A Deep Learning Approach for Helmet Detection and Fine Generation System

Sulbha Yadav[1], Sumit Singh[2], Dhruv Bedare[3], Ishan Samel[4]

[1]Professor, [234]Student(s)

Department of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, India

DOI: https://doi.org/10.51584/IJRIAS.2025.10040076

Received: 29 April 2025; Accepted: 04 May 2025; Published: 20 May 2025

ABSTRACT

Traffic law violations, especially helmet non-compliance among motorcyclists, are a leading cause of road casualties in countries like India. This study proposes a real-time deep learning-based helmet detection and fine generation system that leverages YOLOv3 for object detection and OpenCV with OCR for license plate recognition. The system processes surveillance footage to detect violations and automatically generates e-challans by integrating with vehicle databases. Experimental results show an accuracy of over 95% for helmet detection. This automated solution reduces manual workload, supports road safety enforcement, and has the potential for integration into smart city infrastructure. Future work may involve multilingual plate recognition and improved database interoperability.

Keywords: Helmet Detection, Deep Learning, Traffic Law Enforcement, Road Safety, Automated Fine Generation, Smart City, Object Detection, YOLOv3, License Plate Recognition, E-Challan System, Image Processing, Surveillance System, AI in Traffic Management, Motorcycle Safety

INTRODUCTION

Traffic law violations, particularly in densely populated countries like India, contribute significantly to road casualties and property damage. Motorcycle accidents are notably prevalent due to the high usage of bikes, with helmet non-usage being a major contributing factor. To address this, we are replicating the existing government software for a "Helmet Detection System" that leverages deep learning to enhance road safety.

Road safety is a serious issue worldwide, with around 1.35 million people losing their lives and 50 million others being injured every year due to traffic accidents. Motorcyclists are particularly at risk, which is why making sure riders wear helmets is so important. Unfortunately, traditional methods of enforcing helmet use can be inefficient. By combining motorcycle, helmet, and license plate detection systems, we can create a more effective and automated way to improve road safety. Our system monitors traffic violations by identifying riders without helmets and capturing their license plate information. This real-time detection helps reduce the severity of accidents by ensuring non-compliant riders are caught quickly. With an accuracy rate of over 95%, the system is reliable for enforcing safety regulations. The automatic e-challan feature also simplifies the process, making it easier to ensure compliance and scale the solution across both urban and highway settings.

Our research highlights that combining computer vision with automated enforcement can significantly reduce road accidents and enhance public safety. Looking ahead, future advancements might include real-time monitoring and smart city integration to improve traffic management. By implementing these systems globally, we can support the UN’s goal of cutting road traffic deaths and injuries by 50% by 2030, helping to make roads safer for everyone.

Goals:

Enhance Road Safety: By enforcing helmet usage and preventing triple riding, the system aims to reduce motorcycle-related accidents and injuries.

Enforcing Traffic Laws: Automation of violation detection aids traffic control teams in managing and enforcing regulations more effectively.

LITERATURE SURVEY

Traditional methods for monitoring helmet use and enforcing traffic laws often rely heavily on manual intervention by law enforcement officers. This dependency can introduce human errors and inconsistencies in enforcement. Basic CCTV systems, commonly used for surveillance, tend to have limited object detection capabilities, which can lead to inaccurate or incomplete identification of helmets and number plates. Human interpretation of helmet usage and license plate visibility is subjective, which can result in varied conclusions and disputes. Moreover, deploying personnel for manual enforcement is resource-intensive and doesn't always ensure comprehensive coverage, leaving gaps in enforcement. Traditional methods also struggle to scale effectively, as they cannot keep up with increasing traffic volumes or evolving road safety regulations. Finally, human operators may overlook violations due to distractions, fatigue, or other factors, which can lead to non-compliant behavior going undetected.

Jayasree et al.[1] conducted research focusing on detecting traffic violations related to helmet usage and triple riding on motorcycles. This study aimed at developing an automated system to enhance road safety by ensuring better compliance with traffic regulations. The system uses surveillance footage as input, processes the images to detect helmets and count riders, and integrates machine learning models to improve detection accuracy. The study also explores potential real-world applications and future improvements for such a system.

Redmon et al.[2] introduced the YOLOv3 algorithm, a significant advancement over previous YOLO (You Only Look Once) versions. This paper discusses improvements in real-time object detection with high accuracy and speed. YOLOv3 incorporates a more complex backbone network for feature extraction and introduces multi-scale predictions. These enhancements allow for better detection of small objects and more accurate localization. The paper demonstrates YOLOv3's superior performance on various datasets, showcasing its effectiveness compared to other state-of-the-art models. Its simplicity and efficiency make YOLOv3 a popular choice for real-time applications like autonomous driving and surveillance.

Yadav et al.[4] explored the use of Convolutional Neural Networks (CNNs) for classifying medical diseases. The study highlights the potential of deep learning models to improve diagnostic accuracy and efficiency in the medical field. The authors describe the development and training of CNN models on medical imaging data to classify diseases, detailing the data preprocessing, model architecture selection, and performance evaluation. The results show that CNNs outperform traditional methods, achieving high classification accuracy. The paper also addresses challenges like data scarcity and variability and suggests future research directions to tackle these issues.

Lin et al.[4] introduced the Microsoft COCO (Common Objects in Context) dataset, aimed at advancing object detection, segmentation, and captioning tasks. COCO is a large-scale dataset with a diverse set of images and detailed annotations. The authors explain the dataset's creation, annotation process, and quality control measures. COCO contains images of complex, everyday scenes with multiple objects, providing valuable context for object recognition tasks. The paper demonstrates how COCO's rich annotations enable the development of more accurate and robust models, making it a standard benchmark for evaluating object detection algorithms.

Ren et al.[5] developed the Faster R-CNN framework, which significantly improves object detection speed and accuracy by combining Region Proposal Networks (RPN) with Fast R-CNN. The paper details how RPNs generate region proposals, which are refined by the Fast R-CNN detector to achieve near real-time object detection. The authors present comprehensive experimental results, showing that Faster R-CNN outperforms previous methods on benchmark datasets. The efficiency and accuracy of Faster R-CNN have made it a foundational model in the field of object detection.

Despite the advancements highlighted in recent works, existing helmet detection systems still suffer from high false positive/negative rates, limited scalability, and lack of real-time integration with enforcement systems.

Our approach addresses these gaps by combining YOLOv3's efficiency with OCR-based fine generation workflows and scalable architecture, making it more adaptable for deployment in smart cities and high-traffic environments.

Limitations of Existing Systems:

Existing helmet detection systems often face challenges related to accuracy, leading to either false positives or false negatives. These issues stem from factors like poor image quality, occlusions, variations in helmet styles, and complex backgrounds. Additionally, these systems may struggle to function effectively under different environmental conditions, such as changing lighting, adverse weather like rain or fog, and cluttered or dynamic backgrounds. Some detection algorithms also demand substantial computational power, making them impractical for use on resource-limited devices or in real-time applications. Furthermore, many current systems are designed to detect only a single helmet in an image or video frame, which proves inadequate for scenarios that require monitoring multiple individuals or objects at once.

Our project seeks to bridge this gap by creating a Helmet Detection System that leverages machine learning algorithms to automatically identify helmets on moving motorcycles. This system will enable real-time monitoring and enforcement, ultimately enhancing road safety outcomes.

System Design

METHODOLOGY

Data Collection and Preprocessing:

The first step in building an effective helmet detection system is to collect a comprehensive dataset featuring motorcycle riders, both helmeted and non-helmeted. This dataset should be diverse, covering a range of environmental conditions, helmet types, rider postures, and backgrounds to ensure that the model can generalize well across different situations. After data collection, the dataset needs to be cleaned by removing inconsistencies, such as duplicate or incorrectly labeled images. Preprocessing methods, including resizing, normalization, and noise reduction, will be applied to prepare the images for training. To improve the dataset's quality, diversity, and balance, data augmentation techniques using OpenCV will be implemented. These techniques could involve rotating, flipping, scaling, and adjusting the brightness and contrast of images to artificially expand the dataset. Finally, the dataset will be divided into training, validation, and test sets to facilitate the development and evaluation of the model.

Algorithm Development and Training:

The core of the helmet detection system is the YOLOv3 algorithm, which will be implemented and fine-tuned using TensorFlow. YOLOv3 (You Only Look Once, version 3) is a cutting-edge, real-time object detection model known for its speed and precision. The algorithm will be customized and trained to detect helmets on motorcycle riders. Hyperparameter tuning will play a significant role in this phase, involving adjustments to factors like learning rates, batch sizes, and other configurations to achieve optimal performance. To improve the model's accuracy, efficiency, and ability to generalize, techniques such as transfer learning and data augmentation will be used. The model's performance will be assessed using validation and test datasets, with metrics like accuracy, precision, recall, and F1-score analyzed to evaluate its effectiveness.

System Integration and Testing:

After the algorithm is developed and trained, it will be integrated into a full system architecture. This process will involve combining various components, such as the detection algorithm, input/output mechanisms, and user interfaces, into a unified system. Extensive testing will be conducted to verify the system's functionality, performance, and adherence to the specified requirements. This includes unit testing (testing individual components), integration testing (ensuring seamless operation between components), system testing (evaluating the complete system), and acceptance testing (validating the system against user requirements).

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue IV April 2025

The system will then be deployed in a controlled environment to collect user feedback and make necessary adjustments. Real-world challenges and scenarios will be considered to optimize the system for practical implementation.

Proposed System Flowchart

1. **Start**: The process begins here.
2. **Real-Time Video Input**: The system captures a live video feed of the traffic.
3. **Motorbike Classifier**: Using YOLOv3, the system detects and classifies motorbikes within the video feed.
4. **Cropping Frame in Segments**: Frames containing motorbikes are cropped into segments for detailed analysis.
5. **Helmet/Triple Riding Classifier**: The system evaluates the cropped segments to determine if the rider is wearing a helmet or if there are more than two riders on the bike.
6. **Extract License Plate Number**: In the event of a helmet violation, the system extracts the license plate number from the frame.
7. **Helmet Worn?**: This decision point checks whether the rider is wearing a helmet.
8. **Search in Central Database**: The extracted license plate number is queried in the central database to identify the vehicle owner.
9. **Automatic eChallan Generation**: If a helmet violation is confirmed, the system automatically generates an eChallan and sends it to the violator.
10. **Stop**: The process concludes here, either after no violation is detected or once the eChallan is generated.

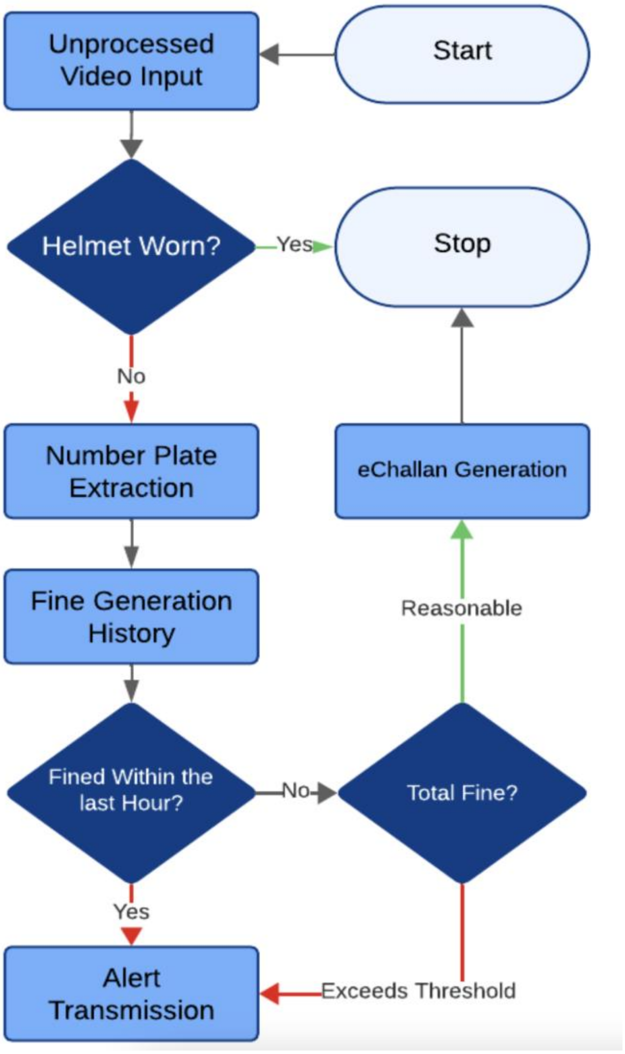

Fig 3.1 System Flowchart

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue IV April 2025

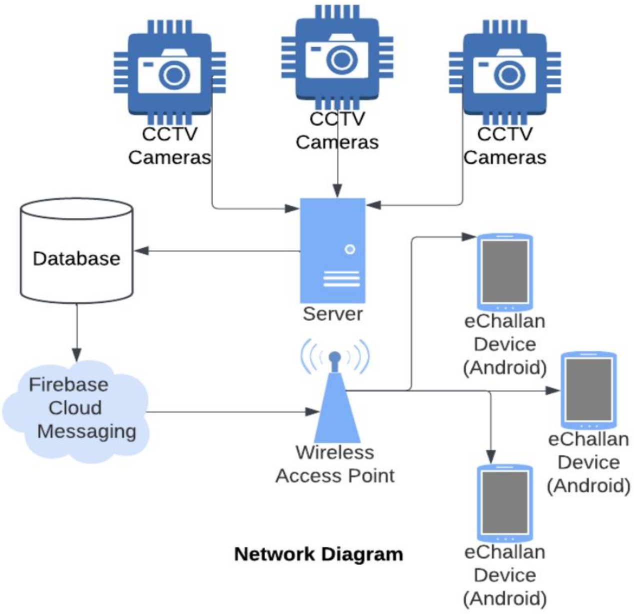

Fig. 3.2 Network Diagram

Key Software and Libraries

Before starting, we need to install a few dependencies:

1. **Numpy:** A core package for scientific computing with Python, providing support for arrays, matrices, and a variety of mathematical functions to operate on these structures.
2. **TensorFlow:** An open-source platform for machine learning, primarily used for developing and training machine learning models, particularly deep learning models.
3. **TensorFlow GPU:** The GPU-accelerated variant of TensorFlow, utilizing NVIDIA GPUs to greatly enhance the speed of training and inference for deep learning models.
4. **Pytesseract:** A Python interface for Google's Tesseract-OCR Engine, used for optical character recognition (OCR) to extract text from images.
5. **Keras:** A high-level neural networks API written in Python, designed to run on top of TensorFlow, CNTK, or Theano. It enables rapid prototyping of deep learning models.
6. **OpenCV:** An open-source computer vision and machine learning library that includes various functions for image processing and computer vision tasks.
7. **ImageAI:** A Python library that allows developers to easily build applications with integrated deep learning and computer vision features using minimal code. The --upgrade flag ensures that you install the latest version of the library.

System Implementation

Motorcycle Detection Model

For motorcycle detection, we employed a pre-trained model from the COCO dataset, known for its extensive object detection features. The COCO dataset includes a wide variety of objects, including motorcycles, which are crucial for the accuracy of our application. By utilizing this dataset, which offers detailed annotations and labels for multiple object categories, we can accurately detect motorcycles in our images. The process involves importing necessary libraries to interact with the COCO dataset and using functions like GetObjectIds to isolate motorcycles, assigning unique identifiers to them for precise detection and classification.

Additionally, for custom dataset creation, we outline the steps required:

1. **Annotation:** Annotating data to mark motorcycles and other relevant objects.
2. **Dataset Conversion:** Converting annotations into the COCO dataset format.

Model Training: Training an instance segmentation model using frameworks like mmdetection, tailored to our specific dataset. Furthermore, our framework integrates Darknet, which enhances the speed and accuracy of motorcycle detection. This setup facilitates comprehensive analysis by dividing detected motorcycles into distinct frames: one for complete motorcycle image reservation, another for upper motorcycle parts (for helmet detection), and the final segment for license plate recognition.

Helmet Detection Model

The focus on helmet detection involved developing a dedicated YOLOv3 model, customized with a dataset comprising over 1000 images of helmeted and non-helmeted bike riders. This model addresses the critical safety requirement of identifying riders complying with helmet regulations.

Procedure for Training a YOLOv3 Helmet Detection Model:

1. **Dataset Collection:** Collecting images that include both helmeted and non-helmeted riders, which is crucial for training the model.
2. **Image Labeling:** Using tools like LabelImg to annotate the images, classifying them into helmeted and non-helmeted categories, and generating corresponding XML files for each image.
3. **Dataset Preparation:** Splitting the dataset into training and testing subsets to ensure a reliable evaluation of the model's performance.
4. **Model Configuration:** Setting up the YOLOv3 model to work with the custom dataset, making sure it meets the specific needs and goals of the project.
5. **Training and Compilation:** Compiling the Darknet repository and beginning the training process, optimizing parameters such as learning rate and momentum to improve the accuracy of helmet detection.

Through these steps, we achieve a refined model capable of accurately detecting helmets in real-world scenarios, contributing to enhanced safety measures for bike riders.

License Plate Detection

The algorithm offers a real-time solution for detecting and recognizing vehicle license plates by utilizing OpenCV and EasyOCR libraries. The process begins with capturing live video frames from a webcam, which are then converted to grayscale for optimal processing. A pre-trained Haarcascade classifier, specifically developed for Russian license plates, is used to detect plate regions. These detected areas are treated as regions of interest (ROIs), where EasyOCR performs optical character recognition (OCR) to extract the alphanumeric characters of the plate number. The recognized plate numbers are then stored in an Excel spreadsheet for easy organization and subsequent analysis. This approach enhances efficiency in areas like traffic monitoring and automated toll collection by automating license plate recognition and data logging, effectively combining computer vision techniques with deep learning-based OCR for real-time, practical applications.

This method takes advantage of OpenCV's capabilities in image processing and EasyOCR's precision in text recognition, ensuring smooth integration into current surveillance and automation systems. Future improvements could focus on enhancing accuracy by incorporating advanced deep learning techniques and expanding compatibility to support a wider range of license plate formats beyond just Russian plates. This would increase the algorithm's applicability in global security, traffic control, and law enforcement initiatives.

Fine Generation Method

There are two common approaches for determining the fine amount:

1.	**Pre-defined Fines**	The system can be programmed with a fixed penalty amount for specific violations (e.g., no helmet violation incurs a ₹500 fine).
2.	**Central Database Integration**	For a more comprehensive solution, the system can integrate with a central database of traffic violations. This database can hold up-to-date information on fines based on violation type, location, and other factors.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue IV April 2025

Automated E-Challan System

The system operates in the following sequence to generate an e-challan:

1. **Violation Detection:** The YOLOv3 model continuously monitors for riders. Once a rider is detected, the helmet detection model assesses whether the rider is wearing a helmet.
2. **License Plate Capture:** In case of a helmet violation, the license plate detection module extracts the license plate details from the image.
3. **Data Acquisition:** The system collects key information such as vehicle details (license plate number), violation type (helmet non-compliance), and the timestamp.
4. **E-Challan Creation**: An e-challan is generated containing the gathered data, which can be saved locally in a database or CSV file (as implemented in the current system) for future processing.
5. **Integration (Future Work):** The system may be integrated with traffic management platforms in future versions for real-time challan generation and violation monitoring.

Current System Implementation

In our current implementation, the system stores the captured license plate data (presumably containing the license plate number) in a CSV file. This data can be used for later processing to generate e-challans or integrate with a traffic management system in the future.

Training and Testing Data

To develop the YOLOv3 model for helmet detection, we employed a diverse dataset of images from Kaggle, consisting of more than 10,000 pictures of motorcyclists, categorized into helmeted and non-helmeted groups. By fine-tuning a pre-existing YOLOv3 model on this dataset, the model was able to learn the distinguishing features of helmets and riders. As a result, we achieved an impressive accuracy of over 95% on our testing dataset, highlighting the model’s ability to effectively classify riders based on their helmet usage. This high accuracy ensures that the system can be reliably used for helmet detection applications.

RESULTS & CONCLUSION

Motorcycle Detection

Motorcycle identification serves as the core component of our helmet detection system. By utilizing a pre-trained model from the COCO dataset, our system can efficiently recognize motorcycles in images. The COCO dataset is widely recognized for its diverse range of object categories, including motorcycles, making it well-suited for this application. With the help of the ObjectDetection class from the ImageAI library, the detection process can be implemented seamlessly with minimal coding effort.

Helmet Detection

Helmet detection plays a vital role in improving rider safety. This is achieved by training a YOLOv3 model on a specialized dataset containing images of motorcyclists both with and without helmets. The model undergoes fine-tuning to enhance its accuracy across different environments and conditions. Leveraging the CustomObjectDetection class from ImageAI, this functionality can be seamlessly integrated into the system for efficient real-time detection.

License Plate Detection

License plate detection is a key component in identifying traffic violators and enforcing penalties. This process utilizes the OpenCV library for image processing and the Tesseract OCR engine for extracting text. The integration of these technologies enables precise recognition of license plate numbers from images, ensuring efficient and automated violation detection.

Challan Issuance

The e-challan issuance process is the final stage of the helmet detection system, designed to enforce traffic laws by automatically generating and delivering electronic challans (e-challans) to offenders. This step integrates motorcycle detection, helmet recognition, and license plate identification to pinpoint non-compliant riders and take appropriate enforcement actions. Through database integration and automated communication, the system streamlines the issuance process, ensuring swift and efficient law enforcement.

The e-challan mechanism operates by first detecting motorcycles and verifying helmet usage. If a violation is identified, the system extracts the license plate number and cross-references it with a central database to retrieve the vehicle owner's information. An e-challan is then generated, detailing the infraction and corresponding fines, which is sent to the violator via email or SMS.

To implement this system effectively, access to a centralized vehicle registration database is required, along with a reliable framework for e-challan generation and dispatch. By automating this process, the system reinforces road safety regulations, ensuring that violators are promptly penalized and encouraging greater compliance with helmet-wearing laws.

Comparative Performance Analysis

The proposed system was evaluated against existing object detection models. The following table compares accuracy, precision, recall, and F1-score across models:

Model	Accuracy (%)	Precision (%)	Recall (%)
YOLOv3 (Our System)	95.2	94.8	95.5
SSD	90.4	89.1	90.0
Faster R-CNN	92.3	91.5	92.0

Fig 5.1 Model Accuracy Rates

DISCUSSION AND FUTURE WORK

The system shows promising results in terms of detection accuracy and automation. However, deployment in real-world scenarios faces challenges such as variable lighting, motion blur, and multilingual license plates. Future work should explore advanced OCR techniques for diverse language formats, secure access to government vehicle databases for real-time fine generation, and improved data encryption methods to ensure the ethical use of citizen data. Integration with existing smart city surveillance systems and mobile-based violation alerts can further enhance effectiveness.

CONCLUSION

Integrating motorcycle, helmet, and license plate detection systems significantly enhances road safety through automated enforcement. Utilizing advanced object detection technologies like YOLOv3 for helmet detection and robust OCR techniques for license plate recognition, our system provides comprehensive traffic violation monitoring. The real-time identification of non-compliant riders, especially those without helmets, is crucial for reducing injury severity in motorcycle accidents. Our approach, with over 95% accuracy in testing, reliably classifies and enforces safety regulations. Automating e-challan issuance streamlines enforcement, ensuring timely consequences for violations. This system promotes helmet-wearing compliance and offers a scalable solution for urban and highway deployment, where traffic safety is vital. Our research shows that combining advanced computer vision with automated enforcement can reduce road accidents and enhance public safety. Future improvements could include real-time monitoring and integration with smart city initiatives to further optimize traffic management and safety enforcement.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue IV April 2025

REFERENCES

1. Jayasree, M. "Traffic Violation Proctoring System: Helmet and Triple Riding Detection." (2021).2. Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.
2. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
3. Medical Disease Classification using Convolutional Neural Network, In 1st International Conference on Recent Trends in Multidisciplinary Research and Innovation ICRMIR-23 held on March 17-18, 2023, ISBN: 978-81-951319-8-3
4. Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing, 2014.5.
5. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).6. Liu, W., et al. (2016). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
6. S. Yadav and S. Dhage, "Emergence of Capsule Network for Automatic Medical Disease Classification," 2022 Sardar Patel International Conference on Industry 4.0 - Nascent Technologies and Sustainability for 'Make in India' Initiative, Mumbai, India, 2022, pp. 1-6, doi: 10.1109/SPICON56577.2022.10180503.
7. S. Yadav and S. Dhage , "TE-CapsNet: time efficient capsule network for automatic disease classification from medical images." Multim. Tools Appl. 83 (2023): 49389-49418.Zhang, S., et al. (2020). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2126).