Assignment 3 Report

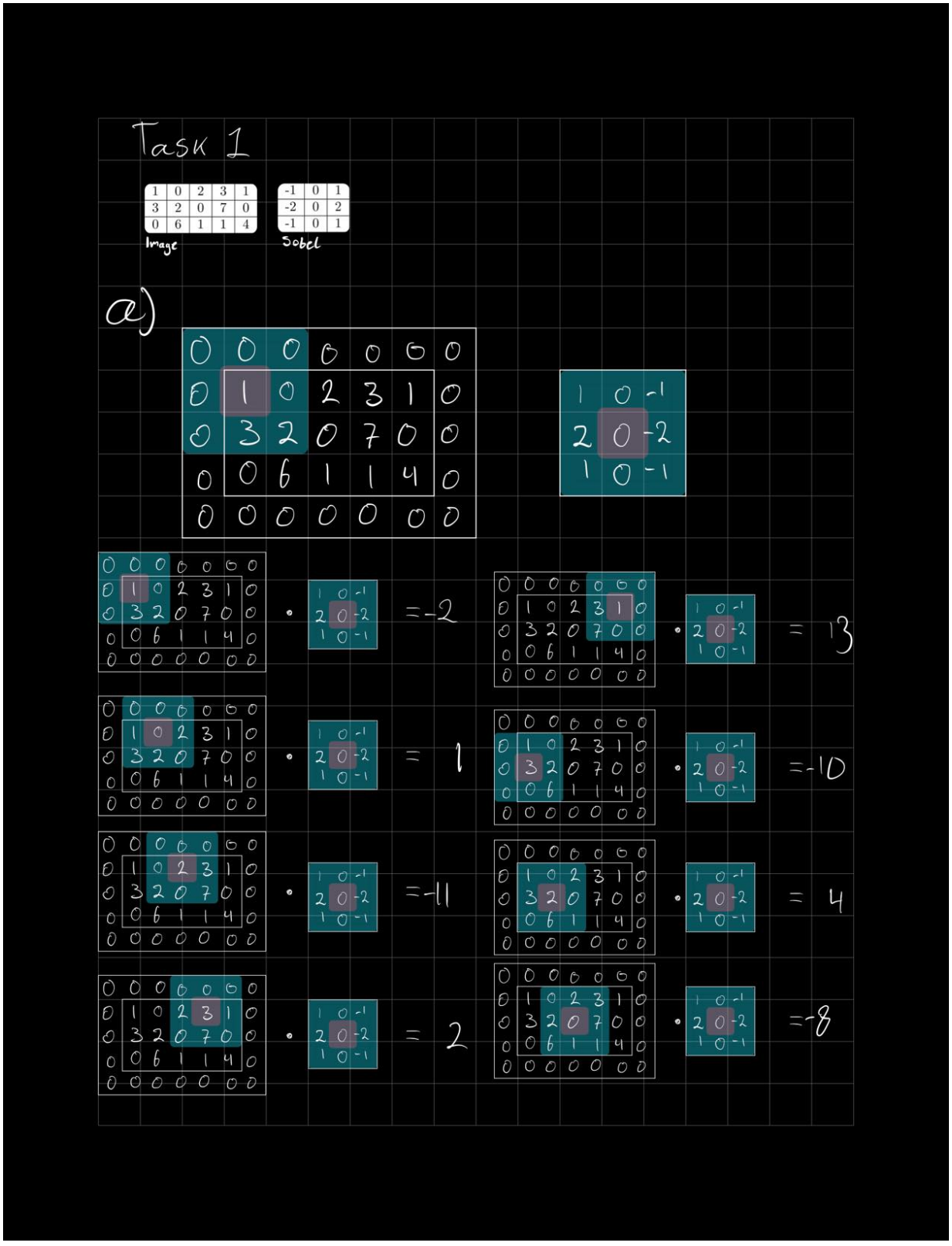
This is an outline for your report to ease the amount of work required to create your report. Jupyter notebook supports markdown, and I recommend you to check out this cheat sheet (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet). If you are not familiar with markdown.

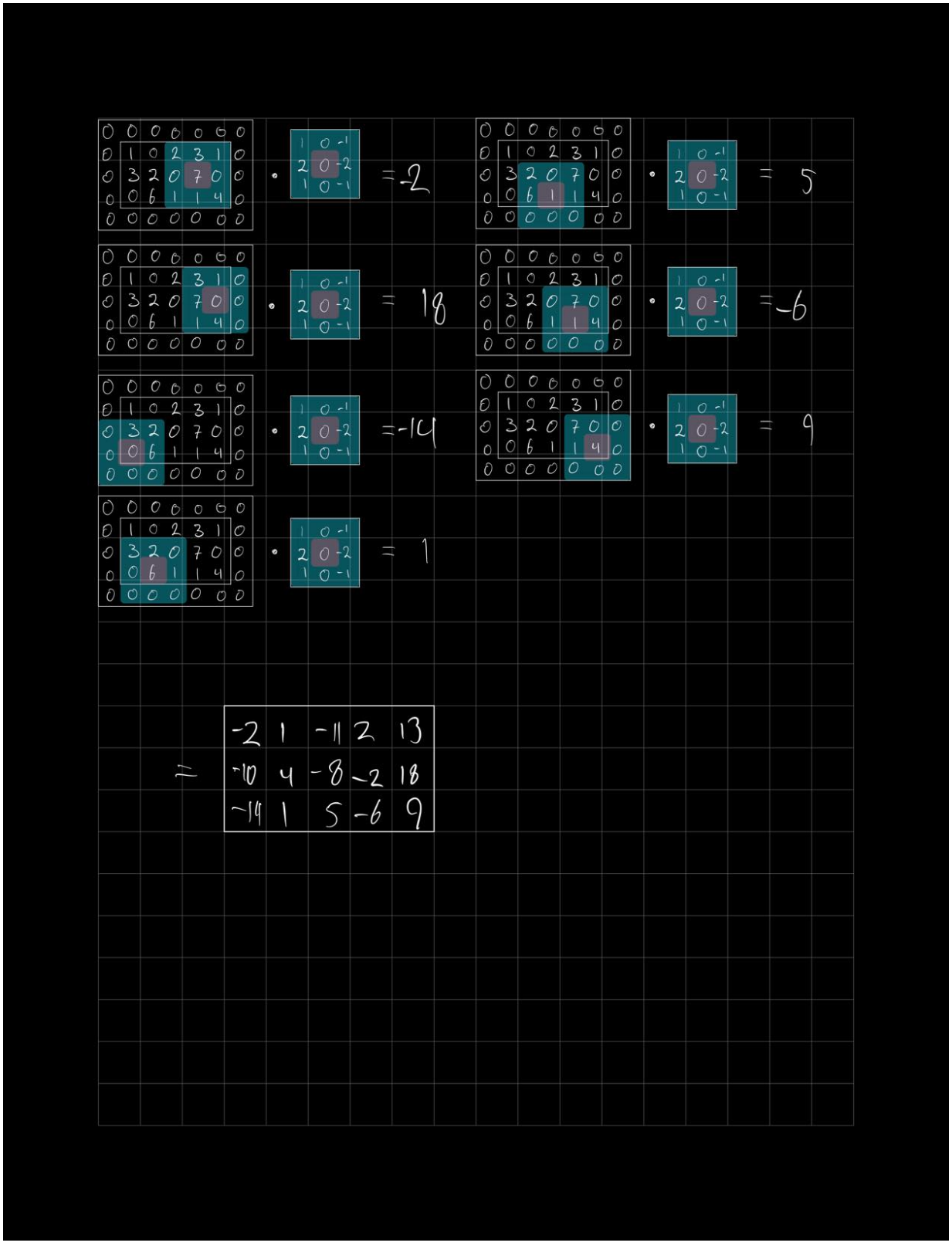
Before delivery, **remember to convert this file to PDF**. You can do it in two ways:

1. Print the webpage (ctrl+P or cmd+P)
2. Export with latex. This is somewhat more difficult, but you'll get somehwat of a "prettier" PDF. Go to File -> Download as -> PDF via LaTeX. You might have to install nbconvert and pandoc through conda; `conda install nbconvert pandoc` .

Task 1

task 1a)

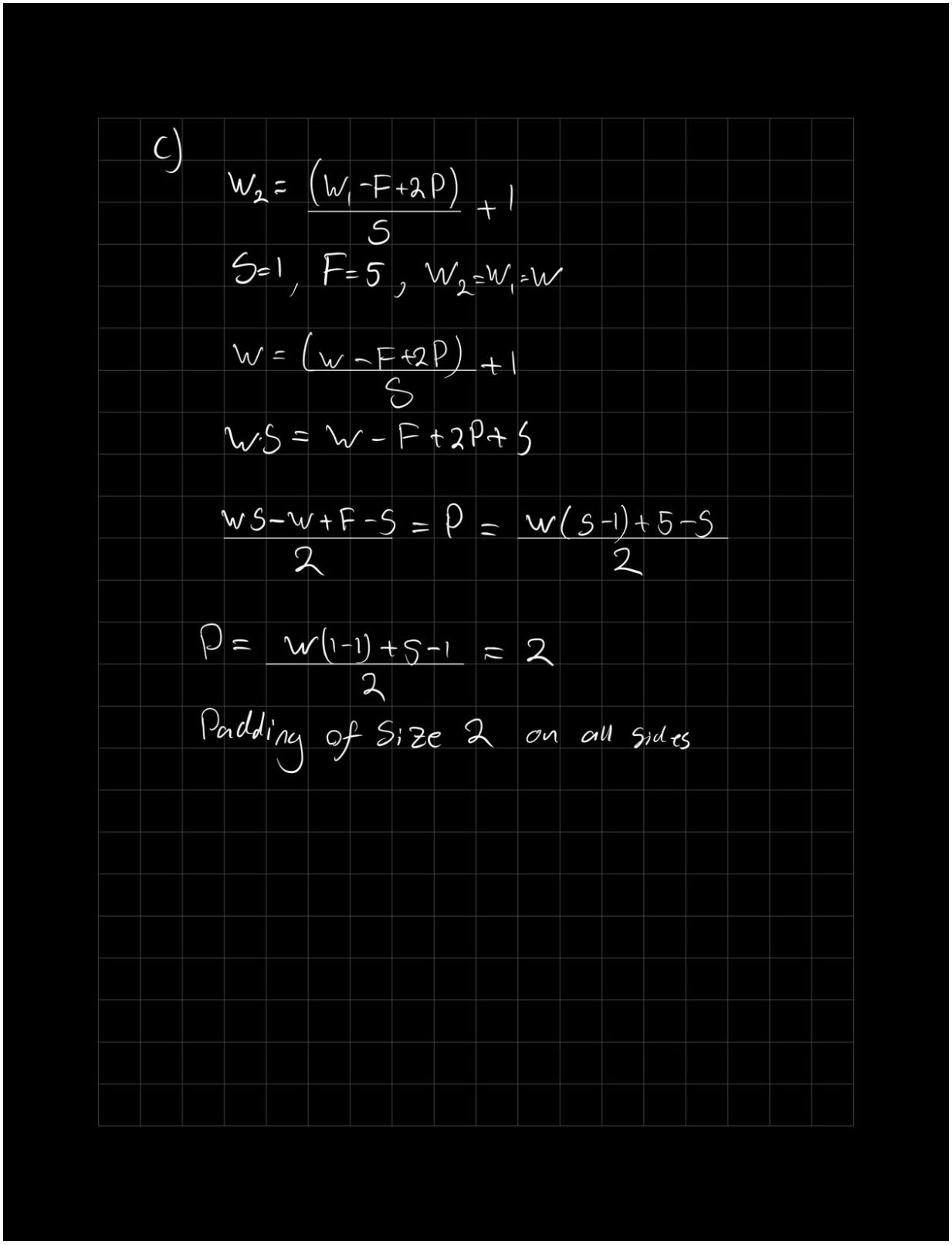

Task 1
Image
Sobel
a)
=-2
=13
=1
=-10
=-11
=4
=2
=-8



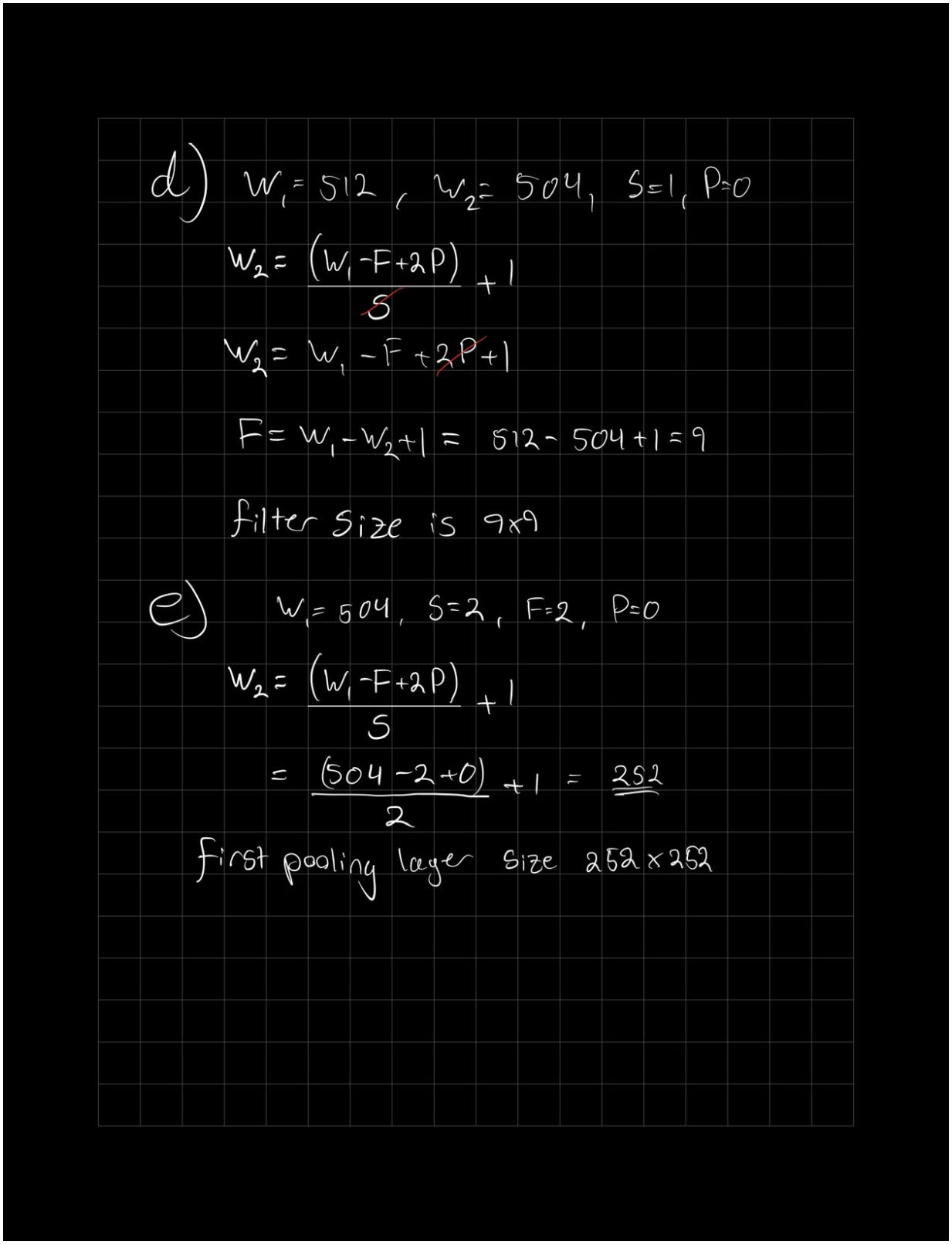
task 1b)

Max pooling is used to lower the resolution of the image by reducing the number of parameters. This is done by selecting the largest structural element and removing details that might be changed and not useful. Max pooling will make a more robust image that wont change much.

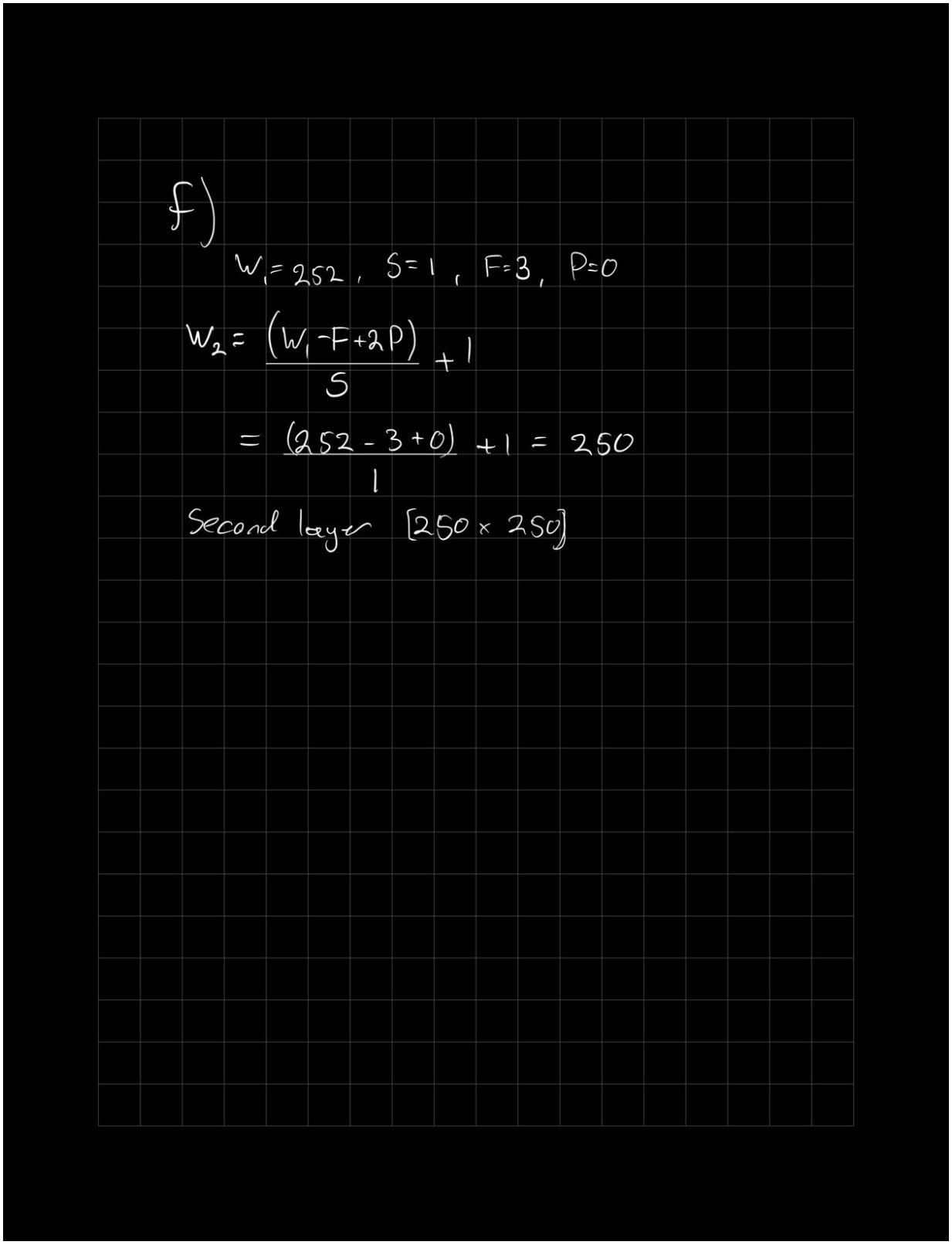
task 1c)


c)
$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
$S=1,\ F=5,\ W_2 = W_1 = W$
$W = \frac{(W - F + 2P)}{S} + 1$
$WS = W - F + 2P + S$
$\frac{WS - W + F - S}{2} = P = \frac{W(S-1) + 5 - S}{2}$
$P = \frac{W(1-1) + S - 1}{2} = 2$
Padding of size 2 on all sides

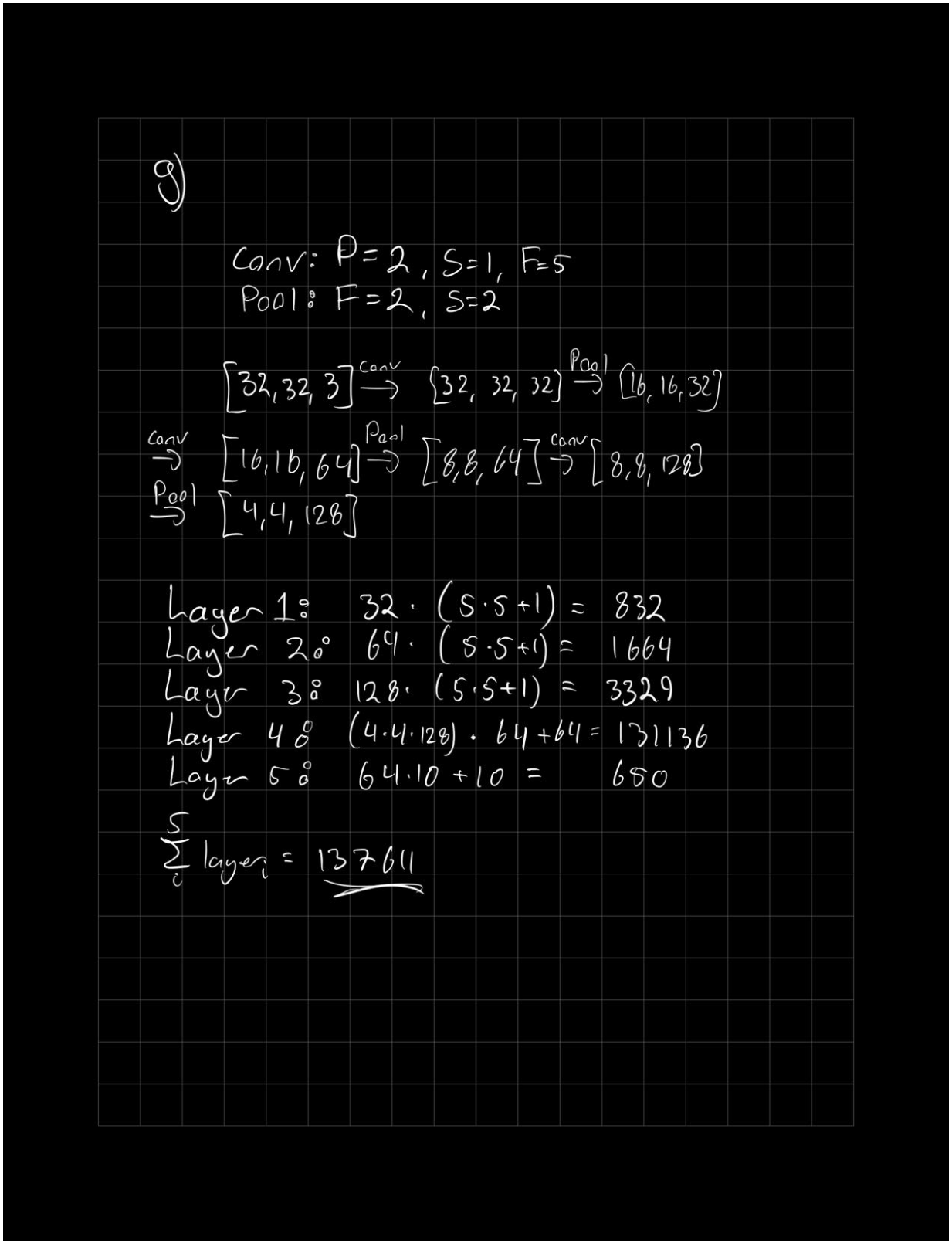
task 1d) 1e)


d) $W_1 = 512$, $W_2 = 504$, $S=1$, $P=0$
$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
$W_2 = W_1 - F + 2P + 1$
$F = W_1 - W_2 + 1 = 512 - 504 + 1 = 9$
filter size is 9x9
e) $W_1 = 504$, $S=2$, $F=2$, $P=0$
$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
$= \frac{(504 - 2 + 0)}{2} + 1 = 252$
First pooling layer size 252 x 252

task 1f)

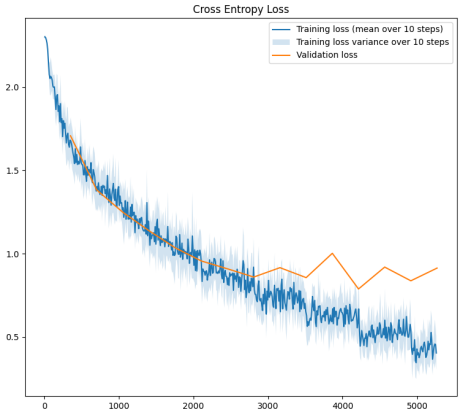

f)
$W_1 = 252$, $S = 1$, $F = 3$, $P = 0$
$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
$= \frac{(252 - 3 + 0)}{1} + 1 = 250$
Second layer [250 × 250]

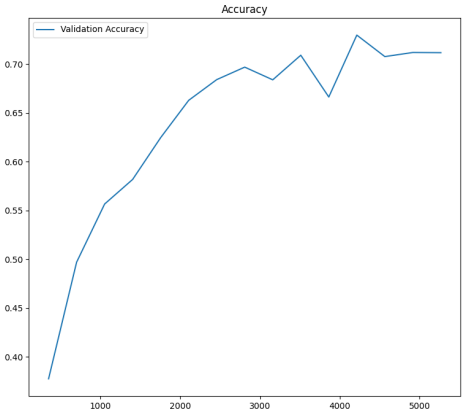
task 1g)



Task 2

Task 2a)





Task 2b)

Training accuracy: 0.8307

Validation accuracy: 0.7296

Test accuracy: 0.7295

Task 3

Task 3a)

Both Networks also use the SDG optimizer

Network 1

We use Kernel Size 3x3, Stride 1 and Padding 1. Added Data Augmentation with horizontal flip, random rotation and random crop.

Hyperparameters for first networks:

Epoch	10
Batch Size	64
Learning Rate	5e-2
Early Stop Count	4

Layer	Layer Type	Number of Hidden Units / Number of Filters	Activation Function
1	Conv2D	32	ReLU
1	BatchNorm2d		
1	MaxPool2d		
2	Conv2D	64	ReLU

Layer	Layer Type	Number of Hidden Units / Number of Filters	Activation Function
2	BatchNorm2d		
2	MaxPool2d		
3	Conv2D	128	ReLU
3	BatchNorm2d		
3	MaxPool2d		
4	Conv2D	256	ReLU
4	BatchNorm2d		
4	MaxPool2d		
	Flatten		
5	Fully-Connected	64	ReLU
6	Fully-Connected	10	SoftMax

Network 2

We use Kernel Size 5x5, Stride 1 and Padding 2. Use Dropout with 20% chance and LeakyReLU.

Hyperparameters for second networks:

Epoch	10
Batch Size	128
Learning Rate	3e-2
Early Stop Count	4

Layer	Layer Type	Number of Hidden Units / Number of Filters	Activation Function
1	Conv2D	32	LeakyReLU
1	BatchNorm2d		
1	MaxPool2d		
2	Conv2D	64	LeakyReLU
2	BatchNorm2d		
2	MaxPool2d		
2	Dropout		
3	Conv2D	128	LeakyReLU
3	BatchNorm2d		
3	MaxPool2d		
3	Dropout		
4	Conv2D	256	LeakyReLU
4	BatchNorm2d		
4	MaxPool2d		

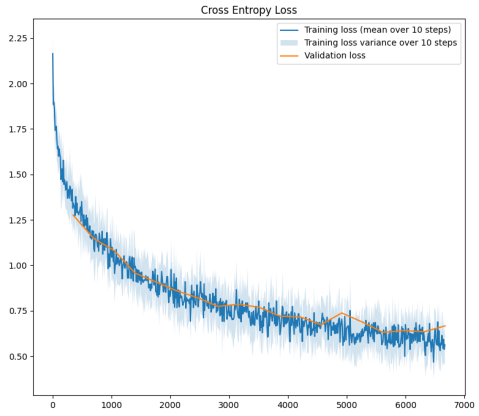
Layer	Layer Type	Number of Hidden Units / Number of Filters	Activation Function
4	Dropout		
	Flatten		
5	Fully-Connected	64	LeakyReLU
5	Dropout		
6	Fully-Connected	10	SoftMax

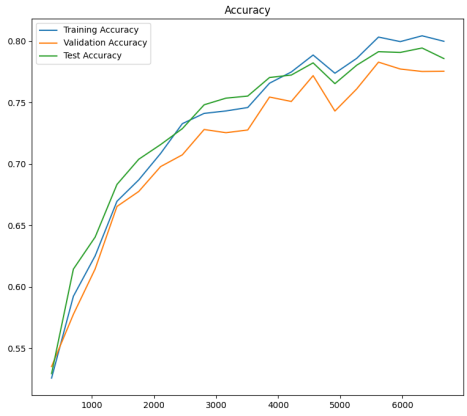
Task 3b)

Include final accuracy scores and plot for two models

Networks	1	2
Training Loss	0.5674	0.5044
Training Accuracy	0.8009	0.887
Validation Loss	0.6578	0.64
Validation Accuracy	0.7684	0.778
Test Loss	0.6606	0.65
Test Accuracy	0.7777	0.7612

Include plot of network 1





Task 3c)

Data augmentation made the network accuracy higher, but the training speed was increased

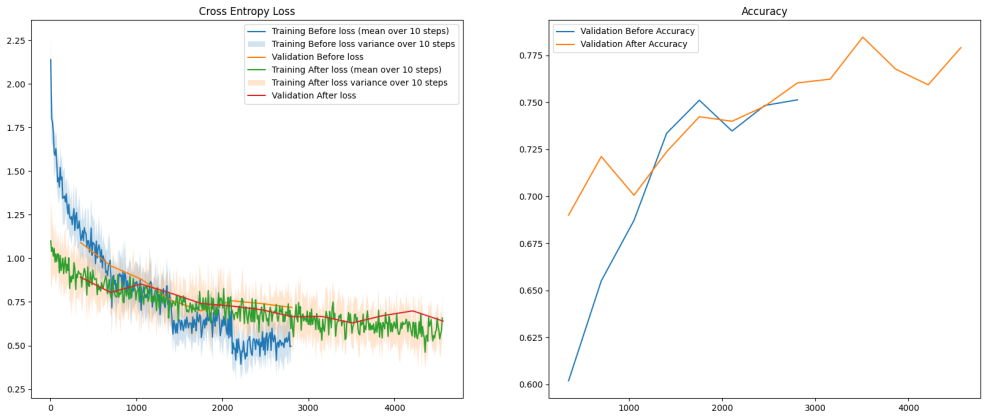
Reducing the filtersize to 3x3 increased the accuracy by a lot

Increasing the number of filters had a positiv and negativ effect depending on how many filters and if the filters was added to the start or end. After testing i found that adding 1 more filter to the end had the best effect

Batch normalization made the training of the network much faster

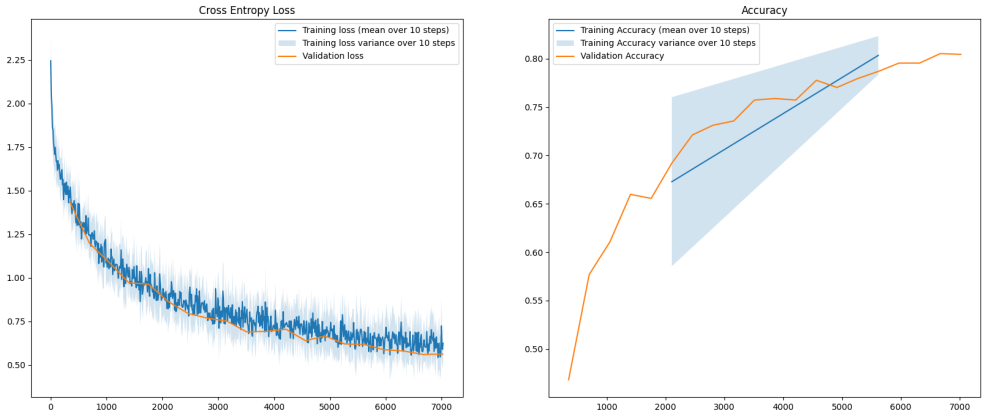
Dropout reduced overfitting and improved the network

Task 3d)



Task 3e)

Include plot



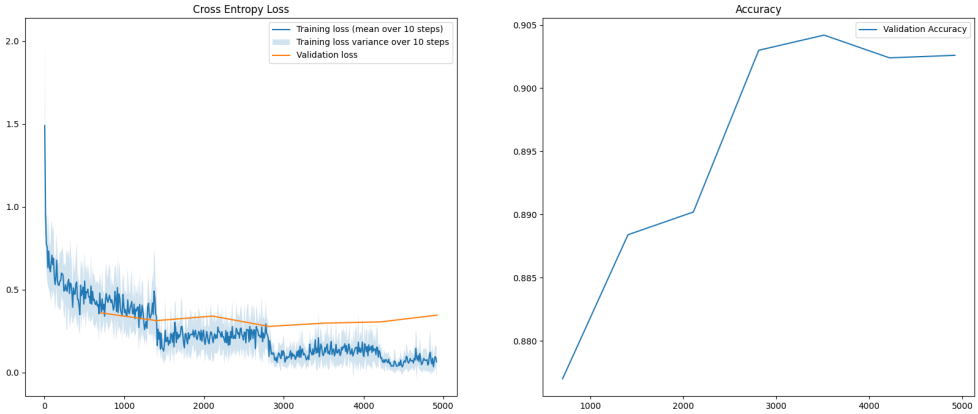
Train Loss: 0.49 Train Accuracy: 0.827 Validation Loss: 0.56 Validation Accuracy: 0.804 Test Loss: 0.55 Test Accuracy: 0.813

Task 3f)

There are not many signs of overfitting, but the the test and training accuracy is higher than the validation accuracy which shows some overfitting.

Task 4

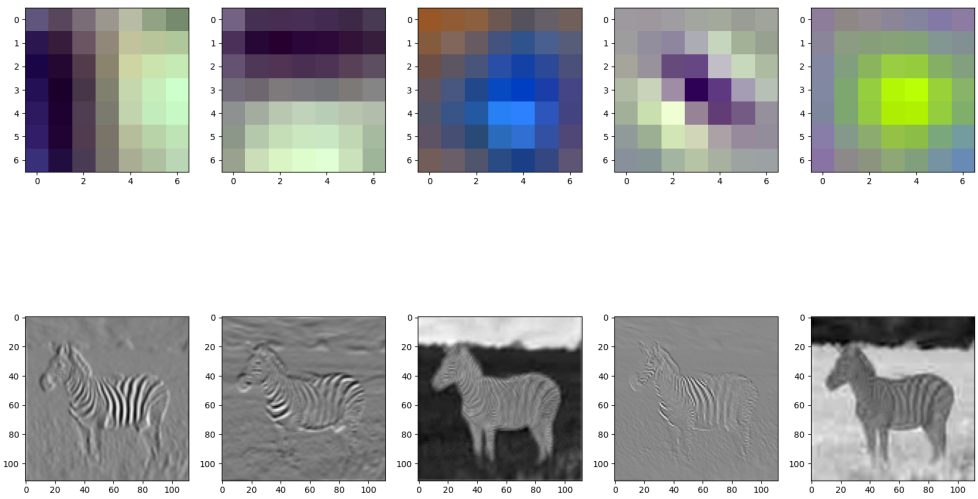
Task 4a)



epochs = 5 batch_size = 32 learning_rate = 5e-4 early_stop_count = 4 resize = 224 optimizer = Adam

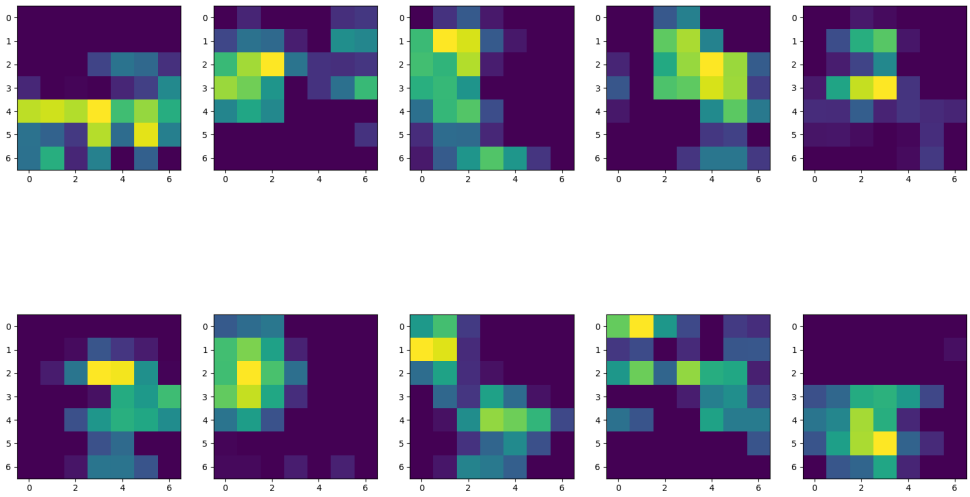
Test accuracy: 0.887

Task 4b)



We can see that these different filters retrive different features of the image. The first shape retrieves the vertival lines in the image, and the second retrieve the horizontal lines. Filter 3 and 5 activate on larges features, the grass and the sky respectivly. Unsure of what the 4 filter is showing, maybe the edges of the zebra.

Task 4c)



Unsure what we are seeing here. The activations seem to look random. Looking at the original zebra image the patterns might look like it is activating on the zebra.

In [ ]: