PROJECT

IMAGE TRANSLATION FROM ENGLISH TO KANNADA

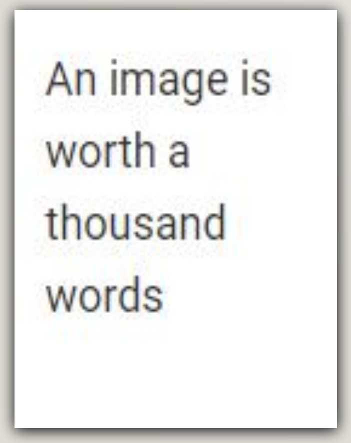
PES1201701562	Apeksha gaonakar
PES1201701655	Shamitha. S
PES1201701156	Astha Singh

INTRODUCTION :

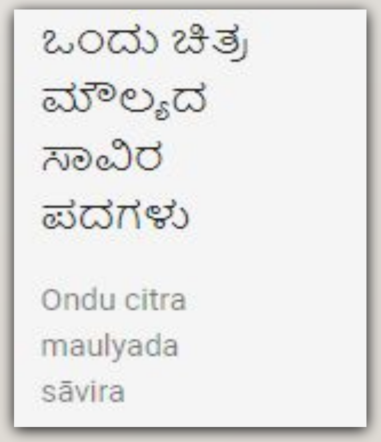
- **Image translation** refers to a technology where we can translate the text on images (posters, banners, menu list, sign board, document, screenshot etc.).
- This is done by **pre-processing** of the image followed by **segmentation** to extract the text contained in the image and then **translating** the text into a language of choice.

PROBLEM STATEMENT

To read text written in English from an Image and translating it into Kannada language.



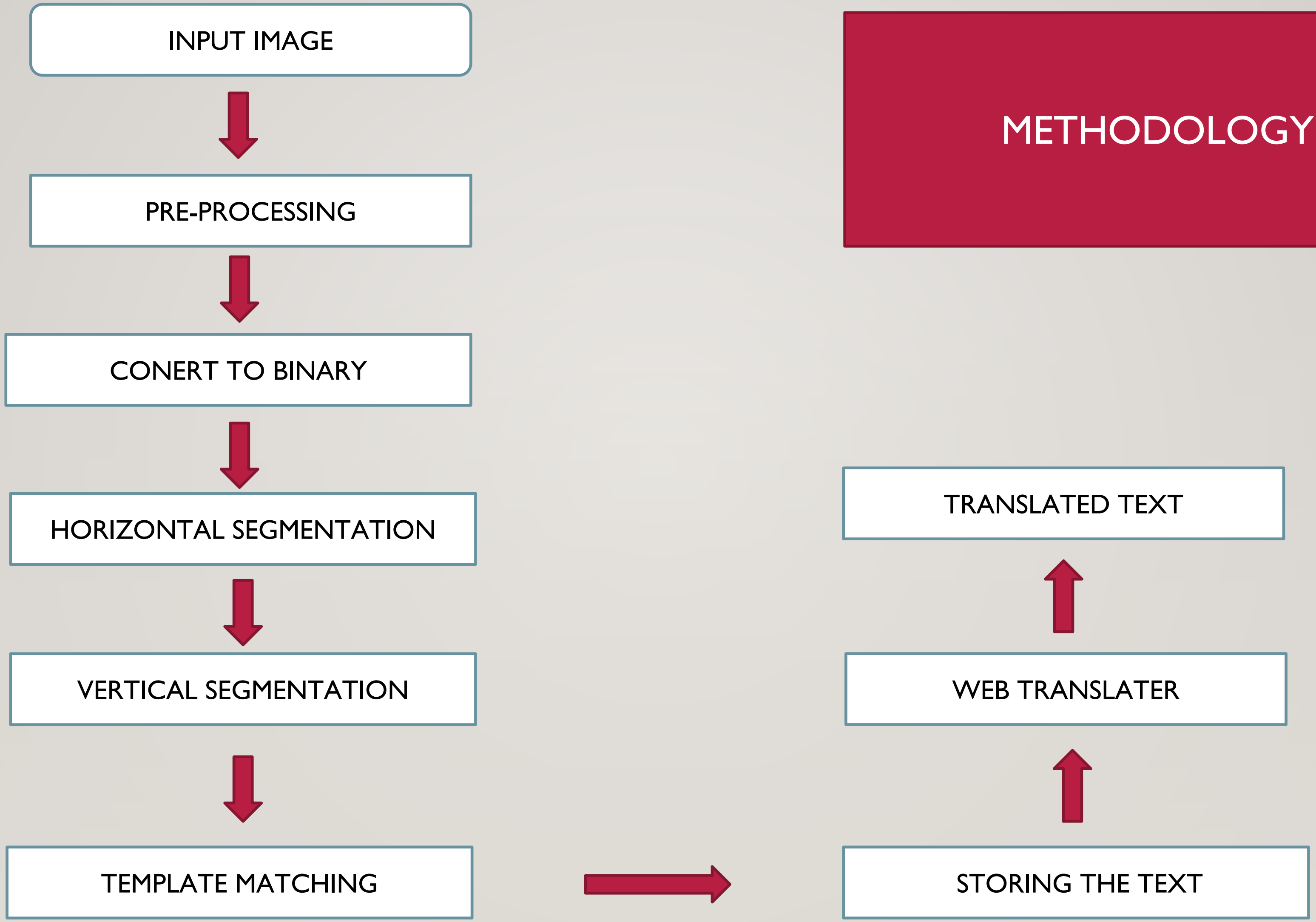
An image is
worth a
thousand
words



ಒಂದು ಚಿತ್ರ
ಮೌಲ್ಯದ
ಸಾವಿರ
ಪದಗಳು

Ondu citra
maulyada
sāvira

METHODOLOGY



INPUT IMAGE

PRE-PROCESSING

CONERT TO BINARY

HORIZONTAL SEGMENTATION

VERTICAL SEGMENTATION

TEMPLATE MATCHING

STORING THE TEXT

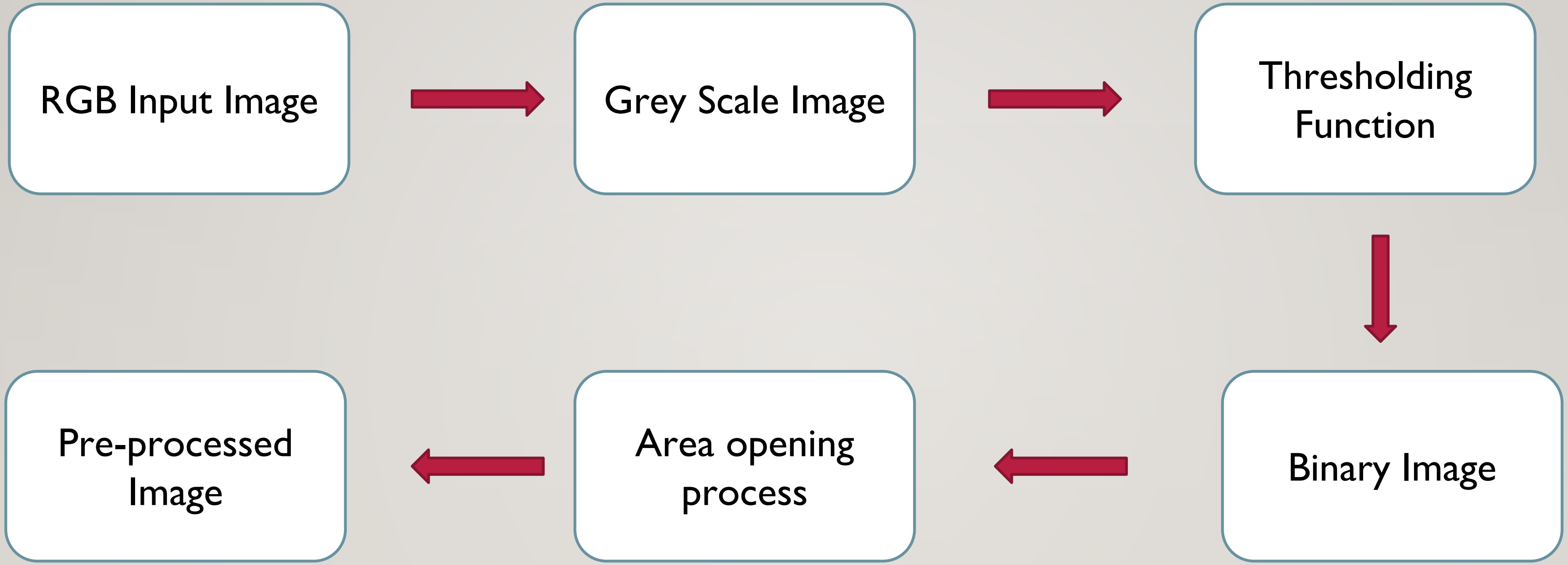
WEB TRANSLATER

TRANSLATED TEXT

IMPLEMENTATION:

- **Pre-processing**:

The aim of pre-processing is an improvement of the image data that suppresses all the unwanted distortions and enhances some image features important for further processing.



RGB Input Image

Grey Scale Image

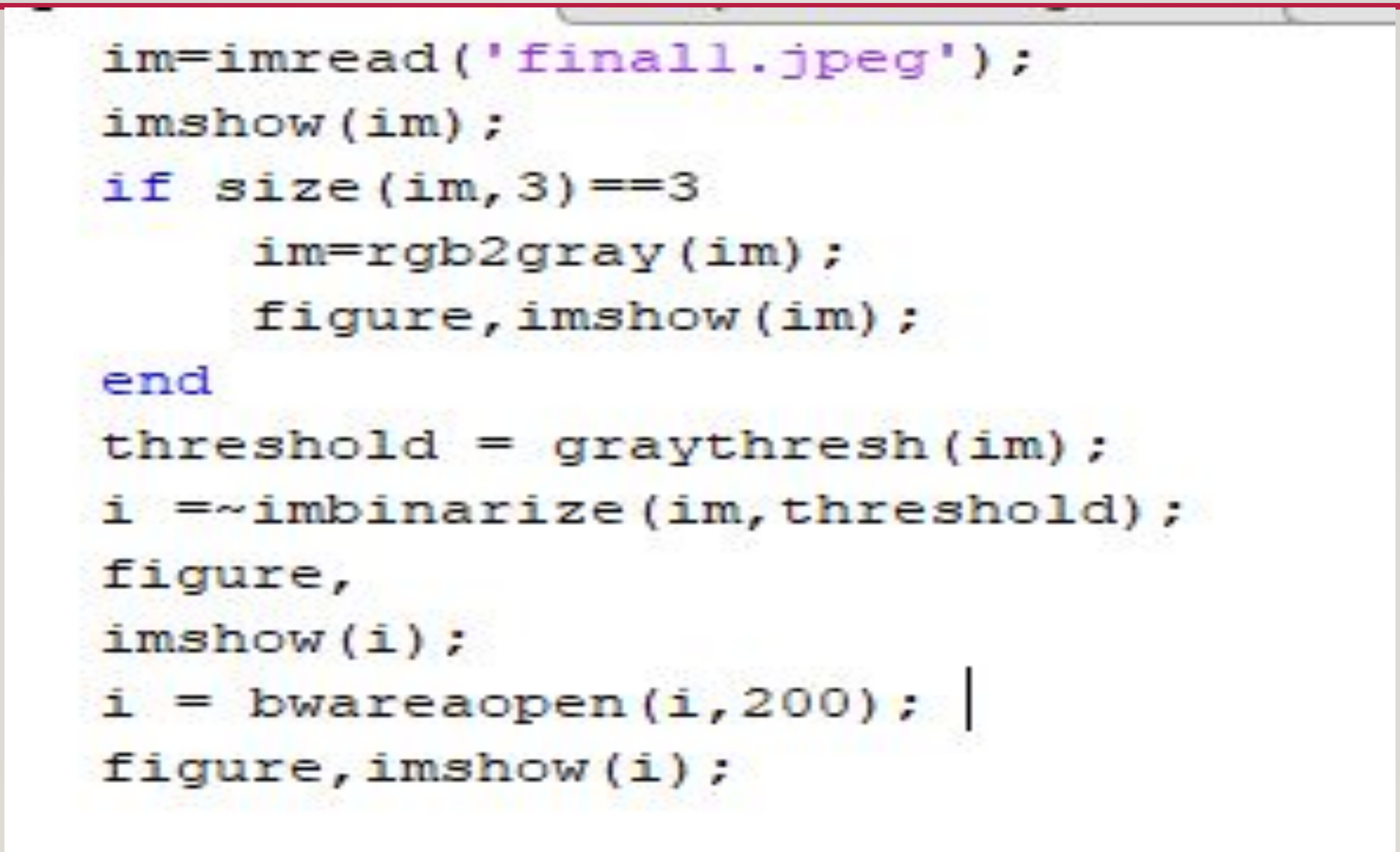
Thresholding Function

Binary Image

Area opening process

Pre-processed Image

PRE-PROCESSING CODE:



```
im=imread('finall.jpeg');
imshow(im);
if size(im,3)==3
    im=rgb2gray(im);
    figure,imshow(im);
end
threshold = graythresh(im);
i =~imbinarize(im,threshold);
figure,
imshow(i);
i = bwareaopen(i,200);
figure,imshow(i);
```

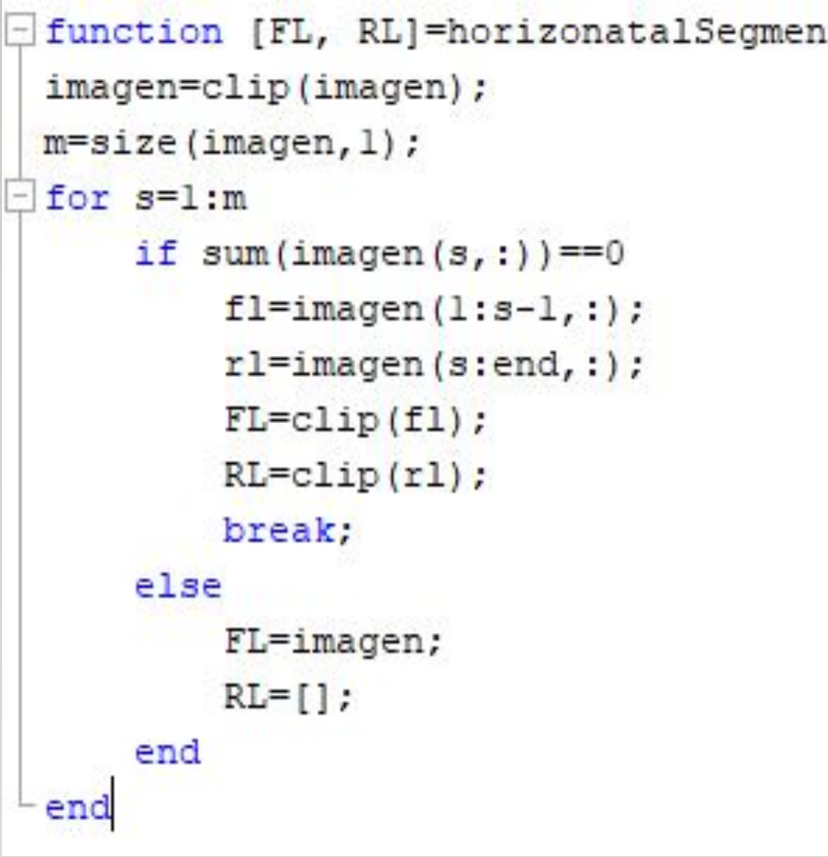
SEGMENTATION:

- Segmentation is an important process in image recognition, its referred to the isolation of the position of the words within the image, and then the extraction of each letter individually.

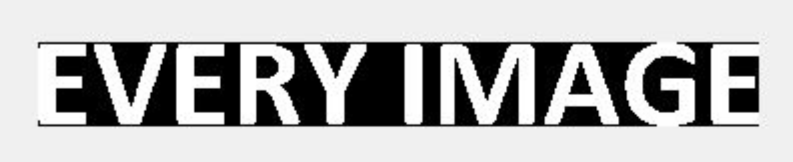
1: Horizontal segmentation

2: Vertical segmentation

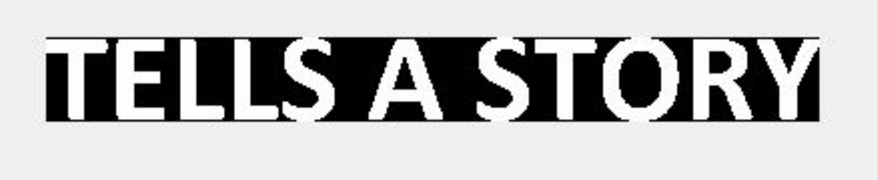
HORIZONTAL SEGMENTATION



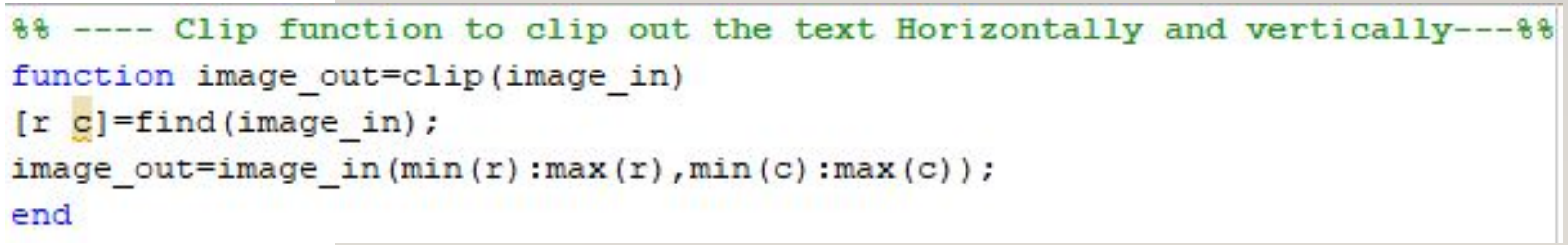
```
function [FL, RL]=horizonatalSegmentation(imagen)
imagen=clip(imagen);
m=size(imagen,1);
for s=1:m
    if sum(imagen(s,:))==0
        fl=imagen(1:s-1,:);
        rl=imagen(s:end,:);
        FL=clip(fl);
        RL=clip(rl);
        break;
    else
        FL=imagen;
        RL=[];
    end
end
```



EVERY IMAGE

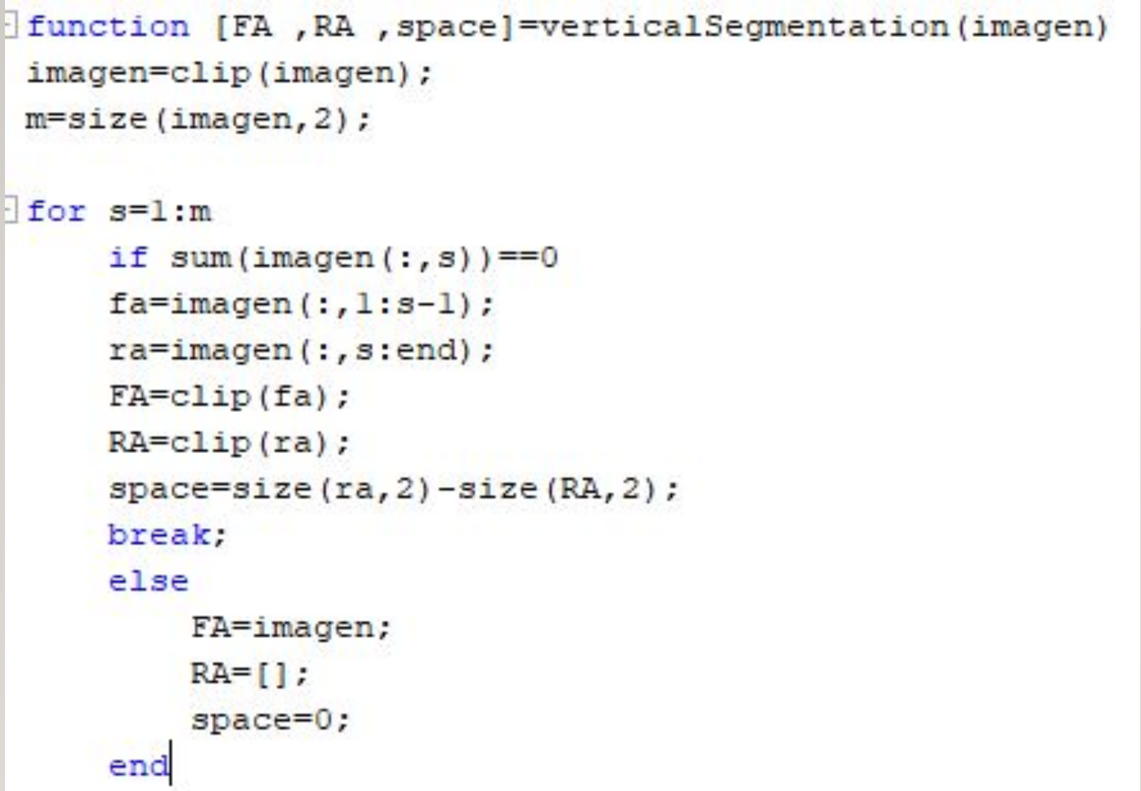


TELLS A STORY

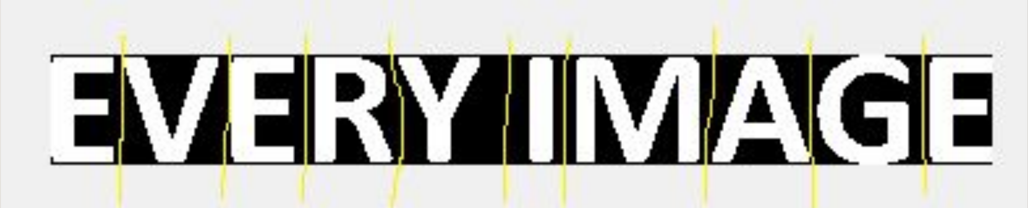


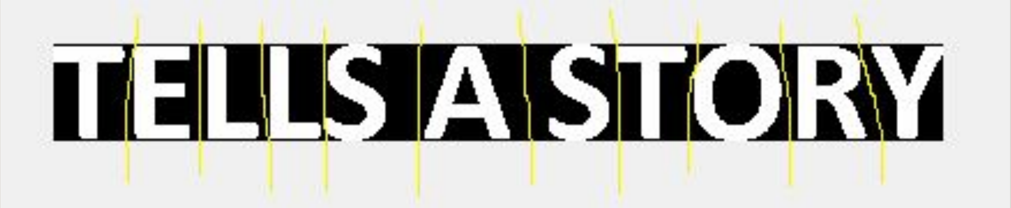
```
%% ---- Clip function to clip out the text Horizontally and vertically---%%
function image_out=clip(image_in)
[r c]=find(image_in);
image_out=image_in(min(r):max(r),min(c):max(c));
end
```

VERTICAL SEGMENTATION



```
function [FA ,RA ,space]=verticalSegmentation(imagen)
imagen=clip(imagen);
m=size(imagen,2);

for s=1:m
    if sum(imagen(:,s))==0
    fa=imagen(:,1:s-1);
    ra=imagen(:,s:end);
    FA=clip(fa);
    RA=clip(ra);
    space=size(ra,2)-size(RA,2);
    break;
    else
        FA=imagen;
        RA=[];
        space=0;
    end
```

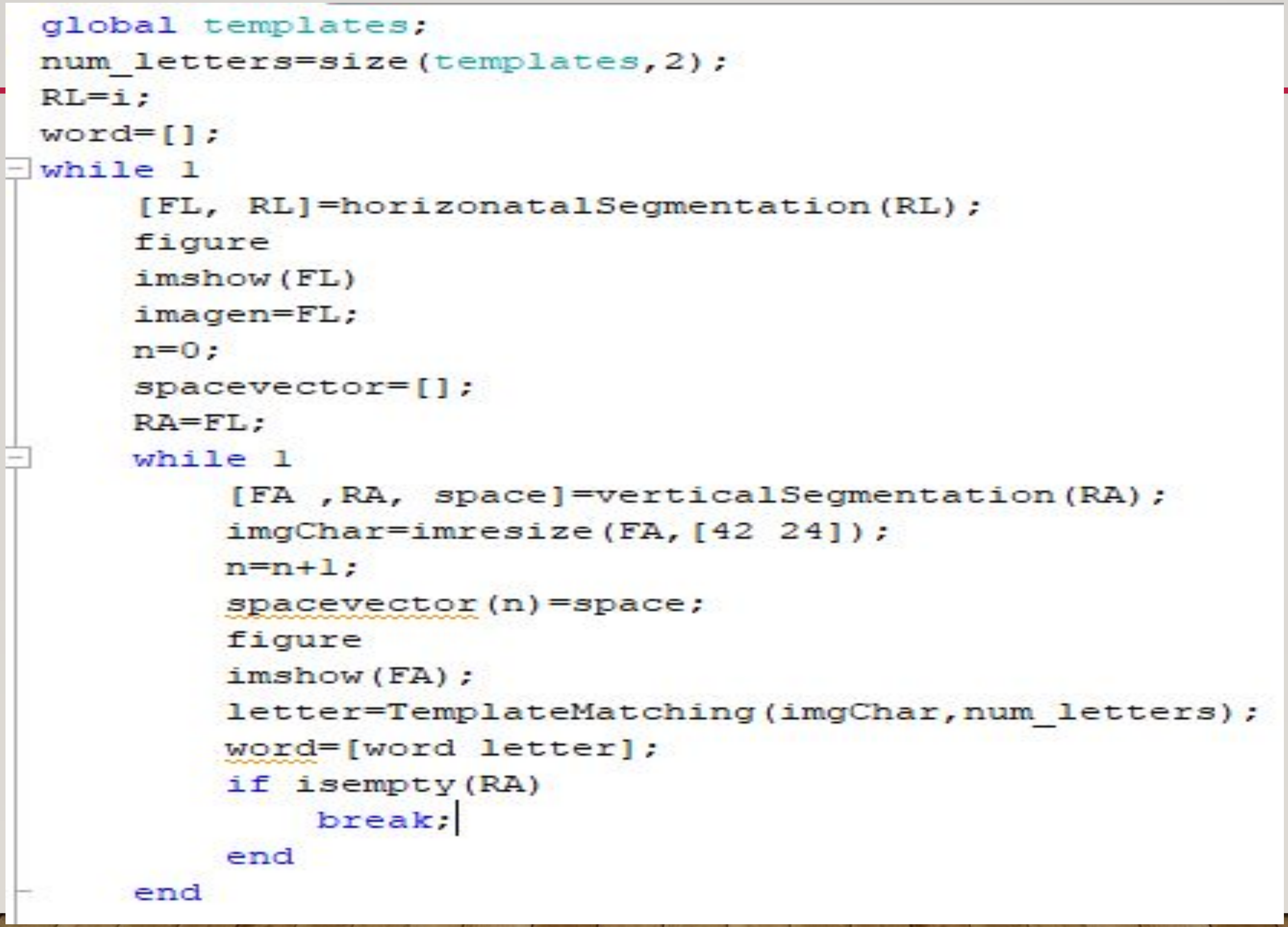




CORRELATION

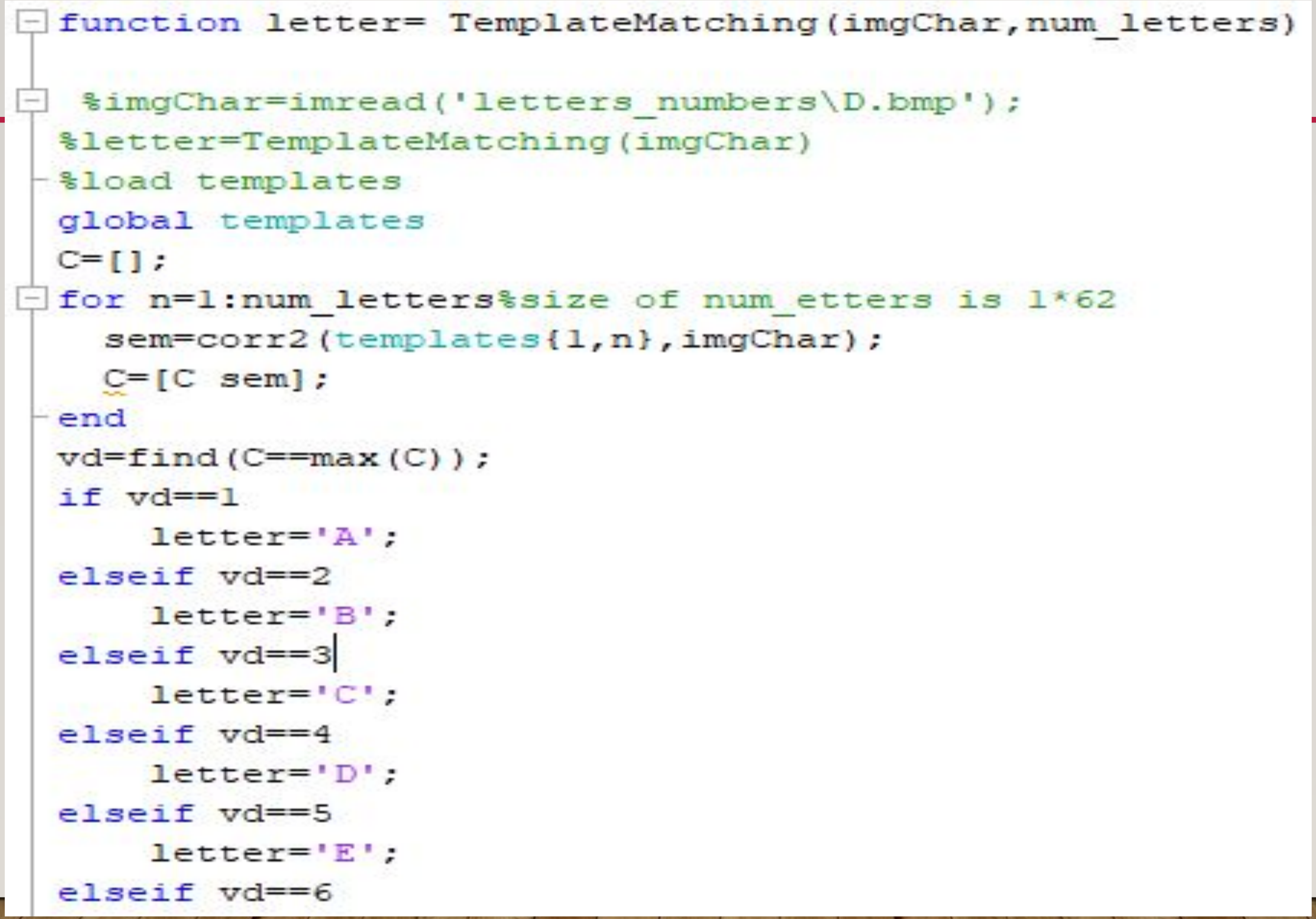
The segmented image is resized to the size of the template provided and correlated with the template image using matlab function corr2. This is stored in an array. After this we find the index value and if the index value matches to the condition a new text file is created and the letters are written in the file.

CORRELATION CODE



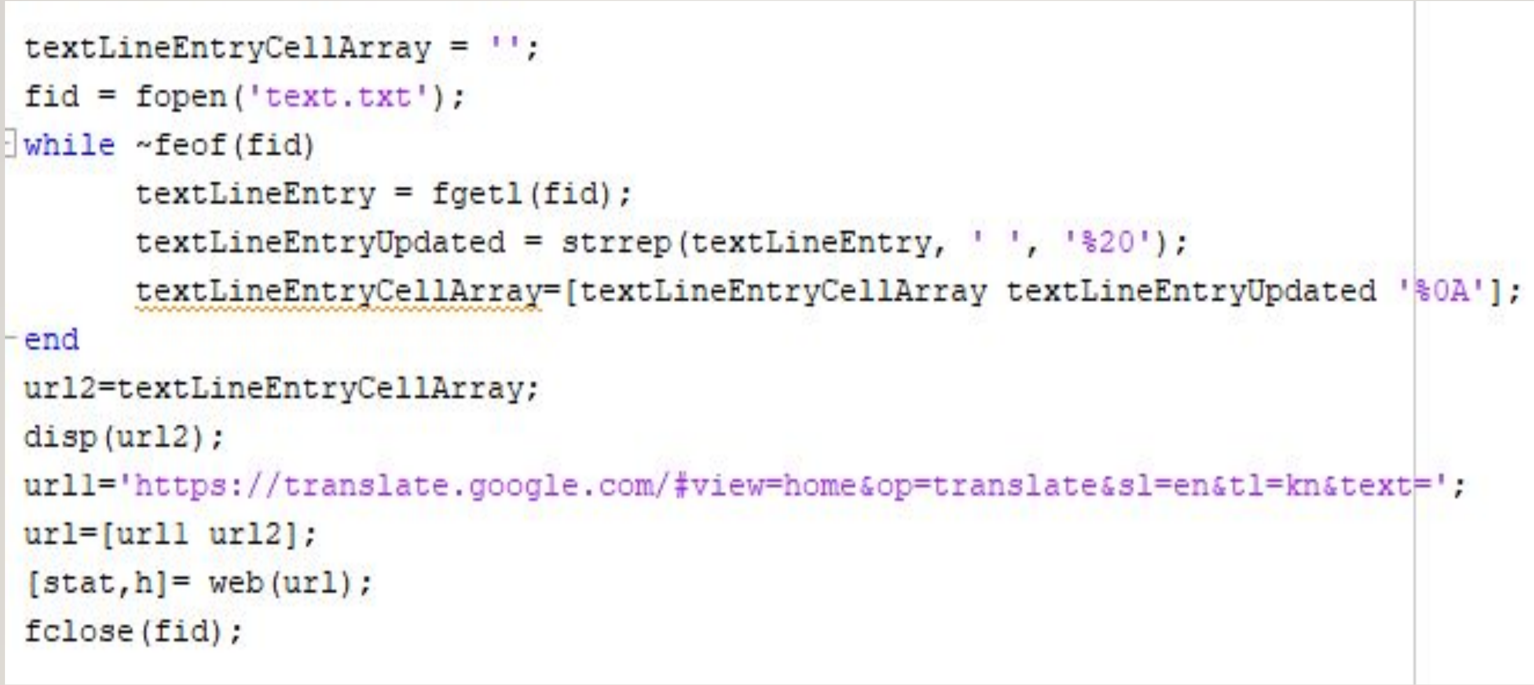
```
global templates;
num_letters=size(templates,2);
RL=i;
word=[];
while 1
    [FL, RL]=horizonatalSegmentation(RL);
    figure
    imshow(FL)
    imagen=FL;
    n=0;
    spacevector=[];
    RA=FL;
    while 1
        [FA ,RA, space]=verticalSegmentation(RA);
        imgChar=imresize(FA,[42 24]);
        n=n+1;
        spacevector(n)=space;
        figure
        imshow(FA);
        letter=TemplateMatching(imgChar,num_letters);
        word=[word letter];
        if isempty(RA)
            break;
        end
    end
```

TEMPLATE MATCHING



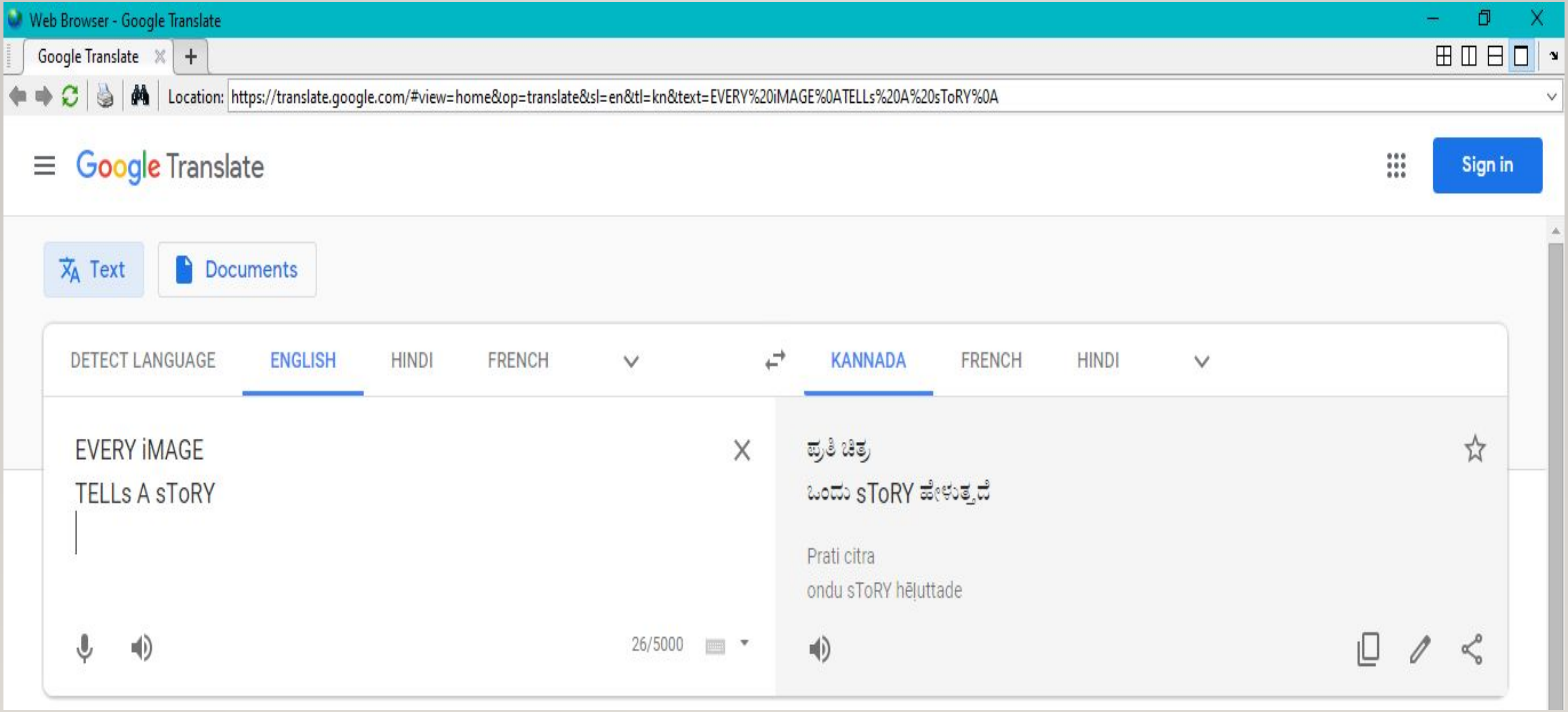
```
function letter= TemplateMatching(imgChar,num_letters)

 %imgChar=imread('letters_numbers\D.bmp');
%letter=TemplateMatching(imgChar)
%load templates
global templates
C=[];
for n=1:num_letters%size of num_etters is 1*62
  sem=corr2(templates{1,n},imgChar);
  C=[C sem];
end
vd=find(C==max(C));
if vd==1
    letter='A';
elseif vd==2
    letter='B';
elseif vd==3
    letter='C';
elseif vd==4
    letter='D';
elseif vd==5
    letter='E';
elseif vd==6
```

TRANSLATING



```
textLineEntryCellArray = '';
fid = fopen('text.txt');
while ~feof(fid)
      textLineEntry = fgetl(fid);
      textLineEntryUpdated = strrep(textLineEntry, ' ', '%20');
      textLineEntryCellArray=[textLineEntryCellArray textLineEntryUpdated '%0A'];
end
url2=textLineEntryCellArray;
disp(url2);
url1='https://translate.google.com/#view=home&op=translate&sl=en&tl=kn&text=';
url=[url1 url2];
[stat,h]= web(url);
fclose(fid);
```

FINAL OUTPUT



Location: https://translate.google.com/#view=home&op=translate&sl=en&tl=kn&text=EVERY%20iMAGE%0ATELLs%20A%20sToRY%0A

EVERY iMAGE
TELLs A sToRY

ಪ್ರತಿ ಚಿತ್ರ
ಒಂದು sToRY ಹೇಳುತ್ತದೆ

Prati citra
ondu sToRY hēḷuttade

CONCLUSION AND FURTHER ENHANCEMENTS

- This code works for fonts which is similar to the given template.
- It can be further enhanced by using neural network instead of correlation where the handwritten dataset is provided and training, testing is performed on that dataset.
- This can also be implemented using OCR inbuilt function of matlab.

REFERENCES

- https://www.mathworks.com/help/images/translate-an-image.html
- https://circuitdigest.com/tutorial/vehicle-number-plate-detection-using-matlab-and-image-processing
- https://ijireeice.com/wp-content/uploads/2013/03/IJIREEICE3G-a-ragini-RECOGNITION-OF-VEHICLE-NUMBER-PLATE-USING-MATLAB.pdf

THANK YOU