

```
In [ ]:
```

```
import cv2
import numpy as np
import os
```

```
In [ ]:
```

```
def dist(x1,x2):
    return np.sqrt(sum((x1-x2)**2))

def KNN(X,Y,query_point,K=5):
    vals = []
    m = X.shape[0]

    for i in range(m):
        d = dist(query_point,X[i])
        vals.append((d,Y[i]))

    vals = sorted(vals)
    # Nearest/First K points
    vals = vals[:K]

    vals = np.array(vals)
    #print("Vals: ",vals)
    new_vals = np.unique(vals[:,1],return_counts=True)
    #print(new_vals)

    index = new_vals[1].argmax()
    pred = new_vals[0][index]

    return pred
```

```
In [ ]:

# Init Camera
cap = cv2.VideoCapture(0)

# Face Detection
face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_alt.xml")
dataset_path = './Data/'
skip = 0
face_data = [] # Load all the files, x for data
labels = [] # y for data
class_id = 0 # Labels for the given file
names = {} # mapping btw id-name

# Data Preparation
for fx in os.listdir(dataset_path):
    if fx.endswith('.npy'):
        # Create mapping between class id and name
        names[class_id] = fx[:-4]
        print('Loaded '+fx)
        data_item = np.load(dataset_path + fx)
        face_data.append(data_item)

        # Create Labels for the class
        target = class_id*np.ones((data_item.shape[0],))
        class_id += 1
        labels.append(target)

face_dataset = np.concatenate(face_data,axis=0)
face_labels = np.concatenate(labels,axis=0)

print(face_dataset.shape)
print(face_labels.shape)
```

```
In [ ]:
```

```
# Testing
```

```
while True:
```

```
    ret, frame = cap.read()
```

```
    if ret == False:
```

```
        continue
```

```
    faces = face_cascade.detectMultiScale(frame,1.3,5)
```

```
    for x,y,w,h in faces:
```

```
        offset=10
```

```
        face_section = frame[y-offset:y+h+offset,x-offset:x+w+offset]
```

```
        face_section = cv2.resize(face_section,(100,100))
```

```
        # Predicted Label
```

```
        out = KNN(face_dataset, face_labels, face_section.flatten())
```

```
        # Display on the screen the name and rectangle around it
```

```
        pred_name = names[int(out)]
```

```
        cv2.putText(frame,pred_name,(x,y-10),cv2.FONT_HERSHEY_SIMPLEX,1,(255,0,0),2,cv2
```

```
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
```

```
cv2.imshow("Faces",frame)
```

```
    # Exiting Code
```

```
    key_pressed = cv2.waitKey(1) & 0xFF
```

```
    x = cv2.waitKey(1)
```

```
    if key_pressed == ord('q'):
```

```
        break
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

```
In [ ]:
```

