

# Video semantic segmentation with low latency

Channappa Gowda D. V., Kanagavalli R.

Department of Information Science and Engineering, The Oxford College of Engineering, Bangalore, India

## Article Info

### Article history:

Received Mar 13, 2023

Revised Mar 14, 2024

Accepted Mar 26, 2024

### Keywords:

Convolutional neural network

Decision network

FlowNet

Latency

Object detection

SegNet

Semantic segmentation

## ABSTRACT

Recent advances in computer vision and deep learning algorithms have yielded intriguing results. It can perform tasks previously requiring human eyes and brains. Semantic video segmentation for autonomous cars is difficult due to the high cost, low latency, and performance requirements of convolutional neural networks (CNNs). Deep learning architectures like SegNet and FlowNet 2.0 on the Cambridge-driving labeled video database (CamVid) dataset enable low-latency pixel-wise semantic segmentation of video features. Because it uses SegNet and FlowNet topologies, it is ideal for practical applications. The decision network chooses an optical flow or segmentation network for an image frame based on the expected confidence score. Combining this decision-making method with adaptive scheduling of the key frame approach can speed up the process. ResNet50 SegNet has a “54.27%” mean intersection over union (MIoU) and a “19.57” average FPS. In addition to decision network and adaptive key frame sequencing, FlowNet2.0 increased graphics processing unit (GPU) frame processing per second to “30.19” with a MIoU of “47.65%”. The GPU is used “47.65%” of the time. This performance gain illustrates that the video semantic segmentation network is faster without sacrificing quality.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Channappa Gowda D. V.

Department of Information Science and Engineering, The Oxford College of Engineering

Bangalore 560068, India

Email: [sudee.info@gmail.com](mailto:sudee.info@gmail.com)

## 1. INTRODUCTION

Semantic segmentation is a popular topic in computer vision research. When each pixel in an image or video is given its own category, the process is known as semantic segmentation [1]. The deep learning model should be able to recognise, classify, and label every pixel in an image or video frame when it is given the input of an image. The four ideas of picture classification, object identification, semantic segmentation, and instance segmentation are contrasted in Figures 1(a)-(d). Prior research on semantic segmentation used dense depth maps [2] and 2D information such as colour and shape, which required a lot of labor-intensive human engineering. The difficulties of semantic segmentation and classification have been made easier to solve without the need for human engineering with the emergence of deep learning approaches and machine learning developments. Convolutional neural network (CNN) has shown superior results [3], [4] in assessing and labelling the characteristics of pictures with enhanced performance. CNN is used for semantic segmentation of images. As a result of CNN’s accomplishments in image semantic segmentation and other image processing tasks, intriguing new advancements in video semantic segmentation are beginning to emerge. Numerous research [5]–[7] have focused on video segmentation with the goal of enhancing its performance. Few studies have been conducted to minimise latency [8]–[11].

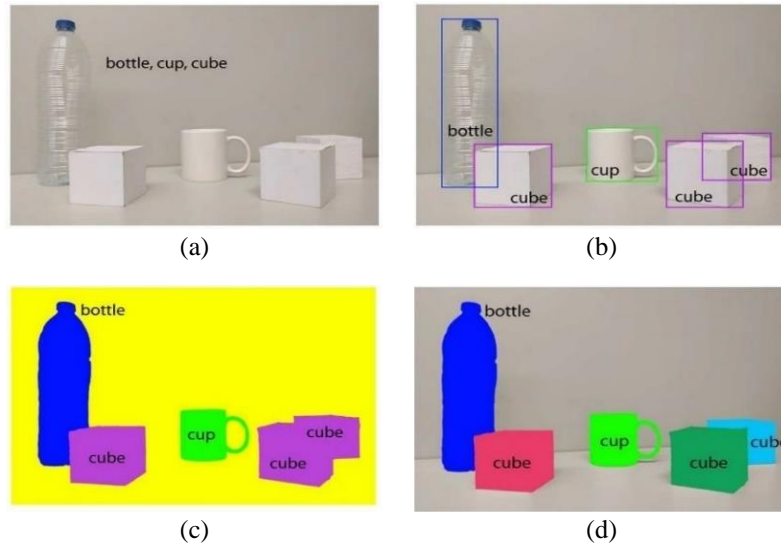


Figure 1. Semantic segmentation of an image source: (a) image classification, (b) object localization, (c) semantic segmentation, and (d) instance segmentation

Numerous significant initiatives, including fully convolutional network (FCN), pyramid scene parsing network (PSPNet), U shaped network (U-Net), segmentation network (SegNet), and DeepLab, have conducted semantic segmentation research [12]–[16]. The aforementioned networks are widely recognised for processing a single frame slowly, but they are quite good at foretelling the future. The goal of this study is to develop a framework for video semantic segmentation that can be applied to real-time tasks like in-building navigation and self-driving automobiles. For real-time applications, it is essential to achieve low latency employing methods from prior publications on dynamic video segmentation network (DVSNet) and deep feature flow (DFF), which include two networks. The optical flow and per-frame segmentation networks are present in both DVSNet and DFF.

One of the most crucial components of this study are the decision network, a system for adaptively updating keyframes, and other techniques like label mapping and depth inferencing. Want to compare the outcomes when an encoder-decoder architecture, such as ResNet50 SegNet, is used as a segmentation network in place of the deep lab while keeping the same flow network. The studies employed the driving scenario dataset Cambridge-driving Labeled Video Database (CamVid) [17], which is quite comparable to the CityScape [18] used in the prior studies.

A summary of the research on the various deep learning methods applied in this study is provided in section 2. The network topologies are explained in depth in section 3, and the training methods and operation of the video segmentation network are covered in section 4. Section 5 details the experiments that were carried out, their findings, a quantitative and qualitative interpretation of those data, and their conclusions. The final portion covers the conclusion, in which we go through the advantages and disadvantages of applying this segmentation technique as well as suggestions for future research.

## 2. RELATED WORK

Prior to the development of FCN [19], [20], only convolutional networks with a fully connected layer at the output were utilised in semantic segmentation networks [21]–[23]. The classification layer's last fully connected layer was given top priority by FCN to be changed to a convolutional layer. Using this classification strategy, which involves removing a fully linked layer, the number of parameters is decreased, input picture size restrictions are abolished, and spatial data is retained. FCN has an impact on both the expanding and shrinking components of Krizhevsky *et al.* [24] U-Net idea for medical pictures. It uses downsampling and upsampling, similar to DeConvNet, to maintain track of data that could otherwise be lost during down sampling. The contracting portion is largely in charge of constructing the feature maps and using  $3 \times 3$  convolution to extract features during downsampling. However, the expansion step uses deconvolution, which produces fewer feature maps but larger spatial dimensions (height and width). U-Net replicates the feature maps that have been cropped from the contracting party to the expanding portion, allowing it to retain track of pattern information. SegNet uses the VGG-16 network (encoder including the top 13 convolution layers of

VGG-16) as an encoder [25]. For its prowess in handling categorization problems, this network is highly known. Alternative to replace, which is also FCN, eliminates the totally connected layer that is the topmost layer in the network to keep the high-resolution feature maps at the encoder output [26]. The final decoder's output is given to the softmax classifier, which creates class probabilities using it. One of the most essential parts of video analysis is optical flow. The difference between the layers, which have the same dimensions as the input frame, is the offset of each pixel along two different axes, namely the x-axis and y-axis [27]. One of the first techniques for figuring out optical flow is the variational approach. This approach [28] was the norm for a sizable period of time. Due to the occurrence of local minima, the variational technique focused mostly on relatively small movements and gave priority to initializations that involved no motion field [29].

The Brox and Malik technique was used to create various algorithms, including PatchMatch [30], [31], FlowField [32], DiscreteFlow [33], and FullFlow [34]. Conditional random field (CRF) is used by both DiscreteFlow and FullFlow to process vector similarities. A significant technique created by Weinzaepfel and colleagues, DeepFlow [35], was influenced by Brox and Malik. A loss function is also included in the variational optical model called DeepFlow. It's built on a deep matching method that enables the fusion of feature descriptors and matching. One or two of the most popular unified or combined matchings techniques are EpicFlow and RichFlow [36], [37]. One of the simplest methods for achieving video segmentation is to use image segmentation networks [38], [39]. However, they work on a "per frame" basis, which complicates the calculation because movies have more frames than static images do. Additionally, image semantic segmentation does not account for the temporal relationships between various frames [40]. Semantic segmentation of video via representation warping was proven by Hu *et al.* [41]. The authors' method involves converting CNN models for image segmentation to video models. In order to warp intermediate CNN representations and combine them with the current frame, they first introduce a warping module called NetWarp. The computation time is effectively cut in half using DFF approaches. Fixed keyframe scheduling was also suggested coupled with the usage of two different networks, a segmentation network and an optical flow network [42], [43]. Furthermore, it lacks both flexibility and customisation since it employs fixed keyframe scheduling (implicating a constant update time between succeeding frames) [5], [44]. Zhang *et al.* [2] unsupervised's learning approach for semantic segmentation of videos presents an entirely traditional adaptive network for semantic segmentation. However, in order to limit the degree of error in their predictions of the target data, they propose to employ labelled instances from the source domain together with a significant number of unlabeled examples from the target domain [45].

### 3. IMPLEMENTATION METHODOLOGY

The DVSNNet, which separates video frames using both per-frame and optical flow networks, is the inspiration for this study. For a video segmentation network, achieving minimal latency is the major objective. We may take use of the advantages of both of these incredibly potent networks by utilising them both. The segmentation network can forecast the outcome of its actions quite accurately. However, optical flow networks are widely renowned for their ability to forecast motion data very fast. This aids in accelerating and improving the performance of segmentation, along with spatial warping (in which the optical flow output is warped with the segmentation output). The segmentation network is slow since it processes data on a per-frame basis, and the optical flow network's flows can occasionally be inaccurate. None of these networks, however, are without their limitations. In this part, network structures and study-related tactics are thoroughly described. Three low-latency networks are used in this study for video semantic segmentation. However, we use SegNet with ResNet50 encoder as a segmentation network, which is distinct from SegNet with FlowNet2S as an optical flow network with spatial warping function. To reach a conclusion based on the difference between two successive frames, judgement networks (DN) are included.

Figure 2 shows the interaction of our semantic segmentation network to the decision network. The initial iteration of our network sends the current frame to the segmentation network path to produce segmentation output. Next, it is established how key and current frames differ from one another. Based on their difference to segments, the decision network calculates each frame's anticipated confidence score. A 0-100 inference threshold is denoted by "t". if a decision network's predicted confidence score exceeds the threshold, the current frame is sent to the segmentation network. Because there are so many pixels separating the keyframe from the current frame, FlowNet2S is unable to make accurate predictions. As a result, changes were made to the key frame and segregated network pathways. The optical flow network, FlowNet2S, is predicted to make forecasts similar to those made by the segmentation network if the expected confidence score is lower than the cutoff.

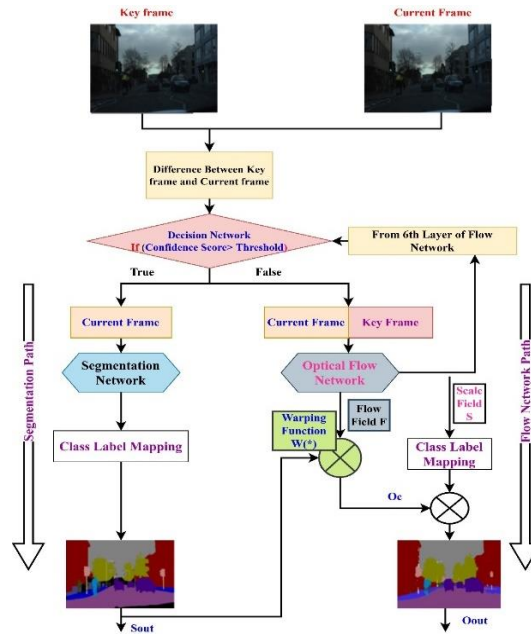


Figure 2. Figure illustrating video segmentation network

Routed network flow only current frames are permitted with segmentation. Flow network path is entered by the keyframe and the current frame. When the decision network chooses to partition the output, ResNet50 SegNet does so. When the decision network picks a flow network path, frames supplied through FlowNet2S are distorted by utilising the most recent segmentation result. To further categorise the 11 classes, label-mapped flow network scale attributes are used. Segmented flow networks are produced as a result of these label-mapped scale fields and distorted output. The optical flow network cannot be segmented by itself. On a flow network route similar to a segmentation network, spatial warping aids in segmentation. ResNet50 Although the per-frame processing of SegNet is slow, the segmentation it creates is precise. A spatially warped optical flow network is rapid but not precise. As a result, segmentation is accelerated and produces efficient segmentation when switching between the two networks, utilising a flow network path the majority of the time, and using a segmentation network only when the frame difference is bigger. A keyframe is changed whenever a significant frame difference is found between two frames. Processing time is decreased through adaptive keyframe scheduling.

### 3.1. Training

This study includes a choices network, a segmented network, an optical flow network with warping, and all three. Networks for segmentation and decision-making will be trained. We'll train the FlowNet2S for the optical flow network using the cityscape dataset. SegNet has been trained using ResNet50 and then fine-tuned using CamVid to serve as encoders and decoders for the imagenet database. Epoch-adaptive handling of the learning rate is provided by the Ada-delta optimizer. We utilised 367 pictures from CamVid, each at a 10-step period, to train ResNet50 SegNet. ResNet50 SegNet uses weights that were trained on ImageNet. Model weights are reused in transfer learning on a related dataset. The output of the segmentation network is label-mapped to eleven classes. During label mapping, we remove the "unlabelled" or "void" class from the CamVid dataset. After integrating the segmentation network (FlowNet2S with ResNet50 SegNet), this approach retains class labels. Then, FlowNet2S was connected to DVSNet in order to leverage its checkpoints, and scale variables were used to map the relevant labels. The CamVid dataset is small enough to train FlowNet on data samples of road scene, therefore using checkpoints from the baseline FlowNet actually wouldn't improve flows when gliding chairs and 3D-object data are separate from road scenes/autonomous driving scenarios. As a result, we employ the FlowNet2S of the DVSNet design, which was created using 1.2 million datasets from a cityscape sample. Finally, we use sixth-layer flow characteristics to train the decision network. The decision network is trained as a regression model using flow network characteristics. The decision network should be able to predict the anticipated confidence score for a certain frame thanks to continuous learning. The 500 training epochs are completed in 32-epoch batches with a decay rate of 0.99. The decision network gets better and better at predicting a frame's expected confidence score through continuous learning and comparison to the ground truth confidence score. As a result, switching network pathways for segmentation and flow is effective. Only decision network training makes use of the ground truth confidence score.

### 3.2. Experiments

#### 3.2.1. Evaluation of baseline network

Resnet50 SegNet was trained at various epochs using the cross-entropy loss function and the Ada-delta optimizer. Accuracy and loss at various epoch rates are measured once the model has stabilised. The mean intersection over union (MIoU) values at various epoch rates were also examined. On 233 current frame pictures, MIoU, and classwise MIoU for epochs 10 to 50 and 100 were compared. Overall, the environment for autonomous driving was divided into more specialised divisions. In result 1, classwise MIoU scores at various epochs are compared to and discussed with ResNet50 SegNet and SegNet's segmentation predictions.

#### 3.2.2. Evaluation of video semantic segmentation network

In this experiment, we combine ResNet50 SegNet and FlowNet2S to decrease latency in the video segmentation network while keeping quality. The decision network forecasts the confidence score after combining the two networks. A video segmentation network known as the integrated network is seen on a test set of 6959 keyframes and current frames. To assess output quality and network speed, the segmentation predictions of the video segmentation network (ResNet50 SegNet+FlowNet2S) and each frame (ResNet50 SegNet) are compared. A thorough analysis of the findings is provided in result 2.

#### 3.2.3. Validation of a video segmentation model at various "t" thresholds

To evaluate the performance of the decision network by changing the threshold from 0-100. The segmentation network path makes an output prediction when t is close to 100. When "t" is near to zero, it is more probable that the flow network path for the predicted output will be selected. We test the system using various threshold values in light of this predicted behaviour.

#### 3.2.4. Depth inferences between objects

The distance of an item as seen by the driver was shown. Calculate the distance in 2D between the driver and the chosen target object's centre. The target object's centre in this approach is the diagonal midpoint of the target object, and the driver's reference point is the x-center of the picture frame. We only measure distances to bicycles, vehicles, people, and poles. The (1) is used to determine the euclidean distance between two places.

$$D_{euclidean} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

The pixel distance is then transformed to metres by applying the (1) pixel in metres=0.0002645833. In self-driving cars, depth information helps adjust the vehicle's speed based on how far or close the item is. Helps avert accidents.

## 4. RESULTS AND DISCUSSION

### 4.1. Results of the baseline model

The input image is on the left, the anticipated output is on the right, and the ground truth is in the middle of Figures 3 and 4, respectively, showing the predicted output of ResNet50 SegNet and original SegNet. According to ground truth photographs, each class is assigned a distinct colour in the expected output, such as a purple automobile or a red structure. The anticipated output of Figure 4 was pretty similar to its ground truth label and the permitted SegNet predictions of Figure 3 (original, ground truth, and predicted). Segmentation maps that were deformed were rare in classes.

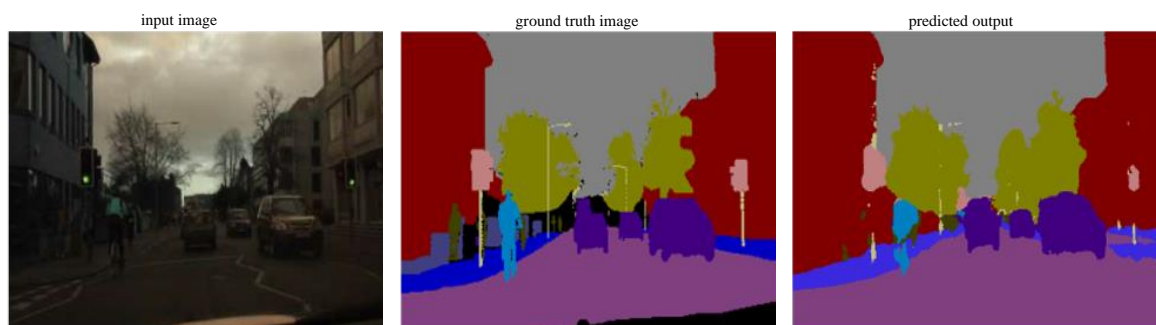


Figure 3. Authorized SegNet implementation

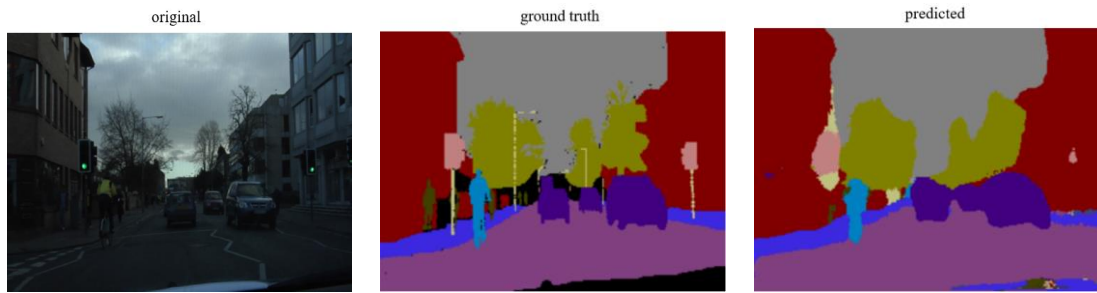


Figure 4. Customised ResNet50 SegNet

Each of the 11 classes IoUs are displayed in Table 1. With the exception of “pavement” where MIoU decreased by 2%, 50 epoch classwise IoU on the test set yielded excellent findings and appears to be an improvement over earlier epochs. The class-wise IoU of the building, bicycle, fence, pedestrian, and tree decreased by more than 1% on testing data when the epoch was raised to 100.

The model’s overfitting, which lacks adaptability and performs poorly on fresh data, may be to blame for this decline in accuracy for more categories as epochs grew. In order to extend the model, errors like this should be avoided as they are risky in automated driving. To maintain class generality, epoch 50 was selected after analysis. ResNet50 SegNet achieved a MIoU of 54.27%, 1.92 mean frames per second, and 19.57 frames per second on central processing unit (CPU) and graphics processing unit (GPU) on the CamVid test set (Table 2). ResNet50, a thin and moderately deep encoder network, outperformed DVSNet’s deeplab-fast network in terms of performance on GPU. We believe that deeper networks, such as ResNet100 and ResNet152, can improve prediction accuracy. Results from the first SegNet and ResNet50 SegNet are compared in Table 2.

Table 1. Each class label’s IoU classwise

Classes	ResNet50 SegNet					
	10 epochs	20 epochs	30 epochs	40 epochs	50 epochs	100 epochs
Sky	89.5	86.34	86.60	86.95	86.86	86.81
Building	73.94	74.64	75.04	74.84	75.088	73.55
Pole	3.7	6.44	10.7	10.88	12.49	14.73
Road	88.01	88.7	89.88	89.86	89.28	90.64
Pavement	64.99	66.89	69.68	69.25	67.67	71.16
Tree	69.22	70.22	70.48	70.82	71.09	69.19
Sign symbol	28.77	31.67	31.62	31.70	34.29	32.02
Fence	20.54	25.85	25.23	25.21	27.74	25.98
Car	70.05	72.13	72.24	73.49	73.44	73.89
Pedestrian	25.56	21.19	31.91	32.71	34.11	32.48
Bicyclist	13.1	22.38	27.5	26.19	32.47	37.16

Table 2. Comparison of SegNet [15] MIoU scores with our developed model ResNet50 SegNet (our baseline) and ResNet50 SegNet+FlowNet2S

Network	MIoU	Average FPS on CPU	Average FPS on GPU
SegNet (original)	60.07	-	-
ResNet50 SegNet (ourbaseline)	54.27	1.92	19.57
ResNet50 SegNet+FlowNet2S	47.65	16.84	31.27

#### 4.1.1. Results of video semantic segmentation network

Results Results from experiment 2 are detailed in this section. The estimated segmentation output quality of the baseline model decreased and seemed distorted when the optical flow network was included. The outcome of ResNet50 SegNet was comparable to the flow network path chosen by the model, however numerous class estimates were off. Additionally, the network only accurately predicted some types, such as poles and fences as shown in Figure 5. A sign symbol was used in inaccurate forecasts. Pole may serve as a powerful symbol. Partial segmentation and distortions may result from information loss during flow construction.



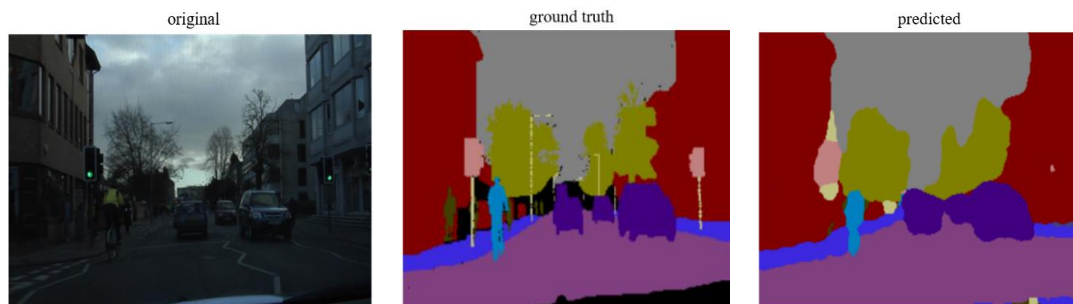


Figure 5. ResNet50 SegNet+FlowNet2S

While in balancing mode, the integrated network scored 47% MioU, CPU, and GPU frames per second (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.

FlowNet2S, a quick optical flow network, helps our video segmentation network function more quickly than per-frame. Instead of segmentation, the optical flow network just generates flow information. The output quality is maintained while segmentation is produced fast thanks to the spatial warping of optical flow network flows with newly segmented output from the segmentation network path. The optical flow network makes use of the slower segmentation network's speed. Compared to DVSNet, our two-network system gained more. Details may be found in Table 3. In contrast to DVSNet's 14.2 FPS, our network increased to 11.7 FPS. Compared to DVSNet, our network boosted FPS by 31.27, although the gain was smaller.

Table 3. Gain in FPS using two network approaches

Inference on GPU	Segmentation network	(Segmentation network+FlowNet2S)	FPS gain
Proposed work	19.57 (ResNet50 SegNet)	31.27	11.7
DVSNet	5.6 (DeepLab_fast)	19.8	14.2

#### 4.1.2. Results collected at various threshold modes

The MIoU and network speed at various CPU and GPU levels are shown in Table 4. The segmentation approach was often used if the threshold was fixed to 90, which corresponds to slow mode, and the network obtained a MIoU of 53.69% and a framerate of 7.87 and 26.69 on CPU and GPU, respectively. The segmentation and flow network pathways were randomly chosen once the threshold was set to 80. A MIoU of 47.65%, 16.84 frames per second, and 31.27 on CPU and GPU were achieved by the network. The majority of the time, the flow network path was chosen if the threshold was fixed at 65. With a 32.55% MIoU, this improved network performance to 19.09 frames per second on CPU and 32.94 frames per second on GPU. IoU is shown in Table 5 per network speed mode.

Table 4. The segmentation approach works at varying speed modes

Speed mode	Threshold "t"	CPU		GPU	
		Speed	MIoU	Speed	MIoU
Very slow mode	95	1.92	54.27	19.57	54.27
Slow mode	90	7.87	53.69	26.69	53.69
Balanced mode	80	16.84	47.65	31.27	47.65
Fast mode	65	19.19	32.55	32.94	32.54
Very fast mode	0	25.45	18.00	35.88	18.00

In fast mode, when FPS climbed, MIoU decreased. When speed was reduced in slow mode, MIoU rose. Thus, speed and output quality are compromised. The balanced mode combines speed and quality. Real-time, users can adjust the ideal threshold based on interface suggestions. Providing speed and MIoU accuracy flexibility. After a certain point, speed saturates and doesn't grow. Table 4 shows that with a threshold of 0, the maximum GPU speed is 35.88 FPS, just 3 FPS greater than fast mode.

Table 5. IoU by classwise in % for different speed modes

Classes	Class wise IoU on slow mode	Class wise IoU on balance mode	Class wise IoU on fast mode
	"t" <sub>0</sub> =90	"t" <sub>0</sub> =80	"t" <sub>0</sub> =65
Sky	86.28	83.54	67.63
Building	75.30	72.49	53.16
Pole	6.44	8.51	3.68
Road	88.7	85.61	76.25
Pavement	66.16	62.24	39.12
Tree	70.38	65.59	42.44
Sign symbol	30.95	25.04	12.79
Fence	29.30	25.63	12.25
Car	70.82	54.43	38.79
Pedestrian	28.89	18.59	5.35
Bicyclist	33.53	22.45	6.6

#### 4.1.3. Results of depth inference

Figure 6 depicts the distances between bicycles and cars on bounding boxes. The cycle rider is 6.68 metres away from the reference location. The distance between two target cars is 8 m and 7.65 m. The 7.65 m car is getting closer, while the 8.00 m car is going away. When moving on a curving route, distance calculations are also erroneous. This may be because the euclidean distance is estimated as a straight line.

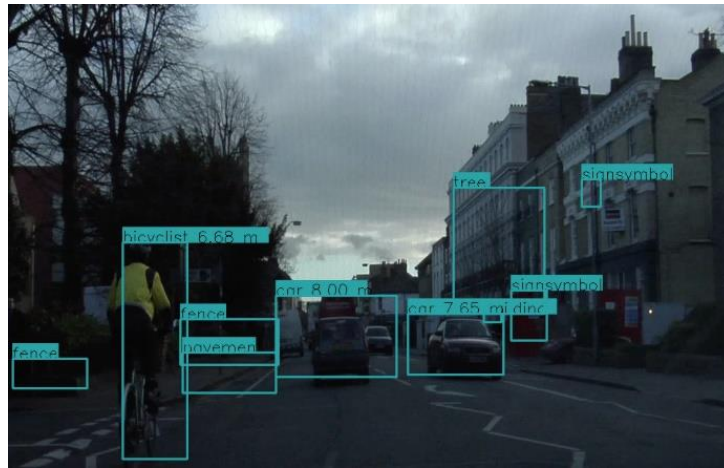


Figure 6. Illustrating distance between objects in streets

## 5. CONCLUSION

We offer a video semantic segmentation network with minimal latency. We were able to construct our ResNet50 SegNet model with 54.27% MioU, 1.92 frames per second on CPU, and 19.57 frames per second on GPU. After introducing FlowNet2S with spatial warping, MioU fell to 47.65%. CPU speed improved by 16.84 FPS and GPU speed by 30.19 FPS in comparison to the base model. We were able to achieve low latency and segmentation quality comparable to ResNet50 SegNet by integrating an optical flow network. In this work, we place special emphasis on label mapping, which allows us to employ a model built on a dataset by converting the labels in our dataset into those in the model. This enables us to deal with well-trained models on large datasets. An optical flow network increases the frame rate in balanced mode by 11.7 FPS. The speed improvement of our network is lower than that of DVSNet. The network speed increased as the threshold went from slow to fast. Rapid mode had a 15% reduction in prediction quality, whereas slow mode had a 6% reduction. calculated the separation of items in a frame using a certain reference (x-axis centre). By figuring out how close an item is to the car, we can avoid collisions. Semantic segmentation may be performed via encoder-decoder networks with little delay. This study may be expanded to include encoder-decoder networks with unique encoder backbones like U-Net and FCN. The depth inference at road bends is improved by using a new distance metric.

## ACKNOWLEDGEMENTS

Author thanks to our research institute and the management to support this research work.

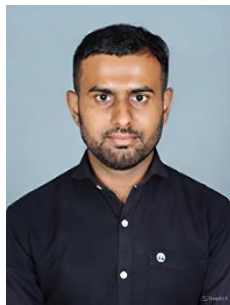





## REFERENCES

- [1] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *26th IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587503.
- [2] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," *Lecture Notes in Computer Science*, vol. 6314, pp. 708–721, 2010, doi: 10.1007/978-3-642-15561-1\_51.
- [3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2015, pp. 431–440, doi: 10.1109/CVPR.2015.7298965.
- [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," *3rd International Conference on Learning Representations, 2015*.
- [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*, Oct. 2017, pp. 4463–4472, doi: 10.1109/ICCV.2017.477.
- [6] S. Jain, X. Wang, and J. E. Gonzalez, "Accel: a corrective fusion network for efficient semantic segmentation on video," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp. 8858–8867, doi: 10.1109/CVPR.2019.00907.
- [7] V. Nekrasov, H. Chen, C. Shen, and I. Reid, "Architecture search of dynamic cells for semantic video segmentation," in *Proceedings-2020 IEEE Winter Conference on Applications of Computer Vision*, Mar. 2020, pp. 1959–1968, doi: 10.1109/WACV45572.2020.9093531.
- [8] Y. Li, J. Shi, and D. Lin, "Low-latency video semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 5997–6005, doi: 10.1109/CVPR.2018.00628.
- [9] Y. S. Xu, T. J. Fu, H. K. Yang, and C. Y. Lee, "Dynamic Video Segmentation Network," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 6556–6565, doi: 10.1109/CVPR.2018.00686.
- [10] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell, "Clockwork convnets for video semantic segmentation," *Lecture Notes in Computer Science*, vol. 9915, pp. 852–868, 2016, doi: 10.1007/978-3-319-49409-8\_69.
- [11] Y. Li, J. Shi and D. Lin, "Low-Latency Video Semantic Segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5997-6005, doi: 10.1109/CVPR.2018.00628.
- [12] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proceedings-30th IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 4141–4150, doi: 10.1109/CVPR.2017.441.
- [13] 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, Sep. 2018, doi: 10.1016/j.asoc.2018.05.018.
- [14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings-30th IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.
- [15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention*, pp. 234–241, Nov. 2015, doi: 10.1007/978-3-319-24574-4\_28.
- [16] 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, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.
- [17] 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, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.
- [18] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: a high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, Jan. 2009, doi: 10.1016/j.patrec.2008.04.005.
- [19] M. Cordts et al., "The cityscapes dataset," *CVPR Workshop on The Future of Datasets in Vision*, vol. 2, p. 1, Jun. 2015.
- [20] S. K. N. Kumar, S. Shankar, and Keshavamurthy, "Compression of PPG signal through joint technique of auto-encoder and feature selection," *International Journal of Healthcare Information Systems and Informatics*, vol. 16, no. 4, 2021, doi: 10.4018/IJHISI.20211001.0a23.
- [21] K. N. Sumilkumar, Shivashankar, and Keshavamurthy, "Bio-signals compression using auto encoder," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 424–433, 2021, doi: 10.11591/ijece.v11i1.pp424-433.
- [22] K. N. Sumilkumar and Shivashankar, "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, doi: 10.5373/JARDCS/V12SP1/20201107.
- [23] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: a survey," *Applied Sciences (Switzerland)*, vol. 12, no. 18, p. 8972, Sep. 2022, doi: 10.3390/app12188972.
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [25] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano, "Toward automatic phenotyping of developing embryos from videos," *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1360–1371, 2005, doi: 10.1109/TIP.2005.852470.
- [26] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," *Advances in Neural Information Processing Systems*, vol. 4, pp. 2843–2851, 2012.
- [27] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *IEEE International Conference on Computer Vision*, Dec. 2015, pp. 1520–1528, doi: 10.1109/ICCV.2015.178.
- [28] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013, doi: 10.1109/TPAMI.2012.231.
- [29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations-Conference Track Proceedings*, pp. 1–14, Apr. 2015, doi: 10.48550/arXiv.1409.1556.
- [30] E. J. Kirkland, "Bilinear interpolation," *Advanced Computing in Electron Microscopy*, pp. 261–263, 2010, doi: 10.1007/978-1-4419-6533-2\_12.
- [31] T. Liu, X. Huang, and J. Ma, "Conditional random fields for image labeling," *Mathematical Problems in Engineering*, vol. 2016, 2016, doi: 10.1155/2016/3846125.
- [32] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," *Advances in Neural Information Processing Systems*, 2005.
- [33] S. Savian, M. Elahi, and T. Tillo, "Optical flow estimation with deep learning, a survey on recent advances," *Deep Biometrics*, pp. 257–287, 2020, doi: 10.1007/978-3-030-32583-1\_12.
- [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, Mar. 2011, doi: 10.1109/TPAMI.2010.143.
- [35] Y. Hu, R. Song, and Y. Li, “Efficient coarse-to-fine patch match for large displacement optical flow,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 5704–5712, doi: 10.1109/CVPR.2016.615.
- [36] C. Bailer, B. Taetz, and D. Stricker, “Flow fields: dense correspondence fields for highly accurate large displacement optical flow estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1879–1892, Aug. 2019, doi: 10.1109/TPAMI.2018.2859970.
- [37] M. Menze, C. Heipke, and A. Geiger, “Discrete optimization for optical flow,” *Lecture Notes in Computer Science*, vol. 9358, pp. 16–28, 2015, doi: 10.1007/978-3-319-24947-6\_2.
- [38] Q. Chen and V. Koltun, “Full flow: optical flow estimation by global optimization over regular grids,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 4706–4714, doi: 10.1109/CVPR.2016.509.
- [39] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, “DeepFlow: large displacement optical flow with deep matching,” in *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2013, pp. 1385–1392, doi: 10.1109/ICCV.2013.175.
- [40] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “EpicFlow: edge-preserving interpolation of correspondences for optical flow,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 1164–1172, 2015, doi: 10.1109/CVPR.2015.7298720.
- [41] Y. Hu, Y. Li, and R. Song, “Robust interpolation of correspondences for large displacement optical flow,” in *Proceedings-30th IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 4791–4799, doi: 10.1109/CVPR.2017.509.
- [42] A. Dosovitskiy et al., “FlowNet: learning optical flow with convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2015, pp. 2758–2766, doi: 10.1109/ICCV.2015.316.
- [43] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: evolution of optical flow estimation with deep networks,” in *Proceedings-30th IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 1647–1655, doi: 10.1109/CVPR.2017.179.
- [44] A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network,” in *Proceedings-30th IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 2720–2729, doi: 10.1109/CVPR.2017.291.
- [45] A. Marcu, D. Costea, V. Licaret, and M. Leordeanu, “Towards automatic annotation for semantic segmentation in drone videos,” *arXiv*, Oct. 2019, doi: 10.48550/arXiv.1910.10026.

## BIOGRAPHIES OF AUTHORS



**Channappa Gowda D. V.**    holds M.Tech degree and currently pursuing the part time research at VTU. He has 10 years of teaching experience, currently working in The Oxford College of Engineering, VTU as an assistant professor. His research interest are video processing, operating systems, and analysis of algorithms. He can be contacted at email: sudee.info@gmail.com.



**Kanagavalli R.**    hold the Ph.D. degree in computer science and Engineering. She has 18 years of teaching experience, currently working in The Oxford College of Engineering, VTU. Her research interest are video processing, cloud computing, and image processing. She can be contacted at email: kanaga.ksr@gmail.com.