ARABIC HANDWRITING RECOGNITION

Practical Use of AI

TABLE OF CONTENTS

CHAPTER #1

INTRODUCTION

INTRODUCTION

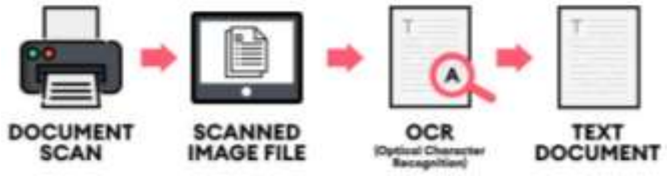
Project Idea:

Our project's idea "Arabic Handwriting Recognition (AHR)" revolves around researching and developing a system that's able to recognize not only characters but also words, sub-words, or even whole paragraphs written in Arabic by a human hands utilizing advanced state-of-the-art techniques, algorithms, and technologies in image processing and artificial intelligence, specializing in a field of AI called "Optical Character Recognition (OCR)"

Optical Character Recognition (OCR):

OCR is a technology that enables the conversion of different types of documents, such as scanned paper documents, PDF files, or images captured by a digital camera, into editable, searchable and machine-readable data, it works by analyzing the text in these documents and translating it into machine-encoded text.

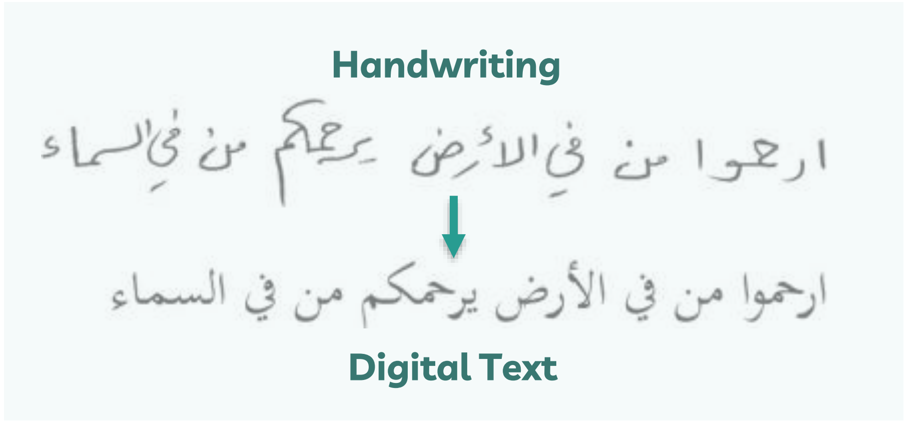
Arabic Handwriting Recognition (AHR) is a specialized subset of OCR focused on recognizing and digitizing handwritten Arabic scripts



INTRODUCTION

WHAT IS AHR?

Arabic Handwriting Recognition (AHR) is a tech that lets computers read handwritten Arabic text by analyzing shapes and patterns of characters, making it easier to digitize and process handwritten documents.



INTRODUCTION

WHY AHR?

An efficient AHR system provides huge benefits to our native beloved language such as:

Cultural Heritage: Digitizing and preserving historical manuscripts, literary works, and cultural documents that are written in Arabic script. This helps safeguard valuable cultural heritage and makes it accessible to future generations.

Education and Learning: aid in the development of educational tools that help learners practice and improve their Arabic handwriting skills. It can also facilitate the digitization and sharing of handwritten educational materials, making them more widely available.

Efficiency and Productivity: Automation of handwritten text recognition can streamline administrative and clerical tasks in various sectors such as education, government, and business, reducing the need for manual data entry and allowing for more efficient processing of documents.

INTRODUCTION

AHR Challenges:

Unlike other languages handwriting recognition like English for example, AHR isn't a quite wide scope for many developers because Arabic deals with special complexity in formatting that's due to several factors in our beautiful language:

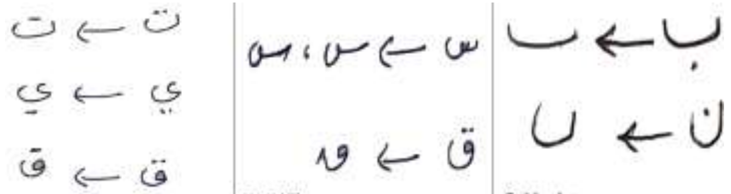
1. **Cursive Script:** Which resembles connectivity between letters that makes segmentation of each character difficult

2. **Style Standards:** There's no single standard to write in Arabic, we have different ways such as "Naskh" & "Riqa'a".

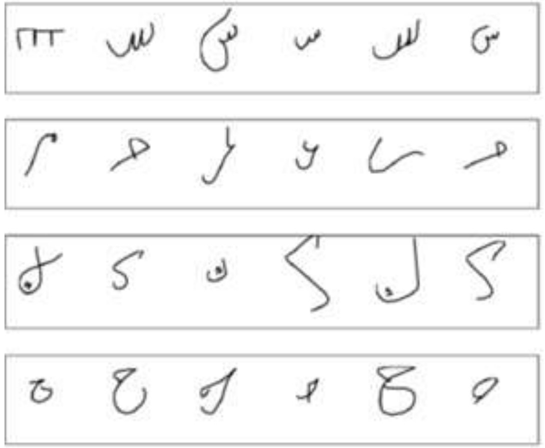
3. **Letter Variations:** A Single letter can have many different shapes.

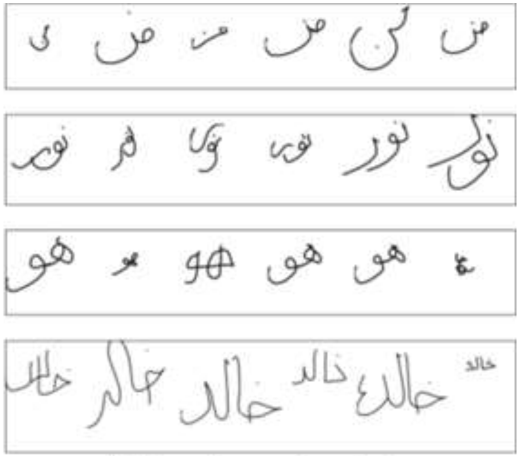
4. **Limited Datasets:** Getting our hands on datasets written in Arabic Handwriting can be quite a hard task

Similar Letters	Same Letter
ث & ت	ك & كـ
ف & ق	ي & يـ
ز & ر	ـلـ & ـل
بـ & يـ	ـفـ & ـف
ي & ى	خ & خـ
عـ & غـ	ه & ـهـ


A: Connected dots B: Different structures C: No dots

INTRODUCTION


(a) Sample character variations.


(b) Sample word variations.

Demonstration of different persons writing the same Characters/ Words in different ways which creates high variation

CHAPTER #2

SYSTEM PLANNING

SYSTEM PLANNING

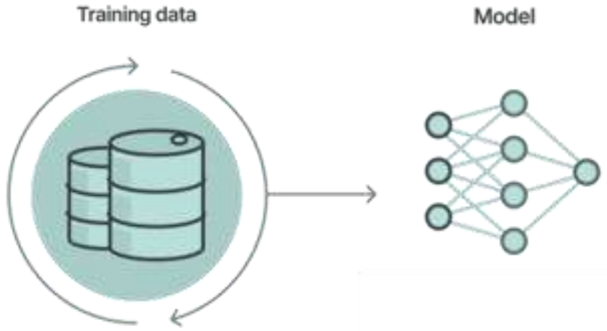
Arabic OCR System:

As we addressed before, developing such a system is a task that requires and advanced AI and Image Processing process.

The methodology of developing any AI System consists of two main aspects:

1. **Dataset:** the data and info for such a task, regularly goes through a wide range of processing steps in order to be ready for the model to train on, processes like labelling, resizing, or just generally organizing the data, it resembles the raw material for the model, so it's crucial and has a huge effect on the model

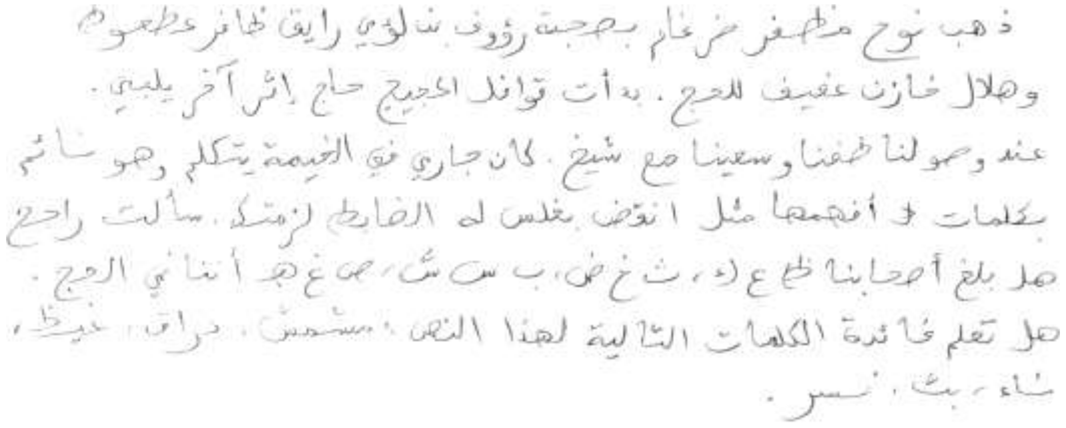
2. **Modelling Algorithm:** the algorithm or process of steps used on that data to create the intelligent system.



SYSTEM PLANNING

Gathering Dataset:
For our problem we had to search for an appropriate datasets which includes many writings of different styles in Arabic. The most suited available dataset for with such characteristics we found was the KHATT (KFUPM Handwritten Arabic TexT) dataset.

KHATT Dataset: An open-source database that consists of 2000 similar-text and 2000 unique-text images of paragraphs written by 1000 different persons as to provide a variety of handwritings in the dataset.


ذهب نوح مظفر ضرغام بصحبة رؤوف شلوفي رايق ظافر عظموم
وهلال خازن عفيف للحج. بدأت قوافل الحجيج حاج إثر آخر يلبي.
عند وصولنا طفنا وسعينا مع شيخ. كان جاري في الخيمة يتكلم وهو نائم
بكلمات لا أفهمها مثل انقض يغلس له الضابط زمشك. سألت راجح
هل بلغ أصحابنا ظ ط ع ء، ث خ ض، ب س ش، ص غ ج أثناء الحج.
هل تعلم فائدة الكلمات التالية لهذا النص: مشمش، دراق، خيط،
شاء، بث، نسر.

A Paragraph Sample Image from KHATT Dataset

SYSTEM PLANNING

Gathering Dataset:
We also used other datasets that were solely made for Arabic Alphabet Characters, unlike KHATT dataset we utilized these datasets for focusing only on providing images that has an Arabic letter inside each image.

1. AHAWP (Arabic Handwritten Alphabets, Words and Paragraphs): The dataset contains 65 different Arabic alphabets (with variations on begin, end, middle and regular alphabets), 10 different Arabic words (that encompass all Arabic alphabets) and 3 different paragraphs. The dataset was collected anonymously from 82 different users.

2. Arabic Handwritten Characters Dataset: The data-set is composed of 16,800 characters written by 60 participants, the age range is between 19 to 40 years, and 90% of participants are right-hand. Each participant wrote each character (from 'alef' to 'yeh').

Sample images from Arabic Handwritten Characters Dataset

A Sample from AHAWP Dataset

SYSTEM PLANNING

Arabic OCR Data Problems:

Camera-Taken images problem: As we thought about the idea we were directly faced with huge problems if we were to deal with paragraphs taken by camera. OCR depends on a clear view of the images passed to the system, taking a picture with a phone's camera from a hard angle might add unneeded noise to the image in the background, or the image being distorted. Which made us use advanced techniques and algorithms for unwarping images.

Paragraphs problem: Even if the images we had were caught by the right angle with no distortion or cropping, creating a model to extract text from entire paragraphs in Arabic OCR involves handling an enormous range of variations in paragraph structure, making it impracticable to create a model that can classify and extract text from every possible paragraph directly, so the paragraphs needed to be segmented.

Example of Warped Images captured from a hard angle

CHAPTER #3

SYSTEM DESIGN

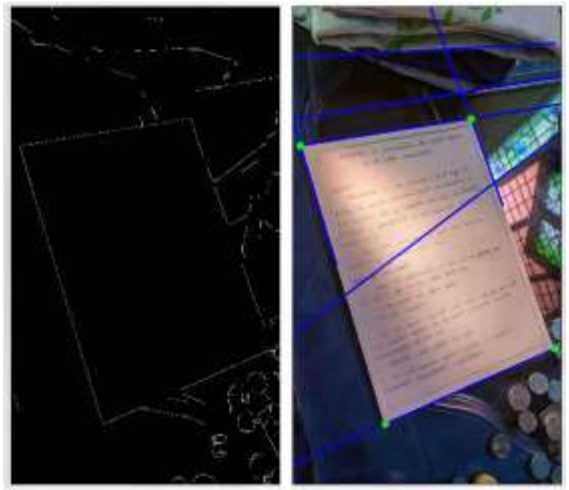
SYSTEM DESIGN

After lots of critical thinking and brainstorming about the system, its requirements, and its challenges, our team decided it's time to get working addressing the obstacles.

1) Geometric Unwarping: Geometric unwrapping is a crucial preprocessing step in Arabic Handwritten OCR to ensure that the text is properly aligned and readable for the OCR system.

"Hough Line Transformation": Initially, we employed a straightforward method using OpenCV to detect and correct the skew in our images. This approach involved detecting straight lines using the "Hough line transformation" within the image and using these lines to crop and align the image using simple steps:

1. Convert the image to grayscale to reduce complexity.
2. Apply edge detection using the Canny edge detector.
3. Use the Hough Line Transform to identify the lines in the image.
4. Identify the lines that correspond to the paper's borders.
5. Crop and warp the image based on these lines.


Document Unwarping Using Hough Line Transform

SYSTEM DESIGN

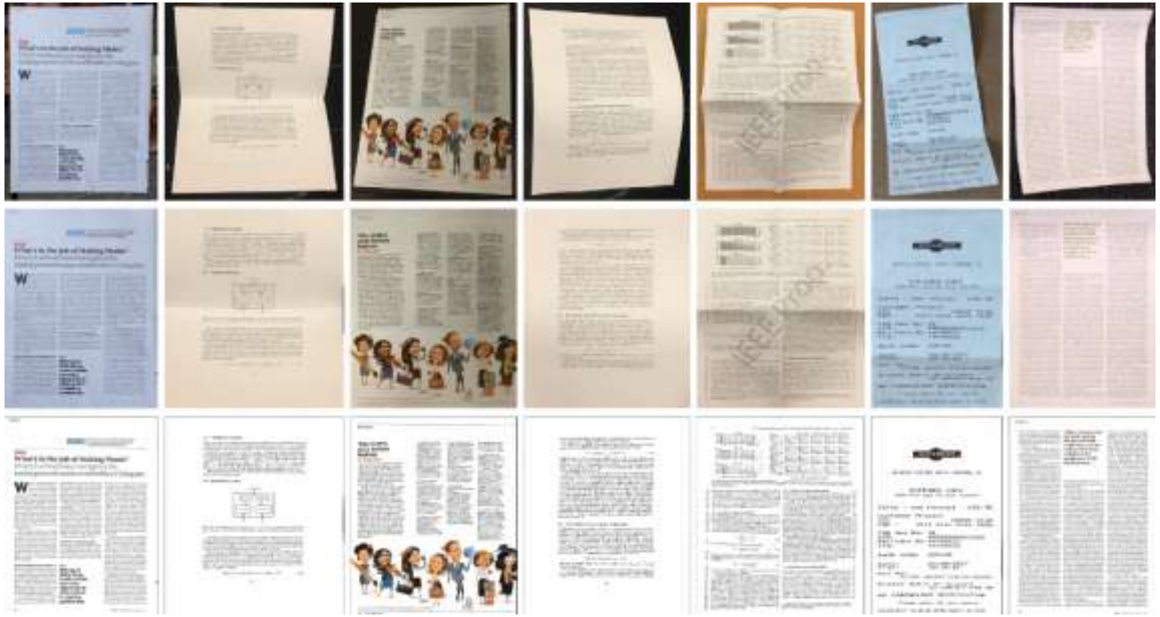
The previous approach wasn't really efficient because of several things:
1. It requires the entire paper to be visible within the image.
2. If any part of the paper is cut off or obscured, the line detection fails, resulting in an error.
3. The background must be clearly distinguishable from the paper.
4. This method assumes the paper is perfectly flat.

These limitations lead us to think that the approach doesn't suit most of the real-life cases.

"DocTR (Document Image Transformer)": "DocTR" utilizes a transformer-based model to understand and correct distortions in document images. The model is trained on a large dataset of document images, learning to recognize and rectify common issues such as geometric distortions and uneven lighting. By analyzing the spatial relationships within the image, the transformer can accurately predict the necessary transformations to straighten the text and normalize the lighting, resulting in a clear and well-aligned image.

The process begins with the model taking an image of the handwritten document and generating a set of transformation parameters. These parameters include adjustments for geometric distortions, such as skew and curvature.

SYSTEM DESIGN



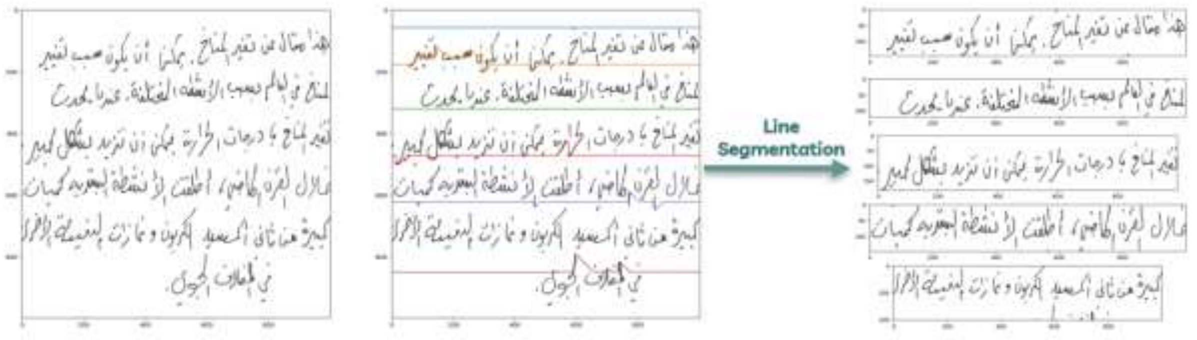
Results of Geometric Unwarping using “DocTR”

SYSTEM DESIGN

2) Paragraph Segmentation: Effective OCR begins with a structured approach to text extraction and analysis. First, the text is split into paragraphs using line breaks and indentation. Each paragraph is then divided into lines, and each line into words, separated by spaces or punctuation. Finally, a detailed analysis at the sub-word level examines characters and features.

“Histogram Projections” Approach: This approach relies on the utilization of histogram projections to accurately extract textual information from document images.

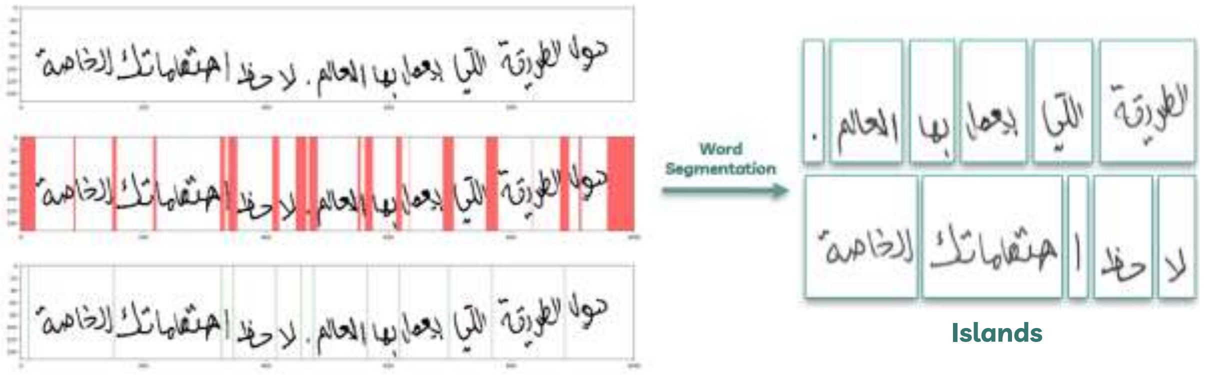
1. Vertical Axis (Line Segmentation): Our system uses vertical histograms of white pixels to identify text lines in document images, calculating a threshold based on the weighted average of histogram values to segment the image. We used the A* (A-star) search algorithm as a pathfinder to traverse and refine these text line regions for segmentation.



Line Segmentation using “Histogram Projections”

SYSTEM DESIGN

2. Horizontal Axis (Word Segmentation): With the text lines now accurately segmented, we turned our attention to the horizontal dimension to extract words, utilizing a similar approach except using the horizontal histogram instead of vertical. This histogram projects the pixel distributions along the horizontal axis, enabling the identification of individual words within each line of text. By applying thresholding and pathfinding techniques, the system can effectively separate the words, preparing them for character-level recognition. We named the resulting regions "Islands".



Word Segmentation using "Histogram Projections"

SYSTEM DESIGN

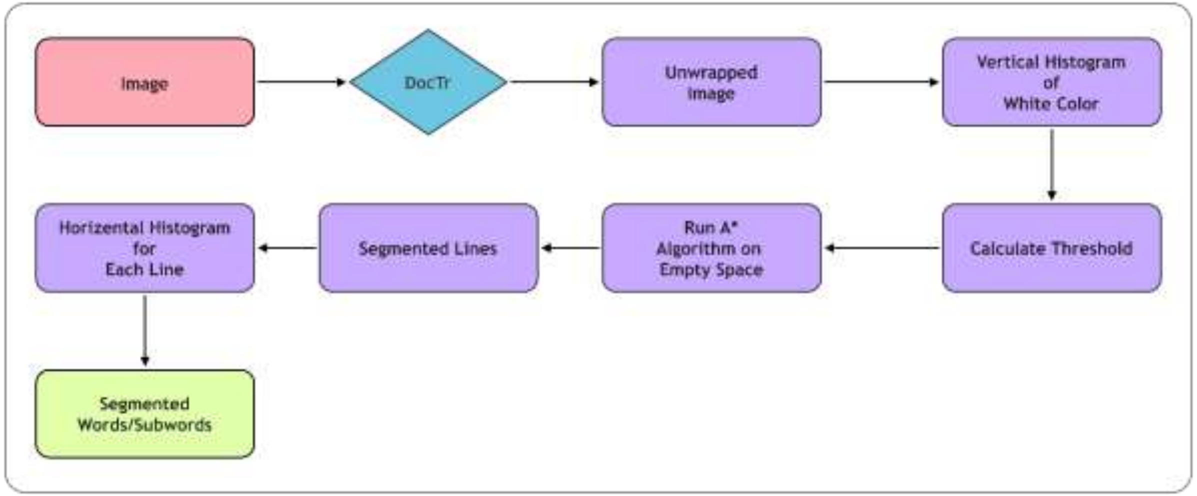


Image
DocTr
Unwrapped Image
Vertical Histogram of White Color
Calculate Threshold
Run A* Algorithm on Empty Space
Segmented Lines
Horizontal Histogram for Each Line
Segmented Words/Subwords

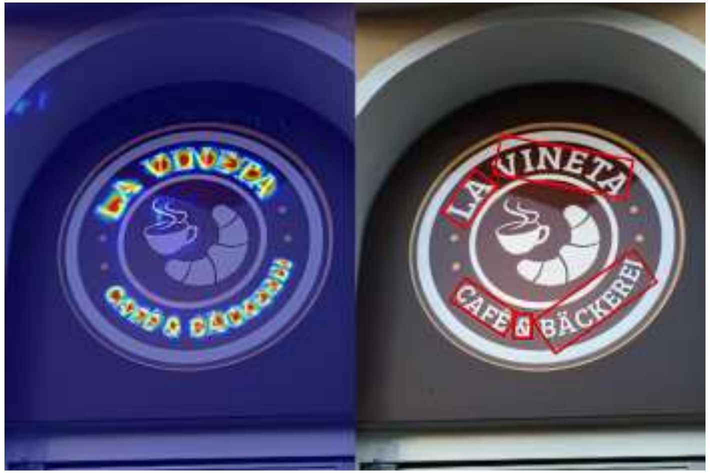
Diagram showing our initial segmentation system

SYSTEM DESIGN

Issues with Histogram Projection Segmentation:

1. Unable to effectively handle skewed or slanted text lines.
2. Difficulty in determining the appropriate threshold for separating text lines and words.

CRAFT (Character Region Awareness for Text Detection) Approach: CRAFT is a model designed to detect text in images. It operates by generating a heat map that highlights areas in the image where text is likely to be present. This heat map assists in identifying and localizing text within the image, making it a valuable tool for text detection in various contexts, including OCR for handwritten documents.



Example of CRAFT's Results

SYSTEM DESIGN

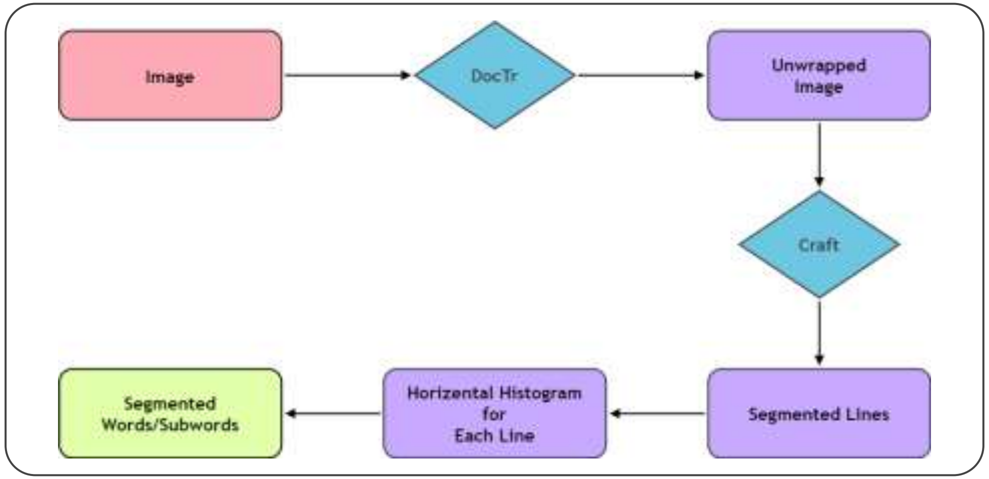
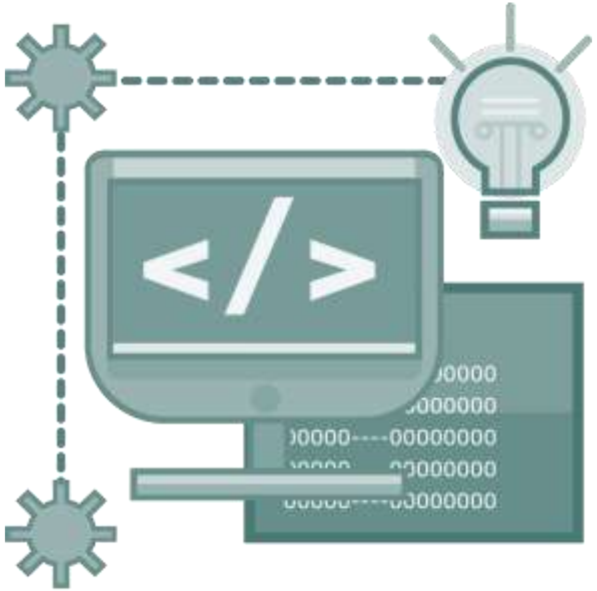


Diagram showing our optimized segmentation system

CHAPTER #4



SYSTEM DEVELOPMENT

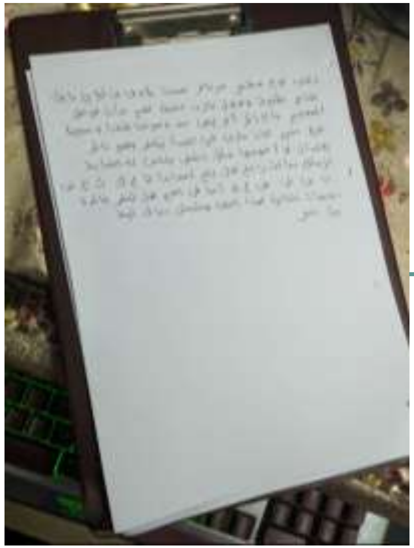
SYSTEM DEVELOPMENT

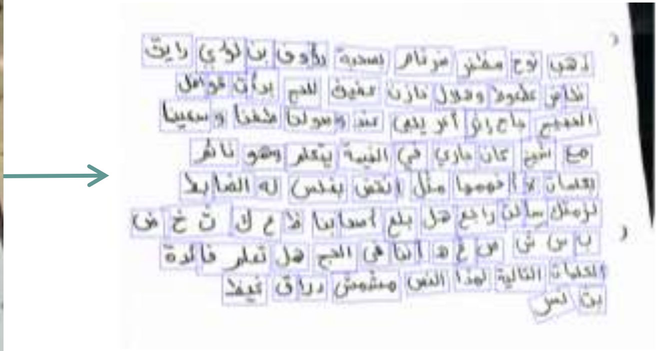
In this stage, it's time to put the designed segmentation system into test, preprocessing our data and preparing it in order to develop our comprehensive AI system and train it.

In order to test the model, we manually wrote a paragraph om a paper, took a picture of it and passed it to the system's flow shown previously in the diagram.

Segmentation System Steps:

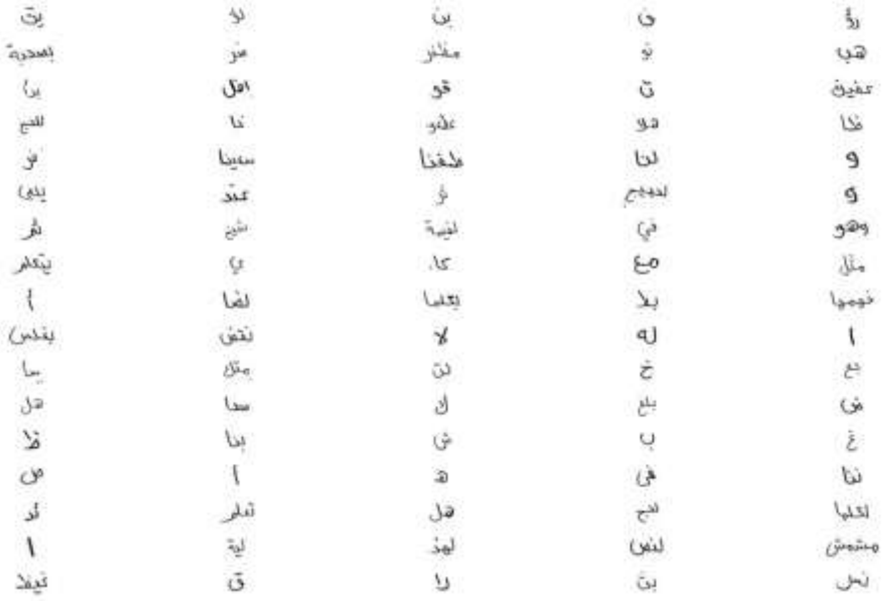
1. Capture the image of a handwritten paragraph.
2. Use DocTr for enhancing & unwarping.
3. Use CRAFT using Keras OCR library for Detection.
4. Extract segmentations.





Tested Results of DocTr followed by CRAFT

SYSTEM DESIGN



Final Results of Our Segmentation System

SYSTEM DEVELOPMENT

Additional Image Processing Steps: After getting the segmented text images or as we like to call them “Islands” there are still few basic processing steps that should be applied to them in order to help the AI Model algorithms recognize the images better:

1. Thresholding: converts a grayscale image to a binary image by setting pixel values to either black (0) or white (255) in image pixels based on a specified threshold. This simplifies the image, making it easier for the AI algorithm to distinguish text from the background, enhancing the accuracy of character recognition.

2. Padding: Padding involves adding extra pixels around the edges of an image making all segmented images the same size without distortion. It ensures that the features of interest (e.g., characters) are not cut off at the edges, providing a complete view of the text for the AI algorithm, which improves recognition accuracy, especially for characters near the borders. All images are padded to 64x64 size.

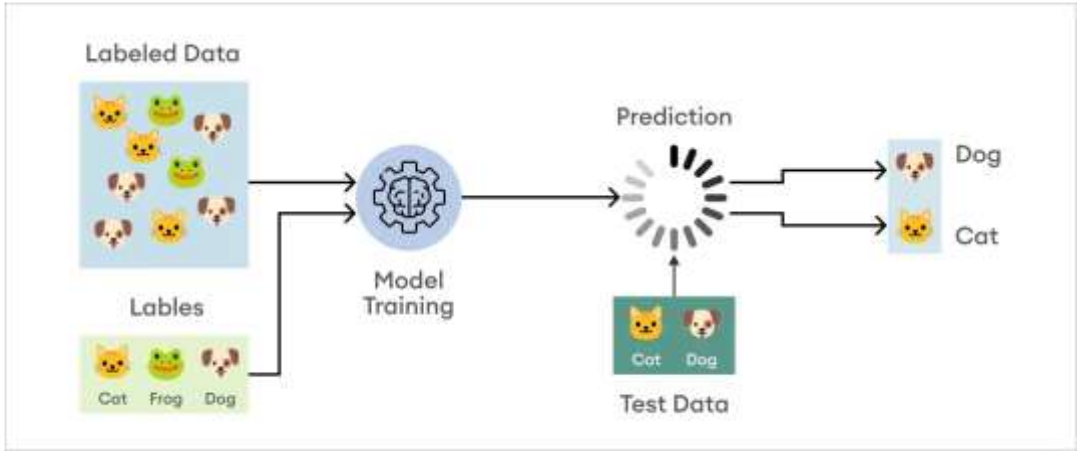
3. Rescaling: Rescaling changes the dimensions of an image to a standard size, typically normalizing pixel values to a range (e.g., 0 to 1). This standardizes input data, making the model’s training and inference processes more efficient and consistent.

SYSTEM DESIGN

Final Results of Our Preprocessing System

SYSTEM DEVELOPMENT

By implementing the previous whole preprocessing system, we were able to successfully solve the problems of our data concerning images, but an image without a label is useless for an AI Model algorithm, the data should be consisting of features (which are the images and their pixels in our case) and the labels, the labels answer the question to “what does that image actually say?”, which is the target of the model.



Demonstration of how AI Algorithms work with Features & Labels

SYSTEM DEVELOPMENT

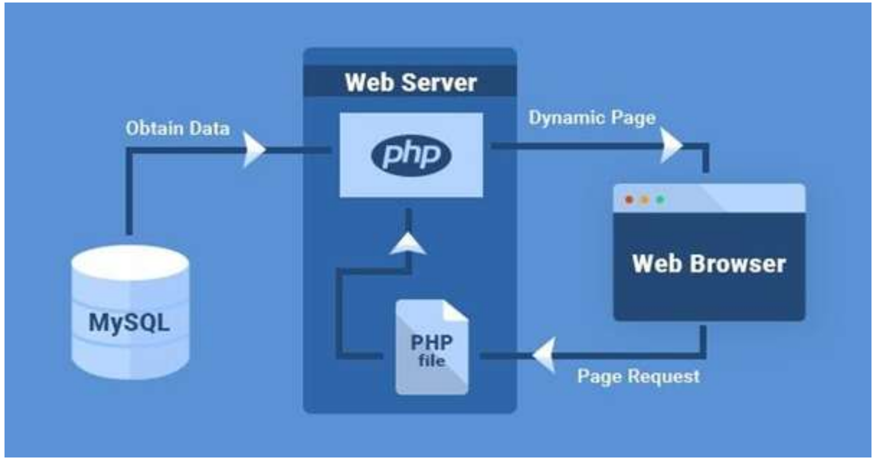
Image Labelling Web App (Prototype): Dealing with tens of thousands of images, we needed an application that could aid us efficiently in the process of labelling the new extracted data images from the paragraphs. The idea of this application is to display a set of images from the KHATT dataset, then we add the appropriate label for each image, and by pressing enter that label is saved in our database.

We used HTML, CSS and JavaScript for the frontend design implementation for the app.

While utilizing PHP for backend implementation for our web application, which is a widely used, open-source, server-side scripting language specifically designed for web development. PHP code is executed on the server, generating HTML that is then sent to the client. The client receives the results of the script but remains unaware of the underlying code.

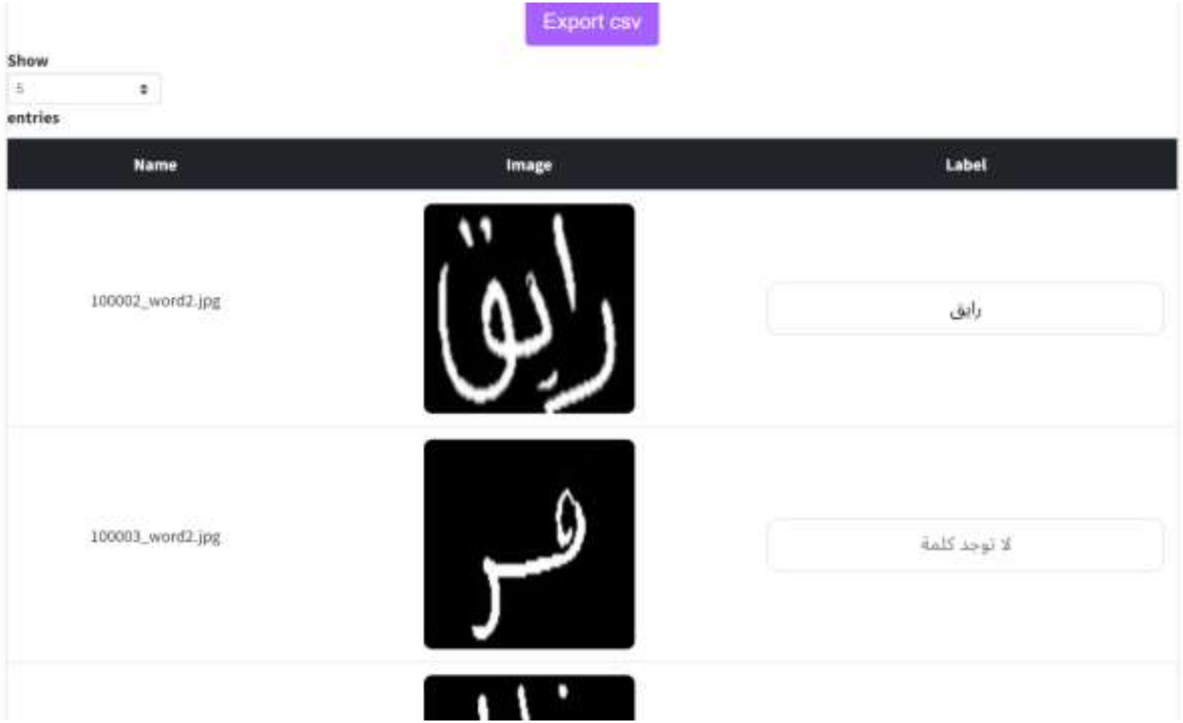
PHP provides robust features and functions for efficient data storage and retrieval. PHP supports a wide range of databases, including MySQL, PostgreSQL, SQLite, and Oracle. It offers built-in functions like mysqli and PDO (PHP Data Objects) that simplify connecting to databases, executing queries, and ensuring secure and reliable data access.

SYSTEM DEVELOPMENT



Demonstration of how PHP Works with Database

SYSTEM DEVELOPMENT



Name	Image	Label
100002_word2.jpg		رايق
100003_word2.jpg		لا توجد كلمة

Main page of our Prototype Labelling APP

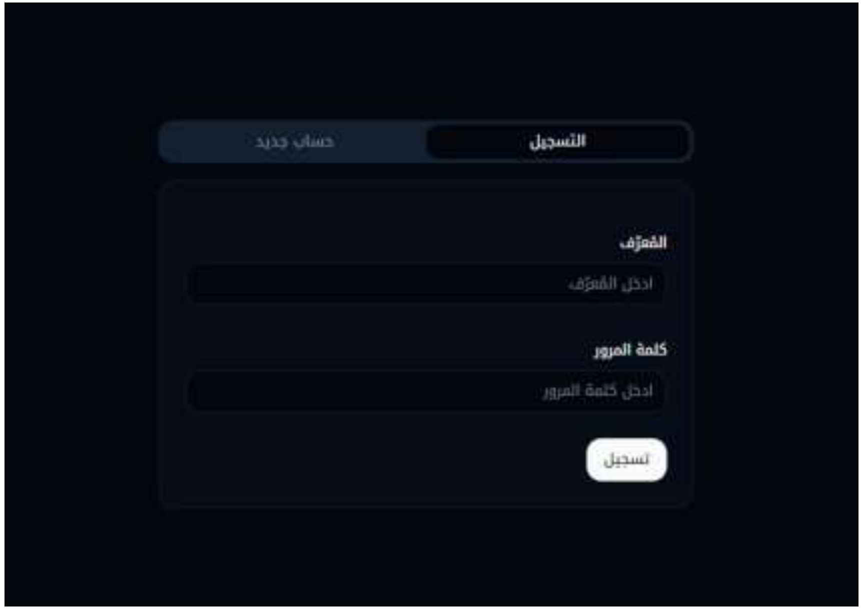
SYSTEM DEVELOPMENT

Optimized Wep APP: Tagger وسّام

To improve our web application, we later rewrote it from scratch and added user login functionality to enable secure access to our app. This login functionality includes creating user accounts using authentication rules such as passwords.

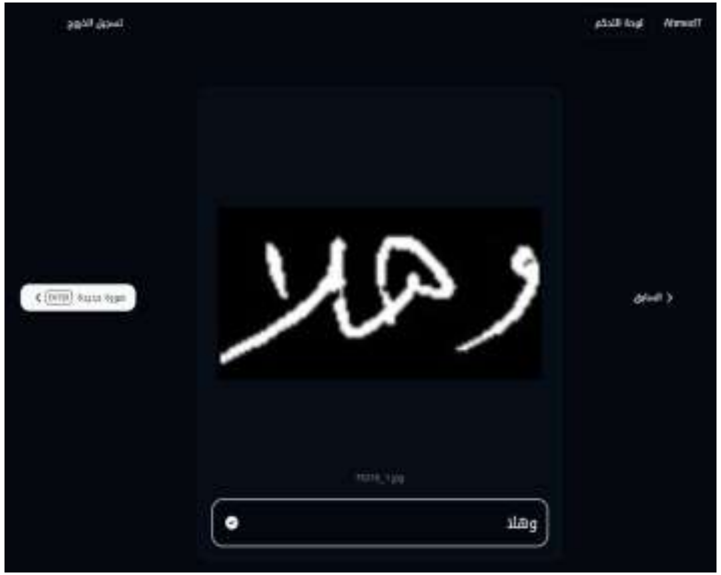
Application concurrent work support was also a crucial thing, allowing multiple users to collaborate on the same project simultaneously with features such as real-time updates and version control.

These improvements make the application more powerful, secure, and easy to use, meeting the needs of individual and group workflows.

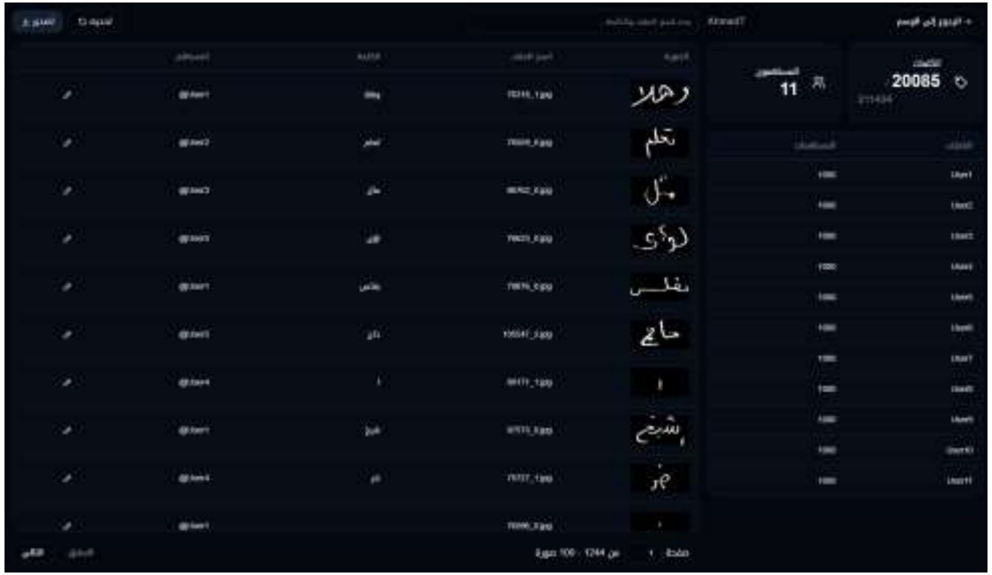


Login Page of Tagger وسّام

SYSTEM DEVELOPMENT



Labelling an Image using Tagger وسّام

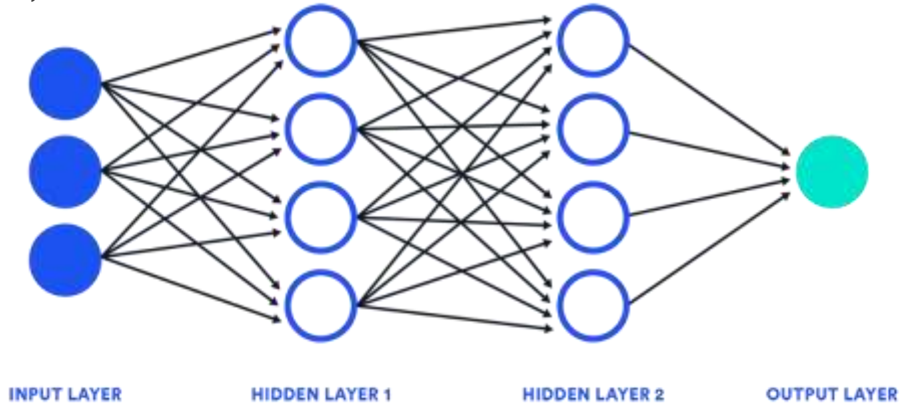


Dashboard of Tagger وسّام

SYSTEM DEVELOPMENT

AI Development: We currently have our data processed, labelled, and ready, all that's left is to use the suitable algorithms and technologies for this task. Dealing with images as unstructured data we'll surely use Artificial Neural Networks (ANNs).

Artificial Neural Networks (ANNs): Artificial Neural Networks (ANNs) are a fascinating and powerful type of machine-learning model inspired by the structure and function of the human brain to deal with unstructured data such as images, text, or even sounds, it's widely used in the AI field specially in what's known as "Deep Learning" for so many applications and projects in the real world. ANNs mainly consist of 3 main layers which are the Input Layer, Hidden Layers, and Output Layer



The General Architecture of ANNs

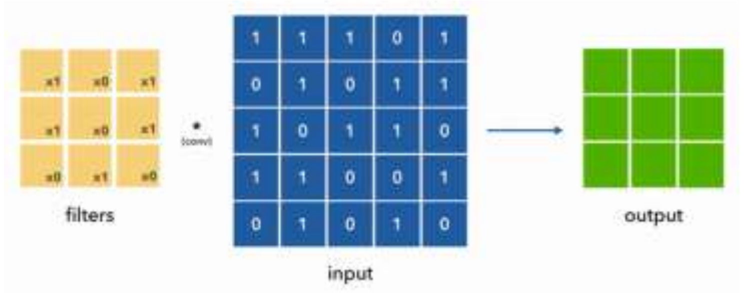
SYSTEM DEVELOPMENT

There are many types of ANNs, but mainly for our task we'll utilize and focus on 2 specific types of ANNs.

1) **Convolutional Neural Networks:** They are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs classifying and dealing with them. This specific neural network will be used as the feature extractor and classifier of our data knowing exactly what text is inside the image.

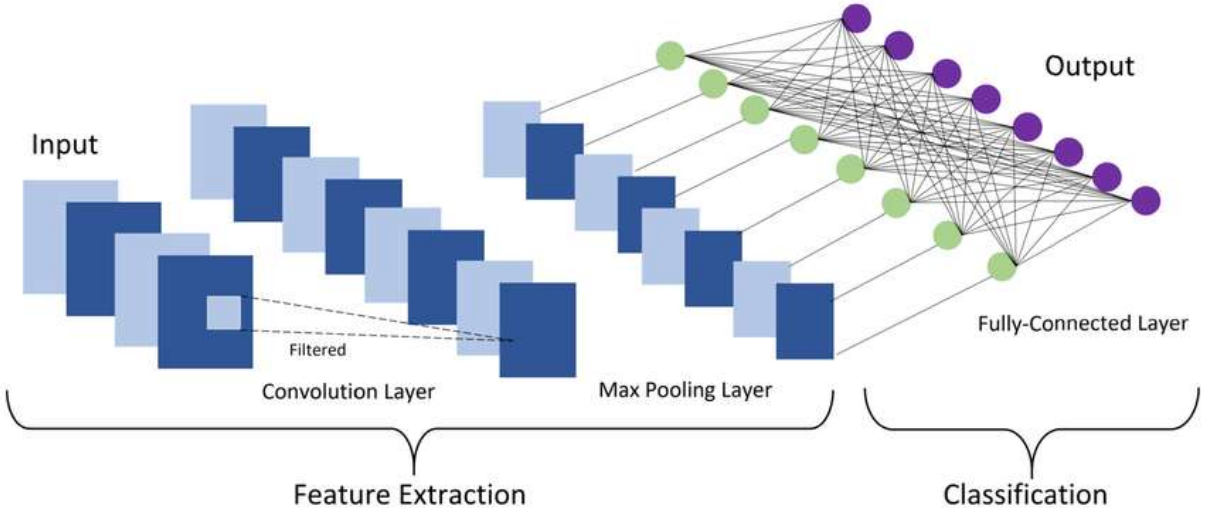
The general layers of a CNN that perform mathematical operations on images, which allows them to capture important features and parts of images :

1. Convolutional layers & Pooling layers which are used for feature extraction of the images
2. Fully connected (FC) layer which provides the output or the classification



The Convolutional Operation of CNNs Extracting Features

SYSTEM DEVELOPMENT

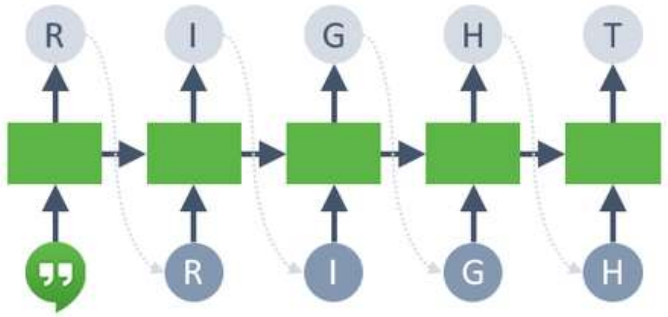


Demonstration of CNN Layers and Architecture

SYSTEM DEVELOPMENT

2) Recurrent Neural Networks (RNNs): RNNs are a type of artificial neural network designed to recognize patterns in sequences of data, such as text, speech, or time series. They have connections that loop back on themselves, allowing information to persist across steps in a sequence.

Arabic handwriting, like any text, has a sequential nature where each character and word follows a specific order. RNNs are particularly well-suited for this because they can maintain contextual information across the sequence of characters. They can also deal well with the varying lengths of text.



An Example of how RNNs Handle Text Sequences

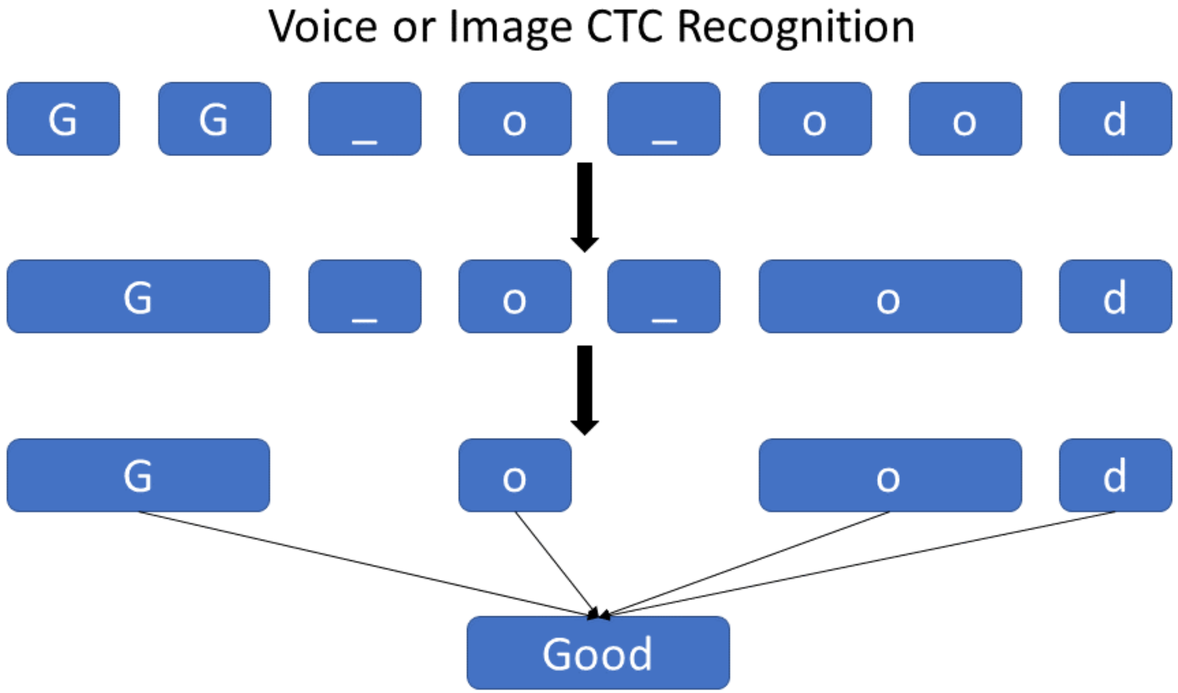
SYSTEM DEVELOPMENT

Loss Function: Every AI task or algorithm should have a loss function, the loss function mainly works as a measure for the algorithm itself, telling it how badly it performed on the data it’s trying to learn, and then it’s optimized using the learning rate and optimizers with other parameters that all work on improving the model’s performance while training over and over by changing the weights it uses at each layer for predicting the output.

CTC (Connectionist Temporal Classification) Loss Layer: CTC loss is a training algorithm used in neural networks for sequence-to-sequence problems where the alignment between input sequences (e.g., image data) and output sequences (e.g., text) is unknown.

CTC allows the model to learn the correct sequence of characters without needing predefined positions, handling variable-length inputs and outputs. It introduces a "blank" token to manage spaces between characters, improving the model's ability to handle gaps and repetitions in handwriting.

SYSTEM DEVELOPMENT



Demonstration of how CTC Works

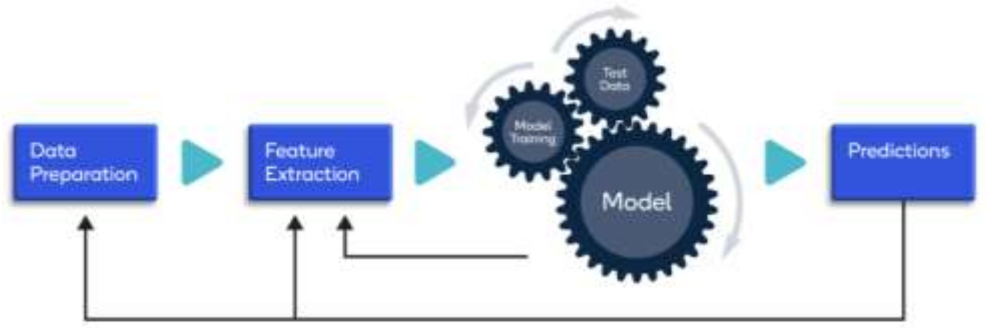
CHAPTER #5

AI SYSTEM

AI SYSTEM

Proceeding to build our Arabic OCR System, it's now the time to the AI System Development stage, but first let's talk about the training process and the model's development cycle.

Training Process: The training process utilizes the prepared data to teach the model to make accurate predictions by iteratively adjusting its parameters to minimize error. During this phase, the model undergoes multiple cycles of learning, where it continuously refines its internal parameters to better capture the patterns and relationships within the data. This process involves feeding the data into the model, calculating the loss or error between the predicted and actual values, and then updating the model's parameters to reduce this loss. Through repeated iterations, the model gradually improves its performance, learning to generalize from the training data to make accurate predictions on unseen data. Several crucial key hyperparameters are adjusted during this phase mentioned below.



AI Training Process

AI SYSTEM

The Training process include setting different hyperparameters, these hyperparameters are essential for the workflow of the process and model's learning performance such as:

1) **Batch Size:** The batch size in machine learning determines the number of training samples used in one iteration.

2) **Epochs**: Epochs indicate how many times the entire dataset is passed through the model during training.

3) **Learning Rate:** The learning rate influences the step size at each iteration while moving toward the minimum of the loss function.

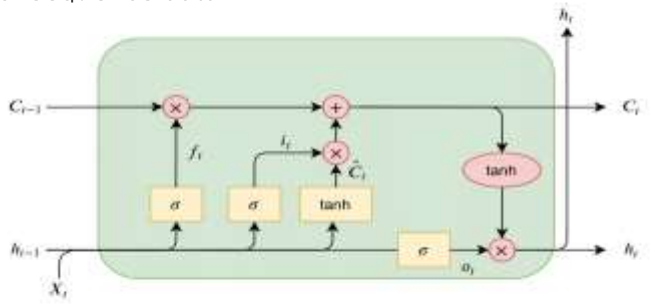
4) **Optimizer:** The optimizer is the algorithm that governs how the model's parameters are updated during training. Common optimizers include **Adam** and **Stochastic Gradient Descent (SGD).** The choice of optimizer significantly affects the speed and quality of convergence to the minimum of the loss function, impacting overall model performance.

AI SYSTEM

CNN Architecture: There are many architectures that could be used as a feature extractor and classifier for the images, most of the work and time spent on the AI System was on different trials of using different architectures with different hyperparameters that we will discuss in detail as we explain the 2 phases, we underwent developing the model.

RNN Architecture: As for the architecture used as a RNN in our project we chose the Bidirectional Long Short-Term Memory (Bi-LSTM) for dealing with text sequences.

Bidirectional Long Short-Term Memory (Bi-LSTM): Bi-LSTMs are a type of recurrent neural network (RNN) designed to capture dependencies in sequences by processing information in both forward and backward directions simultaneously as shown in. Unlike traditional Long Short-Term Memories (LSTMs) that only consider past context, Bi-LSTMs incorporate future context as well, enhancing their understanding of sequence data.



Input (Xt)
Memory Cell (Ct)
Hidden State (ht)
Forget Gate (ft)
Input Gate (it)
Output Gate(ot)

LSTM Unit

CHAPTER #6

AI SYSTEM IMPLEMENTATION

AI SYSTEM IMPLEMENTATION

Our Model's Methodology

Our model for Arabic handwriting recognition project begins with the Image **input layer**, which takes **32x64 images for our initial dataset (prototype), and 64x64 images for the larger one**.

These inputs are then fed into a CNN that extracts core features from the images. The model further processes these features through a **Dense Layer**, providing a meaningful representation of the input.

Following this, a **Batch Normalization layer** stabilizes the network by **normalizing the output of the dense layer**. A **Dropout Layer** is used to reduce overfitting by ignoring a fraction of the input.

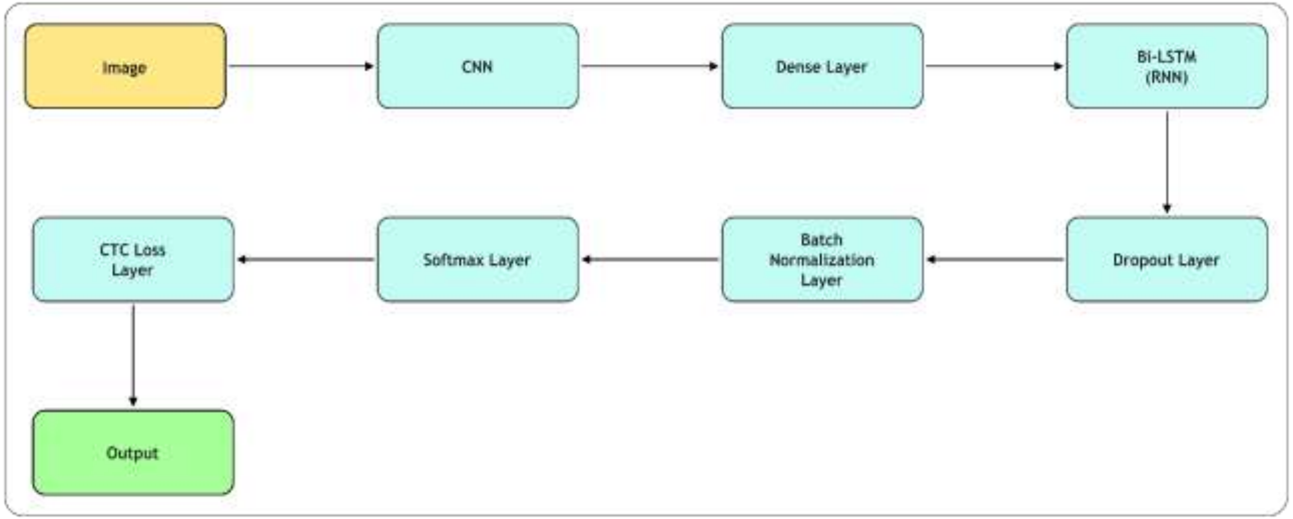
The **Bi-LSTM (RNN)** layer is critical for analyzing sequences of the input image, making it ideal for **recognizing the sequential nature of Arabic handwriting.**

The output of the **Bi-LSTM** is stabilized by another **Batch Normalization layer**, followed by another **Dropout Layer**. The model then uses a **Softmax Layer** to generate the predictive output, ensuring accurate transcription of the handwritten input.

The **CTC Loss Layer** forms the final part of the network, handling the **sequence and reducing the error percentage by accurately calculating the CTC loss and providing feedback for improved performance**.

The model's **output** is generated after the CTC Loss Layer, providing an accurate sequence of **characters that can be used as the final output for the Arabic handwriting recognition.**

AI SYSTEM IMPLEMENTATION



Our Model's Approach

AI SYSTEM IMPLEMENTATION

Prototype Model : Utilizing Initial Dataset

For the initial model prototype, we utilized a subset of the KHATT dataset containing **5,159 samples**, which we augmented to a total of **15,477 samples**.
This dataset covered 36 Arabic letters and included a variety of segmented words and sub-words. As part of our preprocessing steps, all images were resized to a uniform 32x64 resolution.

Samples of Initial Dataset Showing Arabic Words

To reach the optimal CNN architecture and parameters suited for this task, as we implemented our prototype on this small dataset, we began experimenting with different architecture known for their high performance and special tasks such as:

1. EfficientNetB1

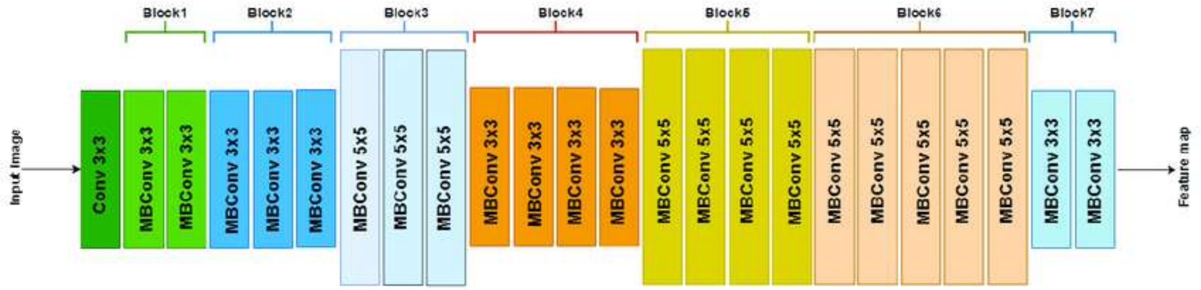
2. VGG19

3. ResNet152

All models were trained on **70 Epochs**.

AI SYSTEM IMPLEMENTATION

EfficientNetB1: A part of the EfficientNet family that uses a unique method called Compound Scaling to make the network both accurate and efficient by focusing on three main aspects for each model (B0,B1,B2..etc):
Deeper: Add more layers.
Wider: Use more channels per layer.
Higher Resolution: Use larger input images.
Scaling them with balance using a coefficient which makes the network efficient in both performance and computational power.

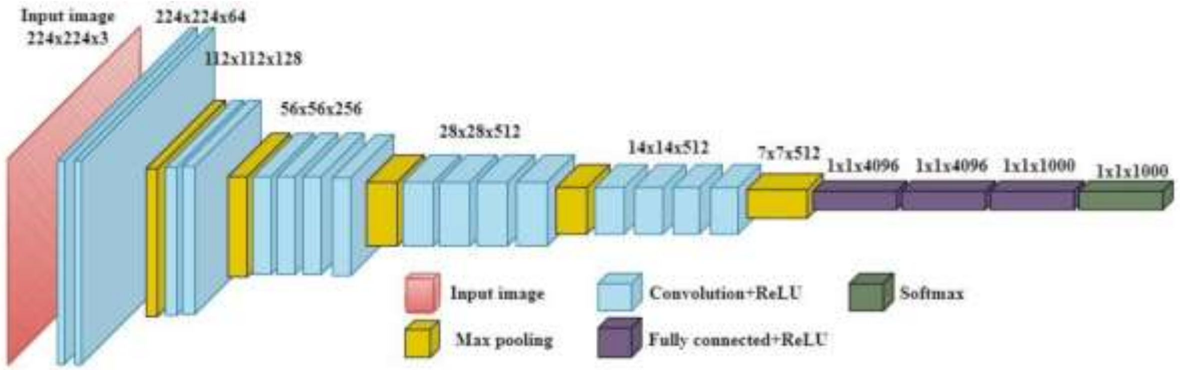


EfficientNetB1 Architecture

→The model achieved a validation loss of 0.78 and a CER of 7.3% (92.7% Accurate).

AI SYSTEM IMPLEMENTATION

VGG19: Is a part of the Visual Geometry Group (VGG) family, known for its simplicity and depth. It consists of 19 layers, including 16 convolutional layers and 3 fully connected layers.
Simplicity: The architecture is straightforward, using only 3x3 convolutional layers stacked on top of each other.
Uniform Design: The use of the same kernel size throughout the network makes it easy to implement and understand.
Performance: Despite its simplicity, VGG19 achieves high accuracy on various image recognition tasks.

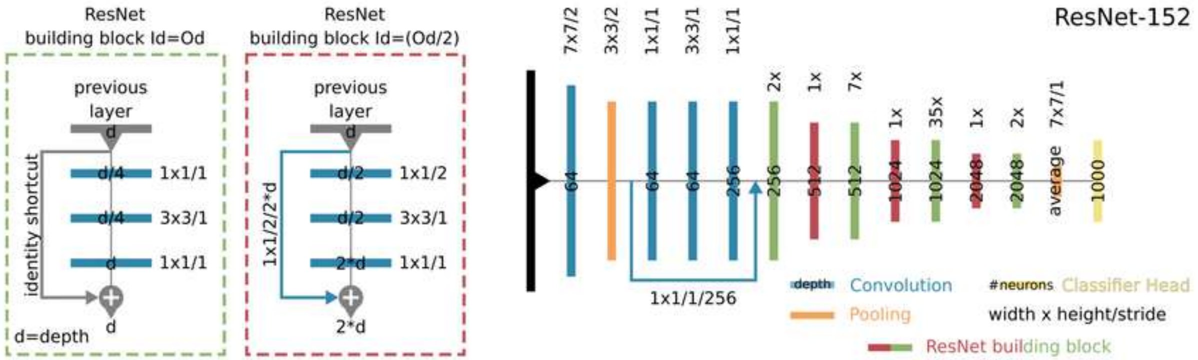


VGG19 Architecture

→The model achieved a validation loss of 0.6 and a CER of 5.4% (94.6 Accurate).

AI SYSTEM IMPLEMENTATION

ResNet152: ResNet152 is part of the Residual Network (ResNet) family, which introduced the concept of residual learning to address the problem of vanishing gradients in deep networks. It consists of 152 layers.
Residual Blocks: ResNet152 uses skip connections (or shortcuts) that bypass one or more layers, allowing the network to learn residual functions.
Very Deep Architecture: With 152 layers, ResNet152 can learn highly complex features and representations.
Performance: ResNet152 achieves state-of-the-art performance on various image recognition benchmarks.

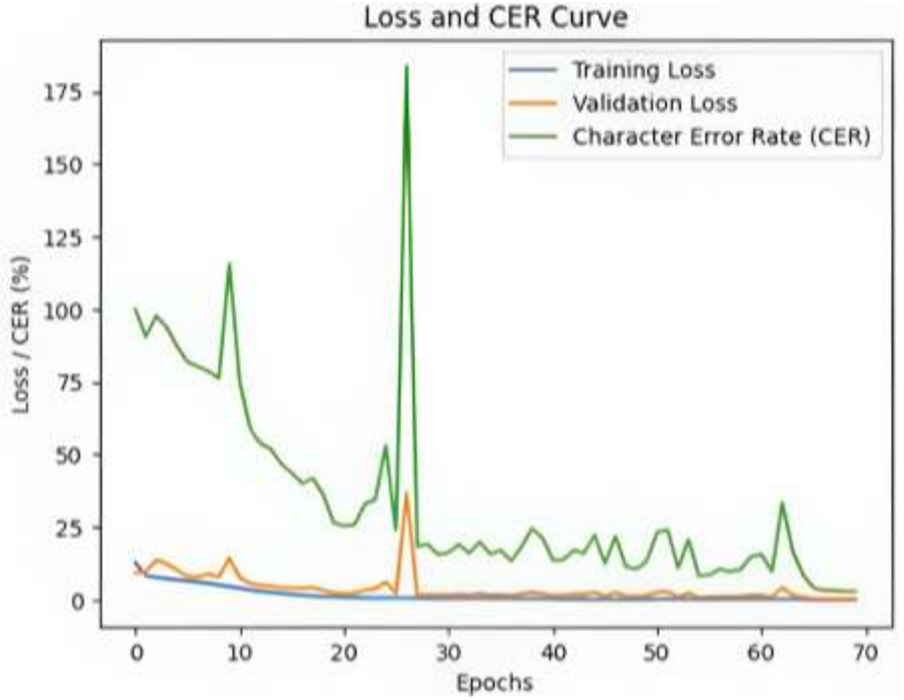


ResNet152 Architecture

→The model achieved a validation loss of 0.3 and a CER of 2.96% (97.04 Accurate).

Which by far is the **best** suited model for this problem

AI SYSTEM IMPLEMENTATION



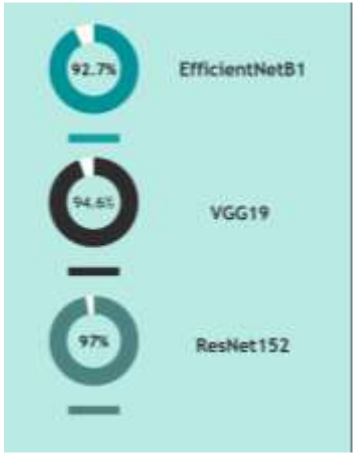
ResNet152 Performance (Training, Validation Loss, & CER) Through the Epochs

AI SYSTEM IMPLEMENTATION

Model	Validation Loss	Character Error Rate	Accuracy
VGG19	0.6	5.4%	94.6%
EfficientNetB1	0.78	7.3%	92.7%
ResNet152	0.3	2.96%	97.04%

Parameters	VGG19	EfficientNetB1	ResNet152
Optimizer	Adam	Nadam	Nadam
Learning Rate	0.0001	0.001	0.001
β_1	0.9	0.9	0.9
β_2	0.999	0.999	0.999
ε	1×10^{-8}	1×10^{-8}	1×10^{-8}
Batch Size	128	128	128

Hyperparameter used for each Architecture's Training Process



Comparison of each Architecture's Performance Results

AI SYSTEM IMPLEMENTATION

Larger Dataset

We curated an extensive dataset incorporating various sources, including a substantial portion of the KHATT Dataset originally comprising approximately 20,000 samples, which after meticulous cleaning and augmentation processes, expanded to 50,145 samples.
Additionally, we integrated data from the AHAWP and Arabic Handwritten Characters datasets, also yielding a combined total of 58,474 samples post-cleaning and augmentation. This unified dataset was meticulously structured to encompass 36 Arabic letters, with all images standardized to a resolution of 64x64 pixels.

Optimized Model: ResNet50V2

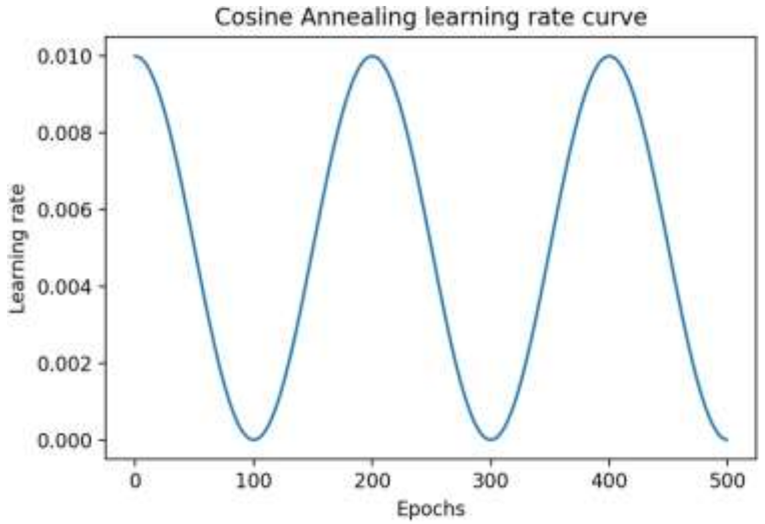
We opted for ResNet50V2 as our optimal model over ResNet152, despite the latter's larger size and higher resolution dataset, due to several considerations. ResNet50V2 strikes a balance between performance and computational efficiency, making it more feasible for scaling up to larger datasets with higher resolutions.

V2 Versions of the Residual Network have some optimized key features such as:
Modified Residual Blocks: The residual blocks are restructured to improve gradient flow and training dynamics.
Pre-Activation: Batch normalization and ReLU activation are applied before the convolutional layers, unlike in the original ResNet where they are applied after.

AI SYSTEM IMPLEMENTATION

Optimized Learning Rate: Cosine Learning Rate Scheduler

To enhance its training efficacy, we implemented advanced techniques like the "Cosine Learning Rate Scheduler". This scheduler dynamically adjusts the learning rate during training according to a Cosine wave as shown in potentially improving model convergence and generalization.



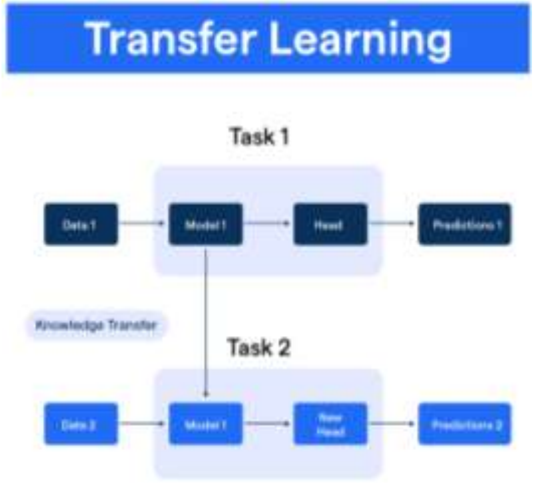
Example of Cosine Scheduler Effect on Learning Rate

AI SYSTEM IMPLEMENTATION

Optimized Transfer Learning: Alphabets Model

Recognizing the need for further improvement, we employed transfer learning techniques to refine the model's performance, leveraging ResNet50V2's capabilities. This approach utilizes knowledge gained from solving one problem and applies it to a different but related problem, typically resulting in improved learning efficiency and performance.

In our case, after initially training ResNet50V2 on an Arabic Alphabet Character dataset consisting of 58,474 samples, we easily achieved a low CER, high accuracy model that's well trained to recognize Arabic Alphabetical Letters. This success prompted us to further enhance the model's capabilities by transferring its learned weights to the main KHATT Dataset



The idea of Transfer Learning

AI SYSTEM IMPLEMENTATION

Final Results:

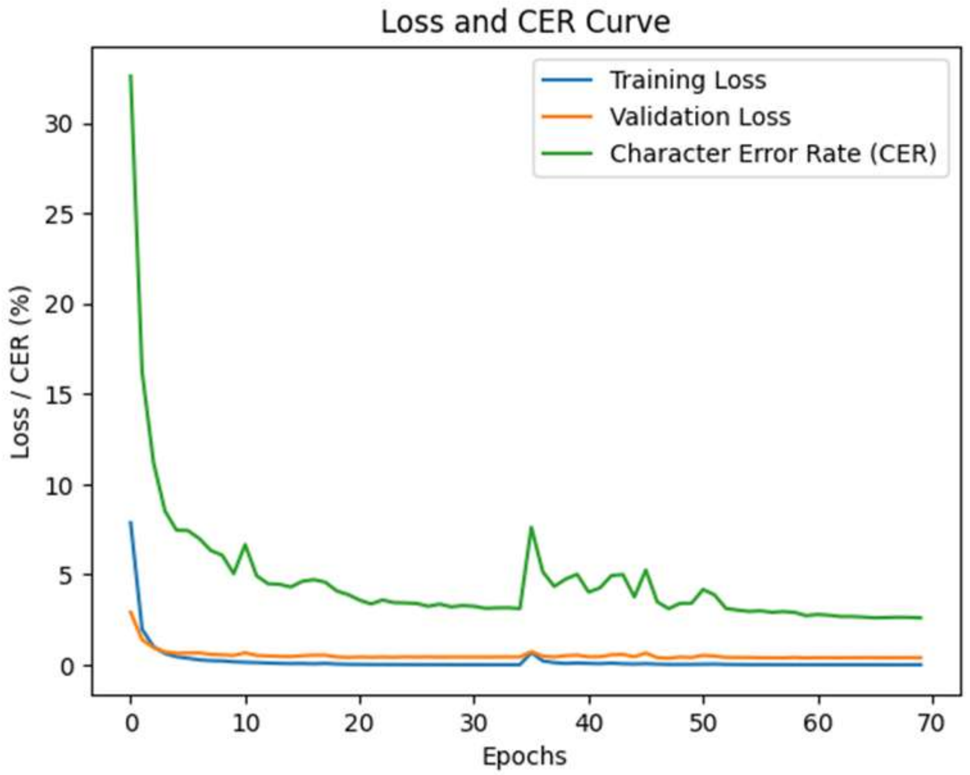
We enhanced the ResNet50V2 model pre-trained on the Arabic Alphabet Character dataset by further training it on the main KHATT Dataset. Using advanced techniques and optimal hyperparameters, the model underwent comprehensive training, achieving remarkable results.

Training Details:

- Datasets:
  - Arabic Alphabet Character dataset: 58,474 samples. (Pre-Trained On It)
  - KHATT Dataset: 50,145 samples.
  - Total Samples: 108,619.Training
- Configuration:
  - Optimizer: Adam
  - Initial Learning Rate: 0.001
  - β1: 0.9
  - β2: 0.999
  - ε: 1e-7
  - Learning Rate Scheduler: Cosine
  - Epochs: 70
- Results:
  - Character Error Rate (CER): **3% (Accuracy: 97%)**

→These results highlight the model's robustness and accuracy in recognizing handwritten Arabic text, proving its effectiveness for real-world applications.

AI SYSTEM IMPLEMENTATION



ResNet50V2 Final Performance

CHAPTER #7

Digital Ḍād- ضاد الرقمية

Digital Ḍād- ض الرقمية

Overview

In the development of our application for Arabic Handwritten OCR, we adopted a robust architecture to ensure efficiency and scalability. The core of our system is built on a front-end and back-end model, which allows us to separate concerns between the user interface and the computational heavy lifting.

We decided to name our app (ض الرقمية) .
it encapsulates how our app work with the Arabic language in a digital world

Frontend and Backend Model

In software engineering, the terms frontend and backend (sometimes written as back end or back-end) refer to the separation of concerns between the presentation layer (frontend), and the data access layer (backend) of a piece of software, or the physical infrastructure or hardware.

Digital Ḍād- ض الرقمية

Frontend

In the Frontend, we designed a web application which can be used by the user. It works by uploading an image that has the text which will be converted to a digital text by our system.

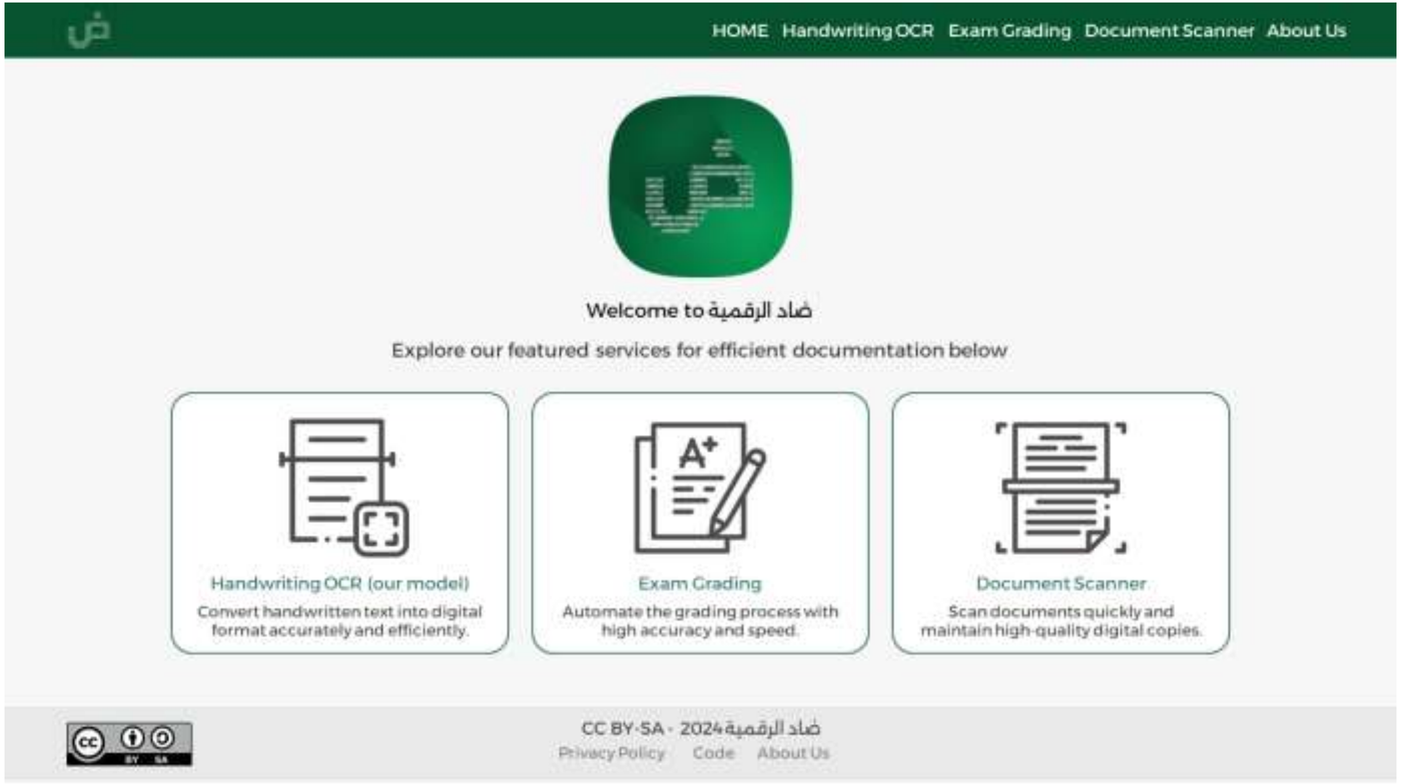
The website was created using HTML5, CSS3, JavaScript, and React JS frontend framework.

Native Application

A native application is a software program designed specifically for use on a particular platform or device. Unlike web applications that run in a browser, native applications are developed using platform-specific programming languages and tools. This allows for better performance, integration with device hardware, and a more seamless user experience.

We decided against using native applications because they are typically limited to a single platform, requiring separate versions for iOS, Android, and other operating systems, which significantly increases development time and costs. Moreover, native applications often demand powerful resources and frequent updates to maintain compatibility with new operating system versions and devices.

Digital Ḍād- ض الرقمية


HOME Handwriting OCR Exam Grading Document Scanner About Us

Welcome to ضاد الرقمية

Explore our featured services for efficient documentation below

Handwriting OCR (our model)
Convert handwritten text into digital format accurately and efficiently.

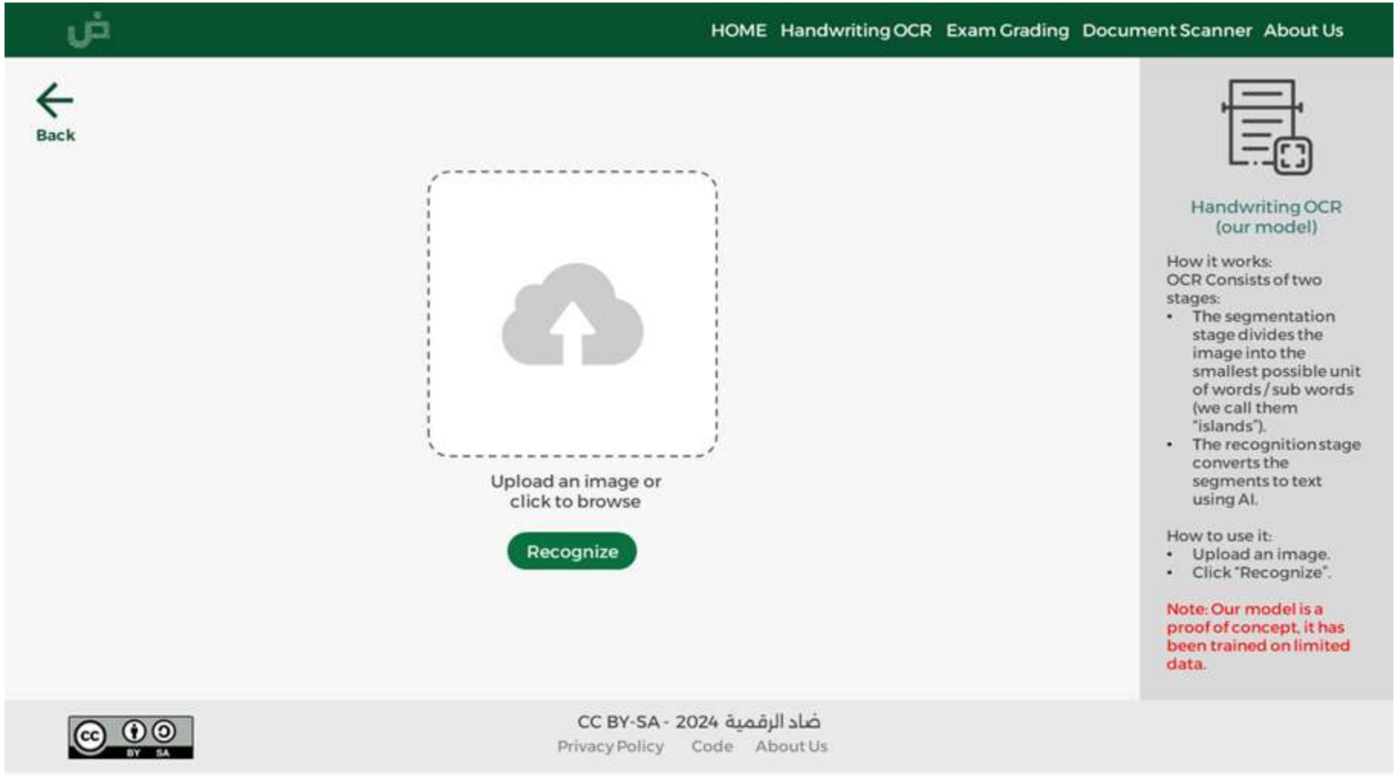
Exam Grading
Automate the grading process with high accuracy and speed.

Document Scanner
Scan documents quickly and maintain high-quality digital copies.

CC BY-SA - 2024 ضاد الرقمية
Privacy Policy Code About Us

Screenshot of Main Page

Digital Ḍād- ض الرقمية



Handwriting OCR (our model)

How it works:
OCR Consists of two stages:

- The segmentation stage divides the image into the smallest possible unit of words / sub words (we call them "islands").
- The recognition stage converts the segments to text using AI.

How to use it:

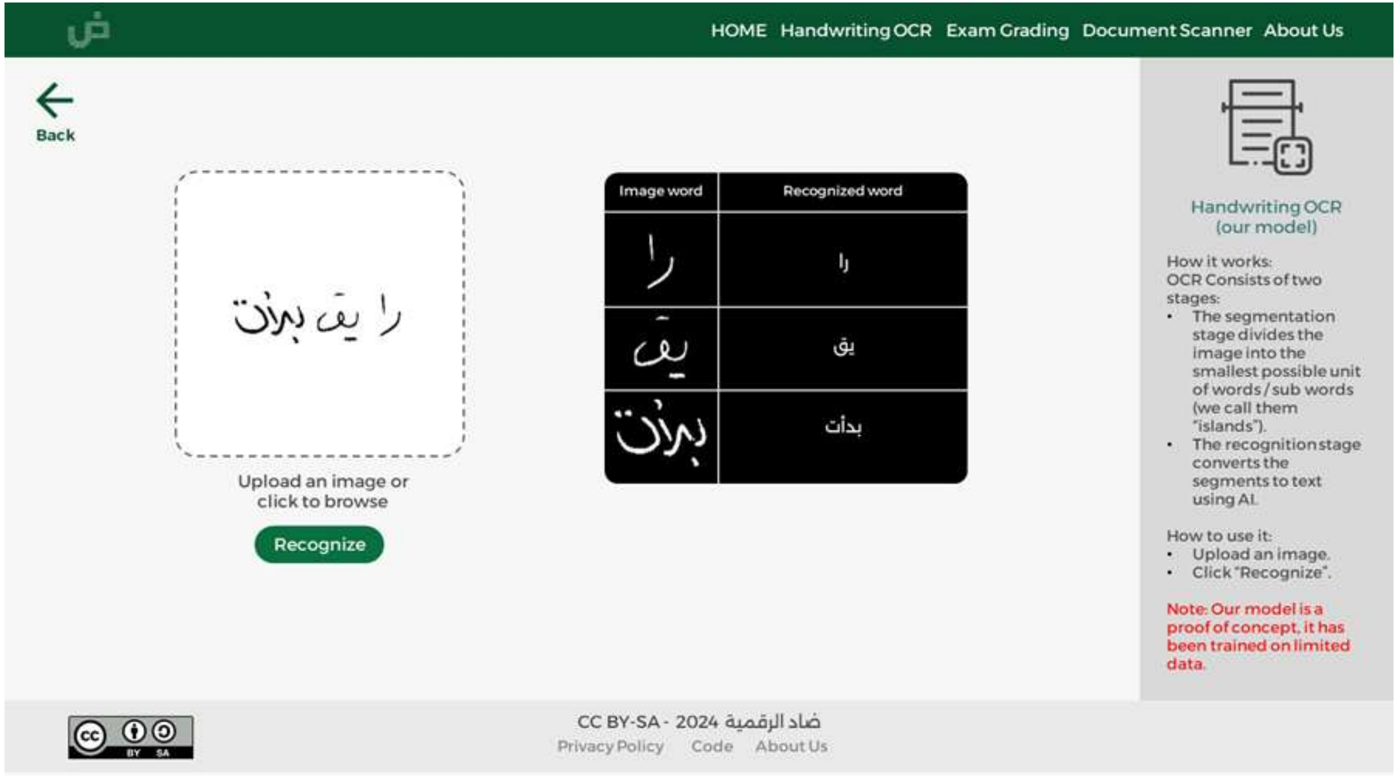
- Upload an image.
- Click "Recognize".

Note: Our model is a proof of concept, it has been trained on limited data.

CC BY-SA - 2024 ضاد الرقمية
Privacy Policy Code About Us

Screenshot of "our model" page

Digital Ḍād- ض الرقمية


HOME Handwriting OCR Exam Grading Document Scanner About Us

Back

Upload an image or click to browse

Recognize

Image word	Recognized word
را	ار
يق	يق
برات	بدأت

Handwriting OCR (our model)

How it works:
OCR Consists of two stages:
- The segmentation stage divides the image into the smallest possible unit of words / sub words (we call them "islands").
- The recognition stage converts the segments to text using AI.

How to use it:
- Upload an image.
- Click "Recognize".

Note: Our model is a proof of concept, it has been trained on limited data.

CC BY-SA - 2024 ضاد الرقمية
Privacy Policy Code About Us

Screenshot of how results are displayed

APPLICATION

Backend

The backend is the part of the system responsible for processing input images using the segmentation system and then passing these images to the model for recognition, ultimately returning the labels. It serves the crucial purpose of offloading the computationally intensive tasks from the frontend, ensuring that the user interface remains responsive and efficient.

Developing an API

In a frontend-backend model application, an API (Application Programming Interface) is essential for communication between the two parts. The API allows the frontend to send image data to the backend, where it can be processed, and then receive the recognized labels.

Fake API : we developed a simple fake API that processed the image in an arbitrary way and returned predefined labels. This fake API was crucial for testing the API's functionality and determining which package to use for developing the final API.

Fast API : is a modern, fast (high-performance) web framework for building APIs with Python 3.6+ based on standard Python-type hints. It is designed to be easy to use and deploy, offering automatic interactive API documentation.

Flask : is a lightweight WSGI web application framework in Python. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. Flask is simple and flexible, providing the essentials without requiring extensive boilerplate code.

Digital Ḍād- ض الرقمية

System Configuration

Model Training process, tuning, trials, development and implementation as well as the backend were all implemented in a local environment (self-hosted) rather than on the cloud. This decision was based on our need to have full control over the hardware and software environment, ensuring optimal performance and ease of maintenance.
The hardware configuration used in the backend includes:

- OS: Windows 11 Pro.
- CPU: -10th Gen Intel ® Core ™ I5-10400F (AI Model Training)

  -12th Gen Intel ® Core ™ I3-12100F (AI Inference & Backend)
- Installed RAM: 16.0 GB DDR4 @ 3200MHZ.
- GPU: NVIDIA GeForce RTX 3060.

We implemented the system in **Python** using **TensorFlow** for model integration, **OpenCV** for image processing.

CHAPTER 8

ACHIEVEMENTS, CONCLUSION& FUTURE WORK

ACHIEVEMENTS, CONCLUSION & FUTURE WORK

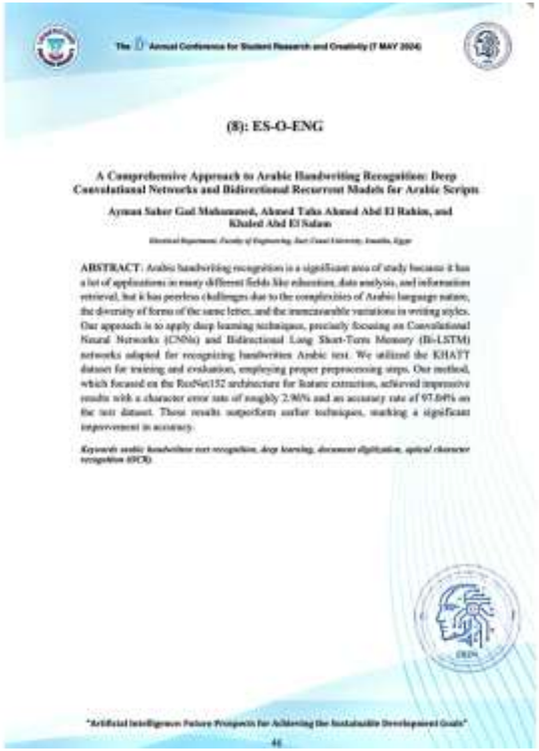
Achievements:

Our project and research on enhancing our model for Arabic handwriting recognition have been highly successful, culminating in two significant publications. One of our papers was presented at a prestigious conference, while another was published in a reputable journal. These achievements underscore the impact and relevance of our work in the field of Arabic OCR, highlighting its contributions to advancing handwritten text recognition technology.

Conference: “Artificial Intelligence: Future Prospects for Achieving the Sustainable Development Goals”:

Our research, titled ”A Comprehensive Approach to Arabic Handwriting Recognition: Deep Convolutional Networks and Bidirectional Recurrent Models for Arabic Scripts” was presented at the 6th Annual Conference “Artificial Intelligence: Future Prospects for Achieving the Sustainable Development Goals” for Student Research and Creativity held on May 7, 2024, at Suez Canal University.

ACHIEVEMENTS, CONCLUSION & FUTURE WORK



The 5th Annual Conference for Student Research and Creativity (7 MAY 2024)

(8): ES-O-ENG

A Comprehensive Approach to Arabic Handwriting Recognition: Deep Convolutional Networks and Bidirectional Recurrent Models for Arabic Scripts

Ayman Saher Gad Mohammed, Ahmed Taha Ahmed Abd El Rahim, and Khaled Abd El Salam

Electrical Department, Faculty of Engineering, Suez Canal University, Ismailia, Egypt

ABSTRACT: Arabic handwriting recognition is a significant area of study because it has a lot of applications in many different fields like education, data analysis, and information retrieval, but it has peerless challenges due to the complexities of Arabic language nature, the diversity of forms of the same letter, and the immeasurable variations in writing styles. Our approach is to apply deep learning techniques, precisely focusing on Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (Bi-LSTM) networks adapted for recognizing handwritten Arabic text. We utilized the KHATT dataset for training and evaluation, employing proper preprocessing steps. Our method, which focused on the ResNet152 architecture for feature extraction, achieved impressive results with a character error rate of roughly 2.96% and an accuracy rate of 97.04% on the test dataset. These results outperform earlier techniques, marking a significant improvement in accuracy.

Keywords: arabic handwritten text recognition, deep learning, document digitization, optical character recognition (OCR)

"Artificial Intelligence: Future Prospects for Achieving the Sustainable Development Goals"

46

Our Publication Page in the Conference's Book

A Demonstration Image of our contribution in the conference

ACHIEVEMENTS, CONCLUSION & FUTURE WORK

Journal: “International Journal of Telecommunications (IJT)”:

Our research, titled ”A Comprehensive Approach to Arabic Handwriting Recognition: Deep Convolutional Networks and Bidirectional Recurrent Models for Arabic Scripts,” has also been published in the International Journal of Telecommunications (IJT), Volume 15, on June 15, 2024. This study, authored by Ayman Saber Gad Mohammed, Ahmed Taha Ahmed Abd El Rahim, and Khaled Abd El Salam from the Faculty of Engineering’s Electrical Department, delves into cutting-edge deep learning techniques for recognizing Arabic handwriting. By leveraging Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (Bi-LSTM) networks

Acceptance Letter

ACHIEVEMENTS, CONCLUSION & FUTURE WORK

Conclusion:

Driven by our passion for developing advanced text-processing techniques in Arabic, we recognized their potential to bolster various areas we mentioned earlier (education, government, data analysis, and information retrieval). We were motivated us to push our own research, culminating in noteworthy results.

Our final model, the ResNet50V2, pre-trained on the Arabic Alphabet Character dataset and further trained on the main KHATT Dataset using advanced techniques and optimal hyperparameters, yielded significant results. This accomplishment fuels our commitment to further research in this domain, with the ultimate goal of achieving even more advanced levels of Arabic handwriting recognition.

Alongside our commitment for creating our very special application "Digital Ḍād- ض الرقمية" to implement our model and deploy it to deal with real life situations and data according to what it was trained on, and providing other services like Document Scanning, or Exam Grading, which resembles a contribution for automating very important processes that contribute to our society and culture.

ACHIEVEMENTS, CONCLUSION & FUTURE WORK

Future Work:

The Arabic language is rich in special characters and diacritics that can significantly alter the meaning of a word. As a result, future research will involve creating a more diverse dataset, and developing more sophisticated models that are specifically trained to recognize and interpret these characters accurately across a more diverse dataset. This could involve creating datasets from scratch; ones that focus on these special characters and incorporating more complex models that can capture the nuances of their usage and placement.

We plan to focus on collecting and preparing more comprehensive and diverse datasets. This includes incorporating more datasets that represent different age groups, writing styles, and handwriting conditions, such as varying levels of pressure, speed, and ink darkness, and if possible, across multiple mediums.

The development of synthetic datasets or the augmentation of existing datasets with techniques like rotation, scaling, and noise could also help improve the model's robustness against real-world conditions and make it a powerful tool to be used in real-world applications.

Thank You