ENEL 645 FINAL PROJECT
TRAINING A SEMANTIC SEGMENTATION MODEL FOR AUTONOMOUS DRIVING

Group 5: Khoi Nguyen, Justin Nguyen, Feras Dahrooge, Seho Chung, Ardit Baboci

University of Calgary, Winter 2023, Dr. Roberto Souza

ABSTRACT

This report is part of the submission for the final project of ENEL 645, with a focus on semantic segmentation in the context of autonomous vehicles. A neural network following an encoder-decoder architecture was trained from scratch on the Cityscapes dataset using PyTorch. The best model performed with a mean IoU of 43.1% on the test set. The neural network was used to semantically segment a video, mimicking its real-world application to autonomous driving.

GitHub repository:
https://github.com/justinknguyen/enel645-winter-2023-project-group5.git

Index Terms**—** semantic segmentation, autonomous vehicles, deep learning, PyTorch, ENEL 645

1. INTRODUCTION

Semantic segmentation is the task of assigning a class label to each pixel in an image. In essence, it is a classification task done on a pixel-by-pixel basis [1]. Some practical applications for this technology include [2]:

- autonomous vehicles
- medical imaging
- augmented reality

The scientific challenge that this report will focus on is the use of deep learning models in the field of autonomous vehicles. Autonomous vehicles have a variety of onboard sensors, most notably cameras, that are used to plan the vehicle's behaviour and avoid obstacles. Using image segmentation models, the images from the vehicle's cameras can be used to map the surrounding environment [3].

Although most autonomous vehicles available on the market today meet the SAE's second level of autonomous driving (i.e., hands off), fully automated vehicles (i.e., mind off) are expected to be commercially available by 2025, rapidly penetrating the new vehicle market after that [4]. Due to scrutiny resulting from the numerous accidents that autonomous vehicles have already been involved in, certain jurisdictions have introduced legislation involving stricter guidelines before approving vehicles for one of the higher levels of autonomous driving [5]. Semantic segmentation and an accurate understanding of the surrounding environment is expected to play an important role in the future of autonomous driving.

For this project, a deep learning model was trained with the purpose of performing semantic segmentation on images in urban settings similar to what an autonomous vehicle may encounter. The model is then used to semantically segment a video, mimicking the process that might take place in the fully autonomous vehicles of the future.

2. RELATED WORK

2.1 Course Work

2.1.1 Assignment 2

The second ENEL 645 assignment served as the basis for this final project. For the first part of the assignment, the team used a pre-trained PyTorch model to semantically segment some example images.

For the second part of the assignment, the team trained a semantic segmentation model using the segmentation_models_pytorch library. This library simplifies the training process by providing an easy-to-use framework with pre-defined architectures.

The "future work" coming out of this assignment was to train a semantic segmentation model from scratch, and to apply the technique to a video in order to mimic its application to autonomous driving. These are the tasks tackled in this report.

2.1.2 Garbage Classification Model

The garbage classification model that was set up throughout the course provided a template for training a deep learning model from scratch using PyTorch. This served as the starting point for training the semantic segmentation model used in this project.

The Datasets, Dataloaders, and neural network were modified as required, but the overall framework was kept consistent. It was also through the training of the garbage classification model that the use of the University of Calgary Teaching and Learning Cluster (TALC) was introduced to the team.

2.2 Literature

Modern state-of-the-art models for pixel-wise semantic segmentation are fully-convolutional networks [6]. Classic semantic segmentation architectures such as FCN, U-net, and DeepLab all follow an encoder-decoder pattern [7].

The encoder compresses the information into lower-dimensional representations while extracting the features, and the decoder projects this information back to the original image resolution. The encoder is usually based on an existing network such as VGG, Resnet, or Xception, while the decoder can make use of bilinear interpolation, deconvolution, or unpooling [6] [7].

Video semantic segmentation is still in its infancy and adds a few layers of complexity over traditional image semantic segmentation. The most important of these is considering temporal information. That is, using the previous frame(s) and the previous output(s) of the network to improve the accuracy of the prediction on the current frame [8]. In the context of autonomous vehicles, there are further requirements in that the predictions need to be made in real-time and often with fewer computational resources due to being ran on embedded devices [9]. Not only is the process of video semantic segmentation a more difficult technical task to tackle, but its use in real-world applications also requires that it be done quicker and with fewer computational resources.

The process of annotating a semantic segmentation dataset is very laborious and time consuming since it requires an individual and independent annotation for each pixel in the image. One ongoing area of research revolves around weakly-supervised semantic segmentation in order to decrease the economic and time costs of manual annotations. Fortunately, many large, well-annotated, datasets have been made available to the public for research and learning purposes. These datasets cover a large variety of situations and environments, including scenery, aerial images, city roads, and CT scans [2].

3. MATERIALS AND METHODS

3.1 Cityscapes Dataset

Cityscapes is one of the most popular benchmarking datasets for comparing model performance on the segmentation of urban scenes, which is consistent with what an autonomous vehicle would normally encounter [6]. The dataset includes 5,000 images with high quality pixel-level annotations, and an additional 20,000 images with coarse annotations covering 50 different European cities. Annotations for the testing portion of the dataset are withheld from the public for benchmarking [10].

For the purposes of this project, only the 5,000 images with high quality pixel-level annotation were used. Furthermore, the 1,525 images belonging to the test set were excluded due

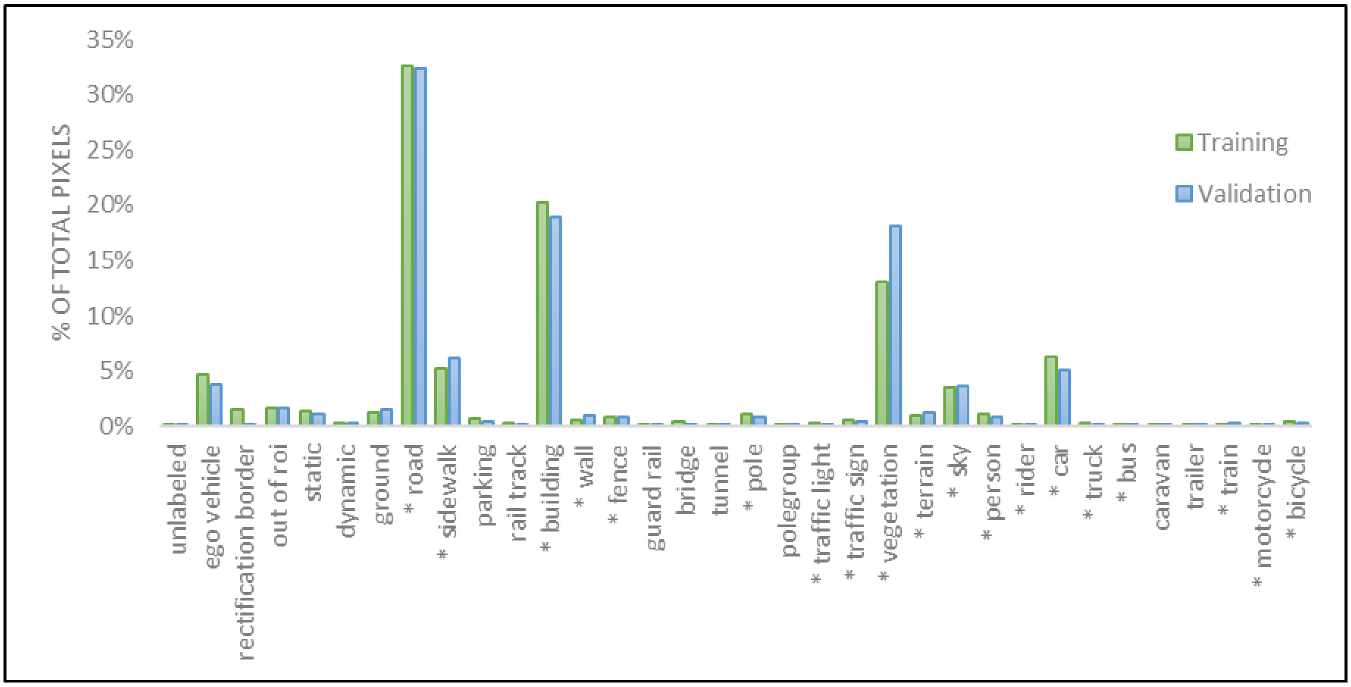

Fig. 1. Distribution of pixel-wise labels in the training and validation sets. An asterisk indicates that it is one of the primary 19 classes used for evaluation of the Cityscapes dataset.

to the lack of labels. This resulted in a total of 3,475 images that were re-split between training, validation, and testing in a ratio of approximately 70%/15%/15%, respectively. The dataset split is summarized below in Table 1.

Table 1. Dataset Split for Model Training and Evaluation

	Training	Validation	Testing	Total
No. of Samples	2,472	503	500	3,475
% of Total	71.1%	14.5%	14.4%	100%

The dataset provides semantic segmentation annotations for 34 different classes that may be commonly encountered in an urban setting. Some examples of the labelled classes are road, building, fence, traffic light, vegetation, person, truck, bus, and bicycle. Figure 1 above provides the distribution of labels in the training and validation sets.

Although the dataset contains annotations for 34 classes, many of the classes are considered too rare and were excluded by the authors for evaluation purposes. The 19 remaining classes that are used for evaluation are indicated in Figure 1 above by an asterisk beside the class name.

3.2 Methodology

After deciding upon and gathering the required data, a PyTorch model was set up to train a model with the task of performing semantic segmentation. The garbage classification model that was set up throughout the course was used as the starting point for this model. The model would be trained on all 34 classes available in the Cityscapes dataset to provide a wholistic segmentation.

Average statistics were initially calculated for the training samples to appropriately normalize the data. Mean values of [0.285, 0.322, 0.282] with a standard deviation of [0.176, 0.181, 0.178] were calculated for each of the 3 input channels. This normalization was included as part of the transformations to help the model train more effectively.

3.2.1 Model Design

Following the literature review, a neural network based on the encoder-decoder architecture was built in PyTorch since this is the most common design when it comes to training semantic segmentation models. For the encoder portion of the network, a VGG16_BN backbone was used. This is similar to the VGG16 architecture discussed in class, except that it also includes batch normalization steps. Batch normalization standardizes the input to a layer for each mini-batch and results in faster, more stable models [11].

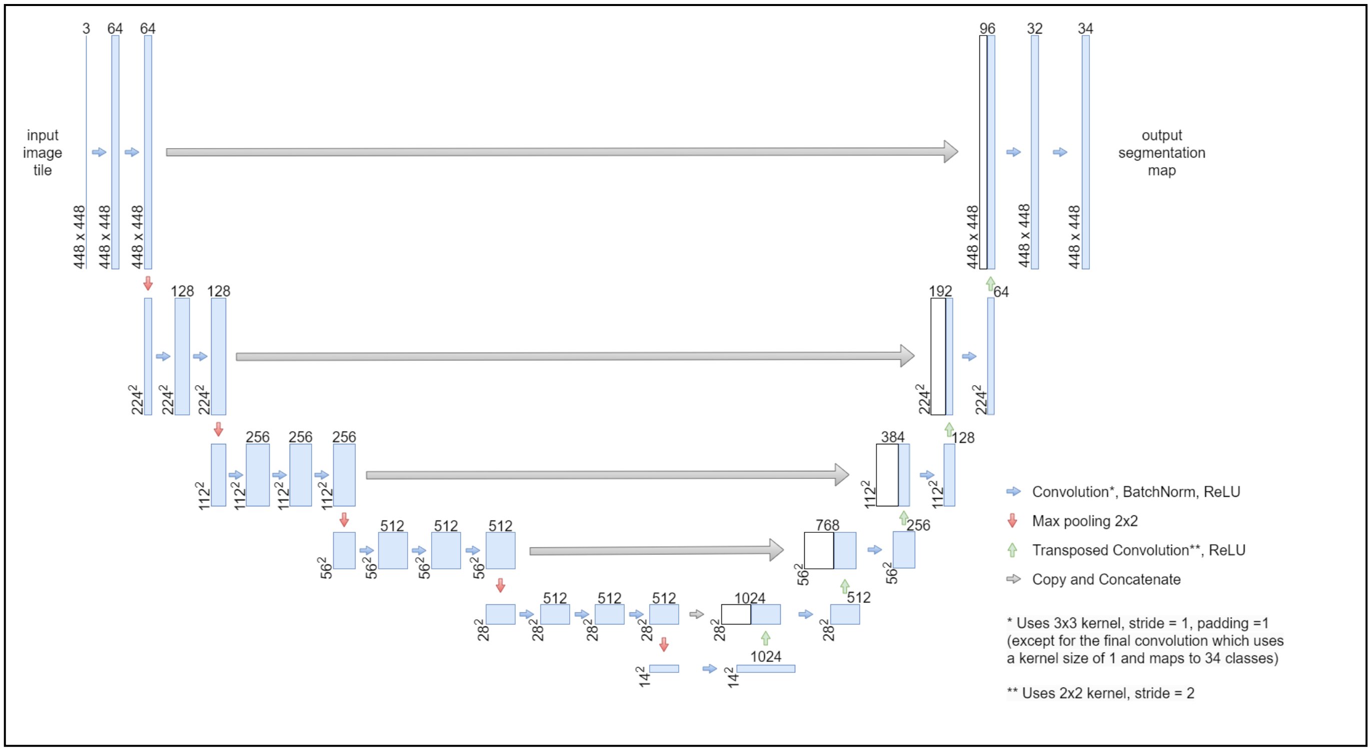

Fig. 2. Overview of the architecture for the neural network trained in this project.

The decoder portion of the network is inspired by U-net but is modified to complement the VGG16_BN backbone chosen for the encoding half of the network [12]. Every decoding step is made up of an up-convolution, followed by one or more regular convolutions.

Up-convolutions are somewhat analogous to max pooling in the encoding portion of the network, except that they serve to increase the image resolution and have an impact on the number of feature channels. They are implemented using a transposed convolution with a 2x2 kernel and a stride of 2. The result of this is a doubling of the resolution, but a halving of the feature channels.

The result of each up-convolution is concatenated with the corresponding feature map from the encoding half of the network before a further 3x3 convolution is applied to the combined tensor. Padding is used during each of the convolutional steps. Figure 2 above provides a complete illustration of the network architecture used for this project.

Given the computational resources available to the team, a batch size of 8 was selected. The neural network used for this project had a total of 29.3 million parameters, all of which were trainable. The estimated total GPU consumption for the model was 9.9 GB based on PyTorch’s summary.

3.2.2 Metrics

Convolutional neural networks trained for image segmentation tasks typically use cross-entropy as the loss function and are subsequently evaluated using the Dice score or Jaccard index [13]. This is consistent with the approach taken in this project.

The Jaccard index, also known as the intersection-over-union ratio (IoU), is calculated exactly as its common name suggests. Figure 3 below illustrates the calculation of this metric.

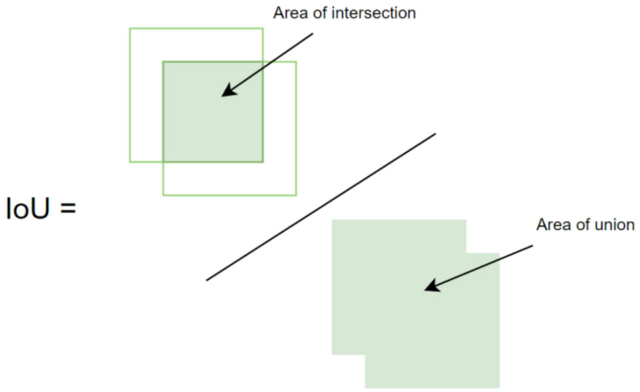

Fig. 3. Jaccard index (aka. IoU) is calculated as the ratio of the area of the intersection to the area of the union.

3.3 Model Training

The model was trained on the University of Calgary Teaching and Learning Cluster (TALC) in order to access a GPU and speed up the training process.

The initial model was set to run for 100 epochs, with early stopping if no improvement was seen on the validation loss for more than 10 epochs. This model terminated after 19 epochs, with the lowest validation loss being achieved on the 9th epoch.

In an attempt to train the model past 19 epochs, a second model was ran with the early stopping threshold increased to 20 epochs. Like the first model, there was no improvement past the 10th epoch and training terminated after the 30th epoch.

One last model was trained with the learning rate increased from the initial 0.001 to 0.005. This model showed improvement for slightly longer, with the best model being reached after the 16th epoch, and final termination 20 epochs later.

During each of the training sessions, the best model was periodically saved on the TALC cluster. This model was then downloaded to the team’s local computers to calculate metrics on the test set and visualize the results. Each of the models took upwards of five hours to train, with a model PTH file size of over 110 MB.

3.4 Applying Semantic Segmentation to Video

Images from the test dataset were combined to form an MP4 video utilizing OpenCV’s VideoWriter [14]. The existing dataset was used to compare the results obtained from the video and images.

After obtaining the input video, each frame was processed independently, essentially analyzing each individual image within the video [15]. Each frame was processed by resizing, normalizing, and converting to a torch tensor, as done when evaluating the images. The model was then used to predict the segmentation mask. A colour mask was applied to the segmentation mask based on the number of classes within the dataset, which is 34. This colour mask was combined with the original frame in order to visualize and showcase the model’s segmentation capabilities. Subsequently, each processed frame was written to the output video file, resulting in a video with a mask overlay.

Utilizing the ground truth masks, an additional video was created to serve as a reference for comparing the model’s performance with the actual ground truth.

4. RESULTS AND DISCUSSION

IoU for each of the three models was calculated using the test set. The first model achieved the highest metrics, with a mean IoU of 43.1%. The other two models were relatively close behind, with an IoU of 42.5% and 42.1%, respectively. The overall results are summarized in Table 2 below, while Table 3 summarizes the IoU for each of the 19 classes used in the evaluation of the best model.

Table 2. Summary of Model Performance on the Test Set

Model #	Hyperparameters	Mean IoU
1	patience = 10 epochs, learning rate = 0.001	43.1%
2	patience = 20 epochs, learning rate = 0.001	42.5%
3	patience = 20 epochs, learning rate = 0.005	42.1%

Table 3. IoU by Class for Model #1 (Best Model)

road 92.6%	sidewalk 65.5%	building 81.3%	wall 19.3%
fence 24.5%	pole 33.9%	traffic light 24.2%	traffic sign 44.2%
vegetation 86.8%	terrain 44.7%	sky 85.5%	person 50.4%
rider 0.02%	car 85.1%	truck 18.4%	bus 0.2%
train 11.4%	motorcycle 0.9%	bicycle 49.3%	**Mean** **43.1%**

Given that this is an imbalanced dataset, there does appear to be some bias towards the more popular classes such as road, building, vegetation, and sky, which all evaluated to an IoU ratio of more than 80%. However, since many of the less prominent labels scored quite a bit lower, the mean IoU comes out to 43.1%. The mean IoU is simply calculated as the arithmetic average for all of the classes.

Another metric that is sometimes reported is the overall IoU in which all of the pixels are considered in aggregate. In this case, the three models mentioned above all achieved a score of more than 80%. However, this metric tends to favour predictions towards the more common classes and is therefore not considered a great representation of the model's capabilities. Since most benchmarking for image semantic segmentation is done on the basis of mean IoU, this was the main metric reported for this project.

Current state-of-the-art semantic segmentation results for the Cityscapes dataset are quite a bit higher than what we were able to achieve in this project. The mean IoU for some of these models is nearing 90%, with the 2022 front-runner achieving 87% [16].

Model predictions were visualized by comparing the original image, the ground truth mask, and the predicted mask for some of the test data side-by-side. Examples for a couple of the video frames are shown in Figure 4 below. Even with a relatively low mean IoU, the visual results were generally quite good.

The strengths of the methodology used for this project are detailed below:

- Dataset: The Cityscapes dataset is one of the most popular datasets for training and benchmarking semantic segmentation models in urban settings. The dataset is also relatively large, providing public access to thousands of images with high quality pixel-level annotations.

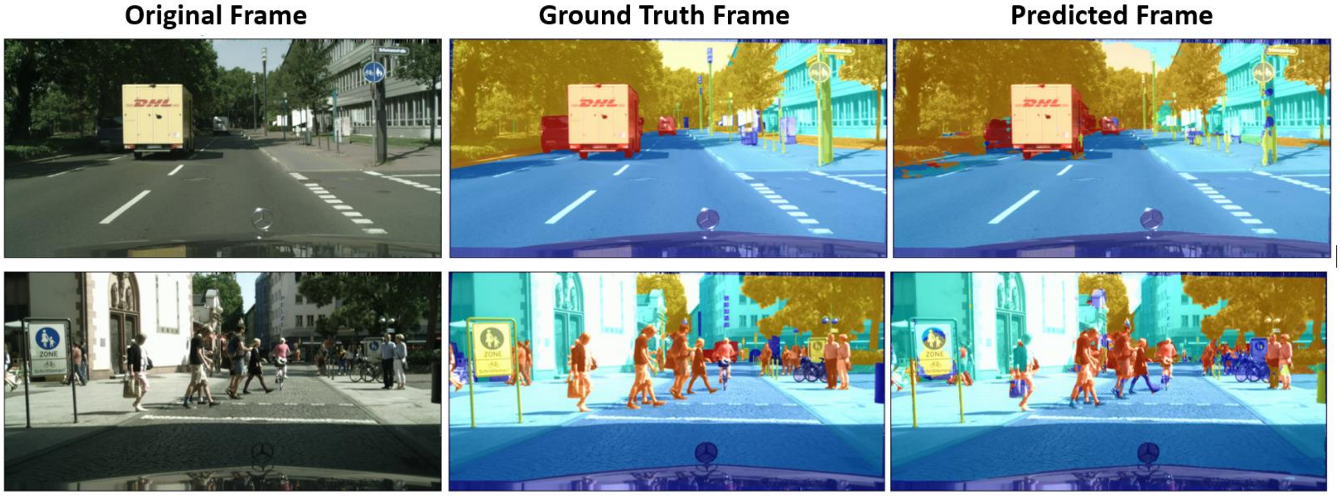

Fig. 4. Example of semantic segmentation on some video frames

- Base Architecture: Most modern state-of-the-art semantic segmentation models are based on the encoder-decoder architecture. This is consistent with the methodology employed in this project.
- Input Size: Since the neural network used was fully convolutional, the model can be used with images that differ in size.
- Video: Segmentation on videos with the purpose of mimicking the perspective of autonomous vehicles was successfully completed, with visual results appearing to be fairly good when compared to the ground truth mask.

Some of the limitations of the methodology used for this project are detailed below:

- Embedded System: This project does not consider that the models used in autonomous vehicles would need to run in real-time with computational resources that are likely more limited [8].
- Geography: The dataset only covers European cities, mainly in Germany. The model may not be generalizable to North America since the roads, traffic signs, and buildings may differ significantly.
- Seasonality and Weather: The images were taken over the spring and summer. Furthermore, the images were deliberately taken during good weather conditions since adverse weather conditions would required specialized techniques [10]. This too would limit the application of this methodology to autonomous vehicles which will encounter a much wider range of conditions.
- Overfitting: With the current methodology, the best model (ie. the one with the lowest validation loss) was achieved after only 9 epochs. The training loss continued to decrease after this, but the validation loss was increasing. Given more time and computational resources, it may be possible to obtain better overall results by applying additional image augmentations, using a larger dataset, and working with a larger range of hyperparameters.
- Video: Modern state-of-the-art video segmentation models make use of past frames in predicting the segmentation of the current frame [8]. The methodology used in this project considers each frame in isolation.
- Data Size: Although the dataset size of 3,475 samples is quite significant for the purposes of this project, it is rather limited when it comes to deep learning in general. For example, COCO is a similar dataset published by Microsoft that includes more than 300,000 images [17].
- Labels: The methodology used in this project is that the model was trained on all 34 classes available in the Cityscapes dataset, but only evaluated on the main 19 classes. The 15 classes that were not considered in the evaluation should have instead been grouped into an “other” category instead of training the model on each of these independently.

5. CONCLUSIONS

Following the second assignment of ENEL 645, the team considered some areas of image semantic segmentation that we wanted to explore further. This included training a semantic segmentation model from scratch and applying the technique to a video in order to mimic its application in autonomous driving. These are the tasks that were tackled for this final project.

After a review of the current literature, a neural network based on an encoder-decoder architecture was set up in PyTorch. The initial framework for this network was based on the garbage classification example covered throughout the course, but modifications were made to the Dataset, Dataloader, and neural network classes as required.

The network was trained on the Cityscapes dataset, which is one of the most popular benchmarking datasets in the field of image semantic segmentation. A total of 3,475 images were split between training, validation, and testing.

Three models with varying hyperparameters were trained, with the best model achieving a mean IoU of 43.1%. Although this is somewhat on the lower end when considering that current state-of-the-art models are achieving around 87%, the predicted results appeared visually promising. Some of the strengths and limitations of the methodology used for this project were explored and discussed.

To mimic the application of semantic segmentation to autonomous driving, the model was subsequently used to semantically segment the frames of a video. The series of frames was recombined back into a video following the prediction. With this, the tasks that the team set at the end of the second assignment were successfully completed.

6. REFERENCES

[1] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey.” arXiv, 2020. doi: 10.48550/ARXIV.2001.05566.

[2] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, “Review the state-of-the-art technologies of semantic segmentation based on deep learning,” Neurocomputing, vol. 493. Elsevier BV, pp. 626–646, Jul. 2022. doi: 10.1016/j.neucom.2022.01.005.

[3] S. Cakir, M. Gauß, K. Häppeler, Y. Ounajjar, F. Heinle, and R. Marchthaler, “Semantic Segmentation for Autonomous Driving: Model Evaluation, Dataset Generation, Perspective Comparison, and Real-Time Capability.” arXiv, 2022. doi: 10.48550/ARXIV.2207.12939.

[4] D. Milakis, M. Snelder, B. V. Arem, B. V. Wee, and G. Homem De Almeida Correia, “Development and transport implications of automated vehicles in the Netherlands: scenarios for 2030 and 2050,” European Journal of Transport and Infrastructure Research, p. Vol 17 No 1 (2017), Jan. 2017, doi: 10.18757/EJTIR.2017.17.1.3180.

[5] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, “Examining accident reports involving autonomous vehicles in California,” PLOS ONE, vol. 12, no. 9. Public Library of Science (PLoS), p. e0184952, Sep. 20, 2017. doi: 10.1371/journal.pone.0184952.

[6] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, “Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving,” 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). IEEE, Oct. 2019. doi: 10.1109/iccvw.2019.00294.

[7] Y. Xing, L. Zhong, and X. Zhong, “An Encoder-Decoder Network Based FCN Architecture for Semantic Segmentation,” Wireless Communications and Mobile Computing, vol. 2020. Hindawi Limited, pp. 1–9, Jul. 07, 2020. doi: 10.1155/2020/8861886.

[8] H. Wang, W. Wang, and J. Liu, “Temporal Memory Attention for Video Semantic Segmentation.” arXiv, 2021. doi: 10.48550/ARXIV.2102.08643.

[9] J. Portillo-Portillo et al., “FASSVid: Fast and Accurate Semantic Segmentation for Video Sequences,” Entropy, vol. 24, no. 7. MDPI AG, p. 942, Jul. 07, 2022. doi: 10.3390/e24070942.

[10] M. Cordts et al., “The Cityscapes Dataset for Semantic Urban Scene Understanding.” arXiv, 2016. doi: 10.48550/ARXIV.1604.01685.

[11] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” arXiv, 2015. doi: 10.48550/ARXIV.1502.03167.

[12] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” arXiv, 2015. doi: 10.48550/ARXIV.1505.04597.

[13] J. Bertels et al., “Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory &amp; Practice,” arXiv, 2019, doi: 10.48550/ARXIV.1911.01685.

[14] BoboDarph, “How to make a movie out of images in Python,” *Stack Overflow*, 06-Jul-2017. [Online]. Available: https://stackoverflow.com/questions/44947505/how-to-make-a-movie-out-of-images-in-python. [Accessed: 24-Mar-2023].

[15] A. Saha, “Read, write and display a video using opencv,” *LearnOpenCV*, 05-May-2021. [Online]. Available: https://learnopencv.com/read-write-and-display-a-video-using-opencv-cpp-python/. [Accessed: 24-Mar-2023].

[16] “Papers with code - cityscapes test benchmark (semantic segmentation),” *The latest in Machine Learning*. [Online]. Available: https://paperswithcode.com/sota/semantic-segmentation-on-cityscapes. [Accessed: 24-Mar-2023].

[17] “COCO (Microsoft Common Objects in Context),” *COCO Dataset | Papers With Code*. [Online]. Available: https://paperswithcode.com/dataset/coco. [Accessed: 24-Mar-2023].