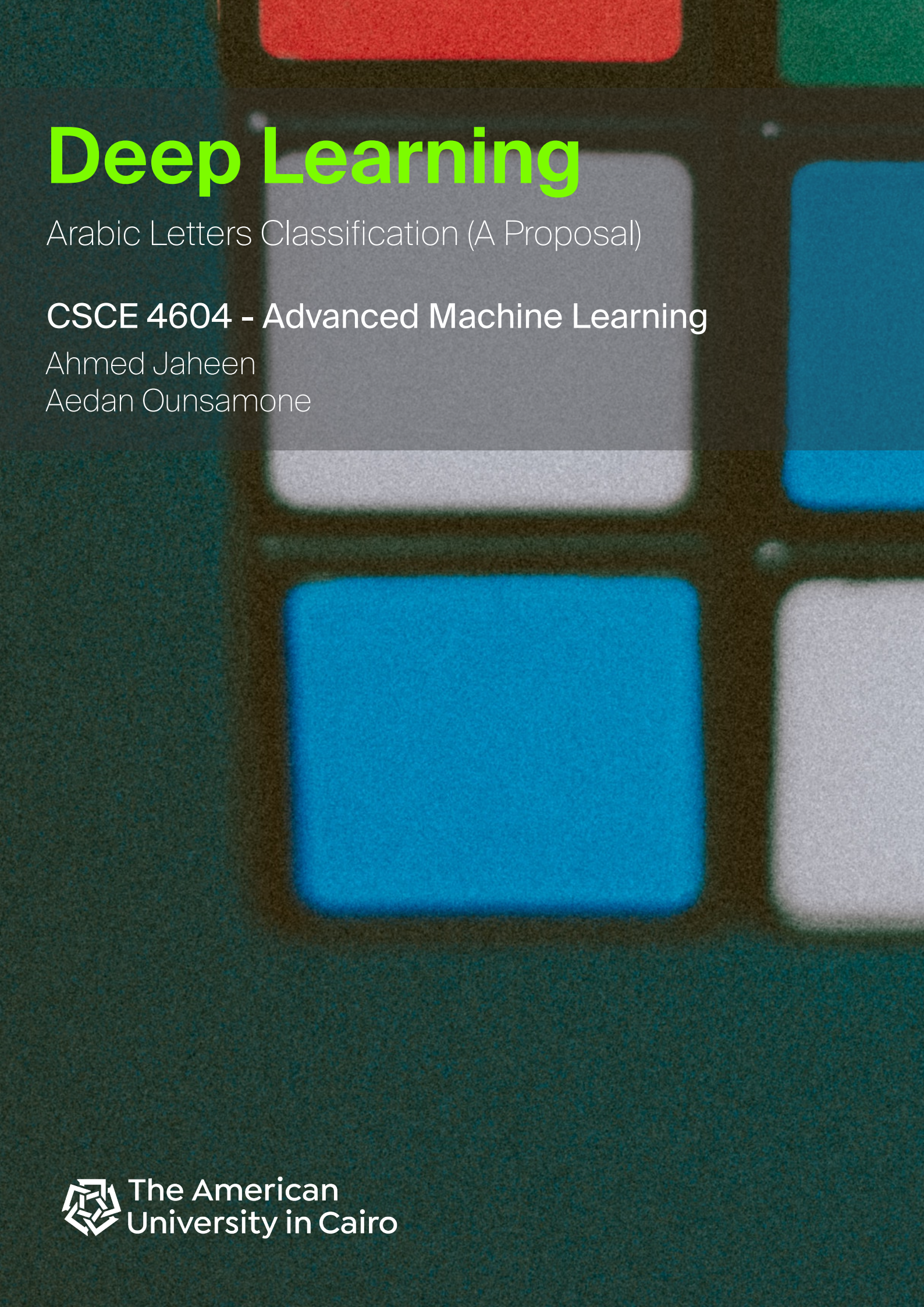


Deep Learning

Arabic Letters Classification (A Proposal)

CSCE 4604 - Advanced Machine Learning

Ahmed Jaheen
Aedan Ounsamone

The American University in Cairo

Deep Learning

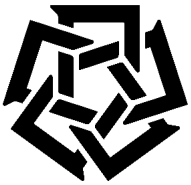
Arabic Letters Classification (A Proposal)

by

Ahmed Jaheen
Aedan Ounsamone

Student Name	Student ID
Jaheen	900212943
Ounsamone	900237691

Faculty: School of Science & Engineering, Department of Computer Engineering
Instructor: Prof. Moustafa Youssef | Spring 2024



Contents

1

Problem Statement

The advancement in text recognition on scanned images has opened up numerous possibilities, from searching texts in extensive documents to automating postal sorting and editing printed documents. Arabic handwriting recognition, given its unique challenges, has garnered attention later than other scripts, leading to diverse methodologies tailored for different image types. On the other hand, recognizing Arabic characters and digits is pivotal due to the script's cursive nature and shape variations, crucial for digital archiving, educational tools, and aiding the visually impaired. Traditional machine learning and basic neural networks often fall short in capturing the script's complexity, highlighting the need for more approaches.

2

Project Overview

2.1. Motivation

The advent of deep learning, particularly convolutional neural networks (CNNs) like EfficientNet, has shown promise in handling complex image classification tasks with higher accuracy and efficiency. EfficientNet B7, known for its scalability and performance, offers a promising solution to the Arabic letters classification problem by leveraging its deep architecture and advanced image processing capabilities. In this project, our goal is to develop a model capable of accurately identifying new images as Arabic letters or numbers. This task can be approached using either machine learning classification techniques or deep learning methodologies such as Convolutional Neural Networks (CNNs). For our solution, we will leverage a CNN, which is known for its superiority in image classification challenges. CNNs effectively distill an image into a highly abstract representation through its layer filters, enhancing the precision of the prediction process.

2.2. State of the Art

Current state-of-the-art solutions in Arabic letter classification include various deep learning models, but there's limited application of EfficientNet B7 in this domain. This project proposes to benchmark the performance of EfficientNet B7 against existing models as shown below, aiming to surpass current accuracy levels.

Paper	Year	Dataset	Method	Data enhancement	Results
Ghanim et al. [41]	2020 IEEE	IFN/ENIT Arabic database (words)	CNN (AlexNet)	Matching, ranking,	95.6% (Accuracy)
Poznanski and Wolf [16]	2016	IFN/ENIT	CNN—multiple separate and parallel fully connected layers	Dropout applied at fully connected layers and Weight decay data augmentation [rotation, shear,	97.07% (Accuracy)
Almodfer et al. [24]	2017	IFN/ENIT	Multi-Column Deep Neural Networks (MCDNN) *they refer to CNN as DNN	–	8.5% error rate
El-Melegy et al. [33]	2019	AHDB	CNN	Data augmentation	97.8% (Accuracy)
Almodfer, Rolla, et al. [25]	2017	IFN/ENIT	CNN (AlexNet + ReLU)	dropout	92.13% (Accuracy)
			CNN (AlexNet + Tanh)		92.55% (Accuracy)

Figure 2.1: Different Models Accuracy

2.3. Evaluation Metrics

We will employ the metrics of Accuracy, Precision, Recall, and F1-score to evaluate our model's performance. Additionally, Log loss will be considered for refining our model, offering insights into the precision of our predictions. Below, we break down the importance of each metric in relation to our task:

2.3.1. Accuracy

This metric offers a straightforward measure of our model's overall effectiveness by calculating the proportion of correct predictions out of all predictions made. It's particularly relevant in scenarios where the distinction between misclassified letters or digits is not a primary concern.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

2.3.2. Precision

Defined as the fraction of true positive predictions in relation to all positive predictions made, Precision helps us assess the model's false positive rate. High precision indicates fewer false positives, a critical aspect when aiming to minimize incorrect letter or digit recognition.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

2.3.3. Recall (Sensitivity)

Recall measures the proportion of actual positives correctly identified by the model, making it crucial in cases where missing a true positive (resulting in a false negative) carries significant consequences. It guides model selection when the cost of misclassification is high.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

2.3.4. F1 Score

By harmonizing Precision and Recall into a single metric, the F1 Score provides a balanced view of the model's performance, especially useful in situations with uneven class distributions or when the costs of false positives and negatives differ markedly.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.3.5. Log Loss

This metric quantifies the accuracy of a classifier by penalizing false classifications. Minimizing Log Loss across predictions indicates a model's reliability in estimating probabilities closely aligned with the actual outcomes. It provides a nuanced perspective on the model's confidence in its predictions.

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

where:

- N is the number of observations.
- y_i is the actual label of the ith observation, which can be 0 or 1.
- $\hat{y}_i$ is the predicted probability of the ith observation being class 1.
- The sum is over all observations in the dataset.

Each of these metrics brings a unique lens through which we can view our model's performance, enabling a comprehensive evaluation of its ability to classify Arabic letters and digits accurately.

2.3.6. Confusion matrix

A confusion matrix is a way to understand how the model is behaving across different classes. In the confusion matrix, the number of correct and incorrect predictions are summarized with count values and broken down by each class. Below is an example of a confusion matrix.

Figure 2.2: Confusion matrix Example

Observe that as the values increase along the diagonal starting from the upper left corner, it indicates improved performance by the model.

2.4. Datasets and Inputs

2.4.1. Arabic Handwritten Alphabets, Words and Paragraphs Per User

The dataset comprises 65 distinct Arabic characters, capturing their variations at the beginning, middle, end, and in isolation, alongside 10 unique Arabic words that collectively include all the characters, and 3 distinct paragraphs. It was anonymously sourced from 82 contributors, with each being instructed to scribe every character and word 10 times. Contributors are anonymously yet uniquely identified by a userid, which is associated with their contributed characters, words, and paragraphs. Altogether, the dataset features 53,199 images of characters, 8,144 images of words, and 241 images of paragraphs.

Examples from the dataset:

		Alphabet Variations				
Group	Alphabet	Regular	Begin	Middle	End	No. of different forms
1	Alif	ا	ا	ـا	ـا	2
2	Beh	ب	بـ	ـبـ	ـب	4
3	Jeem	ج	جـ	ـجـ	ـج	4
4	Dal	د	د	ـد	ـد	2
5	Raa	ر	ر	ـر	ـر	2
6	Seen	س	سـ	ـسـ	ـس	4
7	Sad	ص	صـ	ـصـ	ـص	4
8	Tah	ط	طـ	ـطـ	ـط	2
9	Ain	ع	عـ	ـعـ	ـع	4
10	Feh	ف	فـ	ـفـ	ـف	4
11	Qaf	ق	قـ	ـقـ	ـق	4
12	Kaf	ك, ک	كـ	ـكـ	ـك, ـک	6
13	Lam	ل	لـ	ـلـ	ـل	4
14	Meem	م	مـ	ـمـ	ـم	4
15	Noon	ن	نـ	ـنـ	ـن	4
16	Heh	ه, ہ	هـ	ـهـ	ـه	5
17	Waw	و	و	ـو	ـو	2
18	Yaa	ي	يـ	ـيـ	ـي	4
					Total	65

Figure 2.3: Arabic Alphabets with variations collected in the dataset

اذن	صخر	المستضعفين	الأبجدية	فسيكفيكهم
غزال	غليظ	قشطة	شطيره	محراث

Figure 2.4: Arabic words collected in the dataset

Paragraph-1:
هذا مقال عن تغير المناخ. يمكن أن يكون سبب تغير المناخ في العالم بسبب الأنشطة المختلفة. عندما يحدث تغير المناخ ؛ درجات الحرارة يمكن أن تزيد بشكل كبير. خلال القرن الماضي ، أطلقت الأنشطة البشرية كميات كبيرة من ثاني أكسيد الكربون وغازات الدفيئة الأخرى في الغلاف الجوي.
Paragraph-2:
بدلاً من اتخاذ قرارات محددة بشأن المسار الوظيفي ، أعتقد أنه يجب عليك أن تكون فضوليًا. احصل على فضول حول الطريقة التي يعمل بها العالم. لاحظ اهتماماتك الخاصة وابحث عن طرق صغيرة يمكنك من خلالها ممارسة الشغف في شيء ما. حتى إذا لم تتمكن من العثور على طريقة لكسب المال منه حتى الآن.
Paragraph-3:
أنظر إلى النجوم وليس إلى أسفل عند قدميك. حاول أن تفهم ما تراه ، وتساءل عما يجعل الكون موجودًا. كن فضولي.

Figure 2.5: Arabic paragraphs collected in the dataset

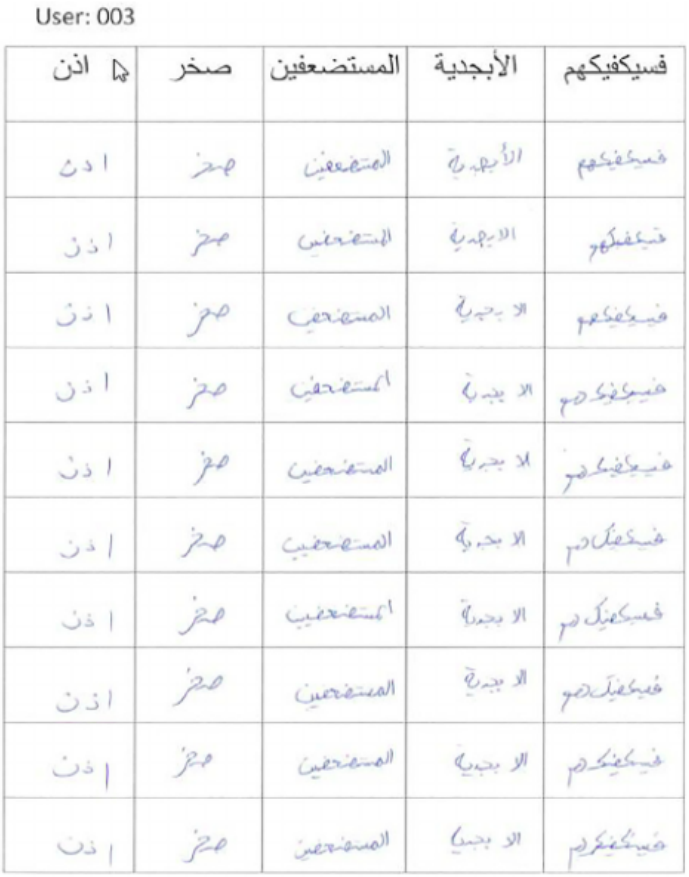


Figure 2.6: Arabic words handwritten by a user

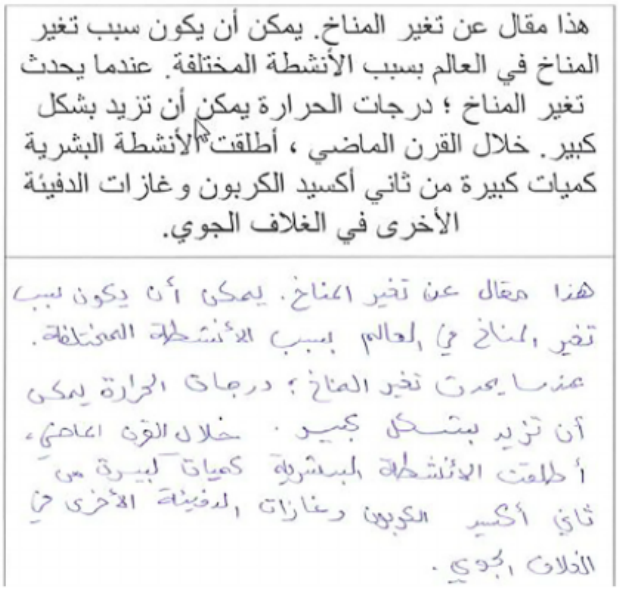


Figure 2.7: Arabic paragraph handwritten by a user

The handwritten forms from all the 82 users were color scanned at 300dpi resulting in an image resolution of 2480Œ3507 pixels. These scanned images are provided as raw data in the folder named raw_dataset in the public repository.

2.4.2. Arabic Handwritten Digits Dataset

The Arabic Digits Dataset is derived from MADBase, a modified database of handwritten Arabic digits featuring 60,000 images for training and 10,000 images for testing. Authored by 700 contributors, MADBase includes each digit (0 through 9) written ten times by each individual.

Examples from the dataset

Figure 2.8: Arabic numbers

2.4.3. Arabic Handwritten Characters Dataset

The Arabic Letters Dataset consists of 16,800 characters penned by 60 individuals aged between 19 and 40 years, with 90% being right-handed. Each participant inscribed each letter (from 'alef' to 'ya2') ten times. These images were captured at a 300 dpi resolution. Using Matlab 2016a, each character block was automatically segmented to identify the coordinates of every block. The dataset is divided into a training set, containing 13,440 characters (about 480 images per category), and a test set, with 3,360 characters (approximately 120 images per category). It's important to note that the contributors for the training and test sets are distinct, ensuring no overlap. Furthermore, the selection of contributors for the test set was randomized to prevent the dominance of any single institution, thereby enhancing the test set's diversity.

Examples from the dataset

Figure 2.9: Arabic characters

2.4.4. Selected Dataset

After careful consideration of the different aspects related to the dataset, we decided to go on with the Arabic Handwritten Alphabets, Words and Paragraphs Per User dataset for the following reasons:

1. The size of these datasets is not large. As a result, training will not take a huge time and we will not face memory issues. Also, because of its good size, we can run the model on some online platforms such as google collab for training and testing.

2. The number of images in these dataset is very good and we will be able to build a good model based on it.
3. The existing Arabic alphabet identication datasets (Hijja, AHCD datasets) do not provide any user information. The existing writer identication of Arabic text datasets (IFN/ENI, KHAT, QUWI datasets) only provide handwritten words or paragraphs and do not contain alphabets. This dataset fills in this gap.
4. Some new models in the field of Arabic Letter Classification were trained and reported results on this dataset. As a result, we will be able to test the performance of our final model against several models.

2.5. Survey of the available models.

We will develop a CNN model designed for high-performance image recognition. The specifics of the model's design and structure will be covered in subsequent sections, but first, let's delve into foundational elements like convolution layers, activation functions, dropout techniques, and optimization algorithms. Essentially, CNNs operate by processing an image through a sequence of convolutional layers, applying non-linear transformations, reducing dimensions via pooling, and making predictions through fully connected layers. The outcome is either a specific class identification or a set of class probabilities that most accurately represent the image. Understanding the function and significance of each layer is crucial.

2.5.1. CNN

To understand how a convolutional layer works, envision a scenario where a flashlight illuminates a 5x5 area on the top left corner of an image. Imagine this flashlight moving across the entire surface of the image. In the context of machine learning, this flashlight represents a filter (also known as a neuron or kernel), and the illuminated area is referred to as the receptive field. This filter comprises an array filled with numerical values, termed weights or parameters. It's crucial that the filter's depth matches that of the input image to ensure compatibility in the calculations, resulting in a filter dimension of 5x5x3. Considering the filter's initial position at the top left corner, as it moves or convolves across the image, it performs element-wise multiplications with the image's pixel values. The sum of these multiplications yields a singular value for each position of the filter on the image. This operation is replicated across every part of the input image, shifting the filter one unit to the right at each step until the entire image has been covered. This process results in an activation map or feature map, essentially a 28x28x1 array of numbers. The creation of a 28x28 array stems from the 784 unique positions a 5x5 filter can occupy on a 32x32 input image, each contributing to the comprehensive feature map.

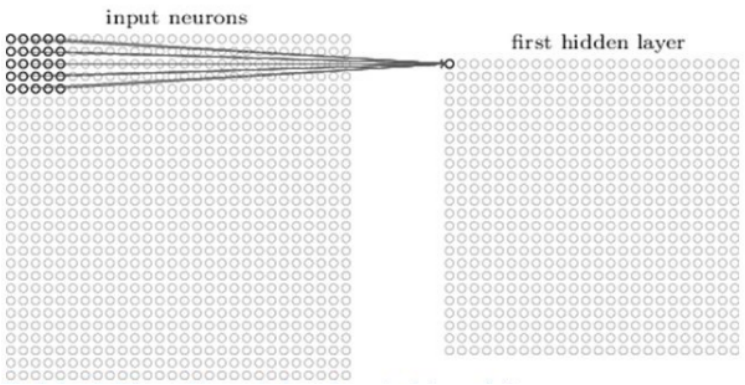


Figure 2.10: A visualization of 5x5 filter

2.5.2. Hybrid Model Using Convolutional Neural Networks (CNNs) and Bi-Directional Short Long Term Memory Networks (BDLSTMs)

The paper, authored by Mohammad Fasha, Bassam Hammo, Nadim Obeid, and Jabir Al-Widian, focuses on the development of a deep learning model for recognizing Arabic text, a task complicated by the cursive nature, joint writing scheme, and the presence of numerous ligatures in Arabic script. The model utilizes a hybrid deep learning network combining Convolutional Neural Networks (CNNs) and Bi-Directional Short Long Term Memory networks (BDLSTMs) to recognize Arabic printed text across 18 different font types, including those mimicking handwritten scripts, without needing character segmentation. Tested on a custom dataset containing over two million word samples, the model demonstrated high accuracy, achieving a 98.76% Character Recognition Rate (CRR) and a 90.22% Words Recognition Rate (WRR) across all tested fonts. However, performance dropped when applied to unseen datasets or noisy images, indicating areas for future research. The authors have made the model, datasets, and toolkit for dataset generation publicly available, supporting further development and application in Arabic optical character recognition.

Architecture

The model consists of a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN), specifically a Bi-Directional Long Short-Term Memory (BDLSTM) network, structured in an end-to-end format that facilitates word-level recognition without necessitating character segmentation. The CNN component is designed with layers of varying filter sizes and pooling windows to effectively capture text features from input images, which are then resized to fit the model's requirements. This resizing is critical for maintaining the integrity of the text's appearance despite the shift variance inherent in CNNs.

After feature extraction by the CNN, the BDLSTM component analyzes the sequence or temporal aspects of the input, enabling the recognition of characters and their order within words. This sequence analysis is crucial for understanding the cursive and connected nature of Arabic script. The model employs a Connectionist Temporal Classification (CTC) loss function to map the observed sequences to the actual text, considering the potential for each character to be represented by multiple sequence observations.

Experiments and results from training and testing the model are detailed, demonstrating its potential effectiveness in Arabic text recognition. The system is designed to be open and accessible, with its implementation shared publicly for further development and application. This research represents a significant step towards addressing the unique challenges of Arabic text recognition, with promising implications for the advancement of Optical Character Recognition (OCR) technologies for Arabic script.

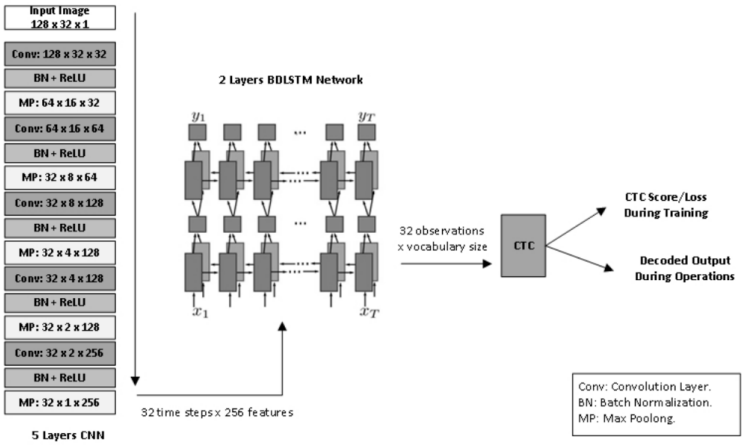


Figure 2.11: BDLSTMs Architecture

Results:

The model showed high accuracy in Character Recognition Rate (CRR) and Words Recognition Rate (WRR) across various datasets, including those with single and multiple fonts, and larger word samples. Although performance slightly decreased with disjoint datasets and in the presence of noise, the model demonstrated promising generalization capabilities, even on untrained datasets and external databases. The findings suggest the model's effectiveness in Arabic text recognition with potential for further improvements in robustness and accuracy through advanced post-processing and noise mitigation techniques.

#	Experiment Name	Dataset	Validation Accuracy %		Test Accuracy %	
			CRR	WRR	CRR	WRR
1	Single font model	1	98.37	90.28	97.5	85.18
2	Single font model – larger dataset	2	99.148	94.66	99.044	94.29
3	Two fonts model	3	99.93	99.5	99.88	99.2
4	(18) fonts, duplicate words across fonts types	4	99.38	94.84	99.27	94.32
5	(18) fonts, unique words across the dataset	5	98.81	90.53	98.76	90.22
6	Testing model generated in expriment 5 above on five character words	-	-	-	98.71	92.4
7	Testing model generated in expriment 5 above on APTI dataset – new font	-	-	-	85.15	23.7
8	Testing model generated in expriment 5 above with salt & pepper noise.	-	-	-	82.01	21.48
9	Testing model generated in expriment 5 above with salt & pepper and speckle noise.	-	-	-	77.29	14.18

Figure 2.12: BDLSTMs Result

.

Link: https://paperswithcode.com/paper/a-hybrid-deep-learning-model-for-arabic-text
Dataset Used: DARPA, APTI, PATDB, APTID/MF, and RCATSS
Model Source Code: https://paperswithcode.com/paper/a-hybrid-deep-learning-model-for-arabic-text

2.5.3. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) with Attention Mechanism

This article introduces a model that combines Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) with an attention mechanism for recognizing Arabic text in images. Initially, the model processes an image to extract features via a CNN, which are then arranged sequentially by a bidirectional RNN. Recognizing the potential for the bidirectional RNN to overlook crucial text segmentation preprocessing, the model incorporates an attention mechanism. This mechanism focuses on pertinent details within the feature sequences, enhancing the model's output. It facilitates an end-to-end learning process utilizing the conventional backpropagation technique, aiming to improve the accuracy and efficiency of Arabic text recognition in images.

Architecture

The proposed CNN-RNN attention model is structured into three primary sections: convolutional layers, recurrent layers utilizing bidirectional Long Short-Term Memory (LSTM), and recurrent layers combined with an attention mechanism. Initially, the model extracts features from input images using convolutional layers. These features are then processed by the recurrent layers, where the bidirectional LSTM labels the sequences sequentially. A unique aspect of this architecture is its utilization of a Connectionist Temporal Classification (CTC) layer, which serves as a dynamic loss function, adapting to the temporal variability in the data.

The model was trained and evaluated using two distinct datasets: ACTIV and ALIF. Prior to their introduction into the model, all images from these datasets undergo preprocessing to standardize their size. The convolutional layers dissect the feature maps column by column,

generating a sequence of features, each packed with critical information. Given CNNs' limitation in handling prolonged dependencies, these sequences are forwarded to the bidirectional LSTM for prediction. This LSTM setup not only addresses the disappearing gradient problem but also enhances text recognition by learning from both right-to-left and left-to-right text sequences.

Incorporating an attention mechanism, the model assigns weights to each feature sequence, thereby prioritizing the most relevant information for the task at hand. This approach significantly refines the model's output by focusing on the most pertinent features, enhancing overall recognition performance. The attention mechanism, alongside the LSTM network, contributes to a robust framework capable of accurately recognizing Arabic text from images, setting a new benchmark in the field.

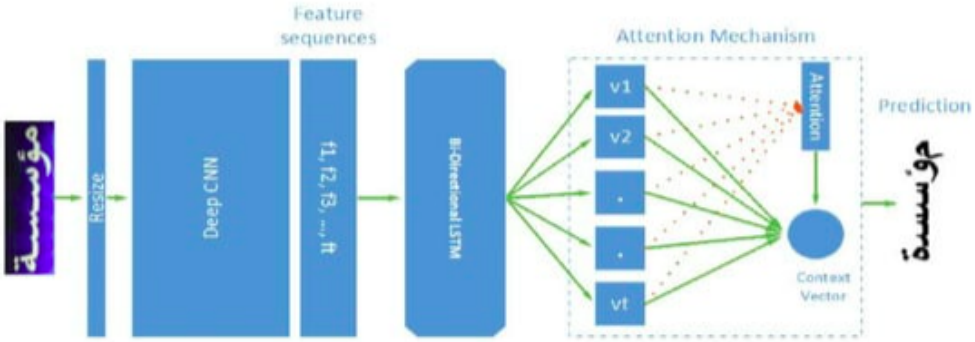


Figure 2.13: Attention Mechanism Architecture

Results:

The architecture discussed is proficient in recognizing Arabic text from natural scene images and video frames, focusing on word-level recognition rather than the more common character-level classification. It leverages two datasets, Alif and Activ, to establish new baseline results for Arabic scene text recognition. Using three evaluation metricsCharacter Recognition Rate (ChRR), Word Recognition Rate (WoRR), and Line Recognition Rate (LiRR)the architecture outperforms existing methods, including the widely used Tesseract OCR, especially in character and line-level accuracy. Notably, the architecture achieves a significant 98.73% ChRR, setting a new state-of-the-art for Arabic text recognition. It also compares favorably against other architectures, such as Deep Belief Networks (DBN), which utilize generative weights for feature detection, and Multi-Dimension Long Short-Term Memory Networks (MDLSTM), tailored for capturing text variations across both dimensions of input images. This highlights the architecture's effectiveness in handling the complexities of Arabic script in varied visual contexts.

On Alif Datasets:

Architecture	Alif_Test_1			Alif_Test_2		
	Ch_{RR} (%)	Li_{RR} (%)	Wo_{RR} (%)	Ch_{RR} (%)	Li_{RR} (%)	Wo_{RR} (%)
ConNet with LSTM	91.27	54.9	70.29	92.37	56.9	71.9
Deep belief net	89.98	40.05	60.58	87.8	43.7	62.78
HC with LSTM	85.44	60.15	53.4	87.14	62.30	50.31
ABBYY	82.4	25.99	50.0	83.26	26.91	49.80
Hybrid CNN-RNN	93.2	78.5	40.5	96.2	79.5	39.5
MLP_AE_LSTM	88.50	33.5	61.22	88.50	33.5	61.22
Hi-MDLSTM	95.55	71.33	85.72	96.55	70.67	85.71
Proposed Architecture	98.73	82.21	87.06	97.09	79.91	85.98

Figure 2.14: Attention Mechanism Result 1

On acTiV Datasets:

Architecture	Alif_Test_1			Alif_Test_2		
	Ch_{RR} (%)	Li_{RR} (%)	Wo_{RR} (%)	Ch_{RR} (%)	Li_{RR} (%)	Wo_{RR} (%)
ConNet with LSTM	91.27	54.9	70.29	92.37	56.9	71.9
Deep belief net	89.98	40.05	60.58	87.8	43.7	62.78
HC with LSTM	85.44	60.15	53.4	87.14	62.30	50.31
ABBYY	82.4	25.99	50.0	83.26	26.91	49.80
Hybrid CNN-RNN	93.2	78.5	40.5	96.2	79.5	39.5
MLP_AE_LSTM	88.50	33.5	61.22	88.50	33.5	61.22
Hi-MDLSTM	95.55	71.33	85.72	96.55	70.67	85.71
Proposed Architecture	98.73	82.21	87.06	97.09	79.91	85.98

Figure 2.15: Attention Mechanism Result 2

.
Link: https://paperswithcode.com/paper/a-hybrid-deep-learning-model-for-arabic-text
Dataset Used: ACTIV and ALIF
Model Source Code: None

2.5.4. Arabic Handwriting Classification using Deep Transfer Learning

This study explores the effectiveness of deep transfer learning models in classifying Arabic handwriting as native or non-native, addressing challenges faced by traditional text recognition algorithms. It tested seven deep learning transfer modelsAlexNet, GoogleNet, ResNet18, ResNet50, ResNet101, VGG16, and VGG19on two datasets containing images of Arabic handwriting. The aim was to identify the most efficient model for distinguishing between native and foreign writers' handwriting. The evaluation included both original and augmented datasets to enhance the training process and improve model accuracy. Among the models, GoogleNet achieved the highest accuracy, recording 93.2% with the normal dataset and 95.5% with the augmented dataset, in correctly identifying native Arabic handwriting. This demonstrates the potential of deep transfer learning models in accurately classifying complex handwriting characteristics associated with language nuances.

Architecture

This section outlines the use of seven transfer learning models for Arabic handwriting classification.

1. **AlexNet:** A convolutional neural network (CNN) by Krizhevsky et al. (2017), trained on over one million ImageNet database images, featuring eight layers (five convolutional and three fully connected), with the capability to classify 1000 classes.
2. **GoogleNet (Inception-V1):** A 22-layer pre-trained CNN also trained on ImageNet, notable for its use of 1 to 1 Convolution layers in the center and global average pooling at the end instead of fully connected layers.
3. **ResNet Models (ResNet-18, ResNet-50, ResNet-101):** Introduced by He et al. (2016), these networks employ skip connections across layers to facilitate training of deep networks. ResNet-18 has 18 layers, ResNet-50 has 50 layers, and ResNet-101 has 101 layers, each trained on one million images from 1000 categories.
4. **VGG16:** Developed by Simonyan and Zisserman (2015), this CNN achieved a top-5 testing accuracy of 92.7% on ImageNet, consisting of 16 weight layers (13 convolutional, three dense) among a total of 21 layers.
5. **VGG19:** Similar to VGG16 but with 19 layers, trained on the ImageNet database with 1000 categories.

Additionally, data augmentation techniques such as mirroring and random crops were employed to enhance the datasets. The effectiveness and analysis of these models and methodologies are detailed in the subsequent sections.

Results:

Performance measure of each DL model using both original and augmented data

DL Model	Original Data			Augmented Data		
	Acc	Spec	Sens	Acc	Spec	Sens
AlexNet	78.3	74.2	82.3	75.0	79.3	70.7
GoogleNet	93.2	92.4	93.9	95.5	93.9	97.0
ResNet18	78.8	75.8	81.8	80.6	78.8	82.3
ResNet50	78.5	75.3	81.8	82.8	87.9	77.8
ResNet101	78.5	83.8	73.2	81.6	79.3	83.8
VGG16	50.3	0.6	0.7	78.5	78.3	78.8
VGG19	79.0	74.2	83.8	78.0	91.4	64.6

Figure 2.16: BDLSTMs Architecture

.

Link: https://www.researchgate.net/publication/313891953_Arabic_Handwritten_Characters_Recognition
Dataset Used: IESK-arDB and Arabic Handwritten Character Dataset (AHCD)
Model Source Code: None

2.5.5. Activation layers

An activation node can be integrated at the output stage of any neural network or between two neural networks. Its primary role is to define the neural network's output, such as binary responses like yes or no. This node transforms the input values to a range between 0 and 1 or -1 and 1, depending on the specific function used. Various activation functions, including relu, linear, sigmoid, and softmax, are employed to achieve this mapping.

2.5.6. Dropout

It is a technique used to mitigate overfitting in neural networks, ensuring that the model does not become overly dependent on the training data. This is achieved by randomly omitting a subset of units (including both hidden and visible ones) within the network during training, thereby hindering the formation of complex co-adaptations among data points.

2.6. Project Benchmark evaluation

For our benchmark, we'll employ a straightforward (vanilla) CNN model and conduct training/testing with the same dataset utilized for our primary model according to "HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT- Database". This basic CNN architecture will include:

- A single Convolutional layer featuring 16 filters and a window size of 3, designed to identify fundamental patterns such as edges within the input images.
- A single Pooling Layer aimed at downsampling the input, facilitating the model's ability to infer features, thereby minimizing overfitting. This also cuts down on the quantity of parameters that need learning, thus shortening training duration.
- An output layer equipped with 38 neurons, corresponding to the number of output classes, and employing a softmax activation function to handle multiple classes. Each neuron will output the probability of its respective class.
- The training of our model will be executed using the Adam Optimizer and cross entropy (Log loss) for the loss function. A batch size of 20 is selected to decrease both training time and the risk of overfitting.

2.7. Description of the Model

2.7.1. Data Preprocessing

We will rescale the images by dividing every pixel in the image by 255 to make them into range [0, 1] to be normalized.

2.7.2. Categorical Labels Encoding

Examining the labels CSV files reveals that the labels are categorical, indicating a multi-class classification challenge. The structure of our outputs is as follows. Digits ranging from 0 to 9 are assigned category numbers from 0 to 9. Letters from 'alef' to 'ya2' are categorized with numbers from 10 to 37. To address this, we will employ One Hot Encoding via Keras to transform these categorical values. One-hot encoding converts integers into a binary matrix characterized by a single '1' in each row, representing the category, and '0's filling the remaining positions, effectively distinguishing each category in a format suitable for model training.

2.7.3. Input Images Reshaping

When utilizing TensorFlow with Keras for CNNs, the input required is a 4D array, also known as a 4D tensor, with the dimensions (nb_samples, rows, columns, channels). Here, 'nb_samples' denotes the total count of images (or samples) in the dataset, while 'rows', 'columns', and 'channels' represent the dimensions and the color channels of each image, respectively. Consequently, we will adjust the format of our input images into a 4D tensor with the dimensions (nb_samples, 64, 64, 1), taking into account that our images are 64x64 pixels in grayscale.

2.7.4. Implementation

We are going to implement over this already existed model Efficientnet_B7 For Arabic Letters Classification. However, we are going to use our own updated implementation:

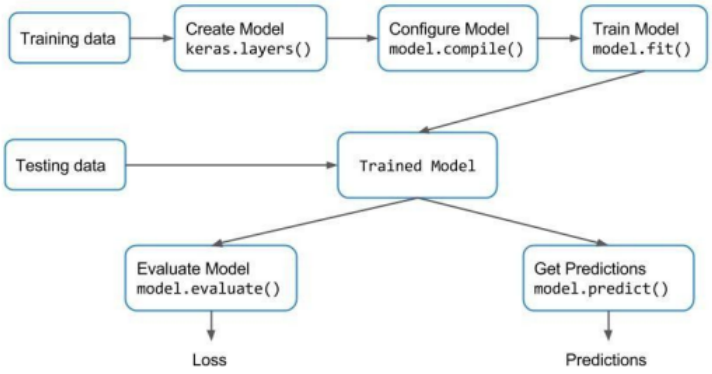


Figure 2.17: Model Implementation

2.8. Proposed Updates to the model

1. Adjusting parameters and incorporating transfer learning to enhance accuracy.
2. Employing image processing techniques for both preprocessing and augmentation to improve accuracy.
3. Implementing simpler deep learning architectures to reduce computational load and model size.
4. Conducting a study on the correlation between the number of classes and accuracy.
5. Comparing the performance across alphabets, words, and paragraphs, with a focus on enhancing accuracy specifically for paragraphs.
6. We will try using different regularisation techniques in an attempt to decrease the number of hidden layers, which will reduce the complexity of the model.
7. We will try to tune different activation functions.

2.9. How to evaluate our results?

In the previous discussion on evaluation metrics, we explored a variety of measures critical to our study. Among these, we will primarily focus on the following:

- **Accuracy**: This is the cornerstone metric for our analysis, crucial for determining state-of-the-art performance in the Arabic Letters Classification domain. It is universally applied across models and research papers addressing the Arabic Letters Classification challenge.
- **Confusion Matrix**: The confusion matrix is particularly vital for our project as it provides insight into the model's class-specific performance. It helps identify if the model underperforms for certain classes, effectively offering a detailed view of accuracy. The confusion matrix is also a common evaluation tool cited in relevant literature.
- **Precision and Recall**: These metrics are key to evaluating our model's class-wise performance, offering a nuanced understanding of its predictive capabilities.

In summary, while accuracy and the confusion matrix will be our primary focus, precision, recall, and the ROC curve will serve as supplementary metrics for a comprehensive evaluation of our model, though they will not be the foundational metrics.

2.10. Graduation Project Statement

None of us is graduating this semester.

2.11. Previous Data science Projects Done

Ahmed: Never worked with Data science project.
Aedan: Housing Price Prediction using logistical regression model in R. Israeli and Palestinian fatality analysis using Python and MySQL. University of Maryland Data Challenge, analyzing traffic data and phone usage while driving to identify trends.

2.12. Each Member Contribution

Both of us: Decided on the datasets to use, the baseline models. We also thought about the updates together, and wrote the introduction and the motivation part.
Ahmed: Did the survey about the available evaluation metrics and the datasets survey. Did the survey on the BDLSTMs model.
Aedan: Did the current state of the art models. Did the survey on the Attention Mechanism model and Deep Transfer Learning.

A

Bibliography

[1] CSDL | IEEE Computer Society. Www.computer.org, www.computer.org/csdl/proceedings-article/icdar/2003/196020890/12OmNyqiaUF.

[2] M. A. Almisreb, A. A. Jamel and S. A. R. Al-Haddad, "Arabic Handwritten Recognition Using Deep Learning: A Survey," in IEEE Access, vol. 9, pp. 15640-15655, 2021, doi: 10.1109/ACCESS.2021.3051732.

[3] Computer Vision with TensorFlow | TensorFlow Core. TensorFlow, www.tensorflow.org/tutorials/images/.

[4] M. Elhoseny, X. Yuan, Z. Yu, S. Mao, K. K. Loo, and A. K. Sangaiah, "Arabic Handwritten Recognition Using Deep Learning: A Survey," 2021. [Online]. Available: https://www.researchgate.net/publication/358068388_Arabic_Handwritten_Recognition_Using_Deep_Learning_A_Survey.

[5] A. El-Sawy, M. Loey, and H. EL-Bakry, "Arabic Handwritten Recognition Using Deep Learning: A Survey," 2022. Available: https://www.researchgate.net/publication/358068388_Arabic_Handwritten_Recognition_Using_Deep_Learning_A_Survey.

[6] S. S., "Complete Architectural Details of all EfficientNet Models," Towards Data Science, https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142

[7] OShea, K., & Nash, R. (2017). An Introduction to Convolutional Neural Networks. Towards Data Science. Available at: https://towardsdatascience.com/deep-convolutional-neural-networks-ccf96f830178

[8] Blurredmachine. (2020, July 28). VGGNet-16 architecture: A complete guide. Kaggle. https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a-complete-guide