Characters Type Recognition in Moroccan Documents Using CNN

Ali Benaissa, Abdelkhalak Bahri and Ahmad El Allaoui

September 3, 2022

Characters Type Recognition In Moroccan Documents Using CNN

Ali Benaissa
Department of Mathematics and Computer Science, Data Sciences and Competitive Intelligence team (DSCI), Laboratory of applied Science National School of Applied Science Abdelmalek Essaadi University
Al Hoceima, Morocco
ali.benaissa.gry@gmail.com

Abdelkhalak Bahri
Department of Mathematics and Computer Science, Data Sciences and Competitive Intelligence team (DSCI), Laboratory of applied Science National School of Applied Science Abdelmalek Essaadi University
Al Hoceima, Morocco
abahri@uae.ac.ma

Ahmad El Allaoui
Department of Engineering Technical Sciences
Business Intelligence and Systems Modeling team
Faculty of Science and Technology
Errachidia, Morocco
a.elallaoui@uae.ac.ma

***Abstract*—In recent years, the classification and recognition became a hot topic in computer studies. Deep Learning algorithms present the most outstanding performance in classification and recognition issues. In this paper, we focus on applying these techniques to extract the characters types from the Moroccan official legal documents. Because of the variety of many pre-trained models. we designed a system able to Loop over each model available on TensorFlow Keras API, based on Transfer Learning technique, we trained the models on the dataset that we have built. And the outcome was that the DensNet and VGGNet models have achieved the best performance, with a validation accuracy of 98%. In addition to this, we proposed a modified model based on DenseNet201, the result achieved is 98.99% of overall accuracy.**

Keywords—Character recognition, Image classification, pre-trained models, Keras, TensorFlow, Transfer Learning.

I. INTRODUCTION

Images classification is one of the most innovative methods to recognize the elements of an image, this method saves the image's parameters detached from a dataset, in order to recognize the similar images, therefore, image classification is useful to recognize them.

Character recognition is basically the use of a group of images that contain characters, with different shapes, color and position as parameters, to recognize the other characters with different parameters.

Deep learning provides the techniques, methods and algorithms for object recognition and images classification, as examples of these magnificent Deep Learning machines we have to mention; Support Vector Machine, k-Nearest Neighbor, Convolutional Neural Network (CNN). However, we will rely in this research on CNN, because it is a new mechanism in the field, and it is approved by an official and respected platform (TensorFlow which is developed by the Google Brain team).

CNN is notorious for extracting and objects recognitions, which is also known for its high predictive accuracy and multi-use to solve different problem with deep learning.

As this research progresses, we will discuss the use of the Deep Learning techniques, especially convolution neural network (CNN), focusing on pre-trained application models' from TensorFlow Keras Library, (a deep learning API programming in Python, it is developed to be fast and runs smoothly in experimentation mode), to expand on this topic, we will be presenting a comparative study between Keras modules to select the most efficient one, a module that can recognize the characters type of Arabic, French and Tamazight languages and categorize them separately.

This paper is structured as follows: in the second section we will talk about the proposed issue, in third section we present a works that is related to ours, in the furth section, we discuss the used methods, as the approach method, system flowchart and dataset construction, in fifth section we will be presenting our experiments, and the results, in final section we finish with a conclusion, and we open about upcoming works.

II. PROBLEMATIC STATEMENT

There are many countries around the world's most of which have multiple languages as their official language. Therefore, these countries tend to use multiple languages in their legal and official documents. However, with technology's development it's important to digitally convert some of these documents if not all, but the similarity between the letters of different languages may impose a problem.

In this research, we take a look at Morocco as an example, to demonstrate how our research can be applied, in the Moroccan legal documents, the text is multilingual most of the time, the first is Arabic, it is the first official language of the country, the second is the French language, then Tamazight (Tifinar the written form of the language) which in later years became recognized by the Moroccan constitution reform made in 2011, which recognizes it as an official language as well. Since then, the country is tying implementing it in everyday lives of the Moroccan people. However, in the digital age we live nowadays. We want to be able to recognize this language, making it accessible and easy to navigate using AI. Unfortunately, the recognition system makes the written form of these languages unrecognizable, and this is mainly due to the similarity between the written forms of these three languages.

III. RELATED WORK

For the past few years, many techniques and methods for classifying images and different datasets have been proposed to extract specific information, as several models in deep learning have been made and developed as well.

N. AHARRANE et al [7], developed the Amazigh printed word images (APWID), it's a dataset for a wide-scale benchmarking, the dataset proposed contain 1795 of different

Tifinagh symbols, with different fonts size and styles to build 114800 images, which form the presented dataset.

Youssef Es-Saady et al. [8], proposed a handwrit tifinagh character recognition system, they are tested the system using AMHCD dataset, that contain 24180 handwritten Amazigh character, they split the dataset to 90% of training and the rest 10% for testing, using ten-fold cross-validation they get 94.62% of recognition rate when integrated the feature of the horizontal and vertical centerlines, and 94.96% of recognition rate when integrated the feature of the horizontal and vertical baselines, they also discuss that the confusion in the system is of the simulation between Tifinagh characters.

Mohamed N AlJarrah et al. [9], proposed a model based on Convolutional Neural Networks, to recognize the Arabic handwritten characters, the model in their experimentation is work on a dataset containing 16800 images of Arabic handwritten characters, with different shapes and forms, their proposed model achieved a 97.2% of accuracy without of data augmentation, when they used data augmentation, the accuracy rose to 97.7%.

Mohammed Aarif K.O and Sivakumar Poruran [10], presented variants of two pre-trained OCR-Nets (AlexNet and GoogleNet), that are based on CNN, applied the transfer learning, they used a dataset of handwritten Urdu characters in a big arrangement of fonts and sizes, to be recognized using the proposed networks, also they developed manually a dataset for objective evaluation as well; as result, the experimentation showed that OCR-AlexNet achieved accuracy of 96.3% and OCR-GoogleNet achieved accuracy of 94.7%.

Najwa Altwaijry and Isra Al-Turaik [11], presented a new dataset called Hijja of Arabic handwritten characters, these characters are written by children aged 7 to 12 years old, in addition to this, they presented a model of Arabic handwritten ystem, to recognize the character of Hijja dataset (29 classes) and Arabic Handwritten Character Dataset (AHCD) dataset, and compared the proposed model with the proposed one for AHCD by El-Sawy A et al. [14], on the Hijja dataset the proposed model achieved 88% of accuracy rate while the model of AHCD achieved an accuracy of 80%.

Md. Mehedi Hasan et al. [12], using transfer learning, the team propped a modified DenseNet-201 architecture, to design a system for recognition of the Bengali handwritten characters, the modified model has a fully connected layer (1024) with ReLu function, a dropout (50%), a fully connected layer (512) with ReLu function and dropout layer (50%) on the except output layer, they proposed model achieved an overall accuracy of 96.89%.

In relation with the different works cited in this section, in the context of the Moroccan documents that contain different characters type, Tifinagh, Arabic, Alphabet… etc. In this paper, we will use pre-trained models in the goal to classify and recognize the Moroccan documents characters.

IV. METHODS

In this section, we describe our double contributions. in the first contribution, we have proposed a new method, to improve the accuracy of character type recognition accuracy. in the second contribution, we have built a new dataset, of used character in Moroccan documents.

A. Approach method:

Based on the results that we will get in the next section, we will choose the best model (based model) with the highest validation accuracy and the lowest parameter's number, then we will try to apply the Fine-Tuning on our proposed model by adding the layers, in order of augments the trainable parameters for get the best result compared to the based model, for whatever reason, if we get unexpected results, we will try with the next module using the same selection criteria. We will manipulate our proposed model by adding the layers: Flatten, BatchNormalization, Dropout, Dense.

B. 2.System flowchart:

We designed a system that is able to load a local dataset, and train the whole Keras models on the loaded dataset, we can describe the system in flowchart showing in Fig.

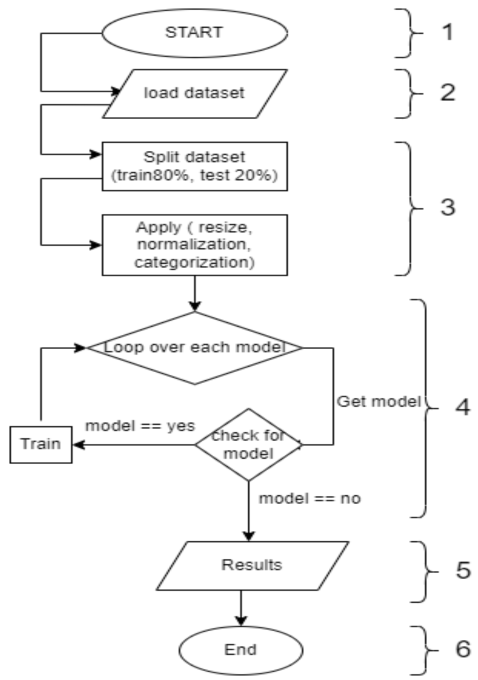


Fig.1. Flowchart system.

We can divide the system's running process into six parts, we start running system in the first part, then we load our local dataset in second part, and we split our dataset afterwards we resize the images to the given dimension. In addition to this, we add specified normalization rate and categorize them. The third part involves the system's conditions, we loop over each available model in Keras library, we pick the appropriate model, train it, check if there's any other model not training yet, until we pick and train all suitable models. In the fifth part of the process, we list the results, and finally we end the system in the sixth part.

C. Dataset construction:

Our objective is to recognize the character's type. To achieve this result, we had to build a new dataset, that contain 6 classes: Tifinagh handwritten [13], Arabic handwritten [14], Alphabet handwritten [15], Alphabet printed, Digits handwritten, and Digits printed, inasmuch as the data big, the model results better, therefore, I combined data from different sources. The dataset that I built proposed some problems, because of the images are different in dimension, shapes and position. The dataset structure is as showing in "Fig 1".

```
DATA/
├── train/
|   ├── Alphabet handwritten/
|   ├── Arabic handwritten/
|   ├── Digits handwritten/
|   ├── Tifinagh handwritten/
|   ├── Alphabet printed /
|   ├── Digits printed /
├── test/
|   ├── Alphabet handwritten/
|   ├── Arabic handwritten/
|   ├── Digits handwritten/
|   ├── Tifinagh handwritten/
|   ├── Alphabet printed/
|   ├── Digits printed /
```

Fig. 1. Dataset structure

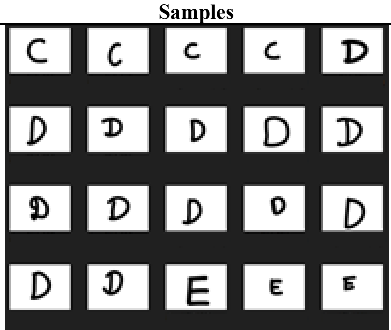
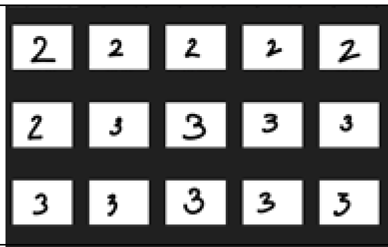
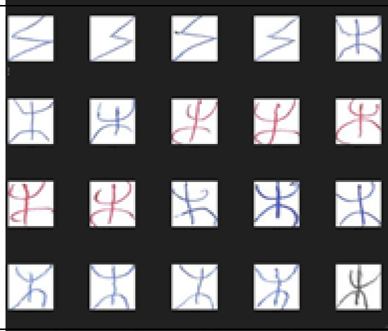
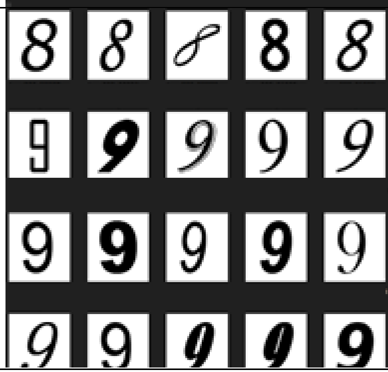
The table 1 shows the number of images in each class before split it to train and test.

TABLE 1. IMAGES NUMBER AND DIMENSION IN EACH CLASS

Classes	Images number
Alphabet handwritten	2860
Arabic handwritten	16680
Digits handwritten	10550
Tifinagh handwritten	15740
Alphabet prited	1560
Digits printed	10460

Table 2 shows samples of the dataset classes.

TABLE 2. DATASET SAMPLES

Classes	Samples
Alphabet handwritten	
Arabic handwritten	
Digits handwritten	
Tifinagh handwritten	
Alphabet printed	
Digits printed	

Using a customized dataset, we got the images in different dimensions, for solving the incompatible dataset images shapes and dimensions, we used OpenCV [16] as a library to resize the images to a unitary dimension (32x32), and we also used prepossessing from Keras library [16] to prepare and process the images, in addition to this, we have used normalization [17] (of 255 as a float32 type), To get full accurate information from an image, and remove the distortions.

V. EXPERIMENTATION

All our experiment is applied on our dataset that we have cited in the section 2 of part III. For the performance measure, we have used accuracy (1) and precision (2):

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

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

Where:

- TP = True Positives.
- TN = True Negatives.

- FP = False Positives.
- FN = False Negatives.

A. *Pre-trained models System:*

The choice of a pre-trained model [18] to be used in Deep Learning tasks became a challenging obstacle in computer vision field. Therefore, we designed a system able to go through each available model in Keras library, and pick one based on validation accuracy rate and the precision rate, on a customized dataset. “Fig. 2” presents the system steps.

Fig 2. The system steps.

B. *Hyperparameters:*

In the used dataset, we split it to 80% of training and 20% of testing, to obtain a 100% of evaluation, we copy 80% from the training and 20% from the testing. All models are trained for 8 epochs with a batch size of 32 as after that the validation loss grew almost constant for the rest of the epochs. ’Adam’ optimizer [19] with a default learning rate of 0.001 was given to maximize the error function. A categorical cross-entropy function was applied for the loss or error function.

C. *Experiment results and analysis:*

After training, we obtained the results presented in the table 3, we also have models that have not been trained, this is due to the dimensions of the images that are not compatible with the input models, which they are NASNetLarge, InceptionResNetV2, InceptionV3, NASNetMobile and Xception.

TABLE 3. PARAMETERS NUMBER AND VALIDATION ACCURACY OF MODELS

	Model name	**Parameters number**	**Validation accuracy**
10	EfficientNetB7	64097687	0.379155
3	EfficientNetB0	4049571	0.380352
8	EfficientNetB5	28513527	0.495534
7	EfficientNetB4	17673823	0.508609
6	EfficientNetB3	10783535	0.531074
5	EfficientNetB2	7768569	0.593039
4	EfficientNetB1	6575239	0.604272
9	EfficientNetB6	40960143	0.613571
11	MobileNet	3228864	0.749839
12	MobileNetV2	2257984	0.889145
14	MobileNetV3Small	1529968	0.907283
17	ResNet152	58370944	0.908664
15	ResNet101	42658176	0.916766
18	ResNet152V2	58331648	0.954148
19	ResNet50	23587712	0.960501
16	ResNet101V2	42626560	0.961698
13	MobileNetV3Large	4226432	0.963447
20	ResNet50V2	23564800	0.963908
0	DenseNet121	7037504	0.984256
1	DenseNet169	12642880	0.986189
21	VGG16	14714688	0.987754
22	VGG19	20024384	0.987754
2	DenseNet201	18321984	0.989148

As well, we generate the graph showed in “Fig 3”

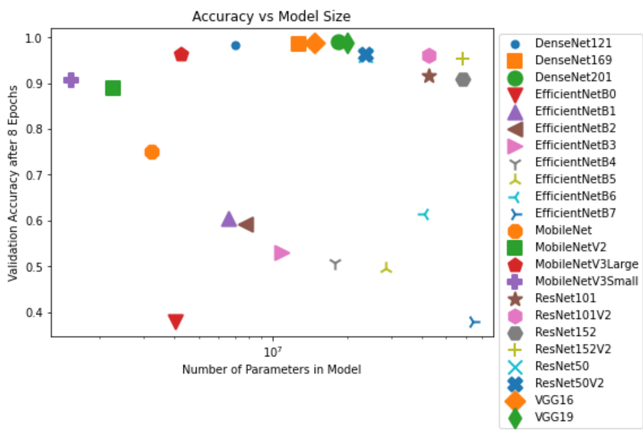


Fig. 3. PARAMETERS NUMBER AND VALIDATION ACCURACY OF MODELS.

“Fig. 3”, presents the validation accuracy of each model against the parameters number, to be more precise, the model that has the fewest parameters number, and the highest validation accuracy, we consider it as the best model.

The table 3 and “Fig. 3” gives us the results as follows; we note that DenseNets provides the best results. In general, we can say that all VGGNet achieved a promising result compared with the paraments numbers, where we find that DenseNet121 reaches 0.977074 of validation accuracy, while VGG16 reaches almost the same as the latter, reaching 0.975048, although the difference in parameter is rather large, as we find at VGG16 number of 20024384 parameters, while the DenseNet121 has only 7037504 parameters.

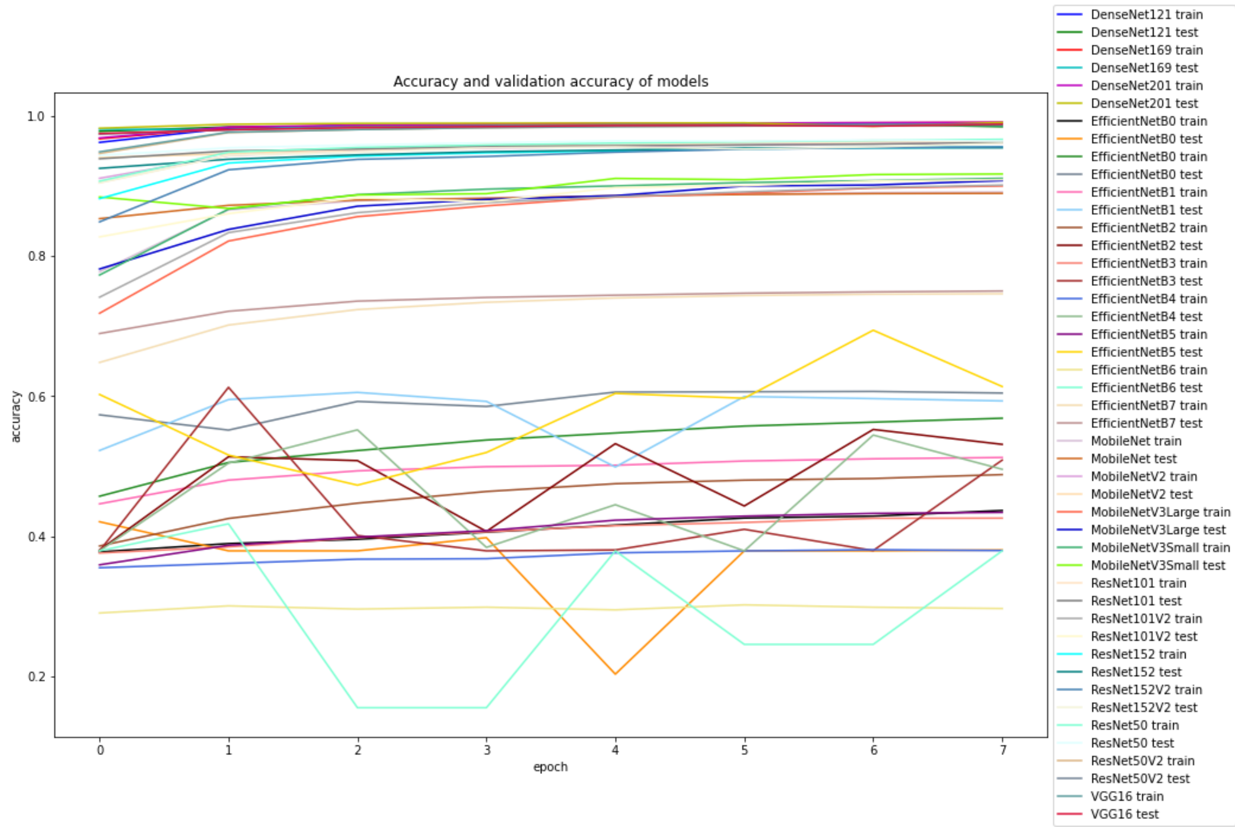


Fig. 4. Accuracy and validation accuracy of models

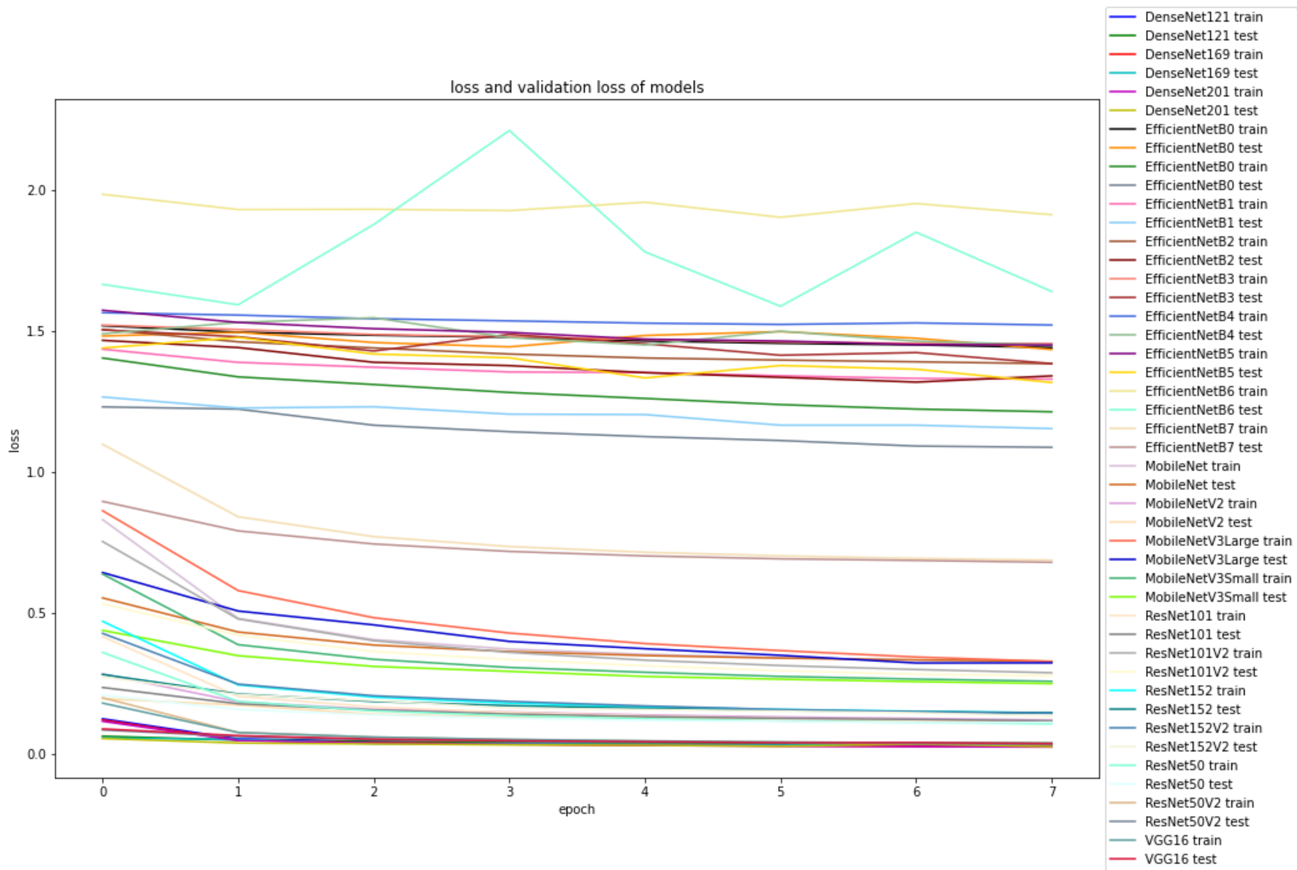


Fig. 5. Loss and validation loss of models

D. Proposed modified model:

Based on the result we got in section C, we choose DenseNet201, to be the basic model to modify, in output model we added a Dense layer (128, ReLu), BatchNormalization layer, Dropout (25%), Dense layer (1024, ReLu), BatchNormalization layer and Dropout (50) layer "fig 4" shows the model architect.

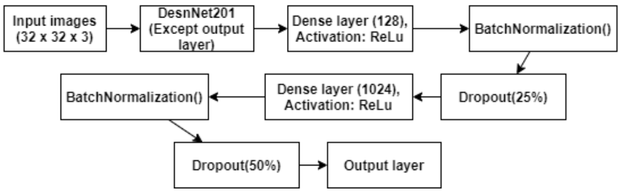


Fig. 6. Proposed modified model architect.

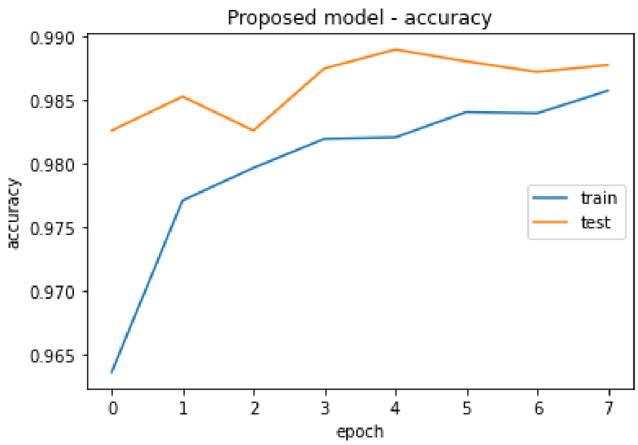


Fig. 7. Training and validation accuracy of proposed model.

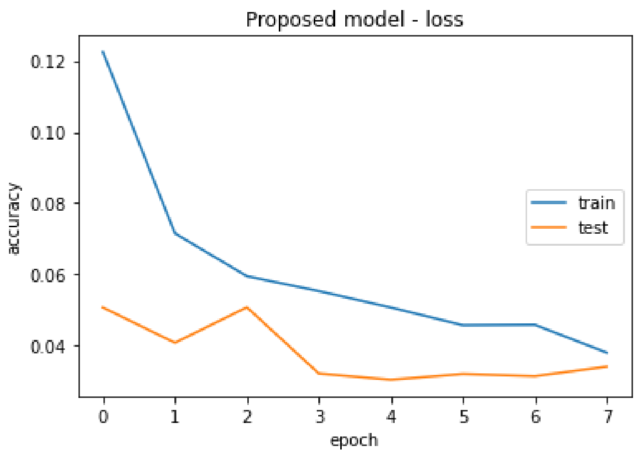


Fig. 8. Training and validation loss of proposed model.

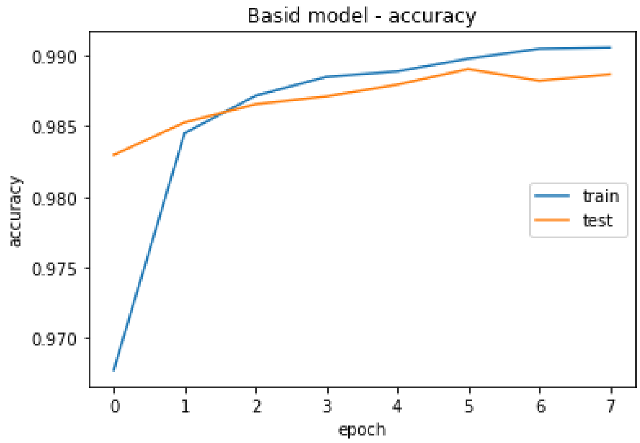


Fig. 9. Training and validation accuracy of basic model.

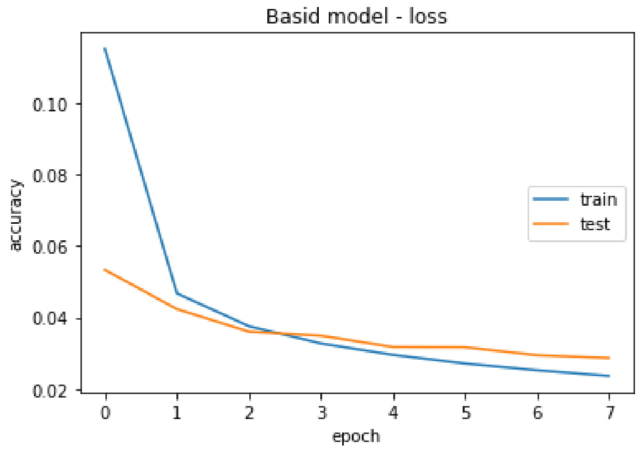


Fig. 10. Training and validation loss of basic model.

The proposed model, compared to basic one, achieved a similar validation accuracy. "Fig 5" represents the training accuracy and "Fig 6" represents the training loss and validation accuracy of our proposed model, while the overall accuracy is 98.78%, in other hand, "Fig 7" represents the training accuracy and "Fig 8" represents the training loss and validation accuracy of the basic model, while the overall accuracy is 98.99%. Despite that the result are close, but the Fine-Tuning [20] open other fields to focus on by researches and improve the using of ore-trained models by added the layers, it may come with higher accuracy and prediction.

VI. Conclusion

Classification and recognition are one of the most important fields on computer studies of our time, and it keeps getting innovated and advanced in order to adapt easily to our daily lives.

In this paper, we built a custom dataset, which used to classify and recognize a character in Moroccan documents. In addition to this, we presented a systematic approach to choose the stable and the smoothly pre-trained model between the available TensorFlow Keras API models, we also proposed a modified pre-trained model, to improve the results.

In the near future, we are looking forward to improving the characters classification and recognition results, by combining several methods of machine learning and semantic information's.

REFERENCES

[1] HORAK, Karel et SABLATNIG, Robert. Deep learning concepts and datasets for image recognition: overview 2019. In: Eleventh international conference on digital image processing (ICDIP 2019). International Society for Optics and Photonics, 2019. p. 111791S.

[2] Anna Syberfeldt, Fredrik Vuoluterä, Image Processing based on Deep Neural Networks for Detecting Quality Problems in Paper Bag Production, Procedia CIRP, Volume 93 ,2020, Pages 1224-1229, ISSN 2212-8271.

[3] Ajeet Ram Pathak, Manjusha Pandey, Siddharth Rautaray, Application of Deep Learning for Object Detection, Procedia Computer Science, Volume 132,2018, Pages 1706-1717, ISSN 1877-0509.

[4] SHARMA, Reya, KAUSHIK, Baijnath, et GONDHI, Naveen. Character recognition using machine learning and deep learning-a survey. In: 2020 International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2020. p. 341-345.

[5] SMYS, S., CHEN, Joy Iong Zong, et SHAKYA, Subarna. Survey on Neural Network Architectures with Deep Learning. Journal of Soft Computing Paradigm (JSCP), 2020, vol. 2, no 03, p. 186-194.

[6] CETINIC, Eva, LIPIC, Tomislav, et GRGIC, Sonja. Learning the Principles of Art History with convolutional neural networks. Pattern Recognition Letters, 2020, vol. 129, p. 56-62.

[7] Aharrane, Nabil & Dahmouni, Abdellatif & Moutaouakil, Karim & Satori, Khalid & Mohammed, Sidi & Abedallah, Ben. (2016). A Dataset of Amazigh Printed Words Images.

[8] Es-saady, Youssef & Amrouch, Mustapha & Rachidi, Ali & El yassa, Mostafa & Mammass, D.. (2014). Handwritten Tifinagh Character Recognition Using Baselines Detection Features. International Journal of Scientific & Engineering Research (IJSER). 5. 1177-1182.

[9] ALJARRAH, Mohammed N., MO'ATH, M. Zyout, et DUWAIRI, Rehab. Arabic Handwritten Characters Recognition Using Convolutional Neural Network. In : 2021 12th International Conference on Information and Communication Systems (ICICS). IEEE, 2021. p. 182-188.

[10] KO, Mohammed Aarif et PORURAN, Sivakumar. OCR-Nets: Variants of Pre-trained CNN for Urdu Handwritten Character Recognition via Transfer Learning. Procedia Computer Science, 2020, vol. 171, p. 2294-2301.

[11] ALTWAIJRY, Najwa et AL-TURAIKI, Isra. Arabic handwriting recognition system using convolutional neural network. Neural Computing and Applications, 2021, vol. 33, no 7, p. 2249-2261.

[12] HASAN, Md Mehedi, SRIZON, Azmain Yakin, SAYEED, Abu, et al. Bengali Handwritten Compound Characters Recognition by Utilizing Transfer Learning. In: 2020 11th International Conference on Electrical and Computer Engineering (ICECE). IEEE, 2020. p. 218-221.

[13] SAADY, Youssef Es, RACHIDI, Ali, YASSA, M., et al. Amhcd: A database for amazigh handwritten character recognition research. International Journal of Computer Applications, 2011, vol. 27, no 4, p. 44-48.

[14] EL-SAWY, Ahmed, LOEY, Mohamed, et EL-BAKRY, Hazem. Arabic handwritten characters recognition using convolutional neural network. WSEAS Transactions on Computer Research, 2017, vol. 5, p. 11-19.

[15] T. E. de Campos, B. R. Babu and M. Varma. Character recognition in natural images. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, February 2009.

[16] Bradski, G., 2000. The OpenCV Library. Dr. Dobb's Journal of Software Tools.

[17] Chollet, F. (2015) keras, GitHub. https://github.com/fchollet/keras

[18] LECUN, Yann A., BOTTOU, Léon, ORR, Genevieve B., et al. Efficient backprop. In : Neural networks: Tricks of the trade. Springer, Berlin, Heidelberg, 2012. p. 9-48.

[19] U.K. Lopes, J.F. Valiati, Pre-trained convolutional neural networks as feature extractors for tuberculosis detection, Computers in Biology and Medicine, Volume 89,2017, Pages 135-143, ISSN 0010-4825.

[20] Z. Zhang, "Improved adam optimizer for deep neural networks," in 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). IEEE, 2018, pp. 1–2

[21] J Praveen Gujjar, R Prasanna Kumar H, Niranjan N. Chiplunkar, Image Classification and Prediction using Transfer Learning in Colab Notebook, Global Transitions Proceedings, ISSN 2666-285X 2021.