Cats vs Dogs classification is a fundamental Deep Learning project for beginners. If you want to start your Deep Learning Journey with Python Keras, you must work on this elementary project.

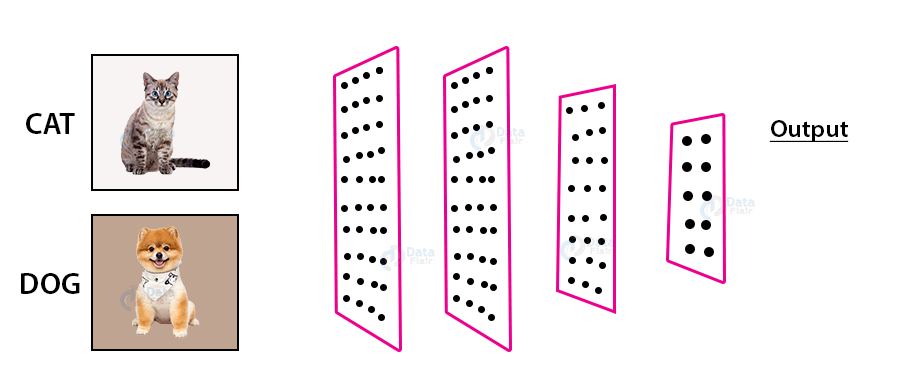
In this Keras project, we will discover how to build and train a convolution neural network for classifying images of Cats and Dogs.

The Asirra (Dogs VS Cats) dataset:

The Asirra (animal species image recognition for restricting access) dataset was introduced in 2013 for a machine learning competition. The dataset includes 25,000 images with equal numbers of labels for cats and dogs.

**Dataset:** [Cats and Dogs dataset](https://www.kaggle.com/c/dogs-vs-cats/data)

Deep Learning Project for Beginners – Cats and Dogs Classification

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2020/05/Cats-Dogs-Classification-deep-learning.gif)

Steps to build Cats vs Dogs classifier:

1. Import the libraries:

1. import numpy as np
2. import pandas as pd
3. from keras.preprocessing.image import ImageDataGenerator,load\_img
4. from keras.utils import to\_categorical
5. from sklearn.model\_selection import train\_test\_split
6. import matplotlib.pyplot as plt
7. import random
8. import os

2. Define image properties:

1. Image\_Width=128
2. Image\_Height=128
3. Image\_Size=(Image\_Width,Image\_Height)
4. Image\_Channels=3

3. Prepare dataset for training model:

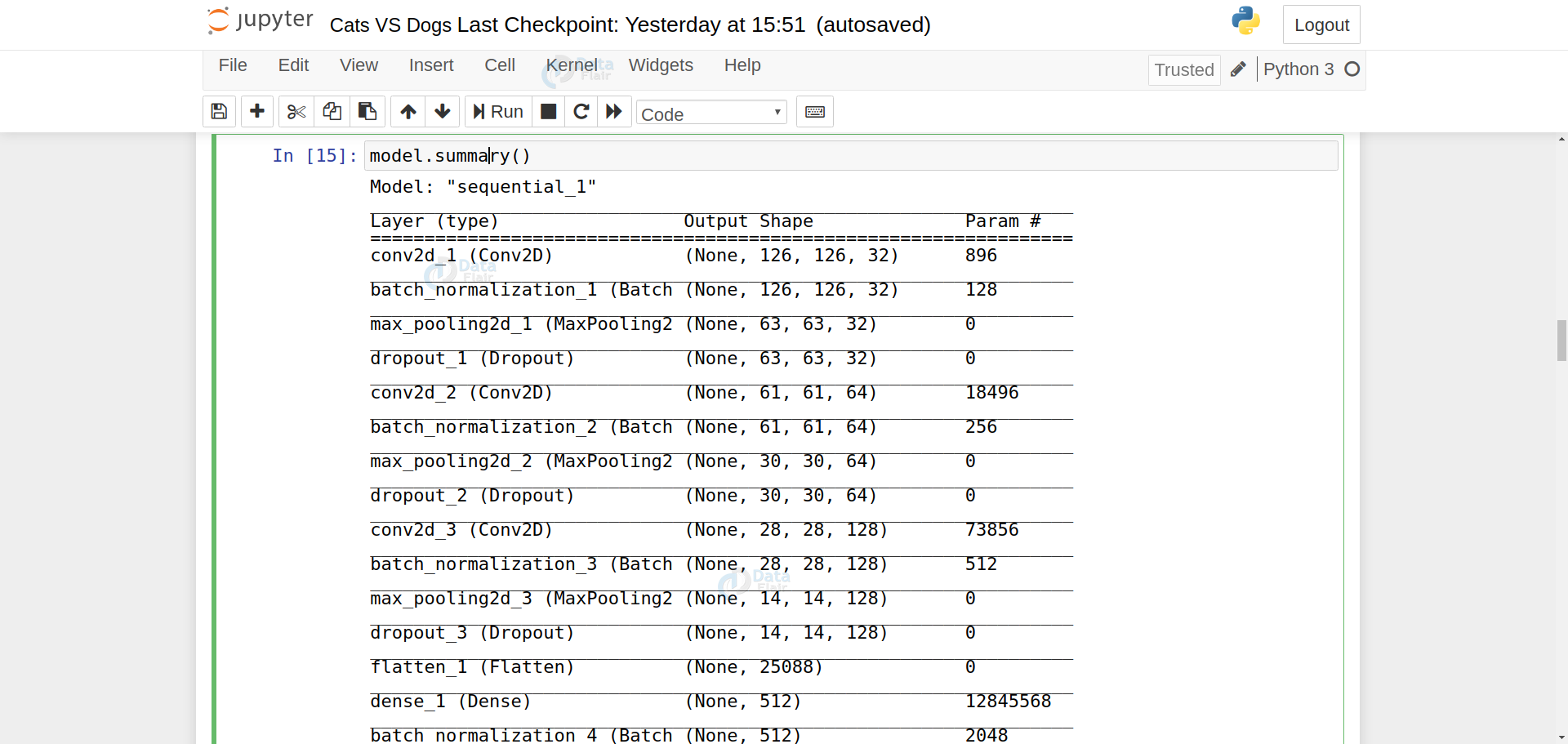
1. filenames=os.**listdir**("../documents/input/train")
2. categories=[]
3. for f\_name in filenames:
4. category=f\_name.**split**('.')[0]
5. if category=='dog':
6. categories.**append**(1)
7. else:
8. categories.**append**(0)
9. df=pd.**DataFrame**({
10. 'filename':filenames,
11. 'category':categories
12. })

4. Create the neural net model:

1. from keras.models import Sequential
2. from keras.layers import Conv2D,MaxPooling2D,\
3. Dropout,Flatten,Dense,Activation,\
4. BatchNormalization
5. model=**Sequential**()
6. model.**add**(**Conv2D**(32,(3,3),activation='relu',input\_shape=(Image\_Width,Image\_Height,Image\_Channels)))
7. model.**add**(**BatchNormalization**())
8. model.**add**(**MaxPooling2D**(pool\_size=(2,2)))
9. model.**add**(**Dropout**(0.25))
10. model.**add**(**Conv2D**(64,(3,3),activation='relu'))
11. model.**add**(**BatchNormalization**())
12. model.**add**(**MaxPooling2D**(pool\_size=(2,2)))
13. model.**add**(**Dropout**(0.25))
14. model.**add**(**Conv2D**(128,(3,3),activation='relu'))
15. model.**add**(**BatchNormalization**())
16. model.**add**(**MaxPooling2D**(pool\_size=(2,2)))
17. model.**add**(**Dropout**(0.25))
18. model.**add**(**Flatten**())
19. model.**add**(**Dense**(512,activation='relu'))
20. model.**add**(**BatchNormalization**())
21. model.**add**(**Dropout**(0.5))
22. model.**add**(**Dense**(2,activation='softmax'))
23. model.**compile**(loss='categorical\_crossentropy',
24. optimizer='rmsprop',metrics=['accuracy'])

5. Analyzing model:

1. model.**summary**()

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2020/05/model-summary-1.png)

6. Define callbacks and learning rate:

1. from keras.callbacks import EarlyStopping, ReduceLROnPlateau
2. earlystop = **EarlyStopping**(patience = 10)
3. learning\_rate\_reduction = **ReduceLROnPlateau**(monitor = 'val\_acc',patience = 2,verbose = 1,factor = 0.5,min\_lr = 0.00001)
4. callbacks = [earlystop,learning\_rate\_reduction]

7. Manage data:

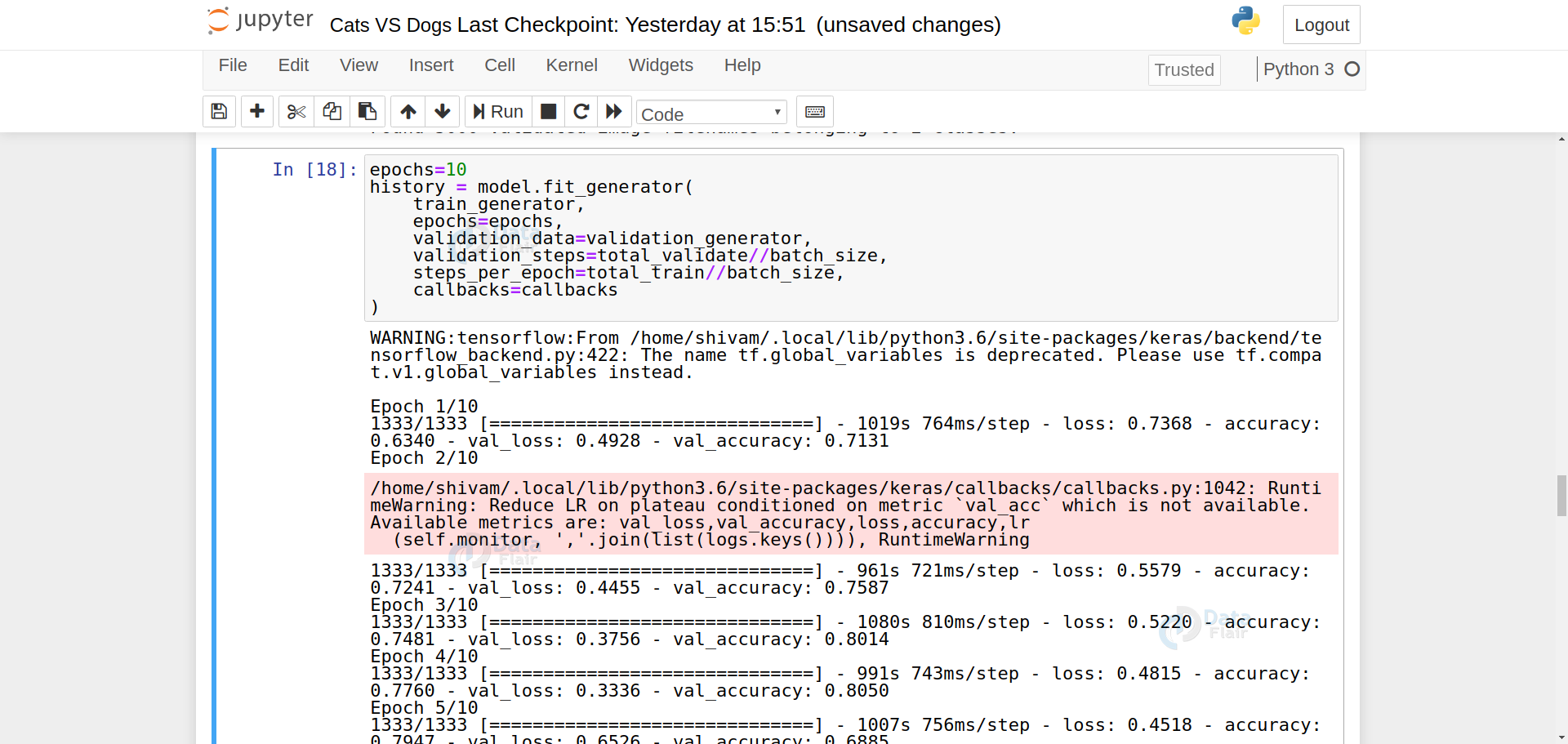
1. df["category"] = df["category"].**replace**({0:'cat',1:'dog'})
2. train\_df,validate\_df = **train\_test\_split**(df,test\_size=0.20,
3. random\_state=42)
4. train\_df = train\_df.**reset\_index**(drop=True)
5. validate\_df = validate\_df.**reset\_index**(drop=True)
6. total\_train=train\_df.shape[0]
7. total\_validate=validate\_df.shape[0]
8. batch\_size=15

8. Training and validation data generator:

1. train\_datagen = **ImageDataGenerator**(rotation\_range=15,
2. rescale=1./255,
3. shear\_range=0.1,
4. zoom\_range=0.2,
5. horizontal\_flip=True,
6. width\_shift\_range=0.1,
7. height\_shift\_range=0.1
8. )
9. train\_generator = train\_datagen.**flow\_from\_dataframe**(train\_df,
10. "./documents/input/train/",x\_col='filename',y\_col='category',
11. target\_size=Image\_Size,
12. class\_mode='categorical',
13. batch\_size=batch\_size)
14. validation\_datagen = **ImageDataGenerator**(rescale=1./255)
15. validation\_generator = validation\_datagen.**flow\_from\_dataframe**(
16. validate\_df,
17. "../documents/input/train/",
18. x\_col='filename',
19. y\_col='category',
20. target\_size=Image\_Size,
21. class\_mode='categorical',
22. batch\_size=batch\_size
23. )
24. test\_datagen = **ImageDataGenerator**(rotation\_range=15,
25. rescale=1./255,
26. shear\_range=0.1,
27. zoom\_range=0.2,
28. horizontal\_flip=True,
29. width\_shift\_range=0.1,
30. height\_shift\_range=0.1)
31. test\_generator = train\_datagen.**flow\_from\_dataframe**(train\_df,
32. "./dogs-vs-cats/test/",x\_col='filename',y\_col='category',
33. target\_size=Image\_Size,
34. class\_mode='categorical',
35. batch\_size=batch\_size)

9. Model Training:

1. epochs=10
2. history = model.**fit\_generator**(
3. train\_generator,
4. epochs=epochs,
5. validation\_data=validation\_generator,
6. validation\_steps=total\_validate//batch\_size,
7. steps\_per\_epoch=total\_train//batch\_size,
8. callbacks=callbacks
9. )

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2020/05/model-traininig.png)

10. Save the model:

1. model.**save**("model1\_catsVSdogs\_10epoch.h5")

11. Test data preparation:

1. test\_filenames = os.**listdir**(".../documents/input/test1")
2. test\_df = pd.**DataFrame**({
3. 'filename': test\_filenames
4. })
5. nb\_samples = test\_df.shape[0]

12. Make categorical prediction:

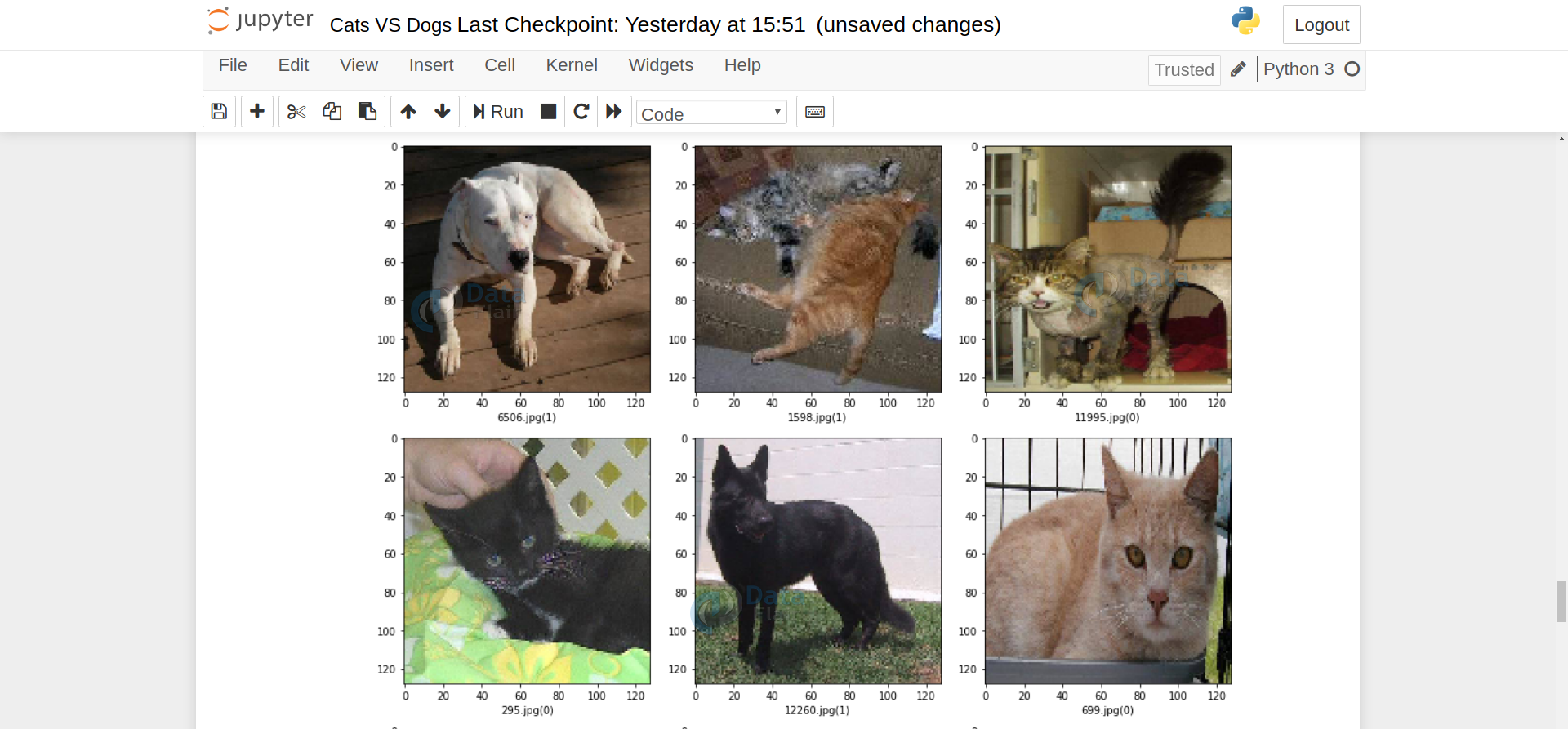
1. predict = model.**predict\_generator**(test\_generator, steps=np.**ceil**(nb\_samples/batch\_size))

13. Convert labels to categories:

1. test\_df['category'] = np.**argmax**(predict, axis=-1)
2. label\_map = **dict**((v,k) for k,v in train\_generator.class\_indices.**items**())
3. test\_df['category'] = test\_df['category'].**replace**(label\_map)
4. test\_df['category'] = test\_df['category'].**replace**({ 'dog': 1, 'cat': 0 })

14. Visualize the prediction results:

1. sample\_test = test\_df.**head**(18)
2. sample\_test.**head**()
3. plt.**figure**(figsize=(12, 24))
4. for index, row in sample\_test.**iterrows**():
5. filename = row['filename']
6. category = row['category']
7. img = **load\_img**("../documents/input/test1/"+filename, target\_size=Image\_Size)
8. plt.**subplot**(6, 3, index+1)
9. plt.**imshow**(img)
10. plt.**xlabel**(filename + '(' + "{}".**format**(category) + ')' )
11. plt.**tight\_layout**()
12. plt.**show**()

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2020/05/sample-data.png)

15. Test your model performance on custom data:

1. results={
2. 0:'cat',
3. 1:'dog'
4. }
5. from PIL import Image
6. import numpy as np
7. im=Image.**open**("\_\_image\_path\_TO\_custom\_image") // for example : 1.jpg
8. im=im.**resize**(Image\_Size)
9. im=np.**expand\_dims**(im,axis=0)
10. im=np.**array**(im)
11. im=im/255
12. pred=model.**predict\_classes**([im])[0]
13. **print**(pred,results[pred])

Cats VS Dogs Classifier GUI:

We do not want to run predict\_classes method every time we want to test our model. That’s why we need a graphical interface. Here we will build the GUI using Tkinter python.

To install Tkinter :

1. sudo apt-get install python3-tk

Now create a new directory, copy your model (“model1\_catsVSdogs\_10epoch.h5”) to this directory.

Create a file gui.py and paste the below code:

1. import tkinter as tk
2. from tkinter import filedialog
3. from tkinter import \*
4. from PIL import ImageTk, Image
5. import numpy
6. from keras.models import load\_model
7. model = **load\_model**('model1\_catsVSdogs\_10epoch.h5')
8. #dictionary to label all traffic signs class.
9. classes = {
10. 0:'its a cat',
11. 1:'its a dog',
12. }
13. #initialise GUI
14. top=tk.**Tk**()
15. top.**geometry**('800x600')
16. top.**title**('CatsVSDogs Classification')
17. top.**configure**(background='#CDCDCD')
18. label=**Label**(top,background='#CDCDCD', font=('arial',15,'bold'))
19. sign\_image = **Label**(top)
20. def **classify**(file\_path):
21. global label\_packed
22. image = Image.**open**(file\_path)
23. image = image.**resize**((128,128))
24. image = numpy.**expand\_dims**(image, axis=0)
25. image = numpy.**array**(image)
26. image = image/255
27. pred = model.**predict\_classes**([image])[0]
28. sign = classes[pred]
29. **print**(sign)
30. label.**configure**(foreground='#011638', text=sign)
31. def **show\_classify\_button**(file\_path):
32. classify\_b=**Button**(top,text="Classify Image",
33. command=lambda: **classify**(file\_path),
34. padx=10,pady=5)
35. classify\_b.**configure**(background='#364156', foreground='white',
36. font=('arial',10,'bold'))
37. classify\_b.**place**(relx=0.79,rely=0.46)
38. def **upload\_image**():
39. try:
40. file\_path=filedialog.**askopenfilename**()
41. uploaded=Image.**open**(file\_path)
42. uploaded.**thumbnail**(((top.**winfo\_width**()/2.25),
43. (top.**winfo\_height**()/2.25)))
44. im=ImageTk.**PhotoImage**(uploaded)
45. sign\_image.**configure**(image=im)
46. sign\_image.image=im
47. label.**configure**(text='')
48. **show\_classify\_button**(file\_path)
49. except:
50. pass
51. upload=**Button**(top,text="Upload an image",command=upload\_image,padx=10,pady=5)
52. upload.**configure**(background='#364156', foreground='white',font=('arial',10,'bold'))
53. upload.**pack**(side=BOTTOM,pady=50)
54. sign\_image.**pack**(side=BOTTOM,expand=True)
55. label.**pack**(side=BOTTOM,expand=True)
56. heading = **Label**(top, text="CatsVSDogs Classification",pady=20, font=('arial',20,'bold'))
57. heading.**configure**(background='#CDCDCD',foreground='#364156')
58. heading.**pack**()
59. top.**mainloop**()

Save this file and run using:

1. python3 gui.py

[..\Videos\Captures\CatsVSDogs Classification 2020-08-30 21-12-56.mp4](../Videos/Captures/CatsVSDogs%20Classification%202020-08-30%2021-12-56.mp4)

Summary:

This Deep Learning project for beginners introduces you to how to build an image classifier. This project takes The Asirra (catsVSdogs) dataset for training and testing the neural network. In this project, we have learned:

* How to create a neural network in [Keras](https://keras.io/) for image classification
* How to prepare the dataset for training and testing
* How to visualize the dataset
* How to save the model
* How to test our model performance on custom data
* How to create a GUI for the execution of deep learning project

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*THANK YOU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*