```
from PIL import Image
import PIL
import sys
import csv
# this program will output the rgb average of each pic to be used in weka to predict what each is


#-------------------Converting image Size----------------
#-----------------------to 256*256-----------------------
def resize(path,name,amount,newSize): # name should be followed by the number of that item
    # path is the path to the image
    # amount is total amount of images
    amount = int(amount) +1
    newWidth = int(newSize)
    for i in range(1,amount):
        im = Image.open(path + name + str(i)+'.jpg', 'r') #1
        percent = (newWidth/float(im.size[0]))
        newsize = int((float(im.size[1]))*float(percent))
        im = im.resize((newWidth,newsize), PIL.Image.ANTIALIAS)
        im.save('/home/rick/forScience/AaO2/'+name+ str(i) +'.jpg')


#--------------------new version-----------------------
#----------------for writing to the file ---------------
def toCsv(amount,path,types,size,path2):
    with open(path2+'test.csv','a',newline='') as fh: # creat the header first do not want to have duplicates
        writeto = csv.writer(fh)
        data = [['red', 'green', 'blue', 'Type']] #type will be used to test the correctness in weka
        writeto.writerows(data)
    for x in range(amount): # iterate through till all the types have been added to the csv file
        writeout(path,types[x],size,path2)



def writeout(path,type,size,path2):
    print("get here")
    with open(path2+'test.csv','a',newline='') as fh:
        writeto = csv.writer(fh)
        Type = str(type).capitalize()
        size = int(size)+1
        for i in range(1,size):
            im = Image.open(path+type+str(i)+'.jpg', 'r') #2
            pix = list(im.getdata())
            pixflat = [x for sets in pix for x in sets]
            red = pixflat[0:256*256]
            red1 = sum(red)/(256*256)
            green = pixflat[(256*256)+1:256*256*2]
            green1 = sum(green)/(256*256)
            blue = pixflat[(256*256*2)+1:256*256*3]
            blue1 = sum(blue)/(256*256)
            writeto = csv.writer(fh)
            data = [[float(red1),float(green1),float(blue1),Type]]
            writeto.writerows(data)


    # got same at end pos #at pos-500 get 253 so minus 499 from max len to get pos
    # just took the mean of this since 256^2 is a big set of num to large for csv file
def home():
    confirm1 = input("Enter yes if you want to resize the images to be 256x256: ")
    if(str(confirm1) == 'yes'):
        name = input("Enter the name of the item: ")
        newSize = input("Enter the new size of the image to be: example 256 = 256x256\n")
        path = input("Enter the path: make sure to enter the end /\nFor example /home/Documents/ would
be the path to any item in the Documents folder: ")
        amount = input("Enter the amount of images you want to resize: ")
        resize(path,name,amount,newSize)
    else:
        print("Going to next step")
    confirm2 = input("Enter yes if you want to convert the images into a csv file:")
    if(str(confirm2) == 'yes'):
        size = input("Enter the amount of the images: make sure all the different types has the same
amount of images\n")
        amount = input("Enter how many different types are you wanting to convert: ")
        path = input("Enter the path: make sure to enter the end /\nFor example /home/Documents/ would
be the path to any item in the Documents folder: ")
        path2 = input("Enter the path where you want to save the csv file: ")
        print("Enter the names one at a time then press enter to continue: ")
        types = [str(x) for x in sys.stdin.readline().split()]
        toCsv(int(amount),path,types,int(size),path2)
        print("Now you should have a file in " + path2 + " that has the file imageRGB.csv\nFinished")

start = home()

#1 http://stackoverflow.com/questions/273946/how-do-i-resize-an-image-using-pil-and-maintain-its-
aspect-ratio - used to resize the image
#2 https://www.hackerearth.com/practice/notes/extracting-pixel-values-of-an-image-in-python/  - used
for converting images to RGB bitmap
```