Home Number Recognition

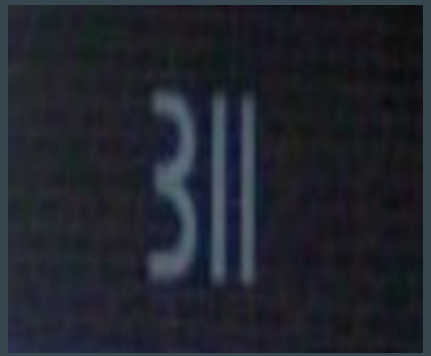
Machine Learning 2017/2018

Omar Mohamed Abdel-latif
Mohamed Ahmed Ismail

Supervised by
Professor Ali Fahmy & Dr. Hanaa Bayomi

Problem

Given an image, the task is to identify the number in the image. The number to be identified is a sequence of digits, s = s1; s2; : : : ; sn. When determining the accuracy of a digit transcriber, we compute the proportion of the input images for which the length n of the sequence and every element si of the sequence is predicted correctly.



Previous attempts

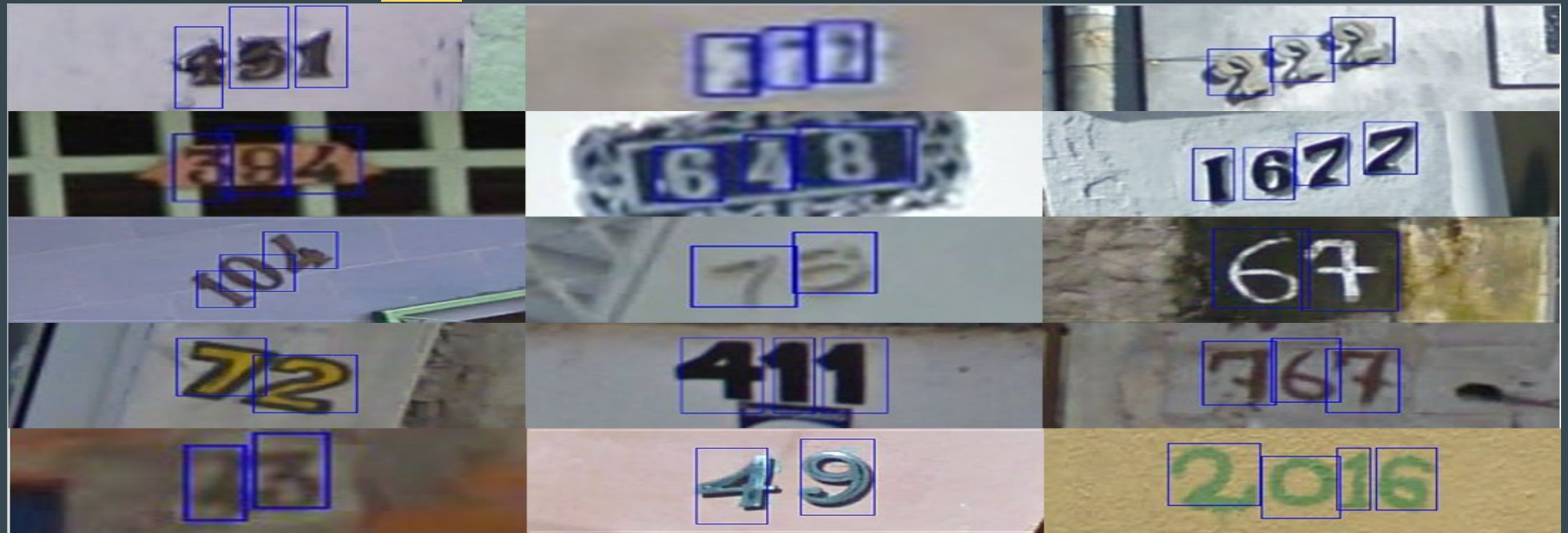
-Traditional approaches to solve this problem typically separate out the localization, segmentation, and recognition steps.

-We will use a unified approach that integrates these three steps via the use of a deep convolutional neural network that operates directly on the image pixels.

SVHN Dataset

SVHN is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. SVHN is obtained from house numbers in Google Street View images.

link for the dataset : Link



Dataset splitting

-The original SVHN dataset contains around 33k training image and around 13k test image.

-Download extra training examples from SVHN website around(202k images)

-Add 150k image to training, 50k to test, and around 2300 to validation set.

-Now the training set is around 183k picture

-The test set is around 63k picture

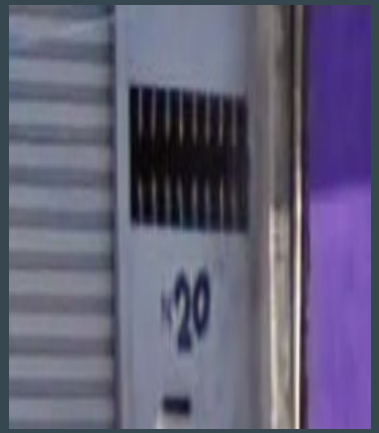
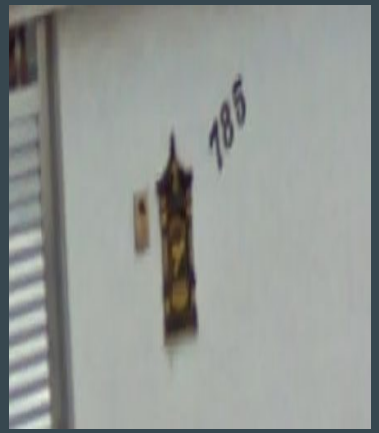
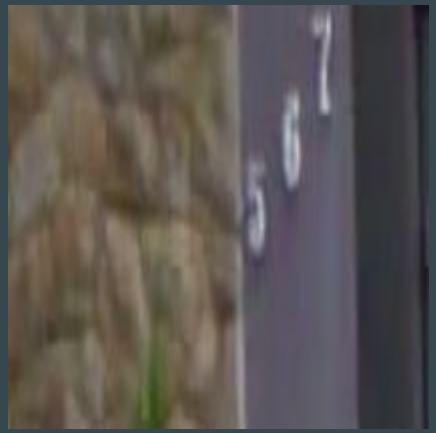
-The validation set is around 2300 picture

SVHN Dataset Problems

-The images are of different sizes, zoom, resolution, and aspect ratio.

-Variability in colors, and lighting conditions.

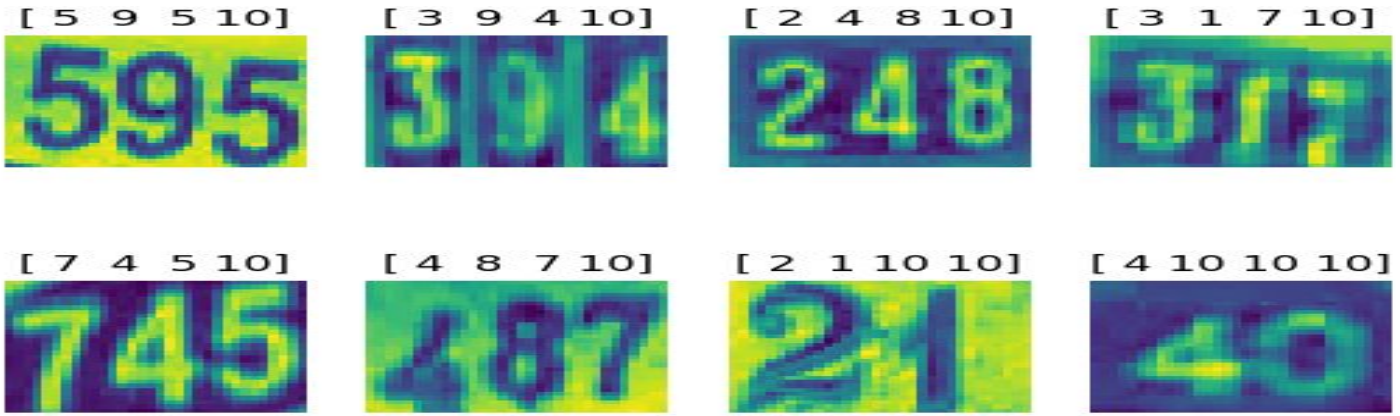
-Some wrong labels

Dataset Preprocessing

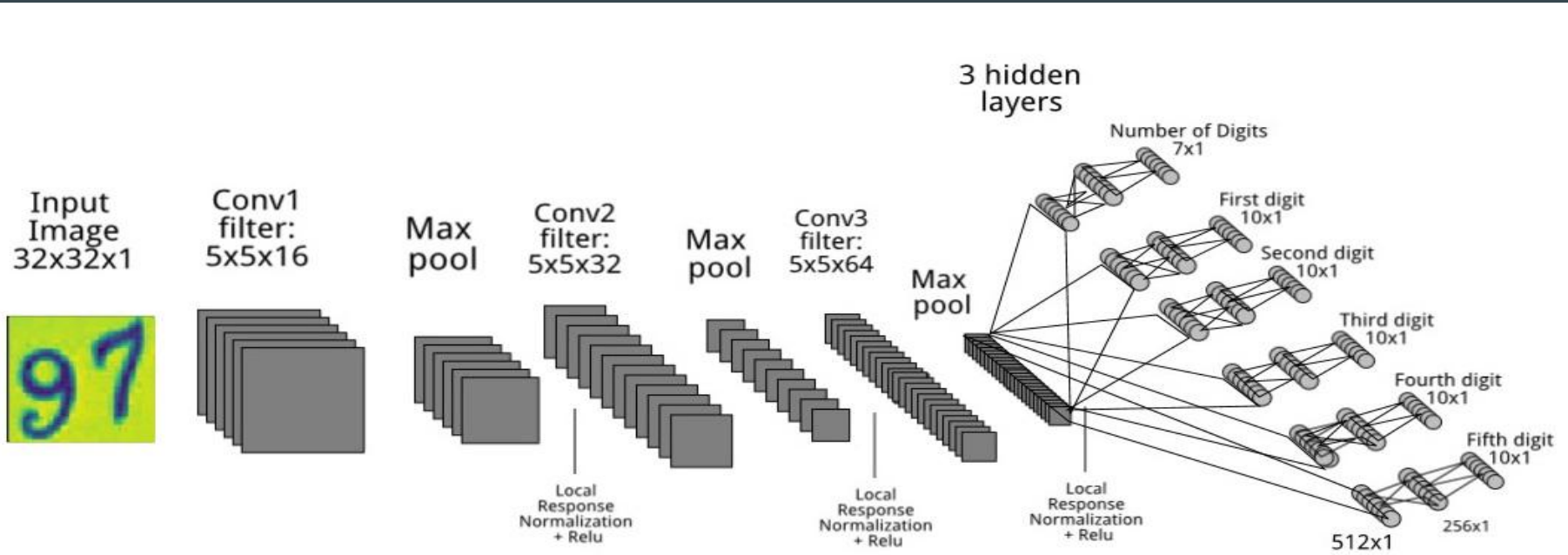
-Cropping using the bounding boxes provided in the dataset 32x32 or 64x64

-Grayscale and Normalization



Training Architecture

-How to know number if digits in the image? (Let it learn it)



Architecture details

-Every convolutional block(conv+max pooling) is followed by a local response normalization and relu activation.

-The hidden layers in the fully connected classifiers do not share weights. Meaning every classifier has 3 weight matrices to learn.

-The fully connected layers use Relu as activation function except for the output layer which uses softmax.

-The cost function is cross entropy over softmax. The error is summed for the different classifiers and propagated to the convolution layers to adjust the weights.

Architecture details

-Used Adam optimizer for backpropagation

-Used mini-batch gradient descent for training with a batch size equal 64 images

-Around 200k training iteration

-Learning rate initially equal 0.001 and used exponential decay by 10% every 10k iterations.

-Used gradient clipping by 1.0 to overcome exploding gradients

-Used dropout on the fully connected hidden layers with a keep rate of 0.7

Architecture details

-When the model is inserted a new image to be classified, we choose the maximum of the first k digit classifiers depending on the output of the first classifier which is the number of digits in the image.

Problems We Faced

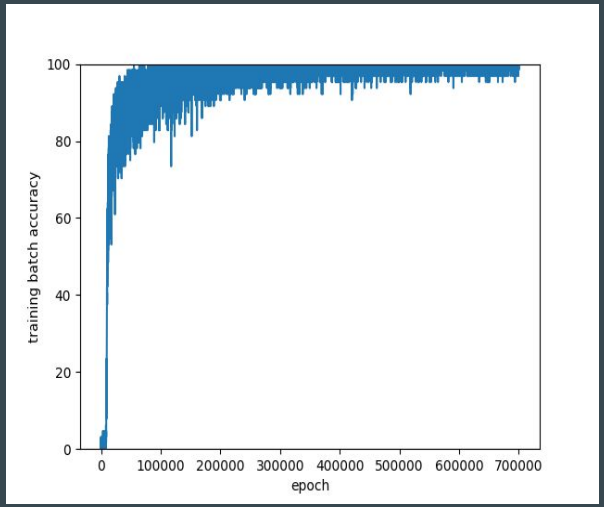
-Overfitting (increased dataset size and added drop out)

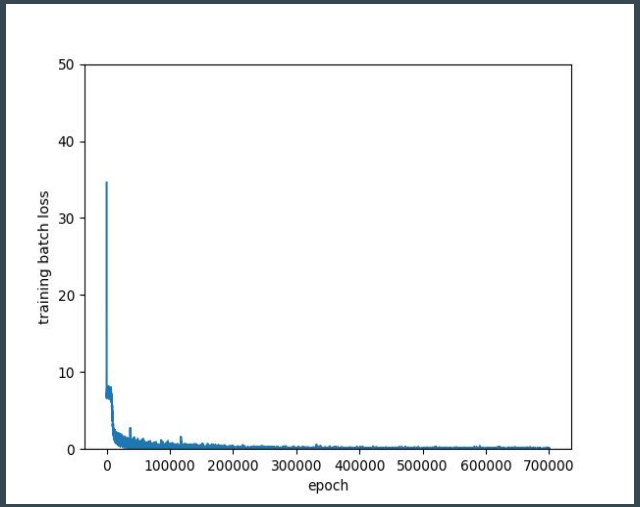
-Exploding gradients (used gradient clipping)

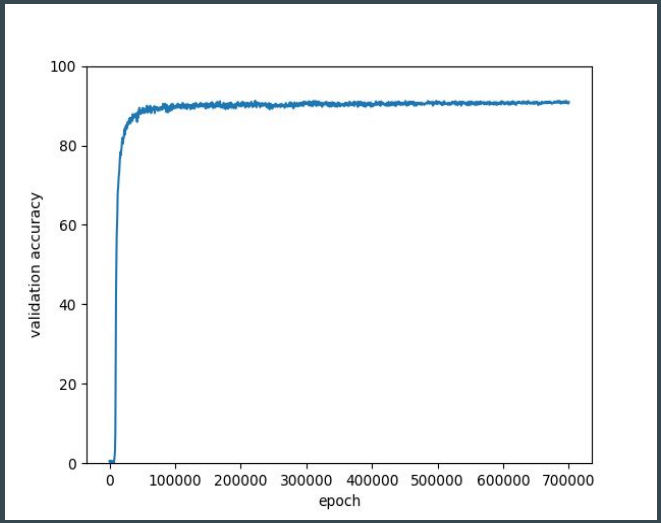
Results

-Reached 89.4% on the 63k test images.

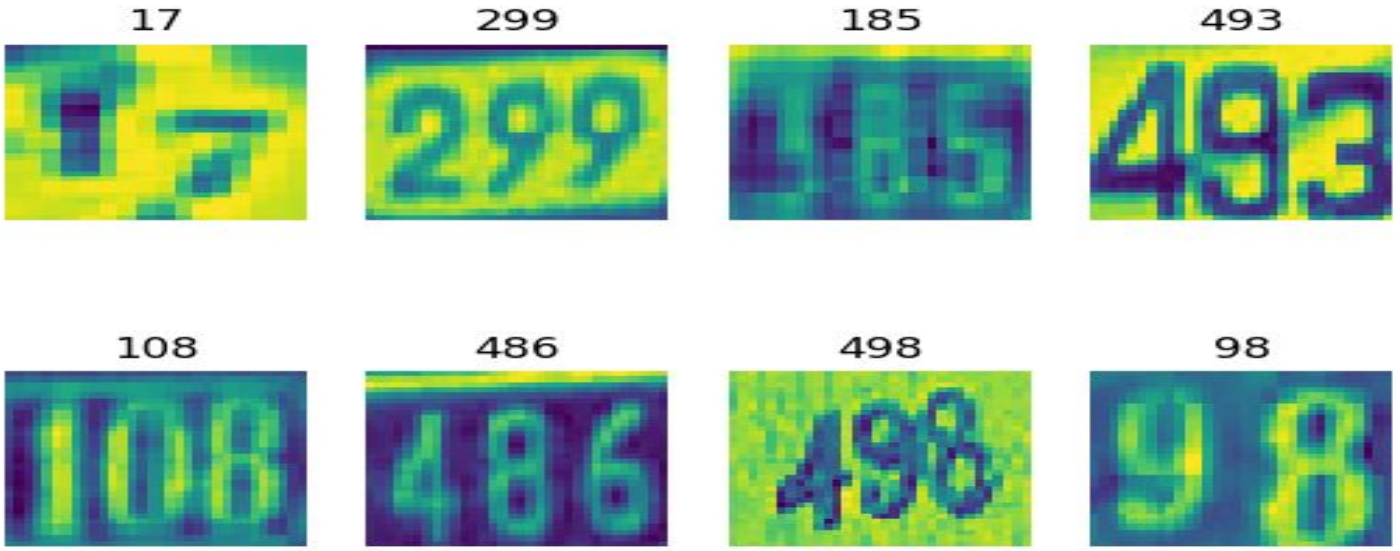
-Reached 91% on 2300 images validation set.



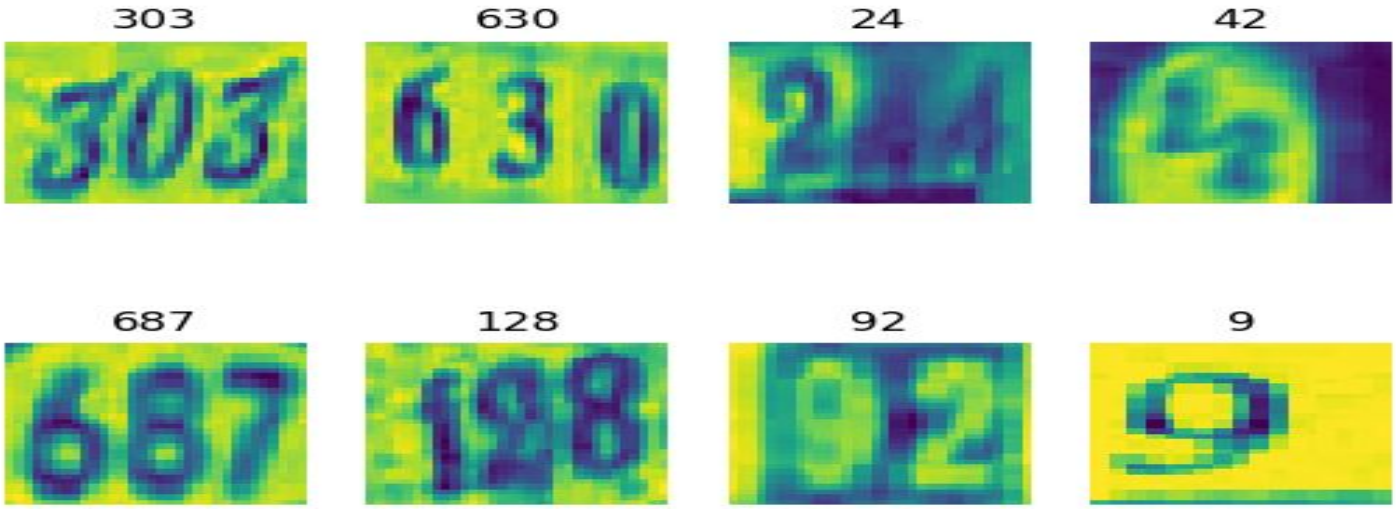




Results



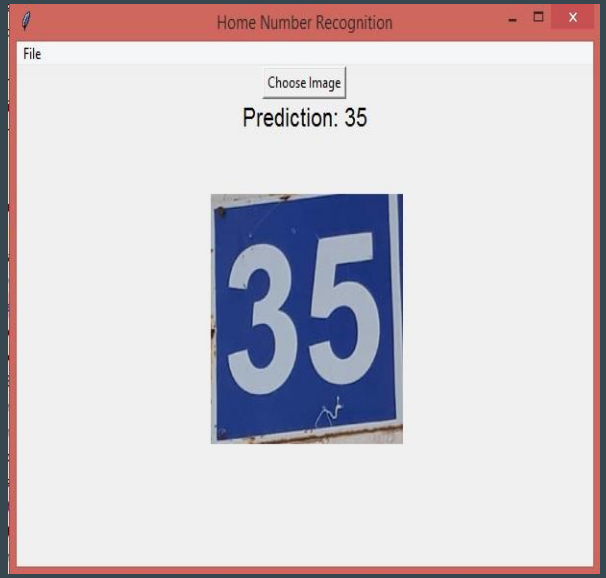
Results

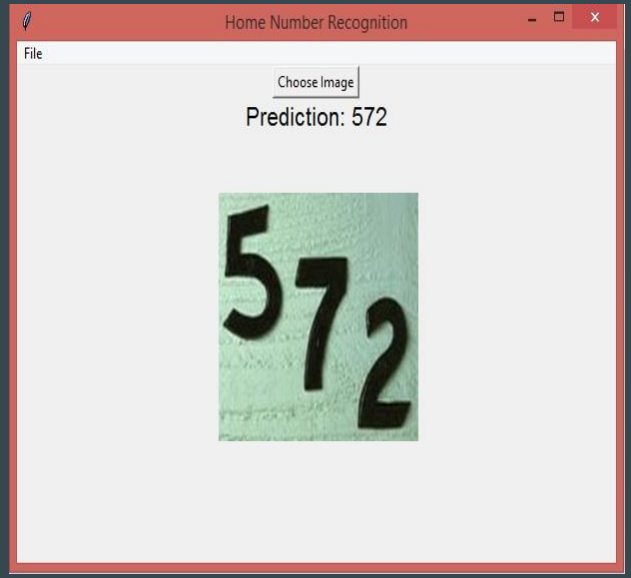


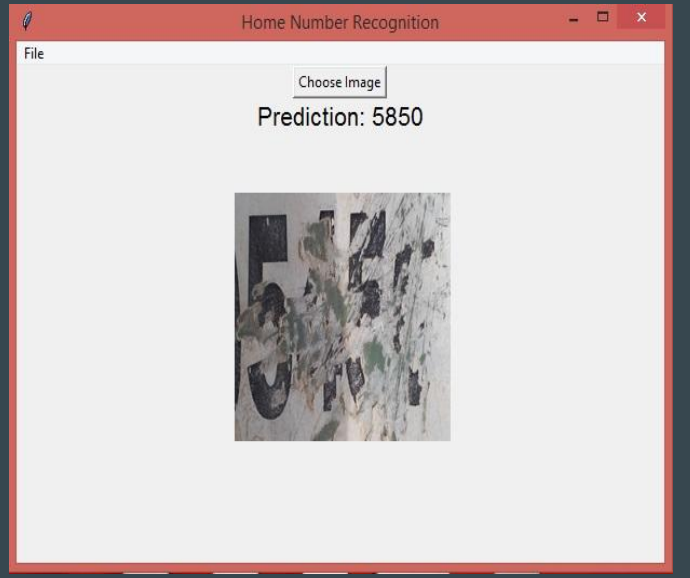
GUI Interface

-We made a GUI interface that takes a picture and classify it using our saved model.

-Link for repository containing code and interface: Link







Things to try

-Go deeper (Res nets)

-Try inception

-Maximize probability over all classifiers when predicting (not just take maximum depending on predicted number of digits)

-Alternating loss computation between classifiers over the training iterations instead of summing the loss in every iteration.

Questions?

References

Ian Goodfellow Paper, Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks

Link: Link