Mehul Raj

18bce1146

Essentials of Data Analytics

Prof: Dr. Sweetlin Hemalatha C ma'am

Digital Assignment 3

Title: Retina face Mask: A Face Mask Detector

This paper focuses on developing a system in which they have used retina face mask to detect whether a person is wearing mask or not, this will definitely can help our frontline warriors on a large scale.

They have mentioned that they have used:

- RetinaFaceMask + Resnet50
- RetinaFaceMask + MobileNet

2 transfer learning algorithms have been used:

- ImageNet
- Wider Face

Using these systems, the complete model has been developed by keeping the transfer learning with MobileNet and Resnet50 model as a backbone.

Experiment results show that RetinaFaceMask achieves state-of-the-art results on a public face mask dataset with 2.3% and 1.5% higher than the baseline result in the face and mask detection precision, respectively, and 11.0% and 5.9% higher than baseline for recall.

Also, they have done this part in 2 steps which include haar cascade face detection and using that as an input for the model to detect whether the person is wearing mask or not.

This basically helps to remove all other unnecessary pixels from the image that might affect the resultant of the model.

In this, I have used different models and have used ImageNet as the transfer learning and found inceptionV3 as the fastest and Resnet50 as the best result giving model.

Citation:

Jiang, M., Fan, X. and Yan, H., 2020. RetinaMask: a face mask detector. *arXiv preprint arXiv:2005.03950*.

Title: Image Segmentation Techniques

This paper focuses on different image segmentation techniques and how these techniques are actually implemented.

Image segmentation techniques are basically used as the fundamental for object detection. As one can see in the paper, these techniques have been used to extract the edges out of the image that helps us in the interpretations for different results based on our requirements.

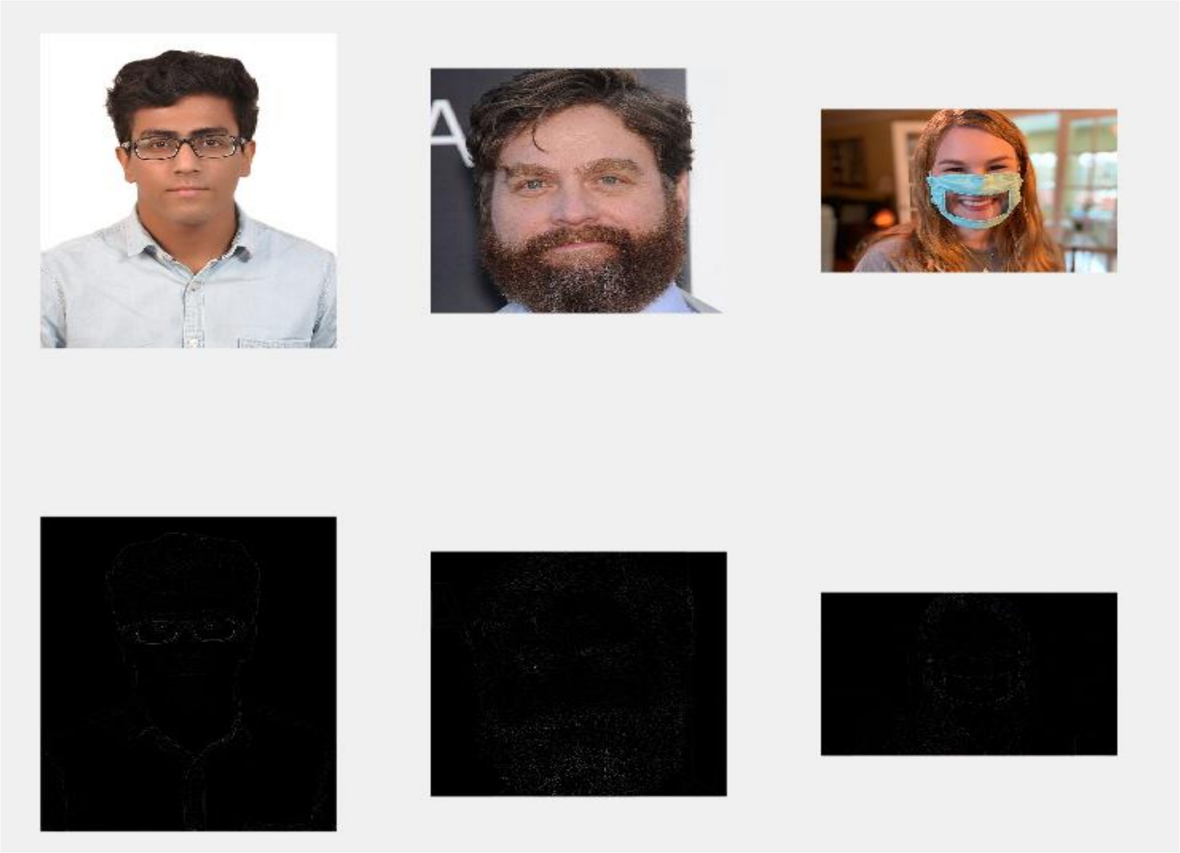
I have used MATLAB and python to develop the code for three images:

1. Person with no mask and no beard
2. Person with no mask and beard
3. Person with transparent mask

Here are the results:

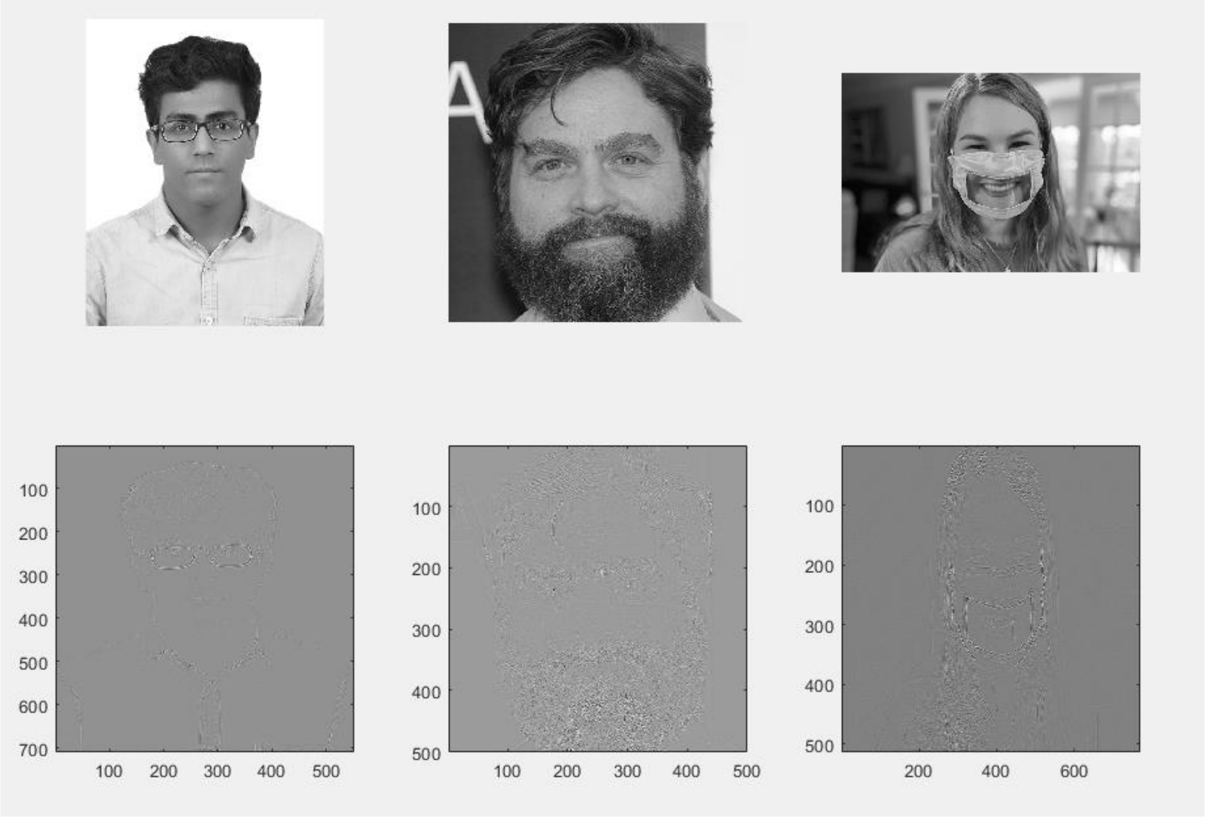
Code 1

```
% images
I1 = imread('mehul.jpeg');
I2 = imread('beard-face.jpg');
I3 = imread('transparent-face-mask.jpg');

% filter
g = fspecial('gaussian',[5 5],1.5);

% Image Plots
subplot(2,3,1), imshow(I1);
subplot(2,3,2), imshow(I2);
subplot(2,3,3), imshow(I3);

% image filtering
Is1 = imfilter(I1, g);
Ie1 = (I1-Is1);
subplot(2,3,4), imshow(Ie1);
Is2 = imfilter(I2, g);
Ie2 = (I2-Is2);
subplot(2,3,5), imshow(Ie2);
Is3 = imfilter(I3, g);
Ie3 = (I3-Is3);
subplot(2,3,6), imshow(Ie3);
```



Code 2

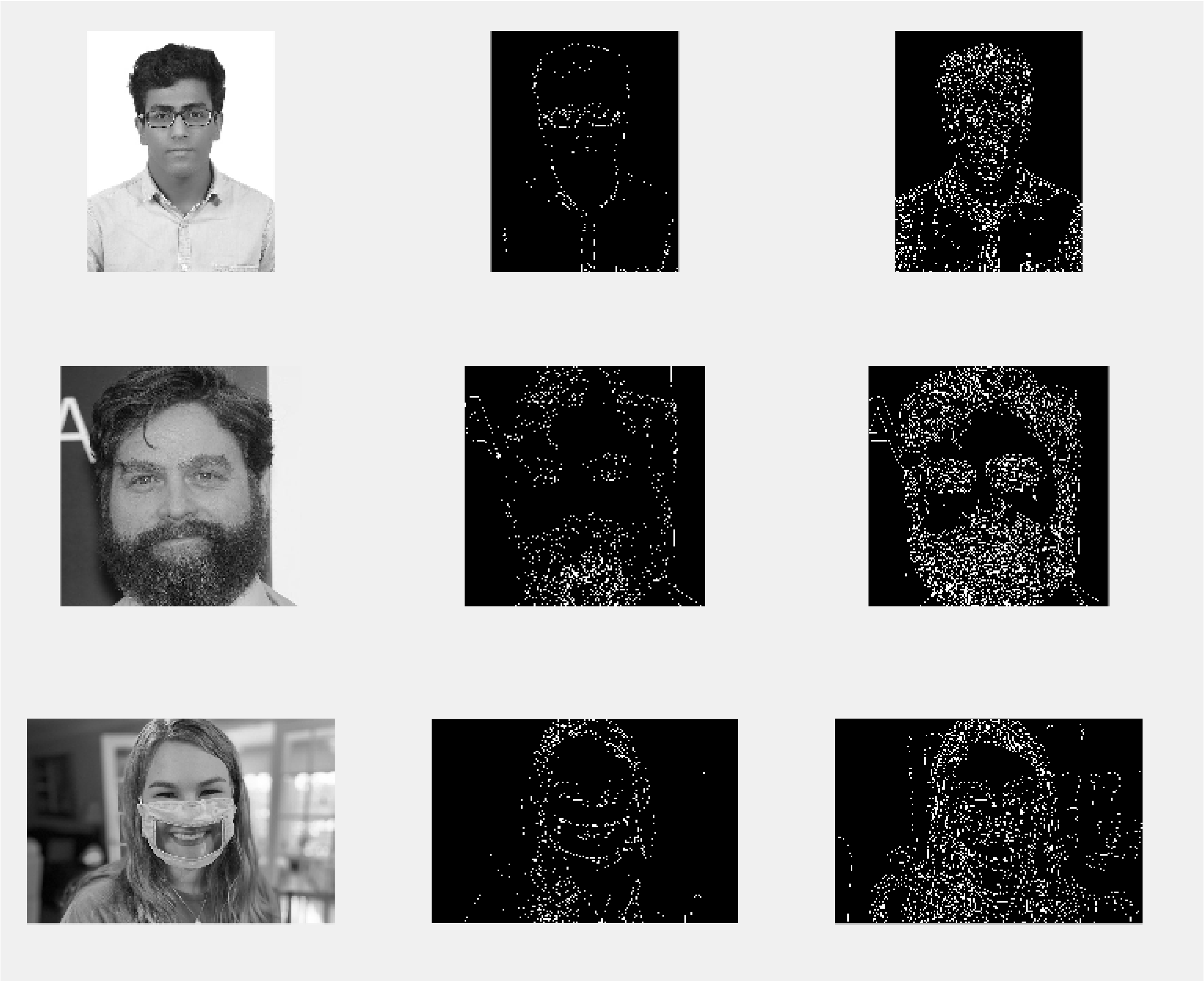
```
% images and grayscaling
I1 = rgb2gray(imread('mehul.jpeg'));
I2 = rgb2gray(imread('beard-face.jpg'));
I3 = rgb2gray(imread('transparent-face-mask.jpg'));

% filter
k = fspecial('laplacian');

% filtering
Ie1 = imfilter(double(I1), k, 'symmetric');
Ie2 = imfilter(double(I2), k, 'symmetric');
Ie3 = imfilter(double(I3), k, 'symmetric');

% displaying
subplot(2,3,1), imshow(I1);
subplot(2,3,2), imshow(I2);
subplot(2,3,3), imshow(I3);
subplot(2,3,4), imagesc(Ie1);
colormap('gray');
subplot(2,3,5), imagesc(Ie2);
colormap('gray');
subplot(2,3,6), imagesc(Ie3);
colormap('gray');
```



Code3

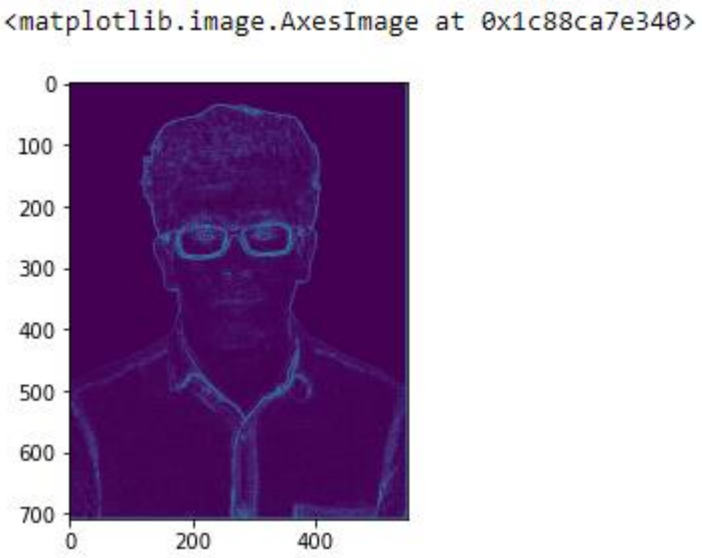
```
% images and grayscaling
I1 = rgb2gray(imread('mehul.jpeg'));
I2 = rgb2gray(imread('beard-face.jpg'));
I3 = rgb2gray(imread('transparent-face-mask.jpg'));
```

```
% image filtering
is1 = edge(I1, 'sobel');
ie1 = edge(I1, 'canny');
is2 = edge(I2, 'sobel');
ie2 = edge(I2, 'canny');
is3 = edge(I3, 'sobel');
ie3 = edge(I3, 'canny');

% image subplot
subplot(331), imshow(I1);
subplot(332), imshow(is1);
subplot(333), imshow(ie1);

subplot(334), imshow(I2);
subplot(335), imshow(is2);
subplot(336), imshow(ie2);

subplot(337), imshow(I3);
subplot(338), imshow(is3);
subplot(339), imshow(ie3);
```



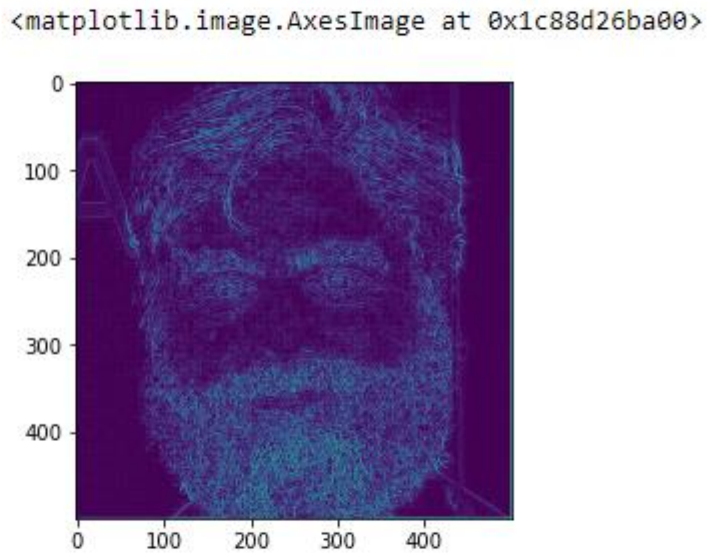
Python filter codes

```
from PIL import Image, ImageFilter

# Opening the image (R prefixed to string

# in order to deal with '\' in paths)

image1 = Image.open(r"mehul.jpeg")

image2 = Image.open(r"beard-face.jpg")

image3 = Image.open(r"transparent-face-mask.jpg")


# Converting the image to greyscale, as edge detection

# requires input image to be of mode = Greyscale (L)

image1 = image1.convert("L")

image2 = image2.convert("L")

image3 = image3.convert("L")


# Detecting Edges on the Image using the argument ImageFilter.FIND_EDGES

image1 = image1.filter(ImageFilter.FIND_EDGES)

image2 = image2.filter(ImageFilter.FIND_EDGES)

image3 = image3.filter(ImageFilter.FIND_EDGES)


from matplotlib import pyplot as plt

plt.imshow(image1)
```



```
<matplotlib.image.AxesImage at 0x1c88ca7e340>
```

```
plt.imshow(image2)
```



```
<matplotlib.image.AxesImage at 0x1c88d26ba00>
```

plt.imshow(image3)

After using these images in my model, model was giving right results for without mask cases but wrong for with mask.

Conclusion: The segmented images may give better results.

Citation:

Haralick, R.M. and Shapiro, L.G., 1985. Image segmentation techniques. *Computer vision, graphics, and image processing*, *29*(1), pp.100-132.