Big Data Course Project

Topic given: Data Science and Machine Learning
Project Task given: Apply ML Algorithm to Dataset

Task chosen: *Recognize Handwriting using CNN + RNN with Keras and TensorFlow package*

Presentation by,
Name: Surya Narayanan S
Reg No: 121015098
Dept: Information Technology
Year: 4th year

Index

- Problem Statement
- Tech Stack
- Methodology
  - Visualize dataset
  - Pre-process Images
  - Model Pipeline
  - Training, Testing, Evaluation of Model
- Result and Analysis
- GUI (Graphical User Interface)
- Conclusion

Problem Statement

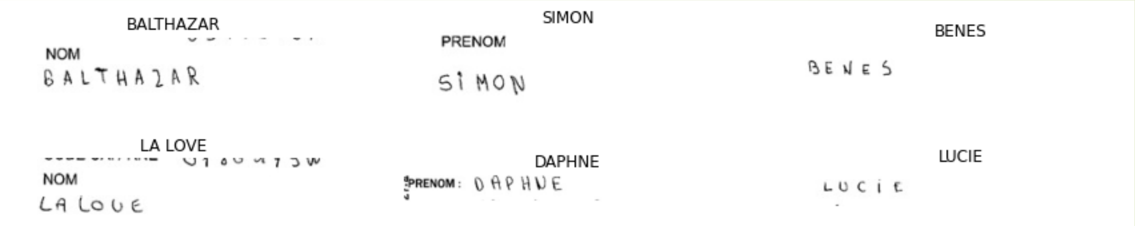
- Task chosen: *Recognize Handwriting using CNN + RNN with Keras and TensorFlow package*
- Dataset: https://www.kaggle.com/landlord/handwriting-recognition
- Dataset size: 1.3 GB
- No of Images: 3.72 Lakhs
- A CRNN (Convolutional + Recurrent Neural Network) Machine Learning Model is built to recognize and convert Handwritten Images to text using TensorFlow and Keras package in python.
- Where is it used ?
  - To digitalize physical forms (Name, Age etc., details from physical form can be recognized and converted to text)
  - Handwriting virtual Keyboard (In mobile phones, we can write in the screen using stylus or hand touch . Which is then converted to text)

Tech Stack

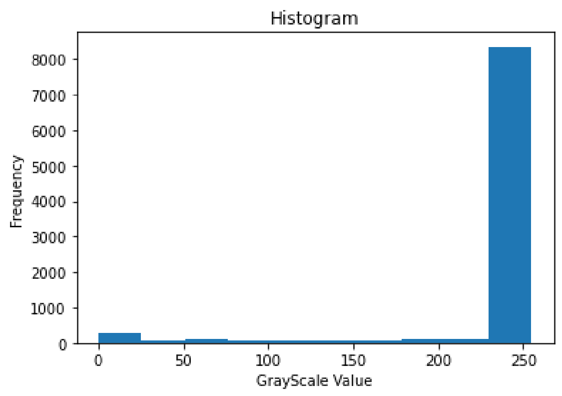
- OpenCV (for Image Processing)
- Matplotlib, Seaborn (for plotting graphs)
- Numpy and Pandas (to store Image and Table)
- Tensorflow-GPU (for training and Testing the model)
- Keras (contains ML Models like Convolution and LSTM)
- Sklearn package (for evaluation metrics)
- Jupyter Notebook (To present the code and output)

What data do we have ?

- Sample 6 Images
- Dataset Consists of 3.3 lakhs Images
- 1 lakhs unique labels
- Handwritten Text Images
- Inference:
  - Black and White Image
  - Characters are black
  - Characters are very thin
  - Image has some prefix like (Prenom and Nom (other than handwritten))



Visualize the Image Distribution



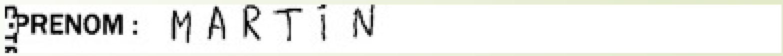
- Histogram of a sample Image
- Inference:
  - Pixel values are between 0 and 255
  - Towards zero (0) is black colour.
  - Foreground(Character) is black and very few pixels less than 1000) contribute to character.
  - Towards 255 is white colour.
  - Background is white and more than 8000 pixels contribute to background
  - In-between 0 and 255, all the pixels contribute noise to image

Pre-processing Dataset (Excel file)

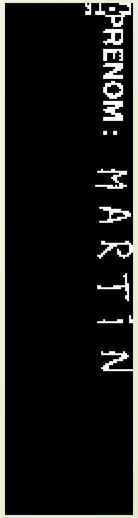
- Remove rows with No Labels (NaN) (Null Cells).
- Remove rows that has labelled as ‘UNREADABLE’.
- Convert all Labels to Upper Case Characters.
- Map all characters to Numbers between 0 and 25 to construct the model easily.

Pre-process Image

- Convert image to Grayscale(Black and White),
- Make High contrast (Threshold and Invert). This makes the Image have more foreground pixels which makes the model predict easily.
- Convert all the images to a fixed size (Height and Width). This makes the Input for the Model Uniform
- Rotate Image to 90 degree. This makes the model predict the first character, first.
- Before Pre-processing (Sample Image):



- After Pre-processing Image (Sample Image)(Black Image) :

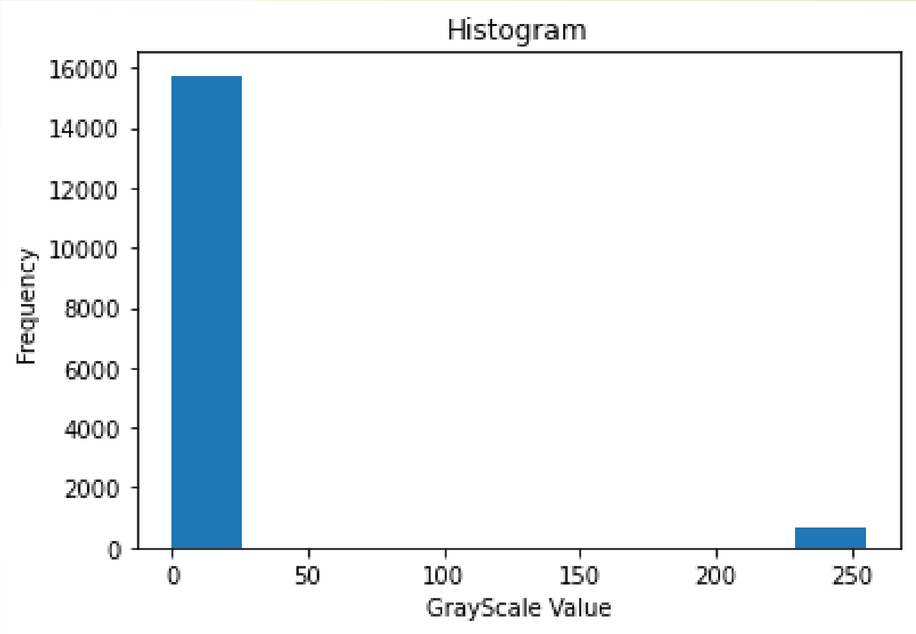


After Pre-processing, Histogram

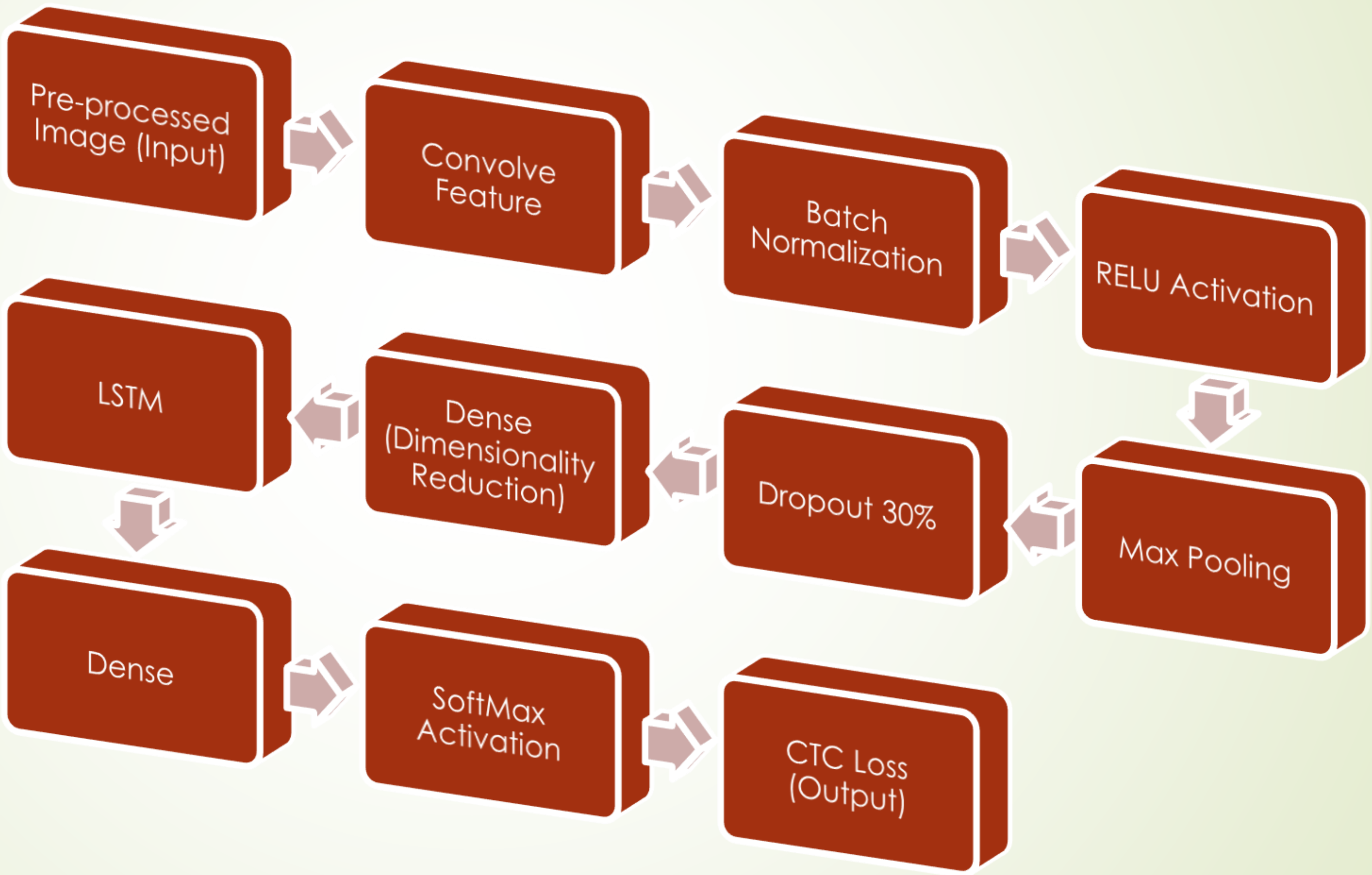
- Histogram of processed Sample Image.
- The Noisy pixels are eliminated

  by converting them either

  to pure black or pure white pixels.
- Now, White pixels are Characters.
- Black pixels are Background.
- The Number of Pixels that represent Character is increased to more than 1000 pixels from less than 1000 pixels. This makes the prediction more accurate.



Pipeline of Model

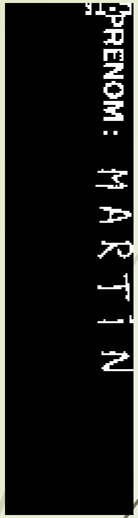


Summary of Model

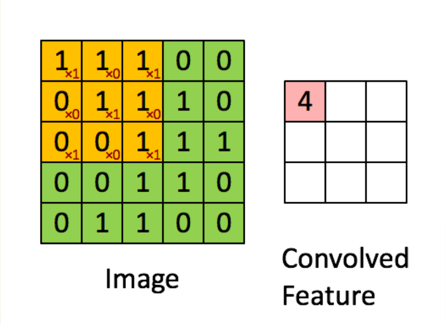
Layer (type)	Output Shape	Param #
input (InputLayer)	[(None, 256, 64, 1)]	0
conv1 (Conv2D)	(None, 256, 64, 32)	320
batch_normalization (BatchNo	(None, 256, 64, 32)	128
activation (Activation)	(None, 256, 64, 32)	0
max1 (MaxPooling2D)	(None, 128, 32, 32)	0
conv2 (Conv2D)	(None, 128, 32, 64)	18496
batch_normalization_1 (Batch	(None, 128, 32, 64)	256
activation_1 (Activation)	(None, 128, 32, 64)	0
max2 (MaxPooling2D)	(None, 64, 16, 64)	0
dropout (Dropout)	(None, 64, 16, 64)	0
conv3 (Conv2D)	(None, 64, 16, 128)	73856
batch_normalization_2 (Batch	(None, 64, 16, 128)	512
activation_2 (Activation)	(None, 64, 16, 128)	0
max3 (MaxPooling2D)	(None, 64, 8, 128)	0
dropout_1 (Dropout)	(None, 64, 8, 128)	0
reshape (Reshape)	(None, 64, 1024)	0
dense1 (Dense)	(None, 64, 64)	65600
lstm1 (Bidirectional)	(None, 64, 512)	657408
lstm2 (Bidirectional)	(None, 64, 512)	1574912
dense2 (Dense)	(None, 64, 30)	15390
softmax (Activation)	(None, 64, 30)	0

Total params: 2,406,878
Trainable params: 2,406,430
Non-trainable params: 448

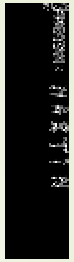
Convolution + Pooling Explained

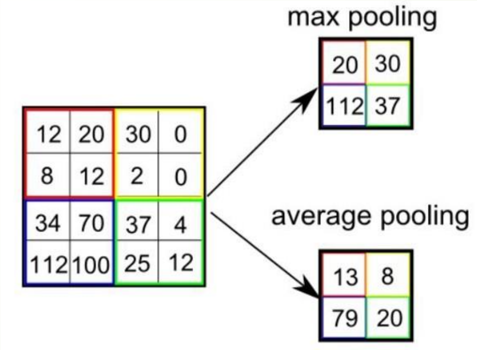


Convolution Filter



Max Pooling

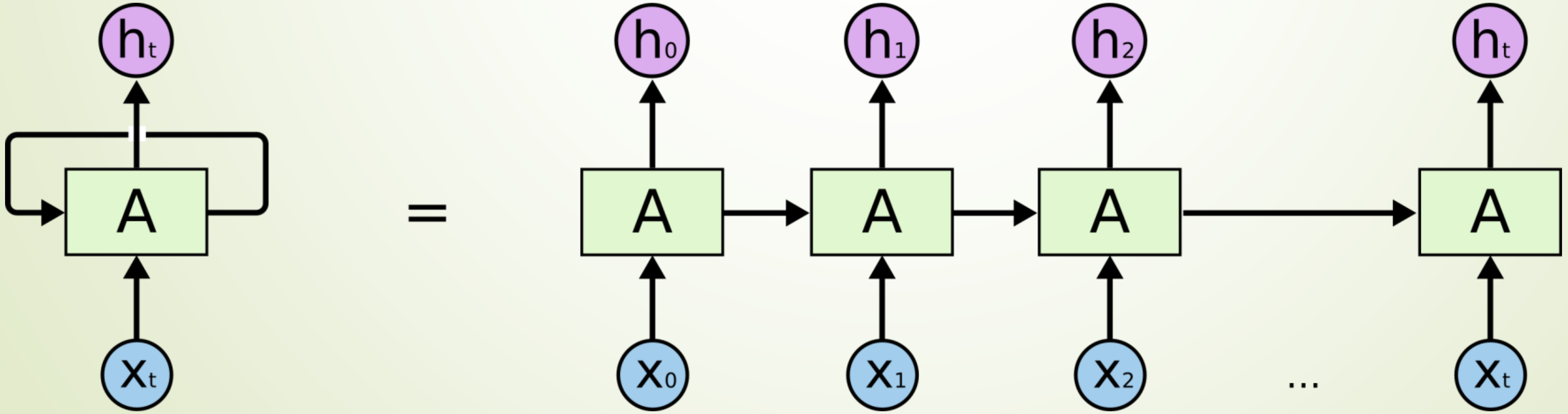




- Convolution is a Feature Extraction process. They use Filter Kernels to extract important features in the image.
- Pooling is a Dimensionality Reduction Process. It reduces the Dimension, yet conserving important features.
- Here, a sample image is convolved and pooled for example.

LSTM - RNN

- LSTM - Long Short Term Memory - Recurrent Neural network
- A RNN - multiple copies of the same network, each passing a message to a successor.
- LSTM - capable of learning long-term dependencies.
- Remembering information for long periods of time

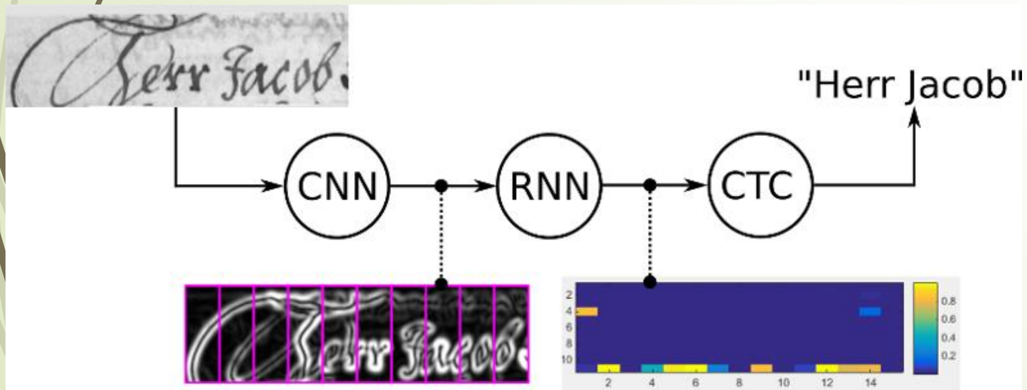


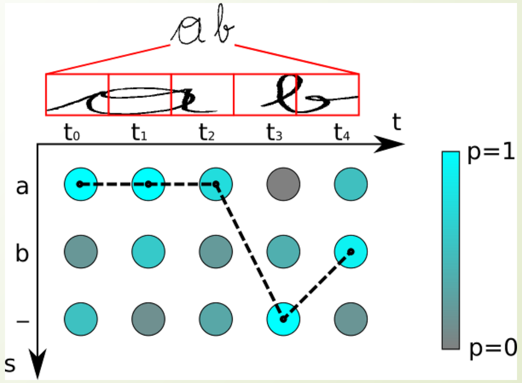
CTC Loss Output

- CTC – Connectionist Temporal Classification

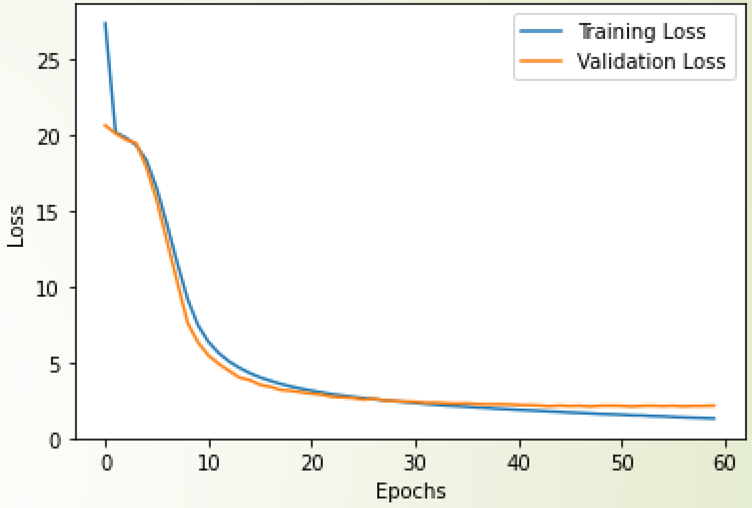
1. train: calculate the loss value to train the NN
2. infer: decode the matrix to get the text contained in the input image

- we only have to tell the CTC loss function the text that occurs in the image. Therefore we ignore both the position and width of the characters in the image.





Evaluation of CRNN Model



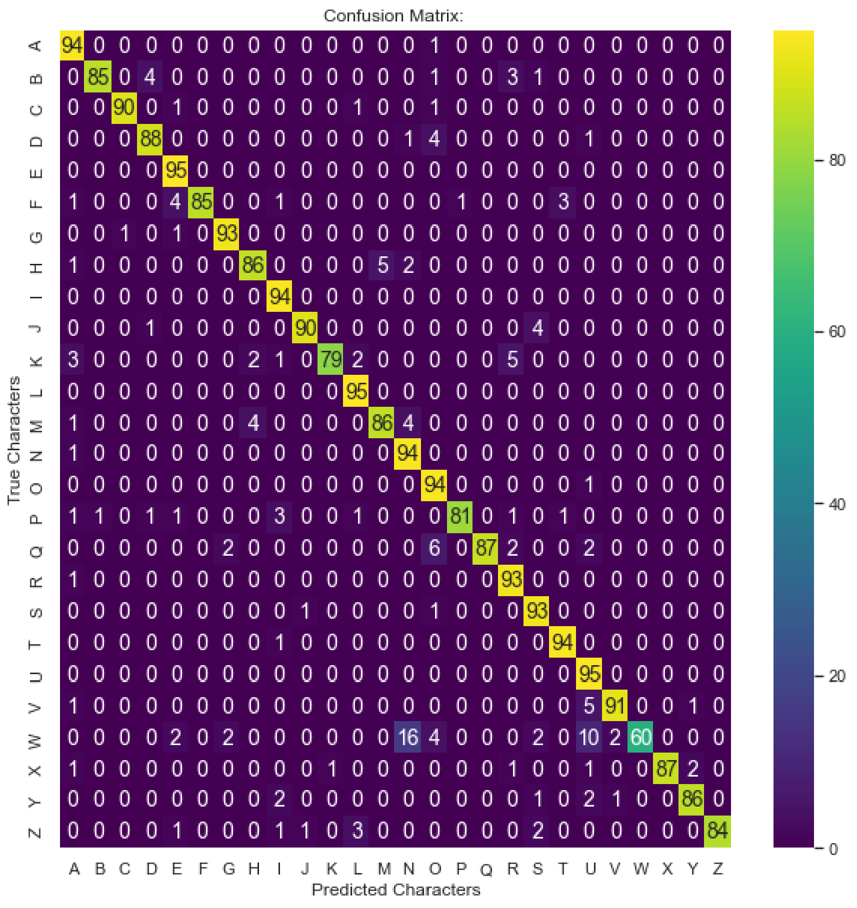
- Fitted – with 30K Images.
- It took nearly 5 hours to train the model.
- 60 epochs , each 30K images.
- Made use of GPU for Image processing.
- Tested with 3K Images.
- Without High Contrast, the model gave an accuracy of 85% of character prediction accuracy
- With using High Contrast colours, it gave 91% character prediction accuracy

Classification Report (3000 Images)

```
In [25]: print(classification_report(y_true_char,y_pred_char))
              precision    recall  f1-score   support

                   0.76      0.67      0.71        84
           '       0.00      0.00      0.00         2
           -       0.66      0.75      0.70        51
           A       0.94      0.94      0.94      2433
           B       0.94      0.85      0.90       446
           C       0.92      0.91      0.91       623
           D       0.92      0.88      0.90       588
           E       0.95      0.95      0.95      2399
           F       0.94      0.86      0.90       166
           G       0.90      0.94      0.92       362
           H       0.88      0.86      0.87       513
           I       0.94      0.95      0.94      1587
           J       0.86      0.90      0.88       140
           K       0.85      0.80      0.82       119
           L       0.95      0.95      0.95      1439
           M       0.90      0.86      0.88       740
           N       0.93      0.94      0.93      1506
           O       0.90      0.94      0.92      1240
           P       0.88      0.82      0.85       251
           Q       0.88      0.88      0.88        48
           R       0.93      0.94      0.93      1314
           S       0.94      0.93      0.93       805
           T       0.94      0.94      0.94       914
           U       0.89      0.95      0.92       869
           V       0.93      0.91      0.92       225
           W       0.76      0.60      0.67        48
           X       0.96      0.87      0.92       118
           Y       0.88      0.87      0.87       267
           Z       0.90      0.85      0.87       113

    accuracy                           0.93     19410
   macro avg       0.86      0.84      0.85     19410
weighted avg       0.93      0.93      0.92     19410
```



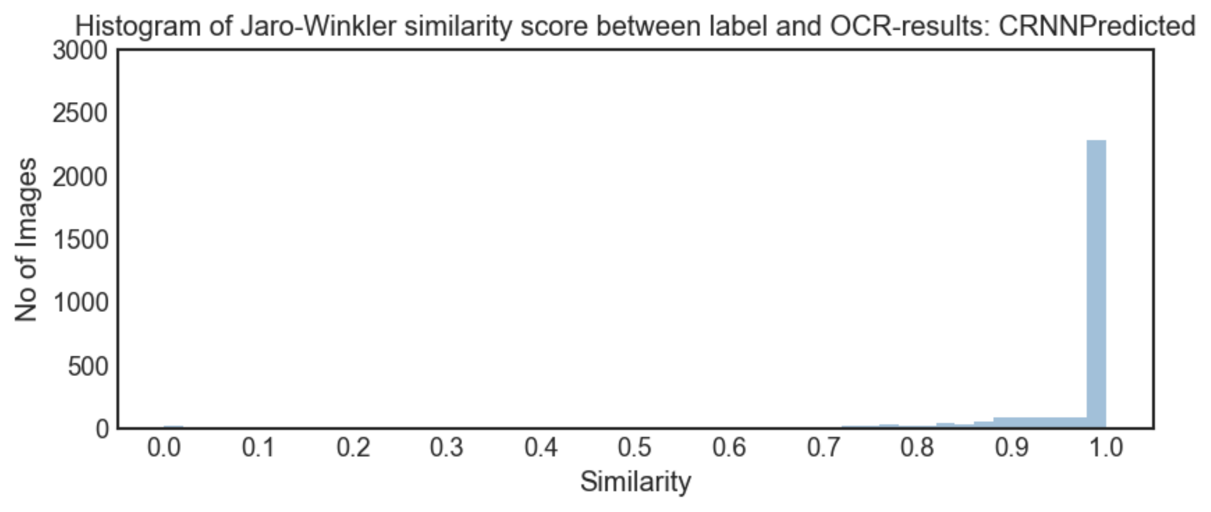

```
Correct characters predicted : 91.25%
Correct words predicted      : 75.17%
```

Similarity, Error Metrics

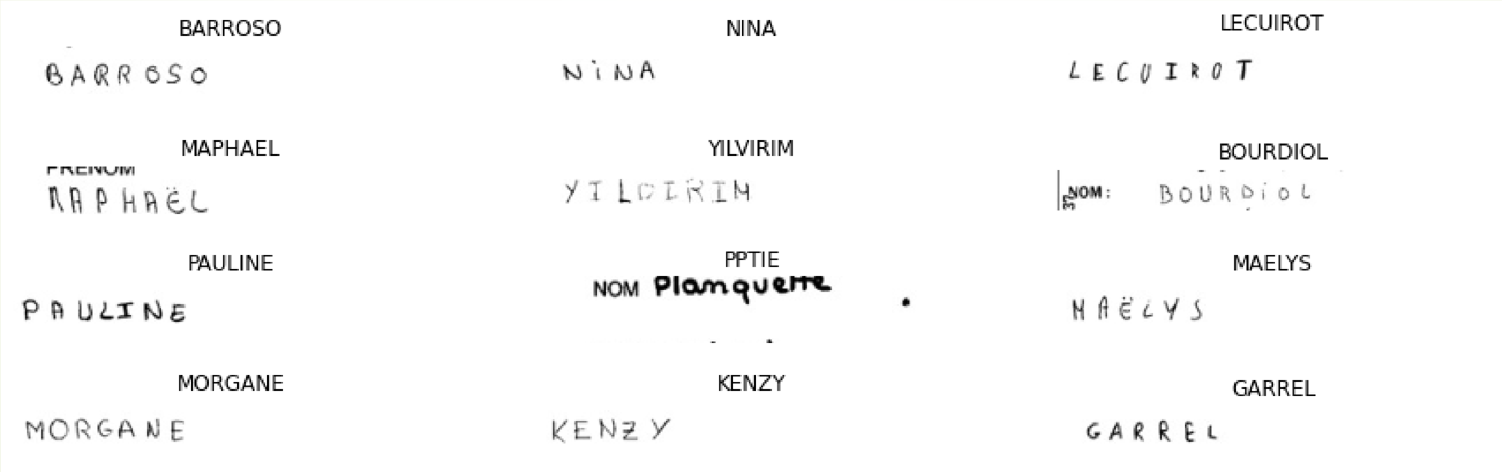
```
Mean Squared Error:  CRNNPredicted   11.183153013910356
Similarity Score between True Label and Predicted Label: CRNNPredicted
```

	CRNNPredicted	IDENTITY	SIMILARITY_SCORE
0	BILEL	BILEL	1.000000
1	LAUMONIER	LAUMIONIER	0.946667
2	LEA	LEA	1.000000
3	JEAN-ROCH	JEAN-ROCH	1.000000
4	RUPP	RUPP	1.000000
5	PICHON	PICHON	1.000000
6	DANIEL	DANIEL	1.000000
7	JEREMY	JEREMY	1.000000
8	JEAN-MICHEL	JEAN-MICHEL	1.000000
9	JULIEN	JULIEN	1.000000

```
Histogram of Similarity: CRNNPredicted
```



Sample Predicted Output



Comparison (RNN vs CNN vs CRNN)

- Likewise, Created RNN Model, CNN Model (Individually).
- Trained all these three models with same 30K images.
- Tested it on 3.3 Lakhs Images.

How do these models(CNN,RNN) look like ?

CNN Model
Model: "model_1"

Layer (type)	Output Shape	Param #
input (InputLayer)	[(None, 256, 64, 1)]	0
conv1 (Conv2D)	(None, 256, 64, 32)	320
batch_normalization_3 (Batch	(None, 256, 64, 32)	128
activation_3 (Activation)	(None, 256, 64, 32)	0
max1 (MaxPooling2D)	(None, 128, 32, 32)	0
conv2 (Conv2D)	(None, 128, 32, 64)	18496
batch_normalization_4 (Batch	(None, 128, 32, 64)	256
activation_4 (Activation)	(None, 128, 32, 64)	0
max2 (MaxPooling2D)	(None, 64, 16, 64)	0
dropout_2 (Dropout)	(None, 64, 16, 64)	0
conv3 (Conv2D)	(None, 64, 16, 128)	73856
batch_normalization_5 (Batch	(None, 64, 16, 128)	512
activation_5 (Activation)	(None, 64, 16, 128)	0
max3 (MaxPooling2D)	(None, 64, 8, 128)	0
dropout_3 (Dropout)	(None, 64, 8, 128)	0
reshape (Reshape)	(None, 64, 1024)	0
dense2 (Dense)	(None, 64, 30)	30750
softmax (Activation)	(None, 64, 30)	0

Total params: 124,318
Trainable params: 123,870
Non-trainable params: 448

RNN Model
Model: "model_4"

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, 16384)]	0
reshape_2 (Reshape)	(None, 64, 256)	0
dense_3 (Dense)	(None, 64, 64)	16448
lstm_4 (LSTM)	(None, 64, 256)	328704
lstm_5 (LSTM)	(None, 64, 256)	525312
dense_4 (Dense)	(None, 64, 30)	7710
activation_2 (Activation)	(None, 64, 30)	0

Total params: 878,174
Trainable params: 878,174
Non-trainable params: 0

How do the predictions look like?

	Index	FILENAME	IDENTITY	CRNNPredicted	CNNPredicted	RNNPredicted
0	0	TRAIN_00001.jpg	BALTHAZAR	BALTHAZAR	BALTHAZAR	BLALTHAIAR
1	1	TRAIN_00002.jpg	SIMON	SIMON	SIMON	BLIMON
2	2	TRAIN_00003.jpg	BENES	BENES	BENES	BVENES
3	3	TRAIN_00004.jpg	LA LOVE	LALOUE	LALOUE	BLALOUE
4	4	TRAIN_00005.jpg	DAPHNE	DAPHNE	DAPHNE	CMAPHNE

	Index	FILENAME	IDENTITY	CRNNPredicted	CNNPredicted	RNNPredicted
330287	330287	TRAIN_330957.jpg	LENNY	LENNY	LENNY	BLENNY
330288	330288	TRAIN_330958.jpg	TIFFANY	TIFFANY	TIEEANY	BLIEFANY
330289	330289	TRAIN_330959.jpg	COUTINHO DESA	COUTINHO DESA	COUTINHODESA	BCOUTINO DEA
330290	330290	TRAIN_330960.jpg	MOURAD	MOURAD	AOURAD	BAOURAD
330291	330291	TRAIN_330961.jpg	HELOISE	HELOISE	HELOISE	BLELOISE

```
No of Images Tested with  330292
```

What have we found in the Comparison ?

Prediction Result for CRNNPredicted
No of Images 330292
No of Characters over all Images 2134747
Correct characters predicted : 92.98%
Correct words predicted : 75.56%
Classification Report: CRNNPredicted

	precision	recall	f1-score	support
	0.76	0.63	0.69	10100
#	0.00	0.00	0.00	1
'	0.61	0.22	0.32	230
-	0.66	0.72	0.69	6493
?	0.00	0.00	0.00	0
A	0.95	0.95	0.95	266965
B	0.93	0.88	0.91	45634
C	0.91	0.93	0.92	67580
D	0.91	0.89	0.90	59587
E	0.95	0.96	0.95	266361
F	0.93	0.88	0.90	18135
G	0.90	0.90	0.90	38807
H	0.90	0.89	0.89	61232
I	0.95	0.95	0.95	171683
J	0.90	0.89	0.90	15745
K	0.91	0.86	0.88	13880
L	0.95	0.95	0.95	159677
M	0.90	0.86	0.88	83273
N	0.93	0.94	0.94	164645
O	0.92	0.94	0.93	134912
P	0.90	0.86	0.88	29106
Q	0.88	0.80	0.84	5131
R	0.94	0.93	0.94	146178
S	0.95	0.95	0.95	89762
T	0.93	0.94	0.94	99376
U	0.90	0.95	0.93	95253
V	0.91	0.87	0.89	24865
W	0.85	0.71	0.78	5464
X	0.94	0.90	0.92	11116
Y	0.88	0.86	0.87	29630
Z	0.92	0.88	0.90	13925
`	0.00	0.00	0.00	1
accuracy			0.93	2134747
macro avg	0.81	0.78	0.79	2134747
weighted avg	0.93	0.93	0.93	2134747

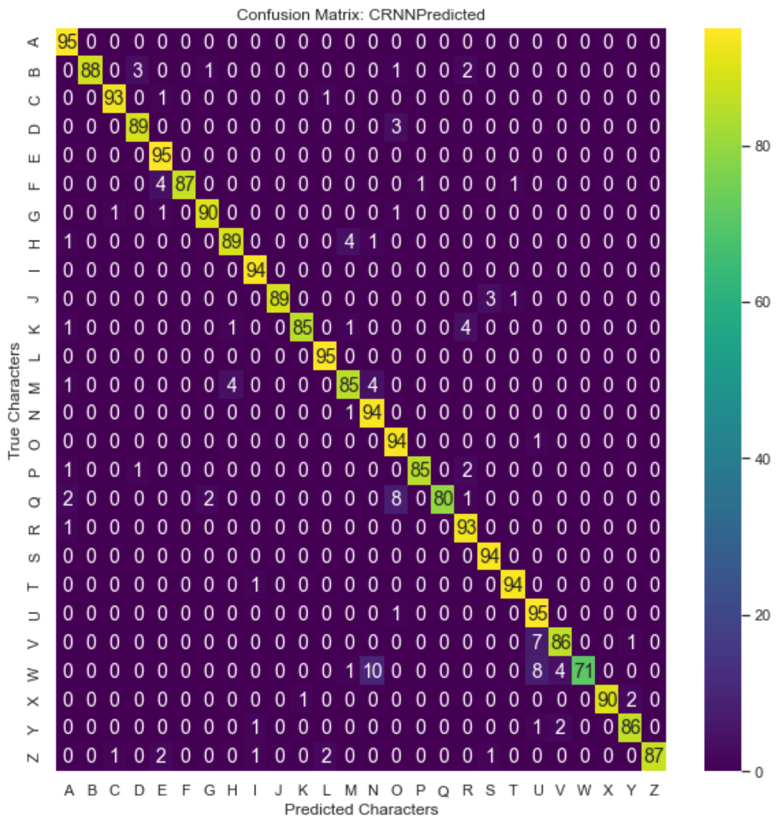
Prediction Result for CNNPredicted
No of Images 330292
No of Characters over all Images 2090486
Correct characters predicted : 84.25%
Correct words predicted : 59.54%
Classification Report: CNNPredicted

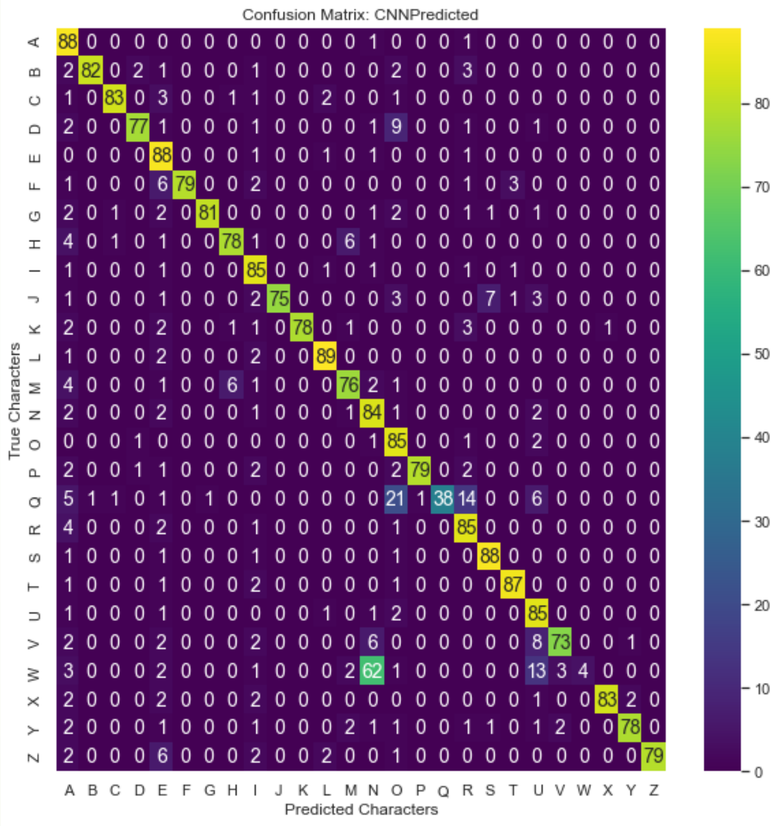
	precision	recall	f1-score	support
	0.01	0.00	0.00	9874
#	0.00	0.00	0.00	1
'	0.00	0.00	0.00	217
-	0.58	0.51	0.55	6406
A	0.86	0.88	0.87	262418
B	0.85	0.82	0.83	45311
C	0.83	0.84	0.84	66669
D	0.83	0.77	0.80	58235
E	0.87	0.88	0.88	258264
F	0.82	0.79	0.81	17890
G	0.85	0.81	0.83	38457
H	0.77	0.79	0.78	60455
I	0.84	0.86	0.85	168646
J	0.86	0.75	0.80	15684
K	0.84	0.78	0.81	13612
L	0.89	0.89	0.89	157627
M	0.80	0.77	0.78	81691
N	0.84	0.85	0.84	159273
O	0.80	0.86	0.83	132511
P	0.84	0.80	0.82	28657
Q	0.77	0.39	0.51	5065
R	0.84	0.85	0.85	143343
S	0.87	0.88	0.87	86890
T	0.88	0.88	0.88	96796
U	0.79	0.86	0.82	93805
V	0.79	0.73	0.76	24637
W	0.57	0.04	0.08	5405
X	0.84	0.83	0.84	10624
Y	0.84	0.78	0.81	28490
Z	0.85	0.79	0.82	13532
`	0.00	0.00	0.00	1
accuracy			0.84	2090486
macro avg	0.71	0.67	0.68	2090486
weighted avg	0.84	0.84	0.84	2090486

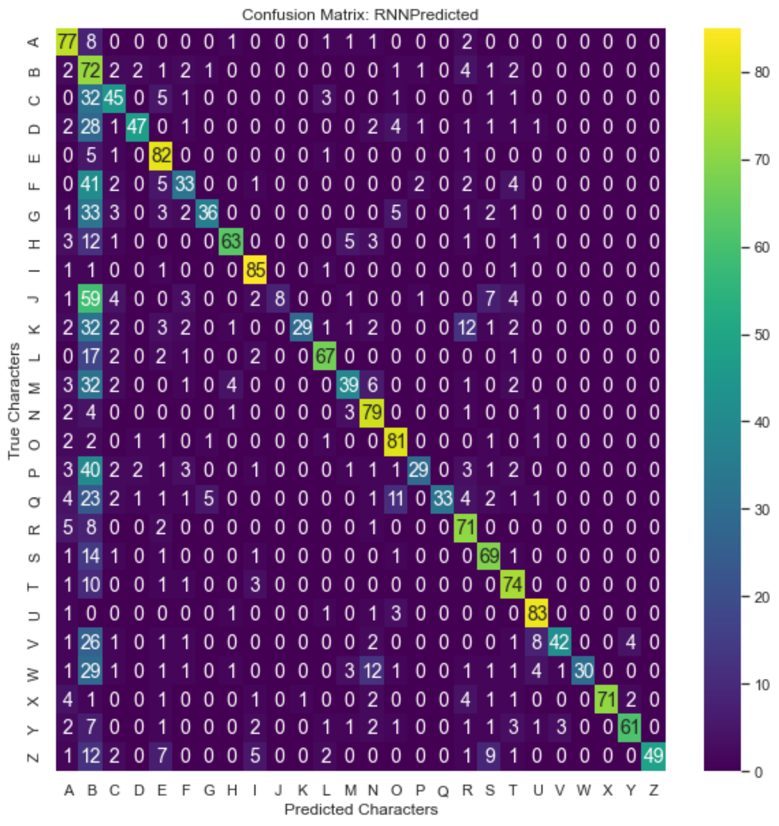
Prediction Result for RNNPredicted
No of Images 330292
No of Characters over all Images 2144334
Correct characters predicted : 69.81%
Correct words predicted : 0.26%
Classification Report: RNNPredicted

	precision	recall	f1-score	support
	0.56	0.42	0.48	9877
#	0.00	0.00	0.00	1
'	0.45	0.10	0.16	216
-	0.40	0.44	0.42	6474
?	0.00	0.00	0.00	1
A	0.84	0.77	0.81	266860
B	0.12	0.72	0.20	45613
C	0.52	0.45	0.48	67799
D	0.73	0.48	0.58	60078
E	0.87	0.82	0.85	268372
F	0.24	0.33	0.28	18198
G	0.64	0.37	0.47	38919
H	0.73	0.63	0.67	61191
I	0.87	0.85	0.86	172547
J	0.27	0.08	0.13	15759
K	0.59	0.30	0.39	14022
L	0.83	0.68	0.74	160140
M	0.60	0.39	0.47	82771
N	0.84	0.79	0.81	166194
O	0.83	0.81	0.82	134731
P	0.47	0.29	0.36	29025
Q	0.56	0.33	0.41	5134
R	0.79	0.72	0.75	146995
S	0.78	0.69	0.73	90876
T	0.75	0.74	0.74	100754
U	0.83	0.83	0.83	95309
V	0.56	0.42	0.48	24921
W	0.57	0.31	0.40	5481
X	0.82	0.71	0.76	11300
Y	0.78	0.61	0.69	30703
Z	0.69	0.50	0.58	14072
`	0.00	0.00	0.00	1
accuracy			0.70	2144334
macro avg	0.58	0.49	0.51	2144334
weighted avg	0.76	0.70	0.72	2144334

Confusion Matrix







Loss & Similarity

Mean Squared Error: CRNNPredicted   10.623822869876383
Similarity Score between True Label and Predicted Label: CRNNPredicted

	CRNNPredicted	IDENTITY	SIMILARITY_SCORE
0	BALTHAZAR	BALTHAZAR	1.000000
1	SIMON	SIMON	1.000000
2	BENES	BENES	1.000000
3	LALOUE	LALOVE	0.933333
4	DAPHNE	DAPHNE	1.000000
5	LUCIE	LUCIE	1.000000
6	NASSIM	NASSIM	1.000000
7	ASSRAOUI	ASSRAOUI	1.000000
8	VLAVIAN	LAVIAN	0.869048
9	MAEVA	MAEVA	1.000000

Histogram of Similarity: CRNNPredicted

Mean Squared Error: CNNPredicted   25.33519956603393
Similarity Score between True Label and Predicted Label: CNNPredicted

	CNNPredicted	IDENTITY	SIMILARITY_SCORE
0	BALTHAZAR	BALTHAZAR	1.000000
1	SIMON	SIMON	1.000000
2	BENES	BENES	1.000000
3	LALOUE	LALOVE	0.933333
4	DAPHNE	DAPHNE	1.000000
5	LUCIE	LUCIE	1.000000
6	NASSIM	NASSIM	1.000000
7	ASSRAOUI	ASSRAOUI	1.000000
8	MLAVIAN	LAVIAN	0.952381
9	MAEVA	MAEVA	1.000000

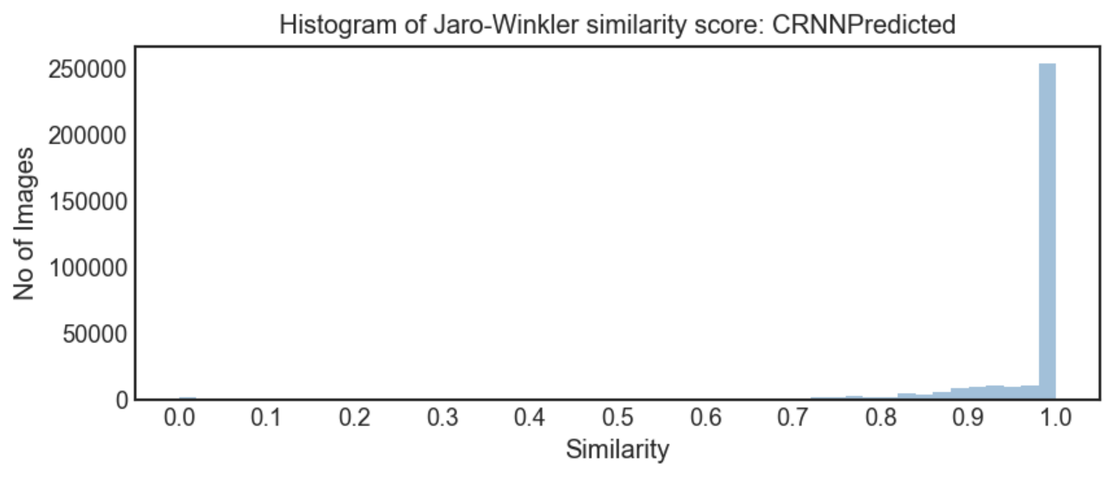
Histogram of Similarity: CNNPredicted

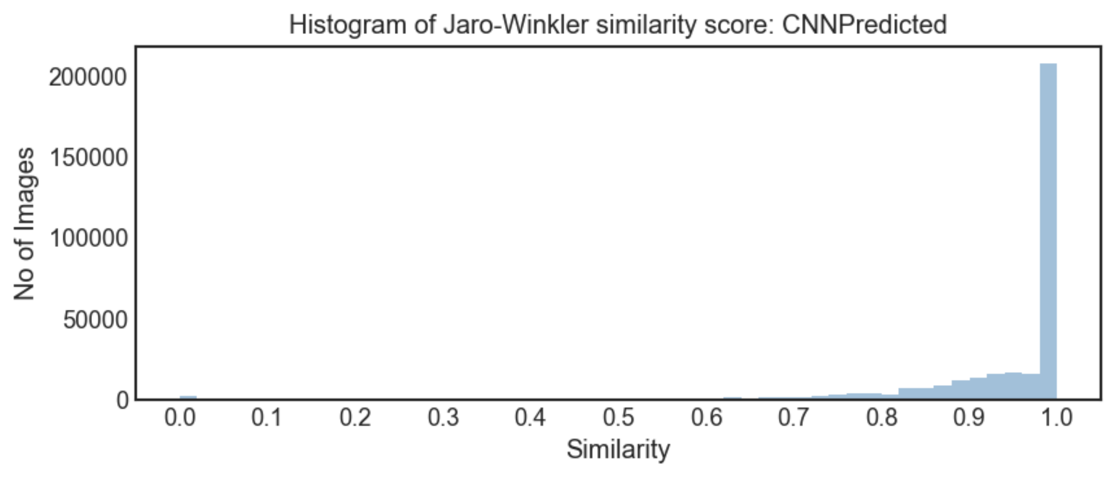
Mean Squared Error: RNNPredicted   41.535935166816365
Similarity Score between True Label and Predicted Label: RNNPredicted

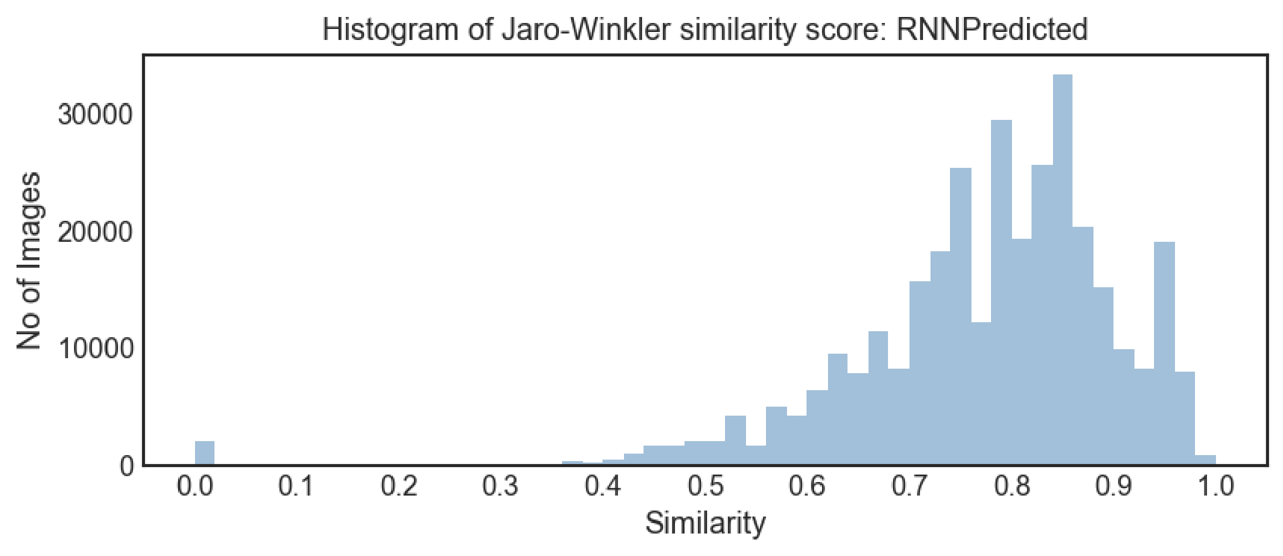
	RNNPredicted	IDENTITY	SIMILARITY_SCORE
0	BLALTHAIAR	BALTHAZAR	0.869167
1	BLIMON	SIMON	0.822222
2	BVENES	BENES	0.950000
3	BLALOUE	LALOVE	0.849206
4	CMAPHNE	DAPHNE	0.849206
5	BLUCIE	LUCIE	0.944444
6	BCMASSIM	NASSIM	0.652778
7	TFBSSRAOUI	ASSRAOUI	0.858333
8	BALAVRAN	LAVIAN	0.686111
9	BLAEVA	MAEVA	0.822222

Histogram of Similarity: RNNPredicted

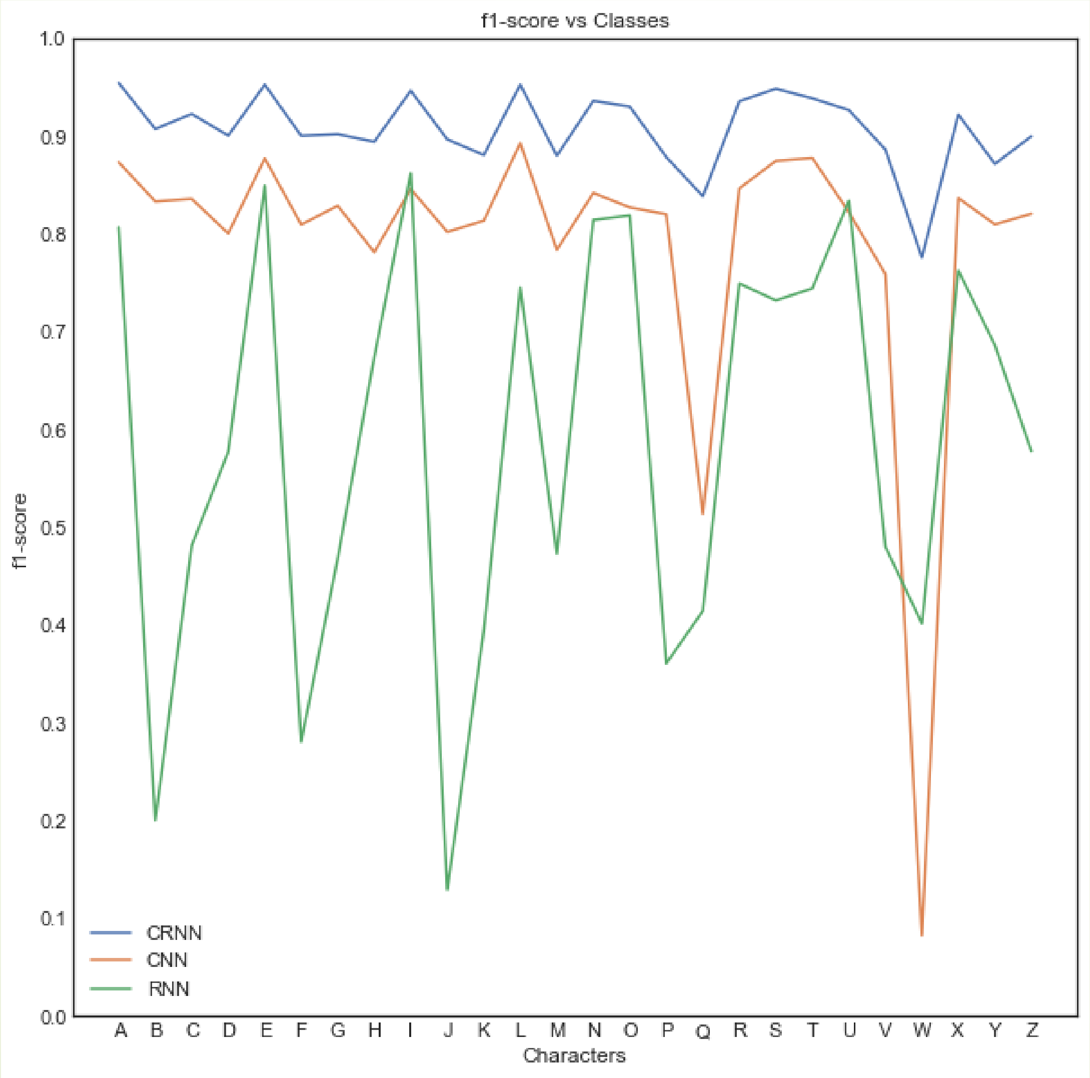
Histogram of Similarity



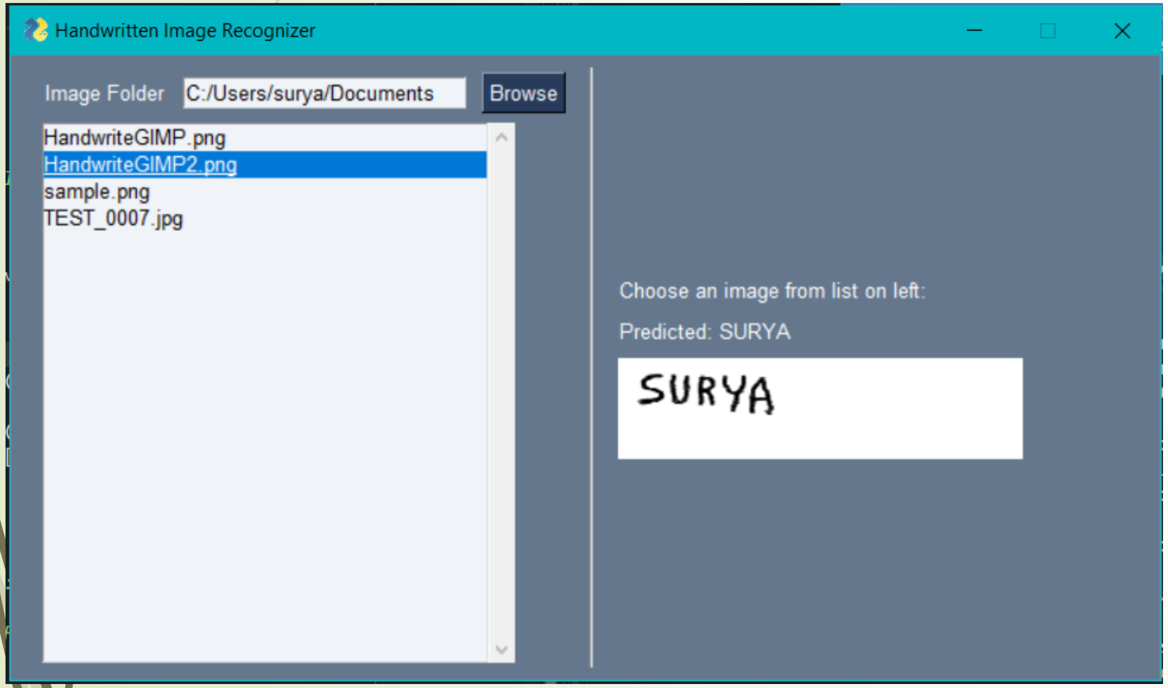


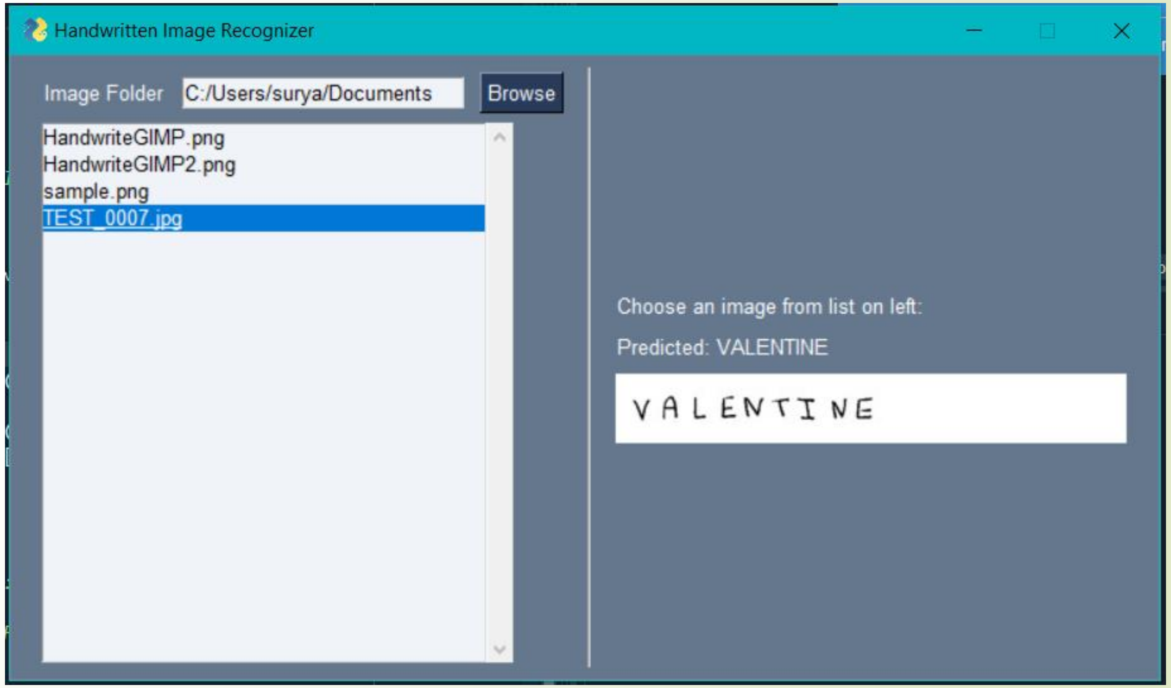


F1-score Plot

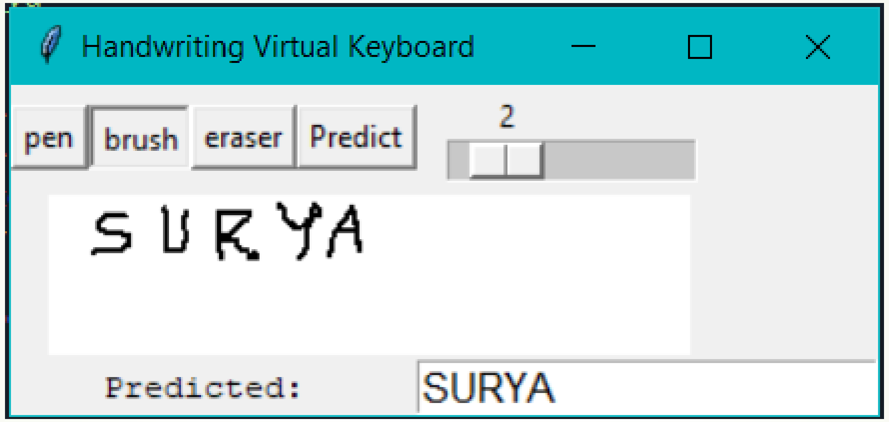


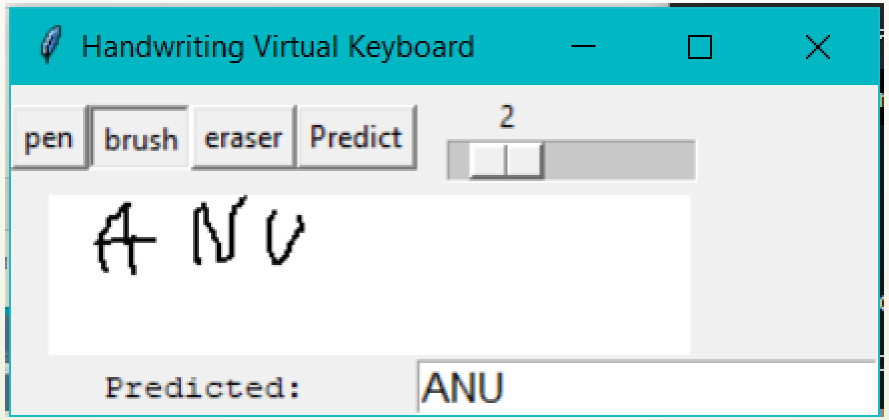
Handwritten Image Recognizer (GUI)





Handwriting Virtual Keyboard (GUI)





Conclusion

- Hence, it is concluded that CRNN performs better than CNN & RNN.
- Pre-processing has effects on prediction.
- Choosing a proper model layers has its effects on prediction.
- Fitting(Training) of images has its effects. (Over fitting etc).
- Plot Evaluation of Prediction gives more information.
- Two GUI is built for use with the best model (CRNN) found.

References

- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c
- https://www.kaggle.com/landlord/handwriting-recognition

Thank You