Task 1

a)

0	0	0	0	0	0	0
0	1	0	2	3	1	0
0	3	2	0	7	0	0
0	0	6	1	1	4	0
0	0	0	0	0	0	0

Using Padding !

First (1,1) Convolution is $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 2 \end{bmatrix} * \begin{bmatrix} 4 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = 2$

(1,2) $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = -2-3+4 = -1$

(1,3) $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 3 \\ 2 & 0 & 7 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = -2+6+7 = 11$

11 Continue this Pattern

Convolution $= \begin{bmatrix} 2 & -1 & 11 & 2 & 13 \\ 10 & -4 & 8 & 2 & -18 \\ 14 & -1 & -5 & 6 & -9 \end{bmatrix}$

Task 1

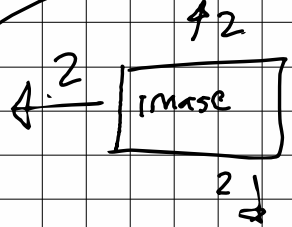
b) Which layer relax the sensitivity to translational variations in the NN

Use the same filter to look for the same features everywhere. Pooling summarises the features present in a region and not precisely positioned features. Used in combination with convolution to find features anywhere.

i) Max Pooling

c) Stride of: 1
Kernel size: 5×5 and 6 filters
Want conv layer to have (Height × W) = input image (Hin × W)
How much padding?

2 on each side, top, bottom



2 from each side removed each time. 6 filters
⇒ 6 outputs

image 3x512x512

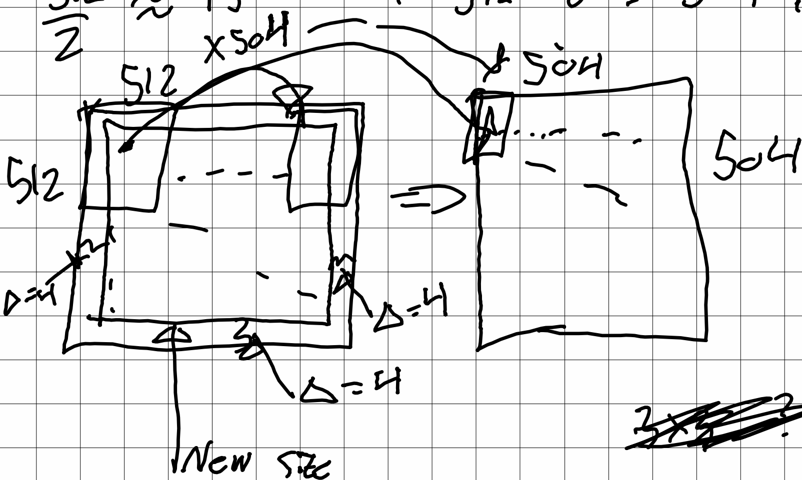
Two convolutional layers

d) • First layer: - features maps 504 x 504
 - 12 feature maps

• No padding, stride of 1, square kernel of odd num
What are the spatial dimensions of these kernels?

8 ← on each side x height

$\frac{512}{2} \approx 43$, $512 - 8 = 504$, $\frac{504}{12} = 42$



~~3x3?~~ Stride on pix her side, skal stride 4

In order to "remove" 4 on each side the kernel must be 9x9 because no padding is used

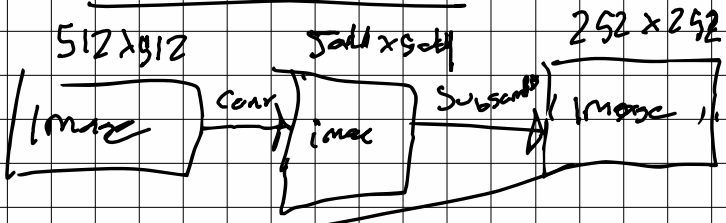
Kernel: 9x9

e) Subsampling after first convolutional layer.

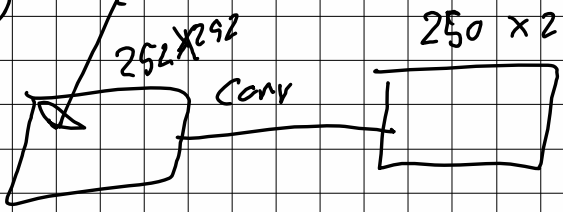
- Neighborhoods of : 2x2
- Stride : 2

What are the spatial dimensions of the pooled feature maps in the first pooling layer?

504/2 = 252



f)



Kernel 3x3, no Padding, Stride : 1

Feature maps is 250x250 with 4 feature maps

9) Convolutional layers: - 5x5 filter
- Padding 2
- Stride 1

MaxPool2D: - Stride 2
- Kernal 2x3

Input: 

$(F \times F \times Din + 1)$ × K for conv

x3

Num Parameters = Num weights + Num biases

Layer:

1: $((5 \times 5 \times 3) + 1) \cdot 32 = 2432$ — 32×32

2: $(5 \times 5 \times 32 + 1) \cdot 64 = 5126$ — 16×16

3: $(5 \times 5 \times 64 + 1) \cdot 128 = 204928$ — 8×8

4: $(128 \times 4 \times 4 + 1) \cdot 64 = 131136$ — 4×4

5: $(64 + 1) \times 10 = 650$

$= 390410$

Assignment 3 Report

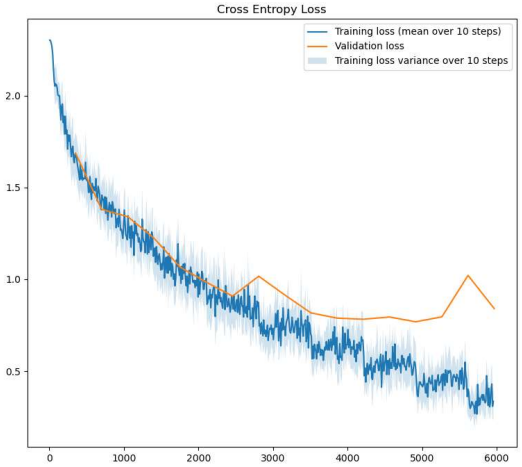
Task 1

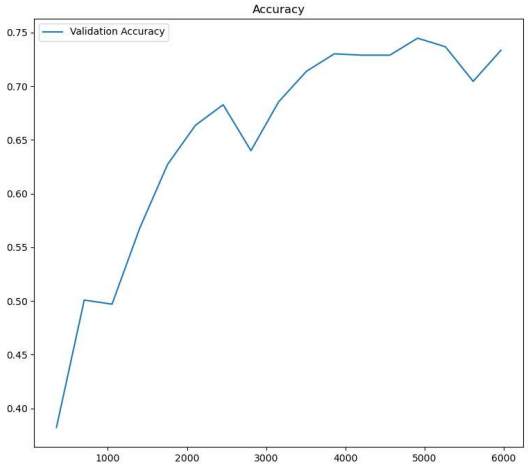
This is appended as scanned handwriting

Task 2

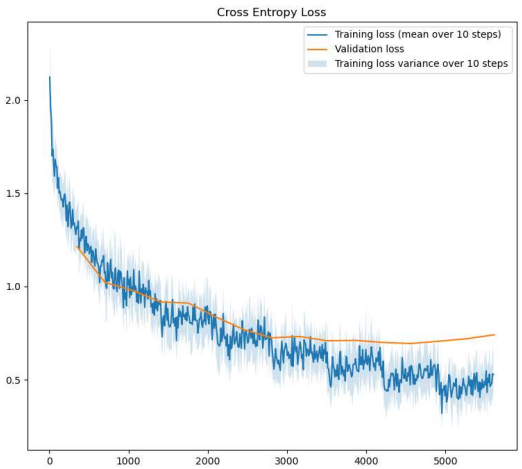
Task 2b)

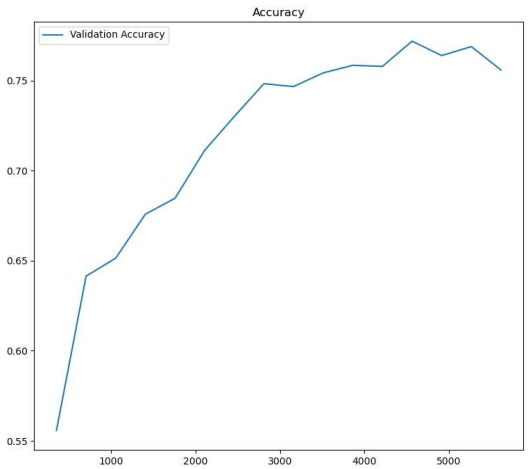
Train acc ca 0.8!





Task 3

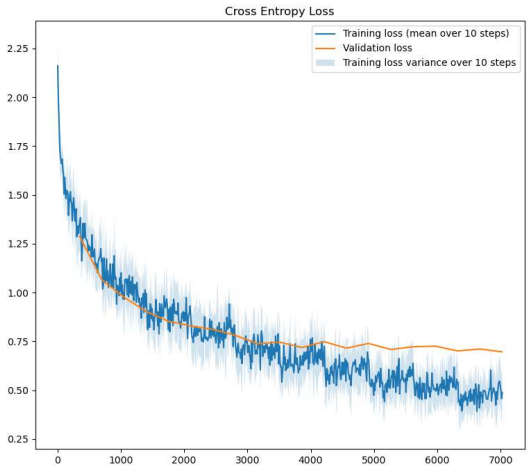


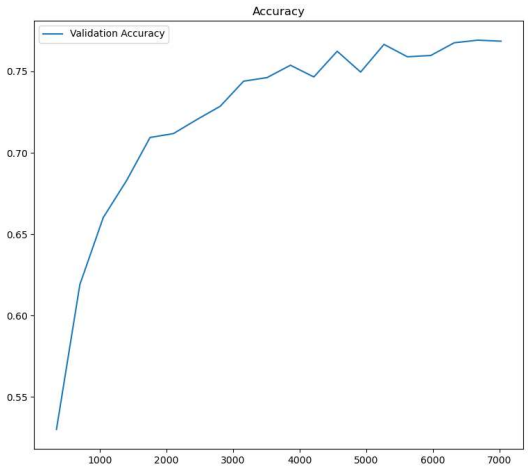


Taask 3a) The first improved model introduces BatchNormmilization between the covulutional and maxpool

layer, and also uses a smaller kernal of 4x4 instead of 5x5. The 2nd improved model uses the sigmoid activation function for the output layer. The traning time is alot longer since it runs for all 10 epochs. As these were enogh to achive the task criteria, but bot at all any good. I Will explain the network architecture of the netowrk in e!

Task 3b) Table of accuracy and loss of both models. And also a plot of the best model.





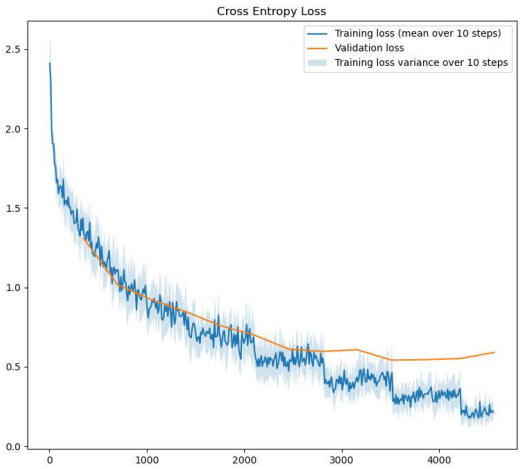
Task 3c) What worked: One size smaller kernal and Sigmoid activation function for output layer What didnt work: Sigmoid for all the layers

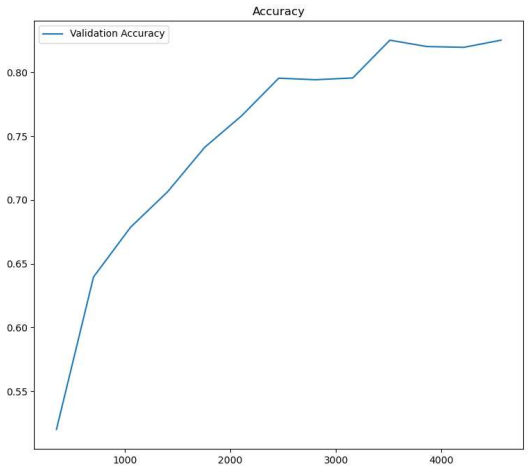
The most amount of work is redesinging the structure but is needed to achive 80% so doing that for task e :)

Task 3d) The plot above with the sigmoid activation function for the output layer

Task 3e) Using ReLU in convulutional layers and sigmoid on output layer. Kernal of 4. Improving this by using adam optimizer with momentum = 0.9.

Redesining the structure to to the propes structure from the assignemtn text: (conv-relu-conv-relu-pool)x3 → (affine)x2 → softmax. Layer 4 and 5 the same, with sigmoid funciton. Final test accuracy is 0.825!





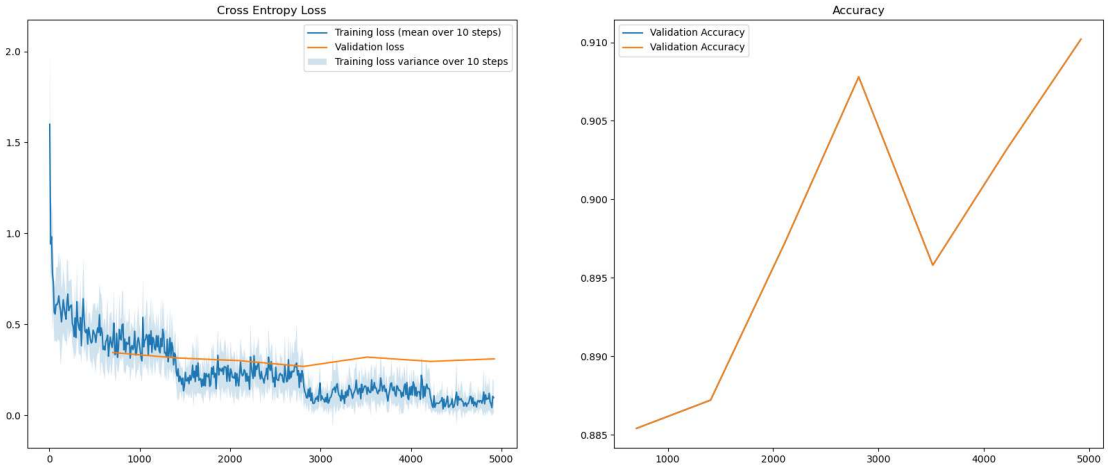
f) We observe that the training loss keeps decreacing even tho the validation loss flattens out. THis could be

because the model is overfitting and therefor not improving on unseen data that it is not trained on.

Task 4

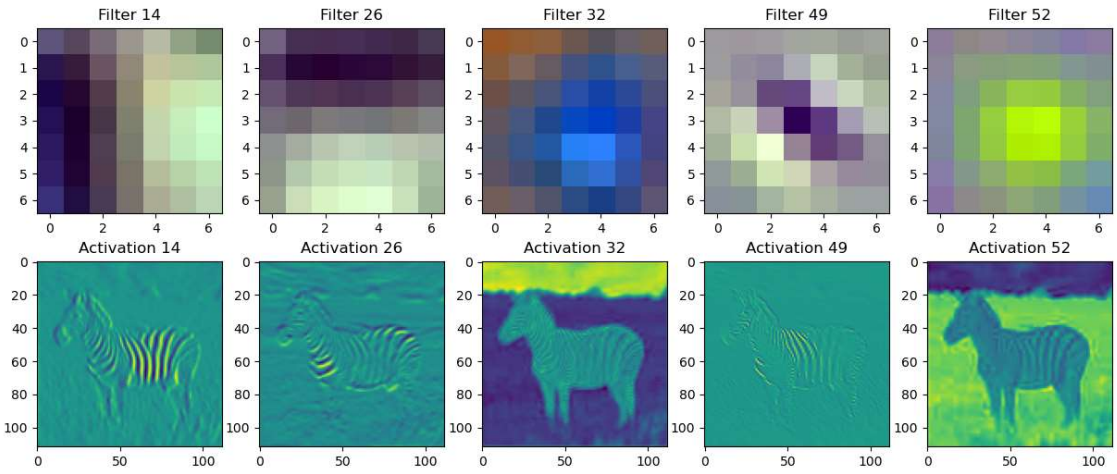
Task 4a)

final test accuracy = 0.9



Task 4b)

This was kinda hard to se but here are my guesses based on structure of the filters. First one is horizontal, next is vertical. The last three is kinda hard to guess, but i think one is contrast and one is edge filtering maby? Maby four is sharpness?



Task 4c)

I could not do this as the last "Øvingstime" was used to complete the other tasks

In [ ]:

```
```