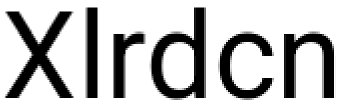
Report on precog's recruitment tasks.

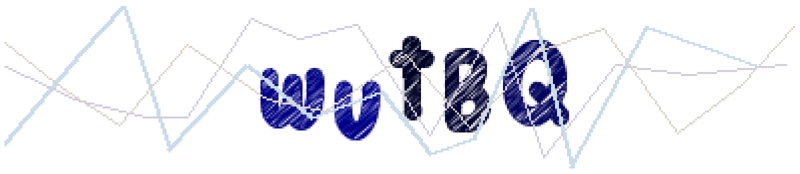
Task : Can you break the CAPTCHA?

Task -0:

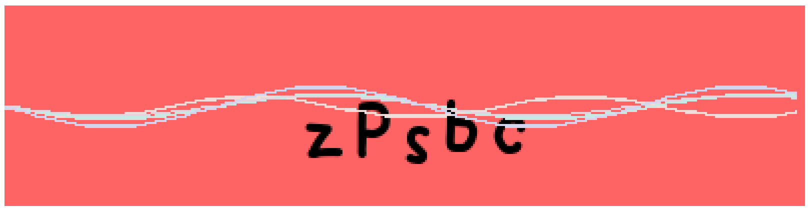
Created a dataset of captcha images with a python script using Pillow (PIL) library.



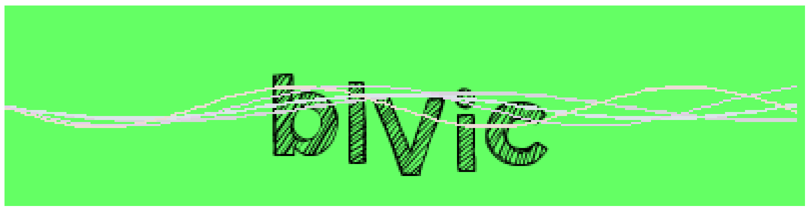
Easy image



Hard Image



Bonus red image



Bonus green image

- Each image contains 3 to 7 uppercase and lowercase alphabets.
- Twenty-two types of fonts are used, including typewriter fonts with partially diminished letters.
- Easy images have no noise and use only the Roboto font, with the first letter capitalised and the remaining letters in lowercase.
- For hard images, noise and varied spacing arrangements of letters are used.
- The bonus set features wavy noise, and the letters do not share a common baseline.

Task-2: Classification
First I used a dataset of size 44 images per class (a total of 4400 images) out of these 80% i.e. 3520, 440, and 440 images are used for training, validation and testing respectively.

Problems faced: Very low accuracy ~10% when trained with this data. I tried with more data 110 images per class. There is no improvement in accuracy.
Then I changed the images to grayscale from RGB as the text only matters not the colour. Then I got these.

	Val loss	Val accuracy	Test loss	Test accuracy
44 images	1.0762	88.64%	1.0911	88.18%
110 images	0.1558	99.00%	0.1268	98.82%

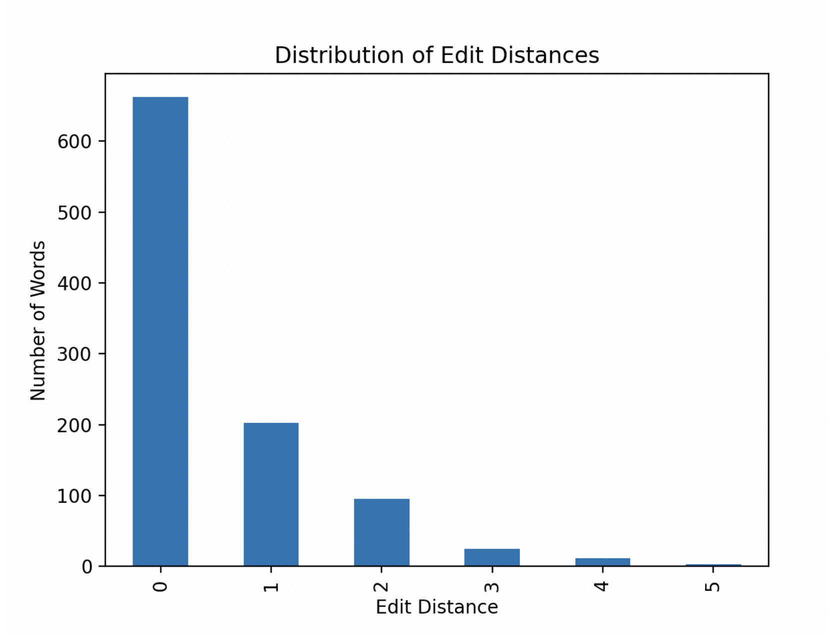
Task -2:

First I used the same dataset generated in the task-0. It contains 999 easy images and 1000 hard images. Out of the 1999 images, 1599 images(80%) are used for training and 400 images(20%) are used for testing. After so many changes in architecture, tuning hyperparameters, and testing with both the RGB and grayscale images this is what I found:

- Grayscale images give the best performance.
- The most optimal learning rate is in the range of 0.001 - 0.003. Increasing lr to 0.005 results in the worst performance, same with the lr<0.001
- The model starts predicting the first letter from the 9th epoch.
- From epoch 22 the model predicts reasonable characters.
- From epoch 49 model predicts almost all the characters correctly.
- At epoch 68 model touched 50% accuracy in the test set.
- At last, by the end of the 100th epoch, the model settles at 58% accuracy and in many of the remaining failed cases the predicted and and the actual word differ by 1-2 letters

Then I increased the data to 4997 images, out of which 3997 images are used for training and 1000 images are used for testing. I have used the observation from the results of the first dataset training. With the same architecture and same code (some changes for saving the best model and results) the model gives 66.2% accuracy on the test set. These results are saved in the ‘easy_hard_4997.csv’ file.

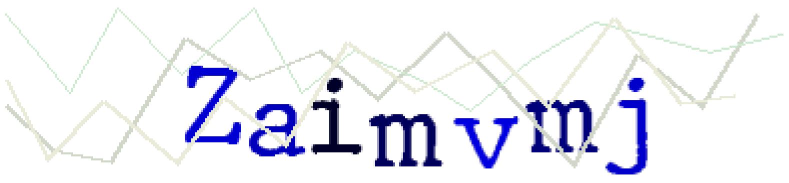
Some of the results.



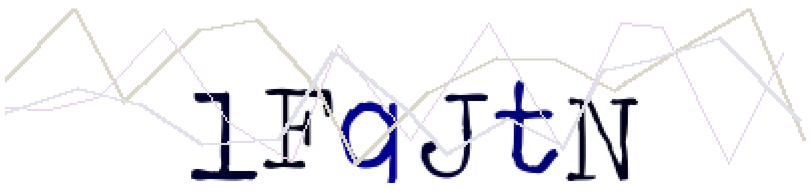
Distribution of edit distance between predicted and original words.

0	662 (66.2%)
1	203 (20.3%)
2	95 (0.95%)
3	25 (0.25%)
4	12 (0.12%)
5	3 (0.03%)

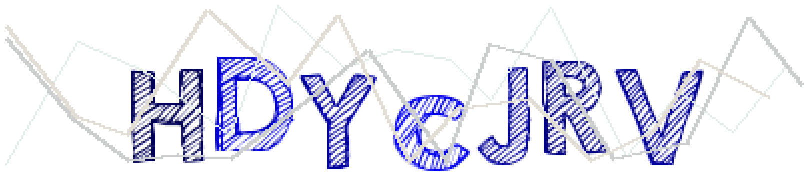
Most the the incorrect predictions differ by one letter from the actual word.



Predicted : Zaimvnj



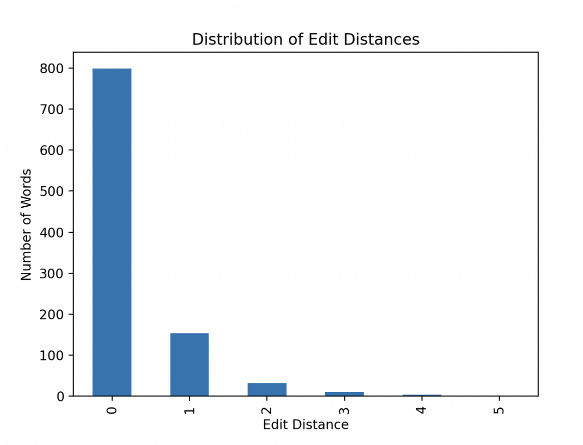
Predicted: IFaJtN


Predicted : HDYeJRV

Predicted : qXEKHfA

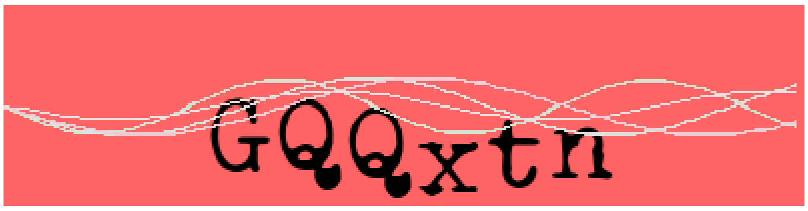
Bonus task:
First I used the same code with the image in RGB but it didn't give any good results. So I kept the image in grayscale and I checked the background of the image from the first pixel. If the first pixel is red then I reversed the label. And trained the model to predict the letters in the same order they appear in both the red and green images. During inference and testing, I again check the background colour. If the colour is red then I reverse the prediction.
This is how I got better results.

I used a dataset of size 4998 images, out of which 3998 are used for training and 1000 images are used for testing.
Learning rate >0.001 results in model not learning anything.

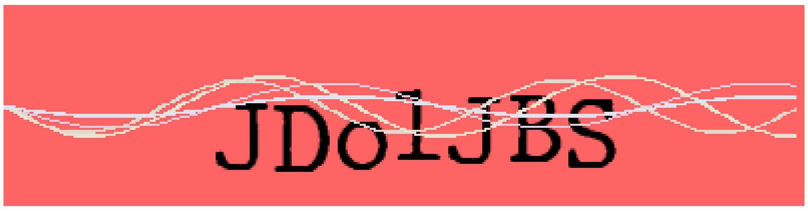


The model gives an accuracy of 79.80% on the test set.

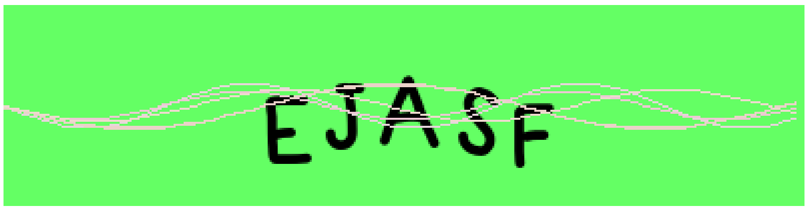
0	798 (79.8)
1	154 (15.4)
2	32 (0.32)
3	11 (0.11)
4	4 (0.04)
5	1 (0.01)



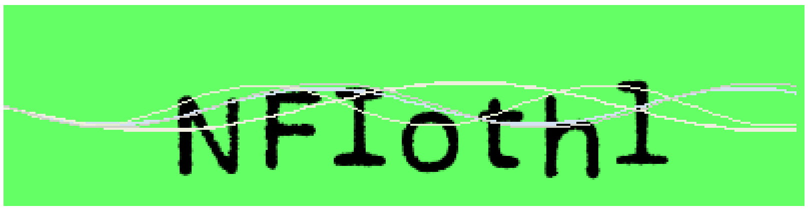
Predicted : ntxQQG



Predicted:SBJloDJ



Predicted: EJASF



Predicted : NFIothl