CS 6320 Project 5.2

Janmesh
u1367465

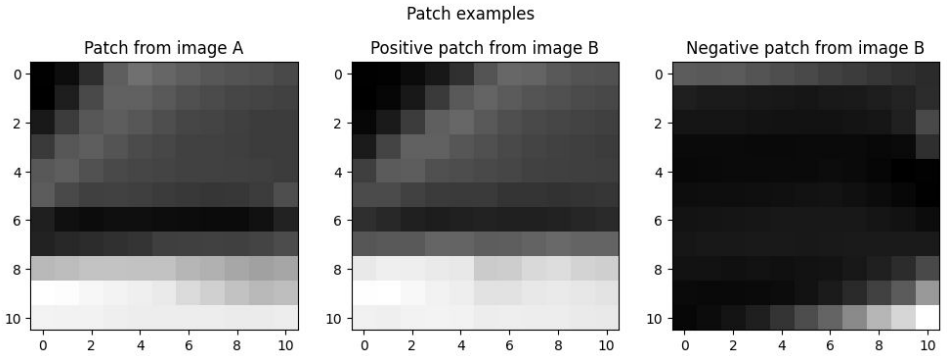
Copy the output of print(net_tr) from the notebook here. You can use screenshot instead of text if that's easier. (5 pts)

```
MCNET(
  (net): Sequential(
    (0): Conv2d(1, 112, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(112, 112, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): Conv2d(112, 112, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): ReLU()
    (6): Conv2d(112, 112, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU()
    (8): Conv2d(112, 112, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU()
    (10): Reshape()
    (11): Linear(in_features=27104, out_features=384, bias=True)
    (12): ReLU()
    (13): Linear(in_features=384, out_features=384, bias=True)
    (14): ReLU()
    (15): Linear(in_features=384, out_features=1, bias=True)
    (16): Sigmoid()
  )
  (criterion): BCELoss()
)
```

What's the purpose of ReLU and why do we add it after every conv and fc layers? (2 pts)

ReLU is an activation function to introduce non-linearity in the network, we usually need a activation function after every layer as the output of the conv and fc layers is essentially linear combination of input so we introduce a non-linearly so that the network can learn complex nonlinear functions and not just linear combination of inputs.

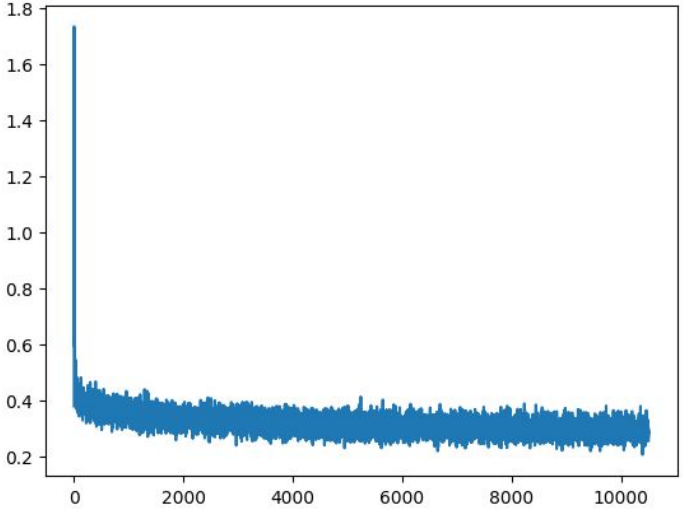
Show the patch example from your gen_patch function. (5 pts)



Giving a true disparity map for each stereo pair, how do we extract positive and negative patches for the training? (2 pts)

We use some x, y, and ws (window size) to extract a patch from an image, this can be considered as positive patch and for negative patch we apply random offset along a particular direction to the to parameters used for extracting positive patch.

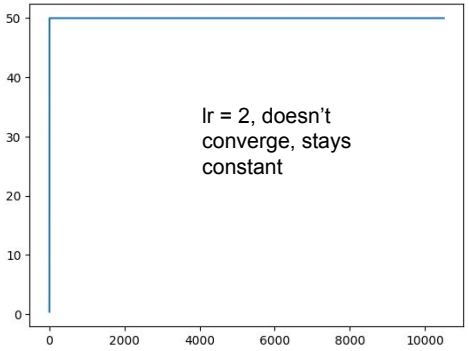
Copy the training loss plot and the final average validation loss of your best network. (5 pts)



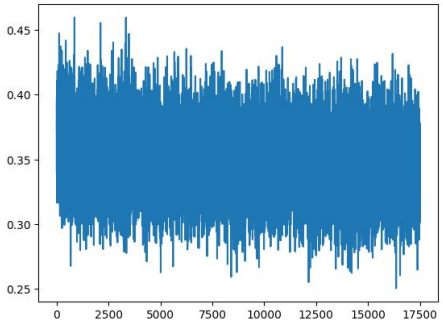
Batch Size = 512, Adam, lr = 6e-4, epochs = 1501. Average validation loss = 0.2968

How does changing learning rate (try using large (> 1) vs small value (< 1e-5)) effect the training? Why? (2 pts)

The learning rate doesn't converge when using a learning rate of 2 (>1) as the learning rate is essentially the step size to move opposite to the gradient directions and when this step size too large then it can't move towards the minima as the large step size is making it move past the dip in energy curve. If it is too small i.e. 1e-6 (< 1e-5) the it will need much more steps to converge as at every step it can only move small amount towards the minima and will require more steps.



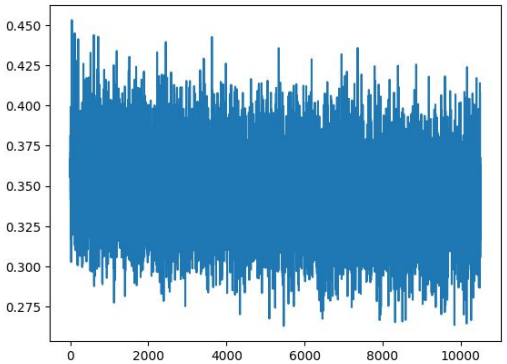
lr = 2, doesn't converge, stays constant



lr = 1e-6, convergence very low, not achieved val acc < 0.3 even after 2500 epochs

What's the difference between Adam and SGD? What do you notice when switching between them? (2 pts)

SGD maintains a constant learning rate throughout the training whereas Adam adapts the learning rate as the training unfolds based on average first moment and average second movement of gradients. From below graph and one in Page 4; one can see that Adam is faster to converge and gives a better validation accuracy of 0.2968 compared of 0.345 of SGD.

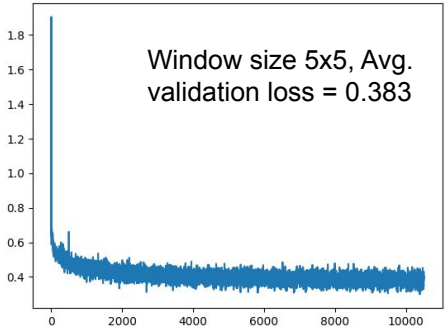


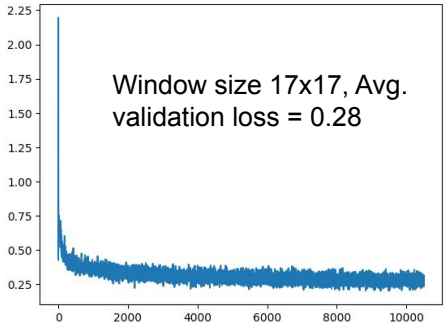
Batch Size = 512, SGD, lr = 6e-4, epochs = 1501. Average validation loss = 0.345

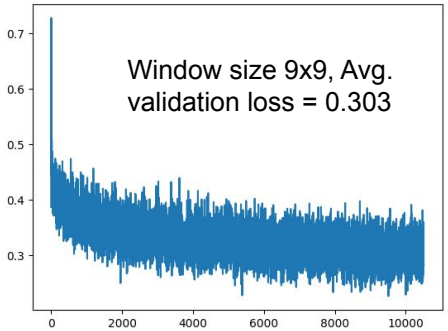
Is your validation loss lower or higher than your training loss? What is the reason for that? How would we get a better validation loss? (2 pts)

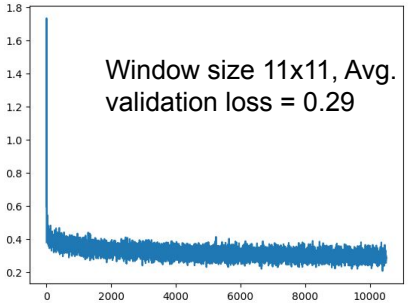
On the best model, the final training loss is around 0.257 whereas the validation loss is 0.2968. Validation data is the held out data on which the model has not been optimized on, the model has not seen the data so it is should be more than the training loss. We can get a better validation using some regularization methods like dropout, weight decay, batch normalization, data augmentation, etc.

Copy the training loss plot and the final average validation loss of the networks with different window sizes (3 pts)







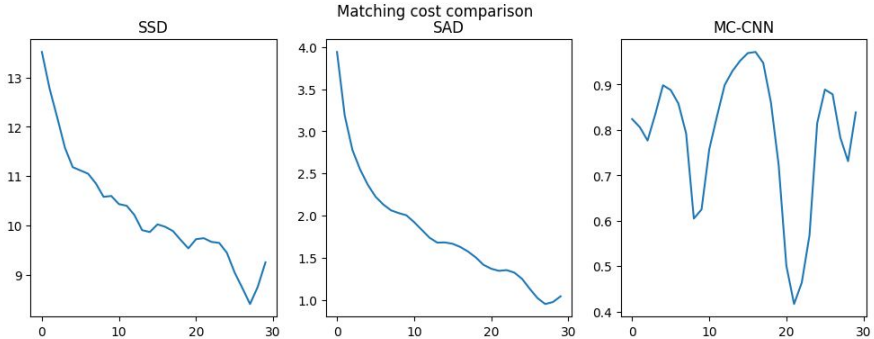


What's the effect of varying the window size on performance? Do you suppose there is an optimal window size for all images? Explain why or why not. (2 pts)

Increasing the window size seems to generally improve the performance, but it should not be increased too much.

There is no optimal window size for all images, optimal window size of one image with some resolution might be too small (noisy output) or too large (more disparity information than required) for another image with some other resolution.

Show the matching cost comparison plot between SSD/SAD/MCCNN (2 pts)

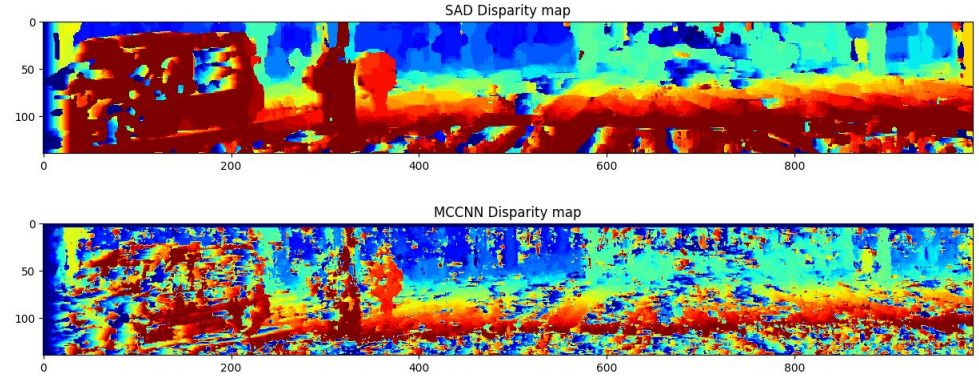


Based on the matching cost comparison plot, how are SSD, SAD, and MC-CNN different? Can you think of a scenario where matching with MC-CNN would be prefered over SSD/SAD? (2 pts)

SSD and SAD calculates cost on based of the pixel values for the patch in an image. Whereas MC-CNN is a learned model which calculates the cost using weights learned during training. It extracts the features which is deems necessary and uses it to calculate loss.

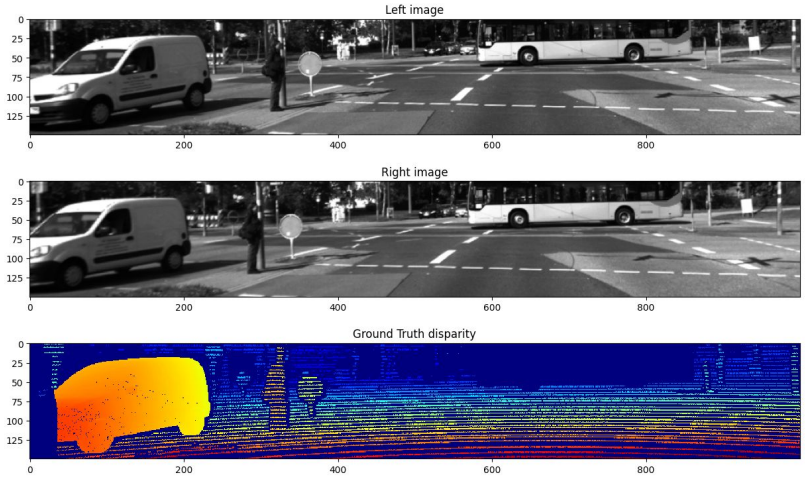
MC-CNN has much lower cost compared to SSD and SAD so when in need of lots of cost calculation then MC-CNN will be preferred.

Show the disparity map comparison between SAD and MCCNN (3 pts)



Qualitatively, between SAD and MCCNN, which disparity map do you think looks better? Explain your reasoning. (2 pts)

MC-CNN output is noisy compared to SAD but is has more detailed structure of objects like contours of car, etc. This is because it is learned directly from ground truth data, the noise can be attributed to the noisy ground truth disparity, like the horizontal lines shown below but overall it seems better.



Show the quantitative evaluation between SAD and MCCNN (average error, etc.) (2 pts)

Metrics:	SAD	MCCNN
Average Error	18.351213	15.651435
%error > 1 pixel	97.43333333333334	93.682
%error > 2 pixels	95.194	91.71933333333332
%error > 4 pixels	91.27866666666667	87.31466666666667

Quantitatively, between SAD and MCCNN, which one achieve better score on the metrics? Why do you think that's the case? (2 pts)

MC-CNN archives better score than SAD, mainly as it is trained using ground truth data, which has been collected and verified by humans where SAD just compares the pixels to get the disparity.

Additional Info

<Any additional information or results to be included? >