

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

Mounted at /content/drive

```
1 import numpy as np
2 import pandas as pd
3 import os
4
5 import tensorflow as tf
6
7 from tensorflow import keras
8 from tensorflow.keras import datasets, layers, models, callbacks
9 import matplotlib.pyplot as plt
10
11 import PIL
12 import PIL.Image
13 import glob
14 import skimage
15 from PIL import Image
16 from skimage import io, color
17
18 import numpy as np
19 import pandas as pd
20 import cv2
21 from keras.models import Sequential
22 from keras.layers import Convolution2D
23 from keras.layers import MaxPooling2D, AveragePooling2D
24 from keras.layers import Flatten, Dense, Dropout
25
26 from sklearn.model_selection import train_test_split
27
28 from tensorflow.keras.callbacks import ModelCheckpoint
29
30
```

```
1 classes = ['Centerstack', 'Forward', 'Left_wing_mirror', 'Rearview_mirror', 'Right_wing_mirror']
```

```
1 num_classes = len(classes)
2 print(num_classes)
```

5

```
1 # shows image size as well as images
2
3 train_root = '/content/drive/MyDrive/Dataset/gaze_dataset/train'
4
5
```

```
6 for i in classes:
7     path = os.path.join(train_root,i,'*')
8     path = glob.glob(path)
9     id = np.random.choice(10)
10    image = io.imread(path[id])
11    imgs = Image.open(path[id])
12    print(imgs.size)
13
14
15    fig, ax = plt.subplots()
16
17    ax.set_title(i)
18    fig.set_figwidth(10)
19    ax.imshow(image)
20
```



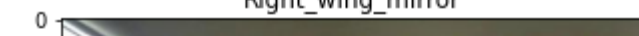
Left_wing_mirror



Rearview_mirror



Right_wing_mirror



Face Extraction



```
1 %cd "/content/drive/MyDrive/Dataset/gaze_dataset/train/face"
```

```
/content/drive/MyDrive/gaze_dataset/train/Face
```



```
1 for c in classes:
2   os.mkdir(c)
```



```

1 train_root = "/content/drive/MyDrive/Dataset/gaze_dataset/train"
2
3 for i in classes:
4     pt = os.path.join(train_root, 'Face', i)
5     os.chdir(pt)
6
7     path = os.path.join(train_root, i, '*')
8     path = glob.glob(path)
9
10
11 for ip in path:
12     fN = ip.split("/")[-1]
13
14     modelFile = "/content/drive/MyDrive/Dataset/gaze_dataset/weights.caffemodel"
15     configFile = "/content/drive/MyDrive/Dataset/gaze_dataset/deploy.prototxt.txt"
16     net = cv2.dnn.readNetFromCaffe(configFile, modelFile)
17     img = cv2.imread(ip)
18     h, w = img.shape[:2]
19     blob = cv2.dnn.blobFromImage(cv2.resize(img, (300, 300)), 1.0, (300, 300), (104.0, 117
20     net.setInput(blob)
21     faces = net.forward()
22     #to draw faces on image
23     for i in range(faces.shape[2]):
24         confidence = faces[0, 0, i, 2]
25         if confidence > 0.5:
26             box = faces[0, 0, i, 3:7] * np.array([w, h, w, h])
27             (x, y, x1, y1) = box.astype("int")
28             cv2.rectangle(img, (x, y), (x1, y1), (0, 0, 255), 2)
29             roi_color = img[y:y1, x:x1]
30
31     cv2.imwrite(fN+'_face.jpg', roi_color)
32     # image = cv2.imread(ip)
33     # gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
34
35     # faceCascade = cv2.CascadeClassifier(cv2.data.harcascades + "haarcascade_frontalface
36     # faces = faceCascade.detectMultiScale(
37     #     gray,
38     #     scaleFactor=1.3,
39     #     minNeighbors=3,
40     #     minSize=(30, 30)
41     # )
42
43     # for (x, y, w, h) in faces:
44     #     cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
45     #     roi_color = image[y:y + h, x:x + w]
46     # cv2.imwrite(fN+'_face.jpg', roi_color)
47

```

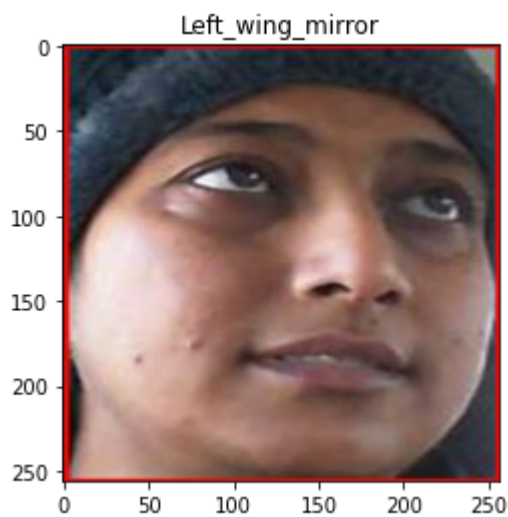
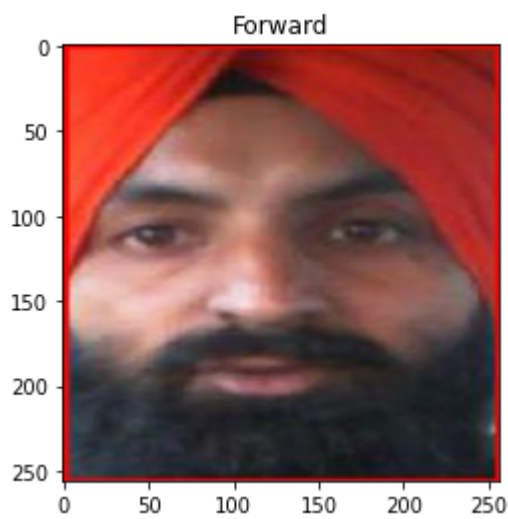
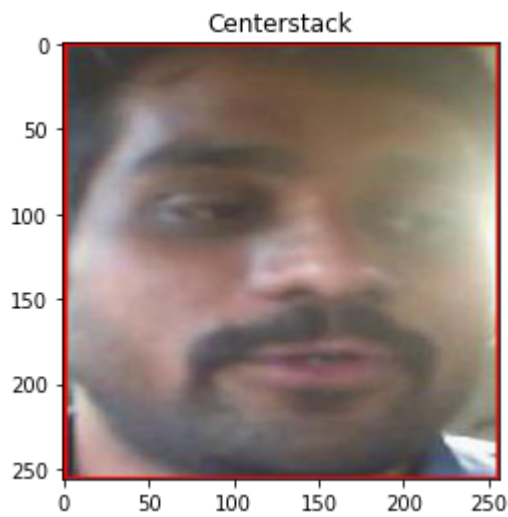
Resizing images

```
1 train_root = "/content/drive/MyDrive/gaze_dataset/train/face"
2
3 for i in classes:
4     path = os.path.join(train_root,i,'*')
5     path = glob.glob(path)
6     for p in path:
7         img = Image.open(p)
8         img = img.resize((256,256))
9         img.save(p)
10
```

```
1 # shows images of face extracted
2
3 train_root = '/content/drive/MyDrive/gaze_dataset/train/face'
4
5
6 for i in classes:
7     path = os.path.join(train_root,i,'*')
8     path = glob.glob(path)
9     id = np.random.choice(10)
10    image = io.imread(path[id])
11    imgs = Image.open(path[id])
12    print(imgs.size)
13
14
15    fig, ax = plt.subplots()
16
17    ax.set_title(i)
18    fig.set_figwidth(10)
19    ax.imshow(image)
```



(256, 256)
(256, 256)
(256, 256)
(256, 256)
(256, 256)





extracting faces for testing

```

1 path = '/content/drive/MyDrive/Dataset/gaze_dataset/test'
2
3 files = glob.glob("/content/drive/MyDrive/Dataset/gaze_dataset/test/*.jpg")
4
5 files.sort()

```

```

1 os.chdir("/content/drive/MyDrive/Dataset/gaze_dataset/face_test")
2
3 for ip in files:
4
5     fN = ip.split("/")[-1]
6
7     image = cv2.imread(ip)
8
9     modelFile = "/content/drive/MyDrive/Dataset/gaze_dataset/weights.caffemodel"
10    configFile = "/content/drive/MyDrive/Dataset/gaze_dataset/deploy.prototxt.txt"
11    net = cv2.dnn.readNetFromCaffe(configFile, modelFile)
12    img = cv2.imread(ip)
13    h, w = img.shape[:2]
14    blob = cv2.dnn.blobFromImage(cv2.resize(img, (300, 300)), 1.0,
15    (300, 300), (104.0, 117.0, 123.0))
16    net.setInput(blob)
17    faces = net.forward()
18    #to draw faces on image
19    for i in range(faces.shape[2]):
20        confidence = faces[0, 0, i, 2]
21        if confidence > 0.5:
22            box = faces[0, 0, i, 3:7] * np.array([w, h, w, h])
23            (x, y, x1, y1) = box.astype("int")
24            cv2.rectangle(img, (x, y), (x1, y1), (0, 0, 255), 2)
25            roi_color = img[y:y1, x:x1]
26
27    cv2.imwrite(fN, roi_color)
28    # gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
29
30    # faceCascade = cv2.CascadeClassifier(cv2.data.harcascades + "haarcascade_frontalface_d
31    # faces = faceCascade.detectMultiScale(
32    #     gray,
33    #     scaleFactor=1.3,
34    #     minNeighbors=3,
35    #     minSize=(30, 30)
36    # )

```