**EXPT NO: 6 A python program to do face recognition using**

**DATE: 27.09.2024 SVM Classifier**

**AIM:**

To write a python program to implement face recognition using the SVM

Classifier

**PROCEDURE:**

Implementing face recognition using the SVM Classifier using the cat and dog dataset involve the following steps:

**Step 1: Import Necessary Libraries**

First, import the libraries that are essential for data manipulation, visualization, and model building.

|  |
| --- |
| import pandas as pd |
| import imageio |
| import os |
| from skimage.transform import resize |
| from skimage.io import imread |
| import numpy as np |
| import matplotlib.pyplot as plt |
| from sklearn import svm |
| from sklearn.model\_selection import GridSearchCV |
| from sklearn.model\_selection import train\_test\_split |
| from sklearn.metrics import accuracy\_score |
| from sklearn.metrics import classification\_report |

**Step 2: Load theDog and cat Dataset**

The dog and cat dataset can be loaded.

Categories=['cats','dogs']

|  |
| --- |
| flat\_data\_arr=[] #input array |
| target\_arr=[] #output array |
| datadir='/content/images' |
| #path which contains all the categories of images |
| for i in Categories: |
|  |
| print(f'loading... category : {i}') |
| path=os.path.join(datadir,i) |
| for img in os.listdir(path): |
| img\_array=imread(os.path.join(path,img)) |
| img\_resized=resize(img\_array,(150,150,3)) |
| flat\_data\_arr.append(img\_resized.flatten()) |
| target\_arr.append(Categories.index(i)) |
| print(f'loaded category:{i} successfully') |
| flat\_data=np.array(flat\_data\_arr) |
| target=np.array(target\_arr) |
| #dataframe |
| df=pd.DataFrame(flat\_data) |
| df['Target']=target |
| df.shape |

**OUTPUT :**



**Step 3: Separate input features and targets.**

|  |
| --- |
| #input data |
| x=df.iloc[:,:-1] |
| #output data |
| y=df.iloc[:,-1] |

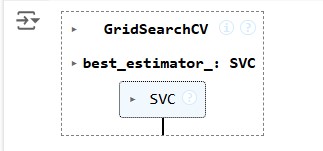
**Step 4 : Separate the input features and target**

|  |
| --- |
| **# Splitting the data into training and testing sets** |
| **x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.20,** |
| **random\_state=77, stratify=y)** |

**Step 5 : Build and train the model**

|  |
| --- |
| **# Defining the parameters grid for GridSearchCV** |
| **param\_grid={'C':[0.1,1,10,100],** |
| **'gamma':[0.0001,0.001,0.1,1],** |
| **'kernel':['rbf','poly']}** |
|  |
| **# Creating a support vector classifier** |
| **svc=svm.SVC(probability=True)** |
|  |
| **# Creating a model using GridSearchCV with the parameters grid** |
| **model=GridSearchCV(svc,param\_grid)** |
|  |
| **# Training the model using the training data** |
| **model.fit(x\_train,y\_train)** |

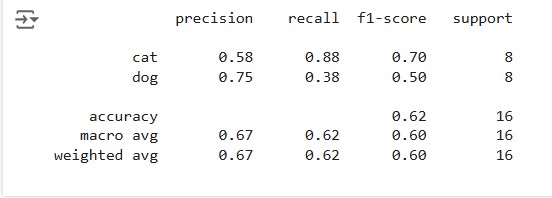
**OUTPUT :**



**Step 6 : Model evaluation**

|  |
| --- |
| **# Testing the model using the testing data** |
| **y\_pred = model.predict(x\_test)** |
|  |
| **# Calculating the accuracy of the model** |
| **accuracy = accuracy\_score(y\_pred, y\_test)** |
|  |
| **# Print the accuracy of the model** |
| **print(f"The model is {accuracy\*100}% accurate")** |
|  |
| **print(classification\_report(y\_test, y\_pred, target\_names=['cat', 'dog']))** |
|  |

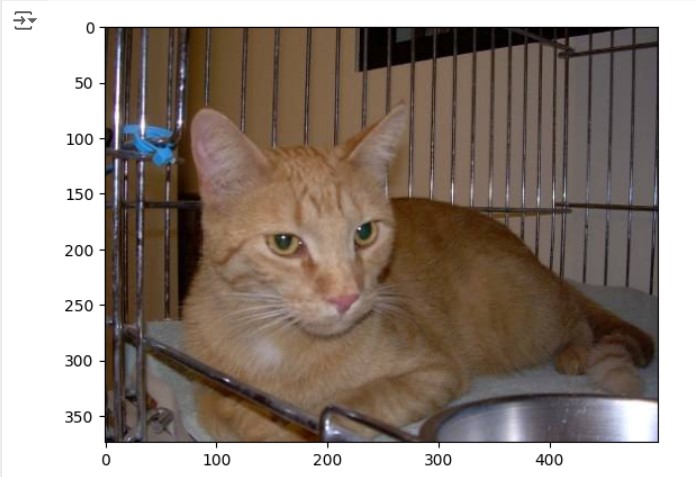
**OUTPUT :**



**Step 7 : Prediction**

|  |
| --- |
| **path='/content/cat.83.jpg'** |
| **img=imread(path)** |
| **plt.imshow(img)** |
| **plt.show()** |
| **img\_resize=resize(img,(150,150,3))** |
| **l=[img\_resize.flatten()]** |
| **probability=model.predict\_proba(l)** |
| **for ind,val in enumerate(Categories):** |
| **print(f'{val} = {probability[0][ind]\*100}%')** |
| **print("The predicted image is : "+Categories[model.predict(l)[0]])** |

**OUTPUT :**



**cats = 52.70216647851706% dogs = 47.29783352148294%**

**The predicted image is : cat**

**RESULT :**

Thus the process helps us to implement the face recognition using SVM Classifier using python program.