

# REPORT ON HUMAN IMAGE CLASSIFICATION PROJECT(B23).

PROJECT IS COMPLETED IN 3 PHASES:

## PHASE 1: (DOWNLOADING THE IMAGES.)

a) To download the image, me and my groupmates used a python scrapper which will download image from google images. It will take input keyword and number of images to download.as shown in image.

```
Enter your search keyword: human images
Enter the number of images you want: 100
Searching Images....
Found 80 images
Start downloading...
Download Completed!
```

b) The library's used are:

(1) os (2) requests (to send GET requests) (3) BeautifulSoup (to parse html)

To store the downloaded images, we used os library to save the images in the desired folder.

```
def main():
    if not os.path.exists(Image_Folder):
        os.mkdir(Image_Folder)
    download_images()
```

We downloaded 1700 images out of which 850 images are Indian images and rest are from Asian country people's images, some American and European.

Stored all the images which contains all the human images from all the continents we noticed that around 5-10% images in our directory are non-human images.so we made a code to classify human and non-human images.

We will see it in 2<sup>nd</sup> phase.

## PHASE 2:(CLASSIFY HUMANS AND NON-HUMANS.)

a) To classify humans and non-humans we used HaarCascade.it is a machine learning-based approach where a lot of positive and negative images are used to train the classifier. Positive images – These images contain the images which we want our

classifier to identify. Negative Images – Images of everything else, which do not contain the object we want to detect.

b) The library's used:

(1) cv2 (2) os (3) shutil

c) We used:

1) frontal face (haarcascade\_frontalface\_default.xml)

2) eye cascade (haarcascade\_eye.xml)

```
def get_cropped_image_if_2_eyes(image_path):
    img = cv2.imread(image_path)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(roi_gray)
        if len(eyes) >= 2:
            return roi_color

path_to_data = '.raw dataset'
path_to_cr_data = '.raw images'

import os
import shutil
if os.path.exists(path_to_cr_data):
    shutil.rmtree(path_to_cr_data)
os.mkdir(path_to_cr_data)

count = 0
for entry in os.scandir(path_to_data):
    print(entry.path)
    count += 1
    roi_color = get_cropped_image_if_2_eyes(entry.path)
    if roi_color is not None:
        cropped_file_name = str(count) + '.jpg'
        cropped_file_path = path_to_cr_data + '/' + cropped_file_name
        cv2.imwrite(cropped_file_path, roi_color)
```

It will detect if the image is of human or non-human using the cascade xml files. human images will be stored in another directory.

After this phase we got a directory which contain only human images which we used to train our model in the 3<sup>rd</sup> phase.

### PHASE 3:(CLASSIFICATION OF INDIAN HUMAN AND OTHER COUNTRY'S HUMAN.)

a) The directory which we created in 2<sup>nd</sup> phase; we used the images of the directory to train our model.

The algorithm used is: K-Means clustering algorithm of sklearn.

b) Library's used:

(1) cv (2) pandas (3) NumPy (4) train\_test\_split (5) KMeans cluster (6) os

c) steps followed for classification:

step 1: we converted all the images of the directory to NumPy array.

```

count += 1
roi_color = get_cropped_image_if_2_eyes(entry.path)
if roi_color is not None:
    np_im=np.array(roi_color)
    print(np_im)

```

Step 2: we stored all the NumPy array in a csv file.

```

#from numpy import savetxt
#savetxt("data1.csv", np_im.reshape((3, -1)),
DF = pd.DataFrame(np_im.reshape(3, -1))

# save the dataframe as a csv file
DF.to_csv("data4.csv")

```

```

df=pd.read_csv("data4.csv")
df

```

Step 3: we cleaned the data and filled the NaN values with the mean (df.fillna(df.mean())).

Step 4: we initialized the x, y with dtype int.

```

x=df.iloc[:,7224]
x=x.astype('int')
y=df.iloc[:,7224]
y=y.astype('int')

```

Step 5: with train\_test\_split we separated the train and test values with X, Y train and test.

```

from sklearn.model_selection import train_test_split

```

```

X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size=0.2, random_state=0)

```

Step 6: we fitted the data in our KMeans algorithm.

```

from sklearn.cluster import KMeans
km=KMeans(n_clusters=2)

```

```

y_pred=km.fit_predict(df)

```

```

y_pred

```

```

array([0, 0, 1])

```

Our model was ready, it was trained in such a way that if the first element of the array is 0 than it's an Indian human and if it's 1 than a non-Indian human. If it's non-human which we will be decided by HaarCascade than it will show a message not a human.

It was time to test the model. we downloaded image of two Indian and two non-Indian people are the result is shown in the screenshot.

[illegible]

(For PM narendra modi)

[illegible]

(FOR Barack Obama)

SUBMITTED BY:

NAME: AAKASH SHARMA

GROUP: B23

MAIL: [coolaakashsharma@gmail.com](mailto:coolaakashsharma@gmail.com)

