**CODE**

**PROGRAM CODE:**

from tqdm import tqdm

#https://www.microsoft.com/en-us/download/confirmation.aspx?id=54765

import numpy as np

import matplotlib.pyplot as plt

import os

import cv2 #to convert image to number

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

from sklearn.svm import SVC

from tqdm import tqdm

from sklearn.preprocessing import StandardScaler #change scale of image

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

import pickle

#Give the full location of the folder containing different folders of images to be classified

#Example: The Animals folder Contains Cat and Dog folders containg their respective images

DATADIR = r"C:\Users\Ananyaa M\Desktop\PROJECT\nature" #directory where everything is stored

#Enter the folder names containing the images to be classified in the list

CATEGORIES = ["buildings", "forest","glacier","mountain","sea","street"] #folder names to be trained

with open('train\_labels.pkl', 'wb') as f:

pickle.dump(CATEGORIES, f)

with open('train\_labels.pkl', 'rb') as f:

CATEGORIES = pickle.load(f)

#Enter the image dimensions to be processed

IMG\_SIZE = 100 #rezized to 100\*100

X = [] #list-data

y = [] #list-names

#Read and load the image as ML processable array in X-data and y-labels

for category in tqdm(CATEGORIES):

path = os.path.join(DATADIR,category)

for img in tqdm(os.listdir(path)):

img\_array = cv2.imread(os.path.join(path,img) ,cv2.IMREAD\_GRAYSCALE) #converted to black n white to reduce computation

img\_array = cv2.resize(img\_array, (IMG\_SIZE,IMG\_SIZE))

X.append(np.array(img\_array))

y.append(CATEGORIES.index(category))

X = np.array(X)

y = np.array(y)

X = X.reshape(len(X),-1) #len(shape[0])-60000,28,28

#Preprocessing the data to a particular scale. Eg: Values from [0 to 255] will become values to [0 to 1]

scaler = StandardScaler()

scaler.fit(X)

X = scaler.transform(X)

#Training and testing split

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

#Support Vector Machine Classifier ML Algorithm

cls = SVC(kernel='linear')

cls.fit(X\_train,y\_train)

print("SVM",cls.score(X\_test,y\_test))

#Decision Classifier ML Algorithm

tree = DecisionTreeClassifier(criterion = 'entropy')

tree.fit(X, y)

print("Decision Tree",tree.score(X\_test,y\_test))

#Random Forest Classifier ML Algorithm

model = RandomForestClassifier(n\_estimators=100,

bootstrap = True,

max\_features = 'sqrt')

model = model.fit(X, y)

#Save Classifier file

pickle\_out = open("RFC\_trained.pickle","wb")

pickle.dump(model,pickle\_out)

#Load Classifier File

pickle\_in = open("RFC\_trained.pickle","rb")

model = pickle.load(pickle\_in)

print("Random Forest", model.score(X\_test,y\_test))

for i in range(6): #randomly pics six images and gives the name of the image

ran = np.random.randint(0,200)

plt.imshow(X\_test[ran].reshape(IMG\_SIZE,IMG\_SIZE),cmap='gray') #cmap-colour mapping

plt.title(CATEGORIES[int(model.predict(X\_test[ran].reshape(1,-1)).astype(int))])

plt.show()

**TRAINING:**

from tqdm import tqdm

import numpy as np

import matplotlib.pyplot as plt

import os

import cv2

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

from tqdm import tqdm

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

import pickle

#Give the full location of the folder containing different folders of images to be classified

#Example: The Animals folder Contains Cat and Dog folders containg their respective images

DATADIR = r"C:\Users\Ananyaa M\Desktop\PROJECT\nature"

#Enter the folder names containing the images to be classified in the list

CATEGORIES = ["buildings", "forest","glacier","mountain","sea","street"]

with open('train\_labels.pkl', 'wb') as f:

pickle.dump(CATEGORIES, f)

#Enter the image dimensions to be processed

IMG\_SIZE = 100

X = []

y = []

#Read and load the image as ML processable array in X-data and y-labels

for category in tqdm(CATEGORIES):

path = os.path.join(DATADIR,category)

for img in tqdm(os.listdir(path)):

img\_array = cv2.imread(os.path.join(path,img) ,cv2.IMREAD\_GRAYSCALE)

img\_array = cv2.resize(img\_array, (IMG\_SIZE,IMG\_SIZE))

X.append(np.array(img\_array))

y.append(CATEGORIES.index(category))

X = np.array(X)

y = np.array(y)

X = X.reshape(len(X),-1)

#Preprocessing the data to a particular scale. Eg: Values from [0 to 255] will become values to [0 to 1]

scaler = StandardScaler()

scaler.fit(X)

X = scaler.transform(X)

#Training and testing split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.15)

#Random Forest Classifier ML Algorithm

model = RandomForestClassifier(n\_estimators=100,

bootstrap = True,

max\_features = 'sqrt')

model = model.fit(X, y)

#Save Classifier file

pickle\_out = open("RFC\_trained.pickle","wb")

pickle.dump(model,pickle\_out)

#Load Classifier File

pickle\_in = open("RFC\_trained.pickle","rb")

model = pickle.load(pickle\_in)

print("Random Forest", model.score(X\_test,y\_test))

**TESTING:**

from tqdm import tqdm

import numpy as np

import matplotlib.pyplot as plt

import os

import cv2

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

from tqdm import tqdm

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

import pickle

#Give the full location of the folder containing different folders of images to be classified

#Example: The Animals folder Contains Cat and Dog folders containg their respective images

DATADIR = r"C:\Users\Ananyaa M\Desktop\PROJECT\nature"

#Loading Labels

with open('train\_labels.pkl', 'rb') as f:

CATEGORIES = pickle.load(f)

TEST\_FOLDER\_NAME = ["test"]

#Enter the image dimensions to be processed

IMG\_SIZE = 100

X = []

#Read and load the test images

for category in tqdm(TEST\_FOLDER\_NAME):

path = os.path.join(DATADIR,category)

for img in tqdm(os.listdir(path)):

img\_array = cv2.imread(os.path.join(path,img) ,cv2.IMREAD\_GRAYSCALE)

img\_array = cv2.resize(img\_array, (IMG\_SIZE,IMG\_SIZE))

X.append(np.array(img\_array))

X = np.array(X)

X = X.reshape(len(X),-1)

#Preprocessing the data to a particular scale. Eg: Values from [0 to 255] will become values to [0 to 1]

scaler = StandardScaler()

scaler.fit(X)

X = scaler.transform(X)

#Load Classifier File

pickle\_in = open("RFC\_trained.pickle","rb")

model = pickle.load(pickle\_in)

#Display classified images with their prediction

for i in range(X.shape[0]):

ran = i

plt.imshow(X[ran].reshape(IMG\_SIZE,IMG\_SIZE),cmap='gray')

plt.title(CATEGORIES[int(model.predict(X[ran].reshape(1,-1)).astype(int))])

plt.show()

**LINK:**

<https://www.kaggle.com/puneet6060/intel-image-classification/data>