PROFESSIONAL

Alphabet recognition-2

INSTRUCTIONS:

Goal of the Project:

In class 123, you wrote a python program which could detect (and understand) handwritten digits and display them on the screen.

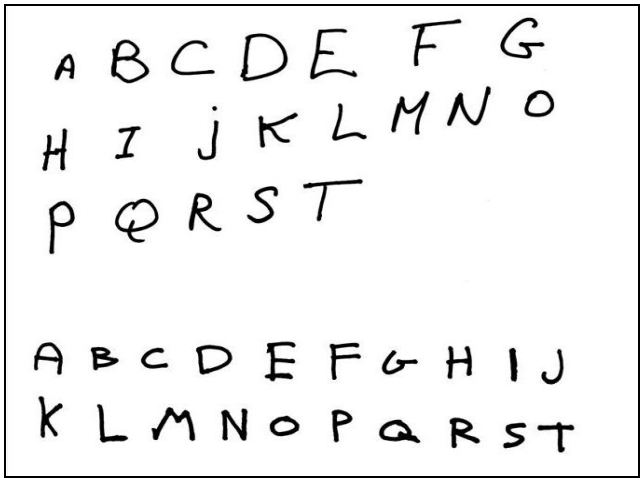
In this project, you will practice and apply what you have learned in the class to detect English Alphabets.

Story:

Lady Ada (the first programmer) and Charles Babbage (the father of modern day computers) worked together to lay the foundation of what computers are today. They exchanged a lot of letters talking about the principles of computing. All these letters are handwritten in English.

Erica, who helps in curating the history of computing has been given the task to digitize these letters. Erica has tried to use her programming skills to train the computer to detect the alphabets in these letters - but she has failed so far.

Can you show Erica - how to detect these letters?



***This is just for your reference. We expect you to apply your own creativity in the project.**

PROFESSIONAL

Alphabet recognition-2

Getting Started:

1. Open the VS Code editor.
2. Create a file called alphabet detection.

Specific Tasks to complete the Project:

1. Import all the necessary libraries to the file.
2. Load the image.npz file and read the labels. Csv file.
3. Split the data to train and test the model.
4. Fit the data into the model.
5. Code to make a prediction and print the accuracy.
6. Code to open the camera and start reading the frames.
7. Draw a box at the center of the screen. And consider only the area inside the box to detect the images.
8. Convert the image to pil format.
9. Convert to grayscale image.
10. Use clip function to limit the values between 0, 255.
11. Create a test sample and make a prediction.
12. Display the resulting frame.
13. Close the window when everything is done.

*Refer to the images given above for reference.

Submitting the Project:

1. Upload your completed project to your own github account.
2. Create a new repository named " **alphabet detection".**
3. **Upload** working code to this github repository.
4. Enable Github pages for the repository.
5. Copy the link to the github pages link in the Student Dashboard. link to the github pages link in the Student Dashboard.

Hints:

1. Code to fetch the data.

```
#Fetching the data
X = np.load('image.npz')['arr_0']
y = pd.read_csv("labels.csv")["labels"]
print(pd.Series(y).value_counts())
classes = ['A', 'B', 'C', 'D', 'E','F', 'G', 'H', 'I', 'J', "K", "L", "M",
"N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"]
nclasses = len(classes)
```

2. Code to create a box and region of interest.

```
#To only consider the area inside the box for detecting the digit
   #roi = Region Of Interest
   roi = gray[upper_left[1]:bottom_right[1],
upper_left[0]:bottom_right[0]]
```

3. Code for the functionality for image processing.

```
# convert to grayscale image - 'L' format means each pixel is
# represented by a single value from 0 to 255
image_bw = im_pil.convert('L')
image_bw_resized = image_bw.resize((28,28), Image.ANTIALIAS)
#invert the image
image_bw_resized_inverted = PIL.ImageOps.invert(image_bw_resized)
pixel_filter = 20
#converting to scalar quantity
min_pixel = np.percentile(image_bw_resized_inverted, pixel_filter)
#using clip to limit the values between 0,255
image_bw_resized_inverted_scaled = np.clip(image_bw_resized_inverted-min_pixel, 0, 255)
max_pixel = np.max(image_bw_resized_inverted)
#converting into an array
image_bw_resized_inverted_scaled = np.asarray(image_bw_resized_inverted_scaled)/max_pixel
#creating a test sample and making a prediction
test_sample = np.array(image_bw_resized_inverted_scaled).reshape(1,784)
test_pred = clf.predict(test_sample)
print("Predicted class is: ", test_pred)
```

REMEMBER... Try your best, that's more important than being correct.
After submitting your project your teacher will send you feedback on your work.

——— XXX ——— XXX ——— XXX ——— XXX ——— XXX ———