Cem Kaya 26440
Uğur Kağan Çakır 27058
Selin Tokman 28495

CS 412 Report

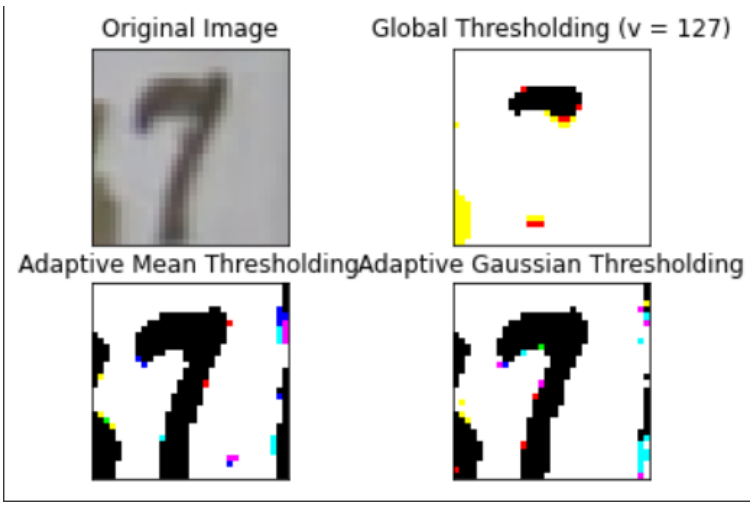
INTRODUCTION

Our data set is Google's Street view house numbers data set. As the name of the data set implies this data came from Google's Street view project and contains house numbers. The data set has two forms: the first one has the coordinates of the house numbers annotated with a bounding box, and the second one contains cropped versions of those bounding boxes.

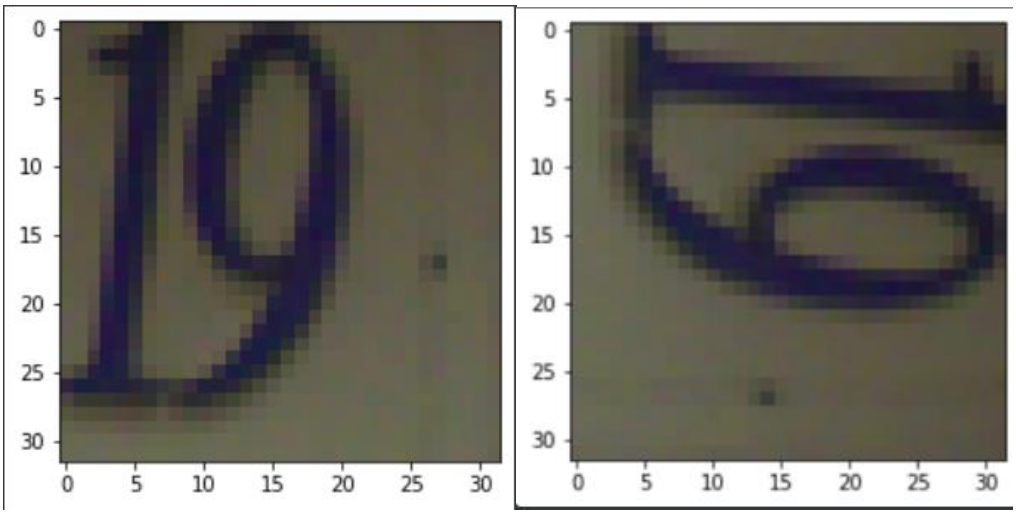
We are going to use the second form of the data set. This data is from the real world and thus contains some impurities which makes the learning process a bit more challenging compared to the Mnist data set. For example, in this data set the colours of the background and the numbers are varied. Secondarily, these cropped versions contain other numbers beside the main number and this is a further challenge that we should overcome.

PREPROCESSING

To overcome the first problem which is diverse color we used a technique for image processing. We obtained the best result from adaptive threshold filters. You can see the result:



We had 72K samples to train and validate our model. We tried to augment the data further. We tried to rotate the images and train and learn the all possible rotations of the images. However, we only turned the images using increments of 90 degrees and these rotated images proved to impair the learning of the models. So we used kerases ImageDataGenerator to augment our data. To rotate in small increments , sheir , shift and zoom a small amount.



MODEL TRAINING

Beside data preprocessing, our general approach was deep learning. Specifically, transfer learning. We gathered several pretrained models and then blended them for our final predictions over the best performing version of the preprocessed data set.

MODEL NAME	MAX TRAIN PERFORMANCE	VALIDATION PERFORMANCE
Vgg16 (without any preprocessing)	0.9155	0.9339
Vgg16 (with color preprocessing)	0.9462	0.880425
Vgg16(manual rotational preprocessing)	0.8583	0.190185
Vgg19(without any preprocessing)	0.9256	0.9407
ResNet(without any preprocessing)	0.9413	0.9223
Decision Tree(with color preprocessing)	NA	0.55700
KNN(with color preprocessing) [Euclidean distance]	NA	0.627491
KNN(with color preprocessing) [Hamming distance as the metric]	NA	0.627491

RESULTS

As a result of what we tested, Vgg19(without any preprocessing) gave us the best result with the 0.9407accuracy. The worst result we got was with theVgg16(manual rotational preprocessing) with nearly 0.19 accuracy.

The models show that in contrast to what we expected color preprocessing to help the learning process however mostly it did not help. Furthermore, in some cases it made the models overfit to the data. Thus we did not pre-processed or further manually augment our data to achieve best results. Except for Decision Tree and kNN.

As the validation data shows DT and kNN did not perform as well as any deep learning based technique thus they were absent from our blending.

We were expecting that the knn with hamming distance metric would perform significantly better than normal knn and decision tree. However it did not.

BLENDING

For the final predictions we blended all the best performing deep learning models with equal weights. We did not change the weights due to similarities of the validation scores. Furthermore, instead of adding them we multiplied the results so that models can inhibit each other better in addition to adding them.

Appendix

Colab Link:
https://drive.google.com/file/d/1YUjyOpo_ff1GmsA2TSGwUUlIRzpIyxKj/view?usp=sharing