HELMET DETECTION USING DEEP LEARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Nadimpalli Tirumala Rama Linga Raju (39110661)
RudrarajuAbhirama Krishnam Raju (39110862)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Nadimpalli Tirumala Rama Linga Raju(39110661)** and **RudrarajuAbhirama Krishnam Raju(39110862)**who carried out the project Phase-2 entitled "**Helmet Detection Using Deep Learning"** undermy supervision from December 2023 to April 2023.

Internal Guide

Ms.NancyKirupanithi.D B.E M.Tech

Head of the Department

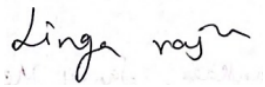
Dr. L. LAKSHMANAN, M.E., Ph.D,

Submitted for Viva voce Examination held on 24.04.23

Internal Examiner **External Examiner**

DECLARATION

I **Nadimpalli Tirumala Rama Linga Raju (Reg No:39110661)**hereby declare that the Project Phase-2 Report entitled ""**HELMET DETECTION USING DEEP LEARNING**" done by me under the guidance of **Ms.Nancy Kirupanithi.D B.E M .Tech** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering.**



DATE: 24.04.23

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D.**, **Dean**, School of Computing **Dr.S.Vigneshwari M.E., Ph.D.** and**Dr.L.Lakshmanan M.E., Ph.D. ,** Headsof the Departmentof Computer Science and Engineering for providing menecessary support and details at the right time during the progressive reviews.

I would like to express my sincere anddeep sense of gratitude to my Project Guide **MS.NancyKirupanithi.D B.E Mtech,** for her valuable guidance, suggestions, and constantencouragement that paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Departmentof Computer Science and Engineering** who were helpful in many ways for thecompletion of the project.

ABSTRACT

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 more than 5000 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. 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.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
DL	Deep Learning
CNN	Convolutional neural networks
LSTMs	Long short-term memory network
MLPs	Multilayer perceptron’s
SOMs	Self organizing maps
YOLO	You only look once
MN	Mobile net

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 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 surveillancebased 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 costeffective. However, in order to adopt such automatic solutions certain challenges need to be addressed: 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.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. 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. 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. 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, 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. Automatic detection of bike-riders without helmet falls under broad category of anomaly detection in surveillance videos. As explained in, effective automatic surveillance system generally involve following tasks: environment modelling, detection, tracking and classification of moving objects. 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

1.1 OVERVIEW

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 5000+ 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 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.

1.2DEEP LEARNING

Deep learning is the branch of machine learning which is based on artificial neural network architecture. An artificial neural network or ANN uses layers of interconnected nodes called neurons that work together to process and learn from the input data.

In a fully connected Deep neural network, there is an input layer and one or more hidden layers connected one after the other. Each neuron receives input from the previous layer neurons or the input layer. The output of one neuron becomes the input to other neurons in the next layer of the network, and this process continues until the final layer produces the output of the network. The layers of the neural network transform the input data through a series of nonlinear transformations, allowing the network to learn complex representations of the input data.

1.3 DEEP LEARNING STRATEGIES

Deep Learning models are able to automatically learn features from the data, which makes them well-suited for tasks such as image recognition, speech recognition, and natural language processing. The most widely used architectures in deep learning are feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

1.3.1 Computer vision

computer vision, Deep learning models can enable machines to identify and understand visual data. Some of the main applications of deep learning in computer vision include:Object detection and recognition: Deep learning model can be used to identify and locate objects within images and videos, making it possible for machines to perform

tasks such as self-driving cars, surveillance, and robotics. Image classification: Deep learning models can be used to classify images into categories such as animals, plants, and buildings. This is used in applications such as medical imaging, quality control, and image retrieval. Image segmentation: Deep learning models can be used for image segmentation into different regions, making it possible to identify specific features within images

1.3.2 Natural Language Processing (NLP)

In NLP, the Deep learning model can enable machines to understand and generatehuman language. Some of the main applications of deep learning in NLP include:

Automatic Text Generation – Deep learning model can learn the corpus of text and new text like summaries, essays can be automatically generated using these trained models.

Language translation: Deep learning models can translate text from one language to another, making it possible to communicate with people from different linguistic backgrounds.

Sentiment analysis: Deep learning models can analyze the sentiment of a piece of text, making it possible to determine whether the text is positive, negative, or neutral. This is used in applications such as customer service, social media monitoring, and political analysis.

Speech recognition: Deep learning models can recognize and transcribe spoken words, making it possible to perform tasks such as speech-to-text conversion, voice search, and voice-controlled devices.

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORK

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.

2.2Related Research into the safety Helmets Detection

The abovementioned 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 pre-processing aspects of image sharpness, object proportion, and the colour difference between background and foreground.Deep learning-based methods are very potential for the purpose of people's unsafe behaviour 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 resultsdemonstrated 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.

CHAPTER 3

METHODOLOGY

3.1 AIM AND SCOPE OF THE PROJECT

The main goal of helmet is to protect the drivers head in case of accident. Incase of accident, if the motorcyclist does not use can be fatal. This paper aims to propose a system for detection of motorcyclist without helmet.

The system detects motorcyclists not wearing helmets and retrieves their motorcycle number plate in real time from videos captured by CCTV cameras at road junctions by making use of Machine Learning. approach is able to detect the object in different illumination and occlusion.

3.2 OBJECTIVE OF THE PROJECT

To apply and compare different ML algorithms on the python code to detect the Helmet for the person.

To, predict the accurate output whether the person is wearing helmet or not. This may improve the safety for a person.

To, develop a Model which analyse the form of the user and gives an instant result within fraction of seconds.

This project improves the user to know who is not wearing the helmet and risking their lives in the worst traffic.

So, that we can take charge or let them know what are the consequences that will happen if we won't wear Helmet.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS

3.3.1 Software Requirements:

- ✓ Python
- ✓ Anaconda
- ✓ Jupyter Notebook

3.3.2 Hardware Requirements:

- ✓ Processor: Intel Core i5
- ✓ RAM: 8GB
- ✓ OS: Windows

3.3.3 Libraries:

- ✓ **Numpy**- NumPy is an open-source numerical and popular Python library. It can be used to perform a variety of mathematical operations on arrays and matrices. It's one of the most used scientific computing libraries, and it's often used by scientists for data analysis. Additionally, its ability to process multidimensional arrays—handling linear algebra and Fourier transformation—makes it ideal for machine learning and artificial intelligence (AI) projects.

- ✓ **SciPy**- SciPy is a free and open-source library that's based on NumPy. It can be used to perform scientific and technical computing on large sets of data. Similar to NumPy, SciPy comes with embedded modules for array optimization and linear algebra. It's considered a foundational Python library due to its critical role in scientific analysis and engineering.

- ✓ **Pandas-** Library of python which can be used easily. It gives speedy results and is also easily understandable. It is a library that can be used without any cost. We have used it for data analysis and to read the dataset.It is built on top of another package named Numpy, which provides support for multi-dimensionalarrays.Thetwoprimarydatastructuresofpandasare**Series**(1-dimensional) and **DataFrame**(2-dimensional).we mainly use dataframe in our machine learningtask.

- ✓ **Scikit-Learn-** Based on NumPy and SciPy, scikit-learn is a free Python library that's often considered a direct extension of SciPy. It was specifically designed for data modeling and developing machine learning algorithms, both supervised and unsupervised.

- ✓ **Seaborn-**Seaborn**i**sadatavisualizationlibrarybuiltontopofmatplotlib andcloselyintegratedwithpandasdatastructuresinPython.Visualization is the central part of Seaborn which helps in exploration and understanding of data.

- ✓ **Theano-** Theano is a numerical computation Python library made specifically for machine learning. It allows for efficient definition, optimization, and evaluation of mathematical expressions and matrix calculations to employ multidimensional arrays to create deep learning models. It's a highly specific library and almost exclusively used by ML and DL developers and programmers.

- ✓ **Pandas Profiling**-This is a library of pythonwhich can be used by anyone free of cost. It is used for data analysis. We have used this for getting the report of the dataset.

- ✓ **TensorFlow-** It is a free and open-source Python library that specializes in differentiable programming. The library offers a collection of tools and resources

that help make building DL and ML models and neural networks straightforward for beginners and professionals. TensorFlow's architecture and framework are flexible and allow it to run on several computational platforms such as CPU and GPU. However, it performs its best when working on a tensor processing unit (TPU).

- ✓ **Keras-** It is an open-source Python library designed for developing and evaluating neural networks within deep learning and machine learning models. It can run on top of Theano and TensorFlow, making it possible to start training neural networks with a little code. The Keras library is modular, flexible, and extensible, making it beginner- and user-friendly. It also offers a fully functioning model for creating neural networks as it integrates with objectives, layers, optimizers, and activation functions.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEMS

4.1 PROPOSED SYSTEM

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.

Algorithm

✓ Data Processing

✓ Working on Algorithms

✓ Model Development

✓ Model Evaluation

✓ Deployment of the model

✓ Detection Analysis

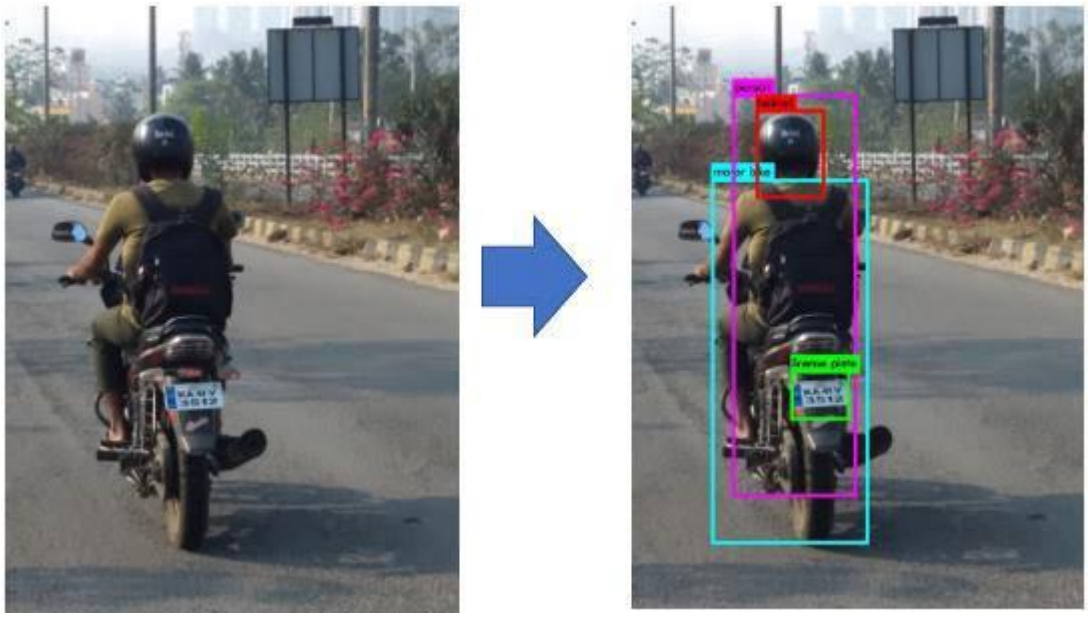


Fig 1 - Proposed approach for detecting helmet.

ADVANTAGES OF THE PROPOSED SYSTEM

• Easily Detects helmet

• High Accuracy

4.2 PROGRAMMING LANGUAGE

4.2.1 PYTHON

Python is the best programming language fitted for Machine Learning. In step with studies and surveys, Python is the fifth most significant language yet because the preferred language for machine learning and information science. It's owing to the subsequent strengths that Python has –

✓ **Easy to be told and perceive-**The syntax of Python is simpler; thence it's comparatively straightforward, even for beginners conjointly, to be told and perceive the language.

✓ **Multi-purpose language −** Python could be a multi-purpose programing language as a result it supports structured programming and object-oriented programming yet as practical programming.

✓ **Support of open supply community −**As an open supply programming language, Python is supported by an awfully giant developer community. Because of this, the bugs square measure simply mounted by the Python community. This characteristic makes Python terribly strong and adaptative.

HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the

Netherlands.Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

APPLICATION OF PYTHON

- ✓ **GUI Programming**: Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- ✓ **Scalable**: Python provides a better structure and support for large programs than shell scripting.
- ✓ **Large standard libraries to solve common tasks**: Python has a number of standard libraries which makes life ofa programme.
- ✓ **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- ✓ **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- ✓ **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- ✓ **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

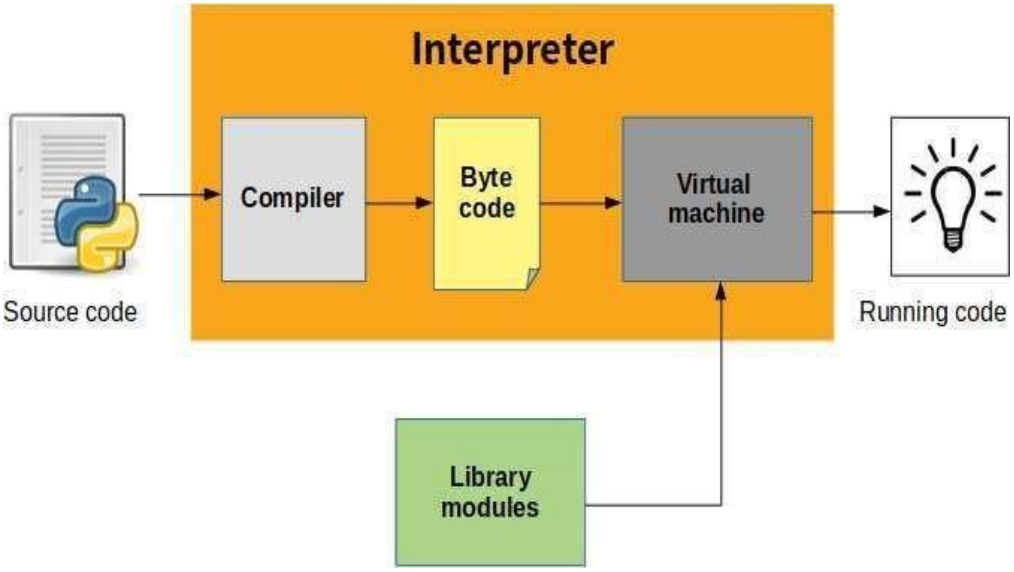


Fig 2 – Execution of source code

FEATURES OF PYTHON

- ✓ It supports functional and structured programming methods as well as OOP.
- ✓ It can be used as a scripting language or can be compiled to byte-code for building large applications.
- ✓ It provides very high-level dynamic data types and supports dynamic type checking.
- ✓ It supports automatic garbage collection.
- ✓ It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

4.2.2 DOMAIN

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars)

4.3 SYSTEM ARCHITECTURE

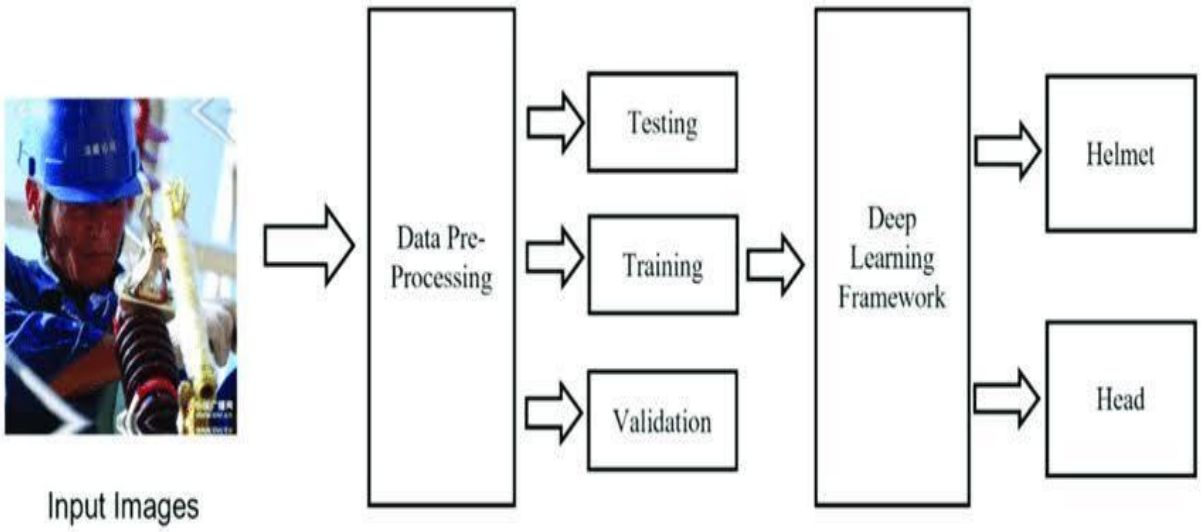


Fig 3 - System Architecture

4.4 ALGORITHMS

4.4.1 Convolutional Neural Networks(CNNs)

CNN popularly known as ConvNets majorly consists of several layers and are specifically used for image processing and detection of objects. It was developed in 1998 by Yann LeCun and was first called LeNet. Back then, it was developed to recognize digits and zip code characters. CNNs have wide usage in identifying the image of the satellites, medical image processing, series forecasting, and anomaly detection.

CNNs process the data by passing it through multiple layers and extracting features to exhibit convolutional operations. The Convolutional Layer consists of Rectified Linear Unit (ReLU) that outlasts to rectify the feature map. The Pooling layer is used to rectify these feature maps into the next feed. Pooling is generally a sampling algorithm that is down-sampled and it reduces the dimensions of the feature map.

Later, the result generated consists of 2-D arrays consisting of single, long, continuous, and linear vector flattened in the map. The next layer i.e., called Fully Connected Layer which forms the flattened matrix or 2-D array fetched from the Pooling Layer as input and identifies the image by classifying it.

Advantage Of CNN:

The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs it learns distinctive features for each class by itself. CNN is also computationally efficient.

Disadvantages of CNN

Some of the disadvantages of CNNs: include the fact that a lot of training data is needed for the CNN to be effective and that they fail to encode the position and orientation of objects. They fail to encode the position and orientation of objects. They have a hard time classifying images with different positions.

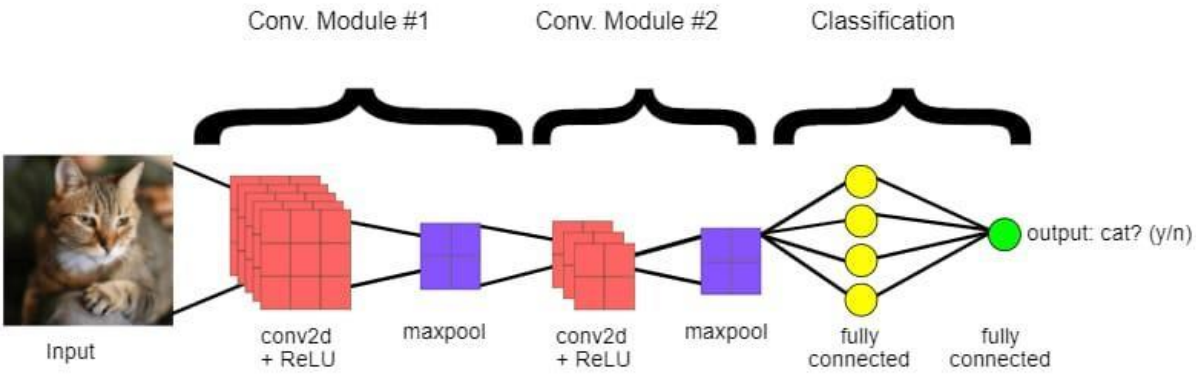


Fig 4 - CNN Architecture

4.4.2 Long Short-Term Memory Networks(LSTMs)

Long Short-Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was designed by Hochreiter&Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long-term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give an efficient performance. LSTM can by default retain the information for a long period of time. It is used for processing, predicting, and classifying on the basis of time-series data.

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is specifically designed to handle sequential data, such as time series, speech, and text. LSTM networks are capable of learning long-term dependencies in sequential data, which makes them well suited for tasks such as language translation, speech recognition, and time series forecasting.

A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network to learn long-term dependencies. LSTMs address this problem by introducing a memory cell, which is a container that can hold information for an extended period of time. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate. These gates decide what information to add to, remove from, and output from the memory cell.

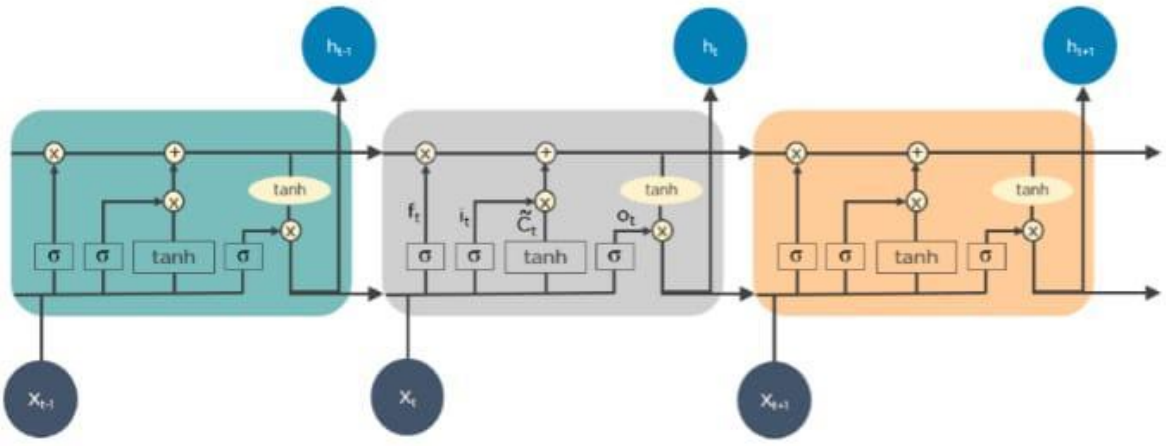


Fig 5–Long Short-term Memory Networks

LSTM work in a sequence of events. First, they don't tend to remember irrelevant details attained in the previous state. Next, they update certain cell-state values selectively and finally generate certain parts of the cell-state as output. Below is the diagram of their operation.

APPLICATIONS OF LONG SHORT TERM MEMORY NETWORKS

There are mainly four sectors where Random-forest mostly used:

1. Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
2. Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.
3. Land Use: We can identify the areas of similar land use by this algorithm.
4. Marketing: Marketing trends can be identified using this algorithm.

Advantages of LSTM

Random Forest is capable of performing both Classification and Regression tasks.

- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of LSTM

- They are more complicated than traditional RNNs and require more training data in order to learn effectively.

4.4.3MultilayerPerceptrons(MLPs)

MLPs are the base of deep learning technology. It belongs to a class of feed-forward neural networks having various layers of perceptron’s. These perceptron’s have various activation functions in them. MLPs also have connected input and output layers and their number is the same. Also, there's a layer that remains hidden amidst these two layers. MLPs are mostly used to build image and speech recognition systems or some other types of the translation software.

The working of MLPs starts by feeding the data in the input layer. The neurons present in the layer form a graph to establish a connection that passes in one direction. The weight of this input data is found to exist between the hidden layer and the input layer. MLPs use activation functions to determine which nodes are ready to fire. These activation functions include tanh function, sigmoid and ReLUs. MLPs are mainly used to train the models to understand what kind of co-relation the layers are serving to achieve the desired output from the given data set. See the below image to understand better

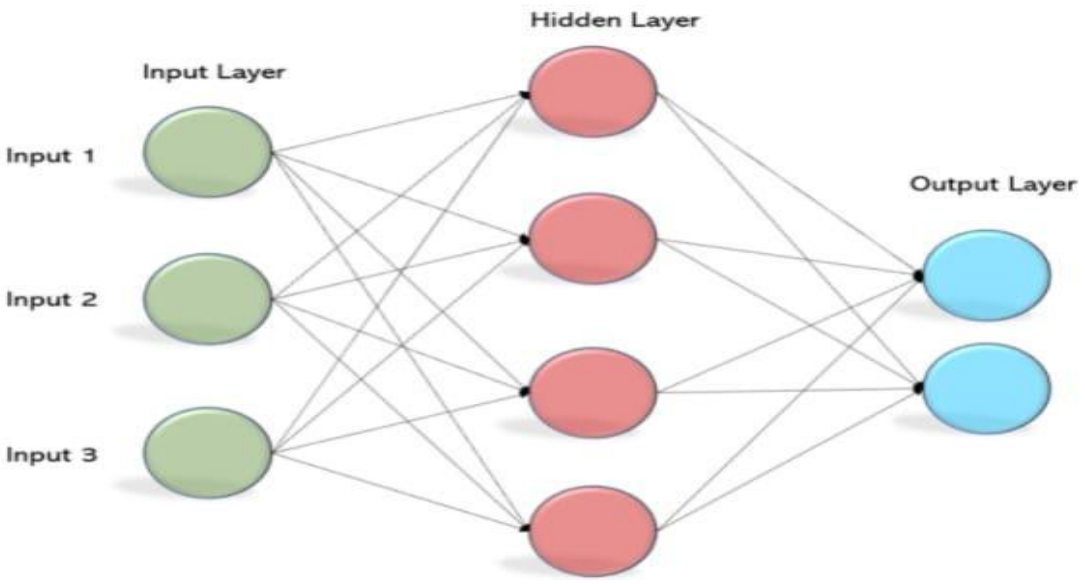


Fig 6- Multilayer Perceptrons(MLPs)

.

Advantages of the MLPs

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The MLPs contains lots of layers, which makes it complex.
- For more class labels, the computational complexity of the decision tree may increase.

4.4.4Self Organizing Maps (SOMs)

SOMs were invented by TeuvoKohenen for achieving data visualization to understand the dimensions of data through artificial and self-organizing neural networks. The attempts to achieve data visualization to solve problems are mainly done by what humans cannot visualize. These data are generally high-dimensional so there are lesser chances of human involvement and of course less error.

SOMs help in visualizing the data by initializing weights of different nodes and then choose random vectors from the given training data. They examine each node to find the relative weights so that dependencies can be understood. The winning node is decided and that is called Best Matching Unit (BMU). Later, SOMs discover these winning nodes but the nodes reduce over time from the sample vector. So, the closer the node to BMU more is the more chance to recognize the weight and carry out further activities. There are also multiple iterations done to ensure that no node closer to BMU is missed. One example of such is the RGB color combinations that we use in our daily tasks. Consider the below image to understand how they function.

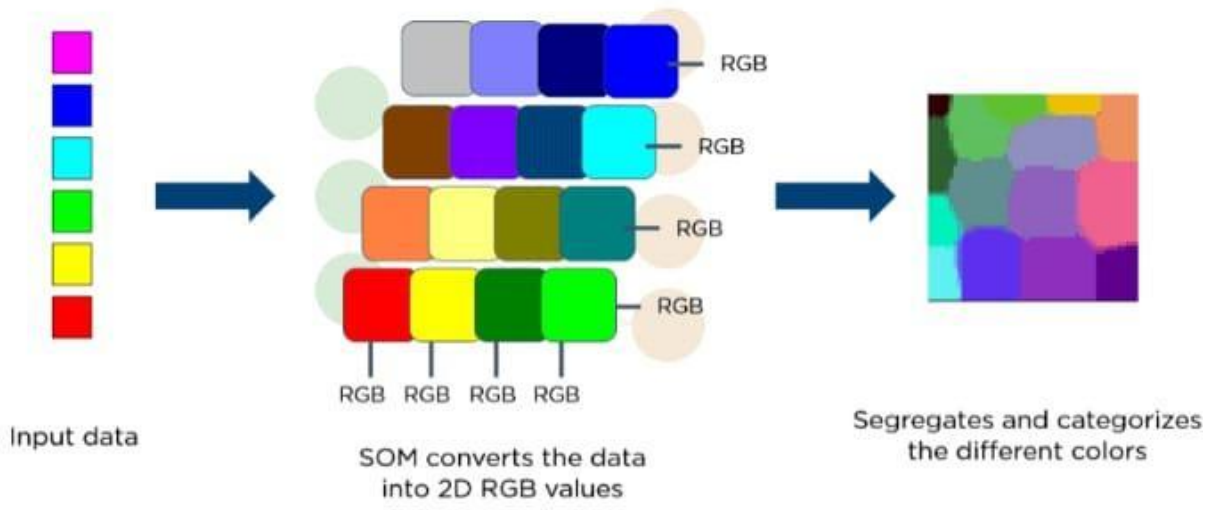


Fig 7- Self Organizing Maps

APPLICATIONS OF SELF ORGANIZING MAPS

- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

- Below are some points to remember while selecting the value of K in the K-NN algorithm:
- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of SOMs Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of SOMs Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

4.4.5 YOU ONLY LOOK ONCE(YOLO)

YOLO was proposed by Joseph Redmond *et al.* in 2015. It was proposed to deal with the problems faced by the object recognition models at that time, Fast R-CNN is one of the state-of-the-art models at that time but it has its own challenges such as this network cannot be used in real-time, because it takes 2-3 seconds to predicts an image and therefore cannot be used in real-time. Whereas, in YOLO we have to look only once in the network i.e. only one forward pass is required through the network to make the final predictions.

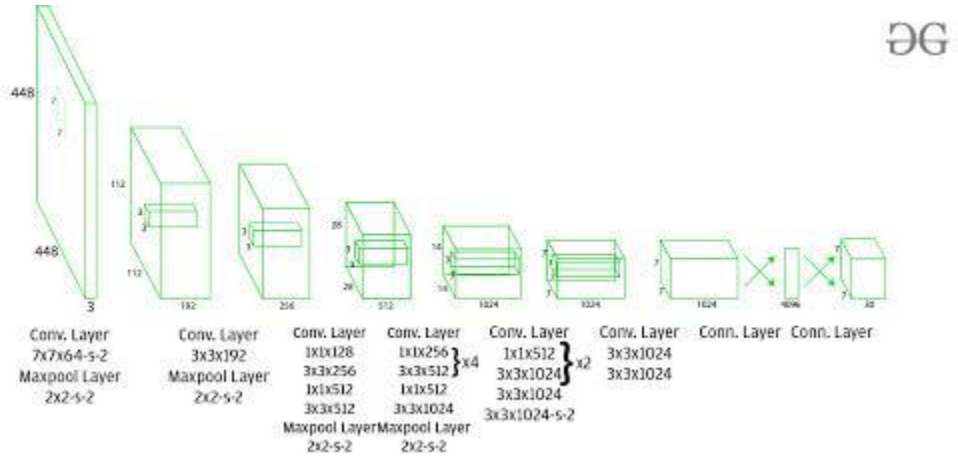


Fig 8 – YOLO Architecture

This architecture takes an image as input and resizes it to *448*448* by keeping the aspect ratio same and performing padding. This image is then passed in the CNN network. This model has *24 convolution layers, 4 max-pooling layers followed by 2 fully connected*

layers. For the reduction of the number of layers (Channels), we use *1*1* convolution that is followed by *3*3* convolution. Notice that the last layer of YOLOv1 predicts a cuboidal output. This is done by generating *(1, 1470)* from final fully connected layer and reshaping it to size *(7, 7, 30)*.

Training:

This model is trained on the *ImageNet-1000* dataset. The model is trained over a week and achieve top-5 accuracy of *88% on ImageNet 2012* validation which is comparable to GoogLeNet (2014 ILSVRC winner), the state of the art model at that time. Fast YOLO uses fewer layers *(9 instead of 24)* and fewer filters. Except this, the fast YOLO have all parameters similar to YOLO.YOLO uses sum-squared error loss function which is easy to optimize. However, this function gives equal weight to the classification and localization task. The loss function defined in YOLO as follows:

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Fig 9 – Yolo Training Set

The first two parts of the above loss equation represent localization mean-squared error, but the other three parts represent classification error. In the localization error, the first term calculates the deviation from the ground truth bounding box. The second term

calculates the square root of the difference between height and width of the bounding box. In the second term, we take the square root of width and height because our loss function should be able to consider the deviation in terms of the size of the bounding box. For small bounding boxes, the little deviation should be more imp.There are three terms in classification loss, the first term calculates the sum-squared error between the predicted confidence score that whether the object present or not and the ground truth for each bounding box in each cell. Similarly, The second term calculates the mean-squared sum of cells that do not contain any bounding box, and a regularization parameter is used to make this loss small. The third term calculates the sum-squared error of the classes belongs to these grid cells.

Fig 10 – YOLO

Advantage of YOLO :

The advantage is that, we have the freedom to attempt anything we want because we only have one life to live and we want to enjoy it.

Disadvantages of YOLO:

- Comparatively low recall and more localization error compared to Faster R_CNN.
- Struggles to detect close objects because each grid can propose only 2 bounding boxes.

4.4.6 MOBILE NET

MobileNet is a computer vision model open-sourced by Google and designed for training classifiers. It uses depthwise convolutions to significantly reduce the number of parameters compared to other networks, resulting in a lightweight deep neural network. MobileNet is Tensorflow's first mobile computer vision model.

MobileNet is a class of convolutional neural network (CNN) that was open-sourced by Google, and therefore, provides an excellent starting point for training classifiers that are insanely small and insanely fast.

The speed and power consumption of the network is proportional to the number of multiply-accumulates (MACs) which is a measure of the number of fused multiplication and addition operations.

This convolution originated from the idea that a filter's depth and spatial dimension can be separated, thus, the name separable. Let's take the example of the Sobel filter used in image processing to detect edges.

You can separate the height and width dimensions of these filters. Gx filter can be viewed as a matrix product of [1 2 1] transpose with [-1 0 1].

You'll notice that the filter has disguised itself. It shows it had nine parameters, but it has six. This is possible because of the separation of its height and width dimensions.

The same idea applies to a separate depth dimension from horizontal (width*height), which gives us a depthwise separable convolution where we perform depthwise convolution. After that, we use a 1*1 filter to cover the depth dimension.

One thing to notice is how much the parameters are reduced by this convolution to output the same number of channels. To produce one channel, we'd need 3*3*3 parameters to perform depth-wise convolution and 1*3 parameters to perform further convolution in-depth dimension.

But if we'd need three output channels, we'd only need 31*3 depth filter, giving us a total of 36 (= 27 +9) parameters, while for the same number of output channels in regular convolution, we'd need 33*3*3 filters giving us a total of 81 parameters.

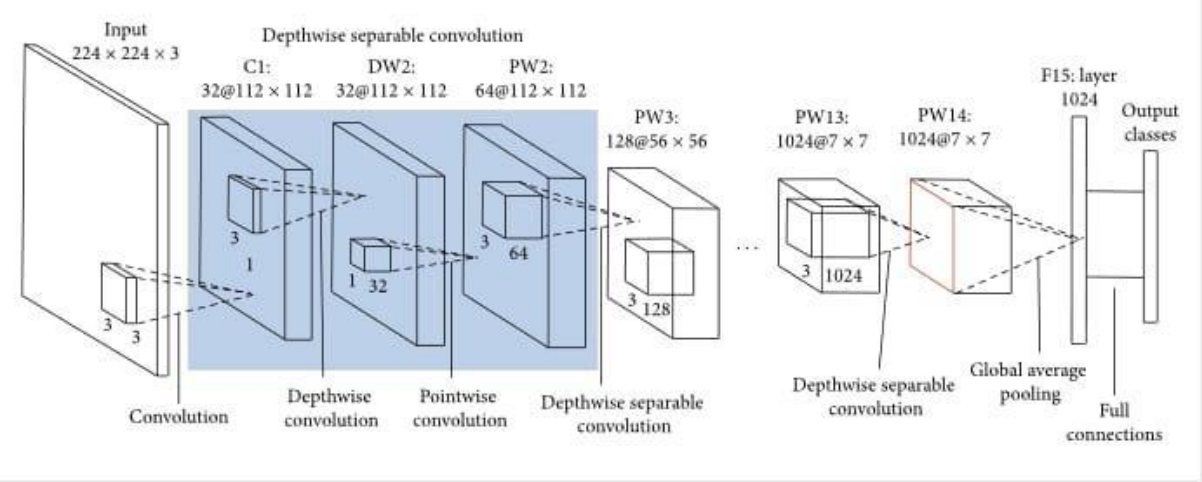


Fig 11 – Mobile Net Architecture

Image Recognition with Mobile NetImage plays an important role in many fields like medical disease analysis, and many more. In this article, we will mainly focus on how to

Recognize the given image, what is being displayed. We are assuming to have a pre-knowledge of TensorFlow, Keras, Python, Machine Learning

We are aimed to recognize the given image using machine learning. We are assuming we are already having a pre-trained model in our TensorFlow which we will be using to Recognize images. So, we will be using Keras of TensorFlow to import architectures which will help us to recognize images and to predict the image in a better way using coordinates and indexing, we will be using NumPy as a tool.

Fig 12 – Mobile Net

Advantages of Mobile Net :

MobileNets are a family of mobile-first computer vision models for TensorFlow, designed to effectively maximize accuracy while being mindful of the restricted resources for an on-device or embedded application.

MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use-cases. They can be built upon for classification, detection, embeddings and segmentation.

4.5 PROJECT MANAGEMENT PLAN

The project contains three parts:

- DATASET COLLECTION.
- TRAIN AND TEST THE MODEL.
- DEPLOY THE MODELS.

Dataset Collection- 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 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. 5000+ 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 5000+ 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.

Fig 13 - Data Collection

- **Deploy the models**- 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.

Following are the steps to do this project (use Jupyter Notebook):

1. Collect the dataset.
2. Import the necessary libraries.
3. Visualize the dataset.
4. Train the dataset using the CNN, LSTMs, MLPs, SOMs, YOLO, Mobile Net
5. Test the model and find the accuracies of Six algorithms.
6. Deploy the mode

7.Detecting Whether a person is wearning Helmet or not

CHAPTER 5

IMPLEMENTATION DETAILS

One important element of deep learning and machine learning at large is dataset. A good dataset will contribute to a model with good precision and recall. In the realm of object detection in images or motion pictures. • For Motorcycle detection: we used trained model with COCO Dataset with accuracy of 99%. • For Helmet Detection: We created our own Yolov3 Model with our own dataset with 5000+ images of helmet and non-helmet riders.

• For License plate: We used API for extraction and web automation for getting details for E- Challan Generation.

5.1MODEL FOR MOTORCYCLE DETECTION:

Coco Dataset for Motorcycle Detection: COCO is a large-scale object detection, segmentation, and captioning dataset. This version contains images, bounding boxes " and labels for the 2017 version. Coco defines 80 classes. COCO stands for Common Object Context. The COCO dataset contains the images which are captured in our daily life scenes. COCO provides multi-object labeling, segmentation mask annotations, image captioning, key-point detection and panoptic segmentation annotations with a total of 80 categories, making it a very versatile and multi-purpose dataset. As we have mentioned above that the COCO dataset contains a total of 80 categories, we import the required libraries that are required to access the COCO dataset. The input contains many objects like Buses, Cars, and Motorcycles etc. The libraries imported will helps divide the required objects from all other objects and the detected objects will be given certain IDs to access them later. Coco.GetObjectIds is a function used to get the IDs for the differentiated objects. The 80 categories are as follows

5.2 MODEL FOR HELMET DETECTION

Fig 14 – Flowchart for Helmet Model

5.2.1 PROCEDURE FOR TRAINING A MOBILENET HELMET MODEL

Gathering images (Creating data set): To detect a bike rider with helmet orwithouthelmet. We need bunch of images of bike-riders with helmet, bike-rider without helmet andbike licenseplate.Inthis project,weused5000+images.

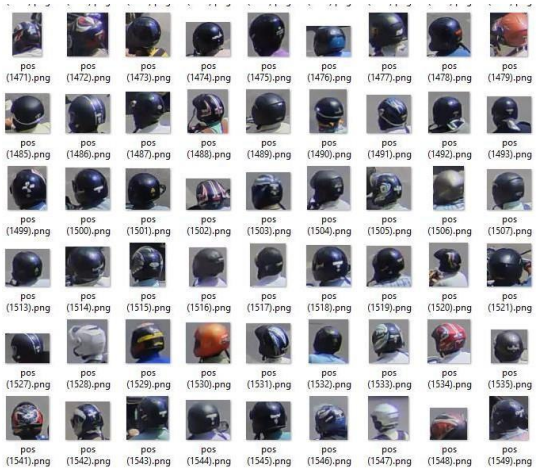


Fig 15– Helmet Images

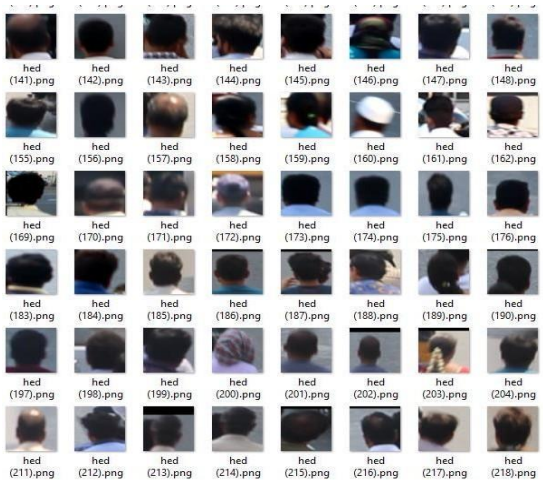


Fig 16 – Non Helmet Images

Label Images -:

Label the all images with the help of**LableImg tool**. In this project, **Helmetclass**was createdwith thehelp of LableImg tool. Create.xmlfilecorrespondingto eachimagewiththeabovefollowingcategoriesofclassesNow that our dataset labels are in the required format, we need to create a train-test split. Ichose to create a test set containing 10% of the images in the dataset.Configuring MOBILENET with your dataset. Now that we have created our train and test sets, we need to make somechangestotraintheMOBILENET modelonthedataset.

Training -:

Now that our dataset is ready to use, we can begin training. Before we start, compilethe darknet repository with the make command. To compile with specific options, such asGPU, CUDNN and OPENCV. This will create a darknet executable. Trained weights formodel. You can set other parameters (learning rate, momentum, weight decay etc by editingthe correspondinglines). Finally,modelis readytouse.

5.3 NUMBER PLATE DETECTION

Platerecognizer:isanopen-sourceAutomatic LicensePlateRecognitionlibrarywritten in C++ with bindings in C#, Java, Node.js, and Python. The library analyses imagesand video streams to identify license plates. The output is the text representation of anylicense platecharacters.

TheASP.NETWebAPIisanextensibleframeworkforbuildingHTTPbasedservices that can be accessed in different applications on different platforms such as web,windows,mobileetc.ItworksmoreorlessthesamewayasASP.NETMVCwebap plication except that itsends data as a responseinsteadof html view. Itis like a webserviceorWCFservicebuttheexceptionisthatitonlysupportsHTTPprotocol.

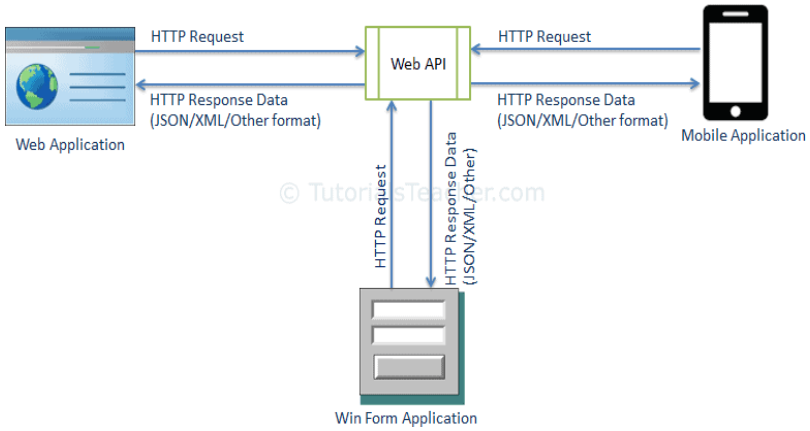


Fig 17 – Working of Web API

Table Plate Recognizer Snapshot API! You can use our API to access our API endpoints,whichcanreadlicenseplatesfromimages.PlateRecognizerprovidesaccu rate,fast,developer-friendly Automatic License Plate Recognition (ALPR) software that works in allenvironments.

Thesoftwarecanbeusedin manyways:

1. Recognizelicenseplatesfromcamerastreams.Theresultsarebrowsable,sea rchable,andcantrigger alerts. The data repository can be in the cloud or stored entirely within your on-sitenetwork.

2. Recognizelicenseplatesfromcamerastreamsandsendthe resultstoyourownapplication.
3. Integratelicense platerecognitionintoyour applicationdirectlyin-code.

CHAPTER 6

RESULTS AND DISCUSSION

RESULTS -:

In the paper, the open-source TensorFlow framework is chosen to train the model. The pretrained 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 detectionshow the variation of the classification loss function, localization loss function, and regularization loss function against the steps. 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.

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 wornby 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 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 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 is the same as the first one. In Figure 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.

DISCUSSION -:

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.

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 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.

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.

CHAPTER 7

CONCLUSION

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.

A Non-Helmet Rider and Triple Riding Detection system is developed where a video file is taken as input. If the motorcycle rider in the video footage is not wearing helmet while riding the motorcycle, or riding with three members, then the license plate number of that motorcycle is extracted and displayed for above cases separately. Object detection principle with YOLO architecture is used for motorcycle, person, helmet and license plate detection. Google Spreadsheet is used for license plate number extraction if rider is not wearing helmet or triple riding. The characters are extracted from LP so that it can be used for other purposes. All the objectives of the project are achieved satisfactorily.

7.1 Future work

The system implemented is a prototype. It can be expanded to process the day-to-daytraffic video by attaining the permissions of the required authorities. A large databaseis created to maintain the records of the violators and their payment of the challans being monitored every few minutes. Also, the identification of the license plate becomes the core part of this project. So, a camera of high resolution is recommended to maintain precision and accuracy. For sending the challan directly to offender's mobile numbers, the subscriptions for SMS are required, as of now it is sent through mail ids, but the motto to send the challan to their mails as well as through SMS.

REFERENCES

1. Kelm, A.; Lausat, L.; Meins-Becker, A.; Platz, D.; Khazaee, M.J.; Costin, A.M.; Helmus, M.; Teizer, J. Mobile passive Radio Frequency Identification (RFID) portal for automated and rapid control of Personal Protective Equipment (PPE) on construction sites. Autom. Constr. 2013, 36, 38–52. [CrossRef

2. Bureau of Labor Statistics. Industries at a Glance. Available online: https://www.bls.gov/iag/tgs/iag23.htm (accessed on 30 March 2022).

3. Chang, X.M.; Liu, X. Fault tree analysis of unreasonably wearing helmets for builders. J. Jilin Jianzhu Univ. 2018, 35, 67–71.

4. Ahn, Y.-S.; Bena, J.F.; Bailer, A.J. Comparison of unintentional fatal occupational injuries in the Republic of Korea and the United States. Inj. Prev. 2004, 10, 199–205. [CrossRef] [PubMed]

5. International Labour Organization (ILO). World Statistic. 2020. Available online: www.ilo.org/moscow/areas-of-work/occupational-safety-and-health/WCMS_249278/lang-- en/index.htm (accessed on 30 March 2022).

6. Ministry of Housing and Urban-Rural Development of the People's Republic of China.Availableonline:https://www.mohurd.gov.cn/wjfb/201903/t20190326_239913.html (accessed on 30 March 2022).

7. Statistics—Work-Related Fatal Injuries in Great Britain. Available online: https://www.hse.gov.uk/statistics/fatals.htm (accessed on 30 March 2022).
8. OSHA. Occupational Safety and Health Administration Commonly Used Statistics. InOSHADataandStatistics.;2020.Availableonline:www.osha.gov/oshstats/commonstats.html (accessed on 30 March 2022).
9. . U.S. Department of Health and Human Services. In Worker Deaths by Falls. A Summary of Surveillance Findings and Investigative Case Reports, No. September 2000.. Available online: www.cdc.gov/niosh (accessed on 30 March 2022).
10. Zhai, H. Research on image recognition based on deep learning technology. In Proceedings of the 2016 4th International Conference on Advanced Materials and Information Technology Processing (AMITP 2016), Guilin, China, 24–25 September 2016; Atlantis Press: Hong Kong, Beijing, 2016; pp. 266–270. [CrossRef]
11. Zhang, H.; Yan, X.; Li, H.; Jin, R.; Fu, H. Real-Time Alarming, Monitoring, and Locating for Non-Hard-Hat Use in Construction. J. Constr. Eng. Manag. 2019, 145, 04019006. [CrossRef]
12. Escorcia, V.; Dávila, M.A.; Golparvar-Fard, M.; Niebles, J.C. Automated vision-based recognition of construction worker actions for building interior construction operations using RGBD cameras. In Proceedings of the Construction Research Congress 2012: Construction Challenges in a Flat World, West Lafayette, IN, USA, 21–23 May 2012.

APPENDIX

A)SOURCE CODE

```
mport necessary packages
fromimutils.videoimportVideoStream
importnumpyasnp
fromimutils.videoimportFPS
importimutils
importtime
import cv2
fromkeras.modelsimportload_model

# initialize the list of class labels MobileNet SSD was trained to detect
# generate a set of bounding box colors for each class
CLASSES = ['background', 'aeroplane', 'bicycle', 'bird', 'boat', 'bottle', 'bus',
'car', 'cat', 'chair', 'cow', 'diningtable', 'dog', 'horse', 'motorbike',
'person', 'pottedplant', 'sheep', 'sofa', 'train', 'tvmonitor']
#CLASSES = ['motorbike', 'person']
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

# load our serialized model from disk
print("[INFO] loading model...")
```

```
net = cv2.dnn.readNetFromCaffe('MobileNetSSD_deploy.prototxt.txt',
'MobileNetSSD_deploy.caffemodel')

print('Loading helmet model...')
loaded_model = load_model('new_helmet_model.h5')
loaded_model.compile(loss='binary_crossentropy', optimizer='rmsprop',
metrics=['accuracy'])

# initialize the video stream,
print("[INFO] starting video stream...")

# Loading the video file
cap = cv2.VideoCapture('vid1.mp4')

# time.sleep(2.0)

# Starting the FPS calculation
fps = FPS().start()

# loop over the frames from the video stream
# i = True
whileTrue:
    # i = not i
    # ifi==True:

    try:
        # grab the frame from the threaded video stream and resize it
        # to have a maxm width and height of 600 pixels
        ret, frame = cap.read()

        # resizing the images
        frame = imutils.resize(frame, width=600, height=600)

        # grab the frame dimensions and convert it to a blob
        (h, w) = frame.shape[:2]

        # Resizing to a fixed 300x300 pixels and normalizing it.
        # Creating the blob from image to give input to the Caffe Model
        blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 0.007843,
(300, 300), 127.5)

        # pass the blob through the network and obtain the detections and
predictions
        net.setInput(blob)

        detections = net.forward()  # getting the detections from the network

        persons = []
```

```
        person_roi = []
        motorbi = []

        # loop over the detections
        foriinnp.arange(0, detections.shape[2]):
            # extract the confidence associated with the prediction
            confidence = detections[0, 0, i, 2]

            # filter out weak detections by ensuring the confidence
            # is greater than minimum confidence
            ifconfidence>0.5:

                # extract index of class label from the detections
                idx = int(detections[0, 0, i, 1])

                ifidx == 15:
                    box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                    (startX, startY, endX, endY) = box.astype("int")
                    # roi = box[startX:endX, startY:endY/4]
                    # person_roi.append(roi)
                    persons.append((startX, startY, endX, endY))

                ifidx == 14:
                    box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                    (startX, startY, endX, endY) = box.astype("int")
                    motorbi.append((startX, startY, endX, endY))

        xsdiff = 0
        xediff = 0
        ysdiff = 0
        yediff = 0
        p = ()

        foriinmotorbi:
            mi = float("Inf")
            forjinrange(len(persons)):
                xsdiff = abs(i[0] - persons[j][0])
                xediff = abs(i[2] - persons[j][2])
                ysdiff = abs(i[1] - persons[j][1])
                yediff = abs(i[3] - persons[j][3])

                if (xsdiff+xediff+ysdiff+yediff) <mi:
                    mi = xsdiff+xediff+ysdiff+yediff
                    p = persons[j]
                    # r = person_roi[j]


            iflen(p) != 0:
```

```
                # display the prediction
                label = "{}".format(CLASSES[14])
                print("[INFO] {}".format(label))
                cv2.rectangle(frame, (i[0], i[1]), (i[2], i[3]), COLORS[14], 2)
                y = i[1] - 15ifi[1] - 15>15elsei[1] + 15
                cv2.putText(frame, label, (i[0], y), cv2.FONT_HERSHEY_SIMPLEX,
0.5, COLORS[14], 2)
                label = "{}".format(CLASSES[15])
                print("[INFO] {}".format(label))

                cv2.rectangle(frame, (p[0], p[1]), (p[2], p[3]), COLORS[15], 2)
                y = p[1] - 15ifp[1] - 15>15elsep[1] + 15

                roi = frame[p[1]:p[1]+(p[3]-p[1])//4, p[0]:p[2]]
                print(roi)
                iflen(roi) != 0:
                    img_array = cv2.resize(roi, (50,50))
                    gray_img = cv2.cvtColor(img_array, cv2.COLOR_BGR2GRAY)
                    img = np.array(gray_img).reshape(1, 50, 50, 1)
                    img = img/255.0
                    prediction = loaded_model.predict_proba([img])
                    cv2.rectangle(frame, (p[0], p[1]), (p[0]+(p[2]-p[0]),
p[1]+(p[3]-p[1])//4), COLORS[0], 2)
                    cv2.putText(frame, str(round(prediction[0][0],2)), (p[0], y),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[0], 2)

    except:
        pass

    cv2.imshow('Frame', frame)  # Displaying the frame
    key = cv2.waitKey(1) &0xFF

    ifkey == ord('q'): # if 'q' key is pressed, break from the loop
        break

    # update the FPS counter
    fps.update()


# stop the timer and display FPS information
fps.stop()

print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

cv2.destroyAllWindows()
cap.release()   # Closing the video stream
```

Code2 -:

```
# mounting google drive to colab to get the training data
fromgoogle.colabimport drive
drive.mount('/content/drive')
```

```
%cp -r 'drive/My Drive/images.zip''images.zip'
```

```
!unzip images.zip
```

```
importnumpyasnp
frommatplotlibimportpyplotasplt
importos
importcv2
importrandom
fromkeras.modelsimport Sequential
fromkeras.layersimport Dense, Dropout, Flatten, Conv2D, MaxPooling2D,
BatchNormalization
fromkeras.optimizersimport RMSprop
```

```
DATADIR = 'images'
CLASS = ['neg','pos']
IMG_SIZE = 50

neg = []
pos = []

# building the training data
defcreate_training_data():
  forclinCLASS:
    path = os.path.join(DATADIR, cl)
    class_num = CLASS.index(cl)
    ifclass_num == 0:
      forimginos.listdir(path):
        try:
          img_array = cv2.imread(os.path.join(path, img))
          img_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
          gray_img = cv2.cvtColor(img_array, cv2.COLOR_BGR2GRAY)
          neg.append([gray_img, class_num])
        exceptExceptionase:
          pass
    ifclass_num == 1:
      forimginos.listdir(path):
        try:
```

```
          img_array = cv2.imread(os.path.join(path, img))
          img_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
          gray_img = cv2.cvtColor(img_array, cv2.COLOR_BGR2GRAY)
          pos.append([gray_img, class_num])
        exceptExceptionase:
          pass

random.shuffle(neg)
  random.shuffle(pos)
  training_data = neg[:len(pos)]+pos
  returntraining_data

training_data = create_training_data()
```

```
#shuffle the data
random.shuffle(training_data)
print(len(training_data))
```

```
X = []
y = []

forfeatures, labelintraining_data:
  X.append(features)
  y.append(label)
```

```
X = np.array(X).reshape(-1, IMG_SIZE, IMG_SIZE, 1)
```

```
# normalizing the data
X = X/255.0
```

```
# building the model
model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
input_shape=X.shape[1:], data_format='channels_last',))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))
```

```
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))

model.add(Conv2D(256, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

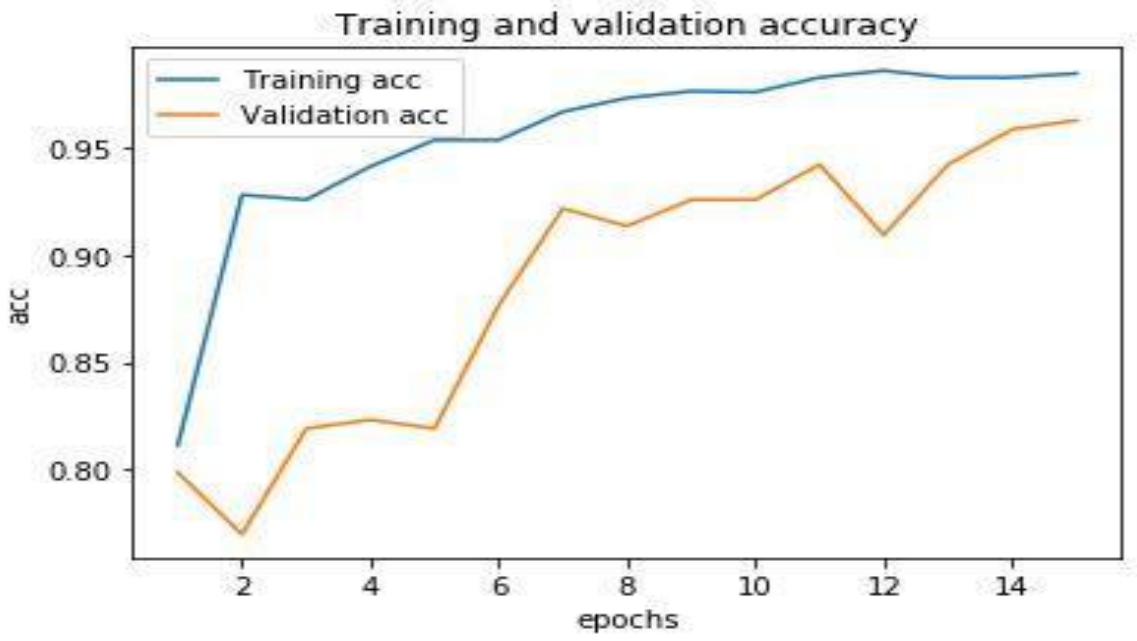
```
model.summary()
```

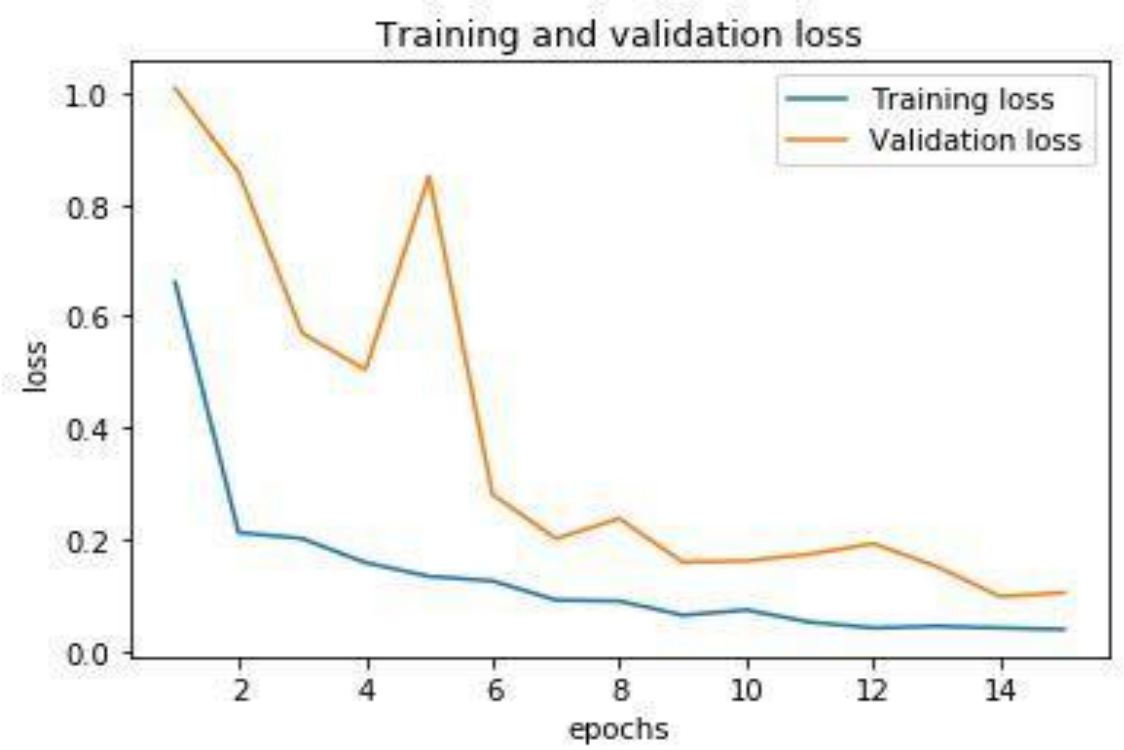
```
# helper function to plot the results
defplot_result(history):
  acc = history.history['acc']
  val_acc = history.history['val_acc']
  loss = history.history['loss']
  val_loss = history.history['val_loss']

  epochs = range(1, len(acc)+1)

  plt.plot(epochs, acc, label='Training acc')
  plt.plot(epochs, val_acc, label='Validation acc')
  plt.title('Training and validation accuracy')
  plt.xlabel('epochs')
  plt.ylabel('acc')
  plt.legend()

  plt.figure()

  plt.plot(epochs, loss, label='Training loss')
```

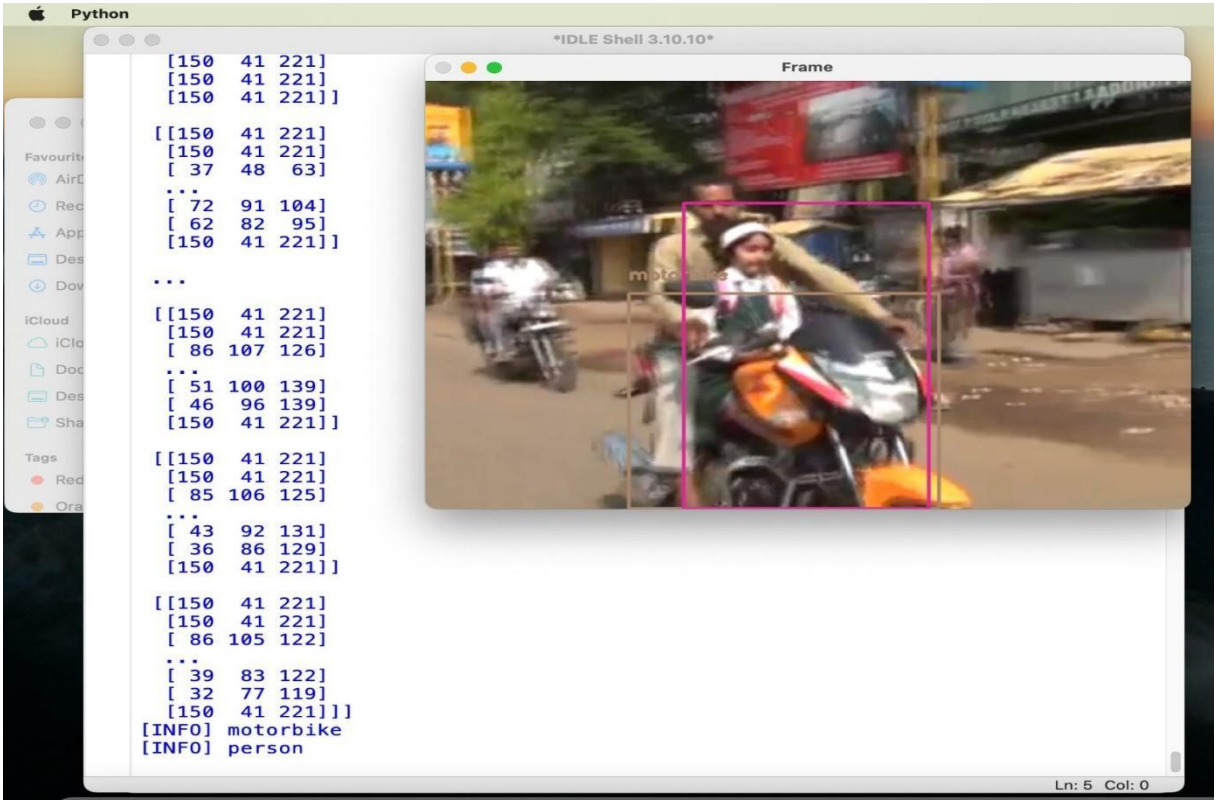
```
  plt.plot(epochs, val_loss, label='Validation loss')
  plt.title('Training and validation loss')
  plt.xlabel('epochs')
  plt.ylabel('loss')
  plt.legend()

  plt.show()
```

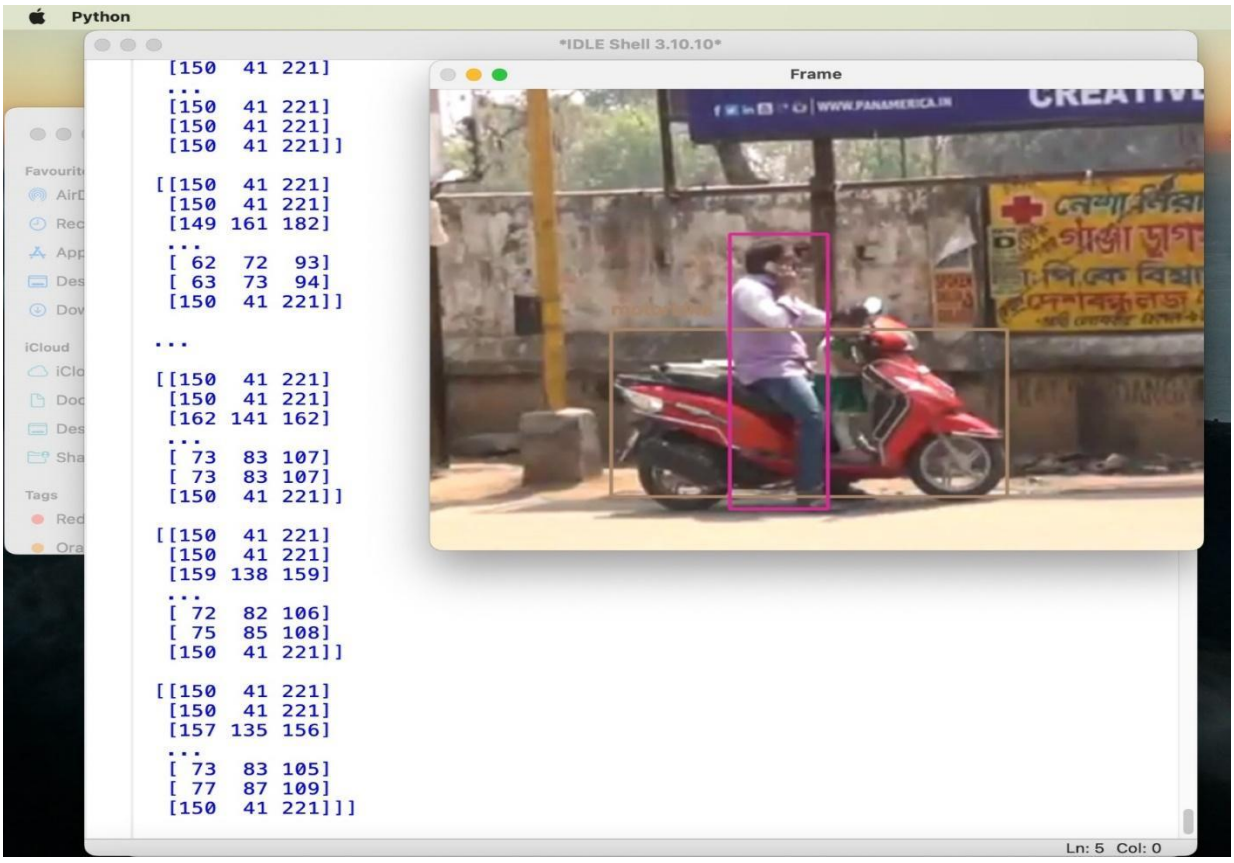
```
plot_result(history)
```

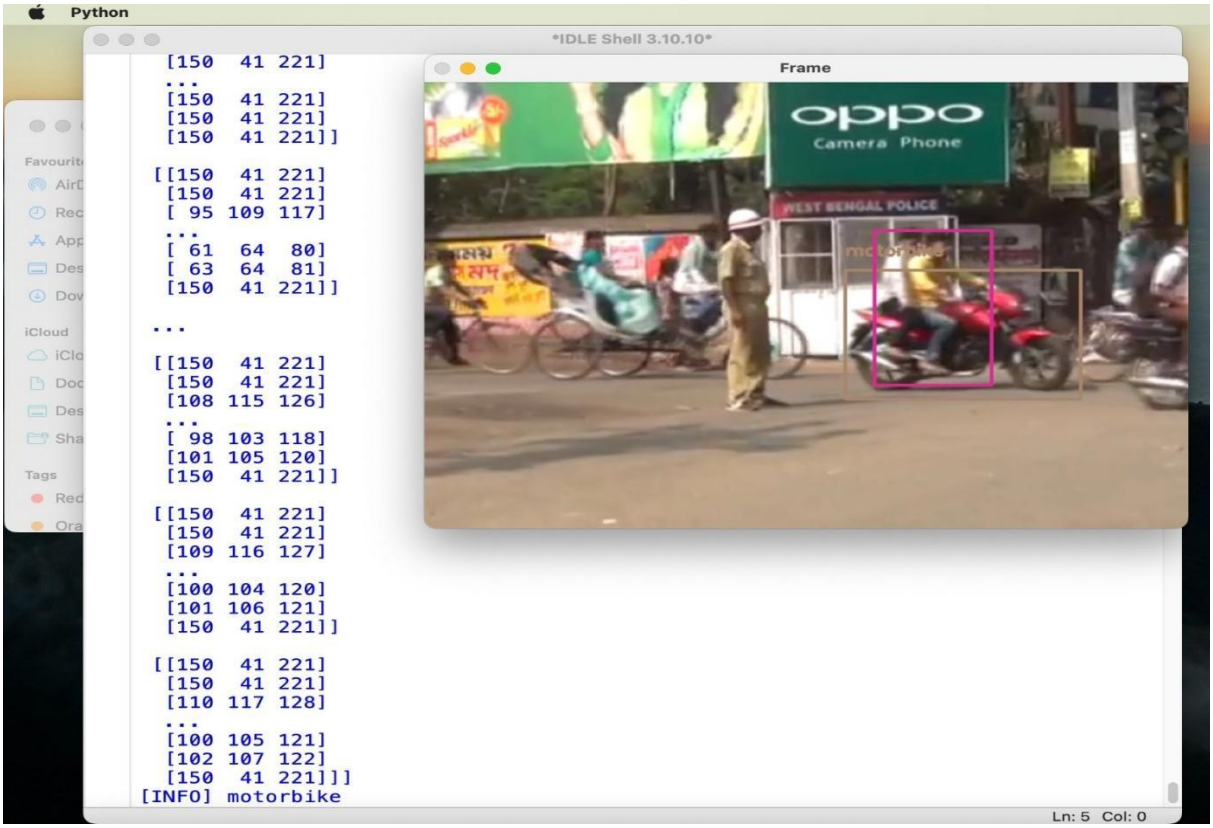
B) SCREENSHOTS

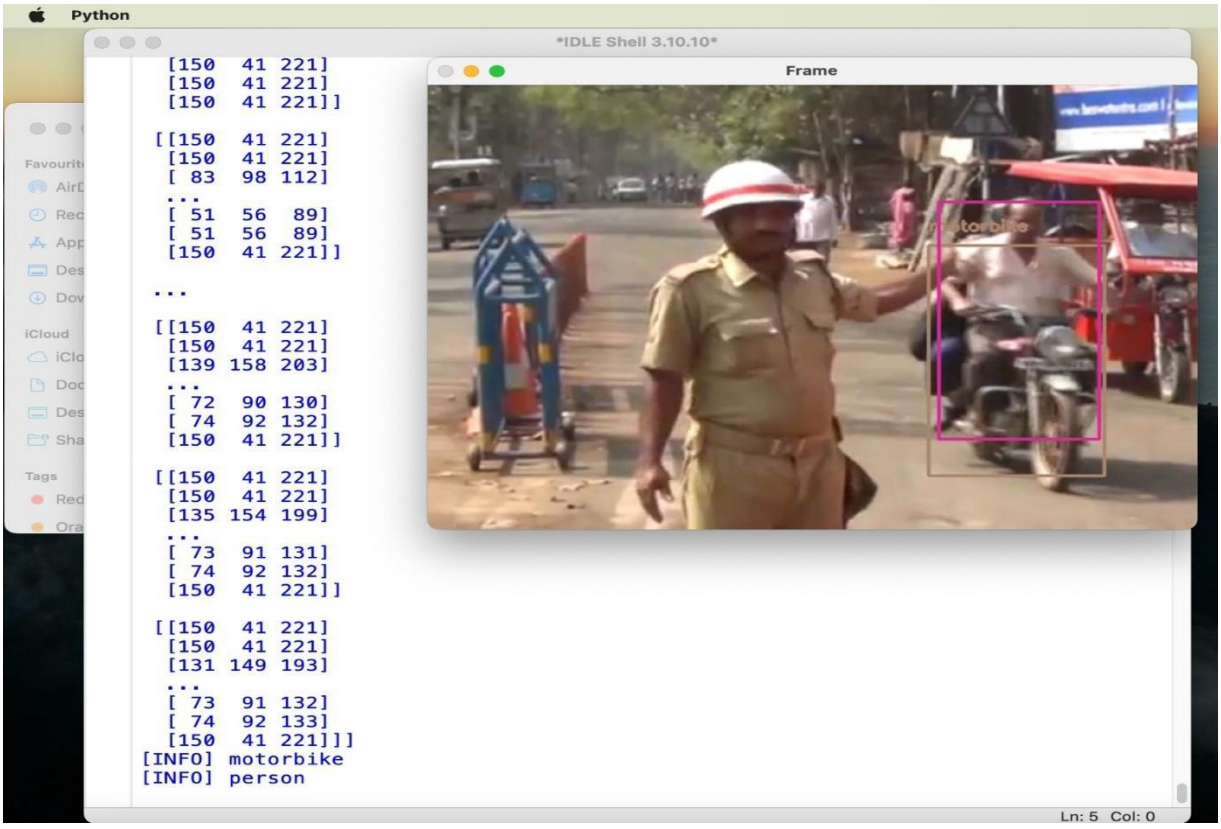


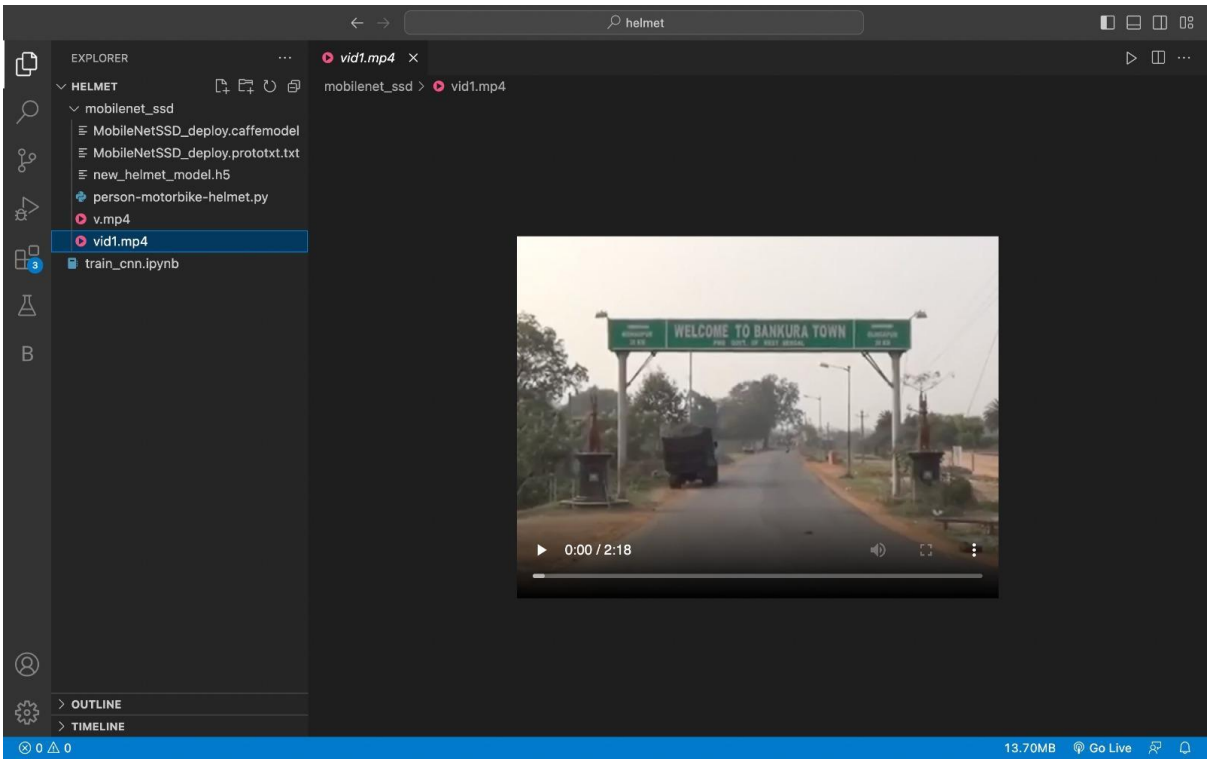

Training and validation loss
Training loss
Validation loss
loss
epochs
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10
12
14

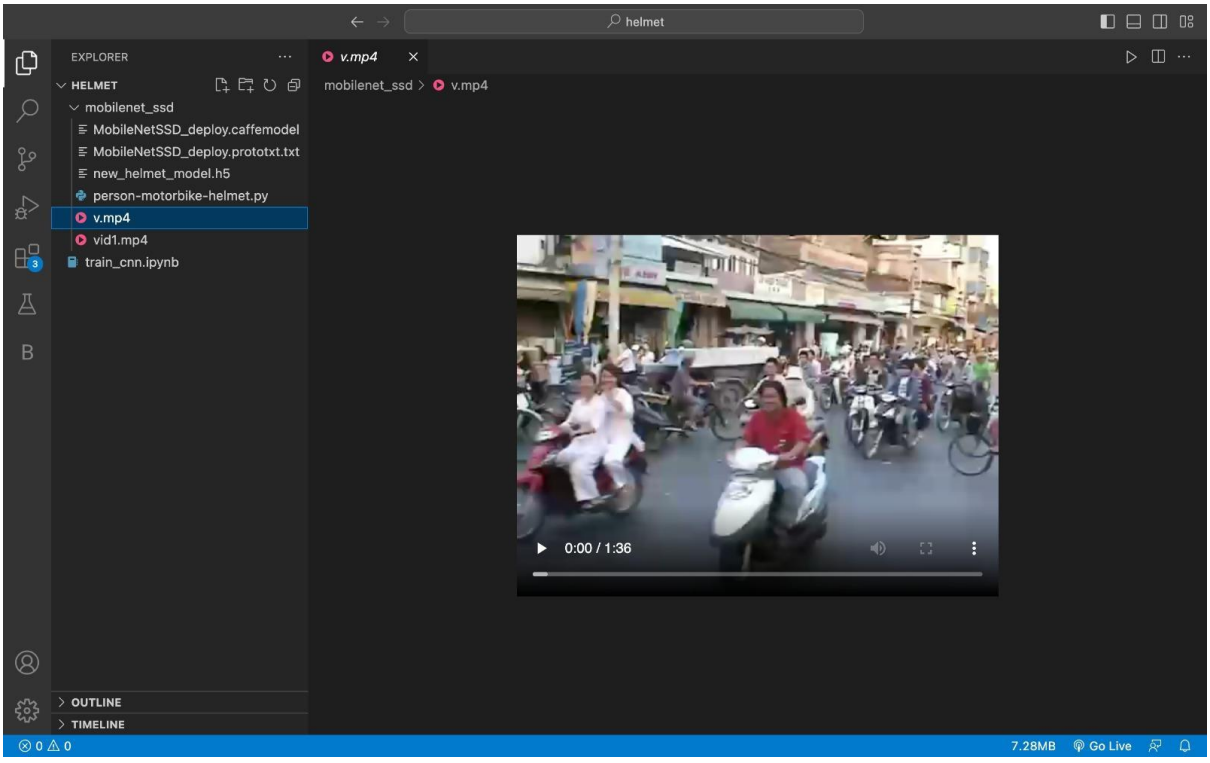

Python
IDLE Shell 3.10.10
Frame
[150 41 221]
[150 41 221]
[150 41 221]]
[[150 41 221]
[150 41 221]
[37 48 63]
[72 91 104]
[62 82 95]
[150 41 221]]
[[150 41 221]
[150 41 221]
[86 107 126]
[51 100 139]
[46 96 139]
[150 41 221]]
[[150 41 221]
[150 41 221]
[85 106 125]
[43 92 131]
[36 86 129]
[150 41 221]]
[[150 41 221]
[150 41 221]
[86 105 122]
[39 83 122]
[32 77 119]
[150 41 221]]]
[INFO] motorbike
[INFO] person
Ln: 5 Col: 0


Python
IDLE Shell 3.10.10
[150 41 221]
...
[150 41 221]
[150 41 221]
[150 41 221]]
[[150 41 221]
[150 41 221]
[149 161 182]
...
[62 72 93]
[63 73 94]
[150 41 221]]
...
[[150 41 221]
[150 41 221]
[162 141 162]
...
[73 83 107]
[73 83 107]
[150 41 221]]
[[150 41 221]
[150 41 221]
[159 138 159]
...
[72 82 106]
[75 85 108]
[150 41 221]]
[[150 41 221]
[150 41 221]
[157 135 156]
...
[73 83 105]
[77 87 109]
[150 41 221]]]
Frame
WWW.PANAMERICA.IN
Ln: 5 Col: 0


Python
IDLE Shell 3.10.10
[150 41 221]
...
[150 41 221]
[150 41 221]
[150 41 221]]
[[150 41 221]
[150 41 221]
[95 109 117]
...
[61 64 80]
[63 64 81]
[150 41 221]]
...
[[150 41 221]
[150 41 221]
[108 115 126]
...
[98 103 118]
[101 105 120]
[150 41 221]]
[[150 41 221]
[150 41 221]
[109 116 127]
...
[100 104 120]
[101 106 121]
[150 41 221]]
[[150 41 221]
[150 41 221]
[110 117 128]
...
[100 105 121]
[102 107 122]
[150 41 221]]]
[INFO] motorbike
Frame
oppo
Camera Phone
WEST BENGAL POLICE
motorbike
Ln: 5 Col: 0


Python
IDLE Shell 3.10.10
[150 41 221]
[150 41 221]
[150 41 221]]
[[150 41 221]
[150 41 221]
[83 98 112]
...
[51 56 89]
[51 56 89]
[150 41 221]]
...
[[150 41 221]
[150 41 221]
[139 158 203]
...
[72 90 130]
[74 92 132]
[150 41 221]]
[[150 41 221]
[150 41 221]
[135 154 199]
...
[73 91 131]
[74 92 132]
[150 41 221]]
[[150 41 221]
[150 41 221]
[131 149 193]
...
[73 91 132]
[74 92 133]
[150 41 221]]]
[INFO] motorbike
[INFO] person
Frame
Ln: 5 Col: 0


helmet
EXPLORER
vid1.mp4
HELMET
mobilenet_ssd
MobileNetSSD_deploy.caffemodel
MobileNetSSD_deploy.prototxt.txt
new_helmet_model.h5
person-motorbike-helmet.py
v.mp4
vid1.mp4
train_cnn.ipynb
mobilenet_ssd > vid1.mp4
WELCOME TO BANKURA TOWN
0:00 / 2:18
OUTLINE
TIMELINE
13.70MB
Go Live


helmet
EXPLORER
v.mp4
HELMET
mobilenet_ssd
MobileNetSSD_deploy.caffemodel
MobileNetSSD_deploy.prototxt.txt
new_helmet_model.h5
person-motorbike-helmet.py
v.mp4
vid1.mp4
train_cnn.ipynb
mobilenet_ssd > v.mp4
0:00 / 1:36
OUTLINE
TIMELINE
7.28MB
Go Live

C) RESEARCH PAPER

HELMET DETECTION USING DEEP-LEARNING

N.Tirumala Rama Linga Raju
Dept.of Computer Science Engineering (B.E.)
Sathyabama Institite Of Science and Technology
Chennai, India
tirumalnadimpalli@gmail.com

R.Abhirama Krishnam Raju
Dept.of Computer Science Engineering (B.E.)
Sathyabama Institute Of Science and Technology
Chennai, India
006abhiram@gmail.com

Abstract*—The current scenario of India's traffic laws poses a number of challenges, each of which may be addressed in a variety of ways. An increasing incidence of motorbike and moped-related fatalities have been linked to the common practice of driving without a headgear in India. The current system relies heavily on CCTV records to keep tabs on traffic offences, forcing cops to manually examine individual frames and, if the rider isn't wearing a helmet, focus in on their numberplate. However, this calls for a significant investment of time and resources, since traffic offences are common and the number of motorcyclists grows daily. What if there were a technology that could detect whether or not a motorcyclist was breaking the law by failing to wear a helmet and, if so, get the vehicle's registration number from the database? This study has been accomplished with recent success using HaaR, HoG, LBP, R-CNN, CNN features, etc. However, the pace, precision, and effectiveness of object identification and categorization are severely constrained by these works. In an effort to automate the process of finding instances of the traffic offense of not donning a headgear and retrieving the numberplate of vehicles, this study develops a Non-Helmet Driver identification method. The core idea is to use three tiers of DL for Image Classification. Individual, motor bike, headgear, and licence plate are all recognised utilising YOLOv2's increasing levels of sophistication. Then, OCR is used to get the licence plate number. Particularly for the extraction of licence plate numbers, all of these methods are governed by a set of predetermined parameters and limits. Since moving images are used as input, the processing time is critical. By combining the aforementioned methods, we have developed a comprehensive system for identifying helmets and extracting licence plate numbers.

***Keywords*—: CNN, LBP, HOG, SVM, OCR**

I. INTRODUCTION

The Globe Health Agency has produced a study titled "The 2016 global study on traffic safety 2018," which states that over 3 million people die and 1.5 million people are wounded each year as a result of traffic collisions throughout the world. It's hard to believe that walkers, bicyclists, and motorists all shoulder the same share of this responsibility.

According to the findings of this paper, an all-encompassing action strategy is needed to prevent needless deaths. When it comes to fatalities caused by automobile accidents, India is, unfortunately, first in the world. Experts have speculated that the rise in traffic fatalities may be attributed to a number of factors, including the increasing prevalence of megacities and the widespread disregard for basic safety precautions like wearing a helmet or a seatbelt while behind the wheel. In the Madrid Statement on Traffic Safety, which India endorsed in 2015, the country pledged to cut the number of fatalities caused by traffic accidents by half by 2020.

In decreasing the number of people killed in road accidents in India, authorities there must admit that issues still exist. In the event of a collision, the rider of a motorcycle or scooter is often removed from the vehicle as a result of the abrupt slowdown that occurs as a result of impact. Even if the neck's movement stops after impact, the body's bulk means it will remain moving until the inside of the skull is struck. It's possible that a severe brain injury might be deadly in certain cases. A helmet is a lifesaver in such a situation. Because a helmet lessens the likelihood that the skull will be actually slowed, it effectively stops the head from moving.

In the event of a collision, the helmet's internal cushion will take the brunt of the force and eventually bring the head to a rest. Moreover, it cushions the blow by dispersing the force across a broader region, protecting the skull from harm. In the event of an accident, a great quality full headgear may prevent or at least lessen the severity of head injuries by acting as an acts as a physical barrier among the rider's head and whatever the rider collides with.

Professionalism brought forth by traffic laws is intended to reduce the potential for accidents and subsequent casualties. However, in practise, there is no observance of these regulations.
This means that approaches that are both effective and practical need to be developed. Manually monitoring traffic using closed-circuit television is now an option. However, in this case, a large number of repetitions are required to reach the goal. As a result, cities with a large population and a high volume of traffic cannot afford to rely on this inefficient, manually technique of helmet identification. With OCR, YOLOv3, & YOLOv2, we offer a technique for comprehensive helmet recognition and licence plate retrieval.

YOLO is a real-time object identification system that can recognise objects in still images and moving videos. There is a newer, more precise version of the YOLO algorithm called YOLO v2. Text in photos and videos may be extracted using OCR. Among the many possible applications is the reading of licence plate numbers from photographs.

Bikers without helmets may be tracked down and their licence plate numbers extracted using a combination of YOLO v2 and optical character recognition. When a picture or video contains motorcyclists, the YOLO v2 algorithm can identify them, and optical character recognition can read the licence plate number. This data might be used to hunt out violators of helmet rules.

II. LITERATURE SURVEY

A. First, we have IIeadgear Recognition.

The discipline of DL has seen a plethora of new proposals in latest days. Thresholding and also the Moc decoder are outlined in [1] as methods for identifying motorbikes in footage. CNN is utilised with handmade characteristics to determine who is and isn't wearing a helmet. When compared to human-created characteristics, CNN clearly performs better. In order to extract a dynamic item from a video, adaptive local removal is used. Motorcycles are now being sorted into several vehicle classes using the CNN technique. Finally, they use CNN to check the heads of motorcyclists to make sure they aren't using protective headgear [3]. To isolate and label the surrounding elements, the GMM is being employed. A quicker Geographical area Convolutional Neural Network is used in this technique to specifically search for motorcycles in the backdrop, confirming the presence of motorcycles. The R-CNN is then used to identify motorcyclists who are or are not using protective headgear. While D-CNN networks are used for headgear identification [1,2,3], the background object in the motorcycle initiation phase is still derived using traditional median filter, which would perform poorly in a high-noise environment.

Neither motorcycle identification is confirmed, although references [4] & [5] imply that the YOLOv3[6] technique is utilised to identify helmeted bikers. We first used the YOLOv3 approach to locate the motorcycle with guy within the picture in [7] &[8], and then they used an evaluation of the overlapping area of grid points containing the two objects to determine who was behind the wheel. The YOLOv3 techniques were subsequently employed to check whether bikers were protecting their heads. Thus, it is unnecessary to precisely detect motorcyclists once transport has been watched due to the high degree of overlap between the two groups. Applying SSD/YOLOv3 techniques to find the motorcycle area, cropping off the head and using a classifier to tell the difference between a helmet and no helmet were all proposed in [9,10,11]. Similarly, if there are more than one rider on the motorcycle, this won't be a useful categorization method. In [12], [13], and [14], for instance, before utilising the Cnn architecture to ascertain whether or not the rider is wearing a helmet, they take into account both the motorbikes and the riders. Single-step, high-fidelity detection approaches are inefficient.

B. Plate Readers

Many authors have looked at the feasibility of using convolutional neural networks (CNNs) for object recognition during the LP detection phase. As Silva&Jung [15] discovered, the Quick algorithm [16] only able to obtain a low identification rate when trying to recognise LPs without prior vehicle detections. Fast-YOLO was used in a cascaded fashion to identify the top perspective of the autos and then its LPs in the indicated locations, yielding good accuracy - recall levels on a sample including Brazilian LPs.

Hsu[17] reworked YOLO and YOLOv2 algorithms for the single goal of LP detection. We believe that LP detection approaches must be quicker than the upgrade of YOLO, which processing 54 FPS with simply a good GPU, since the LP components still need to be recognised. Kurpiel[18] created a nested, overlapping grid using the picture data. Scores were generated for each area using a CNN, and the Lp was determined by looking at the outcomes in neighbouring sub-regions. Real-world data they provided shows that identifying Brazilian LPs in photographs with several car components takes 230 milliseconds on a GT-740MGPU, with just 83percentage reliability.

Using letters culled from regular texts, Li[19] trained a CNN to perform character-based LP detection. Rolling a frame across the whole picture, the networks were used to generate the text salience map. Clusters of characters were used to determine where segments of text existed. Causal research is then used to generate the first sets of criteria and selection boxes. After that, a convolutional neural network (CNN) with LP and non-LP layers was trained to filter out FPs. However, the expense of employing such a collection of methods renders them inappropriate for application areas; on an Electric car Choice of suitable Graphics card, processing a simple photograph takes and over 2 seconds, making them infeasible even if the precision and recall rates achieved were better than just those reached in earlier research.

C. The YOLO Era

First suggested in a 2015 work by JosephR entitled "YOLO: Unified, Context Of physical Identification," the YOLO model is a method for recognising objects with a single pass. Until then, RCNN models had been among the most widely used for object identification. Effective as they were, RCNN models were slow because they needed many steps—including finding the proposed areas for the frame, categorising those regions, and then doing post-processing to enhance the output—to get optimal results. YOLO were created to execute detection in a single phase rather than in several stages, which would shorten the inference time but increase the computation time.

III. METHODOLOGY

Various process stages are discussed in this section.

As illustrated in Figure 1(a) and Figure 1(b), the first step involves collecting images from of the video stream at periodic times. Each shot is saved in a separate directory. The frame count is included in the filename; for instance, frame3-10, frame3-20, etcetera. The numbers 10,20, etc. indicate the frames in the film, or where 3rd video file was inserted. It is evident from the data that many of the frames are unnecessary. In this case, the final image or the very last 2nd frame is selected for given the data on the relative motion of the vehicle and the cameras.

For two different scenarios, the complete process may be broken down into five stages:

Scenario 2: Helmeted motorcyclist

Scenario 2: When the motorcyclist isn't protecting his or her head.

Group of Frames

Fig. 1(a): Intervalically-collected images (Scenario 1)

Fig. 1(b): Intervalically-collected images (Scenario 2)

IV. DETECTION OF MOTORCYCLES AND RIDER

The selected picture is then sent into a YOLOv2 identification and verification model, using "Motorcycle" & "Biker" as the predefined classes. As can be seen in Figure 2 (a) and Figure 2 (b), the final product is an image the with necessary class detection as well as confident of discovery via consecutive frames and significance level (b).

Fig. 2 (a): Frame with 'person' and 'motorcycle' classes detected (Case 1)

Fig. 2 (b): Frame with 'person' and 'motorcycle' classes detected (Case 2)

Fig. 3 (a): Extracted motorcycle and person images (Case 1)

Fig. 3 (b): Extracted motorcycle and person images (Case 2)

In Fig. 3(a) & Fig. 3(b), we see how the capabilities provided by the Imaging AI module are used to selectively select the photos containing the recognised objects, which are then saved as individual files and given names that correspond to the discovered classes and the images numbers. If the extracted item is a motorbike, for instance, the files will be named "motorbike-1," "motorbike-2," "person-1," "person-2," & so on. Information from these collected photos are entered into a lexicon for use in subsequent operations.

V. DETECTION OF HELMETS

The photos of the people are then fed into a helmet classification algorithm after the user combination has been established. False positives were found when evaluating the headgear detection algorithm. As can be seen in Fig. 4, the top quarter of the original human picture was removed by cropping. This prevents false identification & erroneous results from occurring whenever the driver is not really using the headgear but rather carrying it in their hand or leaving it on their motorbike. This headgear recognition technique was applied to the clipped picture.

Fig. 4: Cropped images (Case 1 and Case 2)

Fig. 5: Helmet detection

Fig. 5 depicts the findings. 5. In Fig. 5, we see the helmet's recognition likelihood and its reference image. In Scenario 1, because the rider was wearing a safety helmet, no additional action is required. The absence of a headgear in Scenario 2 eliminates the need for a limiting area to be generated.

VI. NUMBER PLATE RECOGNITION FOR VEHICLES

If the protective headwear has been located, you may skip this procedure. If the headgear cannot be located, the motorbike picture is used as inputs in a licence plate classification algorithm. Bicycle and moped licence plate photos (a total of 832) was gathered as a data for ML purposes. The licence plates in these photographs were then labelled using a labelling tool, thus creating a bounding box around each plate for the model to study. The frame is recorded in a separate.xml file that has the same filename as the picture. Afterwards, the tagged pictures are utilised to create the training set for licence plate detection.

Fig. 6: License plate detection

In a particular source image, the learning algorithm is used to generate a boundary over the licence plate. Bottom-right and top-left subset dimensions, class label, and identification certainty are all included in the corresponding.json file. Afterward, the.json file's barrelling values are utilised to selectively extract the licence plate picture. Figure 6 illustrates an instance where many clusters were found inside the same motorbike picture. The likelihood of detecting is then set at a level of 0.5. The.json file contains data on the structuring element, the one that has a score higher than the criteria is selected.

Fig. 7: License plate extraction and rotation

Fig. 8: License plate image after increasing brightness and rescaling

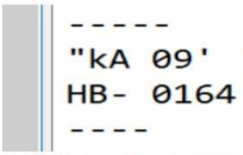


Fig. 9: Output after applying OCR

Before using OCR straight to retrieved licence plate \simage, well before as to be conducted to produce output of better \saccuracy. Hence the picture was rotated. Fig. 7 demonstrates how the \slicense plate picture will appear after it is removed and rotated.

Since cam will stay in the same relative location to the motorbike at all times, the degree from which the retrieved licence plate picture must be flipped will need to be determined experimentally just once; from then on, it will be used in all subsequent instances. The result was determined to be 6°.

To improve OCR's character detection rate, the rotating picture was resampled. Sizing the resized picture was done by selecting a scale ratio, which is the width to body ratio of the resized image to the orginal. Let the initial picture's width & height, w and h, respectively, be h&w, and the resized image's height and width, w' & h'. Let's use r as a proportion to illustrate.

Then the rescaled image size is obtained by:

$$w` = w * r \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

$$h` = h * r \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

wherein r is a dynamic ratio that changes based on which frame is selected throughout frames separation. The results showed that the value in question was within the range of 1.4% and 1.5%. The image's contrast is then cranked up so that the black licence plate numerals stand out more clearly above the white backdrop. The image's Color, Intensity, and Brightness (h,s,v) values were measured. It is well knowledge that a color's v(Value) indicates how light or dark it is. For pixels where the 'v'value is more than a certain maximum, 226 has been designated as the value for that pixel. If the 'v' values of a pixel is less than the minimum, a fixed value has been appended to it. The maximum in this example is 226, while the equilibrium point was 30.

value = 30
limit = 255 – value
if v ≥ limit :
v = 255
else:
v = v + value

Fig. 1. Example of a figure caption. (*figure caption*)

VII. RESULTS AND DICUSSION

In this article, we go through the outcomes for two different scenarios. It's true that they are,

To begin, consider the situation depicted in fig. 5.

Fig. 6 shows what happens in Case 2 when a motorbike rider is not using a helmet..

Table 1. Details of Threshold value with model

Sl. No	Detection model	Number Plate Detection	Threshold value
1	YOLO v2(Without Helmet)	Yes	0.5
2	YOLO v2(With Helmet)	No	0..87

VIII. CONCLUSION

A video clip is used as input in the development of a system for detecting non-helmeted riders. The licence number plate of the motorbike is retrieved and shown if the driver in the video clip is not equipped with a helmet. Motorbike, individual, headgear, & licence plate recognition all employ the YOLO image identification architecture.. If the rider isn't using a helmets, OCR is employed to extract the number from the vehicle's licence plate. Not only are the characters taken out of the frames, but the structure itself is taken out as well. The project's goals have been successfully accomplished.

REFERENCES

[1] J.Chiverton, "Helmet Presence Classification with Motorcycle Detection And Tracking",IET Intelligent Transport Systems,Vol. 6, Issue 3, pp. 259–269, March 2012.

[2] Rattapoom Waranusast, Nannaphat Bundon, Vasan Timtong and Chainarong Tangnoi, "Machine Vision techniques for Motorcycle Safety Helmet Detection", 28th

International Conference on Image and Vision Computing New Zealand, pp 35-40, IVCNZ 2013.

[3] Romuere Silva, Kelson Aires, Thiago Santos, Kalyf Abdala, Rodrigo Veras, Andr´e Soares, "Automatic Detection Of Motorcyclists without Helmet", 2013 XXXIX Latin America Computing Conference (CLEI).IEEE,2013.

[4] Romuere Silva, "Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers", 27th SIBGRAPI Conference on Graphics, Patterns and Images.IEEE, 2014.

[5] Thepnimit Marayatr, Pinit Kumhom, "Motorcyclist's Helmet Wearing Detection Using Image Processing", Advanced Materials Research Vol 931- 932,pp. 588-592,May-2014.

[6] Amir Mukhtar, Tong Boon Tang, "Vision Based Motorcycle Detection using HOG features", IEEE International Conference on Signal and Image Processing Applications (ICSIPA).IEEE, 2015.

[7] Abu H. M. Rubaiyat, Tanjin T. Toma, Masoumeh Kalantari-Khandani, "Automatic Detection of Helmet Uses for Construction Safety", IEEE/WIC/ACM International Conference on Web Intelligence Workshops(WIW).IEEE, 2016.

[8] XINHUA JIANG "A Study of Low-resolution Safety Helmet Image Recognition Combining Statistical Features with Artificial Neural Network".ISSN: 1473-804x

[9] Kunal Dahiya, Dinesh Singh, C. Krishna Mohan, "Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time", International joint conference on neural network(IJCNN). IEEE, 2016.

[10] 1Maharsh Desai, Shubham Khandelwal, Lokneesh Singh, Prof. Shilpa Gite, "Automatic Helmet Detection on Public Roads", International Journal of Engineering Trends and Technology (IJETT), Volume 35 Number 5- May 2016, ISSN: 2231-5381

[11] Pathasu Doungmala, Katanyoo Klubsuwan, "Half and Full Helmet Wearing Detection in Thailand using Haar Like Feature and Circle Hough Transform on Image Processing", IEEE International Conference on Computer and Information Technology.IEEE, 2016.

[12] Shoeb Ahmed Shabbeer, Merin Meleet(2017), "Smart Helmet for Accident Detection and Notification", 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), 2017.

[13] C. Vishnu, Dinesh Singh, C. Krishna Mohan, Sobhan Babu, "Detection of Motorcyclists without Helmet in Videos using Convolutional Neural Network", International Joint Conference on Neural Networks (IJCNN), IEEE, 2017.

[14] Abhijeet S. Talaulikar, Sanjay Sanathanan, Chirag N. Modi "An Enhanced Approach for Detecting Helmet on Motorcyclists Using Image Processing and Machine Learning Techniques", Advanced Computing and Communication Technologies,pp.109-119.

[15] Jie Li, Huanming Liu, Tianzheng Wang ,Min Jiang and Kang Li, "Safety Helmet Wearing Detection Based on Image Processing and Machine Learning", Ninth International Conference on Advanced Computational Intelligence(ICACI), pp.109-119, 2017,.

[16] Jimit Mistry, Aashish K. Misraa, Meenu Agarwal, Ayushi Vyas, Vishal M. Chudasama, Kishor P. Upla, "An Automatic Detection of Helmeted and Non-helmeted Motorcyclist with License Plate Extraction using Convolutional Neural Network", Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA).IEEE, 2017.

[17] Hao Wu, Jinsong Zhao, "An intelligent vision-based approach for helmet identification for work safety", Computers in industry, Vol 100,pp.267-277, Elseivier, 2018.

[18] Wichai Puarungroj, Narong Boonsirisumpun, "Thai License Plate Recognition Based on Deep Learning", Procedia Computer Science, Vol.135, pp.214-221,Elseivier 2018.

[19] Kang Li, Xiaoguang Zhao, Jiang Bian, Min Tan, "Automatic Safety Helmet Wearing Detection", 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER).IEEE, 2017.

[20] Kavyashree Devadiga, Pratik Khanapurkar, Shreya Joshi, Shubhankar Deshpande, Yash Gujarathi, "Real Time Automatic Helmet Detection of Bike Riders", IJIRST – International Journal for Innovative Research in Science and Technology,Volume 4, Issue 11, ISSN: 2349-6010, April 2018.

[21] Yogiraj Kulkarni, Shubhangi Bodkhe, Amit Kamthe, Archana Patil, "Automatic Number Plate Recognition for Motorcyclists Riding Without Helmet", IEEE International Conference on Current Trends toward Converging Technologies(ICCTCT),IEEE, 2018.

[22] Dharma Raj KC, Aphinya Chairat, VasanT imtong, Matthew N. Dailey, Mongkol Ekpanyapong, "Helmet Violation Processing Using Deep Learning", 2018 International Workshop on Advanced Image Technology (IWAIT), IEEE, 2018.

[23] Prajwal M J., Tejas K B., Varshad V., Mahesh Madivalappa Murgod and Shashidhar R "A Review on Helmet Detection by using Image Processing and Convolutional Neural Networks" International Journal of Computer Applications 182(50):52-55, April 2019

.