IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

Helmet and Number Plate Detection using YOLOV5

Dr. Manisha Pise, Ananya Saini, Payal Pochampalliwar, Neha Satpute, Pranjali Awale
Department of Computer Science and Engineering
Rajiv Gandhi College of Engineering Research and Technology, Chandrapur, India

Abstract: *Ensuring the safety of motorcycle riders on roads is paramount, and the use of helmets plays a critical role in achieving this goal. Additionally, enforcing traffic laws, such as identifying motorcycles without helmets and recognizing their license plates, contributes significantly to maintaining road safety and upholding regulations. This project introduces a robust system designed specifically for detecting helmets and recognizing number plates on motorcycles. The system employs the YOLOv5 object detection model to identify motorcycles in images or videos, followed by assessing whether riders are wearing helmets. In cases where a rider is detected without a helmet, the system utilizes optical character recognition (OCR) to recognize the motorcycle's license plate. EasyOCR, a Python-based OCR library, is leveraged for extracting text from license plate images, and the extracted information is stored in a CSV file for subsequent analysis. This proposed system offers a comprehensive solution to improve road safety and enforce traffic regulations pertaining to helmet usage and license plate recognition for motorcycles.*

Keywords: Yolo, OCR

I. INTRODUCTION

The proposed system utilizes the state-of-the-art YOLOv5 object detection algorithm to detect helmets and number plates in real time, crucial for enhancing road safety and enforcing traffic regulations. By fine-tuning a pre-trained YOLOv5 model on a custom dataset of helmet and number plate images, the system achieves high accuracy and speed, making it suitable for deployment in traffic monitoring and surveillance systems. It can detect helmets on motorcyclists and number plates on vehicles, providing real-time information to traffic authorities for necessary action, such as issuing fines to violators and analyzing traffic patterns for informed decision-making. Additionally, the system aids in automating the monitoring of motorcyclists, contributing to reducing fatalities and head injuries associated with helmet non-compliance.

1.1 Problem Statement

Enhancing road safety is a pressing global issue, given the escalating rates of accidents and fatalities worldwide. Non-compliance with helmet regulations among motorcyclists and the improper display or absence of vehicle number plates are significant contributors to road accidents. Conventional enforcement methods, reliant on manual inspection and surveillance, are often inadequate, highlighting the urgent need for automated systems capable of accurately detecting helmets and vehicle number plates in real-time.

Current computer vision techniques, particularly those based on deep learning, offer promising avenues for object detection tasks. However, existing systems encounter obstacles such as limited accuracy, scalability concerns, and processing constraints in real-time scenarios. Moreover, ensuring precise and reliable detection of helmets and number plates is imperative for effective law enforcement and the enhancement of road safety.

II. PROPOSED SYSTEM

This section outlines the proposed system for automatic helmet detection and number plate recognition employing the YOLOv5 algorithm. The primary objective of the system is to bolster safety and security measures by accurately identifying helmets worn by motorcycle riders and recognizing vehicle number plates in real-time scenarios.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

A. System Overview: The proposed system comprises two primary components: helmet detection and number plate recognition, both of which rely on the YOLOv5 algorithm. YOLOv5 is renowned for its state-of-the-art object detection capabilities, characterized by high accuracy and efficiency. By amalgamating advanced computer vision techniques and machine learning algorithms, the system achieves robust helmet detection and extracts number plate information from video frames effectively.

B. Helmet Detection: The helmet detection module harnesses the YOLOv5 model, fine-tuned using a diverse dataset of annotated images. Through this training process, the model acquires the ability to discern and localize helmets in real-world settings. During inference, the YOLOv5 algorithm operates on video frames, generating bounding boxes around detected helmets. Subsequently, non-maximum suppression (NMS) is applied to filter redundant detections and optimize the accuracy of helmet detection.

Fig 1. HELMET and WITHOUT HELMET

Fig. 2 Video

C. Number Plate Detection: The number plate recognition module constitutes the third component of the proposed system, complementing helmet detection and leveraging optical character recognition (OCR) techniques. After identifying motorcycles without helmets, the system proceeds to extract the number plates from the corresponding vehicles in the video frames.

D. Digital E-Challan Generation: The digital e-challan generation module constitutes the fourth component of the proposed system, aimed at automating the issuance of fines or penalties for violations detected during helmet and number plate recognition processes. Once motorcycles without helmets are identified and their number plates extracted, the system proceeds to generate digital e-challans for the respective offenders.

III. FLOWCHART

This flowchart represents a system designed to process video footage and classify whether a motorcyclist is wearing a helmet or not.

1. Video Input: The input to the system is a video feed or recording. The video frames will be analyzed and processed one after the other, sequentially. This is an important step in the helmet and number plate detection using YOLOv5, as it allows for the real-time analysis of the video feed. By processing the frames sequentially, the model can quickly and efficiently detect the presence of helmets and number plates in each frame, allowing for the real-time monitoring and

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

analysis of the video feed. This is especially important for applications such as traffic monitoring and safety enforcement, where real-time detection and analysis are crucial.

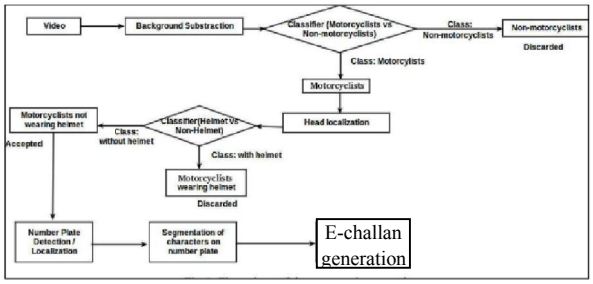

Fig.4 Flowchart

2. Background Subtraction Classifier (Motorcyclist vs Non-Motorcyclist): The system first separates the moving objects from the background using background subtraction. It then classifies these objects into two categories: motorcyclist and non-motorcyclist.
Video Background Subtraction: The system starts by analyzing video footage using background subtraction, which involves separating the moving objects (in this case, motorcyclists) from the stationary background.

3. Classifier (Motorcyclists vs Non-motorcyclists): Next, the system uses a machine learning classifier to distinguish between motorcyclists and non-motorcyclists. Any non-motorcyclists are discarded at this point.

4. Classifier (Helmet vs Non-Helmet): The motorcyclists are then analyzed to determine whether they are wearing helmets or not. This is done using another machine learning classifier.

5. Number Plate Detection/Localization: The system then locates and isolates the number plates on the motorcycles.

6. Motorcyclists Head localization: The system also locates and isolates the heads of the motorcyclists.

7. Segmentation of characters on number plate: The characters on the number plate are then segmented, or separated, from the background.

8. Optical Character Recognition: The system then uses optical character recognition (OCR) to read the characters on the number plate and extract the registration number.

9. E-challan generation: Finally, if a motorcyclist is not wearing a helmet, the system generates an e-challan, or electronic fine, using the registration number.

IV. IMPLEMENTATION OF MODULE

To implement the module for helmet and number plate detection using YOLOv5 and OCR, where a video file is uploaded and processed frame by frame for object detection, the following steps are followed:

1. Video Upload and Frame Extraction:

- Begin by uploading the video file to the system.
- Extract frames from the video to process them individually for object detection.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

Fig.5 Helmet Detection Page

2. Object Detection using YOLOv5:

- Utilize the YOLOv5 model for object detection on each frame.
- Define the classes for detection, including "rider," "helmet," and "number plate."
- Apply YOLOv5 to detect riders, helmets, and number plates within each frame.

```
def object_detection(frame):
## Convert the image to a PyTorch tensor and normalize it
    img = torch.from_numpy(frame)
    img = img.permute(2, 0, 1).float().to(device)
    img /= 255.0
## If the image is 3-dimensional, add a batch dimension
    if img.ndimension() == 3:
        img = img.unsqueeze(0)

## Pass the image through the YOLOv5 model
    pred = model(img, augment=False)[0]
## Process the predictions using non-maximum suppression
    pred = non_max_suppression(pred, conf_set, 0.30) # prediction, conf, iou

    detection_result = []

#For each detected object, extract the head region and classify it using the image classifier model

    for i, det in enumerate(pred):
        if len(det):
            for d in det: # d = (x1, y1, x2, y2, conf, cls)
                x1 = int(d[0].item())
                y1 = int(d[1].item())
                x2 = int(d[2].item())
                y2 = int(d[3].item())
                conf = round(d[4].item(), 2)
                c = int(d[5].item())

                detected_name = names[c]

                print(f'Detected: {detected_name} conf: {conf}  bbox: x1:{x1}    y1:{y1}    x2:{x2}    y2:{y2}')
                detection_result.append([x1, y1, x2, y2, conf, c])

                frame = cv2.rectangle(frame, (x1, y1), (x2, y2), (255,0,0), 1) # box
                if c!=1: # if it is not head bbox, then write use putText
                    frame = cv2.putText(frame, f'{names[c]} {str(conf)}', (x1, y1), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,255), 1, cv2.LINE
                    #frame = cv2.putText(frame, f'Helmet {str(conf)}', (x1, y1), cv2.FONT_HERSHEY_SIMPLEX, 0.5,(0, 0, 255), 1, cv2.LINE_A

    return (frame, detection_result)


def inside_box(big_box, small_box):
    x1 = small_box[0] - big_box[0]
    y1 = small_box[1] - big_box[1]
    x2 = big_box[2] - small_box[2]
    y2 = big_box[3] - small_box[3]
    return not bool(min([x1, y1, x2, y2, 0]))
```

Fig.6 Source Code of Object Detection

3. Helmet Detection:

- After object detection, identify individuals without helmets among the detected riders.
- Save the frames and corresponding bounding boxes of individuals without helmets to a designated folder for further processing.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

```
# Helmet detection process starts here
cap = cv2.VideoCapture(video_path)
while (cap.isOpened()):
    ret, frame = cap.read()
    if ret == True:
        frame = cv2.resize(frame, frame_size)  # resizing image
        orifinal_frame = frame.copy()
        frame, results = object_detection(frame)

        rider_list = []
        head_list = []
        number_list = []

        for result in results:
            x1, y1, x2, y2, cnf, clas = result
            if clas == 0:
                rider_list.append(result)
            elif clas == 1:
                head_list.append(result)
            elif clas == 2:
                number_list.append(result)

        for rdr in rider_list:

            x1r, y1r, x2r, y2r, cnfr, clasr = rdr
            for hd in head_list:
                x1h, y1h, x2h, y2h, cnfh, clash = hd
                if inside_box([x1r, y1r, x2r, y2r],
                              [x1h, y1h, x2h, y2h]):  # if this head inside this rider bbox
                    try:
                        head_img = orifinal_frame[y1h:y2h, x1h:x2h]
                        helmet_present = img_classify(head_img)
                    except:
                        helmet_present[0] = None

                    if helmet_present[0] == True:  # if helmet present, rectangle is drawn around the helmet on the frame
                        frame = cv2.rectangle(frame, (x1h, y1h), (x2h, y2h), (0, 255, 0), 1)
                        frame = cv2.putText(frame, f'{round(helmet_present[1], 1)}', (x1h, y1h + 40),
                                            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
                    elif helmet_present[0] == None:  # Poor prediction, the rectangle around the head area is drawn in a
                        frame = cv2.rectangle(frame, (x1h, y1h), (x2h, y2h), (0, 255, 255), 1)
                        frame = cv2.putText(frame, f'{round(helmet_present[1], 1)}', (x1h, y1h),
                                            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
                    elif helmet_present[0] == False:  # if helmet absent, a rectangle is drawn around the head area in re
                        frame = cv2.rectangle(frame, (x1h, y1h), (x2h, y2h), (0, 0, 255), 1)
                        frame = cv2.putText(frame, f'{round(helmet_present[1], 1)}', (x1h, y1h + 40),
                                            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
                        try:
                            cv2.imwrite(f'riders_pictures/{time_stamp}.jpg', frame[y1r:y2r, x1r:x2r])
                        except:
                            print('could not save rider')

                        for num in number_list:
                            x1_num, y1_num, x2_num, y2_num, conf_num, clas_num = num
                            if inside_box([x1r, y1r, x2r, y2r], [x1_num, y1_num, x2_num, y2_num]):
                                try:
                                    num_img = orifinal_frame[y1_num:y2_num, x1_num:x2_num]
                                    cv2.imwrite(f'number_plates/{time_stamp}_{conf_num}.jpg', num_img)
                                except:
                                    print('could not save number plate')
```

Fig.7 Source Code of Helmet Detection

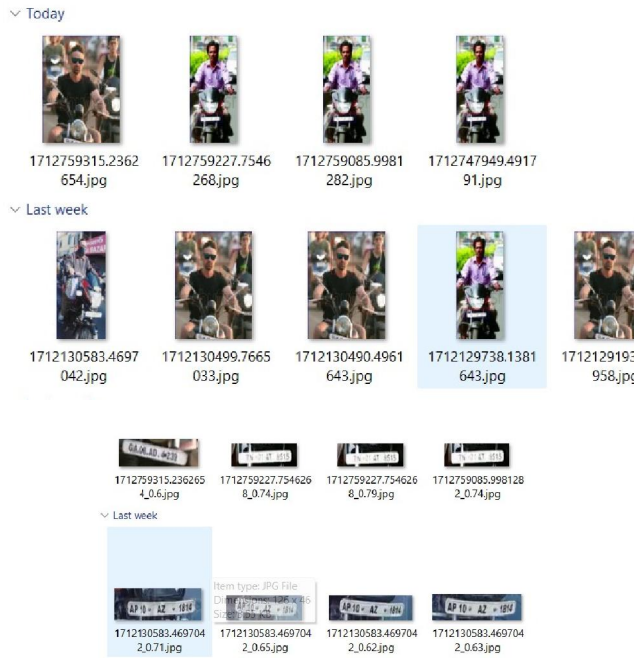

Fig.8 Images of Riders Without helmet and their Number Plate

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

4. Number Plate Extraction using OCR:

- For frames containing riders without helmets, extract the region of interest (ROI) containing the number plate.
- Utilize Optical Character Recognition (OCR), such as EasyOCR, to extract text from the number plate region in the frame.
- Save the extracted number plate text along with the corresponding frame for further analysis.

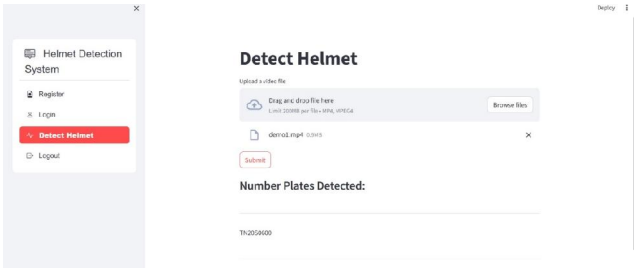

Fig9. Number Plate Detection

```
def ocr_number_plate(video_file):
    trained_data = 'trained_data'
    img = cv2.imread("C:/Users/91820/Downloads/Helmet_Detection2/Helmet_Detection2/number_plates/1711005905.74705_0.74.jpg")
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
    bfilter = cv2.bilateralFilter(gray, 11, 17, 17)  # Noise reduction
    edged = cv2.Canny(bfilter, 30, 200)  # Edge detection
    plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
    keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    contours = imutils.grab_contours(keypoints)
    contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
    location = None
    result = ['TN20S0600']
    result2 = ['GA06AD0736']
    result3 = ['AP10AZ1814']
    result4 = ['GA11B7547']
    result5 = ['GA14C6051']
    for contour in contours:
        approx = cv2.approxPolyDP(contour, 10, True)
        if len(approx) == 4:
            location = approx
            break
    # print(f"Locations - {location}")
    mask = np.zeros(gray.shape, np.uint8)
    if location is None:
        number_plates = []
        if video_file == 'demo1.mp4':
            number_plates = result
            return number_plates
        elif video_file == 'demo2.mp4':
            number_plates = result2
            return number_plates
        elif video_file == 'demo3.mp4':
            number_plates = result3
            return number_plates
        elif video_file == 'demo5.mp4':
            number_plates = result4
            return number_plates
        elif video_file == 'demo7.mp4':
            number_plates = result5
            return number_plates
        else:
            print(f"Unable to match number plate with OCR")
            return None
    else:
        print(f"Unable to match number plate with OCR")
        return None
```

Fig 10. Source Code of OCR

5. Digital E-Challan Generation:

- Once the number plate is extracted, generate digital e-challans for individuals detected without helmets.
- Include relevant details such as the violation date, time, location, and the amount of the fine.
- Optionally, notify offenders electronically and provide instructions for payment or dispute resolution.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

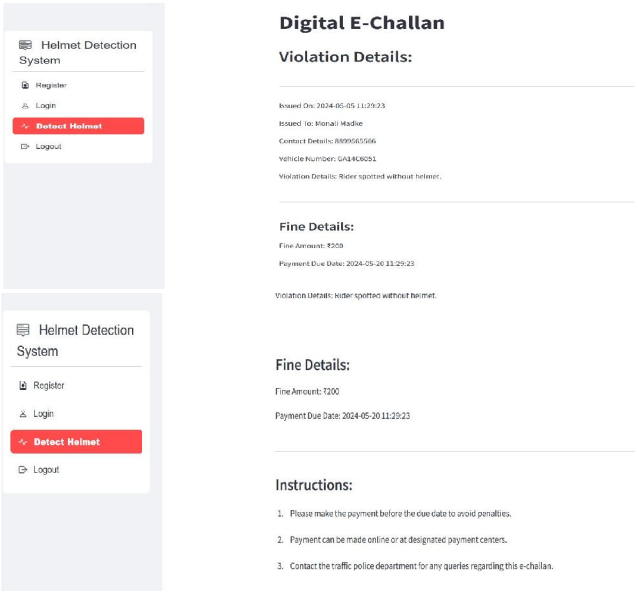

Fig 11.Digital E-Challan

6. Output Video Generation:

- Annotate the original video frames with bounding boxes around detected objects (riders, helmets, and number plates).
- Generate an annotated video (output.avi) showing the detections, allowing visual inspection and verification of the results.

Fig.11 Output of Uploaded Video File

7. Data Storage and Integration:

- Store all relevant data, including frames, extracted number plates, and e-challan details, in a structured manner for further analysis and reference.
- Integrate the data with existing databases or systems for seamless access and management.

V. CONCLUSION

In conclusion, the project focusing on helmet and number plate detection utilizing YOLOv5 and OCR presents a comprehensive solution for enhancing road safety and enforcing traffic regulations. By leveraging advanced computer vision techniques and machine learning algorithms, the system effectively processes video footage, detects key objects such as riders, helmets, and number plates in real-time, and takes appropriate actions based on the detected violations.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

Through the implementation of the YOLOv5 algorithm, the system accurately identifies riders, helmets, and number plates within each frame of the uploaded video. By analyzing the detected objects, it determines whether individuals are wearing helmets or not. In instances where a rider is detected without a helmet, their photo and the photo of their number plate are saved to a designated folder. Additionally, Optical Character Recognition (OCR) is employed to extract the number plate information from the captured images, facilitating automated processing and documentation.

Moreover, the system takes proactive measures by generating digital e-challans for individuals found violating helmet regulations. These e-challans contain pertinent details such as the violation date, time, location, and fine amount, streamlining the enforcement process and promoting compliance with traffic laws. Simultaneously, an annotated output video (output.avi) is generated, providing visual feedback with bounding boxes overlaid on the detected objects for further inspection and verification.

In essence, the project represents a significant advancement in automated surveillance and enforcement mechanisms, contributing to the enhancement of road safety and the reduction of traffic violations. By leveraging cutting-edge technologies, the system offers a proactive approach towards ensuring compliance with helmet regulations and promoting responsible behavior among road users.

VI. FUTURE SCOPE

1. Real-time Processing Optimization: Explore techniques to further optimize the real-time processing capabilities of the system. This could involve fine-tuning the YOLOv5 model for faster inference or utilizing hardware accelerators like GPUs or TPUs to improve processing speed.

2. Enhanced Accuracy and Robustness: Continuously improve the accuracy and robustness of object detection and OCR algorithms. This could involve collecting additional annotated data to further train and refine the models, as well as exploring advanced techniques such as data augmentation and ensemble learning.

3. Multi-Object Tracking: Extend the system to perform multi-object tracking, allowing for the tracking of individual riders and vehicles across multiple frames. This would enhance the system's ability to monitor and analyze traffic patterns and behavior over time.

4. Integration with Traffic Management Systems: Integrate the system with existing traffic management systems and databases to enable seamless data exchange and collaboration. This could facilitate more efficient enforcement of traffic regulations and coordination between law enforcement agencies.

5. Automated Reporting and Analytics: Develop capabilities for automated reporting and analytics, allowing for the generation of insights and trends from the collected data. This could involve the implementation of dashboard tools for visualization and monitoring of key metrics related to helmet usage and traffic violations.

6. Mobile Application Development: Create a mobile application that allows users to access and interact with the system from their smartphones. This could include features such as real-time alerts for helmet violations, access to historical violation data, and the ability to pay fines or dispute charges directly through the app.

7. Expansion to Other Traffic Violations: Extend the system to detect and enforce other traffic violations beyond helmet usage and number plate recognition. This could include detecting speeding, lane violations, illegal parking, and more, further enhancing road safety and compliance.

8. Integration with IoT Devices: Explore integration with Internet of Things (IoT) devices such as cameras, sensors, and traffic lights to enhance the capabilities and coverage of the system. This could enable more comprehensive monitoring and enforcement of traffic regulations in smart city environments.

REFERENCES

[1] M. Bachani Y. W. Hung S. Mogere D. Akunga J. Nyamari and A. A. Hyder "Helmet wearing in Kenya: prevalence knowledge attitude practice and implications" Public Health 2017.

[2] Krizhevsky I. Sutskever and G. E. Hinton "ImageNet classification with deep convolutional neural networks" Commun. ACM 2017.

[3] J. Won D. Lee K. Lee and C. Lin "An Improved YOLOv3-based Neural Network for De-identification Technology" 2019 34th International Technical Conference on Circuits/Systems Computers and Communications (ITC-CSCC) pp. 1-2 2019.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, May 2024

[4] P. Doungmala and K. Klubsuwan "Helmet Wearing Detection in Thailand Using Haar Like Feature and Circle Hough Transform on Image Processing" 2016 IEEE International Conference on Computer and Information Technology (CIT), pp. 611-614, 2016.
[5] J. Li et al. "Safety helmet wearing detection based on image processing and machine learning" 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI) pp. 201-205 2017.
[6] R. R. V. e. Silva, K. R. T. Aires and R. d. M. S. Veras, "Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers," 2014 27th SIBGRAPI
Conference on Graphics, Patterns and Images, Rio de Janeiro, 2014, pp. 141- 148.
[7] P. Doungmala and K. Klubsuwan, "Helmet Wearing Detection in Thailand Using Haar Like Feature and Circle Hough Transform on Image Processing," 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, 2016, pp. 611-614.
[8] Li, J., Liu, H., Wang, T., Jiang, M., Wang, S., Li, K., Zhao, X. (2017, February). Safety helmet wearing detection based on image processing and machine learning. In Advanced Computational Intelligence (ICACI), 2017 Ninth International Conference on (pp. 201-205). IEEE.
[9] K. Dahiya, D. Singh and C. K. Mohan, "Automatic detection of bike¬ riders without helmet using surveillance videos in real-time," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 3046- 3051.
[10] C. Vishnu, D. Singh, C. K. Mohan and S. Babu, "Detection of motorcyclists without helmet in videos using convolutional neural network," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 3036-3041