**##### download the dogs and cats dataset and save it to a local directory called data**

!mkdir data && wget http://files.fast.ai/data/dogscats.zip && unzip dogscats.zip -d data/

**#### Lets intialise some list variables**

data = [ ]

labels = [ ]

dataset\_path = 'data/dogscats/train‘

**##### figure out the label of the image from its path**

for (i, imagePath) in enumerate(imagePaths):

image = cv2.imread(imagePath)

**#### Resize the Image and then use the raw pixel values as features**

features = cv2.resize(image, (32, 32))

**#### Create a train test split**

(trainData, testData, trainLabels, testLabels) = train\_test\_split(data, labels, test\_size=0.25, random\_state=42)

**##### input image dimensions**

input\_shape = data[0].shape

model = Sequential()

**###code for convolution layer and Relu**

model.add(Conv2D(32, kernel\_size=(3, 3), strides=(1, 1),

activation='relu', input\_shape=input\_shape))

**###code for Max pooling layer**

model.add(MaxPooling2D(pool\_size=(2, 2), strides=(2, 2)))

model.add(Conv2D(64, (5, 5), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

***##### Reshape the test image shape to match the network thats expecting a set of images***

test\_feature=test\_feature.reshape(1,test\_feature.shape[0],test\_feature.shape[1],test\_feature.shape[2])

print(test\_feature.shape)

***##### Print Predictions***

test\_prediction = model.predict(test\_feature)

print("[INFO]CAT={:.10f}%,DOG:{:.10f}%".format(test\_prediction[0][0]\*100, test\_prediction[0][1]\*100))