Semantic and Instance Segmentation

Abhilash Kuhikar, Raunak Vijan, Saurabh Mathur, and Shivam Rastogi

Abstract—Robots are being used in a variety of environments from outer-space to deep seas.Vision-based navigation controls a robot's movement by analyzing image frames from the robot's camera. Thus, precise image understanding is necessary for vision-based autonomous robot navigation. Our work deals with semantic segmentation of images by labeling each pixel in the image as one of many classes. We trained and tested the popular segnet architecture for semantic segmentation using various loss functions on the camVid11 dataset. Through our work we show that given the same setting and the environment, the soft dice loss function gives better results for the semantic segmentation task. We also show that with little change in architecture such as Bayesian segnet and some postprocessing methods we achieve better accuracy.

1 Introduction

The research problem that we originally set out to solve was that of robot navigation. Robots require fine-grained perception of their environment to safely navigate it. Our initial motivation was to help the robots navigate in the deep under water environment. Unfortunately we couldn't collect enough images required for the experiments but we still kept our motivation alive and experimented with the other data set. There are three image processing tasks that provide precise mapping of the entire image.

- *Image Segmentation* aims to partition an image into several coherent regions. Many earlier works like Shi and Malik focused on low level features like color and locality to define coherence. Recent works have defined coherence of image partitions on the basis of the type of objects that the partitions represent.
- *Semantic Segmentation* labels each pixel to be belonging to one of several object classes. Thus, pixels belonging to the same object class are considered coherent.
- *Instance Segmentation* attempts to not only label each pixel as one on many object classes but also to separate each instance of each object. Thus, this task combines semantic segmentation with object detection.

In our project, we have tried to evaluate semantic and instance segmentation algorithms on data that simulates the conditions that a vision based robot navigation system may face. We have tried different loss functions and a variant of SegNet. For post processing, we have utilized CRF which removes incoherent regions from images and smoothens the output.

2 Background and Related Work

2.1 Semantic Segmentation

2.1.1 SegNet

SegNet is a fully convolutional neural network based encoder-decoder style architecture developed to make faster inference compared to the Fully Convolutional Network architecture. The segnet architecture works for any arbitrary image size and outputs the semantic segmentation mask of the same size as that of the input [1]. The first 13 convolutions layers of the VGG16 network are used as the encoder. The vgg16 is trained on the ImageNet dataset and the encoder of the segnet is initialized with these trained weights. The decoder is a mirror of the encoder and upsamples the encoded representation of the image using max-unpooling. The pooling indices from the encoder are used in the max-unpooling process. The resulting sparse high-resolution feature maps are convolved and a softmax activation is applied to produce a dense segmentation mask. The segnet architecture is trained on the camvid road dataset with the loss function as the weighted cross entropy. Here we show that the soft dice loss function performs better than the weighted cross entropy loss given the same settings and the environment.

2.1.2 Bayesian SegNet

The Bayesian SegNet reinterprets the main SegNet architecture using the Bayesian probabilistic framework to quantify the uncertainty in predictions [2]. The key idea here is to add a dropout layer in the SegNet architecture and, at inference time, apply dropout to obtain Monte Carlo samples for the pixel-wise class probabilities. The mean and variance of the these samples gives us the Maximum a Posteriori solution and the uncertainty in that solution respectively. The architecture can be seen in figure 1.

This is the same as sampling weights from the posterior distribution over weights to approximate the posterior distribution over the softmax scores for each class. The minimization of training error is also linked to minimization of the KL-Divergence between the true distribution and an approximate variational distribution over the weights of the network.

2.1.3 DenseCRF

In literature, many post processing techniques are utilized to improve upon the segmentation results by looking at the image from a bottom-up level such as superpixels [3]. We used a special type of Conditional Random Field called the DenseCRF to post-process the segmentation mask obtained as the output of SegNet. Most CRFs use the standard Ising prior which encodes that the state at each pixel depends only on its neighbors. However, the DenseCRF or the Fully-Connected CRF assumes that the state at each pixel can be

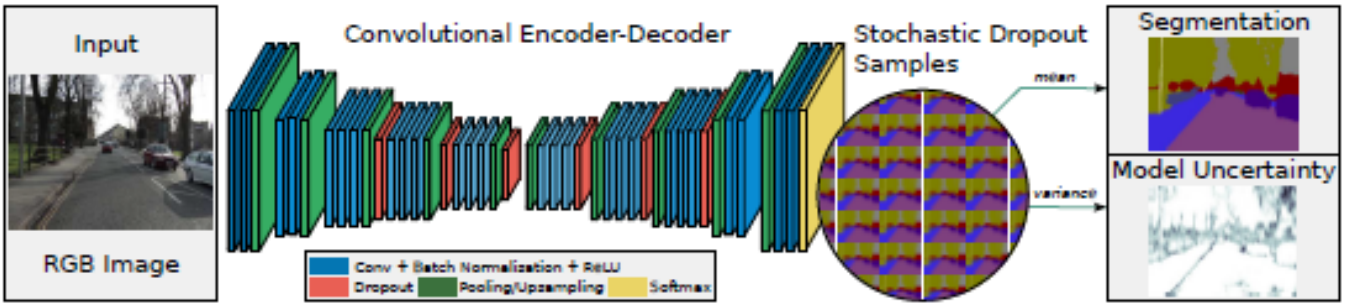

Fig. 1: Bayesian SegNet. Reprinted from Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding

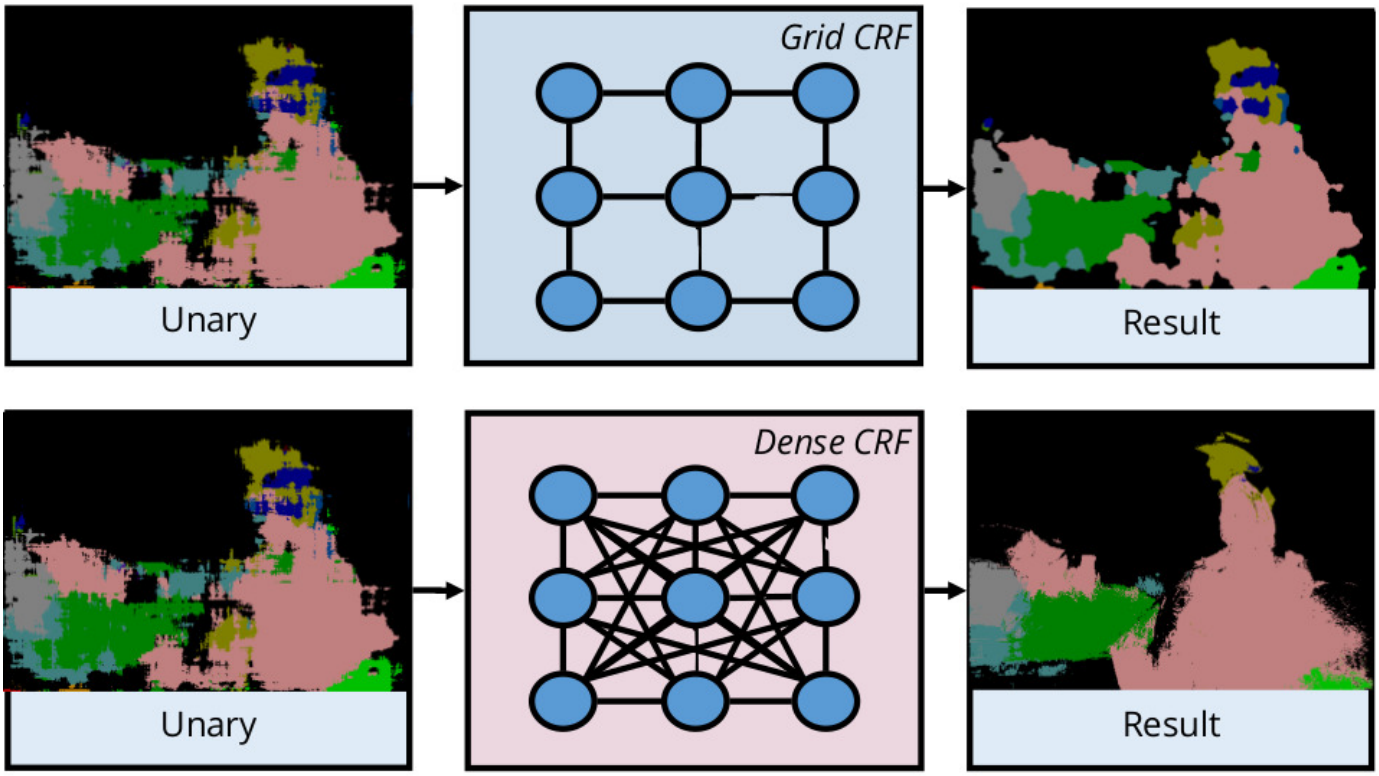

Fig. 2: Grid CRF vs Dense CRF. Reprinted from Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation

affected by any other pixel in the image. Thus, it is a much more expressive model. This can be seen in figure 2.

This however, makes inference much harder. There were two main approaches to inference in the literature:

- Piece-wise training [4]
- Joint estimation using marginal-based [5]

2.2 Instance Segmentation

2.2.1 Mask R-CNN

Mask R-CNN extends the Faster R-CNN architecture (which was used for object detection) to perform instance segmentation by adding a branch to the main backbone that predicts binary masks for each class. This decouples detection from segmentation. Mask R-CNN also mitigates the coarse spatial quantization problem by replacing ROI Pooling with ROI Align.

2.2.2 PsyPhy

PsyPhy is a psychophysics based vision evaluation framework originally used for recognition tasks. The key idea behind the evaluation is to perturb the input to a vision system by a small amount and observe the extent of degradation of its output. Some of the Perturbations used in the framework like Blur, contrast changes and brightness changes were highly relevant to robot navigation.

3 METHODS AND DATASETS

We experimented with training the segnet architecture on the different datasets with different loss functions. We trained the segnet architecture with all the weights initialized randomly and found that it was only able to perform excellent on the training dataset and didn't learn to make sensible inference on the validation or the test dataset. We then initialized the encoder architecture with the trained vgg16 weights and trained the segnet architecture on the CamVid11 dataset. We used the CamVid dataset for the experiments on Semantic segmentation and the MS COCO dataset for the experiments on Instance Segmentation.

3.1 Datasets

CamVid11
The camvid dataset is the Cambridge-driving Labeled Video Database consisting of pixelwise annotation for the images. The dataset consists of the road scenarios and is temporal in nature. We trained the segnet architecture on the camvid11 dataset which is the shortened camvid dataset. The camvid11 dataset contains the 11 classes: (1: building, 2: pole, 3: road, 4: sidewalk, 5: Tree, 6: SignSymbol, 7: Fence, 8: Car, 9: Pedestrian, 10:Bicyclist, 11:Void) The camvid11 dataset contains 367 training images, 101 validation images and 233 test images along with their segmentation masks(pixelwise annotations).

3.2 Semantic Segmentation

We implemented the core SegNet architecture using the PyTorch framework with the encoder initialized with the vgg16 weights. The network is trained on camvid11 dataset using the loss functions described below. We used two separate machines for training the architectures having NVIDIA GeForce GTX 1060 GPU with 6GB memory and 16 GB ram for about 27 hours.

3.2.1 Soft Dice Loss

If p is the output probability vector and g is the ground truth vector then,

$$SoftDiceLoss(p, g) = \frac{2\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2}$$

Cross-Entropy over pixels : Class-prediction for each pixel. This is problematic because of class imbalance. The optimization gets trapped in local minima. Frequency-based Weighted Cross-Entropy over pixels : Assign weights for each class given by 1/Fi. Where Fi is the frequency of pixels belonging to that class.

Soft-dice loss function : The main reason that people try to use dice coefficient or IoU directly is that the actual goal is maximization of those metrics, and cross-entropy is just a proxy which is easier to maximize using back-propagation [6]. In addition, Dice coefficient performs better at class imbalanced problems by design. No need to assign frequencies. For Dice vs wt cross entropy refer [7].

3.2.2 Weighted Cross Entropy

In the weighted cross entropy loss we assign weights to the classes in such a way that the less frequent class is given the higher weight and vice versa. The cross entropy loss is the standard loss function used for the classification problem.

$$WeightedCrossEntropyLoss(p, g) = -2\sum_{i=1}^{N} w_i g_i log(p_i)$$

where p_i is the probability of the ith pixel and g_i is the ground truth label associated with that pixel. We use the weights as suggested by the author in the paper.

3.3 Instance Segmentation

While we weren't able to implement and train the entire Mask R-CNN architecture, we evaluated it using the PsyPhy framework. Specifically, we used the Gaussian Blur, Contrast changes and Brightness changes in the input image to observe the extent of degradation in the output mAP.

3.4 Data collection

No datasets are available for underwater image classification and segmentation tasks. We are creating a new dataset by scraping underwater object images from Google and manually classifying the data in different classes. We used VoTT (Visual Object Tagging Tool) released by Microsoft for tagging and creating the bounding boxes manually. Our work is still in progress, and we have collected 100 images so far. Please refer Fig 3. for some samples.

3.5 Segmentation on video

We also experimented segmentation on the video by applying segmentation on the image frames extracted from the video. We applied post processing on these segmentation masks which improved the segmentation quality of the video as a whole.

4 RESULTS

In figure 4, we can see that soft dice loss performs considerably better than cross entropy loss function and the training time is also shorter for soft dice loss. We also show that the post processing steps on the semantic segmentation masks boost the accuracy.

Table 1 shows the results of evaluation of the variants of the SegNet model on the task of semantic segmentation. We used the following metrics in the evaluation:

- Mean IoU : The mean ratio of intersection over union of predicted segmentation mask and the ground truth segmentation mask.
- Overall Accuracy : Total number of pixels correctly classified over total number of pixels.
- Mean Accuracy : The mean of the class-wise accuracy.
- Frequency Weighted Accuracy : The weighted mean of class-wise accuracy such that the weight is the inverse of the class frequency.

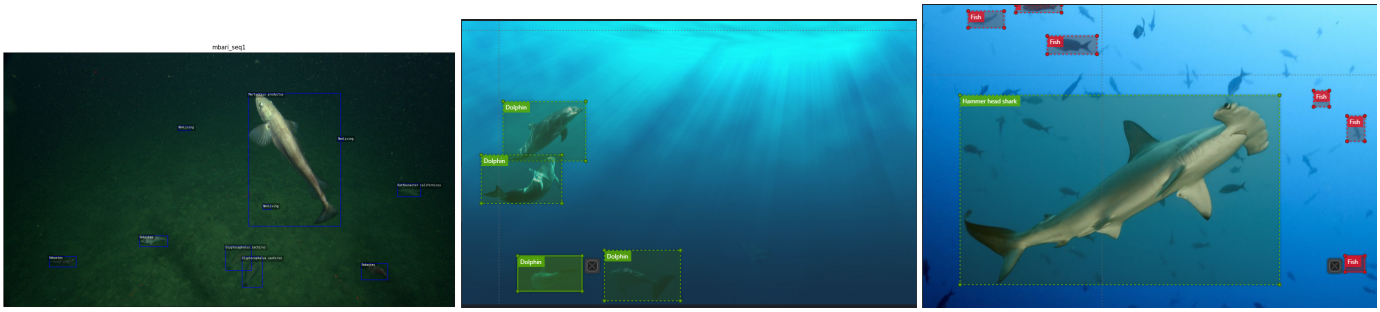

Fig. 3: 1. Underwater images from [Quote]; 2 and 3. Collected by our team

TABLE 1: Model Evaluation

	Mean IOU	Overall Accuracy	Mean Accuracy	Frequency Weighted Accuracy
Cross Entropy	0.5244	0.8312	0.7076	0.7512
Soft Dice	0.5818	0.9135	0.6767	0.8508
Bayesian	0.6023	0.9234	0.6809	0.8472

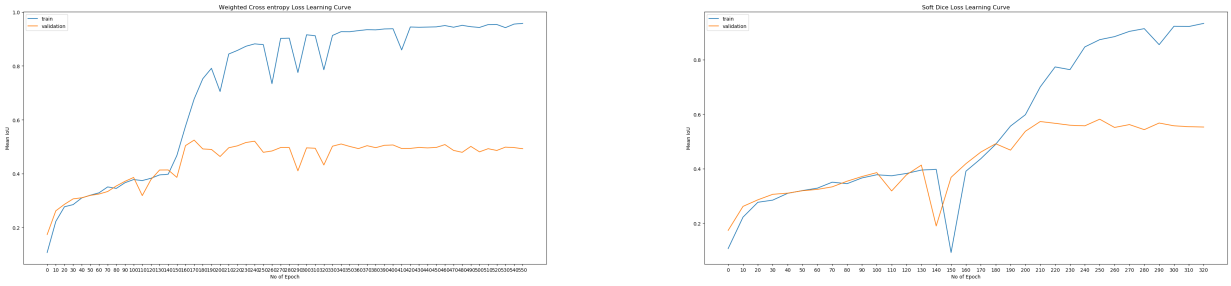

Fig. 4: Learning Curves - Weighted Cross Entropy Loss(left) and Dice Loss(right)

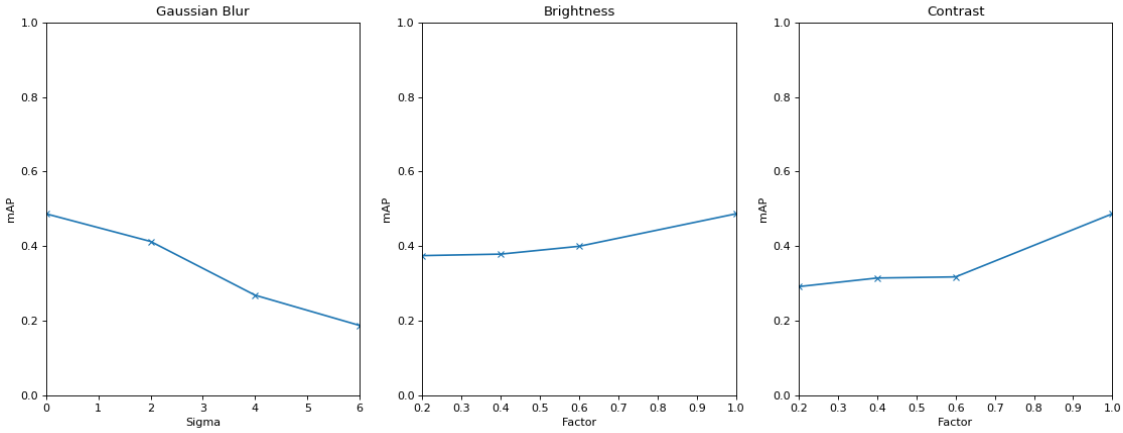

Fig. 5: Degradation of instance segmentation performance of Mask R-CNN with change in amount of Gaussian blur (left), brightness (middle) and, contrast(right).

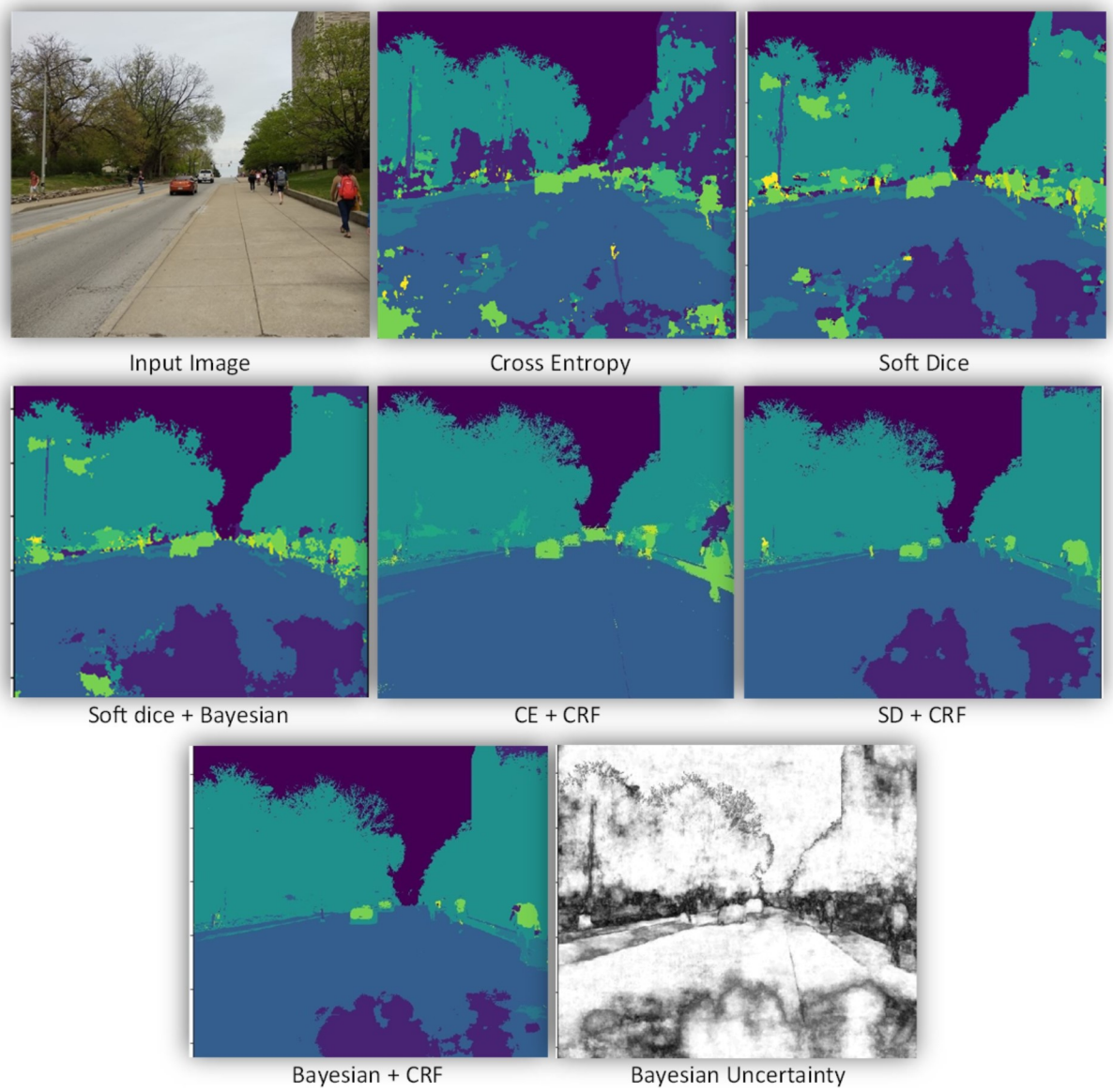

Fig. 6: Segmentation results on IU campus-street images.

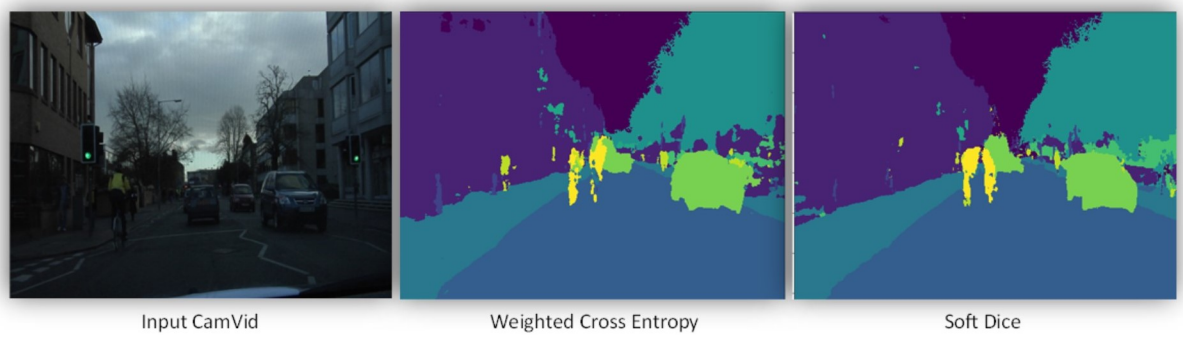

Fig. 7: Segmentation results on CamVid-11.

We see a 8 percent increase in mean IoU when we use Bayesian network and 6 percent increase when we have use Dice Loss as compared to when we use cross entropy loss.

.

Figure 5 depicts the evaluation of the Mask R-CNN model [8]

The model was trained on the MS COCO dataset. The plots show the degradation in performance when the input image is modified. Our experiments were done for 10 images from the test set of the COCO dataset. The mAP scores are averaged over those 10 images. The plot shows that while brightness does not affect the performance of Mask R-CNN much, small changes in Contrast and Blur cause the performance to degrade a lot.

5 Discussion

We can see from figure 6 that the semantic segmentation masks detected by the network are better when the network is trained with the soft dice function. The Bayesian segnet, which acts as an ensemble architecture has better and smoother results. We can also observe that the CRF smoothening on the top of the segmentation mask removes false detections. The DenseCRF ensures that the predicted segmentation mask agrees with the original input image using the unary potentials. It also ensures that the class predictions at each pixel agree with the predictions for every other pixel in the image using pairwise potentials.

The soft dice loss function tries to optimize the Intersection over Union(IoU) for the segmentation and hence intuitively performs better as we have observed from the numbers in the table 1 as well as the figure 6. The Bayesian segnet gives the uncertainty in terms of the soft probabilities which can be useful for the robot to make a more informed decision for navigation. We can observe these uncertainty values on the bottom-right of figure 6. The white pixels correspond to the higher probability values(or less uncertain values) and the black pixels correspond to the higher uncertainty values. We can also observe that the pixels associated with the grass in the original image has higher uncertainty values. This can be attributed to the fact that the 'grass' class is not one of the classes for which the network was trained on.

6 Conclusion and future work

We verified the accuracy of semantic segmentation on CamVid11 dataset by using different loss functions and found that the soft dice loss function performed the best in terms of accuracy. We also re-implemented Bayesian SegNet(which gives uncertainty) and found that it gives better accuracy compared to the SegNet.

We would like to continue collecting the under water image dataset in future and implement the semantic segmentation task for this dataset.

We would also like to use some kind of temporal smoothing(optical flow for example) to smooth the results as the network is trained on the video dataset which is temporal in nature. Having said that we would also like to compress the network using the student-teacher method so as to make the real time segmentation inference on the image frames of the videos.

Acknowledgments

The authors would like to thank Md. Alimoor Reza for extending his support and guiding us by giving his valuable advice.

References

[1] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *CoRR*, vol. abs/1505.07293, 2015. [Online]. Available: http://arxiv.org/abs/1505.07293

[2] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *CoRR*, vol. abs/1511.02680, 2015. [Online]. Available: http://arxiv.org/abs/1511.02680

[3] B. Hariharan, P. Arbelaez, R. B. Girshick, and J. Malik, "Simultaneous detection and segmentation," *CoRR*, vol. abs/1407.1808, 2014. [Online]. Available: http://arxiv.org/abs/1407.1808

[4] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *CoRR*, vol. abs/1210.5644, 2012. [Online]. Available: http://arxiv.org/abs/1210.5644

[5] A. Arnab, S. Zheng, S. Jayasumana, B. Romera-Paredes, M. Larsson, A. Kirillov, B. Savchynskyy, C. Rother, F. Kahl, and P. H.S. Torr, "Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction," *IEEE Signal Processing Magazine*, vol. 35, pp. 37–52, 01 2018.

[6] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *CoRR*, vol. abs/1606.04797, 2016. [Online]. Available: http://arxiv.org/abs/1606.04797

[7] "StackOverflow coefficient-loss-function-vs-cross-entropy," https://stats.stackexchange.com/questions/321460/dice-coefficient-loss-function-vs-cross-entropy, accessed: 2010-09-30.

[8] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[9] P. Kraehenbuehl and V. Koltun, "Parameter learning and convergent inference for dense random fields," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 513–521. [Online]. Available: http://proceedings.mlr.press/v28/kraehenbuehl13.html

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[11] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recogn. Lett.*, vol. 30, no. 2, pp. 88–97, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2008.04.005

[12] "Microsoft VoTT visual object tagging tool: An electron app for building end to end object detection models from images and videos." https://github.com/Microsoft/VoTTvott-visual-object-tagging-tool, accessed: 2010-09-30.

[13] "Matterport Inc." https://github.com/matterport/Mask_RCN.

[2] [1] [8] [4] [9] [5] [10] [11] [6] [12] [7] [13] [3]