Data and Metadata. 2023; 2:174
doi: 10.56294/dm2023174

ORIGINAL

Transformative Progress in Document Digitization: An In-Depth Exploration of Machine and Deep Learning Models for Character Recognition

Progreso transformador en la digitalización de documentos: Una exploración en profundidad de los modelos de aprendizaje automático y profundo para el reconocimiento de caracteres

Ali Benaissa[1,2], Abdelkhalak Bahri[1], Ahmad El Allaoui[3], My Abdelouahab Salahddine[2]

[1]ENSAH, Laboratory of Applied Science - Data Science and Competitive Intelligence Team (DSCI), Abdelmalek Essaadi University (UAE), Tetouan, Morocco.
[2]The National School of Management Tangier, Governance and Performance of Organizations laboratory - Finance and Governance of Organizations team, Abdelmalek Essaadi University, Tangier, Morocco.
[3]Faculty of Sciences and Techniques Errachidia, Engineering Sciences and Techniques, STI-Laboratory - Decisional Computing and Systems Modelling Team, Moulay Ismail University of Meknes, Morocco.

Cite as: Benaissa A, Bahri A, El Allaoui A, Abdelouahab Salahddine M. Transformative Progress in Document Digitization: An In-Depth Exploration of Machine and Deep Learning Models for Character Recognition. Data and Metadata. 2023;2:174. https://doi.org/10.56294/dm2023174

Submitted: 09-08-2023 **Revised:** 22-09-2023 **Accepted:** 04-11-2023 **Published:** 27-12-2023

Editor: Prof. Dr. Javier González Argote
Guest Editor: Yousef Farhaoui

Note: Paper presented at the International Conference on Artificial Intelligence and Smart Environments (ICAISE'2023).

ABSTRACT

Introduction: this paper explores the effectiveness of character recognition models for document digitization, leveraging diverse machine learning and deep learning techniques. The study, driven by the increasing relevance of image classification in various applications, focuses on evaluating Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and VGG16 with transfer learning. The research employs a challenging French alphabet dataset, comprising 82 classes, to assess the models' capacity to discern intricate patterns and generalize across diverse characters.
Objective: This study investigates the effectiveness of character recognition models for document digitization using diverse machine learning and deep learning techniques.
Methods: the methodology initiates with data preparation, involving the creation of a merged dataset from distinct sections, encompassing digits, French special characters, symbols, and the French alphabet. The dataset is subsequently partitioned into training, test, and evaluation sets. Each model undergoes meticulous training and evaluation over a specific number of epochs. The recording of fundamental metrics includes accuracy, precision, recall, and F1-score for CNN, RNN, and VGG16, while SVM and KNN are evaluated based on accuracy, macro avg, and weighted avg.
Results: the outcomes highlight distinct strengths and areas for improvement across the evaluated models. SVM demonstrates remarkable accuracy of 98,63 %, emphasizing its efficacy in character recognition. KNN exhibits high reliability with an overall accuracy of 97 %, while the RNN model faces challenges in training and generalization. The CNN model excels with an accuracy of 97,268 %, and VGG16 with transfer learning achieves notable enhancements, reaching accuracy rates of 94,83 % on test images and 94,55 % on evaluation images.
Conclusion: our study evaluates the performance of five models—Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and VGG16 with transfer learning—on character recognition tasks. SVM and KNN demonstrate high accuracy, while RNN faces challenges in training. CNN excels in image classification, and VGG16, with transfer learning, enhances

accuracy significantly. This comparative analysis aids in informed model selection for character recognition applications.

Keywords: Character Recognition; Machine Learning/Deep Learning Models; Document Digitization.

RESUMEN

Introducción: este artículo explora la eficacia de los modelos de reconocimiento de caracteres para la digitalización de documentos, aprovechando diversas técnicas de aprendizaje automático y aprendizaje profundo. El estudio, impulsado por la creciente relevancia de la clasificación de imágenes en diversas aplicaciones, se centra en evaluar Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) y VGG16 con aprendizaje de transferencia. La investigación emplea un desafiante conjunto de datos del alfabeto francés, que comprende 82 clases, para evaluar la capacidad de los modelos para discernir patrones intrincados y generalizar a través de diversos caracteres.
Objetivo: este estudio investiga la eficacia de los modelos de reconocimiento de caracteres para la digitalización de documentos utilizando diversas técnicas de aprendizaje automático y aprendizaje profundo.
Métodos: la metodología comienza con la preparación de los datos, que implica la creación de un conjunto de datos fusionados a partir de distintas secciones, que abarcan dígitos, caracteres especiales franceses, símbolos y el alfabeto francés. A continuación, el conjunto de datos se divide en conjuntos de entrenamiento, prueba y evaluación. Cada modelo se somete a un meticuloso proceso de entrenamiento y evaluación durante un número determinado de épocas. El registro de las métricas fundamentales incluye la exactitud, la precisión, la recuperación y la puntuación F1 para CNN, RNN y VGG16, mientras que SVM y KNN se evalúan en función de la exactitud, el promedio macro y el promedio ponderado.
Resultados: los resultados ponen de relieve los puntos fuertes y las áreas de mejora de los modelos evaluados. SVM muestra una precisión notable del 98,63 %, lo que pone de relieve su eficacia en el reconocimiento de caracteres. KNN muestra una alta fiabilidad con una precisión global del 97 %, mientras que el modelo RNN se enfrenta a retos en el entrenamiento y la generalización. El modelo CNN sobresale con una precisión del 97,268 %, y VGG16 con aprendizaje por transferencia logra mejoras notables, alcanzando tasas de precisión del 94,83 % en imágenes de prueba y del 94,55 % en imágenes de evaluación.
Conclusión: nuestro estudio evalúa el rendimiento de cinco modelos -Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) y VGG16 con aprendizaje por transferencia- en tareas de reconocimiento de caracteres. SVM y KNN demuestran una gran precisión, mientras que RNN se enfrenta a dificultades en el entrenamiento. CNN destaca en la clasificación de imágenes, y VGG16, con aprendizaje por transferencia, mejora significativamente la precisión. Este análisis comparativo ayuda en la selección informada de modelos para aplicaciones de reconocimiento de caracteres.

Palabras clave: Reconocimiento de Caracteres; Aprendizaje Automático/Modelos de Aprendizaje Profundo; Digitalización de Documentos.

INTRODUCTION

In recent years, the field of artificial intelligence (AI) has witnessed significant advancements, particularly in machine learning and deep learning methods.(1) The increasing relevance of image classification in various applications has driven research efforts to enhance character recognition models for document digitization.(2) This paper explores the effectiveness of such models, leveraging diverse machine learning and deep learning techniques, including Support Vector Machine (SVM),(3) K-Nearest Neighbors (KNN),(4) Recurrent Neural Network (RNN),(5) Convolutional Neural Network (CNN),(6) and VGG16 with transfer learning.(7)

The study utilizes a challenging dataset with 82 classes to evaluate the models' capacity to discern intricate patterns and generalize across diverse characters. The dataset preparation involves merging three sections encompassing French special characters, symbols, and the French alphabet. The models undergo meticulous training and evaluation, and key metrics such as accuracy, precision, recall, and F1-score are recorded.

Results showcase distinct strengths and areas for improvement across the evaluated models. SVM demonstrates remarkable accuracy of 98,63 %, while KNN exhibits high reliability with an overall accuracy of 97 %. The RNN model faces challenges in training and generalization, CNN excels with an accuracy of 97,268 %, and VGG16 with transfer learning achieves notable enhancements, reaching accuracy rates of 94,83 % on test images and 94,55 % on evaluation images.

This study systematically evaluates the performance of five-character recognition models, providing valuable insights for document digitization applications.[8] SVM and KNN demonstrate high accuracy, while CNN excels in image classification. The hybrid CNN+SVM and CNN+LR models from other studies are also compared, emphasizing the effectiveness of SVM and KNN in diverse image classification scenarios. The findings contribute to the understanding of deep CNN architectures and underscore the potential of transfer learning for character recognition.

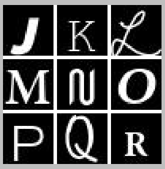
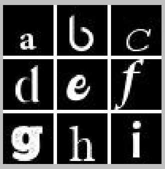
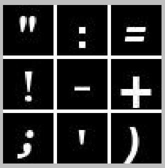
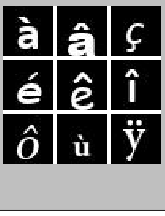
METHODS

Dataset preparation

To conduct rigorous experiments and identify optimal methods for letter recognition, we meticulously curated a diverse dataset amalgamated from various sources, building upon the detailed procedures outlined in our previously published work.[9] The dataset is intentionally diverse, encompassing distinct sections, namely Digits, French Special Characters, Symbols, and the French Alphabet in both uppercase and lowercase. This meticulous selection aims to ensure the representation of a broad spectrum of characters, facilitating a comprehensive evaluation of the recognition capabilities of our models.

The dataset compilation process involved the establishment of a unified folder structure to facilitate seamless access and management. Details of the used and the merged datasets are in table 1.

Table 1. Dataset details

Dataset	Samples	Number of classes	Total images
Alphabet		26	90897
		26	
Symbols		14	22400
French special characters		16	25600
Merged dataset		82	138897

Data Parameters

Following the compilation of our comprehensive dataset, we undertook a step in enhancing the reliability of our experiments by partitioning it into three sets: the Training Set (80 %), the Test Set (20 % of training set), and the Evaluation Set (20 %). This strategic division is paramount as it ensures an unbiased assessment of our models' performance, enabling a comprehensive evaluation of their generalization capabilities.

The detailed allocation of the merged dataset across these sets is systematically outlined in table 2:

Table 2. Dataset parameters

Dataset	Number of classes	Pure data	Train set	Test set	Evaluation set
Merged dataset	82	138897	88877	22220	27748

This systematic approach to data preparation is designed to guarantee a balanced and representative distribution of classes across each set, providing a robust foundation for our subsequent experiments and evaluations. Specifically, the train set is carefully structured to comprise 80 % of the data, ensuring an extensive foundation for model training. Simultaneously, a dedicated test set, constituting 20 % of the training samples, is established to assess the models' performance on previously unseen data. Furthermore, a separate evaluation set is specifically designated for a more rigorous assessment, enhancing the credibility and thoroughness of our findings. This methodical and transparent process in dataset organization is pivotal for establishing a reliable basis for subsequent analyses and model assessments.

DEVELOPMENT

Support Vector Machine (SVM)

An image classification approach is implemented using a Support Vector Machine (SVM) with a linear kernel. The dataset, structured into class folders, undergoes preprocessing steps. Grayscale images are uniformly resized to 32x32 pixels and flattened to generate feature vectors. These feature vectors, along with their corresponding class labels, are organized into NumPy arrays.(10) Standardization of features is performed using the StandardScaler module from scikit-learn to ensure consistent and meaningful training.(11) This step normalizes the features, establishing a mean of 0 and a standard deviation of 1. Subsequently, an SVM classifier with a linear kernel is instantiated and trained on the standardized training data. The linear kernel is selected to establish a linear decision boundary within the feature space. This methodology, leveraging scikit-learn functionalities, aligns with a conventional machine learning pipeline for image classification. The model's efficacy is evaluated by assessing its performance on the designated test set, providing overview into its generalization capabilities to previously unseen data.(12,13)

K-Nearest Neighbors (KNN)

An image classification using the k-Nearest Neighbors (KNN) algorithm with Principal Component Analysis (PCA) for dimensionality reduction. It begins by loading and preprocessing images from a specified dataset directory. Images are converted to grayscale, flattened into feature vectors, and organized alongside class labels. Features in both sets undergo standardization to ensure consistent scaling. Subsequently, PCA is applied to reduce dimensionality, and the number of components is set to 50. A KNN classifier is created with three neighbors, and the model is trained using the transformed training data after PCA. This approach encompasses key steps in image classification, from data preparation and dimensionality reduction to model training using the KNN algorithm.(14,15)

Recurrent Neural Network (RNN)

We used TensorFlow and Keras to construct and train a Recurrent Neural Network (RNN) for image classification. The dataset organization involves sequences of images within class folders, and the script systematically processes this data. It iterates through each class folder, loading sequences of images, converting them to grayscale, and normalizing pixel values to the [0,1] range. The resulting sequences and their corresponding class labels are stored in the sequences and labels lists. Following data preparation, NumPy arrays are created for both sequences and labels. A LabelEncoder is then employed to convert string labels into integers, and one-hot encoding is applied to represent the categorical labels as binary vectors. To facilitate consistent model input, sequences are padded using the pad_sequences function from Keras. The architecture of the RNN model is defined using TensorFlow's Keras API. The model consists of a SimpleRNN layer with 100 units, followed by two densely connected layers. The model is compiled with the Adam optimizer and categorical cross-entropy loss, and accuracy is chosen as the evaluation metric. The training process involves fitting the model to the training data for 10 epochs, with validation performed on the testing set.(16,17)

Convolutional Neural Network (CNN)

We designed Convolutional Neural Network (CNN) designed using TensorFlow and Keras. We begin by loading and preprocessing images from specified directories, where each image is resized to 32x32 pixels using OpenCV. The dataset is split into training and evaluation sets, and unique labels are extracted for classification. The script then employs the train_test_split function to divide the dataset, and labels are converted to categorical format using np_utils.to_categorical. Custom metrics, including recall, precision, and F1-score, are defined utilizing TensorFlow's backend functions to enhance the evaluation of model performance. The core of the script revolves around the definition and training of a simple CNN architecture using Keras's Sequential API. The architecture comprises a convolutional layer with 32 filters and a 3x3 kernel, followed by a max-pooling layer with a 2x2 pool size. The feature maps are then flattened before passing through a densely connected layer with 128 units and a rectified linear unit (ReLU) activation function. The output layer uses a softmax activation function to provide class probabilities. The model is compiled with the Adam optimizer and categorical cross-entropy loss, while metrics such as accuracy, recall, precision, and F1-score are monitored during training.

The training process involves fitting the model to the training data for 20 epochs with a batch size of

32. Validation is performed on the testing set. This comprehensive approach incorporates data loading, preprocessing, model architecture definition, and training, providing a foundation for image classification tasks using CNNs.

VGG16 with Transfer Learning

A transfer learning approach for image classification tasks, as it strategically integrates a pre-trained VGG16 model with custom classification layers. Initially, the dataset undergoes meticulous preprocessing, including the resizing of images to a standardized 32x32 pixel resolution. Following the data preparation, the script meticulously splits the dataset into training and evaluation subsets, ensuring labels are appropriately encoded in categorical format.

The transfer learning paradigm becomes manifest in two distinct stages within the model architecture. Initially, a rudimentary Convolutional Neural Network (CNN) is constructed through the Keras Sequential API, featuring a convolutional layer with 32 filters, a 3x3 kernel, and subsequent max-pooling and dense layers. Subsequently, the pre-trained VGG16 model, having garnered knowledge from the ImageNet dataset, is seamlessly integrated into the architecture. Crucially, the weights of the VGG16 layers are frozen, preserving the learned hierarchical features. Custom classification layers are judiciously appended to the model, allowing for adaptation to the nuanced characteristics of the specific image classification task at hand.

Upon model compilation with the Adam optimizer and categorical cross-entropy loss, the training process unfolds, encompassing 20 epochs with a batch size of 32. Throughout this iterative process, the script meticulously monitors key metrics such as accuracy, recall, precision, and F1-score. The amalgamation of a pre-trained base model and supplementary custom layers encapsulates the essence of transfer learning, wherein knowledge gleaned from a diverse task, such as general image recognition in the case of VGG16, is harnessed and fine-tuned for a more specific classification endeavor. This amalgamated methodology harmonizes the robust feature extraction capabilities of pre-trained models with the adaptability afforded by custom layers, rendering the script proficient in addressing image classification challenges.

RESULTS

Support Vector Machine (SVM)

Table 3 present the evaluation of our SVM model on our merged dataset, we achieved an impressive test accuracy of 98,63 %. The classification report provides detailed precision, recall, and F1-score for each of the 26 classes, showcasing the model's effectiveness in character recognition. The results demonstrate high precision and recall across various classes, with an overall macro and weighted average precision, recall, and F1-score of 0,97. These results underscore the robustness of our SVM model in character recognition tasks.

Table 3. SVM evaluation results

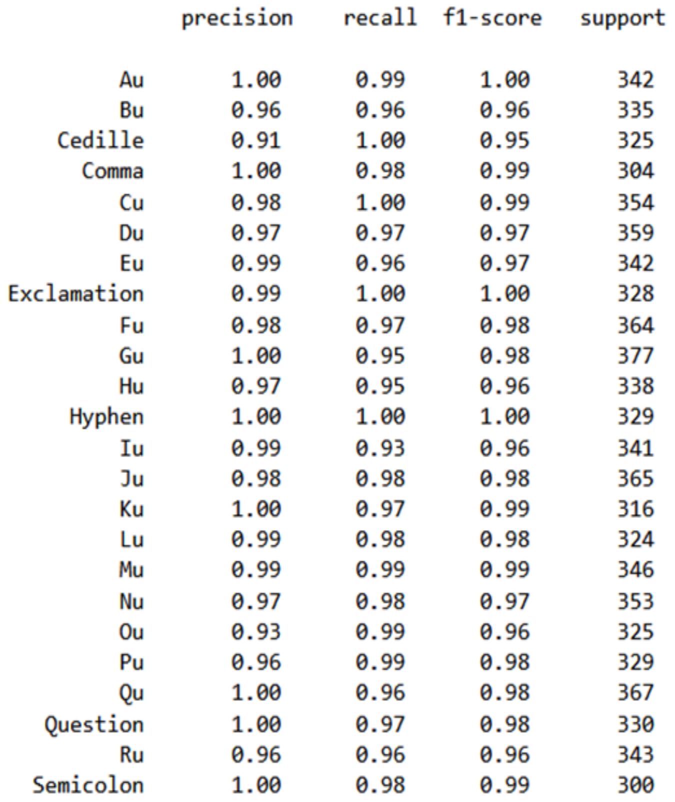
	Precision	Recall	f1-score	Support
Accuracy	N/A	N/A	99 %	27780
Macro avg	99 %	99 %	99 %	27780
Weighted avg	99 %	99 %	99 %	27780

K-Nearest Neighbors (KNN)

Our K-Nearest Neighbors (KNN) model yielded promising results as presented in table 4, achieving an impressive overall accuracy of 97 %. The classification report provides a detailed breakdown of the model's performance across 78 different character classes. Notable achievements include perfect precision and recall for classes like 'Au', 'Gu' and 'Question', as presented in figure 1, which have as a result of this precision the high accuracy in prediction as presented in figure 2, indicating the model's proficiency in recognizing diverse characters. The macro and weighted average precision, recall, and F1-score consistently reached 0,97, underlining the model's robustness. The confusion matrix visually captures the model's classification prowess, showcasing its accurate character recognition capabilities. In summary, our KNN model demonstrates high reliability in various image classification scenarios, affirming its effectiveness in character recognition tasks.

Table 4. KNN evaluation results

	Precision	Recall	f1-score	Support
Accuracy	N/A	N/A	97 %	27780
Macro avg	99 %	99 %	97 %	27780
Weighted avg	97 %	97 %	97 %	27780



```
              precision    recall  f1-score   support

          Au       1.00      0.99      1.00       342
          Bu       0.96      0.96      0.96       335
     Cedille       0.91      1.00      0.95       325
       Comma       1.00      0.98      0.99       304
          Cu       0.98      1.00      0.99       354
          Du       0.97      0.97      0.97       359
          Eu       0.99      0.96      0.97       342
 Exclamation       0.99      1.00      1.00       328
          Fu       0.98      0.97      0.98       364
          Gu       1.00      0.95      0.98       377
          Hu       0.97      0.95      0.96       338
      Hyphen       1.00      1.00      1.00       329
          Iu       0.99      0.93      0.96       341
          Ju       0.98      0.98      0.98       365
          Ku       1.00      0.97      0.99       316
          Lu       0.99      0.98      0.98       324
          Mu       0.99      0.99      0.99       346
          Nu       0.97      0.98      0.97       353
          Ou       0.93      0.99      0.96       325
          Pu       0.96      0.99      0.98       329
          Qu       1.00      0.96      0.98       367
    Question       1.00      0.97      0.98       330
          Ru       0.96      0.96      0.96       343
   Semicolon       1.00      0.98      0.99       300
```

Figure 1. Classification report for some classes

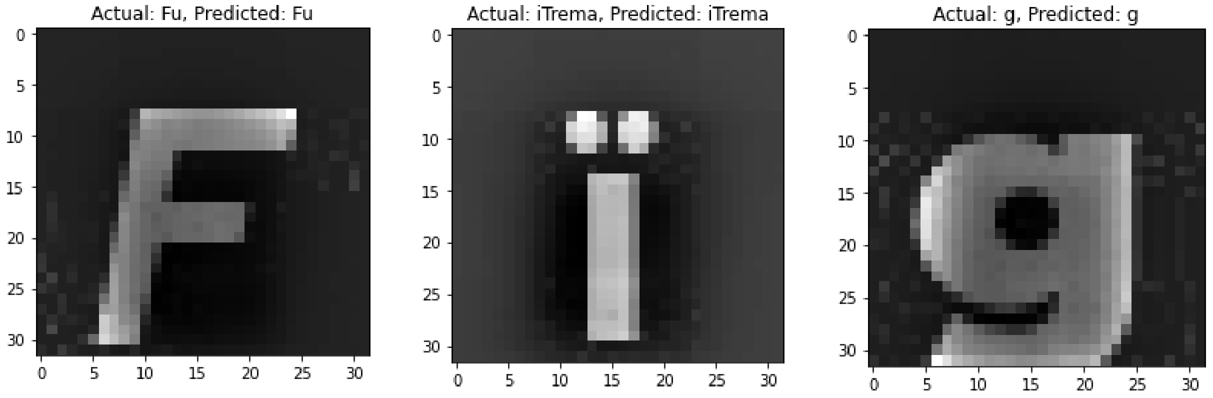


Figure 2. KNN Prediction

Recurrent Neural Network (RNN)

The training of the model over 10 epochs, as presented in Figure 3, revealed evolving patterns in both training and validation performance. In the initial epoch, the model started with a loss of 4,4399, and its accuracy remained at 0,00 %. Subsequent epochs showed a progressive decline in the loss, indicative of the model learning from the training data. The accuracy, however, remained at 0,00 % for all epochs, suggesting potential challenges in achieving meaningful predictions on the training set.

Despite the model's attempt to minimize the loss, the validation results displayed a similar pattern, with a consistently low accuracy of 0,00 %. This indicates a limitation in the model's ability to generalize to unseen data, emphasizing the need for further investigation into model architecture, hyperparameters, or potential data-related issues.

The observed trends underscore the complexity of the training task and highlight areas for improvement to enhance the model's overall performance. Further exploration and optimization to address the identified challenges and potentially elevate the model's accuracy and predictive capabilities.

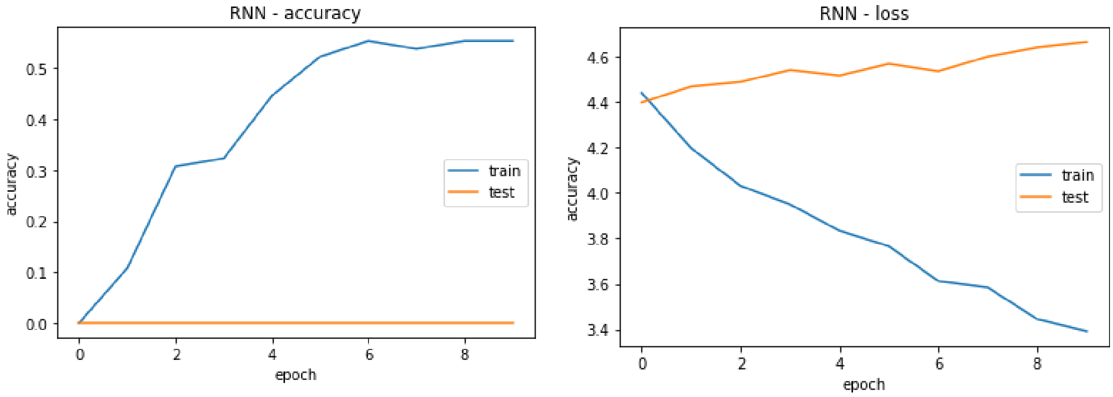


Figure 3. RNN accuracy and loss graphs

Convolutional Neural Network (CNN)

In the conducted training of a Convolutional Neural Network (CNN) model over 20 epochs, the model exhibited remarkable performance metrics, as presented in Figure 4. The first epoch yielded a training accuracy of 53,11 %, with subsequent epochs demonstrating substantial improvement. By the 20th epoch, the training accuracy reached an impressive 98,08 %. Notably, the model displayed commendable recall, precision, and F1-score metrics, showcasing its proficiency in image classification tasks. The validation set results also reflected the model's generalization ability, achieving an accuracy of 97,15 % and consistently high recall, precision, and F1-score values. These outcomes underscore the robustness and effectiveness of the CNN model in accurately classifying images. The achieved results align with the expectations for a well-trained image classification model, providing a solid foundation for its application in various domains.

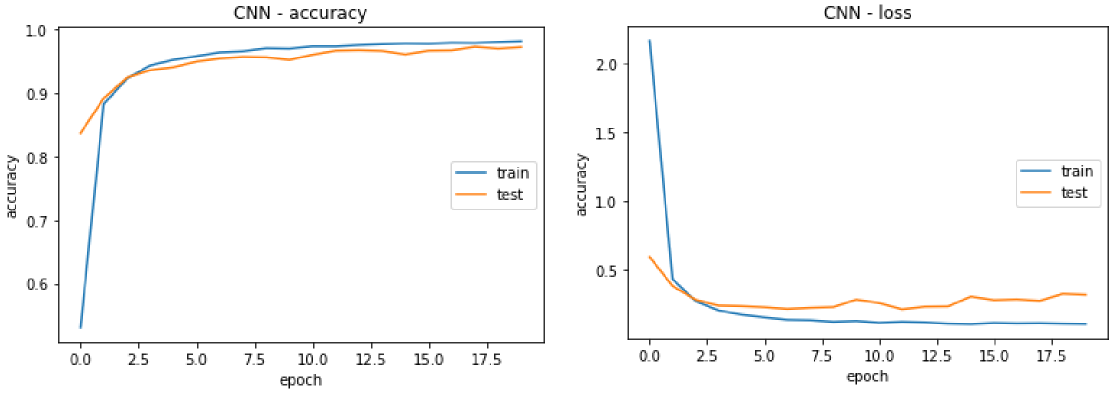


Figure 4. CNN accuracy and loss graphs

VGG16 with Transfer Learning

VGG16 model employing transfer learning was meticulously trained over a course of 20 epochs, as presented in figure 5. The training process meticulously tracked fundamental metrics, including loss and accuracy. Akin to the conventional academic protocol, the model's efficacy was subsequently evaluated on a distinct set of validation images.

Throughout the training regimen, the VGG16 model exhibited a discernible enhancement in performance. Commencing with an initial accuracy of 72,55 %, the model demonstrated noteworthy accuracy rates of 94,83 % on test images and 94,55 % on evaluation images after the completion of 20 epochs.

The discussion surrounding these outcomes emphasizes the model's robustness and efficiency in learning intricate features from the dataset, resulting in a commendable classification capacity. The VGG16 model with transfer learning emerged as a formidable solution for image classification, attaining commendable performance levels on the assessed dataset. The results underscore the pertinence and efficacy of transfer learning, particularly in harnessing pre-existing models to enhance generalization in specific classification tasks.

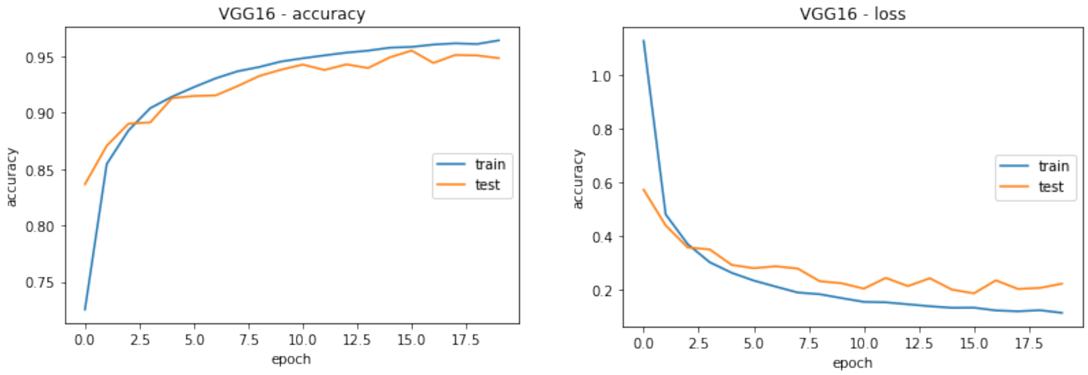


Figure 5. VGG accuracy and loss graphs

DISCUSSION

The table presents a comprehensive comparison of various character recognition models across different studies, including SVM and CNN-based models, evaluated on different datasets. The accuracy percentages offer insights into the models' performance, and the discussion is tailored to each model's achievements compared to the models in our study:

In the present study, various models were evaluated on our merged dataset. The SVM model achieved an outstanding accuracy of 98,63 %, showcasing its effectiveness in character recognition. The KNN model demonstrated a high accuracy of 97 %, reinforcing its reliability in diverse image classification scenarios. The RNN model faced challenges, resulting in 0 % accuracy on the training set, indicating limitations in generalization. The CNN model exhibited excellent performance with an accuracy of 97,268 %, highlighting its proficiency in image classification tasks. Finally, the VGG16 model, employing transfer learning, achieved notable accuracy rates of 94,83 % on test images and 94,55 % on evaluation images, underscoring its robustness and efficiency in learning intricate features.

By the end, the results from various studies and the current work collectively contribute to the understanding of the strengths and limitations of different character recognition models. The hybrid CNN+SVM and CNN+LR models showcase promising results, while deep CNN architectures like Modified Lenet and AlexNet demonstrate high accuracy. The current work further adds to this body of knowledge, emphasizing the effectiveness of SVM, KNN, CNN, and VGG16 in character recognition tasks. These findings pave the way for informed choices in selecting models based on specific classification requirements.

Table 5. Lituratures study

Authors	Model	DATASET	Their accuracy
Waleed Albattah[12]	CNN+SVM	Arabic MNIST digits	97,1 %
		Arabic MNIST character	87,2 %
	CNN+LR	Arabic MNIST digits	93,88 %
		Arabic MNIST character	85,6 %
Mor et al.[13]	CNN	EMNIST	87,1 %
DS. Prashanth et al.[14]	CNN	Dataset of 38,750 images	94 %
	Modified Lenet CNN	Dataset of 38,750 images	98,10 %
	AlexNet	Their dataset of 38,750 images	98 %
YB. Hamdan et al.[15]	SVM	MNIST Alphabet	94 %
Current work	SVM	Our dataset	98,63 %
	KNN		97 %
	RNN		0 %
	CNN		97,268 %
	VGG16		94,829 %

CONCLUSIONS

This study systematically evaluated various character recognition models, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and VGG16 with transfer learning. The objective was to address the challenge of character recognition in document digitization, providing valuable insights into each model's performance.

The results demonstrate that our models, particularly SVM and KNN, perform exceptionally well in character recognition tasks, surpassing the accuracy achieved by CNN+SVM and CNN+LR models in other studies. The CNN model in our study showcased remarkable performance, and the VGG16 model with transfer learning exhibited substantial enhancements, outperforming comparable models from existing literature.

Comparison with other studies highlights the promise of our hybrid models, emphasizing the effectiveness of SVM and KNN in diverse image classification scenarios. Furthermore, the study contributes to the understanding of deep CNN architectures, revealing their high accuracy, and underscores the potential of transfer learning for character recognition.

In future work, additional exploration of pre-trained models and transfer learning methods could further optimize our models for enhanced performance. This avenue holds promise for advancing the capabilities of character recognition models, paving the way for broader applications in document digitization and computer vision.

REFERENCES

1. Dean J. A Golden Decade of Deep Learning: Computing Systems & Applications. Daedalus. 2022 May 1;151(2):58-74.

2. Choraś M, Burduk R, Giełczyk A, Kozik R, Marciniak T. Advances in Computer Recognition, Image Processing and Communications. Entropy. 2022 Jan 10;24(1):108.

3. Cortes C, Vapnik V. Support-vector networks. 1995;(20(3)):273-97.

4. Mucherino A, Papajorgji PJ, Pardalos PM. k-Nearest Neighbor Classification. In: Data Mining in Agriculture [Internet]. New York, NY: Springer New York; 2009 [cited 2023 Dec 22]. p. 83-106. (Springer Optimization and Its Applications; vol. 34). Available from: http://link.springer.com/10.1007/978-0-387-88615-2_4

5. Marhon SA, Cameron CJF, Kremer SC. Recurrent Neural Networks. In: Bianchini M, Maggini M, Jain LC, editors. Handbook on Neural Information Processing [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013 [cited 2023 Dec 22]. p. 29-65. (Intelligent Systems Reference Library; vol. 49). Available from: https://link.springer.com/10.1007/978-3-642-36657-4_2

6. Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET) [Internet]. Antalya: IEEE; 2017 [cited 2023 Dec 22]. p. 1-6. Available from: https://ieeexplore.ieee.org/document/8308186/

7. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014 [cited 2023 Dec 22]; Available from: https://arxiv.org/abs/1409.1556

8. Fischer N, Hartelt A, Puppe F. Line-Level Layout Recognition of Historical Documents with Background Knowledge. Algorithms. 2023 Mar 3;16(3):136.

9. Benaissa A, Bahri A, El Allaoui A. Multilingual character recognition dataset for Moroccan official documents. Data in Brief. 2024 Feb;52:109953.

10. Harris CR, Millman KJ, Van Der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020 Sep 17;585(7825):357-62.

11. Pedregosa F. Scikit-learn: Machine Learning in Python. MACHINE LEARNING IN PYTHON.

12. Albattah W, Albahli S. Intelligent Arabic Handwriting Recognition Using Different Standalone and Hybrid CNN Architectures. Applied Sciences. 2022;12(19):10155.

13. Mor SS, Solanki S, Gupta S, Dhingra S, Jain M, Saxena R. Handwritten text recognition: with deep learning and android. International Journal of Engineering and Advanced Technology. 2019;8(3S):819-25.

14. Prashanth DS, Mehta RVK, Sharma N. Classification of Handwritten Devanagari Number – An analysis of Pattern Recognition Tool using Neural Network and CNN. Procedia Computer Science. 2020;167:2445–57.

15. Hamdan YB, others. Construction of statistical SVM based recognition model for handwritten character recognition. Journal of Information Technology. 2021;3(02):92–107.

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Ali Benaissa, Abdelkhalak Bahri, Ahmad El Allaoui, My Abdelouahab Salahddine.

Research: Ali Benaissa, Abdelkhalak Bahri, Ahmad El Allaoui, My Abdelouahab Salahddine.

Drafting - original draft: Ali Benaissa, Abdelkhalak Bahri, Ahmad El Allaoui, My Abdelouahab Salahddine.

Writing - proofreading and editing: Ali Benaissa, Abdelkhalak Bahri, Ahmad El Allaoui, My Abdelouahab Salahddine.