Optical Character Recognition for a Redaction System Using Machine Learning Techniques.

M. Kanchana, Muskan Sharma and Hrithik Somani

May 28, 2020

Optical Character Recognition for a Redaction System Using Machine Learning Techniques.

Dr M. Kanchana[1,]Muskan Sharma[2], Hrithik Somani[3]
[1] Associate Professor,[2] [3]Senior Year Graduate,

Department of Computer Science Engineering, SRM Institute of Science and Technology.

Abstract

This paper presents the use of OCR in an automatic Redaction System. A Redactor is a system which takes in any electronic document as an input from the user and identifies sensitive information, mainly nouns, such as: Person name, country name, gender, credit card information, phone numbers, email id, any confidential information that is to be not shown to the end user who the document is to be sent to. Initially, the user inputs a document, probably an image. This image is then pre-processed and put into the OCR which extracts the text out of the image. Hence, to be able to identify the sensitive information the very first step is to extract the information. A major application of an OCR is Redaction. Reading of information present in the documents can be read with the help of an OCR Machine.

Keywords: *Machine Learning, Natural Language Processing, Optical Character Recognition, Named Entity Recognition.*

1. Introduction

Redaction is a system designed to identify sensitive, confidential information's in documents and redact them so while sending any such information the end user remains unaware of the confidential information. This system or model is used in many organizations, but only semi – automated systems are being used, semi – automated systems return a list of sensitive words using NLP tools, once the words are returned the user needs to select all the words that he or she wants redacted from the document. On selection the user can click on redact in the interface and a document with the user selected words redacted is returned. Automating this process would eliminate the step where user gets to select any information that he/she wants to redact from the document and simply a redacted document with removal of any sensitive information is returned.

Now, Optical Character Recognition (OCR) is an important step in Redaction. Like OCR, Redaction has many other business applications where there is a need of information extraction. OCR is used for data entries, number plate recognition, passport identification or recognition. Earlier the systems needed to be trained using images of every character and worked with only one font at an instance. Now, the systems have advanced and they have the capability to produce better accuracy for most fonts and also support electronic documents.

2. Literature Survey

On surveying various research papers thoroughly, we came across the most useful applications of OCR and the evolution of the process of extracting text from images.

2.1 Applications of OCR:

OCR helps an incredible number of valuable applications. During the underlying days, OCR has been utilized for mail arranging, bank check perusing and mark confirmation. Also, OCR can be utilized by organizations just as association for mechanized structure handling in places where a lot of information is available in printed structure. OCR is additionally utilized in dealing with visa approval, service bills, robotized number plate recognition. Another helpful utilization of OCR is serving blind and outwardly weakened individuals to understand content. The recovery of the delicate data utilizing OCR would then be able to be evacuated with the goal that it can't be abused by other people who don't reserve the option to get to the data.

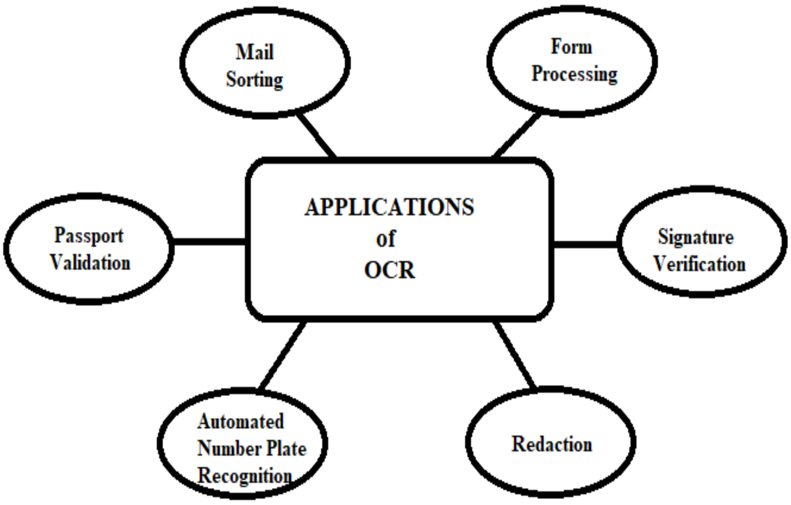


Figure 1: Application of OCR

2.2 Inference from the Literature Survey.

An OCR isn't a nuclear procedure however involves different stages, for example, procurement, pre-preparing, division, include extraction, characterization and post-handling. It is characterized as the way toward digitizing a record picture into its constituent characters'- Processing is done to improve the picture quality which incorporate clamour expulsion, morphological activities, slant evacuation and so on. CNN are cutting edge strategy utilized for include extraction. RNN are utilized for arrangement displaying. CRNN we pass on profound highlights into consecutive portrayals so as to be invariant to the length variety of arrangement like items. It handles arrangements in discretionary lengths. Before appropriating an archive, you might need to ensure the substance so delicate data stays classified.

Redacting data utilizing programming shrewdly expels the touchy data by setting a black box over the substance. Despite the fact that the content is still there it is truly inconceivable for outsiders to recover the data. So essentially, in the wake of doing the writing review of different papers, we surmised and saw how to join the various parts of AI and NLP and incorporate them in our undertaking. At first, we accept contribution as a picture separate content from the picture utilizing OCR which can be executed utilizing a mix of CNN and RNN and afterward play out a named element acknowledgment utilizing Hidden Markov Model.

3. PROPOSED WORK

The steps that are involved for our text recognition models are:

A. Collecting the Dataset
B. Pre-processing Data
C. Creating an appropriate Architecture model.
D. Finding the Loss Function best suited to our model.
E. Once, the above steps our complete we train the model.
F. The last step involves analysing and decoding the outputs.

A. Collecting the Dataset

For collecting the suitable Dataset for our problem statement, we used the dataset that is provided by the Visual Geometry Group, this is an enormous dataset that contains images of 10 GB. Our model uses 1, 35,000 images for training the model and 15,000 images as the validation set. This data set contains text images as shown in the fig.

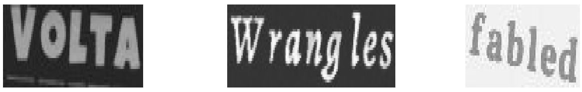


Figure 2: Dataset.

For training our model we have also used synthetic data as our training data. This dataset consists of 8+ million images and the corresponding labels for the words. These images are gathered using a synthetic text engine and are very realistic. We have trained our model using the synthetic dataset and tested it on the real dataset. Despite being trained on synthetic dataset it works quite well on the real-world images.
The IC13[24], this is a test dataset It contains 1.015 ground truths cropped word images. The IIIT5k [28] contains 3,000 cropped word test images that are collected from the internet. Each image is associated to a 50 words lexicon and also a 1k-words lexicon.

B. Pre-Processing

While pre-processing images, the input image as well as the output labels can be pre-processed depending upon the requirements. We have pre-processed both. We use different measures to do so for each of them, separately.

Firstly, to pre-process the input image:

i. Firstly, the image needs to be read.

ii. Once, the image is read it is converted into gray-scale.

iii. Uniform sizing of each image obtained is necessary, the size that we want for each image is 1,28,32, to achieve this padding is used.

iv. It is important for the image obtained from step iii. to be well- matched with the input architecture. This is usually obtained by expanding the dimensions of the images to 1,28,321.

v. The normalization of the values of image pixels is done by dividing them with a numeric value. For our model we use the numeric value 255.

Now, to pre-process the output label:

i. The name given to the image and the text contained within the image is exactly the same. Hence, we read the name given to the image to obtain the text inside it.

ii. On obtaining the name we encode each character. Encoding is done such that each character in a word is referred to as a consequent numeric value, this is done using a function (example: ‘a’ as 0, ‘b’ as 1, ‘c’ as 2,etc). For example, we have a word ‘acacba’ then the encoded label would be [0,2,0,2,1,0].

iii. It is necessary for the output labels to be well-matched to the RNN architecture. Hence, to do so the maximum length is computed amongst all the words and are made of the same size to ensure the compatibility remains intact.

C. Model Architecture

The model architecture that we are using is divided into 3 portions:

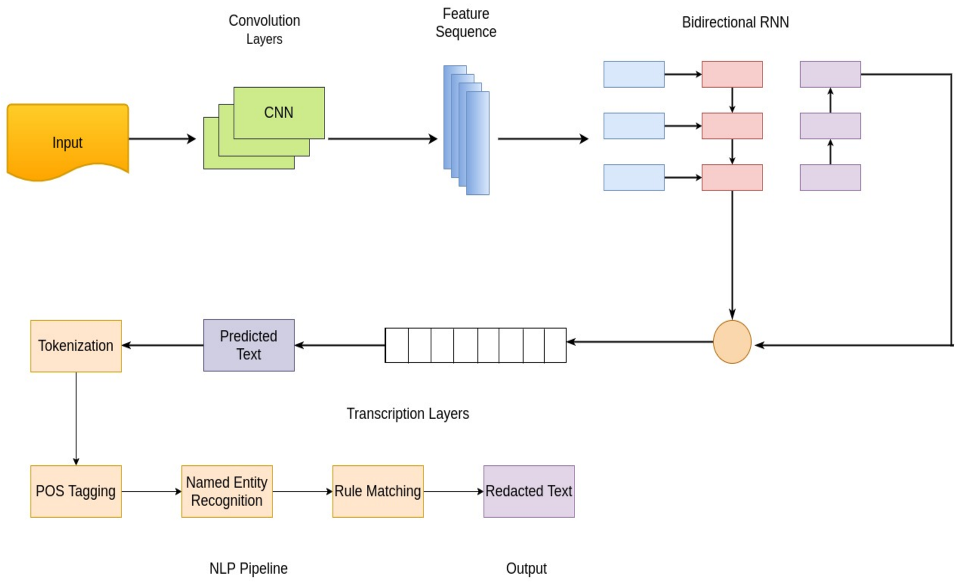


Figure 3: Model Architecture

i. CNN – Extracting features from the pre-processed images that are obtained.

ii. RNN – For the prediction of the consecutive outputs.

iii. The loss function – CTC, in the transcription layer used to anticipate.

Procedure we used for the creation of our architecture model:

i. Our architecture requires a height of thirty-two and width of one hundred and twenty-eight. Hence, the input image should be of the size required as per our model.

ii. Our model architecture requires 7 convolution layers. Across these 7 layers two kernel sizes are used, one is (3,3) and the other (2,2). The 1-6 layers use (3,3) size and the 7th (2,2). As the layers increase the number of filters involved also increases. From 64 at the first to 512 at the last. The number of filters becomes 512 layer by layer from 64, hence increases.

iii. To ensure that our architecture supports feature extraction for larger width and longer predictions, we add a total of four max-pooling layers in adding to the already existing 7 layers. Two of the layers added are of the size (2,2) and the remaining two of (2,1).

iv. The training process of the model can be time consuming, hence the acceleration of the same is necessary. For our architecture we used a technique after two layers, 5th and the 6th convolutional layers. The technique is batch normalization.

v. The output obtained from the Convolutional Layer needs to be well-matched with the LSTM layer. Lambda function can be used to do so.

vi. It squeezes the output and makes it well-matched to the LSTM layer.

The Major Phases involved in the OCR:

OCR is carried out in different stages basically a compound activity that include different phases. Each of the phases that are used in OCR are described below:

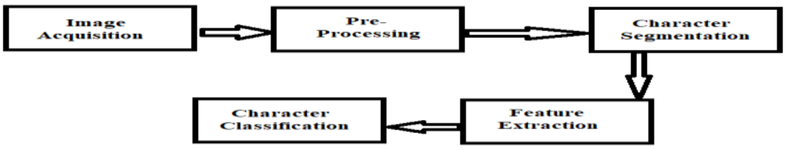


Figure 4: Major Phases of OCR.

Image Acquisition - There are various exterior sources from where the image is taken maybe by using phones, cameras or some kind of scanner.

Pre-Processing – Once the image has been collected the next important step is to pre-process the data as it improves the image quality which helps in better learning of the model. There are many pre-processing techniques such as noise removal, gaussian blur, morphological operations etc.

Segmentation – After pre-processing the images, the characters which are present in the images are alienated which if fed into recognition engine which is referred as Character Segmentation. These techniques include the usage of connected component analysis and projection profiles and other advance character segmentation techniques can be used.

Extraction of Features - The characters which are there in the images are segmented which is then processed to find out different features which helps to identify the characters. The features which are extracted should be competently.

Character Classification - This step maps the features image to the various classes. There are various methods: Classification based on structure of images and categorize characters with the help of decision rules. Statistical classification method which uses probabilities and other statistical methods to categorize the characters that are there in the images.

Algorithm Used

1. Feature Extraction

The convolution, max-pooling, and activation function work on neighbourhood locales, they are invariant and consequently every column of features maps compares to a rectangular area of the input picture called the Receptive field. At the base of architecture, the convolutional layers are utilized to learn the features from the input picture which is then fed to recurrent layers on top of convolution network to make accurate guesses for individual frame which corresponds to the extracted features. The transcription layer on the top of the model architecture translates frame prediction to a label sequence. In the above model, the convolutional layers are built using the layers from the existing CNN model namely convolution and max pooling layers whose top fully connected layers are not taken into account, this architecture is useful in extracting the sequential information from the input images which is called feature vector.

The height of each image is scaled to the same height which then allows to extract the vectors representing features from the feature maps generated by our network of CNN layers. The feature maps are scanned from left to right by columns which helps in generating the feature vector. The m- th column corresponds to the m-th feature vector.

ii. Sequence Labelling

Deep Bi-Directional Recurrent Neural Network. Contextual cues for image-based sequence identification, the Recurrent Neural Network (RNN) has the potential to capture information in a consecutive manner or serially. The RNN back propagates the errors to the convolutional layer, which is the input for the RNN. Due to this the RNN layers and the CNN layers can be trained simultaneously. RNN has the capabilities to process series of different lengths, starting the traversal from the beginning till the end.

A specialised network layer is made, called “Map-to-Sequence”, this acts as an intermediator link between the two types of layers [10]. A deep bidirectional RNN is followed by the convolution layers as recurrent layers, predict a label distribution. There are various advantages of RNN. Firstly, being able to capture of information in a consecutive or serial manner is done later the sameare used for image-based sequence recognition which is more stable. Secondly, as the RNN has the potential to back-propagate the errors to the convolutional layer, which actually is the input to the RNN, due to this back-propagation, both types of layers can be trained simultaneously rather than separately; Single Unified Network. Lastly, RNN has the ability to process series of different length in the traversal pattern from start to end.

iii. Transcription:

The process of transforming the every-frame predictions made by RNN model into a series of a label. Lexicon-free, estimate made without lexicon. Lexicon-based, estimate made taking the label sequence that has the maximum probability from a provided dictionary. Connectionist Temporal Classification (CTC) loss are used. Transcription converts the every-frame predictions made by RNN model into a series of a label. Mathematically it can be said that, transcription refers to finding the label sequence with highest probability conditioned on the pre-frame prediction. In use, there exist two modes, one is the lexicon-free transcription and the other is lexicon-based transcription. A lexicon refers to a set of label sequences that prediction is constraint.[10]

D. Loss Function

Once, the model architecture is prepared, the next step is to choose an appropriate Loss Function. For our model, we use the CTC (Connectionist Temporal Classification) loss function. The reason for choosing the CTC loss is because it is more helpful in text recognition problems, hence, appropriate for our problem statement. This loss function helps us to eliminate the issue in which one character has the abilities to span a more than one step. In case we decide on not using CT than one additional step will be required to deal with this issue separately. This function requires 4 parameters to be able to compute the loss. These 4 parameters are: 'the predicted outputs, the ground truth labels, the input sequence length for the LSTM, the ground truth label length.' Now, to be able to attain this we are required to make a specially designed function as per our requirements which can then be passed to the model. Alongside, we need to ensure it is well-matched with the model. Hence, the models should have a functionality such that it takes in 4 input parameters perform the required task and returns the loss, which is our output.

To train the model we have used 135000 training images and recorded the training loss for 10 epochs and validate our model using 15000 images. It was observed that validation loss was not decreasing further after the 10 epochs. We have tried various optimizers and we found that Adam worked the best in our case. We have recorded both the training loss as well as validation loss in a table and have drawn graphs to visualize how on each iteration our training loss was strictly decreasing and the validation loss after 10 epochs did not decrease significantly. As we can see that the difference between the training and validation loss is not high so we can conclude that our model is not overfitting the training set as it is performing well on validation set also.

E. Training:

Our architecture model is trained with the SGD, the "Stochastic Gradient Descent". For the calculation of the gradients, the algorithm that our model uses are the back-propagation algorithm. In case of the RNN model, for the recurrent layers, the BPTT is used, "Backpropagation through Time". Model trained on Synth90K dataset.

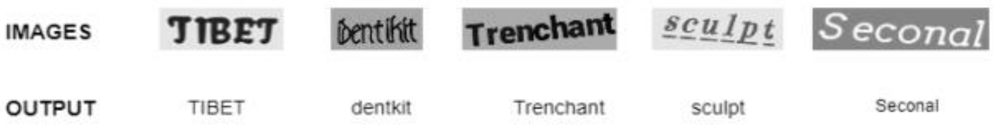
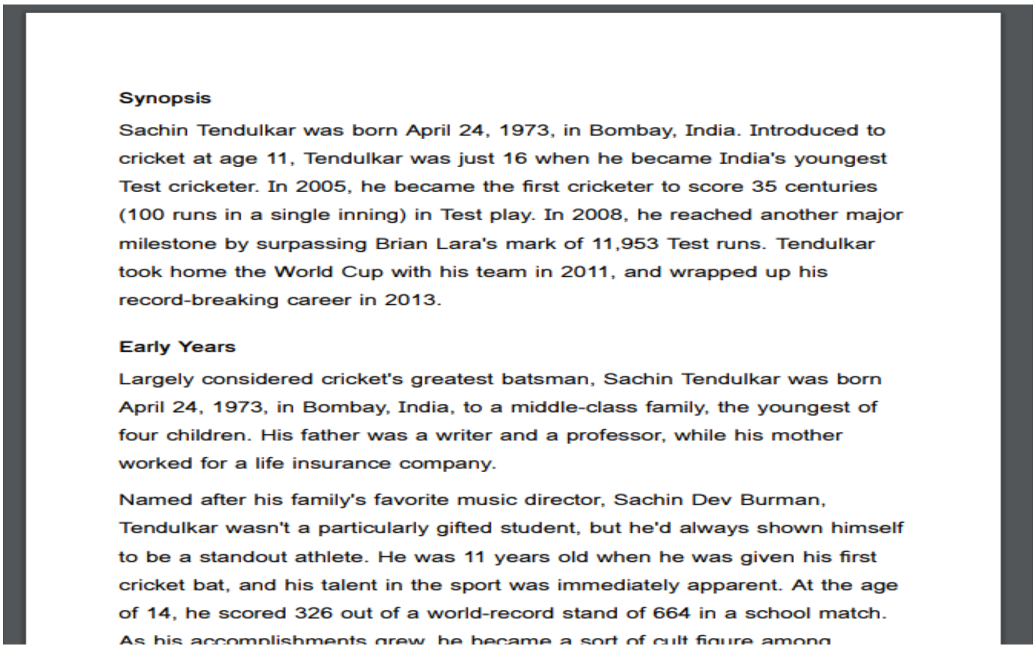


Figure 6: Output Text

F. Redaction

We get the textual data from the the OCR phase which is then redacted with the help of SpaCy, a open source library for Natural Language Processing. We have use named entity recognition, Parts of speech and dependency parsing which gives us the sensitive information like nouns, cardinal values, verbs and adjectives. We are replacing these words with placeholder and finally the output is redacted

Results



Synopsis

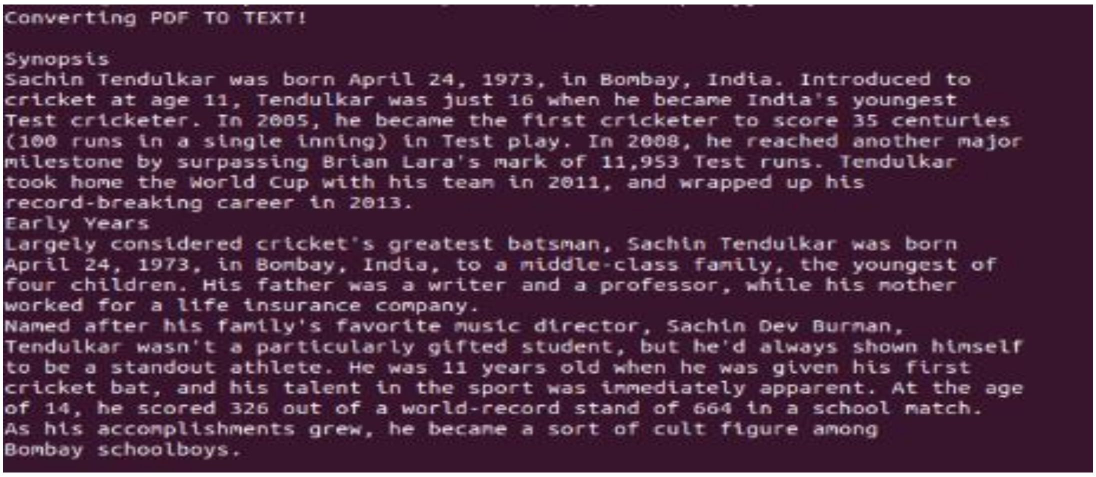
Sachin Tendulkar was born April 24, 1973, in Bombay, India. Introduced to cricket at age 11, Tendulkar was just 16 when he became India's youngest Test cricketer. In 2005, he became the first cricketer to score 35 centuries (100 runs in a single inning) in Test play. In 2008, he reached another major milestone by surpassing Brian Lara's mark of 11,953 Test runs. Tendulkar took home the World Cup with his team in 2011, and wrapped up his record-breaking career in 2013.

Early Years

Largely considered cricket's greatest batsman, Sachin Tendulkar was born April 24, 1973, in Bombay, India, to a middle-class family, the youngest of four children. His father was a writer and a professor, while his mother worked for a life insurance company.

Named after his family's favorite music director, Sachin Dev Burman, Tendulkar wasn't a particularly gifted student, but he'd always shown himself to be a standout athlete. He was 11 years old when he was given his first cricket bat, and his talent in the sport was immediately apparent. At the age of 14, he scored 326 out of a world-record stand of 664 in a school match. As his accomplishments grew, he became a sort of cult figure among

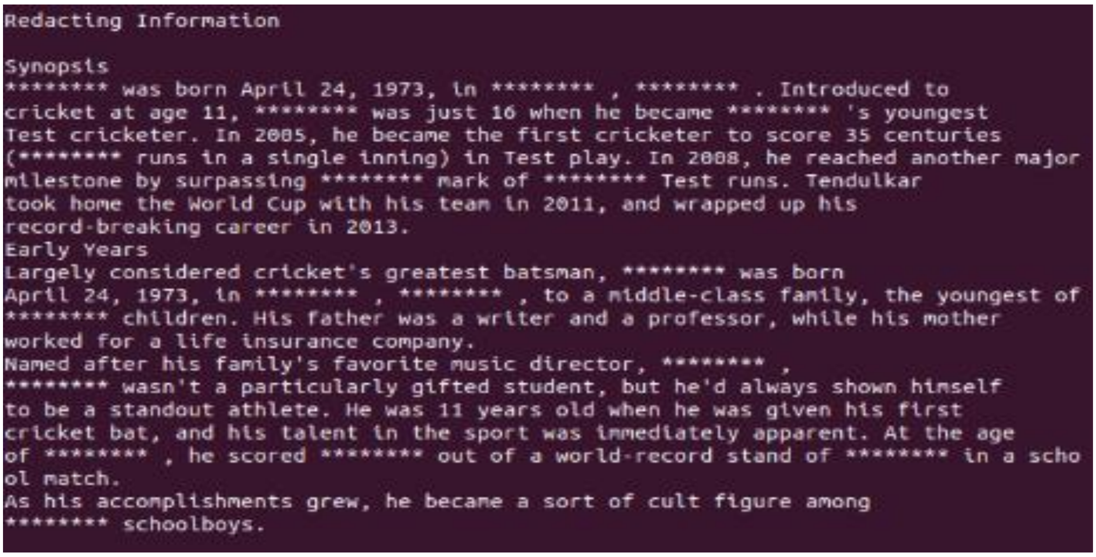
Figure 7: Input PDF file



```
Converting PDF TO TEXT!

Synopsis
Sachin Tendulkar was born April 24, 1973, in Bombay, India. Introduced to
cricket at age 11, Tendulkar was just 16 when he became India's youngest
Test cricketer. In 2005, he became the first cricketer to score 35 centuries
(100 runs in a single inning) in Test play. In 2008, he reached another major
milestone by surpassing Brian Lara's mark of 11,953 Test runs. Tendulkar
took home the World Cup with his team in 2011, and wrapped up his
record-breaking career in 2013.
Early Years
Largely considered cricket's greatest batsman, Sachin Tendulkar was born
April 24, 1973, in Bombay, India, to a middle-class family, the youngest of
four children. His father was a writer and a professor, while his mother
worked for a life insurance company.
Named after his family's favorite music director, Sachin Dev Burman,
Tendulkar wasn't a particularly gifted student, but he'd always shown himself
to be a standout athlete. He was 11 years old when he was given his first
cricket bat, and his talent in the sport was immediately apparent. At the age
of 14, he scored 326 out of a world-record stand of 664 in a school match.
As his accomplishments grew, he became a sort of cult figure among
Bombay schoolboys.
```

Figure 8: Extraction of text from pdf using the OCR.



```
Redacting Information

Synopsis
******** was born April 24, 1973, in ******** , ******** . Introduced to
cricket at age 11, ******** was just 16 when he became ******** 's youngest
Test cricketer. In 2005, he became the first cricketer to score 35 centuries
(******** runs in a single inning) in Test play. In 2008, he reached another major
milestone by surpassing ******** mark of ******** Test runs. Tendulkar
took home the World Cup with his team in 2011, and wrapped up his
record-breaking career in 2013.
Early Years
Largely considered cricket's greatest batsman, ******** was born
April 24, 1973, in ******** , ******** , to a middle-class family, the youngest of
******** children. His father was a writer and a professor, while his mother
worked for a life insurance company.
Named after his family's favorite music director, ******** ,
******** wasn't a particularly gifted student, but he'd always shown himself
to be a standout athlete. He was 11 years old when he was given his first
cricket bat, and his talent in the sport was immediately apparent. At the age
of ******** , he scored ******** out of a world-record stand of ******** in a scho
ol match.
As his accomplishments grew, he became a sort of cult figure among
******** schoolboys.
```

Figure 9: The Final Redacted Output

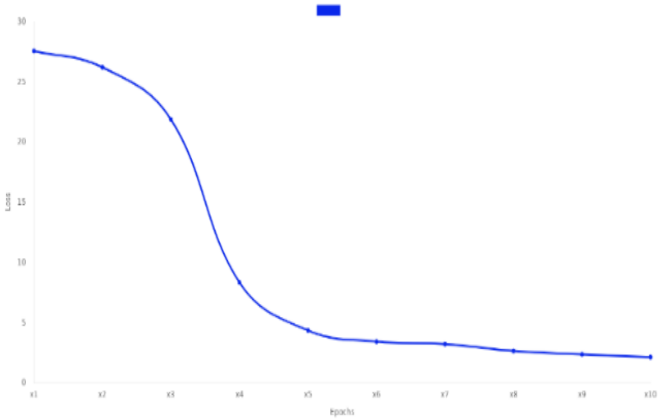


Figure 10: Training Loss

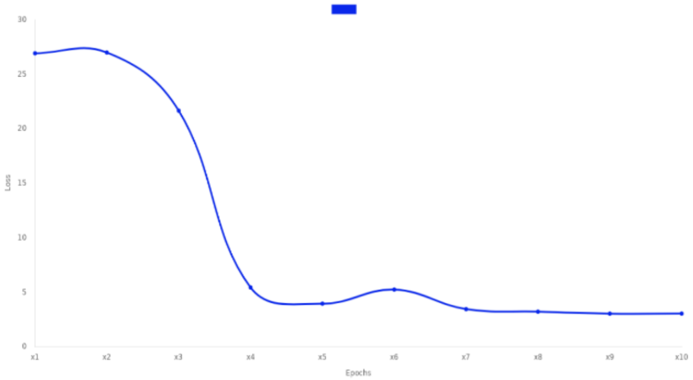


Figure 11: Validation Loss

To train the model we have used 135000 training images and recorded the training loss for 10 epochs and validate our model using 15000 images. It was observed that validation loss was not decreasing further after the 10 epochs. We have tried various optimizers and we found that Adam worked the best in our case.We have recorded both the training loss as well as validation loss in a table and have drawn graphs to visualize how on each iteration our training loss was strictly decreasing and the validation loss after 10 epochs did not decrease significantly. As we can see that the difference between the training and validation loss is not high so we can conclude that our model is not overfitting the training set as it is performing well on validation set also.

Number of Epochs	Training Loss	Validation loss
1	27.5512	26.9004
2	26.1934	26.9729
3	21.8621	21.6336
4	8.3180	5.4096
5	4.3365	3.9244
6	3.4060	5.2163
7	3.2047	3.4186
8	2.6313	3.1864
9	2.3521	3.0067
10	2.1227	3.0125

Table1: Observation

5. CONCLUSION

The use of redaction is very high in organizations of all types: government and private. All types of documents: images, and pdfs can be redacted; that is sensitive words from the document are removed. The purpose of removal of sensitive information varies on organization to organizations. Optical Character Recognition involves reading the documents that need to be redacted, it extracts the information from these images. OCR acts as the first step for the Redaction System. The output generated from the OCR is later fed into the NLP Pipeline.

References

[1] Satti, D.A., 2013, Offline Urdu Nastaliq OCR for Printed Text using Analytical Approach. MS thesis report Quaid-i-Azam University: Islamabad, Pakistan. p. 141.

[2] Mahmoud, S.A., & Al-Badr, B., 1995, Survey and bibliography of Arabic optical text recognition. Signal processing, 41(1), 49-77.

[3] Chad Cumby AND Rayid Ghani: A Machine Learning Based System for Semi-Automatically Redacting Documents in 2014.

[4] Chakaravarthy, V.T., Gupta, H., Roy, P., Mohania, M.: Efficient techniques for document sanitization. In: Proceedings of the ACM Conference on Information and Knowledge Management(2008).

[5] Liang, Zhenkai. (2018). Automated identification of sensitive data from implicit user specification

[6] Bhavani, S., & Thanushkodi, K, 2010, A Survey On Coding Algorithms In Medical Image Compression. International Journal on Computer Science and Engineering, 2(5), 1429-1434.

[7] Bhammar, M.B., & Mehta, K.A, 2012, Survey of
various image compression techniques. International Journal on Darshan Institute of Engineering Research & Emerging Technologies, 1(1), 85-90.

[8] Orekondy, Tribhuvanesh & Fritz, Mario & Schiele, Bernt. (2017). Connecting Pixels to Privacy and Utility: Automatic Redaction of Private Information in Images.

[9] Marcin Namysl and Iuliu Konya, “Efficient, Lexicon-Free OCR using Deep Learning” 5 Jun,
2019

[10] Shi, B., Bai, X. and Yao, C. (2015). An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition.

[11] Orekondy, Tribhuvanesh & Fritz, Mario & Schiele, Bernt. (2017). Connecting Pixels to Privacy and Utility: Automatic Redaction of Private Information in Images.

[12] Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A Search Space Odyssey Klaus Greff, (IEEE
TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 28, NO. 10, OCTOBER 2017).

[13] Sabir, Ekraam & Rawls, Stephen & Natarajan, Prem. (2017). Implicit Language Model in LSTM for OCR. 27-31.

[14] Sánchez, David & Batet, Montserrat & Viejo, Alexandre. (2012). Detecting Sensitive Information from Textual Documents: An Information-Theoretic Approach.

[15] Cumby, C., Ghan, R.: A machine learning based system for semi-automatically redacting documents. In: Proceedings of the 23rd Innovative Applications of Artificial Intelligence Conference, pp. 1628–1635 (2011)