**Lab-7**

Chandrakiran kopparapu

BL.EN. U4AIE23141

Code:

import os

import numpy as np

import cv2

from collections import Counter

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import OneClassSVM

# Path to dataset folder containing glioma images

DATASET\_PATH ="C:\\AIO\\Semster Files\\SEMSTER - 4\\ML\\Lab Work\\ML\_Assignment\_07\_BL.EN.U4AIE23138\\Dataset"

# Load images from the dataset

def load\_images\_from\_folder(folder):

    images = []

    for filename in os.listdir(folder):

        img\_path = os.path.join(folder, filename)

        img = cv2.imread(img\_path)  # Read image

        if img is not None:

            img = cv2.resize(img, (128, 128))  # Resize all images to (128x128)

            images.append(img)

    return np.array(images)

# Load the glioma images

X = load\_images\_from\_folder(DATASET\_PATH)

y = np.zeros(len(X), dtype=int)  # Assign all labels as 0 (glioma class)

# Check class distribution

print("Class distribution before split:", Counter(y))

# Handle the case where only one class is present

unique\_classes = np.unique(y)

if len(unique\_classes) < 2:

    print("Only one class detected. Proceeding with one-class classification.")

# Split the dataset (without stratification)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print("Training set size:", X\_train.shape)

print("Test set size:", X\_test.shape)

# Train a One-Class SVM model for anomaly detection

model = OneClassSVM(gamma='auto').fit(X\_train.reshape(len(X\_train), -1))

# Predict on test data

predictions = model.predict(X\_test.reshape(len(X\_test), -1))

# Output results

print("Predictions:", predictions)

print("Note: In One-Class SVM, +1 means 'normal' (glioma), -1 means 'outliers' (anomalies).")