An Intelligent Road Lane Monitoring System Using Computer Vision

Vanmathi V M
Department of Mathematics
Amrita School of Physical Sciences
Coimbatore
Amrita Vishwa Vidyapeetham
Tamil Nadu, India
vanmathivijay2003@gmail.com
cb.sc.i5das21165@cb.amrita.edu

Aakash P
Department of Mathematics
Amrita School of Physical Sciences
Coimbatore
Amrita Vishwa Vidyapeetham
Tamil Nadu, India
aakash.padmanaban23@gmail.com
cb.sc.i5das21101@cb.students.amrita.edu

Gokul Balajiram
Department of Computer Science and Engineering,
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu, India
cb.en.u4cse22216@cb.students.amrita.edu

Abhishek R
Department of Computer Science and Engineering
Amrita School of Engineering Coimbatore
Amrita Vishwa Vidyapeetham
Tamil Nadu, India
cb.en.p2aie21001@cb.students.amrita.edu

Senthil Kumar Thangavel
Department of Computer Science and Engineering
Amrita School of Computing, Coimbatore
Amrita Vishwa Vidyapeetham, India
t_senthilkumar@cb.amrita.edu

Somasundaram K*
Department of Mathematics
Amrita School of Physical Sciences
Coimbatore
Amrita Vishwa Vidyapeetham
Tamil Nadu, India
s_sundaram@cb.amrita.edu

Selvanayaki Kolandapalayam Shanmugam
Department of Mathematics and Computer Science
Ashland University
USA
skolanda@ashland.edu

Abstract—*Effective lane detection and collision avoidance are vital for the safety of autonomous vehicles, especially in challenging environments where lane markings may be damaged or missing. This paper presents a novel system that processes live camera feeds to tackle these issues, aiming to enhance road safety. This approach utilizes real-time image analysis to identify lane boundaries and detect potential collision threats from pedestrians, animals, and flying objects. A confusion matrix is employed for robust model evaluation, ensuring reliable performance. We also incorporate advanced object detection techniques, such as YOLOv5(You Look Only Once), to classify objects and analyze their movement patterns. Additionally, by monitoring activities on sidewalks, we can better assess the randomness of nearby movements and their implications for lane detection, by integrating sophisticated image processing with predictive algorithms, our system not only addresses ambiguities in lane detection but also identifies potential collision risks in real time. Our results show improved accuracy and fewer errors compared to existing methods, making this system a significant contribution to the field of autonomous driving***

***Keywords—Lane Detection, Collision Avoidance, YOLOv5, Optical Flow, Predictive Modelling, Anomaly Detection, HSL (Hue, Saturation and Brightness) conversion, HSV (Hue, Saturation and Value) conversion, Canny Edge Detection, Hough Transform*.**

I. INTRODUCTION

With the advent of autonomous driving technology, people can largely get rid of the safety problems caused by daily manual driving. Therefore, self-driving cars are sought after by many automobile consumers. The increasing demand for autonomous driving systems has intensified the need for reliable lane detection [1] [2] and collision avoidance mechanisms. Traditional lane detection systems using ML [3]often rely on clearly defined road markings, but real-world conditions frequently present challenges [4] such as faded, worn-out, or missing lanes. This paper proposes a new approach that addresses these limitations by introducing an adaptive lane-detection algorithm complemented by object detection and tracking to ensure safe vehicle navigation even in ambiguous driving conditions.

Our contributions include:

- A robust lane detection algorithm [5] that predicts lane positions even in the absence of clearly defined markings.
- An enhanced object recognition model using YOLOv5 for identifying potential collision objects such as vehicles, pedestrians, and animals.
- An activity detection module capable of predicting dynamic objects' motion, improving decision-making in real time.
- Integration of these systems into a real-time environment with extensive testing on video and image datasets under varying conditions.

Lane Line detection is a one of the most important parts of automatic driving cars which uses computer vision. The method helps to identify the path a self -driving car must take adhering to the traffic rules. In this article, we will build a project to detect lanes and identify all objects and anomalies on the sidewalk which may, in future, come into the path of the vehicle.

The project is aimed at designing a system that is robust and real-time friendly, capable of identifying objects around the vehicle and check for the possibility of collision with it. The collision detection is based on factors such as: movement of vehicles around it, animals, people on the sidewalk and activities near or on the lane. If the system identifies anything as an anomaly, it will give the precaution to the driver and request them to be careful.

II. Data Pre-Processing

The dataset taken into context is the Caltech-lanes dataset. It consists of 1000 images out of which 600 is kept for training, 200 is kept for validation and 200 is kept for testing purposes.

The data as such cannot be taken into the model. So, before we move into the methods, various pre-processing needs to be done. Pre-processing is an inevitable stage where the images are transformed in a way as to reduce the complexity of the algorithm, and increase the efficiency of the program.The input images are colour images; hence the pre-processing begins with colour transforms [6].

A. Colour Transforms

Images in colour form have information encoded in different levels or layers. Extracting only the useful information requires selecting those pixels which have meaning to the proposed work. The main information comes in the format of RGB (Red, Green and Blue) values and hue, saturation, lightness values. Each stage has been processed and shown.

a. RGB Colour transform

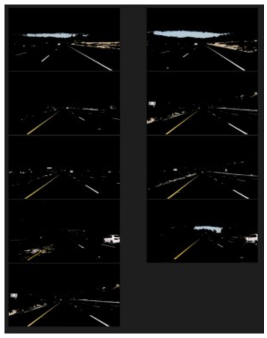

Fig.2 RGB Colour Transform

In the RGB Colour Transform stage, the lines are usually in yellow or white, hence those colours have been selected and plotted from the original set of images.

b. HSL colour selection

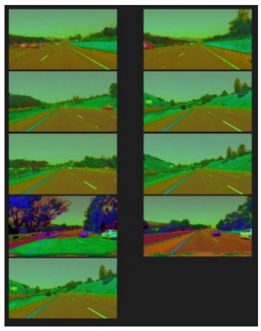

Fig.3 HSL Colour Conversion

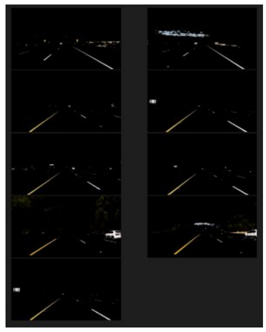

Fig.4 HSL Colour Selection

HSL colour model stands for Hue, Saturation and Lightness model. HSL model gives an idea on how exactly the three colours mix to create all the other colours. The lightness component gives the description on how much black and white exist in the images. Hence, they have been taken as a required field.

c. HSV colour selection

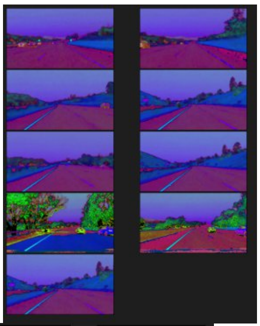

Fig 5. HSV Colour Conversion Fig 6. HSV Colour Selection

HSV model stands for Hue, Saturation and Value model, also known as HSB – Hue, Saturation and Brightness model. It is similar to HSL model, but differs with it as the HSV model depicts how the colours appear under light.

B. Grayscale conversion

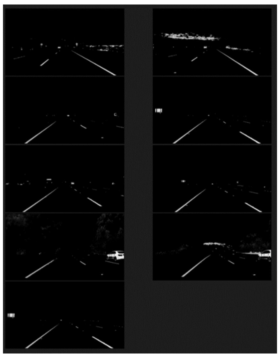

Fig 7. Grayscale Image

After getting the required colour information from the images, the images are transformed into grayscale. This is done so as to extract descriptors from the images. Extracting descriptors from three layered images is exceedingly difficult. Hence, they are converted into grayscale and the required information is extracted.

C. Gaussian Smoothening

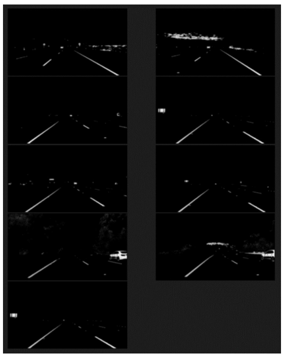

Fig 8. Smoothened Image

In order to reduce the noise in the images, gaussian smoothening is done. It evens out the pixels and noise are removed in essence. This means the important information can be accessed much easier without interference from random information that may arise from surrounding conditions of the camera.

D. Canny Edge detection

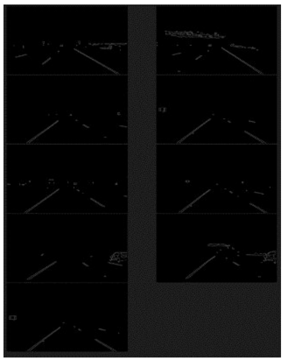

Fig 9. Edge Detected Image

Canny Edge detection [7] is a method of identifying edges by processing the images on multiple stages. This means that further processing uses only those identified edges and hence it is much easier and faster to be processed. This also helps in identifying regions based on edges found.

E. Region Selection

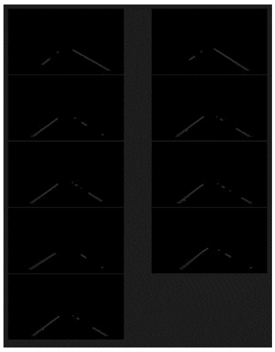

Fig 10. Region of Interest

Now that all the information has been extracted, the region is chosen based on the coordinate system by choosing a bounding box which is in the shape of a trapezium in front of the car. It is clear from the image obtained that only the required information is obtained.

III. METHODOLOGY

This system is designed to analyse images captured from the vehicle's dashcam in real-time, ensuring safe vehicle navigation. The dashcam primarily captures the front view, where key elements such as lanes, pedestrians, animals, traffic signals, and other vehicles can be detected. These images are pre-processed. The 3 main phases of the system are:

- **Lane Detection**
- **Object Detection**
- **Activity Recognition**

Once these tasks are performed, the system can superimpose the lane lines on the image and provide decision-making support for safe vehicle movement based on detected objects and potential hazards. Below is an in-depth explanation of each phase of the methodology.

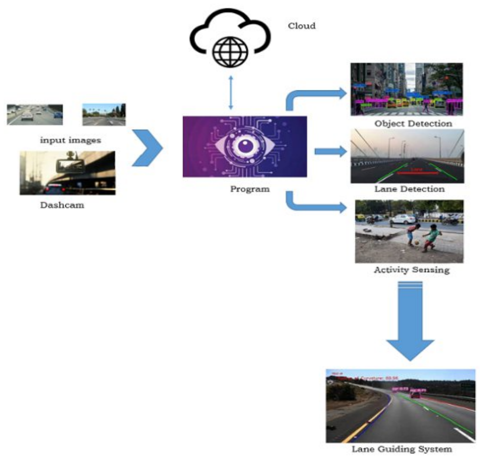

Fig 11. System architecture

1) Phase I: Lane Detection

Lane detection [8]is the first task, as it is critical for keeping the vehicle safely within its driving path. The process involves identifying lane lines and ensuring the vehicle stays on the correct track, even in complex situations like faded lanes or unclear road markings.

- **Hough Transform for Lane Detection** [9]: The **Hough Transform** algorithm is used to detect straight lines, which often correspond to lane lines on the road. By mapping points in the image space to lines in a parameter space, the system can accurately identify lane boundaries, even when they are not perfectly visible. The algorithm plots the lines directly onto the road image.
- **Predictive Lane Detection with Hough Transform**: When lane markings are unclear or missing (e.g., due to wear or dirt), we use a predictive model based on historical lane data. The model predicts the most likely lane positions by analyzing the road's curvature and prior knowledge of the road layout. This allows the vehicle to continue safely, even in less-than-ideal conditions.
- **Handling Missing Lanes Using Optical Flow**: In cases where there are no visible lane lines, we employ **optical flow analysis**. This technique measures the movement of objects and the vehicle relative to the road edges and nearby obstacles. Using this data, the system estimates a safe path, ensuring the vehicle can still navigate effectively without explicit lane markers.

2) Phase II: Object Detection

Object detection [10] is crucial for identifying potential hazards in the vehicle's environment, such as pedestrians, other vehicles, or road obstacles. The system must not only detect objects but also classify and track their movements.

- **YOLOv5 for Real-Time Object Detection**: To ensure fast and accurate object detection [11], we use **YOLOv5** (You Only Look Once). YOLOv5 processes multiple frames per second, making it ideal for real-time detection in autonomous driving. It can identify a wide variety of road objects, including pedestrians [11], vehicles, animals, and obstacles. The algorithm's efficiency and speed make it superior to traditional detection methods like **HOG descriptors**. YOLOv5 is trained on a large dataset (such as the COCO dataset), enabling it to detect objects accurately even in complex or cluttered environments.
- **Object Classification and Tracking**: Once objects are detected, the system classifies them into two categories: static (e.g., parked vehicles, stationary obstacles) or dynamic (e.g., moving pedestrians, cyclists, vehicles). We use the **Lucas-Kanade optical flow** algorithm to track the motion of dynamic objects. This method calculates the movement of each object relative to the vehicle and predicts their trajectory. This helps in determining whether any dynamic object is likely to enter the vehicle's path

This phase provides the vehicle with a clear understanding of its surroundings and potential hazards, allowing for timely responses to avoid collisions.

3) Phase III: Activity Recognition

After detecting lanes and objects, the final task is to predict the movement and behavior of these objects. This helps the vehicle anticipate potential risks and make decisions to avoid them.

- **Lucas-Kanade Optical Flow for Motion Tracking**:
  Initially, the Lucas-Kanade Optical Flow technique[15] was used to track the motion of detected objects. This method calculates the movement of objects by focusing on key features in the image and tracking their positions across

consecutive frames. It helps to determine the direction and speed of dynamic objects, such as pedestrians or vehicles near the lane lines. However, it has limitations in identifying objects near lane boundaries.

- **Dense Optical Flow for Motion Tracking**: **Dense Optical Flow** is used to track the movement of objects and pedestrians around the vehicle. This technique analyzes how each pixel in the image moves between consecutive frames, helping the system determine the direction and speed of nearby objects. Whether it's a pedestrian crossing the street or a vehicle changing lanes, the system uses this information to predict their future position.

This phase is key for real-time risk assessment and ensures the vehicle can respond proactively to changing conditions on the road.

The first 2 phases of the project are focused on images and on video dataset, each frame will undergo this process. The phases 3 happen on video data where we can identify the movement of the things around the vehicle, giving us an understanding on how the objects around are moving. The phase 3 focuses on predicting the further movement of the traced objects and identifying whether they come into the path the vehicle has to follow. If so, the driver needs to be warned.

IV. Results and Discussion

The model was tested with the input images captured from the vehicle's dashcam. After applying the pre-processing techniques discussed in the methodology, the region of interest (ROI) was selected, and the lanes were identified using the **Hough Transform** algorithm. Below is a snapshot of the lane detection results:

Fig 12. Output after lane detection

As seen in the image, the lane lines are clearly plotted on top of the original image. The identified lanes guide the vehicle between the correct boundaries. With successful lane detection, **Phase I** of the project is considered complete.

Moving on to **Phase II** (object detection), we experimented with two methods to identify objects in the vehicle's surroundings. Object detection presented some challenges, particularly in detecting smaller or partially occluded objects.

- The first method used was the **Histogram of Oriented Gradients (HOG) descriptor**, a widely used technique for object detection. The results obtained through HOG are illustrated below:

Fig 13. Object Detection using HOG Descriptor method

In the image, it's evident that several pedestrians were not detected. While HOG performs well in detecting certain features, it struggles with finer details, particularly in dynamic environments. To address this issue, we switched to the **CvLib-YOLOv5** object detection method, as outlined in the methodology.

Fig 14. Object Detection using Cvlib method

Here, **YOLOv5**, integrated through **CvLib**, demonstrates significantly better results. All pedestrians and even two vehicles in the image were successfully detected and labelled. The YOLOv5 architecture, trained on the **COCO dataset**, is particularly effective in real-time detection, as it processes multiple frames per second, ensuring high accuracy and timely detection. With this improvement, **Phase II** of the project is considered complete.

Next, we moved on to **Phase III**, which involved tracking the motion of detected objects using two techniques: the **Lucas-Kanade optical flow** and **Dense Optical Flow**. These methods help predict the movement of objects relative to the vehicle and determine potential collision threats.

Fig 15. Output of Lucas Kanade model

The output of Lucas Kanade model is shown. In this model, the objects near the lane line were not identified properly which is why the next method has been tried.

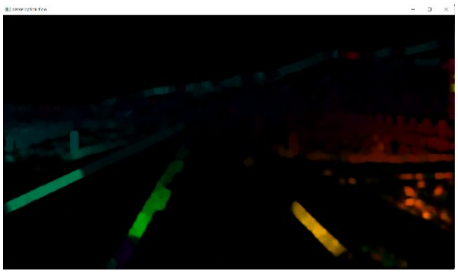

Fig 16. Output of Dense Optical Flow method

In Dense Optical Flow[14], at one glance, it feels as if all the important information is lost. But whenever a pixel is moving, this method has been able to clearly identify it regardless of the position of those moving pixels. Hence this method is taken as the optimal one.

For the **video dataset** evaluation [12], we combined the three phases (lane detection, object detection, and motion tracking) to observe the system's performance on continuous data. The input video shows a car traversing a highway with yellow side lines and white dotted lane lines. During the course of the video, two vehicles appear on the same lane as the subject vehicle.

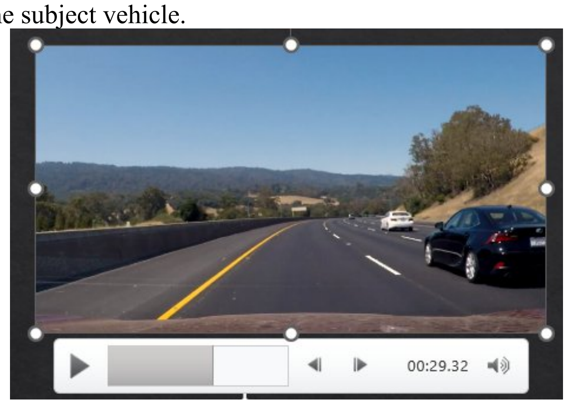

Fig 17. Input video

The first part is lane detecting which is shown in the fig 18 below, where the lane is detected in real time and is masked and shown.

Fig 18. Lane detection output video

In the image of the video, it is clearly visible that the lanes have been identified and the ideal path has been plotted. Now, checking for objects is shown in fig 19.

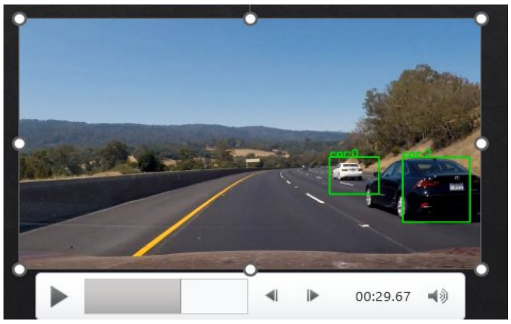

Fig 19. Object Detection output video

It is clear from the image that each of the vehicles have been identified, a bounding box has been drawn around them and they have been labelled. For the final step, both of these are combined and the final output video is obtained.

Fig 20. Final Output Video

PHASE	ALGORITHM	REMARK	EFFICIENCY
1	**Hough Transform**	Performs well on this dataset due to prevalence of straight lane in most images.	HIGH
	Optical Flow Analysis	Performs well on images where lane markings are absent.	MODERATE
2	HOG descriptor	Some people and most objects were not detected	MODERATE to HIGH
	CvLib method (incorporates Yolo v5 Algorithm)	All people and objects were detected	VERY HIGH
3	Lucas Kanade	Motion of the objects near the lane line were not detected	MODERATE
	Dense Optical flow	Effectively identifies moving pixels regardless of their position.	HIGH

NOTE: The selected algorithms are highlighted in bold.

V. CONCLUSION AND FUTURE SCOPE

Automatic driving systems and driver assistance systems are in the rage nowadays. A model of such a system has been made and desirable results have been obtained. The model is capable of working on real time videos and identification of objects is successful. The project can be extended to high traffic areas and tested. In the future, we plan to expand the system's capabilities by introducing multi-lane detection, improving accuracy in extreme weather conditions, and testing the system on larger datasets to further refine its predictive capabilities. Additionally, integrating advanced reinforcement learning models could allow the vehicle to adapt to new driving environments more efficiently.

REFERENCES

[1] A. Assidiq, O. O. Khalifa, M. R. Islam and S. Khan, "Real time lane detection for autonomous vehicles," 2008 *International Conference on Computer and Communication Engineering, 2008, pp. 82-88*, doi: 10.1109/ICCCE.2008.4580573.

[2] Cheng, Hsu-Yung, et al. "Lane detection with moving vehicles in the traffic scenes." *IEEE Transactions on intelligent transportation systems* 7.4 (2006): 571-582.

[3] A. Moujahid et al., "Machine Learning Techniques in ADAS: A Review," 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), Paris, France, 2018, pp. 235-242, doi: 10.1109/ICACCE.2018.8441758.

[4] Z. Kim, "Robust Lane Detection and Tracking in Challenging Scenarios," in *IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 1, pp. 16-26*, March 2008, doi: 10.1109/TITS.2007.908582.

[5] Jung, Heechul, Junggon Min, and Junmo Kim. "An efficient lane detection algorithm for lane departure detection." *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013.

[6] Chiu, Kuo-Yu, and Sheng-Fuu Lin. "Lane detection using [1]color-based segmentation." *IEEE Proceedings. Intelligent Vehicles Symposium, 2005*. IEEE,2005.

[7] P. Ganesan and G. Sajiv, "A comprehensive study of edge detection for image processing applications," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2017, pp. 1-6, doi: 10.1109/ICIIECS.2017.8275968.

[8] Wang, Yifei, Naim Dahnoun, and Alin Achim. "A novel system for robust lane detection and tracking." *Signal Processing* 92.2 (2012): 319-334.

[9] X. Wei, Z. Zhang, Z. Chai and W. Feng, "Research on Lane Detection and Tracking Algorithm Based on Improved Hough Transform," 2018 *IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE),* Lanzhou, China, 2018, pp. 275-279, doi: 10.1109/IRCE.2018.8492932.

[10] Z. Soleimanitaleb, M. A. Keyvanrad and A. Jafari, "Object Tracking Methods:A Review," 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 2019, pp. 282-288, doi: 10.1109/ICCKE48569.2019.8964761.

[11] Hailong Li, Zhendong Wu and Jianwu Zhang, "Pedestrian detection based on deep learning model," 2016 *9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI),* Datong, 2016, pp. 796-800, doi: 10.1109/CISP-BMEI.2016.7852818.

[12] P. Adhiyaman and B. A, "A Detailed Study on Obstacle Detection and Avoidance Techniques for On-road Vehicles," *2022 International Conference on Computer, Power and Communications (ICCPC),* Chennai, India, 2022, pp. 236-241, doi: 10.1109/ICCPC55978.2022.10072046.

[13] L. Kechiche, L. Touil and B. Ouni, "Real-time image and video processing: Method and architecture," 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Monastir, Tunisia, 2016, pp. 194-199, doi: 10.1109/ATSIP.2016.7523067.

[14] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 3061-3070, doi: 10.1109/CVPR.2015.7298925.

[15] D. Sun, S. Roth and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them", *International Journal of Computer Vision (IJCV)*, vol. 106, no. 2, pp. 115-137, 2013.

[16] Janani, P. & Jayaraman, Premaladha & Ravichandran, Kattur Soundarapandian. (2015). Image Enhancement Techniques: A Study. Indian Journal of Science and Technology. 8. 10.17485/ijst/2015/v8i22/79318.

[17] Babu, Tina et al. 'Colon Cancer Prediction on Histological Images Using Deep Learning Features and Bayesian Optimized SVM'. 1 Jan. 2021 : 5275 – 5286

[18] D. Jaswal, S. Vishvanathan, and K. P. S., "Image classification using convolutional neural networks," *Int. J. Sci. Eng. Res.*, vol. 5, no. 6, pp. 1661–1668, Jun. 2014.

[19] S. Amuly, C. Jyotsna, and J. Amudha, "Deep Learning Model for Image Classification," in *Computational Vision and Bio-Inspired Computing*, S. Smys, J. Tavares, V. Balas, and A. Iliyasu, Eds. Cham, Switzerland: Springer, 2020, vol. 1108, pp. 1–8. doi: 10.1007/978-3-030-37218-7_36.

[20] A. Neena and M. Geetha, "Image Classification Using an Ensemble-Based Deep CNN," in *Recent Findings in Intelligent Computing Techniques*, P. Sa, S. Bakshi, I. Hatzilygeroudis, and M. Sahoo, Eds. Singapore: Springer, 2018, vol. 709, pp. 1–8. doi: 10.1007/978-981-10-8633-5_44.