



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 4, April 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

LSTM based OCR

Dr K Shailaja, R Joshua Rahul, B Jahnavi, S Sai Yashwanth, G V Charan

Assistant Professor, Dept. of CSE, Anurag University, Hyderabad, India

UG Student, Dept. of CSE, Anurag University, Hyderabad, India

UG Student, Dept. of CSE, Anurag University, Hyderabad, India

UG Student, Dept. of CSE, Anurag University, Hyderabad, India

UG Student, Dept. of CSE, Anurag University, Hyderabad, India

ABSTRACT: Optical Character Recognition (OCR) is a science that enables the translation of various types of documents or images into analyzable, editable, and searchable data. Researchers have used artificial intelligence and machine learning tools to automatically analyze handwritten and printed documents, converting them into electronic formats.

OCR deals with digitizing and converting handwritten text or documents into digitized form. The main purpose of this conversion is efficient storage, access, preservation, and transfer of knowledge from these documents for the future. In summary, OCR plays a crucial role in transforming handwritten and printed content into machine-readable formats, making it easier to manage and retrieve information.

I. INTRODUCTION

OCR stands for Optical Character Recognition. It's the process that magically transforms a text image (like a scanned document or a photo) into a machine-readable text format that computers can understand. Imagine you scan an invoice or a receipt. Your computer saves it as an image file, but the phrases within that image file cannot be edited, searched, or counted using a regular text editor. OCR takes that image and performs character recognition, converting it into a text file where the contents are saved as editable text data.

Ray Kurzweil invented OCR in 1974 to read text printed in almost any font. Over time, OCR evolved, benefiting from advances in computer vision, artificial intelligence, and machine learning. Now, with smartphones and mobile apps, a wider audience enjoys OCR for tasks like text recognition from photos.

1.1. Applications:

1. Mobile Banking Apps: OCR captures and retrieves data from checks for deposit.
2. Healthcare Records: It manages patient information efficiently.
3. Business Card Scanners: Apps with OCR can scan business cards and save contacts directly.

II. LITERATURE SURVEY

• Optical Character Recognition (OCR) for Telugu:

DATABASE, ALGORITHM AND APPLICATION: Konkimalla Chandra Prakash, Y. M. Srikar, Gayam Trishal, Souraj Mandal, Sumohana S. Channappayya

Telugu is a Dravidian language spoken by more than 80 million people worldwide. The optical character recognition (OCR) of the Telugu script has wide ranging applications including education, health-care, administration etc. The beautiful Telugu script however is very different from Germanic scripts like English and German. This makes the use of transfer learning of Germanic OCR solutions to Telugu a non-trivial task. To address the challenge of OCR for Telugu, we make three contributions in this work: (i) a database of Telugu characters, (ii) a deep learning based OCR algorithm, and (iii) a client server solution for the online deployment of the algorithm. The performance of an OCR system depends hugely on the performance of its classifier. Previous works [14] on Telugu OCR have done the character level segmentation based on histograms along the x and y directions. Assuming that the histogram method for segmentation would work perfectly, they have used SVM based classifiers for character classification. However, we have observed that in real scenarios, the histogram method fails to properly segment out the vattu and the main

character together. It also fails when the characters are rotated or if they share a common region when projected on x-axis or y-axis. Inspired by the success of deep neural networks for feature learning, we have explored CNNs to classify the characters and proposed a new architecture for the same. A CNN is a type of feed-forward neural network or a sequence of multiple layers which is inspired by biological processes. It eliminates the dependency on hand-crafted features and directly learns useful features from the training data itself. It is a combination of both a feature extractor and a classifier and mainly consists of convolutional (weight-sharing), pooling and fully connected layers.

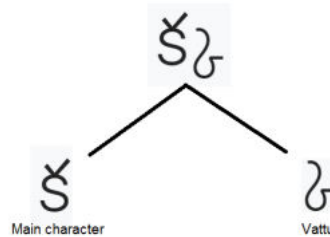


Fig 1: 3: Main character and *vattu*.

Pre-processing consists of skew correction in which tilt in the image is adjusted and binarization after which the image is binarized and segmented into individual words. This is followed by character level segmentation and classification. Our segmentation algorithm assumes that there is no skew in the image. This makes skew correction a matter of utmost importance. We have used a straight line Hough transform based technique for correcting skew that can detect and correct skew upto 90 degrees. We used a modified version of Otsu’s thresholding [15] for our binarization. We used a morphological closing algorithm for noise removal. We then computed the logical OR between the denoised image and that of the Otsu’s thresholding result and applied mode based threshold on it.

III. METHODOLOGY

- Preprocess the input image by applying binarization, noise removal, skew correction, and resizing techniques to enhance the quality and uniformity of the image.
- Segment the image into lines, words, and characters using a combination of projection profiles, connected components, and contour analysis. Alternatively, use a segmentation-free approach that directly extracts features from the whole image or patches of the image.
- Extract features from the segmented or whole image using convolutional neural networks (CNNs), which can learn robust and invariant representations of the characters. Use max-pooling and dropout layers to reduce the dimensionality and overfitting of the features.
- Feed the sequence of features extracted by the CNNs to a recurrent neural network (RNN) with long short-term memory (LSTM) cells, which can learn the temporal dependencies and context information of the characters. Use a bidirectional RNN to capture both forward and backward information from the sequence.
- Use a connectionist temporal classification (CTC) loss function to train the RNN-LSTM network, which can handle variable-length inputs and outputs without requiring explicit alignment. CTC can also output a blank symbol to indicate no character or a repeated character in the sequence.
-

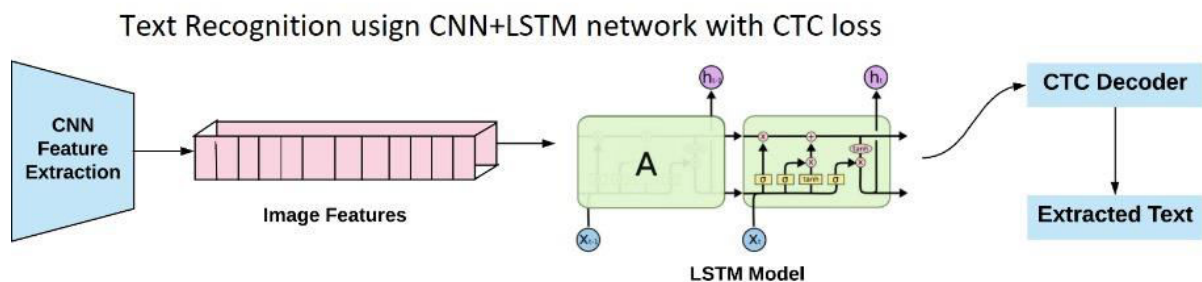


Fig 2 : Convolutional Recurrent Neural Network

3.1 CNN Classifier

A convolutional neural network (CNN) classifier is a type of machine learning model that is used to classify images and videos. CNNs are inspired by the structure of the human visual cortex, which is the part of the brain that is responsible for processing visual information. CNNs are made up of a series of layers, each of which performs a specific task. The first layer of a CNN is typically a convolutional layer, which extracts low-level features from the input image. These features can include things like edges, corners, and textures. The next layer of a CNN is typically a pooling layer, which reduces the size of the feature maps from the previous layer. This is done by merging neighbouring pixels together. Pooling layers help to reduce the number of parameters in the network and make it more robust to noise. The convolutional and pooling layers are typically stacked on top of each other multiple times, with each layer extracting higher-level features from the previous layer. The final layer of a CNN is typically a fully connected layer, which classifies the input image into one of several categories.

CNN classifiers are trained on large datasets of labelled images. During training, the network learns to associate the features that it extracts from the images with the corresponding labels. Once the network is trained, it can be used to classify new images by extracting the features from the images and then using the fully connected layer to predict the label.

CNN classifiers are used in a wide range of applications, including:

- Image classification: Classifying images into different categories, such as cats, dogs, and cars.
- Object detection: Detecting objects in images and videos, such as people, faces, and traffic signs.
- Scene segmentation: Segmenting images into different regions, such as the sky, the ground, and the objects in the foreground.
- Medical image analysis: Detecting and diagnosing diseases in medical images, such as X-rays and MRI scans.

We employ the image classifier Convolutional Neural Network (CNN) to extract features from images. CNN is a powerful and efficient machine learning algorithm known for its speed and accuracy. It's particularly well-suited for handling large datasets and complex feature spaces.

3.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network that can be used in Optical Character Recognition (OCR) to recognize patterns. LSTM is an optical character recognition algorithm that can identify characters from a captured number plate image.

LSTM predicts based on the activation of the memory cell in the previous timestep. LSTM based OCR has low error rates even without language modelling. A Convolutional Recurrent Neural Network (CRNN) model uses a convolutional neural network (CNN) to extract visual features, which are then fed to an LSTM network. The LSTM's output is then mapped to character label space with a Dense layer. The LSTM layers learn the language model of the text.

3.3. Data Sources

We have collected the required datasets from IEEE Dataport, which contains 11,055 images of Telugu characters (achulu, guninthalu, hallulu, otthulu).

3.4. Data Volume and Availability

Our dataset is well classified and pre-processed to maintain the same resolution and quality throughout the dataset. We have also drastically reduced the image size which enabled us to train our model on a large test dataset containing 11000+ images.

3.5. Accuracy and Generalization

Our approach benefits from the structured nature of text data, which can be analysed with a higher degree of accuracy and consistency. It is better suited for recognition of various characters of the language.

3.6. Speed and Efficiency

Our approach is computationally efficient and faster as we have used a CRNN network (Convolutional-Recurrent Neural Network).



3.7. Data Collection

The data consists of 50x50 pixel grayscale images of Telugu handwritten characters. The task is to categorize every character into separate labelled folders with each character containing 600+ train images.

3.8. Libraries Used

3.8.1 OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source software library designed for computer vision and machine learning applications. Originally developed by Intel, OpenCV is now maintained by a community of developers under the OpenCV Foundation. It provides a common infrastructure for various computer vision tasks and accelerates the use of machine perception in commercial products. OpenCV boasts over 2500 optimized algorithms that cover both classic and state-of-the-art computer vision and machine learning techniques.

3.8.2 Numpy

NumPy (pronounced /'nʌmpaɪ/ NUM-py) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The predecessor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

3.8.3 Tkinter

Tkinter is a Python library that provides a Python binding to the Tk GUI toolkit, making it the standard Python interface to the Tk GUI toolkit. It's open-source and comes included with standard Linux, Microsoft Windows, and macOS installs of Python. Tkinter is known for its simplicity and graphical user interface (GUI).

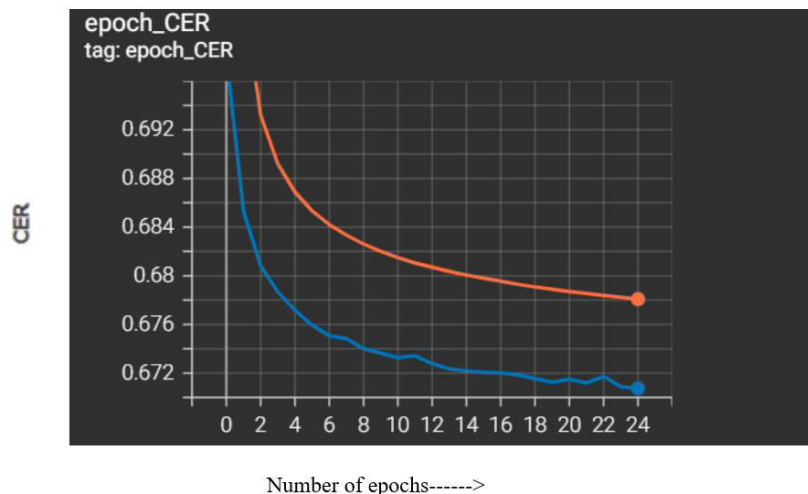
Tkinter calls are translated into Tcl commands, which are fed to the embedded Tcl interpreter, thus making it possible to mix Python and Tcl in a single application.

3.8.4 Pytesseract

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and “read” the text embedded in images. Python-tesseract is a wrapper for Google’s Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.

IV. RESULTS

4.1 Train Loss and Validation Loss





CER – Character Error Rate
Red graph – Training set
Blue graph – Validation set

4.2 Accuracies:

Average CER	7%
-------------	----

V. CONCLUSION

In conclusion, our project has successfully demonstrated the feasibility of predicting characters of Telugu language from images captured from a camera source. The CRNN model emerged as the most effective in terms of accuracy and F1-score. This suggests that our approach offers a robust solution for OCR.

REFERENCES

1. Optical Character Recognition for Handwritten Telugu Text by B Revathi · 2023
2. Telugu Optical Character Recognition Using Deep Learning by G Suresh · 2022
3. Offline Handwritten Basic Telugu Optical Character Recognition by B Kalpana · 2023
4. Optical Character Recognition (OCR) for Telugu by KC Prakash · 2018
5. Implicit Language Model in LSTM for OCR by Ekraam Sabir . 2018
6. A Convolutional Recurrent Neural-Network-Based Machine
7. Learning for Scene Text Recognition Application by Yiyi Liu, Yuxin Wang, and Hongjian Shi . 2023
8. Rakesh Chandra Balabantaray, Surajit Mohanty, Marimuthu Karuppiah, Debabrata Samanta (2022). Approach for Preprocessing in Offline Optical Character Recognition (OCR).
9. Liu, Y., & Yao, J. (2018). Character-level attention for text recognition in
10. scene images. 2018 13th International Conference on Document Analysis and
11. Recognition (ICDAR), 1-8.
12. Wang, Z., Li, W., & Wang, C. (2019). EAST: An efficient and accurate scene
13. text detector. 2019 14th IAPR International Conference on Machine Learning
14. and Applications (ICMLA), 1155-1160.
15. Jiao, Y., He, S., He, Y., & Yang, L. (2020). An end-to-end approach for scene
16. text recognition with parallel attention mechanisms. Pattern Recognition, 101,
17. 107190.
18. Luo, C., Wang, Z., Li, W., & Qiao, Y. (2020). CTPN: Cascaded proposal
19. network for robust scene text detection. IEEE Transactions on Image
20. Processing, 29(12), 6376-6388.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details