Breaking CAPTCHA Using CRNN

One of the internet's first AI benchmarks vs. a modern AI technique

Goals

01 Create a effective CAPTCHA solver

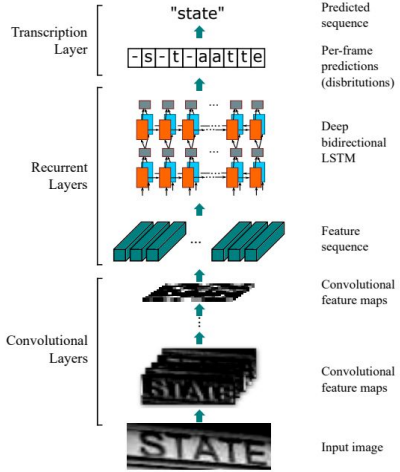
02 See where the approach fails

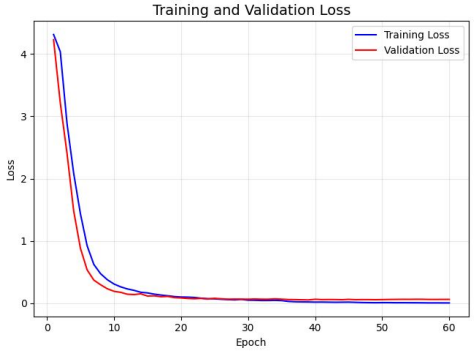
03 Imagine better CAPTCHA systems

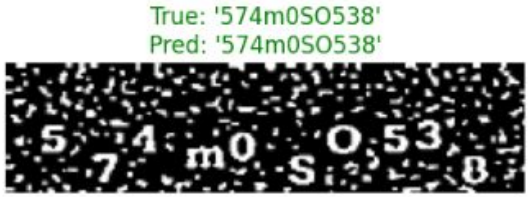
Methodology

Before

After







Preprocessing

Use common OpenCV techniques to prepare image data for feature extraction

Predicting

Leverage the CRNN architecture to make probabilistic predictions about the ground truth over 25 frames

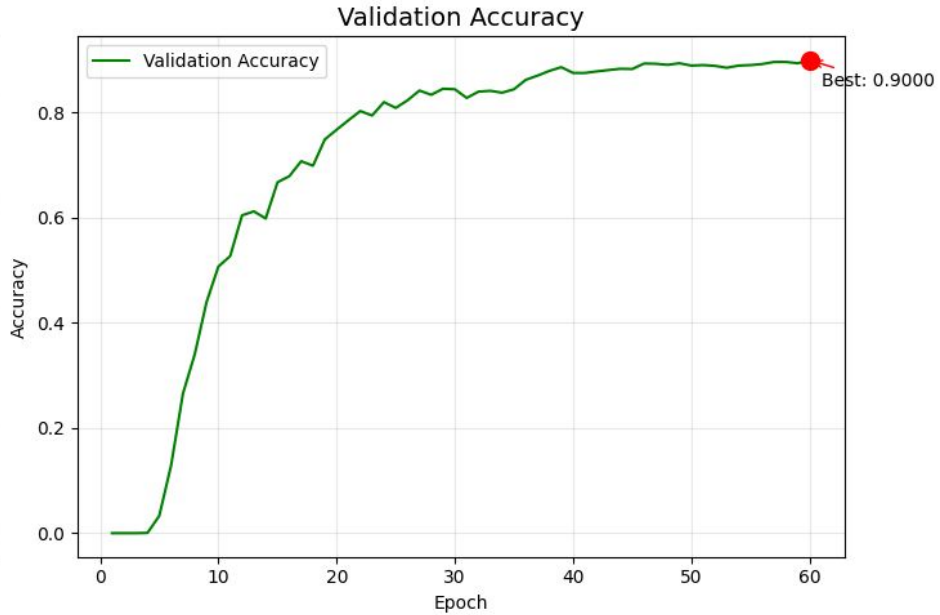
Training

Use these probabilities to calculate CTC loss for training

Evaluation

Compare our decoded string with the ground truth to get our results

Results



Captcha-level Accuracy:
88.0%

Character-level Accuracy:
98.6%

Average error confidence:
99.3%

Most common errors:

I → l: 13 times
I → 1: 13 times
I → i: 6 times

Characters that showed up most often in errors:

'1': appears in 44.2%
'7': appears in 36.7%
'4': appears in 34.2%

Conclusions

CRNN is very effective in OCR

Almost 99% character-level accuracy points to CRNN being a very effective character classifier

High-confidence errors point out flaws

This result indicates that there are still significant optimizations that can be made in this approach

Numbers cause errors in prediction

The prevalence of numbers in our errors show that there is strong correlation between numbers being present in a CAPTCHA and model failure

Thank you!

Questions?