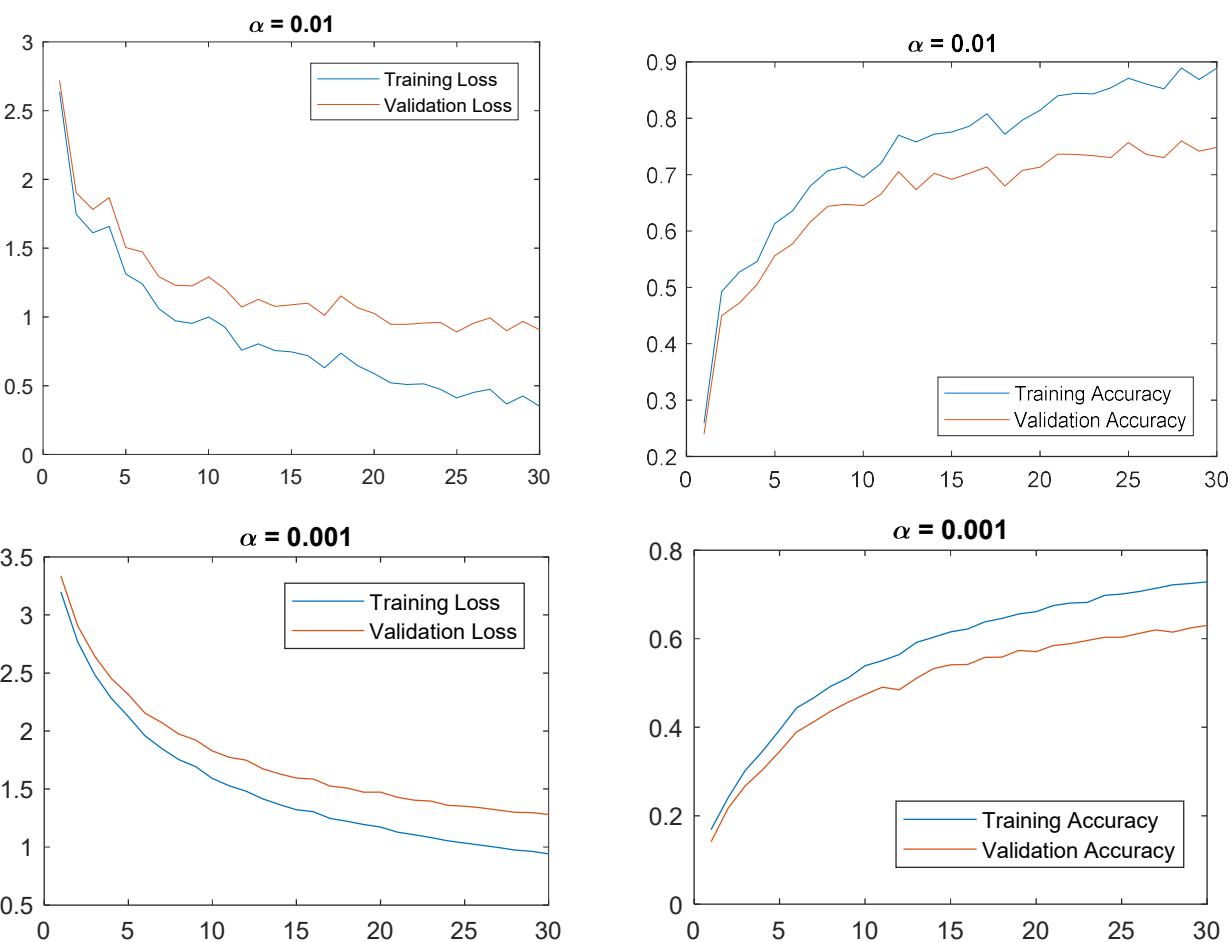


- 1.1.1) Deep networks have many hidden units, so they must apply activation functions many times. A ReLU activation is very efficient to compute compared to sigmoid.
- 1.1.2) If the activation function is linear, any number of layers could be replaced by a single layer. This is because the outputs will be a linear combination of the inputs.
- 2.1.1) Constant values would not be good, because then all the weights will be updated identically (essentially making all hidden units combine to be effectively one hidden unit). Instead, it would be better to have many different values, so they can each provide different outputs given the input.
- 2.1.3) I chose to initialize all weights and biases to be uniform random numbers in the range (-1, 1). This ensures that there's no symmetry in the network and that some weights and biases are initially negative (a likely feature of the ultimate network).
- 2.4.1) Stochastic gradient descent will converge faster because it can handle faster learning rates. However, it'll be more unstable. Batch gradient descent is more stable because it handles the whole dataset, but it'll be slower.

3.1.2)



At neither learning rate did the validation accuracy reach a maximum. Since the larger learning rate learned faster, its accuracy was 74%.

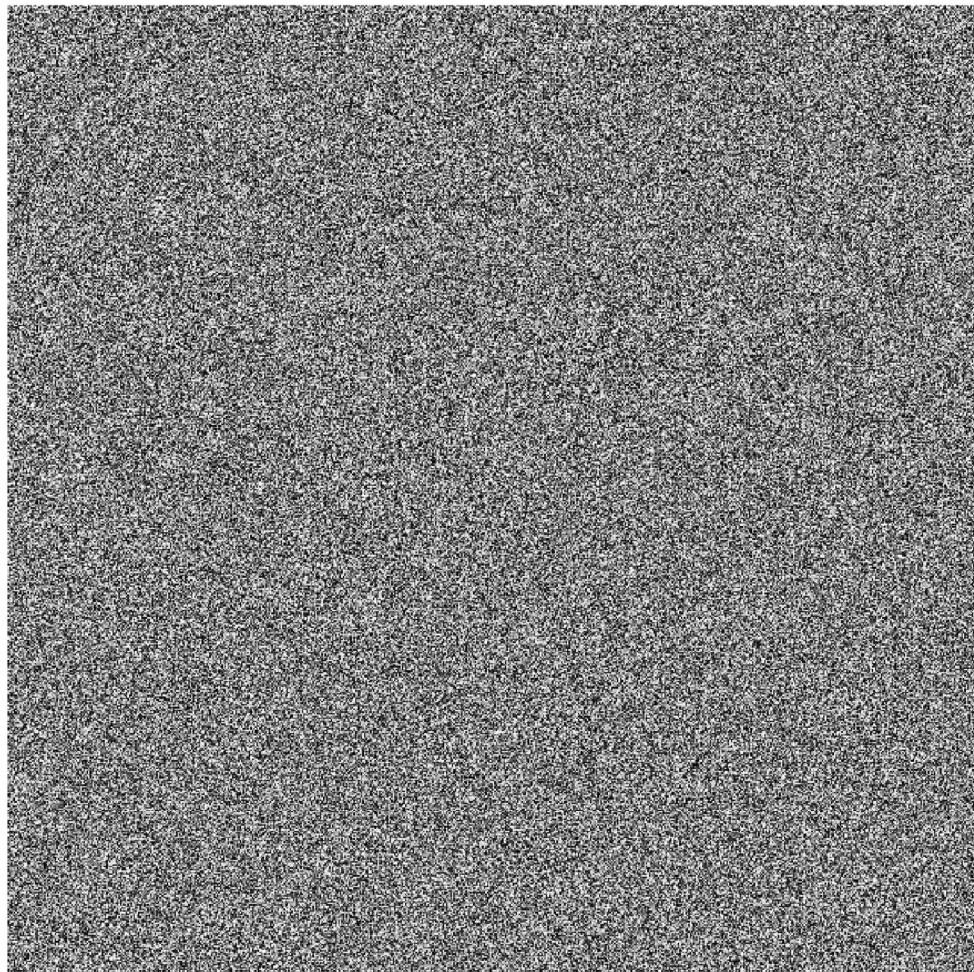
Basel Alghanem

16-720

Homework 3

3.1.3) The accuracy is 75.5% and the average loss is 0.914.

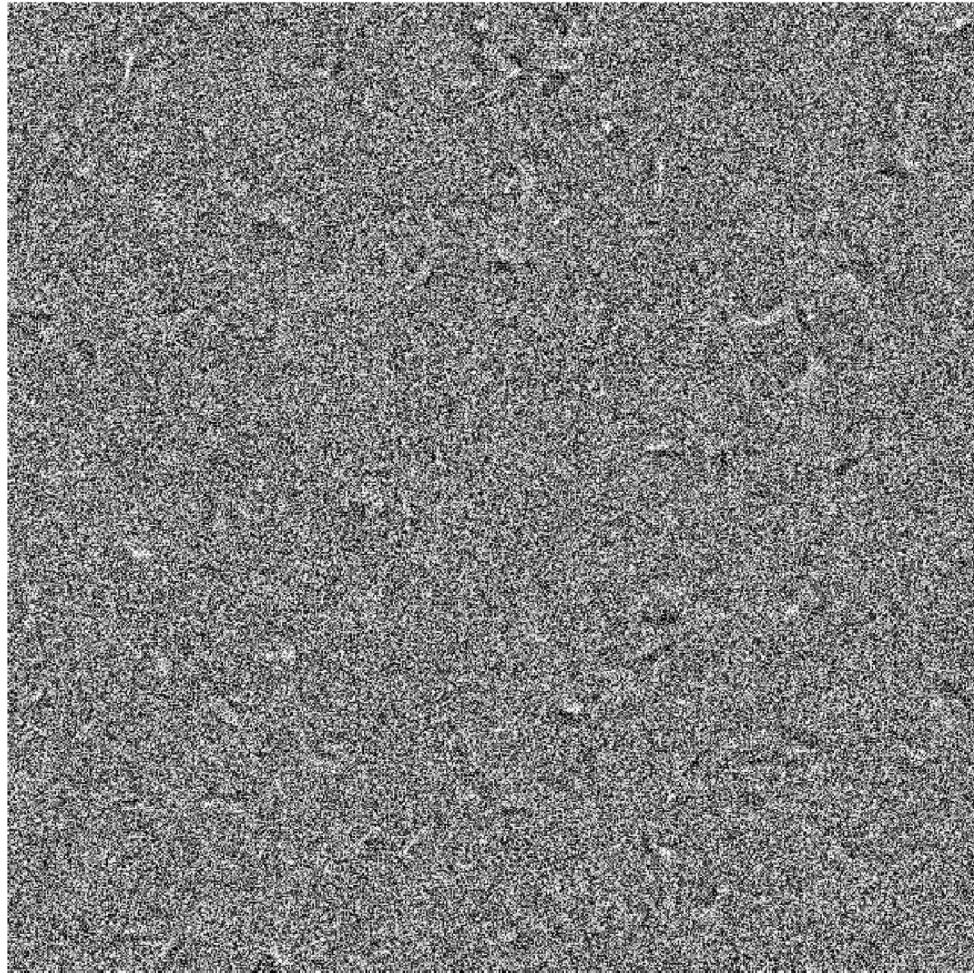
Initial Weights



Basel Alghanem
16-720
Homework 3

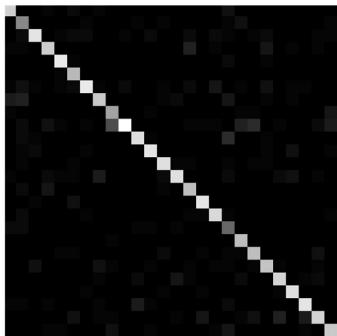
3.1.3)
cont.

Final Weights



The initialized weights look totally random. The final weights are definitely less random, and there are some patterns. However, I definitely cannot tell what the various hidden units are doing.

3.1.4)



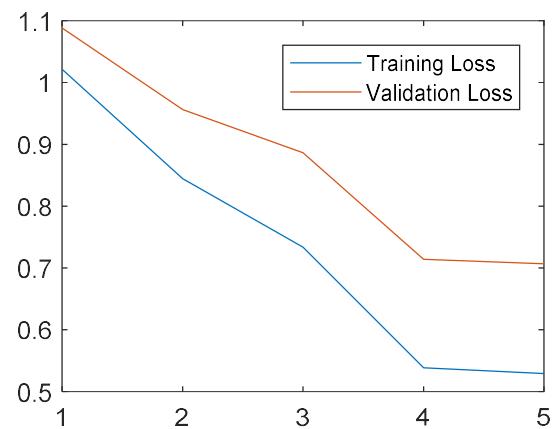
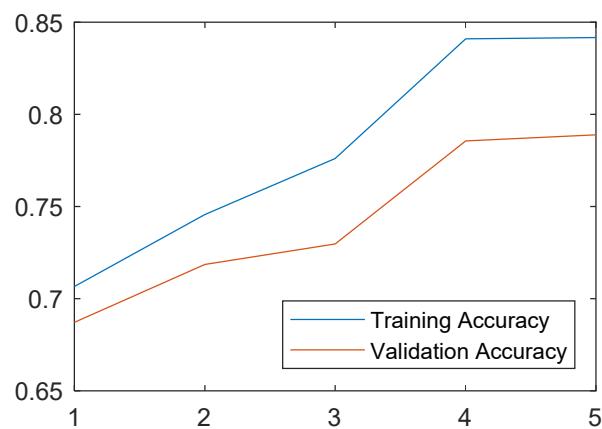
The most common mixup is classes 9 and 10 (letters i and j)

Basel Alghanem

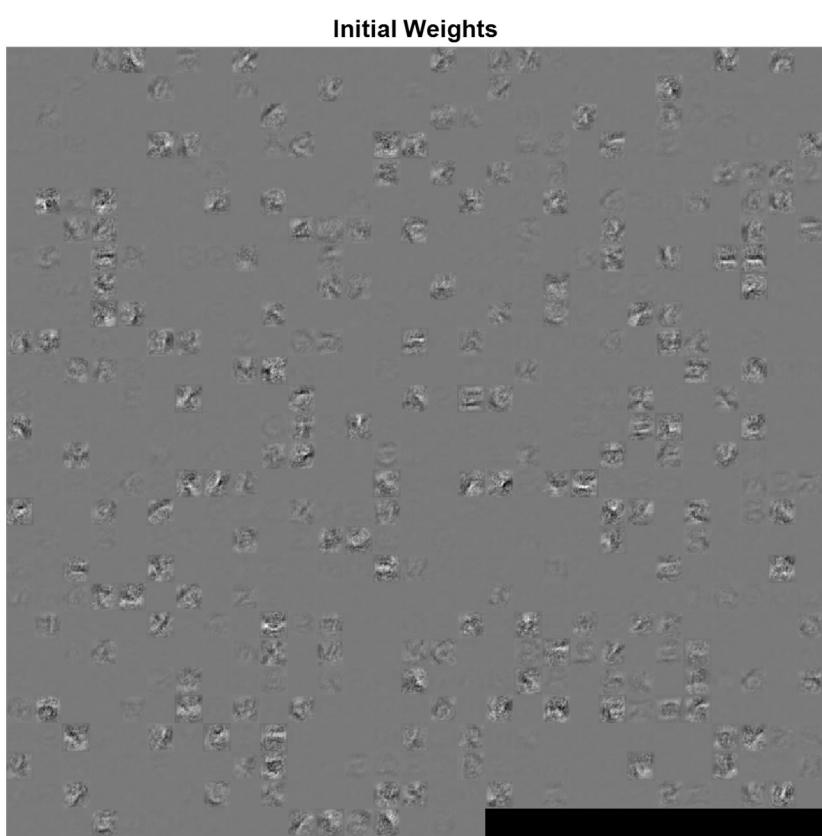
16-720

Homework 3

3.2.1)



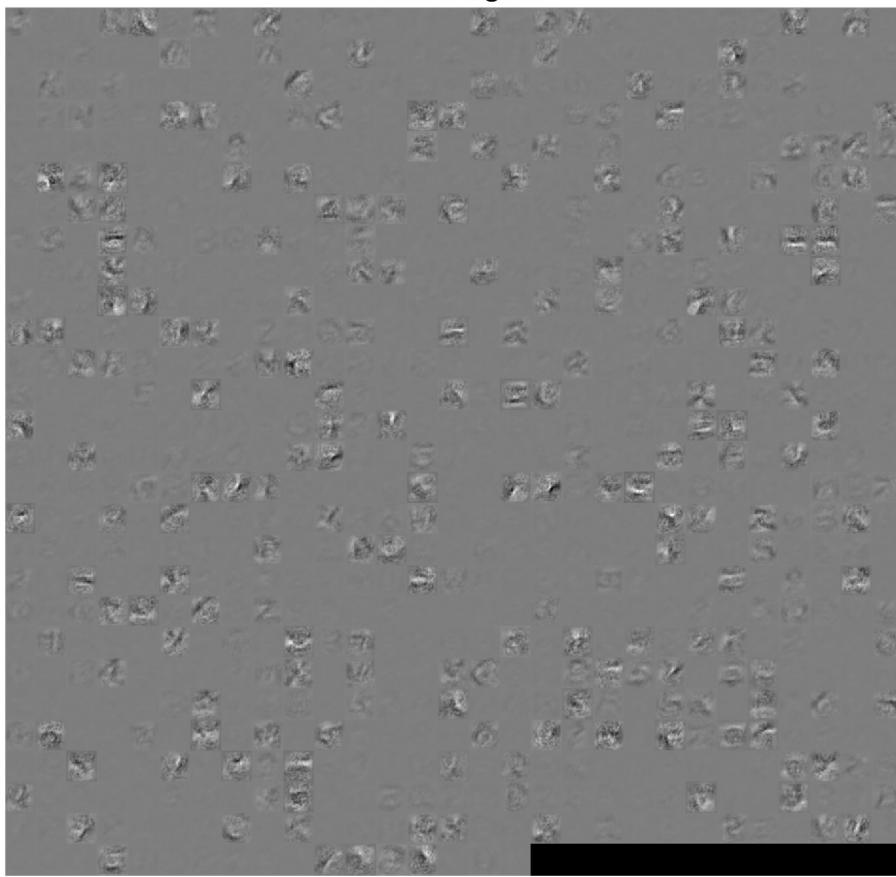
3.2.2)



3.2.2)

cont.

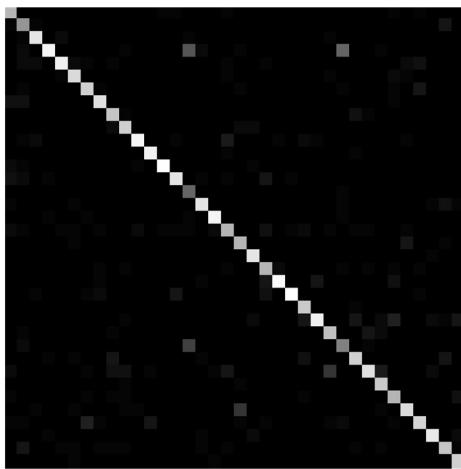
Final Weights



Most of the major patterns are the same, but there are subtle differences around the existing patterns and subtle new patterns.

The final weights had 78.9% accuracy and 0.690 average loss on the test set.

3.2.3)



The most confused classes are 4 and 27 (corresponding to 'd' and '0'). 'd' and '0' were also often confused. While overall, accuracy actually increased on the test set, the worst confusion here is worse than the worst confusion before, because fo the addition of '0'.

Basel Alghanem

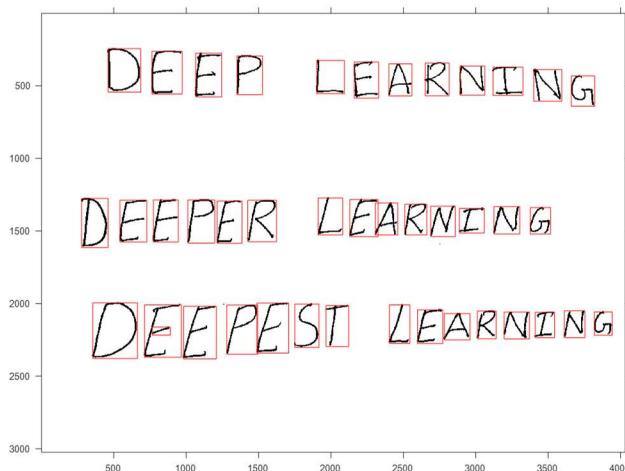
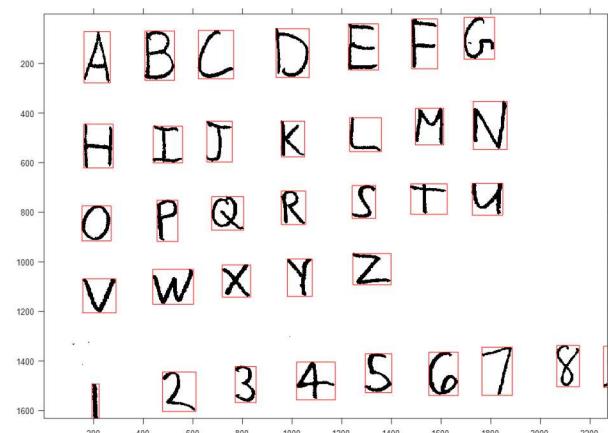
16-720

Homework 3

- 4.1) One assumption is that the separation of the background from the foreground is actually possible. This could be a problem in image two, which has background lines that could easily be mistaken for a 1.

Another assumption is that boxes can be neatly drawn. For example, in image 4, the italicized text means some of the boxes would overlap, which could hurt the accuracy of the network.

4.3)



Basel Alghanem
16-720
Homework 3

4.5) Image 1:

fj cj lc5f
i vake a t3 wj li5t
2 lwlc k jgh twe fikxt
twiwg jn tq dq lixi
b klalize yjw wave alk6ady
cjmv l6tld j twingi
g kfawakj yjwcxelh wiim
a naw

Image 2:

a k c j e f g
w i i k l m w
j k q k s t w
v w x y z
b z 3 g s g 7 k 7 j

Image 3:

w a i wws mre emasy
fwi sjy etim es they ddwt y ak6 sjng6
ee2ri g erm7qr

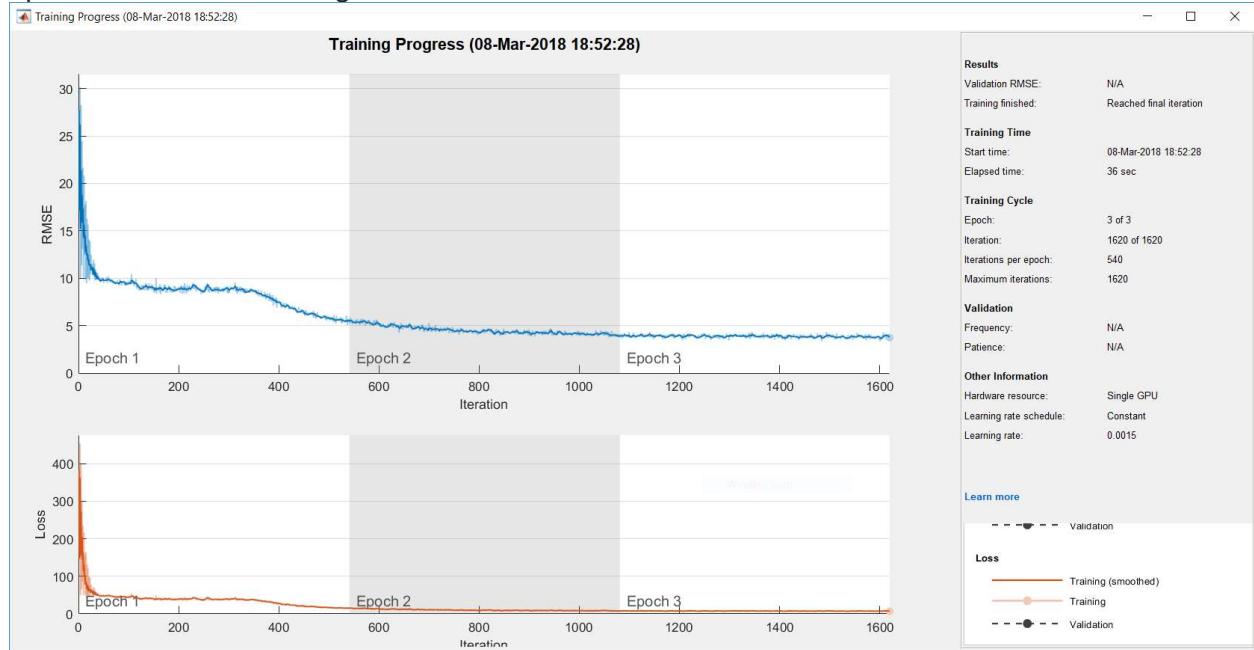
Image 4:

c c c c c 2 k j i n 6
f e e y e k l e d k n i n g
j e v e p c 5 f 1 e 2 k n i n g

5.1) The arguments to the layer generators were not correct to what was defined in the PDF, specifically with regards to the number of output filters & number of input channels in the various layers.

Basel Alghanem
16-720
Homework 3

- 5.2) I modified the weight initialization from normal about 0 to uniform about 0. I set the limits to ± 0.025 . I also updated the initial learning rate to 0.0015.



5.3.1)



Original images on the left and reconstructed images on the right. There are artifacts around each letter, but in each case, the letter is still recognizable.

Basel Alghanem

16-720

Homework 3

5.3.2) The average PSN ratio is 18.76.