Semantic Segmentation for Road Scene Understanding

Rishab Goel

***Abstract*— Semantic Segmentation is the task of recognizing and delineating objects in an image which categorizes each pixel in image. The semantic segmentation found its applications in assisting partially sighted people, path finding for robot, in medical analysis for tumor or cavity detection and in road scene understanding for autonomous vehicles. We would be focusing on challenges to solve the road scene understanding problems. We evaluate SegNet [5,6,7] architecture variants in our project using CamVid [9] dataset of 233 annotated test images . We focus on evaluation of SegNet's pre-trained models available in public domain and closely follow the tutorials to reproduce the claimed results. We aim to analyze the merits, demerits, evaluation criteria, methodology as well as datasets used for testing the pre-trained models. We closely look into discrepancies and inconsistencies in the models, datasets or scripts use for evaluation. The generated results are analyzed for memory requirements, inference time, statistical metrics, and visual quality of segmentation and classification and compared with the actual model.**

***Index Terms*—Semantic Segmentation, Autonomous Vehicles, CamVid , SegNet**

I. Introduction

SEMANTIC segmentation [Fig. 1] is basically a task of clustering of the portions of images into the classes each portions belong to. The usage of the semantic segmentation are very broad, and can be used for characterizing road signs and path mapping to medical analysis for tumor or cavity detection. Our focus is mainly on scene understanding for autonomous vehicles which regions of segmentation like street, traffic lights, cars, bicycle, sidewalks or pedestrians.

Martin [3] in his survey details the nuts and bolts of the problem of semantic segmentation. The paper lists down the quality metrics for the semantic segmentation implementations as accuracy (*mean accuracy, mean intersection over union, and frequency weighted intersection over union),* stability, speed and memory usage. A typical semantic segmentation does preprocessing and feature extraction on the dataset, augments the data follows it by window extraction and its classification culminating with post processing for the training phase. The prediction pipeline eliminates any data augmentation from the training pipeline while keeping other stages intact. The traditional approaches for the semantic segmentation using pixel color, Histogram of gradients and SIFT as features extractors paired with a bag of visual words like classifier. Unsupervised segmentation methods like clustering using k-means, minimum graph cuts, active contour models and watershed segmentation have also proved petty effective of differentiating the regions in an image to an extent. The machine learning models of ensemble learning (random decision forests) and Support Vector Machines have proved effective for training the feature extractors discussed, while graphical models like Markov Random fields and Continuous random fields have been used for tuning and increasing the accuracy. The deep learning networks using the convolution layer and pooling layer at their hearts become popular with the success of AlexNet[1] in the ImageNet 2012 challenge. Textons, a minimal building block in computer vision, like edges are believed to be the features learned by the first filter layer of ConvNets. Many other semantic segmentation methods like by Pinheiro which suggests a recurrent convolutional network. The recurrent network could be also be used for tuning the segmentation as Markov Random Fields or Continuous Random Fields. The segmentation of the samples in datasets as well as in real time prediction may suffer from issues like lens flare, vignetting, blurring, opaque and semi-transparent occlusions as well as varied viewpoints. The segmentation algorithm should be robust to such issues while training and also should be capable of mitigating such issues.

Fig. 1. First one is the original image and second one is the image with semantic segmentation. [12]

.

Our project focuses on SegNet architecture [5,6,7] derived from FCN [2] and VGG16 [4] for semantic segmentation for road scenes. We discusses the pros and cons of architecture variants suggested by the author, and CamVid Dataset. We discuss how we actually evaluate the architecture ranging from repository, evaluation package, scripts, result generation and parameters for evaluation. We analyze the results for inference speed, statistical parameters and visual understanding. We give insights into results and other recent advancements to solve the problem, and then conclude the report.

II. ARCHITECTURE

A. *SegNet : A Encoder-Decoder Architecture*

SegNet-Basic [6] Fig. 2 is an architecture that is inspired from the VGG16 for classification [4] and Fully Convolutional Network (FCN) [2] for segmentation to achieve road scene segmentation. It is to be noted that for a road scene classes like road, pavement , cars, pedestrians occupy majority of pixels on the scene, while others classes like signals, bicycle may not occupy majority pixels. It is required that different classes are trained with different weights to have a well distinguished boundary among them and still preserving the spatial correlation between each of these classes. The varied weighting of classes is termed as class balancing and SegNet uses median frequency balancing for shape, size and orientation distinction and memory as well as computation

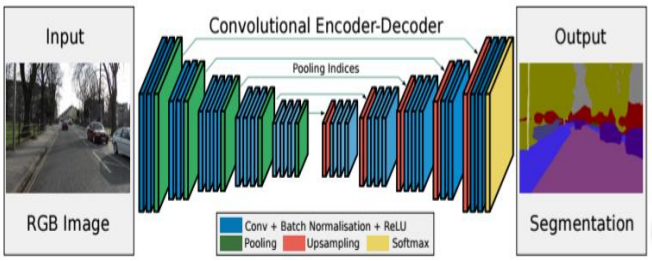


Fig. 2. An illustration of the SegNet Architecture with varied feature maps with decoder upsampling using transferred pooling indices.[8]

reduction.

Segnet-Basic have an encoder and decoder architecture like a stacked autoencoder inspired from FCN[2] and its deep network with 13 convolutional layers is basically VGG16 [4] without last three fully connected layer. The number of parameters gets reduced from 134M to 14.7 M by removing fully connected layer, thus reducing memory utilization. The initialization weights for VGG16 based network can be trained weights of VGG16 network and are used in training and evaluation in SegNet for faster and better convergence. The encoder network consists of convolutional layers (each followed by batch normalization and RELU) and 2x2 max-pooling layer while the decoder consists of convolutional layer (equivalent to deconvolution with batch normalization and RELU too) and upsampling layer.The network ends with SoftMax layer for probabilistic classification of different classes. The max pooling layer is used over average pooling layer in spite of a tendency to loose spatial resolution for two reasons: (i) to maintain translational invariance and (ii) to reduce the number of parameters. The max pooling layer in SegNet is modified to store pooling indices instead of max float value as the number of bits for 2x2 kernel reduces from 32/64 to 2 bits only and later convolved with trainable decoder filter banks. FCN upsampling layer generates dense feature maps instead of sparse maps in a different way. It preserves the encoder feature map in dimensionally reduced form (64 channels convolved with 1x1x64xK filters to reduce to K feature maps; where K is the number of classes). These

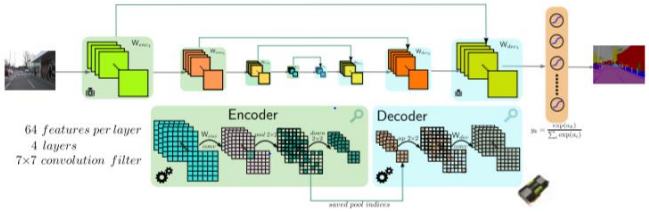


Fig. 3. An illustration of the SegNet Architecture with flat 64 feature maps with decoder upsampling using transferred pooling indices without RELU activation.[7]

encoder feature maps are added to trainable deconvolution based upsampling on the input feature map for the respective decoder. SegNet upsampling reduces memory requirement further by not storing encoder feature maps. It is to be noted that all the layers of Segnet are differentiable by the chain rule and could be trained end to end which makes it faster than networks with separately trained components like CRF (Continuous Random Fields).

The author [6] evaluates multiple decoder variants with a shallow network of 4 encoders and 4 decoders and extend the best one to a deeper network based on VGG16 with 13 encoders and 13 decoders. The decoder variants are discussed here:

a. SegNet-Basic: 4 encoders and 4 decoders with 7x7 constant kernel size for a wider context for smooth feature maps. It uses SegNet technique of not preserving encoder feature maps and preserving indices instead of values for max pooling layers.The number of channels in each decoder filter are equal to the number of feature maps.

b. FCN-Basic : Similar to SegNet-basic except for decoding technique.. It uses FCN technique of preserving encoder feature maps doing deconvolution at the decoder stage.

c. SegNet Basic-Channel Decoder: Similar to SegNet basic, but here the decoder filters are only single channel, which reduces the memory parameters further.

d. FCN Basic No addition : Similar to FCN-Basic , but discards encoder feature maps and only learns upsampling layers.

e. Others include Upsampling with Bilinear Interpolation , SegNet basic with Encoder Addition and FCN basic with No Dimensionality Reduction.

The architecture variant that performs best is the VGG16 based 13 layer architecture with decoder type same as SegNet Basic.

Segnet [7], Fig 3, is a modification of SegNet Basic for robust pixel wise labelling. SegNet decoder don't have a RELU activation for easier optimization of filters in each pair. Encoder and decoder filters are united to increase the degrees of freedom. The SegNet is flat architecture which uses fixed 64 feature maps in each layer to avoid parameter explosion unlike an expanding feature map. The training time for each layer remains similar which otherwise increases with the depth of the layer. The last decoder has the capability to expand into multiple channels 3 or 4 (RGB or RBBD). An additional layer is added at the input for local contrast normalization to remove

non-uniform illumination and enhance edges for better distinction. The additional layer also decorrelates the input dimensions which leads to better convergence. To avoid false channel edges it is applied to each channel (RGB or RGBD) separately. This version of SegNet was also evaluated for 4 layer as well as 13 layer variants.

B. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder

The Bayesian SegNet [5] is a derivation that introduces a measure of model uncertainty which better distinction of object boundaries and identification of complex visual objects. The Bayesian SegNet aims to form a probabilistic encoder-decoder architecture using dropout layer for probabilistic inference. The aim is to get posterior distribution of convolutional layer weights given our observed training data and labels. A posterior distribution of pixel class labels is generated by using drop out at train time and Monte Carlo sampling with dropout at test time to achieve a measure of model uncertainty. The variance of the predicted segmentation is used to generate a measure of uncertainty. The variants of architecture were tried with dropout layer were as follows: Dropout Encoder, Dropout Decoder, Dropout Enc-Dec, Dropout Center (only the deepest encoder and decoder with dropout), Dropout Center Enc-Dec (two deepest encoder and decoder with dropout) and Dropout Classifier (dropout just before classifier).

Fig. 4a. A road scene from CamVid Dataset and Fig 4b. Contrast Enhanced Ground truth label

III. DATASETS

The input for semantic segmentation is an input image as show in **Fig** with a car, road, pedestrians, sky, building, bicycle etc. The output label would be segmented annotated image with each pixel labelled with the class value. Suppose they are 11 classes, the pixel belonging to class 1 has value 0 , for class 2 has value 1, for class 3 has value 3 and so on. They are few popular dataset for semantic segmentation.

a. PASCAL-VOC12 dataset [16]: It has 20 classes with person, animals, vehicles and indoor objects with train/validation data of 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations.

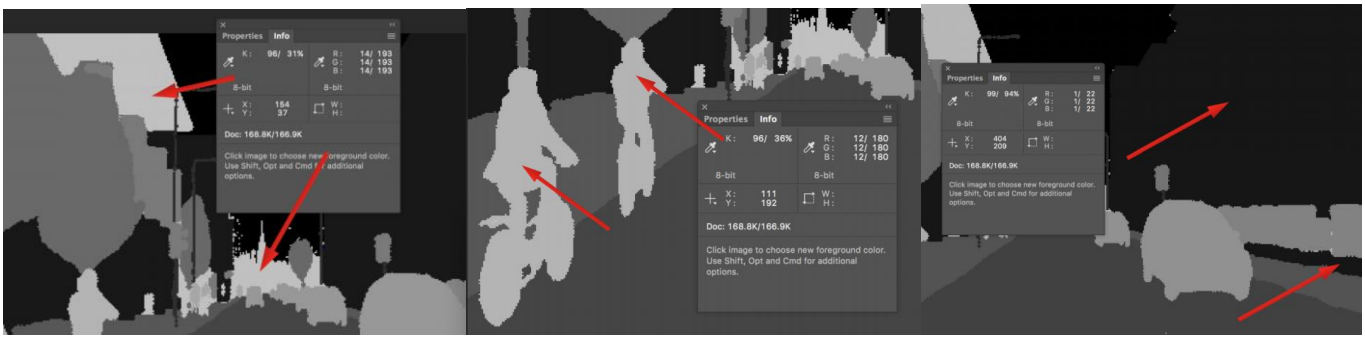
b. CamVid Dataset [9]: 32 semantic classes of moving objects like car, bicycle, pedestrian ; road, sky and buildings. It is mentioned the 701 labelled images, but on the sites they are 5 videos of road scenes with each having less than 7000 frames.

c. CityScapes Dataset [10]: A large scale Stereo Dataset with 30 classes with diverse scenes of 50 cities over different months, daytime and weather conditions. 5000 fine annotated images and 20000 coarse annotated images are available in the dataset.

d. SUN-RGBD dataset [18]: 32 classes inIndoor scenes with 10000 RGBD images densely annotated with 146617 2D polygons and 58657 bounding boxes along 3D layout and category of scenes

e. Kitti road dataset [20,21]: Stereo Image dataset of road scenes of 39.3 km length drive with 200k 3D object annotations with upto 15 cars and 30 pedestrians visible per image

CamVid [9], CityScapes Dataset [10] and Kitti road Dataset [20.21] are good road scene understanding dataset. Kitti is new, CamVid the older one and CityScapes more diverse and descriptive. CityScapes is considered a better dataset because of its diversity. It is used in recent interesting paper of ENet [8] and Speeding Up Semantic Segmentation [15], even a model trained on SegNet architecture is shared but the detail is hared on github. We use CamVid Dataset for our evaluation due to two reasons: (i) SegNet originally evaluated on CamVid dataset, and (ii) multiple SegNet trained model variants are available for CamVid dataset. SegNet is basically trained on CamVid Dataset with 367 train, 101 validation and 233 test images with 11 classes with 360x480x3 (RGB) resolution. The most trained model provided for SegNet different architecture variants is trained on 11 classes , except the webdemo model, which is trained on 12 classes and on additional manually



(a) **(b)** **(c)**

Fig 5 . Bugs in CamVid Dataset[13]

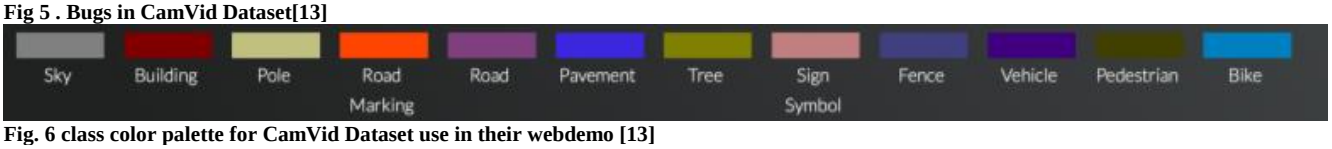


Fig. 6 class color palette for CamVid Dataset use in their webdemo [13]

annotated images. The extra class included is Road Marking The color palette for 12 classes is shown in Fig 6.

The CamVid Dataset has some buggy annotations as evaluated in [13] are shown in Fig 5 with (a) area value 14 in spite the classes being only 11, (b) value at bike is 12 instead of 11 and (c) value of tree is 4 it should be 6. It is to be noted that large scale fine annotations of dataset may suffer some issues when manually annotated or even annotated using tools. These kinds of bugs make the statistical analysis slightly doubtful, so we observe out statistical analysis only as a comparative result not absolute result.

IV. Architecture Implementation and Analysis

A. Implementation

SegNet [5,6,7] architecture (with all its variants) is implemented in a customized repository of Caffe shared on github for training evaluation. Caffe [22] is one of fastest C++ deep learning framework (ConvNets framework) by single or multi-GPU acceleration through **CuDNN** support. The blocks in a convolutional network are defined as layers in Caffe and model definition is done using .prototxt file. Caffe provides python interface to gather, preprocess and post process data and can execute the model definition for training, testing and inference. The authors provides a full training tutorial and also basically four pre-trained models on CamVid dataset for inference. It is to be noted that the machine accessible to me as a small GPU Nvidia GeForce 940M with 4GB GPU memory, while the network was trained on higher end Nvidia Titan GPU with 6 GB memory. This constraint restricts me to train any new or given model, we restrict our evaluation to the trained models available. I use the SegNet Caffe version repository [11] which is supported with CuDNN5 and ahead. We evaluate these models for inference speed and for visual and statistical analysis.

Pre-trained models from repository [12] i.e. *SegNet-Tutorial-master* under *Models/* over CamVid dataset available are as follows:

a. *Segnet_basic_camvid.prototxt*: It is a shallow architecture based on flat Segnet [7] architecture with 4 encoders and 4 decoders each having 64 feature with no RELU in decoders.

b. *Bayesian_segnet_basic_camvid.prototxt*: This is similar to *Segnet_basic_camvid.prototxt* has dropout layers with sample weights test in last two encoders and decoders with batch size 2 or higher. We prepare one more variant named *Bayesian_segnet_basic_camvidw.prototxt* by disabling sample weights and reducing batch size to 1. This makes it closer to SegNet-Basic CamVid with just two additional dropout layers.

c. *Bayesian_segnet_camvid.prototxt*: This is similar to SegNet model with 13 encoders and decoders has dropout layers with sample weights test in last two encoders and decoders with batch size 2 or higher. We prepare one more variant named *Bayesian_segnet_basic_camvid.prototxt* by disabling sample weights and reducing batch size to 1. This makes it closer to SegNet CamVid with just two additional dropout layers.

d. SegNet_model_driving_webdemo: This model is 13 layer variant of SegNet_Basic_CamVid (a) model mentioned above, and trained for 12 classes instead of 11 like all other model. The extra-training class is road marking. It is to be noted that the SegNet tutorial [12] mentions that this same model is used for their web demo which looks real time. This model uses Xavier weight filler ($var(w) = 1/n_{input}$), instead of Gaussian or MSRA ($var(w)= 1/(n_{input}+ n_{ouput})$) as RELU is absent in decoder layer.

The output gifs of all six was added to folders along with groundtruth, two uncertainty gifs and input gif.[24].

The input is a CamVid test image as shown in Fig 4.a and label is class value to which each pixel belongs to. The output generated by Bayesian Segnet based is a probability matrix of 11x360x480 dimensions. The probability would be highest for highly probable class. The SegNet architecture based generates the argmax of the probability among the class, which means the class label for most probable class. The SegNet tutorial guides provides steps with python scripts for testing and training SegNet model. We modified *webcam_demo.py* as well as *test_bayesian_segnet.py* provided in the *Scripts/* folder of SegNet Tutorial evaluation suite for analyzing the model definition mentioned before.

B. Visual and Statistical Analysis Parameters

The model is evaluated on mainly on three statistical parameters namely:

a. *Global accuracy (G)*: percentage of pixels currently classified over total number of pixels

b. *Class accuracy (C)*: Mean accuracy over all the classes.

c. *Mean Intersection per union* (I/U) the percentage of true positive over the sum of true positive, false positive and false negative.

				Median frequency balancing							Natural frequency balancing						
				Test				Train			Test				Train		
Variant	Params (M)	Storage multiplier	Infer time (ms)	G	C	mIoU	BF	G	C	mIoU	G	C	mIoU	BF	G	C	mIoU
Fixed upsampling																	
Bilinear-Interpolation	0.625	0	24.2	77.9	61.1	43.3	20.83	89.1	90.2	82.7	82.7	52.5	43.8	23.08	93.5	74.1	59.9
Upsampling using max-pooling indices																	
SegNet-Basic	1.425	1	52.6	82.7	62.0	47.7	35.78	94.7	96. 2	92.7	84.0	54.6	46.3	36.67	96.1	83.9	73.3
SegNet-Basic-EncoderAddition	1.425	64	53.0	83.4	**63.6**	48.5	35.92	94.3	95.8	92.0	**84.2**	56.5	**47.7**	36.27	95.3	80.9	68.9
SegNet-Basic-SingleChannelDecoder	0.625	1	33.1	81.2	60.7	46.1	31.62	93.2	94.8	90.3	83.5	53.9	45.2	32.45	92.6	68.4	52.8
Learning to upsample (bilinear initialisation)																	
FCN-Basic	0.65	11	24.2	81.7	62.4	47.3	**38.11**	92.8	93.6	88.1	83.9	55.6	45.0	**37.33**	92.0	66.8	50.7
FCN-Basic-NoAddition	0.65	n/a	23.8	80.5	58.6	44.1	31.96	92.5	93.0	87.2	82.3	53.9	44.2	29.43	93.1	72.8	57.6
FCN-Basic-NoDimReduction	1.625	64	44.8	**84.1**	63.4	**50.1**	37.37	**95.1**	**96.5**	**93.2**	83.5	**57.3**	47.0	37.13	**97.2**	**91.7**	**84.8**
FCN-Basic-NoAddition-NoDimReduction	1.625	0	43.9	80.5	61.6	45.9	30.47	92.5	94.6	89.9	83.7	54.8	45.5	33.17	95.0	80.2	67.8

Fig. 7. The performance of Architecture variants of SegNet-Basic on the CamVid dataset with three metrics; global accuracy (G), class average accuracy (C) and intersection over union (I/U). [6]

To analyze above statistical metrics the *Scripts/* folder of *SegNet-Tutorial-master/* contain a matlab script named *compute_test_results.m* The Bayesian Segnet generates another output which is uncertainty across the image which helps for better segmented class model in the final scene. The image generated is used to generate average uncertainty output.

The results from the SegNet [6] implementation are evaluated in Fig. 7 for the various decoder variants. SegNet-Basic is slower than FCN-Basic is inference almost half the speed, but memory requirement is 11 times less as SegNet doesn't preserve any encoder weights. The FCN-Basic in fact achieves real time implementation, as the inference time is less than 33 ms for 30 fps performance. It is to be noted that real time performance is on a high-end GPU Nvidia TX GPU. The table shows that larger decoder better performance in quality, preserving encoder feature maps reduces inference time and, if memory is constrained to get a better inference time pass a dimensionally reduced encoder maps. The results from the paper of Bayesian SegNet [9], shown in Fig. 8**,** that while doing the statistical analysis by Global accuracy, class average accuracy and mean intersection over union. Dropout Enc-Dec performs badly in training itself, and thus could be considered too strong regularizer. Dropout Central Enc-Dec performs well in class average accuracy and mean intersection over union, while Dropout Classifier performs better in global accuracy.

	Weight Averaging			Monte Carlo Sampling			Training Fit		
Probabilistic Variants	G	C	I/U	G	C	I/U	G	C	I/U
No Dropout	82.9	62.4	46.4	n/a	n/a	n/a	94.7	**96.2**	**92.7**
Dropout Encoder	80.6	68.9	53.4	81.6	69.4	54.0	90.6	92.5	86.3
Dropout Decoder	82.4	64.5	48.8	82.6	62.4	46.1	94.6	96.0	92.4
Dropout Enc-Dec	79.9	69.0	54.2	79.8	68.8	54.0	88.9	89.0	80.6
Dropout Central Enc-Dec	81.1	**70.6**	**55.7**	81.6	**70.6**	**55.8**	90.4	92.3	85.9
Dropout Center	82.9	68.9	53.1	82.7	68.9	53.2	93.3	95.4	91.2
Dropout Classifier	**84.2**	62.6	46.9	**84.2**	62.6	46.8	**94.9**	96.0	92.3

Fig. 8. The performance of Architecture variants of Bayesian SegNet-Basic on the CamVid dataset with three metrics; global accuracy (G), class average accuracy (C) and intersection over union (I/U). [5]

Thus, from the results Dropout Central Enc-Dec is suggested to reasonably good architecture.

We create gifs of the outputs each prototxt

V. Results and Discussion

A. Evaluation Suite Description

The folder structure of the Evaluation suite is as follows:
EvaluationSuite/

test_camvid.sh- This shell script executes batches of runs on scripts *webcam_demo.py* and *test_bayesian_segnet.p*y.

SegNet-Tutorial-master/- This repository is obtained from the repository shared in SegNet tutorial [12].
The folder structure of *SegNet-Tutorial-master/* goes as follows:

**.log* – These files contain the inference performance results of each evaluated model definition on Nvidia GeForce 940M 4GB GPU memory with Intel i5 8GB RAM.

Camvid/- This contains CamVid dataset folders *train/*, *test/* ,*val/* , *trainannot/*, *testannot/* ,*valannot/* along with *train.txt* ,*test.txt* ,and *val.txt.* Each text files have input and annotations absolute path mentioned.

Models/- All the .prototxt files of the six model definition, their respective outputs in respectively named folder. The test weights are part of Inferences folder with test weights (*.caffemodel*) of respective model definition.

Scripts/ - The color pallete for camvid dataset of 11 and 12 classes and sun dataset. The *compute_test_results.m* a matlab script for generating statistics results. The groundtruth and predictions file with class labels are required as input.

webcam_demo.py – The script only supported running webcam_demo evaluation for SegNet Web demo. We run this script on the test data of CamVid for both SegNet model definition *segnet_basic_camvid.prototxt* and *segnet_model_driving_webdemo.prototxt*. We have modified to save or show outputs using –save. We allow to save input image, groundtruth colored image, segmented class label image, segmented colored image and horizontal fusion of input image and segmented colored image.

test_bayesian_segnet.py – The script only supports running Bayesian SegNet based model evaluation due to preprocessing of the input image for non-uniform contrast normalization. We run this script for all Bayesian SegNet model definition *(.prototx*. We have modified to save or show outputs using –save and also added profile code. We allow to save input image, groundtruth colored image, segmented class label image, segmented colored image, average uncertainty image and horizontal fusion of input image, groundtruth and segmented colored image.

plotconfusionmatrix.py: Added to visualize confusion matrix to compare the predicted and ground truth class labels over the whole dataset.

B. Inference Speed and Quality Evaluation

I evaluated the pre-trained architecture provided for inference performance on the test set of CamVid Dataset on lower end Nvidia 940M GPU in Fig 9. It could be concluded that Bayesian Segnet are slower in inference runs compared to SegNet Basic as well as have a higher preprocess time. They claim that SegNetWebDemo model is fastest for an expectable quality but it is a lot far away from real time performance. This raise the question whether the architecture is suitable for autonomous vehicles which will have memory as well as resource constraint. We also observe that SegNet Basic speed doesn't matter as the quality is very bad. It could be concluded

	Read (ms	Preprocess (ms)	Run(ms)	Postprocess (ms)
Bayesian_segnet_basic_camvid	25.79	37.88	1938.7	73.11
Bayesian_segnet_basic_camvidw	26.8	37.77	629.52	41.69
Bayesian_segnet_camvidw	27.07	37.86	1256.3	41.3
Bayesian_segnet_camvid	27.01	38.96	5015	74
Segnet_basic	8.106	0.3628	509.683	8.17
SegnetwebDemo	11.03	0.49	1143.98	10.75

Fig. 9. Average Inference Performance of Model definition on Nvidia GeForce 940M 4GB GPU Memory

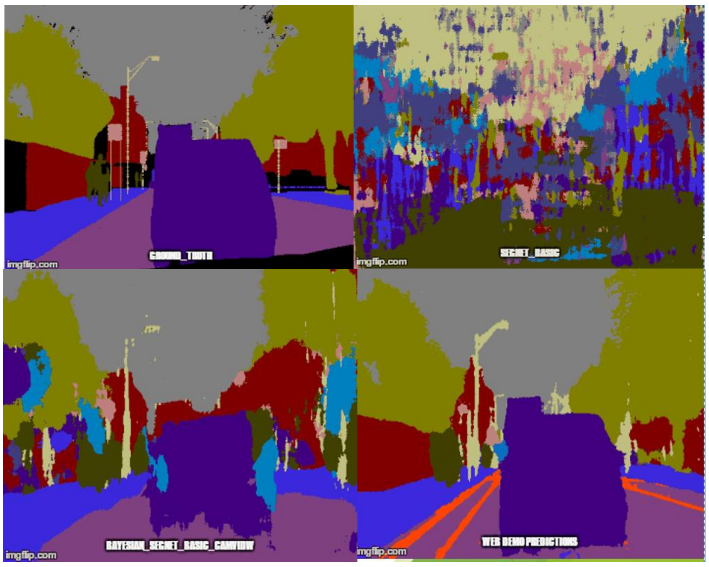
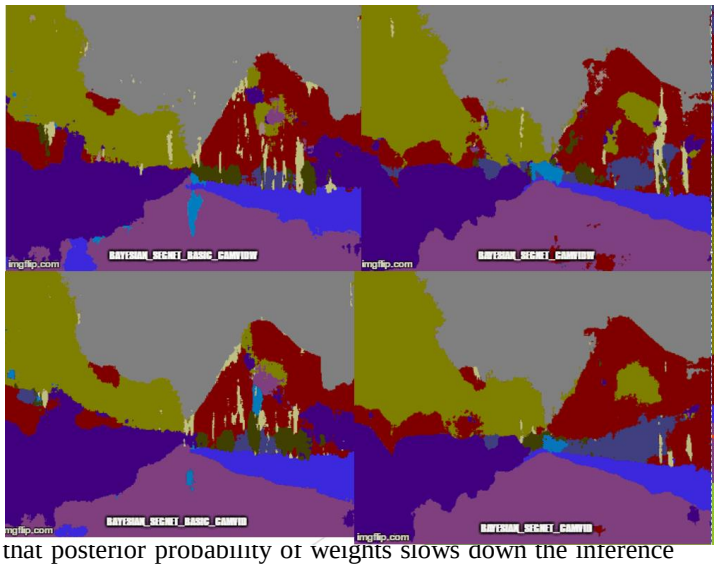


Fig.10. We have ground truth image here followed by compare segnet_basic ,segnet_webdemo, Bayesian_segnet_basic_camvidw (Simple Dropout)in clockwise fashion.

In Fig 11 below, we compare Bayesian_segnet_basic_camvidw (Simple Dropout), Bayesian_segnet_camvidw, (Simple Dropout)Bayesian_segnet_camvid (MonteCarlo Dropout), Bayesian_segnet_basic_camvid (MonteCarlo Dropout) in clockwise fashion



that posterior probability of weights slows down the inference time as low as 3X with similar quality performance.

We use pre-trained model for our evaluation, which are actually trained on small dataset of CamVid dataset with no mention of their training accuracy. Moreover, they also don't clearly mention whether results generated on the tutorial uses the provided models. It raises doubt that why thauthor doesn't want to share to demonstrate SegNet full capability. Bayesian SegNet model hovers around 50% for Global Accuracy, Class average accuracy of around 27% and mean intersection of union of around 17%. For SegNet-basic these parameters all statistical parameters were below 10%.The confusion matrix conveys the classes that remain static in the scenes perform much better than classes with moving objects for the evaluated model. We were not regenerate the statistical results due to the lack of good pre-trained models. But other architecture like ENet [8] were able to regenerate the results for newly trained SegNet and SegNet Basic Model and evaluate it for CamVid Dataset, CityScapes Dataset and SUN RGBD dataset. The results of per class accuracy, average class accuracy as well as mean intersection ofUnion in Fig 11 from ENet analysis [8] shows, that we can achieve the claimed quality performance of SegNet. These observations not only reflects that shared pre-trained models are badly trained, but also leads to questions whether the higher quality was achieved by overfitting.

C. Visual Analysis

Fig 10, we observe that SegNet_basic has smudges all around with no consistent boundary. SegNetWebDemo matches very closely to ground truth and it doesn't have false positives as bayesian_segnet_basic_camvidw. It preserves shape of objects like in dynamic scenes compared to others, but detoriates when lot of objects in the scene.False positives in Bayesian_segnet_basic_camvidw makes bicycle heads as a car. Bayesian_segnet_basic_camvidw performs much better than segnet_basic with addition of dropout in 2 deepest ENC-DEC.

In Fig 11, we observe that Segnet version preserve complex shapes and boundaries better than segnet_basic version. Monte Carlo Dropout architecture performs better boundary distinction and segmentation with complex shapes occur in the scene compared to respective architecture with simple dropout.Monte Carlo Dropout lose a lot on inference time compared to with simple dropout. For real time implementation simple dropout looks a better choice.Dropout helps to regularize the architecture and thus give better results.

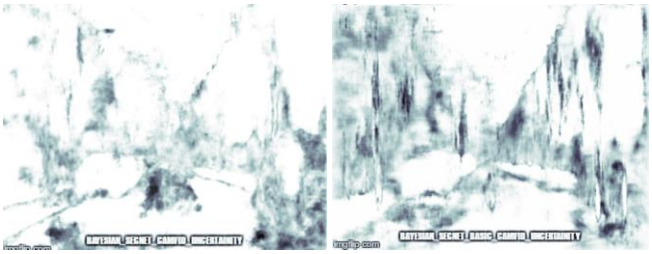


Fig. 12. Uncertainty of all classes in Bayesian_segnet_camvid and Bayesian_segnet_basic_camvid.

Fig 12, we observe that deeper network (segnet) has clearly lesser uncertainty than shallow (segnet_basic. Deeper network have more distinct boundaries than shallower one.

Uncertainty of classes with moving/vanishing objects like car/tree is more than static or large objects like sky or building.

VI. Insights and Future Work

SegNet is a simple end to end trainable stacked autoencoder based network which generates good quality results with single trainable path approach. Multi-path approaches with preserving end to end trainable property of the network will allow better segmentation of complex scenes, while preserving

the speed. It doesn't reach close to real time implementation on a high end GPU like Titan, show that inference kernel needs to be optimized further for speed. 11x memory usage increase is required to get 2X inference speed gain. Thus, more approaches like transferred max pool indices should get evolve to achieve real-time system performance. For classes associated with moving objects using uncertainty would lead to improved segmentation. SegNet does not handle scenes with clutter, but maintains good performance for simpler scenes and objects. Dropout layer increases generalization and increases quality without incurring much inference time. The choice of sharing ill-trained older weights in public domain is a bad move for adoption and spread of the architecture. New architecture comparing its design with SegNet has to also train SegNet from scratch no pre-trained models available for comparison. The choice of small dataset like CamVid of just close to 1000 images and claiming to achieve 80% Global accuracy and 60 % class accuracy seems a premature decision. The semantic segmentation is a complex tasks especially in dynamic scenes like urban road scenes, so the network should be trained much larger and much finely annotated dataset. The author itself proves that training on larger and finer annotated dataset results in a better quality segmentation as their web demo generates visually smoother segmentation and accurate classification.

We will look into Enet[8], which introduces multi-path bottleneck layer and Treml *et.al* [15] introduces fire module and parallel dilated convolution layer. Both networks have multi-path modules which are end to end trainable and claim to perform better in speed and memory utilization, while the quality performance in close proximity. In fact, they

Model	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	Class avg.	Class IoU
1	75.0	84.6	91.2	**82.7**	36.9	93.3	55.0	47.5	**44.8**	74.1	16.0	62.9	n/a
2	**88.8**	**87.3**	92.4	82.1	20.5	**97.2**	57.1	49.3	27.5	84.4	30.7	65.2	**55.6**
3	74.7	77.8	**95.1**	82.4	**51.0**	95.1	**67.2**	**51.7**	35.4	**86.7**	**34.1**	**68.3**	51.3

Fig. 13. ENet (3) comparison with SegNet-Basic(1) and SegNet(2) on CamVid Dataset.

outperform SegNet in some cases for statistical parameters as shown in Fig 11. It should be noted that Semantic Segmentation for road scenes is very dynamic problem. This requires the dataset used to train should be large, extensive and diverse with larger number of classes. The CityScapes dataset [10] is becoming popular for the aforementioned reasons only and in fact for ENet [8] and Treml *et.*al [15] it is there primary dataset for training and evaluation.

The dynamism of the road environment may still not be covered in a very diverse dataset or the network may not arrive for a reasonably good solution. This requires the learning network should have the capability to learn on the fly and to distinguish false positives and false negatives clearly. This suggests that network should have unsupervised learning module to mold itself on the fly. Generative adversarial network (GANs) introduced such an unsupervised learning concept which has proved beneficial for image modelling. It consists of two competitive functions; one being a Discriminator function, which distinguishes ground truth maps from the generator maps. The other being a Generator function, which learns to generate false feature maps to confuse discriminator. The adversarial training approach is evaluated by Paulina *et.al.* [23] over Stanford Background Dataset and PASCAL VOC 2012 dataset with enthusiastic results.

VII. Conclusion

It could be concluded that SegNet is an interesting encoder –decoder based architecture inspired from FCN [2] and VGG16 [4].It aims to reduce the memory requirements and inference time for the problem, instead of just focusing on quality improvements. It introduces some interesting concepts like removes FCC layers in VGG 16, transferred max pool indices, fixed number of feature maps and convolutional layer at decoder without activation. It only tends to accomplish its bits and pieces only, by compromising on memory vs speed in its implementation and unable to achieve real time inference performance on even a high-end GPU. Bayesian network without sample weights test give a balance between quality and inference time.

The lack of consistency and quality in maintaining the pre-trained models in terms of number of classes and training datasets by the author makes its evaluation very fussy, inconsistent. The author doesn't seems to be pushing on working from scratch for usage of the SegNet architecture and gives us a chance to question the reproducibility of its results .It could be concluded that SegNet was an interesting stepping stone for semantic scene segmentation for dynamic road scenes. New architectures like ENet [8], Treml *et.al* [15] and Pauline *et. al* [23] build and evolve using its findings.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS) 25, 2012.

[2] "Fully Convolutional Networks for Semantic Segmentation" by Jonathan Long∗ Evan Shelhamer∗ Trevor Darrell, CVPR 2015, arXiv:1411.4038

[3] "A Survey of Semantic Segmentation", by Martin Thoma, info@martin-thoma.de; arXiv:1602.06541v2 [cs.CV] 11 May 2016.

[4] "Very deep convolutional networks for large-scale image recognition," byK. Simonyan and A. Zisserman, arXiv preprint arXiv:1409.1556, 2014

[5] "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding." by Alex Kendall, Vijay Badrinarayanan and Roberto Cipolla arXiv preprint arXiv:1511.02680, 2015.

[6] "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." by Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." arXiv preprint arXiv:1511.00561, 2015.

[7] "A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling" by V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet. arXiv preprint arXiv:1505.07293, 2015

[8] "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation" by Adam Paszke, (Univ. of Warsaw, Poland) Abhishek Chaurasia, Sangpil Kim, Eugenio Culurciello (Purdue University) arXiv:1606.02147v1 [cs.CV] 7 Jun 2016.

[9] Semantic object classes in video: A high-definition ground truth database," G. Brostow, J. Fauqueur, and R. Cipolla, " PRL, vol. 30(2), pp. 88–97, 2009.
[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[11] https://github.com/TimoSaemann/caffe-segnet-cudnn5
[12] https://github.com/alexgkendall/SegNet-Tutorial
[13] http://5argon.info/portfolio/d/SegnetTrainingGuide.pdf
[14] Nonlinear image representation using divisive normalization. In CVPR, 2008
[15] "Speeding up Semantic Segmentation for Autonomous Driving" by Michael Treml, José Arjona-Medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael Widrich, Bernhard Nessler, Sepp Hochreiter; Institute of Bioinformatics, Johannes Kepler University Linz, Austria, Audi Electronics Venture GmbH, Germany.
[16] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
[17] S. Song, S. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite" in Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)
[18] P. H. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene parsing," arXiv preprint arXiv:1306.2795, 2013.
[19] https://wwwf.imperial.ac.uk/~gmontana/talks/crfasrnn_presentation.pdf
[20] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In CVPR, pages 3354–3361, 2012.
[21] G. Ros, S. Ramos, M. Granados, A. Bakhtiary, D. Vazquez, and A. Lopez. Vision-based offline-online perception paradigm for autonomous driving. In WACV, 2015
[22] "Caffe: Convolutional Architecture for Fast Feature Embedding" by Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor ,arXiv preprint arXiv:1408.5093 {2014}
[23] "Semantic Segmentation using Adversarial Networks " by Pauline Luc, Camille Couprie, Soumith Chintala, Jakob Verbeek; NIPS Workshop on Adversarial Training, Dec 2016, Barcelona, Spain; Cite as: arXiv:1611.08408
[24] https://imgflip.com/