Project Report

On

Safety Helmet Detection by using Python

Submitted by

Amit Mandal

Subhadeep Saha

Papai Sarkar

Pronay Chandra Mridha

Under the guidance of

Dr. BANDANA BARMAN

Assistant professors

Dept. of Electronics and Communication Engineering

Kalyani Government Engineering College

(Affiliated to Maulana Abul Kalam Azad University of Technology)
Kalyani-741235, Nadia, West Bengal

2022-23

Approval Sheet

This project report entitled "Safety Helmet Detection by using Python" by Amit Mandal, Subhadeep Saha, Pronay Chandra Mridha and Papai Sarkar is approved for the degree of B.Tech. (ECE) submitted to Department of Electronics and Communication Engineering at KALYANI GOVERNMENT ENGINEERING COLLEGE, KALYANI under MAKAUT.

Examiners

DECLARATION

We certify that

1. The work contained in the project is original and has been done by us under the general supervision of our supervisor.

2. The work has not been submitted to any other institute for any degree or diploma.

3. We have followed the guidelines provided by the institute in writing of the project.

4. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute and University.

Amit Mandal

Roll No. - 10200321068
Reg No. - 211020100320006

Subhadeep Saha

Roll No. – 10200320036
Reg No. – 201020100310007

Pronay Chandra Mridha

Roll No. – 10200321055

Reg No. – 211020100320007

Papai Sarkar

Roll No. – 10200321070

Reg No. – 2011020100320011

TABLE OF CONTENT

ACKNOWLEDGEMENT

It is not possible to complete our project work and also to prepare a report on this without the assistance & encouragement of people working with us. This one is certainly no exception. On the very outset of this report, We would like to extend our sincere & heartfelt gratitude towards all the personages who have helped in this endeavour. Without their active guidance, help, cooperation & encouragement, We could not have made hardware in the project. First and foremost, we would like to express our sincere gratitude to our supervisor**, Dr. Bandana Barman**.

We are indebted to **Dr. Sourabh kumar Das**, honourable Principle of Kalyani Government Engineering College and our honourable Head of Department **Dr. Angsuman Sarkar** for providing all logistic support and guidance.

We would like to thank all my friends and especially our classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. We have enjoyed their companionship so much during our stay at **KGEC.**

Thanking You

Phone: - 25826680 (PBX)
Fax :- 25821309

Kalyani, 741 235 ,
Nadia, West Bengal, India

Kalyani Government Engineering College

(Govt. of West Bengal)

CERTIFICATE

This is to certify that Amit Mandal, Subhadeep Saha, Pronay Chandra Mridha and Papai Sarkar have successfully completed project work entitled “**Safety Helmet Detection By Using Python**” is being presented. In the partial fulfilment of the requirements for the award of the Bachelor of Technology in Electronics and Communication Engineering and submitted to the Department of Electronics and Communication Engineering of Kalyani Government Engineering College is an authentic record of our own work carried out during our B.Tech course Under the supervision of Dr. Bandana Barman, Assistant Professor of Electronics and Communication Engineering Department.

DR. ANGSUMAN SARKAR
PROFESSOR AND HEAD
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DR. BANDANA BARMAN
PROJECT GUIDE
ASSISTANT PROFESSOR

ABSTRACT

This report explores the use of Python and yolov5 for high-density crowd counting, a task with applications in public safety, event management, and urban planning. Traditional counting methods often struggle in complex crowd scenes, making it challenging to obtain accurate crowd counts. Python, with its versatile libraries and frameworks, offers an effective solution to address this problem.

The report covers data collection, preprocessing, deep learning approaches, density estimation, model training and evaluation, and identification of high-density crowds. Python libraries like OpenCV, NumPy, TensorFlow, yoloV5, Thonny IDE etc are employed for implementation.

Data preprocessing removes noise, resizes images, and normalizes pixel values. Deep learning models, particularly Convolutional Neural Networks (CNNs), learn complex patterns and features from crowd images. Density estimation generates density maps that assign higher values to densely populated areas. Model training and evaluation employ appropriate datasets and evaluation metrics like Mean Absolute Error (MAE) and Mean Squared Error (MSE).

To identify high-density crowds, a density threshold is set, and the number of crowds exceeding this threshold is counted. Python and OpenCV facilitate this process effectively. The report concludes by highlighting the benefits of Python and OpenCV in high-density crowd counting and suggests future directions, such as advanced deep learning architectures and addressing challenges like occlusions and varying lighting conditions.

Keywords : High-density crowd counting, Python, yolov5, deep learning, Convolutional Neural Networks (CNNs), density estimation, data preprocessing, model training, model evaluation, crowd management, urban planning, public safety, event management, data collection, density threshold, occlusions, lighting conditions

1. INTRODUCTION

1.1 Introduction to Crowd analysis

The construction industry is one of the most prone to safety accidents. Therefore, it is of great practical significance to study safety guarantees in this field. Over the past 20 years, it has experienced a decline in accident rates [1].

Head injuries can easily lead to a disability [2]. Reducing head injuries is the primary problem to ensure personnel security in the industry, and safety helmets are widely used to do so. The impact resistance of safety helmets can disperse the impact of rocks. Thus, many industrial regulations require workers to wear safety helmets during working. However, workers do not wear safety helmets as required due to the lack of safety awareness. Because of these reasons, safety accidents are frequent. According to relevant research statistics, 47.3% of people with head injuries in construction site accidents did not wear safety helmets [3]. Thus, it is very important to strengthen the supervision and management of workers. At present, the management of safety helmet wearing in most construction sites still requires manual monitoring. However, the efficiency of manual monitoring is very low due to the large working area and large flow of personnel. With the development of science and technology, video surveillance has become more and more popular. It is a vital part of safety helmet detection. Hence, the optimization of monitoring systems is widely studied [4,5]. Traditional video surveillance is mainly used for continuous monitoring. However, the final judgement still relies on humans' decisions, and the degree of automation is not enough. Intelligence algorithms are a method to enhance automation. They are widely used in image processing, prediction, robotics and so on [6,7,8,9,10,11]. Currently, evolutionary algorithms and deep learning are two important intelligent systems [12,13,14]. Among them, deep learning is widely used in image processing because of its strong learning ability [15,16,17], which can be combined with video surveillance to solve the problems of traditional methods [18].

Slow transmission processes are caused by large image data collected by cameras. Therefore, the obtained original images need to be compressed before transmitting them so that the process can be accelerated. After compressed image data is transferred to the terminal main control unit, they are input into a YOLOv5 target detection network. Since the image compression usually reduces the image resolution, it can be restored by super-resolution (SR) reconstruction.

To solve those problems, a safety helmet detection model is proposed, which is driven by an SR reconstruction network based on YOLOv5. To enhance the learning ability of the network, the double residual channel structure [22] is applied to an SR reconstruction network in the proposed model.

Previous Researches On Safety Helmet Detection (Literature Review):

1) Safety Helmet Detection Based on YOLOv5 by Fangbo Zhou, Huailin Zhao, Zhen (2021) ,this research work proposes a safety helmet detection method based on YOLOv5 and annotates the 6045 collected data sets to establish a digital safety helmet monitoring system and shows the effectiveness of helmet detection based YOLov5.

2) Safety Helmet Wearing Detection Based on Jetson Nano and Improved YOLOv5 by Zaihui Deng,Chong Yao,and Qiyu Yin(2023), This study introduces an improved safety helmet-wearing detection model named YOLOv5-SN, aiming to address the shortcomings of the existing YOLOv5 models, including a large number of model parameters, slow reasoning speed, and redundant network structure.

1.2 Inspiration:

Undertaking a project on Safety helmet detection by using Python, offers a compelling opportunity to address the practical need for accurate crowd estimation in diverse domains. Crowd counting plays a crucial role in urban planning, public safety, transportation management, and event planning. Python's simplicity, extensive libraries, and wide community support make it an ideal choice for implementing crowd counting algorithms. Yolov5 as a comprehensive computer vision library, provides the necessary tools and functions for efficient crowd analysis and counting. Additionally, the availability of publicly accessible crowd counting datasets allows for benchmarking and comparison with existing approaches. By developing accurate crowd counting models, this project can contribute to improved crowd management strategies, enhanced public safety measures, and optimized resource allocation. The combination of Python's versatility and OpenCV's robust functionality provides a solid foundation for undertaking a crowd counting project with practical implications.

1.3 Hardware and Software used:

Software requirement specification

The software used in the development of the project are as follows:

- Windows 10/11
- Python 3.9 and 3.10
- Google Colab
- Thonny Python IDE
- Yolov5

The hardware used in the development of the project are as follows:

- 8 GB RAM
- Core i5 CPU
- 64 bit 0S

2. STUDY AND RELATED WORKS

2.1. Target Detection

At present, research on target detection algorithms include two-stage and one- stage algorithms. Two-stage detection algorithms generate a series of candidate boxes as samples and then classify samples through a convolutional neural network. This kind of detection method has higher task accuracy but slower speed. Girshick et al. [25] proposed the region convolutional neural network, fast regions with CNN [26] and faster regions with CNN [27] algorithms. A onestage detection algorithm directly regresses the category probability and position coordinate values of objects through a backbone network without using a region proposal network (RPN). This kind of detection method sacrifices detection precision but improves detection speed. In 2016, Liu et al. [28] introduced the multiscaledetection method and proposed the SSD (single shot multibox detection) detection algorithm, which improved the detection accuracy. Redmon et al. [29,30,31] proposed YOLOv1, YOLOv2 and YOLOv3. The YOLOv1 network model abstracted the target detection task into a regression problem for the first time, which greatly sped up the target recognition speed. The YOLOv2 network model introduced a new basic model named darknet-19 based on YOLOv1 to realize end-to-end training. Compared with YOLOv1, the YOLOv2 network model realizes more accurate, faster and more target categories. YOLOv3 introduced the feature pyramid network (FPN) algorithm, promoted the new basic model darknet-53 and integrated three feature layers of different sizes for detection tasks. It improved detection speed and accuracy, especially the detection performance of small targets. Bochkovskiy et al. [32] proposed YOLOv4. This detection network takes CSP darknet-53 as the backbone network and uses the PANET path aggregation algorithm. As a result, it improved the detection accuracy of the model. In 2020, Jocher et al. [33] proposed YOLOv5. This network model adds a focus structure to the backbone network of YOLOv4 to obtain a balance between detection speed and accuracy. Carion et al. [23] proposed DETR for end-to-end object detection and brought transformers into the object detection fields. Recently, Wang et al. [34] proposed YOLOv7, which has achieved better accuracy and speed than YOLOv5.

2.2. SR Reconstruction

The image SR reconstruction algorithm is used to recover high-resolution images from one or more low-resolution images. Dong et al. [19] proposed SR Convolution Neural Networks (SRCNNs). SRCNNs effectively improve the results of image SR reconstruction compared with traditional image SR algorithms. However, the network is relatively simple, and the convergence speed is slow during the execution of the algorithm. In subsequent research, researchers added a residual structure to the convolution network to effectively solve the above problems. Kim et al. [23] proposed the VDSR network and increased the number of layers of the CNN to 20. The residual structure and CNN are embedded into image SR reconstruction, and the image reconstruction result is improved. Li et al. [20] proposed a multiscale residual network (MSRN). This network includes image multiscale features in the residual structure to further improve the image reconstruction result. Zhang et al. [35] proposed the residual channel attention network SRCAN. This network applies a channel attention mechanism to the image SR problem .

Difficulty in handling complex scenes with varying densities and limited scalability to large crowds.

While the mentioned research papers contribute significantly to the field of safety helmet detection using YOLOv5, it's common for studies to encounter gaps, difficulties, or areas that warrant further exploration. Here are some potential gaps or challenges that might be found in these research papers:

1. **Handling Highly Occluded Helmets** : Occlusions, where safety helmets are partially or fully hidden, pose a considerable challenge in real-world scenarios.
2. **Generalization Across Diverse Construction Environments** : Construction sites vary widely in terms of lighting conditions, backgrounds, and types of helmets used. Ensuring the model's generalization across diverse environments is crucial.

3. **Ethical and Privacy Concerns :** safety helmet detection systems become more prevalent, there is a growing need to address ethical considerations and privacy implications

4. **Explainability and Transparency :** The interpretability and transparency of YOLOv5-based safety helmet detection systems may not be well-explored in some studies.

Overall, these papers indicate that while significant progress has been made in crowd counting research, there are still several challenges to be addressed. These include handling occlusions and variations in crowd density, particularly in highly congested areas, addressing scalability issues to handle high-density crowds, and improving the robustness of methods to handle complex scenes. Additionally, the limited availability of large-scale datasets and challenges in generating realistic synthetic data remain significant gaps in the field.

2.1 Data Collections:

Data collection and preprocessing are essential steps in crowd counting research to ensure the availability of suitable datasets and to prepare the data for accurate analysis. The process involves capturing crowd images or videos, annotating the data with ground truth crowd counts, and performing preprocessing techniques such as noise removal, image enhancement, and density map generation. These steps contribute to the quality and reliability of the crowd counting models.

Data collection is a crucial step in developing a crowd counting system. It involves gathering a dataset of images or videos with annotations indicating the number of individuals in each frame. Here's an overview of the data collection process for crowd counting:

Define the Scope: Determine the specific context or scenario in which you want to perform crowd counting. It could be public spaces, transportation hubs, events, or any other relevant setting.

Identify Data Sources: Identify potential sources from where you can collect the data. This could include publicly available datasets, online video repositories, surveillance footage, or capturing your own data using cameras.d

Consent and Legal Considerations: Ensure that you comply with legal and ethical requirements for data collection, especially if you are capturing data from public spaces or using third-party sources. Obtain necessary permissions and consents, and respect privacy regulations.

Dataset Size and Diversity: Determine the desired size and diversity of your dataset. The dataset should cover a range of crowd sizes, variations in lighting conditions, different camera angles, and diverse environments to ensure the robustness and generalization of your model.

Annotation Process: Annotate the dataset by manually labeling each image or frame with the corresponding crowd count. This can be a time-consuming task, especially for large datasets. You can use specialized annotation tools or crowdsource the annotation process if feasible.

Data Preprocessing: Preprocess the collected data as necessary. This may involve resizing the images or videos to a consistent resolution, normalizing pixel values, and applying other transformations to enhance the quality and uniformity of the data.

Data Split: Divide the dataset into training, validation, and test sets. The training set is used to train the crowd counting model, the validation set helps monitor the model's performance during training, and the test set is used to evaluate the final model's accuracy and generalization.

Data Augmentation: Consider applying data augmentation techniques to increase the dataset's diversity and improve the model's ability to handle variations in real-world scenarios. Common data augmentation techniques include random cropping, rotation, flipping, and adjusting brightness/contrast.

Remember to document the details of your dataset, including the source, annotation process, and any specific considerations or limitations associated with the data. This documentation will help ensure transparency and reproducibility in your crowd counting research or application.

3. PLAN OF WORK

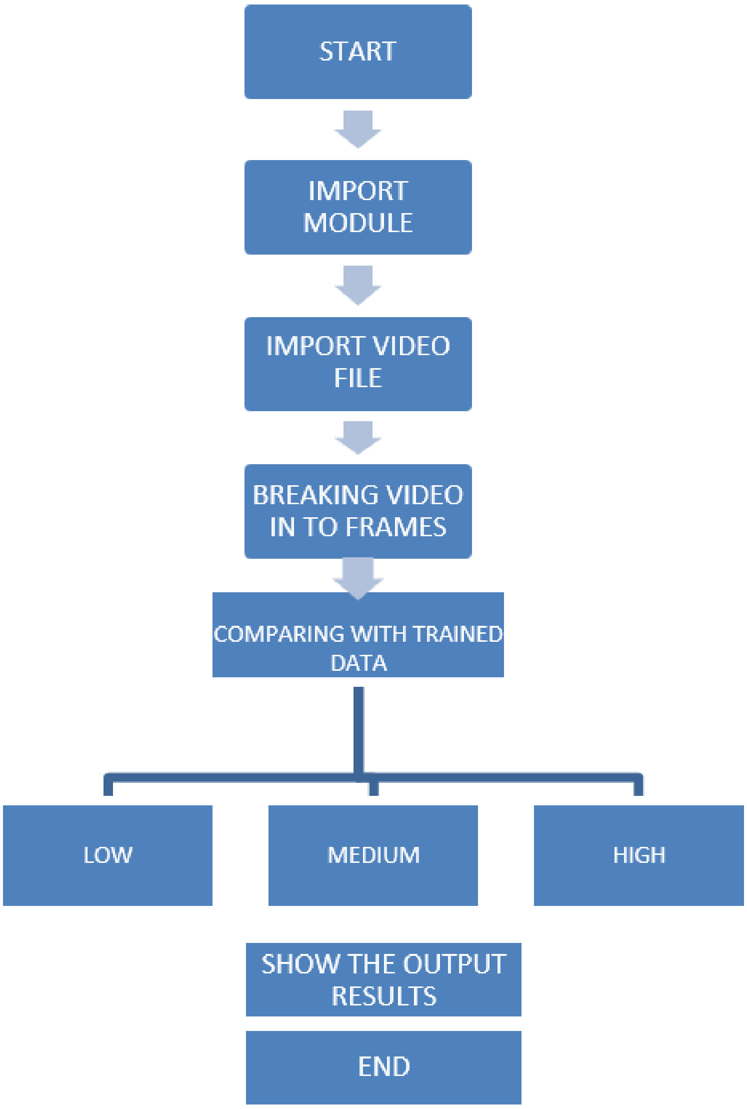
3.1 Algorithm:

Certainly! Here's a basic algorithm for crowd counting:

1) step 1: Install Thonny Python IDE, yolov5 ,open CV module, NumPy module 2) step 2: Download the Images and move our main file.

3) step 3: Created the project video and train all images. we create the 'Rectangle' box to measure the size of our helmets.
4) step 4: Then create a new folder for storing our data and move all pictures here and make a zip file of it and upload in the g-drive.
5) step 5: Opens google collab for collaboration our code with our image file.
6) step 6: Download our custom model and store in the main folder.
7) step 7: Open Thonny Python IDE & give the code for the model, add the path, capture the video file.
8) step 8: Now our basic code is ready , we call the custom model and framing our models.
9) step 9: Finally detecting Safety Helmets in the Images and the project is done.

3.2 : Flow Chart of algorithm



4. METHODOLOGY

A convolutional neural network (CNN) is a multilayer neural network. It is a deep learning method designed for image recognition and classification tasks. It can solve the problems of too many parameters and difficult training of the deep neural networks and can get better classification effects. ,e structure of most CNNs consists of input layerconvolutional layer (Conv layer)-activation function-pooling layer-fully connected layer (FC layer). ,e main characteristics of CNNs are local connectivity and parameter sharing in order to reduce the number of parameters and increase the efficiency of detection. ,e 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. ,e Conv layers can extract and reinforce the object features. ,e pooling layers can filter multiple features, remove the unimportant features, and compress the features. ,e activation layers use nonlinear activation functions to enhance the expression ability of the neural network models and can solve the nonlinear problems effectively. ,e FC layers combine the data features of objects and output the feature values. By this means the CNNs can transfer the original input images from the original pixel values to the final classification confidence layer by layer. In order to better extract the object features and classify the objects more precisely, Hinton et al. [19] proposed the concept of deep learning which is to learn object features from vast amounts of data using deep neural networks and then classify new objects according to the learned features. Deep learning algorithm based on convolutional neural networks has achieved great results in object detection, image recognition, and image segmentation.

Although the SSD algorithm is not capable of the highest accuracy, the detection speed of the SSD algorithm is much faster and comparable to the YOLO algorithm and the precision can be higher than that of the YOLO algorithm when the sizes of the input images are smaller. While the Faster R-CNN algorithm tends to lead to more accurate models, it is much slower and requires at least 100 ms per image [26]. ,erefore, considering the real-time detection requirements, the SSD algorithm is chosen in the research. In order to reduce greatly the calculation amount and model thickness, the MobileNet [27] model is added. ,erefore, in the paper, the SSD-MobileNet model is selected to detect safety helmets worn by the workers. ,e SSD algorithm is based on a feed-forward convolutional network to produce bounding boxes of fixed sizes and generate scores for the object class examples in the boxes. A nonmaximum suppression method is used to predict the final results.

5. **Database**

The data required for the experiment were collected by the author. Since there are few object detection applications of safety helmets using deep learning and there is no off-theshelf safety helmets dataset available, part of the experimental data was collected using web crawler technology, making full use of network 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. ,ere 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. ,before, images were also collected manually besides web crawling. 3500 images were collected in total. ,e images that did not contain safety helmets, duplicate images, and the images that are not in the RGB three-channel format were eliminated and 3261 images were left, forming the safety helmet detection dataset. Some images in the dataset are shown in Figure 4. To increase the detection effect of the safety helmet detection model in detecting helmets with different directions and brightness in images, the image dataset was preprocessed such as rotation, cutting, and zooming. ,en, 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. ,e numbers of the three sets are 2769, 339, and 153, respectively. All the images that contained safety helmets were manually prelabeled, using the open-source tool LabelImage (available in https://github.com/ tzutalin/labelImg). In each labeled image, the sizes and the locations of the object are recorded .

6. RESULT & DISCUSSION

- **Real-time Crowd Counting:**

Real-time crowd counting refers to the process of estimating crowd counts in a live or streaming video feed with minimal delay. It is a challenging task that requires efficient algorithms and architectures capable of processing video frames in real-time while maintaining accurate counting performance.

Real-time crowd counting has significant practical applications in various domains, including crowd management, public safety, transportation, and event planning. It enables timely decision- making, resource allocation, and crowd monitoring, allowing for proactive measures to be taken in response to crowd dynamics.

To achieve real-time crowd counting, several factors need to be considered:

Efficient Architecture: Real-time crowd counting systems require computationally efficient architectures. Models with low inference times, such as lightweight convolutional neural networks (CNNs) or efficient network architectures like MobileNet or ShuffleNet, are commonly used to achieve real-time performance.

Hardware Acceleration: Utilizing hardware acceleration techniques, such as Graphics Processing Units (GPUs), Tensor Processing Units (TPUs), or dedicated hardware like Field-Programmable Gate Arrays (FPGAs) or Neural Processing Units (NPUs), can significantly speed up the inference process and enable real-time crowd counting on resource-constrained devices.

Parallel Processing: Leveraging parallel processing techniques, such as model parallelism or data parallelism, allows for simultaneous processing of multiple video frames, reducing the overall processing time and enabling real-time performance.

Optimization Techniques: Various optimization techniques, including network pruning, quantization, or model compression, can be applied to reduce the computational and memory requirements of the crowd counting models, enabling real-time processing on low-power devices.

Streaming Data Handling: Real-time crowd counting involves handling streaming video data. Efficient data buffering, frame skipping, or adaptive sampling techniques can be employed to balance processing speed and counting accuracy while ensuring a continuous flow of frames for analysis.

Robustness to Environmental Factors: Real-time crowd counting systems should be robust to environmental factors such as changes in lighting conditions, camera motion, or variations in crowd density. Robust feature extraction, normalization techniques, and adaptive algorithms are essential to maintain accurate counting performance in real-world scenarios.

Latency Considerations: In real-time applications, latency, or the delay between video frame capture and counting output, is crucial. Minimizing the latency is important to ensure timely and responsive crowd management decisions.

Ongoing research focuses on developing efficient and lightweight architectures, exploring hardware acceleration techniques, and optimizing algorithms to achieve real-time crowd counting. Additionally, the integration of real-time crowd counting with other technologies such as object tracking, anomaly detection, or behavior analysis further enhances the capabilities of real-time crowd monitoring systems.

Overall, real-time crowd counting enables proactive decision-making and enhances situational awareness in crowd management scenarios. It requires a combination of efficient algorithms, optimized architectures, and hardware acceleration techniques to achieve accurate counting results with minimal delay.

7. Discussion and Future Study

Problems found :

- In mode 3, which is test video that is detecting many people at once, it fails to detect helmets correctly when it overlaps.
- There are some frames that it detects as a helmet even though actual object is hat.
- It fails to detect helmet when person is not wearing it. It only detects as helmet when person is wearing a helmet.
- We have failed to find appropriate video for mode 3.

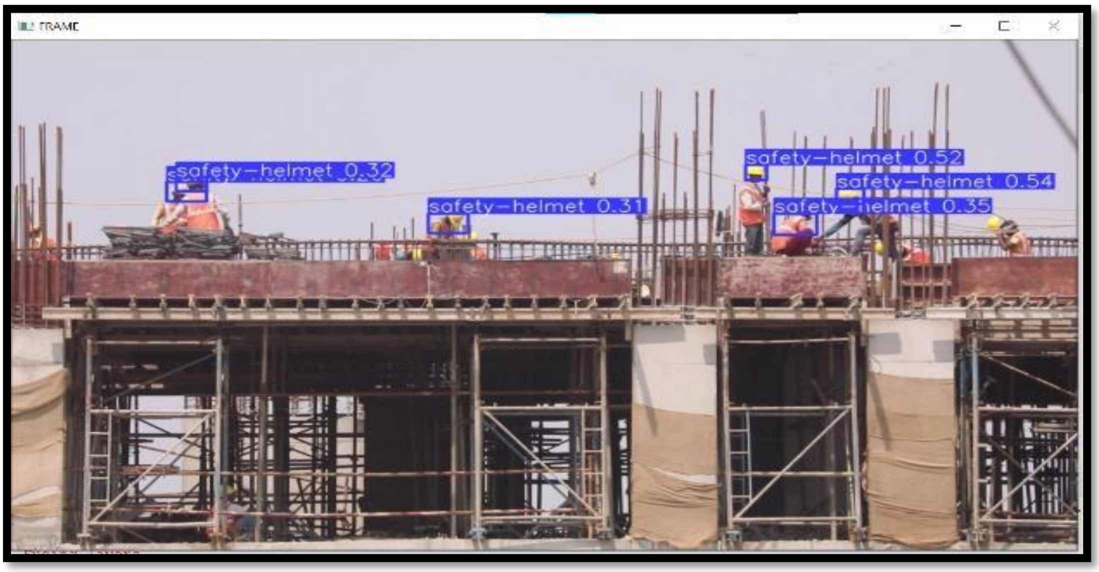
Future Work :

- In order to distinguish helmets from hats and detect helmet without it being on the head, we would like to try training a new model through new datasets.
- Find or take a video that fit mode 3 and evaluate result.
- Online Learning and Adaptation: Crowd counting models that can adapt and learn from evolving crowd dynamics in real-time are essential for dynamic crowd scenarios. Future research should investigate online learning and adaptation techniques to improve the adaptability and robustness of crowd counting models.
- Privacy and Ethical Considerations: With the increasing use of surveillance and crowd monitoring systems, privacy and ethical considerations are of utmost importance. Future directions should address privacy concerns by developing privacy-preserving crowd counting methods and ensuring ethical deployment of crowd counting systems.
- Benchmark Datasets and Evaluation Protocols: Developing standardized benchmark datasets with diverse crowd scenarios and comprehensive evaluation protocols is crucial for fair comparisons and advancements in crowd counting research. Future efforts should focus on creating benchmark datasets that capture various crowd dynamics and provide standardized evaluation protocols.
- Real-Time Crowd Behavior Analysis: Integrating crowd counting with crowd behavior analysis can provide deeper insights into crowd dynamics and facilitate proactive crowd management. Future research should explore real-time crowd behavior analysis techniques, including anomaly detection, crowd flow analysis, and crowd event recognition.

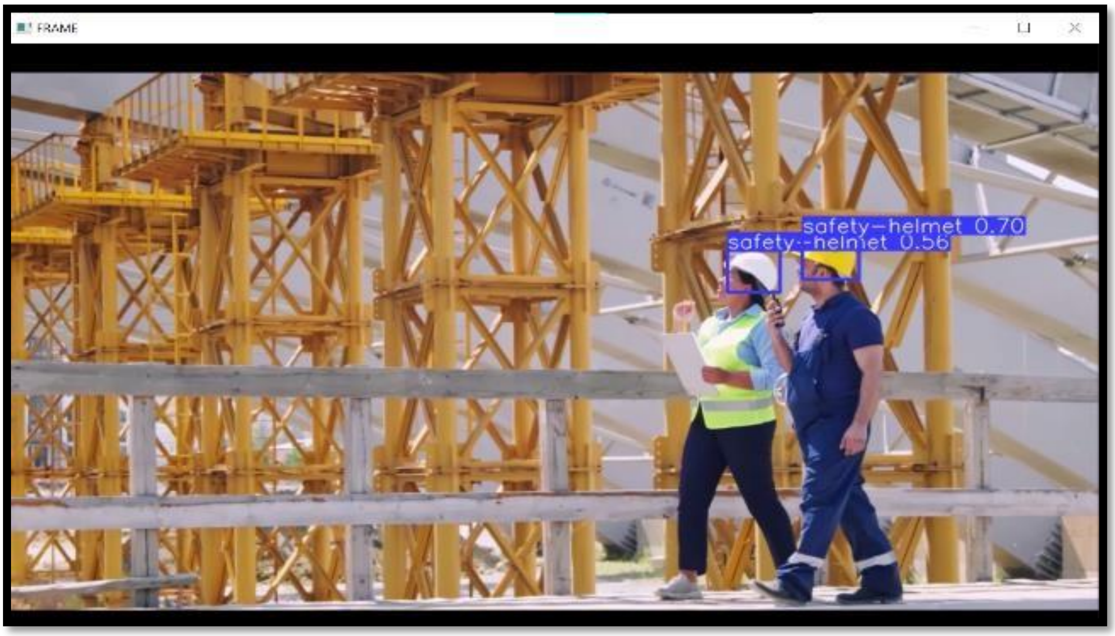
- Human-AI Collaboration: Investigating methods for effective collaboration between humans and AI systems in crowd counting can lead to more efficient and accurate counting results. Future directions should focus on developing interactive interfaces, visualization tools, and decision support systems that facilitate human-AI collaboration in crowd monitoring and management.
- By addressing these future directions, researchers and practitioners can advance the field of crowd counting, develop more accurate and robust models, and contribute to the effective management of crowd-related challenges in various domains.

- **Input-Output**

Example 1 :



Example 2:



8. CONCLUSION:

Conclusions As worker safety is a major concern on construction sites, this study considered helmet detection as a computer vision problem, and proposed a deep learning-based solution. Existing studies have struggled in detecting objects from low-light images and smaller objects (due to the larger distance between the camera and workers). Therefore, a YOLOv5x-based architecture for automatic detection of safety helmets on construction sites was proposed to ensure worker safety. This study used different versions of YOLO architecture, YOLOv3, YOLOv4, and YOLOv5x, to detect safety helmets due to their proven accuracy in object detection tasks.

Among them, YOLOv5x achieved the best mAP (92.44%) in detecting smaller objects and objects in low-light images, thereby showing its efficacy in safety helmet detection. Despite the significant outcomes achieved by YOLOv5x-based architecture, it also possesses several limitations. The proposed deep learning model struggled to perform in some scenarios (e.g., with an obstacle in front of helmets, and objects identical to helmets). Training the model with more images, including the above-mentioned scenarios, could potentially increase the model's efficacy. Moreover, in the future, more safety tools could be added for detection, such as vests, gloves, and glasses, to ensure greater safety for workers.

In conclusion, this project has successfully realized the development and implementation of a safety helmet detection system, contributing to the broader goal of improving construction site safety. The automated monitoring system, empowered by YOLOv5 and Python, stands as a testament to the intersection of technology and safety. By leveraging these tools, we have created a reliable solution that has the potential to mitigate risks and enhance the well-being of construction site workers.

9. REFERENCE

1. "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network" by S. Zhang et al.

2. "Cross-Scene Crowd Counting via Deep Convolutional Neural Networks" by Z. Cao et al.

3. "Switching Convolutional Neural Network for Crowd Counting" by Y. Yuan et al.

4. "CNN-Based Cascaded Multi-task Learning of High-Level Prior and Density Estimation for Crowd Counting" by

   Y. Li et al.

5. "Adaptive Density Map Generation for Crowd Counting" by V. Lempitsky and A. Zisserman.

6. "Convolutional Neural Networks for Crowd Counting" by S. Sindagi and V. M. Patel.

7. "Leveraging Unlabeled Data for Crowd Counting by Learning to Rank" by Y. Zhang et al.

8. "Crowd Counting Using Deep Recurrent Spatial-Aware Network" by X. Shi et al.

9. "Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds" by R. S. Zanotto et al.

10. "Cross-Scene Crowd Counting via Deeply Learned Descriptors" by Z. Cao et al.

11. "Generating Semantically Segmented Density Maps for Crowd Counting" by K. Chen et al.

12. "Crowd Counting via Adversarial Cross-Scale Consistency Pursuit" by Y. Bai et al.

13. "Hydra-CNN: Recursive Learning for Single-Image Crowd Counting" by J. Lian et al.

14. "Weakly Supervised Crowd Counting via Adaptive Density Estimation" by Y. Zhao et al.

15. "Context-Aware Crowd Counting" by C. Zhang et al.

16. Wu, Y.; Meng, Z.; Palaiahnakote, S.; Lu, T. Compressing YOLO network by compressive sensing. Proceedings on the 4th Asian Conference on Pattern Recognition, ACPR2017, Nanjing, China; pp. 19–24. [CrossRef]

17. Pan, H.; Jiang, J.; Chen, G. TDFSSD: Top-Down Feature Fusion Single Shot MultiBox Detector. Signal Process. Image Commun. 2020, 89, 115987. [CrossRef]

18. Li, Y.; Wei, H.; Han, Z.; Huang, J.; Wang, W. Deep Learning-Based Safety Helmet Detection in Engineering Management Based on Convolutional Neural Networks. Adv. Civ. Eng. 2020, 2020, 1–10. [CrossRef]

19. Barro-Torres, S.; Fernández-Caramés, T.M.; Pérez-Iglesias, H.J.; Escudero, C.J. Real-time personal protective equipment monitoring system. Comput. Commun. 2012, 36, 42–50. [CrossRef]

20. Wu, J.; Cai, N.; Chen, W.; Wang, H.; Wang, G. Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset. Autom. Constr. 2019, 106, 102894. [CrossRef]

21. Waranusast, R.; Bundon, N.; Timtong, V.; Tangnoi, C.; Pattanathaburt, P. Machine vision techniques for motorcycle safety helmet detection. In Proceedings of the in International Conference Image and Vision.