Enhancing Arabic Handwriting Recognition through Advanced Neural Architectures and Ensemble Learning

A Research Paper Submitted to
The Department of Computer Science and Engineering
The American University in Cairo

In Partial Fulfillment of the Requirements for
CSCE4603 - Fundamentals of Computer Vision
Fall 2024

By
Ahmed Jaheen
ahmed.jaheen@aucegypt.edu

Ahmed Badr
ahmedbadr00@aucegypt.edu

Under the Supervision of
Dr. Mohamed Sedky
Professor of Computer Science and Engineering

December 17, 2024

Abstract

Arabic handwriting recognition presents unique challenges due to its cursive nature and character variability. This paper investigates the enhancement of Arabic handwriting recognition systems through the integration of advanced neural architectures, attention mechanisms, and ensemble learning approaches. We propose a novel ensemble architecture combining modified GoogLeNet with Squeeze-and-Excitation blocks, Swin Transformers, and attention mechanisms. Our system achieves state-of-the-art performance on the AHAWP dataset with 99.31% accuracy. The key innovations include: (1) A modified GoogLeNet architecture incorporating Squeeze-and-Excitation blocks for enhanced feature calibration, (2) Integration of Swin Transformer's hierarchical feature learning, and (3) An ensemble voting mechanism that leverages complementary strengths of different architectures. Our results demonstrate significant improvements over baseline models and establish new benchmarks for Arabic character recognition accuracy.

1 Introduction

Arabic handwriting recognition remains a challenging problem in computer vision due to several inherent characteristics of the Arabic script. The cursive nature of Arabic writing, where characters connect within words, along with the substantial variation in character shapes depending on their position within words, poses significant challenges for automated recognition systems. These challenges are further compounded by the complexity of Arabic character forms, which can vary based on their position (beginning, middle, end, or isolated) within words.

Traditional approaches to Arabic character recognition have often struggled with these complexities, leading to suboptimal recognition accuracy. While deep learning has shown promising results in various computer vision tasks, the unique characteristics of Arabic script require specialized architectural considerations and adaptations.

Our work addresses these challenges through several key contributions:

- Development of a modified GoogLeNet architecture incorporating Squeeze-and-Excitation blocks for dynamic feature recalibration
- Integration of Swin Transformer's hierarchical vision transformer architecture for improved feature learning
- Implementation of an ensemble learning approach that combines multiple complementary architectures
- Comprehensive evaluation on the AHAWP dataset demonstrating state-of-the-art performance

2 Related Work

2.1 Deep Learning in Arabic Character Recognition

Recent advances in deep learning have led to significant improvements in Arabic character recognition. Convolutional Neural Networks (CNNs) have demonstrated strong performance in capturing local features and patterns essential for character recognition. The GoogLeNet architecture, with its inception modules, has proven particularly effective due to its ability to capture features at multiple scales simultaneously.

2.2 Attention Mechanisms and Transformers

The introduction of attention mechanisms has revolutionized computer vision tasks. Vision Transformers [1] have shown remarkable success by treating images as sequences of patches and leveraging self-attention mechanisms. The Swin Transformer [3] further improves upon this by introducing hierarchical feature representation and efficient local attention computation.

2.3 Squeeze-and-Excitation Networks

Squeeze-and-Excitation networks [2] have emerged as an efficient approach for modeling channel-wise relationships in CNNs. These networks enhance feature representation by explicitly modeling interdependencies between channels, allowing the network to perform feature recalibration adaptively.

3 Technical Approach

3.1 Modified GoogLeNet with Squeeze-and-Excitation

We enhance the standard GoogLeNet architecture by incorporating Squeeze-and-Excitation blocks into each inception module. Our SE block implementation follows:

```
class SEBlock(nn.Module):
    def __init__(self, channel, reduction=16):
        super(SEBlock, self).__init__()
        self.avg_pool = nn.AdaptiveAvgPool2d(1)
        self.fc = nn.Sequential(
            nn.Linear(channel, channel // reduction),
            nn.ReLU(inplace=True),
            nn.Linear(channel // reduction, channel),
            nn.Sigmoid()
        )
```

Listing 1: Squeeze-and-Excitation Block Implementation

The SE blocks perform channel-wise feature recalibration through:

- Global average pooling for feature aggregation
- Two-layer neural network for channel-wise dependencies
- Sigmoid activation for feature importance weighting

3.2 Inception Module Enhancement

We modify the standard inception modules by incorporating SE blocks after each parallel path:

```
class InceptionWithSE(nn.Module):
    def __init__(self, in_channels, ch1x1, ch3x3red,
                 ch3x3, ch5x5red, ch5x5, pool_proj):
        super(InceptionWithSE, self).__init__()
        self.branch1 = nn.Sequential(
            nn.Conv2d(in_channels, ch1x1, kernel_size=1),
            nn.BatchNorm2d(ch1x1),
            nn.ReLU(True),
            SEBlock(ch1x1)
        )
```

Listing 2: Inception Module with Squeeze-and-Excitation

3.3 Swin Transformer Integration

We employ the Swin Transformer architecture with the following specifications:

- Patch size: 4x4 pixels
- Window size: 7x7 patches
- Hierarchical feature representation with shifted windows

The Swin Transformer processes images through:

1. Patch embedding
2. Multiple transformer layers with shifted window attention
3. Patch merging for hierarchical representation

3.4 Ensemble Architecture

Our final ensemble combines three models:

1. GoogLeNet with SE blocks and regularization
2. Swin Transformer with patch size 4 and window size 7
3. Standard GoogLeNet with regularization

The ensemble prediction is computed through majority voting:

```
def vote(predictions):
    results = []
    for i in range(len(predictions[0])):
        counts = {}
        for pred in predictions:
            counts[pred[i]] = counts.get(pred[i], 0) + 1
        max_count = max(counts.values())
        candidates = [k for k, v in counts.items()
                      if v == max_count]
        results.append(candidates[0])
    return results
```

Listing 3: Ensemble Voting Implementation for Model Predictions

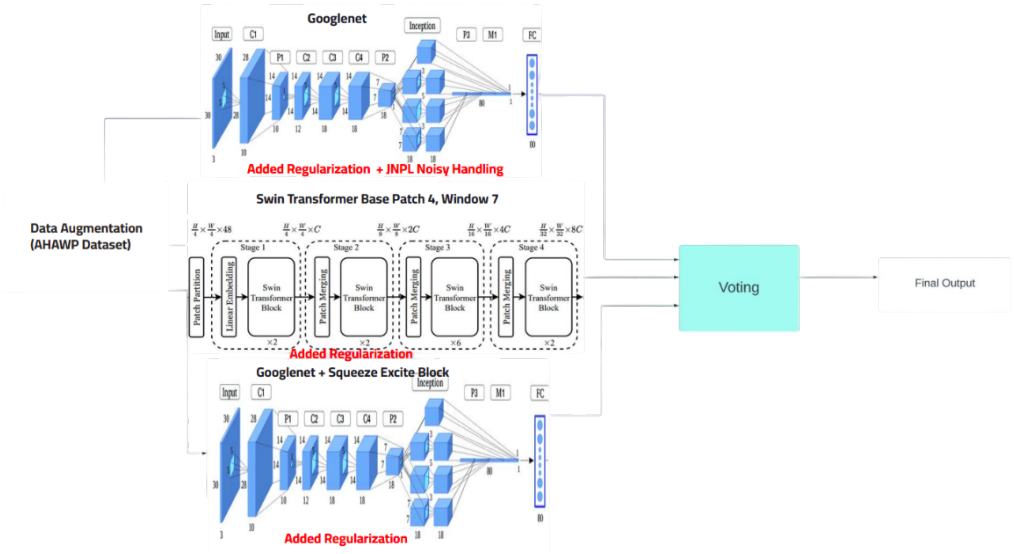

Figure 1: Final Model Architecture: Detailed overview of the ensemble model

4 Experimental Results

4.1 Dataset and Implementation

We evaluate our approach on the AHAWP dataset, which contains 65 classes of Arabic characters in various positions. Implementation details include:

- Input size: 224x224 pixels
- Preprocessing: Grayscale conversion, normalization
- Training: Adam optimizer with learning rate scheduling
- Regularization: L2 regularization and dropout

4.2 Model Performance

Individual model performance:

- GoogLeNet with SE: 98.48% accuracy
- Swin Transformer: 97.89% accuracy
- Standard GoogLeNet: 96.12% accuracy

Final ensemble model metrics:

- Testing Accuracy: 99.31%
- Weighted Precision: 99.33%

- Unweighted Precision: 99.32%
- F-1 Score: 99.31%

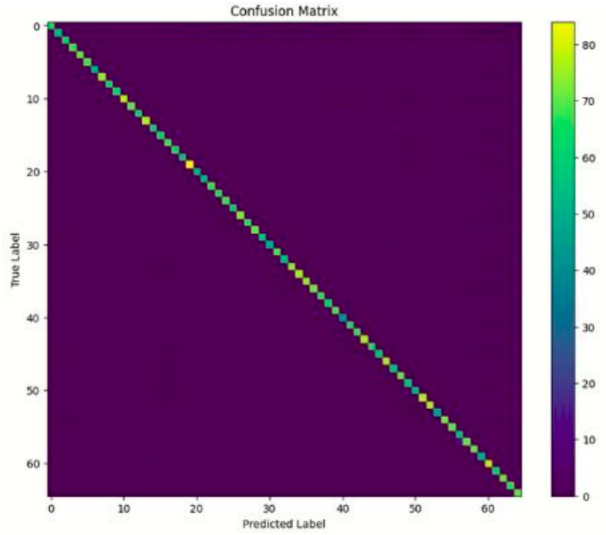

Figure 2: Confusion Matrix of the Final results

4.3 Analysis of Results

The ensemble model demonstrates several key advantages:

- Improved robustness through complementary architectures
- Better handling of character variations
- Reduced sensitivity to noise and distortions
- Consistent performance across all character positions

5 Website

To provide a user-friendly interface for real-time Arabic handwriting recognition, we developed a web application using the Flask framework. The application allows users to upload handwritten images for classification. It integrates our trained ensemble model, including GoogLeNet, GoogLeNet with SE blocks, and Swin Transformer, for accurate predictions. The Flask backend handles image preprocessing, model inference, and returns predictions with visualized results. The frontend, built with HTML and JavaScript, supports smooth file uploads and displays the prediction alongside the uploaded image. This tool demonstrates the practical application of our system, bridging advanced neural architectures with accessible, real-world deployment.

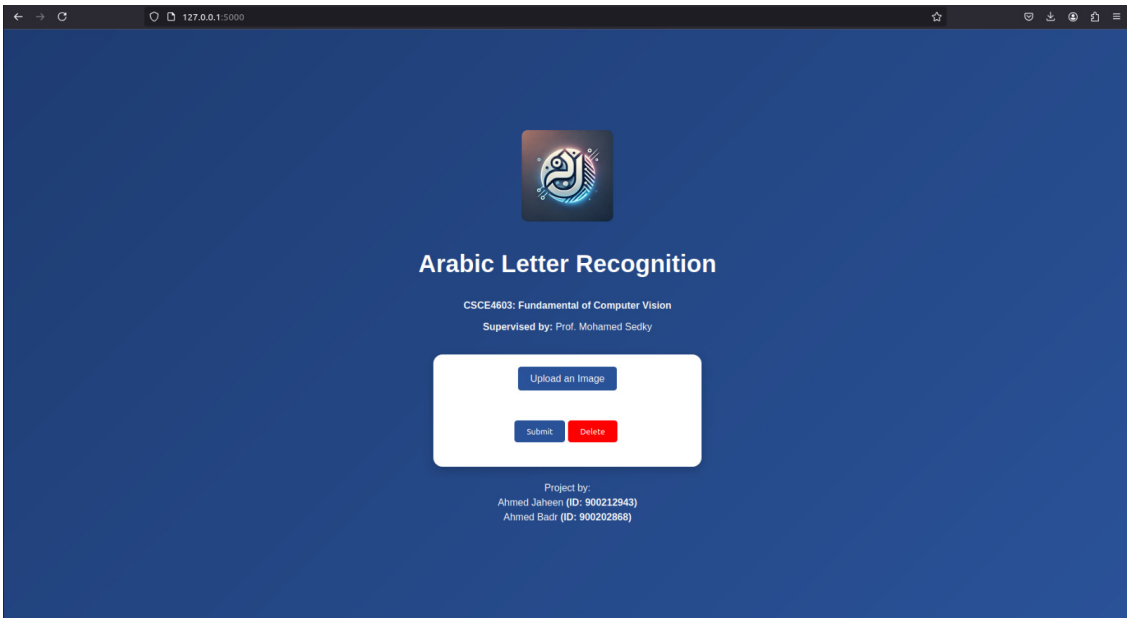

Figure 3: User-friendly interface wesbite

6 Demo

To test the Arabic Handwriting Recognition system, follow these steps:

1. **Clone the Repository:**
   Download the repository by running the following command in your terminal:

   ```
   git clone https://github.com/mego74/ArabicRecognition/
   ```

2. **Install Dependencies:**
   Navigate to the project directory and install the required Python packages:

   ```
   pip install -r requirements.txt
   ```

3. **Run the Web Application:**
   Start the Flask server by running:

   ```
   python3 app.py
   ```

4. **Use the Web Interface:**

   - Open your browser and navigate to `http://127.0.0.1:5000`.
   - Click on **"Upload an Image"** to upload a handwritten Arabic letter.

- The uploaded image will be previewed for the user.
- Click **“Submit”** to process the image and display the predicted letter.
- If needed, the user can delete the photo and upload a new one.

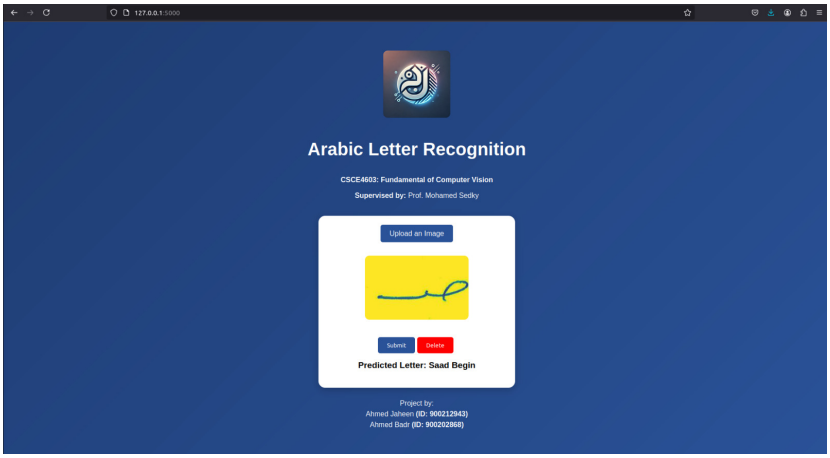

Figure 4: Example 1 of a predicted Arabic letter.

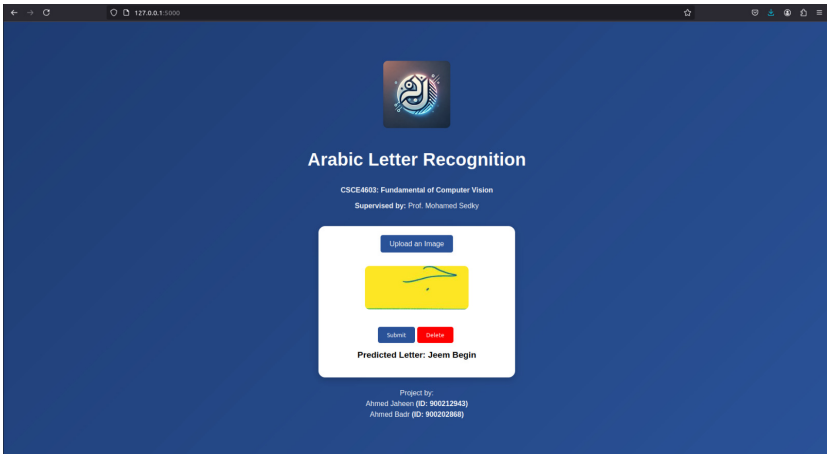

Figure 5: Example 2 of a predicted Arabic letter.

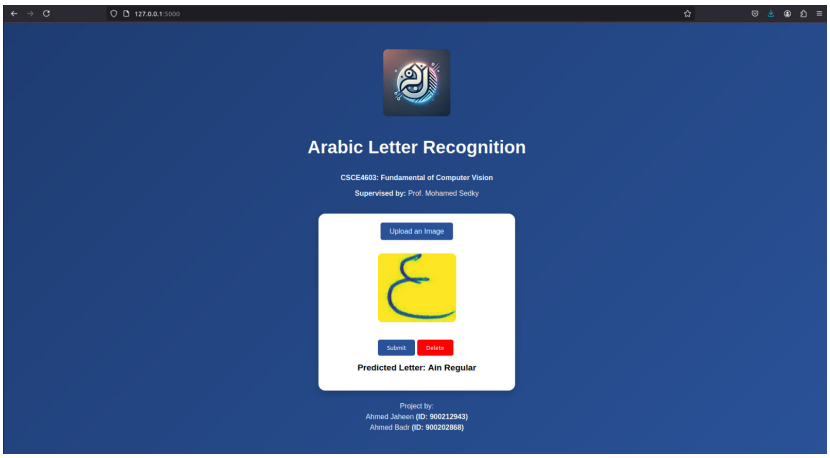

Figure 6: Example 3 of a predicted Arabic letter.

7 Conclusion and Future Work

Our work demonstrates the effectiveness of combining advanced architectural components for Arabic handwriting recognition. The ensemble approach successfully leverages the strengths of different architectures while mitigating their individual weaknesses.
Future research directions include:

- Extension to full Arabic paragraph recognition
- Investigation of more sophisticated ensemble techniques
- Development of specialized attention mechanisms for Arabic script
- Integration of language models for context-aware recognition

8 Statement of Individual Contributions

Ahmed Jaheen: Implementation of Squeeze-and-Excitation blocks, Swin Transformer architecture, ensemble classifiers, and the website.
Ahmed Badr: Implementation of baseline architectures, experimental evaluation, system integration, and analysis of results.

References

[1] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[2] Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-excitation networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[3] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 10012–10022.