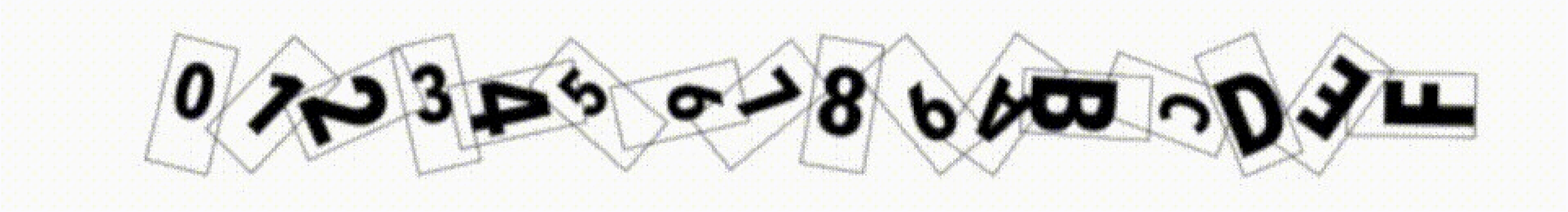
TABLE OF CONTENT

ABSTRACT

The rapid growth of network technology has improved various sectors but also introduced challenges in secure authentication. Traditional methods like passwords are increasingly vulnerable, while CAPTCHA systems face threats from advanced AI algorithms. This paper presents a novel CAPTCHA recognition system using Cellular Neural Networks (CNNs), achieving 85.20% accuracy across diverse CAPTCHA formats. A comparative analysis with machine learning techniques highlights the system's effectiveness and underscores the need for innovative solutions to address evolving AI-driven security threats.



INTRODUCTION

- CAPTCHAs may be referred to those infuriating images containing the text that needs to be typed in before a person can access a particular website.
- As the name suggests it is a way to avert the computer to fill out the form on its own, automatically.

APPLICATION

1. Preventing Fake Accounts
2. Blocking Spam
3. Password Protection

CNN

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data.

Captcha recognition with deep Convolutional Neural Networks (CNNs) is an advanced application of deep learning for automatically identifying and solving captchas. A deep CNN model can be trained to recognize patterns in captcha images and bypass this challenge

01 Feature Extraction

CNNs process CAPTCHA images by detecting and extracting essential features, such as edges, textures, and shapes, that help differentiate between characters or objects within the image.

02 High Accuracy and Speed

CNNs can achieve impressive recognition accuracy, even when dealing with complex distortions and background noise. These models excel at solving CAPTCHAs that were designed to be resistant to traditional machine learning algorithms.

03 End-to-End Recognition

CNNs can automatically interpret raw image inputs and classify them with minimal pre-processing, making them highly efficient for CAPTCHA recognition tasks.

LITERATURE SURVEY

Author	Year	Title	Methodoloyg	Drawback
Department of Natural Science and Mathematics, King's College London WC2R 2LS, United Kingdom	2021	4-Character Captcha Recognition Based on the CNN	Image Preprocessing and CNN	Cannot recognize connected or overlapped characters
Weihang Ding, Yuxin Luo, Yifeng Lin, Yuer Yang, Siwei Lian	2023	VeriBypasser: An automatic image verification code recognition system based on CNN	Deep learning model training based on CNN	VeriBypasser could not sufficiently ensure that every automated login with such image verification codes could be successful - Lacks Accuracy

Author	Year	Title	Methodoloyg	Drawback
Soumen Sinha, Mohammed Imaz Surve	2023	CAPTCHA Recognition And Analysis Using Custom Based CNN Model- Capsecure	Deep Learning	The model's performance can be inconsistent, leading to failed CAPTCHA bypass attempts and unreliable results across various verification scenarios.
Swati Mali, Amisha Waghela, Girish Thatte	2021	DeCaptcha: Cracking captcha using Deep Learning Techniques	Bidirectional LSTM	Cannot solve CAPTCHA of variable lengths and CAPTCHA of different languages

Author	Year	Title	Methodoloyg	Drawback
Yujin Shu, Yongjin Xu School of Mechatronic Engineering and Automation, Shanghai University Shanghai, China	2019	End-to-End Captcha Recognition Using Deep CNN-RNN Network	ResNet-GRU model	Not applicable for desired lengths of string pattern and use residual network structure
Haolin Yang	2020	Captcha Recognition using convolutional neural networks with low structural complexity	CNN	Low-structural-complexity CNNs may struggle with complex CAPTCHA designs, leading to lower accuracy and potential misclassifications. Adversarial attacks and overfitting.

PROBLEM STATEMENT

The problem is to develop a deep learning model capable of recognizing and bypassing CAPTCHAs by extracting text from images. CAPTCHA images contain alphanumeric characters (A-Z and 0-9) with a maximum length of 5 characters, often distorted and accompanied by background noise to prevent automated recognition. The goal is to preprocess these images, encode the characters, and train a Convolutional Neural Network (CNN) to accurately predict the text, overcoming challenges such as distortion, noise, and the complexity of multi-label classification. This work not only highlights the vulnerabilities in CAPTCHA systems but also emphasizes the importance of improving their design to enhance security against automated attacks.

OBJECTIVES

01 **Develop CAPTCHA Recognition System**

Create a system capable of recognizing 4-character CAPTCHA images using a Convolutional Neural Network (CNN) for enhanced accuracy and efficiency.

02 **Enhance Image Preprocessing**

Implement advanced preprocessing steps such as grayscale conversion, binarization, and noise elimination to improve segmentation and recognition accuracy.

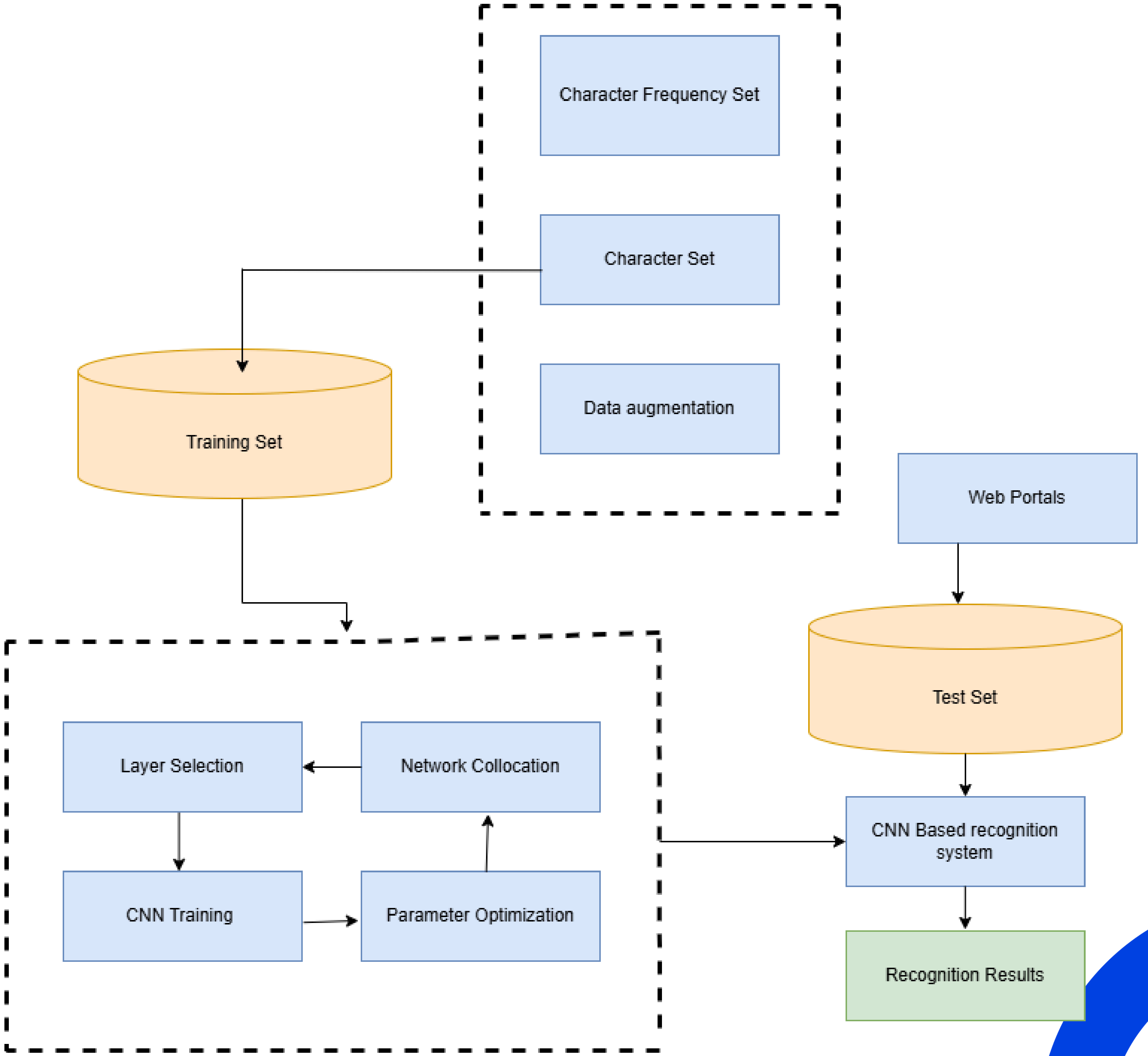
03 **Evaluate Model Performance**

Test the recognition system's effectiveness using single-character and 4-character CAPTCHA datasets, focusing on accuracy and precision metrics.

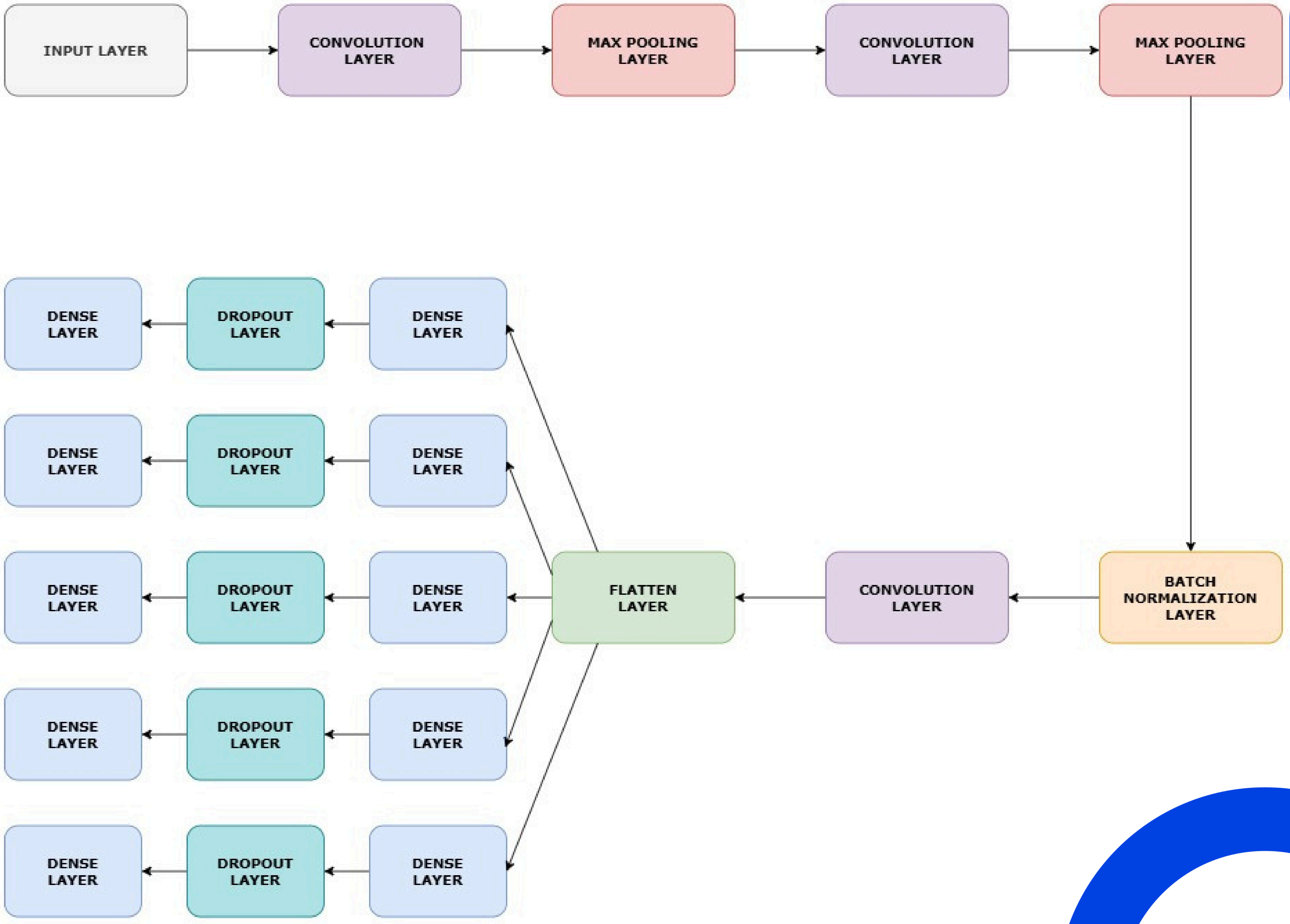
04 **Address Security Vulnerabilities**

Analyze the system’s performance to identify and address weaknesses, such as its inability to recognize connected or overlapping characters.

IMPLEMENTATION



IMPLEMENTATION



IMPLEMENTATION STEPS

1. Data Preparation:

   - Use the Python CAPTCHA library to generate 554,707 single-character CAPTCHA images (60x60 pixels) for training/testing and 1,351 4-character CAPTCHA images (160x60 pixels) for evaluation.

IMPLEMENTATION STEPS

Image Preprocessing:

1. Grayscale Conversion: Reduce RGB dimensions to simplify processing.
2. Binarization: Apply a threshold (e.g., 200) to convert grayscale images to black-and-white, enhancing clarity.
3. Noise Removal: Eliminate extraneous points using a neighborhood-based algorithm to clean the image while retaining character integrity.
4. Segmentation: Divide 4-character CAPTCHAs into individual character sub-images using depth-first search and boundary detection algorithms.

Figure 1. Original captcha image

Figure 2. Grayscale captcha image

Figure 3. Binarized captcha image

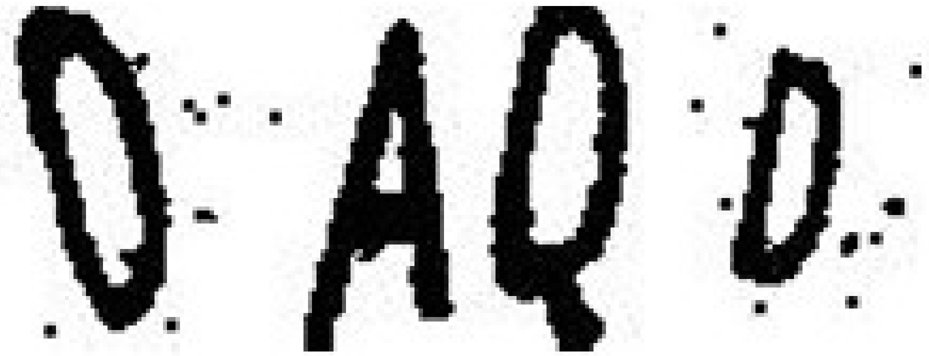


Figure 4. Capthca after noise elimination

Figure 7. Segmented character images

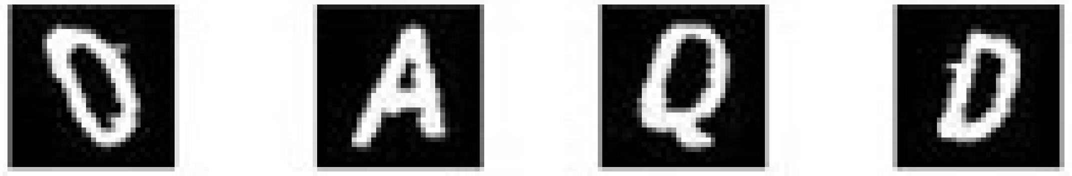
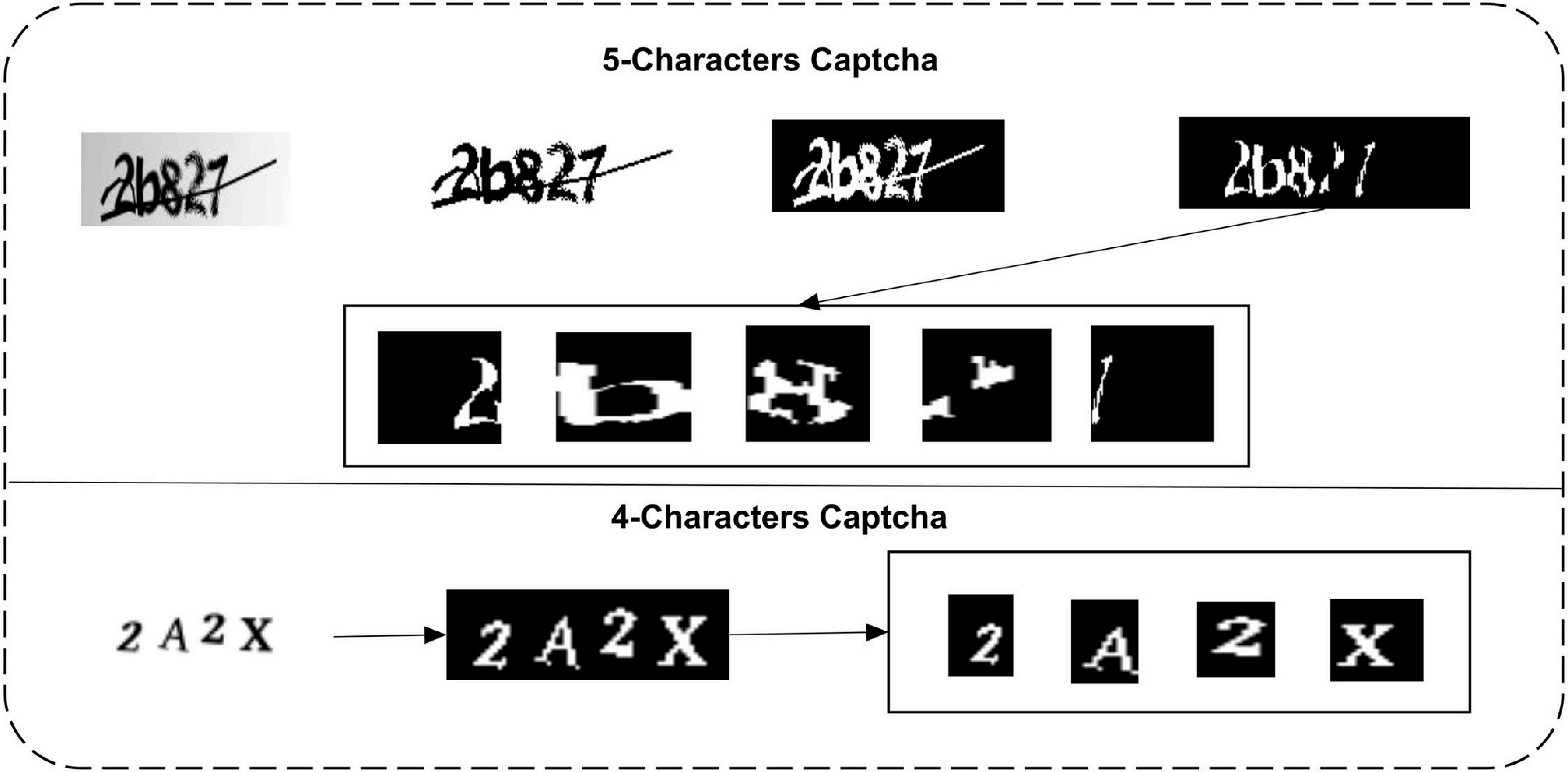


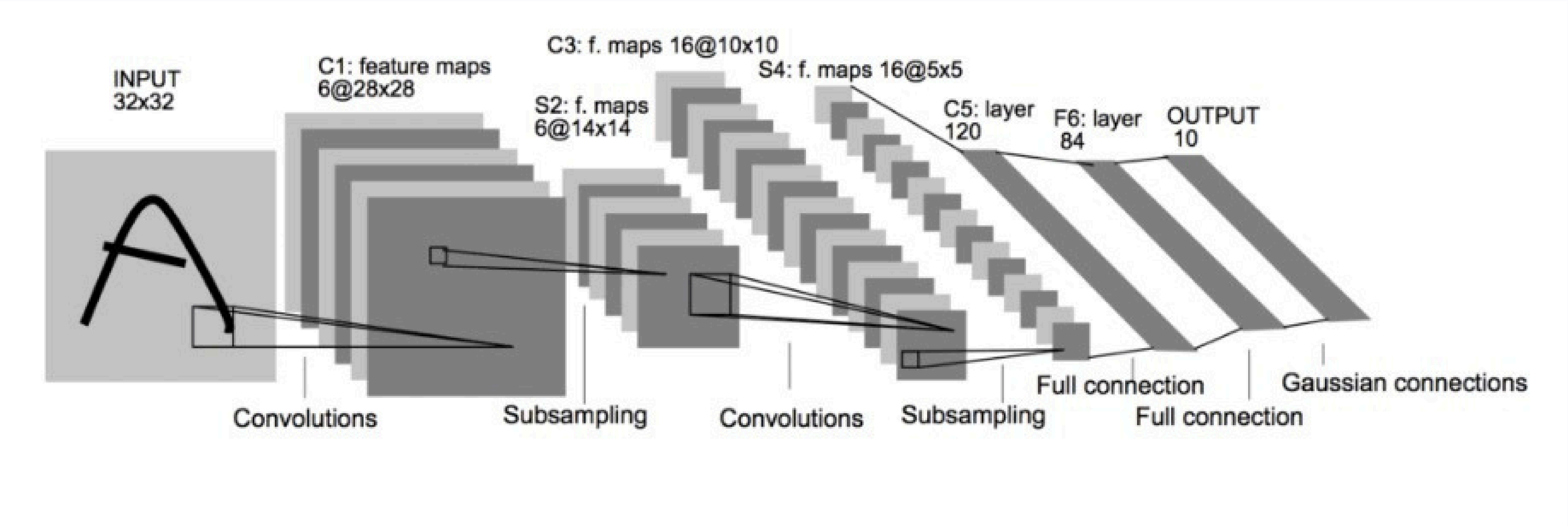
Figure 8. Character images after resizing and color reversing


5-Characters Captcha
4-Characters Captcha

IMPLEMENTATION STEPS

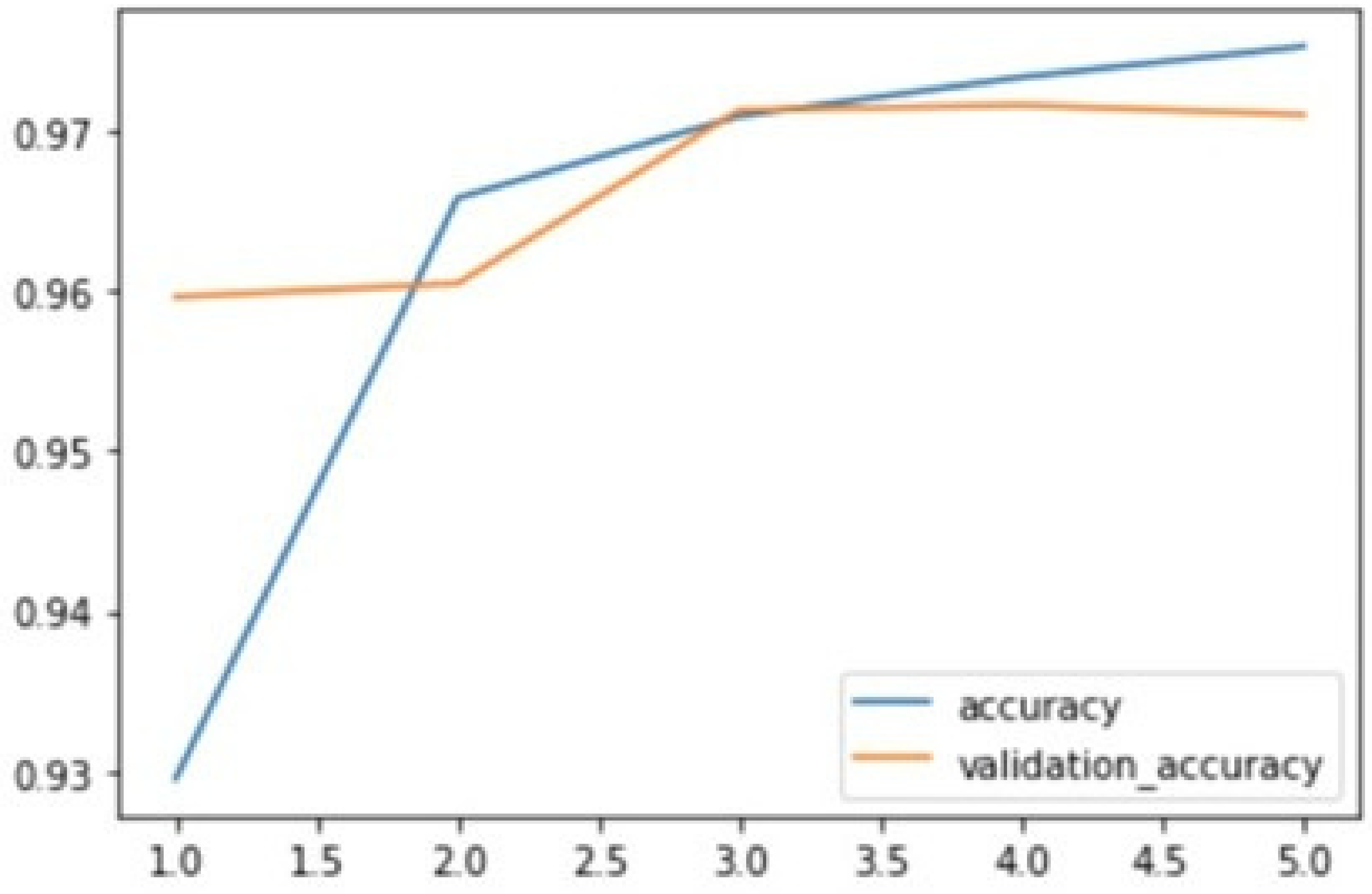
Model Design and Training:

1. Build a CNN with nine layers, including three convolutional layers, two max-pooling layers, one flattening layer, and one fully connected layer.
2. Train the model with 443,765 single-character images for five epochs using the Adam optimizer and sparse categorical cross-entropy loss function.



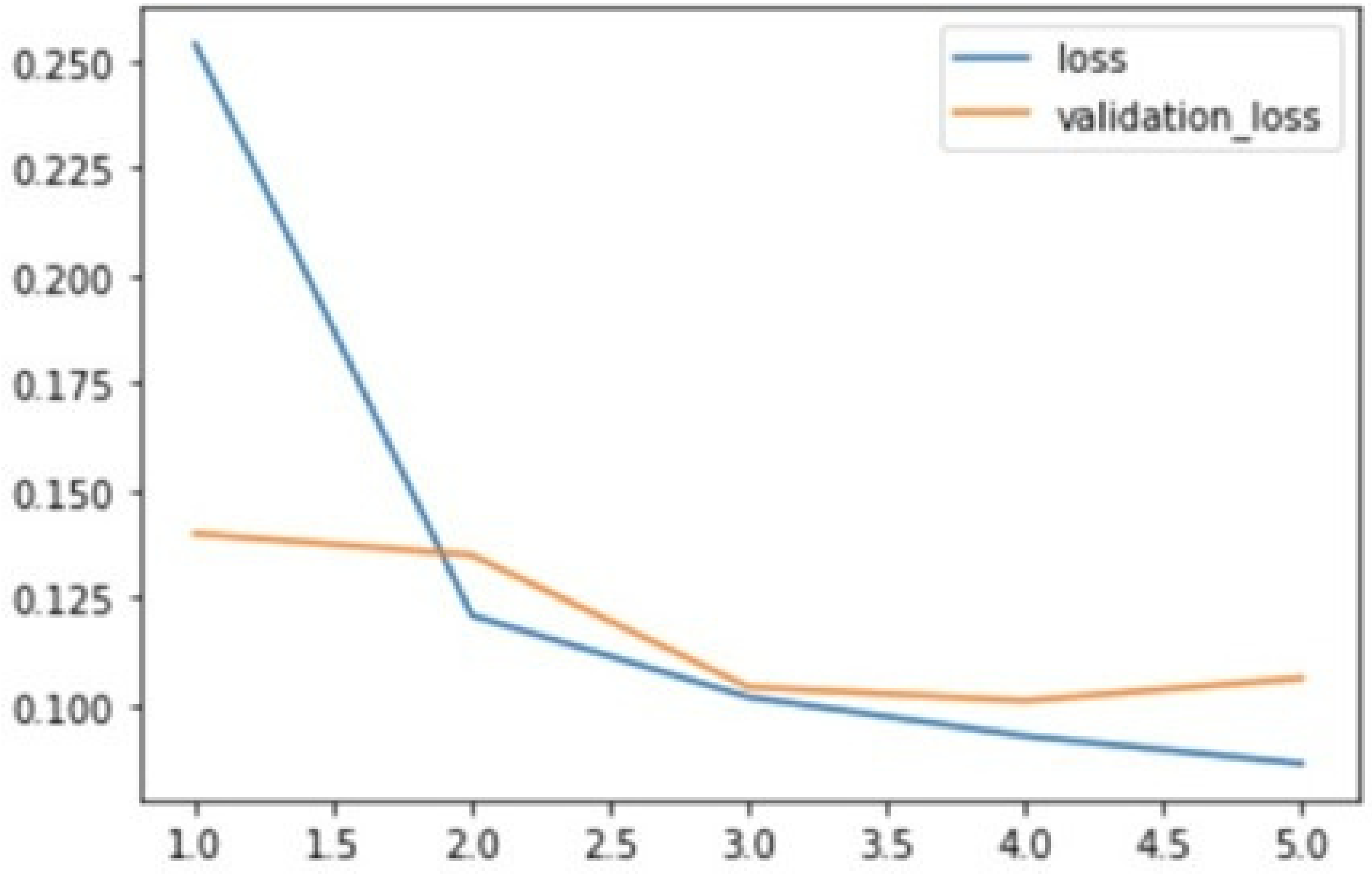
INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

RESULT



Accuracy vs. Epochs

Illustrates the steady improvement in training and validation accuracy over five epochs.



Loss vs. Epochs

Demonstrates the reduction in training and validation loss over epochs.

CONCLUSION

Model Overview

- The CAPTCHA recognition system utilizes improved image processing and segmentation algorithms, combined with a convolutional neural network (CNN).

Input and Preprocessing

- The model takes a 4-character CAPTCHA image, processes it, and segments it into four single-character sub-images for individual recognition.

Limitation

- The model demonstrates high accuracy in recognizing CAPTCHA characters.
- It cannot recognize connected or overlapped characters effectively

Future Work

- The researcher plans to apply the model to different types of CAPTCHAs and focus on improving segmentation techniques for connected and overlapped characters in future studies.

REFERENCES

- S.G. Huang, L. Zhang, P.P. Wang, and H.W. Han, "A CAPTCHA Recognition Algorithm Based on Holistic Verification," in 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, Beijing, China, 2011, pp. 525-528.

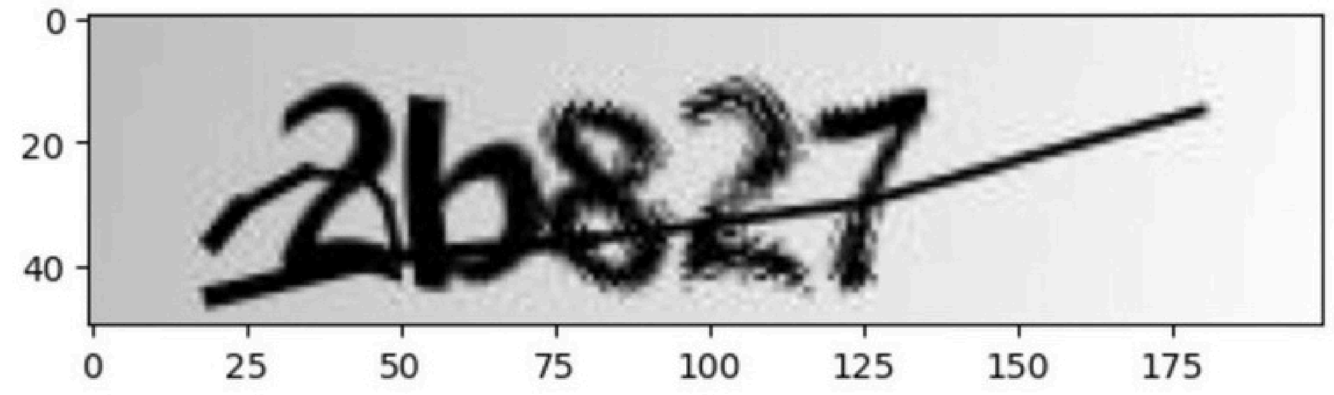
- A. Chandavale and A. Sapkal, "An Improved Adaptive Noise Reduction for Secured CAPTCHA," in 2011 4th International Conference on Emerging Trends in Engineering and Technology (ICETET), Port Louis, Mauritius, 2011, pp. 12- 17.

- H. Shin et al., "Deep Convolutional Neural Networks for Computer Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," in IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1285-1298, May 2016, doi: 10.1109/TMI.2016.252816 Huilgol P. Precision vs. Recall–An Intuitive Guide for Every Machine Learning Person[J]. 2020.

- Morvant, E., Koço, S., & Ralaivola, L. PAC-Bayesian generalization bound on confusion matrix for multi-class classification. arXiv preprint arXiv:1202.6228, 2012.

SNAPSHOTS

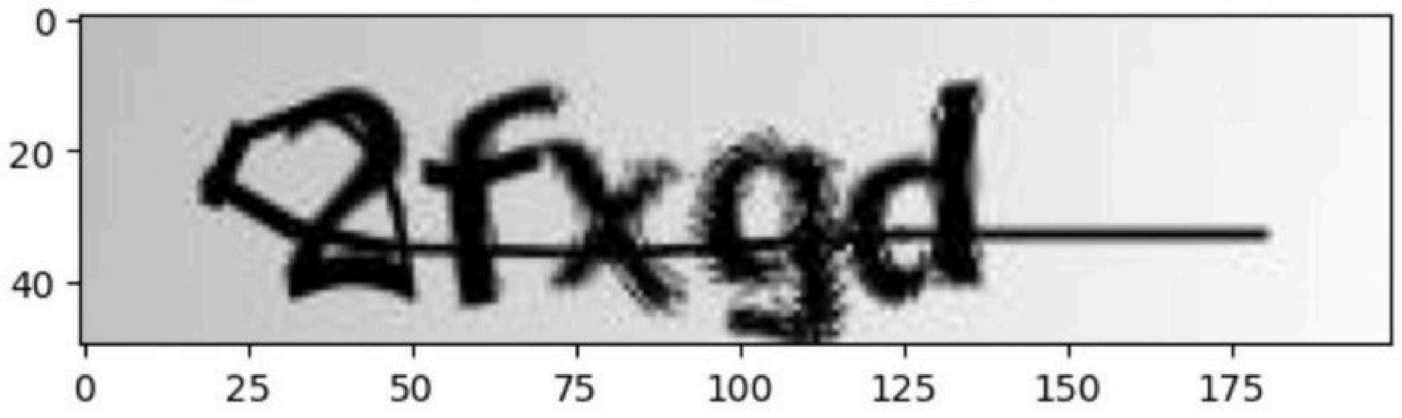
```
img=plt.imread("/Users/apeksh/Desktop/samples/2b827.png")
plt.imshow(img)
plt.show()
```




```
print("predicted Captcha=",predict("/Users/apeksh/Desktop/samples/2b827.png"))
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
predicted Captcha= 2b827
```

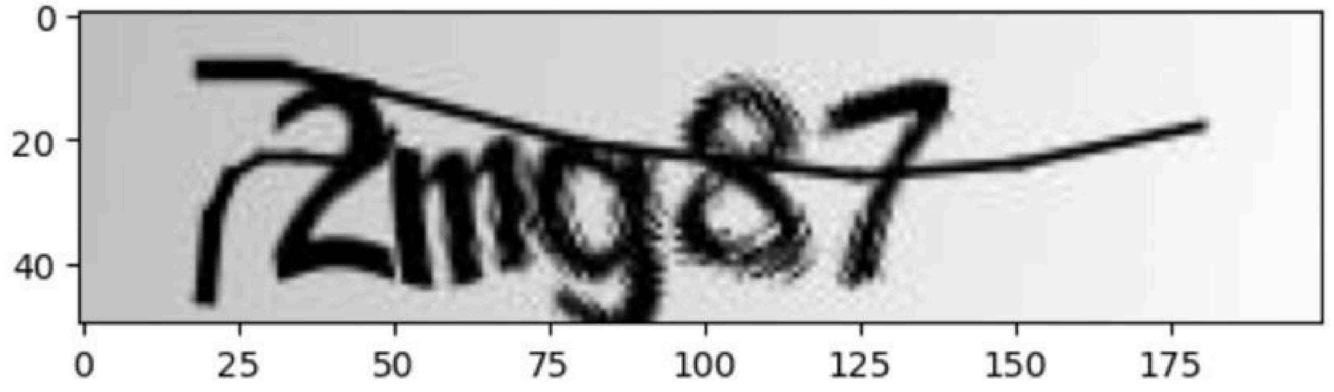
```
img=plt.imread("/Users/apeksh/Desktop/samples/2fxgd.png")
plt.imshow(img)
plt.show()
```




```
print("predicted Captcha=",predict("/Users/apeksh/Desktop/samples/2fxgd.png"))
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step
predicted Captcha= 2fxgd
```

```
[92]: img=plt.imread("/Users/apeksh/Desktop/samples/2mg87.png")
plt.imshow(img)
plt.show()
```




```
[93]: print("predicted Captcha=",predict("/Users/apeksh/Desktop/samples/2mg87.png"))
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
predicted Captcha= 2mg87
```