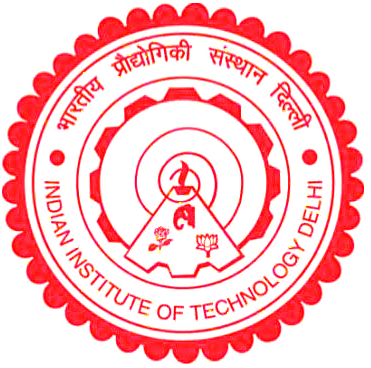
Indian Institute of Technology, Delhi

**Summer Undergraduate Research Award (SURA)**

# Prediction of viscosity of Non-Newtonian fluids at different shear rates and temperature using artificial neural network models.



Applicants:

**Lakshya Kumar Abhiraj Kumar Koslia**

(2022TT12183) (2022TT12187)

Textile Texhnology Textile Texhnology

CGPA : 8.18 CGPA : 8.12

Contact : 8527850957

Contact : 9871187200

Facilitator: Advisor:

**Prof. Chetan Arora Prof. M. Balakrishnan**

Associate Professor, Professor,

Dept. of Textile Technology & Eng. Dept. of Textile Technology & Eng.

IIT Delhi IIT Delhi

chetan@cse.iitd.ac.in mbala@cse.iitd.ac.in

**Prof. Prem K Kalra**

Head of Department,

Dept. of Computer Science & Eng.

IIT Delhi

pkalra@cse.iitd.ac.in

## 1 Introduction

Text has been one of humanity’s most influential invention, being used ever since to effectively record, communicate and inherit culture. The rich and precise semantic information carried by text has proved instrumental in recent Computer Vision based applications The problem of text recognition has been studied extensively for quite some time now due to its importance and challenges. Many vision - based applications, such as image search, intelligent inspection, industrial automation, robot navigation and instant translation rely on text recognition techniques. Due to the availability of higher computing power to the end users, text recognition can be done on-the-fly today.

The problem of text recognition can be categorized broadly into scanned document OCR (Optical Character Recognition) and STR (Scene Text Recognition) i.e. camera based text recognition. Document OCR deals with regular scanned documents, which have consistent fonts, sizes, colors, spacings etc.

On the other hand, Scene Text Recognition deals with text recognition in natural scene images. This presents the challenge of dealing with varied fonts, backgrounds, low resolution and perspective distortion.

## 2 Motivation

Recognizing text in natural scenes has attracted great interest from academia and industry in recent years because of its importance and challenges. Previous work focused on hand-crafted features. With the advent of deep learning and its success in several fields in recent years, new approaches have been developed for the challenging problem of Scene Text Recognition.

Unfortunately however, much of this research has been focused on English text. Current state of the art models and benchmark datasets largely focus on English. Adaptation of these models to solve the unique challenges posed by the Hindi script has not been well studied.

More than 500 million people speak Hindi in India, which makes it all the more important for the text recognition systems to work well on Hindi text. The applications for such a robust Hindi based Scene Text Recognition model are widespread. The visually impaired community, regional industrial automation and language translation all will benefit from advances towards a robust end to end system for Hindi Scene Text Recognition. Recent work on the signboard dataset curated by Mobility Assistant For Visually Impaired (MAVI), IIT Delhi has shown great scope for improvement of Hindi STR which could aid in empowering the Hindi speaking visually impaired community.

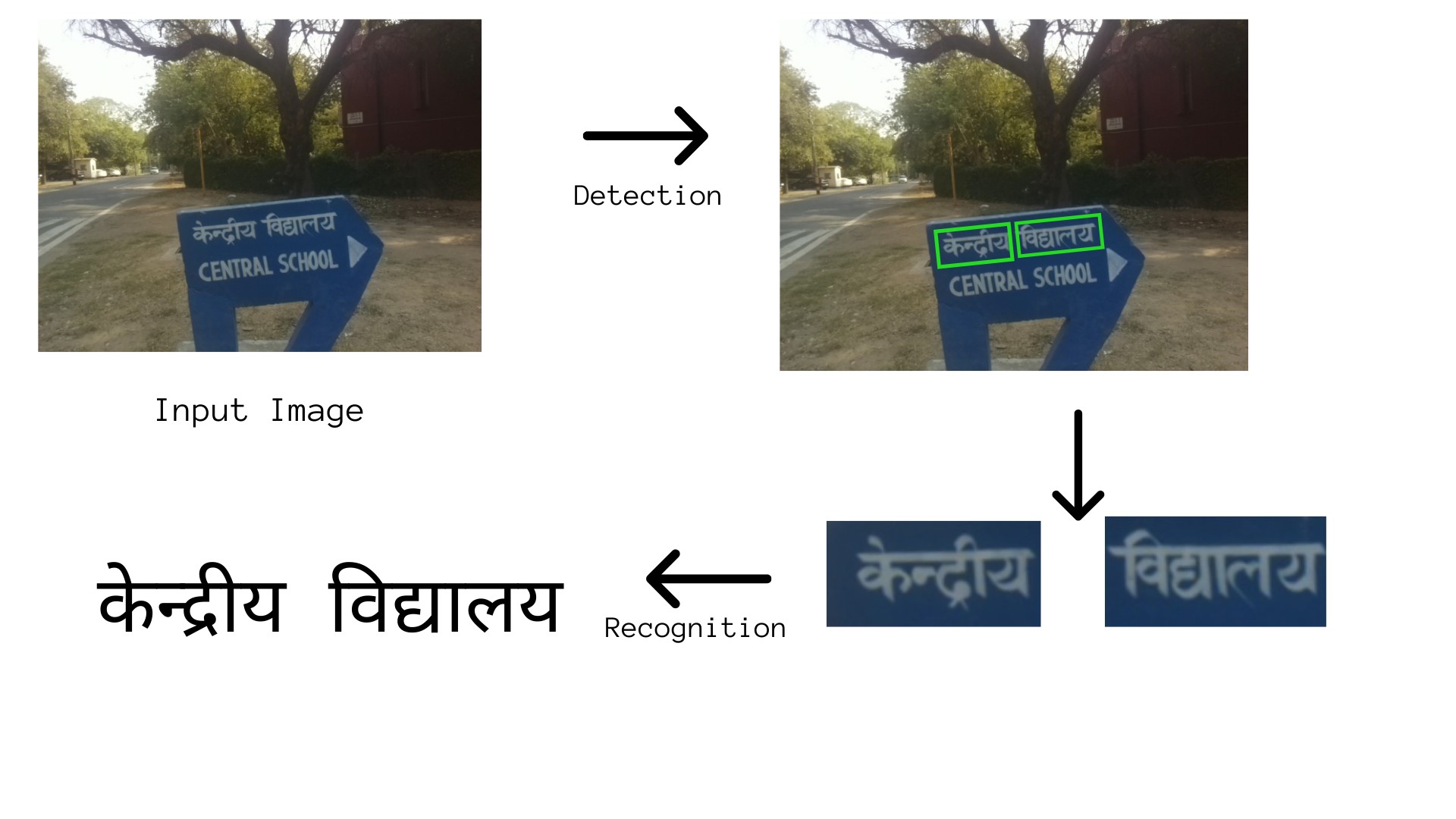


Figure 1: Illustration of Hindi Scene Text Recognition

## 3 Related Work

The problem of Hindi Scene Text Detection and Recognition has not received much attention until only recently. Mathew et. al [2] published their work on Scene Text Recognition in three Indic scripts - Hindi (Devanagari), Malayalam and Telugu. They compared the performance of RNN-OCR and CNN-RNN on the IIIT - ILST dataset, and concluded CNN - RNN performs much better. The accuracy for Hindi text is however, still very low as compared to English. The Devanagari script poses several challenges which makes the task at hand difficult:

1. Characters have a lot of curved information and stroke features which makes it difficult to locate and recognize the text in scene images. Hindi scene images also often have irregular font.
2. Matras pose a unique challenge to text recognition. For eg. the syllable *ke* in English has separate characters which are easier to recognize whereas in Hindi the same is written as one composite character and is expected to be recognized as two different Unicode characters ( the *akshar* and the *matra*).
3. Classification at the composite character level is very challenging as the number of valid classes exceeds 5,000. [7]

There has also been work towards efficiently decomposing Hindi text images into smaller components[7] but an end-to-end solution has not been built. We aim to build on this research towards and end-to-end detection and recognition system.

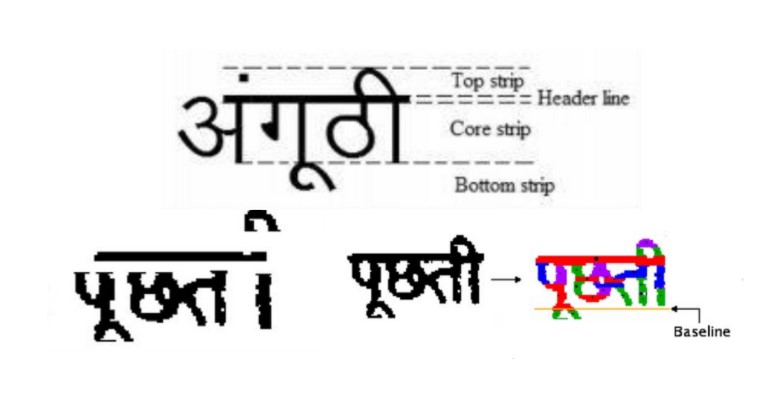


Figure 2: Challenges of the Devnagri script

## 4 Objectives

The main objective of our project is to build a robust end-to-end system for Hindi Scene Text Detection and Recognition with focus on bridging the gap between the accuracy of English and Hindi script for the MAVI dataset. • Implement current state of the art STR models and analyze their performance and sources of error on the MAVI dataset.

* Perform exhaustive comparison of state of the art models across all modules of the pipeline, identifying their shortcomings and merits.
* Implement an image pre-processing module to address challenges of the MAVI dataset and a novel script identification module to differentiate between Hindi and English text, which can be added to other scene text recognition systems and also extended to other Indic languages.
* Build a large Synthetic Hindi scene text dataset, which can also be used for future research in scene text recognition.
* Identify and implement suitable changes to current models to address their shortcomings on the MAVI dataset.
* Build an end-to-end system with the best performing algorithms, tailored to address their shortcomings on the MAVI dataset.

## 5 Concepts

We briefly describe the main components of an end-to-end system for Scene Text Detection and Recognition.

1. **Text Detection** involves detecting text regions in the complex background and label them with bounding boxes.

### (a) Text Localization

This step involves locating the presence of text instances in a scene image and grouping them together into candidate regions with as little background as possible.

### (b) Text Verification

This step aims at verifying whether the previously selected regions are text or not, because text localization often introduces false positives.

### 2. Text Recognition

The text instances detected in the previous step are fed to the recognition module which translates this cropped text image into a target string sequence.

Image pre-processing can also be involved to improve the resolution of image, remove irregularities in font, remove the background, and fix perspective distortion or blurriness in images. These additional modules can help improve the accuracy of the recognition module.

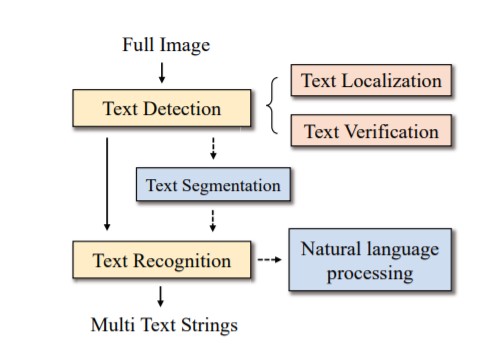


Figure 3: Illustration of an end-to-end system for scene text recognition [5]

## 6 Approach

Traditional approaches to Scene Text Recognition generally involved heuristic based algorithms or classifiers trained on hand-crafted features. These approaches were tedious, entailing demanding and repetitive pre-processing and post-processing. We propose modern Deep Learning based approaches since Deep Learning does not suffer from the problem of manually designing features or heuristics and has shown great promise in several Computer Vision tasks.

### 6.1 Text Detection

Text Detection can be regarded as a special case of Object Detection. However it has its own unique challenges. Text Detection techniques can be broadly categorized into top down approaches and bottom up approaches.

### 1. Bottom Up Approach

The bottom up approach focuses on first detecting the fundamental elements of text, and then aggregating them to build bounding boxes. Such method us used by the Seglink model, and its variant Seglink++. PSENet and PAN define each pixel of the image as a separate segment, and then aggregate them to build the bounding boxes.

### 2. Top Down Approach

The top down approaches directly generate word level or line level detection results. This is made possible by treating text as a special type of object. Top down methods can be further categorized into one stage detectors and two stage detectors.

1. **One Stage Detectors** directly performs box regression on the feature map to produce the bounding boxes.
2. **Two Stage Detectors** first uses a Region Proposal Network using the feature map to produce text proposals, and then regresses the box parameters to produce refined bounding boxes

We plan on using a top down approach since it avoids the complex aggregation process and dependence of accuracy on the aggregation algorithm used.

There is almost no literature comparing various text detection models on Hindi text. Therefore, we aim to implement and compare the performance of text detection models such as TextFuseNet, PMTD etc. Further, it may be possible to tailor the models for the Hindi language, which will improve the performance of the system significantly.

**6.2 Pre-Recognition Module**

### 1. Image pre-processing

Image pre-processing is an important step before the actual recognition task starts, as it helps to improve the quality of the input image by reducing irregularities of font and background, and fixes perspective distortion and blurriness. We aim to implement multiple algorithms for image pre-processing, such as: • **Text Image Super-Resolution (TextSR)**

Scene text is usually distorted by various noise interferences, such as low resolution. Text image super-resolution (TextSR) can output a plausible high-resolution image that is consistent with a given low-resolution image. This approach can help with text recognition in low-resolution images.

### • Background Removal

Text may appear in various scenes with complex backgrounds. Texture features of backgrounds can be visually similar to the text, which causes additional difficulties in recognition. Separating text content from the background can help suppress background noise and improve accuracy.

### • Rectification

This helps in normalizing the text input image, as scene images often have irregular text leading to difficulties in recognition

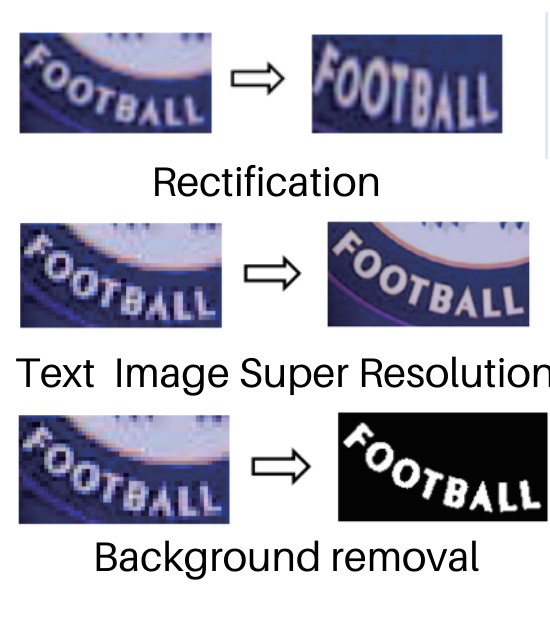


Figure 4: Image Pre-processing [5]

### 2. Script Identification

We plan on implementing a specialized script identification model to help classify the text into Hindi and English. It would be easy to extend the model to various other Indic languages. This would be an important step in the development of multilingual text recognition systems.

### 3. Text Segmentation

Due to less spacing between words in some images, the detection module may sometimes detect multiple words as a single unit. This may lead to poor accuracy for single word recognition models. We aim to introduce a specialized text segmentation stage which will address this problem by implementing word segmentation algorithms and analyzing their performance on Hindi text.

**6.3 Text Recognition**

### 1. Feature Representation Stage

The input image is transformed to a feature space, with an aim to retain features in the original image useful for recognition and suppress useless information like font, color, size and background. Convolutional Neural Networks have become a popular choice as feature extractors. We aim to fine-tune popular CNNs such as the ResNet, DenseNet, UNet and

EfficientNet.

### 2. Sequence Modelling

Sequence Modelling can capture the contextual information within a sequence of characters for the next stage to predict each character, which is more stable and helpful than treating each symbol independently. Different sequence models such as BiLSTMs, CNNs and transformer structures can be experimented with.

### 3. Prediction

There are two major techniques used in the final stage, CTC (Connectionist Temporal Classification) and Attention. While the scope of improvement at this stage is relatively less, trying out the variants of these methods might lead to better performance.

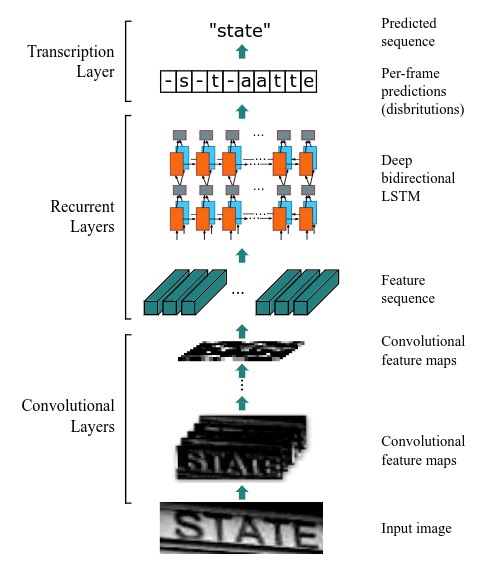


Figure 5: Illustration of model architecture for Text Recognition [6]

## 7 Datasets

We will primarily use the MAVI Signboard dataset curated by Mobility Assistant for Visually Impaired (MAVI), IIT Delhi for training and testing our model. However, it would be possible to generalize the model for other datasets such as the IIIT - ILST dataset.

### 7.1 MAVI Dataset

The MAVI Dataset contains 1493 images for 85 signboards, taken in and around the IIT Delhi campus. For each signboard, multiple images from different perspectives and in various illumination conditions have been taken to better model real - world scenarios. The images contain both English and Hindi text. In all the images, the text is in white, and on a blue background, although blurring and distortions may be present.



Figure 6: MAVI Signboard Dataset[3]

### 7.2 IIIT - ILST Dataset

IIIT - ILST Dataset contains around 1000 annotated scene text word images. The images contain text in varied fonts and backgrounds. As compared to the MAVI dataset, higher proportion of images are taken in good illumination conditions, and text is more prominent. The dataset has images of other Indic scripts as well from which we plan to use the Devnagri images.



Figure 7: IIIT-ILST Dataset[2]

### 7.3 Synthetic Data

Due to the small size of the datasets, synthetic data must be used to train the model first. Thereafter, the real world datasets may be used to finetune the model. Since there are no standard tools to generate synthetic textual images for Hindi, we plan to generate our own synthetic data. The synthetic data generated would be in line with the MAVI dataset, that is, white text on blue background.



Figure 8: Synthetic Dataset

## 8 Novelty

While there has been considerable progress in STR for English text, these has been no attempt to apply these new methods to other languages such as Hindi. We look to implement and analyze these methods for Hindi STR, and tailor them for the language. We also plan to add a script identification module which can be further extended to support other Indic languages. Improved accuracy for Hindi would further motivate research in other languages. Moreover, our work would be of direct use for developing assistive technologies for the visually impaired.

## 9 Budget, Duration and Facilities

The duration of the project will be of 2 months. We will continue to work beyond this point if required to bring the project to a meaningful conclusion.

We will require access to the following resources for the project:

1. NVIDIA v100 GPUs for training the models
2. Intel Skylake CPUs
3. Disk storage for the dataset

We will need approximately |20,000 to get high priority access to HPC resources.

## References

1. Shangbang Long, Xin He, Cong Yao. *Scene Text Detection and Recognition: The Deep Learning Era*. International Journal of Computer Vision (2021)
2. M. Mathew and M. Jain and C. V. Jawahar *, Benchmarking Scene Text Recognition in Devanagari, Telugu and Malayalam*. ICDAR MOCR Workshop 2017.
3. Rajesh Kedia and Anupam Sobti and Mukund Rungta and Sarvesh Chandoliya and Akhil Soni and Anil Kumar Meena and Chrystle Myrna Lobo and Richa Verma and M. Balakrishnan and Chetan Arora *, MAVI: Mobility assistant for visually impaired with optional use of local and cloud resources*. 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID) 2019
4. M. Mathew, A. K. Singh and C. V. Jawahar, *Multilingual OCR for Indic Scripts* 2016 12th IAPR Workshop on Document Analysis Systems
5. Xiaoxue Chen, Lianwen Jin, Yuanhi Zhu, Canjie Luo, and Tianwei Wang *Text Recognition in the Wild: A Survey* Dec 2020
6. Shi, Baoguang and Bai, Xiang and Yao, Cong *An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition* IEEE transactions on pattern analysis and machine intelligence 2016
7. Suryaprakash Kompalli and Srirangaraj Setlur and Venu Govindaraju *Devanagari OCR using a recognition driven segmentation framework and stochastic language models* International Journal on Document Analysis and Recognition (IJDAR) July 2009