PYTHON PROJECT CA-2

Report

Project No: 13

Project Name: Image Base Captcha Using Python

Course Code: INT213

Section: K21PB

Submitted to

Dr. Ramandeep Sandhu (28362)

Assistant Professor

Division of Computer Science Engineering, LPU

Submitted by

Name	Roll No.	Reg No.
Kantheti Venkata Sahil	RK21PBB36	12110580
Suggala N V Sathwik	RK21PBA20	12110366
Madarapu Purna Sai	RK21PBB68	12110664

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

ABSTRACT

Nowadays, security is a major concern and has garnered foremost attention in recent years as a result of the Internet's evident impact on all facets of human life. To provide an appropriate level of security, the Completely Automated Public Turing Test to Tell Computer and Human Apart, or simply CAPTCHA, has been designed for circumstances when bots must be prevented from performing a specific action, such as signing up and downloading. At the same time, it should be designed in such a way that people can execute the same action. Despite its many benefits, there are several major concerns about security, usability, and accessibility that make its use contentious. In this work, we attempted to provide a comprehensive analysis of many features and the state-of-the-art of CAPTCHA in general, as well as its alternatives in particular, to assist researchers in focusing on specific concerns to propose new solutions and ideas. With the advancement of CAPTCHA development, image-based CAPTCHAs have been described and compared. This project could be useful in future studies aimed at developing new strategies for addressing current flaws.

Introduction

Using the Internet for various purposes has become a daily activity for practically all individuals all over the world, who use it to research, shop, use programs, and do a variety of other things. Because there are many unfair users like attackers, hackers, and spammers who intend to have unauthorized access to these applications (information), spam, hack accounts, and do other abuses, the importance of maintaining security attracts more attention as the amount of information and applications grows exponentially. As a result, the crucial importance of (information) security in securing applications and their most valuable assets, namely their private information, has been highlighted more than ever before. The security domain is quite vast, and it should be able to control access, protect programs, detect intrusions, restore damage, and block unauthorized users, among other things. Several policies, processes, and instruments have been proposed to deal with the growing number of security threats . The Completely Automatic Public Turing Test to Tell Computers and Humans Apart (CAPTCHA), which was first introduced more than a decade ago, is now widely recognized as a useful security technique . Several distinct variants of CAPTCHA have been proposed and used for diverse applications in the years since its development. On the other hand, because of the issues and shortcomings of CAPTCHAs, efforts have been made to provide alternate ways.

This python mini project is the implementation of the Image Based Captcha using python. In this project we are using the GUI implementation. CAPTCHAs are, by definition, fully automated, requiring little human maintenance or intervention to administer, producing benefits in cost and reliability. Captcha is used for human verification . Image captcha is more secure than text captcha . The main purpose of a captcha system is to block spambots while allowing human users. In this, based on the images user has to select that image which specify the question. After clicking the submit button, answers should be verify with the user response. If captcha is more complex then user can skip that for the next captcha. In one of the variants of CAPTCHA challenges, images are not incrementally highlighted, but fade out when clicked, and replaced with a new image fading in, resembling whack a mole. Criticism has been aimed at the long duration taken for the images to fade out and in.

A way to tell computers and Humans apart by conducting a test called a Turing test in Turing test human acts as an evaluator. If a Computer program is able to conduct such test and evaluate the result then it is known as CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). A CAPTCHA is a type of challenge-response test used to determine whether or not the user is human. This form of CAPTCHA has a system impulse the user to type the letters of a shown dis manner image, sometimes with the combination of an altered sequence of letters or digits that appears on the screen.

Objectives:

1. The objective of CAPTCHA schemes is mainly to distinguish between human and computer systems.
2. To make websites more secure.
3. Blocks the Automated Systems.
4. Protects online polling

Design:

1)Use of GUI

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. In GUI there are many widgets. Some are buttons, image, label, canvas, check button, entry, frame, menubox, messagebox etc.

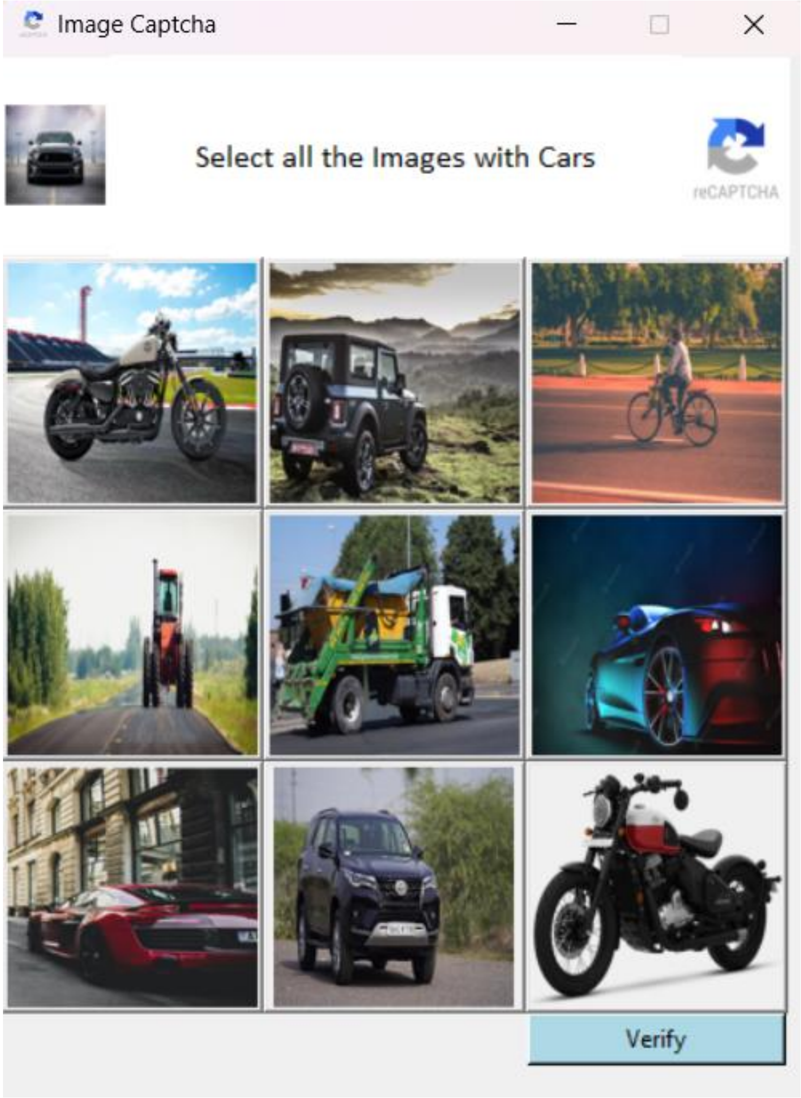
2)Use of GUI and its usability in making platform:

In this project we used the GUI platform. Firstly created the tkinter window. Size of tkinter window is 450x525 and it cannot be resizable. For this tk window, given the title as Image Captcha and the logo. Created the two frames. One frame is used for the 'what to select in captcha'. Another frame is used for the creation of captcha. In image based captcha mainly there is a selection of Images. For selection of images used the buttons to select. For this we used images as buttons using button command. If the user selected a image rather than car images then a same value will be appended into list. For every car image we had given different values to make captcha more releable to use. For the alignment of the images used grid as a tool. At the ending, given verify button. After selecting all the images, user need to click the Verify button. Total two frames used. Total ten buttons are used in the project.and every button will return a command(which call a function) after selection. For overall used grid for alignment. Used the message box to show whether the user selected the correct images or wrong images. If selected images are correct the message box will display 'Captcha Correct'. If selected images are incorrect the message box will display 'Captcha Incorrect'.

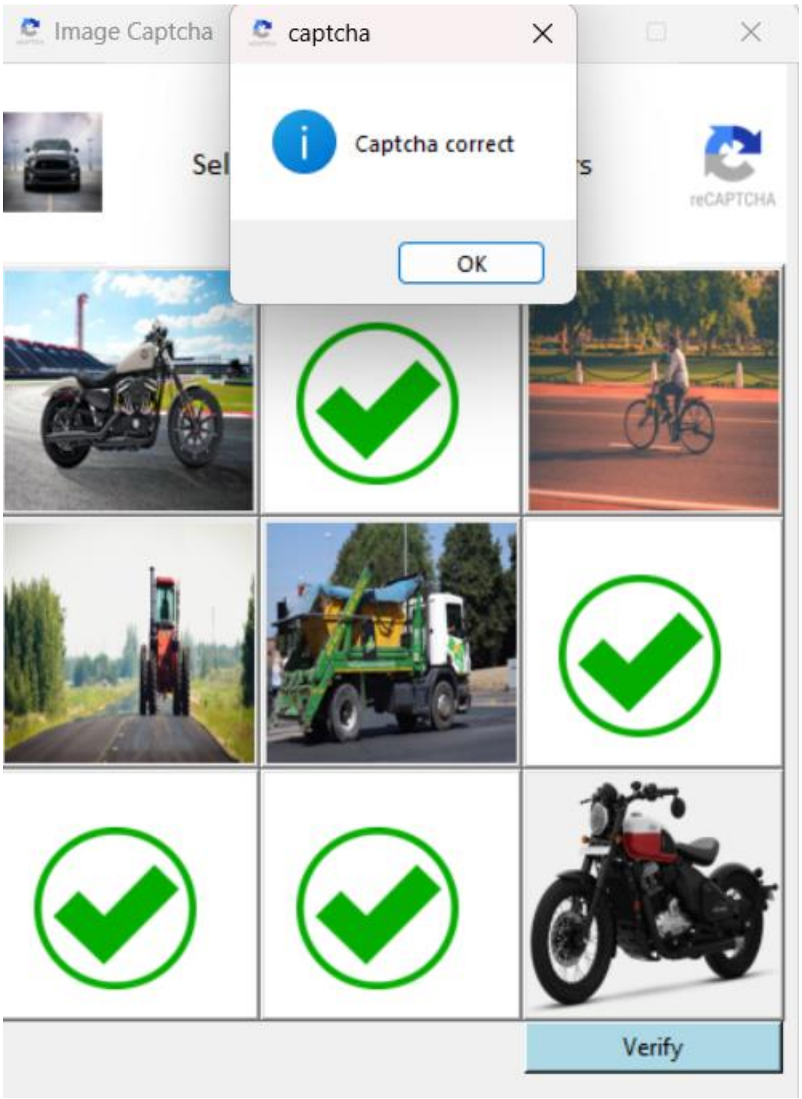
3)Algorithm used:

```
1. Declare list
2. Define a function click()
   Set x=string ‘py1’
   Append x into list
3. Define a function sel1()
   Set x=string ‘py2’
   Append x into list
4. Define a function sel2()
   Set x=string ‘py3’
   Append x into list
5. Define a function sel3()
   set x=string ‘py4’
   Append x into list
6. Define a function sel4()
   set x=string ‘py5’
   Append x into list
7. Define a function verify()
   if py1 in list then,
         print a message “capcha is incorrect”
   elif py2,py3,py4,py5 in list then,
         print a message “captcha is correct”
   else print a message “captcha is incorrect”
```

Screenshot of Project

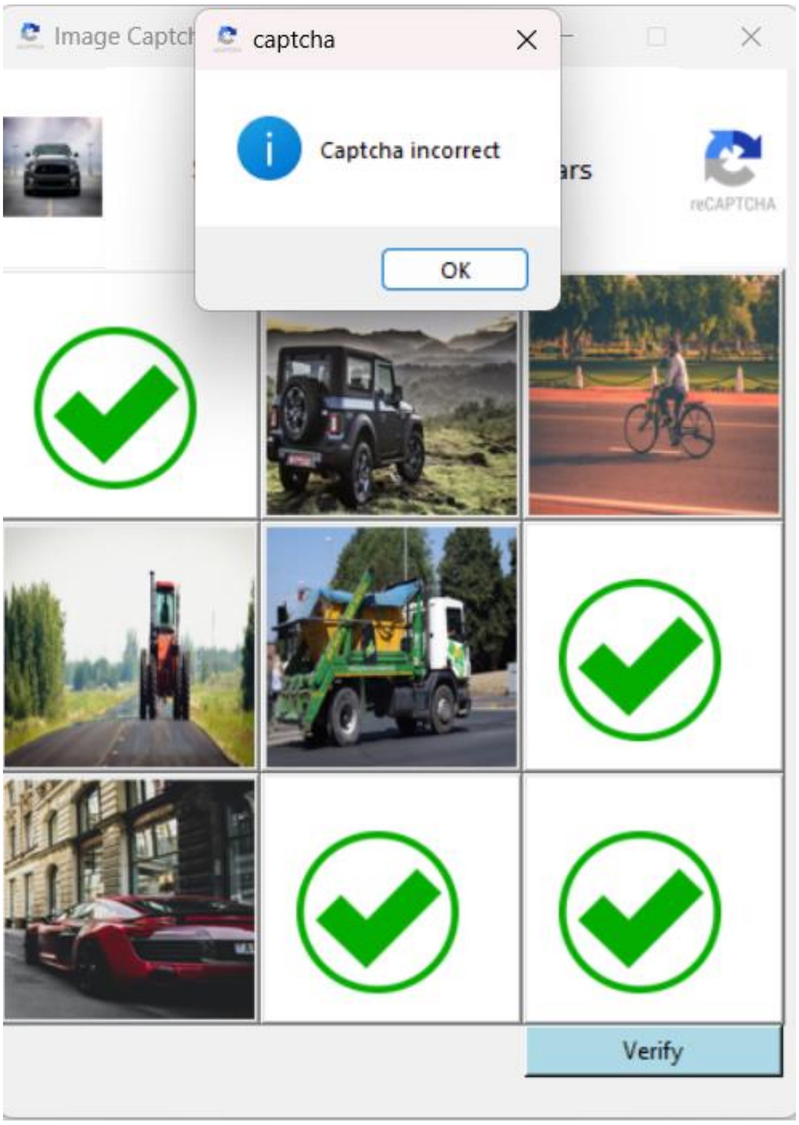


Output if correct Captcha is entered



Captcha correct

Output if incorrect Captcha is entered



Conclusion

As a technique for preventing web bots and scripts from impersonating human activities, the CAPTCHA has a variety of uses, including safeguarding online voting, Ecommerce, and sign up/login. CAPTCHAs come in a variety of forms, including text-based and image-based, and may be used in a variety of scenarios. Here we talk about only Image Captcha. Apart from the inherent difficulties in establishing and securing captchas, usability and accessibility concerns are key roadblocks to their adoption. Several solutions have been offered to address these major issues that might jeopardise the effectiveness of captchas. Even though present options are functional and valuable, some of them have issues owing to increased implementation costs. Overall, a hybrid protection strategy, including captcha, is assessed wherever necessary to acquire better security. In order to address the aforementioned concerns. This is the Image Captcha Project.

References

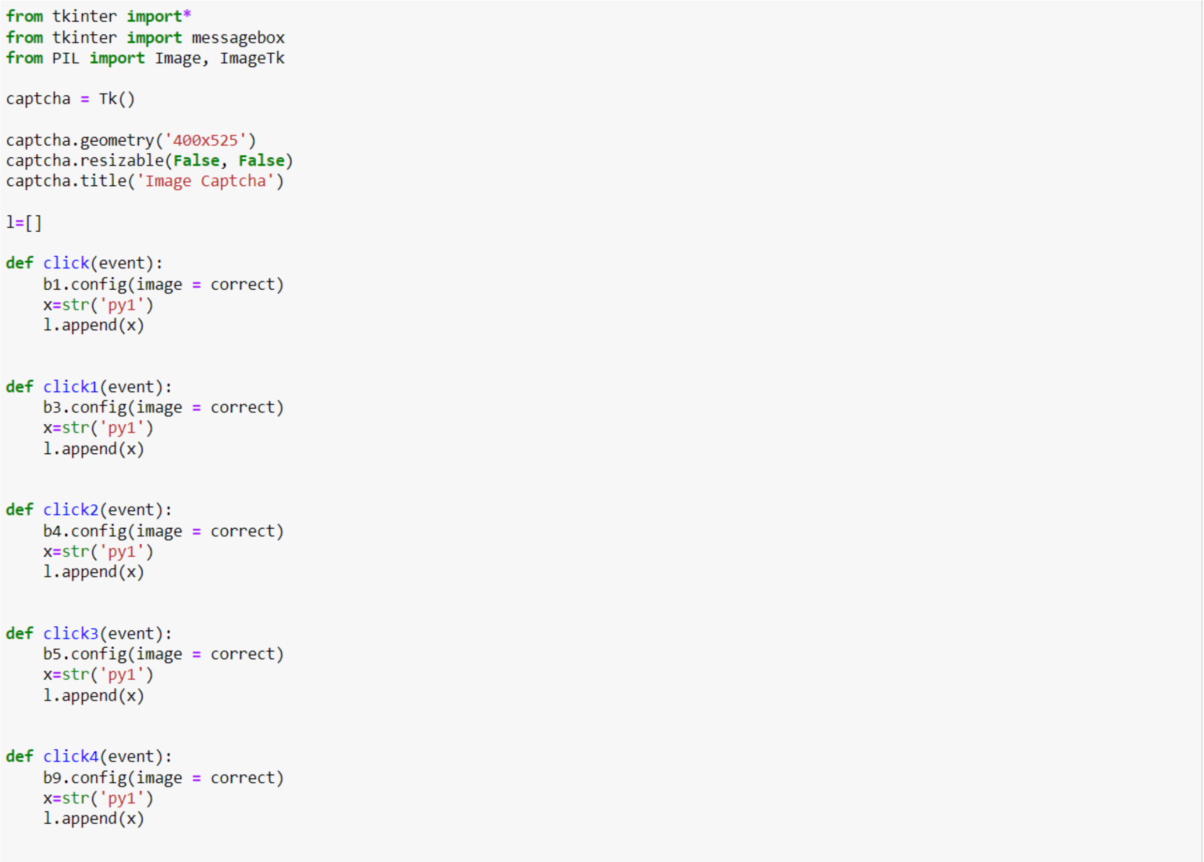
https://curlie.org/Computers/Internet/Abuse/CAPTCHA/

https://www.w3.org/TR/turingtest/

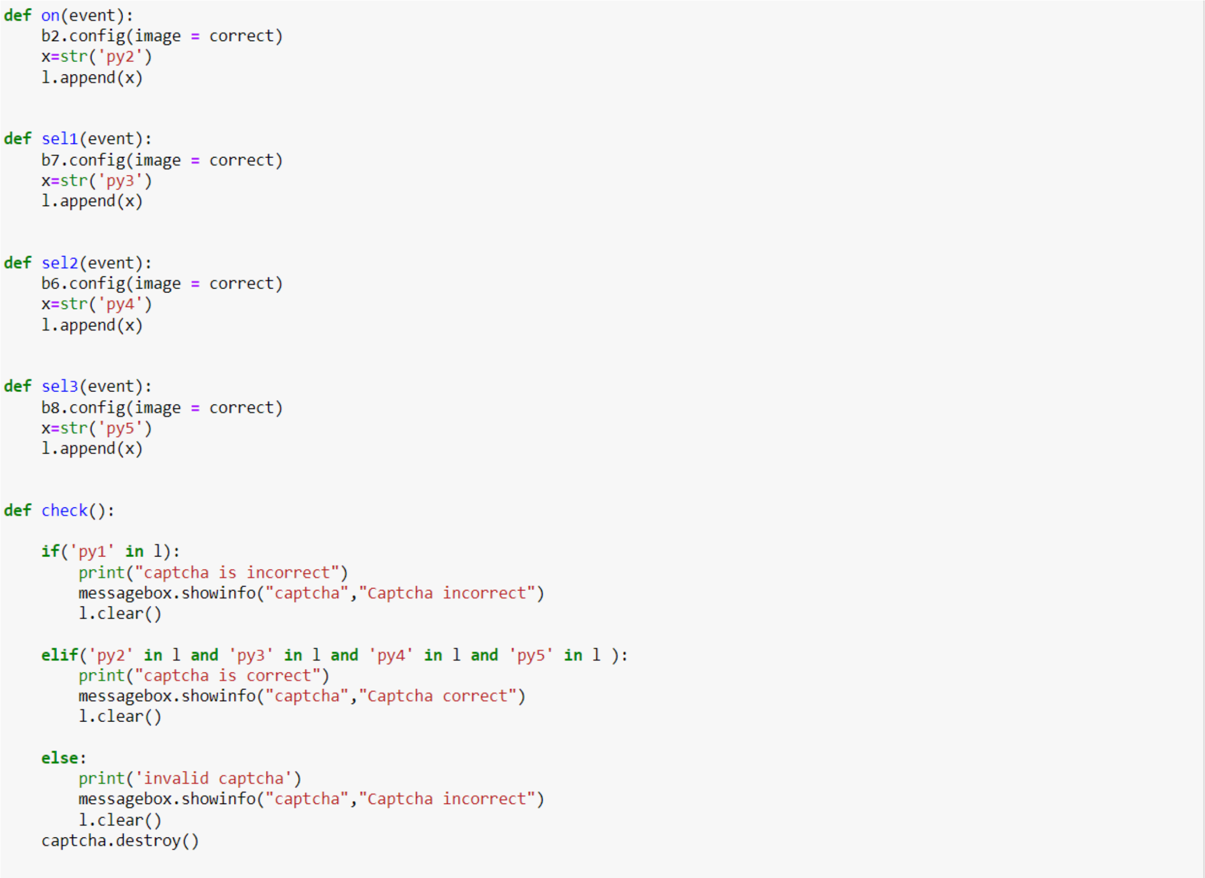
https://web.archive.org/web/20120205201803/http://www2.parc.com/istl/projects/captcha/history.htm

https://web.archive.org/web/20171027203659/https://www.cylab.cmu.edu/partners/success-stories/recaptcha.html

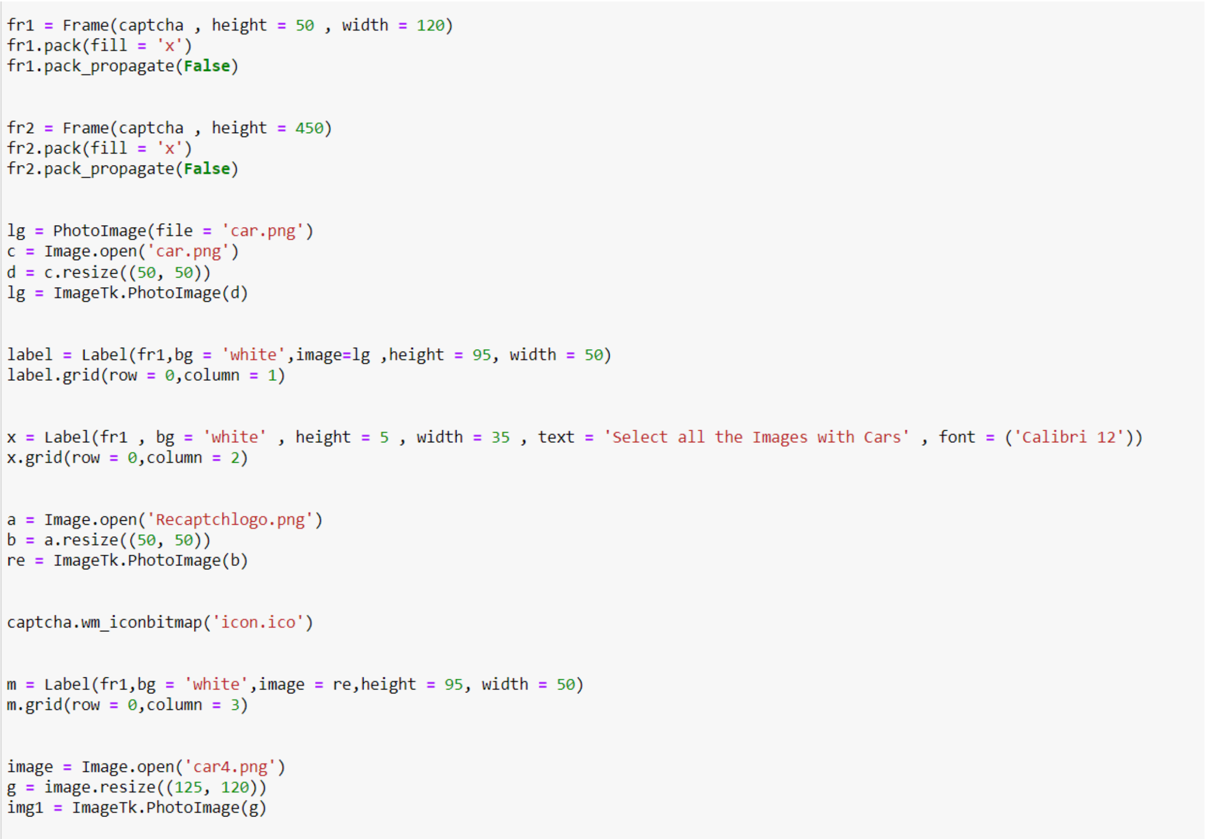
Screenshot of Code



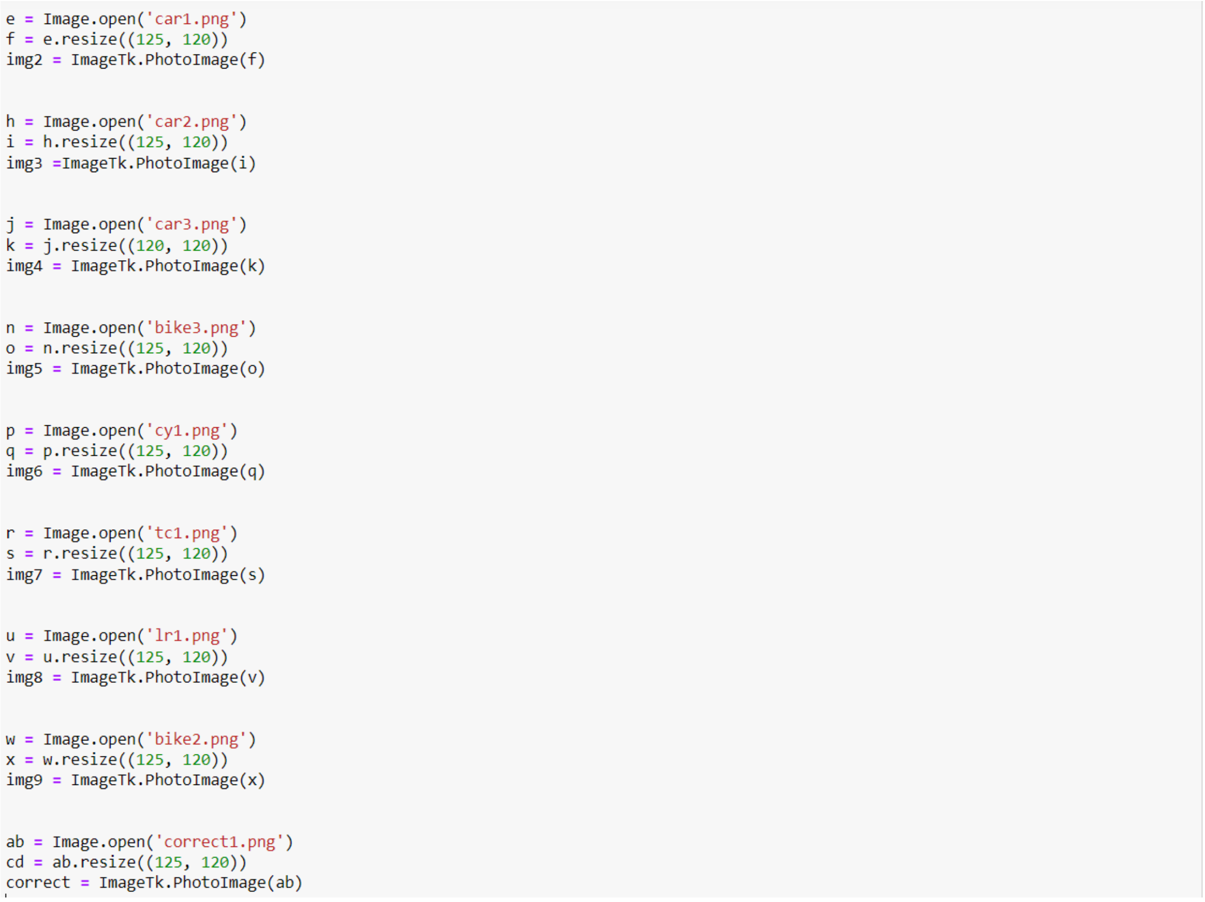
```
from tkinter import*
from tkinter import messagebox
from PIL import Image, ImageTk

captcha = Tk()

captcha.geometry('400x525')
captcha.resizable(False, False)
captcha.title('Image Captcha')

l=[]

def click(event):
    b1.config(image = correct)
    x=str('py1')
    l.append(x)


def click1(event):
    b3.config(image = correct)
    x=str('py1')
    l.append(x)


def click2(event):
    b4.config(image = correct)
    x=str('py1')
    l.append(x)


def click3(event):
    b5.config(image = correct)
    x=str('py1')
    l.append(x)


def click4(event):
    b9.config(image = correct)
    x=str('py1')
    l.append(x)
```



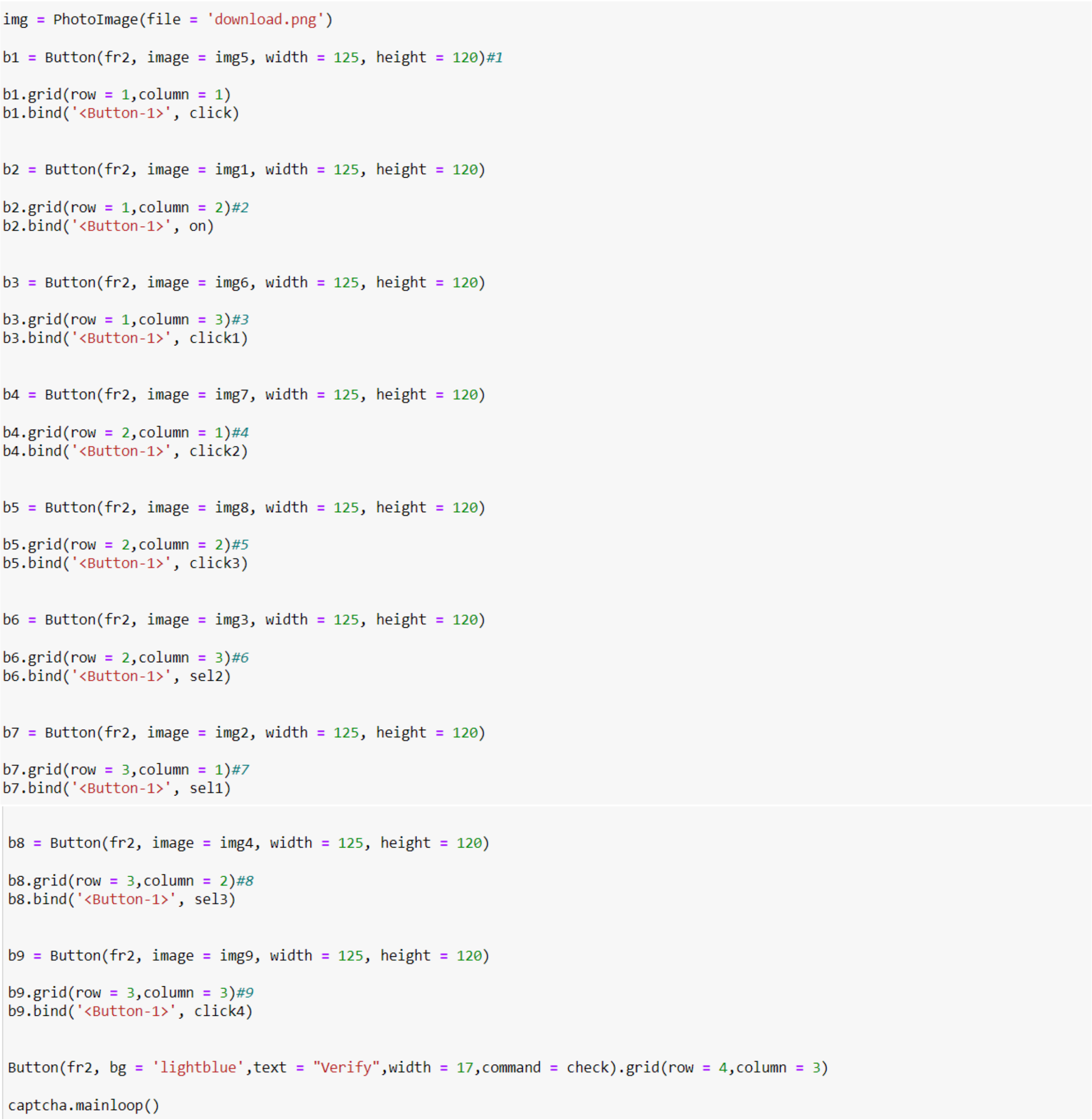
```
def on(event):
    b2.config(image = correct)
    x=str('py2')
    l.append(x)


def sel1(event):
    b7.config(image = correct)
    x=str('py3')
    l.append(x)


def sel2(event):
    b6.config(image = correct)
    x=str('py4')
    l.append(x)


def sel3(event):
    b8.config(image = correct)
    x=str('py5')
    l.append(x)


def check():

    if('py1' in l):
        print("captcha is incorrect")
        messagebox.showinfo("captcha","Captcha incorrect")
        l.clear()

    elif('py2' in l and 'py3' in l and 'py4' in l and 'py5' in l ):
        print("captcha is correct")
        messagebox.showinfo("captcha","Captcha correct")
        l.clear()

    else:
        print('invalid captcha')
        messagebox.showinfo("captcha","Captcha incorrect")
        l.clear()
    captcha.destroy()
```



```
fr1 = Frame(captcha , height = 50 , width = 120)
fr1.pack(fill = 'x')
fr1.pack_propagate(False)


fr2 = Frame(captcha , height = 450)
fr2.pack(fill = 'x')
fr2.pack_propagate(False)


lg = PhotoImage(file = 'car.png')
c = Image.open('car.png')
d = c.resize((50, 50))
lg = ImageTk.PhotoImage(d)


label = Label(fr1,bg = 'white',image=lg ,height = 95, width = 50)
label.grid(row = 0,column = 1)


x = Label(fr1 , bg = 'white' , height = 5 , width = 35 , text = 'Select all the Images with Cars' , font = ('Calibri 12'))
x.grid(row = 0,column = 2)


a = Image.open('Recaptchlogo.png')
b = a.resize((50, 50))
re = ImageTk.PhotoImage(b)


captcha.wm_iconbitmap('icon.ico')


m = Label(fr1,bg = 'white',image = re,height = 95, width = 50)
m.grid(row = 0,column = 3)


image = Image.open('car4.png')
g = image.resize((125, 120))
img1 = ImageTk.PhotoImage(g)
```



```
e = Image.open('car1.png')
f = e.resize((125, 120))
img2 = ImageTk.PhotoImage(f)


h = Image.open('car2.png')
i = h.resize((125, 120))
img3 =ImageTk.PhotoImage(i)


j = Image.open('car3.png')
k = j.resize((120, 120))
img4 = ImageTk.PhotoImage(k)


n = Image.open('bike3.png')
o = n.resize((125, 120))
img5 = ImageTk.PhotoImage(o)


p = Image.open('cy1.png')
q = p.resize((125, 120))
img6 = ImageTk.PhotoImage(q)


r = Image.open('tc1.png')
s = r.resize((125, 120))
img7 = ImageTk.PhotoImage(s)


u = Image.open('lr1.png')
v = u.resize((125, 120))
img8 = ImageTk.PhotoImage(v)


w = Image.open('bike2.png')
x = w.resize((125, 120))
img9 = ImageTk.PhotoImage(x)


ab = Image.open('correct1.png')
cd = ab.resize((125, 120))
correct = ImageTk.PhotoImage(ab)
```



```
img = PhotoImage(file = 'download.png')

b1 = Button(fr2, image = img5, width = 125, height = 120)#1

b1.grid(row = 1,column = 1)
b1.bind('<Button-1>', click)


b2 = Button(fr2, image = img1, width = 125, height = 120)

b2.grid(row = 1,column = 2)#2
b2.bind('<Button-1>', on)


b3 = Button(fr2, image = img6, width = 125, height = 120)

b3.grid(row = 1,column = 3)#3
b3.bind('<Button-1>', click1)


b4 = Button(fr2, image = img7, width = 125, height = 120)

b4.grid(row = 2,column = 1)#4
b4.bind('<Button-1>', click2)


b5 = Button(fr2, image = img8, width = 125, height = 120)

b5.grid(row = 2,column = 2)#5
b5.bind('<Button-1>', click3)


b6 = Button(fr2, image = img3, width = 125, height = 120)

b6.grid(row = 2,column = 3)#6
b6.bind('<Button-1>', sel2)


b7 = Button(fr2, image = img2, width = 125, height = 120)

b7.grid(row = 3,column = 1)#7
b7.bind('<Button-1>', sel1)
```

```
b8 = Button(fr2, image = img4, width = 125, height = 120)

b8.grid(row = 3,column = 2)#8
b8.bind('<Button-1>', sel3)


b9 = Button(fr2, image = img9, width = 125, height = 120)

b9.grid(row = 3,column = 3)#9
b9.bind('<Button-1>', click4)


Button(fr2, bg = 'lightblue',text = "Verify",width = 17,command = check).grid(row = 4,column = 3)

captcha.mainloop()
```