CAPTCHA sequence recognition

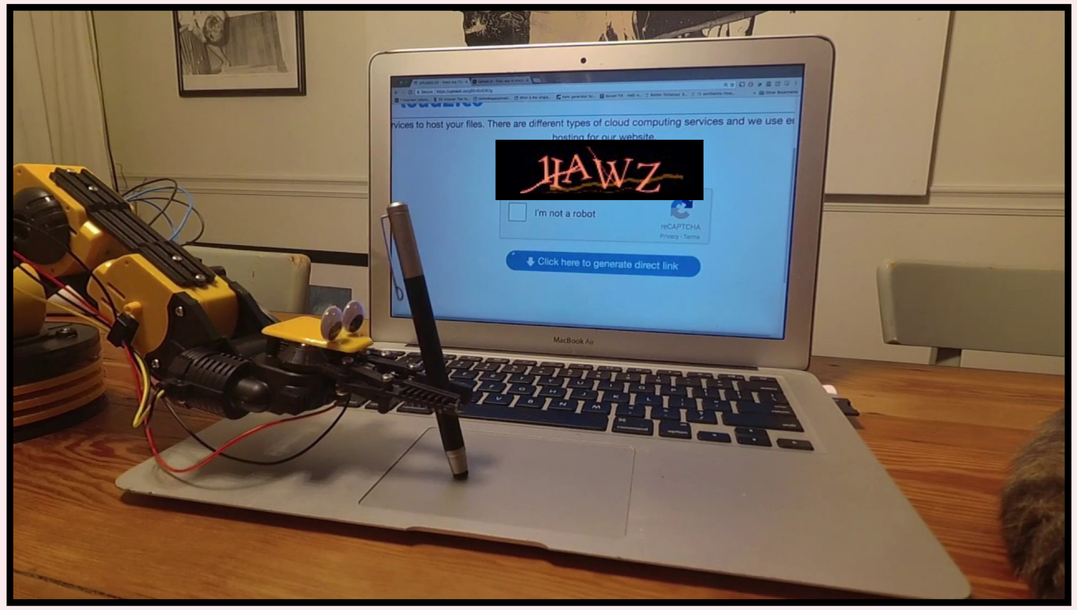
Deep Learning

Nick Cheliotis 2025

Introduction

Introduction

Goal



The goal is to build a deep learning model capable of accurately predicting the full sequence of characters.

Introduction

Candicate Architectures

1.CNN with 5 outputs?

❌ Assumes characters are perfectly spaced and aligned.

Introduction

Candicate Architectures

1.CNN with 5 outputs?

✗ Assumes characters are perfectly spaced and aligned.

2. RNN?

✗ Doesn’t understand visual patterns in an image.Also no horizontal context.

Introduction

Candicate Architectures

1.CNN with 5 outputs?

❌ Assumes characters are perfectly spaced and aligned.

2. RNN?

❌ Doesn’t understand visual patterns in an image.Also no horizontal context.

3.CNN+RNN then?

❌ Almost, but it needs exact alignment of predicted character positions.

Introduction

Candicate Architectures

1.CNN with 5 outputs?

ligned.

2. RNN

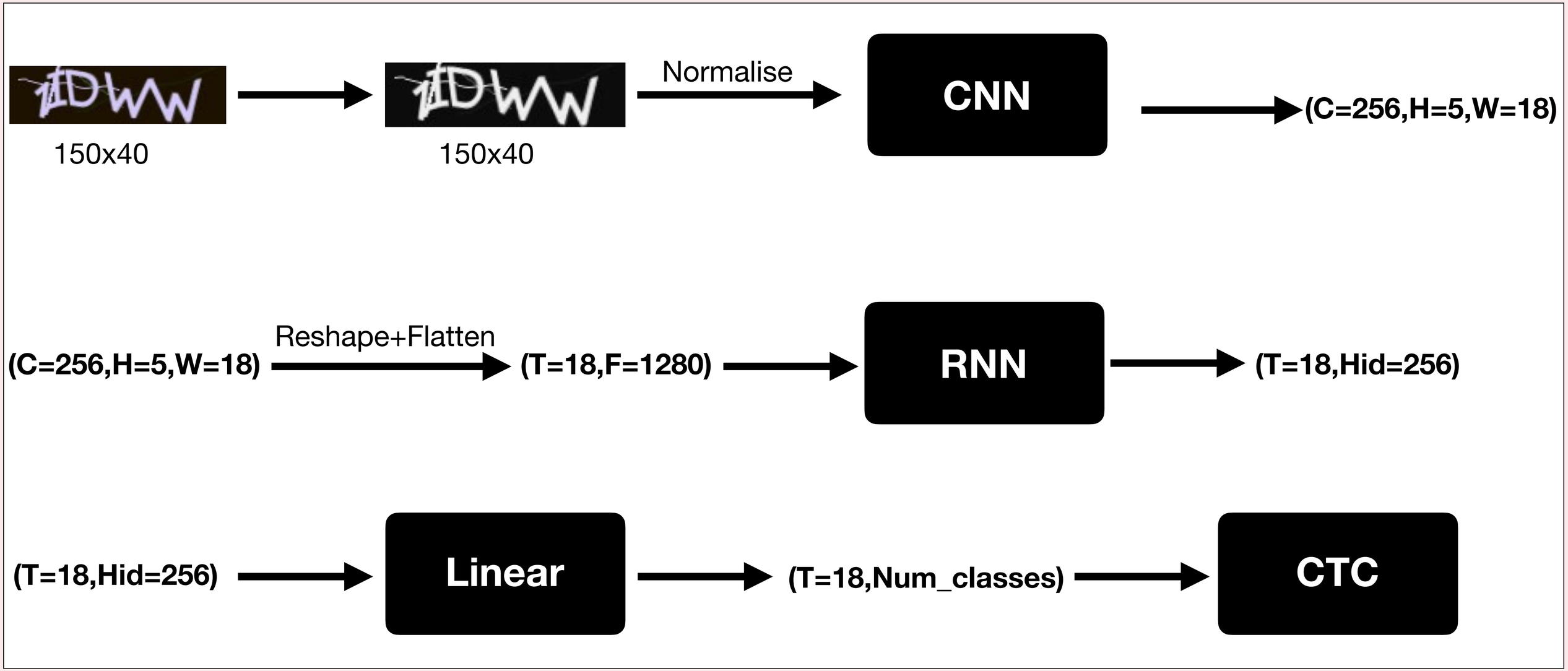
CNN+RNN+CTC

3.CNN-

Almost, but it needs exact alignment of predicted character positions.

Introduction

CNN+RNN+CTC Architecture



Introduction

Challenges

Technical challenges:

1. Training instability with CTC loss: Needs careful tuning (seed,batch size etc).

2. Evaluation is non trivial: Accuracy alone is not enough and common metrics (F1, Precision) are not applicable here.

3. Limited compute: Training large models is slow and GPU expensive.

Experimental Process

Specs

CNN

Input: Bx150x40

Layers: 2

Output: Bx75x5

RNN

Input: 75XBX640

Layers: 1

1-direction

Output:75xBx128

CTC

Greedy Decode

General

Epochs:30

Samples:16

Runs locally

Experimental Process

First Overfitting tests

```
[Overfit Test] Epoch 1, Loss: 56.4784, Accuracy: 0.00%
[Overfit Test] Epoch 2, Loss: 47.4041, Accuracy: 0.00%
[Overfit Test] Epoch 3, Loss: 28.8457, Accuracy: 0.00%
[Overfit Test] Epoch 4, Loss: 11.7846, Accuracy: 0.00%
[Overfit Test] Epoch 5, Loss: 5.4051, Accuracy: 0.00%
[Overfit Test] Epoch 6, Loss: 4.6291, Accuracy: 0.00%
[Overfit Test] Epoch 7, Loss: 4.9704, Accuracy: 0.00%
[Overfit Test] Epoch 8, Loss: 5.2070, Accuracy: 0.00%
[Overfit Test] Epoch 9, Loss: 5.2838, Accuracy: 0.00%
[Overfit Test] Epoch 10, Loss: 5.2752, Accuracy: 0.00%
[Overfit Test] Epoch 11, Loss: 5.2121, Accuracy: 0.00%
[Overfit Test] Epoch 12, Loss: 5.1330, Accuracy: 0.00%
[Overfit Test] Epoch 13, Loss: 5.0031, Accuracy: 0.00%
[Overfit Test] Epoch 14, Loss: 4.9048, Accuracy: 0.00%
[Overfit Test] Epoch 15, Loss: 4.8125, Accuracy: 0.00%
[Overfit Test] Epoch 16, Loss: 4.7125, Accuracy: 0.00%
[Overfit Test] Epoch 17, Loss: 4.6051, Accuracy: 0.00%
[Overfit Test] Epoch 18, Loss: 4.4975, Accuracy: 0.00%
[Overfit Test] Epoch 19, Loss: 4.3868, Accuracy: 0.00%
[Overfit Test] Epoch 20, Loss: 4.2775, Accuracy: 0.00%
[Overfit Test] Epoch 21, Loss: 4.1945, Accuracy: 0.00%
[Overfit Test] Epoch 22, Loss: 4.1384, Accuracy: 0.00%
[Overfit Test] Epoch 23, Loss: 4.1020, Accuracy: 0.00%
[Overfit Test] Epoch 24, Loss: 4.0887, Accuracy: 0.00%
[Overfit Test] Epoch 25, Loss: 4.0943, Accuracy: 0.00%
[Overfit Test] Epoch 26, Loss: 4.1120, Accuracy: 0.00%
[Overfit Test] Epoch 27, Loss: 4.1338, Accuracy: 0.00%
[Overfit Test] Epoch 28, Loss: 4.1518, Accuracy: 0.00%
[Overfit Test] Epoch 29, Loss: 4.1607, Accuracy: 0.00%
[Overfit Test] Epoch 30, Loss: 4.1582, Accuracy: 0.00%
```

Loss drops but accuracy stays 0%.This was to expected, CTC needs lots of epochs to align.

Experimental Process

First Overfitting tests diagnostics

```
[Overfit Test] Epoch 1, Loss: 55.7081, Accuracy: 0.00%
Predicted: ['ei', 'eiei', 'e', 'eieiei', 'e', 'e', 'e', 'e', 'e', 'e', 'ieeiei', 'e', 'e', 'efeqei', 'eeieii', 'e']
Actual   : ['iFWmY', 'IMJvj', 'XvRMg', 'qnkl4', 'tX5Km', '9QTud', '17UFt', '1GTxC', 'Op1HJ', 'Z71ZQ', 'NiyuV', 'nJnPy', 'ZcGyV', 'kmz8S', 'XsnHw', 'DWvkH']
--------------------------------------------------
[Overfit Test] Epoch 2, Loss: 42.4313, Accuracy: 0.00%
Predicted: ['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']
Actual   : ['iFWmY', 'IMJvj', 'XvRMg', 'qnkl4', 'tX5Km', '9QTud', '17UFt', '1GTxC', 'Op1HJ', 'Z71ZQ', 'NiyuV', 'nJnPy', 'ZcGyV', 'kmz8S', 'XsnHw', 'DWvkH']
--------------------------------------------------
[Overfit Test] Epoch 3, Loss: 23.5319, Accuracy: 0.00%
Predicted: ['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']
Actual   : ['iFWmY', 'IMJvj', 'XvRMg', 'qnkl4', 'tX5Km', '9QTud', '17UFt', '1GTxC', 'Op1HJ', 'Z71ZQ', 'NiyuV', 'nJnPy', 'ZcGyV', 'kmz8S', 'XsnHw', 'DWvkH']
--------------------------------------------------
[Overfit Test] Epoch 4, Loss: 10.0506, Accuracy: 0.00%
Predicted: ['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']
Actual   : ['iFWmY', 'IMJvj', 'XvRMg', 'qnkl4', 'tX5Km', '9QTud', '17UFt', '1GTxC', 'Op1HJ', 'Z71ZQ', 'NiyuV', 'nJnPy', 'ZcGyV', 'kmz8S', 'XsnHw', 'DWvkH']
--------------------------------------------------
[Overfit Test] Epoch 5, Loss: 5.1162, Accuracy: 0.00%
Predicted: ['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']
Actual   : ['iFWmY', 'IMJvj', 'XvRMg', 'qnkl4', 'tX5Km', '9QTud', '17UFt', '1GTxC', 'Op1HJ', 'Z71ZQ', 'NiyuV', 'nJnPy', 'ZcGyV', 'kmz8S', 'XsnHw', 'DWvkH']
--------------------------------------------------
[Overfit Test] Epoch 6, Loss: 4.6813, Accuracy: 0.00%
Predicted: ['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']
Actual   : ['iFWmY', 'IMJvj', 'XvRMg', 'qnkl4', 'tX5Km', '9QTud', '17UFt', '1GTxC', 'Op1HJ', 'Z71ZQ', 'NiyuV', 'nJnPy', 'ZcGyV', 'kmz8S', 'XsnHw', 'DWvkH']
```

It seems that greedy decode always predicts the blank space.

Specs

CNN

Input: Bx150x40

Layers: 3

Output: Bx18x5

RNN

Input: 18XBX1280

Layers: 2

bi-directional

Output:18xBx256

CTC

Beam search

General

Epochs:200

Samples:16

Runs locally

Penalty blank

Experimental Process

Second Overfitting tests diagnostics

```
DECODED: lClk
TARGET : 1Cl9k

RAW:     Rllllllllllllllllk
DECODED: lRlk
TARGET : xlztk

RAW:     PyyyyyYYYYYYYYYYYY
DECODED: PyY
TARGET : PDyyY

RAW:     Rllllllllllllllllk
DECODED: lRlk
TARGET : nqlBf

RAW:     PDyyyyYYYYYYYYYYYY
DECODED: PDyY
TARGET : PDyyY

RAW:     Ettttttttttttttttta
DECODED: Eta
TARGET : vgtsa

RAW:     Rllllllllllllllllk
DECODED: RlYlYlk
TARGET : frYdA

RAW:     Ettttttttttttttt2N
DECODED: Et2
TARGET : Et52N
```

The Neural Network makes actual predictions.

Specs

CNN

Input: Bx150x40

Layers: 3

Output: Bx18x5

RNN

Input: 18XBX1280

Layers: 2

bi-directional

Output:75xBx256

CTC

Beam search

General

Epochs:30

Samples:10.000

Batch:32

Penalty blank

COLAB

Char Accuracy

Avg. edit distance

T4 GPU

Experimental Process

Real Tests

```
Epoch 30/30:  0%                                    0/254 [00:05<?, ?it/s, loss=0.409]
Epoch 30: Loss = 68.7434 | Exact Acc = 0.22% | Char Acc = 29.29% | Edit Dist = 3.14
```

```
 FINAL TEST RESULTS:
Exact Accuracy: 0.10%
Char Accuracy: 27.94%
Avg. Edit Distance: 3.22
```

Results were promising but inconsistent across runs.This was due to random weight initialisation and the sensitive nature of CTC.To address this we tested multiple random seeds, identified one that produced good learning behavior, trained it for 30 epochs an saved it.

Specs

CNN

Input: Bx150x40

Layers: 3

Output: Bx18x5

RNN

Input: 18XBX1280

Layers: 2

bi-directional

Output:75xBx256

CTC

Beam search

General

Epochs: 60

Samples:100.000

Batch:64

Gradient clipping

Lr scheduler

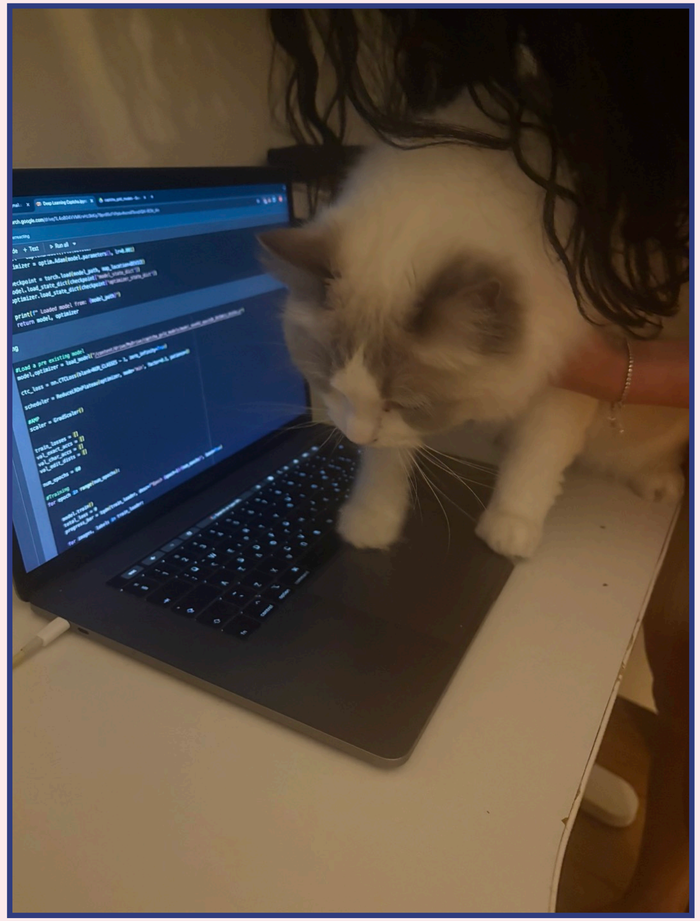
AMP

Loaded model

A100 GPU

Experimental Process

Final test



```
 FINAL TEST RESULTS:
Exact Accuracy: 53.06%
Char Accuracy: 79.74%
Avg. Edit Distance: 0.76
```

Results

Results

How did we do?

According to studies*, similar architectures combining CNNs, RNNs, and CTC loss have achieved near-perfect accuracy in CAPTCHA recognition tasks. This validates our design choice and shows that with further fine-tuning, optimization, and training time, this architecture has the potential to reach state-of-the-art performance.

***1.End-to-End Captcha Recognition Using Deep CNN-RNN Network**

2.Adaptive CAPTCHA: A CRNN-Based Text CAPTCHA Solver with Adaptive Fusion Filter Networks

Thank you!