

## Novel deep learning model for vehicle and pothole detection

Gayathri K, Thangavelu S

Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore,  
Amrita Vishwa Vidyapeetham, Amrita University, India

---

### Article Info

#### Article history:

Received Apr 14, 2021

Revised Jul 8, 2021

Accepted Jul 18, 2021

---

#### Keywords:

Faster R-CNN

Inception model

Transfer learning

---

### ABSTRACT

The most important aspect of automatic driving and traffic surveillance is vehicle detection. In addition, poor road conditions caused by potholes are the cause of traffic accidents and vehicle damage. The proposed work uses deep learning models. The proposed method can detect vehicles and potholes using images. The faster region-based convolutional neural network (CNN) and the inception network V2 model are used to implement the model. The proposed work compares the performance, accuracy numbers, detection time, and advantages and disadvantages of the faster region-based convolution neural network (Faster R-CNN) with single shot detector (SSD) and you only look once (YOLO) algorithms. The proposed method shows good progress than the existing methods such as SSD and YOLO. The measure of performance evaluation is Accuracy. The proposed method shows an improvement of 5% once compared with the previous methods such as SSD and YOLO.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



---

### Corresponding Author:

Thangavelu S

Dept of Computer Science & Engineering, Amrita School of Engineering

Coimbatore Amrita Vishwa Vidyapeetham

Amrita University, India

Email: s\_thangavel@cb.amrita.edu

---

## 1. INTRODUCTION

Economic development and social well-being are both dependent on the road system. Surveillance, intelligent traffic control, public safety and security, and autonomous driving all require vehicle detection. Due to overloaded vehicles and poor drainage systems, road maintenance is difficult. Potholes can also damage lower-riding vehicles, causing flat tyres and wheel damage. Potholes can also cause vehicle collisions and accidents. As a result, detecting potholes is one of the most essential tasks in the road management system. It will aid in the betterment of road conditions, particularly in developing countries with limited resources. This paper proposes a novel approach for detecting vehicles and potholes using deep learning.

The performance, accuracy numbers, detection time, and benefits and drawbacks about the algorithm faster region-based convolution neural network (Faster R-CNN) with the single shot detector (SSD) algorithm and you only look once (YOLO) are discussed and compared in the proposed work. Only highly reliable computing machines are suitable for algorithms like SSD and YOLO. These algorithms require more time to complete the training process. In the proposed work, the Faster-RCNN compares with SSD and YOLO algorithms. The Faster R-CNN is a convolutional neural network-based object detection algorithm that is unified, faster, and more accurate. The algorithm SSD is not suitable to identify small objects. The SSD improves detection by adding the layers of various features into this end network. And mainly when this comes to the case of smaller objects, then the SSD performs significantly worse than this Faster R-CNN. Foremost reasons of this weakness are that higher-resolution layers in the algorithm SSD are

in complete control of detecting small objects. Another drawback of the SSD object detection algorithm is that the SSD training needs a very large quantity of the data. Depending on the application, this can be costly and time-consuming.

YOLO on the other hand, provides an end-to-end network. The YOLO algorithm has a lower recall and a higher localization error than the Faster R-CNN algorithm. It also has trouble detecting close objects because each grid can only propose two bounding boxes. YOLO also has difficulty in finding small things or objects.

A lot of researchers are working on this issue. The high volume of vehicular traffic causes traffic blockage, unpleasant delays, financial loss, accidents, road damages, the passage of emergency vehicles, pollution, violations of traffic, and health concerns. There is a necessity for an enhanced system that would optimize the efficiency of traffic flow. Prasad *et al.* [1] developed a real-time model for estimating traffic density at a junction with the help of image processing methods. To estimate traffic density, the author used techniques such as local binary pattern and histogram of oriented gradient-based approaches.

Choudekar *et al.* [2] proposed a model that controls the traffic light with the help of an image processing technique. This system captures Image sequences. With the help of picture matching and a reference picture of the vacant road, the captured pictures are matched sequentially. The edge detection operator – prewitt operator has been used to perform the detection of edge. The duration of the lights is then adjusted depending upon the percentage of traffic matching.

Kastrinaki *et al.* [3] presented a summary of the image processing tool and analysis tool which are applied in the applications. The authors linked those two tools to finalize traffic-related models. Furthermore, the authors [3] have distinguished between the static and mobile camera cases. The goal of the work is original. As a result, the paper serves three purposes. In the first step [3], used image processing techniques were classified. The next step is to list the benefits as well as the drawbacks of those algorithms. And in the final and third step, an assessment of flaws is attempted from the integrated consideration.

Mednis *et al.* [4] have proposed a model which is based on an accelerometer. Algorithms such Z-DIFF, G-ZERO, and Z-THRESH are used in this accelerometer-based model. The quality of the model was evaluated using a 90 percent true positive value. For the detection of a pothole, Lin *et al.* [5] used a non-linear support vector machine classification tool. Pereira *et al.* [6] have utilized this Neural Network called CNN. Pereira *et al.* [6] showed that their proposed method gives good progress and good performance around 99.80% than the existing methods like support vector machine. Subbiah *et al.* [7] demonstrated that the you only look once the architecture and the variants of YOLO are the most efficient when evaluating metrics in a real-time system. A new image dataset has been created by Dharneshkar *et al.* [8]. The data was gathered from Indian highways. You only look once was used to annotating and training this dataset. YOLO is a phrase that means "you only look once". The mAP, precision, and recall are used to evaluate the results. The authors tested this proposed model on a variety of images containing potholes, and the results were accurate in detecting them.

Ajay *et al.* [9] use aerial images to find small vehicles in a restricted environment. The experiment makes use of the vehicle detection in aerial imagery (VEDAI) dataset. The two phases of this model's architecture are the training and detection phases. This study uses the feature extraction techniques singular value decomposition (SVD) and histogram of oriented gradients (HOG). Ajay *et al.* [9] concluded that the proposed SVD features are comparable to existing HOG features in terms of performance. Pradeep *et al.* [10] develop two improvised models for detecting vehicles during the nighttime. For this, the authors used an optical camera that is facing forward. In the case of a dual tail-light scenario, method one is used to improve accuracy. In the case of single-tail-light functionality, they use the second technique to improve accuracy. The algorithms are tested in real-life traffic scenarios.

## 2. RESEARCH METHOD

Research method for the project involves three main parts-data collection, training, and detection. The proposed vehicle and pothole detection technique detect vehicles and potholes in images with the transfer learning technique. Faster R-CNN and inception v2 models are the techniques used in the proposed model. The ultimate benefit of this proposed method is least time for training along with very high accuracy.

### 2.1. Transfer learning technique

Transfer learning is a technique that involves training a neural network model on a problem that is similar to the solved one. The layers of the trained model use a new model to train on the problem of interest. The implementation of gained knowledge while resolving one problem to another but related problem is known as the transfer learning technique. It is possible to save a significant amount of time with the transfer learning method. Any pre-trained model can be chosen and used in this method as a first step. A model that has had all of its parameters trained refers to a pre-trained model. The final step will be to fine-tune the parameters.

The new dataset fetches to fine-tune the convolutional neural networks (CNNs) [11], [12]. And if latest datasets are similar to an actual datasets, then selfsame weight is used to extract the features. Here this dataset is vastly divergent from these original datasets. Earlier layers of this CNN found a generic property. The layers of CNN [13], [14], on the other hand, become increasingly particular to the specifics of classes in the actual or real dataset. As a result, previous layers will aid in the extraction of current data features. Because of the fewer amount of data, it fixes the earlier layers. Rest of these layers are retrained.

## 2.2. Architecture of F-RCNN

The faster region-based convolutional neural network is also known as F-RCNN or Faster R-CNN. First of all, it is necessary to understand about the Fast R-CNN before moving on to faster RCNN. Fast RCNN (fast region-based CNN) is a detector. Fast region-based convolutional neural network uses external proposal. External selective search, convolutional neural network with maximum pooling, fully connected layers, region of interest pooling layer, then output layers are among the components. With the help of convolutional neural network and pooling layers - max-pooling layers, Fast RCNN receives an input image. It extracts the convolutional feature map. The region of interest pooling layer plays a salient role. These fully connected layer accepts fixed sizes only. As a result, the ROI pooling layers convert CNN's output into a predetermined size. Fast RCNN becomes faster region-based convolutional neural network when fast RCNN combines along with region proposal network. So, the main dissimilarity among fast region-based convolutional neural network and faster region-based convolutional neural network is only about the region proposal. Fast region-based convolutional neural network has an external selective search [15]. Region proposal network combines with the Architecture in faster region-based convolutional neural network. region proposal network also chooses where to look in the image to reduce computational complexity and requirements. It mainly scans this input image. It then generates k output boxes, all with two scores representing probability for an object's availability [16], [17]. Figure 1 depicts the F-RCNN architecture.

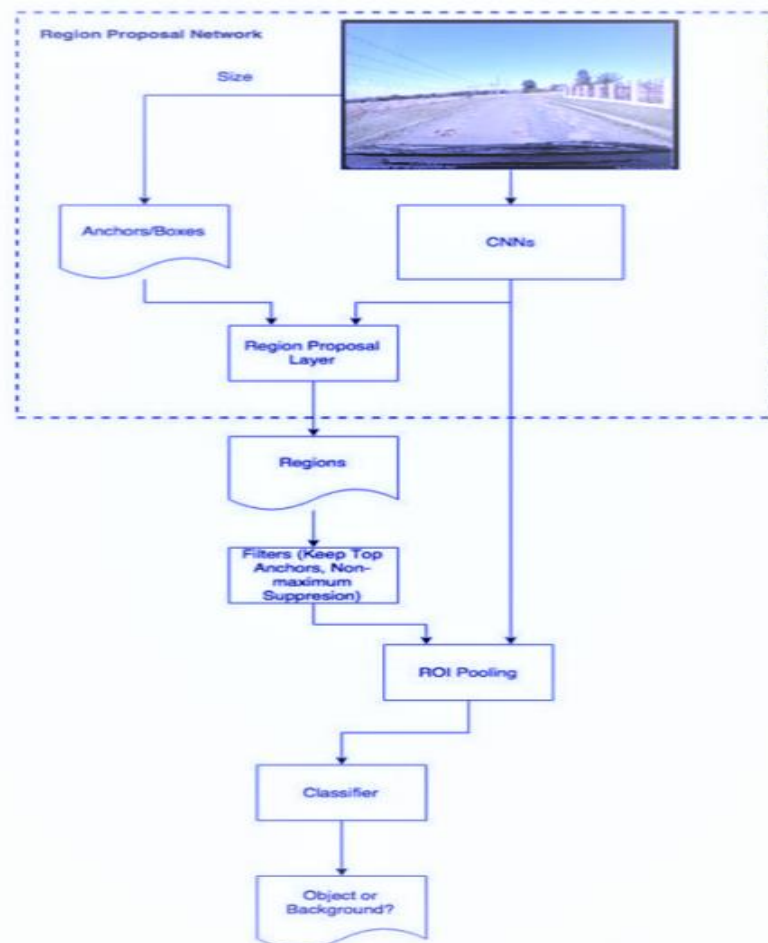


Figure 1. F-RCNN architecture

The following steps are performed to obtain the trained faster region-based convolutional neural network architecture:

- Training of RPN; To propose the expected region, the region proposal network architecture is first trained with the dataset.
- Fast R-CNN training; Faster region-based convolutional neural Network is an amalgamation of the region proposal network and the fast region-based convolutional neural network, to make [18], [19] a faster region-based convolutional neural network, a fast region-based convolutional neural network is trained with the proposals obtained by region proposal network. This is done after the training.
- Fixing convolutional layers; Region proposal network is fine-tuned for unique layers.
- Fixing convolutional layers; Fast region-based convolutional neural network's fully connected layers are then fine-tuned.

**2.3. Architecture of inception network V2**

The Inception network v2 is an improved type in the series of the inception networks. Inception network v2 is a complicated architecture of convolutional neural networks. First version of the inception network series is the Inception network v1. Convolution executes in Inception network v1. In addition, maximum pooling takes place. The outputs are all connected. The outputs are then forwarded to the next inception module. Figure 2 depicts an Inception network v1 module. Computationally, these deep learning models are expensive [20], [21]. To restrict the number of input channels includes an extra convolution. The model uses this inception module to develop the GoogLeNet neural network architecture. There are nine such inception modules in the GoogLeNet neural network architecture [22], [23]. Also, it introduces the inception v2 architecture to improve computational speed, as shown in Figure 3. Also, introduce a new method for improving performance. Filter size convolution  $n \times n$  in this model combines filter size convolution  $1 \times n$  and filter size convolution  $n \times 1$ . The model is 33% less expensive.

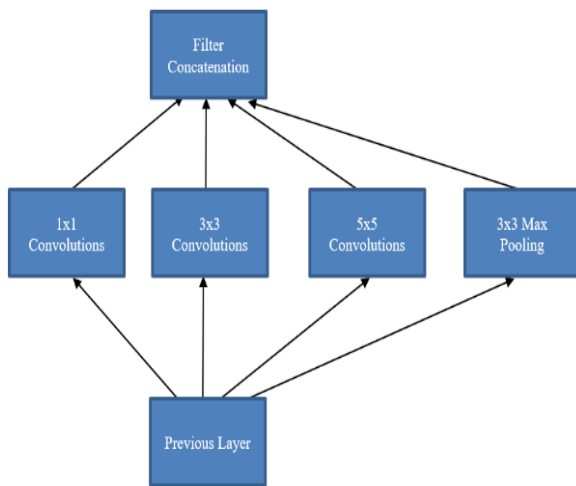


Figure 2. Inception network v1

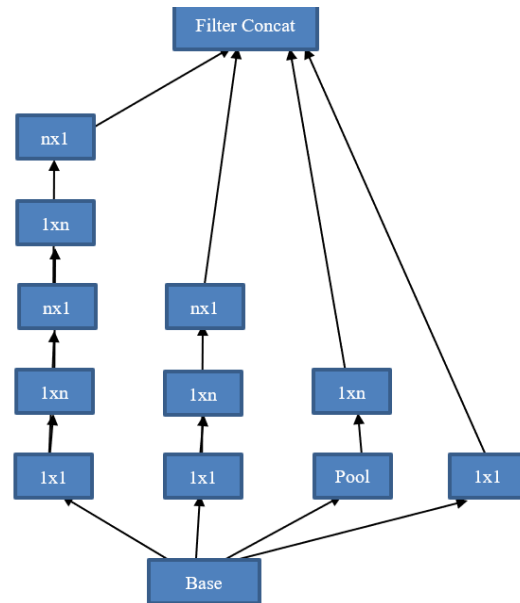


Figure 3. Architecture of inception-v2

**2.4. Dataset**

The datasets identified and used in this proposed work are, traffic surveillance dataset and the Pothole detection dataset. The Traffic Surveillance dataset is composed of 400 images that include different types of vehicles on the road. This dataset is for vehicle detection. This dataset provides images to perform vehicle detection. The Pothole detection dataset is an image dataset that contains around 665 images of roads with potholes. This dataset is from Kaggle. This dataset mainly includes images with different types of potholes on roads. This dataset is for detection models and can work as proof of concept (POC)/proof of value (POV) for the maintenance of roads. For this project, the necessary data and images of both vehicles and potholes are gathered from various sources.

### 2.5. Data pre-processing

There are over a thousand images in the dataset. The vehicle detection and pothole detection tasks consider as uncomplicated only if region in those pictures containing road has cropped. Later it is utilized in the detection. Also, it requires extra time in order to crop one thousand or more images. All of the images are labeled. Then a comma-separated values (CSV) file is created. This CSV file includes the name of the image, path, dimension of the image, and coordinate of the vehicles and potholes. Then TfRecords for the dataset is created. It feeds and fine-tunes the Inception model. The image size is 3680x2760 pixels.

### 2.6. Detection of vehicles and potholes

After the pre-processing of the data, a model to detect vehicles and potholes has created. The model has fine-tuned using the Inception network v2 architecture [24], [25]. Training requires a configuration file. There are nine different classes in this configuration file. Ambulance, auto, bus, car, motorcycle, pothole, potholeGroup, truck, and van are the nine classes. The proposed system will take some time to train model, depending on the hardware configuration. Once the training completes, the testing process begins. During testing, the model performed admirably. It can, however, be improved.

## 3. RESULTS AND DISCUSSION

The result analysis of an algorithm or model relies on a set of parameters. The majority of the evaluation depends on parameters. They are accuracy, detection time, performance, and other parameters. The percentage of the desired result produced by the algorithm is one such important parameter that measures the model's or algorithm's promising element. Another parameter is the time taken which illustrates the time taken by the model or algorithm to give the outcome. Then the performance is also one such important parameter that determines how well the model or algorithm performs. Figure 4 and Figure 5 show the outcomes of the testing process.

The object detection metrics [26] are precision and recall, intersection over union (IoU), average precision (AP), and mean average precision (mAP). Precision is a metric used for determining whether a prediction is correct or not. From all of the correctly predicted data, the recall gives the actual predictions. The IoU metric can be used to find differences between ground truth annotations and predicted bounding boxes. The mean AP or mAP provides precision for the entire model. Here mAP is used to obtain the percentage of correct predictions in the system.

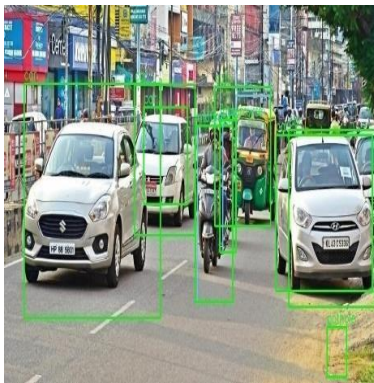


Figure 4. Detection of vehicles

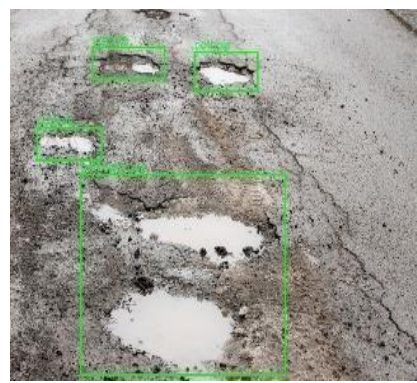


Figure 5. Detection of pothole

The mean AP is the average value of AP. Existing models such as SSD and YOLO had mAP values of 78.70 percent and 81.71 percent, respectively, based on the calculation of mean average precision. And the proposed model showed an mAP value of 86.41%. There has been an increase of around 5% in mean average precision for this proposed system. Table 1 shows a comparison about the performance of the model.

When the most common and widely used parameters are applied to object detection algorithms, the results show that Faster R-CNN is significantly faster than the SSD and YOLO algorithms. The proposed regions are taken into account by an independent network in the Faster R-CNN algorithm. Other methods, such as SSD and YOLO, are slower and slower than the Faster R-CNN method. Table 2 represents a comparison of these three methods or models in terms of detection time taken in ms. It is clear that Faster R-CNN algorithm outperforms the other two algorithms from Table 2.

Figure 6 compares the accuracy numbers for different object sizes, such as a small object, medium object, and large object, for the three algorithms: Faster R-CNN, YOLO, and SSD. From Figure 6, it is evident that when it comes to the case of smaller objects, then the algorithm SSD performs significantly worse than this Faster R-CNN. Ultimate reason for such weakness is that the higher-resolution layers of SSD have complete control over small object detection. Another drawback of this method is that the training of SSD necessarily needs a very large quantity of data. Depending on the application, this can be costly and time-consuming. YOLO also struggles to detect smaller objects. As a result, the Faster R-CNN algorithm performs much better than SSD and YOLO in terms of accuracy numbers. Hence the Faster R-CNN is identified as the best of three algorithms.

Table 1. Performance comparison

No.	Model	mAP (%)
1	Existing Model - SSD	78.70
2	Existing Model – YOLO	81.71
3	Proposed Model – Faster R-CNN	86.41

Table 2. Detection time taken

No.	Model	Detection Time Taken (ms)
1	SSD	2.9 ms
2	YOLO	2.5 ms
3	Faster R-CNN	2 ms

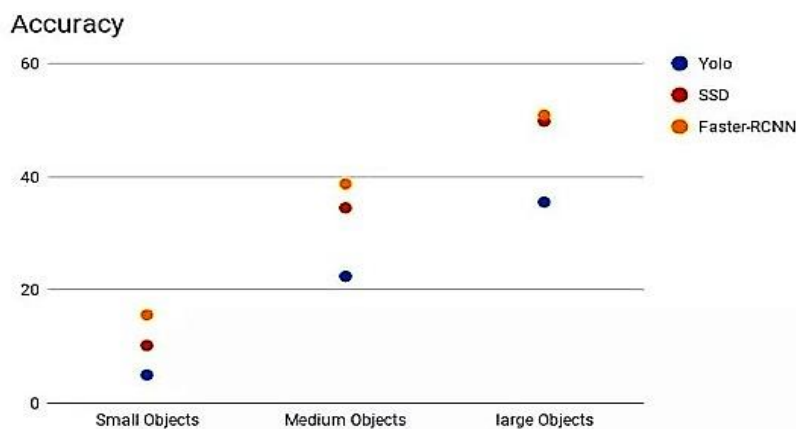


Figure 6. For different sizes: faster R-CNN vs SSD vs YOLO

#### 4. CONCLUSION

This paper proposes a new model for detecting vehicles and potholes in images. The proposed model employs the convolutional neural network's inception network v2. The results of the experiments show that the proposed method outperforms others. The model accurately detects vehicles and potholes. And in the proposed work, the performance, accuracy numbers, detection time, and advantages and disadvantages of Faster R-CNN with SSD algorithm and YOLO are identified, discussed, and compared.

#### REFERENCES

- [1] D. Prasad, K. Kapadni, A. Gadpal, M. Visave, and K. Sultanpure, "HOG, LBP and SVM based Traffic Density Estimation at Intersection," *IEEE Pune Section International Conference (PuneCon)*, Pune, India, pp. 1-5, 2020, doi: 10.1109/PuneCon46936.2019.9105731.
- [2] P. Choudekar, S. Banerjee, and M. K. Muju, "Implementation of Image Processing in Real Time Traffic Light Control," *3rd International Conference on Electronics Computer Technology*, Kanyakumari, India, 2011, pp. 94-98.
- [3] V. Kastinaki, M. Zervakis, and K. Kalaitzakis, "A Survey of Video Processing Techniques for Traffic Applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359-381, 2003, doi: 10.1016/S0262-8856(03)00004-0.
- [4] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," *International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, Barcelona, Spain, 2011, pp. 1-6, doi: 10.1109/DCOSS.2011.5982206.
- [5] J. Lin and Y. Liu, "Potholes Detection Based on SVM in the Pavement Distress Image," *Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, Hong Kong, China, 2010, pp. 544-547, doi: 10.1109/DCABES.2010.115.
- [6] V. Pereira, S. Tamura, S. Hayamizu and H. Fukai, "A Deep Learning-Based Approach for Road Pothole Detection in Timor Leste," *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, Singapore, 2018, pp. 279-284, doi: 10.1109/SOLI.2018.8476795.

- [7] U. Subbiah, D. K. Kumar, S. K. Thangavel, and L. Parameswaran, "An Extensive Study and Comparison of the Various Approaches to Object Detection using Deep Learning," *Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2020, pp. 183-194, doi: 10.1109/ICSSIT48917.2020.9214185.
- [8] J. Dharneeshkar, V. Soban Dhakshana, S. A. Aniruthan, R. Karthika and L. Parameswaran, "Deep Learning based Detection of potholes in Indian roads using YOLO," *International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 2020, pp. 381-385, doi: 10.1109/ICICT48043.2020.9112424.
- [9] A. Ajay, V. Sowmya, and K. P. Soman, "Vehicle detection in aerial imagery using eigen features", *International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2018, pp. 1620-1624, doi: 10.1109/ICCSP.2017.8286664.
- [10] C. S. Pradeep and R. Ramanathan, "An improved technique for Night-time Vehicle detection," *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, India, 2018, pp. 508- 513, doi: 10.1109/ICACCI.2018.8554712.
- [11] S. Ren, K. He, R. Girshick and J. Sun, "Faster RCNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-14, 2017, doi: 10.1109/TPAMI.2016.2577031
- [12] N. F. Sahidan, A. K. Juha, and Z. Ibrahim, "Evaluation of basic convolutional neural network and bag of features for leaf recognition," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 1, pp. 327-332, 2019, doi: 10.11591/ijeecs.v14.i1.pp327-332.
- [13] S. Shindo, T. Goto, T. Kirishima, and K. Tsuchida, "An Optimization of Facial Feature Point Detection Program by Using Several Types of Convolutional Neural Network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 827-834, 2019, doi: 10.11591/ijeecs.v16.i2.pp827-834.
- [14] M. Z. Osman, M. A. Maarof, M. F. Rohani, and N. N. A. Sjarif, "A multi-color based features from facial images for automatic ethnicity identification model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, pp. 1383-1390, 2020, doi: 10.11591/ijeecs.v18.i3.pp1383-1390.
- [15] P. Patel and A. Thakkar, "The upsurge of deep learning for computer vision applications," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 1, pp. 538-548, 2020, doi: 10.11591/ijeecs.v10i1.pp538-548.
- [16] R. Rasetiadi and S. Suharjito, "Foreign exchange prediction based on indices and commodities price using convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, pp. 494-501, 2019, doi: 10.11591/ijeecs.v18.i1.pp494-501.
- [17] N. N. A. Hamid, R. A. Razali, and Z. Ibrahim, "Comparing bags of features, conventional convolutional neural network and alexnet for fruit recognition", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, pp. 333-339, 2019, doi: 10.11591/ijeecs.v14.i1.pp333-339.
- [18] H. F. Pardede, A. R. Yuliani, and R. Sustika, "Convolutional Neural Network and Feature Transformation for Distant Speech Recognition," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 8, no. 6, p. 5381, 2018, doi: 10.11591/ijeecs.v8i6.pp5381-5388.
- [19] K. S. Gautam and S. K. Thangavel, "Video analytics based intelligent surveillance system for smart buildings," *Springer Soft computing*, vol. 23, no. 8, pp. 2813-2837, 2019, doi: 10.1007/s00500-019-03870-2.
- [20] T. Arathi and L. Parameswaran, "Slantlet Transform and Phase Congruency based Image Compression," *International Journal of Computer Applications (IJCA)*, *Proceedings on Amrita International Conference of Women in Computing*, 2013, pp. 12-16.
- [21] L. Suhao, L. Jinzhao, L. Guoquan, B. Tong, W. Huiqian, and P. Yu, "Vehicle type detection based on deep learning in traffic scene," *Procedia Computer Science*, pp. 564-572, 2018, doi: 10.1016/j.procs.2018.04.281.
- [22] W. Lumchanow and S. Udomsiri, "Image classification of malaria using hybrid algorithms: convolutional neural network and method to find appropriate K for K-Nearest neighbor," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 1, pp. 382-388, 2019, doi: 10.11591/ijeecs.v16.i1.pp382-388.
- [23] P. S. Gupta, P. Sharma, D. Sharma, and C. Gupta, "Detection and localization of potholes in thermal images using deep neural networks," *Multimedia Tools and Applications*, Springer, vol. 79, pp. 35-36, 2020, doi: 10.1007/s11042-020-09293-8.
- [24] D. A. Jasm, M. M. Hamad and A. T. H. Alrawi, "Deep image mining for convolution neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 347-352, 2020, doi: 10.11591/ijeecs.v20.i1.pp347-352.
- [25] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Rice false smut detection based on Faster R-CNN," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1590-1595, 2020, doi: 10.11591/ijeecs.v19.i3.pp1590-1595.
- [26] M. A. Anwer, S. M. Shareef, and A. M. Ali, "Accident vehicle types classification: a comparative study between different deep learning models," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1474-1484, 2021, doi: 10.11591/ijeecs.v21.i3.pp1474-1484.