Arabic Document Classification by Deep Learning

Taghreed Alghamdi[1]
Department of Computer Science
College of Computer Science and
Engineering, University of Jeddah
Jeddah, Saudi Arabia

Samia Snoussi[2]
Department of Computer Science
and Intelligence, College of
Computer Science and Engineering
University of Jeddah, Jeddah,
Saudi Arabia

Lobna Hsairi[3]
Data Science Department
University of Jeddah and Tunis
ElManar University
Saudi Arabia, Tunisia

Abstract—**In this paper, we show how to classify Arabic document images using a convolutional neural network, which is one of the most common supervised deep learning algorithms. The main goal of using deep learning is its ability to automatically extract useful features from images, which eliminates the need for a manual feature extraction process. Convolutional neural networks can extract features from images through a convolution process involving various filters. We collected a variety of Arabic document images from various sources and passed them into a convolutional neural network classifier. We adopt a VGG16 pre-trained network trained on ImageNet to classify the dataset of four classes as handwritten, historical, printed, and signboard. For the document image classification, we used VGG16 convolutional layers, ran the dataset through them, and then trained a classifier on top of it. We extract features by fixing the pre-trained network's convolutional layers, then adding the fully connected layers and training them on the dataset. We update the network with the addition of dropout by adding after each max-pooling layer and to the fourteen and the seventeenth layers which are the fully connected layers. The proposed approach achieved a classification accuracy of 92%.**

Keywords—Arabic document; document classification; deep learning; convolutional neural network (CNN); pre_trained network

I. Introduction

Documents classification is traditionally considered an important task and the first step in several document image processing pipelines, including document retrieval, information extraction, and text recognition. The initial classification of documents into various predefined classes not only makes various document processing activities easier but also saves time. It's also possible to enhance document processing systems' overall efficiency [1]. A wide range of classification problems can be solved using the deep learning technique. The ability and flexibility for changeability in the deep learning model with a wide spectrum of datasets make the deep learning algorithm the most important method for the classification task [2]. Several approaches for document image classification have been proposed. There are three approaches to consider. The first category makes use of the document images' layout/structural similarity. Extracting the basic document components and then using them for classification is time-consuming. The creation of local and/or global image descriptors is the focus of the second category of work. These descriptors are then used to categorize documents. Extracting local and global features takes a long time. Finally, the third category of methods employs CNN to automatically learn and extract features from document images, which are then categorized. With CNN's improved success in recent years, it's become easier to distinguish images without having to extract hand-crafted elements. It's the most commonly used neural network model for image classification [6]. Deep learning is preferred in image processing applications because it produces fast and significant results. Several problems in image processing and understanding have been solved using deep learning methods, including document image classification, handwriting recognition, and blind image quality assessment. Previously, various shallow-structure learning methods and handcrafted features were used to solve these issues [7].

We use supervised deep learning algorithms to identify Arabic document images in this paper. Because Arabic is a very rich language with complicated morphology, it has a very different and challenging structure than other languages. Hundreds of millions of people utilize it, and its internet presence is continually expanding. Furthermore, Arabic has several distinct characteristics that render automated handling and understanding of the Arabic language difficult and interesting [3]. As a result, it's critical to create an Arabic text classifier to deal with this difficult language. The importance of document classification stems from the vast number of Arabic document images distributed through many sources from different classes, the majority of which contain a significant amount of important knowledge. Manually classifying Arabic document images takes a long time and is extremely difficult. However, it has become possible to classify Arabic document images into their appropriate classes without requiring human intervention. This paper aims to use deep learning to introduce a model for classifying Arabic documents. The proposed deep learning model is trained using several different Arabic document image classes. It can differentiate between different classes, so it operates as multiple classifiers. The model can then be used to classify an Arabic document image that has been supplied after training. This paper is organized as follows. The current section provides a quick overview of the paper's topic and structure. The second section discusses related document classification studies. The third section discusses image recognition using deep learning. The CNN classification system is defined briefly in the fourth section. The fifth section summarizes the evaluation and its results. The final section summarizes the paper and suggests future work.

II. Related Work

The proliferation and subsequent usefulness of deep learning techniques in a wide range of machine learning tasks have been extensively discussed in the literature over the last decade. Deep neural networks have shown outstanding results in a variety of fields, including document image classification [3].

The classification of document images for the Arabic language is a subject that has received little attention.

Abdulmunim et al. [4] proposed a header-words-based printed Arabic document image classification and retrieval system based on a decision tree classifier enhanced by the bagging technique. This document's Arabic header words are detected by the proposed system. In addition, sets of discriminative features are extracted from detected header words to correctly classify them to the appropriate class. Several structural and statistical features specific to Arabic scripts can be observed. On the official printed Arabic document dataset, the experimental tests score 97.35 % for precision. This dataset contains images of various types of official printed Arabic documents collected from a variety of official websites, including ministries, universities, government departments, and other official states. The dataset includes letters, records, books, forms, notices, administrative instructions, and other official Arabic documents.

Al-Khurabi et al. [5] applied this work to Arabic documents, proposing an Arabic document Image Classification system based on Artificial Neural Networks on a set of 120 captured document images types such as text, geometric, and photographic images. This system was divided into two parts, the first of which was image processing. The second part uses an Artificial Neural Network to classify document images based on the contents into the appropriate class. It received an overall recognition rate of 86%.

On the other hand, many works in the literature in various languages, including English, have addressed the topic of document image classification.

In the field of document classification, Afzal et al. [1] used the deep Convolutional Neural Networks model (CNN). They created a network (except for the last fully connected layer) using weights from AlexNet, which was trained on images from ImageNet, in this study. The proposed network has five convolutional layers, which give the fully connected layer a lot more features for classification. On the Tobacco-3428 Legislation dataset, with 100 samples per class used for training and validation, the proposed method achieved an accuracy of 77.6%.

Kölsch et al. [6] propose a two-stage approach that combines automated deep CNN feature learning with efficient training using Extreme Learning Machines (ELM). As compared to a previous Convolutional Neural Network (CNN) method in [1] accuracy was achieved at 83.24% on the Tobacco-3428 Legislation dataset, resulting in a relative error reduction of 25%.

It's important to highlight the following crucial differences between the proposed approach and previous approaches: Though deep CNN-based approaches have made considerable progress in recent years and are now the current state-of-the-art, training these networks takes a long time. For the classification of Arabic document images, we propose a convolutional neural network supervised deep learning algorithm. To identify the dataset of four classes as handwritten, historical, typed, and signboard, we use VGG16 pertained weights that were trained on ImageNet. We used a convolutional neural network classifier to classify different classes of Arabic document images obtained from various sources. We used VGG16 convolutional layers, ran the dataset through convolutional layers, and then trained a classifier on top of that for Arabic document image classification. We extract features from the pre-trained network's convolutional layers by freezing them, then adding the fully connected layers and training them on the dataset. We added dropout after each max-pooling layer and to the fourteenth and seventeenth layers, which are the fully connected layers, to update the network. The proposed approach had a classification rate of 92%.

TABLE I. Comparison between the Proposed Approach and Previous Approaches

Authors	**Approach**	**dataset**	**Language of Dataset**	**Classification rate**
Abdulmunim et al. [4]	Decision Tree Classifier and Bagging Technique to improve the performance of classification.	Official Arabic printed document images were collected from different authorized websites.	Document images in the Arabic language.	97.35%
al-Khurabi et al. [5]	The proposed system used Artificial Neural Network(NN)	Set of 120 captured document images types.	Document images in the Arabic language.	86%
Afzal et al. [1]	The proposed CNN model used pertained weights from a network (AlexNet)	Tobacco-3428 Legislation	Document images in the English language.	77.6%
Kölsch et al. [6]	The proposed CNN model used pertained weights from a network (AlexNet) and Extreme Learning Machines (ELM)s which provide real-time training	Tobacco-3428 Legislation dataset	Document images in the English language.	83.24%
The proposed work	The proposed approach is based on a convolutional neural network supervised deep learning algorithm. the CNN model uses pertained weights from a network (VGG16) with the addition of the dropout layer applied after each VGG block.	Arabic document images were collected from different sources.	Document images in the Arabic language.	92%

It should be noted that the pre-training in the proposed method did not previously apply to the classification of Arabic document images. The [1] is the first attempt to classify document images using a pre-trained model. Table I compares the proposed approach to previous methods in terms of the method used, type and language of datasets used in the experiment, and the accuracy achieved.

III. Deep Learning for Document Image Processing

DL has opened the door to a new era of AI applications where is a subset of machine learning. It depends on artificial neural networks and representation learning as it is capable of imitating the way the human brain operates by creating patterns and processing data [8][9].

A. Deep Learning for Document Classification

DL contains multilayered neural networks. Its commonly applied to document and image processing. Recurrent Neural Networks (RNN) and convolutional neural networks are two of the most common supervised deep learning architectures (CNN). A recurrent Neural Network (RNN) is suitable for modeling sequential data such as texts, audio, and time series. It has been widely used in machine translation, speech recognition. A convolutional neural network (CNN) is suitable for modeling static data such as images. It has been proved to be a very powerful tool in image processing. Indeed, in the areas of computer vision such as handwriting recognition, image classification. It has become a much better tool compared to previous tools [10], [11].

We are employing deep learning because of its ability to learn useful features from raw data, which makes it extremely useful when working with unstructured data. We realize that one of the most supervised deep learning techniques is the Convolutional Neural Network (CNN). CNN is widely employed in image processing because it performs in image classification and recognition and has significantly improved the efficiency of a number of machine learning tasks. It has since developed into a powerful and widely used deep learning model. The CNN architecture enables it to extract features from images automatically, eliminating the need for manual feature extraction. Different filters are used in the convolutional layer to convolve through images to create complex features that are then passed on to the next layer. This process continues until it reaches the last feature. This makes CNN highly accurate for image classification.

Convolutional Neural Networks (CNN) have shown their efficacy in the classification of document images. In the field of document image classification, Convolutional Neural Networks (CNN) are efficient. Different classification techniques are compared by Lecun et al. [27]. The results reveal that CNN outperformed all other methods when it came to dealing with the range of 2D shapes. As a result, we consider CNN as the model for the challenging tasks of Arabic document image classification [12], [13].

Our contribution is to use CNN to classify Arabic document images and extract information. The proposed model aims to understand CNN and apply it to Arabic document image recognition filters, a CNN extracts feature maps from 2D images.

B. Convolution Neural Network (CNN) Architecture

We use CNN as a model for categorizing Arabic document images. With minimal preprocessing, we built a Convolutional Neural Network to recognize Arabic scripts directly from pixel images. A CNN's central building block is the convolutional layer. The parameters of the layer are made up of a collection of learnable filters (or kernels) for recognizing features (like edges) that span the whole depth of the input image. Each filter convolves over the width and height of the input image during the forward transfer. Produce a features map by computing the dot product of the kernel and the input field. As a result, the network gains an understanding of the filters. When the filter detects a particular type of feature at a specific spatial location in the input, it activates. The function maps are then fed into a pooling layer, where identical convolutions are added one patch at a time. CNN also has a completely connected layer that categorizes performance into one of four classes. Almost all CNN architectures follow the same general design principles: sequentially adding convolutional layers to the input, regular spatial downsampling (Max pooling), and increasing the number of feature maps. Layers that are fully connected, activation, and loss functions are also provided. As a result, before introducing the proposed model, we'll quickly go through these layers. The first layer where it can remove features from images is the convolutional layer. Convolution is the process of reducing the size of an image while maintaining the relationship between pixels by filtering it with a smaller pixel filter. Filter (kernel) is simply a weighted matrix of values that has been trained to detect unique features. The basic concept behind CNNs is to spatially convolve the kernel on an input image to see if the function it's supposed to detect is present. Convolution is performed by computing the kernel's dot product with the input field and then generating a features map [14]. The convolution operation is shown in Fig. 1.

After each convolution process, the ReLU (Rectified-Linear Unit) was used. It replaces all negative pixel values in the function map with zero and is added per pixel. The rectifier function is used to increase non-linearity in the CNN and convert the linear model we train into a network with more expressive capabilities, which aids in faster and more efficient training [17], [18].

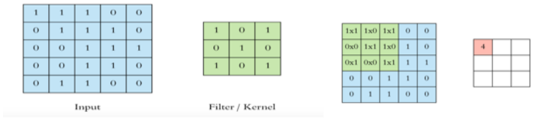

Fig. 1. Convolution Operation [13].

The pooling layer is usually inserted after each convolutional layer in a CNN, and it entails selecting a pooling operation, similar to a filter applied to feature maps, to reduce the representation's spatial size. This layer reduces the computational complexity of each map by minimizing its dimensionality while preserving important data. Overfitting can also be mitigated by pooling layers. The function map typically takes up more space than the pooling or filtering operation. As a result, we pick the maximum, average, or total values within these pixels as a pooling size to reduce the number of parameters. Fig. 2 depicts the maximum pooling operation. In CNN, the max-pooling approach is widely used, which involves placing a 2x2 matrix on the feature map and choosing the largest value in that box. The 2x2 matrix is passed around the function map from left to right, picking the largest value in each pass. These values are then combined to form a pooled feature map, which is a new matrix. It preserves the image's main features while reducing its scale. We'll use dropout to prevent overfitting, which happens when the proposed model is unable to predict new data labels. The dropout layer reduces the random set of activations in that layer to zero as data passes through it, effectively eliminating them from the layer [9].

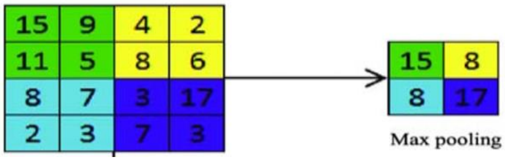

Fig. 2. Max Pooling Operation [14].

A fully connected layer takes the outputs of the convolution and pooling processes and converts them into labels with categories. Convolution and pooling output is flattened into a single vector of values, each representing a probability that defines a specific feature of the class. In the end, we can create a fully connected network to classify the dataset [14]. The fully connected layer is shown in Fig. 3.

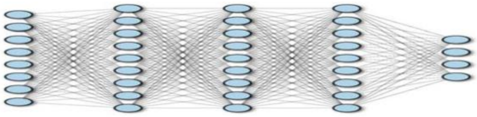

Fig. 3. Fully Connected Layer [14].

Fig. 4 shows the overview look of the proposed convolutional neural network. We can divide the model into seven sequences of layers.

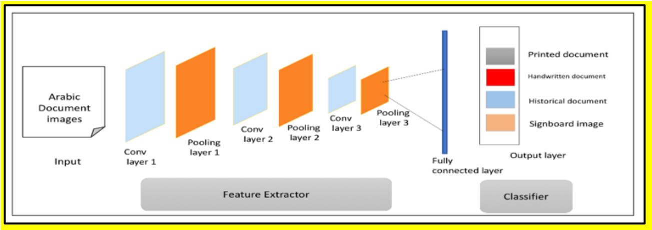

Fig. 4. General Architecture of the Proposed CNN Model.

C. CNN and Feature Extraction

In image processing, we can extract features from images using a variety of convolutional filters in the convolutional layer. Canny filters, for example, can detect edges. Canny normally takes a grayscale image as input and outputs an image that shows where strength discontinuities are located (i.e. edges). Filters aren't described in CNN. During the training phase, the value of each filter is learned. CNN can extract additional meaning from images that humans and human-designed filters might not be able to find. We can get more abstract and in-depth information from a CNN by stacking layers of convolutions on top of each other. Convolution may also be capable of performing hierarchical feature learning, which is thought to be how to brains identify objects. CNN is very effective in image recognition because of its ability to discover abstract and complex features [15], [16].

IV. CNN Classification System

The general architecture of the proposed Convolutional Neural Network (CNN) model is designed for the recognition of Arabic document images, as shown in Fig. 4. We need to do some pre-processing on the document images first, such as resizing them because CNN uses a fixed-size input image So, resize the document images that we'll be experimenting with.

The first step in the proposed approach is to prepare the data. We need to set up training and testing data and reshape it into the right size before we can create the network. The original dataset folder had four subfolders, each containing four different types of Arabic document images. Arabic handwritten documents, Arabic printed documents, Arabic historical documents, and Arabic signboard images are the four main classes. We divided the original dataset folder into training and validation dataset folders using the split-folder python package, with the same four classes in each folder. For the training dataset, we used 80% of document images and 30% for the validation dataset. The training dataset was used to train the model, and the validation dataset was used to fine-tune it.

A. Data pre-processing and Data Augmentation

We'll use Keras, an open-source python library for developing and testing deep learning models, to preprocess the document images. It includes the ImageDataGenerator class, which defines the image data preparation and augmentation configuration and is a powerful tool for image augmentation and feeding these images into the proposed model. All data in the Arabic handwritten document folder will be automatically labeled as Arabic handwritten documents by the ImageDataGenerator. It's the same for the rest of the folders. Data is effectively ready to be transferred to the neural network in this way. The names of the classes will be created automatically from the names of the sub-directories, so we won't have to identify them explicitly. The dataset directory structure is shown in Fig. 5.

Image augmentation, on the other hand, helps to minimize overfitting and improves model generalization by generating more training data based on existing training data. It takes place in memory, and the generators make it simple to set up data for training and testing. It has a lot of resizing, rotating, zooming, and flipping options. Before providing images as input to a deep learning model for training or evaluation, the pixel values in the images must be scaled. The ImageDataGenerator class will rescale pixel values from 0-255 to the preferred range of 0-1 for neural network models [19], [20].

B. Dataset and CNN Training

We have a dataset directory where we will store all of the document image data, as shown in Fig. 5. We'll have subdirectories for each class under each dataset directory, where the actual image files will be stored.

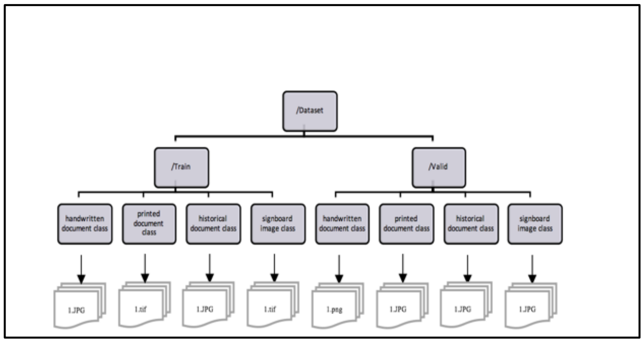

Fig. 5. Dataset Directory Structure.

The training dataset will be stored in the dataset/train/ directory. During training, we may also have a dataset /validate/ for a validation dataset. The document images were categorized into four classes in the train folder. For example, the handwritten document subfolder contains all document images related to the Arabic handwritten document, and so on.

Data is ready to be fed into the model after the required pre-processing. Below, we'll go over the information in greater depth.

Transfer learning is a deep learning technique that involves training a neural network model on a problem that is close to the one being solved. The initial blocks of CNN in the computer vision domain extract low-level features such as edges, shapes, corners, and so on. We transfer the information (features, weights) from a pre-trained network and use them to identify new images since we know that initial blocks use more computational resources. Since the final blocks in CNN are more focused on image data, we freeze the top layers and use the bottom layers as needed. For feature extraction, we use all of the convolution layers. The completely connected and dense layer is then replaced with the layers to identify the document images as (handwritten, historical, printed, and signboard).

We'll use the VGG16 architecture, which is based on ImageNet, a research project that aims to build a broad database of images and labels. There are over 14 million images in this dataset, divided into 1000 categories. VGG16 model architecture for image classification consists of 13 convolutional layers with 3*3 filters, max-pooling, two fully connected layers, and one SoftMax classifier based on the ImageNet database. The feature extractor, which is made up of VGG blocks, and the classifier, which is made up of fully connected layers and the output layer [17], [22], [23], are the two main components of VGG16. We can use the model's feature extraction section and add a new classifier section that is specific to our Arabic document images. Specifically, during training, we may keep the weights of all convolutional layers set and only train new fully connected layers that will learn to understand the features extracted from the model and create a document image classification.

We'll load the dataset and classes, and then begin the training process by learning ImageNet's prior weights. The network was also updated with the addition of dropout. In this case, a 30% dropout is applied after each max-pooling layer, as well as to the fourteenth and seventeenth layers, which are fully connected. We also changed the final layers of the network to identify the classes. Fig. 6 depicts a summary of the model's proposed architecture. One of the reasons we chose the VGG16 model as the base classifier model for the classification of Arabic documents since it has been shown in [3] to be one of the most effective models for dealing with document classification tasks.

C. CNN Implementation

Fig. 7 shows that there are five blocks in total: the first two have two convolutional layers, followed by ReLU and max-pooling layers, while the last three have three convolutional layers, followed by ReLU and max-pooling layers. The first and second convolutional layers are made up of 64 feature kernel filters, each of which is 3*3 in size. The dimensions of the input image (RGB image with depth 3) changed to 500x750x64 as it moved through the first and second convolutional layers. The output is then sent to the max-pooling layer.

The third and fourth convolutional layers are made up of 128 feature kernel filters that are 3*3 in size. The output will be reduced to 250x375x128 after these two layers are accompanied by a max-pooling layer. Convolutional layers with kernel size 3*3 make up the fifth, sixth, and seventh layers. 256 function maps are used for all three. A max-pooling layer comes after these layers.

Two sets of convolutional layers, each with a kernel size of 3*3, are used in the eighth to thirteenth layers. 512 kernel filters are used in each series of convolutional layers. The max-pooling layer comes after these two. The fourteenth through seventeenth layers are fully connected hidden layers with 256, 64, 32, and 8 units, respectively, followed by a four-unit SoftMax output layer (eighth layer). The dropout layer is added after each max-pooling layer, as well as the fourteenth and seventeenth fully connected layers, to prevent overfitting. The actual number of classes is represented by the eighth sheet.

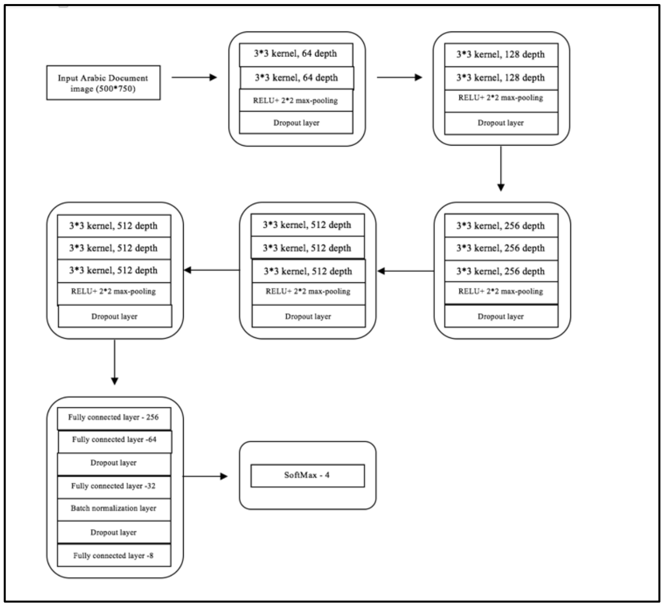

Fig. 6. The Proposed Architecture of the Model.

In this way, as in Fig. 7, CNN transforms the pixel values in the original document image to the final class, layer by layer. While some layers contain parameters, some don't. The convolution and fully connected layers, in particular, perform transformations that are dependent on both the activations in the input volume and the parameters (the weights and biases of the neurons). The Relu/pooling layers, on the other hand, will follow a fixed function. In the convolutional and fully connected layers, we train the parameters. The trained model will be prepared to recognize the document image in the test data. As a result, we can classify document images into four main categories: handwritten, historical, printed, and signboard. Training and testing the model is the third step. Finding kernels in convolution layers and weights in fully connected layers to reduce the gaps between output predictions and given actual classes on a training dataset is the method of training a network. The backpropagation algorithm is a method for training neural networks that involve the use of a loss function and a gradient descent optimization algorithm. A loss function calculates a model's performance under specific kernels and weights using forwarding propagation on a training dataset, and learnable parameters, including kernels and weights, are modified according to the loss value using optimization algorithms which include backpropagation and gradient descent [21]. The adaptive learning rate (Adam) optimization algorithm was used to construct the model and perform random gradient descent training. We used the Adam optimization algorithm to change the weight of the relation between neurons so that the loss is reduced to a minimum or stops after several epochs. Adam is defined as one of the most popular optimization algorithms for optimizing neural networks in deep learning, based on an adaptive learning rate algorithm [25], [26]. Finally, we can start CNN training by providing the training data, the built model, and the current batch of data. Only the data specified for training play a significant role in reducing CNN error. For both the forward and backward passes, we feed the training data into the network. The validation data is only used to see how the CNN reacts to new data that is close to it. The validation data isn't used to train the network. After that, we save the CNN that has been trained and prepare for the testing process. Finally, we can evaluate the model using the testing data.

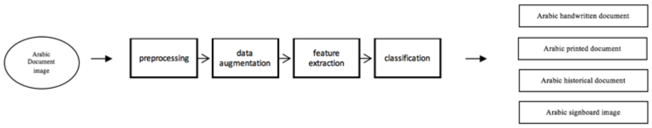

Fig. 7. CNN Implementation Steps.

V. Evaluation and Result

The proposed document collection contains 2373 documents regarding Arabic script, all of which are divided into four classes: handwritten, historical, printed, and signboard in Arabic script. These documents were collected from various sources. We must evaluate the model's performance and estimate the model prediction accuracy, which is how effective the model is at predicting the outcome of a new test dataset that has not been used to train the model, by comparing the expected result values to the actual result values. We use a Confusion Matrix, Accuracy, Recall, Precision, and F1- Score as the most common classification evaluation metrics. In the handwritten document image class, for example, we feed the handwritten document image into the trained model before the model prediction. We compare the prediction to the correct class after the model predicts that this is a handwritten document. As compared to the class of "handwritten document," the prediction is accurate. However, if it predicts that this image is a printed document, the comparison to the correct class will be incorrect. This process is repeated for each of the document images in the test data. We'll eventually get a count of how many test records the model correctly predicted and how many it incorrectly predicted. The Confusion Matrix is a tabular architecture of prediction results that includes counts of test records correctly and incorrectly predicted by the model. It provides information about the types of errors produced by the classifier. Table II displays a confusion matrix of four classes and 20 handwritten document images that have been misclassified as printed documents. The correct classifications are represented by the yellow cells on the diagonal, while the incorrect classifications are represented by the white cells. As can be shown, this provides a much more detailed view of the proposed model's efficiency.

Table III shows the classification accuracy, recall, precision, and f1-score are all factors to consider. The following Table III will clarify everything, shows the classification accuracy, precision, recall, and f1-score for each document type in the dataset. Accuracy is a common metric used by many researchers to evaluate the efficacy of classifiers. It is defined as the percentage of correct predictions for test data. It is easy to calculate by dividing the number of correct predictions by the total number of predictions. The proposed model demonstrates that it is capable of accurately recognizing various documents. Using this dataset to run the model, we were able to achieve a 92% accuracy rate.

Precision is expressed as a percentage of (true positives) correctly predicted out of the total number of positive results predicted by the model. The recall is calculated by dividing the number of positive results correctly classified by the total number of positive examples that should have been found.

TABLE II. CONFUSION MATRIX

Truth					
		Arabic handwritten document	Arabic historical document	Arabic printed document	Arabic signboard image
Predicted	Arabic handwritten document	136	0	0	0
	Arabic historical document	0	134	0	0
	Arabic printed document	20	0	109	0
	Arabic signboard image	0	3	13	62

TABLE III. CLASSIFICATION REPORT

Classification report				
	Precision	Recall	F1-score	Support
Arabic handwritten document	87%	100%	93%	136
Arabic historical document	98%	100%	99%	134
Arabic printed document	89%	84%	87%	129
Arabic signboard image	100%	79%	89%	78
Accuracy			92%	477
Macro avg	94%	91%	92%	477
Weighted avg	93%	92%	92%	477

The f1-score is computed by applying the harmonic mean of precision and recall [23], [24]. All the metrics formulae are shown in the equation (1), (2), (3), and (4).

$$Accuracy\ (acc) = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Recall\ (R) = \frac{TP}{TP+TN} \quad (2)$$

$$Precision\ (P) = \frac{TP}{TP+FP} \quad (3)$$

$$F-Measure = \frac{2\times P\times R}{P+R} \quad (4)$$

The classification report, shown in Table III, provides an overview of the proposed model's performance. The handwritten class results are shown in the third row. The 'support' column indicates how many class handwritten document images were included in the test data. The model performance for the historical class is shown in the fourth row. Tables II and III show how we can describe precision and recall for each of the classes. The precision for the handwritten class, for example, is calculated as the number of correctly predicted handwritten documents (136) out of all predicted handwritten documents (136+20=156), which amounts to 136/156=87%. The recall for the handwritten document, on the other hand, is the number of correctly predicted handwritten documents (136) divided by the number of real handwritten documents (136+0+0=136), which amounts to 136/136=100%. We can measure the precision and recall for the other three classes in the same way. On the other hand, the f1-score for a handwritten document is 93% because it is harmonic between accuracy and recall (2*0.87*1.00)/ (.87+1.00). The three classes are all computed in the same way. The proposed model has proven its effectiveness in classifying Arabic document images by achieving higher accuracy of 92%, as shown in Table III.

VI. CONCLUSION AND FUTURE WORK

The proposed Arabic document collection includes 2373 documents, which are divided into four categories: handwritten, historical, printed, and signboard in Arabic script. These documents were obtained from different sources, the majority of which contain a significant amount of knowledge. As a result, document classification is crucial. It takes a long time and is extremely difficult to manually identify Arabic document images. However, it is now possible to categorize Arabic document images into their appropriate classes. In this paper, we develop a system for classifying Arabic document images into four classes: handwritten, historical, typed, and signboard. The CNN supervised deep learning algorithm was used to create the proposed approach. The pre-trained model VGG16, which was trained on ImageNet, was used in the CNN model. We used the model's feature extraction part and added a new classifier part that is specific to the Arabic document images. Specifically, we may keep the weights of all convolutional layers fixed throughout training and only train new fully connected layers that will learn to understand the features extracted from the model and classify document images. We changed the network by adding dropout after each max-pooling layer and to the fourteenth and seventeenth fully connected layers. The proposed model has proven its effectiveness in classifying Arabic document images by achieving higher accuracy of 92%. We plan to work on big data Arabic document images in the future, and the final data is to apply the same techniques to keyword searches in deep learning classified documents.

ACKNOWLEDGMENT

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. {UJ-20-018-DR}. The authors, therefore, acknowledge with thanks the University of Jeddah technical and financial support.

REFERENCES

[1] Afzal, M. Z., Capobianco, S., Malik, M. I., Marinai, S., Breuel, T. M., Dengel, A., & Liwicki, M. (2015, August). Deepdocclassifier: Document classification with a deep convolutional neural network. In 2015 13th international conference on document analysis and recognition (ICDAR) (pp. 1111-1115). IEEE.

[2] Ezat, W. A., Dessouky, M. M., & Ismail, N. A. (2020). Multi-class Image Classification Using Deep Learning Algorithm. In Journal of Physics: Conference Series (Vol. 1447, No. 1, p. 012021). IOP Publishing.

[3] Das, A., Roy, S., Bhattacharya, U., & Parui, S. K. (2018, August). Document image classification with intra-domain transfer learning and stacked generalization of deep convolutional neural networks. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 3180-3185). IEEE.

[4] Abdulmunim, M. E., & Abass, H. K. (2019). Classification and Retrieving Printed Arabic Document Images Based on Bagged Decision Tree Classifier. AL-MANSOUR JOURNAL, (32).

[5] al-Khurabi, Abd Allah Ali& Mansur, Muhammad Abd Allah. 2004. Arabic document image classification using neural networks. Mansoura Engineering Journal،Vol. 29, no. 1, pp.1-8.

[6] Kölsch, A., Afzal, M. Z., Ebbecke, M., & Liwicki, M. (2017, November). Real-time document image classification using deep CNN and extreme learning machines. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 1, pp. 1318-1323). IEEE.

[7] AL-Saffar, A., Awang, S., Al-Saiagh, W., Tiun, S., & S Al-khaleefa, A. (2018). Deep learning algorithms for Arabic handwriting recognition: A review. International Journal of Engineering & Technology, 7(3.20).

[8] What Is Deep Learning? | How It Works, Techniques & Applications. https://in.mathworks.com/discovery/deep-learning.html.

[9] Deep Learning in Keras - Building a Deep Learning Model. https://stackabuse.com/deep-learning-in-keras-building-a-deep-learning-model/.

[10] Mu, R. (2018). A survey of recommender systems based on deep learning. Ieee Access, 6, 69009-69022.

[11] Demystifying AI, Machine Learning and Deep Learning, from https://developer.hpe.com/blog/demystifying-ai-machine-learning-and-deep-learning/.

[12] Shaheen, F., Verma, B., & Asafuddoula, M. (2016, November). Impact of automatic feature extraction in deep learning architecture. In 2016, International conference on digital image computing: techniques and applications (DICTA) (pp. 1-8). IEEE.

[13] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., ... & Walsh, J. (2019, April). Deep learning vs. traditional computer vision. In Science and Information Conference (pp. 128-144). Springer, Cham.

[14] Hossain, M. A., & Sajib, M. S. A. (2019). Classification of the image using a convolutional neural network (CNN). Global Journal of Computer Science and Technology.

[15] Different Kinds of Convolutional Filters. https://www.saama.com/different-kinds-convolutional-filters/.

[16] Digital Filters. https://homepages.inf.ed.ac.uk/rbf/HIPR2/filtops.htm.

[17] Krishna, S. T., & Kalluri, H. K. (2019). Deep learning and transfer learning approaches for image classification. International Journal of Recent Technology and Engineering (IJRTE), 7(5S4), 427-432.

[18] Sun, X., Li, Y., Kang, H., & Shen, Y. (2019, March). Automatic Document Classification Using Convolutional Neural Network. In Journal of Physics: Conference Series (Vol. 1176, No. 3, p. 032029). IOP Publishing.

[19] Brownlee, J. (2021). Your First Deep Learning Project in Python with Keras Step-By-Step, from https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/.

[20] Brownlee, J. (2021). Image Augmentation for Deep Learning With Keras, from https://machinelearningmastery.com/image-augmentation-deep-learning-keras/.

[21] Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into Imaging, 9(4), 611-629.

[22] Tammina, S. (2019). Transfer learning using VGG-16 with a deep convolutional neural network for classifying images. International Journal of Scientific and Research Publications, 9(10), 143-150.

[23] Burugupalli, M. (2020). Image Classification Using Transfer Learning and Convolution Neural Networks.

[24] Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. International Journal of Data Mining & Knowledge Management Process, 5(2), 1.

[25] Soydaner, D. (2020). A comparison of optimization algorithms for deep learning. arXiv preprint arXiv:2007.14166.

[26] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[27] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.