ISSN 1533-9211

DOI 10.5281/zenodo.7113216

VEHICLE SPEED ESTIMATION BASED ON VIDEOS USING DEEP LEARNING TECHNIQUES

DASARI SIVA KRISHNA[1], PRASAD VADAMODULA[2] and K. JAYASRI[3]

[1, 2] Department of Computer Science & Engineering, GMR Institute of Technology, Rajam, Srikakulam.
[3]Department of Computer Science & Engineering, Vignan Institute of Engineering College, Visakhapatnam.

Abstract

Now-a-days, road accidents are rapidly increasing. There are many reasons for road accidents like vehicle clashing, over-speed of vehicles which are resulting in major loss for not only the person involved but also their entire family. Though rules are imposed, there is no control on speed of the vehicles. Speed can be estimated by calculating the velocity of moving vehicles. This project aims at calculating speed of various vehicles using a video dataset. It is decided to develop a model to calculate the speed of the vehicles using OpenCV. OpenCV is used for the vehicle detection and it increases the prediction accuracy. This algorithm is fast due to no iterations for object detection. Tracking is done and speed is estimated. This project takes a video as input and gives the speed of the vehicles in the video as output.

Keywords: Speed estimation, vehicles, videos, OpenCV, tracking.

INTRODUCTION

Vehicle Speed detection has been an important research topic since the rate of road accidents are increasing day by day. There are many ways to reduce these accidents and decrease the death rate. Imposing restrictions on speed is one of them. To impose restrictions, the speed of the vehicle must be known to the traffic polices or the people who are in charge for the respective department. The vehicle speed can be detected by using deep learning, machine learning or any other techniques. This project uses a technique named OpenCV for vehicle detection. Tracking algorithm is applied for tracking and then the speed is estimated. Vehicle detection may also be done using other algorithms but this algorithm is selected because it saves time. In other algorithms, multiple object detection is not possible. For example, in Convolutional Neural Networks (CNN), if 10 objects are to be identified in an image, then the image needs to be processed 10 times. So, if multiple numbers of vehicles have to be found in a frame, the frame should be iterated number of times under the same process to identify all the vehicles present in the frame. But here which is an enhanced version, the frame is processed only once and multiple vehicles are identified and their exact positions are obtained. This says that there are no iterations in which the frame needs to be processed again and again. So this algorithm proves that it can save time. The algorithm is not only time efficient but also it is a standard algorithm which makes it much more important. This algorithm has highest learning capability which automatically increases the accuracy. When an object needs to be detected from an image, then there should be no unwanted background errors. OpenCV minimizes those errors and produces the correct output. These errors are removed by using Gaussian blur. Here also, there may be some issues like same cell having two object centers and multiple boundary boxes for same object. The robustness of this technique improves accuracy in small object

ISSN 1533-9211

DOI 10.5281/zenodo.7113216

detection. It uses Canny for edge detection in images so that the whole frame is processed giving equal priority to every pixel. Tracking is done by using dlib library which uses the previous positions of the vehicles as reference and detect their current position. Tracking helps in understanding the time that vehicle is taking to move. After tracking is done, speed is estimated with the help of detected vehicles and tracked vehicles.

Deep Learning:

Deep Learning is a subfield of machine learning. The main processing unit in deep learning is neuron. These networks of neurons form neural networks which mimic the functionality of the human brain. It is implemented to recognize complex patterns. Deep neural networks consist of an input layer, multiple hidden layers and an output layer. A deep learning model learns from the data through these layers. The number of layers will be different for different types of problems. Deep learning is mostly used when the amount of data is huge. Deep learning algorithms play an important role in identifying the features and are able to handle a large number of processes for the data which may or may not be structured. Deep Neural networks are of 3 types which include Artificial Neural Networks (ANN) to model complex patterns and predictions, Convolutional Neural Networks (CNN) used for image related tasks and Recurrent Neural Networks (RNN) for sequence classification, sentiment classification and video classification. Most computer vision algorithms use CNN to extract features like texture, edges and spatial data. This project uses deep learning techniques for pre-processing the video dataset to feed as input to the model.

OpenCV:

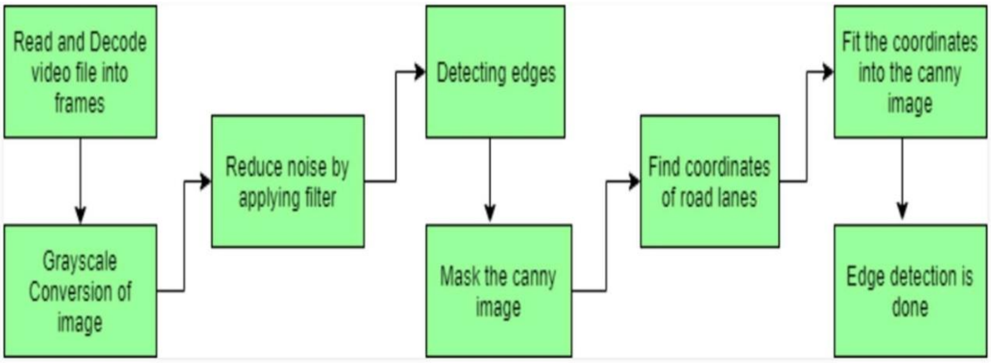
Computer Vision is a technique which is used to understand images and videos and helps in extracting information from them. Python OpenCV is an open-source library for computer vision, machine learning, and image processing. By using it, we can process images and videos to recognize objects, faces, animals and the handwriting of humans. It can also be integrated with other libraries like Numpy in python. With OpenCV, we can also capture video from the camera. OpenCV is a tool used in image processing, segmentation, text summarization, and object detection and computer vision tasks. Computer vision allows the computer to perform the same kind of tasks as humans with the same efficiency. There are two main tasks.

Object Classification - In the object classification, a model is trained on a dataset of particular objects, and the model classifies new objects as belonging to one or more of the training categories.

Object Identification - In the object identification, the model will identify a particular instance of an object.

Our project uses OpenCV in python to calculate the speed of the vehicles from the video provided. The model is trained using the OpenCV library in python to detect the vehicles and track them in order to calculate their speeds. In our project, OpenCV is combined with the dlib library in python to detect the vehicles. Dlib is a Library in python which is used for making real world machine learning and data analysis applications. It is also used in face detection, facial landmark detection, object detection, etc. When dlib is combined with OpenCV it provides more accurate results.

Figure 1.1: OpenCV architecture for vehicle detection


Read and Decode video file into frames
Grayscale Conversion of image
Reduce noise by applying filter
Detecting edges
Mask the canny image
Find coordinates of road lanes
Fit the coordinates into the canny image
Edge detection is done

Major Challenges in the current literature in-line with proposed work:

This project has got wide acknowledgement nowadays and there are many people who researched on this topic using different technologies. So the challenge faced here is finding a technique and a different tracking algorithm which are not previously combined. Another challenge faced is vehicle detection. Vehicle detection becomes difficult with the increase in number of lanes. So vehicle detection is quite tough with 2 lanes. The input of the project is a video. So implementation of the project with the input as a video is difficult.

Solutions to the challenges:

Technique used for vehicle detection is OpenCV. This technique is a standard technique and it has more accuracy when compared with other techniques and algorithms. Tracking is done by the installation on dlib library. This tracker gives the best tracking estimation and provides the correct location of the object with respect to its previous location. The difficulty in detecting vehicles with lanes increased has overcame with proper training. The input video has been divided into frames. Each frame is observed and then vehicles are identified and tracked. With the help of those identified and tracked vehicles, speed is estimated.

Motivation for the proposed work:

Nowadays, vehicle accidents are increasing rapidly. Due to this, death rate is getting increased and many families are losing their loving ones. These accidents happen when people do not follow traffic rules and if there are no strict rules regarding traffic and speed of the vehicle. So if the speed of the vehicle is known, then it will be easy for supervisors or any concerned authorities to impose charges according to their vehicle speeds. If there are more rules and regulations regarding vehicle speed the people will not overcome them and there are more chances of reduction in accidents. Hence, this project helps in detecting the vehicle speed and decreasing accidents.

ISSN 1533-9211

DOI 10.5281/zenodo.7113216

Overview of the proposed work:

This project aims in detecting the speed of a vehicle using OpenCV technique. It installs the required libraries initially. Cascade classifier is used for training purpose. Then video is captured and divided into frames. The frames undergo gray scaling in which all the colors present in the image are changed into the shades of gray. Gray scaling helps in dimensionality reduction because it is single dimensional while RGB colors are 3 dimensional. It also helps in reducing the complexity of the model. After gray scaling is done, image is blurred with the help of Gaussian Blur. It helps in reducing the image noise. Then Canny is used for edge detection so that the total frame gets covered. After Canny, frames undergo erosion and dilation for obtaining accurate features so that the vehicle detection becomes easy. Next, contouring is applied to know the exact shape of the frame. Then vehicles are detected with the help of pre-trained model and then they are tracked and speed is estimated.

LITERATURE SURVEY

Lu, S et al., (2020) this method estimates the vehicle speed by making use of the shadow beneath the vehicle. There are some assumptions made such as input video is captured by a single camera by placing it at a position where shadow is visible, vehicle movement is constant and road lane is a straight line. Firstly, Region of Interest containing moving vehicles is selected and frame difference method is applied on it to obtain some frames. Then projection histograms are drawn on those frames to get key bins. After that, all possible speeds are enumerated with an interval of 5 to decrease time and then tested by drawing projection histograms. The group which has closest histograms with actual bins is selected and then again tested by drawing histograms with an interval of 1. The corresponding speed of the group whose projection histograms are closest to key bins is the speed of the targeted vehicle. This model is tested with 3 different datasets containing 2054 vehicles and accuracy obtained is 99.4%.

Shin, J et al., (2018) in this paper the author proposed the prediction algorithm of velocity based on the Markov chain with speed constraints to improve the prediction performance in the algorithm that reflects the road geometry by designing the roadway position as the domain. They proposed a model that designed with the driver’s data also has the same characteristics with the velocity range of the constraint model becomes narrower in the corner section and can provide more credible predictions using the Markov chain. They proposed prediction algorithm and compared with the Markov chain without the speed constraint model by comparing the performance under the same conditions except for the model type, any improvements in prediction performance as a result of adding the constraint model can be analyzed. They proposed prediction algorithm that does not consider the position error of the ego-vehicle. Therefore, the experiment was conducted under conditions where the position error was minimized as much as possible.

Hua, S et al., (2018) in this paper the author has combined modern deep learning models with classic computer vision approaches to propose an efficient way to predict vehicle speed and introduced some state-of-the-art approaches in vehicle speed estimation, vehicle detection, and object tracking. They generated some ground-truth data to evaluate different models for vehicle

ISSN 1533-9211

DOI 10.5281/zenodo.7113216

detection and a tool was used to annotate data is the VGG Image Annotator (VIA). They focused on applying supervised models to traffic related problems and thus included a large collaborative annotation effort for the dataset. They introduced few methods for tracking vehicles in traffic videos and for estimating their speed and the method takes a detect-then-track approach and can use object detections from any vehicle detection algorithm as input. In this section, we analyze the results obtained by applying our tracking and speed estimation models to the Track 1 videos from the 2018 NVIDIA AI City Challenge. First, challenge result is described and then analyzed the potential avenues of improvement for our model. They have introduced a model for tracking vehicles in traffic videos based on a detect-then-track paradigm, combined with an optical-flow-based data-driven speed estimation approach, and described some solutions for Track 1 of the 2018 NVIDIA AI City Challenge. The model performed well but was not as competitive as some of the other Challenge teams, displaying excessive variability. Due to lack of time, they did not compare the method against other detect then-track algorithms, which they leaved as future work. They planned to investigate smoothing techniques for the predicted vehicle speeds which may lead to improved model performance.

Kim, J. H et al., (2020) this model uses Dual Deep Neural Networks (DDNN) for the estimation of longitudinal speed of vehicles. It uses 640000 training samples and 320000 testing samples. The data is collected and pre-processed and then stored in data storage. Then, both the neural networks are trained with different accelerations i.e, one network with low accelerations and the other with high accelerations. After that estimated longitudinal speed, wheel speed, longitudinal acceleration are used to calculate the speed of the front right wheel. Then the DDNN is equipped with all other needs such as weights and activation functions and uses all the above values for the calculation of the longitudinal speed of the vehicle. The model also compares the accuracies of single DNN and dual DNN.

Famouri, M et al., (2020) Projection Displacement Difference (PDD) is the major problem in video based vehicle speed estimation systems in which there is a plane disparity between street-level indicators and above-plane feature points of vehicles. This results in irregular estimation of speed. A novel motion plane-based approach for vehicle speed estimation is proposed in this paper. It includes two phases' offline and online phases. In the offline phase, the license plate motion plane is determined. A pipeline is used to estimate the speed of vehicles in the online phase. For every vehicle track, the center of the license plate is extracted and the obtained motion plane model in motion plane estimation step is applied to map into its real world 3D position. Displacement is calculated for every two license plates in sequence with respect to their center points.

Javadi, S., Dahl et al., (2019) Speed measurement systems are classified into two types namely active and passive based on their methodology. This paper focuses on passive optical video systems unlike active systems where mostly sensors are used. In this paper, a speed estimation method is proposed which utilizes the camera frame rate, the placement of the intrusion lines and a movement pattern vector and provides a pdf (probability density function) model of the vehicle's speed. To evaluate this method more accurately, four intrusion lines and a GPS

(Global Positioning System)-equipped car are used to obtain the results. For vehicles traveling in the range of 70 km/h– 100 km/h, the average error rates at frame size 50 fps and 30 fps are 1.77%and2.17%,respectively.Error rate can be decreased further more with an increase in frame rate.

Cheng, G et al., (2020) In this paper, they propose a target tracking method based on video image feature matching vehicle to simplify the algorithm complexity and obtain high accuracy. The process of real-time speed detection based on video image can be divided into two aspects: moving target detection and moving target tracking. KNN algorithm is used to identify vehicle targets and obtain good initial vehicle characteristics. K-nearest Neighbour (KNN) classification algorithm is most of the k most similar samples in the feature space belong to a certain category, then the sample also belongs to this category. The biggest limitation of this method is that it cannot detect stationary and slow-moving targets. Experimental results show that the relative error of speed detection can be controlled at about 5%.

Wang, D et al., (2017) in this paper, they propose a new method that recognizes risky driving behaviors purely based on vehicle speed time series. It first retrieves the important distribution pattern of the sampled positive speed-change (value and duration) tuples for individual drivers within different speed ranges. Then, it identifies the risky drivers based on different patterns of drivers. Then, they design a linear supporting vector machine (SVM) model to identify the risky drivers based on the distribution patterns of different drivers. The SVM model will be trained, and can correctly recognize the risky and normal drivers from the newly generated samples. Seventy-six over 80 (95%) testing samples had been correctly classified.

Abuella, H et al., (2019) in this paper, a visible light sensing based speed estimation system, termed as ViLDAR, was proposed. Radio Detection and Ranging (RADAR) system is a popular method for vehicle speed estimation. Light Detection and Ranging (LiDAR) uses the same main principle of the RADAR but utilizes a different portion of the the electromagnetic spectrum. This paper proposed a novel speed estimation system called the Visible Light Detection and Ranging (ViLDAR) that builds upon sensing visible light variation of the vehicle’s headlamp. ViLDAR outperforms RADAR/LiDAR systems in both straight and curved road scenarios. Results showed that using the received light intensity of vehicle’s headlamp, one can estimate the vehicle’s speed with accuracy of more than 90%.

Zhang, Z et al., (2017) the need for Intelligent Transport Systems (ITS) is increasing in which vehicle count and speed monitoring data are important inputs. AMR (Anisotropic Magneto-Resistive) sensor is used for vehicle speed detection. The existing algorithms are not well suitable for time distorted waveforms. So, a speed up version of Dynamic Time Warping (DTW) algorithm based on the vehicle re-identify rule is proposed. Two sensor nodes are placed in the middle of the lane and vehicle signatures are measured at both the nodes which should be equal. The similarity between them is calculated using DTW.

Yang, L et al., (2019) this system uses an optimized motion detector and a novel text detector to efficiently locate vehicle license plates in image regions containing motion. Vehicle speed is measured by comparing the trajectories of the tracked features to known real-world measures. It has also shown that extracting distinctive features from the license plate region led to better results than taking features spread over the whole vehicle, as well as an approach which uses a particle filter for blob tracking. In the experiments, measured speeds had an average error of −0.5 km/h, staying in over 96.0% of the cases inside the +2/−3 km/h error interval determined by the regulatory authorities in several countries. This also aims to apply an OCR(Optical Character Recognition) on the detected license plates in order to create a traffic speed control system with integrated surveillance tools, e.g. to compute the traffic flow, to identify stolen vehicles, etc. Note that the videos contain some vehicles with no visible license plate, and that the ground truth speed meter sometimes fails to properly assign a speed to avehicle.

Grents, A et al., (2020) assessing the speed of vehicles using data from a video surveillance camera. It used a two-stage Faster R-CNN detector together with a SORT tracker to solve the task. The proposed system makes it possible to classify vehicles and estimate their movement speed with an accuracy of over 78%. It is planned to improve the developed system to increase the accuracy of object recognition and speed determination. There are many ready-made software libraries for splitting videos into frames and preparing a frame for recognition, such as OpenCV. Besides, this library can be used to highlight detected objects in video. The data set presented in the study can further be used by other researchers as a complex test or additional training data.

Luvizon, D. C et al., (2016) In this paper Our system is set in a fixed location to capture two view stereo videos for passing vehicles by using a calibrated binocular stereovision system. The optimized single shot multi box detector network that can efficiently detect license plates in the captured two view stereo videos. Vehicle speed is measured by dividing the distance between two selected 3D points by frame intervals. In this paper system performs non-intrusive and stealth measurement in intelligent traffic surveillance and overcomes the challenges of speed measurement for multiple vehicles on multiple lanes in different motions simultaneously. It deals with a speed measurement range of 10m where distance estimation errors are in the order of centimetres'.

METHODOLOGY

Implementation:

Phases of the proposed work:

1) Training
2) VideoCapturing
3) Datapreprocessing
4) Vehicledetection
5) Vehicletracking
6) Speedestimation.

Training:

Training refers to understanding of the problem and producing the correct output with the help of proper help. Training data is usually large and helps in teaching the model to give the best accuracy. OpenCV uses pre trained networks for the training of data according to the requirement. So, there is no need to spend more number of hours only for the sake of training. This project uses Cascade Classifier for training purpose. Cascading classifiers are trained with several hundreds of positive and negative images to obtain a good accuracy while detecting objects. Here, there will be a search window which will help in detecting the objects. The search window will move throughout the image and whenever the required object is identified, it will return the exact location of the object. This project requires the detection of vehicles. So, number of positive and negative images of vehicles are taken for training here. Positive images refer to the vehicles whereas negative images refer to any other objects or shadows of the vehicles. With the help of these images, the classifier understands the correct images of the vehicles and detects them correctly. So after training is done, this classifier is applied to the original input and then vehicles are identified in that image.

Video Capturing:

Video capturing refers to taking the video as an input. In general, many models use images as input because performing operations on images is easy while it is a little bit difficult with videos. So, the captured video is divided into frames and operations are performed on each frame individually. This also helps in reducing the effect of one frame on the other frame. Further processing steps are performed on each and every frame. Here video is captured using Video Capture () method present in cv2 library. Then an infinite while loop is set up and use read () method to divide the video into frames. Frames can be seen with the help of imshow () method. This infinite loop is stopped when the user presses a specific key. After the division of frames, multiple vehicles are detected simultaneously and their locations are identified. In this way video is captured and frames are divided for further processing.

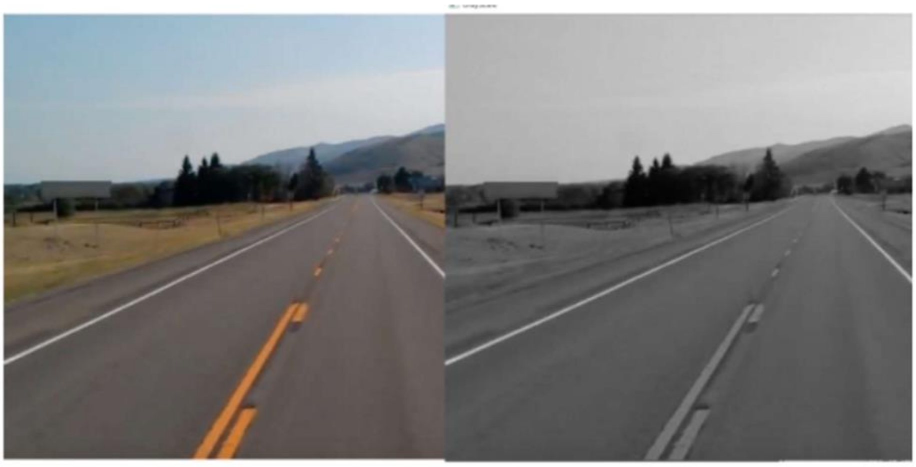
Data Preprocessing:

Data pre-processing means making the available data useful to obtain a better accuracy. Sometimes the available data may contain duplicate data and noisy data. These all problems are reduced here with the help of using different techniques for each problem. Here, Gaussian blur, canny, erosion and dilation are done for the data to obtain the features correctly.

Grayscaling:

Gray scaling is the process of converting an image from other colour spaces such as to shades of gray. It varies between complete black and complete white. Gray scaling helps in dimensionality reduction because it is single dimensional while RGB colours are 3 dimensional. It also helps in reducing the complexity of the model. This is because if RGB image has size of 10x10x3. Then input layer will have 300 nodes. But if we use gray scaling the nodes will be reduced to 100 since the image will become single dimensional. Hence, the complexity of the model is decreased. The image is converted into grayscale using cvt color () function.

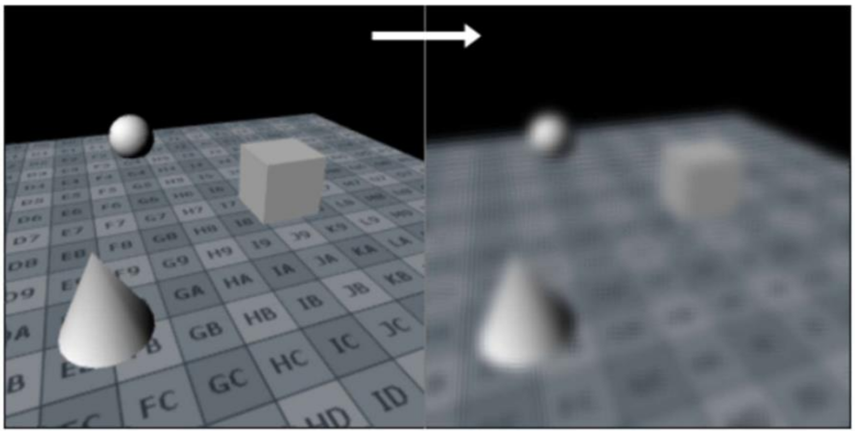
Figure 3.1: Gray scaling



Gaussian Blur:

In image processing, a Gaussian blur is the result of blurring an image by a Gaussian function. Gaussian Blur is also known as Gaussian smoothing. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scale. When Gaussian blur is applied, the key features are highlighted. Because of this, feature extraction becomes easy.

Figure 3.2: Gaussian Blur

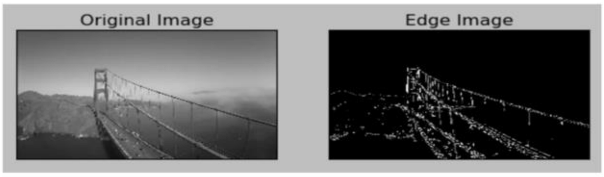


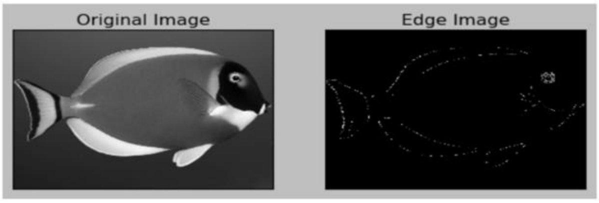
ISSN 1533-9211

DOI 10.5281/zenodo.7113216

Canny:

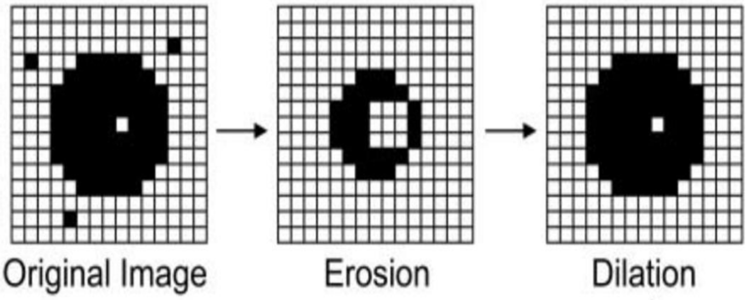
Canny () Function in OpenCV is used to detect the edges in an image. In general, when data is being processed, edges are ignored so accuracy will not be more and there may also be some mistakes in identifying the objects. Edge detection increases robustness and flexibility. It helps us reduce the amount of data (pixels) to process and maintains the structural aspect of the image. So if we use this function, edges are also identified clearly so that the whole image will get covered while processing. It is an inbuilt function which takes 3 arguments. The first argument is the input image and the next arguments are the size of the image.





Erosion and Dilation:

Dilation and Erosion are basic morphological processing operations that produce contrasting results when applied to either gray-scale or binary images. Erosion involves the removal of pixels ate the edges of the region. Dilation is the reverse process with regions growing out from their boundaries. Erosion decreases the size of the object and remove small anamolies. Dilation increases the size of the erosed object and fills the holes if any. Dilation also increases the brightness of the object so that the required features are clearly visible. Erosion is similar to dilation. The difference is that the pixel value calculated is minimum rather than the maximum value indilation.



Vehicle Detection:

Here vehicles are detected from the frames with the help of trained model. Multiple vehicles are detected simultaneously and their locations are identified. After the identification of the locations, each vehicle is given a unique id to avoid confusion between vehicles. These vehicles will be detected correctly only if the training is done properly. After the detection of vehicles, they all are converted into specific size as required so that all the vehicles will be in same size which simplifies further processing. The lanes segmentation will also be sone for easy identification of vehicles. Then tracking is done to identify their positions accurately and then speed is calculated.

Vehicle tracking:

Here, initially the image will be resized into required dimensions after reading the image. Then the image will be copied into another file so that the original copy does not get destroyed. After that the vehicles are tracked according to their ids. In the previous phase, each vehicle is assigned with a unique id. Now, tracking makes use of those ids to track the vehicles. First, the position of a vehicle with specific id is identified and then the position of the same id vehicle is identified in the next image. In this way, it will tell how much time the vehicle took to travel from a specific position to specific position.

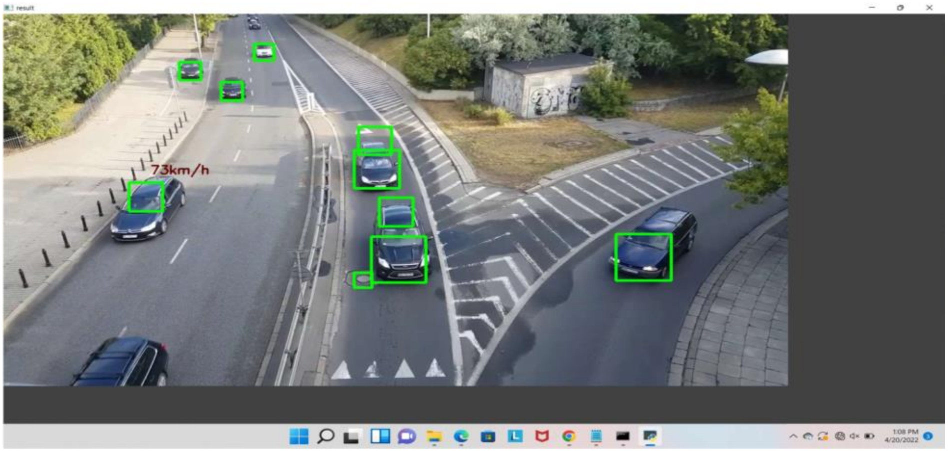
Speed Estimation:

The speed is estimated with the help of time and the tracked vehicles. This can be calculated by knowing the distance between the pixels. This refers to the distance between vehicles. The initial speed is obtained in fps and then it is converted into kmph.

RESULTS:

The input of the model is a video consisting of moving vehicles. The output video consists vehicles with bounding boxes along with their speeds.

Fig 4.1: Vehicle 1 with the detected speed



DOI 10.5281/zenodo.7113216

Fig 4.2: Vehicle 2 with the detected speed

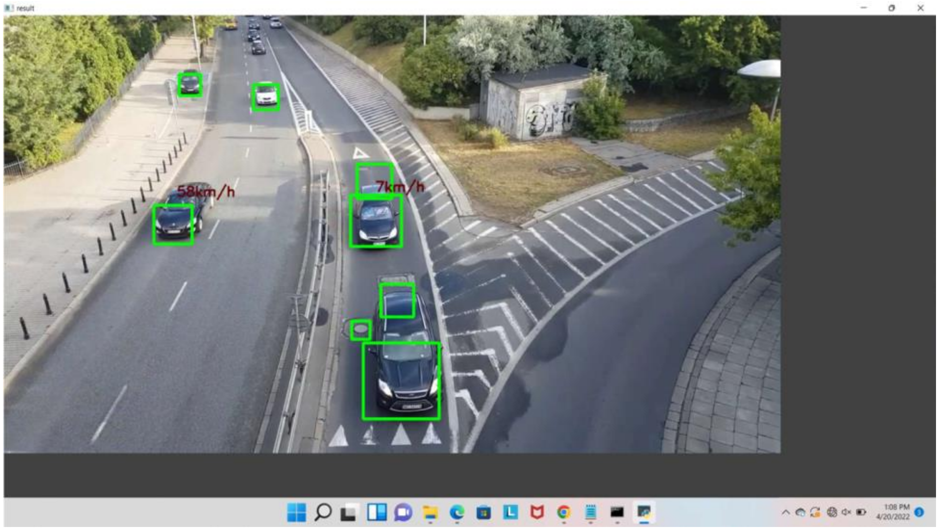


Fig 4.3: Vehicle 4 with the detected speed

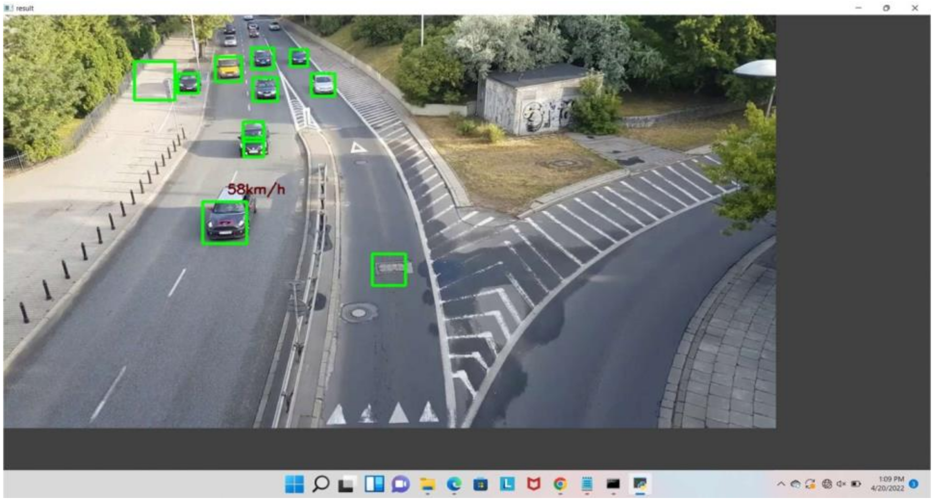
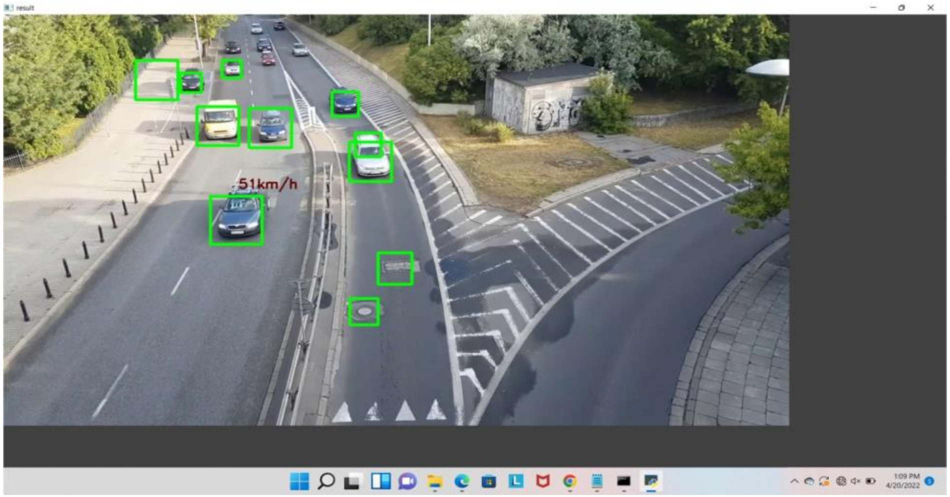


Fig 4.4: Vehicle 5 with the detected speed



CONCLUSION:

This paper proposed a technique using OpenCV for the estimation of vehicle speeds from videos. First, training is done using Cascade classifier. Then video input is taken and divided into frames. After that, pre-processing techniques are applied on those frames for a better accuracy. Then, vehicle detection is done followed by vehicle tracking and speed estimation.

Future Scope:

Vehicle speed detection can be used in traffic videos to detect the speed of the vehicles. This helps in understanding the over speed of the vehicle. In this way, strict traffic rules can be implemented and accidents can be avoided. When accidents get avoided, death rate will also decrease automatically.

Base Paper:

Anandhalli, M., Baligar, P., Saraf, S. S., &Deepsir, P. (2022). Image projection method for vehicle speed estimation model in video system. Machine Vision and Applications, 33(1), 1-13.

REFERENCES:

Lu, S., Wang, Y., & Song, H. (2020). A high accurate vehicle speed estimation method. Soft Computing, 24(2), 1283-1291.

Shin, J., &Sunwoo, M. (2018). Vehicle speed prediction using a Markov chain with speed constraints. IEEE Transactions on Intelligent Transportation Systems, 20(9), 3201-3211.

Hua, S., Kapoor, M., &Anastasiu, D. C. (2018). Vehicle tracking and speed estimation from traffic videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp.153-160).

Kim, J. H., & Yoon, S. W. (2020). Dual Deep Neural Network Based Adaptive Filter for Estimating Absolute Longitudinal Speed of Vehicles. IEEE Access, 8, 214616-214624.

Famouri, M., Azimifar, Z., & Wong, A. (2018). A novel motion plane-based approach to vehicle speed estimation. IEEE Transactions on Intelligent Transportation Systems, 20(4), 1237-1246.

Javadi, S., Dahl, M., &Pettersson, M. I. (2019). Vehicle speed measurement model for video-based systems. Computers & Electrical Engineering, 76,238-248.

Cheng, G., Guo, Y., Cheng, X., Wang, D., & Zhao, J. (2020, February). Real-time detection of vehicle speed based on video image. In 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (pp. 313-317).IEEE.

Wang, D., Pei, X., Li, L., & Yao, D. (2017). Risky driver recognition based on vehicle speed time series. IEEE Transactions on Human-Machine Systems, 48(1), 63-71.

Abuella, H., Miramirkhani, F., Ekin, S., Uysal, M., & Ahmed, S. (2019). ViLDAR—visible light sensing-based speed estimation using vehicle headlamps. IEEE Transactions on Vehicular Technology, 68(11), 10406-10417.

Zhang, Z., Zhao, T., Ao, X., & Yuan, H. (2017). A vehicle speed estimation algorithm based on dynamic time warping approach. IEEE Sensors Journal, 17(8), 2456-2463.

Yang, L., Li, M., Song, X., Xiong, Z., Hou, C., & Qu, B. (2019). Vehicle speed measurement based on binocular stereovision system. IEEE Access, 7, 106628-106641.

Grents, A., Varkentin, V., &Goryaev, N. (2020). Determining vehicle speed based on video using convolutional neural network. Transportation Research Procedia, 50,192-200.

Luvizon, D. C., Nassu, B. T., &Minetto, R. (2016). A video-based system for vehicle speed measurement in urban roadways. IEEE Transactions on Intelligent Transportation Systems, 18(6), 1393- 1404.