VEHICLE DETECTION USING HOG AND LANE DETECTION USING CNN

A MAJOR PROJECT REPORT

Submitted by

CH.EN.U4AIE20001	ANBAZHAGAN E
CH.EN.U4AIE20053	RAMYA POLAKI
CH.EN.U4AIE20069	TRINAYA KODAVATI
CH.EN.U4AIE20031	SMITHIN REDDY K
CH.EN.U4AIE20035	ABHIRAM KUNCHAPU

in partial fulfillment of the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

SIGNAL AND IMAGE PROCESSING

AMRITA SCHOOL OF ENGINEERING, CHENNAI

AMRITA VISHWA VIDYAPEETHAM

CHENNAI – 601103, TAMIL NADU

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, CHENNAI, 601103

BONAFIDE CERTIFICATE

This is to certify that the major project report entitled "**VEHICLE DETECTION USING HOG AND LANE DETECTION USING CNN"** submitted by

CH.EN.U4AIE20001	ANBAZHAGAN E
CH.EN.U4AIE20053	RAMYA POLAKI
CH.EN.U4AIE20069	TRINAYA KODAVATI
CH.EN.U4AIE20031	SMITHIN REDDY K
CH.EN.U4AIE20035	ABHIRAM KUNCHAPU

in partial fulfillment of the requirements for the award of the **bachelor of Master of Technology** in **COMPUTER SCIENCE AND ENGINEERING** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Chennai.

Signature

Dr. S.Aasha Nandini

CSE Dept Assistant professor

This project report was evaluated by us on ……………...

INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

We the undersigned solemnly declare that the thesis "vehicle detection using hog and lane detection using CNN" is based on our own work carried out during the course of our study under the supervision of Dr. Aasha Nandini, Assistant professor of Computer Science & Engineering, and has not for the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice of reporting scientific information, due acknowledgment has been made wherever the findings of others have been cited.

Anbazhagan, Smithin, Ramya, Trinaya, Abhiram

ABSTRACT

The ability of an autonomous vehicle to categorize what it sees on the road is attained when it is taught to the point where it can reliably discern between automobiles, pedestrians, trucks, animals, and so on and label them as such. This research focuses on teaching an autonomous vehicle to identify whether or not there is a car on the road. If we need to pass a car, we will enter the fast lane and pass the vehicle when it is safe to do so. The same holds true for an autonomous car, which depends on visual signals to determine what to do next. We will Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images and train a Linear Support Vector Machine (SVM) classifier and then apply a color transform and combine binned color features, as well as histograms of color, to the HOG feature vector and then we will be implementing a sliding-window technique and use the trained classifier to search for vehicles in images then Run the same on a video stream instead of images and create a heat map of recurring detections frame by frame to reject outliers and find the detected vehicles by estimating a bounding box around it. To complete this mission, a car's front-facing camera captures a video of the road, which is then analyzed frame by frame to fulfill the goal by using SVM and HOG (Histogram Oriented Gradients) in python and OpenCV.

The second part of the project is to detect lanes. Road border lanes are one of the leading causes of car accidents, endangering both drivers and pedestrians. Both computer vision and machine learning algorithms struggle to detect road border lanes. Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks. Lane can be also detected using a canny edge detector algorithm but we can also use some of the CNN models to detect lanes for more accuracy.

ACKNOWLEDGEMENT

I wish to record my deep sense of gratitude and profound thanks to my research supervisor Dr.Aasha Nandini, Professor, Department of Computer Science Engineering, Amrita Vishwa Vidyapeetham, Chennai, for his / her keen interest, guidance, and constant encouragement during all stages of my work, to bring this thesis into fruition.

I am extremely indebted to Dr.Prasanna Kumar, class advisor, Department of CSE-AI, Amrita Vishwa Vidyapeetham, Chennai, and panel members for their valuable suggestions and support during the course of my research work.

I also thank the faculty members of the Department of CSE, Amrita Vishwa Vidyapeetham, Chennai, for their valuable support throughout the course of my research work.

Anbazhagan, Smithin, Ramya, Trinaya, Abhiram

LIST OF FIGURES

INDEX

CHAPTER 1

INTRODUCTION

One of the most difficult challenges in computer vision is the automated identification of objects in pictures. It is also a critical stage in the development of various modern applications that need high-level picture interpretation. Indeed, the difficulty of object recognition is determining whether a picture contains a specific object or an object belonging to a predefined object category. We may characterize the recognition issue as a pairing between the model of the target item and a set of descriptors extracted from the test picture.

The algorithm uses a dense grid of Histograms of Oriented Gradients (HOG) to represent a detection window. The Histogram of Oriented Gradient (HOG) descriptors provide better performance than other existing feature sets. Each stage of the HOG computation has much influence on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. This detection system is based on an Integral Histogram of the Gradient descriptor combined with Support Vector Machines for the recognition stage. SVM is a classification method, which has proved to be very efficient in such cases of high dimensional data. It has been developed to detect cars.

With vehicle detection, Lane detection is a crucial component of both self-driving automobiles and sophisticated driver assistance systems. Many complex strategies for finding lanes have been considered in recent years. The lane is typically identified with a single graphic, though. This can result in a subpar performance in a number of extremely terrible circumstances, such as when there is a great deal of shadow, a great deal of mark damage, a great deal of traffic, etc. The road's lanes are actually long, straight lines. Therefore, if you make use of data from earlier frames, you might be able to identify the lane that is difficult to correctly locate in the current frame.

We propose a hybrid deep architecture that combines the convolutional neural network (CNN) and the recurrent neural network to accomplish this by examining lane detection using many frames of a continuous driving scene (RNN). Each frame's information is specifically extracted by a CNN block, and the CNN features of numerous continuous frames—which have the attribute of being a time series—are then sent into the RNN block for learning features and predicting lanes.

1.1 HISTOGRAM OF ORIENTED GRADIENTS

The Histogram of Oriented Gradients, or HOG, is a feature descriptor that is frequently used to extract features from picture data. It's commonly used for object detection in computer vision tasks. The structure or shape of an entity is the focus of the HOG descriptor. It is only noted if a pixel is an edge or not when it comes to edge. In addition, HOG can offer edge direction. The gradient and orientation (or magnitude and direction) of the edges are extracted to accomplish this.

These orientations are also determined in 'localized' segments. This implies that the entire image is divided into smaller regions, with the gradients and orientation determined separately for each sector. Finally, for each of these regions, the HOG would output a Histogram. The 'Histogram of Oriented Gradients' is named after the histograms are constructed with the gradients and orientations of the pixel values. The main idea behind HOG feature extraction is to determine the gradient magnitude and gradient direction of each pixel within an image, and use this information to look for a type of feature signature for images of cars within an image.

The gradient of the image is calculated. The gradient is obtained by combining the magnitude and angle from the image. Considering a block of 3x3 pixels, first Gx and Gy are calculated for each pixel. First Gx and Gy are calculated using the formulae below for each pixel value.

The horizontal and vertical gradient of each pixel (x, y), are calculated

$$G_x(r,c) = I(r,c-1)$$

$$G_y(r,c) = I(r-1\,,c) - I(r+1\,,c)$$

Where r, and c refers to rows and columns respectively.

After calculating Gx and Gy, the magnitude and angle of each pixel are calculated using the formulae mentioned below.

$$Magnitude(\mu) = \sqrt{G_x^2 + G_y^2}$$

$$Angle(\theta) = \left|\tan^{-1}\frac{G_y}{G_x}\right|$$

1.2 SUPPORT VECTOR MACHINE

SVM (Support Vector Machine) is a supervised machine learning algorithm that can be used to solve classification and regression problems. It is, however, primarily used in classification issues. We plot each data item as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a specific coordinate, using the SVM algorithm.

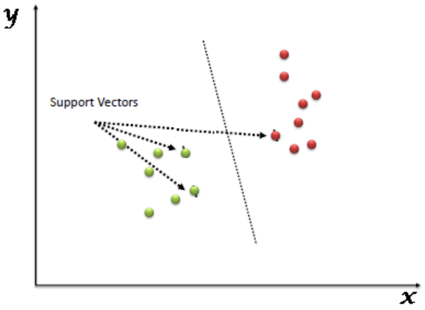

Fig 1 SVM Visualisation

Then we classify the data by locating the hyper-plane that best distinguishes the two groups. The coordinates of each individual observation are what Support Vectors are. The SVM classifier is a frontier that separates the two classes (hyper-plane/line) as well as possible.

1.3 CONVOLUTION NEURAL NETWORKS FOR LANE DETECTION

In a multi-lane urban driving environment, many computer vision methods have been developed for the purpose of solving the lane detection problem. However, these traditional techniques require a lot of computation not to mention the complexity of the code needed for their implementations. The intelligent and efficient lane detection system using TensorFlow and Keras. So, in the past few decades, deep learning has proved to be a very powerful tool because of its ability to handle large amounts of data the interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition one of the most popular deep neural networks is convolutional neural networks.

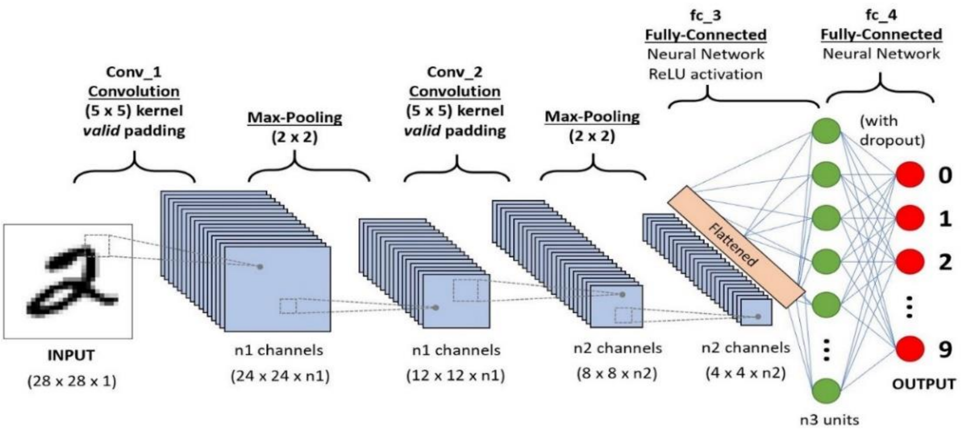

Fig2 Convolutional Neural Network

CNNs are commonly applied to image processing and analysis they are composed of multiple layers of artificial neurons each layer extracts an increasingly complex set of features starting from simple lines to more complex corners and edges and finally arriving at the final prediction of faces objects and so on but before the image is fed to these neuron layers it has to pass through several convolutions and pulling layers these layers are responsible for reducing the spatial size of the features to decrease the computational power required to process the data. Supervised machine learning projects are often separated into two phases the first one concerns the training process where the agent learns to achieve a certain task given labeled examples of the task as inputs at the end of this phase the CNN model is saved and can be then deployed to real-life applications without the need to retrain the model all over again.

CHAPTER 2

LITERATURE SURVEY

2.1 SURVEY

The majority of sophisticated driver assistance systems include on-road vehicle recognition and lane detection as standard features (ADAS). In this paper, researchers introduce an integrated method called Efficient Lane and Vehicle detection with Integrated Synergies (ELVIS). Vehicle detection is done in 2 part based method where the image is divided into 2 parts P1 and P2. In ELVIS, the lower part i.e. P2 is traversed along the road and then the HOG feature of the P2 is computed and fed to an SVM classifier. If the classification of P2 is positive then it proceeded with P1 and it is also classified. Now, the classified scores will be compared with the threshold for the overall classification.[1]

Huang Guan, Wang Xingang, Wu Wenqi, Zhou Han, and Wu Yuanyuan proposed an article that is to achieve real-time and accurate leading vehicle localization, and to do so, it suggests a lane-vehicle identification and tracking system that is both novel and efficient. The following is a list of the components that make up the system. Lane detection as well as recognition of the sky and the road are carried out to reduce the possible area that is being filled by the leading vehicle. After that, a vehicle detection algorithm is applied to each shot to guarantee that every car was captured within it. When information on the ROI is taken into account, it may be possible to correctly pinpoint the leading automobile. As the last phase, vehicle tracking is implemented to make the entire system more dependable and expedient overall. The successful operation of the entire system is contingent upon one's capacity to identify moving cars. Convolutional neural networks are used in an innovative and effective method that is utilized to produce high-quality vehicle detection in real-time. This method is built on the foundation of deep learning. The input image is pre-processed then it does ROI reduction and vehicle detection and then using those two it finds the target leading vehicle and then the vehicle gets tracked.[2]

The study describes a lane-vehicle identification and tracking system that includes vehicle detection, vertical asymmetry measurement, lane boundary recognition, and lane area tracking. The first step is to segment a traffic scene image into potential road regions. Then, utilizing lane markings detection, the region's lane borders are extracted. With the use of a linear-parabolic tracking model, these discovered boundaries are tracked in succeeding video frames. Therefore, using the estimated model parameters, a rough lane region is built. By combining the understanding of lane region with vehicle detection, the vehicle scanning region is limited to the road area so that there is less interference from the road environment and non-vehicle structures when detecting the shadow below a vehicle constantly. They used bounding boxes to extract vehicle regions.[3]

They presented a synergistic method for combined lane and vehicle tracking for driver assistance in the research. The data of the lanes first does the inverse perspective mapping and the mapped output is given to the steerable filter bank then the model is fitted using RANSAC(random access consensus) and then the Lane position is tracked via the Kalman filter. This is the Lane tracking framework. The article includes the module for tracking and detecting on-road vehicles. It includes a vehicle detector based on active learning that is coupled with particle filtering for tracking vehicles.[4]

Wally Chen, Leon Jian, and Sy-Yen Kuo proposed Unified map navigation, lane detection, vehicle identification, and speed camera warning are all included in the provided driving safety system. Vehicle and lane detection are combined to produce reliable recognition results. To evaluate the identification rate and performance through field testing, the system is being ported onto the PAPAGO P3 driving recorder. Lane Departure Warning System and Front Collision Warning System both give drivers additional assistance based on the road model and vehicle appearance.[5]

2.2 SUMMARY AND FINDINGS

To identify on-road cars and lanes, a brand-new integrated method known as ELVIS is introduced. When compared to existing methods, the strategies in the study by Mohan M. Trivedi and Ravi Kumar Satzoda have been found to detect fully visible cars with lower false positive rates and about 90% fewer calculations. By incorporating the vehicle detection data, it is also demonstrated that the lane feature extraction approach benefits from significant cost savings.[1]

The entire system combines lane detection and tracking, recognition of sky and road regions, convolutional neural network-based vehicle detection, and vehicle tracking. The findings demonstrate that in movies of real-world scenes, our technology may achieve higher recall and accuracy.[2]

Without a camera parameter, lane detection and tracking systems can be used for other driving assistance features to calculate ROIs. Additionally, the center of the vehicle can be easily obtained using horizontal symmetry analysis and vertical asymmetry analysis.[3]

The car's behavior is tracked and focus windows are created by analyzing the lane tracing and vehicle bounding box results. The window prevents interference and noise, and it functions well with local searching, which was included to increase the effectiveness of lane detection.[5]

2.3 CHALLENGES

It was difficult to perform the proposed techniques on occluded vehicles.[1]. The researchers found it tough to improve lane detection and tracking. So, they thought of further adding the vehicle information to make the model more robust.[2] Many researchers also found it difficult to perform the techniques mentioned because of the various road conditions and the light conditions (day or night). It was also very challenging to expand the algorithms to urban driving and expansion to contextual tracking. Lane Detection was challenging for a few researchers because of the usage of different lines on the road such as solid lines and dashed lines etc.

2.4 OBJECTIVES OF WORK

- Extract features from training, cross-validation, and test sets of labeled (positive and negative) sample data. develop a classifier.
- Converting images to the desired color space, choosing the desired channels, and then extracting color histogram, and/or spatial characteristics are the steps for feature extraction.
- By utilizing a sliding window and a smoothed heatmap, you can find and create bounding boxes around objects in videos.
- After estimating the bounding boxes detect Lanes using a Convolutional Neural Network.
- Get the shaded region of the lanes for the bounding box of vehicle’s images.

CHAPTER 3

PROPOSED WORK

3.1 ARCHITECTURE OF THE SYSTEM

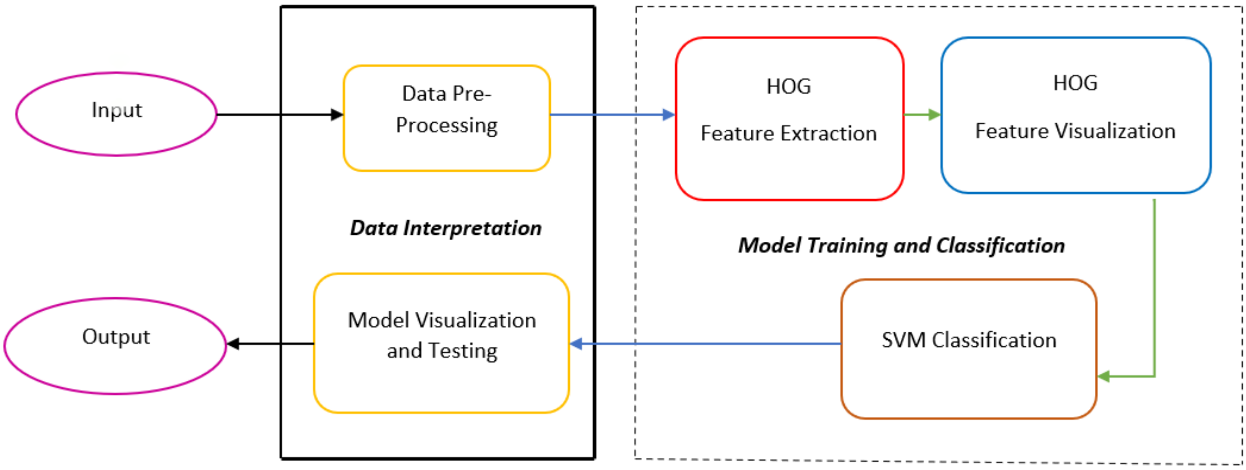

Fig3 workflow for Vehicle Detection

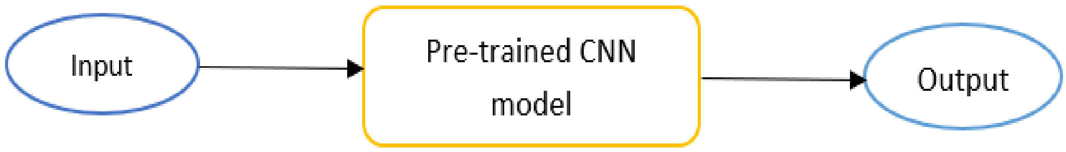

Fig4 workflow for lane detection

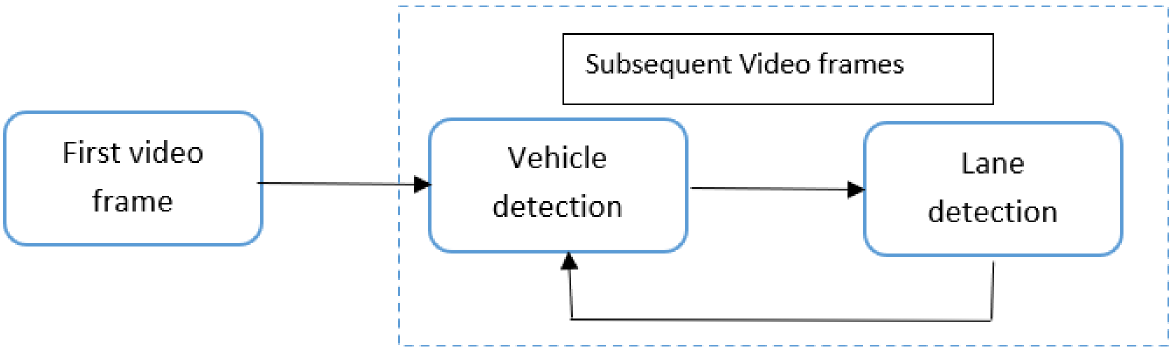

Fig5 Vehicle and lane detection workflow for video

3.2 ALGORITHM DESIGN

3.2.1 DATA PRE-PROCESSING FOR VEHICLE DETECTION

We took 2 datasets from the GTI source where 1st dataset is for vehicles and the 2nd dataset is for non-vehicles but here, we are checking for whether it is a car or not. So, we will be appending the vehicles to the cars list and the other to the non-car list and we will be pre-processing the data.

3.2.2 WORKING OF VEHICLE DETECTION

The Working process can be explained in the below steps and diagram:

1. Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images and train a Linear Support Vector Machine (SVM) classifier.
2. Apply a color transform and combine binned color features to the HOG feature vector.
3. Implement a sliding-window technique and use the trained classifier to search for vehicles in images.
4. Run the pipeline on a video stream and create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles. Estimate a bounding box for vehicles detected.

3.2.3 HOG and SVM

The fundamental idea behind HOG feature extraction is to find a sort of feature signature for images of vehicles inside an image by determining the gradient magnitude and gradient direction of each pixel within the image.

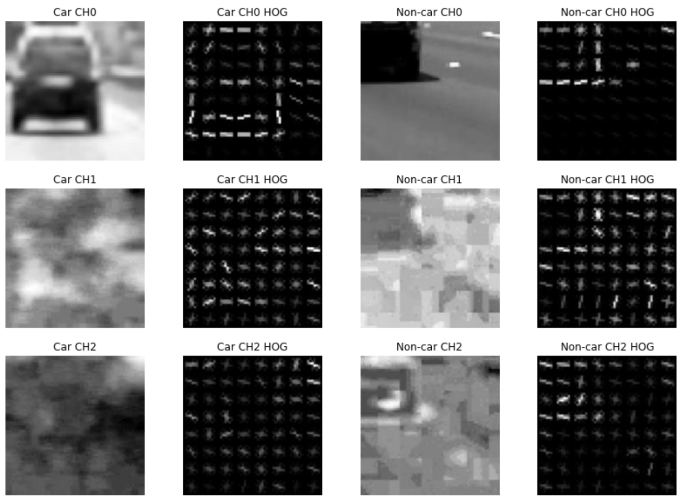

Fig6 HOG Comparison

In Fig. 3, a visualization of the HOG characteristics of an image with an automobile and an image without a car is displayed. An image with a car has a form of feature signature that we can use to train a classifier to detect whether or not it contains a car, as seen in the figure above.

3.2.4 COLOUR TRANSFORM TO THE *hog* FEATURE VECTOR

To make the approach more robust, we added features collected from color spaces and color histograms to the HOG features before training the SVM classifier. we tried with a few various color spaces before settling on YCrCb, which was recommended for HOG applications. One thing to remember is that the characteristics must first be normalized before they can be joined.

We used the YCrCb color space to calculate features for approximately 18,000 photos. After separating the photos into training and test sets, I used the training set to train the linear SVM. Function bin_spatial() computes binned color features by scaling images down, color_hist() computes color histogram features, get_hog_features() returns HOG features and visualization, and extract_features() wraps and combines the above functions.

3.5 IMPLEMENTING A SLIDING-WINDOW TECHNIQUE AND HEATMAP

Sliding window detectors find objects by inspecting every window. Typically, the size of the window is fixed. Since the window size is fixed, we cannot find people of different sizes. So, we should Shrink (down-scale) the image and slide again. It keeps shrinking and sliding we will obtain a full image pyramid, where we have to slide our detector at each scale by making sure that the scale differences across levels are small as in Fig.4

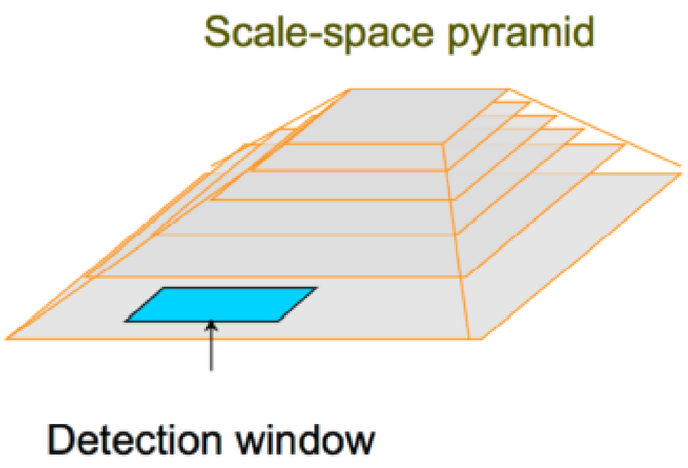

Fig 7 Full image pyramid

If the object is in different poses and orientations, we must devise a solution for this. So here comes the HOG feature extraction helping to plot the histograms of gradients and find suitable data for classification. SVM is the best classifier for multidimensional data and also classifies positive and negative classes. Now with the HOG featured image, take a window, crop out a feature matrix, and vectorize it. Doing this with all windows in an image and classifying them using SVM as car and non-car regions will give out the results which we are expecting. Computing the score $w' \cdot x + b$ in every location is the same as performing cross-correlation with template w (and adding b to the result) will be part of SVM and for threshold, the score can be like e.g., score > -1.

After Running our classifier algorithm, we will find the scores and hence the boxes which are higher than the threshold. So now we might get multiple boxes for the same car. To get rid of multiple detections we use the heatmap to label the frames and draw the bounding boxes using the open-cv library.

A heat map is a two-dimensional representation of data in which values are represented by colors. Heatmaps visualize data through variations in coloring. More elaborate heat maps allow the viewer to understand complex data sets. There can be many ways to display heat maps, but they all share one thing in common -- they use color to communicate relationships between data values that would be would be much harder to understand if presented numerically in a spreadsheet. When applied to a tabular format, Heatmaps are useful for cross-examining multivariate data, by placing variables in the rows and columns and coloring the cells within the table. Heatmaps are good for showing variance across multiple variables, revealing patterns, displaying whether variables are similar, and detecting if any correlations exist between them.

3.2.6 CNN MODEL

Lane detection is done by the CNN model. So, this CNN model is trained with the images. In the CNN architecture that we have built, we used 7 Convolution2D layers with activation function as 'Relu'(Rectified linear unit), and we also added 3 Max pooling layers. We also have 7 Deconvolution layers and 3 upsampling 2D layers. We have compiled the layers with Adam optimizer and Mean Squared error loss. Keras is using the following process when training a model with .fit_generator. Keras calls the generator function supplied to .fit_generator. The generator function yields a batch of size to the fit_generator function. The .fit_generator function accepts the batch of data, performs backpropagation, and updates the weights in our model. This process is repeated until we have reached the desired number of epochs.

3.2.7 LANE DETECTION WORKFLOW

Here, we will import the lane detection CNN model using the Keras model method called the load model. The next step is to create a class called lanes and define the image method of this class that will contain only two lists the recent fit list which is responsible for storing the most recent predictions and the avg_fit which will contain their average predictions. the CNN model will predict whether each pixel in the frame belongs to the lane or not it will do this for every frame of the input video. define a function that does some sort of pre-processing to the input video frame by frame we will call it road lanes first of all we need to resize each. Now we have to feed the image to the CNN model by using this method. we will keep only the predictions part by specifying the zero index and since the prediction will be between 0 and 1 we have to multiply by 255 in order to be able to use it as an image otherwise it will be so dark the prediction is appended to the recent fit list of the lanes class. If the length of this list exceeds 5 then we will discard the first element of the list each time and that means the list will only contain a maximum of 5 elements. This moving window of 5 predictions will be averaged using the main method from NumPy as shown here producing a black-and-white image where the brightness of each pixel indicates the probability of this pixel is a part of the lane or not. Then we will generate fake R and B color dimensions, and stack them with G. Then we will resize the lane image drawn to match the original image. Then merge the line drawing onto the original image. Then we can predict lanes.

The video input will be taken frame by frame and bound the vehicles with boxes. The bounded boxes of the vehicles are given as input to the lane detector.

CHAPTER 4

INFERENCE AND RESULTS

4.1 METRICS FOR EVALUATION

Both the Adam optimizer and the means squared error are two metrics that we have taken into consideration when developing the CNN model. Adam is the optimizer to use if one desires to train the neural network in a shorter amount of time and in a manner that is more effective. The name Adam was given to the optimizer due to the fact that it adjusts the learning rate for each weight in the neural network based on estimations of the first and second moments of the gradient. The Mean Squared Error is a statistical method that determines how closely a regression line corresponds to a given group of data points. It is a risk function with a value that corresponds to the expected squared error loss and its expected value. The square of the difference between the actual readings and the predictions made.

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

4.2 PARAMETER SETTINGS

Convolution Neural Networks take into consideration a parameter known as the Relu activation layer. The Rectified Linear Unit is the activation function that is employed the vast majority of the time in deep learning models. If the function is given any negative input, it will return the value 0, but it will return the value x it was given if the input was positive. Therefore, we can express it as f(x) equals max(0,x). By resolving the issue of vanishing gradients, the rectified linear activation function enables models to learn more quickly and improve their overall performance. One of the parameters that is introduced to the CNN layers is called the Dropout layer. A single model can be used to mimic a very large number of different network architectures simply by excluding nodes at random while it is being trained. This method is known as random node removal. Dropout is a regularisation technique that can reduce overfitting and improve generalisation error in all varieties of deep neural networks. It is also extremely computationally economical and has an excellent success rate in achieving these two goals.

4.3 RESULTS AND DISCUSSION

Here, in this project, we have appended the cars and non-car images separately from the dataset and used them.

Fig 8 Car images and non-car images

Below is a region proposer that scans select portions of the image and checks for the presence of a car. The cell size and cell position are determined by the way the road is laid out, and the distance of vehicles from the camera.

A smaller search area farther away is given smaller cells to search within, given that cars appear smaller further away. It's clear right away that the search region covers areas where automobiles are unlikely to be found (like the left of the highway barrier). Regardless, including this search zone makes logical because the motorist might opt to travel in the center lane instead.

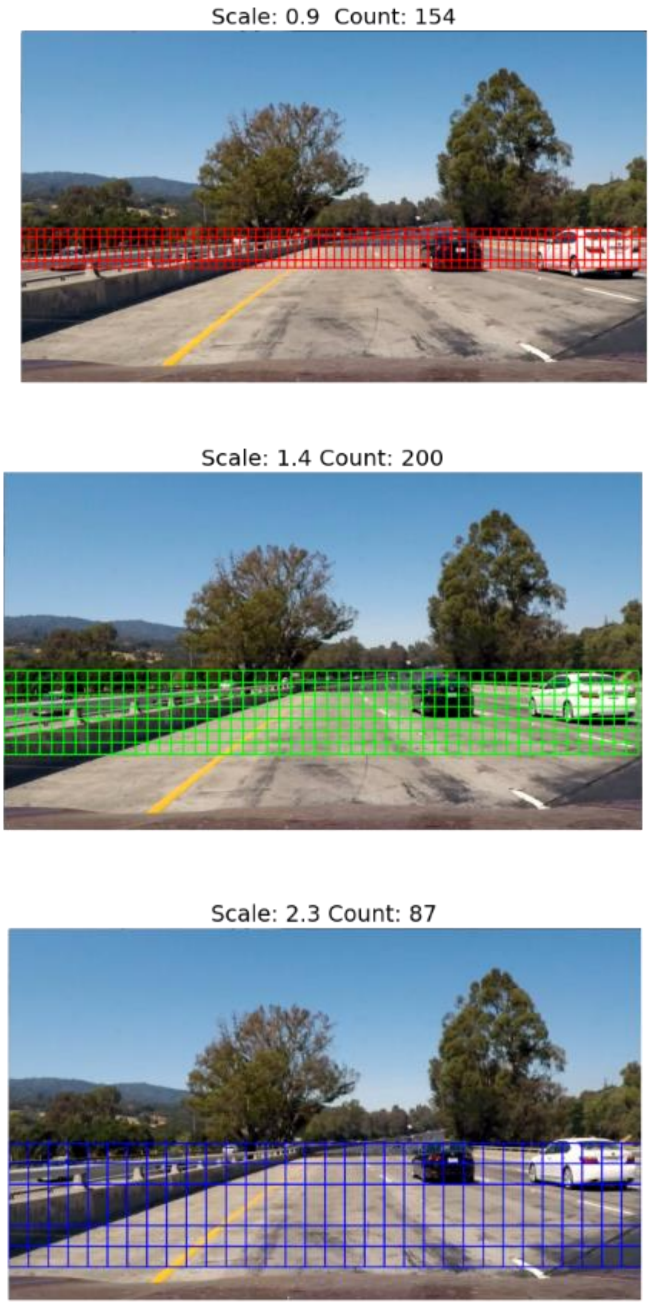

Fig 9 sliding window search

This step helps reduce false positives that would be misclassified over consecutive frames. The result is improved accuracy and a smoother overall result. Below is the visualization of the heatmap next to the resulting bounding boxes.

Fig 10 Temporal Heatmapping

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENTS

5.1 CONCLUSION

Although the SVM classifier approach works for this test scenario and is a good way to grasp the fundamentals behind vehicle detection and tracking, it is also sluggish and computationally costly. As a result, this may not be appropriate in real-world scenarios with a variety of cars, pedestrians, bicycles, and incoming automobiles, among other things. Another concern is that training the SVM to eliminate false positives requires a lot more data, which would make the present technique slower. Some automobiles that are further away and at varying distances are not spotted in the results video. This challenge can be solved by combining many sliding windows of various widths and forming estimated zones of interest based on the car's size about distance. This solution will also take a long time to compute.

5.2 FUTURE ENHANCEMENTS

A neural network or a YOLO (You Only Look Once) classifier system, which operates in real-time and classifies automobiles as well as many other things, might be a superior strategy for car recognition and tracking.

CHAPTER 6

REFERENCES

1. R. K. Satzoda and M. M. Trivedi, "Efficient Lane and Vehicle Detection with Integrated Synergies (ELVIS)," *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 708-713, DOI: 10.1109/CVPRW.2014.108.
2. H. Guan, W. Xingang, W. Wenqi, Z. Han and W. Yuanyuan, "Real-time lane-vehicle detection and tracking system," 2016 Chinese Control and Decision Conference (CCDC), 2016, pp. 4438-4443, DOI: 10.1109/CCDC.2016.7531784.
3. Lim, King Hann, et al. "Lane-vehicle detection and tracking." Proceedings of the international multiconference of engineers and computer scientists. Vol. 2. 2009.
4. S. Sivaraman and M. M. Trivedi, "Integrated Lane and Vehicle Detection, Localization, and Tracking: A Synergistic Approach," in IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 2, pp. 906-917, June 2013, DOI: 10.1109/TITS.2013.2246835.
5. Wally Chen, Leon Jian, and Sy-Yen Kuo, "Video-based on-road driving safety system with lane detection and vehicle detection," 2012 12th International Conference on ITS Telecommunications, 2012, pp. 537-541, DOI: 10.1109/ITST.2012.6425237.