# COMPUTER VISION CS385

ROLL NO : 2101AI25 PROJECT TILE : HANDWRITING RECOGNITON

BRIEF REPORT :

Important links of resources used for projects :

1.Sequence Modeling With CTC  , <https://distill.pub/2017/ctc/> .

2.Edit Distance , <https://en.wikipedia.org/wiki/Edit_distance> .

3.OCR Model , <https://keras.io/examples/vision/captcha_ocr/> .

4.IAM Dataset , <https://fki.tic.heia-fr.ch/databases/iam-handwriting-database> .

Requirements :

Tensorflow , Keras , Numpy , Pandas. I have used anaconda distribution.

My python version is 3.11.4

Google Colab works best with T4 GPU as Runtime. I have attached the custom\_test data for predicting on my handwriting for 6 images.

Folder name is custom\_test.

I have also attached .h5 models for instant loading without training.

Project idea is to develop a model which can recognize the handwritten text and transcribe the words into machine readable format.

For training I have used IAM dataset which is a large dataset.

The database contains forms of unconstrained handwritten text, which were scanned at a resolution of 300dpi and saved as PNG images.  A total of 115320 isolated and labelled words are present in the dataset.

**IMAGE  PRE-PROCESSING**

* We have to work with rectangular images.
* While aspect-unaware resizing square images doesn’t introduce distortion but for rectangular images it isn’t the case.
* But resizing images to a uniform size is a requirement for mini-batching.
* I have resized them as   image\_width = 128, image\_height = 32
* So we need to perform our resizing such that the following criteria are met.
  + Aspect ratio is preserved.
  + Content of the images is not affected.

**DATA SET SPLITTING**

* Split used for train test validation  = 0.90 : 0.5 : 0.5
* Total unique characters : 78
* Max token length of word : 21

# MODEL

I have created a deep learning model with the following layers.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA black rectangle with white lines

Description automatically generated

PARAMETERS CALCULATIONS EXPLANATION :

1. Layer Conv1 uses 32  : (3,3) matrices . So  9 weights in (3,3) matrix + 1 bias parameter.

* Total parameters = (3\*3\*1+1) \* 10 = 320.
* Same padding is used so input and output shapes match together.

  2. Layer Pool1 uses  32 (2,2) matrix. Max has no weights.

* Total parameters = 0
* Width and height of last layer matrix is halved.

3.       Layer Conv2 uses 64  : (3,3) matrices . So  9 weights in (3,3) matrix + 1 bias parameter.

* Total parameters = (3\*3\*32 + 1 )\*64 =  18496   [ 32 is last layer channels ].
* Same padding is used so input and output shapes match together.

4. Layer Pool2 uses 32 : (2,2) matrices. Max has no weights.

* Total parameters = 0
* Width and height of last layer  matrix is halved.

5. Layer reshape : Reshapes the last layer output which is current input.

* Total parameters = 0
* new\_shape = ((image\_width // 4), (image\_height // 4) \* 64).

6. Layer Dense-1 uses 64 units

* Total parameters = (512+1)\*64  = 32832 parameters.
* Width remains same , height of last layer matrix is now 64.

7. Layer Dropout uses 0.2 probability to drop some connections which acts as regularizer.

* Total parameters = 0
* Output shape is same as input shape.

8. Layer Bidir LSTM uses 128 units and 0.25 probability of dropout.

* Total parameters = 2\* ((64 \* 4 \* 128 ) + (128 \*4 \* 128 ) + 4\*128)   = 197632
* 4 -  Input gate, forget gate , cell state , output gate, 2 for bi-directional .
* Output shape is (32,256) 256 is concatenation of outputs from forward & backward 128 units.

Parameters=(Input size×4×Units)+(Units×4×Units)+4×Units

9. Layer Bidir LSTM 2 uses 64 units and 0.25 probability of dropout.

* Total parameters = 2\* ((256 \* 4 \* 64) + (64 \*4 \* 64) + 4\*64 )   = 164352
* 4 -  Input gate, forget gate , cell state , output gate, 2 for bi-directional .
* Output shape (32,128) 128 is concatenation of outputs from forward & backward 64 units.

10.. Layer Dense-1 uses 81 units

* Total parameters = (128+1)\*81  = 10449 parameters.
* Width remains same , height of last layer matrix is now 81.

OUTPUT LAYER HAS 81 UNITS.UNIQUE CHARACTERS ARE 79  + 2 SPECIAL TOKENS .EACH UNIT IS INTERPRETED AS PROBABILITY OF CHARACTER FOR CONSIDERATION FOR PREDICTION OF WORD.

**METRIC FOR LOSS CALCULATION.**

1.The edit distance metric, also known as the Levenshtein distance, is a measure of the similarity between two sequences of items.

2.Definition: The edit distance between two sequences is the minimum number of edits (insertions, deletions, or substitutions) required to transform one sequence into the other.

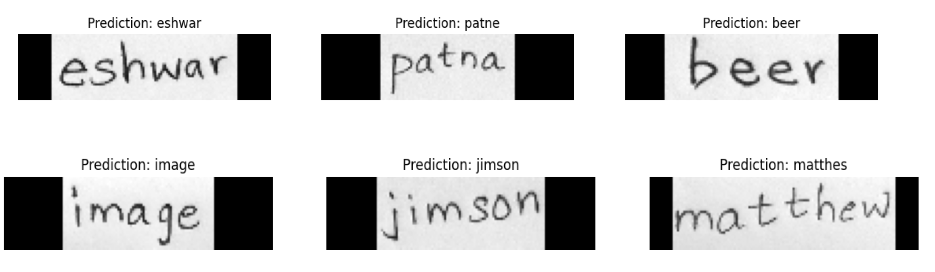
3.Lower edit distance indicates higher similarity between the predicted and ground truth sequences.

4.A perfect match results in an edit distance of 0.

**MODEL TRAINING**

1. Batch size = 64
2. No of epochs -  Tried 10 , 20 , 30 ,40 .
3. Training time = For 40 epoch it takes around 90 minutes.

# RESULTS ON CUSTOM DATASET



**THANK YOU**