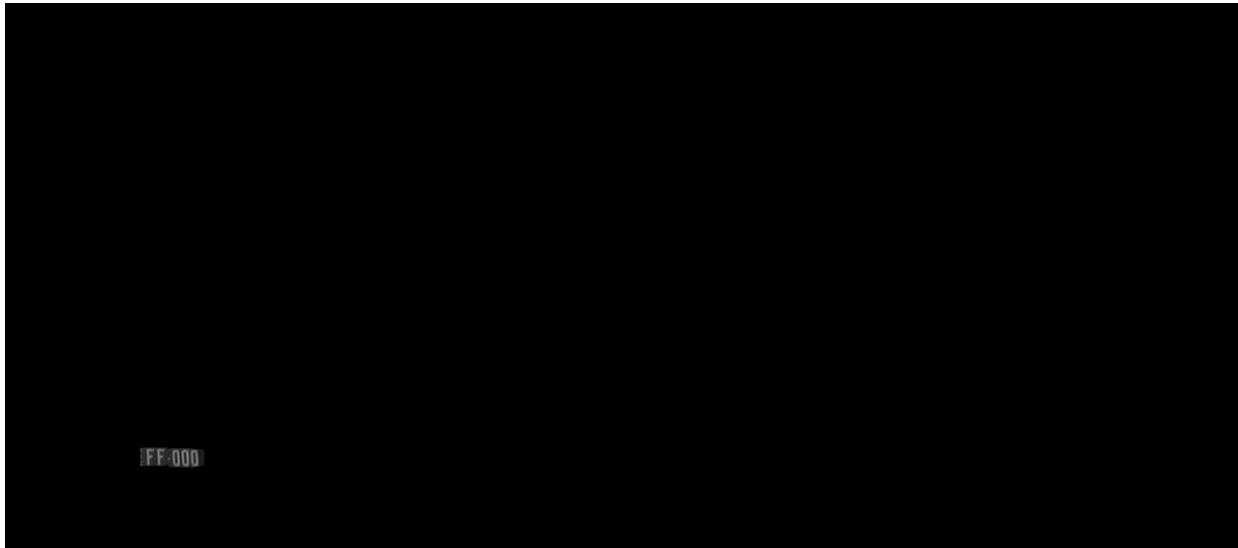


Report
FRI_D2_G15_Soumya
Soumya Sen Gupta 2012EEY7535
Siddharth Srivastava 2012EEY7506

Demo2_a

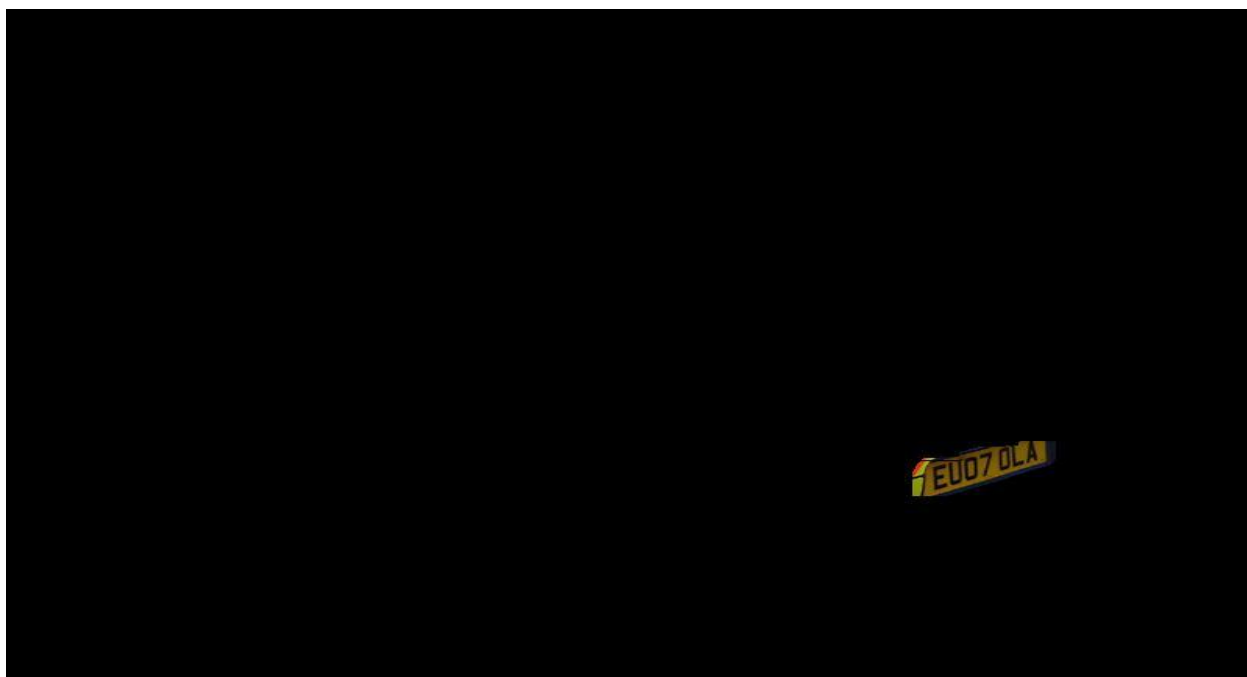


Output

Demo2_b



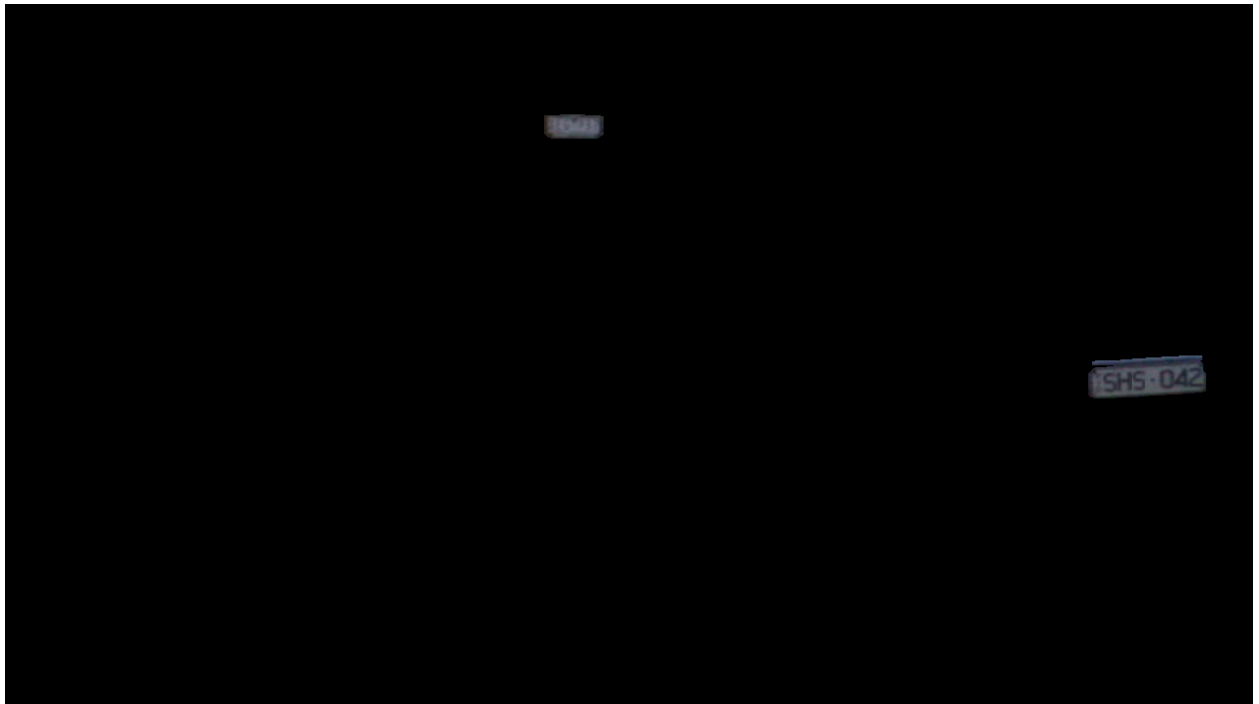
Output



Demo2_c



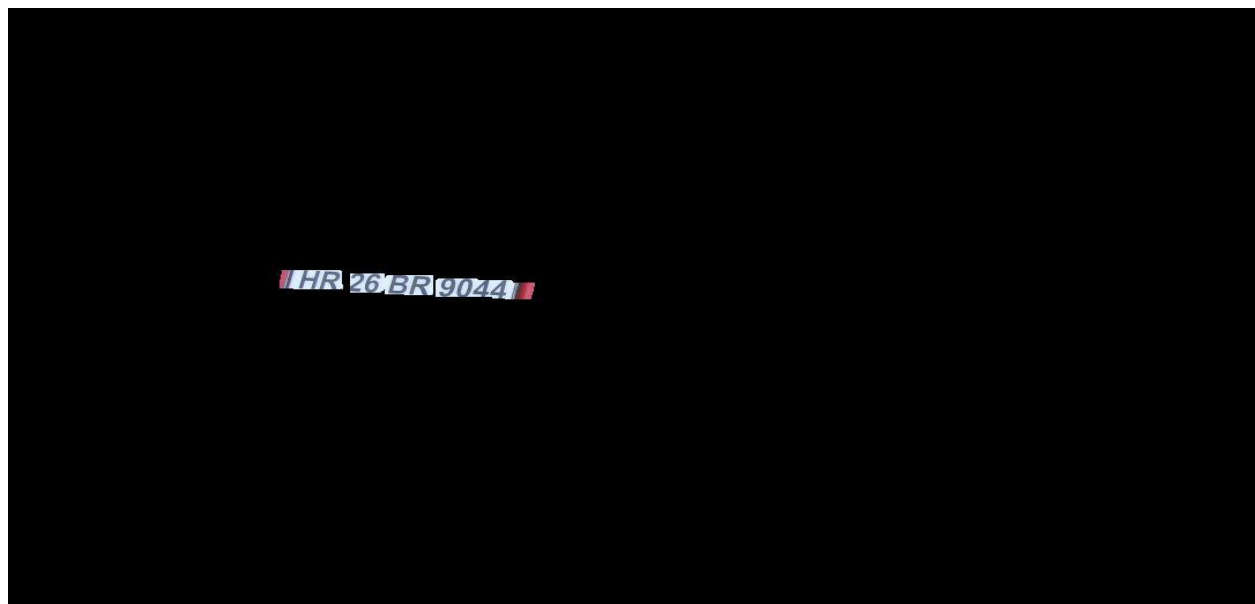
Output



Demo2_d



Output



ALGORITHM

- A. Read the image
- B. Convert it into gray scale
- C. Remove noise using a Median Filter
- D. Do an histogram equalization to increase contrast
- E. Find out edges in the image
- F. Pass the image through a low pass filter to smooth the edges
- G. Convert it to a binary form
- H. Dilate the binary image
- I. Get the connected components from the dilated image having a certain aspect ratio
- J. Fill the connected components
- K. Use a vertical and horizontal mask to remove unnecessary components
- L. Again dilate the new image
- M. Get the connected components from the image. This gives the license plate

CODE

```
clearvars
close all
clc
warning off

I = imread('FRI_D2_G15_Soumya_5.jpg');
figure, imshow(I);

[imageRows, imageCols, noOfColors] = size(I);

gray_I = rgb2gray(I);
% figure, imshow(gray_I);

gray_med_I = medfilt2(gray_I, [3 3]);

I_hist = histeq(gray_med_I);

edge_I = edge(I_hist, 'sobel'); %finding edges
% figure, imshow(edge_I);

filter_freq = 60;
[rows cols] = size(edge_I);
p=rows/2;
q=cols/2;
fft_trans=fft2(edge_I);
fft_shift=fftshift(fft_trans);

for i=1:rows
    for j=1:cols
        distance=sqrt((i-p)^2+(j-q)^2);
        low_pass_filter(i,j)=exp(-(distance)^2/(2*(filter_freq^2)));
    end
end

filtered_I = zeros(rows, cols);

filtered_a=fft_shift.*low_pass_filter;
ifft_shift=ifftshift(filtered_a);
ifft_trans=abs(ifft2(ifft_shift));
filtered_I(:,:)=ifft_trans;

% figure, imshow(filtered_I)

BW = im2bw(filtered_I, 0.1);
% figure, imshow(BW);

se = strel('line',4,1);
dilatedImg = imdilate(BW,se);
figure, imshow(dilatedImg);
% imwrite(dilatedImg, 'dilated_2d.jpg', 'jpg');
```

```

[imx,imy]=size(dilatedImg)

n1 = GetConnectedComponents(dilatedImg, 4, 500, 500, 7000, 300, 150);
original = n1;
% binaryFilledImage = imfill(n1,'holes');
filled = imfill(n1, 'holes');
holes = filled & ~original;
bigholes = bwareaopen(holes, 700);
smallholes = holes & ~bigholes;
binaryFilledImage = original | smallholes;

size(I)
size(binaryFilledImage)
% figure,imshow(binaryFilledImage);
% imwrite(n1, 'result_2c.png', 'png');

se = strel('line',2,0);
binaryFilledImage = imdilate(binaryFilledImage,se);
masked_image = MaskImage(binaryFilledImage, 1, 20);
masked_image = MaskImage(masked_image, 2, 20);

se = strel('line',6,1);
masked_image = imdilate(masked_image,se);
masked_image = GetConnectedComponents(masked_image, 4, 50, 900, 5000, 300,
65);

finalImage = I;
for i=1:imageRows
    for j=1:imageCols
        if(masked_image(i,j)==0)
            finalImage(i,j,1) = 0;
            finalImage(i,j,2) = 0;
            finalImage(i,j,3) = 0;
        end
    end
end
end

figure,imshow(finalImage);

```

```

function [ output_image ] = GetConnectedComponents( input_image,
no_of_connected_comp, least_number_of_comp, min_connected_comp,
max_connected_comp, max_width, max_height)

[imx,imy]=size(input_image);

L = bwlabel(input_image, no_of_connected_comp);% Calculating connected
components
output_image=zeros(imx,imy);
count = 0;
for z = 1:least_number_of_comp
    [r,c] = find(L==z);
    rc = [r c];
    [sx sy]=size(rc);

    if ((sx > min_connected_comp) && (sx < max_connected_comp))
        noOfWhitePixels = 0;
        count = count + 1;
        x2=[];
        y2=[];
        for i=1:sx

            x1=rc(i,1);
            y1=rc(i,2);

            x2(i) = x1;
            y2(i) = y1;

            noOfWhitePixels = noOfWhitePixels + 1;

        end

        height = length(unique(x2));
        width = length(unique(y2));
        val = (width/height);
        if((width/height) < 4) && ((width/height) >1) && (width < max_width)
        && (height < max_height))

            if(noOfWhitePixels > (0.2*(width*height)))
                for k=1:length(x2)
                    output_image(x2(k),y2(k))=255;
                end
            end

        end

        % figure,imshow(n1);

    end

end

end

end

```



```

function [ masked_image ] = MaskImage(image, mask_type, mask_size)

%     figure, imshow(image);
[rows, cols, colors] = size(image);
masked_image = image;
if(mask_type == 1)

    for i=1:cols
        for j = 1:(rows-mask_size-1)
            temp = image(j:j+mask_size-1,i);
            if(temp(1,1) == 0) && (temp(mask_size,1) == 0)
                masked_image(j:j+mask_size-1, i) = 0;
            end
        end
    end

end

if(mask_type == 2)

    for i=1:rows
        for j = 1:(cols-mask_size-1)
            temp = image(i,j:j+mask_size-1);
            if(temp(1,1) == 0) && (temp(1,mask_size) == 0)
                masked_image(i,j:j+mask_size-1) = 0;
            end
        end
    end

end

%     figure, imshow(masked_image);

end

```