PROJECT IDEATION

Problem Statement

The number of different road signs has increased tremendously due to the rising number of vehicles. Thus, an automated system to recognise and classify these traffic signs is essential to prevent road accidents.

Problem Motivation

According to [9], in 2015, more than 30,000 humans died due to motorized vehicle accidents. Accidents can be caused by a variety of reasons and some include ignoring the road signs, obstruction of the traffic sign, and drivers being distracted. Theoretically, leveraging software to classify road signs might prove to be less susceptible to accidents. A Traffic Sign Detection and Recognition (TSR) system plays a significant role in the driver assistant model by regulating traffic, demonstrating the state of the road, and making the drivers and pedestrians aware of new road signs which may have been ignored because of inattentiveness, thereby ensuring safety from accidents. The system will eliminate risks caused due to using a phone while driving, exceeding speed rules, or driving while being drunk as it can alter the automobile's speed depending on the recognised road sign. Additionally, usage of onboard sensors and estimation tools will provide a 360-degree view of their environment all the time.

Literature Review

Driver assistance software is currently a cutting-edge technology, and a significant number of researchers have devoted their work towards creating a framework that aids with detecting, tracking, and recognising various traffic signs. From the 2000s, real-time applications have been successfully integrated with driverless assistant technology to aid safe driving. A traffic sign recognition system is divided into two mains tasks- detection and classification. Detection involves identifying the location of the road sign and extracting the target rectangles while the classification phase decides the category to which the detected sign belongs. Research conducted by [2] proposes a robust, threefold system that uses a combination of color and shape contained by the AdaBoost structure while Chen L., Li Q., and Mao Q. applied a similar color-based segmentation procedure to detect regions of interest (ROI) during the detection step. Traffic signs, in this case, are recognised within ROIs using Haar wavelet features. Speeded up robust features (SURF) are employed to detect local invariant attributes within a potential traffic sign which is then matched to the features of template sign images. Vinston R., Kumar P., Alfred S., Thameem M. (2017) created an image processing system with a segmentation algorithm and the resultant image is processed using Open CV.

Many obstructions are faced while detecting road signs. Some of them include varying brightness depending on the time of the day and climate changes, scene obstruction, objects containing text and logos that resemble a traffic sign, similarity in traffic signs, and the need for a high accuracy rate by a real-time system processing vast amounts of data. However, since traffic signs are placed ahead of time, the driverless system has a sufficient amount of time to analyse and process the traffic events such as ongoing constructions, dead ends, traffic signals, etc., and act accordingly.

The system introduced in [2] is segregated into 3 phases- learning prior knowledge of the environment or color to detect traffic signs, analysing the geometry of the edges that were detected in the first phase to know if the sign is circular or triangular, and cross-validating the signs. Maldonado-Bascon S., et al. (2007) used a similar feature extraction technique as [2], however, Support Vector Machine (SVM) was employed as the classifier to match the numerous road signs. The recognition phase of this framework included finding the difference between every pixel color, shape classification of traffic signs using SVM, and understanding the context through Gaussian-kernel SVM. Pei1 S. et al. (2018) suggested a Multiscale Deconvolution Network system as it solves the issue of a lot of time being taken to pre-process images and use of complex algorithms to improve and find blurred and subpixel image data of road signs. This method combines multi-scale convnets with deconvolution networks, thereby, proving to be a more robust and efficient traffic recognition and classification system. [4] provides a brief on integrating hardware utilising Arduino fixed to a self-driving automobile to solve the problem of detecting traffic signs in real-time. Additionally, the distance between the vehicle and the approaching road sign is calculated. Quite a few approaches using algebra were implemented to aid traffic sign recognition. Rahim A. found contours in the picture, then ellipses and rectangles were detected. Low power consumption is a key component in traffic sign recognition. This is achieved using the NVIDIA Tegra processor in [5] combining it with a YOLO CNN neural network. GPU has also been leveraged along with CUDA and OpenCV libraries to utilise the full computing power.

The dataset used to detect and classify different traffic signs in [2] is the German Traffic Sign Benchmark. This database comprises 43 various classes of traffic signs, a total of more than 50,000 ground-truth images and each physical traffic sign of the dataset is unique. [3] uses a traffic sign dataset from China. These images were captured using a CCD Video Camera placed on the roof of a Chery SUV. 200 images containing 281 road signs formed the test dataset. Belgium traffic signs dataset is employed in [4].

The two main characteristics considered when detecting traffic signs are the color and shape of the sign. Recognition of road signs in [2] is achieved using a deep learning method in combination with a convolutional neural network (CNN). The final implementation base model used is the ResNet50 design from Keras. The hyper-parameters were optimised for this prototype and custom layers were added to match their use case. The most used method for detecting traffic signs is the color segmentation technique according to [3]. Since RGB color space is highly sensitive to the amount of light intensity, HIS and HSV that are not affected by changes in the lighting have been used. The main purpose of a real-time traffic sign recognition system should be to analyse the road beforehand in real-time while precisely detecting road signs and taking minimal corresponding response time which is solved by the system proposed in [4] that suggests the first step to recognising road signs is to acquire data about the land in front of the vehicle using its fitted camera. This information is then transferred to the control station via wireless technology. Google's MobileNet system was employed in [4] which was operated on Anaconda. It is comprised of a 3x3 convolution layer along with a wholly connected layer a single convolution layer. Leveraging TensorFlow with Anaconda is advantageous because of its easy installation as more of the data science Python libraries are provided by the Anaconda structure. Network graphs and view metrics can be easily visualised via TensorBoard. It also takes less time to compute.

Before passing the data through the neural network in [2], it must be pre-processed as the efficiency of the model is determined by the type of data used which then defines the representation of features. Noise can be removed from the images using a total variation filter blended with wavelet denoising and bilateral filters. The resultant images are then grey-scaled and converted into 50x50 pixel values. Reduction of image size and conversion of RGB representation is computationally affordable. The color segmentation process described in [3] involves extracting required color pixels which in turn is used to get ROIs based on the limits of the bounding box of the pixels. The color scheme includes red, blue, yellow, white, and black. As white and black constantly appears with other black or other three colors, this method only aims to work with red, blue, and yellow. In [7] proposed the usage of filter techniques such as unsharp filter, average filter, dilate filter, and erode filter to make the road images more significant.

As time is a crucial factor in real-time applications, template signs are classified of eight depending on the color and proficient AdaBoost classifiers using the Approximate Nearest Neighbor (ANN) algorithm. A candidate traffic sign is matched by drawing a comparison between every feature of the candidate with a specific database. The traffic sign image of the specific database which has the maximum number of matches with the candidate traffic sign is the predicted class.

The architecture of the model in [2] is derived from ResNet50. Pre-trained weights have been set from Image-Net. The final layer of the model is replaced by a three-convolution block which is constructed using a single convolution layer, batch stabilisation max pooling, and a dropout level. The neural network used in [5] is defined as a mathematical function comprised of interrelated artificial neurons. A vector is taken as an input, which is then passed through the hidden layers. The probability of belonging to a certain class is generated as the output. Neural networks work on numerical. YOLO CNN uses bounding boxes which are small rectangles to detect objects. The underlying principle of YOLO is the Single Shot technique that is all images are detected concurrently in a single pass of the framework.

Trained models generate a lower accuracy rate when operated on vast amounts of data, therefore, a deep learning framework is created to resolve this issue. The traffic sign dataset is split into a 70:20:10 ratio for training, testing, and validation respectively in [2]. A custom splitter was leveraged to obtain an equal account of the various classes. The training phase of the network used 500 epochs and 128 images. Categorical accuracy, that is checking if the index of the maximum true target is equivalent to that of the maximum predicted target, was the metric used to determine the weights. Adagrad optimiser was set with default parameters generated by Keras. The learning rate of the model shifts according to the frequency of weights being adjusted while learning takes place. The increasing variation and frequency of updated weights result in fewer updates via the optimiser. [3] uses the AdaBoost algorithm as the classifier learning technique. It is a blend of weak classifiers to make a strong one, then these strong classifiers are combined to create a cascade classifier. Haar-like features are essential to teaching the AdaBoost classifier. In [5] the YOLO model is provided with a 3D image which is then computer to 488x488. This image is then passed through the GooLeNet framework which in turn produces feature maps of size 14x14x1024. Two convolutions are employed. The information is then converted into tensors which are then filtered and displayed.

An accuracy rate of 98.21% is obtained with a precision of 93.94% for the neural network model in [2] with negligible complexity, thereby enabling its usage in real-time applications. It can predict 128 traffic signs in 3 seconds. The size of the model varies from 56.8 megabytes to 57.6 megabytes depending on how much training data was used. Leveraging GPUs like MAGMA can decrease computation time and expenses for object detection. The six classifiers trained in [3] detected traffic signs in 170 milliseconds and classified them in 200 milliseconds on average. Out of 281 signs, only 265 signs were correctly identified, 14 signs were missing, and 2 signs were falsely identified. The system had a detection rate of 94.3% making it efficient and effective while the recognition accuracy was 92.7%. An accuracy rate of 83.7% was achieved in [4] providing precise detection and fast response time to the detected event. Incorrect recognition in [5] occurs due to several factors that include the requirement of large sets of images and all the traffic sign data is used for training purposes. The recognition speed of NVIDIA Jetson TK1 is comparatively lower than the desktop GPUs due to a small RAM. While working with a neural network the entire RAM is used. However, a decent accuracy rate and time were obtained for a mobile system.

In the future better implementable coding standards for hardware could be introduced. Considering two-way lanes while detecting traffic signs can emerge as a breakthrough in driverless technology, especially for countries like India. To ensure real-time response to traffic incidents, the Rasberry Pi microprocessor can be integrated through the different phases of detection and classification. More research could be focused on applications that comprise real-time data and quick response.

Project Statement

Our traffic sign detection and classification system focus on using YOLOv5 on a dataset obtained from Kaggle. This dataset comprises 877 images belonging to 4 unique classes. Bounding box comments have been provided in the PASCAL VOC format. The 4 distinct classes are as follows: traffic light, stop, speed limit, and crosswalk. YOLO framework is mainly implemented in real-time object detection applications. A current state-of-the-art open-source application of YOLOv5 is present on the GitHub repo. It is well-known for its fast and light characteristics about the size of the model. YOLO's provides a faster alternative than the other object detection methods. In comparison to the other versions of YOLO, version 5 is more accurate, however, version 4 is faster. There is yet more research to be done on which version is better. Since YOLO is the latest developing technique in the field of object detection, our application uses this algorithm as it focuses on the entire image then only the content within the bounding box which is a primary requirement when detecting and classifying traffic signs. The IDE is used to create the code in Visual studio along with python extensions.

References

[1] Campbell, A., Both A., Sun C. Q. (2019) 'Detecting and mapping traffic signs from Google Street View images using deep learning and GIS' *Computers, Environment and Urban Systems.*

[2] Shukla U., Mishra A., Jasmine S. G., Vaidehi V., Ganesan S. (2019) 'A Deep Neural Network Framework for Road Side Analysis and Lane Detection', *INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019, ICRTAC 2019.* Location, Date of the conference. Delhi: ATLAN, Volume 165, 2019, Pages 252-258.

[3] Chen L., Li Q., Li M., Mao Q. (2011) 'Traffic Sign Detection and Recognition for Intelligent Vehicle', 2011 *IEEE Intelligent Vehicles Symposium (IV).* Germany,2011.

[4] Swaminathan V., Arora S., Bansal R., Rajalakshmi R. (2019) 'Autonomous Driving System with Road Sign Recognition using Convolutional Neural Networks', *Second International Conference on Computational Intelligence in Data Science, ICCIDS-2019.*

[5] Artamonov N. S., Yakimov P. V. (2018) 'Towards Real-Time Traffic Sign Recognition via YOLO on a Mobile GPU', *The IV International Conference on Information Technology and Nanotechnology.* Russia,2018. Conf. Ser. 1096 012086

[6] Kanjee R. (2020) 'YOLOv5 Controversy- Is YOLOv5 Real?', *medium*, June 18. Available at: https://medium.com/swlh/yolov5-controversy-is-yolov5-real-20e048bebb08 (Accessed: 20 May 2021).

[7] Vinston R. R., Kumar P., Alfred S., Thameem M. (2017) 'Accident Avoidance by Using Road Sign Recognition System', *International Research Journal of Engineering and Technology (IRJET).* Tamil Nadu ,2017.

[8] Wang C. Y., Cheng-Yue R., (2016) 'Traffic Sign Detection using You Only Look Once Framework'. Available at: cs231n.stanford.edu/reports/2016/pdfs/263_Report.pdf

[9] *Self Driving Cars Explained, How do self driving cars work - and what do they mean for the future* (2018) Available at: https://www.ucsusa.org/resources/self-driving-cars-101#:~:text=Self%2Ddriving%20vehicles%20are%20cars,navigate%2C%20and%20drive%20the%20vehicle. (Accessed: 18 May 2021).

[10] *HOW CAN AUTOMATED AND CONNECTED VEHICLES IMPROVE ROAD SAFETY?* Available at: https://roadsafetyfacts.eu/how-can-automated-and-connected-vehicles-improve-road-safety/#:~:text=Self%2Ddriving%20vehicles&text=Using%20on%2Dboard%20sensors%20and,90%25%20of%20all%20accidents%20today. (Accessed: 18 May 2021).