Multi Modal Segmentation

Parth Patel
Made the baseline model and the structure of the Iconseg model
Khushin Patel
Made the RGF module and the data generator

Abstract

We will perform Semantic segmentation on the City Scapes Dataset using a special RGF module proposed in our paper. In our paper, we will introduce the RGF module in detail, presenting it as a component expected to enhance the model's segmentation performance, as proposed by its authors, and we will share our findings on this claim. The authors also suggest that this RGF model solves the problem where a multi-modal model's performance deteriorates when there is a shadow in the image. The proposed RGF model is said to be able to solve this issue by combining information learned from both the depth-image and the RGB image in a way that other multi-modal models don't.

Problem Statement

Multi-modal data refers to information captured from multiple types of sensors or data sources, each providing unique perspectives on the environment. For autonomous vehicles, this includes different data like an RGB image, depth maps, thermal images, LiDAR data, etc. However, this runs into a problem when autonomous vehicles struggle to accurately detect positive and negative obstacles when relying on inconsistent multi-modal data. While multi-modal fusion networks like FuseNet generally outperform single-modal networks, they become ineffective when one modality, such as depth, is missing or degraded leading to poor segmentation performance. This is often caused by environmental challenges like shadows, glares, or limited depth sensing range.

Background Material

A. Single Image Semantic Segmentation Networks
Building on existing methods, SegNet[1] introduces a revolutionary upsampling method that uses the encoder's pooling indices, reducing memory needs and doing away with the need to learn upsampling filters. SegNet strikes a balance between memory economy and competitive accuracy, according to comparative evaluations conducted on indoor and outdoor environments.

B. Multi-Image Semantic Segmentation Networks
Numerous studies have shown the value of using depth information to improve segmentation accuracy in both indoor and outdoor scenarios when it comes to semantic segmentation for

RGB-D data. We go over three noteworthy methods below: Bi-directional Cross-Modality Feature Propagation with Separation-and-Aggregation Gate, Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis, and FuseNet.

Using RGB-D data, FuseNet[2] tackles the problem of integrating depth information into semantic segmentation tasks. This work uses an encoder-decoder CNN architecture that combines depth information at various layers inside the encoder, acknowledging the potential of depth as a complimentary cue to RGB in scene interpretation. Before incorporating depth features into the RGB feature maps, the architecture's two parallel networks process RGB and depth data independently.

A highly optimized RGB-D segmentation model designed for real-time applications in mobile robotics is proposed in Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis[3]. By optimizing the network architecture with NVIDIA TensorRT, this model highlights the durability and efficiency of segmentation on RGB-D inputs, allowing for quick inference. The method works consistently on indoor datasets like NYUv2 and SUN RGB-D and is especially useful for indoor applications like human perception and free space recognition. Additionally, tests conducted on the Cityscapes dataset show how well it adapts to outside landscapes, exhibiting adaptability in a variety of settings while retaining real-time processing capabilities.

A comprehensive method for RGB-D semantic segmentation is introduced by Bi-directional Cross-Modality Feature Propagation with Separation-and-Aggregation Gate[4], which tackles the noise and alignment problems that are frequently present in-depth data. In this work, RGB and depth feature representations are iteratively recalibrated using a Cross-modality Guided Encoder. The Separation-and-Aggregation Gate, which performs cross-modal fusion after jointly filtering and recalibration of features from each modality, is the main invention. To improve long-term information propagation between RGB and depth features while maintaining the unique properties of each modality, the model additionally uses a Bi-direction Multi-step Propagation method. This architecture, which was created to supplement conventional encoder-decoder architectures, sets new standards for accuracy in RGB-D segmentation tasks and performs exceptionally well on both indoor and outdoor datasets.

Proposed Solution

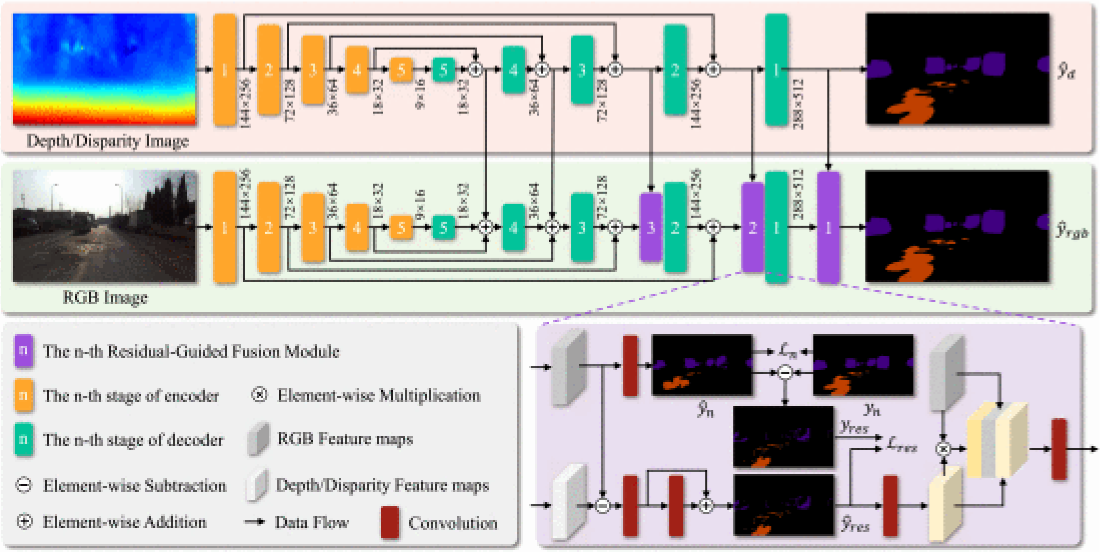


Figure 1: Iconseg Network Structure [5]

To solve this issue we will create a network with two connected streams. The initial stream will use an encoder-decoder design to analyze the depth image. In the same way, the second stream will use an encoder-decoder architecture to handle the RGB image. In the RGB stream the decoder section will include Residual-Guided Fusion modules. The streams will be joined by combining the information from the depth stream's decoder blocks with the Residual-Guided Fusion modules in the RGB stream. In the end, the RGB stream will show the segmentation mask by using both depth and RGB images effectively.

The purpose of the RGF module is to quantify the missing features between RGB features and the ground truth. Rather than directly fusing RGB and depth features, the RGF module extracts complementary features from depth features, effectively addressing the performance degradation that can result from inconsistencies between these data types.

In the Iconseg paper, a 5-stage encoder and decoder are used. However, to keep our model more compact, we employ a 3-stage encoder and a 3-stage decoder. While Iconseg fuses the outputs of the last three decoder stages into the Residual-Guided Fusion module, our model only fuses the output from the final decoder stage into the RGF module.

Implementation

Encoder

Our architecture utilizes the same Encoders as found in the Encoder of the UNET architecture, ie two 3x3 Convolutional Layers with ReLu activation followed by downsampling by a factor .5 using a MaxPooling layer with 2x2 filters. Unlike the UNET model, we perform the encoding and downsampling step not only on our RGB image but also on our Depth Input image and convert it into a bottleneck of shape (16,32,256) just like the RGB image. Both the RGB path and the

Depth path go through 3 encoder layers and then they are made into Bottlenecks which also have two 3x3 Convolutional layers before they are passed into the Decoding step.

Decoder

Both our simple UNET model and the RGF model use Decoders which upsample the bottleneck version of the encoded input images. Just like the Encoders we have separate paths for the Decoder of the RGB image and the Depth image but unlike the Encoders the 2 paths the output of the RGB decoders take in not only the previous RGB representation but also the previous Depth representation and we perform elementwise addition on the two before we finally pass it into the Decoder. We will finally pass the outputs of the last RGB Decoder and the last Depth Decoder into the RGF module to compute the final segmentation output in the RGF model whereas in the regular multi-modal the final output is a 2D convolution on the output of the final RGB decoder.

RGF Module

This module takes two inputs: RGB feature maps and depth feature maps. The RGF module begins by generating the missing features for the RGB modality. Specifically, the RGB feature maps produce an RGB predicted mask y_hat_n using a convolutional layer, where n is the n-th RGF module. A residual mask yres is then generated through element-wise subtraction between y_hat_n and the ground truth yn. This residual mask, yres, represents the missing features of the RGB feature map, with both yn and y_hat_n having the same resolution as the RGB feature maps.

Next, complementary features are extracted for the missing features. Specifically, an element-wise subtraction is performed between RGB feature maps and depth feature maps to compute their difference. This difference is adjusted to the number of classes using a 1×1 convolution. A residual unit with a 3×3 convolution then generates the predicted residual mask y_hat_res, guided by the residual mask yres. The channels of y_hat_res are adjusted to match those of the RGB feature maps through a 1×1 convolution. The adjusted result is then fused with the RGB feature maps via element-wise multiplication. Finally, the adjusted result, fusion result, and RGB feature maps are concatenated along the channel dimension. The output of the RGF module, produced by a 1×1 convolution, is then passed into the next stage in the RGB decoder.

All of these steps are visible in our code for the residual_guided_fusion function. We also only call this RGF module at the very end of the Decoders to produce our final output to save some time during training as calling the RGF multiple times would prolong training way to long on our CPUs.

Loss Function

Our model outputs 2 things the main segmentation output produced by multiple operations performed in the RGF module and an intermediate output produced by the Decoder models but with an additional 2D convolution performed in the RGF module. Both of these outputs are the same size as our true label segmentation masks and hence we compute the loss for both of them based on Categorical Crossentropy as defined in the model.compile function and compute loss and gradients based on the losses from both outputs.

Results

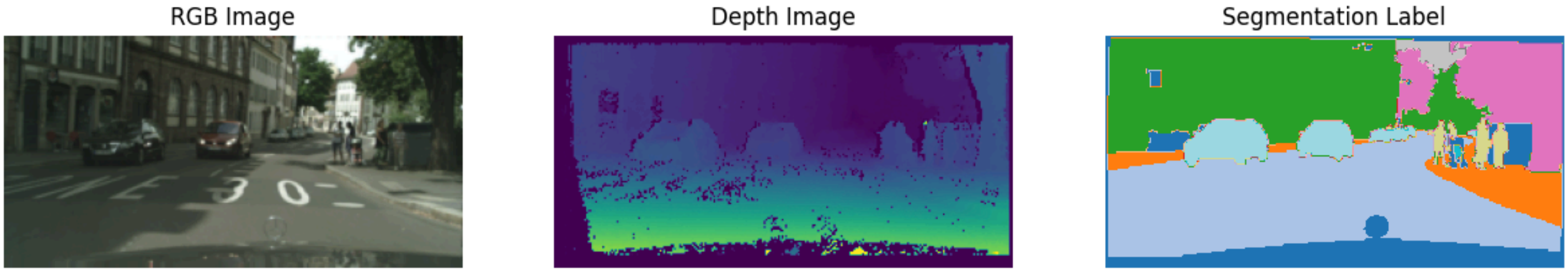
Dataset

The dataset used in this study is a preprocessed version of the CityScapes dataset [7], sourced from the work "End-to-end Multi-task Learning with Attention" [6]. This dataset comprises three types of images: the RGB image, which represents the standard color image; the depth image, which encodes the depth information for each pixel; and the label image, which serves as the segmentation mask. It includes 20 classes for semantic segmentation, with each image having a resolution of 128x256 pixels. The dataset is divided into three subsets: the training set, which contains 2,380 images; the validation set, with 500 images; and the test set, which includes 595 images. The dataset also has a big class imbalance as some of the classes are quite scarce compared to others.

Class Number(original)	Class Number(after offset)	Dataset occurrence(%)
-1	0	12.24624249
0	1	32.44301035
1	2	5.306801676
2	3	20.04192705
5	6	1.079105409
7	8	0.4747958143
8	9	13.95592954
9	10	1.010260061
10	11	3.545020608
11	12	1.033653452
12	13	0.1201418067
13	14	6.146646708
18	19	0.3574544442

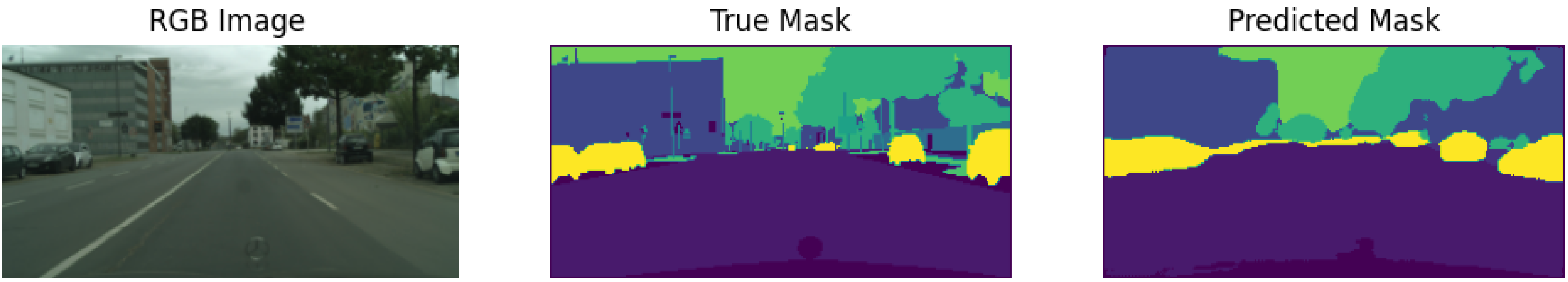
3	4	0.5598603577
4	5	0.7466766614
6	7	0.1800101144
17	18	0.08927545628
14	15	0.2416210014
15	16	0.2122253931
16	17	0.2093416102

This will make it harder for our model to learn how to classify each pixel appropriately but there really isn’t much of a get around in this situation. Here is an example of an instance from our dataset. The original dataset contains classes from -1 to 18 which prompted us to offset all classes by 1 and in our training and evaluation all label images contained classes 0-19.

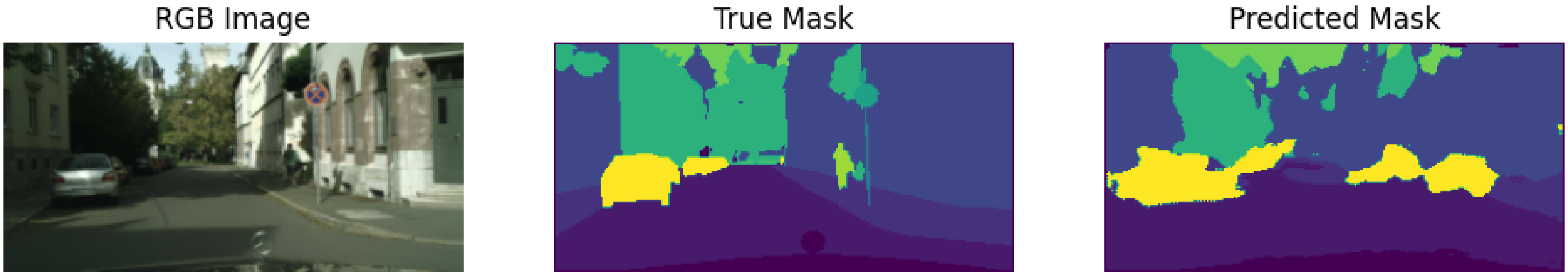


As you can see our dataset has 3 different Images available and the segmentation label has multiple segmentation classes like road, car, humans, sidewalk, building, tree, sky and many more not shown in this particular instance but feel free to change the indices in our code to see more segmentation of different images.

Baseline(UNET)
For our baseline, we trained an encoder-decoder unet with skip connections model on just the RGB input image. This way we can understand what results we would get with just a simple model with only the RGB image. Here is an example of the prediction from the regular UNET model.



Let's evaluate another image but instead in the shadow this time.



The model clearly has a hard time predicting the shapes of objects in the shadows and just the presence of a shadow in the image throws it off from predicting areas outside of the shadows as well. For example, it is struggling with figuring out where the sidewalk is in that image as well as predicting 2 cars on the right-hand side when there is nothing remotely close to a vehicle on the right-hand side.

Here is the evaluation of the model on the test set data using precision, recall, and f1 scores for each of the Classes

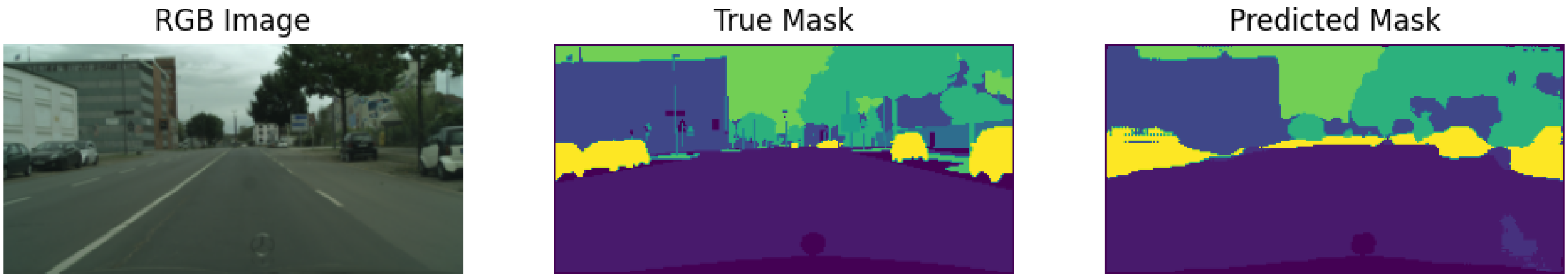
Class	precision	recall	f1-score
0	0.91	0.59	0.72
1	0.82	0.95	0.88
2	0.44	0.42	0.43
3	0.7	0.77	0.73
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0.68	0.77	0.72

10	0	0	0
11	0.81	0.92	0.86
12	0	0	0
13	0	0	0
14	0.59	0.79	0.68
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
accuracy	0.74	0.74	0.74

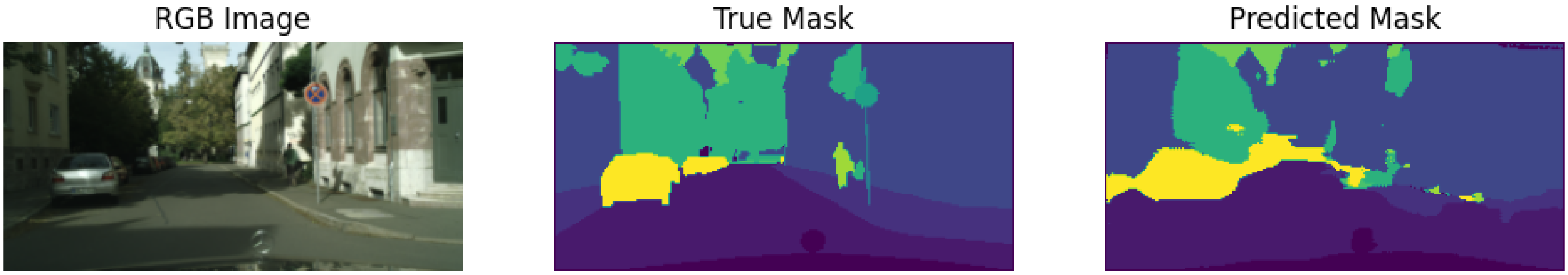
As we can see the model achieved a decent accuracy of .74 but this is kind of misleading because the model is just correctly predicting the classes that have a high frequency in the dataset. It is having a hard time learning the imbalance classes and this is expected because our dataset has quite a big imbalance in it. We were also able to calculate the Mean IoU score for the test data and it is very low at a mere 0.2014 and this can be blamed on the model having a hard time learning the low-frequency classes. The UNET model has a total of 7,760,724 parameters which makes it quite a large model.

Regular Multi-modal model

This model has the same structure as the UNET the only difference here is that we also take into account the depth image. We do this by combining the output of the depth decoders with the output of the RGB encoders using elementwise addition and passing them into the RGB decoders. Once we get the output of the final RGB decoder we perform a 2D convolution to match the number of classes and pass it as our output. Here is the prediction of this model on a sample image.



Let’s also evaluate this model on the image with a shadow in it.



This model from these 2 images seems to be struggling with both regular images and images with shadows, but the surprising part is that this model actually performs the best on both of our metrics as we will discuss later. But it still seems to be doing better with classifying objects on the road when the image doesn’t have a shadow. Let’s look at its precision, recall, and F1 scores.

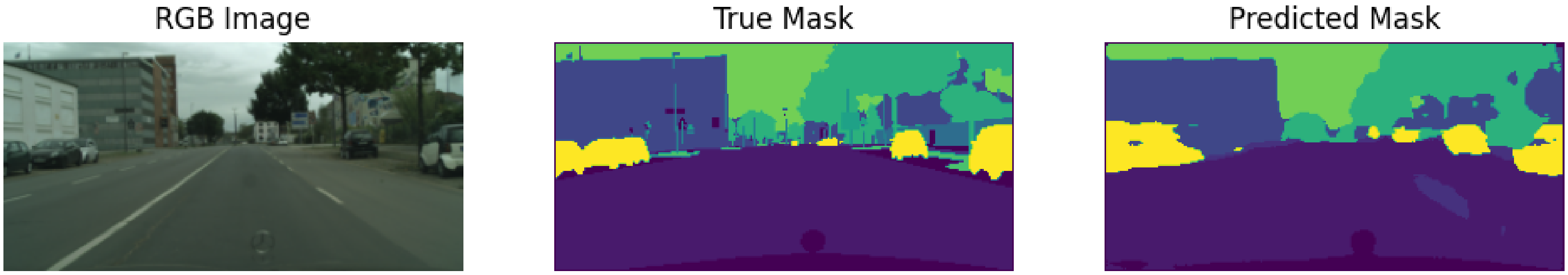
Class	precision	recall	f1-score
0	0.93	0.61	0.73
1	0.87	0.95	0.91
2	0.58	0.55	0.57
3	0.68	0.9	0.78
4	0	0	0
5	0.06	0	0
6	0.47	0	0
7	0	0	0
8	0	0	0
9	0.79	0.73	0.76
10	0.24	0.04	0.07
11	0.88	0.89	0.89
12	0.34	0.17	0.22
13	0	0	0
14	0.62	0.84	0.72

15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
accuracy	0.77	0.77	0.77

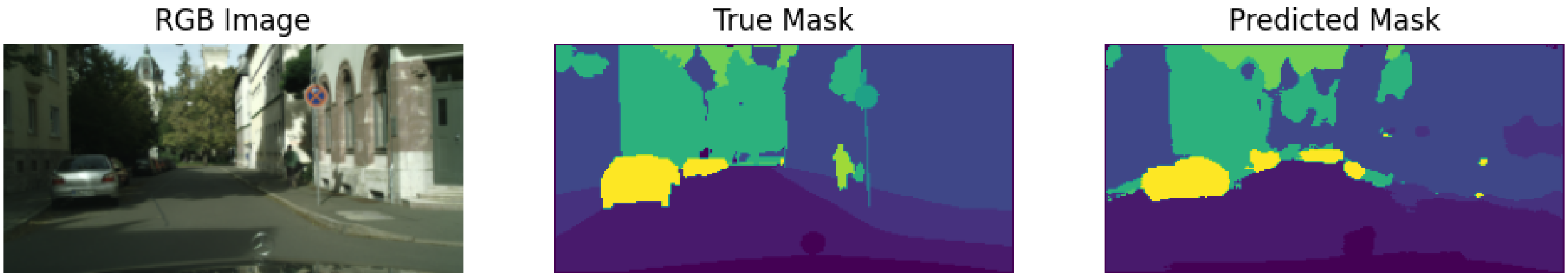
This model has a similar problem as that of UNET where it struggles to learn segmentation for the imbalanced classes. Yet this model achieves the best accuracy out of any of the models. It also achieves a mean IoU of 0.228 on the test set which is the highest of any of our models. This is also not that large of a model as compared to the UNET as it has 3,815,284 total and trainable parameters.

RGF model

This is the full model where we use both the RGB image and the depth image as well as the RGF module. It has the exact same structure as the regular multi-modal model except the output of the final RGB decoder is passed into the RGF model. From there the RGF model performs some operations and it outputs our 3 final outputs. We calculate 3 losses based on these 3 outputs, but only the first output is used as the segmentation the other 2 just allow for extra loss propagation. Here is an example of the prediction from the RGF model on the same image we used for the baseline UNET and the multi-modal.



Let's also evaluate the same shadow image as before and see if the RGF helps identify it better.



This actually does a pretty good job of identifying the shapes of the cars than either of the other 2 models. Yes, it is not perfect but it doesn't just continue to flow the cars on the side like the other models do. And in the image with the shadow it actually does a way better job of not extending the mask of the cars all the way down and instead knows where the outline of the car stops on the left. Let’s look at its precision, recall, and F1 scores.

Class	precision	recall	f1-score
0	0.92	0.61	0.73
1	0.82	0.96	0.88
2	0.5	0.4	0.44
3	0.73	0.8	0.76
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0.67	0.84	0.75
10	0.2	0.02	0.03
11	0.78	0.96	0.86
12	0	0	0
13	0	0	0
14	0.73	0.73	0.73
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
accuracy	0.76	0.76	0.76

It achieves almost the same accuracy as the regular multi-modal but is shy by just .01 and also has the same problems as the other models with learning to identify imbalance classes. This model achieves a mean IoU of 0.21 and does have slightly more parameters at 3,858,112 total and trainable ones. This is very slightly higher than the multi-modal and probably due to the extra RGF at the end. But all in all, I believe this model is the best one because it learns the segmentation shapes a lot better than the other models and does perform quite better under shadows.

Conclusion

We demonstrated a network that uses RGF modules to enhance object segmentation in areas that are shaded. Our model outperforms the baseline UNet and conventional multimodal models in handling shadows thanks to the addition of RGF modules and multimodal input. Importantly, this enhancement is accomplished without appreciably expanding the model's size. The RGF-enhanced model successfully detects objects in shadowed environments while maintaining high accuracy. As suggested in the original research, we propose adding more RGF modules and deepening the network for further enhancement. Longer training could also help all models reach their full potential, as they were only trained for 10 epochs.

Works Cited

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” arXiv.org, https://arxiv.org/abs/1511.00561 (accessed Nov. 8, 2024).
[2] C. Hazirbas, L. Ma, C. Domokos and D. Cremers, "FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture", Proc. Asian Conf. Comput. Vis., pp. 213-228, 2016.

[3] X. Chen et al., "Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation", Proc. Eur. Conf. Comput. Vis., pp. 561-577, 2020.

[4] D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld and H.-M. Gross, "Efficient RGB-D semantic segmentation for indoor scene analysis", Proc. IEEE Int. Conf. Robot. Autom., pp. 13525-13531, 2021.

[5] Z. Feng, Y. Guo, D. Navarro-Alarcon, Y. Lyu and Y. Sun, "InconSeg: Residual-Guided Fusion With Inconsistent Multi-Modal Data for Negative and Positive Road Obstacles Segmentation," in IEEE Robotics and Automation Letters, vol. 8, no. 8, pp. 4871-4878, Aug. 2023, doi: 10.1109/LRA.2023.3272517. keywords: {Decoding;Fuses;Feature extraction;Streaming media;Roads;Convolution;Sun;Negative obstacles;road obstacles;multi-modal fusion;semantic segmentation;autonomous vehicles},

[6] S. Liu, E. Johns, and A. J. Davison, “End-to-end multi-task learning with attention,” arXiv.org, https://arxiv.org/abs/1803.10704 (accessed Nov. 8, 2024).

[7] M. Cordts et al., “The cityscapes dataset for Semantic Urban Scene understanding,” arXiv.org, https://arxiv.org/abs/1604.01685 (accessed Nov. 8, 2024).