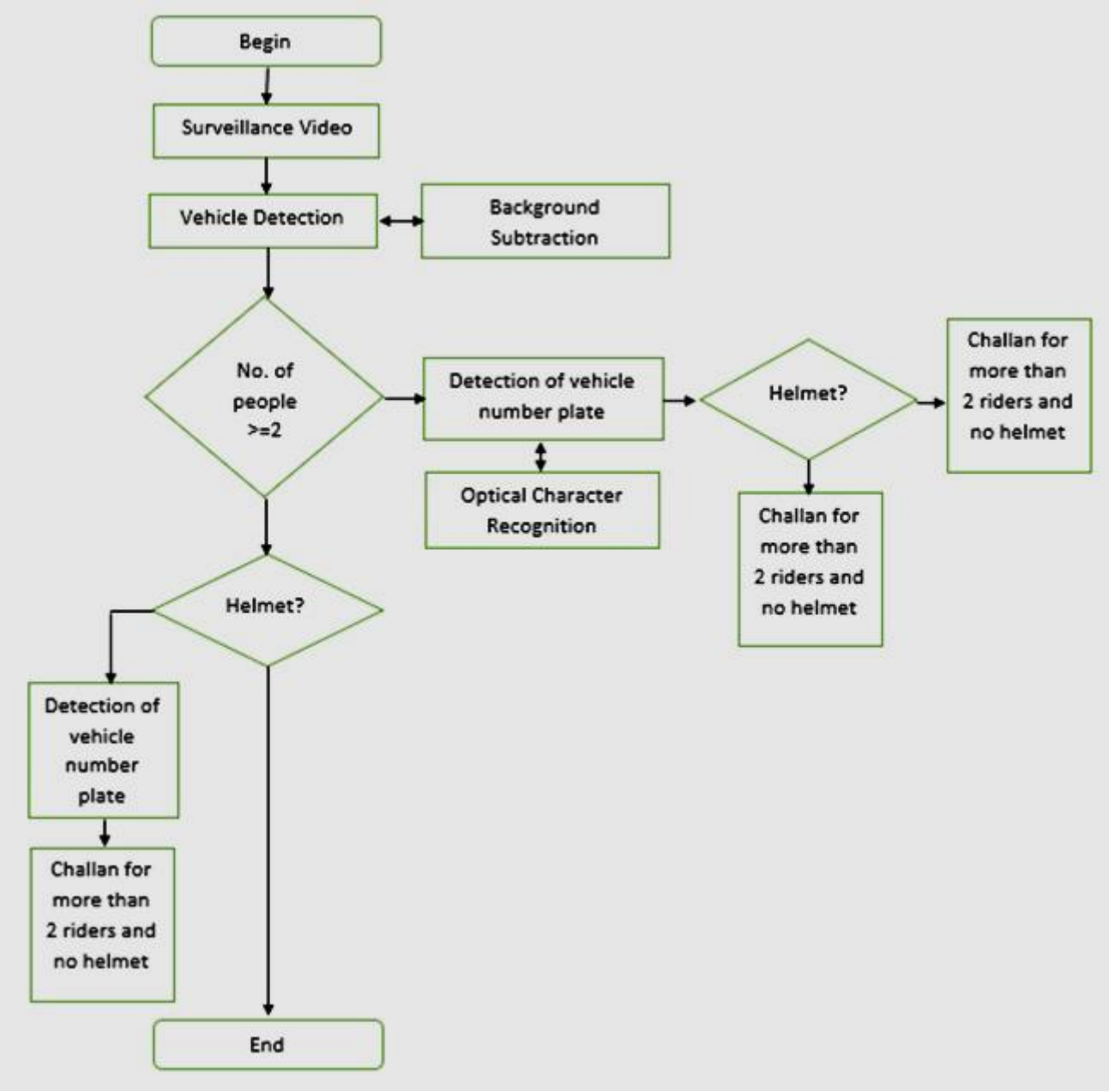
Likewise, propelled following calculations are regularly required to deal with impediment. Night-sight cameras are frequently used to utilize the location framework inside the nonattendance of light. In future, bigger quantities of positive and negative examples can be remembered for requests to expand the speculation capacity of the framework. Likewise work with front-end video catch modules.

The future work on this may include detection of license plates which will help the traffic police to automatically detect the defaulters thus sending an challan to them via SMS. This will not just ease the workload of the traffic police but also be more convenient in all aspects.

Chapter 8

FUTURE WORK

Following flow chart is proposed as a complete solution which can be implemented in future.



License plate recognition:

For the purpose of license plate recognition, we can use Automatic number-plate recognition (ANPR). It is a technology that uses optical character recognition on images to read vehicle registration plates to create vehicle location data. It can use existing closed-circuit television, road-rule enforcement cameras, or cameras specifically designed for the task. ANPR is used by police forces around the world for law enforcement purposes, including to check if a vehicle is registered or licensed. It is also used for electronic toll collection on pay-per-use roads and as a method of cataloguing the movements of traffic, for example by highways agencies.

Fine generation and text intimidation:

Once the license plate number is recognised and stored in a Data manager. The details regarding license plate can be accessed directly by cooperating with local traffic department. Once the details of the defaulter is obtained We can host an SQL based web application that can generate fine based upon the defaults and then it can generate a text and send directly to the defaulter.

When completely implemented, the solution proposed in this project can eliminate the human intervention in the process of detecting the traffic rule violators and imposing fine for their actions. It can further be delocalised by formulating an embedded system inside the surveillance cameras which will detect and directly send the fine amount to the traffic rule violators.

REFERENCES

[1] M. Dasgupta, O. Bandyopadhyay and S. Chatterji, "Automated Helmet Detection for Multiple Motorcycle Riders using CNN," IEEE Conference on Information and Communication Technology, Allahabad, India, 2019, pp.1-4.

[2] K. Dahiya, D. Singh and C.K .Mohan, "Automatic detection of bike riders without helmets using surveillance videos in real-time", in Proceeding of International Joint Conference Neural Networks (IJCNN), Vancouver, Canada, 24-2, 2016, pp.3046-3051.

[3] J. Li et al., "Safety helmet wearing detection based on image processing and machine learning," 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), Doha, 2017, pp.201-205.

[4] N. Boonsiri Sumpun, W. Puarungroj and P. Wairocana Phuttha, "Automatic Detector for Bikers with no Helmet using Deep Learning," 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2018, pp.1-4.

[5] B. Yogameena, K. Menaka and S. Saravana Perumaal, "Deep learning-based helmet wear analysis of a motorcycle rider for intelligent surveillance system," in IET Intelligent Transport Systems, vol. 13, no. 7, 2019, pp.1190-1198.

[6] Kavyashree Devadiga, Yash Gujarathi, Pratik Khanapurkar, Shreya Joshi and Shubhankar Deshpande. "Real Time Automatic Helmet Detection of Bike Riders" International Journal for Innovative Research in Science & Technology Volume 4 Issue 11, 2018, pp.146-148.

[7] R. V. Silva, T. Aires, and V. Rodrigo, " Helmet Detection on Motorcyclists using image descriptors and classifiers", in Proceeding of Graphics, Patterns and Images (SIBGRAPI) ,Rio de Janeiro ,Brazil , 27-30 August 2014, pp.141-148R.

[8] Pattasu Doughmala, Katanyoo Klubsuwan, "Half and Full Helmet Detection in Thailand using Haar Like Feature and Circle Hough Transform on Image Processing" in Proceeding of IEEE International Conference on Computer and Information Technology, Thailand, Bangkok, 2016, pp.611-614.

[9] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[10] Redmon, Joseph, and Ali Farhadi. 'YOLO9000: better, faster, stronger.' IEEE conference on computer vision and pattern recognition, 2017, pp.7263- 7271.

[11] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real time object detection [C]// Computer Vision and Pattern Recognition, 2016, pp.779-786.

[12] J. Mistry, A. K. Mishra, M. Agarwal, A. Vyas, V. M. Chudasama and K. P. Upla, "An automatic detection of helmeted and non-helmeted motorcyclists with license plate extraction using convolutional neural network," 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, 2017, pp.1-6.

[13] Manoharan, S. (2019),"An Improved Safety Algorithm For Artificial Intelligence Enabled Processors In Self Driving Cars",Journal of Artificial Intelligence, 1(02), 95-104.

[14] Jacob, I. J. (2019),"Capsule Network Based Biometric Recognition System",Journal of Artificial Intelligence, 1(02), 83-94.

[15] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 34, no. 3, 2004, pp.334–35.

[16] J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," Intelligent Transport Systems (IET), vol. 6, no. 3, september 2012, pp.259–269.

[17] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, march 2008, pp.555–560.

[18] B. Duan, W. Liu, P. Fu, C. Yang, X. Wen, and H. Yuan, "Real-time on road vehicle and motorcycle detection using a single camera," Procs. of the IEEE Int. Conf. on Industrial Technology (ICIT), 10-13 Feb 2009, pp.1–6.

applied sciences

MDPI

Article

A Super-Resolution Reconstruction Driven Helmet Detection Workflow

Yicheng Liu [1], Zhipeng Li [1], Bixiong Zhan [2], Ju Han [2] and Yan Liu [1,*]

[1] College of Electrical Engineering, Sichuan University, Chengdu 610065, China; liuyicheng@scu.edu.cn (Y.L.); 2020223035121@stu.scu.edu.cn (Z.L.)
[2] China Construction First Group Construction & Development Co., Ltd., Beijing 100102, China; Zhanbixiong@chinaonebuild.com (B.Z.); hanju@chinaonebuild.com (J.H.)
* Correspondence: debbie_ly77@126.com

Abstract: The degrading of input images due to the engineering environment decreases the performance of helmet detection models so as to prevent their application in practice. To overcome this problem, we propose an end-to-end helmet monitoring system, which implements a super-resolution (SR) reconstruction driven helmet detection workflow to detect helmets for monitoring tasks. The monitoring system consists of two modules, the super-resolution reconstruction module and the detection module. The former implements the SR algorithm to produce high-resolution images, the latter performs the helmet detection. Validations are performed on both a public dataset as well as the realistic dataset obtained from a practical construction site. The results show that the proposed system achieves a promising performance and surpasses the competing methods. It will be a promising tool for construction monitoring and is easy to be extended to corresponding tasks.

Keywords: helmet detection; super-resolution reconstruction; you only look once v5 (YOLOv5)

Citation: Liu, Y.; Li, Z.; Zhan, B.; Han, J.; Liu, Y. A Super-Resolution Reconstruction Driven Helmet Detection Workflow. *Appl. Sci.* **2022**, *12*, 545. https://doi.org/10.3390/app12020545

Academic Editor: Vincent A. Cicirello

Received: 22 November 2021
Accepted: 29 December 2021
Published: 6 January 2022

1. Introduction

Given the continued rapid growth of modern society worldwide, the construction industry has developed increasingly fast. However, as well as the traditional hot issues such as design and construction technology, the safety of construction sites is becoming one of the hottest topics in the current construction industry. The construction industry is one of the most prone to safety accidents among all industries. Over the past 20 years, the construction industry has experienced a decline in accident rates, but the industry's accident rate is still about three times that of all industries [1]. Therefore, it is of great practical significance to study the safety guarantee in the construction industry.

Among the safety accidents in the construction industry, the level of disability caused by head injury is the highest, thus, reducing head injuries is obviously the primary objective to ensure the safety of personnel [2]. Currently, the helmet is the most effective way to reduce head injuries because the impact resistance of the helmet can disperse the impact of weights when an accident occurs, so as to greatly reduce the head and neck injuries caused by impact. In order to protect the life safety of employees, the common rule is to force people to wear helmets before entering the construction area and starting working. Practically, it is difficult to implement this requirement since the workload of maintaining supervision is tedious and labor-consuming. Consequently, several kinds of monitoring systems have been developed including fixed camera-based systems and moving camera-based processes. These systems work in a closed loop, which captures information through cameras and performs analysis manually or automatically to produce alarms or any other control signals. It is obvious that moving cameras are more flexible and able to cover wider regions so these have attracted a larger amount of attention from users and researchers. Nonetheless, moving cameras, either cameras on-board an unmanned aerial vehicle (UAV) or web cameras, normally suffer from the problem of data transmission, according to

the fact that monitoring image or video files always have a large size and ask for a wide bandwidth, which is a precious resource in wireless communication. To overcome this problem, compressing files by decreasing the resolution can be performed, but this often degrades the image quality, which is hard to be avoided during the trans

Mission, as shown in Figure 1. This ongoing chain reaction means that the difficulties of monitoring and analyzing images or videos are significantly increasing. In all honesty, this is not a big issue for manual analysis since human brains can adapt to this degrading in most cases, but it really affects the performance of most automatic analyzing methods. Taking into consideration the fact that automatic methods surpass manual approaches due to their advantages of being more effective and less labor-consuming, there is an urgent need for the improvement of automatic helmet detection performance based on low-resolution image or video files, which is the motivation of this paper.

Figure 1. The example of high-resolution images acquired from digital camera and the degraded low-resolution images obtained from the wireless channel due to the constraint of the available transmission bandwidth. (**a**) high-resolution images; (**b**) low-resolution images.

In recent years, there have been a large number of achievements focused on the automatic detection of helmets. Based on the employed research ideas, these methods can be divided into two kinds. Some of them use traditional solutions of object detection tasks, which consist of handcraft feature designing and machine learning models such as a support vector machine. The predesigned features are usually from general computer vision tasks, such as Haar-like features from face detection [3], and the deformable parts model (DPM), which could cascade different single features [4]. However, choosing or designing features is complicated, easy to be prone to poor accuracy and difficult to be extended from one scenario to another. The other methods use deep learning to solve object detection. The region-based convolutional neural network (RCNN) combines the selective search algorithm and convolutional neural network to detect targets, which makes for great improvements in accuracy and speed compared with traditional methods [5]. An RCNN is a two-stage detection network, the speed of which is difficult to meet in real construction engineering. The you only look once model (YOLO) realizes faster detection based on the improvement of the feature extractor backbone. Due to their significant improvement of detection performance, they replace traditional methods in a short time. Actually, most

modern object detection architectures are developed and validated on prepared datasets, which consist of high-quality images. Although they achieve good accuracy on those well-captured images, their performance still decreases quickly as the input quality decreases. In other words, the design of detection architectures assumes that the input image quality meets the demand. However, in our application scenario, when these models are applied on detecting helmets from images obtained through moving cameras, which are degraded in quality, it is hard to prevent the performance crashing.

To solve this problem, we propose a super-resolution reconstruction driven helmet detection workflow to improve detection accuracy under poor image quality. The main contributions of the paper are as follows.

(1) We propose an end-to-end helmet monitoring system, which implements a super-resolution reconstruction driven helmet detection workflow. It works well with poor input image quality and is easy to collaborate with any kinds of image acquisition device, including a wireless web camera or UAV.
(2) We propose to train a super-resolution model with combination loss of l_1 and contextual loss, which enhance its accuracy. We train the super-resolution reconstruction model and the detection model iteratively from scratch to achieve final results.
(3) Validations are performed on both a public dataset as well as the realistic dataset obtained from a practical construction site. The results show the proposed workflow achieves a promising performance and surpasses the competing methods.

2. Related Work

2.1. Object Detection

As one of the fast-developing fields in recent years, deep learning-based object detection algorithms are becoming the leading methods to solve object detection tasks. Most successful methods can be divided into two main categories, two-stage detection and one-stage detection. The most representative methods following two-stage detection including the RCNN, Fast regions with convolutional neural networks (Fast-RCNN) and their variants [6,7]. The common idea of these methods is first to obtain region proposals, which might contain the objects, then change the task into a classification to attach each anchor box a label. It is almost the standard solution for long periods of time due to its relatively high accuracy. However, it has proven to be of less help in practical scenarios that ask for real-time monitoring because of their low detection speed. In contrast, one-stage detection methods try to solve the problem through regression. This first employs a feature extracting backbone, usually a convolutional neural network (CNN)-based one, to produce feature maps, then predicts the position, class and confidence of objects at the same time. Based on the improvement of the feature extractor backbone, YOLO evolves from YOLOv1 to YOLOv3 to achieve better accuracy and speed [8–10]. Adopting a cross stage partial network (CSP)-based darknet-53 as the backbone network and replacing feature pyramid networks (FPN) with a path aggregation network (PANet), YOLOv4 improves the detection accuracy of the model in advance [11]. Recently, the YOLOv5 network model has added a focus structure to the backbone network on the base of YOLOv4, and balanced the detection speed and accuracy. Currently, one-stage detection methods are widely used in engineering practice due to the good time efficiency. Nevertheless, in most cases, the accuracy of YOLO and its variants is not as high as that of two-stage methods, especially for small targets and the low-resolution input.

2.2. Super-Resolution Reconstruction

There have been a large number of attempts to improve the performance of super-resolution reconstruction as it is really a long story in the development of computer vision. The most widely used approaches are kinds of interpolation-based methods, such as bilinear interpolation or nearest-neighboring interpolation [12]. Since the process of interpolation always follows a fixed pattern to calculate the new-generated pixel values from existing ones in a low-resolution image, it is hard to adapt to an unknown image degrading protocol.

Another traditional idea is to treat the SR problem as image reconstruction [13,14]. Inspired by the learning method, recent super-resolution approaches directly learn the nonlinear relationship from the low–high-resolution images. Based on the learning process, it could be divided into supervised SR and unsupervised SR. The supervised SR requires aligned high- and low–high-resolution image pairs to train the CNN models to fit the mapping between images with different resolutions. Dong et al. propose the super-resolution convolutional neural networks (SRCNN), which effectively improves the effect and speed of image SR reconstruction compared with the traditional image SR algorithms [15]. Kim et al. propose a VDSR network, which increases the number of layers of CNN to 20 [16]. The algorithm combines residual structure and CNN with image SR reconstruction, and the image reconstruction effect has been significantly improved. Li et al. propose a multi-scale residual network (MSRN), which applies image multi-scale features to the residual structure to further improve the image reconstruction effect [17]. Zhang et al. propose a residual channel attention network RCAN, which applies a channel attention mechanism to the image super-resolution problem and achieves a better reconstruction effect than previous algorithms [18]. To apply these methods successfully, we need to prepare a large number of strictly aligned image pairs, which is not a simple task in practice. Thus, currently, simulated image pairs are used in research so as not to decrease its performance in real world applications. Unsupervised SR methods employ the GAN model and its variants to generate high-resolution images with the low-resolution input [19–21]. However, without paired training data, its accuracy is not as good as that of supervised methods.

2.3. Helmet Detection

Automatic helmet detection is urgently needed in construction engineering and safety driving monitoring. The traditional helmet detection methods focus on the design of the artificial features to lead the classification towards appropriately discriminating helmet from non-helmet targets. The well-known image descriptors such as the local binary pattern (LBP), local variance (LV) and histogram of oriented gradient (HOG) are used to enhance the feature extraction step, and they achieve promising accuracy through a supporting vector machine [22]. The circular Hough transform (CHT) accompanied with HOG descriptor are applied to extract the helmet attributes, and the multilayer perceptron (MLP) classifier is used to perform the final helmet classification [23]. The method combining multi-feature fusion and a support vector machine (SVM) is used to detect and track the helmet in a factory environment to keep an eye on safety production [24]. However, the choosing of manual features is labor-consuming and poor in generalization, which prevents their application. Nowadays, thanks to the development of deep learning and CNN, a large number of modern helmet detection approaches have been proposed. The CNN-based multi-task learning model has been designed for tracking individual motorcycles through identifying helmets [25]. The faster region-based convolutional neural network (Faster RCNN) is utilized to detect both motorcyclists and helmets [26]. The faster RCNN equipped with the multi-scale training and increasing anchors strategies has proved to be capable of detecting helmets on different scales [27]. Taking the processing speed into consideration, YOLO is even more popular. An improved YOLOv3 model has been applied to detect helmets and successfully increased the average accuracy [28]. Replacing the traditional YOLOv3 backbone of darknet-53 with a deep separable convolution structure, the performance of helmet detection has been further improved [29]. Nevertheless, all these models ask for quality stable input images, which is reachable in training set acquisition but difficult to reach in practical terms. In other words, if there is no assurance about input quality, the detection performance will decrease quickly. To the best of our knowledge, few methods have focused on solving the helmet detection problem with poor quality input images.

3. Method

We propose an end-to-end helmet monitoring system, which implements a super-resolution reconstruction driven helmet detection workflow as shown in Figure 2. There

are two main modules in the workflow, the super-resolution reconstruction module and the detection module. The former implements the SR algorithm to produce high-resolution images, while the detection model performs the helmet detection. Based on the helmet detection results, we can perform semantic analysis of counting or wearing detection based on the specific monitoring task.

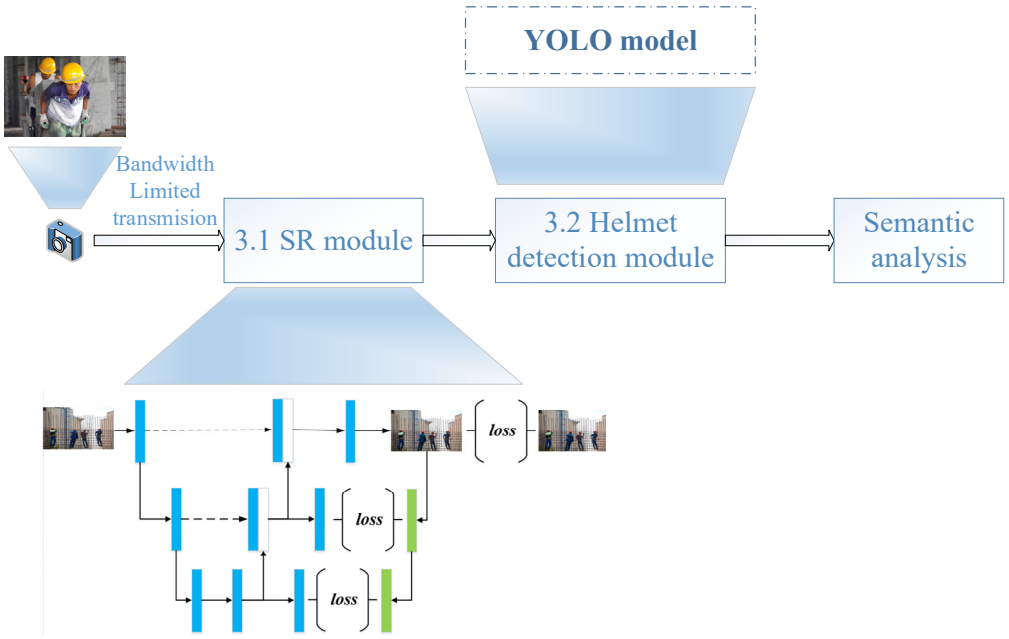


Figure 2. The workflow of the proposed end-to-end helmet monitoring system.

3.1. SR Module

We take the dual regression network as our backbone architecture for super-resolution reconstruction [30]. The main idea is to add a dual regression task (from high-resolution image to low-resolution image) alongside the primal regression task (from low-resolution image to high-resolution image). Through the constraint of the reversible reconstruction, the mapping space between the low–high-resolution images is compressed and it is easier to fit to the real degrading relationship. The details of the SR model are shown in Figure 3.

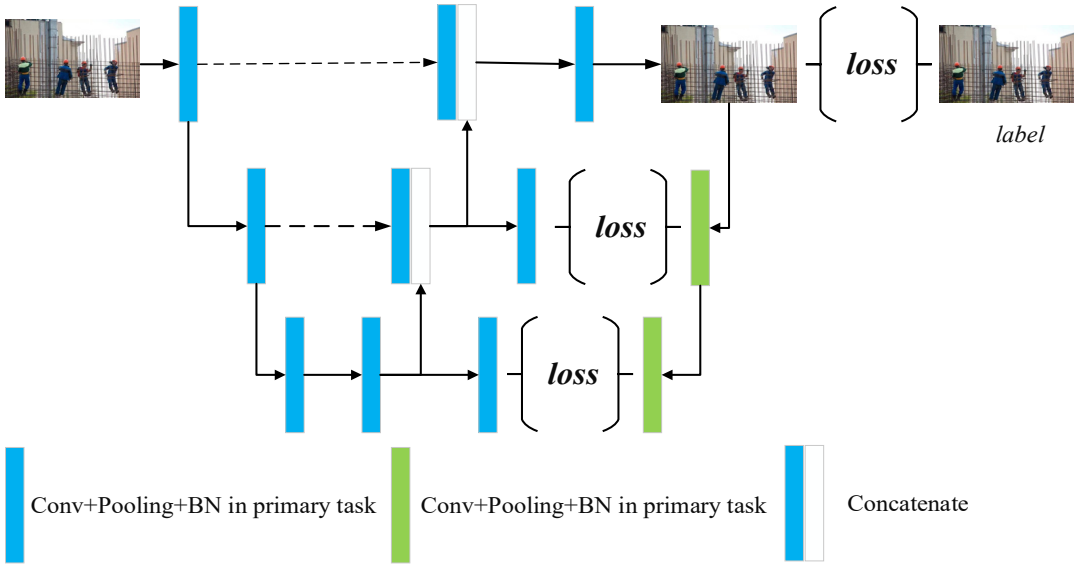


Figure 3. The architecture details of the SR module.

We employ a simple U shape symmetric architecture to produce the primal task. The input low-resolution images are fed into feature extractor consisting of stacked convolution layers. Then, the abstracted features are mapped onto the target image space through trans-convolution. Alongside the easy-to-understand primal architecture, we attach a one-way path to map the resolution of image from high to low. Through improving the deep supervision of the mapping loop, we can achieve the advanced reconstruction performance.

Normally, the two regression tasks in SR are optimized following the classic l_1 loss, that is to say, the mean absolute error (MAE). However, taking into consideration that the main purpose of our SR module is to perform the preparation for the following object detection module, when comparing with pixel-wise similarity, we should pay more attention to the sematic information. Therefore, we employ a combination of l_1 loss and contextual loss as follows

$$l = \sum_{i=1}^{N} [l_1(P(x_i), y_i) + \alpha l_c(P(x_i), y_i) + l_1(D(P(x_i)), x_i)] \tag{1}$$

where (x_i, y_i) is the ith low-resolution and high-resolution image pair. $P(\cdot)$ and $D(\cdot)$ are the primal regression and dual regression, respectively. α is weighting coefficients of the contextual loss component. l_1 indicates the MAE and l_c denotes the contextual loss calculated from Equation (2).

$$l_c(x_i, y_i) = -\log(CX(F_P(x_i), F_D(y_i))) \tag{2}$$

where $F_P(\cdot)$ and $F_D(\cdot)$ are the feature maps obtained from the feature extractor during primal task and dual task, respectively. Based on our symmetric architecture shown in Figure 3, the two feature maps always have the same size N. In order to measure how similar the two feature maps are, we refer to the similarity in [31]. $CX(F_P(x_i), F_D(y_i)) = \frac{1}{N}\sum_j \max_k CX_{kj}$, where CX_{kj} calculates the similarity between the kth and the jth features from $F_P(\cdot)$ and $F_D(\cdot)$.

$$CX_{kj}(F_P(\cdot), F_D(\cdot)) = \frac{\exp\left(\frac{1-\widetilde{d_{kj}}}{h}\right)}{\sum_l \exp\left(\frac{1-\widetilde{d_{kl}}}{h}\right)} \tag{3}$$

where $\widetilde{d_{kj}} = d_{kj} / \left(\min_l d_{kl} + \epsilon\right)$, and d_{kj} is the cosine distance between the kth and the jth features from $F_P(\cdot)$ and $F_D(\cdot)$. The parameters $h = 0.5$ and $\epsilon = 10^{-8}$. Due to the setting, the closer the two features, the smaller the d_{kj}. Consequently, the smoothed $\widetilde{d_{kj}}$ approaches 1 so as to produce large CX_{kj} as well as large $CX(F_P(x_i), F_D(y_i))$. Because the $F_P(\cdot)$ and $F_D(\cdot)$ are abstracted information obtained from backbone network, they are full of sematic information and could help the SR module focus more on high level similarity instead of pixel-wise alignment. This proved to be great help for our detection.

3.2. Detection Module

As one of the most advanced one-stage object detection models, YOLOv5 is chosen as our backbone model to detect helmets. Since there is a clear evolving track for YOLO models and the YOLOv5 is a combination of all the prior tricks and improvements, we do not recap the entire architecture in detail. Here, we only talk about how we use the model. Briefly, the three main components employed in YOLOv5 are backbone for feature extracting, neck for feature fusing and head for prediction. The cross stage partial network combined Darknet is used as backbone, which could abstract abundant information and the path aggregation network is utilized as neck to generate the feature pyramid so that it can enhance the capability of multi-scale detection. The head part follows the traditional YOLO head used in the prior version to obtain the prior box and classification result. In this paper, the detection module containing YOLOv5 model follows the SR module directly to implement the helmet detection. The loss function follows reference [8].

3.3. Dataset

We employ both public dataset as well as the realistic dataset to train our model. For SR task, the public data include the DIV2K and the Flickr2K, which contain 3550 paired images of high resolution, 2× and 4× low resolution [32,33]. There is no requirement with regard to the image content. The realistic dataset includes 5457 images randomly downloaded from the Internet. The chosen standard is that each image contains as least one person wearing helmet. We downsample these images through bicubic algorithm to generate 2× and 4× low-resolution images. All these paired images, 9007 in total, construct our training dataset. We obtain an individual test set, which includes 270 images via the same grabbing way of training set. The testing images are downsampled through randomly chosen methods available in the skimage toolbox of python. For helmet detection task, the aforementioned data excluding DIV2K and the Flickr2K are used as training and testing, respectively.

3.4. Metrics

The quantitative metrics employed in this paper are shown in Table 1. For SR task, we use the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) values to measure the reconstructed image quality. The higher the value of PSNR is the better. The value of SSIM varies between 0 (worst) and 1 (best). For detection task, the precision, recall and average precision (AP) are calculated. Precision measures the capability that the model finds out targets which are real targets. Recall measures the capability that the model finds out real targets without missing. AP is calculated from the area under the precision–recall curve and values approaching 1 are the best.

Table 1. The definition of utilized quantitative metrics.

Task	Metrics	Definition
SR	PSNR	$\mathrm{PSNR} = 10\log_{10}\frac{255}{\frac{1}{3}\sum_{c=\{R,G,B\}}(\mathrm{MSE})_c}$ where MSE indicates the mean square error of the image.
	SSIM	$\mathrm{SSIM}(I_0, I) = \frac{(2\mu_{I_0}\mu_I + c_1)(2\sigma_{I_0 I} + c_2)}{(\mu_{I_0}^2 + \mu_I^2 + c_1)(\sigma_{I_0}^2 + \sigma_I^2 + c_2)}$ where I_0 and I are the original and the reconstructed high-resolution images. μ and σ indicate mean and variance, respectively, and c_1, c_2 are constants
Detection	Precision	$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$ where TP, FP and FN indicate true positives, false positives and false negatives, respectively.
	Recall	$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
	AP	Area under the precision–recall curve

3.5. Training

The entire method is implemented on the workstation equipped with two NVIDIA RTX3090 GPU and Intel i9 CPU. All coding work is based on Python 3.7 and PyTorch 1.8. The SR module and detection module are first trained separately from scratch. Then the two modules are finetuned together, alternately. Specifically, in each iteration, one module will be frozen while the other one is updating, then vice versa. The initial learning rate of the SR and detection are 0.0001 and 0.01, respectively. The Adam optimizer is applied with the momentum 0.9 and the batch size is 2.

4. Results

4.1. Performance of the Proposed SR Module

We compare the super-resolution reconstruction results of the proposed SR module with those of the other popular SR methods. According to the fact that our main purpose of this paper is helmet detection instead of pure super-resolution reconstruction, we choose

the widely used interpolation methods for comparison since they are often utilized to adjust the network input resolution in object detection tasks. Our method starts from the DRN-S model designed for SR tasks so that it is necessary to compare our improvements with the original DRN-S model [30]. The achieved results are shown in Table 2. There is a gap between the PSNR values of SR-based methods and those of interpolation-based methods. Our SR module achieves a higher PSNR value while keeping its SSIM value consistent with that of DRN-S. This means that we can achieve better image quality so as to improve the accuracy of the coming detection module.

Table 2. The results achieved by different methods.

	Interpolation			**DRN-S**	**Our Method**
	Nearest Neighbor	**Bilinear**	**Bicubic**		
PSNR	23.716	25.277	25.343	27.964	27.991
SSIM	0.737	0.782	0.784	0.850	0.850

From Figure 4, we can review the super-resolution reconstructed images directly. Since our real targets are helmets, we focus on them more instead of on the background information. All helmet regions are zoomed in to visualize their details. It can be found that the helmets obtained from super-resolution reconstructed-based methods are clearer than those achieved by the interpolation-based method. To be specific, our SR module produces images that are less blurry but not so piecewise smooth to be able to obtain more median-frequency information.

Based on Table 2, there is a significant improvement using our method compared with the original DRN-S. To evaluate the effort of our newly added contextual loss, we compared the performance of the SR module with different weight α, as shown in Figure 5. DRN-S refers to the original DRN-S model. C-0.001 refers to the SR model trained under the combined loss described in Equation (1) with $\alpha = 0.001$. C-0.0005 and C-0.0001 indicate $\alpha = 0.0005$ and $\alpha = 0.0001$, respectively.

4.2. Performance of the Proposed Helmet Detection Method

We show the detection results produced by different end-to-end workflows in Table 3. Each workflow employs the same YOLOv5 architecture but different input. The Interpolation+YOLOv5 workflow is exactly the same as with normal YOLOv5 since it uses pure interpolation to resize the input images to the standard resolution. The DRN+YOLOv5 workflow is also super-resolution reconstruction driven detection, which consists of the DRN model and YOLOv5. The proposed SR module+YOLOv5 indicates the workflow described in this paper. From Table 3, the combination of the proposed SR module and YOLOv5 achieves the best precision, which means that 88.4% predicted helmets are real helmets. Compared with the other two results, it has the fewest false predictions. However, since precision is normally in contradiction with recall, the recall of the proposed SR module+YOLOv5 lags a little behind that of the DRN+YOLOv5. The AP of the proposed SR module+YOLOv5 is the highest, which indicates it has the best overall performance. It can be seen that the input images produced by the SR module will improve all metrics due to the improvement in image quality. The proposed workflow surpasses the original DRN+YOLOv5 workflow on precision and AP.

Table 3. Detection performance of different workflows.

	Interpolation+YOLOv5	**DRN+YOLOv5**	**Proposed SR Module+YOLOv5**
Precision	0.853	0.878	**0.884**
Recall	0.632	**0.716**	0.715
AP (%)	0.435	0.500	**0.501**

Figure 4. The comparison of the images produced by different SR method. From left to right, columns (**a**,**c**) indicate the complete images, columns (**b**,**d**) indicate the zoom-in regions defined by the blue bounding boxes. From the first row to the last row: original high-resolution image, images achieved via nearest neighbor interpolation, images achieved via bilinear interpolation, images achieved via bicubic interpolation, images achieved via DRN, images achieved via the proposed SR module.

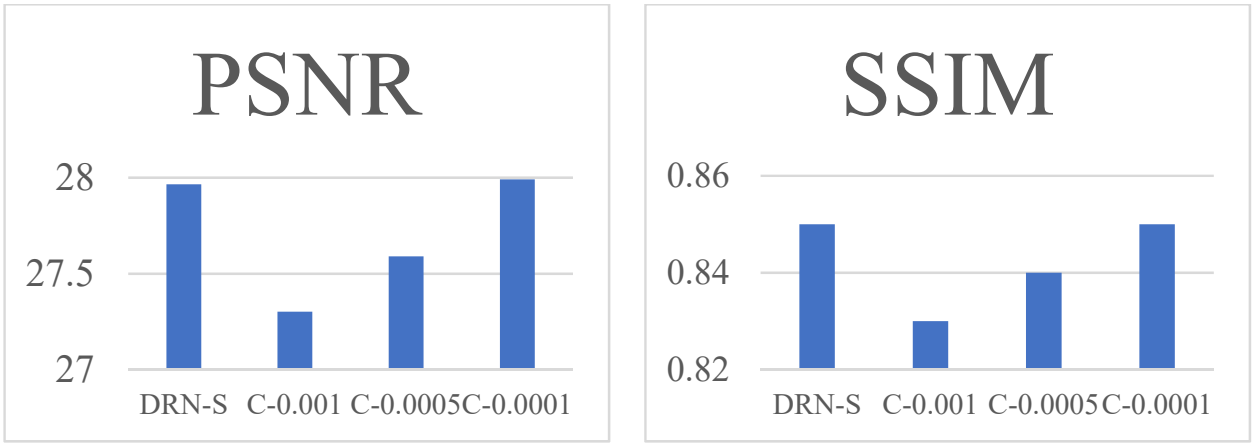


Figure 5. The PSNR and SSIM varied from the weighting of contextual loss.

5. Discussion and Conclusions

We propose an end-to-end helmet monitoring system, which implements a super-resolution reconstruction driven helmet detection workflow. It is designed for the scenario where the input image quality is limited, which is easily faced in engineering practice. For example, input images are acquired from a moving camera and transmitted through a bandwidth-limited wireless channel. Because of the limited bandwidth, images are always compressed and so have poor resolution and quality. The degrading of the input images will consequently decrease the helmet detection precision. To overcome this problem, the proposed SR driven helmet detection workflow consists of two sequential steps in the entire workflow. First, we use a super-resolution reconstruction module to improve the image resolution and quality instead of direct interpolation. Then, the processed images are fed into the detection module consisting of YOLOv5 to perform helmet detection. The two modules are trained separately from scratch and finetuned together, alternately. This is a typical multi-task learning strategy to help increase task specific accuracy by utilizing other tasks as constraints. Validation shows the effectiveness of our workflow. The comparison of the performance of different SR reconstruction methods shows that the proposed SR module could increase the PSNR value while maintaining a consistent SSIM value. The comparison of the performance of different detection workflows shows that the proposed SR module is effective at guiding the YOLOv5 and detection precision and AP are both increased. Generally speaking, based on current results, this will be a promising tool for helmet detection, which can be easily used in construction monitoring or traffic safety monitoring. Moreover, SR driven detection is a general workflow that is easy to be extended to other similar object detection tasks to solve the problem of performance degrading caused by poor inference input quality when the training input quality is good. Currently, our main idea is to use the individual model on specific tasks and combine tasks together. The model will be redundant if there are a large number of tasks. In the future, we will keep working on identifying a semantic subspace to attempt to remove the influence of image quality on detection performance.

Author Contributions: Conceptualization, Y.L. (Yicheng Liu) and Y.L. (Yan Liu); methodology, Y.L. (Yicheng Liu); software, Z.L.; validation, Y.L. (Yicheng Liu), Z.L. and Y.L. (Yan Liu); data curation, B.Z.; writing—original draft preparation, Z.L.; writing—review and editing, Y.L. (Yan Liu); visualization, B.Z.; supervision, J.H.; project administration, B.Z. and J.H.; funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Research on automatic and intelligent safety management technology at construction site, grant number CSCEC-2020Z-10 and Research on intelligent construction site management based on Internet of things and image recognition technology, grant number KJYF-2019-12.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The open access dataset DIV2K can be found in the website https://data.vision.ee.ethz.ch/cvl/DIV2K/ (accessed on 24 December 2021). The open access dataset Flickr2K can be found in the website https://yingqianwang.github.io/Flickr1024/ (accessed on 24 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kurien, M.; Kim, M.K.; Kopsida, M.; Brilakis, I. Real-time simulation of construction workers using combined human body and hand tracking for robotic construction worker system. *Autom. Constr.* **2017**, *86*, 125–137. [CrossRef]
2. Zhong, H.; Yanxiao, W. 448 cases of construction standard statistical characteristic analysis of inductrial injury accident. *Stand. China* **2017**, *2*, 245–247.
3. Viola, P.; Jones, M.J. Robust Real-Time Face Detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [CrossRef]
4. Felzenszwalb, P.F.; Mcallester, D.A.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.
5. Tang, T.; Zhou, S.; Deng, Z.; Zou, H.; Lei, L. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining. *Sensors* **2017**, *17*, 336. [CrossRef] [PubMed]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
7. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
9. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
10. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
11. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-J.M. YOLOv4 Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
12. Kirkland, E.J. *Bilinear Interpolation*; Springer: Manhattan, NY, USA, 2010.
13. Liu, Y. An Improved Feedback Network Superresolution on Camera Lens Images for Blind Superresolution. *J. Electr. Comput. Eng.* **2021**, *2021*, 5583620. [CrossRef]
14. Chen, Y.; Liu, L.; Phonevilay, V.; Gu, K.; Xia, R.; Xie, J.; Zhang, Q.; Yang, K. Image super-resolution reconstruction based on feature map attention mechanism. *Appl. Intell.* **2021**, *51*, 4367–4380. [CrossRef]
15. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a Deep Convolutional Network for Image Super-Resolution. In *Computer Vision—ECCV 2014*; Springer: Cham, Switzerland, 2014; Volume 8692, pp. 184–199.
16. Kim, J.; Lee, J.K.; Lee, K.M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
17. Li, J.; Fang, F.; Mei, K.; Zhang, G. Multi-scale Residual Network for Image Super-Resolution. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 11212, pp. 527–542.
18. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 11211, pp. 294–310.
19. López-Tapia, S.; Lucas, A.; Molina, R.; Katsaggelos, A.K. A single video super-resolution GAN for multiple downsampling operators based on pseudo-inverse image formation models. *Digital Signal Process.* **2020**, *104*, 102801. [CrossRef]
20. Majdabadi, M.M.; Ko, S.B. Capsule GAN for robust face super resolution. *Multimedia Tools Appl.* **2020**, *79*, 31205–31218. [CrossRef]
21. Bulat, A.; Yang, J.; Tzimiropoulos, G. To Learn Image Super-Resolution, Use a GAN to Learn How to Do Image Degradation First. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 11210, pp. 187–202.
22. Badaghei, R.; Hassanpour, H.; Askari, T. Detection of Bikers without Helmet Using Image Texture and Shape Analysis. *Int. J. Eng.* **2021**, *34*, 650–655. [CrossRef]
23. E Silva, R.R.V.; Aires, K.R.T.; Veras, R. Detection of helmets on motorcyclists. *Multimed. Tools Appl.* **2018**, *77*, 5659–5683. [CrossRef]
24. Sun, X.; Xu, K.; Wang, S.; Wu, C.; Zhang, W.; Wu, H. Detection and Tracking of Safety Helmet in factory environment. *Meas. Sci. Technol.* **2021**, *32*, 105406. [CrossRef]

25. Lin, H.; Deng, J.D.; Albers, D.; Siebert, F.W. Helmet Use Detection of Tracked Motorcycles Using CNN-Based Multi-Task Learning. *IEEE Access* **2020**, *8*, 162073–162084. [CrossRef]
26. Yogameena, B.; Menaka, K.; Perumaal, S.S. Deep learning-based helmet wear analysis of a motorcycle rider for intelligent surveillance system. *IET Intell. Transp. Syst.* **2019**, *13*, 1190–1198. [CrossRef]
27. Gu, Y.; Xu, S.; Wang, Y.; Shi, L. An Advanced Deep Learning Approach for Safety Helmet Wearing Detection. In Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 14–17 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 669–674.
28. Xu, K.; Deng, C. Research on Helmet Wear Identification Based on Improved YOLOv(3). *Laser Optoelectron. Progr.* **2021**, *58*, 0615002.
29. Xiao, T. Improved YOLOv3 Helmet Wearing Detection Method. *Comput. Eng. Appl.* **2021**, *57*, 216–223.
30. Guo, Y.; Chen, J.; Wang, J.; Chen, Q.; Cao, J.; Deng, Z.; Xu, Y.; Tan, M. Closed-Loop Matters: Dual Regression Networks for Single Image Super-Resolution. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5406–5415.
31. Mechrez, R.; Talmi, I.; Zelnik-Manor, L. The Contextual Loss for Image Transformation with Non-aligned Data. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 800–815.
32. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1122–1131.
33. Lim, B.; Son, S.; Kim, H.; Nah, S.; Lee, K.M. Enhanced Deep Residual Networks for Single Image Super-Resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1132–1140.

applied sciences

MDPI

Article

Multi-Scale Safety Helmet Detection Based on SAS-YOLOv3-Tiny

Rao Cheng, Xiaowei He *, Zhonglong Zheng and Zhentao Wang

College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China; chengrao@zjnu.edu.cn (R.C.); zhonglong@zjnu.edu.cn (Z.Z.); zhentaowang@zjnu.edu.cn (Z.W.)
* Correspondence: jhhxw@zjnu.edu.cn

Abstract: In the practical application scenarios of safety helmet detection, the lightweight algorithm You Only Look Once (YOLO) v3-tiny is easy to be deployed in embedded devices because its number of parameters is small. However, its detection accuracy is relatively low, which is why it is not suitable for detecting multi-scale safety helmets. The safety helmet detection algorithm (named SAS-YOLOv3-tiny) is proposed in this paper to balance detection accuracy and model complexity. A light Sandglass-Residual (SR) module based on depthwise separable convolution and channel attention mechanism is constructed to replace the original convolution layer, and the convolution layer of stride two is used to replace the max-pooling layer for obtaining more informative features and promoting detection performance while reducing the number of parameters and computation. Instead of two-scale feature prediction, three-scale feature prediction is used here to improve the detection effect about small objects further. In addition, an improved spatial pyramid pooling (SPP) module is added to the feature extraction network to extract local and global features with rich semantic information. Complete-Intersection over Union (CIoU) loss is also introduced in this paper to improve the loss function for promoting positioning accuracy. The results on the self-built helmet dataset show that the improved algorithm is superior to the original algorithm. Compared with the original YOLOv3-tiny, the SAS-YOLOv3-tiny has significantly improved all metrics (including Precision (P), Recall (R), Mean Average Precision (mAP), F1) at the expense of only a minor speed while keeping fewer parameters and amounts of calculation. Meanwhile, the SAS-YOLOv3-tiny algorithm shows advantages in accuracy compared with lightweight object detection algorithms, and its speed is faster than the heavyweight model.

Keywords: YOLOv3-tiny; object detection; attention mechanism; deep learning; intelligent transportation

Citation: Cheng, R.; He, X.; Zheng, Z.; Wang, Z. Multi-Scale Safety Helmet Detection Based on SAS-YOLOv3-Tiny. *Appl. Sci.* **2021**, *11*, 3652. https://doi.org/10.3390/app11083652

Academic Editors: Federico Divina, Javier Alonso Ruiz, Jeroen Ploeg, Martin Lauer, Angel Llamazares Llamazares, Noelia Hernández Parra and Carlota Salinas

Received: 23 March 2021
Accepted: 15 April 2021
Published: 19 April 2021

1. Introduction

Driving a motorcycle or an electric two-wheeler without a safety helmet will cause a high mortality rate. However, many cyclists still have fluke psychology, so wearing safety helmets must rely on traffic policies' way of compulsory supervision to attract people's attention. At present, there are two main ways for traffic management departments to supervise whether riders wear helmets. In general, traffic policies check traffic surveillance videos manually. Another way is that traffic policies manage drivers and passengers on the road. These methods need a lot of human and material resources and cause the phenomenon of missing detection. Whether or not people who ride motorcycles and two-wheelers wear safety helmets to improve safety is crucial for intelligent traffic management, which has a significant research value. With the development of artificial intelligence, intelligent systems based on automatic image detection have been intensely studied and applied in different fields.

The task of object detection is to locate and classify objects in a given image. The uncertainty of object type and number, the diversity of object scale and the external environment's interference will bring different degrees of influence to the task. Object detection algorithms based on convolutional neural networks are mainly divided into two categories: anchor-based and anchor-free. There are two types of anchor-based algorithm: two-stage

algorithms represented by the Region-based Convolutional Neural Network(R-CNN) series and one-stage algorithms represented by the Single-shot multi-box Detector (SSD) series and the YOLO series. R-CNN [1] is a pioneering two-stage object detection algorithm proposed by Girshick et al., He et al. proposed SPP-Net [2] to accelerate R-CNN and learn more different features. Girshick et al. proposed Fast R-CNN [3], which used the Region of Interest (ROI) pooling layer to extract regional features. Object classification and bounding box regression can be optimized end-to-end without requiring additional cache space while it had better detection accuracy and faster reasoning speed than R-CNN and SPP-Net. Even though model learning has improved, the generation of the proposal still relied on traditional methods. Faster R-CNN [4] relied on a new proposal generator methods-Region Proposal Network (RPN), which can be learned by a supervised learning approach. Dai et al. proposed a Fully Convolution Network-based region (R-FCN) [5] to share the computational cost of region classification steps, compared with Faster RCNN, it achieves competitive results. In addition, a single deep feature map is used for final prediction in Faster R-CNN, which makes it difficult to detect objects of different scales, especially for small objects. Facing the problem, Lin et al. took advantage of different features and proposed Feature Pyramid Networks (FPN) [6], which combined deep features with shallow features. In order to make the whole detection process more flexible, He et al. proposed Mask R-CNN [7], which predicted bounding box and mask in parallel and reported the latest results. The two-stage algorithms above always divide the detection into two steps: proposal generation and proposal regression. The one-stage detection algorithms do not generate proposals. They categorize and locate each region of interest directly. Sermanet et al. proposed the one-stage detection algorithm OverFeat [8], which has a significant speed advantage. Redmon et al. developed a real-time detection algorithm called YOLO [9], and its entire framework is a single network, which omits the proposal generation step and can be optimized end-to-end. In 2016, SSD [10] was proposed to solve the limitations of YOLO. In the one-stage algorithm, the imbalance between foreground and background is a serious problem because there is no proposal generation to filter out easily generated negative samples, Lin et al. proposed RetinaNet [11] to solve the class imbalance problem in a more flexible way. Redmon et al. proposed an improved version of YOLO, YOLOv2 [12], which significantly improved the detection performance and maintained the real-time reasoning speed. Later, Redmon et al. proposed YOLOv3 [13], which used the Darknet-53 network structure and the idea of the residual network for reference. In addition, the idea of FPN was used to carry out multi-scale feature detection. The above improvements made YOLOv3 three times faster than SSD, while its accuracy is the same as SSD. Compared with the anchor-based algorithm, the anchor-free algorithms do not depend on the pre-defined anchors and avoid the complicated calculations related to the anchors. The earliest anchor-free methods are the Unifying Landmark Localization with End to End Object Detection (Densebox) [14] and YOLO [9]. The following anchor-free methods [15–17] are detection methods based on keypoints, and compared with YOLO and Densebox, the detection effect of these three methods is significantly improved. Finally, three methods [18–20] belong to the intensive prediction method, and they all obtain the desired result by directly predicting the rectangular box without using the anchor. In recent years, more and more object detection algorithms are applied in the security field. Haikuan Wang et al. put forward a real-time safety helmet wearing detection approach (named CSYOLOv3) [21], it achieved the mAP value of 67.05%, and its FPS reached above 25, so the mAP value was low and the speed was slow. Yang Li et al. proposed a deep learning-based safety helmet detection in engineering management based on convolutional neural networks [22], which would have a deficient performance when the images are not very clear, such as the safety helmets being too small and obscure. In addition, the above works only divide the categories of objects into two categories: Wear and Nowear, but the types of objects on the head are not distinguished. It is necessary to distinguish between different types of objects because ordinary hats are not safe, and different kinds of helmets have different protective effects on the head.

Experimental studies have found that the general object detection algorithms can be applied to the detection task of the safety helmet. However, under complex scenarios, the small-scale object is sheltered and it is dense. The remote small-scale safety helmets and hats with low resolution and blurry pixels have less characteristic information, which leads to the phenomenon of missed detection. In addition, it is challenging to balance the accuracy and complexity in general object detection algorithms, and the imbalance between the two makes it difficult to deploy on mobile devices. Even though YOLOv3 is a widely used object detection algorithm with good recognition speed and detection accuracy by combining several methods such as residual network, feature pyramid and multi-feature fusion network, it has lots of parameters and amount of computation and generates a large model. Hence, it is challenging that the model is transplanted to embedded applications when computing power and storage space are limited. YOLOv3-tiny based on YOLOv3 is a lightweight object detection network applying an embedded platform, but its detection accuracy is low. In this paper, SAS-YOLOv3-tiny is proposed to balance the detection accuracy and speed for a set of self-built helmet datasets. Aiming to promote detection effect while reducing the number of parameters and calculation amount, the Sandglass-Residual module based on depthwise separable convolution and channel attention mechanism is constructed to replace the traditional convolution layer while the convolution layer of stride two is utilized into the backbone to replace the max-pooling layer, which can extract informative and high-dimensional features. The three-scale feature prediction method is introduced into the network structure of SAS-YOLOv3-tiny to improve the two-scale feature prediction for obtaining accurate location information of small objects further. The improved spatial pyramid pooling module is applied to enhance the feature extraction further. CIoU is used to promote the loss function to improve location accuracy. Our algorithm achieved the mAP value of 81.6% on the validation set and the mAP value of 80.3% on the test set with the average detection time of 3.2 ms on each image under an actual traffic environment.

The rest of the paper is organized as follows. Section 2 will explain the principles of the original algorithm YOLOv3-tiny. Section 3 will describe the innovation points of the improved algorithm (SAS-YOLOv3-tiny) in detail. Section 4 will show some experimental results and analyze them. Finally, in Section 5, this paper will be summarized and some future works will be proposed.

2. The Principles of YOLOv3-Tiny

In this section, we will mainly introduce the principles of YOLOv3-tiny. In Section 2.1, the network architecture of YOLOv3-tiny will be defined in detail. In Section 2.2, the principle of bounding box prediction will be explained. The above principles lay a solid foundation for the improved algorithm in Section 3.

2.1. Network Architecture of YOLOv3-Tiny

YOLOv3-tiny is an improved version of YOLOv3, which has changed the YOLOv3's backbone network (named Darknet53) to seven convolution layers with kernel size of 3×3 and six max-pooling layers with stride 2. The idea of FPN is adopted to integrate feature map with low resolution and feature map with high resolution. YOLOv3-tiny utilized last two downsampled feature maps with size of $28 \times 28 \times 256$ and $14 \times 14 \times 1024$ to predict the objects. The reason is that the feature map with size of $14 \times 14 \times 1024$ contains abstract and high-level semantic information while the feature map with size of $28 \times 28 \times 256$ carrying more detailed and lower-level location information, which can obtain feature map containing both semantic and positional information. Specifically, the input image with size of $448 \times 448 \times 3$ is processed through the backbone network and a convolution operation, producing the resultant feature map with size of $14 \times 14 \times 1024$. One part of the processed results is processed through the convolutions and used to output predictions in terms of the current feature map, and the other part is processed through a convolution layer and an up-sampling operations, and then is fused with the corresponding upper

feature map with size of $28 \times 28 \times 256$. The above operations can obtain the feature map with size of $28 \times 28 \times 384$, which is processed by the convolutions, and then used for prediction. At scale y1, the feature map downsampled by $32\times$ is utilized to detect larger objects. At scale y2, the feature map downsampled by $16\times$ is responsible for detecting smaller objects. YOLOv3-tiny network's structure is demonstrated in Figure 1.

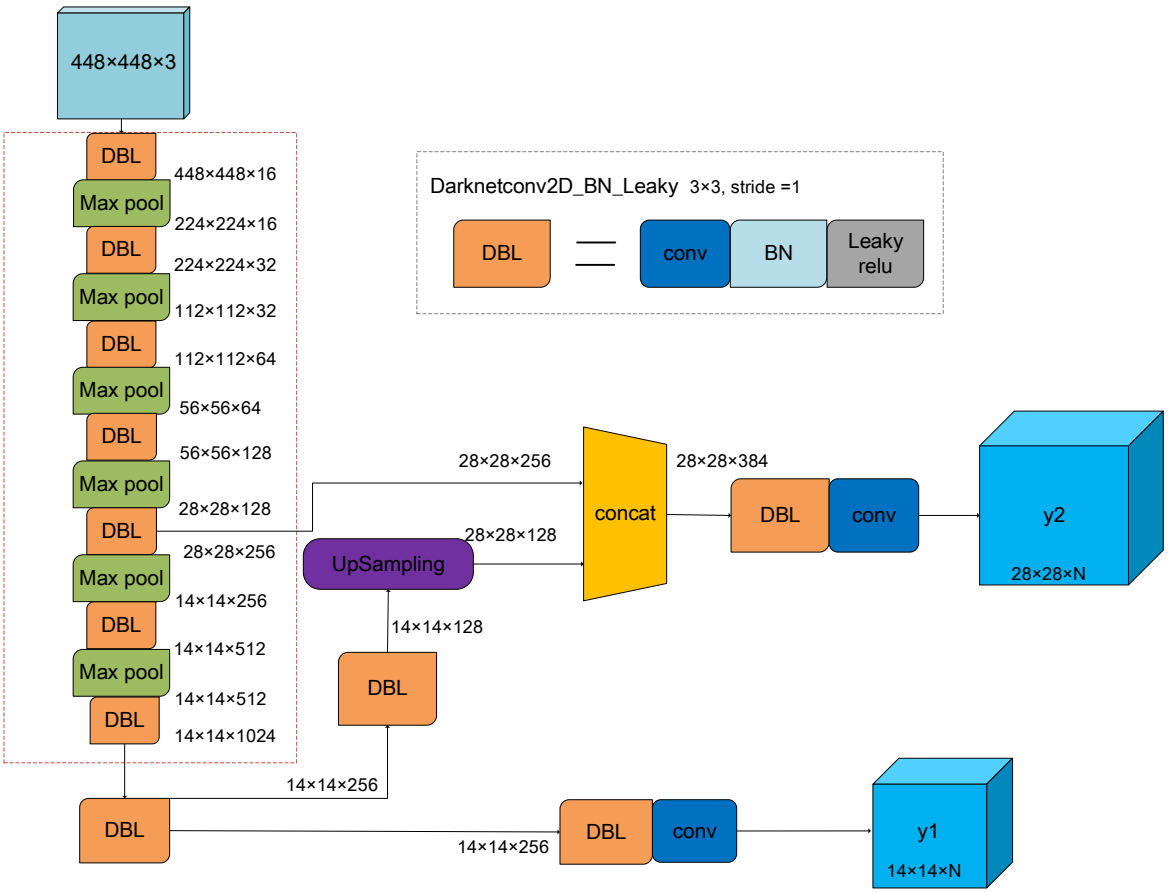


Figure 1. YOLOv3-tiny network structure.

2.2. Bounding Box Prediction

YOLOv3 continued to employ K-means clustering of YOLOv2 to determine the prior boxes, which drew on the anchor box mechanism of RPN in Faster R-CNN. K-means clustering algorithm in YOLOv3-tiny obtained K prior boxes on the Common Objects in Context (COCO) dataset according to the annotated ground truth boxes, which could improve the detection accuracy and speed. Joseph Redmon et al. modified the clustering distance in the k-means algorithm [12]. As shown in Formula (1), it is defined by *IOU*. The larger the *IOU* is, the closer distance of the two bounding boxes is.

$$d(box, centroid) = 1 - IOU(box, centroid) \quad (1)$$

In Formula (1), $d(box, centroid)$ represents the clustering distance, *centroid* represents the box that is selected as the center of mass by the algorithm, *box* represents the other bounding boxes and *IOU* represents the ratio of the intersecting area of the two boxes to the combined area. Even though too many prior boxes can guarantee the detection effects, it greatly affects the efficiency of the algorithm. YOLOv3-tiny used six prior boxes. The corresponding relationship between feature maps and prior boxes is as follows. Feature maps of size 14, 28 correspond [(81,82); (135,169); (344,319)], [(10,14); (23,27); (37,58)], respectively. Generally, large feature maps usually have small receptive fields,

which are very sensitive to small-scale objects, and thus, they will select small prior boxes. On the contrary, small feature maps always have large receptive fields, which are suitable for detecting large objects, and thus, they select large prior boxes.

The final predicted bounding box coordinates of the YOLOv3-tiny network can be obtained by Formulas (2) and (3), and the final bounding box prediction schematic is shown in Figure 2. The confidence is divided into two parts: one is the probability of the existence of the object, showed by $P_r(object)$ (if the object exists, $P_r(object) = 1$, otherwise it is 0), while the other is the accuracy of the predicted bounding box, which is shown in Formula (4).

$$b_x = \sigma(t_x) + c_x \quad b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w e^{t_w} \quad b_h = p_h e^{t_h} \tag{3}$$

$$C_{conf} = P_r(class_i|object) \times P_r(object) \times IOU_{pred}^{truth} = P_r(class) \times IOU_{pred}^{truth} \tag{4}$$

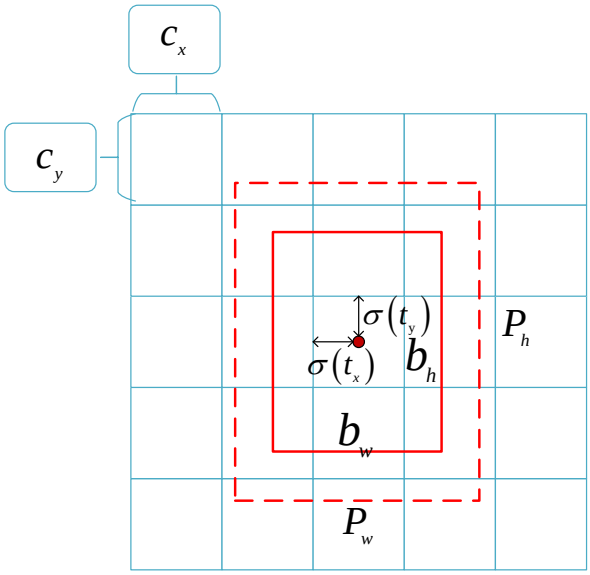


Figure 2. The final bounding box prediction schematic.

In Formulas (2) and (3), b_x and b_y are the coordinates of the center point of the modified bounding box; b_w and b_h represent the width and height of the modified bounding box, respectively; t_x and t_y represent the offset between the object center point and the upper-left corner of the grid; t_w and t_h represent the offset of the width and height of the predicted bounding box, respectively; c_x and c_y represent the offset of the grid relative to the upper-left corner. p_w and p_h are the width and height of the prior box, respectively. The sigmoid function is used to control the range of value within (0, 1) and control the offset of the object center within the corresponding grid cell to ensure that it is not out of bounds. In Formula (4), C_{conf} represents the confidence score of a specific category for each box; $P_r(class_i|object)$ represents the probability of predicting C conditional class in each grid cell (I = 1, 2, . . . , C); $P_r(object) \times IOU_{pred}^{truth}$ represents the confidence score; IOU_{pred}^{truth} represents the intersection ratio of the ground truth box and the prediction box.

3. SAS-YOLOv3-Tiny Algorithm

The original YOLOv3 algorithm has a considerable computation cost and parameters, which are not suitable for deployment on mobile devices. Therefore, YOLOv3 does not satisfy the specific application domain, such as helmet detection. Even though YOLOv3-tiny can meet the practical needs in terms of computation amount and number of parameters, which is not as accurate as YOLOv3 due to model compression. To further reduce the number of parameters and the amount of calculation, the Sandglass-Residual module will be

proposed in Section 3.1. Meanwhile, the channel attention mechanism will be fused into the Sandglass-Residual module to extract more valuable features. In Section 3.2, the improved SPP module will be introduced into the SAS-YOLOv3-tiny network architecture to obtain local and global features. In Section 3.3, we will show the overall network architecture of SAS-YOLOv3-tiny, which utilizes three-scale feature prediction to promote the small-scale objects' detection performance. CIoU loss will be applied to the original loss function to improve position accuracy in Section 3.4.

3.1. Sandglass-Residual Module Based on Channel Attention Mechanism

The inverted residual module of MobileNetv2 [23] places the shortcut on the low-dimensional representations. Feature compression will cause some problems that optimization is complicated, and the gradient is easy to shake, affecting the convergence of the model. MobileNext [24] proposes a new sandglass bottleneck module to solve the inverted residual module problem, which puts the shortcut on the high-dimensional representations. The above operations can retain the advantages of high-speed convergence and training on the high-dimensional network and take advantage of the computational advantages of depthwise separable convolution. In general, the parameters and calculation amount of traditional convolution increase significantly with the increase of convolution layers. So the conventional convolution is replaced with depthwise separable convolution to reduce model complexity, which is transformed into two parts: depthwise convolution and point convolution. We assume that the size of input feature map is $D_F \times D_F \times M$, the size of output feature map is $D_F \times D_F \times N$ and the size of standard convolution kernel is $D_K \times D_k \times M$. The computation amount of standard convolution is $D_K \times D_K \times M \times N \times D_F \times D_F$. In the depthwise convolution operation, the size of convolution kernel is $D_k \times D_k \times 1$ and its number is M. In the point convolution operation, the size of convolution kernel is $1 \times 1 \times M$ and its number is N. so the computation amount of depthwise separable convolution is $D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F$. By comparing the computational amount of the two, the computational amount of depthwise separable convolution can be reduced to $1/N + 1/D_K^2$ of the standard convolution.

In our work, the Sandglass-Residual module based on the lightweight idea is constructed in the feature extraction process, ensuring that more information is passed from the bottom to the top and gradient propagation is facilitated. The Specific operations are as follows. In the high-dimensional space, two depthwise convolutions with kernel size of 3×3 are performed, which can encode more spatial information. The point convolution with kernel size of 1×1 is utilized to reduce and increase channels' dimensions and encode information between channels. The first depthwise convolution and the last point convolution use nonlinear activation functions. In contrast, the first point convolution and the final depthwise convolution directly perform linear output to avoid information loss. The parameters of the Sandglass-Residual module are shown in Table 1.

Table 1. The parameters of Sandglass-Residual module.

Input	Operation	Output
$h \times w \times n$	3×3 Dwise conv, leaky	$h \times w \times n$
$h \times w \times n$	1×1 conv, linear	$h \times w \times \frac{n}{2}$
$h \times w \times \frac{n}{2}$	1×1 conv, leaky	$h \times w \times n$
$h \times w \times n$	3×3 Dwise conv, linear	$h \times w \times n$

The YOLOv3-tiny algorithm is applied to a real-world scenario dataset, objects in the image are treated equally. If the weight is assigned to the features of the object area, the weighted feature maps will be conducive to detecting far-distance and small-scale safety helmets, which can improve detection accuracy without introducing too many parameters. The Squeeze-Excitation (SE) channel attention module in SENet [25] gives different weights to different channels in the feature map of the convolutional neural network, making the network pay more attention to the channels with higher weights.

Thus, it can enhance the learning ability of the network, and its specific operations are as follows. The feature map with size of $H \times W \times C$ is compressed into a vector that its size is $1 \times 1 \times C$ by compression operation (i.e., global average pooling operation). Then the weights of different channels are obtained by excitation operation (i.e., two fully connection operations), and finally, the feature weighting operation is carried out on the obtained feature maps. After the above operations, the attention feature maps are produced. All channels of the feature maps generated by the above Sandglass-Residual module are treated equally, which makes some essential features be overlooked so that these obtained features are not conducive to detecting difficult-to-distinguish objects. Therefore, in this paper, the channel attention is introduced into the Sandglass-Residual module to extract informative features, adjusting the characteristic relationship between network models by squeeze and excitation operations. Its structure is shown in Figure 3. Compared with the original SR block, the Sandglass-Residual module based on the Squeeze-Excitation channel attention enhances the network's nonlinear characteristics, which can improve the model generalization ability without changing the output dimension. The subsequent ablation experiments prove that the Sandglass-Residual module based on the Squeeze-Excitation channel attention is good for improving the detection performance.

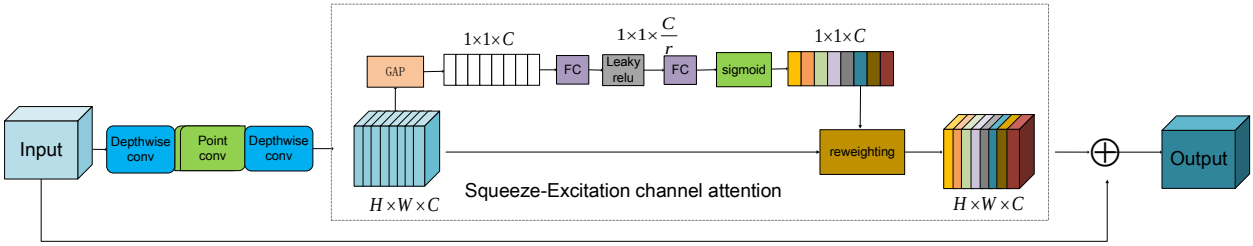


Figure 3. Sandglass-Residual module based on the Squeeze-Excitation channel attention.

3.2. Improved Spatial Pyramid Pooling Module

To obtain the context semantic information of different receptive fields and further improve the detection accuracy of the model, an improved spatial pyramid pooling (SPP) module is added into the improved backbone network. The traditional spatial pyramid pooling [2] is to solve the problem that the input of the fully connected layer must be a fixed eigenvector, which allows us to build a network that supports images of any size to input without cropping and scaling operations. The spatial pyramid pooling module in this paper integrates multi-scale local feature information with global feature information to obtain richer feature representations, which is shown in Figure 4.

After going through the improved SPP module, the feature map's size stays the same, realized by the pooling operation of stride one and the padding method. Specifically, the final feature map with size of $14 \times 14 \times 1024$ extracted from the backbone network already contains rich semantic information. After that, three max-pooling operations are adopted to obtain three kinds of feature maps, which are concatenated with the input feature map with size of $14 \times 14 \times 1024$ along the channel dimension to produce the feature map of size $14 \times 14 \times 4096$ as the output. 5×5, 9×9, 13×13 are the size of the pooling kernel, while the stride is 1. The experiments show that the improved SPP module is added after the backbone network to extract rich features, improving the detection effect.

3.3. Network Architecture of SAS-YOLOv3-Tiny

To solve low detection accuracy and high missing rate of YOLOv3-tiny on small objects such as helmets, we have improved the original network. The network structure of SAS-YOLOv3-tiny is shown in Figure 5. The backbone network of SAS-YOLOv3-tiny is constructed by combining the previously made Sandglass-Residual module based on the Squeeze-Excitation channel attention and the improved SPP module based on spatial pyramid pooling. To be specific, in Figure 5, the dashed line part is the feature extraction

part of the backbone network, in which five brown DBLs in the middle of the backbone network are the 1×1 convolution layer of stride 2 to replace the max-pooling layer to perform down-sampling operations and change the number of channels. The five Sandglass-Residual in the middle of the backbone are the Sandglass-Residual modules based on the Squeeze-Excitation channel attention to replace the standard convolution layer behind the max-pooling layer of the original backbone network. Furthermore, to make the network more robust, we add the improved SPP module at the end of the backbone network to fully extract local and global features.

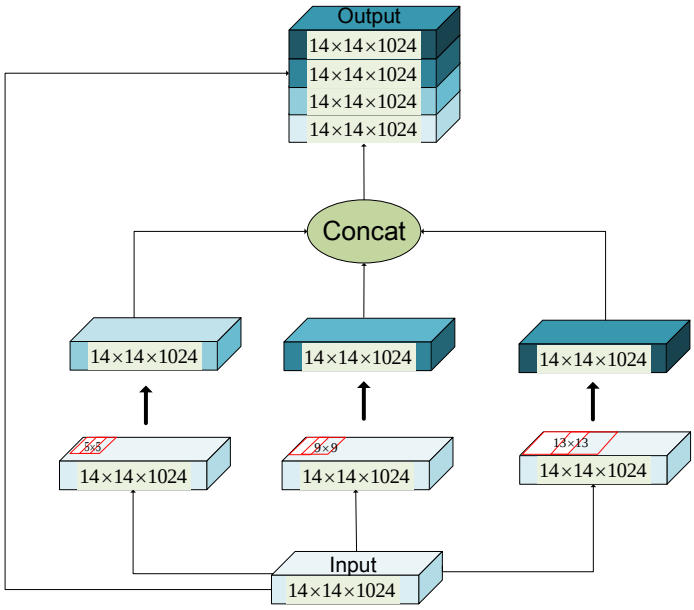


Figure 4. Improved spatial pyramid pooling module.

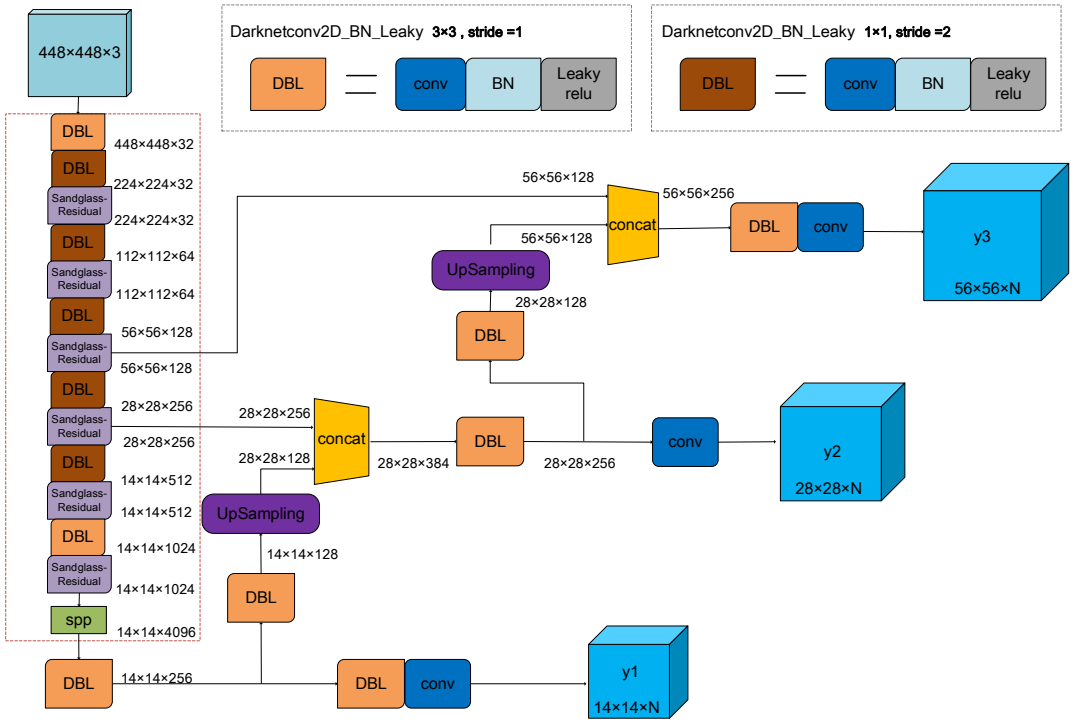


Figure 5. SAS-YOLOv3-tiny network structure.

Simultaneously, we improved the method of multi-scale feature fusion. Based on the original network's two-scale feature prediction, a downsampled feature map with size of $56 \times 56 \times 128$ is used to form the three-scale feature prediction to improve object detection accuracy further. In addition to being used as prediction, the feature map with size of $28 \times 28 \times 384$ after a convolution operation continue to go through a convolution layer and an upsampled layer, and then is concatenated with the feature map with size of $56 \times 56 \times 128$ to perform prediction. At scale y3, the feature map downsampled by $8\times$ is utilized to detect small objects, because it can get more detailed features and location information of small objects. In the improved algorithms, nine prior boxes instead of six prior boxes are utilized, and the corresponding relationship between feature maps and prior boxes is as follows. Feature maps of size 14, 28, 56 correspond [(373,326); (156,198); (116,90)], (59,119); (62,45); (30,61)], [(10,13); (16,30); (33,23)], respectively.

3.4. Improved Loss Function

Recently, in terms of bounding box regression, *IOU* loss optimizations have replaced previous regression loss optimizations (MSE loss, L1-Smooth loss, etc.). One of the most commonly used evaluation criteria for the performance of object detection algorithms is intersection over union (*IOU*), which is the ratio of the overlap area of the ground truth box and the prediction box to the total area of the two boxes, as shown in Formula (5). In Formula (5), $A = (x, y, w, h)$ represents the prediction box and $B = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$ represents the ground truth box.

$$IOU = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

Even though *IOU* can reflect the detection effect of the prediction box and the ground truth box, it only works when the bounding boxes overlap and does not provide any adjustment gradient for the non-overlapped part. The concept of *IOU* is based on the ratio, so it is insensitive to the object scale. In this paper, in Formula (6), the traditional regression loss MSE is replaced with CIoU [26], whose detection effect is more conducive to the actual scene. It inherits the advantages of the Generalized Intersection Over Union (GIoU) [27] and Distance-IoU(DIoU) [28], which not only considers the distance and overlap ratio but also considers the scale and the aspect ratio between the prediction box and the ground truth box so that it can carry out the bounding box regression better. The complete definition of CIoU loss function is shown in Formula (7). Therefore, the loss function of SAS-YOLOv3-tiny is shown in Formula (6), which is divided into three parts: $loss_{CIoU}$ represents the regression loss, $loss_{obj}$ represents the confidence loss and $loss_{\text{class}}$ represents the category loss, they are shown as shown in Formulas (7)–(9).

$$LOSS = loss_{CIoU} + loss_{obj} + loss_{class} \tag{6}$$

$$loss_{CIoU} = 1 - CIoU \quad CIoU = IOU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{7}$$

$$\begin{aligned} loss_{obj} = &- \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{obj} \left[\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i) \right] \\ &- \lambda_{noobj} \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{noobj} \left[\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i) \right] \end{aligned} \tag{8}$$

$$loss_{class} = - \sum_{i=0}^{K \times K} I_{ij}^{obj} \sum_{c \in classes} \left[\hat{p}_i(c) \log(p_i(c)) + (1 - \hat{p}_i(c)) \log(1 - p_i(c)) \right] \tag{9}$$

In Formula (7), b and b^{gt} represent the center points of the prediction box and the ground truth box, respectively. Meanwhile, $\rho^2(b, b^{gt})$ represents the Euclidean distance between two center points, c represents the diagonal distance of the smallest closure region that can contain both the prediction box and the ground truth box, α is the weight parameter

and v is used to measure the similarity of aspect ratios. In Formulas (8) and (9), $K \times K$ represents the size of the final feature map to be detected; I_{ij}^{obj} is used to determine whether the j-th prior box in the i-th grid is responsible for the object. If it is responsible for the object, it has a value of 1. Otherwise, its value is 0. The weight coefficients λ_{coord} and λ_{noobj} are set at 5 and 0.5, respectively, which are used to offset the imbalance between positive and negative samples.

4. Experiments and Results Analysis

In Section 4, some experiments and results analysis will be explained in detail. The basic information of safety helmet detection dataset and evaluation criteria of detection effect will be introduced in Section 4.1. Then, we will explain the experimental progress and do a result analysis in Section 4.2. There are four subsections in Section 4.2. In Section 4.2.1, we will describe the training setting. In Section 4.2.2, we will do ablation experiments to prove the effectiveness of each scheme. In Section 4.2.3, we will conduct the comparison of results with other state-of-the-art detection models. In Section 4.2.4, we will show the detection results of some samples under different detection models.

4.1. Dataset and Evaluation Criteria

4.1.1. Dataset

Dataset is crucial for deep learning-based object detection algorithms. In our work, a set of safety helmet datasets was made, which contained 7656 images and was obtained by searching on the Internet, taking photos with cameras and web crawlers, and the format was produced in VOC format. Labeling software (labelImg) was used to label the collected images. There were four categories of objects: helmet (wear a safety helmet for two-wheelers), cap (wear a non-protective hat), Nowear (wear nothing) and safety-cap (wear an industrial helmet). Additionally, the annotated image coordinate information was saved as an XML file. Next, the training set, the validation set and the test set were randomly divided, and the 8:1:1 ratio was adopted in our study, so there were 6063 training samples, 827 validation samples and 766 test samples. Specifically, the training set was used to train parameters of neural network. The validation set was used to test the effect of the current model after each epoch. The test set was used to test the model's final generalization performance because it did not participate in the training process at all.

4.1.2. Evaluation Criteria

The quality of the detection effect usually needs a certain standard to evaluate, so the following evaluation criteria are introduced.

(1) The formulas of the Precision and Recall are shown in Formula (10), and the formula of F1 is shown in Formula (11). F1 is the harmonic mean of Precision and Recall. In Formula (10), True Positives (TP, $IOU \geq$ threshold) refers to the number of instances that are actually positive examples and are classified as positive examples by the classifier. False Positives (FP, $IOU <$ threshold) refers to the number of instances that are actually negative examples but are classified as positive examples by the classifier. False Negatives (FN, undetected ground truth box) refers to the number of instances that are actually positive examples but are classified as negative examples by the classifier.

$$\mathrm{P}_{precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \quad \mathrm{R}_{recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{10}$$

$$\mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P}+\mathrm{R}} \tag{11}$$

(2) The formulas of Average Precision (AP) and Mean Average Precision (mAP) are shown in Formula (12).

$$\mathrm{AP} = \int_0^1 \mathrm{P}(\mathrm{R})d\mathrm{R} \quad \mathrm{mAP} = \frac{\sum_{i=1}^{\mathrm{N}} \mathrm{AP}_i}{\mathrm{N}} \tag{12}$$

In Formula (12), N represents the number of object categories. In general, the increase of the Recall is often accompanied by a decrease in Precision. To better balance the two, the P-R curve is introduced, and the area below it is the AP value of a specific category.

4.2. Experimental Progress and Result Analysis

4.2.1. Training Setting

This paper's experimental platforms were Intel(R) Core (TM) I7-9700 CPU @3.00 GHz processor and NVIDIA GeForce RTX 2080Ti GPU. The programming language used for the algorithm in this paper was Python 3.8. The deep learning framework Pytorch 1.6.0 was used. The operating system used was Ubuntu18.04, and other dependent libraries were configured. Generally speaking, there were two training methods to train the model. One method was that random initial weights were used to train the model. The other is that pre-training weights were used to train the model. This paper used the first method to train the model to compare the different modification methods. In the experiment, the SAS-YOLOv3-tiny network was trained from scratch by using a self-built dataset. To ensure the fairness of the test, we retrained the YOLOv3-tiny, YOLO v3 and v4 [29] in the same experimental environment to obtain the corresponding detection model for experimental comparison results of the improved algorithm model on the validation set and the test set. Some experimental parameters were set as follows. In the experiment, the batch size was set to 4; 140 epochs were trained; the cosine learning rate strategy was used, which changed the learning rate from 0.01 to 0.0005; momentum was set to 0.937; weight decay was set to 0.000484. In addition, the multi-scale training strategy was adopted to improve the detection effect of the network for images of different input resolutions, and the cut size was selected at {320, 352, 384, 416, 448, 480, 512, 544, 576, 608, 640} for training in each iteration.

4.2.2. Ablation Experiments

In this section, to better understand the influence of each improved method on the detection effect, ablation learning is carried out on the self-built helmet validation set. First of all, we first presented each of our schemes in Table 2. Then, we compared different modification schemes based on YOLOv3-tiny in terms of indicators including P, R, F1, mAP, Weight, Total Parameters and average time of detecting a single image (detection time) in Table 3, and comprehensively analyze how each improvement point promote performance. Finally, we demonstrated the effectiveness of the improvement point by presenting a training curve for each scheme.

Table 2. Different improvement schemes.

Scheme	SR	3-Scale	SPP	SE	CIoU
SR	√				
SR-3s	√	√			
SR-3s-SPP	√	√	√		
SR-3s-SPP-SE	√	√	√	√	
SR-3s-SPP-SE-CIoU (Ours)	√	√	√	√	√

Table 3. Ablation results of different models on the validation set.

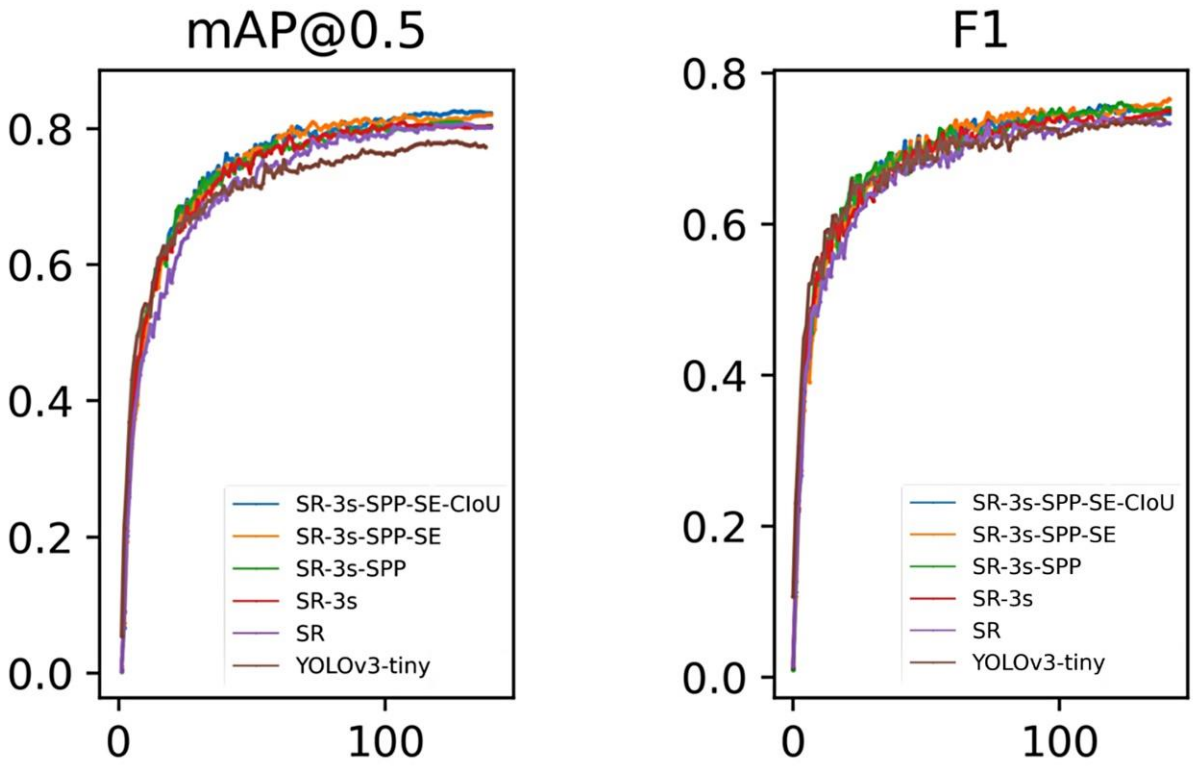
Model	P/%	R/%	mAP/%	F1/%	Weight/MB	Total Parameters/10^6	Detection Time/ms
YOLOv3-tiny	70.7	73.3	73.7	71.9	69.5	8.67681	2.5
SR	69.3	77.9	78.2	73.3	36.5	4.53490	2.8
SR-3s	69.6	79.6	80.2	74.2	39.2	4.86658	3.0
SR-3s-SPP	70.3	80.4	80.1	75.0	45.4	5.65301	3.1
SR-3s-SPP-SE	72.3	80.2	81.2	76.0	46.9	5.82773	3.2
SR-3s-SPP-SE-CIoU (Ours)	73.2	80.2	81.6	76.4	46.9	5.82773	3.2

The different schemes are shown in Table 2. We used the yolov3-tiny algorithm as the baseline. Specifically, in the scheme SR, the Sandglass-Residual (SR) module was used to replace the original convolution layer, and the max-pooling layer was replaced with the convolution layer of stride two. In the scheme SR-3 scale (3 s), on the basis of the scheme SR, a three-scale prediction method was adopted. In the Scheme SR-3s-SPP, to further improve the detection effect, the improved SPP was utilized on the basis of the Scheme SR-3s. In addition, in the Scheme SR-3s-SPP-SE, the Squeeze-Excitation (SE) channel attention mechanism was integrated into the Sandglass-Residual module to extract more representative features on the basis of the Scheme SR-3s-SPP. In the Scheme SR-3s-SPP-SE-CIoU, we used CIoU loss on the basis of the Scheme SR-3s-SPP-SE. A combination of five improvements formed our final algorithm, in other words, the last Scheme SR-3s-SPP-SE-CIoU was our improved algorithm (named SAS-YOLO-v3-tiny).

The ablation results of different models on the validation set are shown in Table 3. From Table 3, we can see that the values of indicators including P, R, mAP, F1 are low in the original YOLOv3-tiny algorithm. Additionally, the indicators, including the weight size of the model and total parameters, still have room for improvement. Compared with the original algorithm, the improved YOLOv3-tiny based on the Sandglass-Residual module made the network more lightweight; this was because the Scheme SR based on depthwise separable convolution reduced the number of parameters and computation amount, reducing the size of weight files and the number of parameters by nearly half. In addition, owing to putting the shortcut on the high-dimensional representations, the SR module could extract rich feature, which could increase R by 4.6%, increase mAP by 4.5% and increase F1 by 1.4% while keeping the detection speed almost unchanged. The Scheme SR-3s changed two-scale feature prediction into three-scale feature prediction, which could incorporate shallow features with sufficient location information, making R, mAP, F1 increase by 1.7%, 2%, 0.9%, respectively. The introduction of the improved SPP module in the Scheme SR-3s-SPP could extract feature with different receptive fields, which can further improve P, R, F1 by 0.7%, 0.8%, 0.8%, respectively. Based on the Scheme SR-3s-SPP-SE, the channel attention mechanism was introduced into the Sandglass-Residual module, which could pay attention to useful feature, improving P, mAP and F1 by 2.0%, 1.1%, 1.0%, respectively. Further, CIoU loss was utilized in the final Scheme SR-3s-SPP-SE-CIoU to promote positioning accuracy, which could improve P by nearly 1%. Due to the combination of the above improved methods, compared with the original YOLOv3-tiny, SAS-YOLOv3-tiny had advantages on model performance and complexity. Specifically, it improved P by 2.5%, improved R by 6.9%, improves mAP by 7.9%, improved F1 by 4.5% over the original algorithm on the validation set and had a smaller number of parameters than the original algorithm at a sacrifice of only 0.8 ms.

To further demonstrate the effectiveness of different schemes, we presented the curves in the training process for six groups of experiments. Two critical performance indicators are mAP and F1, the curves of F1 and mAP in the different models are shown in Figure 6a,b. The horizontal axis in the Figure 6a,b represents the training time, while the vertical axis represents the value of F1 and mAP, respectively. The YOLOv3-tiny represents the training curves of the original algorithm, in which the mAP value and F1 value are the lowest. The SR represents the training results of the Scheme SR, the main reason for the promotion of performance is utilization of the Sandglass-Residual module. The SR-3s represents the training process of the Scheme SR-3s, in which the Sandglass-Residual module and the three-scale feature prediction are applied simultaneously, promoting further enhancement in terms of the mAP and the F1. The SR-3s-SPP represents the training curves of the Scheme SR-3s-SPP, in which not only the Sandglass-Residual module and the three-scale feature prediction are adopted, but also the SPP module is employed. The application of the SPP module showed that the training process was easier to converge and the results were more robust. The SR-3s-SPP-SE represents the training process of the Scheme SR-3s-SPP-SE, in which the channel attention mechanism was introduced on the basis of the above three improvement methods, indicating that the training model had reached a better

level. The SR-3s-SPP-SE-CIoU represents the results of the Scheme SR-3s-SPP-SE-CIoU, in which CIoU was utilized on the basis of the previous methods, proving that the CIoU could promote the positing accuracy. As shown in Figure 6a,b, we can intuitively see that the final model is better than the original algorithm.



(**a**) The curves of mAP (**b**) The curves of F1

Figure 6. The curves of F1 and mAP in the different models.

4.2.3. Result Comparison with Other Detection Models

To prove each algorithm's generalization performance, we compare and analyze different evaluation indexes for different algorithms on the test set, which are shown as shown in Table 4. From Table 4, we can draw the following conclusions. On the test set, compared with the YOLOv3-tiny, SAS-YOLOv3-tiny improves R from 75.4% to 80.9% improves mAP from 74.6% to 80.3%, improves F1 from 72.9% to 75.2% and reduces the size of weight file from 69.5 MB to 46.9 MB, which mainly benefits from the use of SR module based on channel attention and SPP module, the application of three-scale prediction method, and the introduction of CIoU loss. Compared with the latest YOLOv4-tiny, SAS-YOLOv3-tiny improves R from 80.0% to 80.9% and improves mAP from 78.9% to 80.3%, but P and F1 decreases to some extent. The main reason is that the idea of the Cross Stage Partial network (CSPNet) [30] is applied into the YOLOv4-tiny, which strengthen the network feature representation. Compared with the YOLOv3 and the YOLOv4, SAS-YOLOv3-tiny has tremendous advantages in terms of the number of parameters and speed, although its accuracy is inferior to the YOLOv3 and the YOLOv4, and the main reason for the decrease of accuracy lies in less parameters and small computational burden.

Table 4. Comparison of different algorithms for object detection on the test set.

Type / Algorithm	Helmet	AP/ Cap	% Nowear	Safey-Cap	P/%	R/%	mAP/%	F1/%	Weight/MB	Total Parameters/10^7	Detection Time/ms
YOLOv3-tiny	70.9	74.0	76.2	77.5	71.0	75.4	74.6	72.9	69.5	0.86768	2.5
YOLOv4-tiny	80.4	74.7	80.8	79.8	72.6	80.0	78.9	76.0	47.2	0.58779	1.8
YOLOv3	84.2	81.2	92.5	86.6	75.8	85.6	86.1	80.3	492.8	6.15399	8.6
YOLOv4	86.7	80.9	92.7	89.3	79.4	88.6	87.4	83.7	420.7	5.24798	7.4
SAS-YOLOv3-tiny	78.2	73.3	87.8	81.9	71.6	80.9	80.3	75.2	46.9	0.58277	3.2

4.2.4. Detection Results under Application Scenarios

To prove that the improved algorithm is more suitable for natural complex scenes in terms of accuracy, we show the detection effect of some test images, which are shown in Figure 7. For small-scale objects, occluded objects and dense objects, SAS-YOLOv3-tiny is superior to the YOLOv3-tiny algorithm. As can be seen from the first and second set of images, SAS-YOLOv3-tiny and the latest YOLOv4-tiny can detect all objects, but YOLOv3-tiny leaves out an ordinary object. As can be seen from the third set of images, SAS-YOLOv3-tiny can detect all objects while YOLOv4-tiny neglects a helmet object, and YOLOv3-tiny detects some of the objects incorrectly. For detecting small objects at long distances, SAS-YOLOv3-tiny has better performance than YOLOv3-tiny and YOLOv4-tiny. In the last set of images, SAS-YOLO-v3-tiny and YOLOv4-tiny can detect some standard objects, but they will miss objects to be detected when a man deliberately lowers his head. As can be seen from the above test images, the improved algorithm is superior to the original algorithm and sometimes even has a better detection effect than the latest YOLOv4-tiny.

Figure 7. *Cont.*

YOLOv3-tiny SAS-YOLOv3-tiny YOLOv4-tiny

Figure 7. The detection results of some test images.

5. Conclusions

In this paper, the SAS-YOLOv3-tiny algorithm is proposed to solve the problem that the original lightweight algorithm YOLOv3-tiny was low at accuracy. Even though YOLOv3-tiny has a faster speed and fewer parameters, its detection accuracy needs to be improved. First of all, the lightweight Sandglass-Residual module based on depthwise separable convolution and channel attention mechanism was constructed to replace the original convolution layer while the max-pooling layer was replaced with the convolution layer of stride two, which could reduce the number of parameters and improve detection performance. Furthermore, the detection performance is further improved when three-scale feature prediction is utilized. Next, the improved spatial pyramid pooling module was merged behind the backbone network to extract expressive features. Finally, we utilized CIoU to improve the loss function, which also improved the location effect. In conclusion, for the validation set, SAS-YOLOv3-tiny made P from 70.7% to 73.2%, made R from 73.3% to 80.2%, made mAP from 73.7% to 81.6% and made F1 from 71.9% to 76.4%. For the test set, SAS-YOLOv3-tiny had good generalization, and it performed better than the original YOLOv3-tiny at the expense of 0.7 ms speed, which was comparable to YOLOv4-tiny in terms of detection accuracy; compared with the heavyweight algorithms YOLOv3 and YOLOv4, SAS-YOLOv3-tiny had a great advantage in speed although its detection accuracy was not as good as theirs. The experimental results and contrast curves reveal that the improved methods can strengthen the effect of detection. The next work is to expand the safety helmet dataset based on the dataset in this paper and further improve the detection accuracy while maintaining a lower number of parameters and speed.

Author Contributions: Conceptualization, X.H. and R.C.; methodology, R.C. and X.H.; software, R.C.; validation, X.H. and R.C.; formal analysis, X.H. and R.C.; investigation, R.C., X.H. and Z.W.; resources, X.H. and Z.Z.; data curation, R.C. and X.H.; writing—original draft preparation, R.C.; writing—review and editing, X.H. and R.C.; visualization, X.H. and R.C.; supervision, X.H.; project administration, X.H. and Z.Z.; funding acquisition, X.H. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC) (61572023, 61672467).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Some or all data, models or code generated or used during the study are available from the corresponding author by request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Girshick, R.; Donahue, J.; Darrelland, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Seg-mentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014.
2. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
3. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
5. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Proceedings of the Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
6. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
7. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R.B. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef] [PubMed]
8. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; Lecun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv* **2013**, arXiv:1312.6229.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y. SSD: Single Shot Multi Box Detector. In Proceedings of the Europe Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
11. Lin, T.-Y.; Goyal, P.; Girshick, R.B.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
12. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
13. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Scheme. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1854–1862.
14. Huang, L.; Yang, Y.; Deng, Y.; Yu, Y. DenseBox: Unifying Landmark Localization with End to End Object Detection. *arXiv* **2015**, arXiv:1509. 04874.
15. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *Int. J. Comput. Vis.* **2020**, *128*, 642–656. [CrossRef]
16. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-Up Object Detection by Grouping Extreme and Center Points. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 850–859.
17. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6568–6577.
18. Zhu, C.; He, Y.; Savvides, M. Feature Selective Anchor-Free Module for Single-Shot Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 840–849.
19. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9626–9635.
20. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Li, L.; Shi, J. FoveaBox: Beyound Anchor-Based Object Detection. *IEEE Trans. Image Process.* **2020**, *29*, 7389–7398. [CrossRef]
21. Wang, H.; Hu, Z.; Guo, Y.; Yang, Z.; Zhou, F.; Xu, P. A Real-Time Safety HelmetWearing Detection Approach Based on CSYOLOv3. *Appl. Sci.* **2020**, *10*, 6732. [CrossRef]
22. Li, Y.; Wei, H.; Han, Z.; Huang, J.; Wang, W. Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks. *Adv. Civ. Eng.* **2020**, *2020*, 1–10. [CrossRef]
23. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
24. Daquan, Z.; Hou, Q.; Chen, Y.; Feng, J.; Yan, S. Rethinking Bottleneck Structure for Efficient Mobile Network Design. *arXiv* **2020**, arXiv:2007.02269.
25. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.

26. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. *arXiv* **2020**, arXiv:2005.03572.
27. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 658–666.
28. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. 2019, pp. 1458–1467. Available online: https://arxiv.org/abs/1911.08287 (accessed on 9 March 2020).
29. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object detection. *arXiv* **2020**, arXiv:2004.10934.
30. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Glasgow, UK, 23–28 August 2020; pp. 390–391.

Detecting motorcycle helmet use with deep learning

Preprint *in* Accident; Analysis and Prevention · October 2019

CITATIONS
0

READS
2,227

2 authors:

Felix Wilhelm Siebert
Technical University of Denmark
52 PUBLICATIONS **256** CITATIONS
SEE PROFILE

Hanhe Lin
University of Dundee
67 PUBLICATIONS **632** CITATIONS
SEE PROFILE

Detecting motorcycle helmet use with deep learning

Felix Wilhelm Siebert[a,*], Hanhe Lin[b]

[a] *Department of Psychology and Ergonomics, Technische Universität Berlin, Marchstraße 12, 10587 Berlin, Germany*
[b] *Department of Computer and Information Science, Universität Konstanz, Universitätsstraße 10, 78464 Konstanz, Germany*

Abstract

The continuous motorization of traffic has led to a sustained increase in the global number of road related fatalities and injuries. To counter this, governments are focusing on enforcing safe and law-abiding behavior in traffic. However, especially in developing countries where the motorcycle is the main form of transportation, there is a lack of comprehensive data on the safety-critical behavioral metric of motorcycle helmet use. This lack of data prohibits targeted enforcement and education campaigns which are crucial for injury prevention. Hence, we have developed an algorithm for the automated registration of motorcycle helmet usage from video data, using a deep learning approach. Based on 91,000 annotated frames of video data, collected at multiple observation sites in 7 cities across the country of Myanmar, we trained our algorithm to detect active motorcycles, the number and position of riders on the motorcycle, as well as their helmet use. An analysis of the algorithm's accuracy on an annotated test data set, and a comparison to available human-registered helmet use data reveals a high accuracy of our approach. Our algorithm registers motorcycle helmet use rates with an accuracy of -4.4% and +2.1% in comparison to a human observer, with minimal training for individual observation sites. Without observation site specific training, the accuracy of helmet use detection decreases slightly, depending on a number of factors. Our approach can be implemented

*Corresponding author: Tel.: +49-(0)30-314-229-67;
Email address: felix.siebert@tu-berlin.de (Felix Wilhelm Siebert)

Preprint accepted by Accident Analysis & Prevention

in existing roadside traffic surveillance infrastructure and can facilitate targeted data-driven injury prevention campaigns with real-time speed. Implications of the proposed method, as well as measures that can further improve detection accuracy are discussed.

Keywords: Deep learning, Helmet use detection, Motorcycle, Road safety, Injury prevention

---

1. Introduction

Using a motorcycle helmet can decrease the probability of fatal injuries of motorcycle riders in road traffic crashes by 42% [1] which is why governments worldwide have enacted laws that make helmet use mandatory. Despite this, compliance with motorcycle helmet laws is often low, especially in developing countries [2, 3, 4]. To efficiently conduct targeted helmet use campaigns, it is essential for governments to collect detailed data on the level of compliance with helmet laws. However, 40% of countries in the world do not have an estimate of this crucial road safety metric [5]. And even if data is available, helmet use observations are frequently limited in sample size and regional scope [6, 7], draw from data of relatively short time frames [8, 9], or are only singularly collected in the scope of academic research [4, 10]. The main reason for this lack of comprehensive continuous data lies in the prevailing way of helmet use data collection, which utilizes direct observation of motorcycle helmet use in traffic by human observers. This direct observation during road-side surveys is resource intensive, as it is highly time-consuming and costly [11]. And while the use of video cameras allows *indirect* observation, alleviating the time pressure of *direct* observation, the classification of helmet use through human observers limits the amount of data that can be processed.

In light of this, there is an increasing demand to develop a reliable and timely efficient intelligent system for detecting helmet use of motorcycle riders that does not rely on a human observer. A promising method for achieving this automated detection of motorcycle helmet use is machine learning. Machine learning has

been applied to a number of road safety related detection tasks, and has achieved high accuracy for the general detection of pedestrians, bicyclists, motorcyclists and cars [12]. While first implementations of automated motorcycle helmet use detection have been promising, they have not been developed to their full potential. Current approaches rely on data sets that are limited in the overall number of riders observed, are trained on a small number of observation sites, or do not detect the rider position on the motorcycle [13, 14]. In this paper a deep learning based automated helmet use detection is proposed that relies on a comprehensive dataset with large variance in the number of riders observed, drawing from multiple observation sites at varying times of day.

Recent successful deep learning based applications of computer vision, e.g. in image classification [15, 16, 17], object detection [18, 19], and activity recognition [20, 21] have heavily relied on large-scale datasets, similar to the one used in this study. Hence, the next section of this paper will focus on the generation of the underlying dataset and its annotation, to facilitate potential data collection in other countries with a similar methodology. This is followed by a section on algorithm training. In the subsequent sections of this paper, the algorithm performance is analyzed through comparison with an annotated test data set and with results from an earlier observational study on helmet use in Myanmar, conducted by human observers [4].

2. Dataset creation and annotation

2.1. Data collection and preprocessing

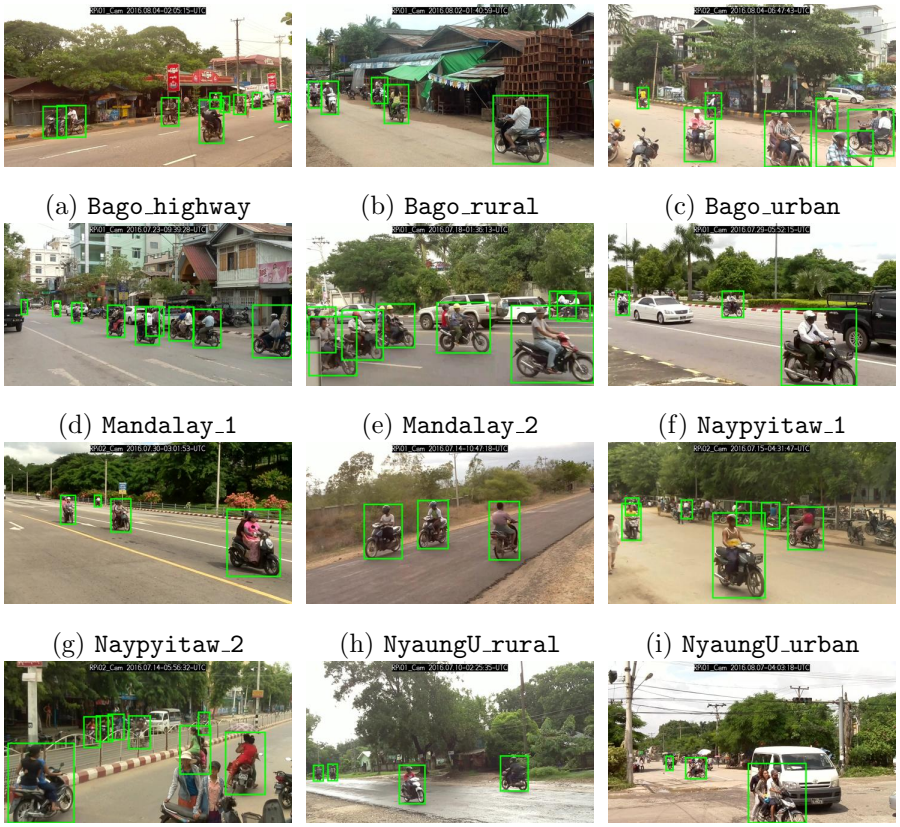
Myanmar was chosen as the basis for the collection of the source material for the development of the algorithm, since its road user aggregate and rapid motorization are highly representative of developing countries in the world [22] and video recordings of traffic were available from an earlier study [4]. Motorcyclists comprise more than 80% of road users in Myanmar [5], and the number of motorcycles has been increasing rapidly in recent years [23].

Throughout Myanmar, traffic was filmed with two video-cameras with a resolution of 1920×1080 pixels and a frame rate of 10 frames per second. Within seven cities, cameras were placed at multiple observation sites at approximately 2.5 m height and traffic was recorded for two consecutive days from approximately 6 am to 6:30 pm (Table 1). As the city of Mandalay has the highest number of motorcyclists in Myanmar the two cameras were installed for 7 days here. Yangon, the largest city of Myanmar, has an active ban on motorcycle in the city center, hence, one camera was placed in the suburbs here. Due to technical problems with the cameras and problems in reaching the selected observation sites, the number of hours recorded was not the same for each observation site. After the removal of blurred videos due to cloudy weather or rain, 254 hours of video data were available as the source material for this study. Video data was

Table 1: 254 hours of source video were available from 13 observation sites in 7 cities across Myanmar, from which 1,000 video clips (10 seconds / 100 frames each) were sampled for further annotation.

City	Population	Site ID	Duration (hours)	Sampled clips
Bago	254,424	`Bago_highway`	9	35
		`Bago_rural`	17	67
		`Bago_urban`	16	63
Mandalay	1,225,546	`Mandalay_1`	58	228
		`Mandalay_2`	48	190
NayPyiTaw	333,506	`Naypyitaw_1`	13	51
		`Naypyitaw_2`	11	43
Nyaung-U	48,528	`NyaungU_rural`	21	83
		`NyaungU_urban`	17	67
Pakokku	90,842	`Pakokku_urban`	19	75
Pathein	169,773	`Pathein_rural`	3	12
		`Pathein_urban`	12	47
Yangon	4,728,524	`Yangon_II`	10	39
			254	1,000

divided into 10 second video clips (100 frames each), which formed the basis for training, validating, and testing the algorithm in this study. The duration of video data available at each observation site is shown in Table 1. The observation sites represent a highly diverse data set, including multilane high traffic density road environments (e.g. Mandalay) as well as more rural environments (e.g. Pathein). Still frames of observation sites are presented in Figure 1.



(a) `Bago_highway` (b) `Bago_rural` (c) `Bago_urban`

(d) `Mandalay_1` (e) `Mandalay_2` (f) `Naypyitaw_1`

(g) `Naypyitaw_2` (h) `NyaungU_rural` (i) `NyaungU_urban`

(j) `Pakokku_urban` (k) `Pathein_rural` (l) `Yangon_II`

Figure 1: Still frames in 12 observation sites of 7 cities throughout Myanmar, where green rectangles correspond to annotations.

2.2. Sampling video clips

Since there were insufficient resources to annotate all 254 hours of recorded traffic, 1,000 videos clips were sampled which were most representative of the

source material. After segmenting the source material into non-overlapping video clips of 10 seconds length (100 frames), we applied the object detection algorithm YOLO9000 [24] with the pre-trained weights to detect the number of motorcycles in each frame, extracting those clips with the highest number of motorcycles in them. Multiple clips were sampled from each observation site, in proportion to the available videodata from each site. The resulting distribution of the 1,000 sampled video clips is presented in Table 1. The observation site `Pathein_urban` (47 video clips) was retroactively excluded from analysis due to heavy fogging on the camera which was not detected during the initial screening of the video data (Section 2.1). In addition, 43 video clips were excluded since they did not contain active motorcycles, as the YOLO9000 algorithm [24] had identified parked motorcycles.

2.3. Annotation

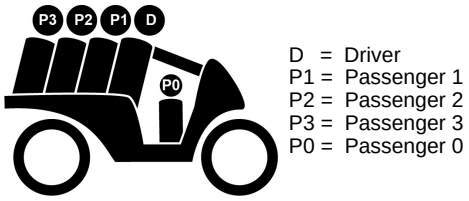
Videodata was annotated by first drawing a rectangular box around an individual motorcycle and its riders (so called *bounding box*), before entering information on the number of riders, their helmet use and position. All bounding boxes containing an individual motorcycle throughout a number of frames form the *motorcycle track*, i.e. an individual motorcycle will appear in multiple frames, but will only have one *motorcycle track*. To facilitate the annotation of the videos, we tested and compared the three image and video annotation tools BeaverDam [25], LabelMe [26], and VATIC [27]. We chose BeaverDam for data annotation, since it allows frame-by-frame labeling in videos, is easy to install, and has superior usability. Annotation was conducted by two freelance workers. An example of the annotation of an individual motorcycle through multiple frames (*motorcycle track*) is presented in Fig. 2.

For each bounding box, workers encoded the number of riders (1 to 5), their helmet use (yes/no) and position (Driver (D), Passenger (P0-3); Fig. 3). Examples of rider encoding are displayed in Fig. 3.

(a) (b)

(c) (d)

Figure 2: An example of motorcycle annotation. An individual motorcycle (marked in light green rectangles) appears on the left side of the frame and disappears on the lower right side of the frame.



(I)

(a) (b) (c) (d) (e)

(II)

Figure 3: Structure (I) and examples (II) of helmet use encoding: (a) `DHelmet`, (b) `DNoHelmet`, (c) `DHelmetP1NoHelmet`, (d) `DHelmetP1NoHelmetP2Helmet`, and (e) `DNoHelmetP0NoHelmetP1NoHelmetP2NoHelmet`.

Table 2: 910 annotated video clips were randomly split into training, validation and test sets according to individual observation site, with a split ratio of 70%, 10%, and 20%.

Site ID	Training set	Validation set	Test set	**Overall**
Bago_highway	24	4	7	35
Bago_rural	41	6	11	58
Bago_urban	44	6	13	63
Mandalay_1	159	23	45	227
Mandalay_2	111	16	31	158
Naypyitaw_1	36	5	10	51
Naypyitaw_2	30	4	9	43
NyaungU_rural	57	8	17	82
NyaungU_urban	47	7	13	67
Pakokku_urban	52	8	15	75
Pathein_rural	8	1	3	12
Yangon_II	27	4	8	39
Overall	636	92	182	910

2.4. Composition of annotated data

The 910 annotated video clips were randomly divided into three non-overlapping subsets: a training set (70%), a validation set (10%), and a test set (20%) (Table 2). Data on the number of annotated motorcycles in all 910 video clips can be found in Table 3. Overall, 10,180 motorcycle tracks (i.e. individual motorcycles) were annotated. As each individual motorcycle appears in multiple frames, there are 339,784 annotated motorcycles on a frame level, i.e. there are 339,784 bounding boxes containing motorcycles in the dataset. All motorcycles were encoded in classes, depending on the position and helmet use of the riders. This resulted in 36 classes, shown in Table 3. The number of motorcycles per class was imbalanced and ranged from only 12 observed frames (e.g., for motorcycles with 5 riders with no rider wearing a helmet) to 140,886 observed frames (one driver wearing a helmet). Some classes were not observed in the annotated video clips, e.g., there was no motorcycle with 4 riders all wearing a helmet.

3. Helmet use detection algorithm

3.1. Method

After the creation of the dataset was finished, we applied a state-of-the-art object detection algorithm to the annotated data, to facilitate motorcycle helmet use detection on a frame-level. In this process, data from the *training set* is used to train the object detection algorithm. In the process of training, the *validation set* is used to find the best generalizing model, before the algorithm's accuracy in predicting helmet use is tested on data that the algorithm has not seen before, the so-called *test set*. The composition of the three sets is presented in Table 2. Generally, the state-of-the-art object detection algorithms can be divided into two types: two-stage and single-stage approaches. The two-stage approaches first identify a number of potential locations within an image, where objects could be located. In a second step, an object classifier (using a convolutional neural network) is used to identify objects a these locations. While two-stage approaches such as Fast R-CNN [28], achieve a higher accuracy than single-stage approaches, they are very time-consuming. In contrast, single-stage approaches simultaneously conduct object location and object identification. Single stage approaches like YOLO [24] and RetinaNet [18] therefore are much faster than two-stage approaches, although there is a small trade-off in accuracy. In this paper, we used RetinaNet [18] for our helmet use detection task. While it is a single-stage approach, it uses a multi-scale feature pyramid and focal loss to address the general limitation of one-stage detectors in accuracy. Figure 4 illustrates the framework of RetinaNet.

3.2. Training

Since the task of detecting motorcycle riders' helmet use is a classic object detection task, we fine-tuned RetinaNet instead of training it from scratch. I.e. we use a RetinaNet model[1] which is already trained for general object detection and fine tune it to specifically detect motorcycles, riders, and helmets.

[1] https://github.com/fizyr/keras-retinanet

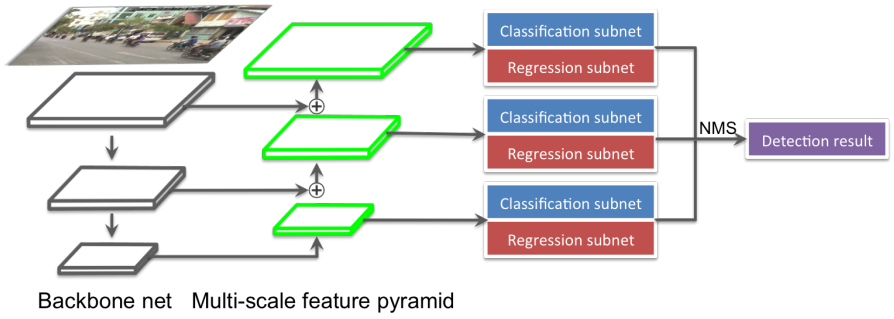


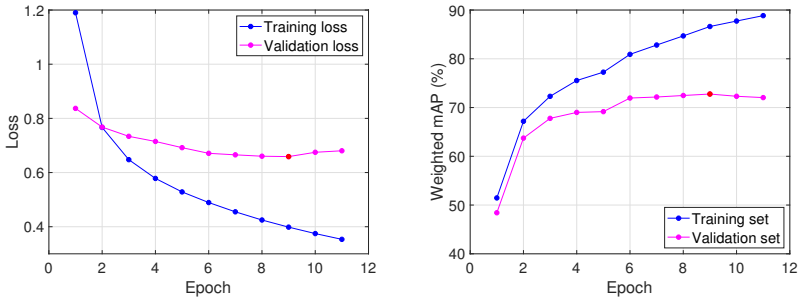
Figure 4: The framework of RetinaNet [18]. A given image is first fed into a backbone convolutional neural network to generate a multi-scale feature map, from which a multi-scale feature pyramid is generated. In each level of the pyramid, there are two subnetworks. One is for regression from anchor boxes to ground-truth object boxes, the other is for classifying anchor boxes. After non-maximum suppression (NMS) across the multi-scale feature pyramid, RetinaNet arrives at the detection results.

In our experiments, we used ResNet50 [15] as the backbone net, initialized with pre-trained weights from ImageNet [29]. The backbone net provides the specific architecture for the convolutional neural network. In the learning process, we used the Adam optimizer [30] with a learning rate of $\alpha = 0.00001$ and a batch size of 4 and stopped training when the weighted mean Average Precision (weighted mAP, explained in the following) on the validation set stopped improving with a patience of 2. To assess the accuracy of our algorithm, we use the Average Precision (AP) value [31]. The AP integrates multiple variables to produce a measure for the accuracy of an algorithm in an object detection task, including *intersection over union*, *precision*, and *recall*. The *intersection over union* describes the positional relation between algorithm generated and human annotated bounding boxes. Algorithm generated bounding boxes need to overlap with human annotated bounding boxes by at least 50%, otherwise they are registered as an incorrect detection. The *precision* presents the number of correct detections of all detections made by the algorithm ($precision = \frac{\text{true positives}}{\text{true positives + false positives}}$). The *recall* variable measures how many of the available correct instances were detected by the algorithm

($recall = \frac{\text{true positives}}{\text{true positives + false negatives}}$). For a more in-depth explanation of AP please see [31] and [32]. Since the number of frames per class was very imbalanced in our dataset (Table 3), the final performance for all classes is computed as weighted average of AP for each class, defined as:

$$\text{weighted mAP} = \sum_{i=1}^{C} w_i \text{AP}_i, \tag{1}$$

where weights w_i across all C classes will sum to one, and set to be proportional to the number of instances. Fig. 5 shows the training loss, validation loss, and weighted mAP in the training and validation sets in the learning process. It can be observed that training loss is constantly decreasing, i.e. the prediction error is getting smaller, while the deep model learns useful knowledge for the helmet use detection from the training set. Consequently, the weighted mAP of the training set is constantly increasing. At the same time, the validation loss, i.e. the prediction error on the validation set is getting smaller in the first 9 epochs. Correspondingly, the mAP on the *validation set* is increasing in the first few epochs before it stops to improve after 9 epochs, which means the algorithm starts to overfit on the training set. Therefore, we stopped training and selected the optimal model after 9 epochs, obtaining 72.8% weighted mAP on the validation set.



(a) (b)

Figure 5: The learning process of RetinaNet for helmet use detection. (a) training and validation loss, (b) weighted mAP on training and validation set. The algorithm achieved 72.8% weighted mAP (red point) on the validation set after 9 epochs.

Our models were implemented using the Python Keras library with Tensorflow as a backend [33] and ran on two NVIDIA Titan Xp GPUs.

3.3. Results

In the following, we report the helmet use detection results of the algorithm on the *test set*, using the optimal model developed on the *validation set* (where it obtained 72.8% weighted mAP).

We achieved **72.3%** weighted mAP on the test set, with a processing speed of **14** frames/second. The AP for each class is shown in Table 3. It can be observed that RetinaNet worked well on common classes but not on under-represented classes due to the small number of training instances. Considering only common classes (up to two riders), our trained RetinaNet achieved **76.4%** weighted mAP. This is a very good performance considering a lot of factors such as occlusion, camera angle, and diverse observation sites. Detection results on some sample frames are displayed in Fig. 6. Due to the imbalanced classes, there are some missing detections, e.g., Fig. 6 (a), (g) and (h). Example videos, consisting of algorithm annotated frames of the *test set* can be found in the *supplementary material*.

Table 3: Composition of annotated data. 339,784 motorcycles were annotated on a frame level. The last column shows the generalized helmet use detection accuracy (mAP= mean Average Precision).

	Position					Motorcycle	Frame level				Helmet use detection
Class	D	P1	P2	P3	P0	tracks	Training	Validation	Test	**Overall**	AP (%)
1	✓	–	–	–	–	4,406	99,029	14,556	27,301	140,886	84.5
2	✓	✓	–	–	–	2,268	50.206	7,071	13,748	71,025	78.5
3	✗	–	–	–	–	1,241	37,664	5,936	10,796	54,396	75.4
4	✗	✗	–	–	–	929	22,723	3,499	5,736	31,958	63.5
5	✓	✗	–	–	–	432	10,729	1,556	2,314	14,599	20.4
6	✗	✗	✗	–	–	211	5,290	377	1,050	6,717	28.0
7	✗	✓	–	–	–	129	2,853	335	511	3699	11.6
8	✓	✗	✓	–	–	125	2,456	639	420	3,515	8.6
9	✓	✗	✗	–	–	75	1,909	269	442	2,620	8.9
10	✗	✗	–	–	✗	55	1,215	113	514	1,842	3.5
11	✓	✓	–	–	✗	49	677	115	466	1,258	9.9
12	✗	✗	✗	–	✗	35	471	208	277	956	18.7
13	✓	–	–	–	✗	34	588	78	369	1035	1.6
14	✗	–	–	–	✗	28	701	95	183	979	0.3
15	✓	✗	–	–	✗	24	600	76	0	676	–
16	✓	✓	✓	–	–	23	492	13	75	580	5.1
17	✓	✓	–	–	✓	22	446	18	146	610	4.2
18	✓	✗	✓	–	✗	22	410	81	24	515	1.6
19	✓	–	–	–	✓	12	352	0	0	352	–
20	✓	✗	✗	–	✗	11	225	0	27	252	0.3
21	✗	✗	✓	–	–	9	123	93	0	216	–
22	✗	✗	✗	✗	–	6	334	28	0	362	–
23	✓	✓	✗	–	–	6	146	0	0	146	–
24	✓	✗	✗	✓	–	5	42	15	0	57	–
25	✓	✗	✗	✗	–	4	50	0	70	120	0.4
26	✓	✗	✓	–	✓	3	62	0	0	62	–
27	✓	✓	✓	–	✗	3	38	11	0	49	–
28	✗	✓	✓	–	–	3	88	0	0	88	–
29	✗	✓	–	–	✗	2	27	0	0	27	–
30	✓	✗	✗	–	✓	2	25	0	0	25	–
31	✗	✗	–	–	✓	1	30	0	0	30	–
32	✗	✗	✗	✗	✗	1	12	0	0	12	–
33	✓	✓	✓	–	✓	1	0	0	21	21	0
34	✗	✗	✓	–	✗	1	0	0	15	15	0
35	✓	✗	✗	✓	✗	1	0	0	53	53	0
36	✓	✗	✗	✗	✗	1	0	0	31	31	0
						10,180	240,013	35,182	64,589	339,784	**weighted mAP: 72.3**

✓ rider in corresponding position wears a helmet

✗ rider in corresponding position does not wear a helmet

– there is no rider in corresponding position

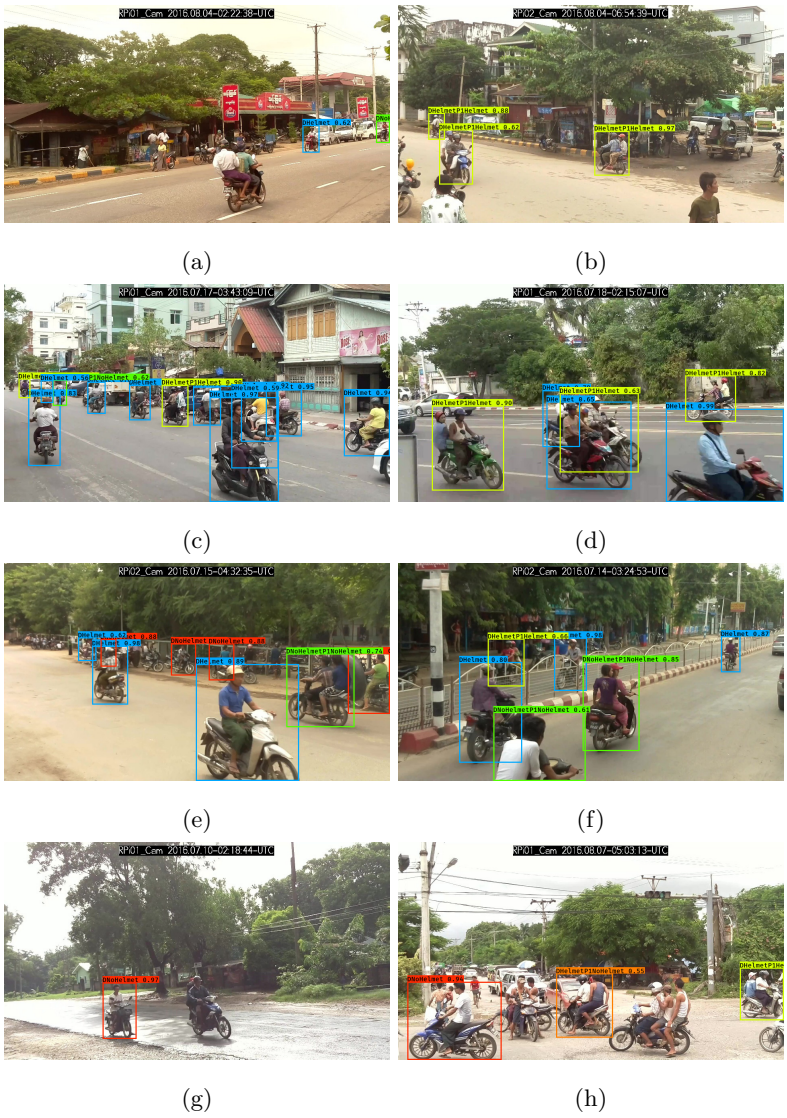


Figure 6: Helmet use detection results on sampled frames using RetinaNet. Bounding box colors correspond to different predicted classes.

4. Comparison to human observation in real world application

Since the video data that forms the basis for the training of the machine learning algorithm in this paper has been analyzed in the past to assess motorcycle helmet use, there is a unique opportunity to compare hand-counted helmet

use numbers in the video data with the calculated helmet use numbers generated by the algorithm developed in this paper. Siebert et al. [4] hand-counted the motorcyclists with and without helmets in the source video data for the first 15 minutes of every hour that a video was recorded. Hence, "hourly" helmet use percentages for every individual observation site in the data set are available. To assess the feasibility of our machine learning approach for real-world observation studies, we compare hourly hand-counted helmet use rates from the Siebert et al. study with hourly computer-counted rates estimated through the application of our algorithm.

4.1. Method

It is important to understand the fundamental difference of the hand-counting method used by Siebert et al. [4] and the frame-based algorithmic approach presented in this paper. In the initial observation of the video data by Siebert et al., a human observer screened 15 minute video sections and registered the number of helmet- and non-helmet-wearing motorcycle riders for individual motorcycles. I.e. helmet use on a motorcycle was only registered once, even though an individual motorcycle was present in multiple frames, driving through the field of view of the camera. The occlusion of an individual motorcycle in some of these frames, e.g. when a motorcycles passed a bus that was located between the motorcycle and the camera, does not pose a problem for the detection of helmet use on that individual motorcycle, as the human observer has the possibility to jump back and forth in the video and register helmet use in a frame with a clear unoccluded view of the motorcycle. Furthermore, a human observer naturally tracks a motorcycle and can easily identify a frame where the riders of the motorcycle and their helmet use is most clearly visible, e.g. when the motorcycle is closest to the camera. The human observer can then use this frame to arrive at a conclusion on the number of riders and their helmet use.

In contrast, the computer vision approach developed in this paper will register motorcycle riders' helmet use in each frame where a motorcycle is detected. This can introduce some error-variance in helmet use detection. The speed of a

motorcycle will influence how many times helmet use for an individual motorcycle is registered, as slower motorcycle riders will appear in more frames than faster ones. Furthermore, occlusion influences how many times a motorcycle will be registered, which can influence the overall helmet use average calculated. Also, helmet use will be registered for motorcycles that are in a sub-optimal angle to the camera, e.g. on motorcycles that drive directly towards the camera, drivers can occlude passengers behind them. However, not all of these differences have a direct impact on helmet use calculated through the algorithm. We assume that occlusion does not introduce a directed bias to detected helmet use rates, as riders with and without helmets have the same chance to be occluded by other traffic. The same can be assumed for differences in motorcycle speed within the observed cities, as riders with helmets won't be faster or slower than those without helmets. We therefore assume that a frame based helmet use registration will lead to comparable results to helmet use registered by a human observer.

Since the algorithm has been trained on specific observation sites, it can be considered to be *observation site trained*. I.e. when the algorithm is applied to the observation site `Bago_rural`, there is data on this specific observation site in the training set (Table 2). In an application of the deep learning based approach, this might not be the case, as the algorithm will not have been trained on new observation sites. Hence, in the following, we also compare algorithmic accuracy for an *observation site untrained* algorithm. For this, we exclude all training data from the observation site that we analyze, before training the algorithm, simulating the application of the algorithm to a new observation site. In the following, *trained algorithm* refers to the algorithm with training on an observation site to be analyzed, while *untrained algorithm* refers to an algorithm that was not trained on an observation site to be analyzed.

4.2. Results

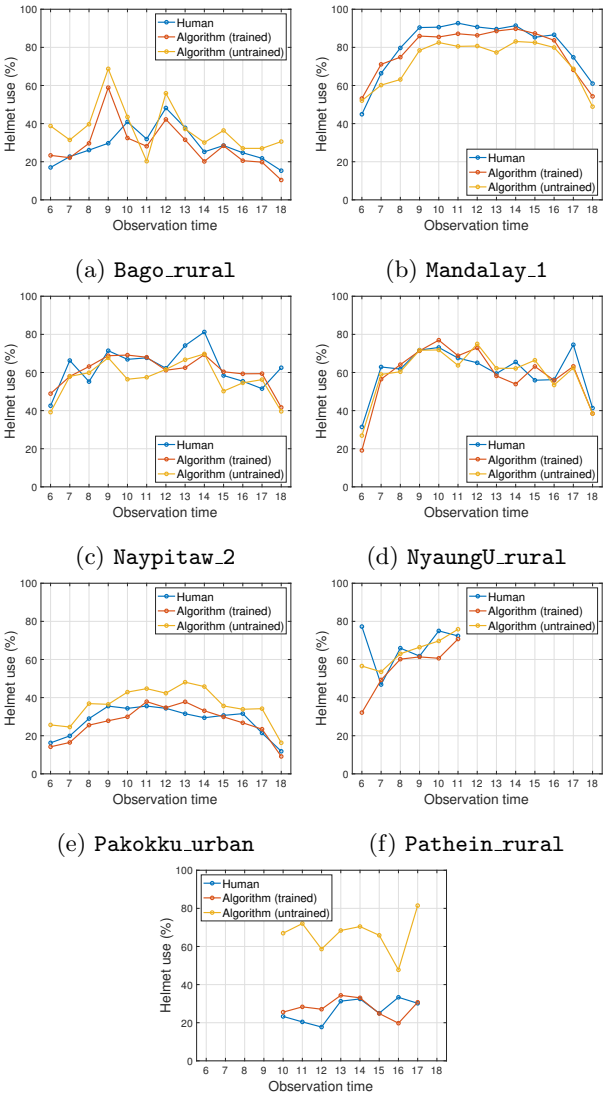
Data on hourly helmet use rates for one randomly chosen day of video data from each observation site is presented in Fig. 7. Helmet use was either reg-

istered by a human observer [4], registered through the *trained* algorithm, or the *untrained* algorithm. It can be observed that hourly helmet use percentages are relatively similar when comparing human and computer registered rates of the trained algorithm. The trained algorithm registers accurate helmet use rates, even when large hourly fluctuations in helmet use are present, e.g. at the `Mandalay_1` observation site (Figure 7(b)). However, some of the observed 15 minute videos show a large discrepancy between helmet use rates registered by a human and the trained algorithm. While it is not possible to conduct a detailed error analysis (as we did in Section 3.3) it is possible to evaluate the video data for broad factors that could increase the discrepancy between human registered data and the data registered by the trained algorithm.

As an example, helmet use rates at the `Bago_rural` observation site at 9 am have a much higher helmet use rate registered by the trained algorithm than by the human observer (Figure 7(a)). A look at the video data from this time frame reveals heavy rain at the observation site (Fig. 8). Apart from an increased blurriness of frames due to a decrease in lighting and visible fogging on the inside of the camera case, motorcycle riders can be observed to use umbrellas to protect themselves against the rain. It can be assumed that motorcycle riders without helmets are more likely to use an umbrella, as they are not protected from the rain by a helmet. This could explain the higher helmet use registered by the trained algorithm at this observation site and time, as non-helmeted riders are less likely to be detected due to umbrellas.

Another instance of a large discrepancy between human and computer registered helmet rates through the trained algorithm can be observed for 6 am at the `Pathein_rural` observation site (Figure 7(f)). A look at the video data reveals bad lighting conditions due to a combination of clouded weather and the early observation time. This results in unclear motorcycles, which are blurred due to their driving speed in combination with the bad lighting conditions (Fig. 9).

Despite singular discrepancies between hourly helmet use rates coded by a human and the trained algorithm, the overall accuracy of average helmet use rates calculated by the trained algorithm per observation site is high. A compar-



(a) `Bago_rural` (b) `Mandalay_1`

(c) `Naypitaw_2` (d) `NyaungU_rural`

(e) `Pakokku_urban` (f) `Pathein_rural`

(g) `Yangon_II`

Figure 7: Hourly helmet use averages for one day of each observation site, registered by human observers and the *trained* and *untrained* algorithm (incomplete data for `Pathein_rural` and `Yangon_II` due to technical problems during the video data collection).

ison of average helmet use per observation site, registered by a human observer and the trained algorithm is presented in (Figure 10). For three observation sites (Naypitaw, Nyaung-U, and Pakokku), trained algorithm registered helmet

Figure 8: Video frames from the Bago observation site at 9 am. Heavy rain, fogging on the camera lens, and umbrella use is visible.

Figure 9: A video frames from the Pathein observation site at 6 am. Low lighting due to heavy clouds results in blurry motorcycles.

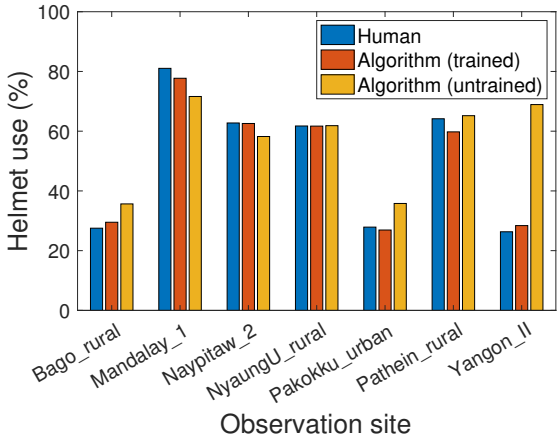


Figure 10: Average helmet use percentage registered by a human observer and the trained and untrained algorithm.

use rates deviate by a maximum of 1% from human registered rates. For the other four observation sites (Bago, Mandalay, Pathein, and Yangon), trained algorithm registered rates are still reasonably accurate, varying between -4.4%

and +2.07%.

For the untrained algorithm, it can be observed that the registered hourly helmet use data is less accurate than that of the trained algorithm, while it is still relatively close to the human registered data for most observation sites (Figure 7). The effects of decreased visibility at the `Bago_rural` and `Pathein_rural` sites are also present. However, at the `Yangon_II` observation site, the registered helmet use of the untrained algorithm is notably higher than helmet use registered by the trained algorithm and the human observer, registering more than double the helmet use present at the observation site. A comparison of the frame-level helmet use detection at the `Yangon_II` site between the trained and untrained algorithm revealed a large number of missed detections of the untrained algorithm (Figure 11). Excluding `Yangon_II`, the helmet use rates registered through the untrained algorithm vary between -8.13% and +9.43% from human registered helmet use (Figure 10).

Figure 11: Comparison of the trained (left) and untrained algorithm (right) at the `Yangon_II` observation site.

5. Discussion

In this paper, we set out to develop a deep learning based approach to detect motorcycle helmet use. Using a large number of video frames we trained an algorithm to detect active motorcycles, the number and position of riders, as well as their helmet use. The use of an annotated test data set allowed us to evaluate the accuracy of our algorithm in detail (Section 3.3, Table 3). The

algorithm had high accuracy for the general detection of motorcycles. Further, it was capable of accurately identifying the number of riders and their position on the motorcycle. The algorithm was less accurate however, for motorcycles with a large number of riders or for motorcycles with an uncommon rider composition (Table 3). Based on these results, the present version of the algorithm can be expected to generate highly accurate results in countries, where only two riders are allowed on a motorcycle and where riders adherence to this law is high. Our implementation of the algorithm can run on consumer hardware with a speed of 14 frames per second, which is higher than the frame rate of the recorded video data. Hence, the algorithm can be implemented to produce real-time helmet use data at any given observation site.

Our comparison of algorithm accuracy with helmet use registered by a human observer (Section 4) revealed an overall high average accuracy, if the algorithm had been trained on the specific observation sites (Figure 10). If there was no prior training on the specific observation site, the (untrained) algorithm had an overall lower accuracy in helmet use detection. There was a large deviation of registered helmet use at the `Yangon_II` observation site, where a large number of missed detections resulted in a highly inaccurate detection performance. The lack of training data with a camera angle similar to the `Yangon_II` observation site is the most likely cause for this low detection accuracy. Potential ways to counteract this performance decrement are discussed in the Section 6.

A comparison of hourly helmet use rates revealed a small number of discrepancies between human and algorithm registered rates (Fig. 7). Further analysis revealed a temporary decrease in the video source material quality as the reason for these discrepancies (Fig. 8 & 9). This decrease in detection accuracy has to be seen in light of the training of the algorithm, in which periods with motion blur due to bad lighting or bad weather were excluded. Hence, decrements in detection accuracy are not necessarily the result of differences in observation sites themselves.

6. Conclusion and future work

The lack of representative motorcycle helmet use data is a serious global concern for governments and road safety actors. Automated helmet use detection for motorcycle riders is a promising approach to efficiently collect large, up-to-date data on this crucial measure. When trained, the algorithm presented in this paper can be directly implemented in existing road traffic surveillance infrastructure to produce real-time helmet use data. Our evaluation of the algorithm confirms a high accuracy of helmet use data, that only deviates by a small margin from comparable data collected by human observers. Observation site specific training of the algorithm does not involve extensive data annotation, as already the annotation of 270 s of video data is enough to produce accurate results for e.g. the `Yangon_II` observation site. While the sole collection of data does not increase road safety by itself [34], it is a prerequisite for targeted enforcement and education campaigns, which can lower the rate of injuries and fatalities [35].

For future work we propose three ways in which the software-side performance of machine learning based motorcycle helmet use detection can be improved. First, there is a need to collect more data in under-represented classes (Table 3) to increase rider, position, and helmet detection accuracy for motorcycles with more than two riders. Second, diverse video data should be collected in regards to the camera angle. This would prevent detection inaccuracies caused by missed detections in camera setups with unusual camera angles. Third, it appears promising to add a simple tracking method for motorcycles to the existing approach. Tracking would allow the identification of individual motorcycles within a number of subsequent frames. Using a frame based quality assessment of an individual motorcycle's frames together with tracking, would allow the algorithm to choose the most suitable frame for helmet use and rider position detection, which will improve overall detection accuracy. Tracking would further allow the algorithm to register the number of individual motorcycles passing an observation site, providing valuable information on traffic flows and density.

On the hardware-side, future applications of the algorithm can greatly benefit from an improved camera system, that is less influenced by low light conditions (Fig. 9) and less susceptible to fogging and blur due to rain on the camera lens (Fig. 8). An increase of the resolution of the video data could allow the detection of additional measures, such as helmet type or chin-strap usage. Apart from a generally increased performance through software and hardware changes, future applications of the developed method could incorporate a more comprehensive set of variables. Within the deep learning approach, the detection of e.g. age categories, chin-strap use, helmet type, or mobile phone use would be possible.

There are a number of limitations to this study. Algorithmic accuracy was only analyzed for road environments within Myanmar, limiting the type of motorcycles and helmets present in the training set. Future studies will need to assess whether the algorithm can maintain the overall high accuracy in road environments in other countries. A similar limitation can be seen in the position of the observation camera. While the algorithm is able to detect motorcycles from a broad range of angles due to diverse training data, there was no observation site where the observation camera was installed in an overhead position, filming traffic from above. Since traffic surveillance infrastructure is often installed at this position, future studies will need to assess whether the algorithm would produce accurate results from an overhead angle. This is especially important in light of the results of the `Yangon_II` observation site, where an unusual camera angle lead to a large number of missed detections. Furthermore, a more structured variation of camera to lane angle would help to better understand optimal positioning of observation equipment for maximum detection accuracy. While it was included in the data annotation process, the algorithmic accuracy in detecting the position of riders was not compared to human registered data in this study. In light of large differences of motorcycle helmet use for different rider positions [4], future studies will need to incorporate deeper analysis of position detection accuracy. For the comparison of human- and machine-registered helmet use rates, it appears promising to enable a detailed error analysis (false

positive/ false negative) through the use of an adapted data structure of human helmet use registration.

In conclusion, we are confident that automated helmet use detection can solve the challenges of costly and time-consuming data collection by human observers. We believe that the algorithm can facilitate broad helmet use data collection and encourage its active use by actors in the road safety field.

Acknowledgement

This research was supported by the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) project-id 251654672 TRR 161 (Project A05).

References

[1] B. Liu, R. Ivers, R. Norton, S. Blows, S. K. Lo, Helmets for preventing injury in motorcycle riders, Cochrane database of systematic reviews 4 (2004) 1–42.

[2] A. M. Bachani, C. Branching, C. Ear, D. R. Roehler, E. M. Parker, S. Tum, M. F. Ballesteros, A. A. Hyder, Trends in prevalence, knowledge, attitudes, and practices of helmet use in cambodia: results from a two year study, Injury 44 (2013) S31–S37.

[3] A. Bachani, Y. Hung, S. Mogere, D. Akunga, J. Nyamari, A. A. Hyder, Helmet wearing in kenya: prevalence, knowledge, attitude, practice and implications, Public health 144 (2017) S23–S31.

[4] F. W. Siebert, D. Albers, U. Aung Naing, P. Perego, S. Chamaiparn, Patterns of motorcycle helmet use a naturalistic observation study in myanmar, Accident Analysis & Prevention 124 (2019) 146–150.

[5] World Health Organization, Global status report on road safety 2015, World Health Organization, 2015.

[6] M. C. Fong, J. R. Measelle, J. L. Dwyer, Y. K. Taylor, A. Mobasser, T. M. Strong, S. Werner, S. Ouansavanh, A. Mounmingkham, M. Kasuavang, et al., Rates of motorcycle helmet use and reasons for non-use among adults and children in luang prabang, lao peoples democratic republic, BMC public health 15 (1) (2015) 970.

[7] R. D. Ledesma, S. S. López, J. Tosi, F. M. Poó, Motorcycle helmet use in mar del plata, argentina: prevalence and associated factors, International journal of injury control and safety promotion 22 (2) (2015) 172–176.

[8] K. Karuppanagounder, A. V. Vijayan, Motorcycle helmet use in calicut, india: User behaviors, attitudes, and perceptions, Traffic injury prevention 17 (3) (2016) 292–296.

[9] Y. Xuequn, L. Ke, R. Ivers, W. Du, T. Senserrick, Prevalence rates of helmet use among motorcycle riders in a developed region in China, Accident Analysis & Prevention 43 (1) (2011) 214–219.

[10] J. Oxley, S. O'Hern, A. Jamaludin, An observational study of restraint and helmet wearing behaviour in malaysia, Transportation research part F: traffic psychology and behaviour 56 (2018) 176–184.

[11] D. W. Eby, Naturalistic observational field techniques for traffic psychology research, in: Handbook of traffic psychology, Elsevier, 2011, pp. 61–72.

[12] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE conference on Computer vision and Pattern Recognition (CVPR), Vol. 1, IEEE, 2005, pp. 886–893.

[13] K. Dahiya, D. Singh, C. K. Mohan, Automatic detection of bike-riders without helmet using surveillance videos in real-time, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 3046–3051.

[14] C. Vishnu, D. Singh, C. K. Mohan, S. Babu, Detection of motorcyclists without helmet in videos using convolutional neural network, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 3036–3041.

[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778.

[16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 2818–2826.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[19] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 2980–2988.

[20] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, J. Dambre, Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video, International Journal of Computer Vision 126 (2-4) (2018) 430–439.

[21] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 2625–2634.

[22] World Health Organization, Powered two-and three-wheeler safety: a road safety manual for decision-makers and practitioners, World Health Organization, 2017.

[23] F. Wegman, B. Watson, S. V. Wong, S. Job, M. Segui-Gomez, Road Safety in Myanmar. Recommendations of an Expert Mission invited by the Government of Myanmar and supported by the Suu Foundation., Paris, FIA, 2017.

[24] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 6517–6525.

[25] A. Shen, Beaverdam: Video annotation tool for computer vision training labels, Master's thesis, EECS Department, University of California, Berkeley (Dec 2016).
URL http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-193.html

[26] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: a database and web-based tool for image annotation, International Journal of Computer Vision 77 (1-3) (2008) 157–173.

[27] C. Vondrick, D. Patterson, D. Ramanan, Efficiently scaling up crowdsourced video annotation, International Journal of Computer Vision 101 (1) (2013) 184–204.

[28] R. Girshick, Fast R-CNN, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2015, pp. 1440–1448.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 248–255.

[30] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[31] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338.

[32] G. Salton, M. J. McGill, Introduction to modern information retrieval, McGraw Hill Book Co., 1983.

[33] F. Chollet, et al., Keras, https://keras.io (2015).

[34] A. A. Hyder, Measurement is not enough for global road safety: implementation is key, The Lancet Public Health 4 (1) (2019) e12–e13.

[35] World Health Organization, Helmets: a road safety manual for decision-makers and practitioners, Geneva: World Health Organization, 2006.

HELMET DETECTION ON TWO-WHEELER RIDERS USING MACHINE LEARNING

[1]RAMESH BABU D R, [2]AMANDEEP RATHEE, [3]KRISHNANGINI KALITA, [4]MAHIMA SINGH DEO

[1,2,3,4]Computer Science Department, DSCE, Bangalore, Karnataka, India
E-mail: bobrammysore@gmail.com, arathee2@gmail.com, krishnangini.kalita@gmail.com, mahimasinghdeo@gmail.com

Abstract - Road safety is often neglected by riders worldwide leading to accidents and deaths. To address this issue, most countries have laws which mandate the use of helmets for two-wheeler riders. In addition to the law, there is a significant proportion of the police force that discourages this behavior by issuing a traffic violation ticket. As of now, this process is manual and tedious. This project aims to solve this problem by automating the process of detecting the riders who are riding without helmets. Furthermore, the system also extracts the license plate so that it could be used to issue traffic violation tickets. The system implements machine learning and image processing techniques to detect riders, riding two-wheelers, who are not wearing helmets. The system takes a video of traffic on public roads as the input and detects moving objects in the scene. A machine learning classifier is applied to the moving object to identify if the moving object is a two-wheeler. If it is a two-wheeler, then another classifier is used to detect whether the rider is wearing a helmet. The license plate is provided as the output in case the rider is not wearing a helmet.

Keywords - Machine Learning, Supervised Learning, Feature Extraction, Background Subtraction, MATLAB Functions

I. INTRODUCTION

The proposed system aims to provide complete safety for bike riders. Recently helmets have been made compulsory, but still, people drive without helmets. The amount of deaths has been rising each year, especially in developing countries. Therefore, keeping public safety in mind, there needs to be a mechanism for automatic helmet detection which can extract the number plates of those who don't wear helmets on roads. This sort of automation will help the administration to issue helmet violation tickets more efficiently and ultimately aims to inhibit the violation by two-wheeler riders.

II. RELATED WORK

A. Existing state of the art

There is a proposed circular arc detection method based on the modified Hough transform. This transformation has been applied to detect a helmet [1] by an ATM surveillance system. Since the safety helmet location will be in the set of the obtained possible circles/circular arcs (if any exists), we use geometric features to verify whether a safety helmet exists in the set. Here, first, the camera scans the image of the driver and sends it to the controller.

$$L = 2\pi r * (rac)$$

In the equation, rac is the ratio of the arc length to the circumference of the circle and r is the radius.

B. Other relevant works

There have been other works in the field of object detection on roads. Let us look at some of these works, one by one, in each of the following paragraphs:

The main task is to detect whether a motorcycle rider is wearing a helmet or not [2]. Even though motorcycles are convenient means of transportation, they are not safe when compared to four-wheelers such as cars. Here, we have given a traffic video as input. In this video, we apply a background extraction algorithm. The algorithm is used to extract the foreground objects in the video which is then extracted as frames. In the next stage, the SIFT (Scale Invariant Feature Transform) algorithm is used to detect a moving object, that is a motorcycle. Using the Region of Interest (ROI), it chooses the location where the helmet can be found. This area is extracted and the helmet is detected using a machine learning classifier.

The driver of the vehicle is involved in a high-speed accident without wearing a helmet and seat belt [3]. It is highly dangerous and can cause death. Wearing a seat belt and helmet can reduce shock from the impact and may save a life. The aim of this research work is to development of smart helmet detection system and seat belt detection system for dune buggy to avoid or reduce the accident fatigue on drivers during the accident. The driver will be unable to start the vehicle without wearing a seat belt and helmet.

Motorcycles being an obvious choice a convenient transportation mode, it has a significant contribution to road accident casualties and injuries [4]. Despite the Government traffic regulation, people still avoid using a helmet. The proposed system is an effort to create awareness in society by endorsing the use of helmet and lead people to safety. This paper proposes effective enforcement of the use of a helmet by implementing RF communication-based helmet detection system.

Automatic text extraction from number plates is used in most of the countries in Present [5]. In this paper, semiautomatic text extraction from Sri Lankan vehicle number plates is presented. Number plate first corrected as can be seen from a camera situated in front of the camera. Then the area of the numbers is extracted. Texts and numbers are then separated in to separate images. Finally, isolated texts and numbers

are matched with the original template created using template matching. A data set consisted of 93 number plates is used to demonstrate.

III. PROPOSED METHOD

A. Flowchart of the proposed methodology

B. Moving object detection

The first task in helmet identification is to detect a moving vehicle. It is the first step before performing more sophisticated functions such as tracking or categorization of vehicles. Rather than immediately processing the entire video, the example starts by obtaining an initial video frame in which the moving objects are segmented from the background. Processing only the initial few frames helps to take the steps required to process the video. The foreground detector needs a certain number of video frames to initialize the Gaussian mixture model [6]. The foreground segmentation process is not perfect and often includes undesirable noise. Next, we find bounding boxes of each connected component corresponding to a moving vehicle. Generally, more than one blob is detected apart from moving vehicles such as pedestrians, trees, dogs and other small noises. All the blobs that consist of less than n number of pixels are discarded (in our case n is 150 pixels). This way, we only remain with the moving vehicle. But there are a lot of gaps in the blob, that is, it is not one coherent blob. We use the morphological opening to remove the noise and to fill gaps in the detected objects which makes the blob more coherent. Once the blob is found, the raw image is extracted that is hidden behind the blob.

C. Vehicle classification

The next step is to classify the moving vehicle extracted in the last part. To classify vehicle, we have used a number of machine learning algorithms, from classical machine learning algorithms to modern deep neural networks, to see which approach works best in vehicle classification with limited data. A vehicle can be classified into two categories twowheelers or four-wheelers. We are only interested in twowheelers Figure 1 since we want to detect the presence of a helmet. The system proceeds further only if a two-wheeler is detected. Else, it discards this vehicle and looks for other vehicles and the cycle continues.

We collected the training data required for the classification of a vehicle on our own. We captured images of various vehicles in various positions. Almost same number of images, 1000, were gathered for both the classes two-wheelers or four-wheelers. If there are equal number of training images from both classes then it eliminates the problem of class imbalance and leads to better performance of the classifier.

The training images contain a vehicle surrounded by other objects of interest such as trees, footpath, buildings and other noise objects. The images mimic how a vehicle is normally seen on roads. Using synthetic images is convenient, and it enables the creation of a variety of training samples through the use of image augmentation. For testing the classifier, a different set of images is used. Although this dataset is not the most representative of the real world moving objects, it is still enough to train and test the effectiveness of various machine learning algorithms to check the feasibility of the approach. The images were converted to grayscale. Raw pixel values were fed to the classifier.

D. Helmet detection

Using the same approach as applied to identify the type of vehicle, we detect whether the rider is wearing a helmet. The images that have been used to train a helmet detector are the cropped version of the two-wheeler images focusing on the head region of the rider. Using this technique, we were still able to maintain the class balance, that is, there was the same number of images where the rider was wearing a helmet and where the rider was not wearing a helmet. We used numerous machine learning classifiers in order to select the best one for this task.

E. License plate extraction

After the previous steps, in case if the rider of a two-wheeler is not wearing a helmet, our next step is to extract the license plate of the vehicle. We extract the region of interest from our cropped image by giving the appropriate coordinates.

IV. MACHINE LEARNING APPROACHES

A total of five machine learning classifiers were used to test which one performs better in our scenario. The classifiers are:

A. Random Forest

This algorithm [7] is based on decision trees. Here, instead of building one tree, a lot of trees are grown in parallel. All these trees are fed only a subset of data points and a subset of features. The subsetting ensures diversity among the trees. After training, each tree votes for a class and a final class is chosen based on the highest number of votes.

B. Gradient Boosted Trees

This algorithm [8] is also based on decision trees like the random forest. However, instead of constructing a lot of trees in parallel, trees are constructed sequentially one after the other. Each tree improves the loss by rectifying the error made by the previous tree while training.

C. Support Vector Machine

An SVM [9] creates a hyperplane (a plane in n-dimensions) which divides all the classes in the training data from one another in such a way that the

difference between the two classes is maximum. This algorithm takes a lot of computing resources to complete the training as compared to the aforementioned classification techniques.

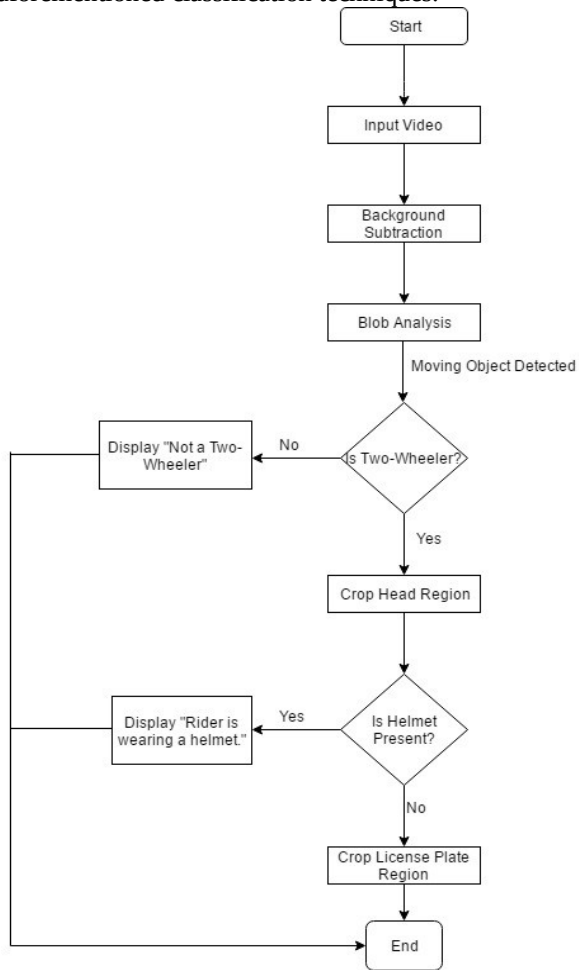


Fig. 1. Flowchart of the proposed method

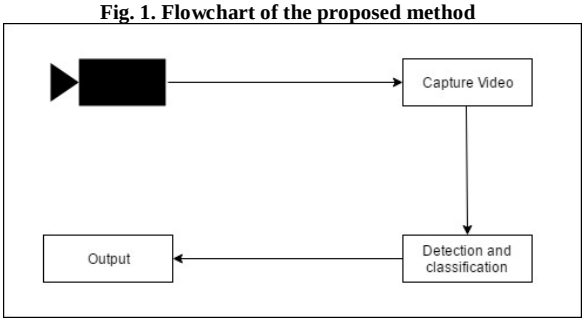


Fig. 2. System architecture

Fig. 3. Moving object detection

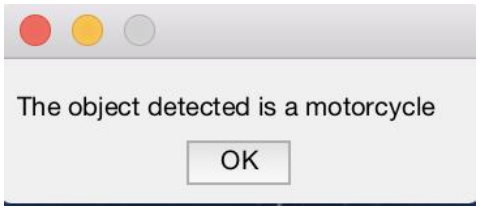


Fig. 4. Vehicle classification result

Fig. 5. Helmet detection result

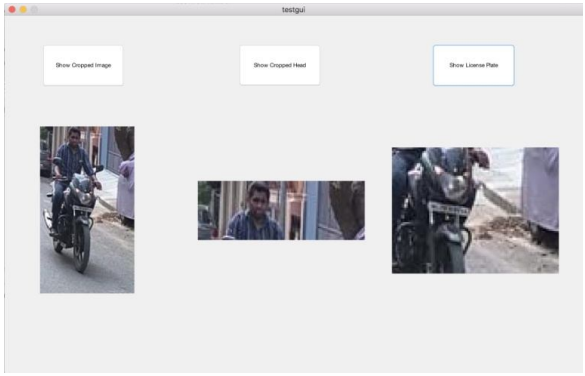


Fig. 6. License plate region cropped

D. Deep Neural Networks

Deep neural networks are an improvement over conventional neural networks. These are neural networks with a large number of layers where each layer has a plethora of nodes. Deep learning is being used to achieve state-of-the-art results in the field of computer vision and natural language processing. Deep neural networks require a lot of training data as compared to conventional machine learning algorithms to outperform them. And that's why we were interested to see how well would this technique work in case of a small training data. If we had a large training data (hundreds of thousands of images for each class), the choice obviously would have been a deep neural network. However, the small number of images becomes a bottleneck for such networks. We used a 10-layer network with 50 nodes in each layer.

RESULTS

All of the five classifiers are trained on about 2000 images. The training images contain both, the front view and the back view of the vehicle. The surveying of various algorithms was done in R [10]. However, the final end-to-end system was implemented in Matlab using the best classifier. All algorithms were tested separately for vehicle classification and helmet detection task. Raw pixel values are used as features in vehicle classification as well as helmet detection. The metric that's chosen is accuracy since all the

classes are balanced, that is, each class has almost the same number of images in both the tasks. 20% of the entire data that was collected wasn't used to train the classifiers. Instead, it was used to test the classifiers. The results are shown in the following tables.

Classifier	Accuracy on test images
Random Forest	91%
Gradient Boosted Trees	71%
Support Vector Machine	73%
Deep Neural Network	76%

TABLE I
VEHICLE CLASSIFICATION RESULTS

Classifier	Accuracy on test images
Random Forest	92%
Gradient Boosted Trees	72%
Support Vector Machine	76%
Deep Neural Network	79%

TABLE II
HELMET DETECTION RESULTS

CONCLUSION AND FUTURE WORK

From the results shown above, one can infer that random forest outperforms all the other algorithms by a significant difference. A deep neural network is expected to perform better than a random forest in image recognition, but due to lack of data, it does not perform as expected. As stated earlier, deep learning algorithms shine when there is a lot of training data.

In the future, the system can be improved by scrutinizing its drawbacks. There are a couple of drawbacks. First, the system doesn't work when there are multiple vehicles in the scene. We have intentionally left that part out because our focus was more on surveying performance of different machine learning algorithms in this scenario rather than making the system best at detecting helmet. However, for the system to be practical, it needs to recognize multiple vehicles and successfully perform all the tasks as it does in the case of a single vehicle. Multiple vehicle detection has already been implemented [11]. The second drawback is, instead of outputting an image of the number plate at the end, the system can output a license number using an OCR (Optical Character Recognition) [12] Extracting the license number will allow the system to automatically send a ticket to the registered owner of the two-wheeler, in case they are not wearing a helmet.

If the above two issues are addressed, and a lot of training data is gathered from surveillance cameras, the system can become much more robust and reliable than it is now.

REFERENCES

[1] "Che-Yen Wen, Shih-Hsuan Chiu, Jiun-Jian Liaw, ChuawPin Lu, 18 May 2004 "The safety helmet detection for ATM'," s surveillance system via the modified Hough transform.

[2] G. S. Gopika, R.Monisha, and S.Karthik, "International Journal of Trend in Research and Development," 3 2016.

[3] S. A. Babu, S. Ayyalusamy, R. R. Singh, S. Dharmarijan, James, Jason, and M. Anas, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), vol. 2015.

[4] G. Sasikala, K. Padol, and A. A. S. Dhanasekaran, Safeguarding of Motorcyclist through helmet recognition, vol. 2015.

[5] J. M. N. D. B. Jayasekara and W. G. C. W, Text Extraction for Sri Lankan Number Plates, vol. 2015.

[6] "MathWorks." [Online]. Available: https://in.mathworks.com/help/vision/ref/vision.foregrounddetector-system-object.html

[7] L. Breiman, Machine Learning (2001) 45, vol. 5. [Online]. Available: https://doi.org/10.1023/A

[8] S. Si, C.-J. Hsieh, I. S. Dhillon, H. Zhang, S. S. Keerthi, and D. Mahajan, "Gradient Boosted Decision Trees for High Dimensional Sparse Output."

[9] C. Cortes and V. L. Vapnik, 1995. [Online]. Available: https:

[10] //doi.org/10.1007/BF00994018

[11] "R." [Online]. Available: https://www.r-project.org/

[12] K. Mu*, F. Hui*, and X. Zhao, "Multiple Vehicle Detection and Tracking, 6 2016, vol. 12, no. 2.

[13] B. V. Kakani and D. S. Jani, "Improved OCR based automatic vehicle number plate recognition using features trained neural network,," in 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017.

★★★

Hindawi
Advances in Civil Engineering
Volume 2020, Article ID 9703560, 10 pages
https://doi.org/10.1155/2020/9703560

Research Article

Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks

Yange Li,[1] Han Wei,[1] Zheng Han ,[1] Jianling Huang,[1] and Weidong Wang[1,2]

[1]*School of Civil Engineering, Central South University, Changsha 410075, China*
[2]*The Key Laboratory of Engineering Structures of Heavy Haul Railway, Ministry of Education, Changsha 410075, China*

Correspondence should be addressed to Zheng Han; zheng_han@csu.edu.cn

Received 8 August 2019; Revised 14 May 2020; Accepted 10 September 2020; Published 19 September 2020

Academic Editor: Antonio Formisano

Visual examination of the workplace and in-time reminder to the failure of wearing a safety helmet is of particular importance to avoid injuries of workers at the construction site. Video monitoring systems provide a large amount of unstructured image data on-site for this purpose, however, requiring a computer vision-based automatic solution for real-time detection. Although a growing body of literature has developed many deep learning-based models to detect helmet for the traffic surveillance aspect, an appropriate solution for the industry application is less discussed in view of the complex scene on the construction site. In this regard, we develop a deep learning-based method for the real-time detection of a safety helmet at the construction site. The presented method uses the SSD-MobileNet algorithm that is based on convolutional neural networks. A dataset containing 3261 images of safety helmets collected from two sources, i.e., manual capture from the video monitoring system at the workplace and open images obtained using web crawler technology, is established and released to the public. The image set is divided into a training set, validation set, and test set, with a sampling ratio of nearly 8 : 1 : 1. The experiment results demonstrate that the presented deep learning-based model using the SSD-MobileNet algorithm is capable of detecting the unsafe operation of failure of wearing a helmet at the construction site, with satisfactory accuracy and efficiency.

1. Introduction

Construction is a high-risk industry where construction workers tend to be hurt in the work process. Head injuries are very serious and often fatal. According to the accident statistics released by the state administration of work safety from 2015 to 2018, among the recorded 78 construction accidents, 53 events happened owing to the fact that the workers did not wear safety helmets properly, accounting for 67.95% of the total number of accidents [1].

In safety management at the construction site, it is essential to supervise the safety protective equipment wearing condition of the construction workers. Safety helmets can bear and disperse the hit of falling objects and alleviate the damage of workers falling from heights. Construction workers tend to ignore safety helmets because of weak safety awareness. At the construction site, workers that wear safety helmets improperly are much more likely to be injured. Traditional supervision of the workers wearing safety helmets on construction sites often requires manual work [2]. There are problems such as a wide range of operations and difficult management of site workers. These factors make manual supervision difficult and inefficient and it is difficult to track and manage the whole workers at the construction sites accurately in real time [3]. Hence, it is hard to satisfy the modern requirement of construction safety management only relying on the traditional manual supervision. In this context, it remains a significant issue to study on the automatic detection and recognition of safety helmets wearing conditions.

The automatic monitoring method can contribute to monitoring the construction workers and confirm the safety helmet wearing conditions at the construction site. In particular, considering that the traditional manual supervision of the workers is often costly, time-consuming, error-prone, and not sufficient to satisfy the modern requirements

of construction safety management, the automatic supervision method can be beneficial to real-time on-site monitoring.

In this paper, based on the previous studies on computer vision-based object detection, we develop a deep learning-based method for the real-time detection of safety helmet at the construction site. The major contributions are as follows: (1) a dataset containing 3261 images of safety helmets collected from two sources, i.e., manual capture from the video monitoring system at the workplace and open images obtained using web crawler technology, is established and released to the public. (2) The SSD-MobileNet algorithm that is based on convolutional neural networks is used to train the model, which is verified in our study as an alternative solution to detect the unsafe operation of failure of wearing a helmet at the construction site. The article is organized as follows. Section 2 gives a brief description of the related work. Section 3 describes the methodology of the research. Section 4 introduces the construction of the database. Section 5 reports the experiment results of the study. Sections 6 and 7 discuss the pros and cons of the study and conclude the paper.

2. Literature Review

2.1. Related Research into the Safety Helmets Detection. At present, previous studies of safety helmets detection can be divided into three parts, sensor-based detection, machine learning-based detection, and deep learning-based detection. Sensor-based detection usually locates the safety helmets and workers (Kelm et al. [4], Torres et al. [5]). The methods usually use the RFID tags and readers to locate the helmets and workers and monitor how personal protective equipment is worn by workers in real time. Kelm et al. [4] designed a mobile Radio Frequency Identification (RFID) portal for checking personal protective equipment (PPE) compliance of personnel. However, the working range of the RFID readers is limited and the RFID readers can only suggest that the safety helmets are close to the workers but unable to confirm that the safety helmets are being properly worn.

Up to date, machine learning-based object detection technologies are widely used in many domains for its powerful object detection and classification capacity (e.g., Rubaiyat et al. [6], Shrestha et al. [7], Waranusast et al. [8], Doungmala et al. [9], Jia et al. [10], and Li et al. [11]). Remarkable studies are made by Rubaiyat et al. [6], who proposed an automatic detection method to obtain the features of construction workers and safety helmets and detect safety helmets. The method combines the frequency domain information of the image with the histogram of oriented gradient (HOG) and the circle Hough transform (CHT) extractive technique to detect the workers and the helmets in two steps. The detection methods based on machine learning can detect safety helmets accurately and precisely under various scenarios but also have some drawbacks. Sometimes the method can only detect safety helmets with a specific color and it is difficult to distinguish the hats with similar color and shape to the safety helmets. Moreover, the method cannot detect faces and safety helmets thoroughly under some circumstances; for example, some workers do not turn their faces towards the camera at the construction site.

2.2. Deep Learning-Based Object Detection. The above-mentioned methods are commonly based on traditional machine learning to detect and classify the helmets and choose features artificially with a strong subjectivity, a complex design process, and poor generalization ability. In recent years, with the rapid development of deep learning technology, the object detection algorithm turns to the one based on convolutional neural networks with a great promotion of speed and accuracy (e.g., Wu et al. [12]).

The methods construct convolutional neural networks with different depths to detect safety helmets. Some other strategies such as multiscale training, increasing the number of anchors and introducing the online hard example mining, are added to increase the detection accuracy (e.g., Xu et al. [13]). However, these methods have some limitations in the preprocessing aspects of image sharpness, object proportion, and the color difference between background and foreground.

Deep learning-based methods are very potential for the purpose of people's unsafe behavior identification. Many previous studies have presented a solution to this topic. Remarkable studies include the following: Ding et al. [14] developed a hybrid deep learning model that integrates a convolution neural network (CNN) and long short-term memory (LSTM) that automatically recognizes workers' unsafe actions. The results demonstrated that the model can precisely detect safe and unsafe actions conducted by workers on-site. However, some behaviors cannot be recognized owing to the lack of data, the small sample size used for training, and the limited number of unsafe actions that were considered. Fang et al. [15] proposed a novel deep learning-based framework to check whether a site worker is working within the constraints of their certification. The framework includes key video clips extraction, trade recognition, and worker competency evaluation. Results demonstrate that the proposed framework offers an effective and feasible solution to detect noncertified work. However, some workers cannot be detected when the workers' faces hardly appear or are obstructed by the safety helmets or other equipment. Also, the worker close to the camera failed to be recognized. Fang et al. [16] integrated a Faster R-CNN and a deep CNN to detect the presence of a worker from images and the harness, respectively, which can identify whether workers wear safety harness while working at heights or not. The research is limited by the restricted activities working at heights and the dataset size. Fang et al. [17] developed a computer vision-based approach which uses a Mask R-CNN to detect people and recognize the relationship between people and concrete supports to identify unsafe behaviors. The study has some restrictions: it focuses on a limited number of activities related to the construction of deep foundation-pits. Luo et al. [18] proposed an increased CNN that integrates Red-Green-Blue,

optical flow, and gray stream CNNs to monitor and assess workers' activities associated with installing reinforcement at the construction site. The research is limited by occlusions, insufficient knowledge of a time series of actions definition, and lack of a large-scale database.

Considering its excellent ability to extract features, in the paper, we use the convolutional neural network (CNN) to build a safety helmet detection model. Automatic detection of safety helmets worn by construction workers at the construction site and timely warning of workers without helmets can largely avoid accidents caused by workers wearing safety helmets improperly. The designed CNN is trained using the TensorFlow framework. The contributions of the research include a deep learning-based safety helmet detection model and a safety helmet image dataset for further research. The model provides an opportunity to detect the helmets and improve safety management.

Deep learning-based methods are commonly used to detect unsafe behaviors on-site. Nevertheless, many traditional measures of safety helmet detection are commonly sensor-based and machine-based, thus limited by problems such as sensor failure over long distances, the manual and subjective features choice, and the chaotic scene interference. Based on the previous studies, we present a deep learning-based method to detect the safety helmets in the workplace, which is supposed to avoid the abovementioned limitations.

3. Methodology

A convolutional neural network (CNN) is a multilayer neural network. It is a deep learning method designed for image recognition and classification tasks. It can solve the problems of too many parameters and difficult training of the deep neural networks and can get better classification effects. The structure of most CNNs consists of input layer-convolutional layer (Conv layer)-activation function-pooling layer-fully connected layer (FC layer). The main characteristics of CNNs are local connectivity and parameter sharing in order to reduce the number of parameters and increase the efficiency of detection.

The Conv layer and the pooling layer are the core parts, and they can extract the object features. Often, the convolutional layer and the pooling layer may occur alternately. The Conv layers can extract and reinforce the object features. The pooling layers can filter multiple features, remove the unimportant features, and compress the features. The activation layers use nonlinear activation functions to enhance the expression ability of the neural network models and can solve the nonlinear problems effectively. The FC layers combine the data features of objects and output the feature values. By this means the CNNs can transfer the original input images from the original pixel values to the final classification confidence layer by layer.

In order to better extract the object features and classify the objects more precisely, Hinton et al. [19] proposed the concept of deep learning which is to learn object features from vast amounts of data using deep neural networks and then classify new objects according to the learned features. Deep learning algorithm based on convolutional neural networks has achieved great results in object detection, image recognition, and image segmentation. Girshick et al. [20] proposed R-CNN detection framework (region with CNN features) in 2014. Many models based on R-CNN were proposed after that including SPP-net (spatial pyramid pooling network) [21], Fast R-CNN (fast region with CNN features) [22], and Faster R-CNN (faster region with CNN features) [23].

Classification-based CNN object detection algorithms such as Faster R-CNN are widely used methods. However, the detection speed is slow and cannot detect in real time. Regression-based detection algorithms are becoming increasingly important. Redmon et al. [24] proposed YOLO (You Only Look Once) algorithm in 2016. At the end of 2016, Liu et al. [25] combined the anchor box of Faster R-CNN with the bounding box regression of YOLO and proposed a new algorithm SSD (Single Shot MultiBox Detector) with higher detection accuracy and faster speed.

Although the SSD algorithm is not capable of the highest accuracy, the detection speed of the SSD algorithm is much faster and comparable to the YOLO algorithm and the precision can be higher than that of the YOLO algorithm when the sizes of the input images are smaller. While the Faster R-CNN algorithm tends to lead to more accurate models, it is much slower and requires at least 100 ms per image [26]. Therefore, considering the real-time detection requirements, the SSD algorithm is chosen in the research. In order to reduce greatly the calculation amount and model thickness, the MobileNet [27] model is added. Therefore, in the paper, the SSD-MobileNet model is selected to detect safety helmets worn by the workers.

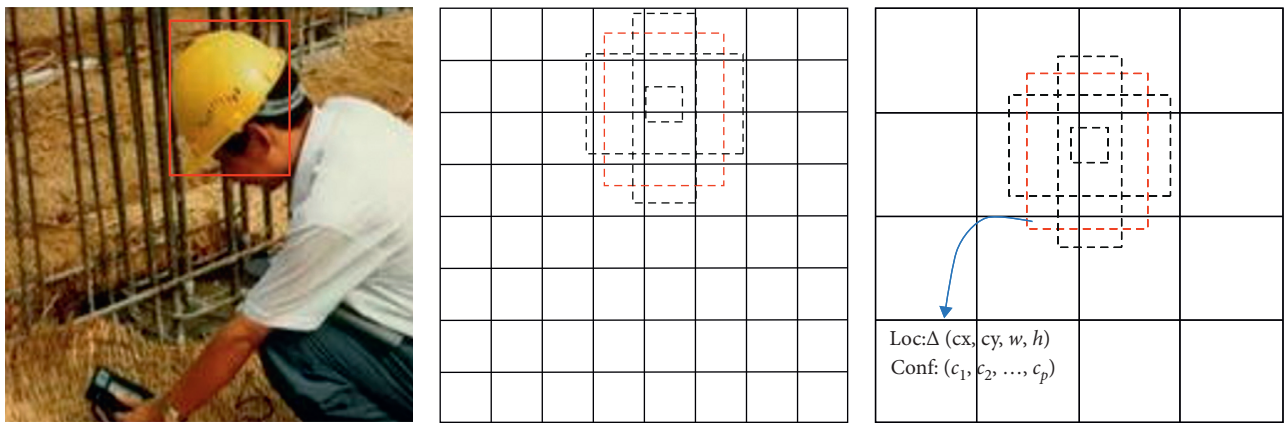
The SSD algorithm is based on a feed-forward convolutional network to produce bounding boxes of fixed sizes and generate scores for the object class examples in the boxes. A nonmaximum suppression method is used to predict the final results.

The early network layers of the SSD model are called the base network, based on a standard framework to classify the image. The base network is truncated before the classification layers, and the convolutional layers are added at the end of the truncated base network. The sizes of the convolutional feature maps decrease progressively to predict the detections at multiple scales.

The SSD algorithm sets a series of fixed and different size default boxes on the cell of each feature map as shown in Figure 1. Each default box predicts two kinds of detections. One is the location of bounding boxes including 4 offsets (cx, cy, w, h), which represent, respectively, x and y coordinates of the center of the bounding box and the width and height of the bounding box; the other is the score of each class. If there are C classes of the objects, the SSD algorithm predicts a total of C+1 score including the score of the background.

The setting of default boxes can be divided into two aspects: size and aspect ratio. The sizes of the default boxes in every feature map will be calculated as follows:

$$S_K = S_{\min} + \frac{S_{\max} - S_{\min}}{m - 1} k - 1, \quad k \in [1, m]. \tag{1}$$



(a) (b) (c)

Figure 1: Detection process. (a) Image with GT box. (b) 8×8 feature map. (c) 4×4 feature map.

In the formula, $S_{\min}$ is 0.2, $S_{\max}$ is 0.95. The aspect ratios are set different for the default boxes, expressed as $a_r \in \{1, 2, 3, (1/2), (1/3)\}$. The width of the default boxes is calculated as follows:

$$w_k^a = S_k \sqrt{a_r}. \tag{2}$$

The height of the default boxes is calculated as follows:

$$h_k^a = \frac{S_k}{\sqrt{a_r}}. \tag{3}$$

When the aspect ratio is 1, a default box size is added: $S_k' = \sqrt{S_k S_{k+1}}$. Therefore, there are six default boxes of different sizes for each feature cell.

The default boxes will be matched to the ground truth boxes. Each ground truth box can choose default boxes of different locations, aspect ratios, and sizes to match. The ground truth box will be matched to the default box with the best Jaccard overlap. The Jaccard overlap is also called the IoU (Intersection over Union), or the Jaccard similarity coefficient. The IoU is the ratio of the intersection and the union of the default box to the ground truth box. The schematic illustration of IoU is shown in Figure 2:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| + |A \cap B|}. \tag{4}$$

After the match process, most default boxes are negative examples which do not match the objects but the background. Therefore, the SSD algorithm uses the hard negative mining strategy to avoid the significant imbalance between the positive and negative training examples. The default boxes are ranked in the descending order according to the confidence error and the top ones are chosen to be the negative examples so the ratio between the negative and positive examples is almost 3 : 1.

The SSD algorithm defines the total loss function as the weighted sum between localization loss and confidence loss:

$$L(x, c, l, g) = \frac{1}{N}\left(L_{\text{conf}}(x, c) + \alpha L_{loc}(x, l, g)\right). \tag{5}$$

In the prediction process, the object classes and confidence scores will be confirmed according to the maximum class confidence score and the prediction box that belongs to the background will be filtered out. The prediction boxes with confidence scores below 0.5 are also removed. As for the left boxes, the location will be obtained according to the default boxes. The prediction boxes are ranked in the descending order according to the confidence score and the top ones are retained. Finally, the nonmaximum suppression algorithm is used to filter out the prediction boxes with higher but not the highest IOU and the left prediction boxes are the results.

Although the SSD algorithm performs well in the speed and the precision, the large model and a large amount of calculation make the training speed a bit slow. Therefore, the base network of the SSD model is replaced by the MobileNet model to reduce the calculation amount and the model thickness. In the paper, the SSD-MobileNet model is chosen to detect the safety helmets worn by the workers.

The core concept of the MobileNet model is the factorization of the filters. The main function is to reduce the calculation amount and the network parameters. The model is used to factorize a standard convolution into a depthwise convolution and a pointwise convolution. The model is shown in Figure 3.

The model also introduces two hyperparameters: width multiplier and resolution multiplier to reduce the channel numbers and reduce the image resolutions, respectively. The network model with less calculation amount can be built. Hence, using the SSD-MobileNet model can reduce the thickness of the SSD model effectively.

4. Database

The data required for the experiment were collected by the author. Since there are few object detection applications of safety helmets using deep learning and there is no off-the-shelf safety helmets dataset available, part of the experimental data was collected using web crawler technology, making full use of network resources. By using several

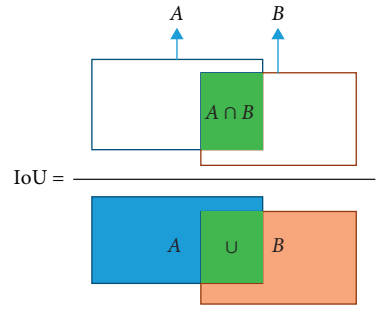


Figure 2: Schematic illustration of IoU.

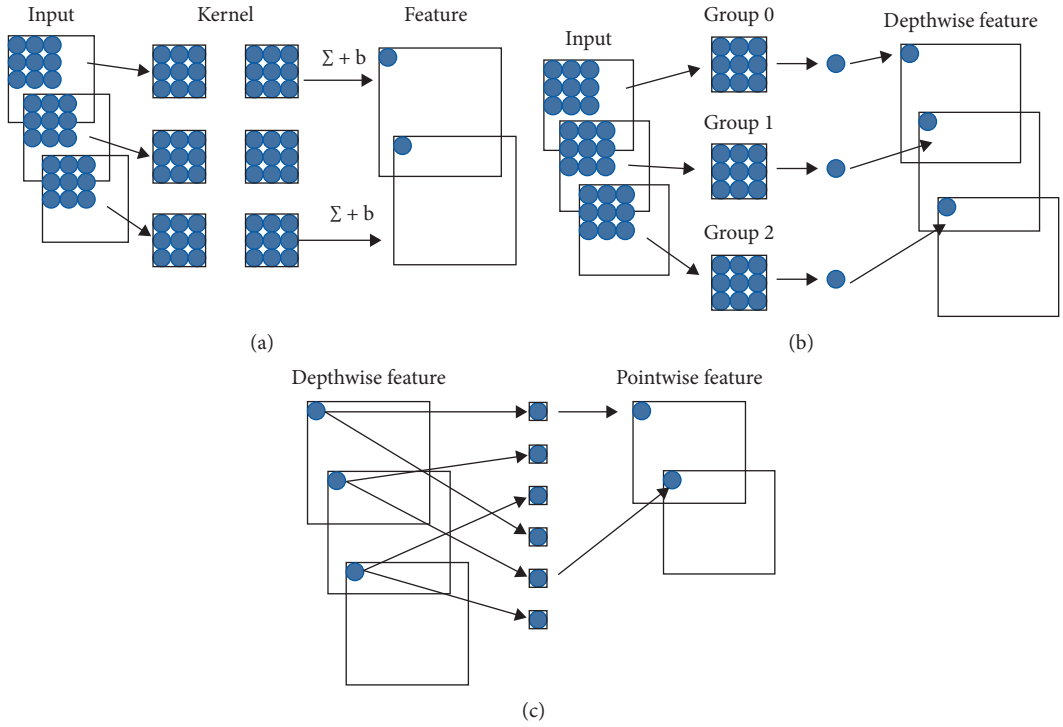


Figure 3: Schematic illustration of MobileNet separate convolution. (a) Standard convolution. (b) Depthwise convolution. (c) Pointwise convolution.

keywords, such as "workers wear safety helmets" and "workers on the construction site," python language is used to crawl relevant pictures on the Internet.

However, the quality of the crawled images varies greatly. There are problems that there is an only background and no objects in some images, the size of the safety helmet is small, and the shape is blurred. Therefore, images were also collected manually besides web crawling. 3500 images were collected in total. The images that did not contain safety helmets, duplicate images, and the images that are not in the RGB three-channel format were eliminated and 3261 images were left, forming the safety helmet detection dataset. Some images in the dataset are shown in Figure 4. To increase the detection effect of the safety helmet detection model in detecting helmets with different directions and brightness in images, the image dataset was preprocessed such as rotation, cutting, and zooming.

Then, the samples in the dataset are divided into three parts randomly: training set, validation set, and test set. Commonly, a ratio of 6 : 2 : 2 is suggested for dividing the training set, validation set, and test set in the previous machine learning studies, such as the course of Andrew Ng from deeplearning.ai. In deep learning, the dataset scale is much larger and the validation and test sets tend to be a smaller percentage of the total data which are commonly less than 20% or 10%. In this sense, an adequate ratio of 8 : 1 : 1 according to the previous experience is adopted in our study. The numbers of the three sets are 2769, 339, and 153, respectively. All the images that contained safety helmets were manually prelabeled, using the open-source tool LabelImage (available in https://github.com/tzutalin/labelImg). In each labeled image, the sizes and the locations of the object are recorded (Figure 5).

5. Results

In the paper, the open-source TensorFlow framework is chosen to train the model. The pretrained

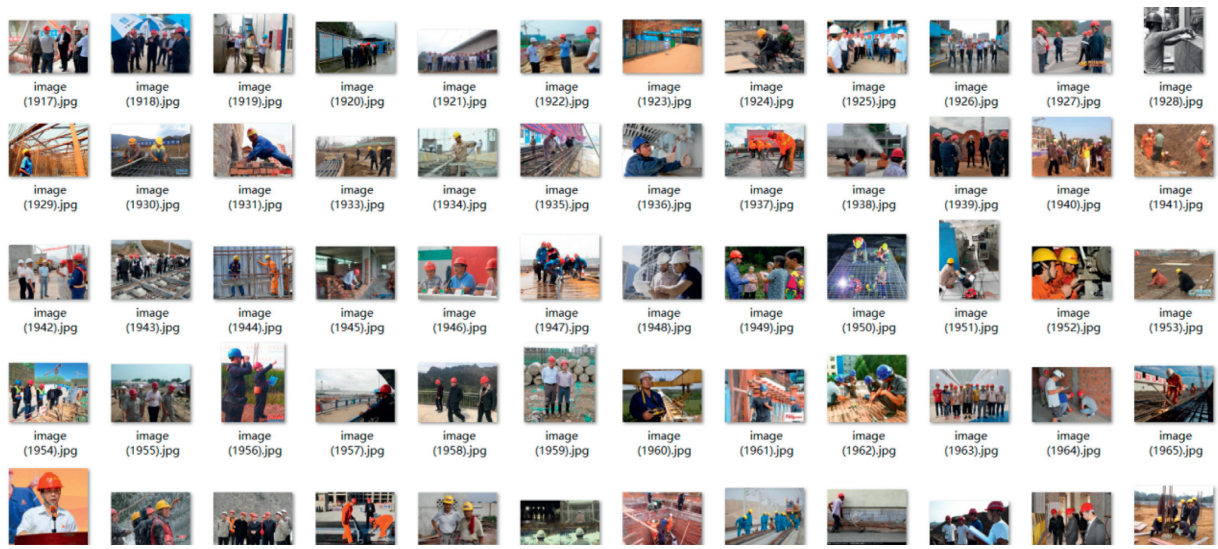


Figure 4: Image dataset.

(a)

Filename	Width	Height	Class	Xmin	Ymin	Xmax	Ymax
Image (1).jpg	315	207	Helmet	100	6	152	51
Image (1).jpg	315	207	Helmet	162	35	200	61
Image (1).jpg	315	207	Helmet	206	53	229	69
Image (1).jpg	315	207	Helmet	238	45	258	60
Image (1).jpg	315	207	Helmet	1	30	33	68
Image (1).jpg	315	207	Helmet	46	45	83	74

(b)

Figure 5: (a) Manual labeling. (b). Data recording.

SSD_mobilenet_v1_COCO model with the COCO dataset is used to learn the characteristics of the safety helmet in the built dataset to reduce the training time and save the computing resources. The initial weights and the parameter values of our own model are the same as the SSD_mobilenet_v1_COCO model. Finally, the weights and the parameter values of the safety helmet detection model are trained and obtained through the training process.

Among the 3261 images, 2769 images were divided into the training set, 339 images were divided into the validation set, and 153 images were divided into the test set. The training set is used to train the model or to determine the parameters of the model. The validation set is used to adjust the hyperparameters of the model and to evaluate the capacity of the model preliminarily. The test set is used to evaluate the generalization ability of the final model [28].

In the course of training, the change of the mean average precision (mAP) and the loss function during training was recorded by TensorBoard. As a measure index, the mean average precision (mAP) [29] is generally used in the field of object detection. Figure 6 illustrated that the mean Average Precision shows an overall upward trend, and the trend has ups and downs and is not a steady rise. When training rounds up to 50,000, the mean average precision of the detection model is 36.82%. Figure 7 shows that the total loss values decrease slowly at the beginning of the training and converge at the end of the training. The values of the loss function are the differences between the true value and the predicted value in general speaking. The change in the values of the loss function represents the training process of the model. The smaller the values are, the better the model is trained. The convergence of the loss functions demonstrates that the training of the model is completed. Hence, loss functions mainly influence the training process but not detection. Figures 8(a) to 8(c) show the variation of the classification loss function, localization loss function, and regularization loss function against the steps. Figures 8(a) and 8(b) demonstrate that the values of the classification loss function decrease slowly at first and, then, decrease rapidly when training rounds up to nearly 7,000; the values of the localization loss function decrease rapidly at first and converge at the end of the training. The convergence of the loss functions demonstrates that the training of the model is completed.

Figure 6: Mean average precision change during training.

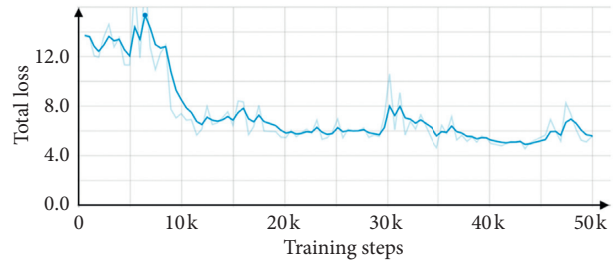


Figure 7: Total loss change during training.

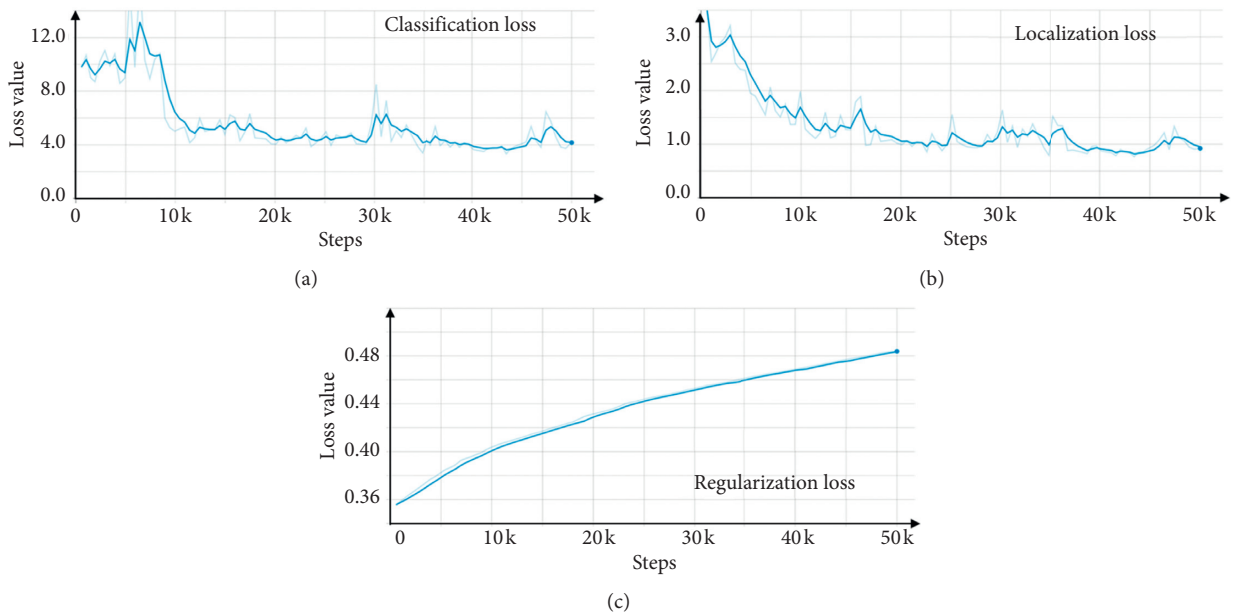


Figure 8: Partial loss function value change. (a) Classification loss. (b) Localization loss. (c) Regularization loss.

After the model was trained, it was used to validate the collected validation set by using the Spyder software. The 153 images of the validation set were input into the model and the detected images were output. The output images show the predicted labels and the confidence scores of safety helmets. Some validation results are shown in Figure 9.

The precision and recall are the commonly used metrics to evaluate the performance and reliability of the trained model. Precision is the ratio of true positive (TP) to true positive and false positive (TP + FP). TP + FP is the number of helmets detected. Recall is the ratio of true positive (TP) to true positive and false negative (TP + FN). TP + FN means the actual number of helmets. There are 250 true positive objects, 12 false positive objects, and 73 false negative objects in the detected images. The precision of the trained model is 95% and the recall is 77%, which demonstrates that the proposed method performs well in safety helmet detection.

As the pictures above show, the probabilities of recognizing the safety helmets worn by workers as safety helmets are more than 80%. However, the output images of the model demonstrate some errors in the detection model. For example, it is hard for the model to detect the safety helmets of small sizes or large rotation angles. It is possible to recognize the objects of the same colors in the images as the

Figure 9: Validation results.

safety helmets. When the illumination intensity of the construction site in the images and the objects are not clear, the safety helmets are difficult to be recognized. That suggests the detection model established in the paper is not accurate enough.

As shown in Figure 10(a), the probability predicted by the model is 98%, but the probability of recognizing the background as safety helmets is 78%. This fake detection generates false positive. This is a case of false detection which predicted the false object as correct. The case of Figure 10(c) is the same as the first one. In Figure 10(b), the red helmet is missed and this is a case of false negative. The errors occur because of the interference of the complex background, the limitation of the number of the image dataset, and the safety helmets proportion in the images. In order to improve the performance of the model, some measures must be taken such as increasing the number of the image dataset and adding the preprocessing operations of the images. Besides the above measures, ameliorating the nonmaximum suppression algorithm, adjusting the parameters and weights, and so forth can also be a great solution to reduce the false positives.

In summary, there are several detection errors of the model. (1) The hats with the same shapes and colors or the background are recognized mistakenly as the safety helmets. (2) The safety helmets of incomplete shapes and small sizes are hard to be recognized. (3) The two or more helmets that are very close to each other are often recognized as a safety helmet.

6. Discussion

6.1. Effect of the Presented Method. The proposed automatic detection method based on deep learning to detect safety helmets worn by workers provides an effective opportunity to improve safety management on construction sites. Previous studies have demonstrated the effectiveness of locating the safety helmets and workers and detecting the helmets. However, most of the studies have limitations in practical application. Sensor-based detection methods have a limited read range of readers and cannot be able to confirm the position relationship between the helmets and the workers. The machine learning-based detection methods choose features artificially with a strong subjectivity, a complex design process, and poor generalization ability. Therefore, the study proposed a method based on deep learning to detect safety helmets automatically using convolutional neural networks. The experimental results have suggested the effectiveness of the proposed method.

In the paper, the SSD-MobileNet algorithm is used to build the model. A dataset of 3261 images containing various helmets is trained and tested on the model. The experimental results demonstrate the feasibility of the model. And the model does not require the selection of handcraft features and has a good capacity of extracting features in the images. The high precision and recall show the great performance of the model. The proposed model provides an opportunity to detect the helmets and improve construction safety management on-site.

6.2. Limitations. However, the detection model has a poor performance when the images are not very clear, the safety helmets are too small and obscure, and the background is too complex as shown in Figure 10. Moreover, the presented model is limited by the problems that some images of the dataset are less in quantity; the preprocessing operations of the images are confined to rotation, cutting, and zooming; the manual labeling is not comprehensive and may miss some objects. In some extreme cases, for example, only part of the head is visible and the safety helmet is obstructed, the model cannot detect the helmets accurately. This is the common limitation of the-state-of-art algorithms. Due to the above reasons, the detection performance is not good enough and there are some detection errors.

(a) (b) (c)

Figure 10: Detection errors.

The algorithm we use emphasizes the real-time detection and fast speed. However, the accuracy of the detection is also quite important and the performance needs to be improved. Hence, in the ongoing studies, we are working at the expansion and improvement of the dataset in order to solve the problems of inadequate data with poor quality. More comprehensive preprocessing operations should be done to improve the performance of the model.

7. Conclusions

The paper proposed a method for detecting the wearing of safety helmets by the workers based on convolutional neural networks. The model uses the SSD-MobileNet algorithm to detect safety helmets. Then, a dataset of 3261 images containing various helmets is built and divided into three parts to train and test the model. The TensorFlow framework is chosen to train the model. After the training and testing process, the mean average precision (mAP) of the detection model is stable and the helmet detection model is built. The experiment results demonstrate that the method can be used to detect the safety helmets worn by the construction workers at the construction site. The presented method offers an alternative solution to detect the safety helmets and improve the safety management of the construction workers at the construction site.

Data Availability

The database used to train the CNN of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Professor Y.G. Li collected the dataset and wrote the manuscript. Professor Z. Han designed the study. H. Wei trained and optimized the model. Professor J.L. Huang and Professor W.D. Wang participated in the analysis of the results. All the authors discussed the results and commented on the manuscript.

Acknowledgments

This study was financially supported by the National Key R&D Program of China (Grant no. 2018YFC1505401); the National Natural Science Foundation of China (Grant no. 52078493); the Natural Science Foundation of Hunan (Grant no. 2018JJ3638); and the Innovation Driven Program of Central South University (Grant no. 2019CX011). These financial supports are gratefully acknowledged.

References

[1] X. Chang and X. M. Liu, "Fault tree analysis of unreasonably wearing helmets for builders," *Journal of Jilin Jianzhu University*, vol. 35, no. 6, pp. 67–71, 2018.

[2] Z. Y. Wang, *Design and Implementation of Detection System of Warning Helmets Based on Intelligent Video Surveillance*, Beijing University of Posts and Telecommunications, Beijing, China, 2018.

[3] H. Zeng, *Research on Intelligent Helmets System for Engineering Construction*, Harbin Institute of Technology, Harbin, China, 2017.

[4] A. Kelm, L. Laußat, A. Meins-Becker et al., "Mobile passive radio frequency identification (RFID) portal for automated and rapid control of personal protective equipment (PPE) on construction sites," *Automation in Construction*, vol. 36, pp. 38–52, 2013.

[5] S. Barro-Torres, T. M. Fernández-Caramés, H. J. Pérez-Iglesias, and C. J. Escudero, "Real-time personal protective equipment monitoring system," *Computer Communications*, vol. 36, no. 1, pp. 42–50, 2012.

[6] A. H. M. Rubaiyat, T. T. Toma, M. Kalantari-Khandani et al., "Automatic detection of helmet uses for construction safety," in *Proceedings of the 2016 IEEE ACM International Conference on Web Intelligence Workshops (WIW)*, ACM, Omaha, NE, USA, October 2016.

[7] K. Shrestha, P. P. Shrestha, D. Bajracharya, and E. A. Yfantis, "Hard-hat detection for construction safety visualization," *Journal of Construction Engineering*, vol. 2015, Article ID 721380, 8 pages, 2015.

[8] R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, "Machine vision techniques for motorcycle safety helmet detection," in *Proceedings of the Image & Vision Computing New Zealand*, IEEE, Wellington, New Zealand, November 2013.

[9] P. Doungmala and K. Klubsuwan, "Helmet wearing detection in Thailand using haar like feature and circle hough transform

on image processing," in *Proceedings of the IEEE International Conference on Computer & Information Technology*, IEEE, Nadi, Fiji, December 2016.

[10] J. S. Jia, Q. J. Bao, and H. M. Tang, "Method for detecting safety helmet based on deformable part model," *Application Research of Computers*, vol. 33, no. 3, pp. 953–956, 2016.

[11] K. Li, X. G. Zhao, J. Bian, and M. Tan, "Automatic safety helmet wearing detection," 2018, https://arxiv.org/abs/1802.00264.

[12] H. Wu and J. Zhao, "Automated visual helmet identification based on deep convolutional neural networks," in *Proceedings of the 13th International Symposium on Process Systems Engineering (PSE 2018)*, vol. 44, pp. 2299–2304, San Diego, CA, USA, July 2018.

[13] S. Xu, Y. Wang, Y. Gu, N. Li, L. Zhuang, and L. Shi, "Safety helmet wearing detection study based on improved faster RCNN," *Application Research of Computers*, vol. 37, no. 3, pp. 901–905, 2019.

[14] L. Ding, W. Fang, H. Luo, P. E. D. Love, B. Zhong, and X. Ouyang, "A deep hybrid learning model to detect unsafe behavior: integrating convolution neural networks and long short-term memory," *Automation in Construction*, vol. 86, pp. 118–124, 2018.

[15] Q. Fang, H. Li, X. Luo et al., "A deep learning-based method for detecting non-certified work on construction sites," *Advanced Engineering Informatics*, vol. 35, pp. 56–68, 2018.

[16] W. Fang, L. Ding, H. Luo, and P. E. D. Love, "Falls from heights: a computer vision-based approach for safety harness detection," *Automation in Construction*, vol. 91, pp. 53–61, 2018.

[17] W. Fang, B. Zhong, N. Zhao et al., "A deep learning based approach for mitigating falls from height with computer vision: convolutional neural network," *Advanced Engineering Informatics*, vol. 39, pp. 179–177, 2019.

[18] H. Luo, C. Xiong, W. Fang, P. E. D. Love, B. Zhang, and X. Ouyang, "Convolutional neural networks: computer vision-based workforce activity assessment in construction," *Automation in Construction*, vol. 94, pp. 282–289, 2018.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2014.

[22] R. Girshick, "Fast R-CNN. computer science," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal network," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, p. 1137, 2016.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[25] W. Liu, D. Anguelov, D. Erhan et al., "Single shot multibox detector," in *Proceedings of the ECCV 2016: Computer Vision-ECCV 2016*, vol. 9905, pp. 21–37, Springer, Amsterdam, The Netherlands, October 2016.

[26] J. Huang, V. Rathod, C. Sun et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3296-3297, Honolulu, HI, USA, July 2017.

[27] A. G. Howard, M. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[28] J. Grum, "Book review: pattern recognition and neural networks by B.D. Ripley," *International Journal of Microstructure and Materials Properties*, vol. 4, no. 1, p. 146, 2009.

[29] M. Everingham, L. V. Gool, and J. Winn, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.