Comparing CNNs for Smile Detection

Tabib Wasit Rahman

Motivation

- CNNs are not very explainable
- Try to gain insights into the performance of CNN by visualizing the trained kernels of the convolution network
- Aim to explain the inner workings/intuition of the model
- What happens when we vary kernel size?
- What happens when we vary number of filters?
- What happens if we do data augmentation?

Dataset: CelebA

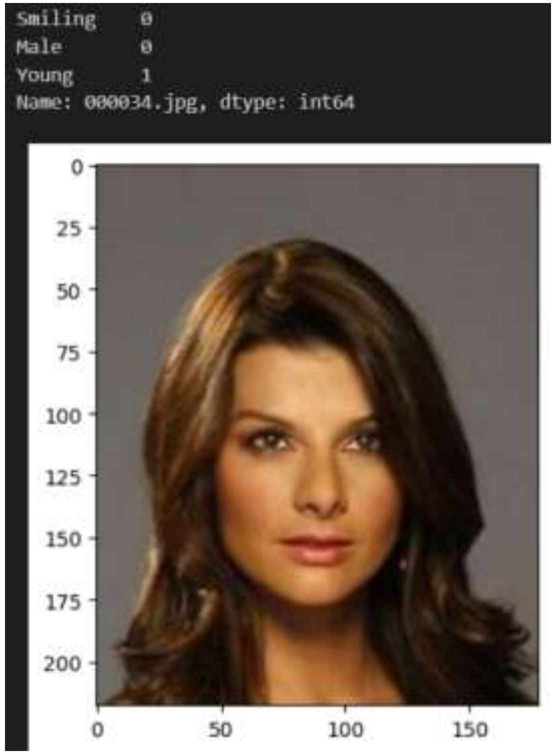
- 202,599 face images
- 10,177 unique identities
- 40 binary attributes:

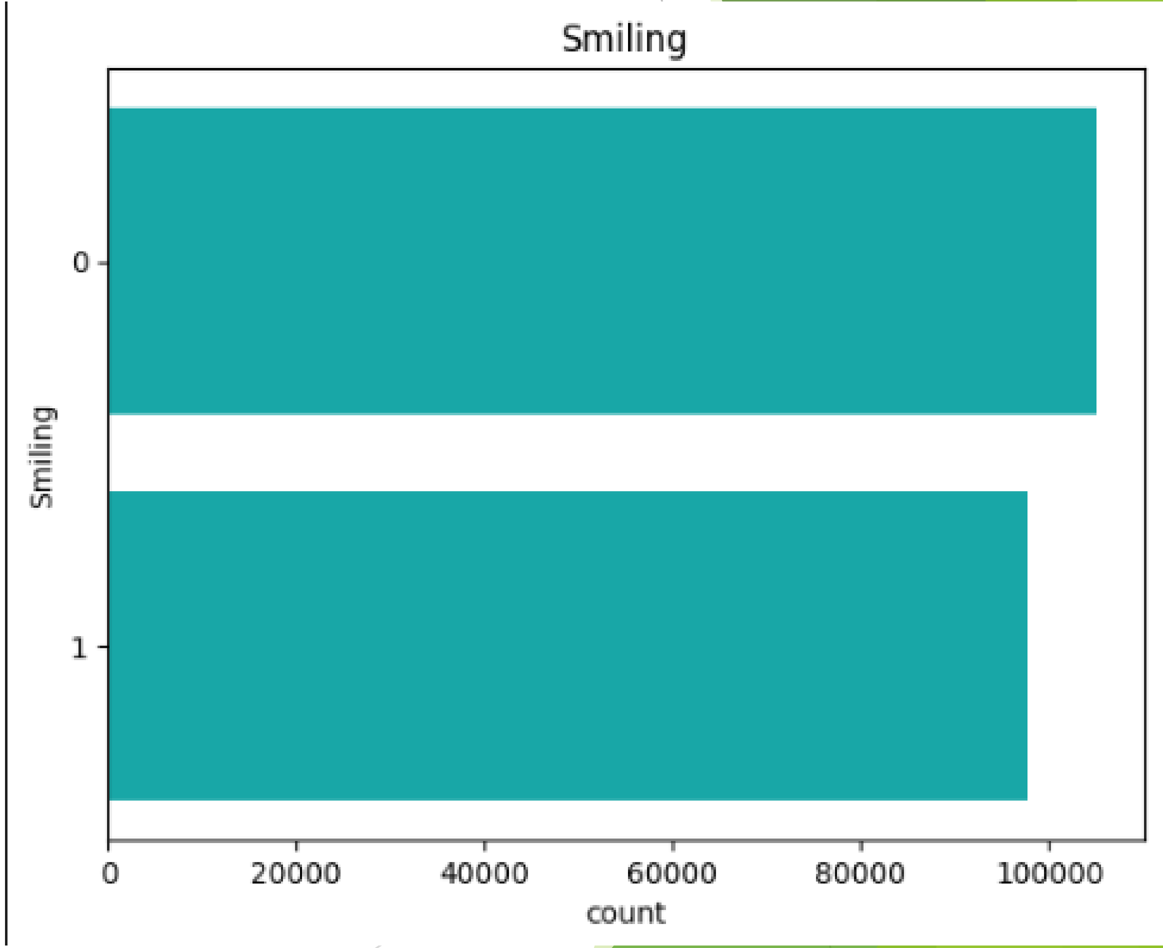
```
30 Sideburns
31 Smiling
32 Straight_Hair
33 Wavy_Hair
34 Wearing_Earrings
35 Wearing_Hat
36 Wearing_Lipstick
37 Wearing_Necklace
38 Wearing_Necktie
39 Young
```

- The dataset is partitioned into training, validation, and testing sets
  - 162,770 training images
  - 19,867 validation images
  - 19,962 testing images
- We pick a subset of images:
  - 10,000 training images
  - 2,000 validation images
  - 2,000 testing images

The data

- Images are 178x218 pixels





Training

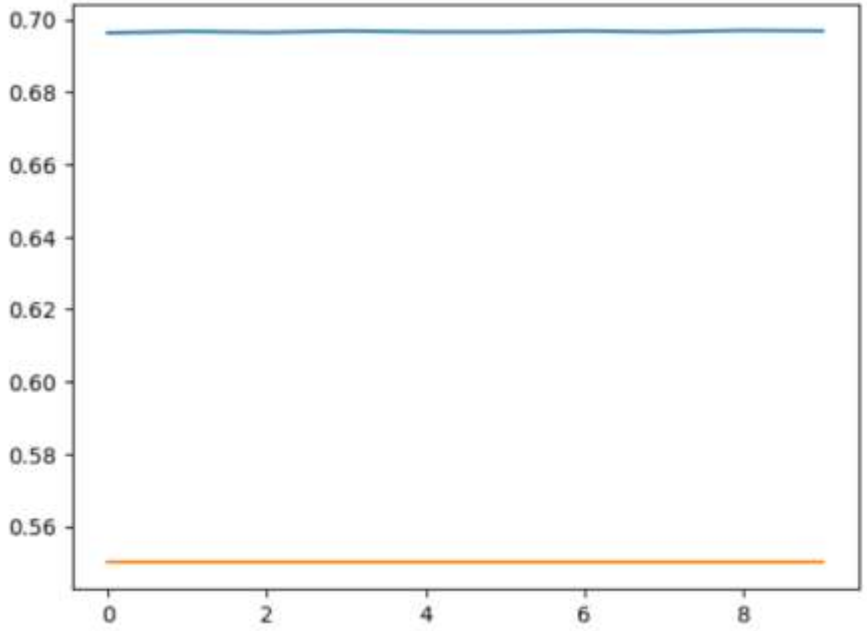
- 10 epochs, with early stopping
- SGD optimizer,
  - Learning rate = 0.001
  - Momentum = 0.9
- Two 2D convolution layers with ReLU activation
- Followed by 2x2 max pooling
- 3 fully connected linear layers
- BCELoss()

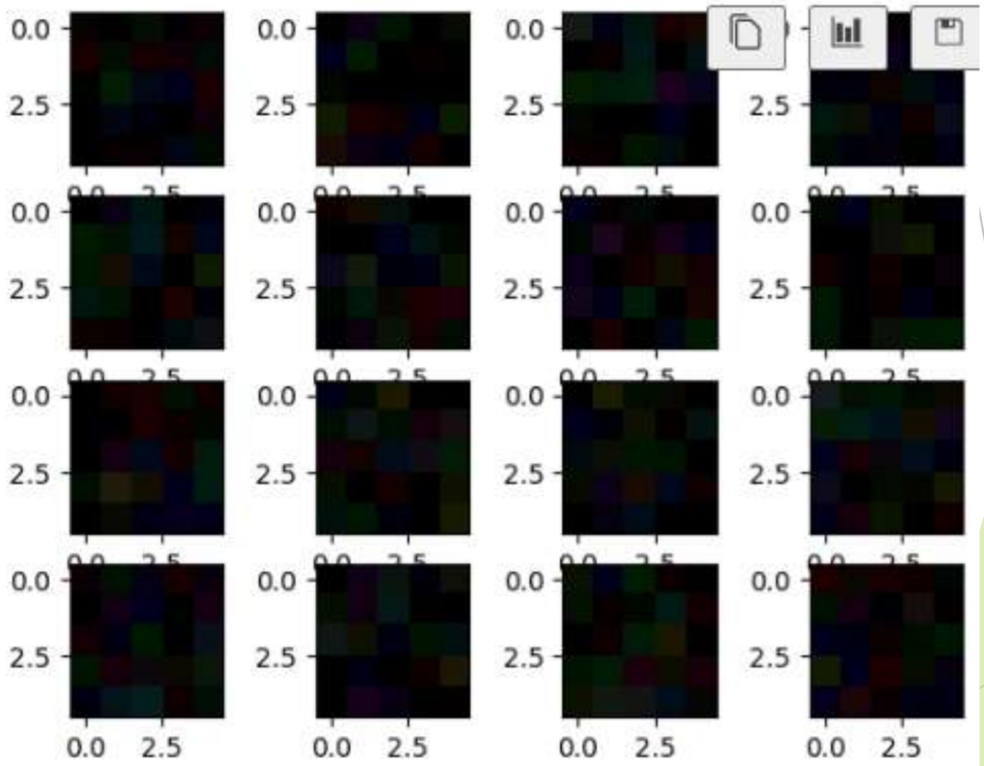
- What happens when we vary kernel size?
- What happens when we vary number of filters?
- What happens if we do data augmentation?

First model – 5x5 kernel

```
==========================================================================
Layer (type:depth-idx)                            Output Shape
==========================================================================
Net                                               [16]
├─Conv2d: 1-1                                     [16, 16, 214, 174]
├─MaxPool2d: 1-2                                  [16, 16, 107, 87]
├─Conv2d: 1-3                                     [16, 64, 103, 83]
├─MaxPool2d: 1-4                                  [16, 64, 51, 41]
├─Linear: 1-5                                     [16, 128]
├─Linear: 1-6                                     [16, 64]
├─Linear: 1-7                                     [16, 1]
--------------------------------------------------------------------------
```

First model

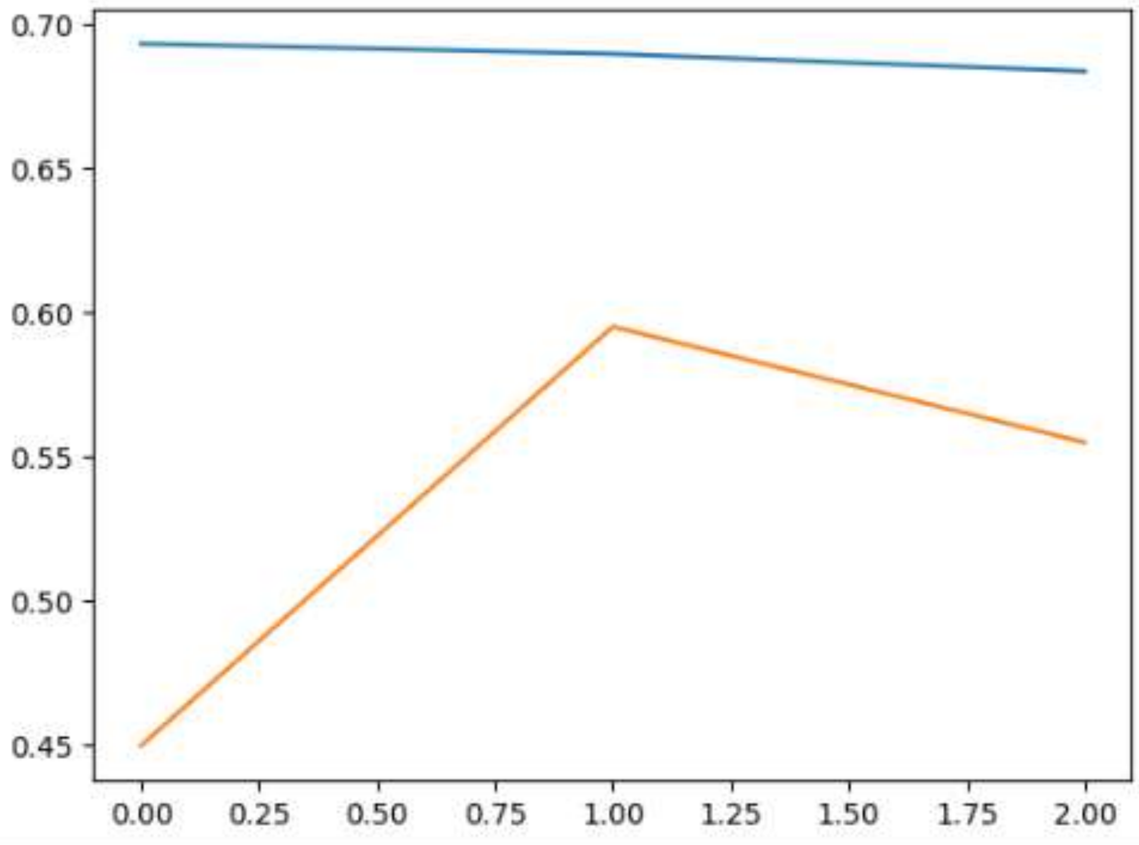


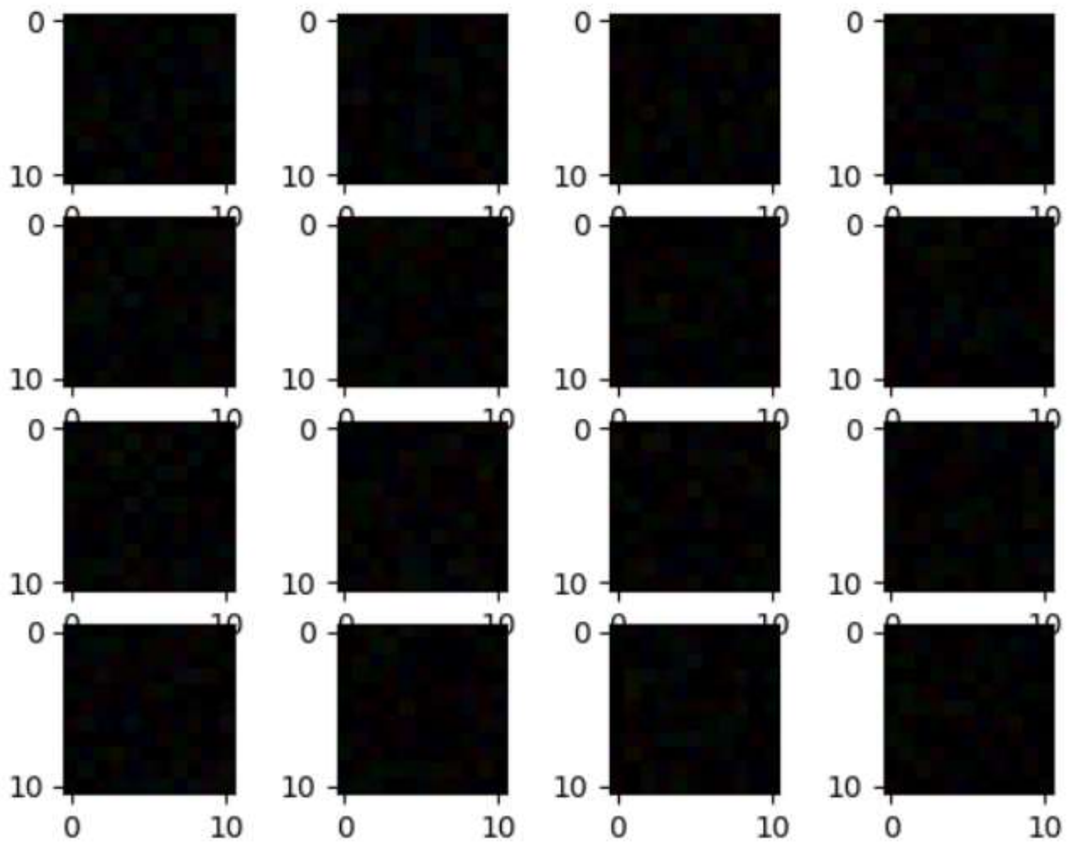


Second model – 11x11 kernel

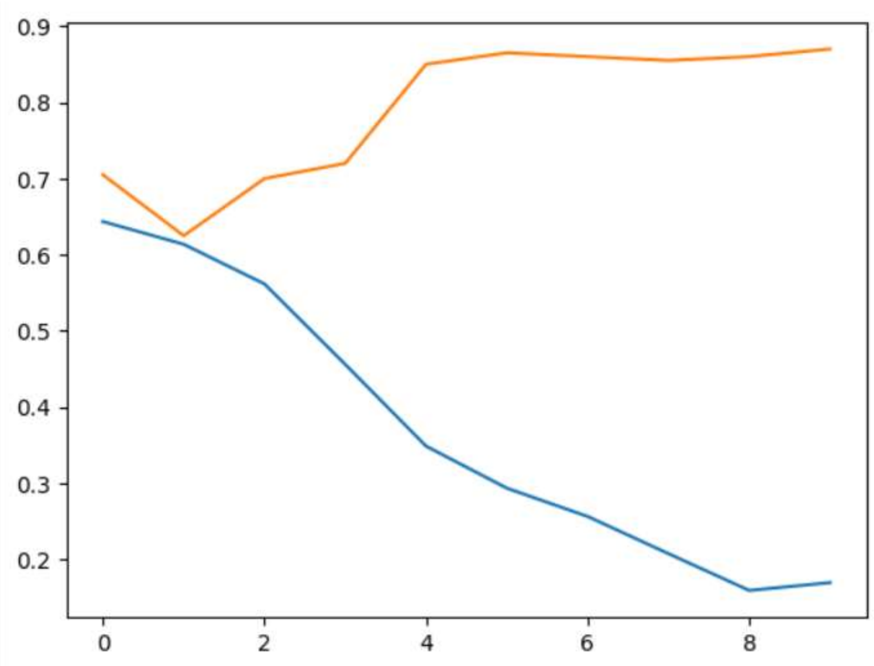
```
==========================================================================================
Layer (type:depth-idx)                             Output Shape
==========================================================================================
Net                                                [16]
├─Conv2d: 1-1                                      [16, 16, 208, 168]
├─MaxPool2d: 1-2                                   [16, 16, 104, 84]
├─Conv2d: 1-3                                      [16, 32, 94, 74]
├─MaxPool2d: 1-4                                   [16, 32, 47, 37]
├─Linear: 1-5                                      [16, 128]
├─Linear: 1-6                                      [16, 64]
├─Linear: 1-7                                      [16, 1]
==========================================================================================
```

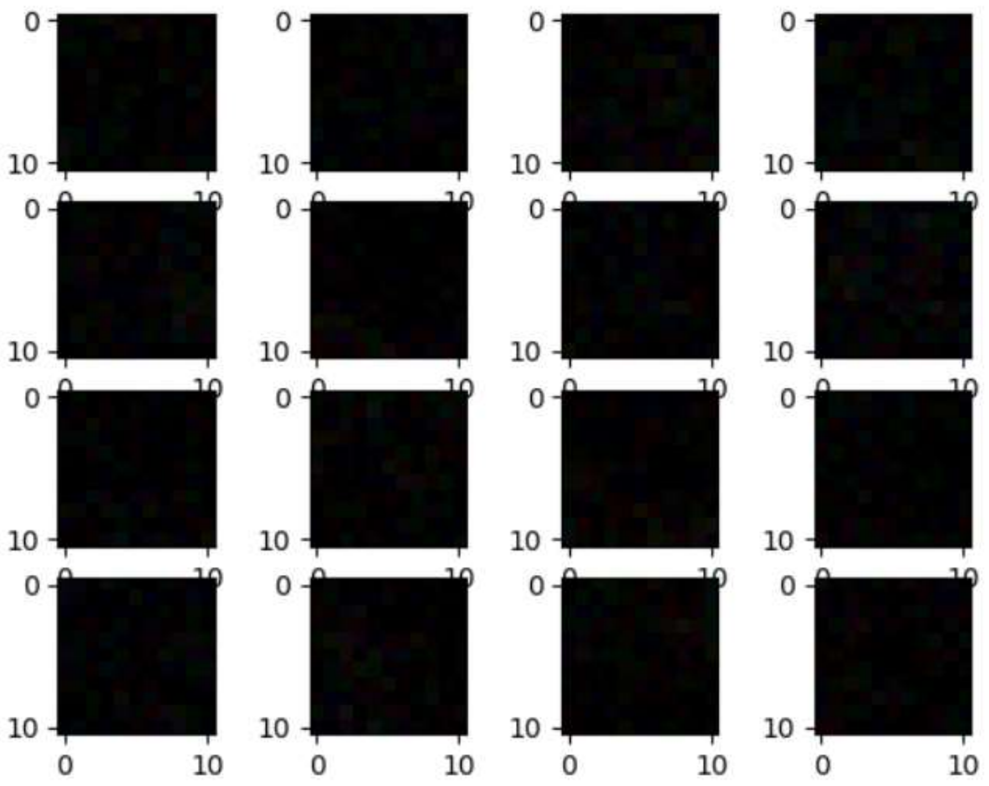
Second model





Remove early stopping

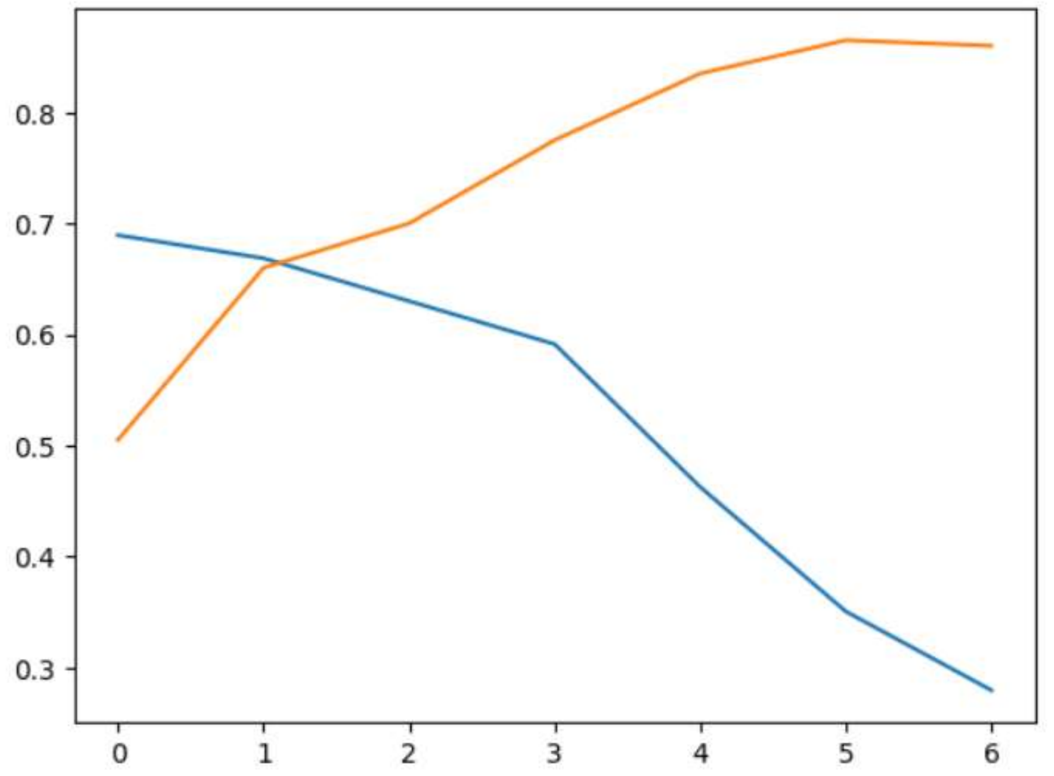


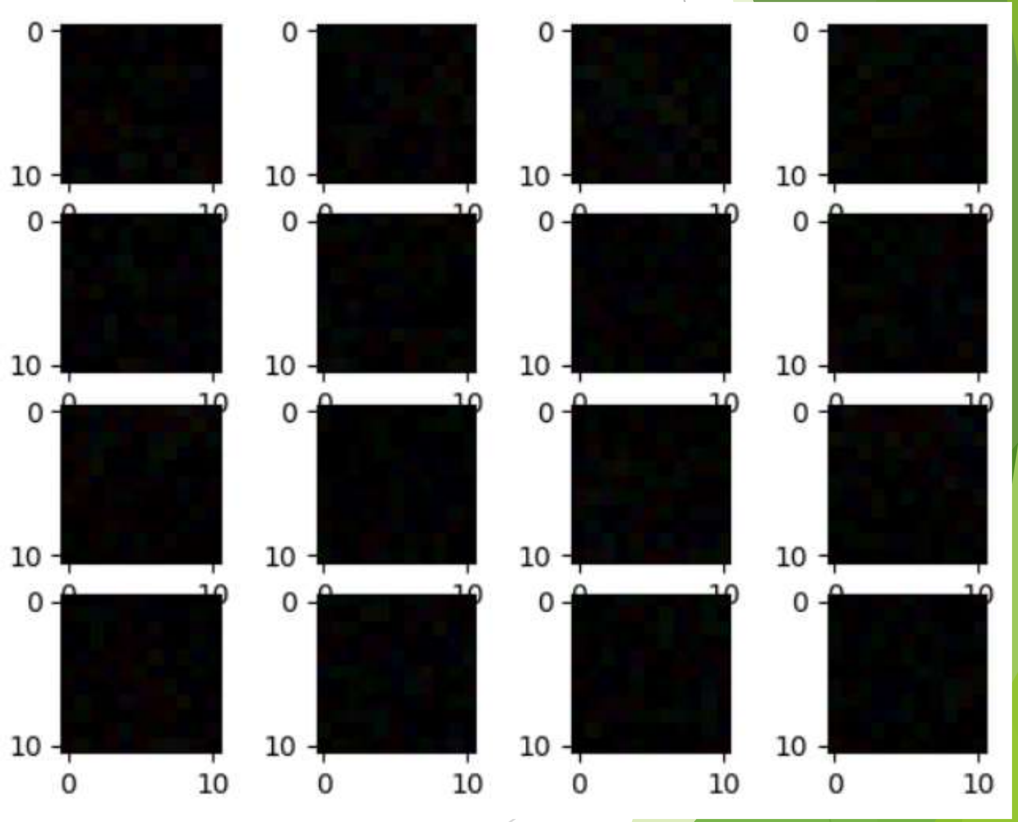


Third model: with normalization

```
==========================================================================
Layer (type:depth-idx)                        Output Shape
==========================================================================
Net                                           [16]
├─Conv2d: 1-1                                 [16, 16, 208, 168]
├─MaxPool2d: 1-2                              [16, 16, 104, 84]
├─Conv2d: 1-3                                 [16, 64, 94, 74]
├─MaxPool2d: 1-4                              [16, 64, 47, 37]
├─Linear: 1-5                                 [16, 128]
├─Linear: 1-6                                 [16, 64]
├─Linear: 1-7                                 [16, 1]
--------------------------------------------------------------------------
```

Third model: with normalization



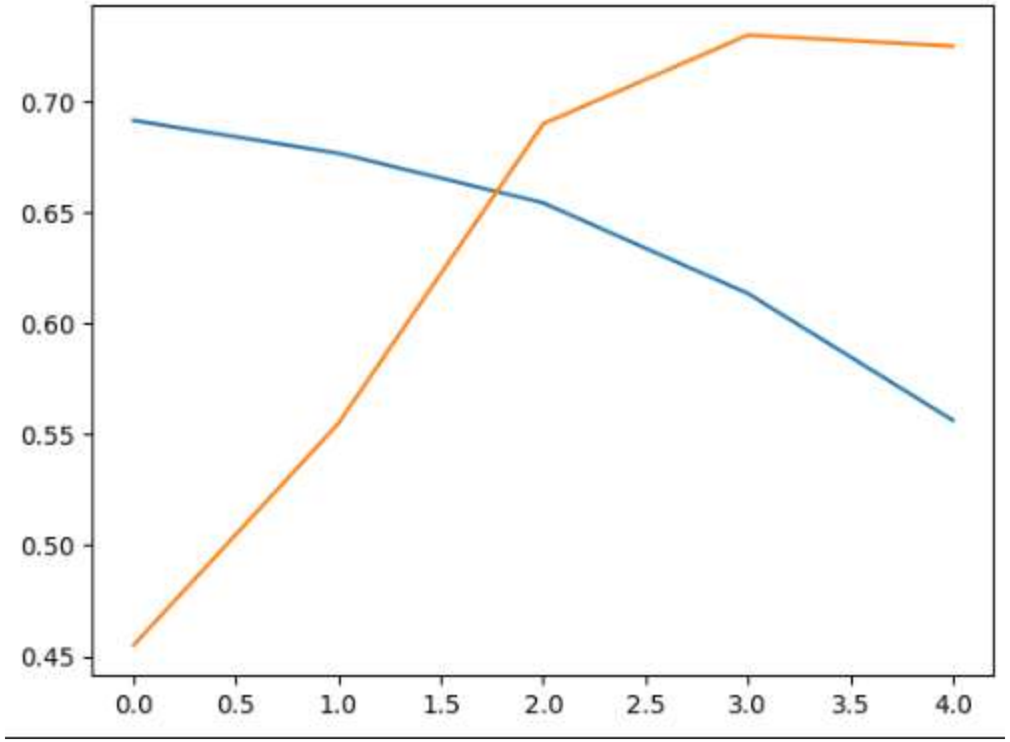


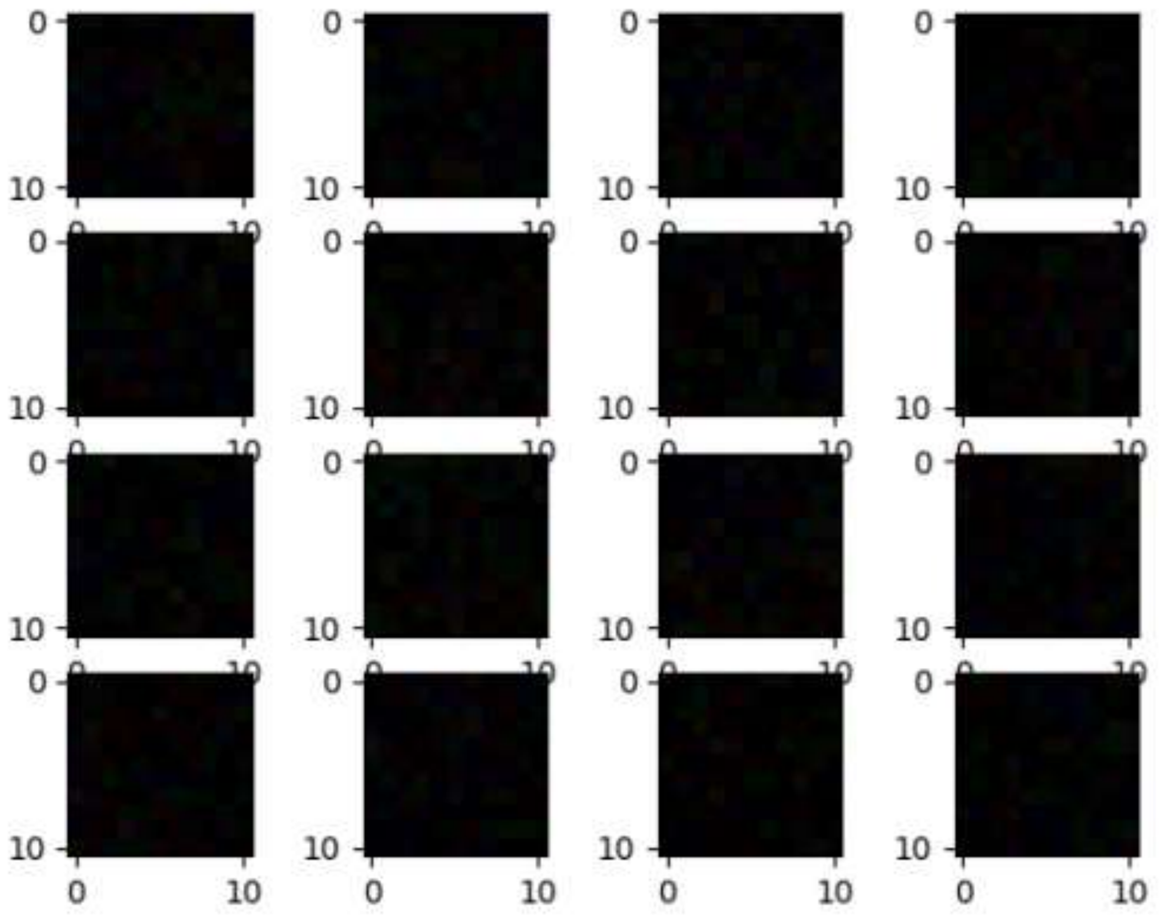
Fourth model: data augmentation

- Random Affine
  - Rotation = 30 degrees
  - Translation = 20%
  - Scale = 70%
  - Shear = 20%
- Random Horizontal Flip

```
==========================================================================
Layer (type:depth-idx)                   Output Shape
==========================================================================
Net                                      [16]
├─Conv2d: 1-1                            [16, 16, 208, 168]
├─MaxPool2d: 1-2                         [16, 16, 104, 84]
├─Conv2d: 1-3                            [16, 64, 94, 74]
├─MaxPool2d: 1-4                         [16, 64, 47, 37]
├─Linear: 1-5                            [16, 128]
├─Linear: 1-6                            [16, 64]
├─Linear: 1-7                            [16, 1]
==========================================================================
```

Fourth model: data augmentation





Conclusion

- Kernel size improves accuracy
- The number of filters did not show a significant improvement
- Both kernel size and number of filters affect the training time significantly
- Normalization improves the performance of the model
- Random Affine transformation does not have much significant effect

- Issues:
- Visualization of kernels in not very helpful
  - Need better way to examine kernels
  - Larger kernel size
- Training time is high (requires lot of GPU resources)
- Small model (use more convolution layers for a better performance)

Thank you!