|  |  |
| --- | --- |
| Internship Project Title | TCS iON RIO – 125 Automate extraction of hand written text from an image. |
| Project Title | Automate extraction of hand written text from an image |
| Name of the Company | TCS iON |
| Name of the Industry Mentor | Debashish Roy |
| Name of the Institute | St. Joseph’s College of Engineering and Technology, Palai |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
| 31-05-2021 | 29-06-2021 | 125 | Google Colab, Jupyter Notebook | Tensorflow==1.14.0  Keras==2.3.1  Google Colab GPU |
| Project Synopsis:   * Title of the project - Automate extraction of handwritten text from an image * Objective and aim – To develop machine learning algorithms in order to enable entity and knowledge extraction from documents with handwritten annotations with an aim to identify handwritten words on an image. * Project Guidelines – Prepare or collect some sample images containing handwritten text * The chosen images may include the following: * 1.Cursive handwriting * 2.Poor image quality generated from frequently scanned documents * 3.Skewed images | | | | |
| Solution Approach:  The main objective of the project is to automatically extract handwritten text from an image. The most common approach is go for is Convolution Neural Network (CNN).  For implementing my project I used the IAM dataset, Google Colab Virtual GPU, Tensorflow == 1.14.0, Keras ==2.3.1, Numpy library, Ski-kit learn library, Matplotlib library, Pandas library, Seaborn library.  The following steps were followed:  1.Collecting and importing the datasets on google drive  2.Preprocessing data  3.Creating network architecture  4.Training model  5.Testing and Prediction  1)Collecting and importing the datasets:  This is one of the important step in the project. The datasets I used was the IAM datasets. This is a very large dataset. So it was better to import it in google drive. This data consist of text images which is shown as below:    2) Preprocessing data:  We have preprocess both the input images and output labels   * Read the image and convert into gray-scale image * Normalize the image pixel with dividing it with 255   To preprocess output labels  Read the text from the forms\_for\_parsing.txt. This file contains every image text  Compute the maximum length from words and pad every output label to make it the same size as the maximum length.  3) Creating network architecture:  Here I used three convolutional layers. Also we used normalization layers and lambda function to increase the training process  4)Training model:  For training the model I used the Adam optimizer. Also we can use the keras callbacks functionality to save the weights of the best model on the basis of validation loss. In model. Now we have trained the model.  5)Testing and prediction:  Our model is now trained and now it is time to test the model. As our model predicts the probability for each class at each time step, we need some transcription function to convert it into actual text.  I also used other approaches but when we increased the complexity of the image then the result was not good. | | | | |
| Assumptions:  The assumptions considered are as follows:  1)The handwritten text must be in English  2)The image should not be tilted  3)Only image is provided for text recognition  4)All libraires must be compatible and of the latest version | | | | |
| Project Diagrams:  1)  2)    3)    4)    5) | | | | |
| Algorithms:  A Convolutional Neural Network (CNN) is a deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower compared to other classification algorithm  Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts , are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in CNN each layer generates several activation functions that are passed on to the next layer.  The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces etc. | | | | |
| Outcome:  The following is the model architecture obtained:  <https://colab.research.google.com/drive/1u-27AIEOhqPACN4DrW08cdYbFTRIy9x3#scrollTo=ON2qu9TDy7p-&line=1&uniqifier=1>  Worked till this segment  Model: “sequential\_1”  Layer (type) Output Shape Param #  zero\_padding2d\_1 (ZeroPaddin (None, 115, 115, 1) 0  lambda\_1 (Lambda) (None, 56, 56, 1) 0  conv1 (Conv2D) (None, 28, 28, 32) 832  activation\_1 (Activation) (None, 28, 28, 32) 0  pool1 (MaxPooling2D) (None, 14, 14, 32) 0  conv2 (Conv2D) (None, 14, 14, 64) 18496  activation\_2 (Activation) (None, 14, 14, 64) 0  pool2 (MaxPooling2D) (None, 7, 7, 64) 0  conv3 (Conv2D) (None, 7, 7, 128) 73856  activation\_3 (Activation) (None, 7, 7, 128) 0  pool3 (MaxPooling2D) (None, 3, 3, 128) 0  flatten\_1 (Flatten) (None, 1152) 0  dropout\_1 (Dropout) (None, 1152) 0  dense1 (Dense) (None, 128) 147584  activation\_4 (Activation) (None, 128) 0  dropout\_2 (Dropout) (None, 128) 0  dense2 (Dense) (None, 64) 8256  activation\_5 (Activation) (None, 64) 0  dropout\_3 (Dropout) (None, 64) 0  output (Dense) (None, 50) 3250  activation\_6 (Activation) (None, 50) 0  Total params: 252,274  Trainable params: 252,274  Non-trainable params: 0  None  The algorithm was able to detect and extract handwritten text from an image with an overall accuracy of 91.72% while implementing and testing.    Extracted word: boycotting  The model is quite accurate in extracting the text  The following are the reference links used:  <https://software.intel.com/en-us/ai/courses/artificial-intelligence>  <https://software.intel.com/en-us/ai/courses/machine-learning>  <https://www.python-course.eu/machine_learning.php>  <https://numpy.org/doc/>  <https://www.scipy.org/docs.html>  <https://scikit-learn.org/stable/tutorial/index.html>  <https://pandas.pydata.org/pandas-docs/version/0.15/tutorials.html>  <https://www.tensorflow.org/tutorials>  <https://www.tensorflow.org/tutorials/keras/classification>  <https://www.tensorflow.org/tutorials/images/cnn>  <https://www.tensorflow.org/tutorials/text/text_classification_rnn>  <https://www.tensorflow.org/tutorials/images/classification>  <https://software.intel.com/en-us/ai/courses/deep-learning> | | | | |
| Exceptions considered:   1. The text across the input image must be of the same color not multicolor handwritten text 2. The image should not have aggressive multicolor backgrounds across the text of the image 3. The image should not have too much aggressive background noises across the text of the image 4. The image should not be tilted or rotated 5. The libraries installed should be compatible to meet its requirements | | | | |
| Enhancement Scope:  The following are the enhancements scope for this project:   1. The accuracy of the model can be increased with predefined models and also powerful machine learning GPU processors can be used to attain good percentage of accuracy. 2. In the near future we can use this algorithm with more than one particular language. 3. This model can also be used for paragraph extraction if we increase the number of CNN and its pre processing layers. 4. This algorithm can be used in extracting text from a video if we combine it with OpenCV algorithms | | | | |
| Link to Code and executable file:  <https://colab.research.google.com/drive/1u-27AIEOhqPACN4DrW08cdYbFTRIy9x3?usp=sharing> | | | | |
|  | | | | |
|  | | | | |