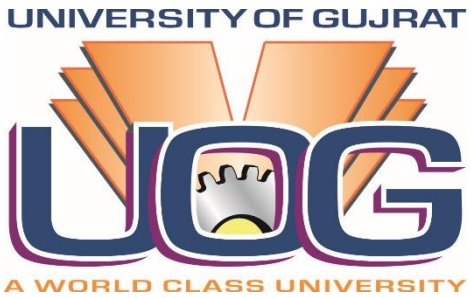

Artificial Intelligence

Course Code: CS-329

Lab-Task #4

Submitted to: Mr. Naveed Anwar Butt

Submitted by: Abdul Rehman (20021519-049)

Submission Date: 02 January, 2023

BS_CS 5th B

Human Face Detection System

1. **Determine the critical data values you need to train/test your model.**

   To determine the critical data values for training and testing a human face detection system, you should consider the *types of faces* that the system should be able to detect, *the size and resolution of the faces in the data, the pose and orientation of the faces, the lighting and background conditions, and the number and diversity of the faces in the data*.

2. **Collect the data you need, then sort and organize it.**

```
import pandas as pd
import numpy as np
from scipy import stats
from scipy.stats import zscore
import matplotlib.pyplot as plt

# Load the dataset into a Pandas dataframe
df = pd.read_csv("data.csv")


# Sort the data by the label column
df = df.sort_values(by="label")


# Group the data by the label column
df_grouped = df.groupby("label")
```

3. **Identify duplicate or irrelevant values and remove them.**

```
# Identify and remove duplicates
df_deduped = df.drop_duplicates()
```

4. **Search for missing values and fill them in or totally nullify them, so you have a complete dataset.**

```
# Search for missing values
missing_values = df.isnull().sum()

# Fill in missing values with a certain value
df_filled = df.fillna(value=0)

# Nullify rows with missing values
df_no_missing = df.dropna()
```

5. **Identify outliers and remove them, so they will not interfere in training/testing.**

```
#define a function called "outliers" which returns a list of indexes of all the
outliers using interquartile method

def outliers(dataframe, col):
    Q1 = dataframe[col].quantile(0.25)
    Q3 = dataframe[col].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    ls = dataframe.index[ (dataframe[col] < lower_bound) | (dataframe[col] >
upper_bound) ]

    return ls

# removing outliers using Z-Score Method

threshold = 3

# Calculate z-scores of each column
z_scores = df.apply(zscore)

# Identify which rows have outliers
outliers2 = (z_scores < -threshold) | (z_scores > threshold)

# Drop rows with outliers
df_clean = df.drop(df[outliers2].index)
```

6. **Validate your dataset to ensure it is ready for your model.**

```
# check for errors or inconsistencies

    assert df_cleaned["label"].isnull().sum() == 0

    # Check for missing values
    assert df_cleaned["label"].notnull().all()

    # Check the quality and consistency of the data using statistical techniques
    assert df_cleaned["label"].mean() > 0
```