International Journal of Communication Networks and Information Security
ISSN: 2073-607X, 2076-0930
Volume 15 Issue 03 Year 2023

Vehicle Detection and Speed Estimation Using Semantic Segmentation with Low Latency

Mr. Sathisha G*
Research Scholar, BGSIT, Adichunchanagiri University, B G Nagar
write2sathisha@gmail.com
Subbaraya C K
Registrar, Adichunchanagiri University, B G Nagar
registrar@acu.edu.in
Ravikumar G K
Principal, BGSCET, Bengaluru
ravikumargk@yahoo.com

Article History

Received: 28 April 2023
Revised: 18 June 2023
Accepted: 10 July 2023

Abstract

Computer vision researchers are actively studying the use of video in traffic monitoring. TrafficMonitor uses a stationary calibrated camera to automatically track and classify vehicles on roadways. In practical uses like autonomous vehicles, segmenting semantic video continues to be difficult due to high-performance standards, the high cost of convolutional neural networks (CNNs), and the significant need for low latency. An effective machine learning environment will be developed to meet the performance and latency challenges outlined above. The use of deep learning architectures like SegNet and Flownet2.0 on the CamVid dataset enables this environment to conduct pixel-wise semantic segmentation of video properties while maintaining low latency. In this work, we discuss some state-of-the-art ways to estimating the speed of vehicles, locating vehicles, and tracking objects. As a result, it is ideally suited for real-world applications since it takes advantage of both SegNet and Flownet topologies. The decision network determines whether an image frame should be processed by a segmentation network or an optical flow network based on the expected confidence score. In conjunction with adaptive scheduling of the key frame approach, this technique for decision-making can help to speed up the procedure. Using the ResNet50 SegNet model, a mean IoU of "54.27 per cent" and an average fps of "19.57" were observed. Aside from decision network and adaptive key frame sequencing, it was discovered that FlowNet2.0 increased the frames processed per second to "30.19" on GPU with such a mean IoU of "47.65%". Because the GPU was utilised "47.65%" of the time, this resulted. There has been an increase in the speed of the Video semantic segmentation network without sacrificing quality, as demonstrated by this improvement in performance.

Keywords: Semantic Segmentation, Vehicle Detection, Decision Network, SegNet, FlowNet, ResNet

1. Introduction

Intelligent Transportation Systems (ITSs) utilize high-tech cameras and smart sensors to monitor transportation networks, mainly using video cameras for their depth of information and ease of installation and maintenance. Unlike older technologies like inductive loops, laser, and infrared sensors, which categorize vehicles based on features like length, axles, and wheel distance, video-based ITSs offer more comprehensive data without the need for invasive installations. Older systems had limitations in providing advanced traffic characteristics and were costly to install. Video-based ITSs, on the other hand, leverage existing traffic control center cameras, reducing installation costs and disruptions. They provide detailed information enabling automatic incident detection, law enforcement, monitoring of weather conditions, and tracking vehicle trajectories. These systems can identify vehicle classification, speed, and additional details like make and owner, through sophisticated video-processing techniques, making them a more efficient and informative solution for managing traffic flow and reducing congestion.

Every pixel in an image or video frame should be able to be recognised, categorised, and labelled by the deep learning model when it receives the input of an image. Previously, semantic segmentation was investigated utilizing 2D information including colour and shape [1] and dense depth maps [2] needed substantial human engineering that was computationally expensive. With the advent of deep learning methodologies and machine learning advancements, the issues of semantic segmentation and classification have been made easier to address without the need for human engineering. Using CNN for semantic segmentation of images has yielded impressive results [3], [4] in analysing and labelling the features of images with improved performance. Both in terms of the quality of the findings and the speed of computation, CNN has had a significant impact on computer vision.

Video segmentation has been the focus of a large number of studies [5], [6], [7] aimed at improving its performance. There has been little study done to reduce latency [8]-[11]. Semantic segmentation research has been carried out by several significant projects, including FCN, PSPNet, U-Net, SegNet, and DeepLab [12]-[15]. It is well-known that the aforementioned networks take a long time to process a single frame; nonetheless, they are extremely accurate in predicting the future. This research aims to provide a video semantic segmentation framework that can be used in real-time applications like self-driving cars and in-building navigation. Achieving low latency using approaches employed in previous studies on dynamic video segmentation (DVSNet) and deep feature flow (DFF) and involving two networks is critical for real-time applications [9], [11]. In both DVSNet and DFF, the optical flow and per frame segmentation networks are present. The decision network, a mechanism for adaptively updating keyframes, and other approaches such as label mapping and depth inferencing are among the most essential parts of this research. To see how the network responds when an encoder-decoder design like ResNet50 SegNet is used instead of the deep lab as a segmentation network while maintaining the same flow network as [9] and [11], wish to compare the results. CamVid [16] was used for the experiments, which is a driving scenario dataset very similar to the CityScape [17] that was used in the previous two papers.

This paper presents a formulation for the solution of Track1. Vehicle recognition and tracking play a crucial role in our methodology. However, due to the lack of available labels, training a vehicle detection model from scratch is a formidable task in this Challenge. Instead, we use the 3D Deformable model [15] to do inference on our dataset, a transfer learning technique, for vehicle detection. We also took into account an alternative to the 3D Deformable model by Zhang et al. [2], whose model performed similarly well in the 2017 challenge. We evaluate the two models' performance on the target reference data by extracting all frames from one of the Track 1 films and measuring their performance on the frames. We next compare the models' vehicle identification ability by measuring the mean Average Precision (mAP) Of comparison to the model by Zhang et al. [2], the experimental results reveal that the 3D Deformable model [15] obtains 74% mAP. Our tracking system employs a detect-then-track approach. Thus, the detection precision greatly affects the tracker's efficiency. In order to identify the automobiles that have been spotted in each frame, we first extract key features from them. To calculate the speed of a moving object, it is important to take into account the shift in the frame's location of these elements.

2. Related Works

Accurate speed estimation methodology requires the use of vehicle tracking. Several object-tracking techniques employ computer vision and machine learning. In their study, Kale et al. employed optical-flow and motion vector estimation techniques for object tracking [11]. The authors introduced a method that involves analyzing the track of an object using optical-flow detection and estimating the speed of motion vectors. Geist et al. combined reinforcement learning with a Kalman Filter to address dynamic settings [8]. According to the study, the act of watching video objects can be interpreted as predicting the location of the bounding box for each frame. Zhang et al. developed a model using a combination of reinforcement learning and recurrent convolutional neural networks [18]. Faragher provided an explanation of Kalman Filters [6]. The paper presents an introduction to Kalman Filters and systematically models a tracking problem. Mihaylova et al. employed a particle filtering technique to track objects that are both nonlinear and non-Gaussian by including characteristics [19]. Several studies have been conducted to detect the speed of different vehicles. Rad et al. suggested using the comparison of vehicle positions in consecutive frames to predict the speed of traffic using stationary digital film [14]. The camera was calibrated by applying geometric formulas. The method developed by Rad et al. can be applied to various places and exhibits an average speed inaccuracy of 7 km/h.

Liu and Yamazaki employed panchromatic and multispectral QuickBird imagery to discern velocity. Liu et al. [9] utilized Kalman filters and optical flow to estimate speeds. The first option prevents temporary blockages, while the second option provides more precise speed delivery. In his study, Wang presented a technique for identifying mobile objects by converting the distance between pixels into real-world distance [7]. This strategy employed three methods: frame differencing, background differencing, and feature extraction, to identify and extract attributes from moving vehicles. The process of tracking and determining the position was accomplished by extracting the centroid feature of the vehicle.

Semantic segmentation is one of the most recent advancements in computer vision. Providing pixel-level categorization and masking of an image in the past was extremely difficult due to the high number of manual computations that were required necessary to segment an image, Dense depth mapping, 3D geometry, and superpixel co-existence are a few examples of these computations. Many picture segmentation and classification data sets were generated manually in the later part of the 21st century using various software tools. It has been completely revolutionized with the introduction of deep learning and convolutional neural networks. Research on semantic segmentation and easy data creation has resulted from this. In semantic segmentation networks before the invention of FCN, only convolutional networks with a fully connected layer at the output were used [20], [21], and [22]. FCN made it a top priority to convert the classification layer's last fully connected layer to a convolutional layer. The number of parameters is reduced, input image size constraints are removed, and spatial information is preserved by using this classification approach, which entails eliminating a layer that is fully connected.

The expanding and contracting elements of Ronneberger et al. 's U-Net concept for medical images were both influenced by FCN [23]. As with DeConvNet [24], it makes use of downsampling and upsampling to keep track of data that might otherwise be lost during downsampling. During downsampling, the contracting section is responsible for primarily computing the feature maps and employing 3x3 convolution to extract features. A further fully convolutional network called SegNet [15] was developed specifically for road scene segmentation. It features an encoder-decoder architecture, convolutional layers, batch normalisation, max pooling, ReLU activation function, and a Softmax classifier. The VGG-16 network (encoder including the first 13 convolution layers of vgg-16) is used as an encoder in SegNet [25].

Another significant design is known as the deep lab, and its purpose is to solve the problem of loss of spatial resolution that occurs as a result of the repetitive max pooling and downsampling that is carried out at successive deep convolution layers (DCNNs). In addition to this, it solves the problems that arise as a result of the presence of objects of varying scales and a lack of precise localization. They decided to take a novel method to restore feature maps at full resolution by employing a technique known as "atrous convolution." This technique involves upsampling the filters before conducting the convolution. To accomplish this, they replace the downsampling process with filter upsampling in the convolutional layers that come after the maximum pooling layers of fully convolutional layers. This is done since the downsampling process causes errors in the maximum pooling layers. The computation of feature maps at large sample rates is facilitated as a result of this. The original image dimensions are obtained by performing an atrous convolution, then following it up with a bilinear interpolation of the feature maps [16], [26].

A variational technique is one of the oldest methods for determining optical flow. This method was the standard for a considerable amount of time. The variational method primarily concentrated on relatively tiny motions and gave preference to initializations that involved no motion field because of the existence of local minima [27]. In addition, Brox and Malik [28] came up with the unify technique, which is a patch-based approach that incorporates variational refinement.

PatchMatch [29], FlowField [30], DiscreteFlow [31], and FullFlow [32] are some examples of algorithms that were built utilising the Brox and Malik method. CRF [33], [34] is utilised in the processing of vector similarities by both DiscreteFlow and FullFlow. DeepFlow [35] is a noteworthy method developed by Weinzaepfel and colleagues, which was influenced by Brox and Malik. DeepFlow is a variational optical model that also incorporates a loss function. It is founded on a deep matching algorithm, which makes it possible to combine feature descriptors and matching. EpicFlow and RichFlow are a couple of the few more well-known unified or combined matchings approaches [36], [37].

Semantic segmentation and computer vision have begun to apply deep learning and optical flow approaches in recent years. FlowNet uses a single-stacked architecture to estimate the flow of optical data by employing the generic CNN architecture. It takes two photos as its input and produces flow fields as its output. FlowNet demonstrated two distinct architectural designs, which were referred to as FlowNetS (FlowNet simple) and FlowNetC. (FlowNet correlation). FlowNet was meant to manage huge displacements and was trained on an image dataset consisting of a flying chair and three-dimensional items [38].

3. Methodology

This research builds on the dynamic video segmentation network (DVSNet) to detect vehicles and estimate their speed from video frames with low latency. By integrating a per-frame segmentation network with an optical flow network using spatial warping, the model achieves accurate speed estimation. The study employs three low-latency networks for video semantic segmentation, differing from SegNet and FlowNet2S. It uses a SegNet with a ResNet50 encoder and spatial warping function for segmentation, and Judgment networks (DN) to analyze differences between consecutive frames. Unlike DVSNet's frame region technique, this study adopts an adaptive key frame scheduling policy for the entire frame, enhancing the performance and accuracy of vehicle speed detection in video analysis.

3.1 Segmentation Network

Segmentation Network (SegNet) is an convolution neural network architecture which comprises of several layers like Convolution, batch normalisation, ReLU activation and maximum pooling layers from which all features of input sigals are encoded in the SegNet. Upsampling, deconvolution, and a Softmax classifier are all included in the decoder as shown in Figure 1. Thirteen convolution layers are used in the SegNet encoder. Our research work, on the other hand, utilises ResNet50, a more complex network than VGG16, as an encoder and decoder. ResNet50 weights can be loaded into ImageNet for initial training. ReSNet is the name of the ResNet50 encoder/SegaNet decoder combination we use. For the encoder output, SegNet is a fully convolutional network that has had its fully connected layer removed. So the encoder network has fewer variables to deal with.

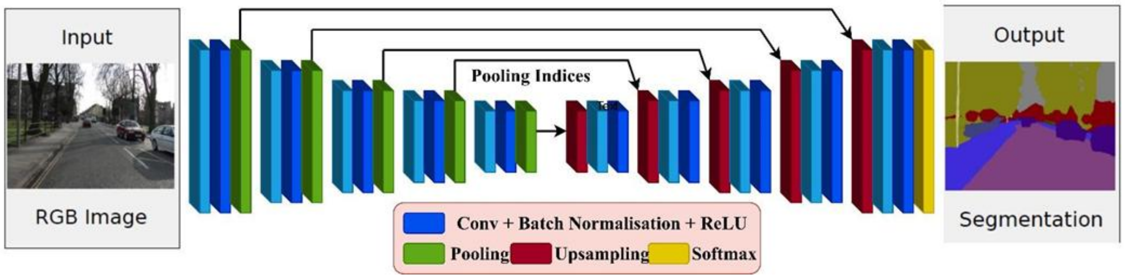

Figure 1. Figure Illustrating SegNet Architecture

In SegNet, an encoder convolutes input frames to create feature maps, which are then batch normalized and processed with ReLU. Max pooling with a 2x2 non-overlapping window and stride 2 ensures translation invariance, but reduces resolution, impacting semantic segmentation. SegNet addresses this by storing max-pooling indices, not all border information. These indices help decoders upsample sparse feature maps, which are then convolved into dense maps. A softmax classifier predicts class probabilities. Although processing each frame of a video is time-consuming, SegNet, especially with a ResNet50 encoder trained on CamVid, excels in segmentation accuracy.

3.2 Flow Network

FlowNet is an optical flow technique that employs a convolutional neural network. Optical flow is the term used to describe the displacement of each pixel in an image resulting from the relative motion of objects between frames or due to a camera in motion. The vehicle tracking technique we use depends on the localization findings obtained from the vehicle detection methods explained in Section 4.1. These results are improved by using optical-flow based features, resulting in a reliable vehicle trajectory.

3.2.1 The Process of Tracking an Object by Detecting its Presence

In our 1080p, 30 fps footage, object tracking leverages the small movement between frames. We use detection confidence to assign IDs and track objects by intersecting bounding boxes across frames, calculating Intersection over Union (IOU) scores within a frame window (h frames). While effective, this method risks identity changes if detection fails over h frames and can yield imprecise bounding box localizations, affecting speed calculation accuracy due to pixel shifts in consecutive frames.

3.2.2 Flow-based Tracking

In order to improve traditional tracking methods that rely purely on object recognition, we utilize the iterative Lucas-Kanade technique with pyramids, which was invented by Bouguet [3, 13]. This technique allows us to calculate the optical flow given a limited number of observed object features, notably Shi-Tomasi corners [17]. Within the realm of flow estimation, corners are characterized as small regions in the UV space that exhibit significant intensity change across the whole visual field. The Harris corner detector [10] identifies potential corners by calculating the eigenvalues 1 and 2 of the matrix, as shown in Equation 1 and 2. However, the Shi-Tomasi corner detector [11], [12] enhances this approach.

$$M = \sum_{x,y} w(x,y) \left[I_xI_x \; I_xI_y \; I_xI_y \; I_yI_y \right] \qquad (1)$$

w is a function that weighs the contribution of derivative windows in the composition, and I_x and I_y are the frame derivatives in the x and y directions, respectively.

$$R1 = (\lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \qquad (2)$$

Harris's criterion for identifying corners in windows was based on a certain threshold. However, Shi and Tomasi showed that windows with an R2 value equal to or less than 1 were more likely to exhibit corner features. Our method consists of delaying the filtering of the points that form the selected corners of a frame until all of them are contained within at least one bounding box provided by our vehicle detector. The Lucas-Kanade technique relies on the assumption that the intensity of a point remains constant between consecutive frames and that both the point and its neighboring points move in a similar manner. It accomplishes this by comparing the positions of the chosen point and its adjacent points in the two frames and approximating the resulting movement. In our methodology, we store the recorded coordinates of each corner of a tracked object in the vast majority of preceding frames. The obtained tracklets function as an indicator for quantifying the movement of the vehicles indicated by the corner points.

3.3 Speed Estimation

Our system for estimating vehicle speeds is based on data and relies on several underlying assumptions. A stationary traffic camera is mandatory for the 2018 AI City Challenge. Furthermore, we assume that there is at least one vehicle traveling at the highest permissible speed on the recorded sections of the route. The algorithm calculates the speeds of vehicles based on their local motion and the maximum speed, Smax, at which a vehicle is predicted to be driving in the movie. Equation 3 establishes the relationship between local vehicle motion and the highest historical corner point motion seen within the vehicle's tracklets.

$$\Delta m = \ percp\ \left(\left(\| T_i(j) - T_i(j-1) \|_2 \right)\ \right) \qquad (3)$$

The research tracks vehicle movement in video frames, where Ti represents a vehicle's track, Ti(j) a point in its history, and |Ti| the tracklet size. The Perc function removes outliers by analyzing tracklet motion distributions. A 3×4 projection matrix P, derived from vanishing points and camera settings, standardizes object mobility, considering varying camera perspectives. This approach accounts for apparent velocity changes due to camera angles, especially for vehicles moving vertically towards the horizon, and uses a tile-based anticipated velocity (PS) model to accurately estimate vehicle speeds.

$$S = \frac{\Delta m}{\Delta m} \times S_{max} \qquad (4)$$

where max T Δm is the greatest distance traveled locally by any tracklet belonging to a vehicle that traversed the tile while the current vehicle was in its window. We take into account the highest estimated vehicle speed during the previous h frames to further remove extreme values [0, smax]. Finally, we restrict the projected speed to be between zero and the maximum speed. The majority of the vehicles in Loc 1 and 2 films go at consistent speeds because the roads are not congested. After first filtering based on confidence and track overlap, we think about a second constant speed (CS) model that gives all detected vehicles in the video the input max speed.

To compute the object motion correction, we will utilize FlowNet 2.0. Correlation, Simple, and Small Displacement are the three levels that make up FlowNet2.0's layered network architecture as shown in Figure 2. Massive displacements between objects in an image frame can be handled by FlowNet-C, FlowNet-S, and FlowNet-SD respectively. The datasets that are used by FlowNet-S and FlowNet2S are distinct.

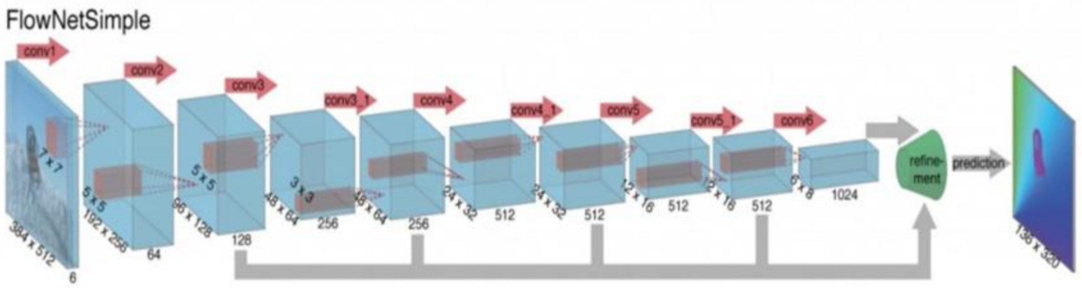

Figure 2. Architecture of FlowNet-Simple

FlowNet2-Simple, an encoder-decoder architecture, generates flow fields for two images. For warping, FlowNet2S requires 2 frames key frame (Ik) and current frame (Ii) as the input and output flow field as F. Only flow fields are provided by FlowNet2S; segmented output is not available. To forecast the final segmentation output, we will combine the output of the segmentation network with the output of the optical flow network. Or the flow of the network. The feature maps that are generated at each convolutional layer represent information about the spatial movement that has occurred between the keyframe and the frame that corresponds to it. When compared to per-frame segmentation, FlowNet2 is significantly faster. This decreases the amount of time spent computing, which increases segmentation speed.

3.3.1 Spatial Warping

Additional layers of convolutional processing encode both the semantics and the spatial location of an image Because of these spatial maps, the propagation of feature maps through spatial warping can be done at a low cost. Flow fields Mi>k can be determined by employing a technique for flow estimate known as F, which utilizes the Equation 5:

$$\mathrm{M}_{\mathrm{i}\to\mathrm{k}} = F\,(Ik, Ii) \qquad (5)$$

Propagating feature maps are expanded to the same dimensions as these flow fields bi-linearly. Object location p in a current frame I become p + δp in keyframe k, Where, δp = M_(i→k)(P). Feature warping is achieved by bilinear interpolation using Equation 6 and 7 respectively.

$$f_i^c(p) = \sum_q G(q, p + \delta p) f_k^c(q) \qquad (6)$$

Where, q means Enumeration of all spatial locations in feature maps and F means Feature maps.G means Two-dimensional bilinear interpolation kernel,C means Total channels in feature maps.

$$G(q, p + \delta p) = g(q_x, p_x + \delta p_x).\, g\left(q_y, p_y + \delta p_y\right) \qquad (7)$$

Where, g(a,b) will be equal to max(0, 1-|a-b|).

There is a possibility that the spatial warping is incorrect because of flaws in the calculated flows, object occlusion, and other factors. It is necessary to modify the amplitudes of the features using "scale fields" S_(i→k) to achieve a more accurate estimation of the characteristics. The spatial and channel dimensions of a scale field are identical to those of a feature map. Scale fields have the same dimensions as feature maps. The following formulae are used to calculate the scale field, which is the scale function applied to both the keyframe and the current frame which is as depicted in Equation 8:

$$S_{i\to k} = S(I_k, I_i) \qquad (8)$$

The following Equation 9 & 10 are used to calculate propagated feature maps:

$$fi = W(f_k, M_{i\to k}, S_{i\to k}) \qquad (9)$$

$$fi = W(f_k, M_{i\to k}, S_{i\to k}) \qquad (10)$$

In this case, W applies feature warping in Equation 10 to all of the regions and channels in feature maps before multiplying them with scale fields.

3.3.2 Label mapping and Working of Decision Network

Label mapping combines FlowNet2S and ResNet50 SegNet target labels. FlowNet2S has 19 labels and ResNet50 SegNet has 12 labels on the cityscape dataset. Both datasets contain road scene data. Both datasets have similar labels, hence their intersection produces our label set. Our 11-label collection. After mapping, one CamVid label is 'Unlabelled' and eight Cityscape labels are disregarded.

The decision network (DN) consists of four layers: one convolutional layer, three fully linked layers, and the top layer. DN forecasts segmentation output using a flow or segmentation network path. DN takes hidden layers (4 to 6 of Convolutional layer) and generates a predicted confidence score depending on the ground truth pixel difference between O_{out} and S_{out} (segmentation network path and output). F predicts O_{out} on the decision network (estimated flows). Frame ground truth confidence score:

$$\sum_{p\epsilon P} \frac{\left(C\left(O_{out}(p), S_{out}(p)\right)\right)}{P} \qquad (11)$$

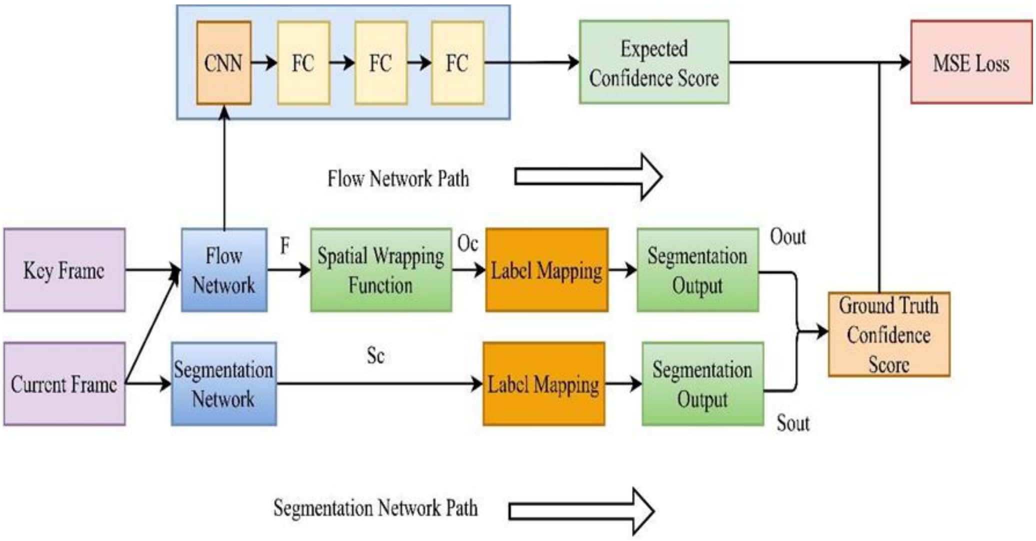

Figure 3. Figure Illustrating a Decision Network Architecture

Our network employs an inference threshold 't' to alternate between segmentation and flow paths, based on frame confidence scores. High scores trigger the ResNet50 SegNet for accurate segmentation, while lower ones engage FlowNet2S for optical flow analysis, refined by spatial warping and label-mapping for 11 classes. The decision network, informed by key and current frame differences, dynamically switches paths for efficiency. We tested inference times using NVIDIA Tesla T4 GPUs on Google Colab, noting SegNet's precision but slower speed compared to the quicker, less accurate flow network. An adaptive keyframe scheduling policy further optimizes processing time, updating keyframes when significant frame differences are observed, thereby balancing speed and segmentation quality.

4. Results and Discussion

4.1 Data Set

In contrast to the AI City Challenge that took place in 2017, which centered on the application of supervised models to traffic-related issues and consequently included a substantial amount of collaborative annotation work for the dataset, the focus of this competition was on transfer learning approaches, and the dataset does not contain any annotations at all. The data set that is now accessible was acquired from stationary cameras that were installed at a number of different metropolitan intersections and highways. The description that can be seen below provides a comprehensive breakdown of the dataset, which includes the following information at length.

• The Track 1 dataset contains thirty-seven 1080p videos, each of which is one minute in length and was captured at thirty frames per second (fps). These videos were recorded in 1920 x 1080 resolution. The filming of these movies took place in four different locations, with sites 1 and 2 being highways and locations 3 and 4 being crossroads, respectively.

• There are a total of one hundred videos that are included in the Track 2 dataset. Each of these videos is around fifteen minutes long and was captured at a resolution of 800 by 410 pixels at a frame rate of thirty frames per second.

A total of fifteen videos with a resolution of 1080p and a frame rate of thirty frames per second are included in both the Track 3 data set and the Track 4 data set. These films were captured in four different locations across the world. Every movie can last anywhere from half an hour to one and a half hours.

4.2 Training

In our research, we integrate a decision network, a segmentation network, and a FlowNet2S optical flow network with warping. SegNet, trained on ImageNet using ResNet50 and fine-tuned on CamVid, handles segmentation, while FlowNet2S is trained on the Cityscapes dataset. Transfer learning with ResNet50 accelerates training and reduces generalization error. The segmentation output is mapped to 11 classes, excluding the "unlabelled" category from CamVid. The decision network, trained on 6th-layer flow features from FlowNet2S, undergoes 500 epochs in 32-epoch batches with a 0.99 decay rate, enhancing prediction accuracy for frame confidence scores. This setup facilitates efficient path selection between segmentation and flow, evaluated using the S1 metric combining vehicle detection rate and normalized RMSE.

4.2.1 Outcome Results of the Baseline Model

Here, we compare experiment 1's results. Figure 4 and 5 show the expected output of ResNet50 SegNet and Original SegNet, with input picture on the left, anticipated output on the right, and ground truth in the middle. In the expected output, each class is allocated a different hue, such as a purple car and a red building, comparable to ground truth images. Figure 5's predicted output was quite comparable to its ground truth label and Figure 4's authorized SegNet predictions. Few classes had distorted segmentation maps. Figure 5's pole segmentation mask is not smooth compared to the ground truth.

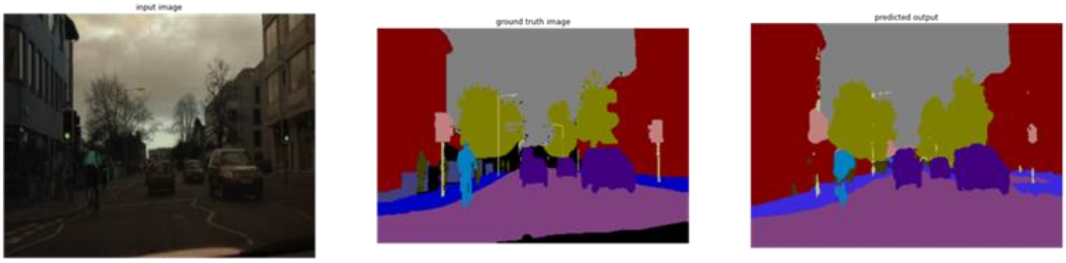

Figure 4. Authorized SegNet Implementation

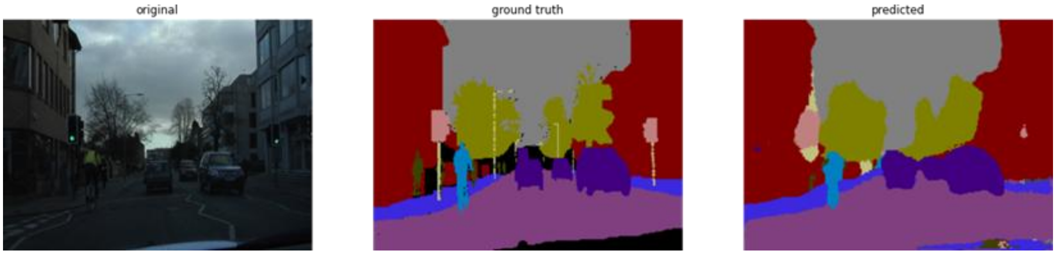

Figure 5. Customised ResNet50_SegNet

Table 1 shows the IoU for each of the 11 classes. At 50 epoch, classwise IoU on the test set produced great findings, and the results appear to be better compared to previous epochs, with the exception of 'pavement,' in which MIoU declined by 2%. When the epoch were raised to 100, the class-wise IoU of building, cyclist, fence, pedestrian, and tree declined by over than 1% on testing data. This decrease in accuracy for more categories as epochs increased could be triggered by the model's overfitting, which lacks versatility and performs poorly on new data. Errors like this are dangerous in automation driving and should be avoided to generalize the model. After analysis, epoch 50 was chosen to retain class generality.

Table 1. Each Class Label's IoU Classwise

ResNet50_SegNet						
Classes	ClassWiseIoU in % on Test Set					
	10 Epochs	20 Epochs	30 Epochs	40 Epochs	50 Epochs	100 Epochs
Road	87.24	87.35	88.54	87.69	89	90.64
Fence	21.87	27.24	26.88	26.87	28.55	28.21
Sky	85.93	87.34	87.60	87.95	87.68	88.53
Tree	70.22	71.45	72.24	73.25	73.05	74.21
Pole	4.5	7.88	12.8	12.57	14.24	16.87
Building	74.41	75.24	76.51	75.88	76.87	77.52
Pavement	65.24	67.45	70.24	70.57	69.25	72.45
Pedestrian	25.56	22.36	33.21	34.52	34.87	34.01

Bicyclist	12.8	23.04	28.35	27.54	33.33	38.87
Sign Symbol	29.87	32.87	32.05	32.52	36.45	35.99
Car	71.20	73.01	72.88	74.52	74.68	74.29

On the CamVid test set, ResNet50 SegNet obtained a mean IoU of 54.27 percent and 1.92 mean fps, and 19.57 fps on CPU and GPU (table 2). On GPU, a lightweight and moderately deep encoder network like ResNet50 achieved greater speed than DVSNet's deeplab-fast network. We assume deeper networks like ResNet100, ResNet152 can increase prediction quality. Table 2 compares original SegNet and ResNet50_ SegNet results. MIoU of our baseline network differs by roughly 6% from SegNet benchmark findings, demonstrating that ResNet50 SegNet achieved equivalent segmentation quality. Although we failed to obtain better scores than the original method utilizing ResNet50 as an encoder, we did so in fewer iterations (nearly 100k).

4.2.2 Outcome Results of Video Semantic Segmentation Network

This section describes experiment 2's results. When the Optical flow network was combined with the baseline model, Figure 6's estimated segmentation output quality fell and looked distorted. ResNet50 SegNet's result was similar to that of the flow network path selected by the model, however, several class forecasts were distorted. Also, the network only partially predicted classes like poles and fences. False predictions included a sign symbol. Pole can be a significant symbol. Information loss during flow creation may produce partial segmentation and distortions.

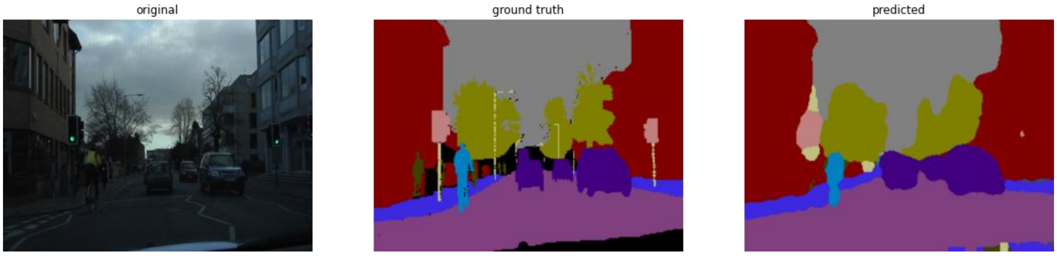

Figure 6. ResNet50_SegNet+FlowNet2S

While in balancing mode, the integrated network scored 47% MIoU and CPU and GPU fps of 16.84 and 31.27 (Table 2). Flow network integration boosts processing speed while reducing quality by 6%. Loss of information during flow generation may explain the 6% quality drop. When the Flow network path is chosen, spatial warping helps preserve quality near the segmentation network. As the difference between consecutive frames grows, spatial warping of segmentation network output with lossy flows produces a drop in quality and MIoU values.

Our video segmentation network is faster than per frame thanks to FlowNet2S, a fast optical flow network. The optical flow network simply creates flow information, not segmentation. The spatial warping of optical flow network flows with newly segmented output from the segmentation network path helps generate segmentation quickly and preserves output quality. The segmentation network is slower, thus the optical flow network uses that speed. Our two-network system gained more than DVSNet [9]. Table 3 has details. Our network gained 11.7 fps, compared to 14.2 fps for DVSNet. Our network increased fps by 31.27 compared to DVSNet, however, the benefit was less. The slower a network's segmentation, the greater benefit it will see. DVSNet's DeepLab-Fast is slower than Resnet50 Segnet, therefore gain is comparable.

Table 2. Comparison of SegNet [15] Mean IoU Scores with our Developed Model ResNet50_SegNet (Our Baseline) and ResNet50_SegNet+ FlowNet2S

Network	**Mean IoU**	**Average fps on CPU**	**Average fps on GPU**
SegNet (original implementation)	60.07%	-	-
ResNet50_SegNet (our baseline)	54.27%	1.92	19.57
ResNet50_SegNet+	47.65%	16.84	31.27

FlowNet2S			

Table 3. Gain in fps Using Two Network Approaches

Inference on GPU	**Segmentation Network**	**(Segmentation Network + FlowNet2S)**	**fps Gain**
Our Implementation	19.57 (resnet50_segnet)	31.27	11.7
DVSNet*	5.6 (deepLab-Fast)	19.8	14.2

We apply our tracking and speed estimation models to the Track 1 video that is included in the CamVid dataset, and then we examine the findings that we get from doing so in this section. First, we will present our result, and then we will investigate alternative ways in which our model could be improved. As shown in Figure 7, our best performing model was a constant speed model with maximum speeds of 70, 70, 50, and 30, respectively, for all of the sites ranging from track 1 to track 4.

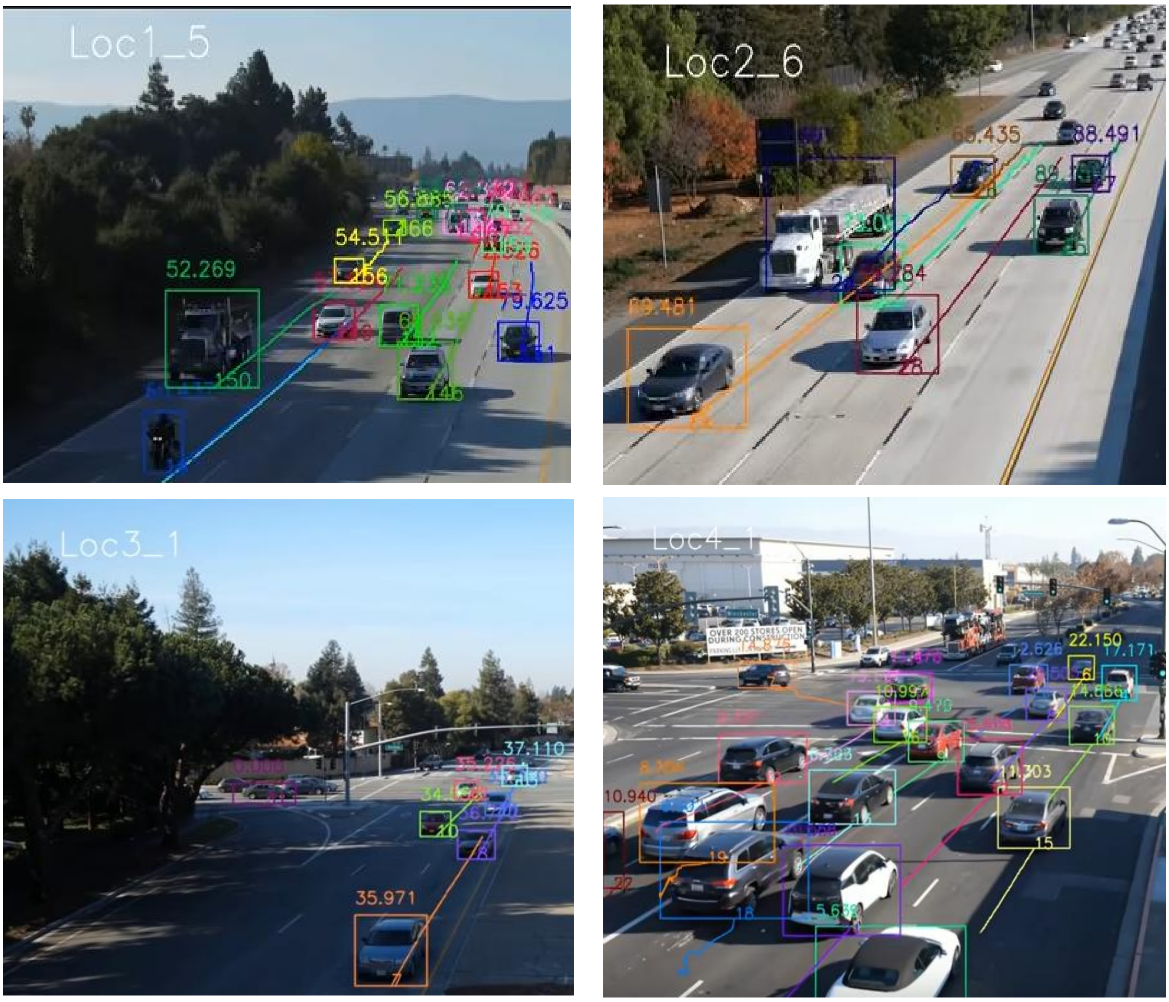

Figure 7. Tracklets Obtained Through Optical-flow Estimation

The Predictive Speed (PS) model estimates vehicle speeds using the optical-flow-based tracklet detection, and it will only output a detection if the vehicle's speed can be assessed accurately. As a consequence of this, some of the vehicles that are accounted for by the Constant Speed model are not included in the output of the Predictive Speed model. We chose 10 random tracks from each location, with a minimum length of 45 frames and a maximum length of 60 frames, and plotted them in Figure 8 so that we could have a better understanding of the restrictions imposed by the PS model. Many of the tracks reveal unexpected spikes in speed, which seems to indicate that the corner feature detector may be choosing alternate similar corners in certain frames. This is in contrast to the fact that some tracks show expected smooth transitions that are typical of regular traffic.

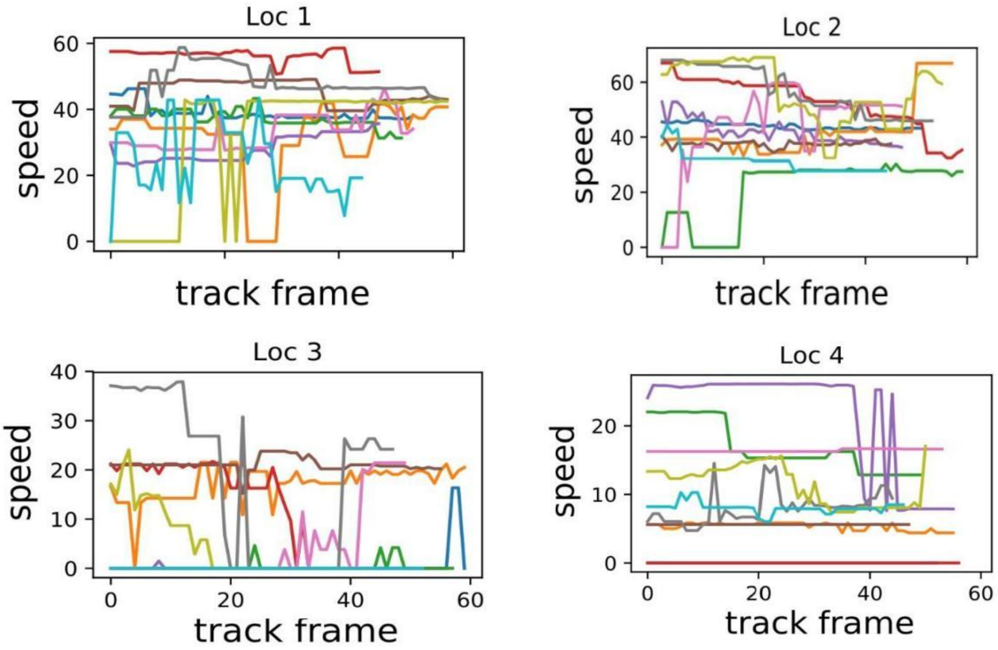

Figure 8. Speed of Random Tracks

We further test the variability in speed predictions in the Predicted Speed (PS) model by visualising the distributions of speed ranges (the difference between the maximum and minimum speed) in tracks of films in all four locations. This helps us to better understand how the model works. Because the majority of cars are only visible for a few seconds when they pass through the frame, it is reasonable to anticipate that their speeds will be quite consistent and will vary only little. Each quadrant of Figure 9 displays, using a line for each movie at a specific point, a uniform random sample from the speed range distribution in the given video. This sample was obtained by drawing a random distribution from the whole video. Our model demonstrates significant variability, with almost 20% of vehicles reporting a range of more than 15 miles per hour, despite the fact that some degree of variation is to be expected as a result of regular traffic conditions. The performance is much more deplorable at Location 3, where the quality of the movies captured was degraded due to continual camera movement brought on by the wind or bridge vibrations.

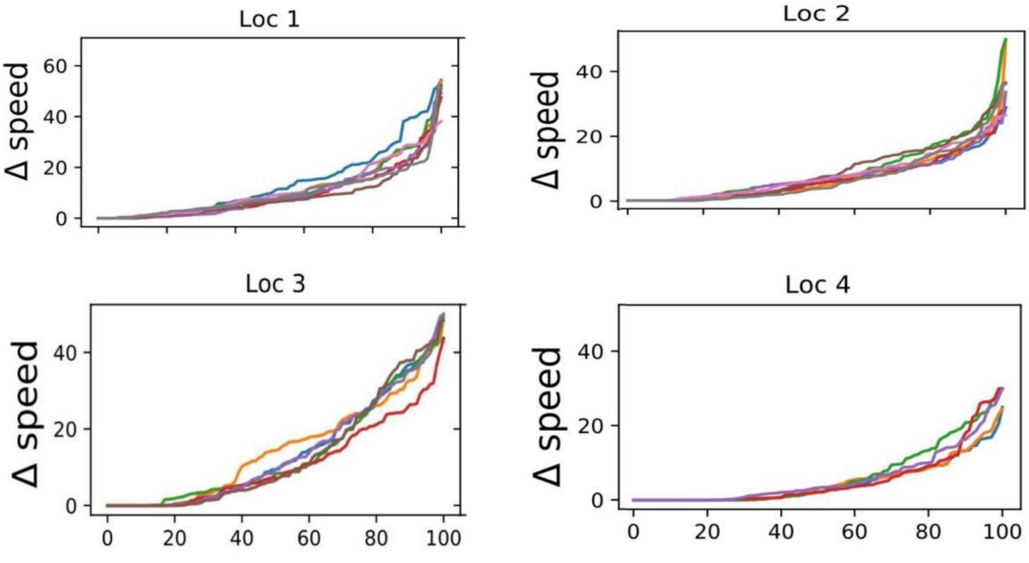

Figure 9. Speed Range Distribution

A total of 27 videos, each exactly one minute long and captured at 30 frames per second and 1080p resolution make up the dataset for the traffic flow analysis. Our mission is to determine the average speed of all vehicles travelling on major thoroughfares throughout all frames of all movies that have been provided to us. The ground-truth speed data were gathered by in-vehicle tracking for a subset of cars in each film that we refer to as ground-truth vehicles. This subset of cars was used in each video. The ability to pinpoint these cars and anticipate their speed will be used to evaluate the candidate's performance.

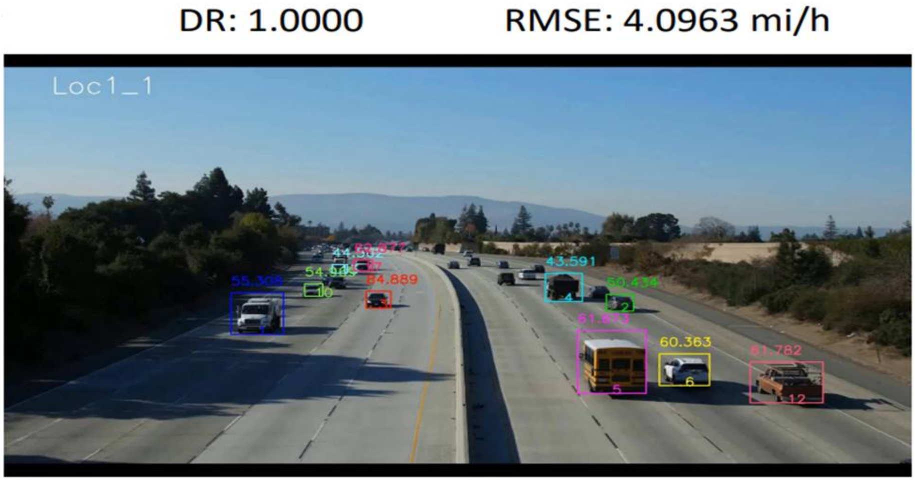

Figure 10. Quantitative Comparison of Speed Estimation

The performance evaluation score for this track is calculated as follows: S1 = DR × (1 − NRMSE), where DR represents the vehicle detection rate for the set of ground-truth vehicles and NRMSE represents the RMSE score across all detections of the ground-truth vehicles, normalised via min-max normalisation with regards to the RMSE scores of Track 1. Where DR represents the detection rate and NRMSE stands for the normalised Root Mean Square Error (RMSE) of speed.

The S1 score, ranging from 0 to 1, gauges overall success, with higher values indicating better performance. It combines the vehicle detection rate (DR), calculated by the ratio of detected ground-truth vehicles, and the normalized RMSE (NRMSE), derived from min-max normalization of all team entries. A vehicle is considered detected if localized in 30% of frames with an IOU score of 0.5 or higher. Our method, using tracklet-based clustering, achieves a 100% DR. Additionally, our camera calibration approach minimizes re-projection error, leading to the lowest RMSE in speed estimation (4.0963 mi/h), enhancing overall accuracy.

5. Conclusion and Future Works

In this paper, we introduced a model for tracking vehicles in traffic videos based on a detect-then-track paradigm, coupled with an optical-flow-based data-driven speed estimation approach, and described our solutions for Track 1 of the 2018 NVIDIA AI City Challenge. Here we present a video semantic segmentation network with low latency. With 54.27 percent MioU and 1.92 frames per second on CPU and 19.57 frames per second on GPU, we were able to build our ResNet50 SegNet model. MIoU dropped to 47.65% after adding FlowNet2S with spatial warping. Compared to the basic model, CPU speed increased by 16.84fps and GPU speed by 30.19fps. Optical flow network integration allowed us to achieve low latency and similar segmentation quality to ResNet50 SegNet. In this study, we emphasize label mapping, which enables us to use a model developed on a dataset by mapping the labels in our dataset to those in the model. This lets us use well-trained models on vast datasets in our work. In balanced mode, adding an optical flow network gains 11.7fps. Our network's speed improvement is less than DVSNet's. As the threshold shifted from sluggish to fast, the network speed rose. Slow mode to balanced mode decreased prediction quality by 6%, and rapid mode by 15%. Calculated the distance between objects in a given frame from a specific reference (x-axis centre). Encoder-decoder networks can do semantic segmentation with low latency. Encoder-decoder networks such as U-Net, and FCN with distinct encoder backbones can be added to this study. Using a different distance measure helps enhance depth inference at road curves.

References

[1] J. Shotton, M. Johnson, and R. Cipolla, “Semantic text on forests for image categorization and segmentation,” *IEEE*, pp.1-8. 2008.

[2] C. Zhang, L. Wang, and R. Yang, “Semantic segmentation of urban scenes using dense depth maps,” *In Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pp. 708-721, 2010.

[3] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431-3440, 2015.

[4] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 2018.

[5] R. Gadde, V. Jampani, and P. V. Gehler, “Semantic Video CNNs Through Representation Warping,” *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 4453-4462, 2017.

[6] R. Faragher, “Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Lecture Notes],” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128-132, 2012.

[7] S. Jain, X. Wang, and J. E. Gonzalez, “Accel: A corrective fusion network for efficient semantic segmentation on video,” *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8866-8875, 2019.

[8] M. Geist, O. Pietquin, G. Fricout, “Tracking in Reinforcement Learning,” In *Neural Information Processing: 16th International Conference, ICONIP 2009, Bangkok, Thailand, December 1-5, 2009, Proceedings, Part I 16*, pp. 502-511, 2009.

[9] W. Liu, F. Yamazaki and T. T. Vu, “Automated Vehicle Extraction and Speed Determination From QuickBird Satellite Images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 1, pp. 75-82, 2011.

[10] V. Nekrasov, H. Chen, C. Shen, and I. Reid, “Architecture Search of Dynamic Cells for Semantic Video Segmentation,” *In Proceedings of the ieee/cvf winter conference on applications of computer vision*, pp. 1970-1979, 2020.

[11] K. Kale, S. Pawar and P. Dhulekar, “Moving object tracking using optical flow and motion vector estimation,” *2015 4th International Conference on Reliability, Infocom Technologies and Optimization*, pp. 1-6, 2015.

[12] Y. Li, J. Shi, and D. Lin, “Low-Latency Video Semantic Segmentation,” *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5997-6005, 2018.

[13] Y. S. Xu, T. J. Fu, H. K. Yang, and C. Y. Lee, “Dynamic Video Segmentation Network,” *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6556-6565, 2018.

[14] M.R. Karim, and A. Dehghani, “Vehicle speed detection in video image sequences using CVS method,” *International journal of the Physical Sciences*, vol. 5, no. 17, pp. 2555-2563, 2010.

[15] K.N. Sunilkumar, G.A. Kumar, R. Gatti, S.S. Kumar, D.A. Bhyratae, and P. Satyasrikanth, “Design and implementation of auto encoder based bio medical signal transmission to optimize power using convolution neural network,” *Neuroscience Informatics*, vol. 3, no. 1, 2023.

[16] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez- Gonzalez, and J. Garcia-Rodriguez, “A survey on deep learning techniques for image and video semantic segmentation,” *Applied Soft Computing Journal*, vol. 70, pp. 41-65, 2018.

[17] S. S. Ramaprasad and K. N. Sunil Kumar, “Intelligent traffic control system using GSM technology,” *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering*, pp. 830-834, 2017.

[18] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, “Fully Convolutional Adaptation Networks for Semantic Segmentation,” *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6810-6818, 2018.

[19] L. Mihaylova, P. Brasnett, N. Canagarajah, and D. Bull, "Object tracking by particle filtering techniques in video sequences," *Advances and challenges in multisensor data and information processing,* vol. 8, pp. 260-268*,* 2007.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234-241, 2015.

[21] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834-848, 2018.

[22] Sunil Kumar K N and Dr. Shivashankar, "Compression of PPG Signal through Joint Technique of Auto-encoder and Feature Selection," *International Journal of Health CareInformation Systems and Informatics-ESCI Indexed ACM-Digital Library*, vol. 16, no. 4, pp.1-15, 2021.

[23] S. K. Shivashankar, "Bio-signals Compression Using Auto Encoder," *Journal of Electrical and Computer Engineering (Q2 Indexed)*, vol. 11, no.1, pp. 424-433, 2021.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2012.

[25] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *In Proceedings of the IEEE international conference on computer vision*, pp. 1520-1528, 2015.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] S. Savian, M. Elahi, and T. Tillo, "Optical Flow Estimation with Deep Learning, a Survey on Recent Advances," *Deep biometrics*, pp. 257-287, 2020.

[28] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patch match for large displacement optical flow," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5704-5712, 2016.

[29] C. Bailer, B. Taetz, and D. Stricker, "Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation," *In Proceedings of the IEEE international conference on computer vision*, pp. 4015-4023, 2015.

[30] S. S. Kumar, "Security Framework for Physiological Signals using Auto Encoder," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 1, pp. 583-592, 2020.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.

[32] M. Menze, C. Heipke, and A. Geiger. "Discrete optimization for optical flow," *In Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings 37*, pp. 16-28, 2015.

[33] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1164-1172, 2015.

[34] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 500-513, 2010.

[35] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, Van Der Smagt, P. Cremers, D. and T. Brox, "Flownet: Learning optical flow with convolutional networks," *In Proceedings of the IEEE international conference on computer vision*, pp. 2758-2766, 2015.

[36] S. Wang, S. Xu, Z. Ma, D. Wang, and W. Li, "A Systematic Solution for Moving-Target Detection and Tracking While Only Using a Monocular Camera," *Sensors*, vol. 23, no. 10, p. 4862, 2023.

[37]V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481-2495, 2017.

[38]A. K. GB, S. K. KN, R. Prasad, R. Gatti, S. S. Kumar and N. Nataraja, “Implementation of Smart Card for Vehicles Documentation Verification Using IoT,” *In 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pp. 965-969, 2021.