```python
import string
from tkinter import*
from tkinter import ttk
from tkinter.font import BOLD
from PIL import Image, ImageTk
from tkinter import messagebox
import mysql.connector
import cv2
from numpy import imag, true_divide
import os
import numpy as np

class Train:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1270x700+0+0")
        self.root.title("Train Data")

        title_label = Label(self.root,text="TRAIN DATA SET",font=("times new roman",30,"bold"),bg="steelblue",fg="black")
        title_label.place(x=0,y=0,width=1280,height=40)

        #img1
        img_top=Image.open(r"college_img\traindataaa.png")
        img_top=img_top.resize((1270,250),Image.ANTIALIAS)
        self.photoimg_top=ImageTk.PhotoImage(img_top)

        f_lbl3=Label(self.root,image=self.photoimg_top)
        f_lbl3.place(x=5,y=40,width=1270,height=250)

        #button
        b2 = Button(self.root,text="Train Data",command=self.train_classifier,cursor="hand2",font=("times new roman",26,"bold"),bg="darkblue",fg="white")
        b2.place(x = 85,y = 290,width = 1100, height = 50)

        #img2
        img_down=Image.open(r"college_img\traint.jpg")
        img_down=img_down.resize((1270,350),Image.ANTIALIAS)
        self.photoimg_down=ImageTk.PhotoImage(img_down)

        f_lbl3=Label(self.root,image=self.photoimg_down)
        f_lbl3.place(x=5,y=340,width=1270,height=350)

    def train_classifier(self):
        data_dir=("data")
        path=[os.path.join(data_dir,file) for file in os.listdir(data_dir)]

        faces=[]
        ids=[]

        for image in path:
            img=Image.open(image).convert("L")   #GRAYSCALE IMG
            imageNp=np.array(img,'uint8')
            id=int(os.path.split(image)[1].split('.')[1])

            faces.append(imageNp)
            ids.append(id)
            cv2.imshow("training",imageNp)
            cv2.waitKey(1)==13
        ids=np.array(ids)

        #============TRAIN THE CLASSIFIER AND SAVE===========
        clf=cv2.face.LBPHFaceRecognizer_create()
        clf.train(faces,ids)
        clf.write("classifier.xml")
        cv2.destroyAllWindows()
        messagebox.showinfo("Result","Training Datasets Completed")


if __name__=="__main__":
    root=Tk()
    obj=Train(root)
    root.mainloop()

```