



An Agri vigilance System based on Computer Vision and Deep Learning

Karan Owalekar¹, Junaid Shaikh², Meghna Daftary³, Megha Gupta⁴

^{1,2,3,4}Department of Computer Engineering, NHITM, Thane, India

karanowalekar270399@gmail.com¹, junaid.shaikh.980@gmail.com², meghna.daftary@gmail.com³, meghagupta@nhitm.ac.in⁴

How to cite this paper: K. Owalekar, J. Shaikh, M. Daftary and M. Gupta (2021) An Agri Vigilance System based on Computer Vision and Deep Learning. *Journal of Informatics Electrical and Electronics Engineering*, Vol. 02, Iss. 02, S. No. 017, pp. 1-8, 2021.

<https://doi.org/10.54060/JIEEE/002.02.017>

Received: 07/04/2021

Accepted: 27/05/2021

Published: 09/06/2021

Copyright © 2021 The Author(s).

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In an agricultural-based country like India, farming and farming activities play a vital role in the growth of the economy as it is the main source of GNI (Gross National Income). This dependence of GNI on agriculture makes it important to address the issues faced by the farmers. The main area of concern for farmers revolves around crops and livestock. Precise farming techniques like cattle counting and crop disease detection are the need of the hour. The introduction of computer vision and deep learning has enabled us to make improvements in farming techniques. To accomplish this, a computer vision-based system is proposed which will be implemented using ResNet and YOLOv3-tiny. The proposed system will take images and videos as input and run them on the inference. The output will be updated in the database and the farmer will be notified in case of any inconsistency. The detailed report can be accessed by government agencies. The system will increase efficiency in farming processes like crop monitoring, livestock tracking, crop disease detection by providing fast and efficient solutions for the problems faced by the farmers.

Keywords

Precise farming, crop disease detection, livestock monitoring, ResNet, YOLOv3-tiny, Smart Agriculture

1. Introduction

Our civilization started with agriculture, and it continues to play a significant role in providing humans with basic necessities like food, fodder, and raw materials. Agriculture has been instrumental in sustaining and driving the country's economy as it provides employment opportunities and helps in rural development. Especially in an agricultural-driven country like India, farming is the backbone of India's economic system, where about 60% of the population depend directly on farming as a livelihood. This dependence of national income on cultivation necessitates addressing the problems faced by the farmers. Issues such as an increase in food prices, malicious activity on the farm, and no significant development in crop production have come to light.



The two principal problems that demand urgent attention are crop disease detection and cattle monitoring. Diseased crops can drastically reduce the quality and quantity of crops. If this continues on a larger scale, it could lead to scarcity of food. Therefore, it becomes vital to identify unhealthy plants at the initial stages to save the yield from ruining. At present, the crops are manually checked periodically for the presence of diseases. This task is time-consuming and can prove inefficient if not done properly. The aim is to automate the process of crop disease detection. The objective is to reduce manual labour by leveraging computer vision and deep learning technologies in farming activities. Hence, we propose a Resnet based architecture to detect and identify some common crop diseases with near-perfect accuracy [1][2].

Furthermore, considering that there has been an increase in the illegal theft or killing of animals, the farmers need to know the number of cattle grazing in their field. By keeping an accurate count of the cattle helps the farmer to know whether their cattle are within the boundaries of the farm or not. The counting of individual animals manually over acres of farmlands is difficult and prone to human errors. The proposed system uses the YOLOv3-tiny model, which will help farmers monitor their livestock and keep track of the number of cattle on their farms [13-22]. There will be a notable enhancement in farm productivity and improvement in the overall efficiency of agricultural practices by catering to these problems.

2. Literature Survey

Custom Convolution Neural Network (CNN) for classifying the plant into diseased or healthy based on the leaf image has been proposed by Trivedi J. et.al [1]. They used a dataset that consists of 54,305 images which are divided into 15 different classes of healthy and diseased plant leaf images. The CNN architecture is used consisting of several convolution layers along with pooling layers and dropout layers. A fully connected layer is then stacked at the end of the architecture that does the final classification. The model achieved an accuracy of 95.81%. V. Kumar et.al. have used residual network- 'RESNET-34' architecture, a type of CNN used to classify a plant disease based on a leaf image. RESNET-34 is a pre-trained model that is trained on the ImageNet dataset. Transfer learning is done on this network, retraining it on a dataset consisting of 15200 images covering 14 unique crops and divided into 38 different classes. The data is derived from - Original Plant Village Dataset. Using the proposed technique, they have achieved an accuracy of 99 percent.[2]

In [3], the authors have compared AlexNet and SqueezeNet architecture for the detection of diseases in Tomato plants. They have chosen 10 different classes for their study from the 'Plant village dataset'. Both the CNN architecture performed well, although the performance of Alexnet was slightly better. Further, they suggested that these neural networks can be added to the robot [4] for real time disease detection capability.

To increase the prediction accuracy, we need to have a robust network so that network could learn various features and classify accordingly. But training deep neural networks sometimes lead to vanishing gradient problems due to which the network may perform poor. In [5], the study has been done to address the degradation problem due to vanishing gradient. It has been suggested that the shortcut connections can be introduced in the network which will enable the network to retain the information and hence can perform well. These shortcuts are essentially Identity mapping and introducing them in the network does not increase any extra parameter.

Jayme Garcia Arnal Barbedo, Luciano Koenigkan, and Andrea Roberto Bueno Ribeiro have proposed a four-step approach to count the cattle. Steps are CNN to delineate the region of interest, colour transformation to segment animals, cluster separation, and image matching to compensate for image overlap. The results showed that the calves were 99% correctly detected when they were alone, but in a cluster, i.e., when the calves were either touching one another or shadowing the other, the results dropped to 71%.[6] Wen Shao, Kawakami Rei, Ryota Yoshihashi, and Shaodi You have used a CNN-based system called YOLOv2 which is a fast and precise CNN to detect the number of cattle in different images. They have merged all the images to estimate the total count. They utilize a pipeline and an algorithm to merge detection results using a



three-dimensional model reconstructed by SfM to avoid double count. The estimated cattle count was +3 to the true value of dozens of cattle.[7]

Gemert, Jan & Verschoor, Camiel & Mettes, Pascal & Epema, Kitso & Koh, Lian & Wich, Serge defined two tasks: i) animal detection and ii) animal counting. To evaluate these tasks, they manually recorded and annotated a new dataset with a quad-copter drone. The animal detection task was benchmarked with three lightweight object detection algorithms that are suitable for on-board implementation since their potential detection speed is less than 1 second per image. The animal counting was based on frame-based detections in combination with point tracks obtained by a KLT tracker. [8]

In [9], The authors presented their research on using deep learning to detect cattle. They put 15 different CNN architectures to the test, looking at things like low illumination, excessive brightness, blur, and so on. NasNet Large produced the best results in terms of accuracy, with the Xception architecture producing very high accuracies with a fraction of NasNet's training times. They suggested using YOLO to draw a bounding box around the subject as the future work.

In [10], The authors evaluated the performance of Faster RCNN, YOLOv3, and SSD for cattle detection and counting. Across a range of scenarios, average precision scores of 86%, 91%, 95% of bounding box and 84%, 90%, 94% for mask, as well as counting accuracy of 90%, 92%, 94% and recall of 91%, 95%, 96% were detected. The results presented indicate that Mask R-CNN could be utilized in practical settings as a method of livestock detection and counting using a quadcopter.

It can be observed that in the case of crop diseases, Resnet based architecture has proved to be very efficient for achieving more than the desired accuracy. However, the architecture is a bit computationally heavy. The network size can be decreased by using various optimization techniques like pruning and at the same time preserving its robustness. Additionally, the systems for monitoring the cattle proposed by various researchers used CNN-based models which delivered good results, but performance decreased in case of huge clusters or when motion was detected.

3. Problem Definition

Agriculture in India faces many challenges in terms of productivity, reliable detection and accurate identification of crop diseases, and labour shortage. The system will detect and identify the diseased crops, and the user will be notified via a message. The system will keep an accurate count of the livestock to prevent illegal killing or theft of cattle of particular farmland. The aim is to automate and increase the efficiency of the agriculture processes.

4. Proposed System Overview and Methodology

The proposed system will strengthen the agricultural practices by eliminating the problems faced by the farmers - monitoring the health of the crop and cattle management. The model will function in two modes - plant disease detection and cattle monitor. The user will have to select the mode for which they want the information. According to the mode selected, the pre-processed image will be passed either through the disease detector or cattle count detector. In crop disease detection mode, the images of the leaves will be taken as input. It will give the type of disease as output if any. In cattle count mode, the system will give the count of the cattle as output from the input images.

The disease detector is based on ResNet, which is a CNN-based architecture. Unlike normal CNN, it contains residual blocks that help in retaining the information of the image when passed through the network. ResNet is a model that is pre-trained on ImageNet Database. A pre-trained model helps get higher accuracy by just using a small amount of data, which saves time. In disease detector mode, the pre-processed image will undergo segmentation. The region of interest (ROI) from the segmented image will be localized. The final step is to classify whether a plant is healthy or not. If the plant is infected, then it will identify the type of disease.

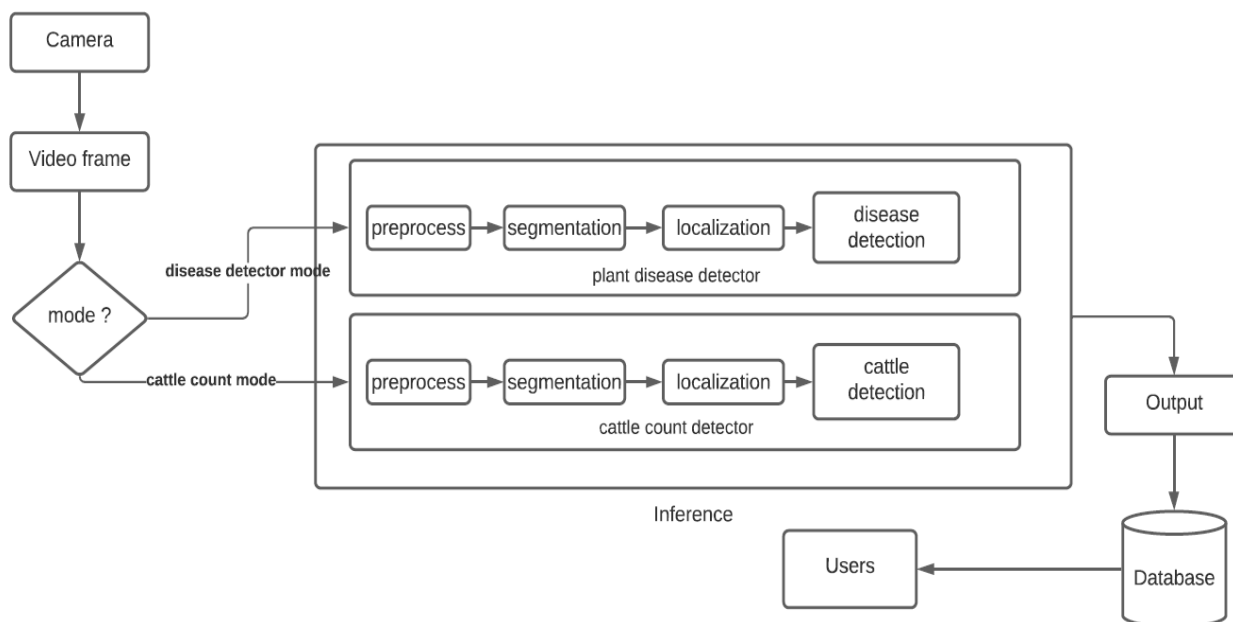


Figure. 1 Proposed System Architecture

For plant disease classification, we have used the “New plant village dataset” that contains 87,867 RGB images of healthy and diseased crop leaves which are categorized into 38 different classes. This dataset is divided into an 80/20 ratio of training and testing sets. Various augmentation has been done on this data before feeding it to the model to make the model more robust. The model which we have used is Resnet as we can see in figure 2. Resnet is a state-of-art residual neural network that has proved to be very efficient in computer vision and object detection tasks. A residual block makes the use of skip connection to addresses the vanishing gradient problem. This shortcut guarantees the network integrity if the regular connection’s co-efficient converges to zero.

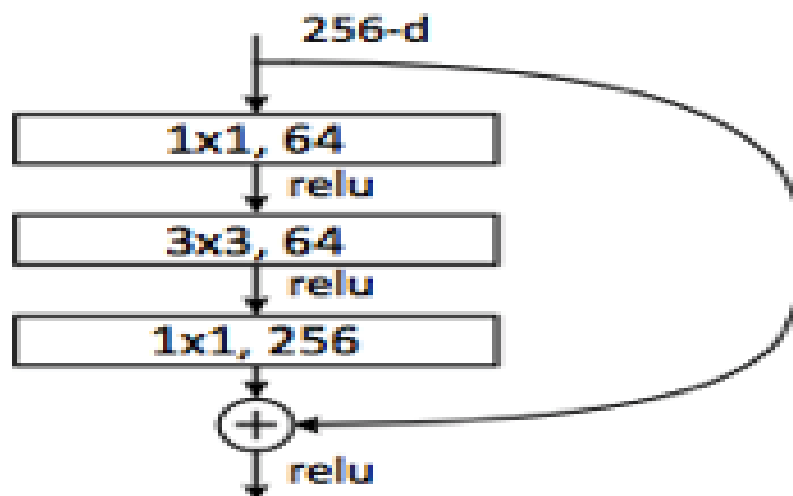


Figure. 2 Resnet 50 Residue Block

Transfer learning is used for training the model for our use-case. Using a pre-trained model is beneficial because it saves a lot of training time. Pre-trained Resnet architecture trained on ImageNet - which is a huge dataset comprising of different image categories, is taken as starting base model. This is because we wanted the model to retain its feature extraction knowledge which it has learned on the ImageNet dataset. At the end of model architecture various layers such as convolution layer, batch normalization, and finally fully connected dense layers are stacked to finetune it for our purpose. After training the model is integrated into the system for inferencing. An image of plant leaves is to be fed to the system. This image will be reshaped into 256*256 pixels and given to the model for classification. The model gives out the class of images. The cattle count detector is implemented using YOLOv3. The architecture of YOLOv3 can be seen in figure 3 below. YOLOv3-tiny is used because it gives better accuracy compared to other algorithms. In cattle count mode, the pre-processed image will undergo segmentation. The next step is feature extraction for cattle detection. After that, the inference will run, and it will give bounding boxes for the cattle. Finally, the cattle will be counted based on the number of bounding boxes.

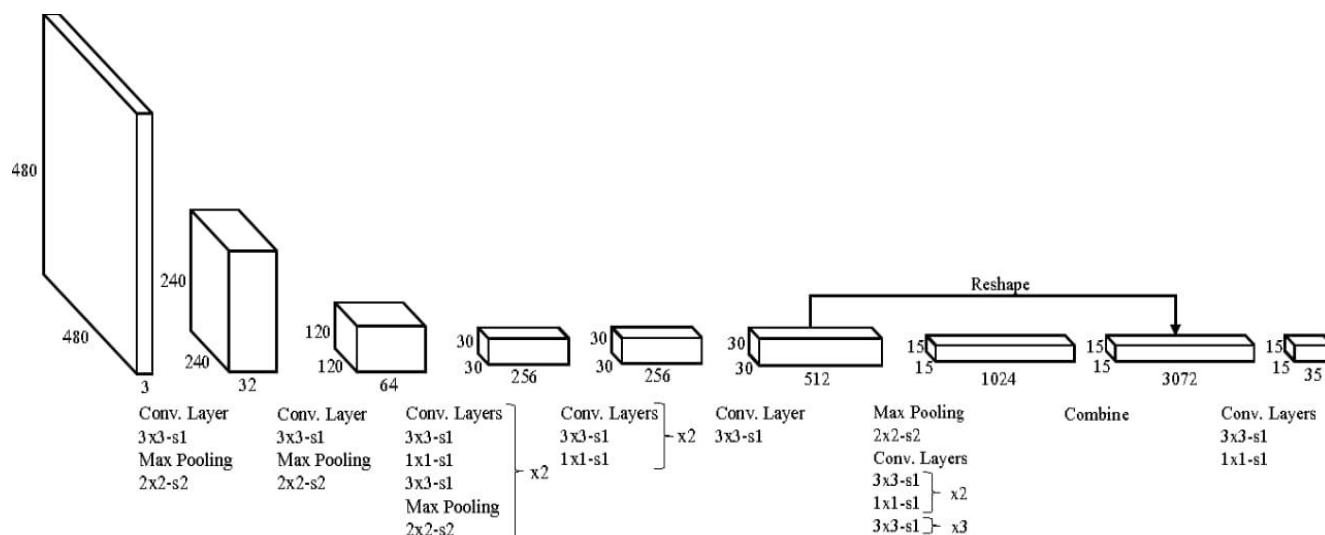


Figure. 3 YOLOv3 Architecture

We have created our own dataset consisting of around 50 images that contain plenty of cattle. We are considering four types of cattle i.e., cows, buffalos, sheep, and camels. Individual animal from each image is extracted and labelled for training. This dataset is randomly divided into train and test in the ratio of 90/10. For training, we will be using the YOLOv3-Tiny model - a variant of Darknet. YOLO is an object detection algorithm that is fast and accurate as a Single Shot MultiBox (SSD). We are using YOLOv3-Tiny and not YOLOv3 because it requires less computation, and it is good for embedded devices like Raspberry-pi. It applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. We retrain the already trained model for our data. After every 1000 epoch of training weights will be updated and saved into the file. After training, the model is integrated into the system for inferencing. An image or video frame can be passed to the model where the segmentation will take place. The model will divide images into different segments of images. These individual segments will be labeled into a class. The model then returns a plethora of bounding boxes that are then reduced through Non-max Suppression, a type of filtering. This results in fairly accurate detection of different classes. As an output, we will get all the detected objects along with the class labels, confidence score and location of the bounding box in a form of a list. The resultant data from either mode will then be updated into the database. The user will request the database for the required data, and the database will respond to the user request.

5. Result and Analysis

Table 1 Model Summary

Parameters	Value
Total parameters	48,868,305
Trainable parameters	25,255,469
Non-trainable parameters	23,612,836
Learning rate	0.001 to 0.0001

For Plant Disease detection, our model gave almost 99% accuracy. The model summary is given in Table 1 above. Previously, many authors have got similar accuracy, but none of them have used augmented data. Our training method used artificially augmented data that makes our model ready to use on real-world images. Machine learning and deep learning models are as good as the data on which they are trained and for their performance, they require the type of data on which they are trained on. The dataset which we have used contains a single leaf image in each image file, and that too on the background which is easily distinguishable from the foreground object (i.e., leaf). But in the real world, we can't expect to get such an image. Our use case is images from drones i.e., the system will be getting images containing several leaves with a background containing some more vegetation. To make the system work with such images, we artificially augmented the images to produce similar conditions and used those images to train our model. The proposed system will be able to work in the real world reliably. We already got the validation accuracy of 99% i.e., similar to that of training accuracy, and several tests on real-world images also show a good prediction as seen in figure 4.

Categories	total images	predicted
Apple__Apple_scab	489	484
Apple__Black_rot	497	463
Apple__Cedar_apple_rust	370	363
Apple__healthy	502	495
Blueberry__healthy	454	451
Cherry_(including_sour)__Powdery_mildew	401	398
Cherry_(including_sour)__healthy	455	444
Corn_(maize)__Cercospora_leaf_spot Gray_leaf_spot	410	409
Corn_(maize)__Common_rust	477	472
Corn_(maize)__Northern_Leaf_Blight	477	471
Corn_(maize)__healthy	465	465
Grape__Black_rot	472	466
Grape__Esca_(Black_Measles)	420	400
Grape__Leaf_blight_(Isariopsis_Leaf_Spot)	430	419
Grape__healthy	423	422
Orange__Haunglongbing_(Citrus_greening)	484	453
Peach__Bacterial_spot	459	457
Peach__healthy	432	427
Pepper,_bell__Bacterial_spot	478	477
Pepper,_bell__healthy	497	473
Potato__Early_blight	416	405
Potato__Late_blight	485	477
Potato__healthy	456	448
Raspberry__healthy	445	438
Soybean__healthy	505	500
Squash__Powdery_mildew	434	427
Strawberry__Leaf_scorch	444	442
Strawberry__healthy	445	421
Tomato__Bacterial_spot	425	421
Tomato__Early_blight	443	429
Tomato__Late_blight	459	457
Tomato__Leaf_Mold	470	467
Tomato__Septoria_leaf_spot	436	434
Tomato__Spider_mites Two-spotted_spider_mite	435	433
Tomato__Target_Spot	407	394
Tomato__Tomato_Yellow_Leaf_Curl_Virus	490	487
Tomato__Tomato_mosaic_virus	448	447
Tomato__healthy	471	461

Figure. 4 Validation result of crop disease detection

For cattle count detection, our main task was to correctly detect the position, height and width of the box around the animal present in the image. For this, our model was correctly able to detect these parameters with an accuracy of 73.2319%. Previously, authors narrowed down their focus on counting only one type of animal, but we have considered 4 types of animals where along with good detection accuracy, we can classify the detected animal and provide the count of each animal present. Other studies show that researchers used a four-step-approach to count cattle with a large dataset which gave them good accuracy in detecting the cattle and non-cattle. With our model, we got close enough results in detecting the animal from its surrounding with little false positives after training on an extremely small-sized custom dataset as seen in figure 5.



Figure.5 Cattle detection

Our model was able to process 6-7 images/frames per second with an estimated cattle count of ± 3 to the true value per dozen when the animals were scattered around in the field. The error rate increases when cattle were close to each other i.e., when animals in the image were overlapping each other. The analysis showed that our single model performed better than most of the previous works even against all of them combined.

6. Conclusion

The proposed system shows how computer vision and deep learning can help eliminate the problems faced by farmers by monitoring their livestock and crop health. The system will enhance agricultural productivity and improve the overall efficiency of agricultural practices. The farmers will be able to receive updates regarding the inconsistencies in their crops or cattle count by using this system. The system will serve as an aid to farmers enabling fast and efficient recognition of plant diseases and assist in keeping track of their cattle. The government agencies will be able to get regular updates regarding the crop health data, which was traditionally done by visiting the farm once a year.

References

- [1]. J. Trivedi, Y. Shamnani, and R. Gajjar, "Plant leaf disease detection using machine learning," in Communications in Computer and Information Science, Singapore: Springer Singapore, pp. 267–276, 2020.
- [2]. V. Kumar, H. Arora, Harsh, et al., "ResNet-based approach for detection and classification of plant leaf diseases," in 2020 International

- al Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 495-502, 2020.
- [3]. H. Durmus, E. O. Gunes, and M. Kirci, "Disease detection on the leaves of the tomato plants by using deep learning," in 6th International Conference on Agro-Geoinformatics, 2017.
- [4]. H. Durmuş, E. O. Güneş and M. Kırıcı, "Data acquisition from greenhouses by using autonomous mobile robot," *Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, pp. 1-5, 2016.
- [5]. K. He, X. Zhang, S. Ren, et al., "Deep residual learning for image recognition," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770-778, 2016.
- [6]. J. G. A. Barbedo, L. V. Koenigkan, P. M. Santos et al., "Counting cattle in UAV images—dealing with clustered animals and animal/background contrast changes," *Sensors (Basel)*, vol. 20, no. 7, p. 2126, 2020.
- [7]. W. Shao, R. Kawakami, R. Yoshihashi, et al., "Cattle detection and counting in UAV images based on convolutional neural networks," *Int. J. Remote Sens.*, vol. 41, no. 1, pp. 31–52, 2020.
- [8]. J. C.V Gemert, C. R. Verschoor, P. Mettes, et al., "Nature conservation drones for automatic localization and counting of animals," in *Computer Vision – ECCV Workshops*, Cham: Springer International Publishing, pp. 255–270,2015.
- [9]. J. G. A. Barbedo, L. V. Koenigkan, T. T. Santos, et al., "A study on the detection of cattle in UAV images using deep learning," *Sensors (Basel)*, vol. 19, no. 24, 2019.
- [10]. B. Xu, W. Wang, G. Falzon, et al., "Automated cattle counting using Mask R-CNN in quadcopter vision system," *Comput. Electron. Agric.*, vol. 171, no. 105300, 2020.
- [11]. P. Singhal, P. Singh, and A. Vidyarthi, "Interpretation and localization of Thorax diseases using DCNN in Chest X-Ray," *Journal of Informatics Electrical and Electronics Engineering*, vol.1, no.1, pp.1-7,2020.
- [12]. M. Vinny, & P. Singh, "Review on the Artificial Brain Technology: Blue Brain," *Journal of Informatics Electrical and Electronics Engineering*, vol.1, no.1, pp.1-11,2020.
- [13]. A. Singh and P. Singh, "Object Detection," *Journal of Management and Service Science*, "vol.1, no.2, pp. 1-20,2021
- [14]. A. Singh, & P. Singh, "Image Classification: A Survey," *Journal of Informatics Electrical and Electronics Engineering*, vol.1, no.2, pp.1-9,2020.
- [15]. A. Singh and P. Singh, "License Plate Recognition," *Journal of Management and Service Science*, vol.1, no.2, pp. 1-14,2021.
- .