EYE CLASSIFICATION

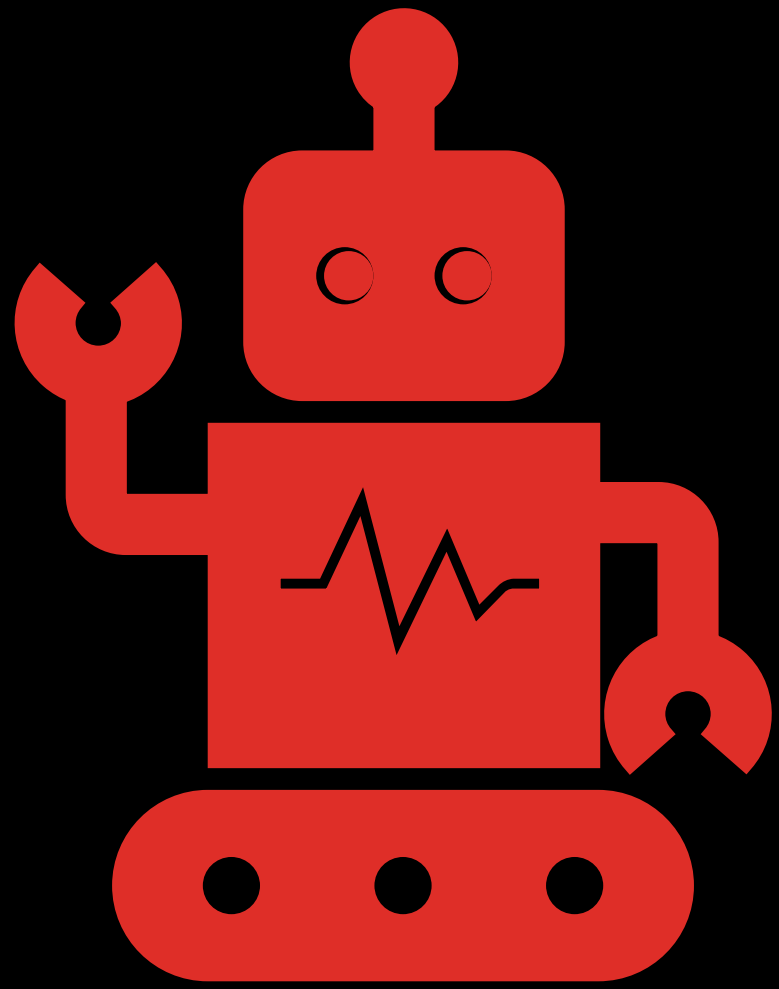
By: 1. Aya Ahmad Saad.
2. Manar Sameh Ramadan.
3. Fayza Khaled Mashaal.

REQUIRED PACKAGES

- For User Interface (UI):
  - Tkinter
- For Machine Learning (ML):
  - Sklearn.
  - Numpy.
  - Matplotlib.

FIRST

Machine learning code.



DATA

```
# By: Aya Ahmad Saad - Fayza Khaled Mashaal - Manar Sameh Ramadan
# Data
x_ = [115, 91, ……, 197, 145]  # first feature
y_ = [136, 98, ……, 131, 76]  # second features
w_ = [243*207, 224*225, ……, 252*200, 273*184] #third feature

class_ = ['Right', 'Right’, ……, 'Left', 'Left']  # Label or target varible.
```

PREPROCESING

```
# Preprocessing
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
x = le.fit_transform(x_)
y = le.fit_transform(y_)
z = le.fit_transform(w_)

features = list(zip(x, y, z))
label = le.fit_transform(class_)
```

VISULIZATION

```
# Visulization
import matplotlib.pyplot as plt
def visulize():
      plt.scatter(x[0:25], y[0:25], color='g', marker='s', label='Right')
      plt.scatter(x[25:49], y[25:49], color='r', marker='*', label='Left')
      plt.show()
```

TRAIN & TEST DATA

```
# Train and Test data
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(features, label,
                                                    test_size=0.3, random_state=42)
```

MODEL & PREDICTION

```
# Model
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=5)
model.fit(x_train, y_train)

# predict
predicted = model.predict(x_test)
```

ACCURACY (EVALUATION)

```
# For accuracy and evaluation
from sklearn.metrics import accuracy_score
score = accuracy_score(y_test, predicted)
def accuracy():
      return score*100
```

USER

```
# User
import numpy as np
def enterData(x_user, y_user, w_user):
      x_.append(x_user)
      y_.append(y_user)
      w_.append(w_user)
      x = le.fit_transform(x_)
      y = le.fit_transform(y_)
      z = le.fit_transform(w_)
      new_user = np.array([(x[-1], y[-1], z[-1])])
      new_out = model.predict(new_user)
      return new_out[0]
```

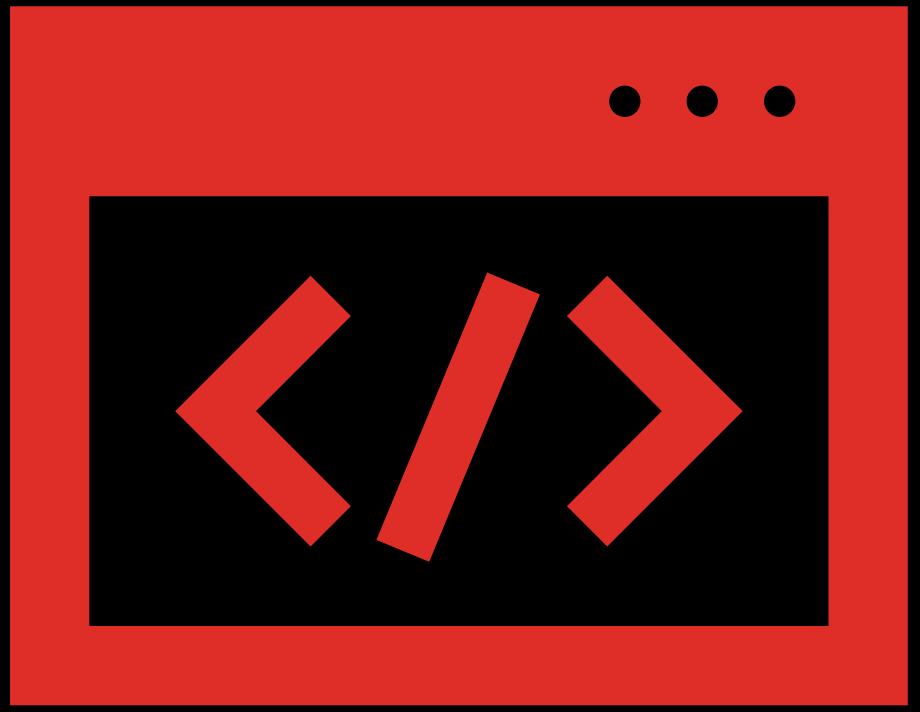
```
# Import the required Libraries
from tkinter import *
from tkinter import ttk
from Right_Left import enterData, visulize, accuracy
# Create an instance of tkinter frame
win= Tk()

# Set the geometry of tkinter frame
width = win.winfo_screenwidth()
height = win.winfo_screenheight()
win.geometry(f"{width}x{height}")
global y

x = Label(win,text="x-axis").place(x=550,y=40)
y = Label(win,text="y-axis").place(x=550,y=65)
w = Label(win,text="width").place(x=550,y=90)
h = Label(win,text="height").place(x=550,y=115)
```

SECOND

User Interface code.



```
# Initialize a label to display the user Input
label=Label(win, text="", font=("Courier 22 bold"))
label.pack()

# Create an entry widget to accept user Input
entry1= Entry(win, width= 20)
entry1.focus_set()
entry1.pack(pady=3)
# Create an entry widget to accept user Input
entry2= Entry(win, width= 20)
entry2.pack(pady=3)
# Create an entry widget to accept user Input
entry3= Entry(win, width= 20)
entry3.pack(pady=3)
# Create an entry widget to accept user Input
entry4= Entry(win, width= 20)
entry4.pack(pady=3)
```

```
def result():
    new1 = int(entry1.get())
    new2 = int(entry2.get())
    new3 = int(entry3.get())
    new4 = int(entry4.get())
    new5  = int(new3)*int(new4)
    new_out = enterData(new1, new2, new5)

    if  new_out == 0:
        print("left")
        y = "left"
        return y
    else:
        print("right")
        y = "right"
        return y

def display_text():
    string= result()
    label.configure(text=string)
```

```
# Create a button to validate Entry Widget
ttk.Button(win, text= "Okay",width= 20, command= display_text).pack(pady=20)

# Create button for visulization
ttk.Button(win, text= "Visulization?",width= 20, command= visulize).pack(pady=20)

acc = accuracy()
label=Label(win, text=f"Accuracy: {acc:.2f}", font=("Courier 22 bold"))
label.pack()

win.mainloop()
```