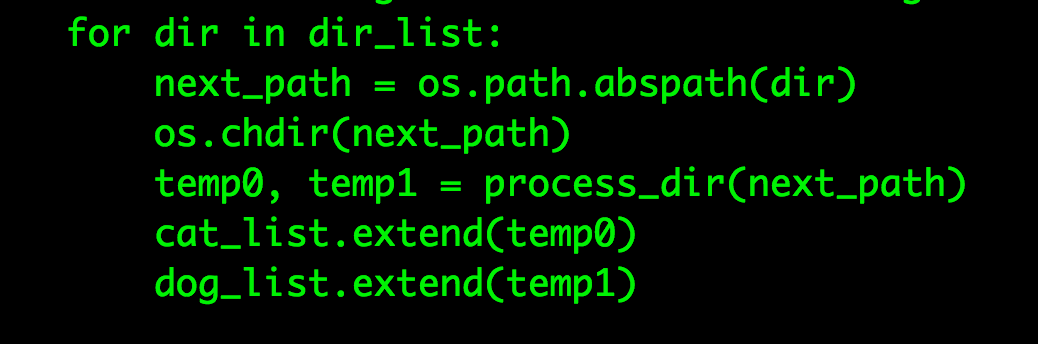
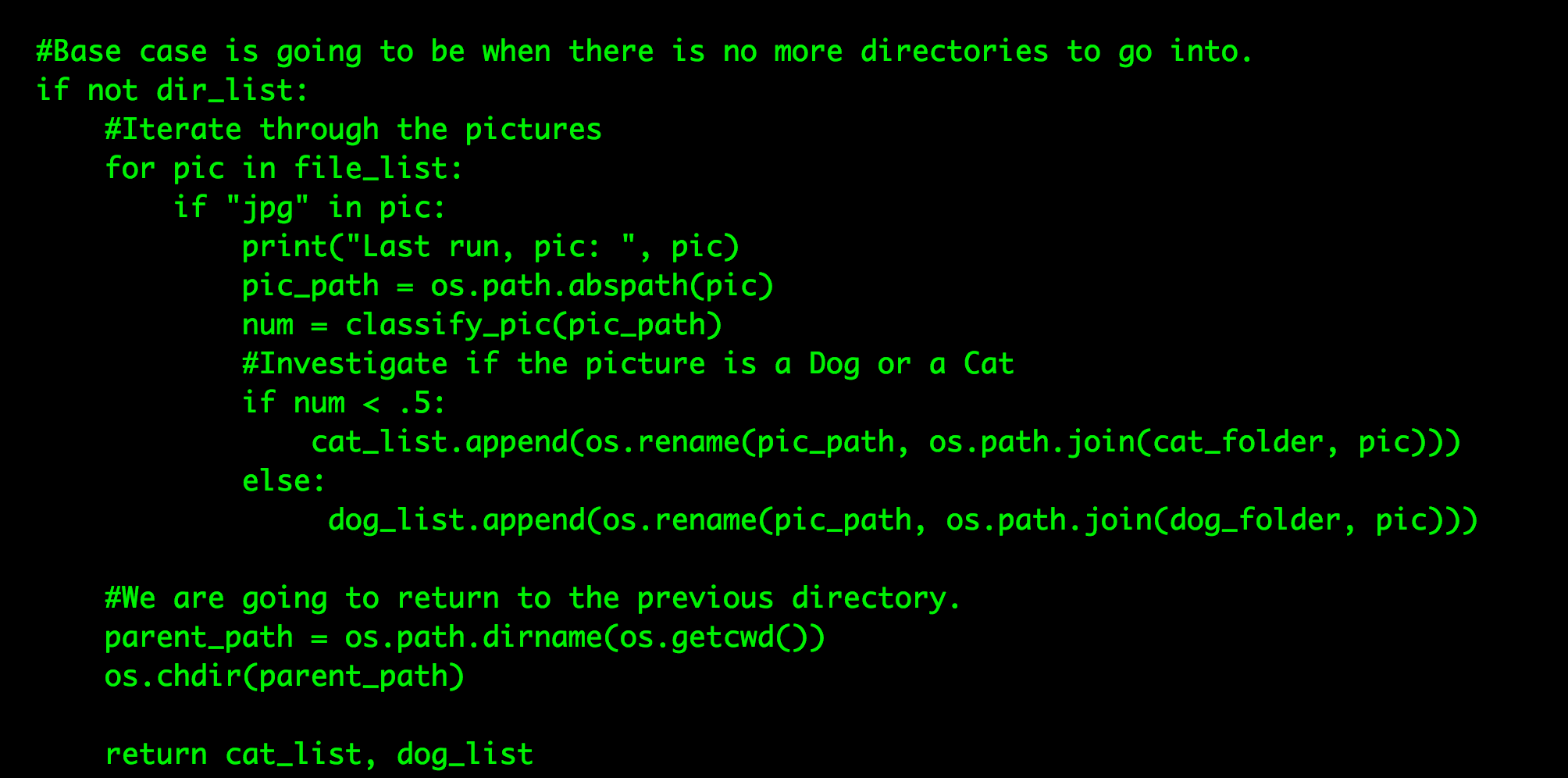
Lab 1a

For this lab the problem is that there is a folder were people were going to be saving the images of dogs and cats. However, they were not organized, they were placing new folders for the dogs and cats within folders, the pictures were not organized in any way. The pictures are all over many layers of directories. Through recursion we suppose to traverse the directories and find all of the pictures in how ever many directories they are in. We are going to use the method *classify\_pic,* which implements machine learning to classify the picture to be either a dog or a cat. Depending on the result they are going to be moved to the corresponding folder.

The first thing that I did was that I created *global* folders called “Dog” and “Cat”, these folders are going to hold the corresponding pictures. The way that I approach this problem is to first think about it in one layer of directories, for example, I am in the *Pictures* directory and there are two directories that contain only jpegs files inside. For this I implemented a for-loop that is going to call its self with how ever many directories they are in the current working directory.



The *os.chdir()* method is going to change the current directory to the next layer of the directory. Then we are going to call itself but now with a new directory path. The two last lines are going to add the cat and dog pictures to the current list. This is going to run until it hits the base case, this is when there are not more layers of directories to advance to.



From there we are going change directories again to the previous parent directory and continue to the next directory. Finally, before changing any directories we are going to collect the pictures in the current directory. The names of the files are going to be given to us through the *get\_dirs\_and\_files ()* method, from there we are going to use *os.path.abspath()* method to get the absolute path of the file. Then we are going to use the *os.rename* and *os.path.join* methods to rename its path to the either the dog’s or cat’s folder. This is going to be decided based on the *classify\_pic()* method that is going to return a number. If the number is greater than .5 the file is a dog, otherwise it’s a cat. Depending on the number they are going to be moved to the corresponding folder.

For the test cases I created a variety of directories and moved pictures around to see if the program will traverse correctly. The program did manage to get all the pictures and once again move them to the correct folder. The final test was that I placed PDF files inside random directories, to see if the program could manage to ignore the files and focus on the pictures of the dogs and the cats. The program did not crash, and it manage to ignore all other files that were not of type *jpeg*, this is because of the if statements that checks for *jpegs*. If other files are not of that extension, then the program is going to ignore it and continue with the other files.

Before running the program.


Figure Before running the program, I inserted a PDF to see if will ignore this file.

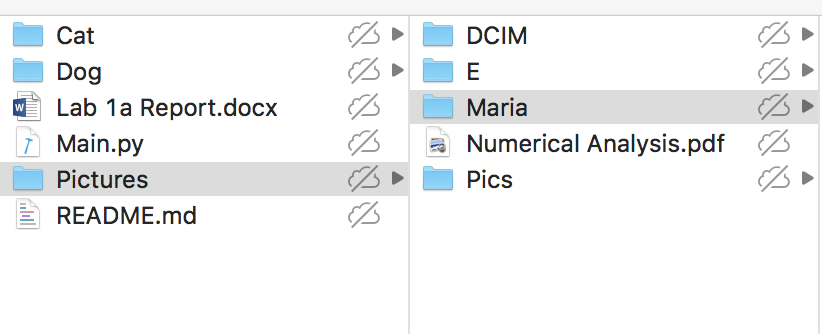


Figure The program will create a Dog and Cat folder, it will ignore the files that are not jpegs.

For this lab I learned how to use the *os* module of python. This module lets me move file to other directories, create directories, remove directories, among other methods for directory manipulation. I also learned how to use tuples in python, were I was saving the path of the picture in relation to the date that they we created.

“I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.”

Source Code

import os

import random

import io

from PIL import Image

'''

Eduardo Herrera

CS2303

Professor Aguirre

Date: 09/12/2018

Going to make a folder for the dogs and another folder for the cats pictures. Then

we are going to traverse the directories through recusion and get all the dog and cat pictures. Finally,

all the pictures are going to be moved to the corresponding folder.

'''

#Going to get the files and directories names of all the files

def get\_dirs\_and\_files(path):

dir\_list = [directory for directory in os.listdir(path) if os.path.isdir(path + '/' + directory)]

file\_list = [directory for directory in os.listdir(path) if not os.path.isdir(path + '/' + directory)]

return dir\_list, file\_list

#Going to classify the picture is they are a cat or a dog

def classify\_pic(path):

# To be implemented by Diego: Replace with ML model

if "dog" in path:

return 0.5 + random.random() / 2

return random.random() / 2

#Going to go trough the directories and move the pics to the corresponding folder

def process\_dir(path):

dir\_list, file\_list = get\_dirs\_and\_files(path)

cat\_list = []

dog\_list = []

#Base case is going to be when there is no more directories to go into.

if not dir\_list:

#Iterate through the pictures

for pic in file\_list:

if "jpg" in pic:

#print("Last run, pic: ", pic)

pic\_path = os.path.abspath(pic)

num = classify\_pic(pic\_path)

#Investigate if the picture is a Dog or a Cat

if num < .5:

cat\_list.append((os.rename(pic\_path, os.path.join(cat\_folder, pic))), os.path.getmtime(pic\_path))

else:

dog\_list.append((os.rename(pic\_path, os.path.join(dog\_folder, pic))), os.path.getmtime(pic\_path))

#We are going to return to the previous directory.

parent\_path = os.path.dirname(os.getcwd())

os.chdir(parent\_path)

return cat\_list, dog\_list

else:

#Iterate through the pictures

for pic in file\_list:

if "jpg" in pic:

pic\_path = os.path.abspath(pic)

num = classify\_pic(pic\_path)

#Investigate if the picture is a Dog or a Cat

if num < .5:

cat\_list.append((os.rename(pic\_path, os.path.join(cat\_folder, pic))), os.path.getmtime(pic\_path))

else:

dog\_list.append((os.rename(pic\_path, os.path.join(dog\_folder, pic))), os.path.getmtime(pic\_path))

#Iterate through the Directories through recursion

for dir in dir\_list:

next\_path = os.path.abspath(dir)

os.chdir(next\_path)

temp0, temp1 = process\_dir(next\_path)

cat\_list.extend(temp0)

dog\_list.extend(temp1)

#We are going to return to the previous directory

os.chdir(os.path.dirname(os.getcwd()))

return cat\_list, dog\_list

def main():

global main\_path

global dog\_folder

global cat\_folder

main\_path = os.getcwd()

path = os.path.join(os.getcwd(), "Pictures")

dog\_folder = os.path.join(os.getcwd(), "Dog")

cat\_folder = os.path.join(os.getcwd(), "Cat")

if not os.path.exists(dog\_folder) and not os.path.exists(cat\_folder):

os.makedirs(dog\_folder)

os.makedirs(cat\_folder)

os.chdir(path)

print(path)

cats, dogs = process\_dir(path)

main()