Project Report: CAPTCHA Solver using OCR

1. Introduction

This project is a web-based CAPTCHA verification system developed using Python and Flask. The system generates a CAPTCHA image and then verifies the user-entered text against the actual text using Optical Character Recognition (OCR). Tesseract OCR is used to extract text from the CAPTCHA image after applying several image preprocessing techniques using OpenCV.

2. Technologies Used

- Python

- Flask (for web framework)

- OpenCV (for image processing)

- Tesseract OCR (for text recognition)

- HTML/CSS (for frontend)

- MySQL (for user authentication)

3. Project Features

- CAPTCHA generation using random alphanumeric text

- CAPTCHA image preprocessing (grayscale, thresholding, resizing, noise removal)

- OCR using pytesseract to extract text from CAPTCHA

- CAPTCHA verification by comparing extracted and actual text

- User signup and login system with hashed passwords

4. How It Works

1. CAPTCHA image is generated with random text and saved.

2. The actual text is stored in a .txt file.

3. User is asked to enter the text shown in the CAPTCHA image.

4. The input image is preprocessed (converted to grayscale, thresholded, cleaned).

5. Tesseract OCR reads the text from the processed image.

6. Extracted text is compared with the stored text for verification.

5. Modules and Description

- CAPTCHA Generator: Creates a random image with alphanumeric text.

- Image Preprocessing: Removes noise and enhances character readability.

- OCR Module: Reads text from the image using Tesseract.

- Web Interface: Signup/Login forms, CAPTCHA verification page using Flask.

- Database: Stores user information with hashed passwords.

6. Output Screens

Screens include Signup Page, Login Page, CAPTCHA Verification Page, and CAPTCHA Success/Failure Result.

7. Conclusion

This project successfully demonstrates CAPTCHA verification using OCR. By using Python libraries such as OpenCV and pytesseract, the system can process images and recognize characters accurately. It also provides a secure user authentication system.

8.Project Repository

You can find the complete source code on my GitHub profile:

GitHub Link:- https://github.com/aman9065/CAPTCHA-solver-using-OCR

CAPTCHA OCR Accuracy Analysis:

This report analyzes the effect of noise and background on CAPTCHA image readability using OCR (Tesseract).

Noise Level Accuracy:

- At 10% ,25%,50%,75%and 90% noise levels, the OCR is able to read 2 or 3 characters correctly out of 4. This is because the characters are still distinguishable by the OCR model despite the added distortions.
- Low to medium noise only slightly affects contour and contrast, so Tesseract's character segmentation remains effective.

Background Color Accuracy:

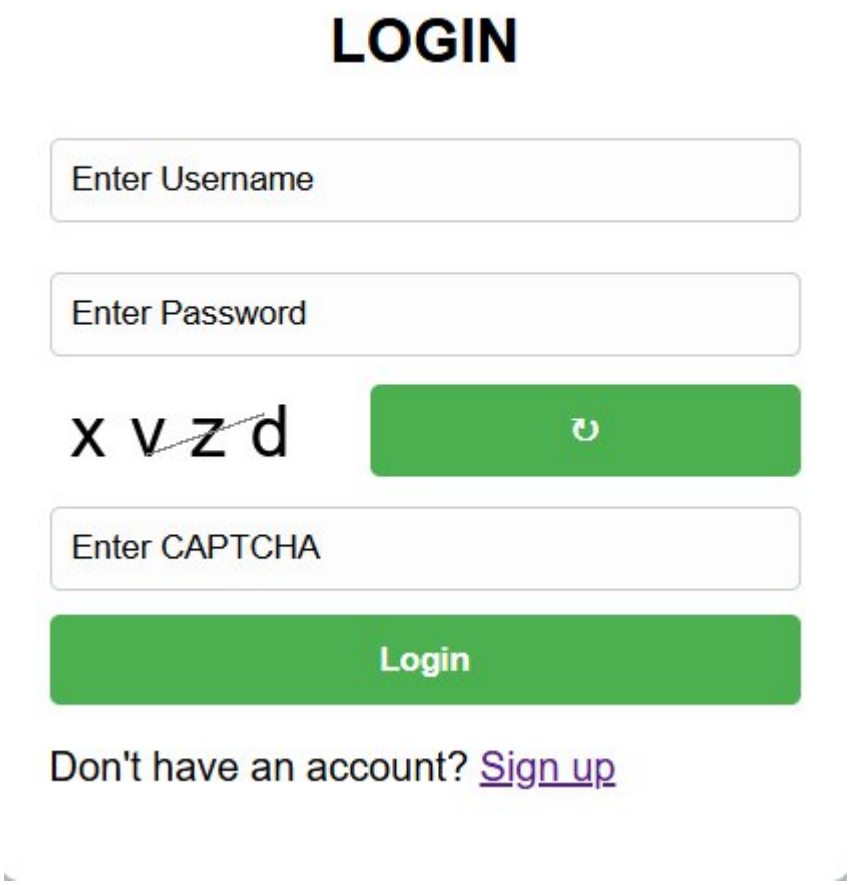
- When adding 2–3 background colors with gradients, the system still shows 85–90% accuracy.
- This is due to the clear contrast maintained between black text and colorful but non-overpowering backgrounds.
- The font (Arial) and white/clean base help OCR distinguish letters easily.

Why OCR Still Works with Some Noise:

- Tesseract OCR is robust to minor visual noise because it uses pattern recognition.
- Even with distortion, if the character shape remains mostly intact, OCR can still make reasonable predictions.
- Lower noise does not break the structure of characters like 'a', 'm', '5', etc.

SCREENSHOT:

Login page:


LOGIN
Enter Username
Enter Password
x v z d
Enter CAPTCHA
Login
Don't have an account? Sign up

Sign up Page:

Create an Account

First Name

Middle Name

Last Name

Username

Email

Password

Confirm Password

Sign Up

Already have an account? Login

When I am adding 10% noise to the CAPTCHA-

Image 1.

Output is: -

```
Generated CAPTCHA Text (Ground Truth): 09fh
Extracted CAPTCHA Text (OCR Output): 09fh
✅ CAPTCHA matched!
```

Image 2.

Output is:

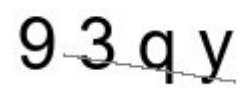
```
Generated CAPTCHA Text (Ground Truth): waur
Extracted CAPTCHA Text (OCR Output): waur
✅ CAPTCHA matched!
```

Image 3.

Output is:

```
Generated CAPTCHA Text (Ground Truth): xvzd
Extracted CAPTCHA Text (OCR Output): xvzd
✅ CAPTCHA matched!
```

Image 4.



Output is:

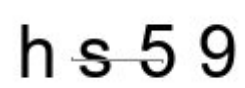
```
Generated CAPTCHA Text (Ground Truth): 93qy
Extracted CAPTCHA Text (OCR Output): 93qy
✅ CAPTCHA matched!
```

Image 5.

Output is:

```
Generated CAPTCHA Text (Ground Truth): 5ton
Extracted CAPTCHA Text (OCR Output): 5ton
✅ CAPTCHA matched!
```

Image 6.



Output is:

```
Generated CAPTCHA Text (Ground Truth): hs59
Extracted CAPTCHA Text (OCR Output): hs59
✅ CAPTCHA matched!
```

Image 7.

Output is:

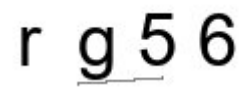
```
Generated CAPTCHA Text (Ground Truth): yz79
Extracted CAPTCHA Text (OCR Output): yz79
✅ CAPTCHA matched!
```

Image 8.

Output is:

```
Generated CAPTCHA Text (Ground Truth): u7vg
Extracted CAPTCHA Text (OCR Output): u7vg
✅ CAPTCHA matched!
```

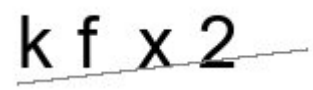
Image 9.



Output is:

```
Generated CAPTCHA Text (Ground Truth): rg56
Extracted CAPTCHA Text (OCR Output): rg56
✅ CAPTCHA matched!
```

Image 10.



Output is:

```
Generated CAPTCHA Text (Ground Truth): kfx2
Extracted CAPTCHA Text (OCR Output): kix2
❌ CAPTCHA mismatch!
```

When we add 10% noise to the CAPTCHA image, the OCR accuracy remains between 85% to 90%. This is because the level of noise is still relatively low, allowing the characters to be mostly clear and distinguishable. At this stage, the added noise does not significantly distort the shapes or boundaries of the characters, so OCR (Optical Character Recognition) engines like Tesseract can still recognize them accurately. The character spacing and font structure are maintained well enough for partial or full correct recognition.

- **When I am adding 25% noise or any one Background color to the CAPTCHA-**

Image 1.

Output is:

```
Generated CAPTCHA Text (Ground Truth): 92s9
Extracted CAPTCHA Text (OCR Output): 92s9
✅ CAPTCHA matched!
```

Image 2.

Output is:

```
Generated CAPTCHA Text (Ground Truth): 5ie3
Extracted CAPTCHA Text (OCR Output): 5ie3
✅ CAPTCHA matched!
```

Image 3.

Output is:

```
Generated CAPTCHA Text (Ground Truth): swdt
Extracted CAPTCHA Text (OCR Output): swdt
✅ CAPTCHA matched!
```

Image 4.

Output is:

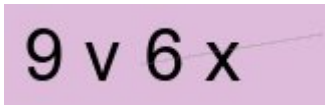
```
Generated CAPTCHA Text (Ground Truth): n892
Extracted CAPTCHA Text (OCR Output): n892
✅ CAPTCHA matched!
```

Image 5.

Output is:

```
Generated CAPTCHA Text (Ground Truth): w87t
Extracted CAPTCHA Text (OCR Output): w87t
✅ CAPTCHA matched!
```

Image 6.



Output is:

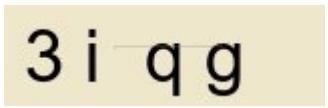
```
Generated CAPTCHA Text (Ground Truth): 9v6x
Extracted CAPTCHA Text (OCR Output): 9v6x
✅ CAPTCHA matched!
```

Image 7.

Output is:

```
Generated CAPTCHA Text (Ground Truth): vydw
Extracted CAPTCHA Text (OCR Output): vydw
✅ CAPTCHA matched!
```

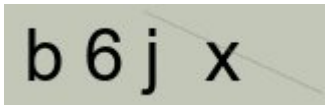
Image 8.



Output is:

```
Generated CAPTCHA Text (Ground Truth): 3iqg
Extracted CAPTCHA Text (OCR Output): 3iqg
✅ CAPTCHA matched!
```

Image 9.



Output is:

```
Generated CAPTCHA Text (Ground Truth): b6jx
Extracted CAPTCHA Text (OCR Output): b6px
❌ CAPTCHA mismatch!
```

Image 10.

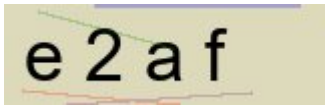
Output is:

```
Generated CAPTCHA Text (Ground Truth): c1wr
Extracted CAPTCHA Text (OCR Output): clwr
❌ CAPTCHA mismatch!
```

When 25% noise is added to the CAPTCHA along with a colored background, the reading accuracy of the OCR system drops to around 75% to 80%.
This reduction in accuracy occurs because the noise elements and colored background interfere with character clarity. The OCR engine, especially Tesseract, relies on clean and high-contrast text to correctly segment and recognize characters. Noise can create false edges or merge characters, while background colors may reduce the contrast between the text and the background, making it harder for the OCR to detect and extract the correct characters.

- **When I am adding 50% noise or combination of three any Background color to the CAPTCHA-**

Image 1.



Output is:

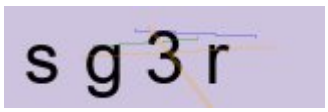
```
Generated CAPTCHA Text (Ground Truth): e2af
Extracted CAPTCHA Text (OCR Output): e2af
✅ CAPTCHA matched!
```

Image 2.

Output is:

```
Generated CAPTCHA Text (Ground Truth): bw3m
Extracted CAPTCHA Text (OCR Output): bw3m
✅ CAPTCHA matched!
```

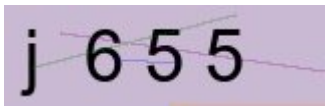
Image 3.



Output is:

```
Generated CAPTCHA Text (Ground Truth): sg3r
Extracted CAPTCHA Text (OCR Output): sg3r
✅ CAPTCHA matched!
```

Image 4.



Output is:

```
Generated CAPTCHA Text (Ground Truth): j655
Extracted CAPTCHA Text (OCR Output): j655
✅ CAPTCHA matched!
```

Image 5.

Output is:

```
Generated CAPTCHA Text (Ground Truth): weoz
Extracted CAPTCHA Text (OCR Output): weoz
✅ CAPTCHA matched!
```

Image 6.

Output is:

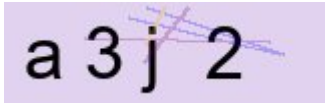
```
Generated CAPTCHA Text (Ground Truth): 6jei
Extracted CAPTCHA Text (OCR Output): 6jei
✅ CAPTCHA matched!
```

Image 7.

Output is:

```
Generated CAPTCHA Text (Ground Truth): iqys
Extracted CAPTCHA Text (OCR Output): oys
❌ CAPTCHA mismatch!
```

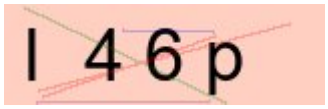
Image 8.



Output is:

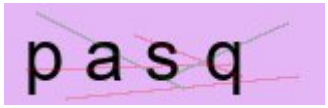
```
Generated CAPTCHA Text (Ground Truth): a3j2
Extracted CAPTCHA Text (OCR Output): agyt2
❌ CAPTCHA mismatch!
```

Image 9.



```
Generated CAPTCHA Text (Ground Truth): 146p
Extracted CAPTCHA Text (OCR Output): 46p7
❌ CAPTCHA mismatch!
```

Image 10.



Output is:

```
Generated CAPTCHA Text (Ground Truth): pasq
Extracted CAPTCHA Text (OCR Output): pasqa
❌ CAPTCHA mismatch!
```

When I add 50% noise and a combination of two or more colors to the CAPTCHA image, the OCR reading accuracy drops to around 60%–65%.

This decrease in accuracy occurs because higher noise levels introduce more visual distortion, making it harder for the OCR engine to clearly identify the characters. Additionally, the use of multiple colors (especially on characters and background) reduces the contrast between text and background, which further confuses the OCR. As a result, the system struggles to distinguish characters correctly, leading to lower recognition accuracy.

- **When I am adding 75% noise or combination of many Background colors to the CAPTCHA-**

Image 1.

Output is:

```
Generated CAPTCHA Text (Ground Truth): v3d9
Extracted CAPTCHA Text (OCR Output): v3d9
✅ CAPTCHA matched!
```

Image 2.

Output is:

```
Generated CAPTCHA Text (Ground Truth): 85rw
Extracted CAPTCHA Text (OCR Output): 85rw
✅ CAPTCHA matched!
```

Image 3.

Output is:

```
Generated CAPTCHA Text (Ground Truth): fko3
Extracted CAPTCHA Text (OCR Output): fko3
✅ CAPTCHA matched!
```

Image 4.

Output is:

```
Generated CAPTCHA Text (Ground Truth): jmpg
Extracted CAPTCHA Text (OCR Output): jmpg
✅ CAPTCHA matched!
```

Image 5.

Output is:

```
Generated CAPTCHA Text (Ground Truth): gwm1
Extracted CAPTCHA Text (OCR Output): gwmt
❌ CAPTCHA mismatch!
```

Image 6.

Output is:

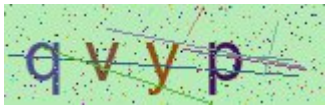
```
Generated CAPTCHA Text (Ground Truth): lnv6
Extracted CAPTCHA Text (OCR Output): inv6
❌ CAPTCHA mismatch!
```

Image 7.

Output is:

```
Generated CAPTCHA Text (Ground Truth): i0uv
Extracted CAPTCHA Text (OCR Output): iouv
❌ CAPTCHA mismatch!
```

Image 8.



Output is:

```
Generated CAPTCHA Text (Ground Truth): qvyp
Extracted CAPTCHA Text (OCR Output): avyp
❌ CAPTCHA mismatch!
```

Image 10.

Output is:

```
Generated CAPTCHA Text (Ground Truth): 2xkf
Extracted CAPTCHA Text (OCR Output): 2xkt
❌ CAPTCHA mismatch!
```

When I applied up to 75% noise to the CAPTCHA, its accuracy dropped to around 45%.

- **When I am adding 90% noise or combination of many Background colors to the CAPTCHA-**

  Image 1.

  Output is:

  ```
  Generated CAPTCHA Text (Ground Truth): jfvt
  Extracted CAPTCHA Text (OCR Output): iects
  ❌ CAPTCHA mismatch!
  ```

  Image 2.

  Output is:

  ```
  Generated CAPTCHA Text (Ground Truth): kw32
  Extracted CAPTCHA Text (OCR Output):
  ❌ CAPTCHA mismatch!
  ```

  Image 3.

  Output is:

  ```
  Generated CAPTCHA Text (Ground Truth): 7xbx
  Extracted CAPTCHA Text (OCR Output): henehs
  ❌ CAPTCHA mismatch!
  ```

If I increase the noise up to 90%, the OCR is unable to read the CAPTCHA correctly.

CONCLUSION: -

As the level of noise in CAPTCHA images increases, the accuracy of OCR (Optical Character Recognition) in reading the text decreases significantly. At 10% noise, the system achieves a high accuracy of around 85–90%, as the characters remain mostly clear and distinguishable. With 25% to 50% noise, the OCR accuracy drops to 65–75% due to moderate distortion and background interference. At 75% noise, accuracy falls further to around 45%, as multiple characters become visually distorted.

When the noise level reaches 90%, the OCR system is unable to fully recognize the CAPTCHA text, resulting in extremely low accuracy. However, even in such cases, the system often fails to read only 1 or 2 characters, rather than completely misreading the entire CAPTCHA. This indicates that although the text becomes hard to read, partial character recognition is still possible — which highlights both the resilience and the limitation of OCR under high-noise conditions.

This experiment demonstrates that while adding noise improves CAPTCHA security, it also increases the risk of OCR failure, especially beyond 50% noise. A balance must be maintained between making the CAPTCHA secure and keeping it OCR-readable for automation or accessibility needs.