# Text Recognition from Scenic Images

**CS4099D Project**

**Final Report**

*Submitted by*

|  |  |
|---|---|
| **Anoop Babu** | **(B170065CS)** |
| **Hulle Nikhil Vaijanath** | **(B170023CS)** |
| **Tarun Thaliath** | **(B170024CS)** |

*Under the Guidance of*

**Dr. Vinod Pathari**
**Associate Professor**

**Department of Computer Science and Engineering**
**National Institute of Technology Calicut**
**Calicut, Kerala, India - 673 601**

**May, 2021**

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# CERTIFICATE

*Certified that this is a bonafide report of the project work titled*

**TEXT RECOGNITION FROM SCENIC IMAGES**

*done by*

**Anoop Babu (B170065CS)**
**Hulle Nikhil Vaijanath (B170023CS)**
**Tarun Thaliath (B170024CS)**

*of Eighth Semester B. Tech, during the Winter Semester 2020-'21, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology Calicut.*

Vinod

Dr Vinod Pathari
Associate Professor
**Project Guide**

13 May 2021
**Date**

# DECLARATION

We hereby declare that the project titled, **Text Recognition from Scenic Images**, is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place : NIT Calicut
Date : 13 May 2021

Signature : Anoop
Name : Anoop Babu
Roll. No. : B170065CS

Signature : Nikhil
Name : Hulle Nikhil Vaijanath
Roll. No. : B170023CS

Signature : Tarun
Name : Tarun Thaliath
Roll. No. : B170024CS

# Abstract

According to Wikipedia, “Optical Character Recognition or Optical Character Reader(OCR) is the conversion of images of typed, handwritten, or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo, or from subtitle text superimposed on an image.” A typical OCR software operates as a 3 step process. These three steps are preprocessing, text recognition, and post-processing. The overall performance of the OCR engine depends on the efficiency and effectiveness of these three steps. This proposed model is able to recognize text from images of objects with irregular surfaces (the scope of inputs for this project is limited to medicine strips) and then match the input to a more general case and then re-route to an external website to display the search results which may give more information or details for the provided input. We extend the existing open source OCR software Tesseract's functionality to be able to identify and provide a more accurate output specifically for images with uneven characteristics.

# ACKNOWLEDGEMENT

We would like to thank our guide, Dr Vinod Pathari, Department of Computer Science and Engineering, National Institute of Technology, Calicut, for his continuous guidance and support. We would also like to thank The National Institute of Technology, Calicut, for giving us the opportunity to work on such a project.

# Contents

# List of Figures

# Chapter 1

# Introduction

Most of the OCR softwares are designed primarily for simplistic cases like document images such as those from a flatbed scanner. But when such software is used as it is on scenic images which differ in intensity, contrast and orientation, they give unsatisfying results. This is because they typically rely on techniques such as binarization,where the first stage of processing is a simple thresholding operation used to categorize an image into text and non text pixels. With advancements made in areas such a deep learning, there has been a marked improvement in the performance of these OCR softwares when it comes to scenic images. But there is still room for improvement when it comes to the processing of the input image before the recognition step. Since such scenic images are becoming more and more common due to usage of mobiles and other similar devices, these improvements become all the more important. Extraction of text from images of medicine strips is one such example where increased OCR performance will be of tremendous help.

# Chapter 2

# Literature Survey

Having discussed about the existing revelations/shortcomings of the OCR technologies in the introductions, we see that in contrast to existing popular OCR engines, methods shown in papers like [1] display computationally expensive characteristics in order to achieve optimal recognition rates, such methods are typically tried on smaller amounts of text per image and are often constrained by predefined lexicons which limits the diversity of symbols that can be recognized e.g alpha numeric characters, etc. OCR is classified into two methods i.e. segmentation-based and segmentation-free methods.

Segmentation-based OCR mainly work to divide the given image or document into smaller components like lines,words,characters, etc.., and the output is a recognition lattice containing various segmentation and recognition alternatives weighted by the classifier. The lattice is then decoded, e.g., via a greedy or beam search method .We see this method used in [2]. They make use of histogram oriented gradients trained neural networks for character classification and use a character level language model score function.

The drawbacks of using such a method is that it is highly dependent on the process of segmentation which often suffers from segmentation error and lack of context in a specific segmented window. It also highly depends on the language model that is being employed that might constrain the model as mentioned earlier. A lot of effort is usually required to tune the the model for specific application scenarios. Moreover the precise weighting of all involved hypothesis must be re-computed from scratch every time one component is updated.

Thus using segmentation free OCR solves most of the above problems as it does not need pre-segmented inputs. Most of the recently developed segmentation free methods use recurrent and convolutional neural networks. Thus in this report, what we aim to use is a segmentation free text recognition method with much lesser computational weights, and for achieving optimal recognition rates we employ methods like edge preserving MSER, edge detection methods and Stroke Width Transform[13]. By using this, we hope to improve on the robustness of the OCR which will help us to achieve appropriate results.

While MSER has been widely used for region detection problems due to its robustness against view point, scale and lighting changes as shown in [3], it appears to be sensitive to image blur. Thus, small letters cannot be detected or distinguished in case of motion or blur by applying MSER to images of limited resolution. To cope with this we incorporate complimentary properties of canny edges[4] and MSER. Here, the outlines of the regions are enhanced and optimal pruning of MSER is done in the MSER detection stage.

Reference [9] uses a slightly modified version of stroke-width transform algorithm that accepts an image as input, processes it and produces a new image, with text segments marked, as output. The stroke width algorithm works by assigning each pixel of the image with the value corresponding to the stroke width of the stroke containing that pixel. The pixels with similar stroke width value are considered to be in the same text segment upto a certain limit.

Reference [10] proposes using a combination of the methods mentioned in the earlier papers. Text detection is first done using the intersection of the outputs obtained from processing the image using mser and LoG algorithms. Laplacian of Gaussian (LoG) is an image edge detection technique. In this technique, the areas of the image having high difference in intensities are assumed to be the edges between the text and the background thereby giving us a more identifiable outline of the text. The image is first smoothened using a Gaussian filter and edges are identified using the principle of zero-crossings. This is explained in [11].

The combined output is further refined using strokes width transform and other processing techniques and is passed through the OCR for text extraction. The text obtained from this stage is then matched with the stored text in the database upto a 70 percent confidence level to give a more accurate final output.

# Chapter 3

# Problem Definition

We aim to improve the accuracy of text recognition from scenic images such as those of medicine strips, i.e. to improve on the output of raw OCR software. We provide an image (of medicine strip) as input and expect the textual information present in the image (name of the medicine) to be extracted with a high level of accuracy and forwarded to an external website that would provide more information about the product.

## 3.1 Motivation

A brief search on the internet can tell us that there is no sure shot method for getting text out of scenic images and moreover no solid method which gives satisfactory results in the case of medicine strips. Extraction of text from medicine strips is still a very niche area and not much solid work has been done on it. We believe that this sort of an application can be very easily extended to other medical products which can help large medicine conglomerates and hospitals efficiently log their products and can easily find substitutes to any medicines. These are just a few of the many applications that can be though of. What makes this application even more challenging is that the input for this model is a very uneven surface with multi-colored fonts, different font sizes, etc. Moreover since mostly any user would take a picture from an handheld device and not a professional picture capturing device, this brings added uncertainties in the form of fluctuating resolutions, angle of lights falling on to the medicine strip while taking the image, angled text on the medicine strip etc. Thus considering all these factors with respect to the image, the current plain OCRs do not give satisfactory results because they are primarily designed to detect text which are consistent in terms of fonts, color, background, etc. Thus, this current situation of available products for text recognition was our motivation to create a model which will account for all these variables and give viable results. We aim to provide the most correct string for the medicine name which can be extracted from the image.

## 3.2 Input/Output

Input: An image (of medicine strip).
Output : The user is redirected to an external website that contains more information about the product based on the detected text.

# Chapter 4

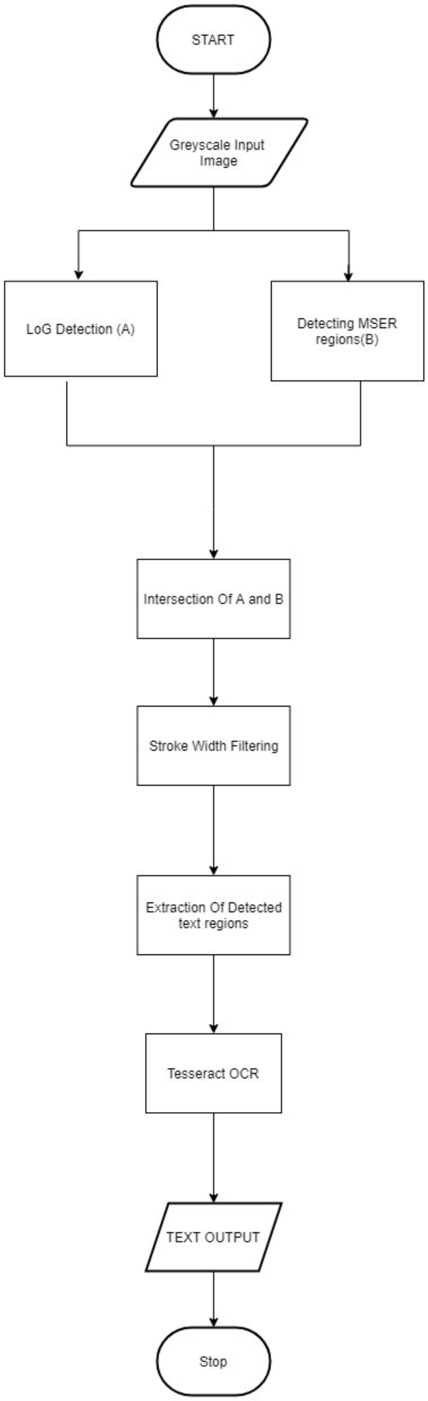
# Methodology

For a rough pictorial overview, refer the pictorial representation of the workflow in Section 4.1.1. The first step involves the selection of a medical strip for which the user wishes to detect the name. After supplying the said image to the model as an input, we then go ahead and apply various algorithms to facilitate detection and extraction of text from the image. MSERs are efficiently computed for the image followed by the edge detection for medical strips. To overcome any limitations of MSER we perform geometric checks to filter the regions. In the next step, Stroke Width Transform is applied to further filter out the non-text regions. Bounding boxes are then created around the identified text regions and segregated into various small images containing only the text identified from the above process and that is finally passed into the Tesseract OCR. Furthermore, to account for any lost characters in the identified string, we use a Fuzzy string matching/editing algorithm to complete the identified string if needed.

We implement identifying text from medicine strips via a web application. The web application allows the user to upload an image of a medicine strip, identifies text from the image, and redirects the user to a website that gives more details about the medicine. The workflow for the text identification process is given below.

## 4.1 Workflow

### 4.1.1 Workflow Model



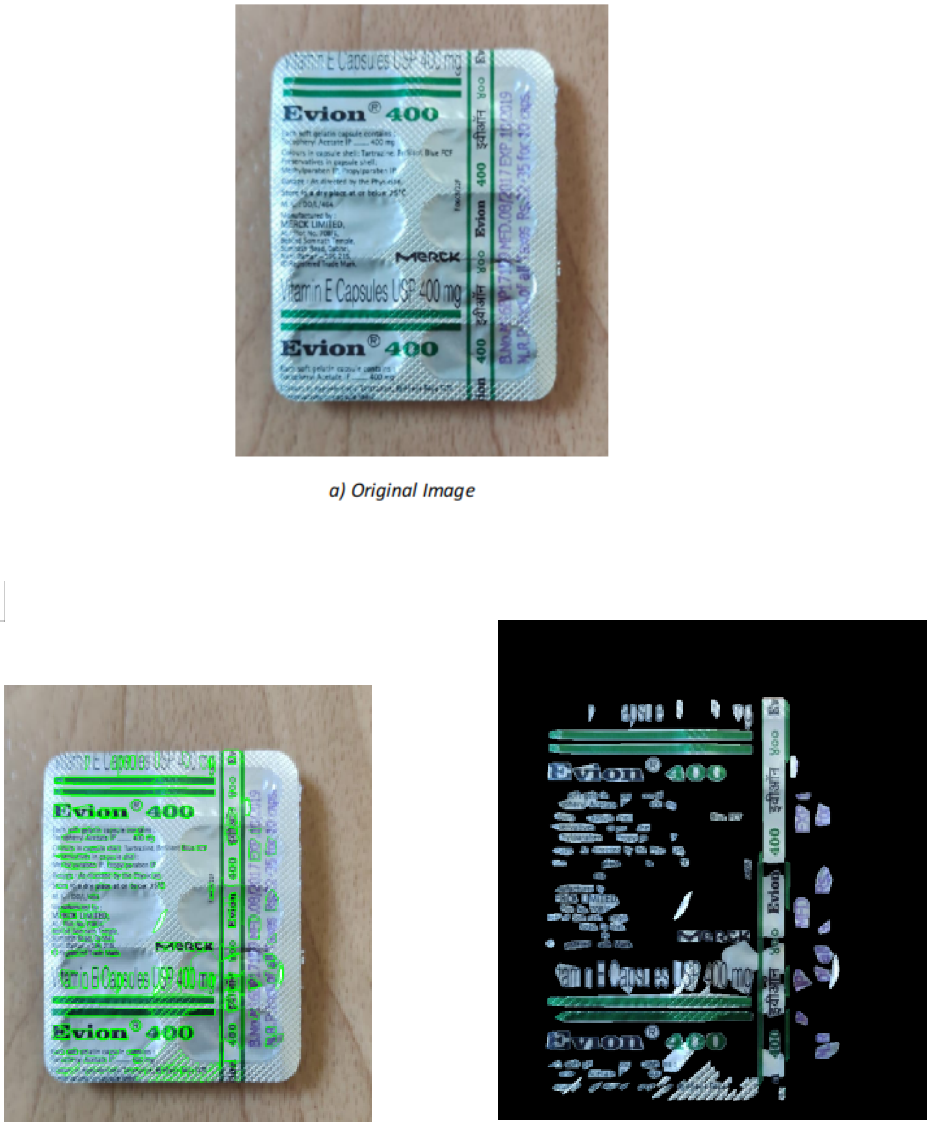
### 4.1.2 Detecting MSER Regions

Maximally Stable Extremal Regions (MSER) Algorithm is used for blob detection. The MSER algorithm returns from an image numerous co-variant regions called MSERs: each of these regions is a balanced component of some gray-level sets of images.Ultimately potential text regions are identified in the image based on the consistency of color and contrast.

The algorithm returns the regions that stay almost indistinguishable through a wide range of threshold values.The pixels preset below a certain preset threshold are considered white, and all those that are above or equal to this preset threshold are considered to be black. If shown a ordered series of thresholded images $\mathrm{I_t}$ with frame t corresponding to a threshold value t, At first, a black or dark image would appear, and then some sort of a white discoloration occurs which match the local intensity minima and will then these white spots appear to spread throughout and grow bigger in size until the whole image is white. The set of all connected components in order is the set of all extremal regions. The "EXTREMAL" word refers to the characteristic that all pixels inside a specific detected regions have either higher(bright extremal regions) or lower(dark extremal regions) intensity when compared to all the other pixels existing on its outer boundary.

Edge-preserving MSER (eMSER)[5].

Input: RGB input image and necessary parameters
Output: Extracted text with highest possible accuracy .
1 Convert the RGB image to an intensity image I.
2 Smooth I using a guided filter [6].
3 Compute the gradient amplitude map ∇I, and then normalize it to [0, 255].
4 Get a new image $I^* = I + \gamma \nabla I$(resp. $I^* = I - \gamma \nabla I$).
5 Compute the MSER algorithm on $I^*$ to retrieve dark/black regions on the bright background (vice versa i.e for bright regions on dark background).



*a) Original Image*

*b) Blob Detection*

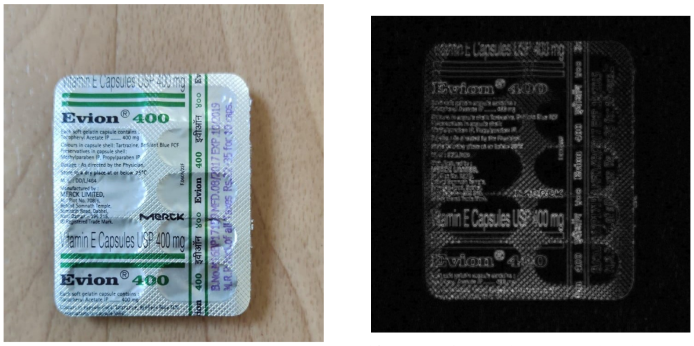
*c) Blob Detection(Blacked Out)*

Figure 4.1: *MSER*

### 4.1.3 Edge Detection

Edge Detection Algorithms use differences in intensity between text regions and background regions to sketch out the edges of the text. Canny Edge Detection and Laplacian of Gaussian are 2 edge detection methods used for this purpose.

Laplacian of Gaussian (LoG)
The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. A matrix is generated marking the intensity levels of each pixel and the second derivative is taken. When this is done the only remaining task is to locate the zeros and these are the edges.
1. Compute the mask that has to be used for Laplacian of Gaussian
2. Convolve the image along each row and call it $I_x$.
3. Convolve the image along each column and call it $I_y$.
4. Add $I_x$ and $I_y$.
5. Find zero crossings from each row and column.
6. Apply a threshold value for the slopes of the zero crossings (to identify whether its actually an edge or just a continuous progression)



a) Original image

b) Image after applying LoG edge detection algorithm

Figure 4.2: *Laplacian of Gaussian Edge Detection*

### 4.1.4 Geometric Filtering

Each MSER region detected is filtered based on the geometric properties such as area, perimeter, aspect ratio and compactness before stroke width transform is applied.

PARAMETERS USED [12]

1. AREA LIM = 2.0e$^{-4}$
2. PERIMETER LIM = 1e$^{-4}$
3. ASPECT RATIO LIM = 15.0
4. OCCUPATION INTERVAL = (0.23, 0.90)
5. COMPACTNESS INTERVAL = (3e$^{-3}$, 1e$^{-1}$)

The parameters are applied relative to the corresponding properties of the image.

### 4.1.5 Stroke Width Filtering

The Stroke Width Transform associates each pixel (x.y) to its most probable stroke width value. Potential components obtained from the edge detection methods, are filtered out based on the variation in their stroke width values. Higher variation in stroke widths within a component indicates that the component is not a text character as text characters generally have consistent stroke width.

The stroke width corresponding to a particular pixel is computed by sending rays from edge pixels in the direction of the gradient. The end point of the particular ray either coincides with an edge directly opposite to the source edge or does not meet any edge at all. All pixels traversed in the path of the terminating well-defined rays emanating from such source edges are assigned a stroke width equal to the length of the ray.

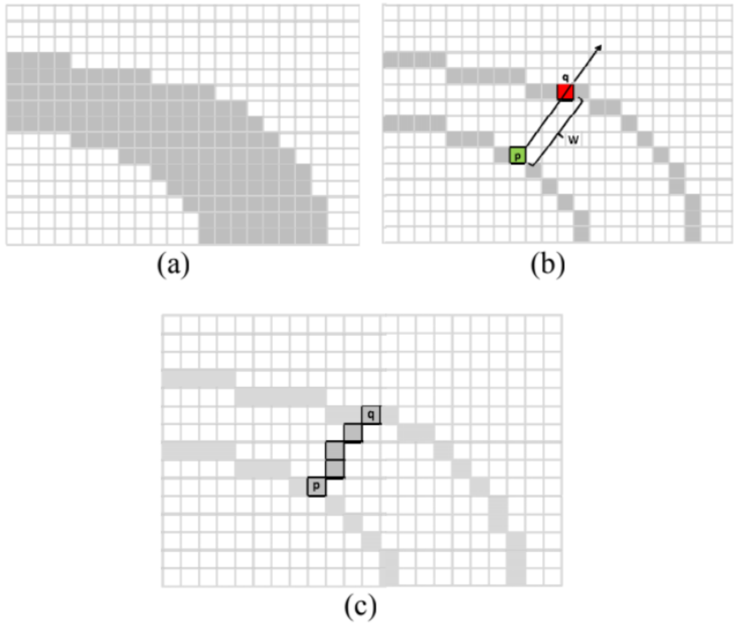


Figure 4.3: *a) A stroke. b) p is a pixel on the boundary of the stroke. A ray is drawn in the direction of the gradient of p and it meets the opposite edge at q. c) The pixels along the ray pq are assigned the minimum of its current value and that of the width of the stroke. Image obtained from [9]*

Stroke Width Transform Algorithm

```
Let set P=set of pixels identified as edge pixels
Assign all pixels in image with stroke width $\infty$
for pixel in P
    for every possible gradient $d_p$ //All 8 possible directions from pixel
        n=0
        while(true)
            q=pixel+n$d_p$ //General form of the ray considered
            if(q belongs to P)
        if(gradient(q)=(-$d_p$))
            Traverse the ray again and assign each stroke value to
            min(value,$||pixel - q||$), where value is the current stroke
            width assigned to the pixel
        break
    n=n+1
```

On calculating the stroke widths, pixels are grouped into components based on the similarity of their stroke width values upto a certain error (variance). This stroke width variance ratio is taken to be 0.15.

### 4.1.6 Extraction of Detected Text Regions

The image is binarized based on the regions remaining after filtering and is then dilated over a 3x3 kernel over 7-10 iterations. Dilation smoothens the text in the image by adding pixels around the object boundaries. Each text region is made straight by rotation and is passed to the Tesseract OCR for text recognition and the text is obtained.

## 4.2 Implementation Details

The workflow mentioned was implemented in python with the help of the following libraries:

- OpenCV: for standard image processing routines.
- numpy: for mathematical operations on arrays.
- scipy : for solving scientific and mathematical problems.
- pytesseract: For Tesseract functions.

- Fuzzywuzzy: for fuzzy string matching.
- Flask: micro web python framework for web development.

HTML and Node.js were also used for basic web development.

# Chapter 5

# Results

## 5.1 Test Cases

### 5.1.1 Test Case 1 (Strip name: Mefex)

The input image is of the medicine “Mefex” (Refer Fig 5.1). The extracted text after applying our model is shown in Fig 5.2 and finally the output as mentioned in the Problem Definition(Refer Subsection 3.2) is shown in Fig 5.3. The extracted text from the input image consists of the medicine strip name i.e “Mefex”.

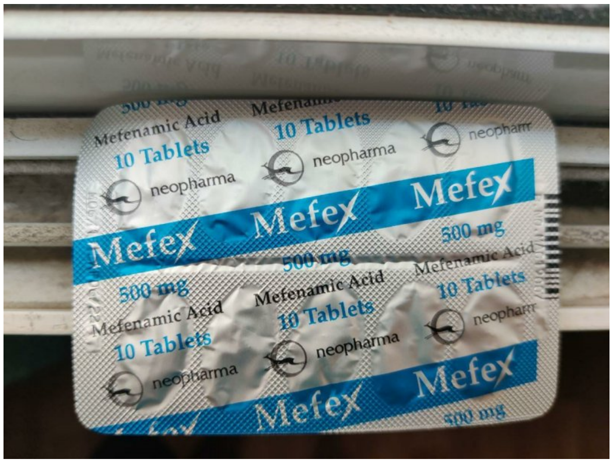
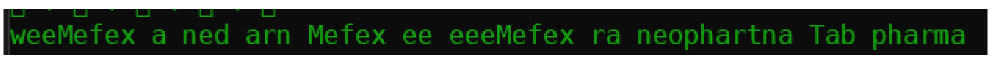
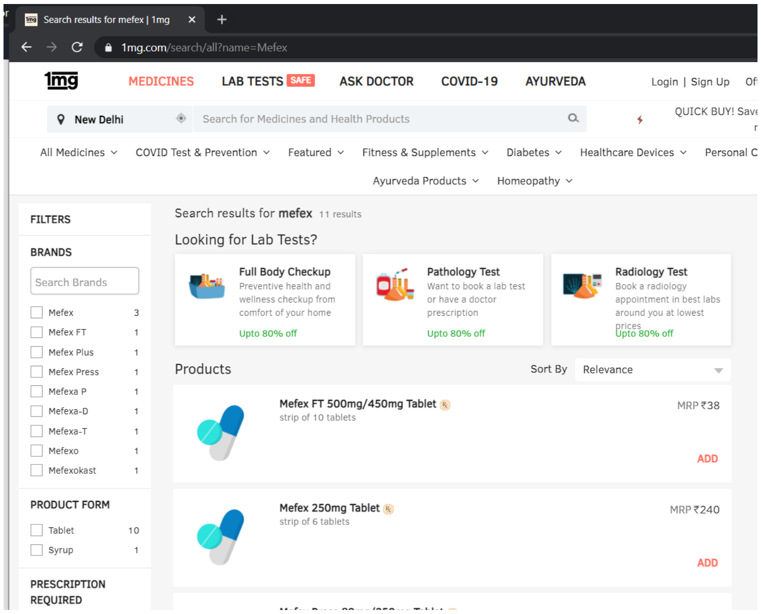


Figure 5.1: *Mefex Input*



```
weeMefex a ned arn Mefex ee eeeMefex ra neophartna Tab pharma
```

Figure 5.2: *Extracted Text*



Search results for **mefex** 11 results

Looking for Lab Tests?

**Full Body Checkup** Preventive health and wellness checkup from comfort of your home Upto 80% off

**Pathology Test** Want to book a lab test or have a doctor prescription Upto 80% off

**Radiology Test** Book a radiology appointment in best labs around you at lowest prices Upto 80% off

| | |
|---|---|
| □ Mefex | 3 |
| □ Mefex FT | 1 |
| □ Mefex Plus | 1 |
| □ Mefex Press | 1 |
| □ Mefexa P | 1 |
| □ Mefexa-D | 1 |
| □ Mefexa-T | 1 |
| □ Mefexo | 1 |
| □ Mefexokast | 1 |
| □ Tablet | 10 |
| □ Syrup | 1 |

Products

**Mefex FT 500mg/450mg Tablet** strip of 10 tablets MRP ₹38

**Mefex 250mg Tablet** strip of 6 tablets MRP ₹240

Figure 5.3: *Mefex Output*

### 5.1.2 Test Case 2 (Strip name: Evion)

The input image is of the medicine "Evion" (Refer Fig 5.4). The extracted text after applying our model is shown in Fig 5.5 and finally the output as mentioned in the Problem Definition(Refer Subsection 3.2) is shown in Fig 5.6. The extracted text from the input image consists of the medicine strip name i.e "Evion".

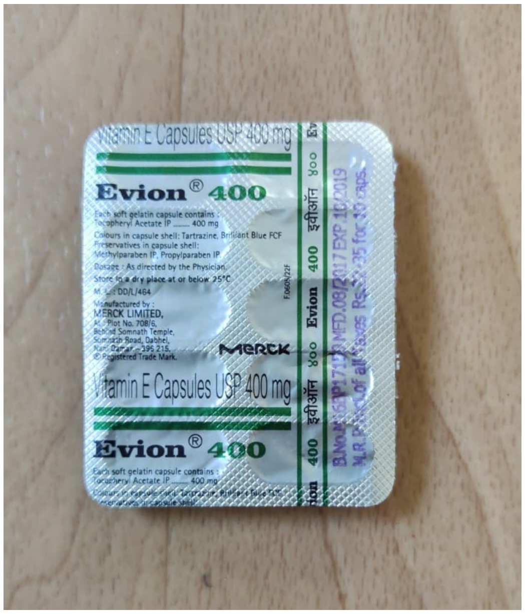
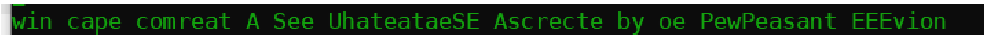


Figure 5.4: *Evion Input*



```
win cape comreat A See UhateataeSE Ascrecte by oe PewPeasant EEEvion
```

Figure 5.5: *Extracted Text*

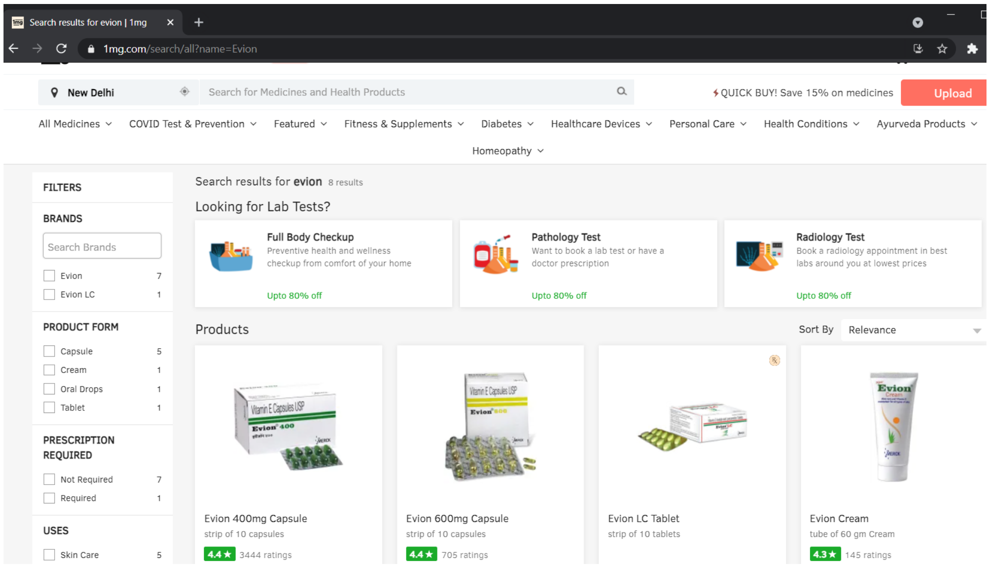


Figure 5.6: *Evion Output*

### 5.1.3 Test Case 3 (Strip name: Paracetamol)

The input image is of the medicine “Paracetamol” (Refer Fig 5.7). The extracted text after applying our model is shown in Fig 5.8 and finally the output as mentioned in the Problem Definition (Refer Subsection 3.2) is shown in Fig 5.9 . The extracted text from the input image consists of the medicine strip name i.e “racetamol”.

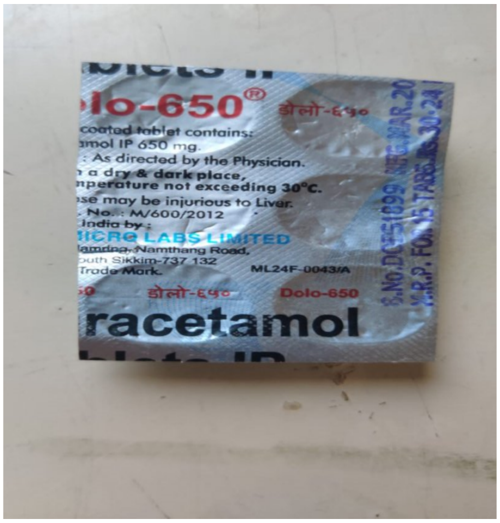
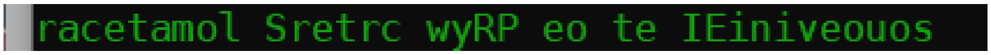
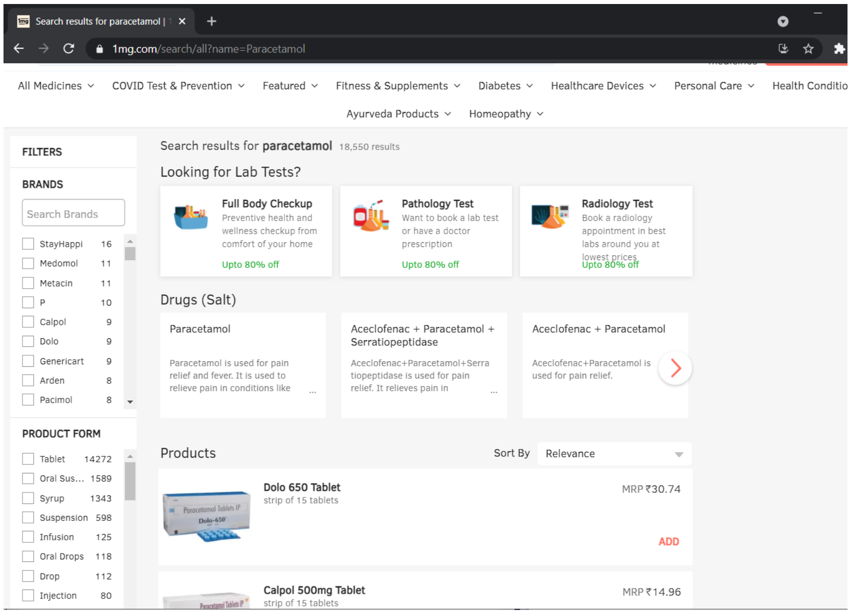


Figure 5.7: *Paracetamol Input*



```
racetamol Sretrc wyRP eo te IEiniveouos
```

Figure 5.8: *Extracted Text*



Search results for **paracetamol** 18,550 results

Looking for Lab Tests?

**Full Body Checkup** Preventive health and wellness checkup from comfort of your home Upto 80% off

**Pathology Test** Want to book a lab test or have a doctor prescription Upto 80% off

**Radiology Test** Book a radiology appointment in best labs around you at lowest prices Upto 80% off

Drugs (Salt)

Paracetamol — Paracetamol is used for pain relief and fever. It is used to relieve pain in conditions like ...

Aceclofenac + Paracetamol + Serratiopeptidase — Aceclofenac+Paracetamol+Serratiopeptidase is used for pain relief. It relieves pain in ...

Aceclofenac + Paracetamol — Aceclofenac+Paracetamol is used for pain relief.

Products

**Dolo 650 Tablet** strip of 15 tablets MRP ₹30.74

**Calpol 500mg Tablet** strip of 15 tablets MRP ₹14.96

Figure 5.9: *Paracetamol Output*

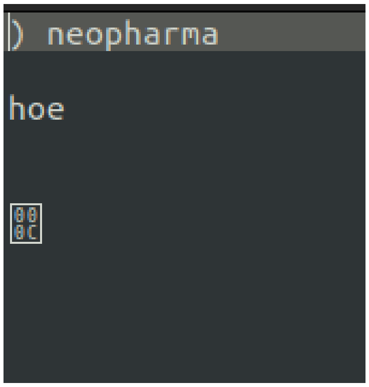
## 5.2 Comparison and Analysis

Here the images below will show the comparison between the outputs in the case of applying the plain Tesseract OCR and applying our model on the imput images.

### 5.2.1 Test Case 1

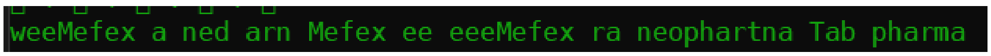
Fig 5.10 shows the result of applying plain Tesseract OCR on the image strip (Refer Fig 5.1 for input image) and Fig 5.11 shows the result of applying our model on the input image.

It can be noticed that due to the unevenness, contrasting background and the angle of the text, Tesseract OCR is not able to detect the required name, whereas our model is able to extract the closest match for the medicine name which in this case is the full medicine name “Mefex”.



```
) neopharma

hoe
```

Figure 5.10: *Tesseract on Mefex*



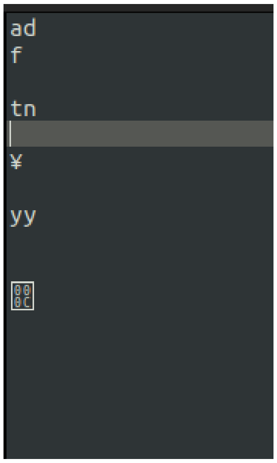
```
weeMefex a ned arn Mefex ee eeeMefex ra neophartna Tab pharma
```

Figure 5.11: *Our model on Mefex*

### 5.2.2 Test Case 2

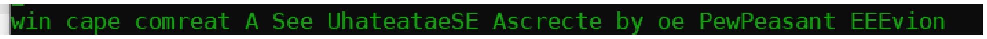
Fig 5.12 shows the result of applying plain Tesseract OCR on the image strip (Refer Fig 5.4 for input image) and Fig 5.13 is the result of applying our model on the input image.

It can be noticed that due to the unevenness, contrasting background, picture resolution,etc., Tesseract OCR is not able to detect the required name, whereas our model is able to extract the closest match for the medicine name which in this case is the full medicine name “Evion”.



```
ad
f

tn
|
¥

yy
```

Figure 5.12: *Tesseract on Evion*



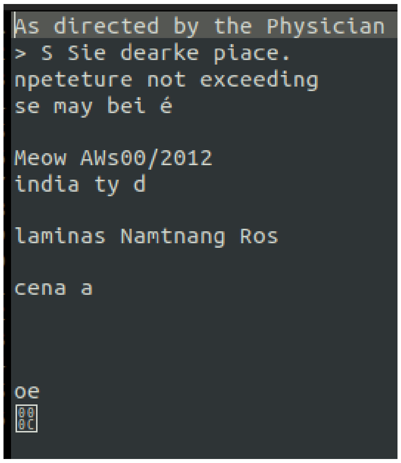
```
win cape comreat A See UhateataeSE Ascrecte by oe PewPeasant EEEvion
```

Figure 5.13: *Our model on Evion*

### 5.2.3 Test Case 3

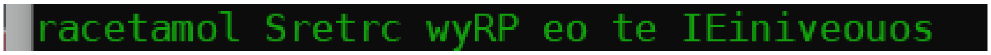
Fig 5.14 shows the result of applying plain Tesseract OCR on the image strip (Refer Fig 5.7 for input image) and Fig 5.15 is the result of applying our model on the input image.

It can be noticed that due to the unevenness, contrasting background, picture resolution and incomplete name on the strip, Tesseract OCR is not able to detect the required name, whereas our model is able to extract the closest match for the medicine name which in this case is the medicine name “racetamol” (which is further string matched to Paracetamol).



```
As directed by the Physician
> S Sie dearke piace.
npeteture not exceeding
se may bei é

Meow AWs00/2012
india ty d

laminas Namtnang Ros

cena a



oe
```

Figure 5.14: *Tesseract on Paracetamol*



```
racetamol Sretrc wyRP eo te IEiniveouos
```

Figure 5.15: *Our model on Paracetamol*

# Chapter 6

# Conclusion and Future work

## 6.1 Conclusion

In this work, we have proposed a method for the text extraction from a given picture of medical strip. This method employs pre-processing methods like MSER, edge detection and Stroke Width Transform for detecting text regions. Using MSER, potential text regions are identified in the image based on consistency of colour and contrast. Further ahead, we use edge detection algorithms to identify pixels belonging to an "edge". These edge pixels are then to be used in the stroke width algorithm to detect the strokes in the image and further filter out non-text regions. The remaining text regions are then filtered based on their geometric properties and fed as input to Tesseract, a segmentation-free OCR software to extract text from the image. The extracted text string is matched with our medicine database to retrieve the name of the medicine and is then used to search in an external website which will display the search results based on the name of the said medicine strip.

## 6.2 Future Work

A possible extension is to create a mobile application which allows the user to take a photo of a medicine strip on his/her phone and redirects the user to a website with additional information about the medicine. The application would use the same backend as that of the very basic web application developed in this paper.

# References

[1] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end to-end trainable neural network for spotting text with arbitrary shapes," in *Proc. European Conference on Computer Vision (ECCV)*, Researchgate, 2018.

[2] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Reading text in uncontrolled conditions," in *IEEE International Conference on Computer Vision*, Google Inc, 2013.

[3] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," in *Int. J. Comput. Vision, vol. 65*, Researchgate, 2005.

[4] J. Canny, "A computational approach to edge detection," in *IEEE Trans. Pattern Anal. Mach. Intell., vol. 8*, University of Berkely, 1986.

[5] Y. Li, W. Jia, C. Shen, and A. van den Hengel, "An indicator of text in the wild," in *IEEE Transactions on Image Processing, vol. 23*, 2014.

[6] K. Hea, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Comp. Vis*, 2010.

[7] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *18th IEEE International Conference on Image Processing*, Stanford University, 2011.

[8] S. A. Pote and M. A. Mehta, “An improved technique to detect text from scene videos,” in *International Conference on Communication and Signal Processing (ICCSP)*, Sarvajanik College of Engineering and Technology, 2017.

[9] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Microsoft, 2010.

[10] T. Mehra, “Design and development of medicine text identification system,” Thapar University, Patiala, Punjab, 2017.

[11] D. Marr, “Zero-crossings,” in *Vision*, pp. 54–61, The MIT Press, 2010.

[12] A. C. Özgen, M. Fasounaki, H. Kemal, and Ekenel, “Text detection in natural and computer-generated images,” SiMiT Lab, Department of Computer Engineering Istanbul, Turkey, 2018.

[13] M. Namysl and I. Konya, “Efficient, lexicon-free ocr using deep learning,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Fraunhofer IAIS, 2019.