

## Introduction

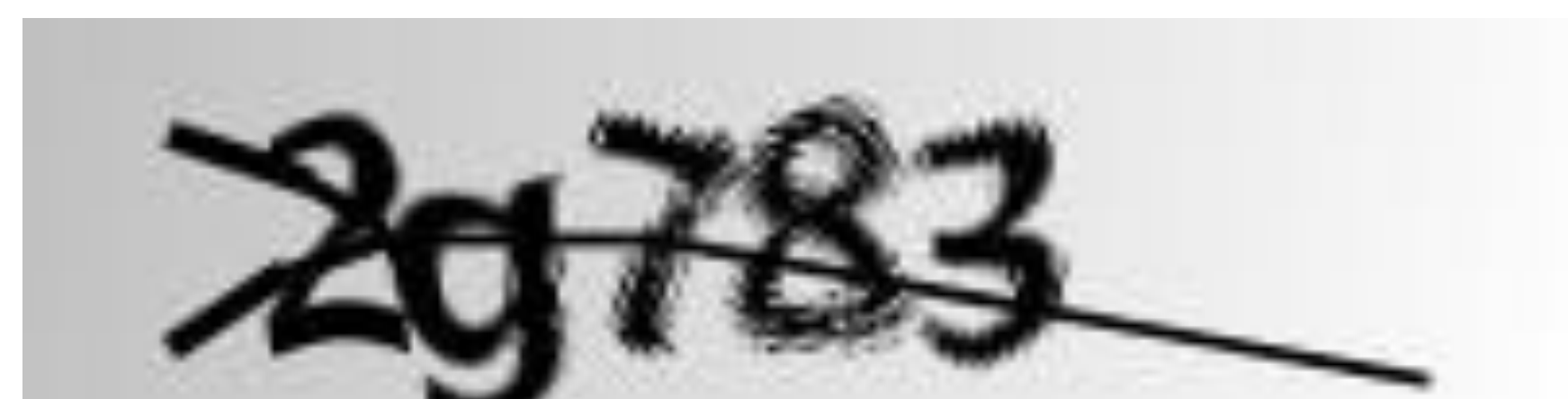


CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). A typical captcha consists of a distorted test, which a computer program cannot interpret but a human can still read. Captcha consists of texts or images with distortions and noise. The aim of this project to recognize and solve captchas using deep learning.

## Data Set

[Wilhelmy, Rodrigo & Rosas, Horacio. \(2013\). captcha dataset.](#)

The images are 5-letter-words that can contain numbers. The images have had noise applied to them (blur and a line). They are 200 x 50 PNGs.



## Method

- Remove noises and unstructured line using kernel
- Extracting letters by making simple vertical cuts at 20%, 40%, 60% and 80% of the captcha,
- Add a third channel to the image,
- Save each letter as a single image,
- Then Train the model on single letter.

### Tuning :

There are a lot of hyperparameters that can be tuned, and different values are explored:

1. Number of Convolution Layers : tried with 2 and 3 layers, both provide similar results, however, the 3 layers approach consumes much time
2. Kernel Size: When increasing kernel sizes from 3X3 to 5X5 the performance enhanced
3. Used Random Search to find parameters for epochs and batch size .

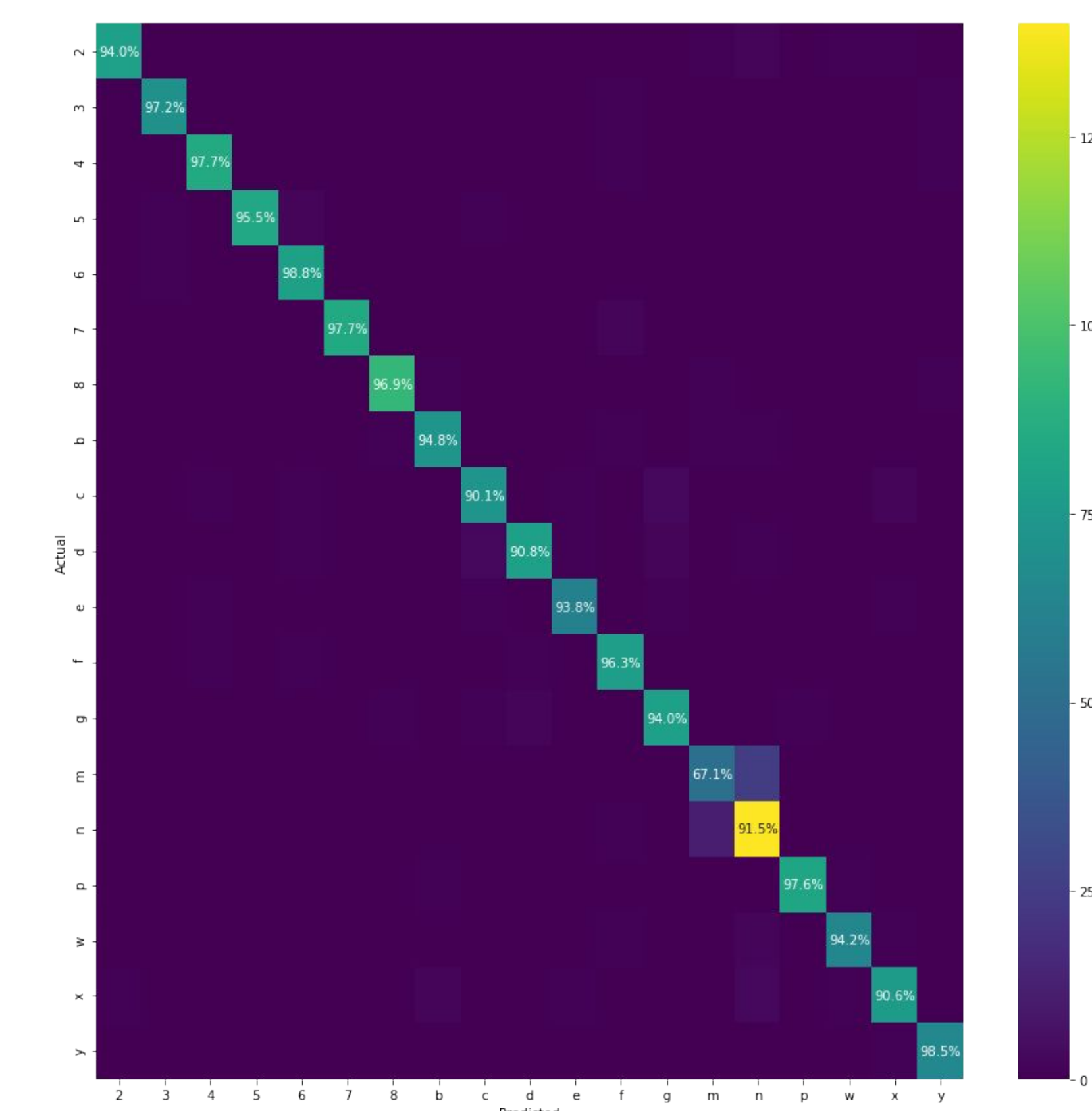
### Build Model:

A convolutional neural network (CNN) trained on single letters was built with 2 convolution layers (5 X 5 kernel) ,flatten layer , Dens layer(500 units) and softmax output. The model train over 100 epochs with batch of size 10.

## Results

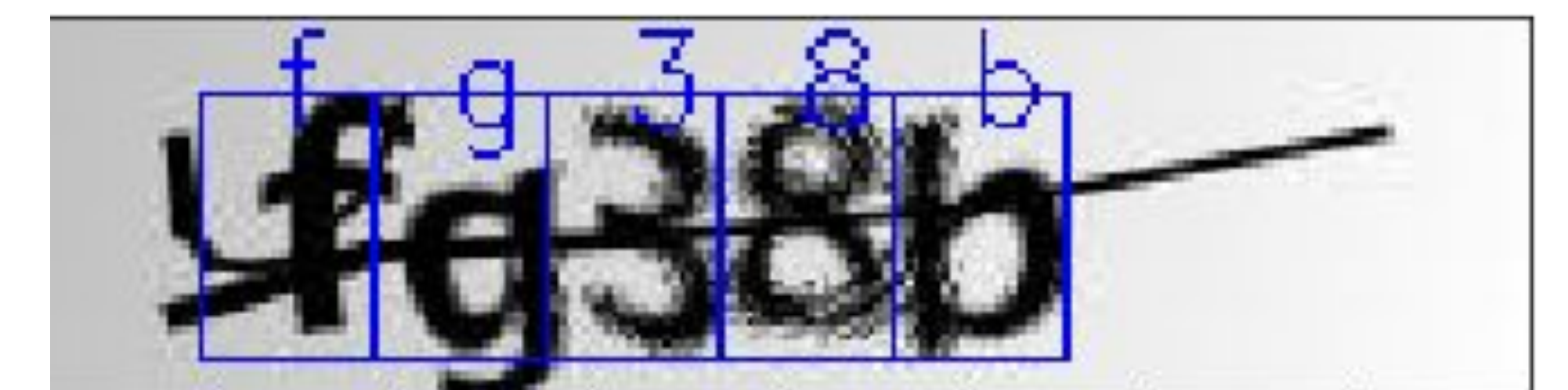
	Train Accuracy	Test Accuracy
CNN	93.7%	93.5%

### Confusion Matrix :



The model sometimes confuses character pairs with similar structures such as "M" and "N".

## Simple output image



## Conclusion

With the power of deep learning, the accuracy of cracking captcha generated by Wilhelmy, Rodrigo & Rosas is pretty high. It is also strongly suggest that same performance can be guaranteed with other captchas dataset generated from other libraries as well. In the future any discussion about creating a new captcha should *start with the power and the limitations of a convolutional neural network.*

### Future work:

- Improve segmentation techniques ,
- Build a Multitype CAPTCHA Recognition and see if it can improve letter detection ,
- Solve misrecognition of confusable characters.

## References

- sklearn documentation: <http://scikit-learn.org/stable/>
- Keras documentation: <https://keras.io>
- DataSet : <https://www.researchgate.net>

## Pipeline

