UDC 004.912.032.26(64) **doi: 10.32620/reks.2024.1.02**

Ali BENAISSA[1,3], Abdelkhalak BAHRI[1], Ahmad EL ALLAOUI[2], My Abdelouahab SALAHDDINE[3]

[1] ***Science and Competitive Intelligence Team, Applied Sciences Laboratory, The National School of Applied Science, Abdelmalek Essaadi University, Al-Hoceima, Morocco***

[2] ***Decisional Computing and Systems Modelling Team, Engineering Sciences and Techniques Laboratory, Faculty of Sciences and Techniques, Moulay Ismail University of Meknes, Errachidia, Morocco***

[3] ***Governance and Performance of Organizations, The National School of Business and Management, Abdelmalek Essaadi University, Tangier, Morocco***

ADVANCED APPROACH FOR MOROCCAN ADMINISTRATIVE DOCUMENTS DIGITIZATION USING PRE-TRAINED MODELS CNN-BASED: CHARACTER RECOGNITION

In the digital age, efficient digitization of administrative documents is a real challenge, particularly for languages with complex scripts such as those used in Moroccan documents. The subject matter of this article is the digitization of Moroccan administrative documents using pre-trained convolutional neural networks (CNNs) for advanced character recognition. This research aims to address the unique challenges of accurately digitizing various Moroccan scripts and layouts, which are crucial in the digital transformation of administrative processes. Our goal was to develop an efficient and highly accurate character recognition system specifically tailored for Moroccan administrative texts. The tasks involved comprehensive analysis and customization of pre-trained CNN models and rigorous performance testing against a diverse dataset of Moroccan administrative documents. The methodology entailed a detailed evaluation of different CNN architectures trained on a dataset representative of various types of characters used in Moroccan administrative documents. This ensured the adaptability of the models to real-world scenarios, with a focus on accuracy and efficiency in character recognition. The results were remarkable. DenseNet121 achieved a 95.78% accuracy rate on the Alphabet dataset, whereas VGG16 recorded a 99.24% accuracy on the Digits dataset. DenseNet169 demonstrated 94.00% accuracy on the Arabic dataset, 99.9% accuracy on the Tifinagh dataset, and 96.24% accuracy on the French Special Characters dataset. Furthermore, DenseNet169 attained 99.14% accuracy on the Symbols dataset. In addition, ResNet50 achieved 99.90% accuracy on the Character Type dataset, enabling accurate determination of the dataset to which a character belongs. In conclusion, this study signifies a substantial advancement in the field of Moroccan administrative document digitization. The CNN-based approach showcased in this study significantly outperforms traditional character recognition methods. These findings not only contribute to the digital processing and management of documents but also open new avenues for future research in adapting this technology to other languages and document types.

Keywords: *Character Recognition; Pre-trained Models; Convolutional Neural Networks (CNNs); Moroccan Official Documents; Digital Transformation.*

Introduction

Advancements in machine learning [1] have significantly improved the field of optical character recognition [2], which is crucial for document digitization. These advancements [3] have led to the development of more accurate character recognition systems [4] and innovative methods such as image classification [5, 6], which have proven effective in recognizing characters from various sources, including different fonts and handwriting styles. Object detection, a key component of pattern recognition in computer vision [7], has seen widespread application across sectors such as education [8], medicine [9], video surveillance [10], entertainment [11], and security [12]. The Moroccan government has recently begun trying to digitize its public administration under Law No. 55.19, which the Ministry of Economy, Finance, and

Administration Reform issued on March 19, 2020. This move is crucial because many older archives and new documents are still in paper form, often in multiple languages such as French, Arabic, and Amazigh. The challenge lies in efficiently processing these multilingual documents to enhance Morocco's public administration system. Recent studies have contributed to addressing these challenges through innovative OCR technologies. Mohammad Anwarul Islam and Ionut E. Iacob [13] delves into the OCR of historical manuscripts, using the Beowulf manuscript as a case study. Their research demonstrates the effectiveness of combining machine learning and deep learning techniques, particularly convolutional neural networks (CNNs), for character recognition in documents with unique challenges such as cursive writing and historical wear. This approach underscores the importance of adaptable OCR systems capable of handling the diversity of scripts found in Moroccan administrative documents, aligning with the need for robust solutions that accommodate script diversity and document quality variations. Furthermore, Another study by Yiyi Liu, Yuxin Wang, and Hongjian Shi [14] proposed a novel OCR model that integrates CRNN with Differentiable Binarization (DBNet) and incorporates a text direction classifier and the Retinex algorithm for image enhancement. This model significantly improves the detection and recognition of text within complex scenes, including multi-oriented texts, offering a promising solution for digitizing multilingual documents. Their methodology presents a pathway for enhancing OCR technologies to meet the digitization requirements of the Moroccan government's initiative, offering a promising direction for our research in developing a comprehensive character recognition system for diverse administrative documents. By focusing on these two recent studies, this research benefits from cutting-edge OCR advancements, aiming to develop a system that effectively addresses the challenges presented by the multilingual and multi-font nature of Moroccan administrative documents. This approach contributes to the broader efforts of digital transformation in public administration, highlighting the importance of innovative OCR solutions in the efficient processing and management of multilingual documents. This analysis underscores the gap between existing OCR advancements and the practical requirements of the multilingual document digitization initiative, pointing towards an area for focused research and technological adaptation.

Research is motivated by the need to improve the digitization of administrative documents on a global scale, addressing the substantial challenge posed by the diversity of scripts and languages. Focusing on the intricacies of converting paper-based records into digital formats, this study tackles the limitations of current optical character recognition (OCR) technologies. By focusing on pre-trained CNN models, we endeavor to improve the accuracy and efficiency of processing multilingual administrative documents. This effort is rooted in the understanding that effective digitization is essential not only for streamlining administrative operations worldwide but also for ensuring equitable access to and preservation of vital government records. The motivation lies in leveraging technological innovations to support the global digital transformation ambitions of administrations, thereby facilitating a more connected and efficiently managed digital future.

The objective of this research is to automate the recognition and extraction of content from these multilingual Moroccan documents using advanced digitization and character recognition techniques, with a measurable target to significantly enhance character recognition accuracy beyond current standards. This automation is intended to streamline administrative procedures, save time and resources, and increase the overall effectiveness of Morocco's public administration.

The approach involves utilizing deep learning with pre-trained models based on CNNs [15] from the TensorFlow Keras Library [16] and applying transfer learning techniques [13], which are known for their effectiveness in classification problems. To support this research, we constructed a new dataset including various characters from the languages used in official Moroccan documents and selected 11 models for evaluation, such as MobileNet, ResNets, DenseNets, and VGGNets. We will evaluate the effectiveness of each model using metrics such as the number of parameters, accuracy validation, recall validation, F1-score validation, and predict validation. This study identifies the most effective model for different datasets and present a comparative analysis between the presented approach and existing techniques.

1. State of the art

In the past few years, many approaches and methods of deep learning for character identification and recognition have been proposed. To apply these approaches and methods, numerous datasets have been built for different languages, such as Arabic, Alphabet, Tifinagh, and Digits.

CNN has made a breakthrough by introducing advanced performance in character recognition.

Hinton et al. [17] investigated Deep Belief Networks (DBN), which have three layers and an assimilation algorithm, and scored an accuracy of 98.75% for the MNIST dataset.

Using the dropout regulation method for unrestricted handwriting recognition, Pham et al. [18] made recurrent neural networks (RNNs) work better by lowering the number of word and character errors.

A portion of the approach described in the paper [19] had already been initiated: a custom dataset with five categories was created, each representing a group of characters in the Moroccan language, to identify characters in Moroccan documents. The custom dataset was trained using pre-trained models, achieving a 98% accuracy score with the DenseNet201 model. A modified pre-trained model based on the DenseNet201 model was proposed, which achieved a 99% accuracy score.

Mohamed N AlJarrah et al. [20] proposed a model based on CNN to recognize Arabic handwritten characters. The model used in their experiment worked on a dataset containing 16800 images of Arabic handwritten characters in different shapes and forms. Their proposed model achieved a 97.2% accuracy rate without data augmentation; when they also used data augmentation, the accuracy increased to 97.7%.

Tapan Kumar Hazra et al. [21] presented an application of pattern recognition using KNN to recognized handwritten or printed text with some advantages. One of these advantages is that it works well with multimodal classes because its inference is based on comparison, regardless of whether the target class is multimodal or not. This technique always leads to high accuracy.

Duddela Sai Prashanth et al. [22] produced a dataset of 38,750 images of Davangari digits, and experiments were conducted by applying three networks based on CNN to their proposed dataset. Using the proposed CNN architecture, they achieved a recognition rate of 99% on training data and 94% on evaluation data. Using the proposed CNN architecture, a modified LeNet achieved an accuracy rate of 99% and 98% with less computational cost. Finally, AlexNet achieved a recognition rate of 99% on training data and 98% on evaluation data. They performed a series of experiments using a different split of data, and the outcome was that they found a minimal loss of 0.001%.

Waleed Albattah et al. [23] created several deep-learning and hybrid models. Using two datasets in the experiments, the transfer-learning model on the Arabic MNIST digit dataset achieved 99.67% accuracy, and the hybrid models achieved an accuracy between 93.88% and 97.10%. The results of the hybrid models on the Arabic MNIST dataset were poor, not exceeding 87.2% accuracy.

This state-of-the-art review encapsulates the dynamic and rapidly evolving field of character recognition, highlighting the significant strides made through the adoption of CNNs and other deep learning architectures. The continuous development of tailored datasets and innovative model architectures promises further advancements, pushing the boundaries of what is achievable in character recognition and offering new opportunities for practical applications in document digitization and beyond.

2. Problem statement

Take Morocco as an example to show how this method is used in practice. In Morocco, official documents are often multilingual, featuring content in Arabic (the primary official language), French (the secondary official language), and Tamazight (Tifinagh as the written form of the language), which was recognized as an official language in a 2011 constitutional reform. In addition, recent news suggests that English may soon become another official language in the country. Despite the presence of multiple languages, the aim is to recognize all of them, even when they appear within the same document, using the proposed approach. However, recognition systems such as OCR struggle to identify these languages because of the similarities between their written forms. Table 1 presents similarities between characters across different scripts used in Moroccan administrative documents, specifically highlighting the resemblance among characters in Alphabet, Digits, Arabic, and Tifinagh scripts. This comparison illustrates the potential challenges faced by optical character recognition (OCR) systems because of the similarities between characters in different scripts. Examples are listed where characters from the Alphabet script resemble those in other scripts, such as Arabic and Tifinagh, potentially complicating the OCR process for multilingual documents containing these scripts. This insight underscores the complexity of accurately digitizing Moroccan administrative documents that feature multiple languages and scripts, motivating the need for the advanced approach discussed in the paper to improve character recognition accuracy using pre-trained CNN models.

Table 1

Similarities between the characters

Alphabet	**Digits**	**Arabic**	**Tifinagh**
Q	9	هـ	ⵕ
Y	-	-	ⵖ
E	-	-	ⴹ and ⵟ
A	-	-	ⵍ
O and o	0	ه	ⵔ and ∘
l	1	ا and إ	ⵉ

To better understand why recognition systems may struggle with multilingual documents, it is necessary to look at how CNNs work. In this type of machine learning, a model is trained on a dataset that contains various classes or categories. These classes can represent different languages, letter forms, or other characteristics of the data. However, as the number of categories increases, the model's performance can degrade significantly. In a

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

study by Chao Luo et al. [24], it was found that performance begins to decrease significantly when there are eight categories. Given that there are more than 153 categories in the built dataset, one can imagine how CNN models can give disastrous performances both in terms of accuracy and precision. Therefore, in this research, a hierarchical approach is proposed that uses pre-trained CNN models to identify and recognize Moroccan characters smoothly, considering the challenges posed by multiple languages, various letter forms, and similarities between them.

3. Proposed system and datasets

3.1. Proposed system

This is an overview of the system, as shown in

Fig. 1. The system is designed to perform two tasks: character type identification and character recognition, which are described below.

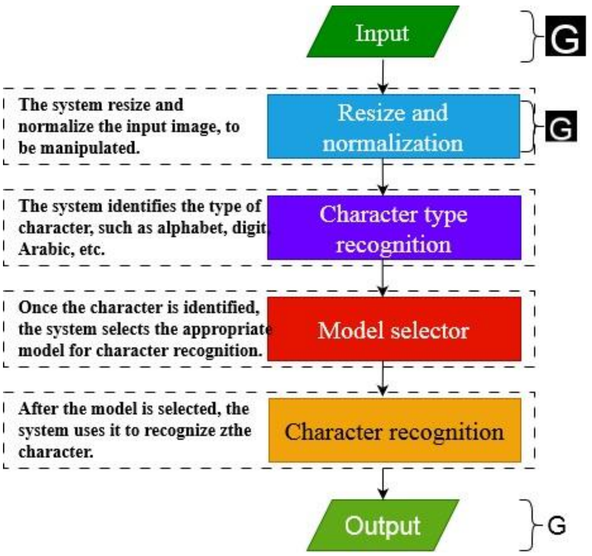


Fig. 1. Proposed Character Recognition System

The proposed character recognition system consists of four main steps to accurately recognize different types of characters from the input images:

1. Input Image: The first step in the process involves providing an input image containing a character to the system. The system then identifies the type of character in the input image, which can be alphabet, Arabic, digit, or other types of characters. This identification is crucial for selecting a suitable pre-trained model that can accurately recognize the character.

2. Model Selection: after character type identification, the system employs a meticulous model selection procedure from a curated set of pre-trained models. The identified character type, ensuring the use of an appropriate model tailored to recognize the specific character category present within the input image, drives this selection process.

3. Character Recognition: Following meticulous pre-trained model selection, the system leverages the chosen model to perform character recognition within the input image. By employing a range of techniques, including machine learning, computer vision, and pattern recognition, the selected model processes the input image to recognize the specific character contained within it.

4. Result Output: Finally, the system outputs the result of the character recognition process, which is the recognized character in digital format.

3.2. Datasets

This section provides an overview of the datasets used in this study, which encompass characters from various languages used in Moroccan official documents, including Arabic, French, and Tamazight. The dataset was developed to facilitate research and analysis in this domain. The previously published datasets [25] provide a comprehensive collection of characters, presenting valuable resources for further research. The datasets cover various characters, including lowercase and uppercase letters, numbers, special characters, and symbols. These datasets were meticulously curated to ensure accuracy and representativeness. By utilizing a diverse selection of commonly used fonts, the aim is to capture the richness and diversity of character forms. The resulting dataset provides a reliable foundation for further investigations in this field.

3.3. Datasets parameters

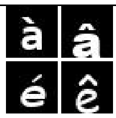
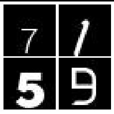
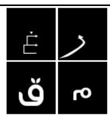
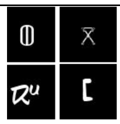
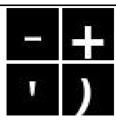
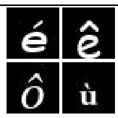
To proceed, a discussion of the dataset parameters is necessary. In this study, each dataset is partitioned into a train set and an evaluation set, following an 80% and 20% split, respectively. Within the training set, a subset comprising 20% of its samples is designated as the test set. It is crucial to highlight that the evaluation set is exclusively used for evaluation. Consequently, during the training phase, the model solely uses the training set and the designated test set, ensuring the integrity of the training process and accurate assessment of the model's performance.

As an example, the Alphabet dataset contains 52 categories with 91505 total images; these images comprise the training set (17242), test set (4311), and evaluation set (5385). The same is true for the other datasets, with a different number of pure data, as

Table 2 details the other datasets.

Table 2

Datasets parameters

N	Dataset	Samples	Categories number	Pure data	Train set	Test set	Evaluation set
1	Alphabet		52	91505	17242	4311	5385
2	Digits		10	64567	6816	1705	2127
3	Arabic		28	66164	42358	10590	13216
4	Tifinagh		33	6930	4435	1109	1386
5	French Special Characters		16	25600	16384	4096	5120
6	Symbols		14	22400	14336	3584	4480
7	Character Type		6	205758	131685	32922	41151
8	All-Classes		135	203921	130535	32634	40752

Two additional datasets were built: the Character Type Dataset and the All-Classes Dataset. The Character Type Dataset encompasses the combined data from the Alphabet, Digits, Arabic, Tifinagh, French Special Characters, and Symbols datasets. It served the purpose of determining the category to which a specific character belonged. The All-Classes Dataset incorporated all categories from the Alphabet, Digits, Arabic, Tifinagh, French Special Characters, and Symbols datasets. Its purpose was to provide a comparative basis during the experiment, allowing for exploration of the challenges associated with character recognition when dealing with several categories.

4. Experiments setup

4.1. Models training

A system was developed that is capable of loading local datasets and training Keras models on them. The system's flowchart is shown in Fig. 2. The system's running process can be divided into six parts.

In the first part, the system is initiated, and in the second part, a local dataset is loaded and then split into training, validation, and testing sets. The images in the dataset were resized to the given dimensions, normalized, and categorized. In the third part, the system loops over each available Keras model, downloads it, and trains it.

If there are any other models that have yet to be trained, the system trains them until all suitable models are trained. In the fourth part, the system evaluates the models and saves their weights. In the fifth part, the system lists the results of the training and evaluation process. Finally, in the sixth part, the system ends its execution.

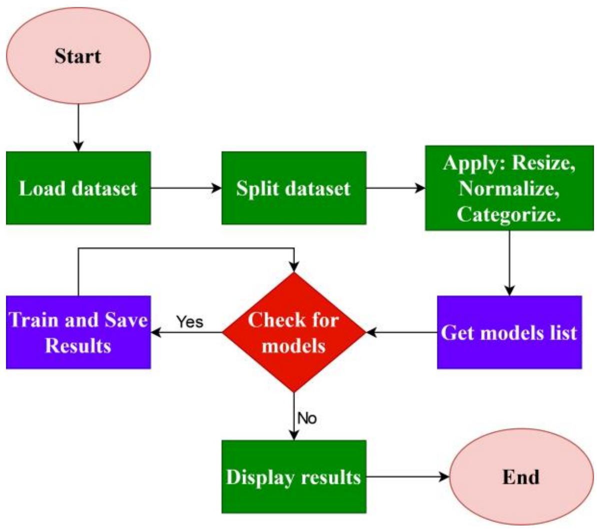


Fig. 2. Model training flowcharts

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

4.2. Hyper-parameters

Began with a pure dataset, which was then divided into training and evaluation sets. Provided both directory paths to the scripts and ensured that the training and evaluation sets contained different images. Selected the test dataset from the training set, which accounted for 20% of the data. Finally, we evaluated the pre-trained models on the evaluation dataset, which the models had not seen before.

To compile the models, used the 'Adam' optimizer [26, 27] with a default learning rate of 0.001 to minimize the error function. We applied the categorical cross-entropy function as the loss or error function and measured the performance using accuracy (1), precision (2), recall (3), and F1-score (4) metrics:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

$$\text{precision} = \frac{TP}{TP + FP}, \quad (2)$$

$$\text{recall} = \frac{TP}{TP + FN}, \quad (3)$$

$$F1 - \text{score} = \frac{2 * (\text{recall} * \text{precision})}{\text{recall} + \text{precision}}, \quad (4)$$

where TP – True Positives, TN – True Negatives, FP – False Positives, FN – False Negatives.

During the training process, validation accuracy was monitored, and if it did not improve for two consecutive epochs, the learning rate was reduced. The learning rate is updated by multiplying the default learning rate (0.001) by a default factor (0.01). Furthermore, to prevent overfitting and optimize accuracy, an early stopping mechanism was employed that stops the model fitting process if there is no improvement in the accuracy metric. This criterion serves as a reliable indicator to ensure that the model's training terminates at an optimal point, avoiding unnecessary computational efforts.

5. Theory validation

To validate the theory, the dataset categories were collected in one directory, forming a single dataset. The same hyper parameters as mentioned earlier were used and the dataset was trained on different pre-trained models. The results are presented in Fig. 3 and Table 3.

Based on Fig. 3, we observe that the pre-trained models perform poorly. The models with the highest accuracy measures are DenseNet169 and DenseNet201, with accuracy rates of 89.34% and 89.27%, respectively. Finally, MobileNetV2 is at the bottom of the list with the weakest performance, achieving only 36.80% validation accuracy.

.

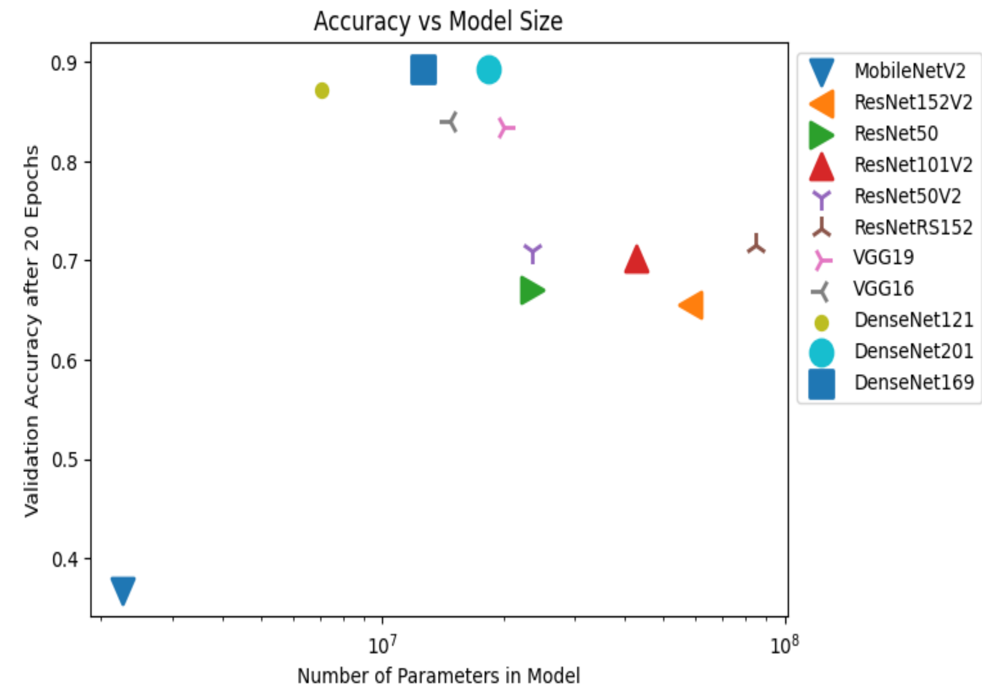


Fig. 3. Pre-trained models’ accuracy against the parameters number on Al l Classes dataset

Table 3

Models' performance results

Model name	Parameters number	Validation Accuracy	Validation recall	Validation F1-score	Validation precision
DenseNet169	12642880	89.34%	87.25%	91.46%	96.28%
DenseNet201	18321984	89.27%	86.99%	91.29%	96.21%
DenseNet121	7037504	87.17%	84.63%	89.33%	94.78%
VGG16	14714688	84.07%	78.08%	85.63%	95.16%
VGG19	20024384	83.37%	76.56%	84.53%	94.75%
ResNetRS152	84724256	71.50%	62.45%	72.48%	87.01%
ResNet50V2	23564800	71.04%	60.36%	71.15%	87.38%
ResNet101V2	42626560	70.19%	58.30%	69.84%	87.93%
ResNet50	23587712	67.00%	56.12%	67.87%	86.70%
ResNet152V2	58331648	65.50%	50.81%	63.76%	86.75%
MobileNetV2	2257984	36.80%	19.05%	30.31%	79.98%

Based on Table 3, it is evident that DenseNet169 achieved the highest accuracy, precision, recall, and F1-score of 89.34%, 87.25%, 91.46%, and 96.28%, respectively. These results were achieved because of the large parameter numbers, as presented in Fig. 3. On the other hand, MobileNetV2 has the smallest number of parameters and achieves the lowest accuracy, precision, recall, and F1-score of 36.80%, 19.05%, 30.31%, and 79.98%, respectively. However, despite the large parameter numbers of pre-trained models, their performance on the All-Classes dataset was poor because the results achieved by DenseNet169 were not sufficient to accurately identify characters. Furthermore, we applied the method proposed by Waleed Albattah [23] to the All-Classes dataset, which is a combination of CNN and SVM. The result was poor, with an accuracy rate of 0.6%. To enhance character recognition, we will use a hierarchical structure in the upcoming experiments, replacing the current flat structure used in this experiment (a single dataset with all classes). Each language part will be used as a separate dataset to build a hierarchical structure, as shown in Fig. 4.

6. Results

This section presents the approach, in which we embark upon the selection of the most suitable model for a given dataset. We conducted a comprehensive analysis, comparing multiple models in terms of their accuracy and parameter count. Through a thorough examination of the outcomes, we identify the model that attains the highest accuracy while simultaneously ensuring reasonable computational efficiency. This endeavor seeks to strike an optimal equilibrium between accuracy and the allocation of computational resources. The findings, which delineate the performance evaluation of the pre-trained models, are expounded upon in the ensuing discussion. Furthermore, we contextualize the results by juxtaposing them with recent advancements in the field of character recognition.

6.1. Alphabet

The performance of the pre-trained models on the Alphabet dataset is presented in Fig. 5. DenseNets and VGGs achieved the best performance followed by ResNets with medium performance, and MobileNetV2 with the worst results.

Table 4 shows the performance of each model, including validation accuracy, recall, F1-score, and precision. DenseNet201 had the highest performance, with a validation accuracy of 95.78%. DenseNet121 and DenseNet169 followed, with validation accuracy of 95.00% and 94.64%, respectively. VGG16 and VGG19 achieved

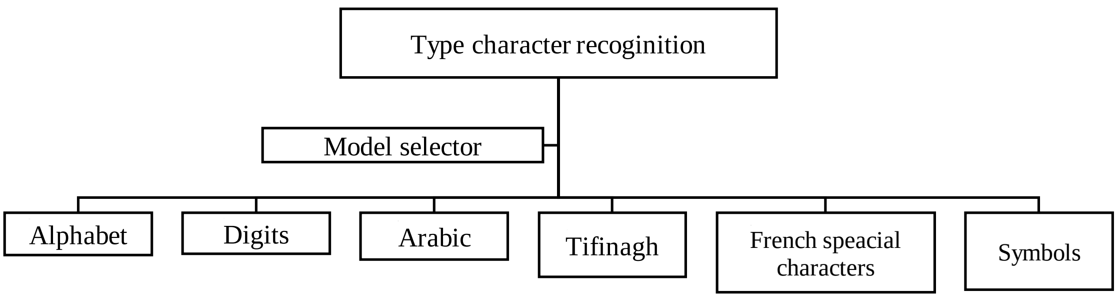


Fig. 4. Hierarchical approach

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

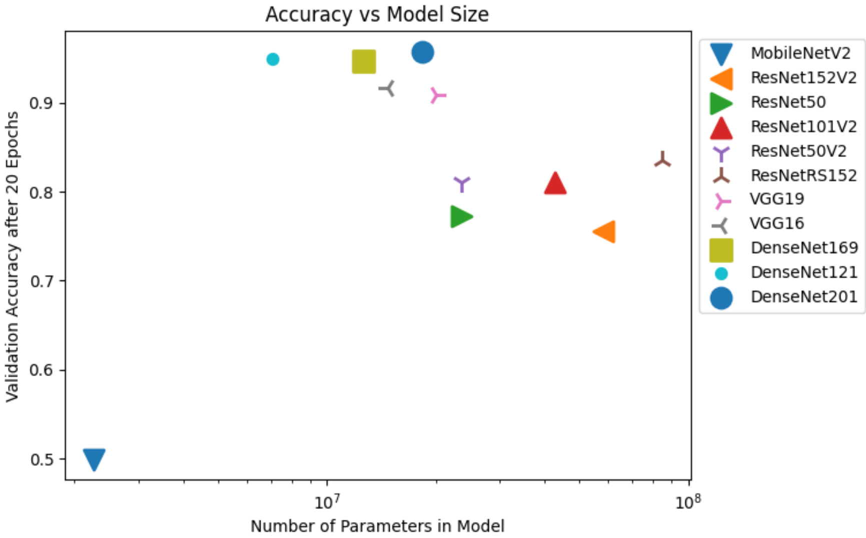


Fig. 5. Pre-trained models' accuracy against the parameters number on the alphabet dataset

Table 4

Pre-trained models' performance on the alphabet dataset

Model name	Parameters number	Validation Accuracy	Validation recall	Validation F1-score	Validation precision
DenseNet201	18321984	95.78%	94.10%	95.62%	97.24%
DenseNet121	7037504	95.00%	93.14%	94.99%	96.98%
DenseNet169	12642880	94.64%	92.39%	94.53%	96.85%
VGG16	14714688	91.71%	87.86%	91.39%	95.33%
VGG19	20024384	90.84%	86.88%	90.74%	95.09%
ResNetRS152	84724256	83.61%	75.59%	82.70%	91.64%
ResNet50V2	23564800	81.09%	71.41%	79.47%	90.03%
ResNet101V2	42626560	81.02%	72.62%	80.28%	90.14%
ResNet50	23587712	77.20%	61.25%	72.94%	91.05%
ResNet152V2	58331648	75.52%	63.51%	73.34%	87.32%
MobileNetV2	2257984	49.92%	27.30%	40.58%	83.35%

validation accuracy of 91.71% and 90.84%, respectively. ResNets had validation accuracy ranging from 83.61% to 75.52%. Finally, MobileNetV2 had the lowest validation accuracy at 49.92%.

To determine the optimal model for the task of recognizing the alphabet based on the Alphabet dataset, we will compare the performance of the top two models. After analyzing the accuracy and number of parameters for each model, it is evident that DenseNet121 is the superior choice. This is because DenseNet121 achieved a high level of accuracy while also having a relatively low number of parameters, which can lead to improved computational efficiency and reduced overfitting. Therefore, we can conclude that DenseNet121 is the optimal model for the alphabet recognition task.

6.2. Digits

Fig. 6 displays the results of the pre-trained models on the Digits dataset. VGG16, DenseNet201, and ResNet50 achieved the highest performance on this dataset, whereas VGG19, DenseNet121, ResNetRS152, and DenseNet169 showed moderate performance. On the other hand, MobileNetV2 achieved the poorest results compared with the other models. Overall, the analysis of these results highlights the superiority of the VGG16, DenseNet201, and ResNet50 models for the task of digit recognition on this particular dataset.

Table 5 presents the performance results of the different pre-trained models on the Digits dataset. The models were evaluated based on four metrics: validation accuracy, validation recall, validation F1-score, and validation precision. Among the tested models, VGG16,

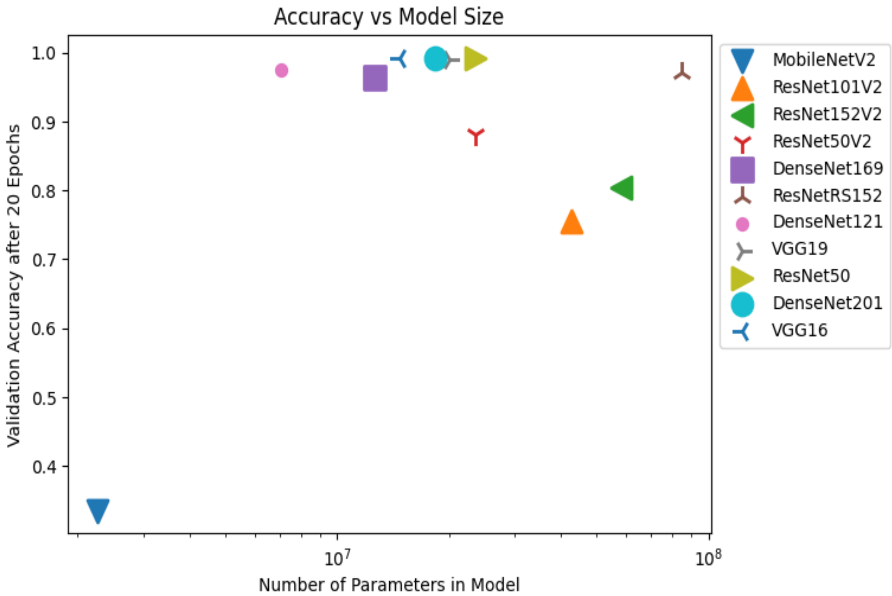


Fig. 6. Pre-trained models' accuracy against the parameters number on the digits dataset

Table 5

Pre-trained models' performance on the digits dataset

Model name	Parameters number	Validation Accuracy	Validation recall	Validation F1-score	Validation precision
VGG16	14714688	99.24%	99.14%	99.19%	99.25%
DenseNet201	18321984	99.21%	99.17%	99.35%	99.53%
ResNet50	23587712	99.14%	99.10%	99.26%	99.43%
VGG19	20024384	99.03%	99.03%	99.06%	99.10%
DenseNet121	7037504	97.48%	97.31%	97.41%	97.51%
ResNetRS152	84724256	97.05%	95.97%	97.09%	98.29%
DenseNet169	12642880	96.33%	96.23%	96.42%	96.61%
ResNet50V2	23564800	88.08%	88.07%	88.07%	88.07%
ResNet152V2	58331648	80.44%	80.39%	80.43%	80.47%
ResNet101V2	42626560	75.50%	75.51%	75.53%	75.54%
MobileNetV2	2257984	33.57%	5.88%	10.61%	68.35%

DenseNet201, ResNet50, and VGG19 achieved the highest validation accuracy with 99.24%, 99.21%, 99.14%, and 99.03%, respectively. DenseNet121, ResNetRS152, and DenseNet169 followed with 97.48%, 97.05%, and 96.33% validation accuracy, respectively. ResNet50V2, ResNet152V2, and ResNet101V2 achieved validation accuracy values of 88.08%, 80.44%, and 75.50%, respectively. MobileNetV2 had the lowest performance, with a validation accuracy of 33.57%.

Therefore, VGG16, DenseNet201, ResNet50, and VGG19 are the top-performing models, whereas MobileNetV2 is the worst performing model.

We found that VGG16 outperforms DenseNet201 in terms of validation accuracy, achieving the highest accuracy with the fewest parameters. VGG16's high accuracy on the validation set indicates that it can generalize well and make accurate predictions on unseen data. Furthermore, VGG16's lower number of parameters compared with DenseNet201 makes it a more computationally efficient model. Therefore, based on these findings, we can confidently conclude that VGG16 is the best model for recognizing digits.

6.3. Arabic

The performance of the pre-trained models on the Arabic dataset is depicted in Fig. 7. The results indicate that ResNet50, DenseNet169, DenseNet201, and VGG19 are more effective than DenseNet121, VGG16, and ResNetRS152, whereas MobileNetV2 outperforms DenseNet121, VGG16, and ResNetRS152.

Table 6 displays a detailed breakdown of the performance results for each model, including their validation accuracy, recall, F1-score, and precision. Among the

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

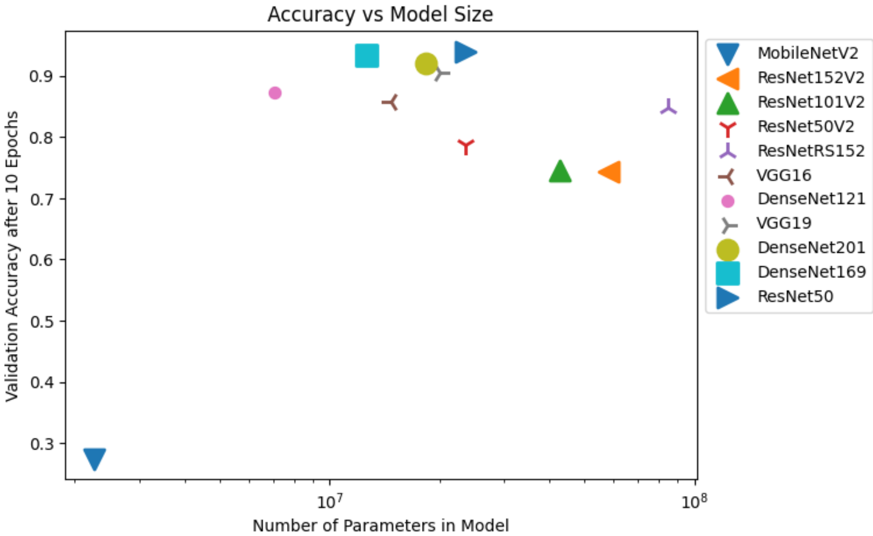


Fig. 7. Pre-trained models' accuracy against the parameters number on Arabic dataset

Table 6

Pre-trained models' performance on Arabic dataset

Model name	Parameters number	Validation Accuracy	Validation recall	Validation F1-score	Validation precision
ResNet50	23587712	94.00%	92.01%	93.83%	95.78%
DenseNet169	12642880	93.34%	92.36%	93.61%	94.94%
DenseNet201	18321984	92.10%	89.15%	91.92%	94.96%
VGG19	20024384	90.58%	88.93%	90.68%	92.56%
DenseNet121	7037504	87.18%	85.78%	87.62%	89.67%
VGG16	14714688	85.73%	81.37%	85.89%	91.16%
ResNetRS152	84724256	84.78%	76.13%	83.47%	92.77%
ResNet50V2	23564800	78.77%	78.08%	78.79%	79.53%
ResNet101V2	42626560	74.56%	74.51%	74.81%	75.11%
ResNet152V2	58331648	74.35%	73.40%	74.57%	75.89%
MobileNetV2	2257984	27.46%	4.88%	8.88%	60.73%

models tested on the Arabic dataset, ResNet50, DenseNet169, DenseNet201, and VGG19 achieved the highest validation accuracy of 94%, 93.34%, 92.10%, and 90.58%, respectively. The middle performers were DenseNet121, VGG16, and ResNetRS152, with validation accuracy of 87.18%, 85.72%, and 84.77%, respectively. In addition, ResNet50V2, ResNet101V2, and ResNet152V2 obtained validation accuracy values of 78.77%, 74.34%, and 74.34%, respectively. MobileNetV2 had the weakest performance, with a validation accuracy of 27.45%. ResNet152V2 obtained a validation accuracy of 78.77%, 74.34%, and 74.34%, respectively. MobileNetV2 had the weakest performance, with a validation accuracy of 27.45%.

To choose the optimal model for recognizing Arabic characters, we compare the top two performing models, ResNet50 and DenseNet169. By evaluating the models based on their validation accuracy and parameter count, we can confidently state that DenseNet169 is the superior model. With a higher validation accuracy and a lower number of parameters, DenseNet169 demonstrates better performance and efficiency than ResNet50. Therefore, we selected DenseNet169 to recognize Arabic characters.

6.4. Tifinagh

The performance of the pre-trained models on the Tifinagh dataset is shown in Fig. 8. DenseNet201, DenseNet169, ResNet50, ResNet50V2, and VGG19 achieved the best performance, whereas VGG16 and ResNetRS152 achieved good performance. Medium performance was achieved by ResNet152V2 and ResNet101V2, and DenseNet121 also showed decent performance. However, MobileNetV2 achieved the worst performance among the models evaluated.

Table 7 provides a detailed summary of the performance results of each model on the Tifinagh dataset, including validation accuracy, validation recall, validation

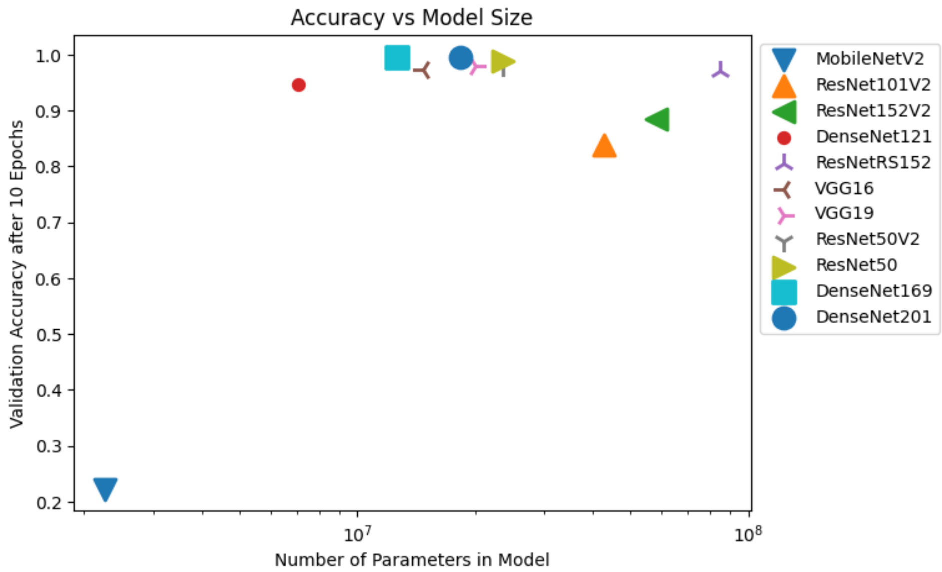


Fig. 8. Pre-trained models' overall accuracy against the parameters number on Tifinagh dataset

Table 7

Pre-trained models' performance on Tifinagh dataset

Model name	Parameters number	Validation Accuracy	Validation recall	Validation F1-score	Validation precision
DenseNet121	7037504	99.91%	99.55%	99.73%	99.91%
DenseNet201	18321984	99.82%	99.82%	99.82%	99.82%
DenseNet169	12642880	99.64%	99.64%	99.73%	99.82%
ResNet50	23587712	98.83%	98.21%	98.87%	99.55%
VGG19	20024384	98.02%	97.90%	98.20%	98.52%
VGG16	14714688	97.66%	96.61%	97.85%	99.18%
ResNetRS152	84724256	97.57%	95.45%	97.25%	99.17%
ResNet50V2	23564800	96.30%	95.94%	96.27%	96.62%
ResNet101V2	42626560	90.35%	90.13%	90.77%	91.44%
ResNet152V2	58331648	61.14%	60.38%	61.25%	62.16%
MobileNetV2	2257984	22.00%	1.12%	2.16%	32.86%

F1-score, and validation precision. Among the pre-trained models evaluated, DenseNet121, DenseNet201, DenseNet169, ResNet50, and VGG19 exhibited the highest performance with validation accuracies of 99.91%, 99.82%, 99.64%, 98.83%, and 98.02%, respectively. Following closely were VGG16, ResNetRS152, and ResNet50V2, which achieved validation accuracies of 97.66%, 97.57%, and 96.30%, respectively.

ResNet101V2 also achieved good performance, with a validation accuracy of 90.35%. At the bottom of the list were ResNet152V2 and MobileNetV2, which achieved the worst results with validation accuracies of 61.14% and 22.27%, respectively.

To choose the best model, we compare the top two performing models in terms of validation accuracy and parameter count. Based on these criteria, DenseNet121 emerges as the best model.

6.5. French Special Characters

The performance of the pre-trained models on the French Special Characters dataset is presented in Fig. 9, DenseNet169, DenseNet201, and DenseNet121, along with VGG16, VGG19, ResNet50V2, and ResNet101V2, achieved the highest performance. Models such as ResNetRS152, ResNet152V2, and ResNet50 achieved moderate performance, whereas MobileNetV2 had the worst performance.

Table 8 provides a detailed comparison of the performance results for each model on the French Special Characters dataset. The Table 8 reports the validation accuracy, validation recall, validation F1-score, and validation precision metrics for each model. Among the models, DenseNet169, DenseNet201, and DenseNet121

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

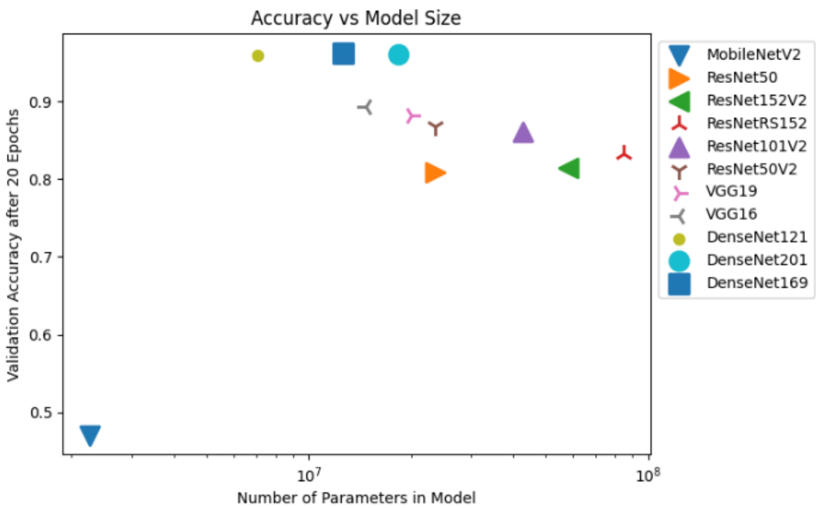


Fig. 9. Pre-trained models' accuracy against the parameters number on French Special Characters dataset

Table 8

Pre-trained models' performance on French special characters dataset

Model name	Parameters number	Validation Accuracy	Validation recall	Validation F1-score	Validation precision
DenseNet169	12642880	96.24%	95.12%	96.74%	98.48%
DenseNet201	18321984	96.04%	94.38%	96.37%	98.54%
DenseNet121	7037504	95.95%	94.46%	96.48%	98.67%
VGG16	14714688	89.33%	81.40%	88.38%	97.02%
VGG19	20024384	88.16%	78.61%	86.57%	96.76%
ResNet50V2	23564800	86.82%	79.79%	85.98%	93.52%
ResNet101V2	42626560	86.01%	77.39%	84.61%	93.61%
ResNetRS152	84724256	83.33%	73.73%	82.07%	93.03%
ResNet152V2	58331648	81.45%	70.70%	79.95%	92.37%
ResNet50	23587712	80.88%	64.60%	76.41%	94.41%
MobileNetV2	2257984	47.07%	21.34%	33.08%	80.39%

achieved the highest performance with validation accuracy scores of 96.24%, 96.04%, and 95.95%, respectively. The VGG16 and VGG19 models achieved medium performance, with validation accuracy scores of 89.33% and 88.16%, respectively. On the other hand, the ResNet models achieved moderate performance, with ResNet50V2, ResNet101V2, ResNetRS152, and ResNet50 obtaining validation accuracy scores of 86.82%, 86.01%, 83.33%, and 80.88%, respectively.

To select the best model, the comparison will be made between the two highest-performing models based on two criteria: accuracy and parameter count. Based on this comparison, we can conclude that DenseNet169 is the best model. It achieved a validation accuracy of 96.24%, which is slightly higher than DenseNet201's accuracy of 96.04%. In addition, DenseNet169 has fewer parameters than DenseNet201, which makes it computationally more efficient. Therefore, DenseNet169 is the best model for the French Special Characters dataset.

6.6. Symbols

Fig. 10 displays the performance of the pre-trained models on the Symbols dataset. DenseNet169 demonstrated the best performance, followed by ResNet50, VGG19, VGG16, DenseNet121, ResNetRS152, and DenseNet201, respectively. Note that ResNet50V2 and ResNet101V2 achieved good performance. However, MobileNetV2 exhibited poor performance on the dataset.

Table 9 displays the validation accuracy of the pre-trained models on the Symbols dataset. Notably, DenseNet169, VGG19, VGG16, DenseNet121, ResNetRS152, and DenseNet201 achieved the highest performance, attaining validation accuracies of 99.14%, 98.83%, 98.49%, 98.41%, 98.35%, and 98.33%, respectively. ResNet50V2 and ResNet101V2 also demonstrate ed strong results with validation accuracies of 93.50% and 91.55%, while ResNet152V2 achieved an accuracy of 87.50%. In contrast, MobileNetV2 exhibited lower performance, with a validation accuracy of only 50.45%.

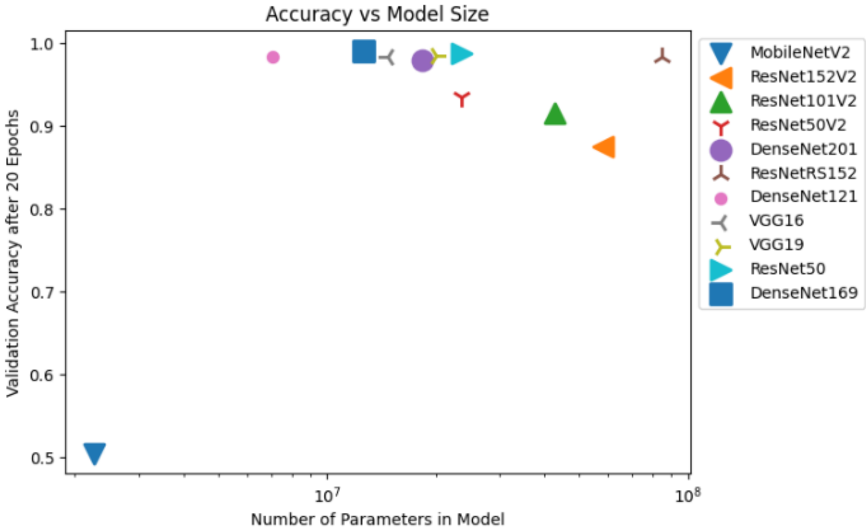


Fig. 10. Pre-trained models' accuracy against the parameters number on Symbols dataset

Table 9

Pre-trained models' performance on Symbols dataset

Model name	Parameters number	Validation Accuracy	Validation recall	Validation F1-score	Validation precision
DenseNet169	12642880	99.14%	99.02%	99.17%	99.33%
ResNet50	23587712	98.83%	98.83%	98.83%	98.83%
VGG19	20024384	98.49%	98.35%	98.49%	98.63%
VGG16	14714688	98.41%	98.19%	98.51%	98.85%
DenseNet121	7037504	98.35%	98.24%	98.36%	98.49%
ResNetRS152	84724256	98.33%	98.21%	98.43%	98.66%
DenseNet201	18321984	97.94%	97.88%	98.05%	98.24%
ResNet50V2	23564800	93.50%	93.42%	93.52%	93.63%
ResNet101V2	42626560	91.55%	91.43%	91.56%	91.69%
ResNet152V2	58331648	87.50%	87.42%	87.55%	87.69%
MobileNetV2	2257984	50.45%	24.58%	36.04%	70.36%

To select the best model, a comparison is needed based on two criteria: accuracy and parameter numbers. The aim is to find a model with the highest accuracy and the fewest parameters. After comparing the performance of all models on the Symbols dataset, it was found that DenseNet169 achieved the highest validation accuracy of 99.14%, making it the best model based on the criteria.

6.7. Character type

In this section, the outcomes obtained from pre-trained models on the Character Type dataset are presented. These results will help in choosing the best model to act as the selector for another specified model.

In Fig. 11, the performance of the pre-trained models on the Character Type dataset is displayed. Notably, ResNet50, DenseNet201, and DenseNet169 exhibited the highest performance, whereas VGG19, ResNetRS152, VGG16, ResNet50V2, ResNet152V2, and ResNet101V2 achieved intermediate performance. MobileNetV2, however, demonstrated the lowest performance among the tested models.

Table 10 presents the detailed performance results of each model in terms of validation accuracy, validation recall, validation F1-score, and validation precision. The top three performers, ResNet50, DenseNet201, and DenseNet169, achieved validation accuracy scores of 99.90%, 92.39%, and 92.29%, respectively. VGG19, ResNetRS152, VGG16, ResNet50V2, ResNet152V2, and ResNet101V2 achieved medium performance, with validation accuracy scores ranging from 89.24% to 84.21%. MobileNetV2 had the weakest performance with a validation accuracy score of 45.92% and was placed at the bottom of the list.

Based on Table 10, ResNet50 was selected as the best model because it achieved the highest validation accuracy, validation recall, validation F1-score, and validation precision.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

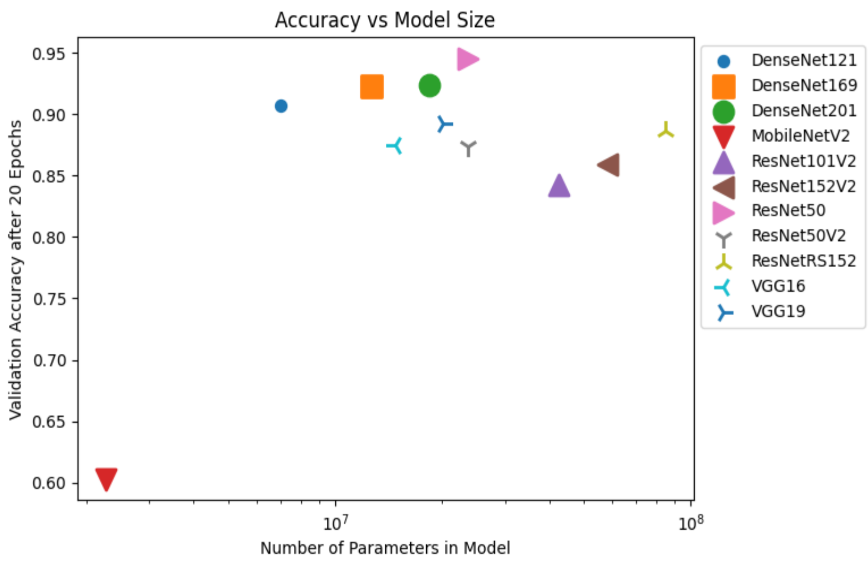


Fig. 11. Pre-trained models' accuracy against the parameters number on the Character Type dataset

Table 10

Pre-trained models' performance on Character Type dataset

Model name	Parameters number	Validation Accuracy	Validation recall	Validation F1-score	Validation precision
ResNet50	23587712	99.57%	98.05%	99.54%	99.05%
DenseNet201	18321984	92.39%	91.62%	92.34%	93.09%
DenseNet169	12642880	92.29%	91.73%	92.38%	93.07%
DenseNet121	7037504	90.71%	89.97%	90.76%	91.59%
VGG19	20024384	89.24%	87.88%	89.28%	90.76%
ResNetRS152	84724256	88.63%	86.79%	88.53%	90.41%
VGG16	14714688	87.47%	85.30%	87.30%	89.48%
ResNet50V2	23564800	87.38%	87.25%	87.41%	87.57%
ResNet152V2	58331648	85.95%	85.74%	85.93%	86.13%
ResNet101V2	42626560	84.21%	84.06%	84.22%	84.39%
MobileNetV2	2257984	60.30%	45.92%	55.39%	70.53%

7. Discussion

Here, we evaluate the models on the test, evaluation datasets, the models already know the first of which, and the second of which is new to them. The results are shown in Table 11. It can be observed that there is no significant difference between the test results and the evaluation results of each model, for example, on the French alphabet, where DenseNet121 has 95.0% validation accuracy on the test dataset and 95.09% validation accuracy on the evaluation dataset.

The same thing is true for the Symbols Dataset; we see that for DenseNet169, while we have 99.14% of validation accuracy on Test and 98.73% of validation accuracy on Evaluation, and so on.

Researchers from all over the world have produced a great deal of work on character recognition. In the current research, we present a system that includes many models for character recognition, and each model achieves very improved results. Recent works have shown how researchers process different models and datasets. A summary is presented in Table 12, which shows the models and different datasets used for character recognition. Different types of datasets, including Arabic MNIST digits, Arabic MNIST characters, AHCD, Hijja, EMNIST, a proposed dataset by D.S. Prashanth, and MNIST Alphabet, have been used to train CNN or CNN with addition (e.g., CNN+SVM), "Their accuracy" refers to the accuracy figures reported by researchers in their respective studies, using specific datasets tailored for those analyses. On the other hand, "our accuracy" refers to the performance of these methods when applied to our "All Classes" dataset. Most models from recent studies achieved high performance on their dataset, but compared with the All Classes dataset, most of them achieved from bad to medium performance.

Table 11

Evaluation results for all datasets

Model		Dense Net121	Dense Net169	Dense Net201	Mobile NetV2	ResNet 101V2	ResNet 152V2	ResNet 50	ResNet 50V2	ResNet RS152	VGG16	VGG19
Dataset	%											
Alpha-bet	Test	95.00	94.64	95.78	49.92	81.03	75.52	77.20	81.09	83.61	91.71	90.84
	Eva.	95.09	94.77	96.08	48.10	81.07	76.77	76.77	81.97	83.64	91.64	91.18
Digits	Test	97.48	96.33	99.21	33.57	75.50	80.44	99.14	88.08	97.05	99.24	99.03
	Eva.	98.38	97.12	99.50	34.22	77.39	99.64	99.64	89.68	97.59	99.54	99.47
Arabic	Test	87.18	93.34	92.10	27.46	74.56	74.35	94.00	78.78	84.78	85.73	90.58
	Eva.	87.22	91.86	90.94	25.92	74.55	92.45	92.45	79.54	83.09	85.71	89.47
Tifinagh	Test	99.91	99.64	99.82	22	90.3	61.14	98.83	96.30	97.57	97.66	98.02
	Eva.	99.06	99.64	99.71	25.33	59.45	98.56	98.56	97.19	96.03	97.19	97.48
Special charac-ters	Test	95.95	96.24	96.05	47.07	86.01	81.45	80.88	86.82	83.33	89.33	88.16
	Eva.	93.93	95.84	94.26	44.61	84.65	78.52	78.52	87.13	82.38	87.64	85.63
Sym-bols	Test	98.35	99.14	97.94	50.45	91.55	87.50	98.83	93.50	98.33	98.41	98.49
	Eva.	98.10	98.73	97.86	49.55	90.25	98.30	98.30	93.26	98.62	97.59	97.90
Charac-ter type	Test	90.71	92.29	92.39	60.30	84.21	85.96	99.90	87.38	88.63	87.47	89.24
	Eva.	90.63	92.61	92.82	60.16	84.33	94.67	98.68	87.54	88.69	87.33	89.27
All classes	Test	87.17	89.34	89.27	36.80	70.19	65.51	67.00	71.04	71.50	84.07	83.37
	Eva.	87.17	89.34	89.27	36.80	70.19	65.51	67.00	71.04	71.50	84.07	83.37

Table 12

Comparison with other models

Authors	**Year**	**Model**	**Dataset**	**Their accuracy**	**Our accuracy**
Waleed Albattah [23]	2022	CNN+SVM	Arabic MNIST digits	97.1%	
			Arabic MNIST character	87.2%	
			All classes (Our dataset)		0.6%
		CNN+LR	Arabic MNIST digits	93.88%	
			Arabic MNIST character	85.6%	
D.S.Prashanth et al. [28]	2021	CNN	Their dataset of 38,750 images	94%	
			All classes (Our dataset)		89.92%
		Modified Lenet CNN	Their dataset of 38,750 images	99%	
			All classes (Our dataset)		93.02%
		AlexNet	Their dataset of 38,750 images	98%	
			All classes (Our dataset)		93.61%
Y. B. Hamdan et al [29]	2021	SVM	MNIST Alphabet	94%	
			All classes (Our dataset)		91.71%
Ali Benaissa et al. [19]	2022	Modified DenseNet201	Custom dataset	99%	
Current work		DenseNet169	All Classes		89.34%
		DenseNet121	Alphabet		**95.78%**
		VGG16	Digits		**99.24%**
		DenseNet169	Arabic		**94.00%**
		DenseNet121	Tifinagh		**99.91%**
		DenseNet169	French Special Characters		**96.24%**
		DenseNet169	Symbols		**99.14%**
		ResNet50	Character Type		**99.90%**

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

In terms of performance on the built dataset, we trained them with pre-trained models, some of which achieved high performance and were generally better than CNN classic models or CNN with addition, especially when the dataset contained fewer classes. Thus, we can conclude that pre-trained models in our usage case are better and more effective.

This comparison illustrates the adaptability of various existing methods to complex datasets and underscores the need for more flexible and robust character recognition models. Such models must effectively handle the diverse linguistic and scriptural elements found in datasets such as All Classes, while maintaining high accuracy. This highlights the importance of this approach in developing a character recognition system that incorporates multiple models, with each model specifically tailored to a particular language/script.

Conclusion

In this era of rapidly advancing digital technology, the field of character recognition systems is becoming increasingly significant in computer vision. This study concentrates on developing eight datasets that cover a variety of languages present in Moroccan documents. We have trained these datasets using selected pre-trained models based on convolutional neural networks (CNN) through transfer learning, and contribute to the development of a character recognition system that incorporates multiple models, each tailored for recognizing different languages. This multipronged approach not only highlights the adaptability of fine-tuned models to complex datasets like "All Classes" but also emphasizes the importance of creating flexible and robust character recognition systems capable of handling the varied linguistic and scriptural elements encountered in such datasets. These advancements underscore our methodology's significance in the broader context of digitizing and managing multilingual documents, marking a pivotal step towards overcoming the challenges of character recognition in diverse and evolving digital landscapes. The primary goal of this research is to determine the most effective model for meeting digitization requirements.

We used pre-trained models such as DenseNet121, VGG16, DenseNet169, and ResNet50, which were trained on datasets including Alphabet, Digits, Arabic, Tifinagh, French Special Characters, Symbols, and Character Type. These models achieved validation accuracies ranging from 94.00% to 99.91%. In addition, we created a comprehensive dataset named 'All-Classes' by merging all individual datasets to test the versatility of the models. Unfortunately, the models exhibited lower performance on this combined dataset. Comparison with recent studies in the same field shows that our pre-trained models generally perform better.

Moving forward, we plan to enhance this approach by incorporating a novel feature extraction technique. This technique blends the methodology discussed in this paper with semantic knowledge and Natural Language Processing (NLP) techniques. The aim is to improve the recognition capabilities of complex fonts and handwriting styles. This future direction in our research is expected to significantly advance the field of character recognition, particularly in handling more challenging and diverse text representations.

Contributions of authors: conceptualization, methodology – **Ali Benaissa, Abdelkhalak Bahri;** formulation of tasks, analysis **– Ali Benaissa;** development of model, software, verification **– Ali Benaissa;** analysis of results, visualization **– Ahmad El Allaoui, My Abdelouahab Salahddine;** writing – original draft preparation, writing – review and editing **– Ali Benaissa, Abdelkhalak Bahri.**

Conflict of interest

The authors declare no conflict of interest related to this research.

Financing

This study was conducted without financial support.

Data availability

The manuscript has associated data in a data repository.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current study.

Acknowledgments

The authors would like to thank the staff of Al Hoceima/Izemmouren commune for providing us with some examples of official documents.

All the authors have read and agreed to the published version of this manuscript.

References

1. Geron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd edition, O'Reilly Media, Inc., 2019. 856 p. ISBN: 978-1492032649.

2. Thorat, C., Bhat, A., Sawant, P., Bartakke, I., & Shirsath, S. A Detailed Review on Text Extraction Using

Optical Character Recognition. *In:* FONG, S., DEY, N. & JOSHI, A., eds. *ICT Analysis and Applications, Lecture Notes in Networks and Systems, 2022,* vol. 314, Singapore: Springer Nature Singapore, pp. 719-728. DOI: 10.1007/978-981-16-5655-2_69.

3. Radwan, M. A., Khalil, M. I., & Abbas, H. M. Neural Networks Pipeline for Offline Machine Printed Arabic OCR. *Neural Processing Letters.* 2018, vol. 48, no. 2, pp. 769-787. DOI: 10.1007/s11063-017-9727-y.

4. Gong, Y., Yang, J., & Shi, X. Towards a comprehensive understanding of digital transformation in government: Analysis of flexibility and enterprise architecture. *Government Information Quarterly*. 2020, vol. 37, no. 3, article no. 101487. DOI: 10.1016/j.giq.2020.101487.

5. Kadhim, A. I. Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review.* 2019, vol. 52, no. 1, pp. 273-292. DOI: 10.1007/s10462-018-09677-1.

6. Sen, P. C., Hajra, M., & Ghosh, M. Supervised Classification Algorithms in Machine Learning: A Survey and Review. *In:* MANDAL, J. K. & BHATTACHARYA, D., eds. *Emerging Technology in Modelling and Graphics. Advances in Intelligent Systems and Computing,* 2020, vol. 937, Singapore: Springer Singapore, pp. 99-111. DOI: 10.1007/978-981-13-7403-6_11.

7. Khan, A., Laghari, A., & Awan, S. Machine Learning in Computer Vision: A Review. *ICST Transactions on Scalable Information Systems,* 2021, iss. 32, aricle no. e4. DOI: 10.4108/eai.21-4-2021.169418.

8. Chukhray, A., Chernenko, M., Stoliarenko, T., & Leshchenko, O. Diagnostic support of intelligent tutor system for teaching skills to construct control object frequency characteristics. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems,* 2022, no. 3, pp. 190-199. DOI: 10.32620/reks.2022.3.14.

9. Oumoulylte, M., Omari Alaoui, A., Farhaoui, Y., El Allaoui, A., & Bahri, A. Convolutional neural network-based skin cancer classification with transfer learning models. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems,* 2023, no. 4, pp. 75-87. DOI: 10.32620/reks.2023.4.07.

10. Hosen, M. A., Moz, S. H., Khalid, M. M. H., Kabir, S. S., & Galib, D. S. M. Face recognition-based attendance system with anti-spoofing, system alert, and email automation. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems*, 2023, no. 2, pp. 119-128. DOI: 10.32620/reks.2023.2.10.

11. Cob-Parro, A. C., Losada-Gutiérrez, C., Marrón-Romera, M., Gardel-Vicente, A., & Bravo-Muñoz, I. A new framework for deep learning video based Human Action Recognition on the edge. *Expert Systems with Applications,* 2024, vol. 238, article no. 122220. DOI: 10.1016/j.eswa.2023.122220.

12. Dovbysh, A., Liubchak, V., Shelehov, I., Simonovskiy, J., & Tenytska, A. Information-extreme machine learning of a cyber attack detection system. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems,* 2022, no. 3, pp. 121-131. DOI: 10.32620/reks.2022.3.09.

13. Islam, M. M., Hossain, M. B., Akhtar, M. N., Moni, M. A. & Hasan, K. F. CNN Based on Transfer Learning Models Using Data Augmentation and Transformation for Detection of Concrete Crack. *Algorithms,* 2022, vol. 15, no. 8, article no. 287. DOI: 10.3390/a15080287.

14. Liu, Y., Wang, Y., & Shi, H. A Convolutional Recurrent Neural-Network-Based Machine Learning for Scene Text Recognition Application. *Symmetry,* 2023, vol. 15, no. 4, article no. 849. DOI: 10.3390/sym15040849.

15. Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., Han, W., Huang, M., Jin, Q., Lan, Y., Liu, Y., Liu, Z., Lu, Z., Qiu, X., Song, R., Tang, J., Wen, J.-R., Yuan, J., Zhao, W. X., & Zhu, J. Pre-trained models: Past, present and future. *AI Open,* 2021, vol. 2, pp. 225-250. DOI: 10.1016/j.aiopen.2021.08.002.

16. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G. & Isard, M. TensorFlow: A System for Large-Scale Machine Learning. *OSDI'16: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, November 2016, pp. 265–283.

17. Hinton, G. E., Osindero, S., & Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* 2006, vol. 18, no. 7, pp. 1527–1554. DOI: 10.1162/neco.2006.18.7.1527.

18. Pham, V., Bluche, T., Kermorvant, C., & Louradour, J. Dropout Improves Recurrent Neural Networks for Handwriting Recognition. *14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 285-290. DOI: 10.1109/ICFHR.2014.55.

19. Benaissa, A., Bahri, A., & Allaoui, A. E. Characters Type Recognition In Moroccan Documents Using CNN. *2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET),* 2022, pp. 1-7. DOI: 10.1109/IRASET52964.2022.9738376.

20. AlJarrah, M. N., Zyout, M. M., & Duwairi, R. Arabic Handwritten Characters Recognition Using Convolutional Neural Network. *12th International Conference on Information and Communication Systems (ICICS),* 2021, pp. 182-188. DOI: 10.1109/ICICS52457.2021.9464596.

21. Hazra, T. K., Singh, D. P., & Daga, N. Optical character recognition using KNN on custom image dataset. *8th Annual Industrial Automation and*

Electromechanical Engineering Conference (IEMECON), 2017, pp. 110-114. DOI: 10.1109/IEMECON.2017.8079572.

22. Prashanth, D. S., Mehta, R. V. K., & Sharma, N. Classification of Handwritten Devanagari Number – An analysis of Pattern Recognition Tool using Neural Network and CNN. *Procedia Computer Science.* 2020, vol. 167, pp. 2445-2457. DOI: 10.1016/j.procs.2020.03.297.

23. Albattah, W., & Albahli, S. Intelligent Arabic Handwriting Recognition Using Different Standalone and Hybrid CNN Architectures. *Applied Sciences*, 2022, vol. 12, iss. 19, article no. 10155. DOI: 10.3390/app121910155.

24. Luo, C., Li, X., Wang, L., He, J., Li, D., & Zhou, J. How Does the Data set Affect CNN-based Image Classification Performance? 5th International Conference on Systems and Informatics (ICSAI), 2018, pp. 361-366. DOI: 10.1109/ICSAI.2018.8599448.

25. Benaissa, A., Bahri, A., & El Allaoui, A. Multilingual character recognition dataset for Moroccan official documents. *Data in Brief.*, 2024, vol. 52, article no. 109953. DOI: 10.1016/j.dib.2023.109953.

26. Kingma, D. P., & Ba, J. Adam: A Method for Stochastic Optimization. *CoRR.*, 2014, vol. abs/1412.6980. DOI: 10.48550/arXiv.1412.6980.

27. Sashank, Kale, S., & Kumar, S. On the Convergence of Adam and Beyond. *arXiv pre-print server*, 2019. DOI: 10.48550/arXiv.1904.09237.

28. Prashanth, D. S., Mehta, R. V. K., Ramana, K., & Bhaskar, V. Handwritten Devanagari Character Recognition Using Modified Lenet and Alexnet Convolution Neural Networks. *Wireless Personal Communications*, 2022, vol. 122, no. 1, pp. 349-378. DOI: 10.1007/s11277-021-08903-4.

29. Hamdan, Y. B., & Sathish. Construction of Statistical SVM based Recognition Model for Handwritten Character Recognition. *Journal of Information Technology and Digital World*, 2021, vol. 3, no. 2, pp. 92-107. DOI: 10.36548/jitdw.2021.2.003.

Received 17.09.2023, Accepted 20.02.2024

УДОСКОНАЛЕНИЙ ПІДХІД ДО ЦИФРОВОЇ ОБРОБКИ АДМІНІСТРАТИВНИХ ДОКУМЕНТІВ МАРОККО ЗА ДОПОМОГОЮ ПОПЕРЕДНЬО НАВЧЕНИХ МОДЕЛЕЙ НА ОСНОВІ CNN: РОЗПІЗНАВАННЯ СИМВОЛІВ

Алі Бенаісса, Абдельхалак Бахрі, Ахмад Ель Аллауї,
Мій Абдельуахаб Салахддін

У цифрову епоху ефективна **цифрова** обробка адміністративних документів є справжнім викликом, особливо для мов зі складними скриптами, які використовуються у марокканських документах. **Предметом** цієї статті є цифрова обробка марокканських адміністративних документів за допомогою попередньо навчених згорткових нейронних мереж (CNN) для розширеного розпізнавання символів. **Метою** цього дослідження є вирішення унікальних викликів у точній цифровій обробці різних марокканських скриптів і макетів, що є важливим у цифровій трансформації адміністративних процесів. **Нашою метою** було розробити ефективну та високоточну систему розпізнавання символів, спеціально адаптовану для марокканських адміністративних текстів. **Задачі** включали всебічний аналіз та налаштування попередньо навчених моделей CNN та ретельне тестування продуктивності на різноманітному наборі марокканських адміністративних документів. **Методологія** включала детальну оцінку різних архітектур CNN, навчених на наборі даних, що представляє різні типи використовуваних символів у марокканських адміністративних документах. Це забезпечило **адаптивність** моделей до реальних сценаріїв з акцентом на точність та ефективність у розпізнаванні символів. **Результати** були суттєвими, зокрема, DenseNet121 досяг 95.78% точності на наборі даних Alphabet, тоді як VGG16 зафіксував 99.24% точності на наборі даних Digits. DenseNet169 продемонстрував 94.00% точність на арабському наборі даних, а також 99.9% на наборі даних Tifinagh та 96.24% точності на наборі даних French Special Characters. Крім того, DenseNet169 досяг 99.14% точності на наборі даних Symbols. Також ResNet50 досяг 99.90% точності на наборі даних Character Type, що дозволяє точно визначити, до якого набору даних належить символ. **На закінчення**, це дослідження є суттєвим удосконаленням у галузі цифрової обробки марокканських адміністративних документів. Підхід на основі CNN, представлений у дослідженні, перевершує деякі традиційні методи розпізнавання символів. Ці результати не лише сприяють цифровій обробці та управлінню документами, але й відкривають нові можливості для майбутніх досліджень у адаптації цієї технології для інших мов та типів документів.

Ключові слова: розпізнавання символів; попередньо навчені моделі; згорткові нейронні мережі (CNNs); офіційні документи Марокко; цифрове перетворення.

Алі Бенаісса – аспірантка у галузі штучного інтелекту та цифрової трансформації в Національній школі прикладних наук Аль-Хосейми, Лабораторія прикладних наук, Команда SDIC, Університет Абдельмалека Есааді, Тетуан, Марокко.

Абдельхалак Бахрі – доктор з інформаційних технологій, професор Національної школи прикладних наук Аль-Хосейми, Лабораторія прикладних наук, Команда: SDIC, Університет Абдельмалека Есааді, Тетуан, Марокко.

Ахмад Ель Аллауї – доктор з інформаційних технологій, професор факультету науки і технологій FSTE, Еррачідія, Марокко.

Мій Абдельуахаб Салахддін – доктор з менеджменту, професор Національної школи бізнесу та менеджменту, Університет Абдельмалека Ессааді, Танжер, Марокко.

Ali Benaissa – PhD student in the field of Artificial Intelligence and Digital Transformation at the National School of Applied Sciences Al-Hoceima, Laboratory of Applied Sciences, Team: SDIC, Abdelmalek Essaadi University, Tétouan, Morocco,
e-mail: ali.benaissa@etu.uae.ac.ma, ORCID: 0009-0000-8944-5708.

Abdelkhalak Bahri – Doctor in Information Technology, Professor at the National School of Applied Sciences Al-Hoceima, Laboratory of Applied Sciences, Team: SDIC, Abdelmalek Essaadi University, Tétouan, Morocco,
e-mail: abahri@uae.ac.ma, ORCID: 0000-0002-8527-7281.

Ahmad El Allaoui – Doctor in Information Technology, Professor at the Faculty of Science and Technology FSTE, Errachidia, Morocco,
e-mail: a.elallaoui@umi.ac.ma, ORCID: 0000-0002-8897-3565, Scopus Author ID: 57195110231.

My Abdelouahab Salahddine – Doctor in Management, Professor at the National School of Business and Management, Abdelmalek Essaadi University, Tangier, Morocco,
e-mail: salahddine@uae.ac.ma, ORCID: 0009-0001-0997-6099.