EE735: Computer Vision
Assignment # 01 – Deep Depth From Focus

Part # 02 **Dated: 1st November 2019**

Step 1: Reproducing the Baseline

Deep depth from focus uses convolutional neural network (CNN) architecture to predict depth map given a stack of images with different optical focus. The network architecture comprises of an encoder and a decoder part. The decoder is a mirrored version of the encoder. It uses VGG-16 net model as a baseline for accessing the pre-trained weights and hence allowing the small dataset to fine-tune to this specific task. Thus, the network comprises of 13 convolutional layers and 5 poolings layers in each encoder and decoder part. The model concatenates layers of encoder with the decoder feature maps in order to preserve the sharp object boundaries.

Dataset: the authors used light-field camera to generate the dataset for the model. They used 10 images per stack for each depth map and training was done in frame-by-frame manner. Final scoring was done by 1x1 convolution through the stack of 10 feature maps.

Results: the original paper model was run for 200 epochs, however due to time constraints and unavailability of faster GPUs, in this assignment all the modified networks were trained for 10 epochs. In order to have fair comparison, baseline model was trained for 10 epochs as well.

Fig.1 below shows the result of two sample images. Compared to ground-truth depth map, the depth generated with baseline with 200 epochs have better results than the baseline with 10 epochs. This indicates the network needs to be run for more epochs in order to have better results. This applies for all the models defined later in this document.

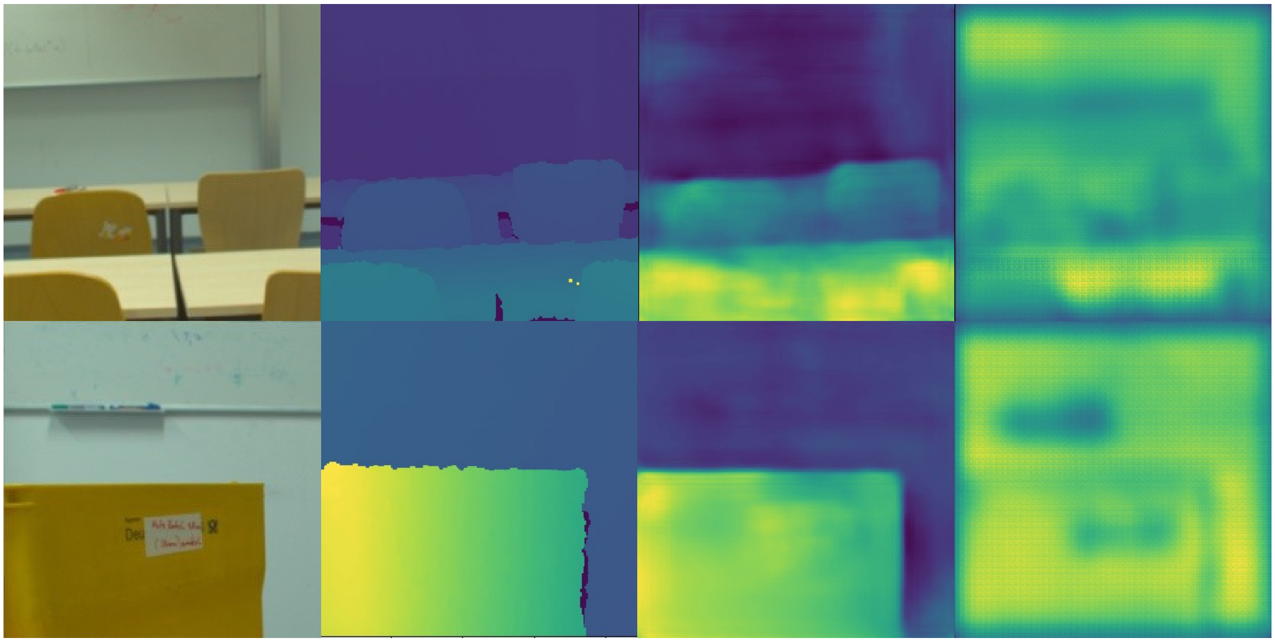

Image **ground-truth** **baseline-200-epochs** **baseline-10-epochs**

Fig.1

Step 2: Improving the Baseline by Learning Inter-Frame Information

This was performed using 3d-convolution across the frames instead of 2d-convolution frame-by-frame. The model architecture remained same except all the convolutions and upconvolutions were done in 3d in order to generate inter-frame relationship. This model gave better results than the baseline considering the average losses per epoch given in table 1. However, the model took over 24 hours to train for 10 epochs as compared to around 7 hours for the baseline. (on Nvidia Geforce GTX 1080)

The evaluation metrics could not be evaluated for the model because the model generated cuda out of memory error. Hence, due to the unavailability of better GPU, the visual and metrics results are not shown for this model.

Step 3: Improving the Baseline by Spatial and Volumetric Attention

1. Spatial Attention:

For spatial attention, CBAM module was used in between various convolutional layers without any changes to the overall baseline architecture. However, in CBAM, the channel wise attention function was deleted since the baseline is run in frame-by frame manner. Spatial attention, thus, helps in identifying the regions of optimal focus areas in the stack. Fig. 2 shows the results of depth maps using spatial attention.

2. Volumetric Attention:

For volumetric attention, CBAM module was modified to perform 3d convolution across stack of images as compared to 2d in spatial attention case. Channel wise attention function was deleted in this case too, since 3d convolution is already catering for stack-wise attention. Fig. 2 shows the results of depth maps using volumetric attention.

For both models, 6 attention modules were added. Both modules were added in same locations in between convolutional layers. (after 6th, 9th, 12th, 17th, 20th, 23rd)

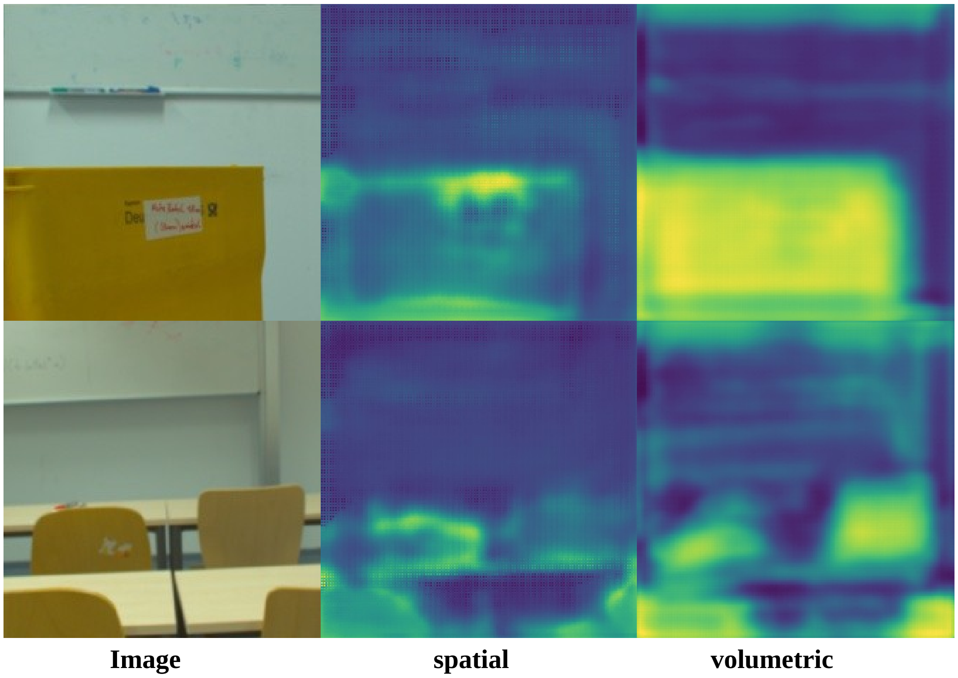

Fig.2

Overall Comparison Between All Networks

Comparing the losses given in table 1, baseline with volumetric attention gives the best results with the smallest average loss in the 10th epoch. Baseline with spatial attention does not seem to perform any better than the baseline. Inter-frame performs better than the baseline, however, it takes a lot of time for training as compared to the baseline and baseline with spatial and volumetric attention.

Comparing the evaluation metrics in table 2, baseline with volumetric attention gives the best results with respect to all the metrics – MSE, RMS, log RMS, absolute square, badpix and bump are smallest and three accuracy measures are the highest among all models for 10 epochs. Inter-frame results could not be generated due to GPU memory issue. However, baseline with spatial performs better than the baseline in terms of evaluation metrics. It performs better in all metrics than the baseline, which suggests that the baseline with spatial attention might be better if the model is run for more epochs.

Fig. 3 shows the visual results of all the networks and in all cases volumetric gives best results as compared to ground truth for the 10 epochs results.

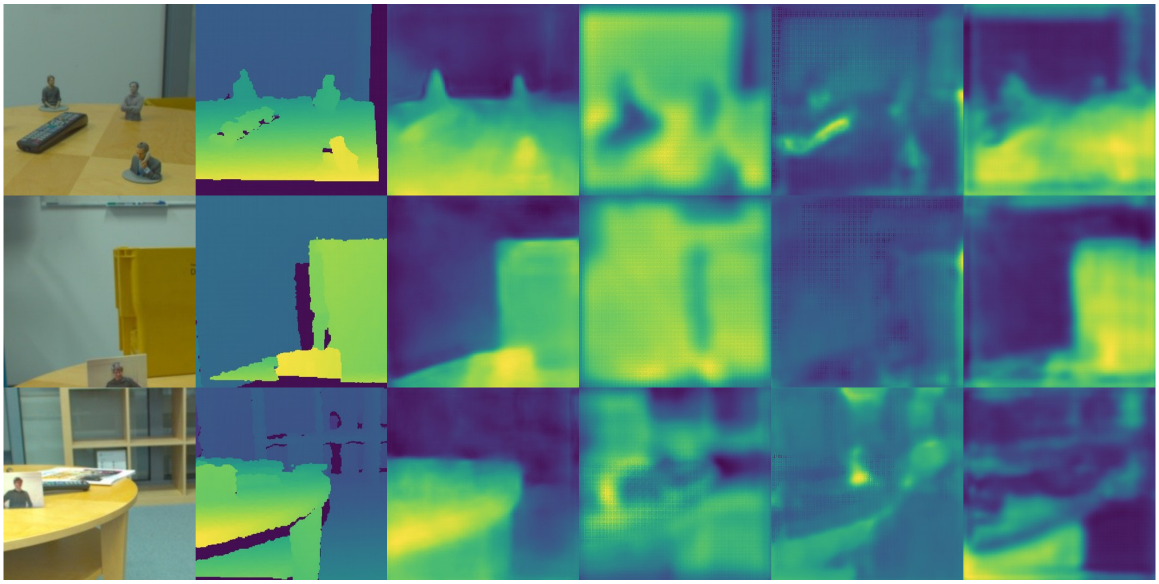

Image **Ground truth** **Baseline-200-epochs** **Baseline-10-epochs** **spatial** **volumetric**

Fig. 3

Average losses per epoch for 10 epochs for each of the network										
Network Architecture	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Baseline	415	58.4	39.5	32.9	27.4	23.8	20.2	15.6	16.9	14.9
Inter-frame Learning	122	77.4	46.7	34.0	26.3	20.2	16.9	15.2	13.8	11.4
Baseline with Spatial Attention	284	95.5	61.9	41.5	32.5	26.4	22.5	19.6	17.7	14.9
Baseline with Volumetric Attention	113	32.6	18.9	14.7	11.7	9.14	7.78	6.91	6.64	5.38

Table. 1

Errors for Each of the network (10 epochs training)										
Network Architecture	**MSE**	**RMS**	**Log RMS**	**Abs. rel**	**Sqr. rel**	**δ**	**δ^2**	**δ^3**	**Badpix**	**Bump**
Baseline (200 Epochs)	3.6e-4	1.7e-2	2.7e-1	2.1e-1	5.3e-3	7.0e+1	9.0e+1	9.5e+1	7.2e-1	3.6e-1
Baseline (10 Epochs)	4.7e-2	2.1e-1	1.4e+0	3.7e+0	9.5e-1	4.8e+0	1.1e+1	1.8e+0	8.9e+1	9.5e-1
Inter-frame Learning	cuda out of memory error									
Baseline with Spatial Attention	6.8e-3	7.1e-2	1.4e+0	6.8e-1	5.3e-2	1.0e+1	1.9e+1	2.9e+1	3.4e+1	5.1e-1
Baseline with Volumetric Attention	1.9e-3	4.0e-2	4.9e-1	5.5e-1	2.8e-2	3.4e+1	6.1e+1	7.9e+1	1.0e+1	3.8e-1

Table. 2

Step 4: Comparison with Part 1

1. Comparison

Part 2 with the original baseline (200 epochs trained model) seems to perform better than the part 1 as can be seen by the depth maps shown in Fig. 4. The part1 results are very noisy and clearly fails to identify regions with different depths as compared to the results of the baseline. Part 1 depth maps were generated using greater stack of images (balls = 25, keyboards = 32) and with higher resolution in the original case, hence, they gave better results, while in part 2, stacks of 10 images and of lower resolutions are used to generate a depth image.

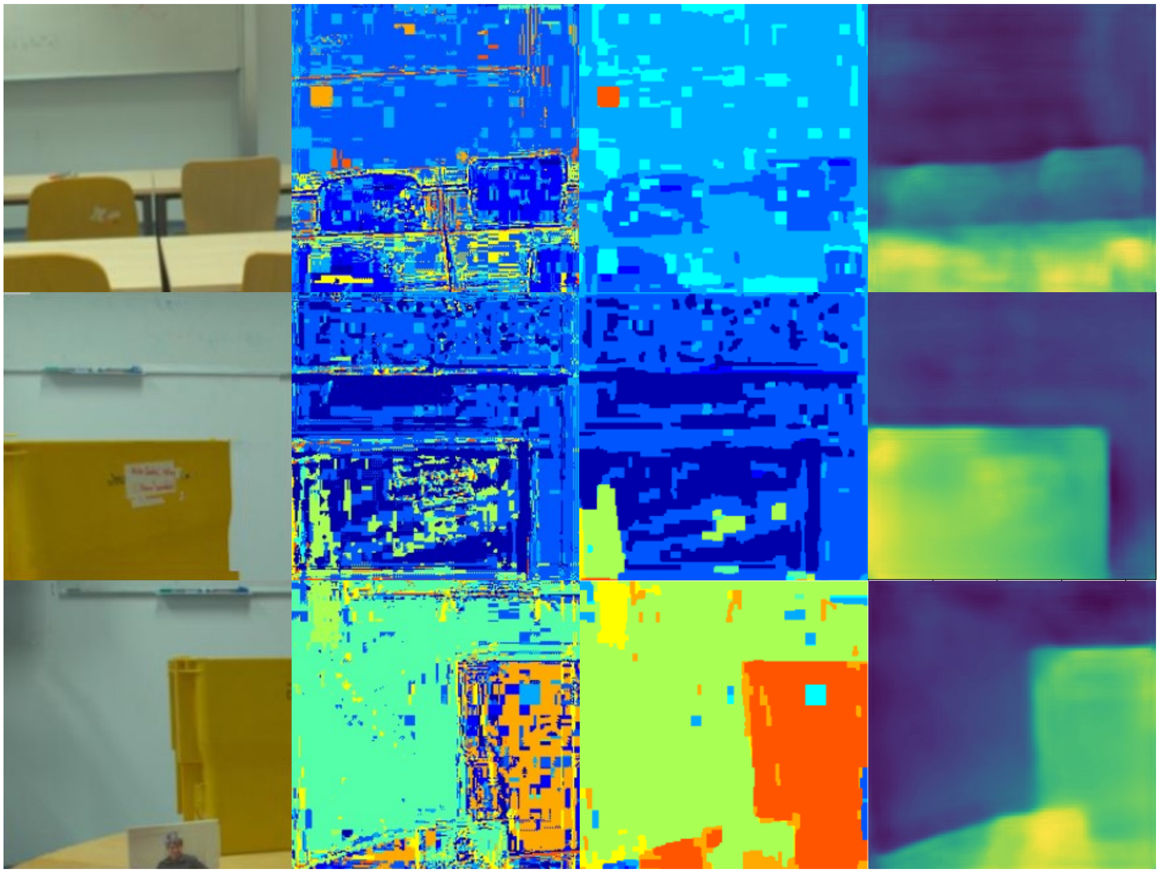

Image **Focal Measure** **Graph-cut** **DDFF**

Fig. 4

2. Pros and Cons of each Methods

Part 2 – Network Architecture

Pros	Cons
End-to end learning, automatic feature learning. Hence no need of hand-engineered feature extraction	Training deep neural networks (DNNs) is very time-consuming
Once trained, the depth map could be predicted in short amount of time with GPU	DNNs require huge data-sets
Light field imaging could be used to generate stacks of images whereby from one image, focal stacks could be generated with depth map using RGB-D sensor	Manually obtaining a stack of refocused images is very time-consuming

Part 1 – Focal Measure & Graph Cuts

Pros	Cons
Gives more refined depth maps	Requires hand-engineered feature extraction
Does not require GPU for computations	Requires larger focal stack for each image to generate an all-in-focus image
	Takes a long time to generate the depth map – image-alignment, focal measure and graph-cuts takes time for computations