**Step 1: Data Preprocessing and Model Training**

1. Load and preprocess your letter images and organize them into classes (folders named after each letter).
2. Extract features from the images, such as resizing, converting to grayscale, flattening, etc. This is typically done using libraries like OpenCV or scikit-image.
3. Split your data into training and testing sets.
4. Train a classifier using scikit-learn's **SVC** (Support Vector Classifier) or other suitable classifier.
5. Evaluate the trained model's performance on the testing set.

from sklearn.svm import SVC

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report

# Load and preprocess images, organize into classes, and extract features

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, labels, test\_size=0.2, random\_state=42)

# Initialize and train the classifier

clf = SVC()

clf.fit(X\_train, y\_train)

# Evaluate the model on the testing set

y\_pred = clf.predict(X\_test)

print(classification\_report(y\_test, y\_pred))

**Step 2: Predict Letters in Captcha Images**

1. Load and preprocess the captcha image in a similar way as you did with the training data.
2. Pass the preprocessed image through the trained model to predict the letters.

# Load and preprocess the captcha image

captcha\_image = preprocess\_captcha\_image(captcha\_path) # Implement this function

# Predict letters using the trained classifier

predicted\_letters = clf.predict(captcha\_image)

# Join the predicted letters to form the captcha text

captcha\_text = ''.join(predicted\_letters)

print("Predicted Captcha:", captcha\_text)

Please note that the code snippets provided are for illustrative purposes and need to be adapted according to your data and requirements. You'll need to implement the **preprocess\_captcha\_image** function to preprocess the captcha image before passing it through the trained model.

Additionally, consider using more advanced neural network architectures like Convolutional Neural Networks (CNNs) for better accuracy in image classification tasks. You can use libraries like TensorFlow or PyTorch for building and training neural networks.

Top of Form