CS482/682 Final Project Report Group 10

Un-Supervised Traffic Segmentation using Encoder-Decoder Structure

Kiana Bronder/kbronde1
Sohan Gadiraju/sgadira2
Priya Gillan/pgillan2

1 Introduction

Background In recent years, there has been a rapid increase in the development of autonomous and driving-assisted cars. The implementation of self-driving features into vehicles can greatly decrease the number of car accidents. In 2022, Tesla reported one crash for every 4.85 million miles driven while using their Autopilot technology, whereas people who were not, reported one crash for every 1.40 million miles driven [3]. A major obstacle in developing autonomous vehicles is having detailed data for training, as annotating these images for segmentation is a tedious process. However, it is a lot easier to annotate object detection datasets as only objects of value have to be annotated. Therefore having a way to transform object detection datasets into segmentation masks, as well as being able to get segmentation masks without any labels, is very crucial to improving self driving models quickly.

In this project, we attempt to make a robust deep network that is able to categorize segments of an image into cars, pedestrians, and cyclists for vehicles to have a more accurate representation of their environment using unsupervised training. Utilizing the segmented representation, the car then can determine if something is in front of it and what that object is.

Related Work One recent work in traffic/road detection is Alvarez et al., which creates a self-supervised convolutional neural network to learn high–order features from noisy labels for road scene segmentation. Their network learns the random stochastic patterns of the roads, yielding a 15% improvement in 3D road segmentation from still images compared to the baseline [4]. Another paper that tackles this problem is Bichen et al. which utilizes a convolutional neural network, known as "SqueezeSeg", that trains on 3D LiDAR point clouds provided by the KITTI dataset. They trained their model by creating a LiDAR simulator in GTA V, yielding accuracies of approximately 60%, 20%, and 25% for cars, pedestrians, and cyclists, respectively, within a runtime of 8.7 ± 0.5 ms per frame [5].

2 Methods

Dataset The dataset we will be using to train our model is the "Traffic Detection Project" dataset from Kaggle [8]. The dataset contains 6633 images taken from traffic cameras from many different countries with detailed bounding-boxes for object detection tasks. The annotations include various objects, including vehicles, pedestrians, traffic signs, cyclists, and motorcyclists. No major preprocessing is needed for this dataset as it is already split into training, validation, and test sets. We incorporated data normalization and data augmentation in our training set to zero-center our data while also trying to avoid overfitting, as well as downsampling all images to 256x256, to enable faster training and lower computation costs.

Setup, Training and Evaluation We used an encoder-decoder-like structure (similar to UNet) with a Cross Entropy loss function, Adam optimizer, and learning rate scheduler to prevent early plateauing. The encoder architecture of this model comes from

Yolo V5 (refer to Figure 1). The decoder block consists of bilinear upsampling, which allows us to keep finer detail information by factoring in all nearby pixel values, followed by double convolutions with batch normalization and ReLU activation. The architecture (shown in Figure 2) consists of a 24-layer encoder block and 8-layer decoder block, as skip-connections are only added at the "C3" blocks in the Yolo V5 model. The number of filters/depth increases at each "C3" block, making it the ideal skip-connection to gain more localization information for our segmentation masks. After going through the encoder and decoder, we use bilinear interpolation to upsample to our original image size in order to retain more local information as per all nearby pixels. Lastly, we utilize a sigmoid activation to get class probabilities for the segmentation and one-hot encoding to determine the label.

We pretrained the YOLOv5 model for 50 epochs to perform object detection on the training set in order to enable supervised learning of key patterns later used for segmentation. We then augmented the data with flips and crops to better train the model for another 25 epochs (compute limits) at a learning rate of 0.1, predicting a mask with all 5 image classes. As we did not have true masks for comparison to calculate loss, we attempted K-Means clustering to help our model learn by comparing model outputs to cluster predictions. We also tried to incorporate just K-means clustering without CE loss, but to no avail. After training, we tuned our hyperparameters using the validation set and tested our model on the test set. Due to the lack of labels, most of our segmentation masks were evaluated qualitatively. In order to get quantitative measurements (e.g., precision and recall), we attempted to convert our segmentation masks back into bounding boxes to evaluate it against the labels but failed to do so.

3 Results

Our pretrained model with YOLOv5 achieves a precision of 85% with a recall of 80% on our dataset, indicating it is relatively good at detecting the various different objects without overfitting the data (Figures 4 and 5). However, this accuracy from pretraining does not seem to correlate greatly to our unsupervised model, as it is able to segment parts of the road and some cars, but struggles with segmentation of crowded areas (shown in Figure 3). It is not very successful at segmenting people or smaller objects on the road as we had hoped. As our loss did not decrease by much, we believe our model was not learning properly. Our code and more information can be found at this link.

4 Discussion

As our dataset is specifically for object detection, one of our main challenges was implementing unsupervised learning to generate masks. Initially, we were only able to use the first six layers of YOLOv5 for our encoder with 3 layers for decoder. This led to a lot of local and spatial information missing due to the large depth of YOLO and our lack of skip-connections. After we were able to get our encoder block fully working, we added skip connections to every layer where the number of filters changed, as we wanted general features and local information from the early layers injected into our higher-representation layers and allow for more learning. Our masks significantly improved, but neither the learning nor the loss improved much throughout the training. One way we could have improved our model is by properly implementing K-means clustering into our loss function or using strictly K-means clustering. Utilizing cross-entropy loss for unsupervised learning was not very helpful, as we did not have class labels to properly calculate loss, making it difficult for our model to learn. K-means clustering, on the other hand, groups based on distance/similarity between features,enabling us to group features of classes together to better segment objects such as a person versus car. Lastly, we would have also liked to implement a better method to evaluate our model as we were only able to do so qualitatively. One approach to this would have been transforming our segmentation mask into multiple masks and creating bounding boxes to evaluate it with the original labels.

References

[1] Volpe National Transportation Systems Center. "How Much Time Do Americans Spend Behind the Wheel?" [Online]. Available: volpe.dot.gov [Accessed: November 6, 2023]

[2] Office of Highway Policy Information. "Average Annual Miles per Driver by Age Group," May 31, 2022. [Online]. Available: fhwa.dot.gov

[3] Tesla. "Tesla Vehicle Safety Report." [Online]. Available: tesla.com[Accessed: November 6, 2023]

[4] Alvarez, J.M., Gevers, T., LeCun, Y., Lopez, A.M. (2012). "Road Scene Segmentation from a Single Image." In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds) Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7578. Springer, Berlin, Heidelberg. [Online]. Available: doi.org [Accessed: November 8, 2023]

[5] Wu, B., Wan, A., Yue, X., Keutzer, K (2018). "SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud." 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane, Australia. [Online]. Available: ieeexplore.ieee.org [Accessed: November 8, 2023]

[6] Gupta, A., Anpalagan, A., Guan, L., Shaharyar Khwaja, A (2021). "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues." Array, Volume 10. [Online]. Available: doi.org [Accessed: November 8, 2023]

[7] N. Tomar, "What is UNET?," Medium, medium.com (accessed Nov. 8, 2023).

[8] kaggle.com

5 Appendix

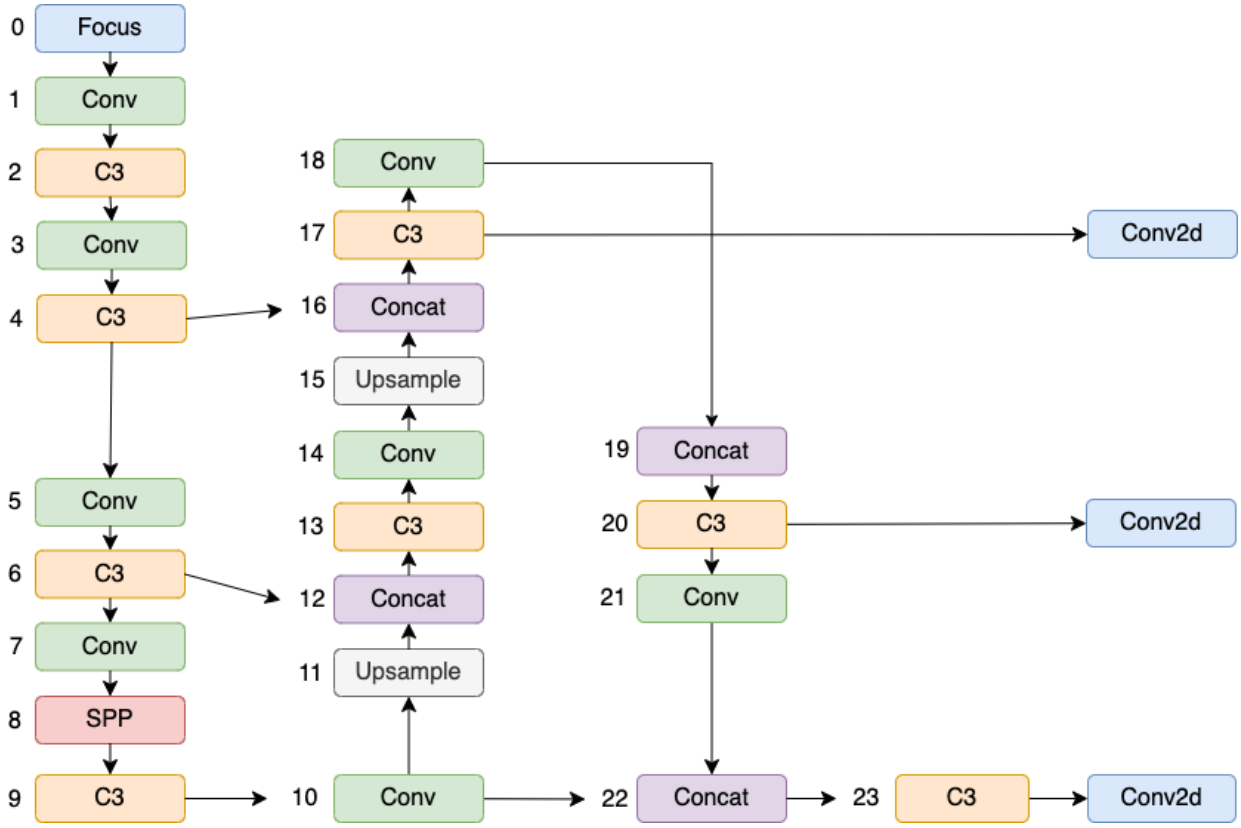

Figure 1: Yolov5 architecture

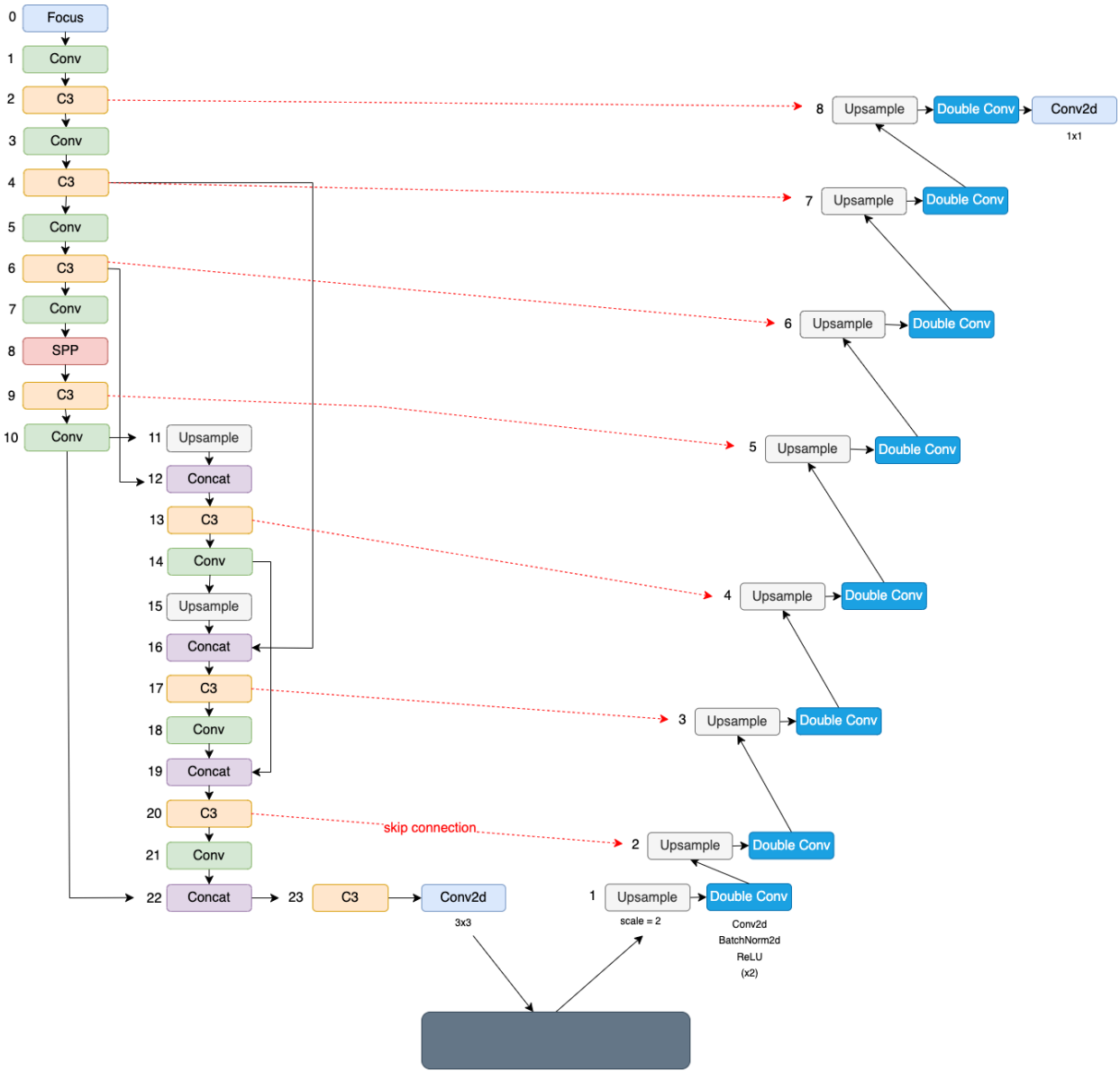

Figure 2: Full encoder-decoder model architecture

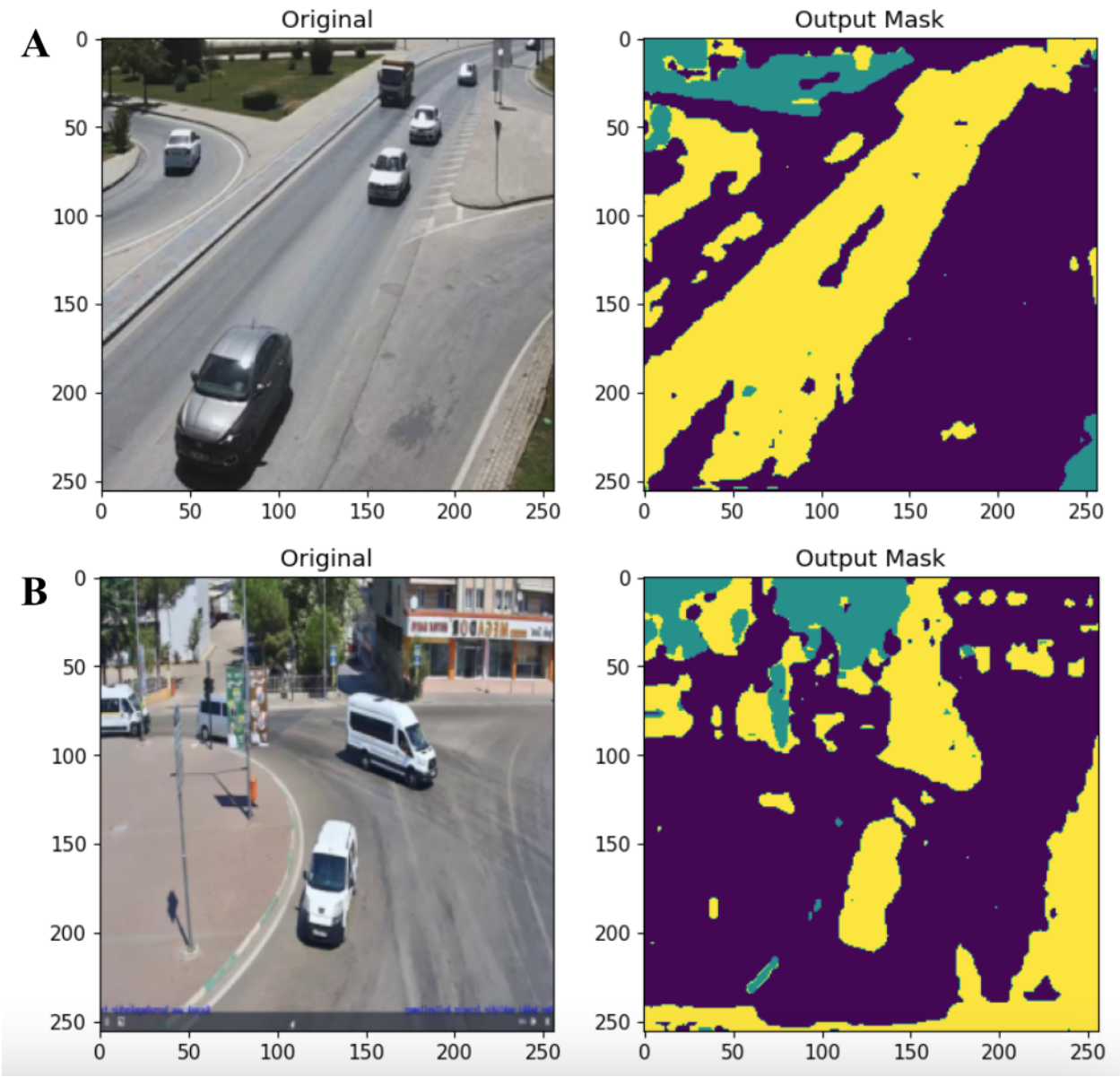

Figure 3: Example input and output images.
A) Errors in car and road segmentation. B) Segmented cars but error in separating the van from the tree.

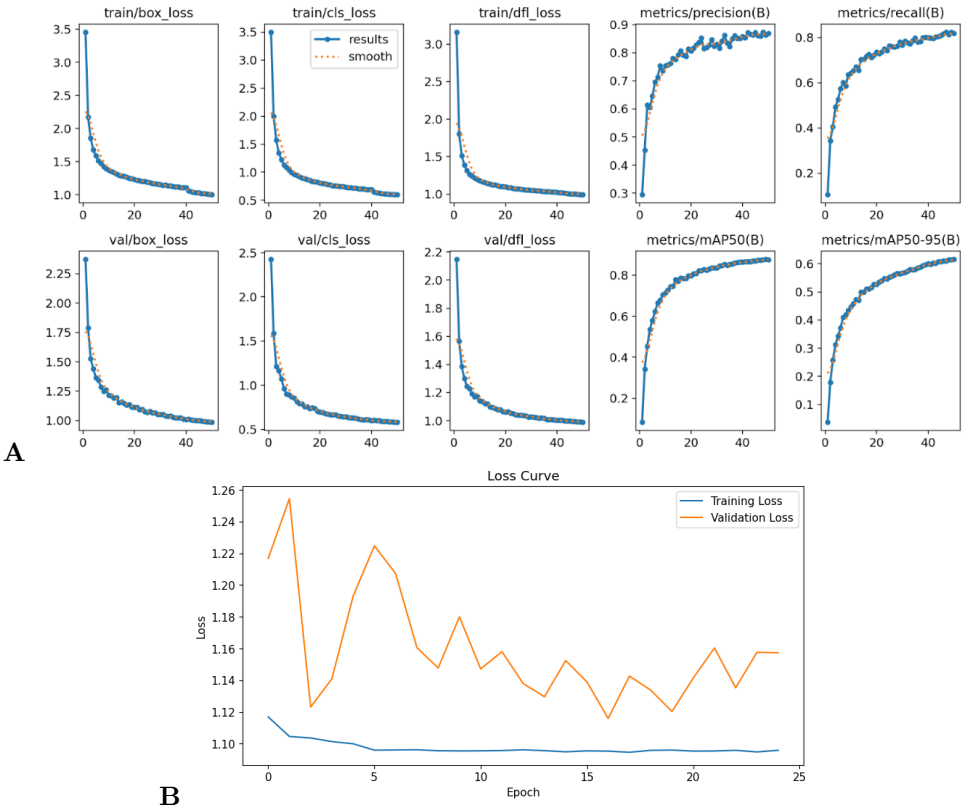

Figure 4: Comparison of loss curves.
A) Loss and Evaluation Metrics for Pre-trained Yolov5 Model. B) Loss curve for full encoder-decoder model

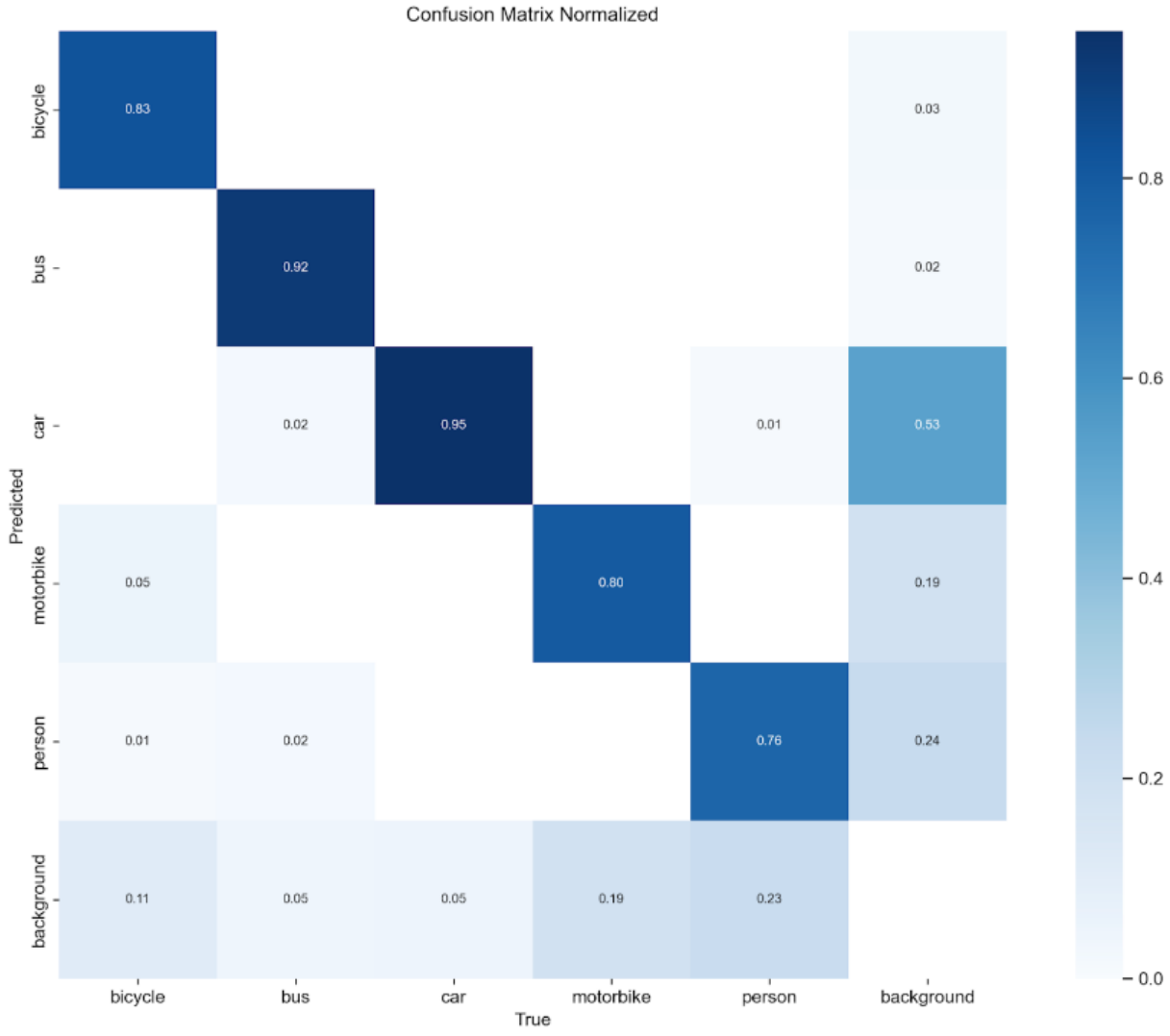

Figure 5: Normalized Confusion Matrix for Pre-trained Yolov5 model