Street View Housing Numbers

Authors: Aneeka Latif, Antonio Llorens, Indra Sulijoadikusumo, Connor Borrego

Specify

- Explore – digit recognition in real world images (SVHN Dataset)
- Applications – identifying missing postal data in mapping apps as opposed to GPS approximation.
- Challenge - classifying a sequence of numbers as opposed to single digits
- Plan – focus on recognizing them simultaneously.

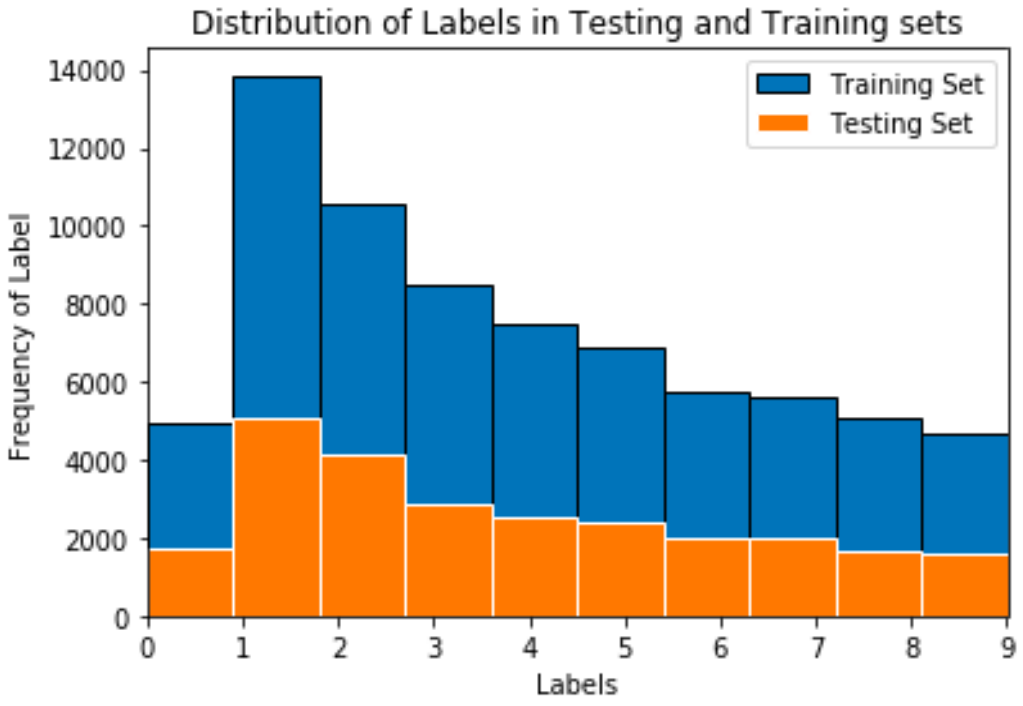
About the Data

- Single Digit Recognition
- MNIST 32x32
  - Training Set:
    - 73257 digits
  - Test Set:
    - 26032 digits
  - Images
    - Single digit
  - Bounding Boxes [Entire Image]
  - Labels
    - Digits labeled 1 – 10
      - 0 is 10

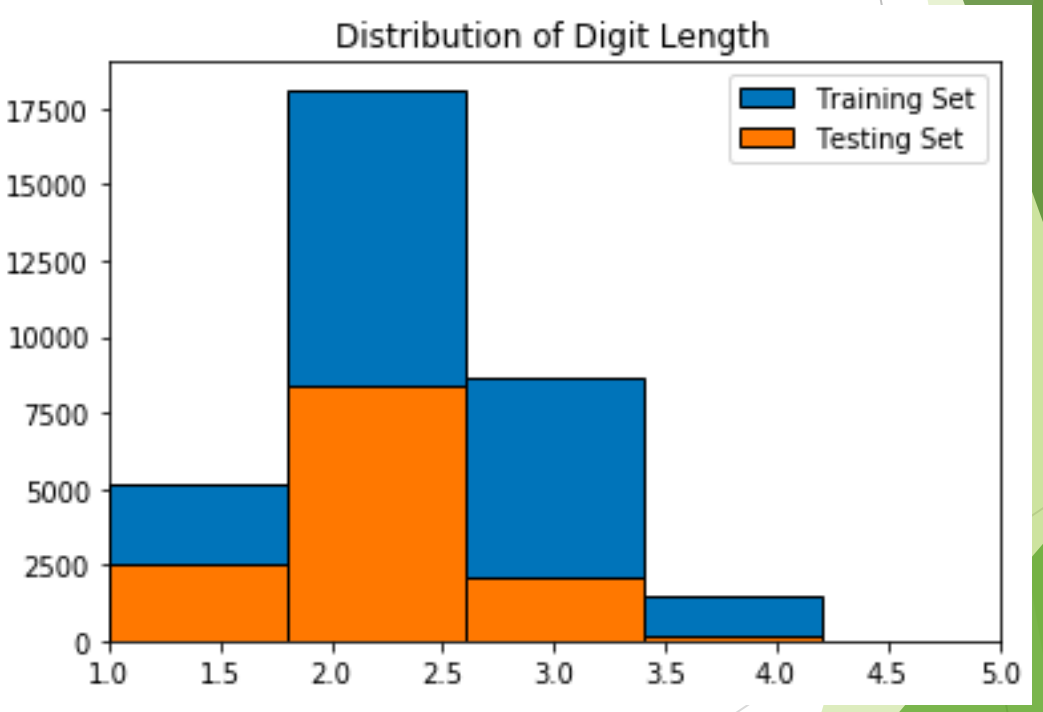
- Multi Digit Recognition
  - Training Set:
    - 33402 images
  - Test Set:
    - 13068 images
  - Images
    - Between 1-6 digits
  - Bounding Boxes [Entire Image + Each Digit]
  - Labels
    - Arrays between 1-6 digits in length
    - Digits labeled 1-10
    - 0 is 10

About the Data

- Single Digit Recognition

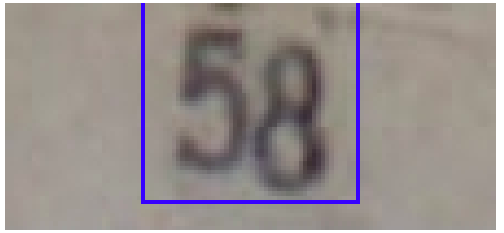


- Multi-Digit Recognition



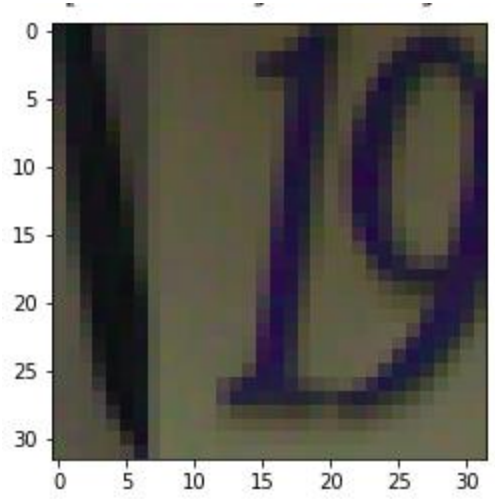
Observe

- Enormous appearance variations in natural images
  - Different fonts
  - Scale
  - Rotations
  - Illumination conditions
- Bounding boxes
  - Entire image & individual digit



Preprocessing - Bounding Boxes

- Single digit
  - MNIST images determined using provided bounding boxes

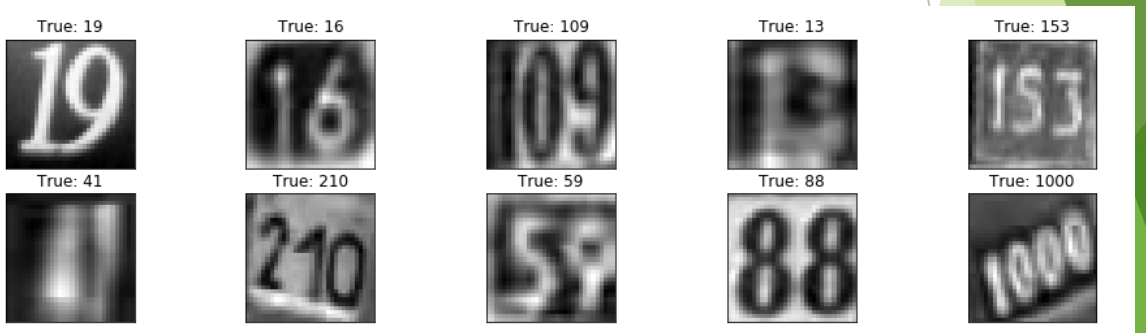


- Multi Digit
  - Increase/decrease the bounding boxes as necessary

Preprocessing

- Single Digit
  - Reshape the input
  - Set label "10" as 0
  - Normalize the images

- Multi-digit
  - Reshape the input
  - Set arrays of length 5 with "10" as a placeholder
  - Convert images to greyscale
  - Resize images
  - Drop poor quality images
  - Drop images with >5 digits

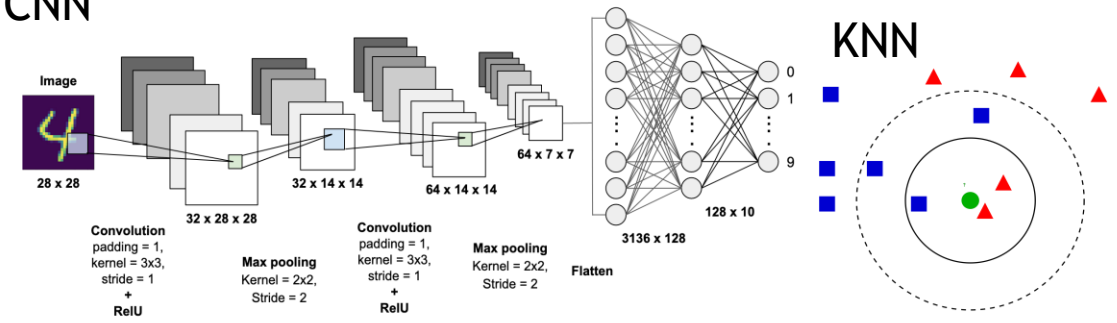


19 -> 1 9 10 10 10
123 -> 1 2 3 10 10
9876 -> 9 8 7 6 10
5 -> 5 10 10 10 10

Analyze - Models

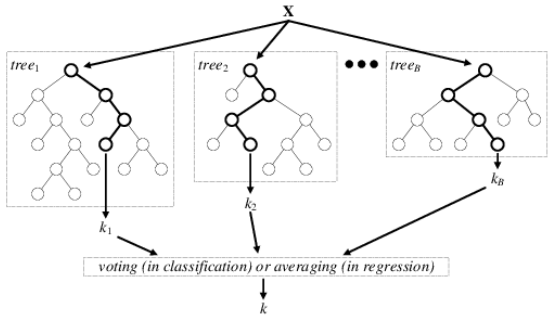
Single

CNN

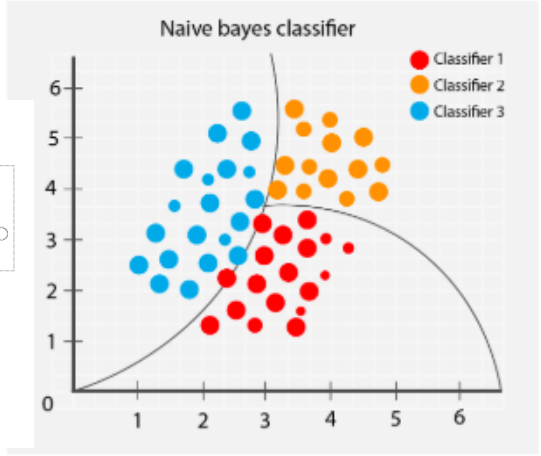


KNN

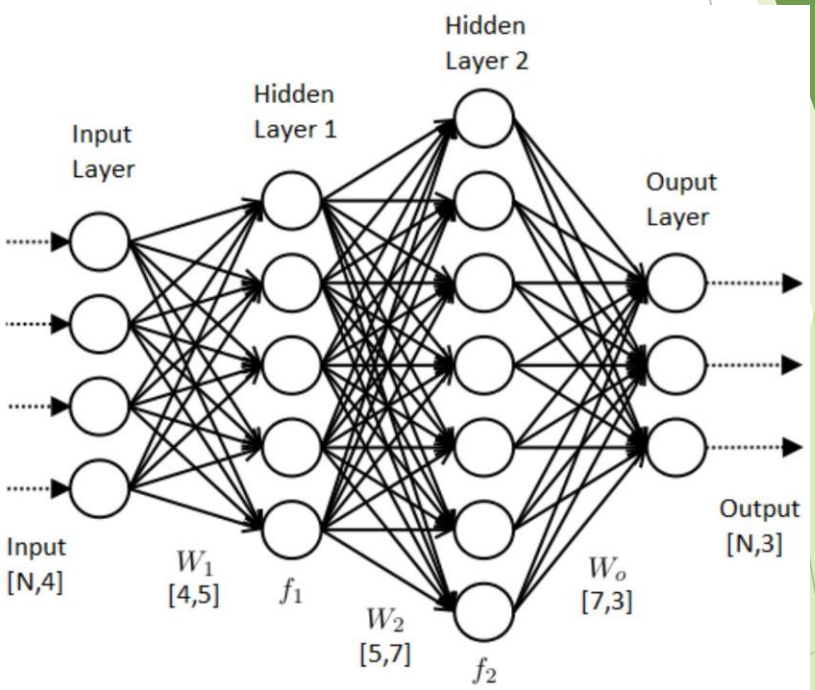
Random Forest



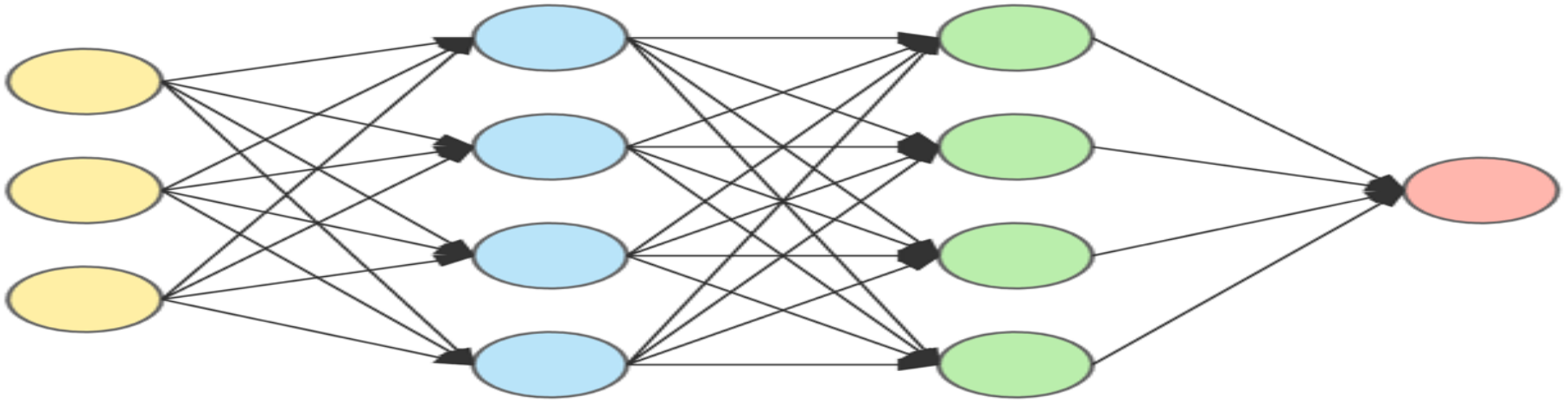
MNB



Multi



Multi Digit – Neural Network Architecture



Subsampling Subsampling Flatten () & Dense

Input Layer

- 2 Conv 2d
- 4 Activation Relu
- 4 Batch normalization
- Maxpooling
- Dropout

Hidden Layer 1

- 2 Conv 2D
- 2 Activation relu
- 2 Batchnormalization
- MaxPooling
- Dropout

Hidden Layer 2

- 2 Conv 2D
- 2 Activation relu
- 2 Batchnormalization
- MaxPooling
- Dropout

Output Layer

- Dense
- Softmax Func
- 5 outputs layer

Compiling Step

- Ndam Optimizer
- Binary_crossentropy
- Acc Metrics

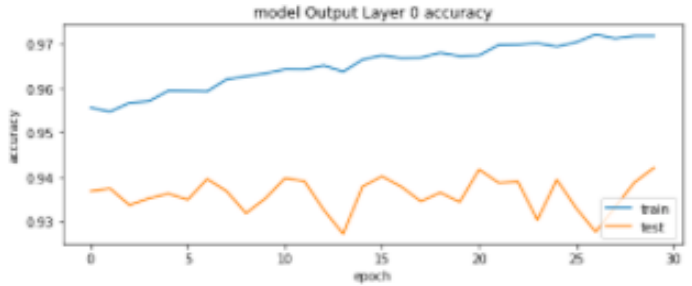
Accuracy

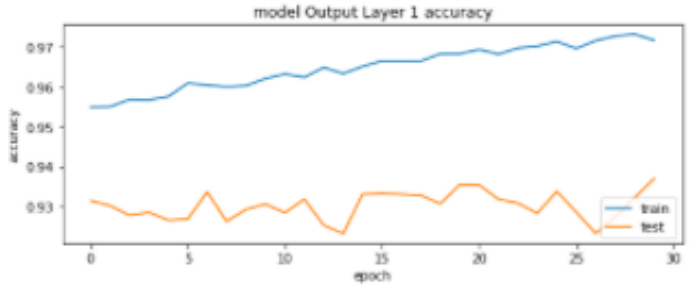
Single Digit

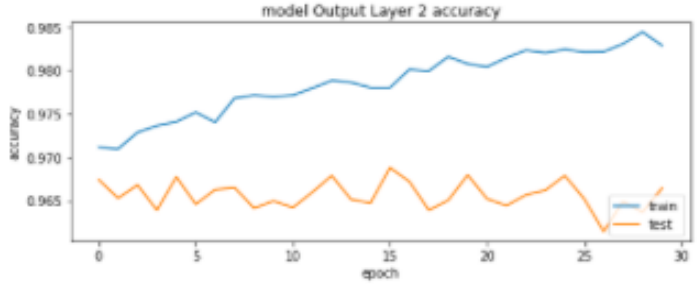
Model	Parameters	Accuracy
CNN	Relu(2), epoch (3)	19%
MNB	Alpha = .8	19%
KMeans 1	Clusters = 1,822	17.1% Train 7.8% Test
KMeans 2	Clusters = 10,000	38.2% Train 11.4% Test
Random Forest	Criterion = entropy	25.93%

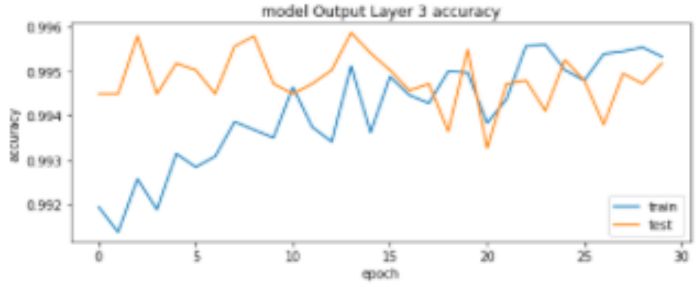
Multi-Digit

Layers	Individual Accuracy	Sequence Accuracy
[Conv2D Maxpooling Dropout] Adam X2 Epoch (30)	91.53%	65.03%
[Conv2D Maxpooling Normalization Dropout] nadam Epoch (30)	96.80%	87.74%



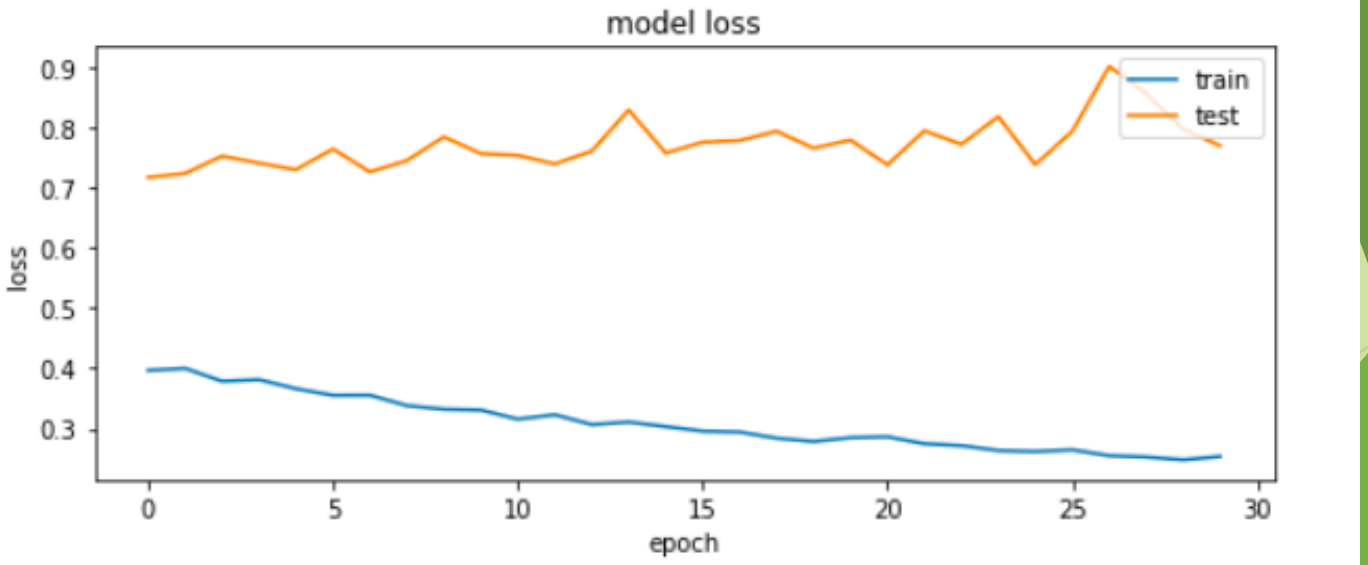






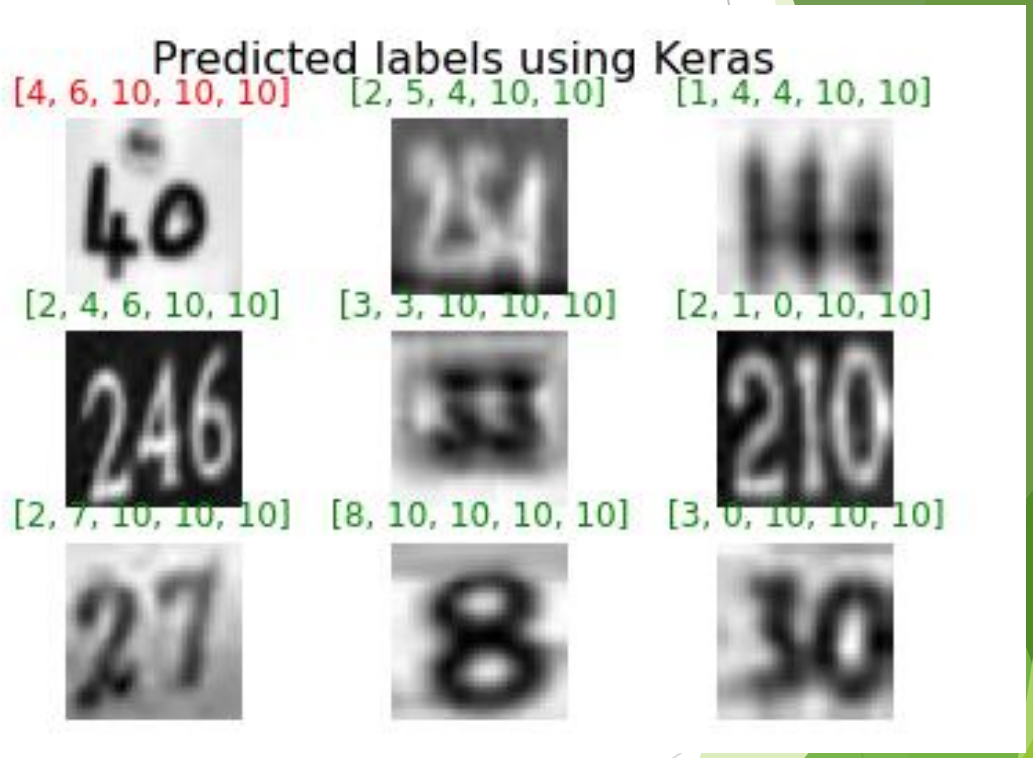
Model Training and validation

- Accuracy Plots for the model's 5 output layers
- Signs of overfitting for ouput layers 0 and 1
- Improvement output layer 3 and 4



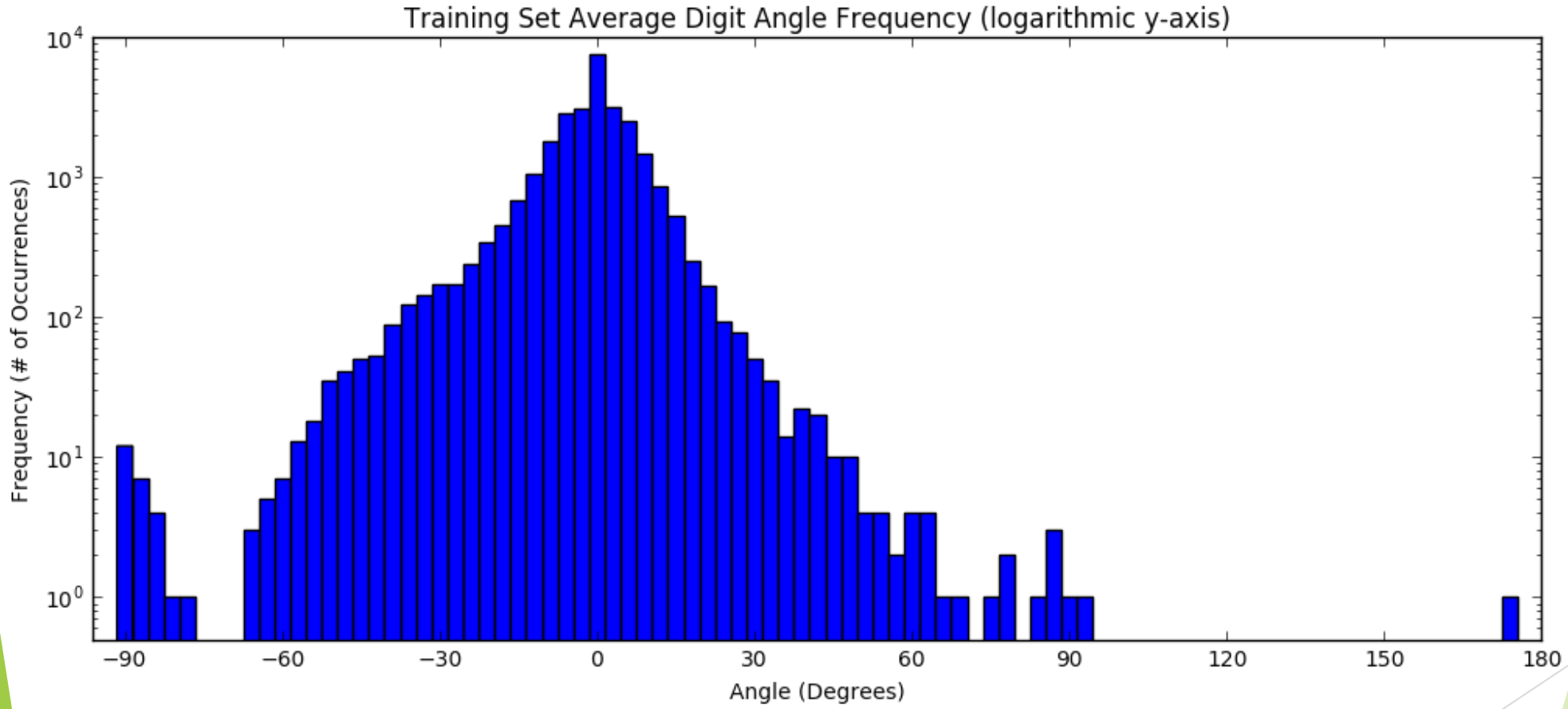
Validation loss showed a sign of overfitting similar to validation accuracy

Predictions



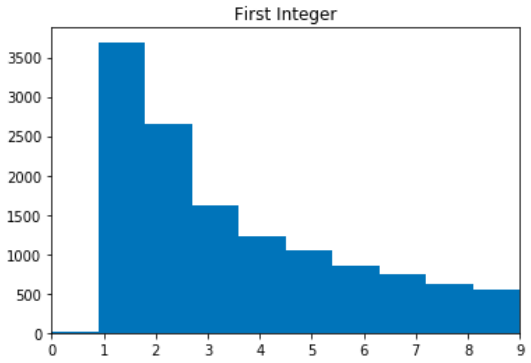
Reasons for Difficulty:

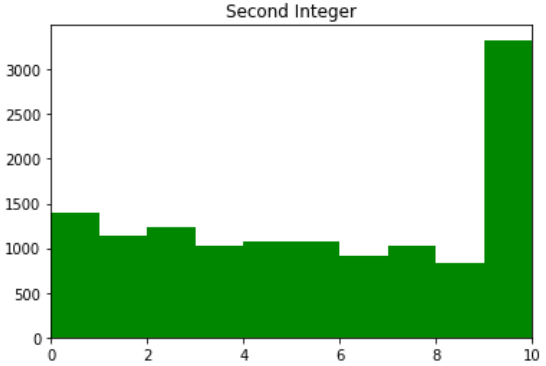
Angle of Street Numbers

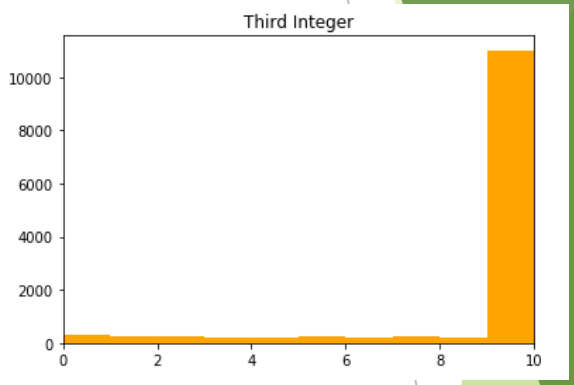


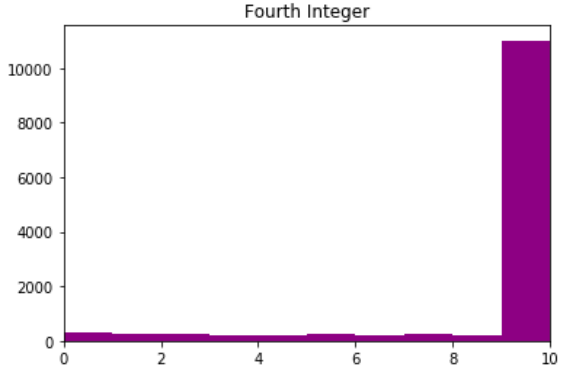
https://github.com/mgruben/digit-recognition/blob/master/report.md

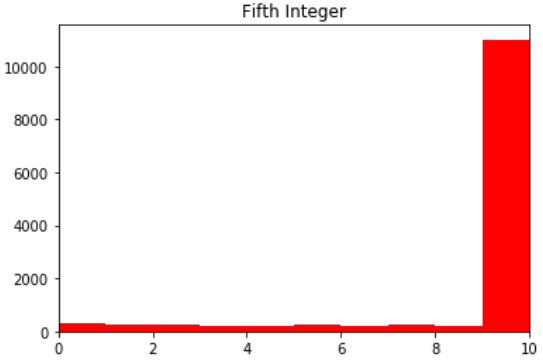
Imbalanced Digit Distribution











Obstructions

Recommendations

Improve the model:

- More data (evenly distributed)
- Splitting training data into validation – cross validation
- Image augmentation

Applications:

- Basemap data
- Scanning check amounts at ATM
- Scanned images of patient intake data