SafeLid Vision: A Machine Learning Approach to Automated Helmet Detection

Mrs. K.S.Saraswathi Devi
Assistant Professor of Computer Science, Government First Grade College -VIJAYANAGAR, Bangalore 40

Abstract -In major Indian cities, a surge in vehicle ownership has created a gridlock – traffic congestion. This, coupled with lax enforcement of helmet use among motorcycle riders, has become a major safety concern. Despite regulations mandating helmets, many riders disregard this rule, leading to a rise in accidents and fatalities. Current enforcement methods heavily rely on reviewing CCTV footage, requiring officers to manually zoom in and identify license plates of riders without helmets. This process is not only time-consuming but also inefficient. This paper proposes a solution that leverages machine learning for automated helmet detection. Helmet detection is a critical but complex task in computer vision, with applications beyond traffic monitoring. The proposed method tackles this challenge in three stages: pre-processing, feature extraction, and classification. The paper also reviews existing technologies and strategies for helmet detection, highlighting the promising results achieved by recent research using features extracted from techniques like Convolutional Neural Networks (CNNs). Our research aims to develop a system that automatically detects riders without helmets and captures their license plate numbers. The core concept revolves around a three-level deep learning object detection approach. First, the YOLOv3 algorithm identifies objects like motorcycles within the video frame. Then, YOLOv5 focuses on these detected motorcycles to determine if the rider is wearing a helmet. Finally, if a helmet violation is identified, YOLOv3 zooms in on the motorcycle to extract the license plate number using Optical Character Recognition (OCR). This extracted license plate number allows authorities to verify the vehicle's registration and potentially identify stolen vehicles within a database. By automating helmet detection and license plate capture, this system empowers traffic police to not only identify helmet violators but also track down stolen vehicles, significantly improving traffic safety and enforcement efficiency.

Keywords: CNN, ORC, YOLO3, Machine Learning, Helmet Detection

INTRODUCTION

Wearing a helmet is mandatory by law and essential for safety, yet enforcing helmet use among motorcycle riders in India remains a challenge. Limited manpower and human fatigue hinder traditional methods relying on traffic police vigilance. Similarly, CCTV surveillance used in major cities requires manual review, creating a bottleneck in enforcement. This research proposes an automated solution for motorcycle helmet detection using machine learning. The system addresses the growing number of motorcycles on the road and the increasing concern for rider safety. It automatically detects riders without helmets and captures their license plate numbers from CCTV footage, particularly useful at intersections where traffic flow is more predictable. Object recognition and tracking are crucial components of the system's computer vision algorithm. However, robust object detection can be difficult due to variations in scenery and occlusions (e.g., another vehicle partially blocking the rider). To overcome these challenges, the system employs the TensorFlow object detection API for identifying moving objects. The core technology utilizes Convolutional Neural Networks (CNNs) for object identification, a prevalent technique for tasks like image classification and real-time object detection due to its efficiency and accuracy. While region-based CNNs offer high accuracy, their computational complexity makes them less suitable for real-time applications. In an effort to improve motorcycle rider safety and address the growing number of motorcycles on Indian roads, this research proposes an automated system for helmet detection using machine learning. Traditional enforcement methods, limited by manpower and the time-consuming nature of CCTV review, are no longer sufficient. This system tackles the challenge of object recognition and tracking in video footage by leveraging a TensorFlow object detection API. The core technology relies on

Convolutional Neural Networks (CNNs), a popular choice for their efficiency and accuracy in tasks like image classification and real-time object detection.

LITERATURE REVIEW

Automatic license plate detection systems have become increasingly sophisticated, with YOLO being a popular foundation for many approaches. Rayson et al. achieved a 96.8% recognition rate using YOLO, highlighting its potential for real-time detection of multiple vehicles. However, challenges like traffic, lighting variations, and weather can affect performance. To address this, Hsu et al. proposed a YOLO-based deep learning technique that incorporates these environmental factors. Preprocessing techniques like skew correction have also been explored to improve accuracy. Abdussalam et al. implemented this approach before applying deep learning algorithms. Lele Xie et al. introduced the MDYOLO framework, which tackles rotational challenges in real-time scenarios. Pan Gao et al. achieved real-time license plate recognition using a 30-class convolutional neural network that merges the benefits of resident networks and dense convolutional networks. Researchers are also exploring YOLO variations for license plate detection. Diogo M. F et al. employed the tiny YOLOv3 architecture for Brazilian license plates, combining it with a second convolutional network for character identification. Weishan Zhang et al. proposed a scalable approach using YOLO for localization and CNN for recognition, achieving good results. While some studies, like Vitalii Varkentin et al., achieved a 73% accuracy rate with a YOLO-based approach, there's room for improvement. MJ Prajwal et al. addressed motion blur, a common challenge in real-world scenarios, using a Wiener filter. They also created a custom dataset based on Indian character sets to improve recognition accuracy. Their system utilizes YOLOv3 for initial detection followed by a trained neural network for character recognition. Overall, these studies showcase the continuous advancements in automatic license plate detection systems. YOLO and convolutional
neural networks play a key role, with researchers exploring various techniques to improve accuracy and address real-world challenges like environmental variations and motion blur.

SYSTEM ARCHITECTURE

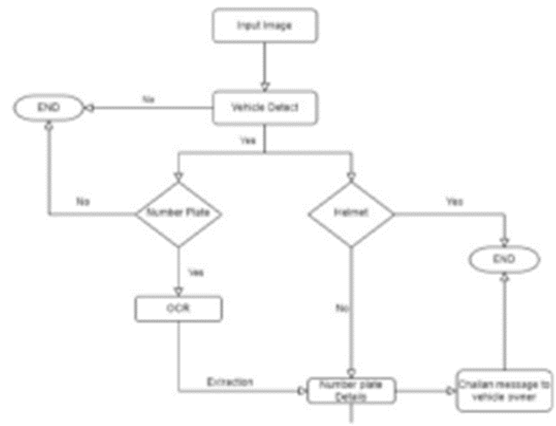

Fig.1.1: System Architecture

● Real-time Image Capture: A CCTV or traffic camera snaps a picture of every vehicle. ● Motorcycle Detection: A Convolutional Neural Network (CNN) classifier analyzes these images, separating motorcycles from other vehicles. Cars, buses, or anything not a motorcycle is filtered out, leaving only potential motorcycle riders for further inspection.

● Helmet Detection: Another CNN classifier takes over to determine if the rider is wearing a helmet. Since heads are typically located in the upper portion of the image, the system focuses on the top quarter. A separate CNN, trained on a large dataset of images with and without helmets, analyzes this specific region to identify riders with or without helmets.

● License Plate Detection: If a helmet violation is detected, the system needs to identify the vehicle. A dedicated license plate detection function comes into play, pinpointing the license plate within the captured image.

● Character Segmentation and Recognition: Once the license plate is located, it's time to decipher the characters. Techniques like character segmentation come into play, separating individual letters and numbers from the plate. Finally, Optical Character Recognition (OCR) technology, often powered by OpenCV, reads and recognizes each character, extracting the entire license plate number.

● Challan Generation: With the confirmed violation (no helmet) and the identified vehicle (license plate number), the system can automatically generate a challan (traffic ticket). It offers a significant improvement over traditional methods, enhancing traffic safety and enforcement efficiency.

CONCLUSION

This study proposes a framework to automatically identify and penalize motorcyclists who violate helmet laws using CCTV footage. Convolutional Neural Networks (CNNs) with transfer learning will ensure high accuracy in detecting riders without helmets. But simply identifying violators isn't enough. The system goes a step further by capturing their license plates as well. These license plates are then stored and made accessible to the Transport Office. By cross-referencing with their vehicle database, the office can obtain information about the riders, allowing them to issue challans (traffic tickets) to those who disregard helmet safety regulations. Additionally, the system can be enhanced to detect stolen vehicles, adding another layer of security to traffic monitoring.

FUTURE SCOPE

Our research successfully built a helmet detection application using Python and the Jupyter Notebook environment. We rigorously tested the application and explored its potential applications and future directions. The exciting part is that with some modifications, the system can be deployed in real-time. Imagine connecting it directly to traffic cameras! This would enable real-time helmet detection, allowing for immediate action. Furthermore, the system can be integrated with automatic license plate identification and stolen vehicle algorithms. For riders who violate helmet laws, this combined system could not only identify them but also potentially flag stolen vehicles, significantly improving traffic safety and enforcement efficiency.

REFERENCE

[1] Rubaiyat, A.H., Toma, T.T., Kalantari-Khandani, M., Rahman, S.A., Chen, L., Ye, Y. and Pan, C.S., 2016, October. Automatic detection of helmet uses for construction safety. In 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW) (pp. 135-142). IEEE.

[2] Li, J., Liu, H., Wang, T., Jiang, M., Wang, S., Li, K. and Zhao, X., 2017, February. Safety helmet wearing detection based on image processing and machine learning. In 2017 Ninth international conference on advanced computational intelligence (ICACI) (pp. 201-205). IEEE.

[3] Boonsirisumpun, N., Puarungroj, W. and Wairotchanaphuttha, P., 2018, November. Automatic detector for bikers with no helmet using deep learning. In 2018 22nd international computer science and engineering conference (ICSEC) (pp. 1-4). IEEE.

[4] Khan, F.A., Nagori, N. and Naik, A., 2020, July. Helmet and number plate detection of motorcyclists using deep learning and advanced machine vision techniques. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 714-717). IEEE.

[5] Dasgupta, M., Bandyopadhyay, O. and Chatterji, S., 2019, December. Automated helmet detection for multiple motorcycle riders using CNN. In 2019 IEEE Conference on Information and Communication Technology (pp. 1-4). IEEE.

[6] Silva, R., Aires, K., Santos, T., Abdala, K., Veras, R. and Soares, A., 2013, October. Automatic detection of motorcyclists without helmet. In 2013 XXXIX Latin american computing conference (CLEI) (pp. 1-7). IEEE.

[7] Long, X., Cui, W. and Zheng, Z., 2019, March. Safety helmet wearing detection based on deep learning. In 2019 IEEE 3rd information technology, networking, electronic and automation control conference (ITNEC) (pp. 2495-2499). IEEE.

[8] Vishnu, C., Singh, D., Mohan, C.K. and Babu, S., 2017, May. Detection of motorcyclists without helmet in videos using convolutional neural network. In 2017 International joint conference on neural networks (IJCNN) (pp. 3036-3041). IEEE.

[9] Wu, F., Jin, G., Gao, M., Zhiwei, H.E. and Yang, Y., 2019, May. Helmet detection based on improved YOLO V3 deep model. In 2019 IEEE 16th International conference on networking, sensing and control (ICNSC) (pp. 363-368). IEEE.

[10] Wang, H., Hu, Z., Guo, Y., Yang, Z., Zhou, F. and Xu, P., 2020. A real-time safety helmet wearing detection approach based on CSYOLOv3. Applied Sciences, 10(19), p.6732.

[11] Wu, H. and Zhao, J., 2018. Automated visual helmet identification based on deep convolutional

neural networks. In Computer aided chemical engineering (Vol. 44, pp. 2299-2304). Elsevier. 12. Shine, L. and CV, J., 2020. Automated detection of helmet on motorcyclists from traffic surveillance videos: a comparative analysis using hand-crafted features and CNN. Multimedia Tools and Applications, 79(19), pp.14179-14199.

[12] e Silva, R.R.V., Aires, K.R.T. and Veras, R.D.M.S., 2014, August. Helmet detection on motorcyclists using image descriptors and classifiers. In 2014 27th SIBGRAPI Conference on graphics, patterns and images (pp. 141-148). IEEE.

[13] Benyang, D., Xiaochun, L. and Miao, Y., 2020, November. Safety helmet detection method based on YOLO v4. In 2020 16th International conference on computational intelligence and security (CIS) (pp. 155-158). IEEE.

[14] Li, P., Meziane, R., Otis, M.J.D., Ezzaidi, H. and Cardou, P., 2014, October. A Smart Safety Helmet using IMU and EEG sensors for worker fatigue detection. In 2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings (pp. 55-60). IEEE.

ANNEXURE (ALGORITHM)

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
# Pre-trained models (replace with your trained
models)
motorcycle_detector =
load_model("motorcycle_detector.h5")
helmet_detector = load_model("helmet_detector.h5")
license_plate_detector =
load_model("license_plate_detector.h5")
def detect_helmet(frame):
# 1. Motorcycle Detection with YOLOv3
   # Likely involves feeding the frame into
motorcycle_detector and getting bounding boxes
# around detected motorcycles
    motorcycle_boxes = detect_motorcycles(frame) #
Placeholder for your detection function
for box in motorcycle_boxes:
# 2. Helmet Detection
x_min, y_min, x_max, y_max = box
motorcycle_region = frame[y_min:y_max,
x_min:x_max]
          # Preprocess motorcycle region (resizing,
normalization) before feeding to helmet_detector
preprocessed_region =
preprocess_image(motorcycle_region)
        helmet_probability =
helmet_detector.predict(np.expand_dims(preprocesse
d_region, axis=0))[0][0]
if helmet_probability < 0.5: # Threshold for helmet
presence (adjust as needed) # 3. License Plate
Detection (if helmet violation)
       license_plate_box = detect_license_plate(frame,
box) # Placeholder for your detection function
if license_plate_box is not None:
# 4. Character Segmentation and Recognition
                license_plate_image =
   frame[license_plate_box[1]:license_plate_box[3],
        license_plate_box[0]:license_plate_box[2]]
plate_number =
extract_license_plate(license_plate_image)
# 5. Challan Generation
generate_challan(plate_number)
# Implement functions for:
# - detect_motorcycles (YOLOv3 implementation for
motorcycle detection) # - preprocess_image
(functions to resize and normalize image for CNN) #
- detect_license_plate (potentially another YOLO
model for license plate detection) # -
extract_license_plate (character segmentation and
OCR logic) # - generate_challan (logic to store
violation details and potentially send challan) # Main
loop (replace with your video capture logic)
while True:
ret, frame = cap.read() # Assuming cap is a video
capture object detect_helmet(frame)
cv2.imshow("Helmet Detection", frame)
if cv2.waitKey(1) == ord('q'):
break
cap.release()
cv2.destroyAllWindows()
```