International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No. 5 (Sep-2020)

http://dx.doi.org/10.12785/ijcds/090502

Towards Author Recognition of Ancient Arabic Manuscripts Using Deep Learning: A Transfer Learning Approach

Manal M. Khayyat[1,2] and Lamiaa A. Elrefaei[1,3]

[1] Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia
[2]Computer Science Department, Umm Al-Qura University, Makkah, Saudi Arabia
[3]Electrical Engineering Department, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt

Received 25 Feb. 2020, Revised 11 May 2020, Accepted 2 Jul. 2020, Published 1 Sep. 2020

Abstract: Due to the significance of ancient Arabic manuscripts and their role in enriching valuable historical information, this study aims to collect Arabic manuscripts in a dataset and classify its images to predict their authors. We accomplished this study through two main phases. First is the data collection phase. Arabic manuscripts gathered, including 52 Arabic Authors. Second is the models' development phase to extract the visual features from the images and train the networks on them. We built four deep learning models named: MobileNetV1, DenseNet201, ResNet50, and VGG19. We configured the models by tuning their learning hyperparameters toward optimizing their recognition process. Afterward, we performed a comparative analysis between all the models to measure their performance. Eventually, we reached that minimizing the learning rate, combining "Sigmoid" with "Softmax", and increasing the number of neurons on the final classification dense layer improved the networks' recognition performance significantly since all utilized deep learning models reached above 95% validation accuracy.

Keywords: Ancient Arabic Manuscripts, Authors Recognition, Convolutional Neural Networks, Deep Learning Models, Learning Hyper-parameters.

1. INTRODUCTION

Until now, there has been a gap between extracting low-level features from images as captured by electronic devices and between extracting high-level semantic concepts as viewed by real humans' brains. Deep learning is a rigid technique that utilizes Convolutional Neural Networks (CNN) to mimic humans' brains in distinguishing and classifying images.

Therefore, there is a significant need to explore the ability to utilize deep learning models for images' classifications. Al-Ayyoub et al. [1] claim that new developments in the field of deep learning showed innovative solutions in natural language processing, speech recognition, and computer vision, which includes images classification and prediction.

Classical CNN consists of three main layers. First is the convolution layer, second is the pooling layer, and last is the fully-connected layer.

Rawat & Wang [2] state that the convolutional layers are playing the role of features extractor. Thus, they learn and extract the features from the input images to organize the neurons located on the convolutional layers into feature maps. On the other hand, the role of the pooling layers is to minimize the spatial resolution exited from the previously arranged feature maps to reach spatial invariance. Usually, there are several numbers of stacked convolutional and pooling layers on top of each other to extract the features and reduce the distortion in the data. Finally, the fully-connected layer that is responsible for computing the final loss function to resolve the classification problems.

Dureja & Pahwa [3] believe that the techniques to images classification started with depending on the visual features only and then developed into using the distance metric learning; until it reached using deep learning technology. They admit that the deep learning techniques that leverage the use of convolutional neural network layers to extract the images' features are currently the best techniques for classifying images successfully. Therefore, using deep CNN would improve the performance of images' classification, especially when dealing with large and complex datasets. In fact, CNN can be trained on datasets and become able to select the best distinguishing

E-mail: mkhayyat012@stu.kau.edu.sa, laelrefaei@kau.edu.sa, mmkhayat@uqu.edu.sa, lamia.alrefaai@feng.bu.edu.eg

features using either supervised learning or unsupervised learning [4].

Collected Arabic manuscripts used in this study are historical because they established a very long time ago. Some of the manuscripts created before the hijra of the prophet "Mohammed" were the Islamic calendar got started. Hence, all available ancient Arabic manuscripts are handwritten and have poor visualization quality, which made them harder to visualize and read. Sometimes the same person establishes and writes the manuscript. While in other situations, one person creates the manuscript called "author"; and another different person writes it called "writer". There might also be an "editor" that reviews the written manuscript and modifies it. However, this study considers only the "authors" of the Arabic manuscripts.

The motivation of this research is to modulate the primary learning hyperparameters that affect the deep learning models' evaluation parameters to be able to reach the best strategy that would improve the classification and recognition accuracies.

Contributions of this paper are as follows:

1. Collect ancient Arabic manuscripts in a dataset and classify its images to be able to recognize their authors
2. Experiment three hypothesis
   - 2.1. Minimizing the learning rate allows the model to learn slowly and hence, it will improve the learning process
   - 2.2. Increasing the number of final classification dense layers improve the classification accuracy
   - 2.3. Increasing the number of neurons entering the final classification layer enhance the learning performance
3. Test a range of five various values from each hypothesis on four different deep learning models named: MobileNetV1, DenseNet201, ResNet50, and VGG19.

The rest of the paper organized as follows: Section 2 discusses the literature review and previous work done on the field. Section 3 explains the collection of our ancient Arabic manuscripts. While section 4 clarifies the augmentation and preprocessing of the collected dataset, as well as, it explains the development of the various deep learning models. Section 5 highlights the conducted experiments of the developed models and modulating their learning hyperparameters toward reaching high results of authors recognition. Section 6 analyzes and compares the generated results of the four deep learning models. Also, it relatively compares between the proposed method and the existing state-of-the-art methods. Finally, section 7 concludes the paper.

2. Related Work

Much of the literature has been focusing on authors' identification and prediction. In this section, we review previous studies in the field, and we organize them as either performed using the traditional algorithms or performed using the trending deep learning algorithms.

A. Prediction Using Traditional Algorithms

Bagasi & Elrefaei [5] propose predicting the authors of historical Arabic manuscripts using visual local-based features. The dataset they used manually collected from 30 books. It contains 1670 images of historical Arabic manuscripts. The authors initially classified their dataset into 29 classes based on the authors of the manuscripts. They preprocessed their manually collected dataset by converting the colored images into a grey-scale and then resized them into (256×256) pixels. The last step in the preprocessing phase was to use the Otsu's method to binarize grey-scale images to be able to visualize their contents better.

The authors recommend extracting local visual features from the ancient Arabic manuscripts using two Content-Based Images Retrieval (CBIR) techniques, which are Speeded-up Robust Feature (SURF) and Binary Robust Invariant Scalable Key points (BRISK). Afterward, the authors employed the Hamming Distance (HD) measurement to find-out the matching images for each predicted author using the BRISK feature extraction technique. While they used the Sum of Square Differences (SSD) measurement for the SURF technique. Finally, the authors computed both precision and recall reaching that the SURF technique extracts the local visual features better than the BRISK method. That is because SURF accomplished 70% for both precision and recall, while BRISK accomplished 53% recall and 50% precision. Noting that the overall accuracy of the system is 61% using the SURF technique and 37% using the BRISK technique.

Adam et al. [6] used ancient Arabic manuscripts to discover and test a unique algorithm for manuscripts' age and author's prediction. They utilized the KERTAS dataset, which consists of more than 2000 images of high-quality scanned ancient Arabic manuscripts. To tackle the features extraction problem, the authors employed two techniques. First, is the sparse representation-based technique that uses normalization to choose the nearest sub-space of the manuscript assisted. Second, is the handwriting style-based features. The features measure the run-length, which concern with both the edge hinge and edge direction measurements.

Afterward, both the accuracy with the predefined folds and the accuracy with the random training and testing partitions calculated. Moreover, the k-nearest neighbor with k=3 estimated. KERTAS dataset utilized for the testing scenario. The authors used complete images

without any cropping because they were interested in studying both the writing and the layout styles to help them in identifying the age and author of each manuscript. However, they resized the images looking for the best size to discover manuscripts' features. Hence, they started with (12×12) pixels. Then, they increased the sizes to become (25×25), (50×50), (100×100), (200×200) and till (250×250) pixels. They concluded that with reducing the size of the images, most of the features become unclear, which minimizes the chances to find the right matching manuscript successfully. Similarly, increasing images size dramatically might cause the same un-clarity in visualizing images features. Eventually, they concluded that the most accurate size for visualizing images was (50×50) pixels.

Asi et al., [7] recommend a new algorithm for identifying the writer(s) of ancient Arabic manuscripts successfully. Their algorithm also includes determining the number of writers. They propose utilizing the Intra-Document Analysis (IDA) process in conjunction with the integrated local and global features for reaching their goal. Two datasets were used, which are: WAHD and KHATT. WAHD dataset consists of 353 manuscripts, while the KHATT dataset consists of around 1000 short manuscripts. For the preprocessing purpose, the authors cropped the background of the scanned images and then segmented their primary text.

To identify the manuscript writer successfully, the researchers recommend extracting both local and global features. Regarding the local features, which are the low-level features represented through the curves and roundness of each manuscript handwritings, they captured utilizing the "modified contour-based feature". On the other hand, the global features, which are the high-level features based on observing the uniqueness of each writer's handwriting style. That accomplished employing the "globalizing local keypoint descriptors".

After extracting the features from each page and the entire manuscript, three classifications techniques used. Then, based on the similarity measurement of the handwriting style, the query image was classified into one of three classifications techniques, which are: averaging, voting, or weighted voting. The similarity between the query manuscript and the manuscript stored in the dataset was measured using the cosine or the Chi-square distance metric.

For the evaluation purpose, the authors used the "leave-one-out cross-validation" strategy. In this strategy, the authors randomly choose one manuscript for the query process. For the testing scenario, the authors tested all the manuscripts in KHATT dataset as it is relatively small. However, they have also tested the Islamic Heritage Project (IHP) section from the WAHD dataset, which consists of 333 manuscripts written by 302 distinct writers. They eventually concluded that using the proposed algorithm would reach accurate identification of manuscript writers.

Yahia [8] recommends integrating both the Content-Based Image Retrieval (CBIR) techniques with the Latent Semantic Indexing (LSI) approach to facilitate the indexing of historical Arabic manuscripts. The used dataset was only two pre-scanned ancient Arabic manuscripts named: "Sahih Al-Bukhari" (صحيح البخاري) and "Mawaqeet Al-Haj wa Al-Umra" (مواقيت الحج والعمرة). For the preprocessing of the dataset purpose, the author did two operations, which were binarization and smoothing the images by getting rid of their noise. Binarization involves converting colored images into greyscale images and then, into binary images. On the other hand, the main goal behind the smoothing algorithm is to remove any unnecessary parts in the image.

After preprocessing the images, they segmented into words. The author constructed latent semantic indexing by computing the values of four local features as following: 1) concentric circle features, 2) angular line features, 3) rectangular region features, and 4) circular polar grid features. Moreover, the similarities among the query image and the rest of the images existed in the dataset were measured utilizing the singular value decomposition. To evaluate the performance of the proposed model, both precision and recall of the ancient Arabic manuscripts computed. For the testing purpose, the author implemented the system using "Matlab" and stored his information using Microsoft Excel. The same two pre-scanned Arabic manuscripts tested through pre-processing them, segmenting them into individual words. Using the four features sets, the author concluded that the perfect set is the circular polar grid with 78.8% recall.

Aghbari & Brook [9] introduce an approach for segmenting and classifying ancient Arabic manuscripts. The authors used a hardcopy dataset called Historical Arabic Handwritten (HAH) manuscripts for their study by scanning them to convert their images into digital copies. The scanned images then preprocessed through four steps as following: 1) binarization, 2) noise removal, 3) smoothing, and 4) thinning. These preprocessing steps improved the original poor-quality presented in the ancient Arabic manuscripts and simplified the rest of the retrieval steps.

After preprocessing the original manuscripts' images, they segmented them into words, and then each word segmented into its connected parts. The features then extracted from the connected parts by recognizing both the structural and statistical features. In addition, the feedforward technique of multi-language processing neural network used to classify the feature vectors. For the

testing scenario, the authors used one historical Arabic manuscript named "كشف اللثام عن وجه الإسلام". There are 27 pages in the testing manuscript. After preprocessing the manuscript, segmenting it, and extracting its features using the neural network. The average accomplished accuracy computed as 89.3%.

B. Prediction Using Deep Learning Algorithms

Bagnall [10] designed a multi-headed Recurrent Neural Network (RNN), which is a particular type of deep neural networks that execute sequential elements identically. In contrast, the generated output is depending on the preceding execution. RNN has been excessively used in Natural Language Processing (NLP) applications because they inspect sequential information [11].

RNN differs from CNN in that there is no connection between the nodes of the layers. However, the layers are fully connected. Hence, we can imagine that each layer is presenting the network computations at a specific time slot. RNN employs a backpropagation algorithm in their training, which made them rigid but more challenging to train since the back-propagated elements might get smaller or more prominent in every step [12]. However, the author in [10] used RNN to identify authors successfully utilizing the texts as inputs to his deep learning model. His task designed for the "PAN 2015" authors identification competition, and he was able to record higher than 80% average Area Under Curve (AUC).

Similarly, Schaetti [13] participated in the "PAN 2017" competition and utilized a deep learning model that is using CNN for authors profiling. His model was confusion between deep learning and Term-Frequency-Inverse Document Frequency (TFIDF) model. The researcher experimented with the confusion model in many different languages; the Arabic language was one of them. The used CNN deep learning model leverages both "ReLU" and "Softmax" activation functions at the last two dense layers. To evaluate the author's model, he collected four tweet collections from the Twitter application. After assessing his model, he reached a final accuracy equals to 64% for Arabic language authors identification.

Moreover, Qian et al. [14] evaluated four deep learning models on two different datasets to identify authors. The used datasets are "Reuters_50_50" and "Gutenberg". While, the used deep learning models are "sentence-level GRU, article-level GRU, article-level LSTM, and article-level Siamese network". Finally, they concluded that the best performing model was the article-level GRU. That is because it recorded 69.1% accuracy on the Reuters dataset and 89.2% accuracy on the Gutenberg dataset.

He & Schomaker [15] experimented with three methods for identifying the writers of images. They are as following: baseline, linear adaptive, and deep adaptive learning methods. They utilized images, including one single handwritten word—the images taken from two freely available datasets named CVL and IAM. The researchers trained their convolutional neural network employing the "Tensorflow" deep learning library and NVIDIA GPU GTX 960. They concluded that the deep adaptive learning algorithm is the best algorithm for writers' identification. That is because it recorded 78.6% top-1 and 93.7% top-5 recognition rates using the CVL dataset. In addition, it recorded 96.5% top-1 and 86.1% top-5 recognition rates using the IAM dataset.

After reviewing previous researches in the field, we found out that even though some efforts made on classifying and recognizing the authors of the Arabic manuscripts, there still a need to do much research on implementing the deep learning technology for Arabic authors' prediction. That is because the deep learning technology "in particular" has been recording the highest evaluation metrics in many various domains. Thus, we focus in this paper on experimenting with multiple deep learning models for Arabic authors classifications and recognitions and tuning their learning hyperparameters looking for the best strategy that generates the highest evaluation parameters.

3. DATASET COLLECTION

Due to the lack of an existing and freely available historical Arabic manuscripts, and to be able to conduct our research study on recognizing authors of the Arabic manuscripts, we had to collect the dataset illustrated in Table 1 manually. Thereby, we started by arbitrary collecting the required ancient Arabic manuscripts from the "wqf"[1] online website. We gathered (8638) images included within (64) ancient Arabic manuscripts. A total of (52) Arabic authors has written the collected Arabic manuscripts because one author may write more than one manuscript. Hence, we assigned an ID for each one of the authors in our dataset.

Table 1 lists each author unique identification number along with its Arabic name and its translation into the English language. It also contains each manuscript details, including its identification number, Arabic title, the time the manuscript written at, the genera of the manuscript that indicates its specialized type, and the exact number of pages inside the manuscript.

[1] http://wqf.me/?p=15619

TABLE 1. LIST OF AUTHORS AND THEIR ASSOCIATED MANUSCRIPTS DETAILS.

Author ID	Arabic Name	English Name	Manuscript(s) Details				
			ID	Title	Period in Hijri	Genre	No. of Images
1	أبو الضياء عبد الرحمن بن علي بن محمد بن عمر بن الربيع الشيباني الشافعي	Abu Theyaa Abdulrahman Bin Ali Bin Mohammed Bin Omar Bin Rabea Alshebani Alshafeai	1	تيسير الوصول إلى جامع الأصول	1004	حديث	191
2	المناوي	Al-Manawe	2	شرح الجامع الصغير	*0	حديث	42
3	الشيخ حسام الدين الشهير بالمتقي الهندي	Alshaikah Hosam Aldin	3	منتخب كنز العمال (نسخة أولى)	*0	حديث	293
			17	منتخب كنز العمال (نسخة ثانية)	*0	حديث	292
4	أبو جعفر الطحاوي المصري	Abu Jafar Altahawe Almasri	4	قطعة من شرح معاني الآثار	*0	حديث	80
5	محمد الشيبي	Mohammed Alshaibi	5	الأعمال الموجبة	1135	حديث	12
6	شمس الدين محمد بن الجزري	Shams Aldin Mohammed Bin Aljazri	6	الهداية في علم الرواية	1305	علم الرواية	16
7	الإمام أحمد بن حنبل	Alimam Ahmed Bin Hanbal	7	مسند الإمام أحمد بن حنبل رواية ابنه عبد الله	*0	حديث	309
8	سيدي علي القارئ	Sidi Ali Alqari	8	مرقاة المفاتيح على مشكاة المفاتيح	1180	حديث	313
9	عبد الرحمن بن أبي بكر السيوطي	Abdulrahman Bin Abibakr Alsayoti	9	الجامع الصغير	1233	حديث	277
10	أبو زكريا محي الدين النووي	Abu Zakariya Mohe Aldin Alnawawi	10	شرح صحيح مسلم بن الحجاج	1075	حديث	162
			13	رياض الصالحين	*0	حديث	114
11	ولي الدين التبريزي	Wali Aldin Altbrizi	11	مشكاة المصابيح (نسخة أولى)	1033	حديث	260
			12	مشكاة المصابيح (نسخة ثانية)	1183	حديث	264
12	عقيل بن عمر	Aqeel Bin Omar	14	ذكر أسباب إصلاح البيوت	*0	حديث	16
13	محمد بن إسماعيل البخاري	Mohammed Bin Ismail Albukhari	15	قطعة من صحيح البخاري	1232	حديث	114
14	ابن حجر الهيتمي	Ibn Hajar Alhythami	16	شرح الأربعين، المسمى الفتح المبين	1335	حديث	101
			18	الزواجر في الكبائر	*0	حديث	16
15	محمد بن محمد الأمير	Mohammed Bin Mohammed Alamer	19	ثبت الأمير	1307	حديث	30
16	شهاب الدين أحمد ابن محمد بن محمد بن علي ابن حجر الهيتمي	Shihab Aldin Ahmed Ibn Mohammed Bin Mohammed Bin Ali Ibn Hajar Alhythami	20	مسانيد	1246	فقه	139
17	حسن الشرنبلالي	Hasan Alshernulaly	21	العقد الفريد لبيان الراجح في جواز التقليد	1384	فقه	27
			25	نظم الفوايد شرح المقاصد	1096	فقه	216
			28	تحفة التحرير	1064	فقه حنفي	7
18	محمد أفندي عابدين	Mohammed Afandi Abbdin	22	الرحيق المختوم شرح قلائد المنظوم	1305	فقه	44
19	محمد مكي بن عزوز التونسي	Mohammed Maki Bin Azoz Altonisy	23	الأجوبة المكية على الأسئلة الحفظية	*0	فقه	10
20	حكمة الهندي	Hekma Alhindi	24	خزانة الروايات	*0	فقه	49
21	إبراهيم بن محمد الحلبي	Ibrahim Bin Mohammed Alhalabi	26	ملتقى الأبحر	1064	فقه حنفي	158
22	عبد المعطي السملاوي	Abdulmoati Alsimlawy	27	المربع في حكم العقد على المذاهب الأربع	1306	فقه حنفي	6
23	حضر بن أحمد	Hathar Bin Ahmed	29	مقدمة عن الصلاة و شروطها	1284	فقه حنفي	25
24	حافظ الدين النسفي	Hafez Aldin Alnsfy	30	كنز الدقائق	*0	فقه حنفي	104
25	محب الدين الحموي	Moheb Aldin Alhamawy	31	عمدة الحكام ومرجع القضاة في الأحكام	1243	فقه حنفي	65
26	تاج الدين بن أحمد الدهان	Taj Aldin Bin Ahmed Aldahan	32	إجادة الجدة بمنع القصر في طريق جدة	1310	فقه حنفي	11
			34	رسالة في القنوت في النوازل	1171	فقه	8
27	أحمد بن محمد الحموي	Ahmed Bin Mohammed Alhamawy	33	القول البليغ في حكم التبليغ	1066	فقه حنفي	9
28	أحمد بن محمد الناطفي	Ahmed Bin Mohammed Alnatefy	35	أحكام الناطفي	1037	فقه	34
29	زين بن نجيم	Zain Bin Nejam	36	الفوائد الزينية في مذهب الحنفية	1239	فقه	50
30	أكمل الدين	Akmal Aldin	37	العناية على شرح الهداية (نسخة أولى)	1334	فقه	274
			38	العناية على شرح الهداية (نسخة ثانية)	1334	فقه	214
31	عمر بن عمر الزهري الدفري الحنفي	Omar Bin Omar Alzahri Aldafri Alhanafy	39	الدرة المنيفة على مذهب أبي حنيفة	1197	فقه	40
32	ملا خسرو	Mala Khasro	40	درر الحكام شرح غرر الأحكام	*0	فقه	82
33	بدر الدين	Badr Aldin	41	شرح التسهيل	*0	نحو	71
34	حسين بن إبراهيم الشهير بزيني زاده	Hussain Bin Ibrahim	42	الفوائد الشافية في إعراب الكافية	1209	نحو	254
35	علي النبيتيتي	Ali Alnubity	43	شرح الآجرومية	1233	نحو	179
36	الشنواني	Alshenwany	44	تعليق الدرة الشنوانية على شرح الآجرومية	1019	نحو	129
37	حسن بن أحمد زيني زاده	Hassan Bin Ahmed Zaini Zadah	45	تعليق الفواصل على إعراب العوامل	1165	نحو	82
38	ابن هشام النحوي	Ibn Hesham Alnahwi	46	شرح شذور الذهب	1233	نحو	140
39	حاج بابا ابن عثمان الطرسيوي	Haj Baba Ibn Othman Althrsiwi	47	لطائف الإعراب في شرح قواعد الاعراب	1086	نحو	119
40	أحمد بن زيني دحلان	Ahmed Bin Zaini Dahlan	48	حاشية على متن السمرقندية	1283	بلاغة	9
41	قاسم الحريري	Qasem Alhariry	49	مقامات الحريري	1064	أدب	132
42	محمد النواجي المصري	Mohammed Alnawajy Almasri	50	حلبة الكميت	*0	أدب	138
43	أحمد بن محمد الخفاجي	Ahmed Bin Mohammed Alkhafagy	51	ريحانة الألبا وزهرة الحياة الدنيا	1330	أدب	271
44	يوسف الحنفي الشافعي	Yousef Alhanafi Alshafei	52	شرح الرسالة العضدية	1168	علم وضع	6
45	محمد الخطيب الشربيني	Mohammed Alkhatib Alsherbini	53	تفسير الخطيب الشربيني	*0	تفسير	280
46	محي الدين التالجي	Mohi Aldin Altaljy	54	حاشية على شرح الكافي	1135	منطق	96
47	محمد بن عبدالرسول	Mohammed Bin Abdulrasol	55	الاشاعة لاشراط الساعة	1368	دين عام	98
48	جلال الدين السيوطي	Jalal Aldin Alsayoti	56	شرح الصدور في شرح حال الموتى في القبور	*0	دين عام	103
49	عبد الوهاب الشعراني	Abdulwahab Alshearani	57	الأشباه والنظائر الفقهية	1160	فقه	297
50	الزيلعي	Alzailai	58	تبيين الحقائق شرح كنز الدقائق (الجزء الأول)	*0	فقه	283
			59	تبيين الحقائق شرح كنز الدقائق (الجزء الثاني)	*0	فقه	381
			60	تبيين الحقائق شرح كنز الدقائق (الجزء الثالث، القسم الأول)	*0	فقه	141
			61	تبيين الحقائق شرح كنز الدقائق (الجزء الثالث، القسم الثاني)	*0	فقه	175
			62	تبيين الحقائق شرح كنز الدقائق (الجزء الرابع)	1132	فقه	373
51	عثمان ابن محمد قاري الطايفي	Othman Ibn Mohammed Qari Altaifi	63	فتح القدير	1233	فقه	170
52	عبدالله بن النسقي	Abduallah Bin Alnasqi	64	أسباب الإختيار	1053	دين عام	237
Total:			64				8638

The symbol *0 inside the "Period in Hijri" field, means that the time the manuscript was written at is unknown.

4. METHODOLOGY

We began by augmenting and preprocessing our collected ancient Arabic manuscripts to optimize their generated results. Afterward, we developed four pre-trained deep learning models. Figure 1. illustrates the architecture of the developed models.

We notice from Figure 1 that the models accept an input query image and then preprocess the image through augmenting and resizing it to prepare it for entering the deep convolutional neural networks. On the second step of the architecture, the four pre-trained deep learning models extract the visual features from the preprocessed dataset images and get trained on the extracted features. While on the last step, we solve the classification problem through transfer learning from the models.

To transfer learning from the pre-trained models while adapting them to fit with our dataset, we utilized all the layers in the chosen deep learning models with their corresponding weights. But, we deleted the last "Fully Connected" layer included in the original models and

added on the top of it three layers to improve the prediction of the authors. The first added layer is the "Flattened" layer to convert the generated two-dimensional features map into one vector. While the second and the third added dense layers are triggered through the "Sigmoid" and the "Softmax" activation functions to solve the final prediction problem.

The output from the models is the predicted (52) Arabic authors existed in our dataset.

A. Dataset Augmentation and Preprocessing

We employed offline data augmentation to enhance the prediction process since it increases the original dataset size through generating new arbitrary modified versions of the images, which assist the model in getting more generalized with the user data. The data augmentation method modifies the original images by changing their brightness, colors, noising, rotation, zooming, twisting, stretching, cropping, and flipping.

The data augmentation method could be implemented offline to generate the new images before training and have the new samples existed on the hard disk, or it could be implemented online, so the new samples will not exist on the hard drive. Instead, the new augmented samples will be generated and used during the training. The main difference between the two types of augmentation methods is that the online real-time augmentation saves more space on the users' hard disk. The data augmentation method works well with visual-based images (spatial-based) such as images including faces, animals, clothing, flowers, etc. while we should be cautious about implementing it on text-based images because we don't want to add just a random scribble into the text-images that might make them lose their distinguishable features.

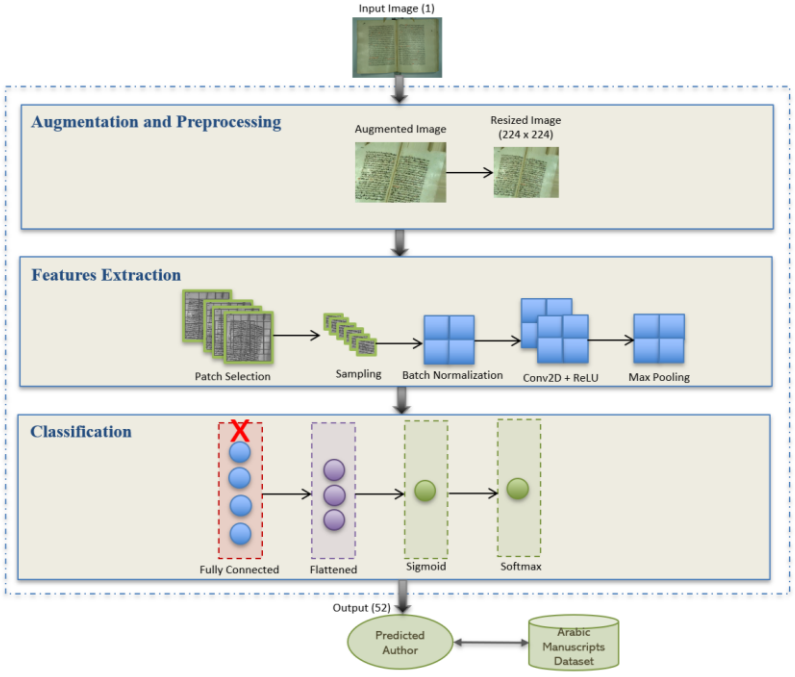

Figure 1. Architecture of the developed models.

Thereby, we performed wisely offline data augmentation on our ancient Arabic manuscripts utilizing the "ImageDataGenerator" function under "Keras" deep learning library. Five different modifications to the images' angles implemented as follows:

- Rotate the images up-to 30 degrees from the center
- Zoom up-to 10% more inside images
- Increase both the width and height by 10%
- Twist/shear images by pulling them from the top toward the right or left up-to 20%
- Fill the corner of images through repeating closest values to each pixel

Figure 2.a. illustrates original manuscripts' images. While Figure 2.b. illustrates the same images after they have augmented (zoomed, rotated, and shifted from both dimensions).

Figure 2.a. Original images.

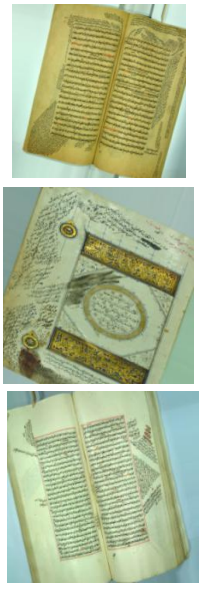

Figure 2.b. Augmented images.

After augmenting our dataset, we resized all the images into (224 x 224) pixels because it is the accepted size by the four chosen deep learning models.

B. Models Development

There are many open-source deep learning packages that researchers can use to develop their models. Including Theano, Caffe, Torch, PyTorch, MLC++, OpenCV, OpenNN, Scikit, Accord, cuDNN, BigDL, Chainer, Deeplearning4j, Dlib, Keras, Microsoft Cognitive Toolkit (CNTK), Apache MXNet, Apache SINGA, PlaidML, and Tensorflow. The first library, called MLC++, which released in 1994. While the most recent deep learning library is Tensorflow that released in 2016 [16]. We leveraged four pre-trained deep learning models that all fall under "Tensorflow"[2] deep learning library to predict the authors of our ancient Arabic manuscripts. All utilized models initially trained to classify images from the "ImageNet Large Scale Visual Recognition Challenge" that conducted in the year of 2012 and abbreviated as (ILSVRC-2012-CLS)[3].

Tensorflow deep learning package is distinguished from other deep learning packages in that it supports distributed execution from multiple devices and on different platforms, which makes it more flexible [17]. The developed deep learning models were four as follows:

1) MobileNet_V1_100_244

MobileNetV1 deep learning model is the simplest model we used in our experiments. That is because it consists of a small number of layers contained within plain blocks and stacked on the top of each other without any residual connections between them. Instead, the convolutional layers connected linearly, and the signals move only in forward propagation. Moreover, MobileNetV1 model decreases the spatial dimensions among its tensors, which makes it small compared with other large deep learning models. This characteristic enables it to execute faster and in a short time. Concerning the number of multi-adds, MobileNetV1 includes 569 million of them that authorize the model to realize and comprehend the learned features efficiently [18].

2) ResNet_V2_50

ResNet50 is a deep residual convolutional neural network. In other words, it includes residual connections and multiple branches between its 50 convolutional layers, which makes it a non-linear model. Hence, its signals can move in a backpropagation or forward propagation manner, making skip connections as needed. The second version of ResNet50 model uses batch normalization as a pre-activation function before calculation the weights to improve the training on the extracted features [19]. The model includes over 25 million of learning parameters that makes it efficient in the learning process [20].

3) DenseNet_201

DenseNet201 deep learning model is an extensive model since it includes (201) convolutional layers. Each layer in the DenseNet201 model is passing its features to all incoming next layers while collecting previous knowledge from all preceding layers [21]. This increases the number of channels moving forward in the model. However, every two contiguous blocks in the model are separated by one convolutional layer and one average pooling layer to decrease the model's complexity. DenseNet201 model is similar to ResNet50 in that it also uses the batch normalization before the weights' computation function.

[2] https://tfhub.dev/

[3] http://www.image-net.org/challenges/LSVRC/2012/

4) VGG_19

According to Tsang [22], the performance of VGG19 deep learning model outperformed all other models since it won the ILSVRC-2014 competition for classifying images. In addition, VGG19 generated the highest evaluation parameters on both Caltech and VOC datasets. Thus, it is a rigid deep learning model even though it includes the least number of convolutional layers comparing it with the other three experimented deep learning models. On contrast, VGG_19 model is having the largest number of learning parameters, among other utilized models. It includes 144 million parameters [23]. This means that increasing the number of layers without efficient use of the other learning parameters will not improve the learning process.

Figure 3. Illustrates the layers' structure of the leveraged convolutional neural networks. (a) MobileNet-V1 model [18], (b) ResNet-50 model [24], (c) DenseNet-201 model [21], and (d) VGG-19 model [24].

The figure highlights the output size written in orange color to the right side of each layer. If the same layer repeated then, we indicated this by the dashed blue square with the number of repetitions written in blue above the output size. The straight arrows are for the plain blocks with forwarding propagated signals, while the slanted arrows are referencing the residual blocks with both forward and backward propagated signals.

Even though the residual connections in ResNet50 model exist after each 3-layers block, for simplicity, we drew them between main blocks. The (1x1/ 3x3/ 7x7) written before each layer indicates the size of the kernel. On the other hand, the number of filters highlighted inside the layers' boxes after the "-" symbol. The number written with the Fully Connected (FC) layer indicating the size of the feature that produced from previous training and features extraction steps and entering the layer.

5. Experiments and Tests Results

In this section of the study, we explain in detail the hardware and software used to conduct our experiments. As well as. We define the mathematical representations of the employed evaluation parameters to assess the four developed deep learning models. Moreover, we clarify the tested hypotheses to tune the hyperparameters essential in the model's learning process.

A. Hardware and Software Used

We developed our models on "ABS Battelbox" personal computer that is having Ubuntu 16.04 operating system and Nvidia Gefore RTX 2080 GPU. Regarding the programming language, we used Python version 3.7 on Pycharm application programming interface.

B. Evaluation Parameters

After developing the models, we trained them utilizing the manually collected ancient Arabic manuscripts dataset.

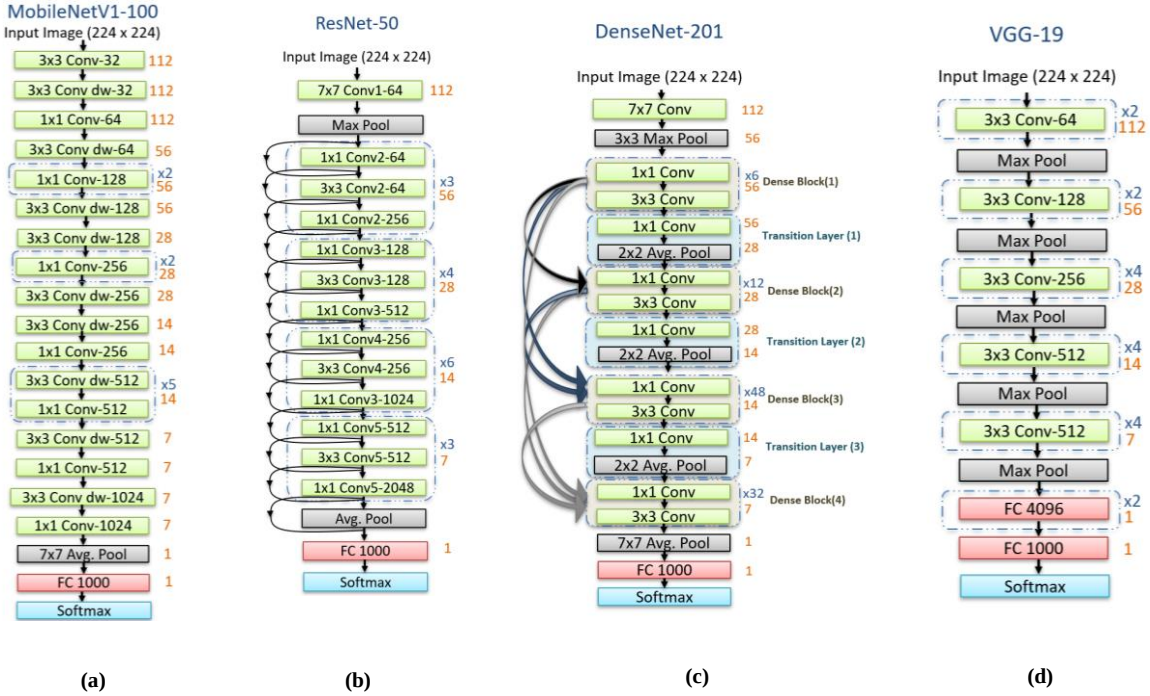

Figure 3. Layers' structure of the Convolutional Neural Networks (CNN).

We divided the utilized dataset into three categories as following: train, test, and validate. The ratios used in splitting the dataset are as follows: 70% from the entire size of the dataset allocated for the training subset, 15% for the validation subset, and 15% for the testing subset. To evaluate the models, we recorded the generated accuracy by each model. The equation for calculating the accuracy evaluation metric presented in (1) [9]:

$$\text{Accuracy} = \frac{S_{cw}}{T_w} \quad (1)$$

Where S_{cw}, represents the number of successfully predicted authors and T_w, represents the total number of authors.

Moreover, we evaluated the effectiveness of the developed deep learning models by computing the precision (P), recall (R), and the F-score (F-score) of each correctly retrieved author in our dataset. Following equations (2-4) illustrate their computations [25]:

$$P = \frac{\text{number of correctly retrieved authors}}{\text{total number of retrieved authors}} \quad (2)$$

$$R = \frac{\text{number of correctly retrieved authors}}{\text{total number of relevant authors in the dataset}} \quad (3)$$

$$\text{F-score} = 2 * (P * R)| (P + R) \quad (4)$$

We depended on both the validation accuracy and the average F-score metrics in evaluating the performance of our conducted experiments. That is because there is a trade-off between the recall and the precision. However, the F-score metric combines the measurements of both the recall and the precision [26]. Thus, we can rely on it as a trustable general evaluation parameter for evaluating the developed deep learning models.

C. Modulating The Learning Hyperparameters

We started our experiments by executing the four deep learning models ten times (10 epochs) and utilizing (1e-3) learning rate. Furthermore, we used one final classification dense layer that includes "Softmax" activation function with "adam" optimizer and "spare categorical crossentropy" loss.

We used a global shuffling buffer while building the "TF" record, which is a zipped simplified version of our collected dataset. As well as, we performed another local shuffle of (64) buffers while doing the training on our data. We trained the models on the same dataset and used the same batch size as (32).

Table 2 summarizes the results from the initial execution of the four deep learning models. We recorded five evaluation metrics as following: Training Accuracy (TA), Validation Accuracy (VA), Average Precision (AP), Average Recall (AR), and Average F-score (AF).

The highest generated results highlighted in bold. Considering that, the high recorded numbers of the evaluation metrics indicate better models' performance. On the other hand, the worst generated results were highlighted by red color to indicate the low performance of the models.

TABLE 2. INITIAL EXECUTION RESULTS OF THE FOUR MODELS.

	MobileNetV1	ResNet50	DenseNet201	VGG19
TA	0.3777	0.4075	0.2936	**0.9152**
VA	0.3542	0.4006	0.2957	**0.8737**
AP	0.2167	0.2402	0.1589	**0.8371**
AR	0.3716	0.3932	0.2972	**0.8533**
AF	0.2516	0.2775	0.1896	**0.8362**

From Table 2, we notice that all the models were not able to perform well in recognizing the authors of our ancient Arabic manuscripts except VGG19 deep learning model. That is because all the models realized only around 20% of the authors except VGG19, which recognized approximately 80% of the authors. We also notice from table 2 that the DenseNet201 deep learning model was the weakest in identifying the Arabic authors since it generated the lowest recognition results among the other tested deep learning models. Therefore, we set a goal to modulate and tune the primary hyperparameters essential in the learning process of the deep learning models to reach the best strategy for recognizing the Arabic authors of our ancient manuscripts. Hence, we experimented with three hypotheses to reach the best evaluation metrics.

Hypothesis (1): Minimizing the learning rate allows the model to learn slowly, and hence it will improve the learning process.

The learning rate is the step size in seeking images within the dataset to get trained on them. Therefore, it shouldn't be too small, either too large to enable the deep learning model to learn effectively with a suitable speed in memory [27].

To test the correctness of the hypothesis, we conducted new experiments that employ different learning rates ranging from 1e-2 (0. 01) to 1e-6 (0.000001). Generated results summarized in the tables from Table 3 to Table 6.

Analysis and findings from the tables' 3-6 results:

1. Even though there is a little bit fluctuation in the results, three deep convolutional neural networks (MobileNetV1, DenseNet201, and VGG19) recorded the highest evaluation parameters at (1e-6) learning rate. The average F-score recorded by MobileNetV1 deep learning model was 0.2965, and the average F-score recorded by DenseNet201 deep learning model was 0.2408, as well as, the average F-score recorded by VGG19 deep learning model was 0.9217. Hence, we can claim that the hypothesis holds true, and we will use (1e-6) for the rest of the experiments.

TABLE 3. MOBILENETV1 WITHIN DIFFERENT LEARNING RATES

	1e-2	1e-3	1e-4	1e-5	1e-6
TA	0.3777	0.3848	0.3970	0.3733	**0.4279**
VA	0.3542	0.3678	0.3710	0.3750	**0.4135**
AP	0.2167	0.2199	0.2212	0.2299	**0.2653**
AR	0.3716	0.3790	0.3861	0.3779	**0.4047**
AF	0.2516	0.2490	0.2685	0.2688	**0.2965**

TABLE 4. RESNET50 WITHIN DIFFERENT LEARNING RATES

	1e-2	1e-3	1e-4	1e-5	1e-6
TA	0.3599	0.4075	0.4323	**0.4596**	0.4512
VA	0.3349	0.4006	0.4087	**0.4415**	0.4295
AP	0.1859	0.2402	0.2526	**0.2884**	0.2583
AR	0.3545	0.3932	0.4162	**0.4598**	0.4431
AF	0.2247	0.2775	0.2973	**0.3398**	0.3110

TABLE 5. DENSENET201 WITHIN DIFFERENT LEARNING RATES

	1e-2	1e-3	1e-4	1e-5	1e-6
TA	0.3080	0.2936	0.3331	**0.3618**	0.3606
VA	0.3165	0.2957	0.3357	0.3253	**0.3438**
AP	0.1609	0.1589	0.1890	0.1914	**0.2072**
AR	0.3219	0.2972	0.3346	0.3366	**0.3506**
AF	0.1888	0.1896	0.2230	0.2289	**0.2408**

TABLE 6. VGG19 WITHIN DIFFERENT LEARNING RATES

	1e-2	1e-3	1e-4	1e-5	1e-6
TA	0.9088	0.9152	0.9564	0.9250	**0.9799**
VA	0.8478	0.8737	0.8622	0.8686	**0.9431**
AP	0.8331	0.8371	0.8936	0.8577	**0.9184**
AR	0.8631	0.8533	0.8559	0.8801	**0.9304**
AF	0.8418	0.8362	0.8588	0.8638	**0.9217**

2. All the four deep learning models recorded the lowest evaluation parameters at (1e-2) and (1e-3) learning rates. That is because the lowest average F-scores were 0.2490 and 0.8362 recorded at (1e-3) by MobileNetV1 and VGG19 deep learning models, respectively. While the lowest average F-scores were 0.2247 and 0.1888 recorded at (1e-2) by ResNet50 and DenseNet201 deep learning models, respectively. Thus, we shouldn't use fast learning rates for training our deep learning models.

To easily visualize the improvements in the learning process, we drew the F-score values of the four models at the different learning rates in Figure 4.

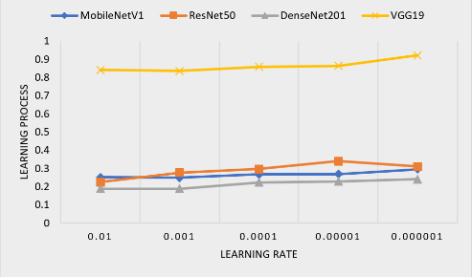

Figure 4. Learning process through different learning rates.

In general, there were no considerable improvements in the results after minimizing the learning rate. Therefore, we had to tune another learning hyperparameter that is crucial to the models' operation. Hence, we made all the models deeper through increasing their layers in the next hypothesis.

Hypothesis (2): Increasing the number of final classification dense layers improve the classification accuracy.

To test the correctness of this hypothesis, we added more classification dense layers before the formerly existing "Softmax" layer, denoted as (F). The cases we tested are as following:

- Add "ReLU" dense layer, denoted as (R)
- Add "Sigmoid" dense layer, denoted as (G)
- Add both "ReLU" and "Sigmoid" dense layers
- Add two "ReLUs" and one "Sigmoid" dense layer

We set the number of neurons in all added new classification dense layers to (256). Table 7 to Table 10 summarizes the generated results.

Analysis and findings from the tables' 7-10 results:

1. Making the convolutional neural networks deeper through adding two "ReLUs" and one "Sigmoid" classification dense layers didn't record the highest results in any one of the four tested deep learning models. That is because the recorded average F-scores were 0.9454, 0.9599, 0.9300, and 0.9492 by MobileNetV1, ResNet50, DenseNet201, and VGG19 dep learning models respectively, which were not the highest recorded values. Thus, we can't claim that this hypothesis holds true.
2. Both MobileNetV1 and DenseNet201 recorded their highest results when we added one "ReLU" and one "Sigmoid" classification dense layers before the existing "Softmax" classification layer. MobileNetV1 model recorded 0.9578, and the DenseNet201 model recorded 0.9685 average F-scores, which were the highest recorded F-scores by both models during the entire experiments. On the other hand, both ResNet50 and VGG19 deep learning models recorded their most top results when we added the "Sigmoid" classification dense layer before the existing "Softmax" layer. ResNet50 model recorded 0.9655, and the VGG19 model recorded 0.9647 average F-scores, which were the highest recorded F-scores by both models during the entire experiments. Thus, we recommend adding the "Sigmoid" activation function either alone or with the "ReLU" activation function before the original

TABLE 7. MOBILENETV1 THROUGH DIFFERENT CLASSIFICATION LAYERS

	F	F + R	F + G	F + R + G	F + 2R + G
TA	0.4279	0.2809	0.9894	**0.9998**	0.9967
VA	0.4135	0.2804	0.9511	**0.9631**	0.9495
AP	0.2653	0.1521	**0.9613**	0.9611	0.9526
AR	0.4047	0.2859	0.9576	**0.9581**	0.9486
AF	0.2965	0.1804	0.9555	**0.9578**	0.9454

TABLE 8. RESNET50 THROUGH DIFFERENT CLASSIFICATION LAYERS

	F	F + R	F + G	F + R + G	F + 2R + G
TA	0.4512	0.3060	0.9955	**1.0000**	**1.0000**
VA	0.4295	0.3197	0.9583	**0.9631**	0.9567
AP	0.2583	0.2019	**0.9670**	0.9578	0.9607
AR	0.4431	0.3249	**0.9665**	0.9576	0.9600
AF	0.3110	0.2266	**0.9655**	0.9569	0.9599

TABLE 9. DENSENET201 THROUGH DIFFERENT CLASSIFICATION LAYERS

	F	F + R	F + G	F + R + G	F + 2R + G
TA	0.3606	0.0189	0.9827	**1.0000**	0.9895
VA	0.3438	0.0176	0.9399	**0.9599**	0.9327
AP	0.2072	0.0004	0.9567	**0.9634**	0.9340
AR	0.3506	0.0192	0.9555	**0.9629**	0.9308
AF	0.2408	0.0009	0.9539	**0.9685**	0.9300

TABLE 10. VGG19 THROUGH DIFFERENT CLASSIFICATION LAYERS

	F	F + R	F + G	F + R + G	F + 2R + G
TA	0.9799	0.9974	**0.9997**	0.9981	0.9957
VA	0.9431	0.9471	**0.9583**	0.9551	0.9503
AP	0.9184	0.9459	**0.9659**	0.9543	0.9497
AR	0.9304	0.9441	**0.9657**	0.9548	0.9504
AF	0.9217	0.9439	**0.9647**	0.9539	0.9492

"Softmax" dense layer since it had the highest effects on the results.

3. Three deep learning models recorded the lowest results when we used the "ReLU" classification dense layer before the existing "Softmax", which were MobileNetV1, ResNet50, and DenseNet201. In fact, DenseNet201 deep learning model decreased its evaluation parameters dramatically since it recorded 0.0009 average F-score when using the "ReLU" classification dense layer before the existing "Softmax". In addition, MobileNetV1 recorded 0.1804, and ResNet50 recorded 0.2266 average F-scores, which were the lowest recorded F-scores by both models. Thereby, we should never combine between "ReLU" and "Softmax" alone.

To highlight the improvements in the learning process, we drew the F-score values of the four models utilizing the different number of final classification dense layers in Figure 5.

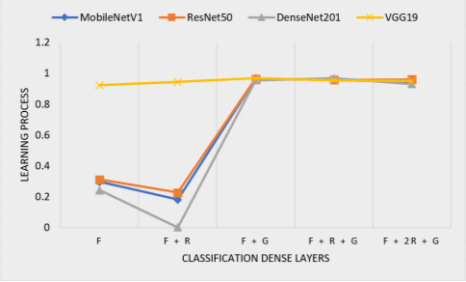

Figure 5. Learning process through a different number of dense layers

Since all the four utilized deep learning models reached higher than 90% successful recognition of our Arabic authors after adding the "Sigmoid" classification dense layer, we accomplished satisfying results. The final recorded validation accuracy of the MobileNetV1 deep learning model raised from 41.35% to 95.11%.

Similarly, the validation accuracy of both ResNet50 and VGG19 deep learning models increased from 42.95% and from 94.31%, respectively, to become 95.83%. Moreover, the validation accuracy of the DenseNet201 deep learning model risen from 34.38% to 93.99%. Thus, we will use "Sigmoid" in addition to the existing "Softmax" classification dense layer for the rest of the experiments. But, we want to conduct one more examination that tests the effects of increasing the neurons number on the added classification dense layer.

Hypothesis (3): Increasing the number of neurons on the last classification layer enhances the learning performance.

To test the correctness of this hypothesis, we increased the number of neurons in the new added "Sigmoid" classification dense layer from (64) to (1024) neurons. Generated results presented from Table 11 to Table 14.

Analysis and findings from the tables' 11-14 results:

1. All the four tested deep learning models slightly raised their evaluation parameters by increasing the number of neurons from (64) neurons to become (1024) neurons. The MobileNetV1 deep learning model raised its recorded F-score from 0.9356 to 0.9566.

   Similarly, The ResNet50 deep learning model increased its recorded F-score from 0.9457 to 0.9646. DenseNet201 deep learning model raised its recorded F-score from 0.9083 to 0.9606, as well as, VGG19 deep learning model raised its recorded F-score from 0.9524 to 0.9649. These slight improvements in the results allow us to admit that the hypothesis holds true.
2. All the four tested deep learning models recorded the lowest evaluation parameters using the (64) neurons number. That is because the recorded final validation accuracies were 0.9239, 0.9463, 0.9022, and 0.9367 by MobileNetV1, ResNet50, DenseNet201, and VGG19 deep learning models respectively.

TABLE 11. MOBILENETV1 THROUGH DIFFERENT NUMBER OF NEURONS

	64	128	256	512	1024
TA	0.9638	0.9815	0.9894	0.9906	**0.9930**
VA	0.9239	0.9447	0.9511	**0.9639**	0.9559
AP	0.9401	0.9557	0.9613	0.9559	**0.9599**
AR	0.9366	0.9553	0.9576	0.9552	**0.9584**
AF	0.9356	0.9549	0.9555	0.9545	**0.9566**

TABLE 12. RESNET50 THROUGH DIFFERENT NUMBER OF NEURONS

	64	128	256	512	1024
TA	0.9816	0.9866	0.9955	0.9955	**0.9971**
VA	0.9463	0.9471	0.9583	0.9511	**0.9623**
AP	0.9477	0.9518	0.9670	0.9543	**0.9698**
AR	0.9459	0.9516	**0.9665**	0.9538	0.9663
AF	0.9457	0.9508	**0.9655**	0.9494	0.9646

TABLE 13. DENSENET201 THROUGH DIFFERENT NUMBER OF NEURONS

	64	128	256	512	1024
TA	0.9294	0.9766	0.9827	0.9885	**0.9899**
VA	0.9022	0.9431	0.9399	0.9511	**0.9583**
AP	0.9116	0.9398	0.9567	0.9490	**0.9637**
AR	0.9115	0.9345	0.9555	0.9487	**0.9637**
AF	0.9083	0.9347	0.9539	0.9477	**0.9606**

TABLE 14. VGG19 THROUGH DIFFERENT NUMBER OF NEURONS

	64	128	256	512	1024
TA	0.9943	0.9995	0.9997	**1.0000**	0.9995
VA	0.9367	0.9583	0.9583	**0.9663**	0.9591
AP	0.9532	0.9652	0.9659	0.9609	**0.9710**
AR	0.9528	0.9649	**0.9657**	0.9613	0.9652
AF	0.9524	0.9647	0.9647	0.9605	**0.9649**

Hence, we recommend not to use this low number of neurons on the last classification dense layer.

To simplify the visualization of reached results, we summarized the generated F-score of the four models during the used different number of neurons in Figure 6.

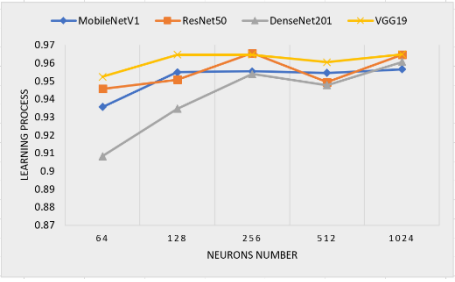

Figure 6. Learning process through various neurons number.

From Figure 6. we notice that there is a little bit fluctuation in the final recorded F-score. However, all the models improved their performance after increasing the neurons number to (1024) neurons on the final classification "Sigmoid" dense layer. Therefore, we reached a successful recognition of the Arabic authors.

After experimenting the effects of various learning hyperparameters on the performance of the four deep neural networks, we conclude that the best strategy to follow on developing our deep learning models utilizing our ancient Arabic manuscripts is to use "Sigmoid" classification dense layer before the exiting "Softmax" layer as they produced high results. In addition, we will use (1024) neurons in the added classification dense layer, and we will employ (1e-6) as the learning rate since it allowed the models to learn the extracted features more slowly, and that makes them more knowledgeable. Furthermore, we found out that running the learning cycles (10) epochs saved our time and accomplished satisfying results.

From the conducted experiments, we noticed that initially, the accuracy was low in all the models except the VGG19 deep leering model. There was a massive difference in the results between the VGG19 deep learning model and the other three deep learning models. However, after the wise and careful tuning of the main learning hyperparameters, we were able to increase the evaluation parameters for all the models, which optimized their performance and generated accuracies that were all above 95% successful recognition of the Arabic authors.

6. COMPARATIVE ANALYSIS

This section begins by comparing the generated results of the four developed and tested deep learning models. Afterward, we relatively compare our proposed method with other existing, state-of-the-art techniques.

A. Comparison Between The Results of The Four Developed Deep Learning Models

After reaching the best strategy in developing our deep learning models in the previous section, we compare the four models through computing their precision, recall, and F-score for each author, as illustrated in Table 15.

This comparison conducted to ensure that we reached our goal, which is confirming that all the tested four deep learning models are performing well in recognizing the Arabic authors.

To fairly compare between the models, we ensured that all utilized models are accepting the same input image size. In addition, we leveraged the same learning hyperparameters for all used models in one base script that contains a simple convolutional neural network for configuring the initial file structure.

From Table 15, we notice that the deep learning models were able to 100% successfully recognize a close number of authors out of the existing (52) Arabic authors in our dataset. For instance, the MobileNet model recognized (21) authors. ResNet50 model recognized (23) authors, DenseNet201 model recognized (26) authors, and VGG19 model recognized (24) authors. In addition, we notice that none of the authors were completely un-recognized, which

TABLE 15. COMPARATIVE ANALYSIS OF THE PERFORMANCE OF THE FOUR DEEP LEARNING MODELS.

Author ID	MobileNet_100			ResNet_50			DenseNet_201			VGG_19		
	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score
1	0.6333	1.0000	0.7755	0.6207	0.9000	0.7346	0.5714	1.0000	0.7273	0.7308	1.0000	0.8444
2	0.8519	1.0000	0.9200	0.8800	1.0000	0.9362	0.8889	1.0000	0.9412	0.8519	1.0000	0.9200
3	0.9394	0.9688	0.9538	0.9375	1.0000	0.9677	0.9143	0.9412	0.9275	0.8709	0.9310	0.9000
4	0.7273	0.6400	0.6809	0.7097	1.0000	0.8302	0.7241	0.9130	0.8077	0.7353	1.0000	0.8475
5	**1.0000**	**1.0000**	**1.0000**	0.9629	1.0000	0.9811	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
6	0.9714	1.0000	0.9855	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
7	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
8	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
9	0.9545	0.9545	0.6545	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
10	1.0000	0.9677	0.9836	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9677	0.9836	0.9032	0.9655	0.9333
11	0.9583	1.0000	0.9787	**1.0000**	**1.0000**	**1.0000**	0.9583	1.0000	0.9787	0.9615	1.0000	0.9804
12	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
13	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9500	0.9744
14	0.9655	0.9655	0.9655	1.0000	0.9643	0.9818	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
15	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9524	1.0000	0.9756	0.9545	1.0000	0.9767
16	0.9474	1.0000	0.9729	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
17	0.9583	0.9583	0.9583	**1.0000**	**1.0000**	**1.0000**	0.9565	1.0000	0.9778	**1.0000**	**1.0000**	**1.0000**
18	1.0000	0.9524	0.9756	**1.0000**	**1.0000**	**1.0000**	0.9545	1.0000	0.9767	0.9545	0.9545	0.9545
19	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
20	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
21	0.9615	1.0000	0.9804	0.9600	1.0000	0.9796	0.9231	1.0000	0.9600	**1.0000**	**1.0000**	**1.0000**
22	1.0000	0.6207	0.7659	0.8636	0.6333	0.7308	1.0000	0.6000	0.7500	0.9130	0.7500	0.8236
23	**1.0000**	**1.0000**	**1.0000**	0.9130	1.0000	0.9545	0.9583	1.0000	0.9787	**1.0000**	**1.0000**	**1.0000**
24	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9688	0.9841	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9667	0.9831
25	0.9615	1.0000	0.9804	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9615	1.0000	0.9804
26	0.9524	1.0000	0.9756	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9545	0.9767
27	1.0000	0.8636	0.9268	0.9545	0.8750	0.9130	0.9524	0.8696	0.9091	0.9524	0.8696	0.9091
28	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
29	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9655	0.9825	1.0000	0.9667	0.9831	1.0000	0.9667	0.9831
30	0.9655	0.9655	0.9655	0.9655	1.0000	0.9825	1.0000	0.9667	0.9831	**1.0000**	**1.0000**	**1.0000**
31	**1.0000**	**1.0000**	**1.0000**	0.9474	1.0000	0.9729	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9444	0.9714
32	0.9524	1.0000	0.9756	**1.0000**	**1.0000**	**1.0000**	0.8696	1.0000	0.9302	**1.0000**	**1.0000**	**1.0000**
33	0.9565	1.0000	0.9778	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
34	0.9643	0.9643	0.9643	1.0000	0.9615	0.9804	0.9615	0.9615	0.9615	1.0000	0.9643	0.9818
35	1.0000	0.8636	0.9268	1.0000	0.9583	0.9787	1.0000	0.9130	0.9545	1.0000	0.9565	0.9778
36	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9687	0.9841	0.9655	0.9333	0.9492	**1.0000**	**1.0000**	**1.0000**
37	**1.0000**	**1.0000**	**1.0000**	0.9600	1.0000	0.9796	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
38	0.9130	1.0000	0.9545	0.9524	0.9524	0.9524	**1.0000**	**1.0000**	**1.0000**	0.9524	0.9524	0.9524
39	1.0000	0.9583	0.9787	1.0000	0.9583	0.9787	1.0000	0.9167	0.9565	1.0000	0.9615	0.9804
40	0.9444	0.8500	0.8947	1.0000	0.8500	0.9189	0.8889	0.8421	0.8649	0.8947	0.8500	0.8718
41	0.9600	0.9231	0.9412	1.0000	0.9259	0.9615	1.0000	0.9231	0.9600	1.0000	0.9643	0.9818
42	1.0000	0.8000	0.8889	1.0000	0.9500	0.9744	1.0000	0.8636	0.9268	1.0000	0.9524	0.9756
43	0.9565	0.9565	0.9565	1.0000	0.9565	0.9778	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
44	0.5714	0.7059	0.6316	1.0000	0.5000	0.6667	0.7857	0.5789	0.6667	1.0000	0.4706	0.6400
45	**1.0000**	**1.0000**	**1.0000**	0.9583	1.0000	0.9787	**1.0000**	**1.0000**	**1.0000**	1.0000	0.9091	0.9524
46	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
47	**1.0000**	**1.0000**	**1.0000**	0.9500	1.0000	0.9744	**1.0000**	**1.0000**	**1.0000**	0.9091	1.0000	0.9524
48	0.9500	1.0000	0.9744	0.8947	1.0000	0.9444	0.8889	1.0000	0.9412	0.9474	1.0000	0.9729
49	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
50	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
51	1.0000	0.9565	0.9778	1.0000	0.9583	0.9787	1.0000	0.9565	0.9778	1.0000	0.9565	0.9778
52	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

validate the effectiveness of the utilized strategy in developing the models.

After reaching and validating the effectiveness of the best strategy to develop our models, we generated the confusion metrics for each model, as illustrated in the figures from Figure 7. to Figure 10. Theses metrics assist in comparing the predicted authors with the ground truth ones since the rows include the true authors, while the columns contain the predicted authors. Moreover, the sum of the total numbers inside each confusion matrix references the test portion of the total dataset size.

From the generated confusion matrices by the four deep learning models, we admit that the authors' recognition process is performing well. That is because we can notice a clear diagonal created inside the confusion matrices, including the most significant numbers, which indicates that the models were able to recognize the authors of our ancient Arabic manuscripts successfully. In addition, we notice that the author with (21) number in the confusion matrix and (22) id in our dataset, as well as, the author with (43) number in the confusion matrix and (44) id in our dataset were the worst recognized Arabic authors. That is because they had

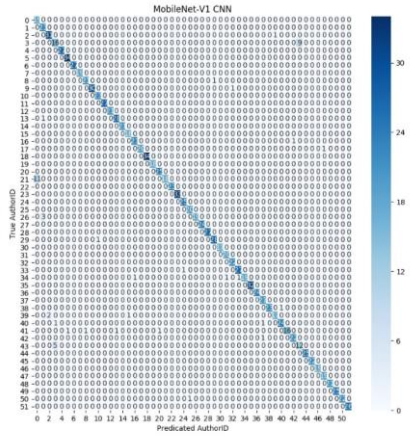

Figure 7. Authors' Confusion Matrix by MobileNetV1 Model.

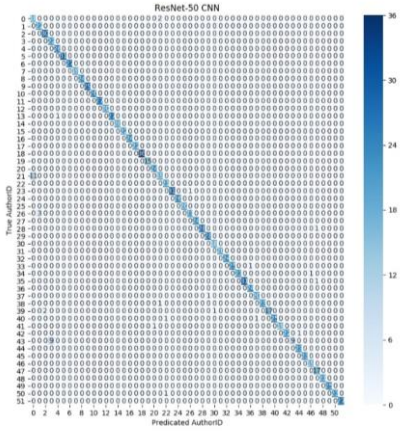

Figure 8. Authors' Confusion Matrix by ResNet50 Model.

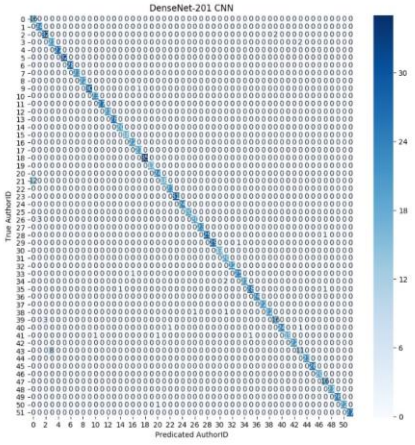

Figure 9. Authors' Confusion Matrix by DenseNet201 Model.

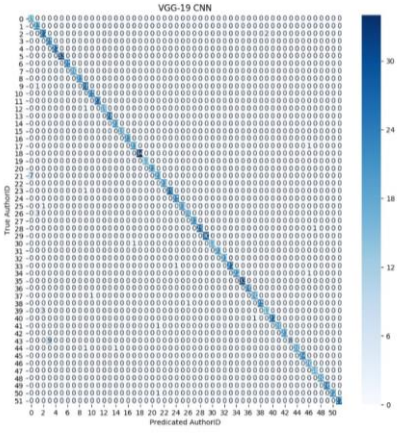

Figure 10. Authors' Confusion Matrix by VGG19 Model.

the largest numbers of miss-predicted images. But, we find this result satisfactory because even though these two authors only wrote (6) pages, the models were able to recognize them, and this was due to the performed offline data augmentation on the training subset. About the author with (22) id, 76.59%, 73.08%, 75%, and 82.36% of its images were recognized successfully using MobileNet, ResNet50. DenseNet201 and VGG19 deep learning models, respectively. Similarly, the author with (44) id; 63.16%, 66.67%, 66.67%, and 64% of its images were recognized successfully using MobileNet, ResNet50. DenseNet201 and VGG19 deep learning models, respectively.

B. Relative Comparison Between The Proposed and Existing Methods

In Table 16, we evaluate our approach with other state-of-the-art approaches. This accomplished through comparing the results of existing methods used in the related work and our proposed method, which utilizes pre-trained deep learning models using the following learning hyperparameters:

- Employ (1e-6) for the learning rate
- Add "Sigmoid" before the original existing "Softmax" classification dense layer
- Use (1024) neurons on the added "Sigmoid" final classification dense layer

TABLE 16. RELATIVE COMPARISON WITH THE STATE-OF-THE-ART METHODS.

	Reference# (Year)	Features Extraction	Classification	Dataset	No. of Images	Results
Manual Handcrafted Features	[5] (2019)	SURF and BRISK CBIR	*SF: Hamming distance and Sum of square distance	Manually collected Arabic manuscripts	1670	61% overall accuracy using SURF technique and 37% using BRISK
	[6] (2018)	Sparse representation-based technique and handwriting style-based features	*ML: K-nearest neighbor	KERTAS ancient Arabic manuscripts	2505	94.77% accuracy with predefined folds and 42.31% accuracy with random train/test split using (50×50) size
	[7] (2017)	IDA process combined with modified contour-based feature and globalizing local key point descriptors	*SF: Cosine or Chi-square distance metric	IHP and KHATT ancient Arabic manuscripts	(IHP, 2313) (KHATT, 4000)	88.9% and 73% identification accuracies using KHATT and IHP datasets respectively
	[8] (2011)	CBIR techniques with the LSI approach	*SF: Singular Value Decomposition (SVD)	"Sahih Al-Bukhari" and "Mawaqeet Al-Haj wa Al-Umra" Arabic manuscripts	34	The most accurate feature set is the circular polar grid with 78.8% recall
	[9] (2009)	Feedforward technique of multi-language processing neural network	*SF: Error signal function	Historical Arabic Handwritten manuscript	27	89.3% average accuracy
Automatic Deep Learning Features	[10] (2015)	Multi-headed Recurrent Neural Network (RNN)	*DL: Rectified Shifted Square Root (ReSQRT)	PAN 2014	—	Higher than 80% AUC
	[13] (2017)	Confusion between deep learning and (TFIDF) model	*DL: Softmax dense layer	Four tweet collections from Twitter	—	64% Arabic authors identification accuracy
	[14] (2018)	The authors tested four deep learning models named: sentence-level GRU, article-level GRU, article-level LSTM, and article-level Siamese network	*DL: Softmax dense layer	"Reuters_50_50" and "Gutenberg" datasets	(Reuters, 5000) (Gutenberg, 1286)	Article-level GRU was the best performing model recording 69.1% and 89.2% accuracy on Reuters and Gutenberg datasets respectively
	[15] (2018)	The authors tested three methods: 1) Baseline, 2) linear adaptive, and 3) deep adaptive learning	*DL: Sigmoid dense layer	CVL and IAM datasets	(CVL, 99513) (IAM, 49625)	The deep adaptive learning was the best method recording 78.6% and 69.5% top-1, as well as, 93.7% and 86.1% top-5 recognition rates using the CVL and IAM datasets respectively
	Proposed method	Transfer learning from MobileNet_V1_100_244	*DL: Sigmoid + Softmax dense layers	Collected ancient Arabic manuscripts	8638	**95.59%** validation accuracy
		Transfer learning from ResNet_V2_50				**96.23%** validation accuracy
		Transfer learning from DenseNet_201				**95.83%** validation accuracy
		Transfer learning from VGG_19				**95.91%** validation accuracy

The comparison is relative as different datasets were used, as well as, various features extraction algorithms employed, and different classification methods used. We categorized the papers according to the performed features extraction algorithm. Hence, we divided them as either manual handcrafted features or automatic deep learning-based features.

The (SF) in the classification column in Table 16 stands for Static Formula, (ML) stands for Machine Learning, and (DL) stands for Deep Learning. We notice from Table 16 that our proposed approach achieved the highest results among other state-of-the-art methods, which prove its correctness and effectiveness.

7. Conclusion

In this study, we developed and compared four pre-trained models that fall under the "Tensorflow" deep learning package to classify ancient Arabic manuscripts and successfully recognize their authors. The models were: MobileNet_V1, Resnet_50, DenseNet_201, and VGG_19.

We started by collecting the dataset manually, combining a total of (8638) images that were written by (52) Arabic authors. We resized all the images to (224 x 224) pixels to prepare them for entering the deep learning models. In addition, we performed an offline data augmentation on the images to optimize the authors' recognition process. Afterward, we developed the models utilizing the commonly used learning hyperparameters in most studies and trained them on the collected dataset.

The initially generated evaluation metrics were not satisfactory. Thus, we set three hypotheses and experimented each hypothesis with five different values, looking for the best strategy in recognizing the authors. The hypotheses were seeking to tune and control the main parameters affecting the models' learning process. Thus, we experimented 1) minimizing the learning rate, 2) increasing the number of the final classification dense layers, and 3) increasing the neurons number on the dense layers. We found out that the first and the third hypotheses hold true since the models recorded highest evaluation parameters after minimizing the learning rate from (1e-2) to (1e-6) and after increasing the number of neurons from (64) to (1024). Moreover, we found out that the second hypothesis didn't hold true. That is because none of the tested four deep learning models improved their performance after increasing the number of final classification dense layers to become two "ReLUs" and one "Sigmoid" beside the original "Softmax" classification dense layer. However, the second hypothesis helped us to reach the most crucial learning hyperparameter that raised the evaluation parameters significantly in all tested deep learning models, which was utilizing the "Sigmoid" activation function in addition to the existing original "Softmax" classification dense layer.

The accuracy, precision, recall, and F-score of each developed model computed. Furthermore, we produced the confusion matrix for each model to compare the predicted authors with the ground truth ones. Eventually, we reached high results since all the utilized four deep learning models recorded a final validation accuracy that is higher than 95% successful recognition of the Arabic authors.

References

[1] M. Al-Ayyoub, A. Nuseir, K. Alsmearat, Y. Jararweh, and B. Gupta, "Deep learning for Arabic NLP: A survey," *J. Comput. Sci.*, vol. 26, pp. 522–531, 2018. doi: 10.1016/j.jocs.2017.11.011.

[2] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review, " *Rom. J. Phys.*, vol. 61, pp. 1120–1132, 2017. doi: 10.1162/NECO.

[3] A. Dureja and P. Pahwa, "Image retrieval techniques: a survey," *Int. J. Eng. Technol.*, vol. 7, no. 2, p. 215-219, 2018. doi: 10.14419/ijet.v7i1.2.9231.

[4] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *J. Geotech. Geoenvironmental Eng.*, vol. 12, 04015009, 2015. doi:10.1061/(ASCE)GT.1943-5606.0001284.

[5] B. Bagasi and L. Elrefaei, "Arabic Manuscript Content Based Image Retrieval: A Comparison between SURF and BRISK Local Features," *Int. J. Comput. Digit. Syst.*, vol. 7, no. 6, pp. 355–364, 2019. doi: 10.12785/ijcds/070604.

[6] K. Adam, A. Baig, S. Al-Maadeed, A. Bouridane, and S. El-Menshawy, "KERTAS: dataset for automatic dating of ancient Arabic manuscripts," *Int. J. Doc. Anal. Recognit.*, vol. 21, no. 4, pp. 283–290, 2018. doi: 10.1007/s10032-018-0312-3.

[7] A. Asi, A. Abdalhaleem, D. Fecker, V. Märgner, and J. El-Sana, "On writer identification for Arabic historical manuscripts," *Int. J. Doc. Anal. Recognit.*, vol. 20, no. 3, pp. 173–187, 2017. doi: 10.1007/s10032-017-0289-3.

[8] M. H. Yahia and W. G. Al-Khatib, "Content-Based Retrieval of Arabic Historical Manuscripts Using Latent Semantic Indexing," *4th International Conference on Arabic Language Processing, Rabat, Morocco*, pp. 165–169, 2012.

[9] Z. Al Aghbari and S. Brook, "Word Stretching for Effective Segmentation and Classification of Historical Arabic Handwritten Documents," *Proc. 2009 3rd Int. Conf. Res. Challenges Inf. Sci. RCIS 2009*, pp. 217–224, 2009. doi: 10.1109/RCIS.2009.5089285.

[10] D. Bagnall, "Author identification using multi-headed recurrent neural networks, " *CLEF (Working Notes)* arXiv:1506.04891v2, pp. 1–11, 2015.

[11] V. Nigam, "Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning," *Lect*, 11-Sep-2018. [Online] *Towards Data Science.* Available at: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90 [Accessed 07 Jan 2019].

[12] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," *Eurasip J. Wirel. Commun. Netw.*, vol. 211, no. 1, pp. 1–12, 2017. doi: 10.1186/s13638-017-0993-1.

[13] N. Schaetti, "UniNE at CLEF 2017 : TF-IDF and Deep-Learning for Author Profiling Notebook for PAN at CLEF 2017," pp. 1–11, 2017.

[14] C. Qian, T. He, R. Zhang, "Deep Learning based Authorship Identification," *Report, Stanford University*, pp. 1–9, 2017.

[15] S. He, L. Schomaker, "Deep Adaptive Learning for Writer Identification based on Single Handwritten Word Images," *Pattern Recognit.* arXiv:1809.10954v1 , pp. 06–27, 2018. doi: 10.1016/j.patcog.2018.11.003.

[16] P. Goldsborough, "A Tour of TensorFlow," *arXiv preprint* arXiv:1610.01178, pp. 1-16, 2016.

[17] S. Bahrampour, N. Ramakrishnan, L. Schott, M. Shah, "Comparative Study of Deep Learning Software Frameworks," *Res. Technol. Center, Robert Bosch LLC.* arXiv:1511.06435, pp. 1-9, 2015.

[18] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint* arXiv:1704.04861, pp. 1-9, 2017.

[19] K. He, X. Zhang, S. Ren, J. Sun, "Identity Mappings in Deep Residual Networks," *European Conference on Computer Vision, Springer*, arXiv:1603.05027, pp. 630-645, 2016.

[20] S. Zagoruyko, N. Komodakis, "Wide Residual Networks," *Univ. Paris-Est, École Des Ponts, Paris Tech, Fr. arXiv preprint* arXiv:1605.07146, pp. 1-15, 2017.

[21] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, "Densely Connected Convolutional Networks," *In Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 4700-4708, 2018.

[22] S.H. Tsang, "Review: VGGNet — 1st Runner-Up (Image Classification), Winner (Localization) in ILSVRC 2014," 22-Aug-2018 [Online] *Medium.* Available at: https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvlc-2014-image-classification-d02355543a11 (accessed 03 March 2019).

[23] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," *Am. J. Heal. Pharm. ICLR 2015 conference paper,* arXiv:1409.1556, vol. 75, p. 398–406, 2015. doi: 10.2146/ajhp170251.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Enzyme Microb. Technol. In Proceedings of the IEEE conference on computer vision and pattern recognition.* vol. 19, no. 2, pp. 107–117, 2015.

[25] F. Alaei, A. Alaei, U. Pal, and M. Blumenstein, "A Comparative Study of Different Texture Features for Document Image Retrieval," *Expert Syst. Appl.*, vol. 121, pp. 97–114, 2018. doi: 10.1016/j.eswa.2018.12.007.

[26] S. Minaee, "20 Popular Machine Learning Metrics. Part 1: Classification & Regression Evaluation Metrics," 28-Oct-2019 [Online] *Medium.* Available at: https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce (accessed 25 December 2019).

[27] J. Brownlee, "How to Configure the Learning Rate When Training Deep Learning Neural Networks," 23-Jan-2019 [Online] *Machine Learning Mastery.* Available at: https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks (accessed 25 September 2019).

Manal M. Khayyat received the B.Sc. degree (Hons.) in Computer Science from King Abdulaziz University, Saudi Arabia, in 2007 and received M.Sc. degree of Applied Science in Quality Systems Engineering from Concordia University, Canada, in 2015. She is currently a PhD student in the Department of Computer Science at King Abdulaziz University. She worked at the IT department of Effat University, Saudi Arabia, from 2007 to 2010. Then, she worked as a lecturer at King Abdulaziz University, from 2012 to 2019 and she is currently working as a lecturer at Umm Al-Qura University, Saudi Arabia. Her research interests include computer vision, image processing, natural language recognition, and deep learning.

Lamiaa A. Elrefaei received the B.Sc. degree (Hons.) in electrical engineering (electronics and telecommunications), and the M.Sc. and Ph.D. degrees in electrical engineering (electronics) from the Faculty of Engineering at Shoubra, Benha University, Egypt, in 1997, 2003, and 2008, respectively. She held a number of faculty positions at Benha University, as a Teaching Assistant, from 1998 to 2003, as an Assistant Lecturer, from 2003 to 2008, and has been a Lecturer, since 2008. She is currently an Associate Professor with the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. Her research interests include computational intelligence, biometrics, multimedia security, wireless networks, and nano networks.