**INTERNSHIP: PROJECT REPORT**

------------------------------------------------------------------------------------------------------------------------------------------

|  |  |
| --- | --- |
| Internship Project Title | RIO-125: AUTOMATIC EXTRACTION OF HANDWRITTEN TEXT FROM AN IMAGE |
| Project Title | DEEP LEARNING CLASSIFICATION ALGORITHM TO PREDICT HANDWRITTEN TEXT FROM AN IMAGE |
| Name of the Company | TATA CONSULTANCY SERVICES |
| Name of the Industry Mentor | MR.DEBASHIS ROY |
| Name of the Institute | MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
| 15-04-2022 | 24-04-2022 | 30 | Google Colab, Visual Studio code(jupyter notebook),Windows 10 | Python 3(Tensorflow, Keras) |

|  |
| --- |
| PROJECT SYNOPSIS:  It is widely used method to produce effective communication between Hand Documents and computers. Mostly we use OCR to extract text from different kinds of documents. We know there are two types of Handwritten text extraction from image offline mode and online mode, Where dataset is taken, follow a Deep learning classification algorithm is called Offline mode. Handwriting Recognition which is still considered a challenging problem statement. The high variance in handwriting styles across people and poor quality of the handwritten text compared to printed text pose significant hurdles in converting it to machine readable text. Nevertheless it's a crucial problem to solve for multiple industries like healthcare, insurance and banking. |

|  |
| --- |
| ASSUMPTIONS:  I Had imported the dataset from Kaggle. I had downloaded the files of the datasets in the notebook in runtime from API tokens. It useful while working notebooks in cloud platforms like colab, google virtual machines, etc. |
| PROJECT DIAGRAMS:  MODEL: |
| ALGORITHM:  Step 1:Cloned the Github repository using “git clone”. To download the Kaggle API token.  Step 2: Created a Directory for Kaggle file and Dataset download.  Step 3: Imported the Required Libraries to build algorithm.  Step 4:Csv files of images data had imported and pre-processed.  Step 5:Had Data Exploratory analysis with data visualization.  Step 6.Again reassign csv file and data files form training data and validation data.  Step 5: Build the model with 4 layers.   1. The first layer is input layer and CNN layer as CONV2D(32, (3, 3)) with BatchNormaization, Activation layer of relu and a Maxpooling2d layer(2, 2) . 2. The Second Layer CNN layer as CONV2D(64, (3, 3)) with BatchNormaization, Activation layer of relu and a Maxpooling2d layer(2, 2). And it also has with Dropout layer of 0.3. Same as Third layer with conv3 name. 3. The Fourth layer had different approach with RNN layer. We use to make model more accurate. The layer consists of two bidirectional LSTM with dense layer and activation function of Softmax   Step7: Compile the model with the loss function and optimizer. Here optimizer as “ADAM”.  Step 8:Fit the model train data, with epochs =12 , making a validation with test data.  Here  Epoch 1/15  235/235 [==============================] - 126s 500ms/step - loss: 24.5291 - val\_loss: 20.6190  Epoch 2/15  235/235 [==============================] - 114s 487ms/step - loss: 20.1761 - val\_loss: 20.0821  Epoch 3/15  235/235 [==============================] - 115s 489ms/step - loss: 19.8377 - val\_loss: 19.7400  Epoch 4/15  235/235 [==============================] - 115s 488ms/step - loss: 19.4342 - val\_loss: 19.3906  Epoch 5/15  235/235 [==============================] - 115s 490ms/step - loss: 18.7245 - val\_loss: 18.3208  Epoch 6/15  235/235 [==============================] - 114s 486ms/step - loss: 17.5416 - val\_loss: 16.9298  Epoch 7/15  235/235 [==============================] - 115s 488ms/step - loss: 15.9787 - val\_loss: 15.3284  Epoch 8/15  235/235 [==============================] - 114s 487ms/step - loss: 14.1126 - val\_loss: 14.5479  Epoch 9/15  235/235 [==============================] - 114s 486ms/step - loss: 11.9024 - val\_loss: 15.6335  Epoch 10/15  235/235 [==============================] - 114s 485ms/step - loss: 9.4170 - val\_loss: 14.9164  Epoch 11/15  235/235 [==============================] - 114s 487ms/step - loss: 7.5285 - val\_loss: 11.1751  Epoch 12/15  235/235 [==============================] - 115s 488ms/step - loss: 6.3317 - val\_loss: 9.1714  Epoch 13/15  235/235 [==============================] - 114s 486ms/step - loss: 5.5405 - val\_loss: 9.0098  Epoch 14/15  235/235 [==============================] - 116s 493ms/step - loss: 5.0015 - val\_loss: 7.0696  Epoch 15/15  235/235 [==============================] - 114s 486ms/step - loss: 4.5772 - val\_loss: 6.7395  Step 8: Predicted handwritten text from an image. |
| EXPECTATIONS CONSIDERED:  Accuracy & Loss Function values vary slightly on every compilation of the model. This may give a slight difference in the prediction value which may impact the neutral network. In this model if we train with more epochs we can build more accurate and decrease loss. |
| ENHANCEMENT SCOPE:  Increasing the accuracy is the biggest enhancement scope. This can be achieved by adjusting the CNN layers with RNN to more accurate results. Three “Conv2d” layers is mixed with two “LSTM” layers. |
| LINKS TO CODE AND EXCUTABLE FILE:  Link of project work (notebook)  [Colab\_Link](https://colab.research.google.com/drive/12dQgo_WcifPTF3NTRCazbajRl5GlvY-d?usp=sharing) |