Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 6, December 2024, pp. 4294~4305
ISSN: 2302-9285, DOI: 10.11591/eei.v13i6.8178

Improving Arabic handwritten text recognition through transfer learning with convolutional neural network-based models

Hicham Lamtougui[1], Hicham El Moubtahij[2], Hassan Fouadi[1], Khalid Satori[1]
[1]LISAC Laboratory, Department of Computer Science, Faculty of Sciences Dhar EL Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[2]Higher School of Technology, Ibn Zohr University, Agadir, Morocco

Article Info

Article history:

Received Jan 13, 2024
Revised Apr 16, 2024
Accepted May 17, 2024

Keywords:

Arabic handwritten
IFN/ENIT
ResNet50
Transfer learning
VGG16

ABSTRACT

Arabic handwritten text recognition is a complex and challenging research domain. This study proposes an offline Arabic handwritten word recognition system based on transfer learning. The system exploits four pre-trained convolutional neural network (CNN) architectures, namely VGG16, ResNet50, AlexNet, and InceptionV3. In addition, a specialized image recognition model derived from the ImageNet dataset is incorporated. A combination strategy is designed to combine transfer learning with specific fine-tuning techniques, aiming to improve recognition accuracy. The study is conducted on the IFN/ENIT dataset, which includes images of Tunisian City and village names. The results show that the proposed system achieves a recognition accuracy of 94.73%, which is significantly higher than the accuracy rates achieved by previous approaches. These results suggest that the proposed system is a promising approach for Arabic handwritten text recognition.

Corresponding Author:

Hicham Lamtougui
LISAC Laboratory, Department of Computer Science, Faculty of Sciences Dhar EL Mahraz
Sidi Mohamed Ben Abdellah University
Fez, Morocco
Email: hicham.lamtougui@usmba.ac.ma

1. INTRODUCTION

Arabic, spoken by over 400 million individuals worldwide, has not yet been integrated into an industrial system for the automatic recognition of its handwriting. Despite the vast potential applications, such as automating postal sorting, processing checks, managing forms, or indexing ancient manuscripts, the automatic recognition of Arabic handwriting remains a significant challenge. This challenge stems from the inherent complexity of handwriting recognition, both in terms of specific applications and associated processing techniques.

Handwriting recognition aims to convert handwritten text into a symbolic form interpretable by a computer system. It primarily divides into two categories: online recognition, which captures pen tip movements during writing, and offline recognition, focused on analyzing digital images of handwritten texts [1]. This research field is continually evolving, encompassing issues ranging from simple character recognition to cursive writing transcription.

Within the context of these challenges, contemporary research is gravitating towards more sophisticated modeling strategies, thereby transcending the inherent limitations of classical hidden Markov models (HMM) [2] and artificial neural networks (ANN) [3]. A prominent trend involves the amalgamation of these architectures with advanced machine learning techniques, such as support vector machines (SVM) [4], random forests (RF) [5], and decision trees (DT) [6]. Concurrently, deep learning methodologies, notably convolutional neural networks (CNN), are emerging as indispensable tools, especially in demanding computer vision domains like handwritten recognition [7].

Previous research has addressed various approaches to Arabic handwritten text recognition, demonstrating significant advancements in the classification and recognition of Arabic characters and words. However, despite these advancements, a lack of critical discussion, thorough comparison, and interpretation of results is often observed in the existing literature. Areas requiring improvement include generalizing recognition models to diverse handwriting styles and considering variations in writing conditions, such as different Arabic calligraphy styles. Our work addresses this gap by aiming to explore the innovative application of transfer learning techniques to enhance the accuracy of Arabic handwritten content recognition.

Numerous articles, publications, and research projects focus on the classification and recognition of handwritten Arabic texts. Siddhu proposed a method combining statistical and structural approaches for recognizing Arabic handwritten characters. This method focuses on the main body of the character and uses point descriptors to identify its exact shape. Results from the IFN/ENIT database demonstrate high accuracy of 96.71% for main bodies and 94.52% for complete characters. Subsequently, Khémiri *et al.* [8] introduced the use of Bayesian and CNN for recognizing handwritten Arabic words. Structural features are extracted from word images and serve as inputs for various Bayesian networks, such as Naïve Bayes, tree augmented Naïve Bayes, forest augmented Naïve Bayes, and HMMs. Additionally, a 2D convolutional network architecture is designed to process word images, and a combination of deep belief network (DBN) and CNN is suggested to enhance classification accuracy. Experiments on the IFN-ENIT database show 91.20% accuracy for the DBN-CNN combination compared to other models. Abandah *et al.* [9] presented a system for recognizing cursive Arabic words that utilizes an efficient segmentation approach, feature extraction, and a recurrent neural network. This segmentation approach reduces labeling errors by 18.5%, sequence errors by 22.3%, and execution time by 31% compared to a holistic approach. Elleuch *et al.* [10] introduced a novel model combining a CNN and a SVM for offline Arabic handwriting recognition. This model employs dropout techniques to prevent overfitting and outperforms models based solely on CNN or SVM. The model's performance, evaluated on Arabic handwritten datasets, demonstrates superior efficiency. Ahmad and Fink [11], introduced a text recognition system for handwritten Arabic based on HMM models. This system offers an innovative representation of Arabic characters by separating central forms from diacritical marks and using smaller secondary forms. This method significantly reduces the number of models required for text recognition and enhances the performance of the contextual HMM system compared to classical Arabic character models, even with limited training data. In their study, Eltay *et al.* [12] explored various deep learning architectures to address challenges posed by cursive handwritten text. They also highlighted the importance of handling imbalanced datasets and proposed a novel adaptive data augmentation algorithm that assigns weights to words based on the average probability of each class. The objective of this approach is to promote class diversity within datasets. Lamtougui *et al.* [13] introduced a new model combining a CNN and a bidirectional long short-term memory (BLSTM), complemented by a connectionist temporal classification (CTC) layer for recognition on the IFN/ENIT database. A data augmentation algorithm is also employed to enhance training data quality. Experimental results reveal that the proposed model achieves a precision rate of 92.11%.

Our new contributions consist of the exploration and innovative application of transfer learning techniques specifically for Arabic handwritten content recognition. We have deepened this approach by integrating various CNN models, including VGG16, ResNet50, AlexNet, and InceptionV3, to assess their potential for improving the accuracy of the IFN/ENIT dataset. Additionally, we have introduced a novel methodology that merges transfer learning with specific fine-tuning techniques to enhance word recognition accuracy within the IFN/ENIT corpus.

The application of transfer learning with CNN models and fine-tuning offers several significant advantages for Arabic handwritten text recognition. This approach improves accuracy by fine-tuning the general knowledge of pre-trained CNN models on massive ImageNet datasets for the specific task of Arabic handwritten text recognition. Furthermore, it reduces overall training time by avoiding expensive training on large datasets. This approach also enhances the model's robustness to variations in Arabic handwriting and optimizes data utilization while providing ease of implementation due to the availability of deep learning frameworks and the simplicity of fine-tuning. The results of our research promise to significantly improve the accuracy of Arabic handwritten text recognition and pave the way for new applications in various fields.

The document is structured as follows: section 2 introduces the various characteristics and databases of the Arabic language. The methods employed in this research are discussed in section 3, and a detailed analysis of experimental results is presented in section 4. Finally, conclusions and future perspectives are addressed in section 5.

2. BACKGROUND

2.1. Characteristics of the Arabic language

Arabic is one of the most widely spoken Semitic languages in the world and its script is the third most widely used. The Arabic script is used by many other languages such as Urdu, Persian, and Uyghur. Arabic script is cursive, both in printed and handwritten form. It is made up of letters that represent consonants. Arabic is written from right to left and has 28 basic characters, with no distinction between upper and lower case.

Arabic script recognition falls within the general framework of cursive script recognition, but little work has been done in this area compared to other writing systems such as Latin or Chinese. Arabic characters are distinguished by their structure and the way they are linked to form words. The characteristics of Arabic writing include the fact that characters are written from right to left, that there are no capital letters, that the same character can have up to four different forms, that certain characters are distinguished by the presence and position of dots, and that an Arabic word is composed of related components (Table 1).

Table 1. Arabic characters in printed and handwritten forms

Printed Arabic character	Name	Handwritten Arabic character	Printed Arabic character	Name	Handwritten Arabic character
ا	Alif	أ	ض	Dad	ض
ب	Ba	ب	ط	Ta	ط
ت	Ta	ت	ع	Ayn	ع
ث	Tha	ث	غ	Ghay	غ
ج	Jim	ج	ف	Fae	ف
ح	Ha	ح	ق	Qaf	ق
خ	Kha	خ	ك	Kaf	ك
د	Dal	د	ل	Lam	ل
ذ	Dhal	ذ	م	Mm	م
ر	Ra	ر	ن	Nun	ن
ز	Zay	ز	ه	Ha	ه
س	Sin	س	و	Waw	و
ص	Sad	ص	ي	Yae	ي

The size of Arabic characters varies from character to character and within the same character. Arabic texts are not vocalized, meaning readers must infer meaning from context. The overlapping of characters and the presence of ligatures make it difficult to recognize handwritten Arabic words. Recognition of handwritten Arabic text faces several challenges, such as handwriting variability between authors, overlapping issues, and variation in character shapes depending on their position. Some Arabic characters have diacritics (dots above or below), which can be written in different ways by different authors. It is also important to note that some ligatures, such as the "lam-alif" ligature, are mandatory, while other ligatures are not. Authors can choose to write ligatures or characters separately, even in similar contexts (Figure 1).

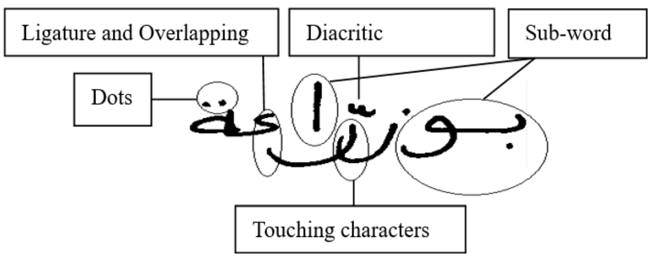

Figure 1. Some of Arabic script characteristic

2.2. Challenges of handwritten Arabic recognition

Recognition of Arabic handwritten script poses a complex challenge that is actively being researched. Several obstacles encountered, such as variability in writing styles among authors, illegible texts, and ligatures, mirror those found in other writing systems. The main difficulties in Arabic handwritten script recognition include:

- Variability in character forms Arabic script exhibits significant variability in character forms based on their position within a word, writing style, and author, making it challenging to distinguish between characters.
- Diacritics: Arabic characters may be accompanied by diacritics such as dots or lines, indicating pronunciation or word meaning. The presence of diacritics complicates writing and increases the risk of recognition errors.
- Ligatures: Arabic letters can be linked together to form ligatures. The presence of ligatures can make word segmentation and recognition of individual characters difficult.
- Letter connectivity: Arabic letters are typically connected to form words. This connectivity can make word segmentation and recognition of individual characters challenging.
- Writing styles: Arabic writing styles are generally classified into two main categories: typewritten (*Nuskah*) and handwritten (*Ruqqah*).

2.3. Arabic handwriting database

The need for a freely available standard Arabic script database has become crucial for the recognition of Arabic script. Previously, the lack of such databases hampered research on recognizing Arabic text against other languages. However, several databases of Arabic texts and words are now available, addressing various aspects of research in Arabic handwriting recognition as illustrated in Table 2. These databases include datasets focused on the recognition of legal amounts, widely used handwritten Tunisian city names, handwritten text images by multiple authors, numerals, isolated Arabic letters, and more. Additionally, an efficient system was developed to generate synthetic handwritten words and lines of text, providing an alternative to resource-intensive offline databases.

Table 2. Summary of database used in Arabic handwritten text recognition

Database	Writers	Description
CENPARMI [14]	180 writers	2,499 words
AHDB [15]	100 writers	Words and sentences that used in in writing checks
IFN/ENIT [16]	411writers	26,459 handwritten Tunisian town
AHTID/MW [17]	53 writers	3,710 text lines and 22896 word images
KHATT database [18]	1,000 writers	2,000 similar-text paragraph images and their extracted text line images
HACDB [19]	50 writers	6,600 segmented characters
Al-Isra [20]	500 writers	37,000 words, 10,000 digits, 2,500 signatures, and 500 sentences
ADBase/MADBase [21]	700 writers	70,000 digits

3. PROPOSED METHOD

The method for developing our Arabic handwritten recognition model begins with meticulous data preparation, involving the collection and selection of data from the IFN/ENIT database and other relevant sources. Images are cleaned and preprocessed, normalized to grayscale, and resized to a uniform size, with the application of noise reduction and contrast enhancement techniques. Subsequently, four CNN architectures are evaluated, including VGG16, ResNet-50, AlexNet, and InceptionV3, along with exploration of architectures specialized in Arabic handwritten recognition. The most performing model is selected based on validation set and adapted to Arabic handwriting specifics by adjusting hyperparameters.

Once the models are selected, a training and optimization process commences, utilizing a transfer learning strategy on the training set. Cross-validation is used to evaluate and fine-tune the model, employing regularization techniques to avoid overfitting and enhance model robustness. The learning algorithm is optimized to minimize training time and energy consumption. After training, model performances are assessed on the test set, using various metrics to compare results with other Arabic handwritten recognition methods. Errors are analyzed to identify potential improvement areas.

Finally, fine-tuning technique is employed to enhance the performance of pretrained models on limited datasets such as IFN/ENIT. For each model, fine-tuning specifically targets upper layers, adjusting weights to focus on Arabic recognition-specific features. The choice of the number of layers for fine-tuning is crucial, depending on the size and complexity of the training dataset. Partial fine-tuning is used to mitigate overfitting risk, by freezing lower layers to effectively focus on domain-specific features.

3.1. Pre-processing

Before training models specifically designed for Arabic handwritten text recognition using transfer learning, meticulous data preparation is paramount. To tailor a model to the ImageNet data for Arabic handwritten text recognition with the IFN/ENIT dataset, the IFN/ENIT database must undergo several transformations. These steps involve converting the images to a format suitable for ImageNet, adjusting their resolution to ImageNet standards, standardizing pixel values to match ImageNet's scale, and finally, organizing the images according to ImageNet's specific folder structure.

Additionally, we divide our IFN/ENIT database into three categories to maximize the efficiency of the model. By allocating 60% for training, 20% for validation, and 20% for testing, one can fine-tune hyperparameters, mitigate overfitting risks, and attain a reliable assessment of the model's performance on previously unused data. Hence, this tripartite segmentation of the IFN/ENIT dataset is a foundational step in effectively training a robust model for Arabic handwritten text recognition using transfer learning.

3.2. Deep transfer learning for Arabic handwritten text recognition

CNNs, introduced by LeCun in the 1980, are widely used for visual data processing, where they learn hierarchical representations from data. Popular architectures include residual networks, VGG, ResNet, Inception, and AlexNet, each suitable for different applications and levels of data complexity [22]. Despite potential degradation and vanishing gradient issues, deep CNNs (DCNNs) are commonly used in the commercial domain, especially for text recognition [22], [23]. The study in question focuses on four architectures of pre-trained CNN: VGG16, ResNet-50, AlexNet and InceptionV3.

In the context of Arabic handwritten text processing, transfer learning is an essential technique leveraging DCNN architectures [24]. However, these intricate architectures typically demand a large number of training samples, which can be costly and challenging to obtain. Transfer learning addresses this challenge by transferring knowledge gained from similar previous tasks to the target task. This is achieved by utilizing a pretrained model on a large dataset. The process involves six steps: obtaining a pretrained model, creating a base model, freezing layers of the pretrained model, adding new trainable layers, training these layers on the target dataset, and optionally fine-tuning the model to enhance performance [25]. Additionally, during the training phase, various hyperparameters, including learning rate, batch size, and input image dimensions, exert significant influence on the performance of the DCNN. By leveraging previously acquired knowledge, transfer learning overcomes the lack of training data and enhances the performance of models in Arabic handwritten text processing, as illustrated in Figure 2.

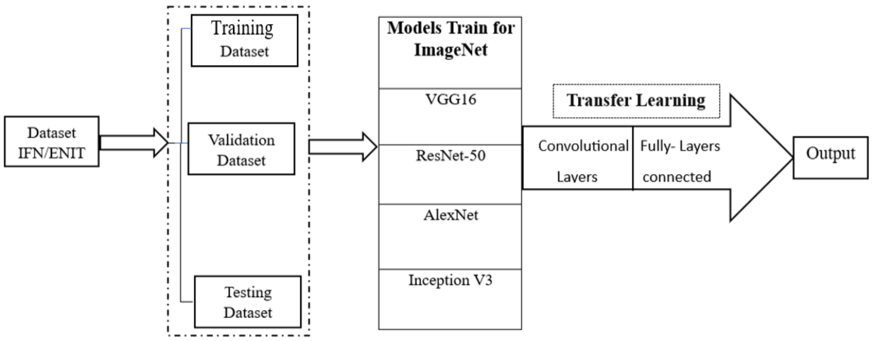

Figure 2. System architecture of transfer learning

3.2.1. Transfer learning from VGG16 convolutional neural network model

The VGG16 model is a CNN composed of 16 layers, designed to extract complex features from images. It is organized into five convolutional blocks, each followed by two pooling layers, and culminates in three classification layers. The final layer uses a sigmoid activation function for datasets containing two categories or fewer, while it employs the SoftMax function for datasets with more than three categories. Developed by Karen and Andrew [26], the VGG16 architecture primarily utilizes 3×3 dimension filters in its convolutional layers. Additionally, 1×1 dimension filters are used to perform linear transformations of input channels, followed by non-linear activation. A one-pixel padding operation is applied to the 3×3 convolutional layers to preserve spatial resolution, while spatial pooling is achieved using 2×2 windows with a stride of 2.

Moreover, the model incorporates three fully connected (FC) layers with 4,096 channels for the first two and 1,000 channels for the last one, where each channel corresponds to a specific class. The final layer is a SoftMax layer dedicated to classification. The VGG16 architecture referenced in this study is detailed in Table 3. It's also noteworthy that the VGG16 model has been pre-trained on the ImageNet dataset, which comprises over a million images distributed across 1,000 categories. This pre-training phase equips the model with the ability to identify distinctive features for image recognition.

Table 3. VGG16 model for imageNet

No layers	Convolution	Output dimension	Pooling	Output dimension
1, 2	Convolution layer of 64 channel of 3×3 kernel with padding 1, stride 1	224×224×64	Max pool stride=2, size 2×2	112×112×64
3, 4	Convolution layer of 128 channel of 3×3 kernel	112×112×128	Max pool stride=2, size 2×2	56×56×128
5, 6, 7	Convolution layer of 256 channel of 3×3 kernel	56×56×256	Max pool stride=2, size 2×2	28×28×256
8, 9, 10	Convolution layer of 512 channel of 3×3 kernel	28×28×512	Max pool stride=2, size 2×2	14×14×512
11, 12, 13	Convolution layer of 512 channel of 3×3 kernel	14×14×512	Max pool stride=2, size 2×2	7×7×512

3.2.2. Transfer learning from ResNet50 convolutional neural network model

ResNet50 is a deep neural network based on residual architectures, designed to build very deep architectures with more than 150 layers. The ResNet50 model includes 50 layers, some of which were pre-trained on the ImageNet dataset. Through the use of bottleneck blocks, ResNet50 manages to be more efficient. It consists of five CNN blocks interconnected by shortcuts. Deep residual features (DRF) are extracted using the last convolution layer.

Proposed by He *et al*. [27] the ResNet50 model follows a similar approach to that of VGG networks, using convolutional layers of size 3×3, with an input size fixed at 224×224. The layers in the model are designed so that those with the same number of filters have the same output. If the convolved output size is halved, the number of filters is doubled to preserve time complexity per layer. The model ends with a medium pooling layer, followed by a FC layer at 1000 nodes using the SoftMax function. Compared to VGG networks, ResNet50 uses fewer filters and has reduced complexity. Other variants such as ResNet101 and ResNet152 also exist. The layer configuration of the ResNet50 network is shown in Figure 3.

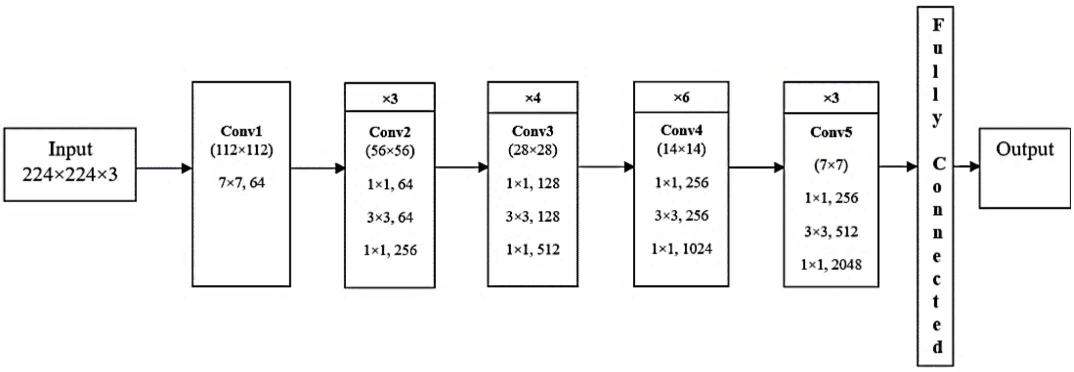

Figure 3. ResNet50 architecture

3.2.3. Transfer learning from AlexNet convolutional neural network model

AlexNet won the ImageNet LSVRC-2012 competition by a significant margin with an error rate of 15.3%, thus surpassing the other competitors [28]. This 8-layer CNN has been pre-trained on millions of images, allowing it to classify images into 1,000 different categories, with rich representations for various image types such as medical imaging, biometrics, scene classification, embedded systems, and character recognition. AlexNet landslide victory marks a major breakthrough in computer vision, demonstrating the superiority of feature learning over manually designed ones. Adaptations have been made to the AlexNet architecture to accommodate two small GPUs, although it has similarities to the LeNet architecture illustrated in Figure 4.

3.2.4. Transfer learning from inceptionV3 convolutional neural network model

Inception-v3 is an improved version of Inception-v1, using convolutional filters of different sizes to extract features from the input while preserving local information [29]. The final model is composed of this feature extraction part and a stack of convolutional layers concatenated with the first one. The FC layer is the only editable layer to avoid overfitting, with 1,024 nodes and a ReLU activation function, as well as a dropout of 0.4.

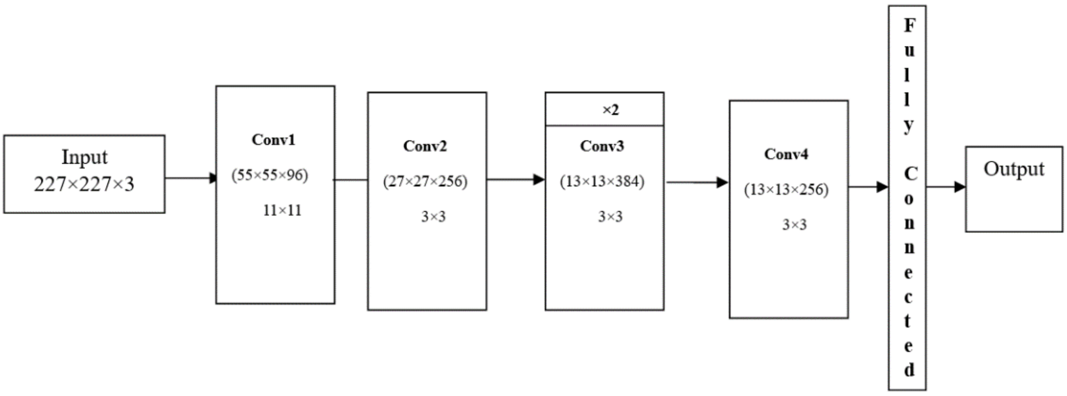

Figure 4. AlexNet architecture

The classification is performed with a softmax function and the loss function used is "binary_crossentropy", optimized with the "Adam" optimizer. The initial learning rate was gradually reduced to facilitate model convergence. For transfer learning, the feature extractor was prepared, and a linear classifier was trained in the FC layer to fit the new dataset, which was smaller but different from the original set indicated in Figure 5.

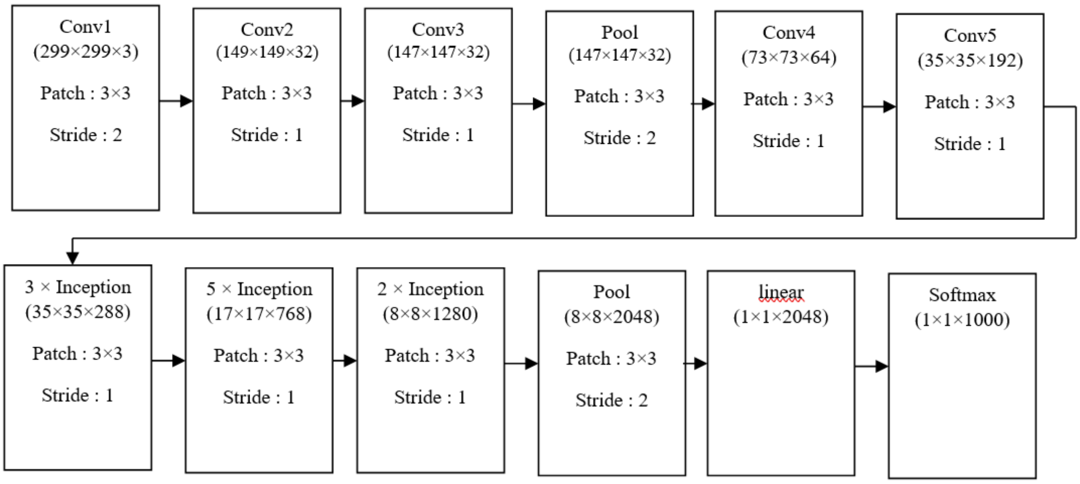

Figure 5. InceptionV3 architecture

3.3. Partial fine-tuning for Arabic handwritten text recognition

In the domain of Arabic handwritten text recognition, fine-tuning proves to be a fundamental technique designed to enhance the performance of a pre-trained model when applied to new datasets. The pre-trained model consists of a diverse set of images of handwritten text from various contexts [30]. A specific approach to fine-tuning, referred to as partial fine-tuning, is emphasized in scenarios where datasets are relatively limited, thereby reducing the risk of overfitting [31].

This study focuses on the utilization of the VGG16 model, pre-trained on ImageNet, for the recognition of the IFN/ENIT database. For VGG16, fine-tuning is directed towards the upper layers, adopting a partial approach where the lower layers remain fixed. The decision to fine-tune the last five layers aims to concentrate learning on domain-specific features of Arabic handwritten text recognition, thereby decreasing the risk of overfitting. A remarkable accuracy of 94.73% was achieved on the test dataset.

In the case of ResNet50, partial fine-tuning targets the last four layers, responsible for extracting features specific to Arabic handwritten text recognition. The results demonstrate an accuracy of 93.88% on the test dataset. The justification for the choice of fine-tuning layers underscores their relevance in feature extraction. By freezing the lower layers, the study effectively concentrates on domain-specific characteristics.

For AlexNet, fine-tuning the last five layers aims to adapt the model to Arabic handwritten script while preserving general knowledge from ImageNet. By freezing the lower layers, attention is directed towards learning specific traits from the IFN/ENIT dataset, resulting in an accuracy of 92.70% on the test dataset.

The approach with InceptionV3 focuses on the last five layers, resulting in an accuracy of 92.87% on the test dataset. Partial fine-tuning thus emerges as a promising strategy, effectively adapting models to Arabic handwritten text recognition, demonstrating its potential in limited datasets such as IFN/ENIT. The results suggest a fruitful avenue for enhancing performance in the specific context of this task. The choice of the number of layers for fine-tuning is a crucial decision, dependent on the size and complexity of the training dataset. In the case of the relatively small IFN/ENIT dataset, mitigating the risk of overfitting is essential. Freezing the lower layers allows effective concentration on domain-specific features, ultimately improving the model's performance on the test dataset.

4. EXPERIMENTAL RESULT AND DISCUSSION

In this section, we outline our IFN/ENIT dataset and implementation details. Subsequently, we delve into the detailed experimental frameworks used. The results obtained with the proposed systems are presented and compared, employing different sets for training, validation, and testing.

4.1. IFN/ENIT dataset

There is a growing interest in Arabic handwriting recognition, highlighting the need for an accessible database representing different handwriting styles. The lack of Arabic databases has been considered as one of the reasons for the lack of research in the field of Arabic text recognition compared to other languages. Several databases of Arabic texts and words have now been created to support research in Arabic handwriting recognition, covering characters, numbers, words and texts. Among them, the IFN/ENIT database of handwritten names of Tunisian cities is the most commonly used by researchers working on Arabic handwriting recognition systems. This database was developed in collaboration between the Institute of Communication Technologies (IFN) of the Technical University of Braunschweig in Germany and the National School of Engineers of Tunis (ENIT) in Tunisia, as detailed in Table 4.

Table 4. Statistics of IFN/ENIT

Sets	Words	Characters
A	6537	51984
B	6710	53862
C	6477	52155
D	6735	54166
E	6033	45169
Total	32492	257336

4.2. Evaluation measures

In handwritten text recognition problems, several metrics are identified to evaluate the performance of the models. Simply using precision as the only metric can be inappropriate, because it does not measure different errors in the same way, including false negatives (FNs) and false positives (FPs). It is therefore best to use metrics such as precision, recall, and F1-score to get a better understanding of model results in this context.

It is essential to note that a decision can be considered either true or false. So this gives us four possibilities in total: true positive (TP) is positive data that is correctly classified as positive, true negative (TN) is negative data correctly classified as negative, and FP is for negative data misclassified as positive. Using as (1) to (4):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1-Score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{4}$$

4.3. Implementation details

We present our innovative approach for training text recognition models using transfer learning on the IFN/ENIT database. To ensure a fair comparison, all models share an identical design architecture. We employ partial fine-tuning to adapt the models to domain-specific features of Arabic handwritten text recognition, thereby concentrating learning on relevant aspects. Table 5 details the implementation of four neural network architectures (VGG16, ResNet-50, AlexNet, and InceptionV3), each with specific input size characteristics. The models are configured with a learning rate of 0.0001, chosen for its ability to ensure stable convergence on pre-trained models. The RMSprop optimizer is favored due to its compatibility with a low learning rate, particularly suited to our text recognition application. The CategoricalCrossentropy loss function is used to address multi-class classification aspects. The choice of a batch size of 32 and training for 20 epochs aims to strike a balance between computational efficiency and training quality. The architecture of CNNs and their variants is central to this approach, implemented using TensorFlow, a dedicated deep learning library in Python. This method aims to establish a standard for training text recognition models, thereby simplifying comparisons and performance evaluations across different architectures and input dimensions.

Table 5. Implementation details

Model	Input shape	Learning rate	Epochs	Batch size	Optimizer	Loss function
VGG16	(224, 224,1)	0.0001	20	32	RMSprop	CategoricalCrossentropy
ResNet-50	(224, 224,1)	0.0001	20	32	RMSprop	CategoricalCrossentropy
AlexNet	(227,227,1)	0.0001	20	32	RMSprop	CategoricalCrossentropy
InceptionV3	(229,229,1)	0.0001	20	32	RMSprop	CategoricalCrossentropy

The use of partial fine-tuning has yielded exceptional results. For VGG16, it led to effective adaptation with notable performance on the test dataset, although the exact accuracy is not explicitly mentioned. For ResNet50, partial fine-tuning is directed towards the last four layers, focusing on extracting features specific to Arabic handwritten text recognition. This approach resulted in compelling outcomes with a positive impact on accuracy on the test dataset, although the precise figure is not explicitly stated. AlexNet and InceptionV3 follow a similar strategy by adapting their last layers, thus enhancing their performance in the specific context of IFN/ENIT.

4.4. Results and discussion

In the context of this experimental study, we assessed the performance of four CNN architectures, namely VGG16, ResNet50, AlexNet, and Inception-v3, using the transfer learning technique on the IFN/ENIT database for offline recognition of Arabic handwritten text. The selection of these architectures was based on their previous success in image classification tasks. The overall recognition rates achieved by each model are presented in Table 6. The results highlighted that VGG16 and ResNet-50 architectures, leveraging transfer learning, reached remarkably high recognition rates of 94.73% and 93.88%, respectively, demonstrating the effectiveness of the technique for this specific task.

Table 6. Comprehensive comparison of each model using accuracy, precision, recall, and F1-score

Architecture	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
VGG16	94.73	94.83	94.15	92.40
ResNet-50	93.88	94.19	93.32	92.15
AlexNet	92.70	93.45	92.33	90.47
InceptionV3	92.87	93.67	92.50	92.22

To leverage pre-trained models on the ImageNet dataset as the source dataset, we adapted the layers of the models using the IFN/ENIT dataset. This adjustment allowed the model to enhance its capability to recognize simple Arabic handwritten words. We also applied the data augmentation technique using a specific preprocessing function for each model. To assess the model's quality in Arabic word classification, we utilized evaluation metrics such as recall and F1-score. By removing the top layer of the pre-trained models during the partial fine-tuning for our classification task and loading pre-trained weights from the specified file, we froze all other layers of the model except the last three. This approach enabled us to finely adjust the model for our specific task. Figure 6 illustrates the validation curves obtained during the training phase, highlighting the rapid convergence resulting from the application of the transfer learning method.

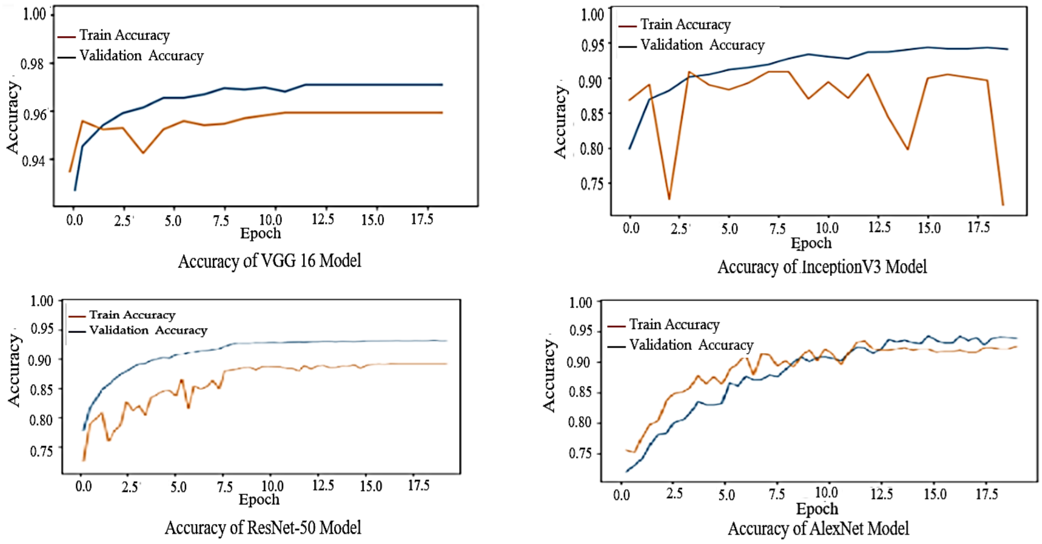

Figure 6. Accuracy of the VGG-16, ResNet50, AlexNet and inception-V3 model

4.5. Comparison to other approaches

We evaluate the performance of our models (VGG16, ResNet50, AlexNet, and Inception-v3) by applying the transfer learning method, and we present detailed results in Table 7. These data explicitly confirm the superiority of our method in terms of recognition accuracy compared to previous approaches. Our approach, utilizing four pretrained CNN models (VGG16, ResNet, AlexNet, and Inception-v3), has demonstrated a remarkable average precision rate for our models. This performance significantly exceeds the results obtained by comparison approaches. The effectiveness of our approach relies on two key aspects. Firstly, the use of pretrained models endows our method with extensive generalization capability to new data. These models, pretrained on the ImageNet database, capture the general features of handwritten text, enabling our approach to maintain accurate recognition of Arabic handwritten text even when confronted with styles different from those used during training.

Table 7. Comparison with state-of-the-art methods on the IFN/ENIT dataset

Reference	Years	Training/testing	Approach	Accuracy (%)
Khémiri *et al.* [8]	2019	abcd/e	DBN+CNN	91.20
Abandah *et al.* [9]	2014	abcd/e	BLSTM-RNN	75.72
Elleuch *et al.* [10]	2015	56 classes	CNN-SVM	92.95
Ahmad and Fink [11]	2019	abcde/f	sub-core-shape HMM	93.32
Eltay *et al.* [12]	2020	abcde/f	BLSTM-CTC-WBS	93.57
Lamtougui *et al.* [13]	2022	abcd/e	CNN-BLSTM-CTC	92.11
Our models	2023	abcde/f	VGG16	94.73
			ResNet-50	93.88
			AlexNet	92.70
			Inception-V	92.87

Furthermore, the strategy of partial fine-tuning adjustment has played a decisive role in our approach. This transfer learning technique, which involves training only the top layers of pretrained models, has considerably reduced training time while preserving high performance. For instance, the accuracy rate of the VGG16 model increased to 94.73% with fine-tuning, illustrating a significant improvement. These results highlight the effectiveness of partial fine-tuning adjustment in the context of our approach. Our findings indicate that our method, leveraging transfer learning and partial fine-tuning adjustment, achieves high recognition performance without requiring full model training. This approach may be beneficial in other areas of Arabic handwritten text recognition.

5. CONCLUSION

Arabic handwritten text recognition is an active research domain that poses specific challenges, such as variability in writing styles, differences in size, and writing quality. In our research, we introduced an approach based on transfer learning to address these challenges. Our method leverages four pre-trained CNN models (VGG16, ResNet, AlexNet, and Inception-v3), which we adapted to the IFN/ENIT Arabic handwritten dataset. We also explored the significance of partial fine-tuning, reducing trainable parameters and training time while maintaining exceptional performance. Our evaluations demonstrated that the VGG16 model achieved an accuracy rate of 94.73%, surpassing all other individual CNN models considered for Arabic handwritten text recognition. These results hold promise for enhancing Arabic handwritten text recognition in applications such as archive digitization, check recognition, and machine translation. Our study has certain limitations, particularly regarding the generalization of results to other datasets.

Our future work is directed towards integrating zero-shot learning (ZSL). Aiming to further improve the performance of Arabic handwritten text recognition. We are confident that ZSL can enhance the capabilities of this technology by enabling the classification of more extensive and diverse datasets, thereby opening new avenues for application.

REFERENCES

[1] R. Plamondon and S. N. Srihari, "Online and offline handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000, doi: 10.1109/34.824821.
[2] H. El Moubtahij, A. Halli, and K. Satori, "Recognition of Off-Line Arabic handwriting words using HMM toolkit (HTK)," in *Proceedings - Computer Graphics, Imaging and Visualization: New Techniques and Trends, CGiV 2016*, 2016, pp. 167–171, doi: 10.1109/CGiV.2016.40.
[3] J. J. Hopfield, "Artificial neural networks," *IEEE Circuits and Devices Magazine*, vol. 4, no. 5, pp. 3–10, Sep. 1988, doi: 10.1109/101.8118.
[4] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning*, 2020, pp. 101–121, doi: 10.1016/B978-0-12-815739-8.00006-7.
[5] L. Breiman, *Random Forests*, vol. 2012-CVIM-, no. 31. Kluwer Academic Publishers, 2001.
[6] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.
[7] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "a survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022, doi: 10.1109/TNNLS.2021.3084827.
[8] A. Khémiri, A. K. Echi, and M. Elloumi, "Bayesian versus convolutional networks for Arabic handwriting recognition," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9301–9319, 2019, doi: 10.1007/s13369-019-03939-y.
[9] G. A. Abandah, F. T. Jamour, and E. A. Qaralleh, "Recognizing handwritten Arabic words using grapheme segmentation and recurrent neural networks," *International Journal on Document Analysis and Recognition*, vol. 17, no. 3, pp. 275–291, 2014, doi: 10.1007/s10032-014-0218-7.
[10] M. Elleuch, R. Maalej, and M. Kherallah, "A New design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition," in *Procedia Computer Science*, 2016, vol. 80, pp. 1712–1723, doi: 10.1016/j.procs.2016.05.512.
[11] I. Ahmad and G. A. Fink, "Handwritten Arabic text recognition using multi-stage sub-core-shape HMMs," *International Journal on Document Analysis and Recognition*, vol. 22, no. 3, pp. 329–349, 2019, doi: 10.1007/s10032-019-00339-8.
[12] M. Eltay, A. Zidouri, and I. Ahmad, "Exploring deep learning approaches to recognize handwritten Arabic texts," *IEEE Access*, vol. 8, pp. 89882–89898, 2020, doi: 10.1109/ACCESS.2020.2994248.
[13] H. Lamtougui, H. El Moubtahij, H. Fouadi, and K. Satori, "An efficient hybrid model for Arabic text recognition," *Computers, Materials and Continua*, vol. 74, no. 2, pp. 2871–2888, 2023, doi: 10.32604/cmc.2023.032550.
[14] Y. Al-Ohali, M. Cheriet, and C. Suen, "Databases for recognition of handwritten Arabic cheques," *Pattern Recognition*, vol. 36, no. 1, pp. 111–121, 2003, doi: 10.1016/S0031-3203(02)00064-X.
[15] S. Al-Ma'adeed, D. Elliman, and C. A. Higgins, "A data base for Arabic handwritten text recognition research," in *Proceedings - International Workshop on Frontiers in Handwriting Recognition*, 2002, pp. 485–489, doi: 10.1109/IWFHR.2002.1030957.
[16] H. El Abed and V. Margner, "The IFN/ENIT-database - a tool to develop Arabic handwriting recognition systems," in *2007 9th International Symposium on Signal Processing and Its Applications*, Feb. 2007, pp. 1–4, doi: 10.1109/ISSPA.2007.4555529.
[17] M. E. M. Musa, "Arabic handwritten datasets for pattern recognition and machine learning," in *2011 5th International Conference on Application of Information and Communication Technologies*, 2011, pp. 1–3, doi: 10.1109/ICAICT.2011.6110959.
[18] S. A. Mahmoud *et al.*, "KHATT: Arabic offline handwritten text database," in *Proceedings - International Workshop on Frontiers in Handwriting Recognition, IWFHR*, 2012, pp. 449–454, doi: 10.1109/ICFHR.2012.224.
[19] A. Lawgali, M. Angelova, and A. Bouridane, "HACDB: Handwritten Arabic characters database for automatic character recognition," in *2013 4th European Workshop on Visual Information Processing, EUVIP 2013*, 2013, pp. 255–259.
[20] N. Kharma, M. Ahmed, and R. Ward, "A new comprehensive database of handwritten Arabic words, numbers, and signatures used for OCR testing," in *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.99TH8411)*, Canada, 1999, pp. 766-768 vol. 2, doi: 10.1109/CCECE.1999.808042.
[21] E. Ali El-Sherif and S. Abdelazeem, "A two-stage system for Arabic handwritten digit recognition tested on a new large database," in *International Conference on Artificial Intelligence and Pattern Recognition 2007, AIPR 2007*, 2007, pp. 237–242.
[22] H. Imaduddin, F. Y. A'la, A. Fatmawati, and B. A. Hermansyah, "Comparison of transfer learning method for COVID-19 detection using convolution neural network," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 11, no. 2, pp. 1091–1099, 2022, doi: 10.11591/eei.v11i2.3525.
[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, vol. 86, no. 11, pp. 2278–2324, doi: 10.1109/5.726791.
[24] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.
[25] M. Maimaiti, Y. Liu, H. Luan, and M. Sun, "Multi-round transfer learning for low-resource NMT using multiple high-resource languages," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 4, pp. 1–26, 2019, doi: 10.1145/3314945.

[26] S. Karen and Z. Andrew, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, p. 14.
[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
[30] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, “An ensemble of fine-tuned convolutional neural networks for medical image classification,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 31–40, 2017, doi: 10.1109/JBHI.2016.2635663.
[31] M. F. Abdelfattah, M. W. Fakhr, and M. A. Rizka, “ArSentBERT: fine-tuned bidirectional encoder representations from transformers model for Arabic sentiment classification,” *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 12, no. 2, pp. 1196–1202, 2023, doi: 10.11591/eei.v12i2.3914.

BIOGRAPHIES OF AUTHORS

Hicham Lamtougui received a master’s degree in computer science from the Faculty of Science Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco. where he is currently pursuing a Ph.D. degree. His research interests include machine learning, deep learning, Arabic handwriting recognition, and text mining. He can be contacted at email: hicham.lamtougui@usmba.ac.ma.

Hicham El Moubtahij is currently a Professor of computer science at the University of Ibn Zohr, Agadir, Morocco. He received his Ph.D. in computer science from the University of Sidi Mohamed Ben Abdellah, Fez, Morocco in 2017. He is now a member of the Systems and Technologies of Information Team at the High School of Technology at the Ibn Zohr University, Agadir. His current research interests include machine learning, deep learning, Arabic handwriting recognition, text mining, and medical imagery. He can be contacted at email: h.elmoubtahij@uiz.ac.ma.

Hassan Fouadi received a master’s degree in computer science from the High School of Technology at the University Sidi Mohamed Ben Abdellah, Fez, Morocco. Where he is currently pursuing a Ph.D. degree. His research interests include sentiment analysis, machine learning, deep learning, and natural language processing. He can be contacted at email: Hassan.fouadi@usmba.ac.ma.

Khalid Satori received the Ph.D. degree from the National Institute for the Applied Sciences INSA at Lyon, in 1993. He is currently a Professor in computer science with Sidi Mohamed Ben Abdellah University, Fez. He is also the director of the LISAC Laboratory. His research interests include real-time rendering, image-based rendering, virtual reality, biomedical signal, camera self-calibration, genetic algorithms, and 3D reconstruction. He can be contacted at email: khalid.satori@usmba.ac.ma.