MAJOR PROJECT

RAVITEJA.K

ravitejakondagorla@gmail.com

Machine Learning September_2023 Batch

```
#===IMPORTING  LIBRARIES===
import numpy as np
import randomimport matplotlib.pyplot as plt
from tensorflow.keras.models import Sequentialfrom tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten
#===LOAD DATASET===
X_train = np.loadtxt("C:\\Users\\tejon\\Downloads\\input.csv", delimiter = ',')
Y_train = np.loadtxt("C:\\Users\\tejon\\Downloads\\labels.csv", delimiter = ',')
X_test = np.loadtxt("C:\\Users\\tejon\\Downloads\\input_test.csv", delimiter = ',')
Y_test = np.loadtxt("C:\\Users\\tejon\\Downloads\\labels_test.csv", delimiter = ',')X_train = X_train.reshape(len(X_train), 100,
100, 3)
Y_train = Y_train.reshape(len(Y_train), 1)
X_test = X_test.reshape(len(X_test), 100, 100, 3)
Y_test = Y_test.reshape(len(Y_test), 1)
X_train = X_train/255.0
X_test = X_test/255.0

print("Shape of X_train: ", X_train.shape)
print("Shape of Y_train: ", Y_train.shape)
print("Shape of X_test: ", X_test.shape)
print("Shape of Y_test: ", Y_test.shape)

idx = random.randint(0, len(X_train))
plt.imshow(X_train[idx, :])
plt.show()
```

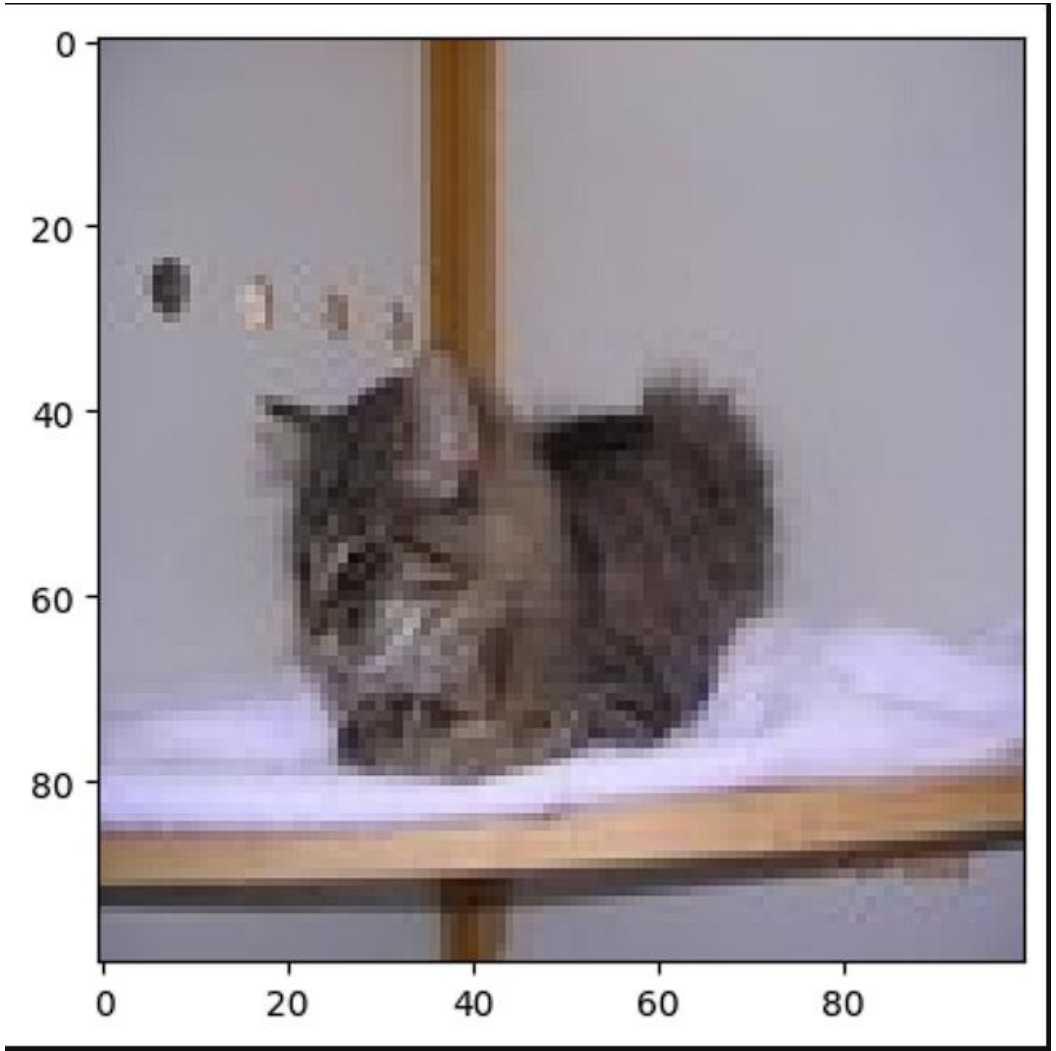
```
#===MODEL===
model =Sequential([    Conv2D(32, (3,3), activation = 'relu', input_shape = (100, 100, 3)),
MaxPooling2D((2,2)),
Conv2D(32, (3,3), activation = 'relu'),
MaxPooling2D((2,2)),
Flatten(),
Dense(64, activation = 'relu'),
Dense(1, activation = 'sigmoid')])

model = Sequential()
model.add(Conv2D(32, (3,3), activation = 'relu', input_shape = (100, 100, 3)))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(32, (3,3), activation = 'relu'))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense(64, activation = 'relu'))
model.add(Dense(1, activation = 'sigmoid'))
model.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
model.fit(X_train, Y_train, epochs = 5, batch_size = 64,validation_data=(X_test,Y_test))
model.evaluate(X_test, Y_test)
```

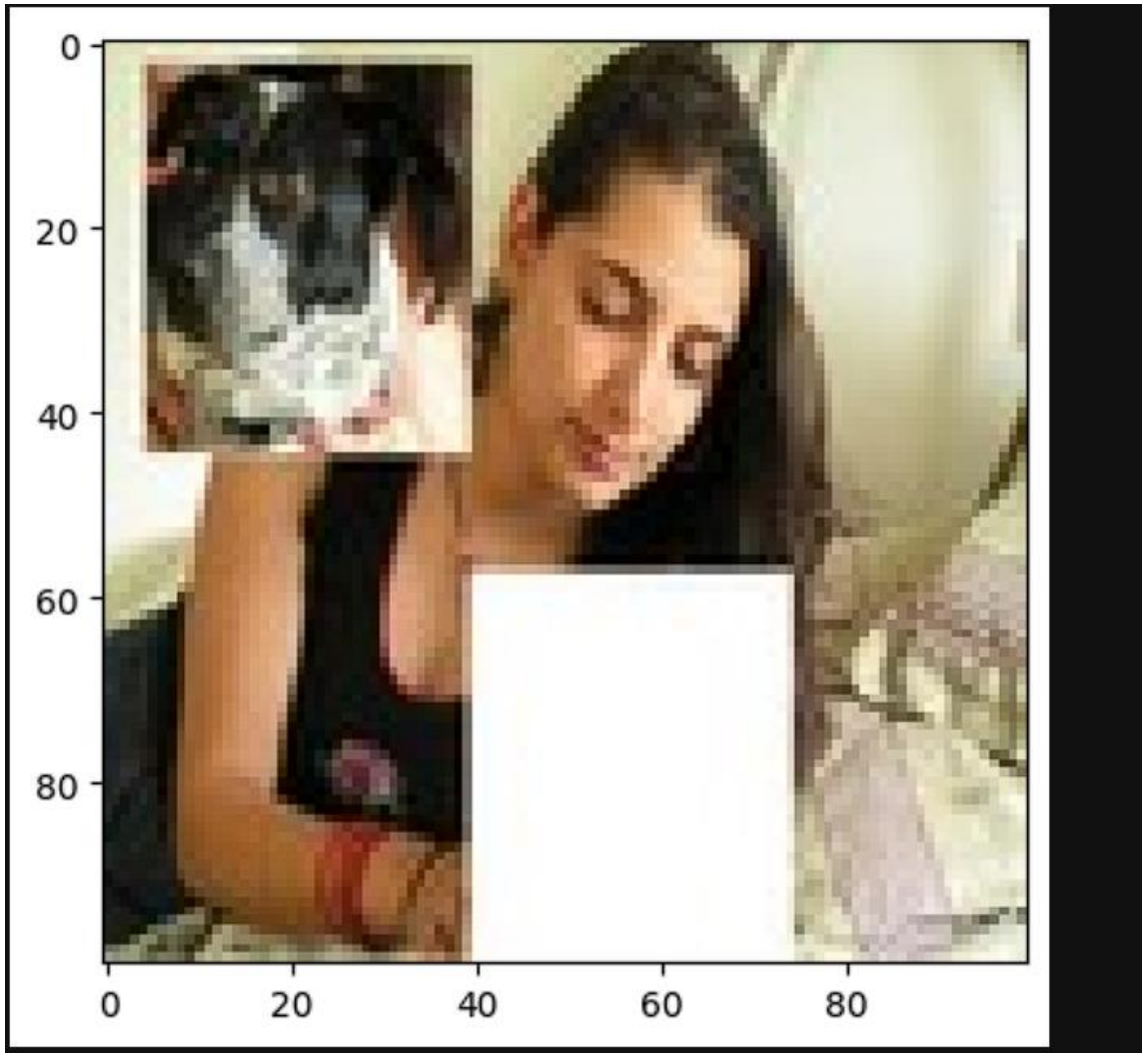
```
#===MAKING PREDICTIONS===

idx2 = random.randint(0, len(Y_test))
plt.imshow(X_test[idx2, :])
plt.show()
y_pred = model.predict(X_test[idx2, :].reshape(1, 100, 100, 3))
print(y_pred)
y_pred = y_pred < 0.5
if y_pred :
        pred = 'dog’
else:
        pred = 'cat'
print("Our model says it is a :", pred)
```

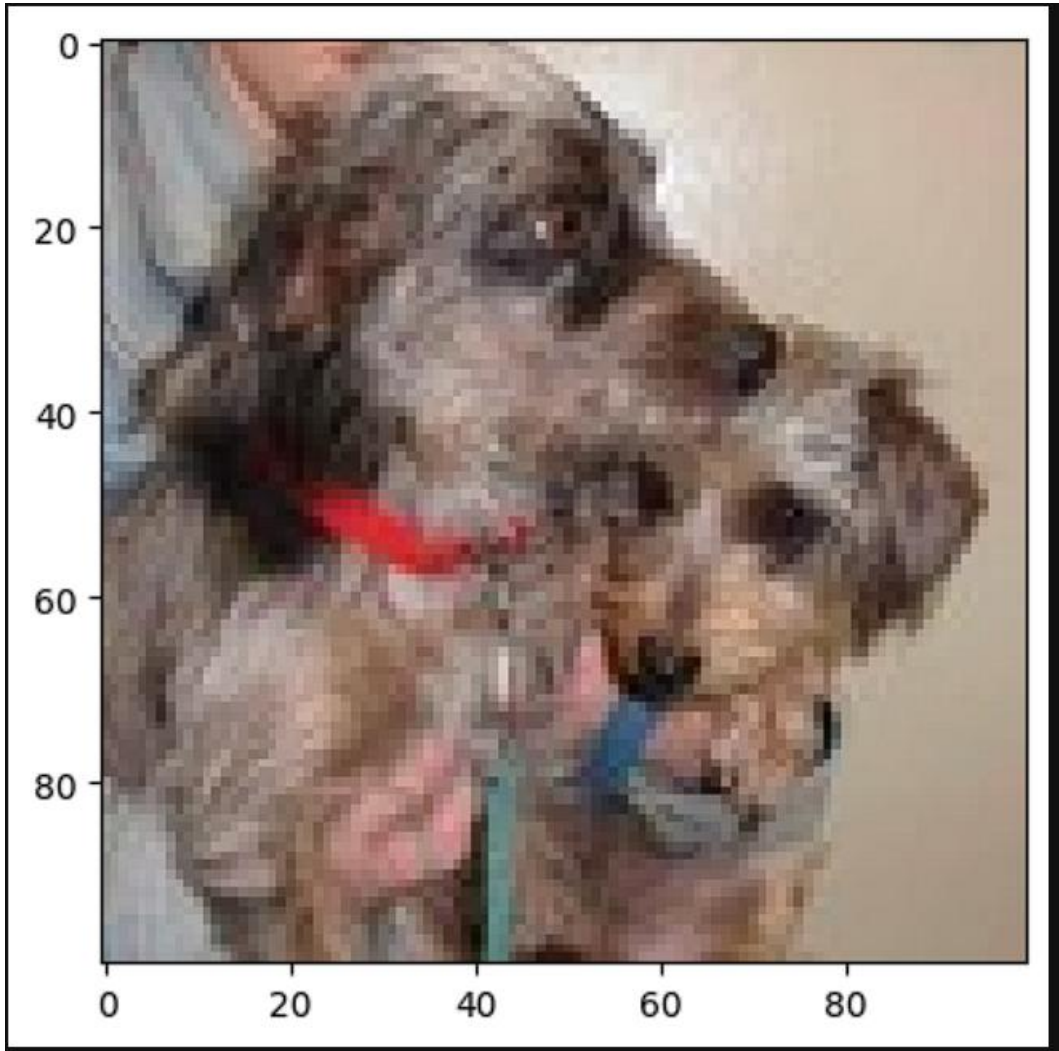
#==DATASET==
https://drive.google.com/drive/u/0/folders/1dZvL1gi5QLwOGrfdn9XEsi4EnXx535bD



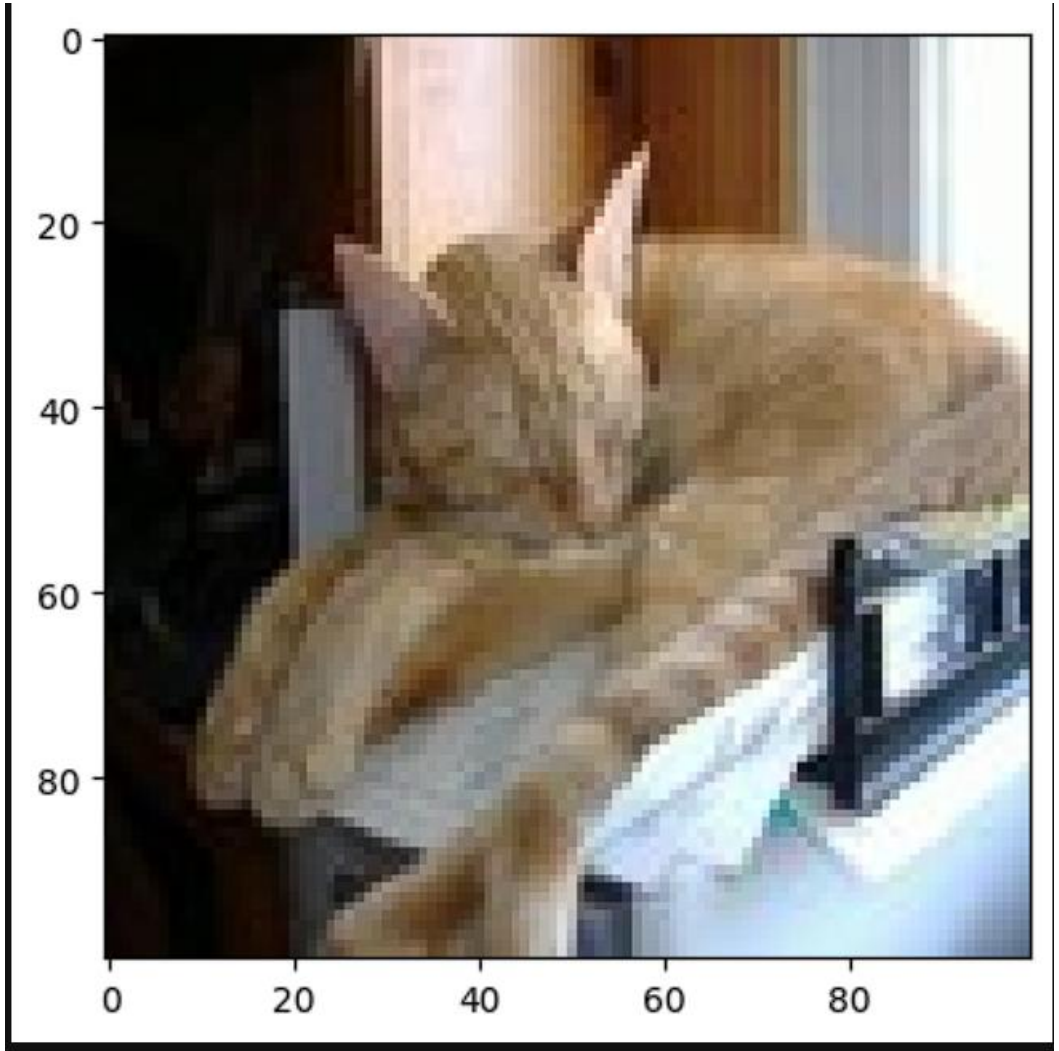
```
1/1 [==============================] - 0s 35ms/step
[[0.80278057]]
Our model says it is a : cat
```



```
1/1 [==============================] - 0s 16ms/step
[[0.26688728]]
Our model says it is a : dog
```




```
1/1 [==============================] - 0s 18ms/step
[[0.24166147]]
Our model says it is a : dog
```




```
1/1 [==============================] - 0s 51ms/step
[[0.91859144]]
Our model says it is a : cat
```