



REPORT FOR WEEK 6

MOHAMMADREZA KAZEMI

MAJOR IMPROVEMENTS



The position of generators are not fixed anymore

Convolutional NN was implemented

ResNet was implemented

THE POSITION OF GENERATORS ARE NOT FIXED ANYMORE

- Previously, the Position of generators were fixed to bus #1 to 3
- Now, the position for demand and supply is randomly generated

| Bus Data | | | | | | |
|----------|---------|----------|------------|----------|--------|----------|
| Bus # | Voltage | | Generation | | Load | |
| | Mag(pu) | Ang(deg) | P (MW) | Q (MVar) | P (MW) | Q (MVar) |
| 1 | 1.103 | 12.067* | - | - | - | - |
| 2 | 1.000 | 0.000 | -69.00 | -10.00 | - | - |
| 3 | 1.020 | 12.257 | - | - | 0.00 | 40.00 |
| 4 | 1.103 | 12.067 | 87.00 | 0.00 | - | - |
| 5 | 1.099 | 11.920 | - | - | - | - |
| 6 | 1.043 | 12.257 | 108.00 | 0.00 | - | - |
| 7 | 1.013 | 6.289 | - | - | 16.00 | 30.00 |
| 8 | 1.008 | 2.473 | - | - | 104.00 | 31.00 |
| 9 | 1.085 | 8.768 | - | - | - | - |
| Total: | | | 126.00 | -10.00 | 120.00 | 101.00 |

CONVOLUTIONAL NN WAS IMPLEMENTED

- A Convolutional NN was implemented
- CNN outperformed Fully Connected NN

input

output

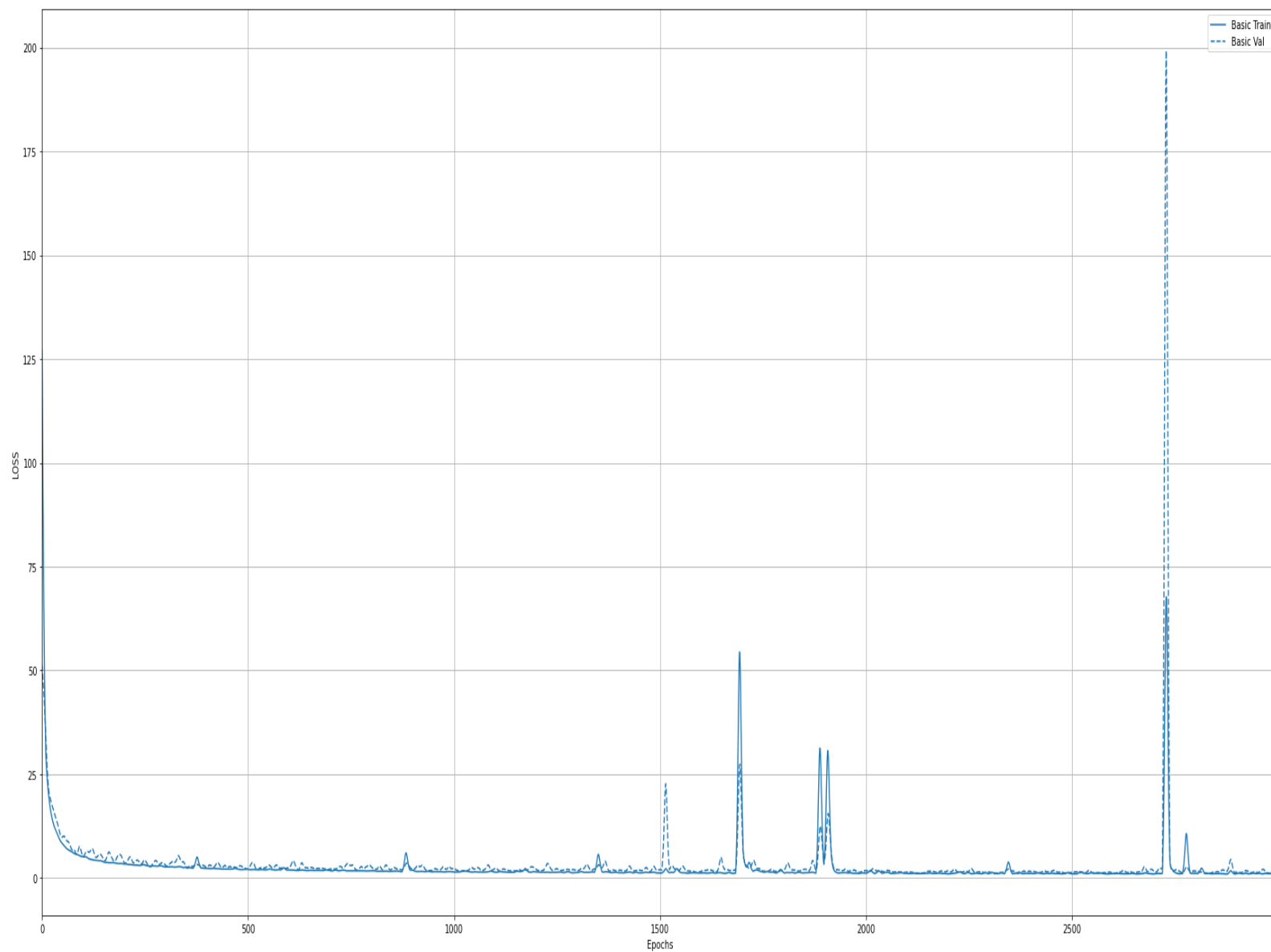
| Layer (type) | Output Shape | Param # |
|--------------------------------|--------------------|---------|
| conv2d_13 (Conv2D) | (None, 9, 12, 128) | 1280 |
| max_pooling2d_7 (MaxPooling2D) | (None, 5, 6, 128) | 0 |
| conv2d_14 (Conv2D) | (None, 5, 6, 128) | 147584 |
| conv2d_15 (Conv2D) | (None, 5, 6, 128) | 147584 |
| conv2d_16 (Conv2D) | (None, 5, 6, 128) | 147584 |
| max_pooling2d_8 (MaxPooling2D) | (None, 3, 3, 128) | 0 |
| conv2d_17 (Conv2D) | (None, 3, 3, 128) | 147584 |
| max_pooling2d_9 (MaxPooling2D) | (None, 2, 2, 128) | 0 |
| conv2d_18 (Conv2D) | (None, 2, 2, 128) | 147584 |
| flatten_2 (Flatten) | (None, 512) | 0 |
| dense_8 (Dense) | (None, 128) | 65664 |
| dense_9 (Dense) | (None, 128) | 16512 |
| dense_10 (Dense) | (None, 128) | 16512 |
| dense_11 (Dense) | (None, 18) | 2322 |
| Total params: 840,210 | | |
| Trainable params: 840,210 | | |
| Non-trainable params: 0 | | |

COMPARISONS BETWEEN FCNN AND CNN

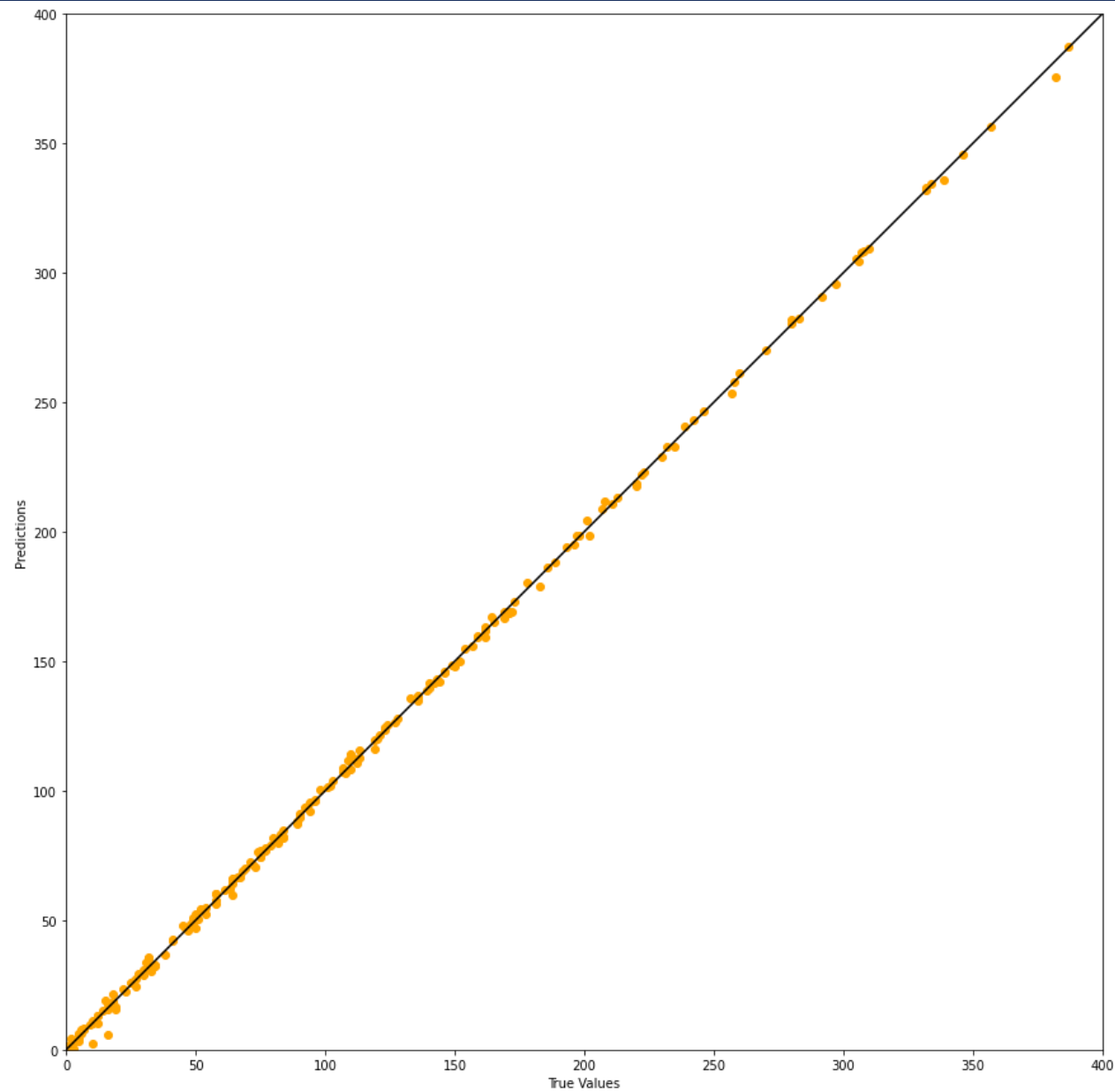
- Mean Absolute Error for FCNN with fixed generation and load position: 0.85
- Mean Absolute Error for FCNN with variable generation and load position: NaN (No experiment was done)
- Mean Absolute Error for CNN with fixed generation and load positions: 0.35
- Mean Absolute Error for CNN with variable generation and load positions: 0.52
- I believe it is safe to say that CNN outperformed FCNN

SOME NUMBERS FOR CNN

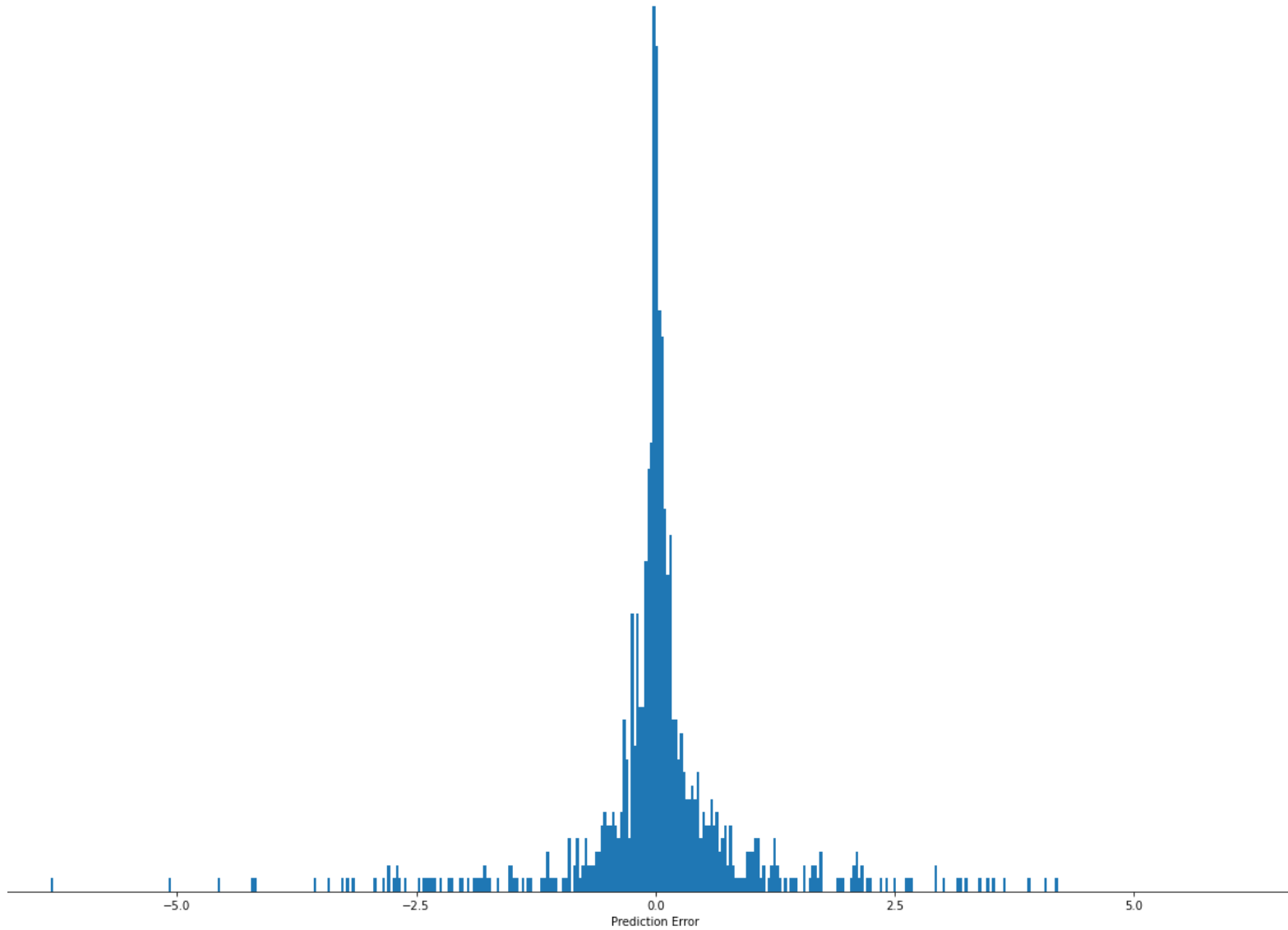
- Training:
 - loss: 0.9694 - accuracy: 0.9928 - mae: 0.5137 - mse: 0.9694
- Testing:
 - loss: 1.0681 - accuracy: 1.0000 - mae: 0.5234 - mse: 1.0681



LOSS VS. EPOCHS FOR CNN



TRUE VALUE
VS.
PREDICTION



PREDICTION
ERROR VS.
COUNT

TEST CASE I (CNN PREDICTION)

| Bus # | Generation P (True Value) | Generation P (Prediction) | Generation Q (True Value) | Generation Q (Prediction) |
|-------|------------------------------|------------------------------|------------------------------|------------------------------|
| 0 | -124.0 | -121.693504 | 30.0 | 30.201191 |
| 1 | 0.0 | -0.284206 | 0.0 | 0.907698 |
| 2 | 0.0 | 0.535467 | 0.0 | 0.606632 |
| 3 | 0.0 | -0.372101 | 0.0 | -0.136823 |
| 4 | 92.0 | 92.520256 | 0.0 | -0.042954 |
| 5 | 176.0 | 175.870483 | 0.0 | -0.110085 |
| 6 | 0.0 | 0.881297 | 0.0 | -0.027453 |
| 7 | 0.0 | -0.302840 | 0.0 | 0.061926 |
| 8 | 0.0 | 0.365462 | 0.0 | -0.041741 |

TEST CASE 2 (CNN PREDICTION)

| Bus # | Generation P (True Value) | Generation P (Prediction) | Generation Q (True Value) | Generation Q (Prediction) |
|-------|------------------------------|------------------------------|------------------------------|------------------------------|
| 0 | 170.0 | 169.818253 | 67.0 | 66.918182 |
| 1 | 0.0 | 0.195781 | 0.0 | -0.370815 |
| 2 | 91.0 | 89.946724 | 15.0 | 14.529804 |
| 3 | 0.0 | -0.208835 | 0.0 | 0.058383 |
| 4 | 0.0 | 0.099604 | 0.0 | 0.026538 |
| 5 | 0.0 | 0.288580 | 0.0 | 0.073228 |
| 6 | 99.0 | 100.341103 | 0.0 | 0.005307 |
| 7 | 0.0 | 0.622381 | 0.0 | -0.046634 |
| 8 | 0.0 | 0.025451 | 0.0 | 0.075404 |

TEST CASE 3 (CNN PREDICTION)

| Bus # | Generation P (True Value) | Generation P (Prediction) | Generation Q (True Value) | Generation Q (Prediction) |
|-------|------------------------------|------------------------------|------------------------------|------------------------------|
| 0 | 0.0 | -0.401420 | 0.0 | -0.176424 |
| 1 | 0.0 | -0.301592 | 0.0 | 0.460730 |
| 2 | -45.0 | -45.227913 | 89.0 | 89.443909 |
| 3 | 0.0 | -0.561829 | 0.0 | 0.044143 |
| 4 | 0.0 | -0.152657 | 0.0 | 0.002158 |
| 5 | 0.0 | 0.348304 | 0.0 | 0.050586 |
| 6 | 11.0 | 12.743074 | 0.0 | 0.123529 |
| 7 | 226.0 | 224.611450 | 0.0 | -0.065079 |
| 8 | 0.0 | 0.313832 | 0.0 | 0.166100 |

RESNET

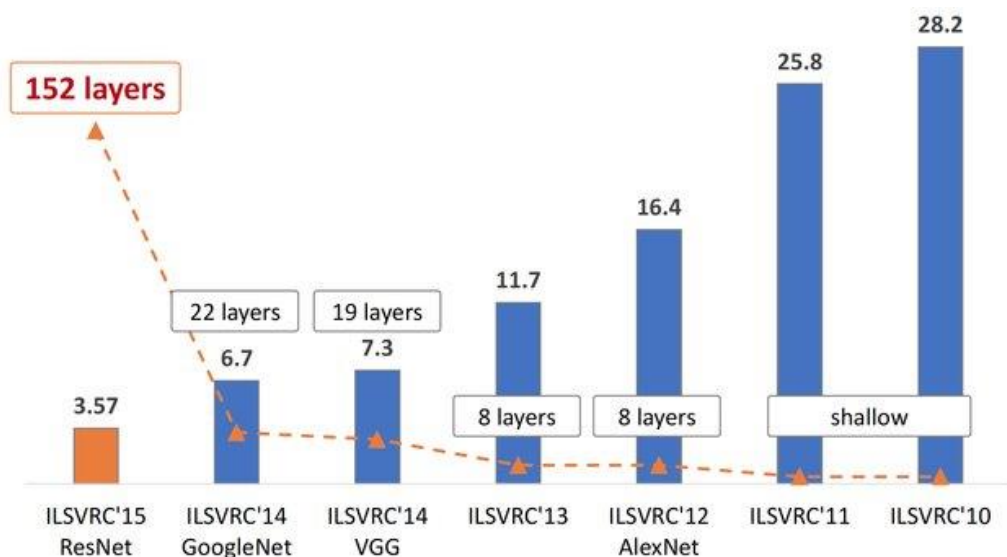


Fig. 1. The evolution of the winning entries on the ImageNet Large Scale Visual Recognition Challenge from 2010 to 2015. Since 2012, CNNs have outperformed hand-crafted descriptors and shallow networks by a large margin. Image re-printed with permission.

Source: https://www.researchgate.net/publication/321896881_Iris_Recognition_with_Off-the-Shelf_CNN_Features_A_Deep_Learning_Perspective/figures?lo=1

- Beside CNN, I implemented ResNet, but unfortunately its implementation was done just last night, and I didn't get a chance to **fully** run it. But over 53 epochs, the result was not promising compared to other models. However, I insist on running it, because it is the winner of "ImageNet Large Scale Visual Recognition Challenge" and it beat other models in every category.



THANK YOU