Detectia Université BOUIRA

ID: bl2yn6-67996

Rapport d'analyse de la similarité textuelle

- Nom du document: **MEMIORE__test2_MEZIANE AHLEME.pdf**
- Soumis par: **Djouabi Abderezak**
- Faculté: -
- Date de soummission: **2024-06-19**

25.5%

Taux global de similarité

25.5% Similarité Forte
0.0% Similarité Proche
0.0% Exclu manuellement

131

Nombre de sources

131 sources internet
0 sources Thèses-Algérie
0 sources dépôtprivé

126

Passages surlignés

16800 mots
133326 caractères

Ce document est un rapport d'analyse et de détection de similarité textuelle qui peut fournir des preuves pour l'établissement d'un rapport de plagiat. Il revient à l'examinateur, l'encadrant ou bien au comité déontologique de l'université ou de l'école d'émettre un avis quant au statut de plagiat du document analysé.

◎ Consultez **l'arrêté N° 1082 du 27 Décembre 2020 fixant les règles relatives à la prévention et la lutte contre le plagiat** pour en savoir plus concernant ce qui est considéré comme étant un acte de plagiat, les procédures ainsi que les sanctions.

Taille minimale des passages: **15 mots.** Niveau de tolérance de la similarité: **Majeur**

Signature d'intégrité

📚 Sources identifiées

Cette section fournit une liste des principales sources à partir desquelles une similarité textuelle a été détectée avec le document analysé. Ces sources peuvent provenir d'internet, la base Thèses-Algérie, ou bien des bases de données propriétaires.

◉ Les scores de similarité individuels pour chaque source sont fournis. Les sources sont classées par ordre décroissant de similarité globale.

◉ Vous pouvez cliquer sur le lien (🔗) d'une source pour la consulter en ligne.

Origine	Détails	Résultats
Page web	**Mastering Data Cleaning & Data Preprocessing** 🔗 https://encord.com/blog/data-cleaning-data-preprocessing/ Data cleaning involves identifying and correcting errors in the dataset, such as dealing with missing or inconsistent data, removing duplicates, and ...	**Sim. globale: 4.6%** **Sim. forte: 4.6%** **Sim. proche: 0%**
Page web	**F1 Score in Machine Learning Explained** 🔗 https://encord.com/blog/f1-score-in-machine-learning/ Jul 18, 2023 — The F1 score or F-measure is described as the harmonic mean of the precision and recall of a classification model. The two metrics contribute ...	**Sim. globale: 4.4%** **Sim. forte: 4.4%** **Sim. proche: 0%**
Fichier web	**A detailed analysis of optical character recognition technology** 🔗 https://dergipark.org.tr/en/download/article-file/236939 ... to first scan the documents and then store them as image files... The segmentation process is crucial as it converts the image ... and an official choice relies on the complicated collaboration ...	**Sim. globale: 4.4%** **Sim. forte: 4.4%** **Sim. proche: 0%**
Page web	**Fine-tuning Models: Hyperparameter Optimization** 🔗 https://encord.com/blog/fine-tuning-models-hyperparameter-optimization/ Aug 22, 2023 — At its core, this process systematically searches for the best set of hyperparameters to elevate a model's performance. But what distinguishes ...	**Sim. globale: 4.3%** **Sim. forte: 4.3%** **Sim. proche: 0%**
Fichier web	**Genetic algorithms** 🔗 https://link.springer.com/content/pdf/10.1007/978-3-540-73190-0_2.pdf ... , natural selection plays a major role in this survival process. ... ones selected from the last generation, the genetic algorithm is ... The principle of GAs is simple: imitate genetics and natural ...	**Sim. globale: 4.2%** **Sim. forte: 4.2%** **Sim. proche: 0%**
Fichier web	**Adaptive methods for robust document image understanding** 🔗 https://core.ac.uk/download/pdf/304641877.pdf ...), the effects of time and/or imperfect storage. Chapter 2 will ... page may have been rotated or skewed at an arbitrary angle on ... In chapter 4 we address the issue of document skew and ...	**Sim. globale: 3.2%** **Sim. forte: 3.2%** **Sim. proche: 0%**
Fichier web	**Convolutional Networks II** 🔗 https://deeplearning.cs.cmu.edu/F20/document/slides/lec11.CNN.pdf • Each pooling layer with stride decreases the size of the maps by a factor of ... – Pool2: 2x2 max pooling with stride 2 for second conv layer. • Result 16 5x5 ...404 pages	**Sim. globale: 3.1%** **Sim. forte: 3.1%** **Sim. proche: 0%**

Type	Source	Similarité
Fichier web	**Convolutional neural networks: A survey** https://www.mdpi.com/2073-431X/12/8/151 ... or average value of non-overlapping regions. Pooling helps to ... operation in CNNs that involves sliding a filter or kernel over an ... by convolving a set of filters with the input image. Each ...	**Sim. globale: 2.7%** **Sim. forte: 2.7%** **Sim. proche: 0%**
Fichier web	**Evolving Deep Neural Networks with Explanations for Image...** ... ... of-the-art accuracy in their eras, respectively. Furthermore, as ... EC method to automatically design CNN architectures, and ... As a result, in the next iteration, the controller will give higher ...	**Sim. globale: 2.7%** **Sim. forte: 2.7%** **Sim. proche: 0%**
Fichier web	**Ensemble Optimization for Histological Image Classication** https://search.proquest.com/openview/264d58963409eb7daf8a23cbddd0be3b/1?pq-... ... a noisystudent model using hyper-parameters obtained from the ... well on benchmark image datasets such as MNIST and CIFAR, ... We use the confusion matrix of the models over the ...	**Sim. globale: 2.6%** **Sim. forte: 2.6%** **Sim. proche: 0%**
Fichier web	**Handwritten digit recognition of MNIST dataset using deep...** https://ieeexplore.ieee.org/abstract/document/9396870/ ... major Deep learning algorithms Artificial Neural Network and ... -entropy loss and ADAM optimizer on the MNIST dataset. ... The preprocessing part is done by the convolutional layer by ...	**Sim. globale: 2.5%** **Sim. forte: 2.5%** **Sim. proche: 0%**
Fichier web	**Deep learning feature extraction for image processing thesis** https://www.researchgate.net/profile/Baptiste-... ... La première, la reconnaissance de nombres écrits à la main, est ... Chapitre 1 Introduction Ce chapitre présente le contexte de la ... from which the exact normalization constant can be ...	**Sim. globale: 2.5%** **Sim. forte: 2.5%** **Sim. proche: 0%**
Fichier web	**1 INTRODUCTION TO COMPUTER** http://gpnanakpur.ac.in/wp-content/uploads/2019/04/computerfundamentals-e-notes.pdf This chapter discusses the history and evolution of computer, the concept of input-process-output and the characteristics of computer. This chapter also ...628 pages	**Sim. globale: 2.5%** **Sim. forte: 2.5%** **Sim. proche: 0%**
Fichier web	**Swarm intelligence in optimal management of aquaculture farms** https://link.springer.com/content/pdf/10.1007/978-1-4939-2483-7.pdf#page=232 ... to the pheromone model and heuristic information. The heuristic information is represented by a new matrix (ηij) ... where α and β are two positive parameters whose values determine the ...	**Sim. globale: 2.2%** **Sim. forte: 2.2%** **Sim. proche: 0%**
Fichier web	**Ant colony optimization: artificial ants as a computational ...** https://typeset.io/pdf/ant-colony-optimization-artificial-ants-as-a-computational-4d2lxwbp0x.pdf 1) Ant System (AS): It is the first ACO algorithm proposed in the literature [6]–[8]. Its main characteristic is that, at each iteration, the pheromone ...14 pages	**Sim. globale: 2.1%** **Sim. forte: 2.1%** **Sim. proche: 0%**
Page web	**Introduction to Deep Learning** https://www.geeksforgeeks.org/introduction-deep-learning/ May 26, 2024 — Deep learning AI can be used for supervised, unsupervised as well as reinforcement machine learning. it uses a variety of ways to process these.	**Sim. globale: 2.0%** **Sim. forte: 2.0%** **Sim. proche: 0%**

Fichier web	**Facial expression recognition using Deep Convolutional Neural...** 🔗 https://dione.lib.unipi.gr/xmlui/handle/unipi/12567 ... Fully Connected layers are denoted as dense followed by the number ... We can see that the ReLU activation function culls the ... By using dropout, we randomly alter the architecture of the ...	**Sim. globale: 2.0%** **Sim. forte: 2.0%** **Sim. proche: 0%**
Fichier web	**Digits Recognition System using Machine Learning** 🔗 http://www.ir.juit.ac.in:8080/jspui/handle/123456789/9891 ... and categorise digits, the digit recognition system leverages ... Google Colab: - Google Colab was created by Google to give ... In this project, CNN (Convolutional Neural Network) model is ...	**Sim. globale: 1.7%** **Sim. forte: 1.7%** **Sim. proche: 0%**
Fichier web	**Ant Colony Optimization: a literature survey** 🔗 https://repositorio-aberto.up.pt/bitstream/10216/70687/2/40116.pdf ... is given by the pheromone quantity τij. The transition rule ... ants can deposit pheromone in the arcs of their solution, the ... importance of the pheromone values from one iteration to the ...	**Sim. globale: 1.7%** **Sim. forte: 1.7%** **Sim. proche: 0%**
Fichier web	**Recognition of Japanese handwritten characters with Machine...** 🔗 https://rua.ua.es/dspace/handle/10045/109318 ... The smoothing implies both filling and thinning. For this task ... creation of a new dataset following the MNIST format. Their most ... preprocessing, making it easy to apply for real-world data (...	**Sim. globale: 1.6%** **Sim. forte: 1.6%** **Sim. proche: 0%**
Fichier web	**Hyperparameter optimization for 1D-CNN-based network intrusion...** 🔗 https://www.mdpi.com/2227-7390/11/17/3724 ... contribute to achieving higher model performance metrics, thereby ... precision and recall. Finally, on the NSL-KDD dataset, the GA-optimized model achieves marginally better accuracy ...	**Sim. globale: 1.6%** **Sim. forte: 1.6%** **Sim. proche: 0%**
Fichier web	**A survey of methods for explaining black box models** 🔗 https://dl.acm.org/doi/abs/10.1145/3236009 ... In DNNs the data typically flows from the input to the output layer ... function through their interaction with other features. Moreover, MFI ... influence the outputs of black box predictors. These ...	**Sim. globale: 1.6%** **Sim. forte: 1.6%** **Sim. proche: 0%**
Fichier web	**Artificial neural networks based optimization techniques: A...** 🔗 https://www.mdpi.com/2079-9292/10/21/2689 ... neural networks are a family of deep learning (DL) and ML ... CNN is the most widely used type of neural network for ... regressions for ANN deep neural network after using the optimization ...	**Sim. globale: 1.6%** **Sim. forte: 1.6%** **Sim. proche: 0%**
Fichier web	**Handwritten word recognition: Application to Arabic cheque...** 🔗 https://spectrum.library.concordia.ca/id/eprint/1739/ ... and analytical classification methods to recognize Arabic sub-... Corporation (one of the largest banking corporations in ... , though they do exist in the lexicon of handwritten Arabic legal ...	**Sim. globale: 1.6%** **Sim. forte: 1.6%** **Sim. proche: 0%**

Fichier web	**Optimizing convolutional neural network parameters using...** https://mspace.lib.umanitoba.ca/handle/1993/34684 ... epochs (each epoch refers to a full pass through the training ... the number of batch sizes with increasing number of iterations, ... the batch size (128 in our case) or increase the number of ...	**Sim. globale: 1.5%** Sim. forte: 1.5% Sim. proche: 0%
Fichier web	**Optical character recognition systems** https://link.springer.com/chapter/10.1007/978-3-319-50252-6_2 ... between two points, isolated dots, a bend between two points, ... Direct matching: A gray level or binary input character is ... , the recognition rate of this method is very sensitive to noise. ...	**Sim. globale: 1.5%** Sim. forte: 1.5% Sim. proche: 0%
Fichier web	**Automatic Design of Deep Neural Network Architectures with...** https://search.proquest.com/openview/bae423e665dd25579dc445518578ae33/1?pq-... ... were developed using the concepts from genetic algorithms, ... (N); the total number of iterations that the algorithm will run (... CNN model is trained for a certain number of epochs using ...	**Sim. globale: 1.5%** Sim. forte: 1.5% Sim. proche: 0%
Fichier web	**Neural Network - an overview** https://www.sciencedirect.com/topics/physics-and-astronomy/neural-network When applied to forecasting, neural networks can be regarded as a nonlinear black box (input-output) model. A neural network is simply a set of interconnected ...	**Sim. globale: 1.4%** Sim. forte: 1.4% Sim. proche: 0%
Fichier web	**A systematic review of hyperparameter optimization techniques...** https://www.sciencedirect.com/science/article/pii/S2772662224000742 ...), and M-3 (PSO-MLP), with M-1 producing the most favorable ... of our study was to examine datasets utilized in CNN ... By optimizing the parameters of CNN models, these algorithms ...	**Sim. globale: 1.4%** Sim. forte: 1.4% Sim. proche: 0%
Fichier web	**Improving multilayer perceptron classifiers AUC performance.** https://paginas.fe.up.pt/~niadr/PUBLICATIONS/LIACC_publications_2011_12/pdf/TD9_Pollan.pdf ... TPR and FPR levels which can then be plotted in a ROC ... FPR-TPR space, such as the dashed lines shown in figure 3. ... a random guess and we seek classifiers above the diagonal and ...	**Sim. globale: 1.3%** Sim. forte: 1.3% Sim. proche: 0%
Fichier web	**A comprehensive survey on optimizing deep learning models by...** https://link.springer.com/article/10.1007/s10462-021-09992-0 ... distinct initialization methods, and activation functions. The ... aspects of DL models (training or hyper-parameter optimization) or ... inputs to prevent overfitting during training. The reshape ...	**Sim. globale: 1.3%** Sim. forte: 1.3% Sim. proche: 0%
Fichier web	**How to Develop a CNN for MNIST Handwritten Digit ...** https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-... Nov 14, 2021 — How to develop a test harness to develop a robust evaluation of a model and establish a baseline of performance for a classification task.	**Sim. globale: 1.3%** Sim. forte: 1.3% Sim. proche: 0%

Fichier web	**Statistical pattern recognition: A review** https://ieeexplore.ieee.org/abstract/document/824819/ ... from statistical decision theory are utilized to establish decision ... use the optimal Bayes decision rule (with a 0/1 loss function) ... found the maximum likelihood estimate ˆmˆm of m and used ...	**Sim. globale: 1.3%** **Sim. forte: 1.3%** **Sim. proche: 0%**
Page web	**The Complete Guide to Image Preprocessing Techniques ...** https://medium.com/@maahip1304/the-complete-guide-to-image-preprocessing-techniques-in-python-... Preprocessing is a crucial first step to prepare your image data before feeding it into machine learning models. There are several techniques ...	**Sim. globale: 1.2%** **Sim. forte: 1.2%** **Sim. proche: 0%**
Page web	**A Basic Introduction to Convolutional Neural Network** https://medium.com/@himadrisankarchatterjee/a-basic-introduction-to-convolutional-neural-... Examples include time-series data, which can be thought of as a 1D grid taking samples at regular time intervals, and image data, which can be ...	**Sim. globale: 1.2%** **Sim. forte: 1.2%** **Sim. proche: 0%**
Fichier web	**Deep learning black box problem** https://www.diva-portal.org/smash/record.jsf?pid=diva2:1353609 ... return outputs, or make decisions, by running internal ... can be simple neural network models with one hidden layer, ... a continuous interaction of “online learning an input-output mapping ...	**Sim. globale: 1.2%** **Sim. forte: 1.2%** **Sim. proche: 0%**
Fichier web	**ConfusionVis: Comparative evaluation and selection of multi-...** https://www.sciencedirect.com/science/article/pii/S0950705122002982 ... are tuned, eg the convolutional neural network VGG-16 [18] ... data set consists of 100 data points of class C 1 , 200 of C 2 , ... in fact not misclassify any data points other than C 1 → C 3 (...	**Sim. globale: 1.2%** **Sim. forte: 1.2%** **Sim. proche: 0%**
Fichier web	**A framework for evolutionary optimization with approximate...** https://ieeexplore.ieee.org/abstract/document/1041556/ ... criterion is used to determine when the approximate model must be ... values lie within the respective error bounds of the other ... without affecting the correct convergence of the algorithm. ...	**Sim. globale: 1.2%** **Sim. forte: 1.2%** **Sim. proche: 0%**
Fichier web	**Improved handwritten digit recognition using convolutional...** https://www.mdpi.com/1424-8220/20/12/3344 ... evaluate various SGD optimization algorithms in improving ... a recognition accuracy of 99.87% for a MNIST dataset. ... to avoid complex pre-processing, costly feature extraction and a ...	**Sim. globale: 1.2%** **Sim. forte: 1.2%** **Sim. proche: 0%**
Fichier web	**The basic ideas in neural networks** ... ... output value, a set of input connections, a bias value, and a ... the effect of the incoming input on the unit's activation level. ... the neuron functions and how neurons communicate with one ...	**Sim. globale: 1.1%** **Sim. forte: 1.1%** **Sim. proche: 0%**

Fichier web	**A survey of convolutional neural networks: analysis,...** https://ieeexplore.ieee.org/abstract/document/9451544/ ... When setting a convolution kernel with a certain size, we will ... Besides, to control the density of convolving, the stride is ... convolution kernels (such as 1 × 1, 1×n, and n×1 convolution ...	**Sim. globale: 1.1%** **Sim. forte: 1.1%** **Sim. proche: 0%**
Fichier web	**A Gentle Introduction to Object Recognition With Deep ...** https://machinelearningmastery.com/object-recognition-with-deep-learning/ Jan 27, 2021 — Input: An image with a single object, such as a photograph. Output: A class label (e.g. one or more integers that are mapped to class labels).	**Sim. globale: 1.1%** **Sim. forte: 1.1%** **Sim. proche: 0%**
Fichier web	**Particle swarm optimization** https://link.springer.com/chapter/10.1007/978-3-642-37846-1_3 ... (particles or living organisms) in a bird flock or fish school [2], ... tribe is generated and initialized with its first particle, and so on. ... The PSO process can then start with a random initialization ...	**Sim. globale: 1.1%** **Sim. forte: 1.1%** **Sim. proche: 0%**
Fichier web	**Deep neural networks for object detection** https://proceedings.neurips.cc/paper/5207-deep-neural-networks-for-object-detection ... but also precisely localizing objects of various classes. We ... rectangle in the image for which the presence of an object is ... For each scale, we apply the bounding box inference from Sec. ...	**Sim. globale: 1.1%** **Sim. forte: 1.1%** **Sim. proche: 0%**
Fichier web	**Classification and Variable Selection by Maximizing the Area...** https://repository.lib.ncsu.edu/bitstream/handle/1840.20/33224/etd.pdf?sequence=1 ... space and the estimation of AUCROC is denoted as ... The plot of the ROC curve represents the trade-off between true positive rate (TPR) and false positive rate (FPR), where TPR ...	**Sim. globale: 1.1%** **Sim. forte: 1.1%** **Sim. proche: 0%**
Page web	**What is Forward Propagation in Neural Networks?** https://www.geeksforgeeks.org/what-is-forward-propagation-in-neural-networks/ May 1, 2024 — Forward propagation is the process in a neural network where the input data is passed through the network's layers to generate an output. It ...	**Sim. globale: 1.0%** **Sim. forte: 1.0%** **Sim. proche: 0%**
Fichier web	**Genetic algorithms: principles and perspectives: a guide to GA...** ... ... The GA works on the Darwinian principle of natural selection.... 2, we review the basic principles of the genetic algorithm in ... , or per gene; in this book we shall use the gene-wise mutation ...	**Sim. globale: 1.0%** **Sim. forte: 1.0%** **Sim. proche: 0%**
Fichier web	**Comparison of data preprocessing approaches for applying deep...** https://www.mdpi.com/1424-8220/18/7/2146 ... [26] defined data preprocessing as including data cleaning, ... These methods transform the raw data segments into different type of representations, from which the deep learning models ...	**Sim. globale: 1.0%** **Sim. forte: 1.0%** **Sim. proche: 0%**
Page web	**Activation functions in Neural Networks** https://www.geeksforgeeks.org/activation-functions-neural-networks/ May 3, 2024 — The purpose of the activation function is to introduce non-linearity into the output of a neuron. Explanation: We know, the neural network has ...	**Sim. globale: 0.9%** **Sim. forte: 0.9%** **Sim. proche: 0%**

Page web	**Classification and analysis of the MNIST dataset using PCA and...** https://cyberleninka.ru/article/n/classification-and-analysis-of-the-mnist-dataset-using-pca-and-... ... different individuals to properly evaluate the performance of ... Table 2 - Statistical distribution of the MNIST training dataset ... used the whole MNIST training dataset which contains 60,000 ...	**Sim. globale: 0.9%** **Sim. forte: 0.9%** **Sim. proche: 0%**
Page web	**Activation Functions in Neural Networks [12 Types & Use ...** https://www.v7labs.com/blog/neural-networks-activation-functions May 27, 2021 — A neural network activation function is a function that is applied to the output of a neuron. Learn about different types of activation ...	**Sim. globale: 0.9%** **Sim. forte: 0.9%** **Sim. proche: 0%**
Fichier web	**Statistics-based data preprocessing methods and machine...** https://www.aut.upt.ro/~rprecup/IJAI_59.pdf ... clean, noise free and consistent to model in various real life ... the performance of other two models, and while the data size ... effectively we can process the raw data and then analyze it to ...	**Sim. globale: 0.9%** **Sim. forte: 0.9%** **Sim. proche: 0%**
Fichier web	**Detecting dead weights and units in neural networks** https://arxiv.org/abs/1806.06068 ... we perform 10 epochs of training and the loss value is plotted ... , even though we start pruning after the second epoch. We can ... train a small CNN (Appendix C) with CIFAR10 [19] dataset. ...	**Sim. globale: 0.9%** **Sim. forte: 0.9%** **Sim. proche: 0%**
Fichier web	**Niching in Particole Swarm Optimization.** https://bee22.com/resources/PhDthesis_Passaro 2007.pdf ... Particle Swarm Optimization (PSO) algorithm, inspired by social behavior of bird flocking, fish schooling or ... ia critical value Ci (the criticality of the particle), which is initialized to Ci(0) = 0. ...	**Sim. globale: 0.9%** **Sim. forte: 0.9%** **Sim. proche: 0%**
Fichier web	**Prediction model using SMOTE, genetic algorithm and decision...** https://link.springer.com/article/10.1007/s00530-021-00817-2 ... model is trained, and its effectiveness is evaluated in the fourth layer in terms of classification accuracy (CA), classification error (... attributes [50, 51]. The computation procedure of C4.5 ...	**Sim. globale: 0.9%** **Sim. forte: 0.9%** **Sim. proche: 0%**
Fichier web	**Deep Learning - Object recognition** https://homel.vsb.cz/~pla06/files/dl/dl_11.pdf Input: An image with one or more objects, such as a photograph. • Output: One or more bounding boxes (e.g. defined by a point, width, and height). 1. Page 4 ...	**Sim. globale: 0.9%** **Sim. forte: 0.9%** **Sim. proche: 0%**
Fichier web	**Frequency domain methods in recurrent neural networks for...** https://bonndoc.ulb.uni-bonn.de/xmlui/handle/20.500.11811/9245 ... To move a modern recurrent neural network (RNN)-cell into the ... the FWT into convolutional and recurrent neural networks. ... Given a network output o and a ground truth value t, the ...	**Sim. globale: 0.8%** **Sim. forte: 0.8%** **Sim. proche: 0%**

Fichier web	**Image classification using machine learning techniques** https://search.proquest.com/openview/9f21050fd15c033f3463ef70c0b3a961/1?pq-... ... in this study, and different classifiers will be used to analyse ... in this chapter the different classifier methods are identified and ... referred to as the decision boundary can be represented as (...	**Sim. globale: 0.8%** **Sim. forte: 0.8%** **Sim. proche: 0%**
Fichier web	**Sentiment analysis with recurrent neural networks on turkish...** https://open.metu.edu.tr/handle/11511/43354 ... Supervised and unsupervised learning are the main ... backpropagation algorithm is known as Backpropagation Through ... such as Convolutional Neural Networks (CNNs), FastText for text ...	**Sim. globale: 0.8%** **Sim. forte: 0.8%** **Sim. proche: 0%**
Fichier web	**A framework for data-driven fault detection and identification...** http://studentsrepo.um.edu.my/9357/ ... discrimination analysis, in which multi-scale KFDA method ... , a kernel PCA (KPCA) has used kernel functions to complete ... In summary, SVM embodies many important principles. It has ...	**Sim. globale: 0.8%** **Sim. forte: 0.8%** **Sim. proche: 0%**
Page web	**Precision and recall** https://en.wikipedia.org/wiki/Precision_and_recall F-measure. measure, because recall and precision are evenly weighted. measure, which puts more emphasis on precision than recall.	**Sim. globale: 0.7%** **Sim. forte: 0.7%** **Sim. proche: 0%**
Fichier web	**a genetic algorithm based optimized convolutional neural ...** https://sciendo.com/pdf/10.34768/amcs-2023-0002 by N Karlupia · Cited by 6 — The experimental results indicate that the proposed GA-CNN model generates an improved model accuracy in comparison with existing CNN models. In ...	**Sim. globale: 0.7%** **Sim. forte: 0.7%** **Sim. proche: 0%**
Fichier web	**(PDF) Evaluation: From precision, recall and F-measure to ...** https://www.researchgate.net/publication/276412348_Evaluation_From_precision_recall_and_F-... Commonly used evaluation measures including Recall, Precision, F-Measure and Rand Accuracy ... The easiest performance metric to understand is accuracy, which is ...	**Sim. globale: 0.7%** **Sim. forte: 0.7%** **Sim. proche: 0%**
Fichier web	**Deep Convolutional Neural Network for Facial Expression** https://uhcl-ir.tdl.org/items/bbdb9ab5-5e71-4fc0-8214-76b8fde1c0cf ... in a fully connected layer which is used to learn global ... Given an input value x, The ReLU layer computes the output ... in a fully connected layer (FC) which is used to learn global ...	**Sim. globale: 0.7%** **Sim. forte: 0.7%** **Sim. proche: 0%**
Fichier web	**Deep convolutional neural networks for image classification: A...** https://ieeexplore.ieee.org/abstract/document/8016501/ ... on the backpropagtion algorithm and general training protocols for deep neural ... fully supervised fashion using backpropagation, which was in contrast to the unsupervised reinforcement ...	**Sim. globale: 0.7%** **Sim. forte: 0.7%** **Sim. proche: 0%**

Page web	**Receiver operating characteristic** https://en.wikipedia.org/wiki/Receiver_operating_characteristic The ROC curve is the plot of the true positive rate (TPR) against the false positive rate (FPR) at each threshold setting. The ROC can also be thought of as ...	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**
Page web	**OCR Pre-Processing Techniques | Image ...** https://medium.com/technovators/survey-on-image-preprocessing-techniques-to-improve-ocr-... Lines Straightening: When the lines are curvy as in the case of the above image, it may result in OCR issues and can cause issues with line ...	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**
Fichier web	**Handwritten optical character recognition (OCR): A...** https://ieeexplore.ieee.org/abstract/document/9151144/ ... 2) To highlight research weakness in order to eliminate them ... multipliers of a dual optimization problem that describe the ... challenges posed by the Nasta'liq style of Urdu handwriting. ...	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**
Fichier web	**Optimized Three Deep Learning Models Based-PSO ...** https://arxiv.org/pdf/2306.07296 by A Pranolo · 2023 · Cited by 12 — This research attempts to optimize the deep learning architecture of Long short term memory (LSTM), Convolutional neural network (CNN), and ...	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**
Fichier web	**Modified genetic algorithm for feature selection and hyper...** https://ieeexplore.ieee.org/abstract/document/9851666/ ... a novel GA variation that optimizes the parameters of a ... TF-iDF vectorizer parameters in Tweets text preprocessing are ... accuracy obtained using GA in our approach was 95.88% ...	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**
Fichier web	**Envision: reinventing the integrated development environment** https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/214522/eth-50652-02.pdf ... interfaces for software development, we designed and created the Envision open-source IDE... support elements such as embedded diagrams (editable directly in the IDE), tables, and ...	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**
Fichier web	**Indian script character recognition: a survey** https://www.sciencedirect.com/science/article/pii/S003132030400055X ... letter shapes which the OCRs read. Such machines ... was the IBM 1418, which was designed to read a special IBM font, 407 [2]. The recognition method was logical template matching ...	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**
Fichier web	**3679 Research Article Optimizing Convolu** https://turcomat.org/index.php/turkbilmat/article/download/6450/5342 by P Kalaiarasi · 2021 · Cited by 4 — The obtained results are observed to be excellent in both training and testing. In [21], an auto encoder was optimized using PSO algorithm.	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**
Fichier web	**Tour of Data Preparation Techniques for Machine Learning** https://machinelearningmastery.com/data-preparation-techniques-for-machine-learning/ Jun 30, 2020 — Data cleaning involves fixing systematic problems or errors in "messy" data. The most useful data cleaning involves deep domain expertise and ...	**Sim. globale: 0.6%** **Sim. forte: 0.6%** **Sim. proche: 0%**

Fichier web	**A brief survey of recent edge-preserving smoothing algorithms...** https://arxiv.org/abs/1503.07297 ... to smooth as non-linear Gaussian filter tends to work like a ... Wp is the normalization factor, σs and σr denotes the standard ... more on the use of bilateral filter to remove noise from digital ...	**Sim. globale: 0.6%** Sim. forte: 0.6% Sim. proche: 0%
Page web	**How to explain the ROC curve and ROC AUC score?** https://www.evidentlyai.com/classification-metrics/explain-roc-curve It plots the True Positive rate (TPR) against the False Positive rate (FPR). ... to 1, where 0.5 indicates random guessing, and 1 indicates perfect performance.	**Sim. globale: 0.5%** Sim. forte: 0.5% Sim. proche: 0%
Page web	**Convolutional Neural Networks (CNNs / ConvNets)** https://cs231n.github.io/convolutional-networks/ FC (i.e. fully-connected) layer will compute the class scores, resulting in ... In particular, the CONV/FC layers perform transformations that are a function ...	**Sim. globale: 0.5%** Sim. forte: 0.5% Sim. proche: 0%
Page web	**Confusion Matrix in Machine learning** https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/ The confusion matrix can be used to calculate a variety of metrics, such as accuracy, precision, recall, and F1 score.	**Sim. globale: 0.5%** Sim. forte: 0.5% Sim. proche: 0%
Page web	**7 steps of image pre-processing to improve OCR using ...** https://nextgeninvent.com/blogs/7-steps-of-image-pre-processing-to-improve-ocr-using-python-2/ 2. Skew Correction. While scanning or taking a picture of any document, it is possible that the scanned or captured image might be slightly skewed sometimes.	**Sim. globale: 0.5%** Sim. forte: 0.5% Sim. proche: 0%
Fichier web	**Genetic algorithms** https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119175391.ch19 ... especially in the GBML (Genetics Based Machine Learning) ... elements of an evolutionary algorithm, namely a population ... of nature's important principles employed by GAs is the opera...	**Sim. globale: 0.5%** Sim. forte: 0.5% Sim. proche: 0%
Fichier web	**Segmentation Based Method For Handwritten Connected Digits...** http://dspace.univ-tebessa.dz:8080/xmlui/handle/123456789/10962 ... preprocessing and augmentation techniques applied to the ... of the CNN model, we used the cvl and mnist data sets together ... had undergone preprocessing, including the skeletonization ...	**Sim. globale: 0.5%** Sim. forte: 0.5% Sim. proche: 0%
Fichier web	**How to Avoid Overfitting in Deep Learning Neural Networks** https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-... Aug 6, 2019 — Training a deep neural network that can generalize well to new data is a challenging problem. A model with too little capacity cannot learn ...	**Sim. globale: 0.5%** Sim. forte: 0.5% Sim. proche: 0%
Fichier web	**Convolution in convolution for network in network** https://ieeexplore.ieee.org/abstract/document/7879808/ ... of alternating convolutional layers and pooling layers. Instead ... number of feature maps (channels) obtained by the kernel ki. ... "pooling 3 × 3" and "pooling 8 × 8" mean that max pooling ...	**Sim. globale: 0.5%** Sim. forte: 0.5% Sim. proche: 0%

Page web	**Data Preprocessing in Machine Learning: 7 Easy Steps To ...** https://www.upgrad.com/blog/data-preprocessing-in-machine-learning/ Feb 18, 2024 — It involves refining, transforming, and structuring data to enhance the performance of new models. This critical data pre processing steps ...	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Page web	**Intelligent Character Recognition (ICR)** https://encord.com/blog/intelligent-character-recognition-icr/ May 3, 2024 — By implementing OCR, the firm can swiftly digitize these documents, making them editable and significantly reducing manual entry errors.	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Page web	**Applied Deep Learning - Part 4: Convolutional Neural ...** https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-... Nov 8, 2017 — So we flatten the output of the final pooling layer to a vector and that becomes the input to the fully connected layer. Flattening is ...	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Fichier web	**Deep learning: a comprehensive overview on techniques,...** https://link.springer.com/article/10.1007/s42979-021-00815-1 ... as data understanding and preprocessing, DL model building, ... conclude that the future prospects of deep learning modeling ... A combination of algorithmic advancements with improved ...	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Fichier web	**What is more preferable in machine learning, the accuracy ...** https://www.quora.com/What-is-more-preferable-in-machine-learning-the-accuracy-of-model-A-... High training accuracy does indicate that the model learned the training data. ... model's performance between the training and validation sets.	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Fichier web	**Real-time assistive reader pen for Arabic language** https://dl.acm.org/doi/abs/10.1145/3423133 ... Arabic OCR dataset of handwritten, printed, and embedded texts. ... of the recognized text to mobiles or computers, in an editable ... the process of Optical Character Recognition (OCR) is ...	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Fichier web	**Automated copper alloy grain size evaluation using a deep-...** https://arxiv.org/abs/2005.09634 ... introduced in the mid-twentieth century [20]. The CCD embeds ... Besides the dataset creation, the CNN training represented ... 35 training epochs, where for each epoch the CNN model is ...	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Fichier web	**An effective digit recognition model using enhanced...** https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs211242 ... effort as the performance of CNN-OLCGWO for handwritten digit ... for handwritten digit recognition using MNIST corpus. The stages of research are: Data acquisition and pre-processing, ...	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Fichier web	**Review and comparison of commonly used activation functions...** https://link.springer.com/chapter/10.1007/978-981-15-5495-7_11 ... commonly used additive functions, such as swish, ReLU, Sigmoid, ... to the simplest example of a linear equation, the equation output ... of the final layer which always uses softmax AF. The ...	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**

Fichier web	**Literature review: Efficient deep neural networks techniques...** https://link.springer.com/article/10.1007/s00521-022-06960-9 ... processing, computer vision, machine learning, and health ... Deep learning (DL) has changed forms and definitions slowly ... learning method) which required no preprocessing step and ...	**Sim. globale: 0.4%** **Sim. forte: 0.4%** **Sim. proche: 0%**
Page web	**What Is Image Segmentation?** https://www.ibm.com/topics/image-segmentation Image segmentation is a computer vision technique that partitions a digital image into discrete groups of pixels—image segments—to inform object detection ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Page web	**Python - Our Technologies - DataSense** https://datasense.be/blog/technology/python/ Its high-level built in data structures, combined with dynamic typing and dynamic ... use as a scripting or glue language to connect existing components together.	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Page web	**What is Python? Executive Summary** https://www.python.org/doc/essays/blurb/ Its high-level built in data structures, combined with dynamic typing and dynamic ... use as a scripting or glue language to connect existing components together.	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Page web	**45 Questions to test a data scientist on Deep Learning ...** https://www.analyticsvidhya.com/blog/2017/01/must-know-questions-deep-learning/ A. A neuron has a single input and a single output only. B. A neuron has multiple inputs but a single output only. C. A neuron has a single input but multiple ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Fichier web	**An Integrated Web Platform for Remote Control and Monitoring...** https://webthesis.biblio.polito.it/30829/ ... The frontend of the application is developed using React.js, ... By strategically leveraging these libraries, the server stands ... The user interface is intuitive, ensuring a seamless interaction ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Fichier web	**Fault diagnosis of nonlinear and large-scale processes using...** https://www.tandfonline.com/doi/abs/10.1080/00207721.2014.912780 ... -weighted kernel Fisher discriminant analysis (KFDA) based on ... KFDA, as a typical kernel technique, is used to diagnose ... for kernel function and parameter optimization in support vector ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Fichier web	**Using the confusion matrix for improving ensemble classifiers** https://ieeexplore.ieee.org/abstract/document/5662159/ ... the learning phase so that a misclassified record gets a higher weight during the current ... diagonal in the matrix represent the ideal case in which the instance was correctly classified...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Fichier web	**Understanding of machine learning with deep learning:...** https://www.mdpi.com/2073-431X/12/5/91 ... a seasoned oncologist, will serve as the performance metric. ... steps: data comprehension and preprocessing, DL model ... to more recent deep learning methods and breakthroughs in a ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**

Fichier web	**A survey of the recent architectures of deep convolutional...** 🔗 https://link.springer.com/article/10.1007/s10462-020-09825-6 ... for processing of grid-like topological data (images and time ... the distribution of hidden units' values, which slows down the ... these networks with recurrent neural network (RNN) and thus ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Fichier web	**A surrogate loss function for optimization of score in binary...** 🔗 https://arxiv.org/abs/2104.01459 ... true proportions, and z = (1 − q0) · (p(β2 + 1 − q0 − q1)+ q1). ... the surrogate Fβ loss with a large β value, ie, β = 3 or β = 2, ... see that the surrogate Fβ loss achieves the best F1 score on ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Fichier web	**Python - Valentina DB** 🔗 https://valentina-db.com/en/store/category/25-python Its high-level built in data structures, combined with dynamic typing and dynamic ... use as a scripting or glue language to connect existing components together.	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Fichier web	**Deep Learning-Based Brain Image Segmentation for Automated...** 🔗 https://arxiv.org/abs/2404.05763 ... convolutional neural network-CNN called the U-'Net' design ... As we delve further into the network, the number of filters or ... Focal loss: Deep learning tasks involving object detection use a ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Fichier web	**Introduction to convolutional neural networks** 🔗 https://cs.nju.edu.cn/wujx/paper/CNN.pdf ... see that the convolution kernels in a convolution layer of a ... Its size is shown in the format n1 × n2, where n1 is the size of ... of a max pooling, we need values in a 2 × 2 spatial extent in ...	**Sim. globale: 0.3%** **Sim. forte: 0.3%** **Sim. proche: 0%**
Page web	**A CNN Hyperparameters Optimization Based on Particle ...** 🔗 https://www.mdpi.com/2313-433X/10/2/30 by K Aguerchi · 2024 · Cited by 1 — The present study was based on the Particle Swarm Optimization (PSO) technique to look for the CNN model with the highest accuracy for breast cancer detection.	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Page web	**Advancements and Challenges in Arabic Optical Character ...** 🔗 https://arxiv.org/html/2312.11812v1 by MSE Kasem · 2023 · Cited by 2 — Optical character recognition (OCR) is a vital process that involves the extraction of handwritten or printed text from scanned or printed images, converting it ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Page web	**Confusion matrix, Precision, Recall, and F1 Score ...** 🔗 https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-... Sep 13, 2020 — Confusion matrix, precision, recall, and F1 score provides better insights into the prediction as compared to accuracy performance metrics.	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Page web	**Accuracy, precision, and recall in multi-class classification** 🔗 https://www.evidentlyai.com/classification-metrics/multi-class-metrics How to use accuracy, precision, and recall in multi-class classification? This illustrated guide breaks down how to apply each metric for multi-class ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**

Page web	**Medical image analysis using deep learning algorithms - PMC** https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10662291/ by M Li · 2023 · Cited by 19 — Medical image processing techniques have revolutionized the field of medicine by providing a non-invasive means to visualize and analyze the internal ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Fichier web	**What should I do next? The loss on the training set is ...** https://www.quora.com/What-should-I-do-next-The-loss-on-the-training-set-is-decreasing-but-the-... Apparently, this is happening because you model is overfitted (low bias but high variance). ... However, the validation loss initially decreases ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Fichier web	**Indian script character recognition: a survey** https://www.sciencedirect.com/science/article/abs/pii/S003132030400055X by U Pal · 2004 · Cited by 727 — But most of these systems work for Roman, Chinese, Japanese and Arabic characters. ... color document, low-quality noisy documents like photocopy and fax, etc.	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Fichier web	**Evaluation: from precision, recall and F-measure to ROC,...** https://arxiv.org/abs/2010.16061 ... of F1 is not known in AI/ML/CL/IR but is just as well known as ... Note that in the above equations (20-22) the denominator is ... weighting up of the true unbiased performance in both these ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Fichier web	**Advanced metaheuristic optimization techniques in applications...** https://link.springer.com/article/10.1007/s00521-021-05960-5 ... module such in convolutional neural network (CNN) architecture [18]. Thus, in ... Figure 1 depicts these groups. ... is very costly in terms of computational time. To tackle this problem, authors ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Fichier web	**Thresholding classifiers to maximize F1 score** https://arxiv.org/abs/1402.1892 ... , the results can be generalized to Fβ metrics for β = 1. ... from the score assigned to a complementary prediction Pc given ... variability of F1 results in high weight for rare labels in macro ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Fichier web	**F1 confidence, Precision-Recall curve ...** https://github.com/ultralytics/ultralytics/issues/7307 Jan 4, 2024 — Precision is the ratio of true positive predictions to the total number of positive predictions (true positives + false positives), while recall ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Fichier web	**Handwritten Character Recognition System** https://journal.esrgroups.org/jes/article/download/3553/2761/6417 The dataset used for training and assessment is made up of a large collection of handwritten characters gathered under various writing styles, sizes, and.	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**
Fichier web	**A survey on image data augmentation for deep learning** ... ... the training and validation accuracy at each epoch during ... augmented data will represent a more comprehensive set of ... of the loss function for the augmentation net (Content loss, Style ...	**Sim. globale: 0.2%** **Sim. forte: 0.2%** **Sim. proche: 0%**

Page web	**Image Thresholding in Image Processing** https://encord.com/blog/image-thresholding-image-processing/ Sep 12, 2023 — Image thresholding is a technique that simplifies a grayscale image into a binary image by classifying each pixel value as either black or white ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Page web	**What is REST API (RESTful API)?** https://www.techtarget.com/searchapparchitecture/definition/RESTful-API A RESTful API is an architectural style for an application programming interface that uses HTTP requests to access and use data. That data can be used to ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Page web	**FastAPI** https://fastapi.tiangolo.com/ FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints. The key features are: Fast ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Page web	**Mnist Dataset for Machine Learning | by Binaya Puri** https://medium.com/@binaya.puri/mnist-dataset-for-machine-learning-8987e0b20bb3 Data: The dataset contains 70,000 grayscale images of handwritten digits (0 through 9). Each image is a 28x28 pixel square, making the images ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Page web	**Hyper-parameter optimization of convolutional neural ...** https://www.slideshare.net/slideshow/hyperparameter-optimization-of-convolutional-neural-network-... Jan 27, 2022 — The document proposes using a particle swarm optimization (PSO) algorithm to optimize the hyperparameters of a convolutional neural network ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Page web	**Microsoft Lens for Android** https://support.microsoft.com/en-us/office/microsoft-lens-for-android-... Microsoft Lens is an app for your Android device that lets you easily capture, use, and share information from documents, whiteboards, photos, ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Page web	**Exploring the Realm of Machine Learning in Julia with Flux.jl** https://medium.com/@manuedavakandam/exploring-the-realm-of-machine-learning-in-julia-with-flux-... Flux.jl is a powerful machine learning library in Julia, providing a flexible and intuitive framework for building, training, and evaluating ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Fichier web	**particle swarm optimization algorithm - an overview** https://www.sciencedirect.com/topics/computer-science/particle-swarm-optimization-algorithm The PSO algorithm is a population-based optimization method inspired by the social behavior of bird and fish flocks searching for food, which means that ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Fichier web	**Deep learning for big data applications in CAD and...** https://www.sciencedirect.com/science/article/pii/S0166361517305560 ... recognition methods are usually combined with machine ... by a higher strength in preprocessing converges toward zero. ... Thus, it can be concluded that the model is able to handle 3D ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**

Fichier web	**Disruptive developments in document recognition** https://www.sciencedirect.com/science/article/pii/S0167865515004109 ... Computer vision used to be easily distinguished from the image ... to image processing still rely on character recognition. ... on our way to converting all existing books to electronic files. ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Fichier web	**A review: Data pre-processing and data augmentation techniques** https://www.sciencedirect.com/science/article/pii/S2666285X22000565 ... in Machine learning, focusing on all types of problems while ... So, raw data cannot be directly fed to the machine learning ... Cleaning methods are used to remove unnecessary data ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**
Fichier web	**A survey of data mining and deep learning in bioinformatics** https://link.springer.com/article/10.1007/s10916-018-1003-9 ... To transfer useful knowledge from original raw data, data mining ... to enhance data mining performance, a preprocessing step is ... to implement data cleaning and preprocessing, data ...	**Sim. globale: 0.1%** **Sim. forte: 0.1%** **Sim. proche: 0%**

Texte analysé

Cette section rapporte le texte original extrait du document analysé, avec les passages similaires surlignés. Ces passages ont été identifiés comme tels à partir des sources listées dans la section précédente.

◉ Similarité forte ◉ Similarité proche ◉ Exclu par l'utilisateur

General Introduction

The recognition of handwritten digits is an important field in artificial intelligence and computer vision that has played a significant role in converting traditional paper-based archives to electronic archives. Earlier, archives relied heavily on paper documents, which were difficult to maintain and organized and required a large amount of storage space. The advent of digital technology has enabled documents to be stored electronically, making them more accessible, searchable, and manageable.

Individual writing styles present a significant challenge in this field. Unlike printed characters, handwritten digits exhibit considerable variation in shape, size, slant, and stroke thickness. A variety of factors contribute

to this variability, including the rate at which the writer writes, the situation at the time of writing,

as well as cultural differences in the formation of digits. It is particularly challenging to recognize digits with fine or delicate handwriting, as these subtle strokes can be mistaken for background noise or incorrectly identified as different digits.

To address these challenges, the project explored Convolutional Neural Networks (CNNs) optimized using techniques such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). These methods aim to improve the model's ability to generalize from diverse training data and provide high performance on test datasets.

Additionally, our experiments were enhanced by using Julia, a programming language

that is less common but promising in the field of machine learning and scientific computing. Julia is known for its high performance and ease of use. Our aim with Julia is not
only to take advantage of these strengths, but also to explore new ways of optimising and

1

Introduction générale

improving handwritten digit recognition.

This project is more than just developing a robust model for handwritten digit recognition. It aims to identify and overcome specific obstacles associated with fine handwriting
while experimenting with Julia for model building. This will not only contribute to the advancement of handwritten character recognition technology, but also accelerate its implementation.

The structure of this work is divided into four chapters :

– The first chapter introduces optical character recognition (OCR) and covers its de finition, historical background, applications, and challenges. We also review existing OCR tools and outline the various stages of OCR.

– Our second chapter focuses on form recognition and discusses preprocessing tech niques, deep learning methods, and their applications in computer vision.

– The third chapter describes our experimental setup in detail, including the use of Julia and Google Colab, and introduces CNN models trained using genetic algo rithms (GA) and particle swarm optimization (PSO) on the MNIST dataset. We compare

the results and present our proposed preprocessing methods.

– The final chapter details the development and

testing of our platform, encompas sing the construction of a RESTful API that links multiple servers with the client interface. We conclude with a comprehensive overview of the project's success.

2

Chapitre 1

OCR

1.1 Introduction

Ever wondered how the human brain is able to process and analyze images so

effortlessly ? As soon as we come into contact with a picture, our brains instinctively break it
down into recognizable chunks, allowing us to identify its various components effortlessly.
But have you ever wondered if computers could perform similar tasks ? If you are aware that image processing is a branch of computer science that attempts to do the same thing with computers. How does OCR work, and what are its challenges and phases ?
1.2

Optical Character Recognition

1.2.1 definition

Optical Character Recognition (OCR) is a sub-field of image processing that deals with
the process of recognizing characters from an image (OCR)

. This approach examines an
image containing scanned text or handwritten characters and attempts to recall

them
using a variety of algorithms[?].
the format of the handwritten text :
the format of the handwritten text can be :
Isolated

characters :This task involves the
straightforward classification of individual
characters or numbers, independent of their context. It's a relatively simple task focused
on identifying single characters without considering their surrounding context.
3

Chapitre 1 Premier chapitre
Continuous characters : In contrast, recognition of continuous characters involves identifying and interpreting sequences of characters or numbers within a larger context, such

as words, sentences, or paragraphs. It's a more complex task that requires algorithms and techniques to handle factors like variable spacing, font styles, and contextual dependencies.

1.2.2 Historical

The development of retina scanners originates from early character recognition concepts,
which involved a framework for image transmission utilizing a mosaic of photocells. A significant breakthrough occurred in 1890 with Nipkow's invention of the sequential scanner,
laying the groundwork for modern television and reading machines. Initially, OCR technology was considered an aid for visually impaired individuals, but it soon evolved into a
vast field of research and development.

In 1929, a patent filed by Tauschek in Germany marks the first documented evidence of an optical character recognition system. This innovation led to a US patent granted to Tauschek in 1935, following an earlier public disclosure by Handel in 1933. Both early machines employed template symbols on a circular disc, allowing light to pass through specific cutouts. The image to be recognized was placed in front of the disc and illuminated. Reflected light passing through the template holes was focused onto a photosens or
for detection.

The commercially available OCR systems can be categorized into four generations based on their robustness, efficiency, and flexibility. The first generation OCRs, which emerged
in the mid-1960s, could read only specific fonts

and character shapes. The IBM 1418 was the first widely marketed OCR of this generation, employing logical template

matching

techniques.

The second generation of OCRs, available from the mid-1960s to mid-1970s, showed significant improvements. These systems could recognize both machine-printed and handwritten characters, although initially limited to numerals. The IBM 1287 exemplifies this
generation, incorporating both analog and digital technologies.

Subsequent advancements led to the development of third-generation OCRs between 1975 and 1985. These systems could handle a broader range of handwritten characters and poor-quality prints, enhancing their utility for diverse applications.

4

The fourth generation OCRs are capable of processing complex documents with interspersed text, mathematical symbols, and tables. They can recognize

spontaneous handwritten characters and handle low-quality noisy documents, such as photocopies, faxes, and color documents.

Modern OCR systems are now sophisticated enough to support multiple languages, including Arabic, Chinese, Japanese, and Roman scripts.[?].

1.2.3 domain of application OCR

Many applications of handwritten digit recognition and classification exist, including :

– Processing of checks and other financial documents using handwritten digit recog nition is becoming increasingly popular with banks and corporations.

– The technology can also be used to read and recognize license plate numbers on vehicles, which is particularly useful for law enforcement agencies and highway toll companies.

– Systems of security use handwritten digit recognition to protect assets and sensitive information, such as combination locks and PIN-based authentication systems.

– Automatic reading of handwritten digits using character recognition : Addresses on envelopes, postal codes, telephone numbers on application forms are often read au tomatically using character recognition. In this way, large volumes of administrative documents can be processed more efficiently and quickly.

1.2.4 impact on society and industry

Optical Character Recognition (OCR) is a transformative technology that has significantly influenced both society and industry. Let's delve into its impact :[?]

Efficiency and Automation :

- OCR automates the process of extracting text from printed documents, reducing the need for manual data entry. This efficiency boost saves time and minimizes errors compared to manual typing.

- Businesses can digitize vast amounts of paper-based information swiftly, making it accessible for further analysis and processing.

chapitre
Cost Reduction :
- By streamlining data extraction and storage, OCR reduces costs associated with manual data entry.
- Small businesses benefit from faster data processing and efficient data utilization, as OCR eliminates the need for extensive human involvement.
Consistency and Accuracy :
- OCR ensures consistent results across multiple users and projects. Unlike humans, it doesn't suffer from fatigue or variations in performance.
- Improved data accuracy leads to better decision-making and reliable information.
Data Security and Storage :
- Digitized documents are easier to store, search, and retrieve. OCR contributes to efficient data management.
- Enhanced security measures can be applied to digital records, safeguarding sensitive information.
Industry-Specific Applications :
- Healthcare : OCR assists in digitizing patient records, prescriptions, and medical reports, enabling efficient data sharing and analysis. - Automotive : OCR streamlines paperwork related to vehicle registration, insurance claims, and maintenance records.
- Finance : OCR automates invoice processing, expense management, and financial document handling.
- Legal : Legal professionals use OCR to convert paper-based contracts, court documents, and case files into digital formats.
1.2.5 challenges
OCR is a challenging task due to multiple problems that complicate the process of re-knowledge, among which are [?] :
— Document quality : a document that is faxed or photocopied multiple times is more difficult to process than the original copy. Writing may become thinner or other
6

Chapitre 1 Premier
chapitre
wise thicker, degraded with missing parts of text or tasks that appear, openings or
closures of loops . . .
— Printing : a composite document is better than a document typed which, in turn, is clearer than text from a printer dot matrix. An inkjet printer can introduce ink stains and a spread of characters, a laser printer can generate lines or funds. . .
— The discrimination

of form : depending on the style of the font used, its body and fatness, the character changes its graphics. The number of shapes is all the more important as the number of writing

styles is high. In addition, several characters have strong

similarities, such as :

- for Arabic : W and 7, / and 3,O and [
- for Latin : U and V, O and 0, S and 5, Z and 2.

- The medium of information, such as paper,

also plays on performance recognition by its quality : its grammage, granulation and color.

- • Acquisition : real-time

scanning often introduces distortions in the image. In the offline case, the quality of the scanned text is a compromise between the variations of the position (tilt, translation, shrinkage, etc.), the cleanliness of the glass of the scanning device

and its resolution.

1.3 existing applications

1.3.1 iScanner

The iScanner app is yet another example of cleverly leveraging an always-connected camera to do more than just intelligently make photos look prettier. The app's actually designed to turn a smartphone's camera into a document scanner by automating the process of color-correcting and straightening documents snapped at an angle as well as converting a page's content to editable text using optical character recognition[?].

1.3.2 Google Lens

Google Lens is a visual search tool developed by Google that uses machine

learning
to understand and analyze images and videos. It can be used to :
– Identify objects.

– Translate text.
– Shop for product.
– Copy text.
– Get help with homework[?].

1.3.3 Pen To Print

Pen to Print is an application that utilizes Optical Character Recognition (OCR) technology to convert handwritten text in images and PDFs into digital, editable format. Available as a mobile app on Android and iOS, as well as an online OCR tool on the Pen to Print website, this versatile tool offers the ability to convert handwritten material into
digital text with ease. With Pen to Print, users can store the scanned handwriting on any digital platform or cloud service, making it accessible from anywhere [?].

1.3.4 Microsoft Lens

microsoft Lens (formerly Microsoft Office Lens) is a free mobile app available on Android and iOS designed to enhance and make pictures of whiteboards and documents readable. This versatile app offers features to trim images, enhance their quality, and convert them into various digital formats. With Microsoft Lens, users can capture text, images, documents, and whiteboards, utilizing Optical Character Recognition (OCR) and other technologies to digitize printed or handwritten text. The app allows users to convert
captured content into PDF, Word, PowerPoint, and Excel files, offering flexibility in how information is utilized. Additionally, Microsoft Lens provides options to save content to OneNote, OneDrive, or the local device, ensuring accessibility and easy sharing of digitized
materials. [?]

1.3.5 comparative

The table below presents a comparison between the existing OCR platforms.

Applications	Advantages	Disavantages
Google Lens	-Convenience : Lens is a quick and easy way to get information about the world around you. - Versatility : Lens can be used for a variety of tasks, from identifying objects to translating text. -Accuracy : Lens is constantly being improved, and its accuracy is impressive. -Offline functionality : Some Lens features can be used offline, which is handy when you don't have an internet connection.	-Limited functionality : Lens is not perfect, and it may not always be able to identify objects or translate text accurately. Privacy concerns : Some people are concerned about the privacy implications of using Lens, as it collects data about the things you scan. -Reliance on internet connection : Most of Lens's features require an internet connection[?].
iScanner	-Comprehensive Features : iScanner offers a wide range of features including OCR (Optical Character Recognition), document scanning, editing, and sharing functionalities. -Intuitive Interface : The app boasts a user-friendly interface that effectively organizes its multitude of features into easily accessible categories like Scan, Edit, and Share. -Efficient Document Testing : iScanner demonstrates good performance in standard document testing,providing mostly accurate digitized text from various types of documents[?].	-Accuracy : Accuracy can vary depending on camera quality, lighting, and app capabilities. File size : Scanned images can be large, consuming device storage. Security : Pay attention to data privacy policies, as some apps may collect or share data. Limited features : Free versions might have limited features like scan quality, page limits, or file formats. - Internet connection : Some features might require an internet connection for functionality[?].

9

Chapitre 1 Premier chapitre

	-Unique Functionalities : Apart from basic scanning, iScanner offers unique features such as solving math problems and counting objects, adding value beyond traditional scanning apps. Text Blurring Feature : iScanner allows users to blur out text on documents with the ability to match the background color, offering enhanced privacy and document manipulation options. Overall Strength : With its combination of features, performance, and user experience, iScanner emerges as one of the strongest choices among OCR apps.	

Pen To Print -Readable to everyone, unlike some - the free version typically
im handwritings that are hard to read poses limitations on
conversions,
like cursive, scribbles and other illegible file sizes, and supported OCR
lan writing. - Easy editing capabilities - guages, which may hinder
high making changes, adding comments or volume conversion needs or
ad suggestions, deleting irrelevant parts, vanced usage. - Its
susceptibility
reordering sections, spell-checking, and to data breaches and malware
at more. -Easily stored, shared, and acces- tacks necessitates the
implemen sed from various devices, providing un- tation of strong
security mea paralleled convenience and flexibility. sures,
particularly when uploa - Enables simplified searches through ding
sensitive information - the large documents, making it easy to lo-
re's a risk of unauthorized parties
cate specific information quickly. accessing or storing your
uploa ded data, necessitating
compre hension of the app's
privacy po licy and
restricting its use to trus
ted content[?].

10

Chapitre 1 Premier chapitre

Microsoft Lens -Straightforward Interface : Users ap- -Limited File Management : Mi
preciate the simplicity of Microsoft crosoft Lens lacks robust file ma
Lens, as it allows them to start scan- nagement features, such as the
ning documents without going through ability to create folders or
orga sign-up processes or tours. -Minimal nize scans through sorting
and fil Fuss : Unlike other apps that may have tering options. -Basic
Export Op introductory processes, Microsoft Lens tions : After editing
documents,
skips these steps and lets users focus the app directly leads users
to
on scanning immediately. -Integration export options without provi
with Microsoft Ecosystem : Users who ding additional file organization
are already using Microsoft products capabilities. -OCR Accuracy :
benefit from seamless integration with While generally accurate,
Micro apps like OneDrive, OneNote, Word, soft Lens may have slight
inac and PowerPoint, making it easy to save curacies, such as missing
words
and import documents[?]. or letters, particularly when
dea ling with complex
documents like

books[?].

Table 1.1 – comparision between application of OCR

11

Chapitre 1 Premier

chapitre

1.4 OCR Phases :

In this section we describe the main important phases and architecture of optical character recognition[?].

Figure 1.1 – OCR phases[?]

1.4.1 Pre-processing

The objective of pre-processing is to remove undesired characteristics or noise from an image while preserving all significant information. Pre-processing techniques are essential for color, grey-level, or binary document images that contain text and/or graphics. Given that processing color images is computationally more intensive, most character recognition systems employ binary or grey-scale images. Pre-processing minimizes inconsistent data and noise, enhancing the image and preparing it for subsequent phases in OCR systems.

These are some important pre-processing operations :

12

Chapitre 1 Premier chapitre

Processes	Description
Binarization	Separates image pixels as text or background.
Noise Reduction	Improves image quality by reducing noise.
Skew Correction	Corrects document skew caused by image capture devices.
Morphological Operations	Add or remove pixels to characters, correcting imperfections.
Thresholding	Separates information from background in an image.
Thinning and Skeletonisation	Thinning reduces the width of characters to one pixel, Skeletonisation regularizes the text map.

Table 1.2 – Some important pre-processing operations

Figure 1.2 – binarisation phase[?]

1.4.2 Segmentation Phase

Document segmentation is a critical pre-processing step in the implementation of an

OCR system. It involves classifying a document image into homogeneous zones, where each zone contains only one type of information, such as text, a figure, a table, or a halftone
image. The accuracy of OCR systems is significantly influenced by the effectiveness of the
page segmentation

algorithm employed.

Document segmentation algorithms can be broadly categorized into three types :

–

Top-down methods

13

Chapitre 1 Premier chapitre

– Bottom-up methods

– Hybrid methods

Top-down methods segment a document by recursively dividing large regions into smaller sub-regions.

This process continues until a specified criterion is met, at which point the segmentation halts, and the resultant sub-regions form the final segmentation. In contrast,
bottom-up methods begin by identifying interest pixels within the document. These interest pixels are then grouped into connected components that form characters. These characters are subsequently

combined into words, lines, or text blocks. Hybrid methods
integrate both top-down and bottom-up approaches to leverage the advantages of both

strategies, aiming for improved
segmentation accuracy and efficiency.
Figure

1.3 – segmentation phase[?]

1.4.3 Normalization Phase

As a result of the segmentation process, isolated characters are obtained, which are then ready to proceed to the feature extraction phase. These isolated

characters are typically resized
according to

the specific algorithms employed. The
segmentation process
is crucial as it converts the image into an m × n

matrix. These matrices are commonly
normalized by reducing their size and eliminating extraneous information from the image while preserving all essential details.

14

Chapitre 1 Premier
chapitre

Figure 1.4 – normalization phase[?]

1.4.4 Feature Extraction Phase

Feature extraction involves identifying

and extracting relevant features from
objects
or alphabets to construct feature vectors. These feature vectors are then used by classifiers to match the input unit with the corresponding output unit.

This process simplifies classification, as the classifier can more easily differentiate between distinct classes by

examining these features. According to Suen, there are two primary categories of features : statistical features and structural features. In a character matrix, statistical features are derived from the statistical distribution of each point, such as zoning, moments, crossings, Fourier transforms, and projection histograms.

- Statistical features, also known as global features, are generally averaged and extracted from sub-images, such as meshes. Initially, statistical features were used

to recognize machine-printed

characters.

- Structural or topological features pertain to the geometry of the character set being analyzed. Examples of these features include convexities and concavities in the characters, the number of holes in the characters, and the number of endpoints.

1.4.5 Classification Phase

OCR systems extensively utilize pattern recognition methodologies, which assign each example to a predefined class. Classification is the process of assigning

inputs to their corresponding classes based on detected information, thereby creating groups with homogeneous characteristics and separating different inputs into distinct classes. Classification is

performed using stored features in the feature space, such as structural features, global features, and others.

In essence, classification divides the feature space into

several classes based on a deci
15

Chapitre 1 Premier chapitre

sion rule. The choice of classifier depends on several factors, including the number of free parameters and the available training set. Researchers have explored various procedures for OCR to enhance the accuracy and efficiency of classification.

Template Matching

Template matching is the simplest method for character recognition, based on comparing stored templates against the character or word to be recognized. By analyzing shapes, pixels, curvature, and other features, the matching

operation determines the degree of similarity between two vectors. A gray-level or binary input character is compared to a standard set of stored templates. The recognition rate of this method is highly sensitive to noise and input distortions.

Statistical Techniques

The theory of statistical decision involves statistical decision functions and a set of optimality criteria, which, for a given model of a specific class, can maximize the likelihood of the observed pattern. The main statistical methods used in OCR include Nearest Neighbor (NN), Likelihood or Bayes classifier, Clustering Analysis, Hidden Markov Modeling (HMM), Fuzzy Set Reasoning, and Quadratic Classifier.

Neural Networks :

Character classification is a problem that aligns

with heuristic reasoning, as humans recognize characters and documents through knowledge and experience. Neural networks, which are inherently heuristic, are therefore well-suited for this type of problem. A neural network is a computational architecture that consists of a massively parallel interconnection of adaptive node processors. The output from one node influences the next node in the network, and the final decision depends on the complex interactions of all nodes.

Due to its parallel nature, a neural network can perform computations at a higher rate compared to traditional

methods. Neural network architectures can be categorized into feed-forward neural networks and feedback neural networks.

16

Chapitre 1 Premier chapitre

Figure 1.5 – Neural network[?]

Kernel Methods :

Among the most important kernel methods are Support Vector Machines (SVMs). Techniques such as Kernel Fisher Discriminant Analysis (KFDA) and Kernel Principal Component Analysis (KPCA) also utilize kernel methods. SVMs are one of the most widely used and effective supervised learning techniques, suitable for both binary and multi-class classification. In classification techniques, the data set is conventionally divided into training and testing sets. The objective of SVM is to develop a model that accurately predicts the outcomes for the test set. The enhancement rule for SVM is to maximize the width of the margin between classes, which is the empty region around the decision boundary defined by the distance to the nearest training example.

Combination Classifier :

Different classification strategies have their own particular advantages and shortcomings. Thus ordinarily various classifiers are consolidated together to solve a given classification problem. Matei, Oliviu, Petrica C. Pop, and H. Vălean by utilizing neural networks and k-Nearest Neighbor, proposed Optical character recognition in real environments such as electricitymeters and gas-meters.

17

Chapitre 1

Premier chapitre

1.5 Conclusion

As a result of the quest to emulate the brain's image processing capabilities, OCR was developed. Even with its challenges, OCR continues to advance, offering promising solutions to tasks such as document digitization and automated data extraction. It is becoming increasingly apparent that the intersection of human cognition and computational intelligence will open up new avenues of exploration and innovation in the field of image processing and optical character recognition.

18

Chapitre 2

FORM RECOGITION

2.1 Introduction

Form recognition is a critical component in the field of document processing, enabling
the automated identification and extraction of structured data from scanned documents, images, and digital forms. This chapter delves into the methodologies and techniques used to preprocess and recognize forms accurately. By leveraging advances in both traditional image preprocessing methods and modern deep learning approaches, we aim to enhance the precision and efficiency of form recognition systems.
The chapter begins by exploring the essential preprocessing steps required for form recognition, emphasizing the significance of preparing images for subsequent analysis. Following
this, we introduce deep learning techniques, which have revolutionized the field of computer vision, providing robust solutions for form recognition tasks. We discuss the basics
of neural networks, various deep learning architectures, and the importance of data preprocessing in training effective models. The chapter concludes with a discussion on

the
evaluation metrics used to measure the performance of form recognition systems, highlighting the advancements

and future directions in this domain.

2.2 Preprocessing for

Form Recognition

2.2.1 Introduction

Image preprocessing is the process of manipulating raw image data into a usable and meaningful format. It allows you to eliminate unwanted distortions and enhance specific

19

Chapitre 2 Deuxième chapitre

qualities essential for computer vision applications. Preprocessing is a crucial first step to prepare your image data before feeding it into machine learning models[?].

2.2.2 The Role of image Preprocessing

let's explore why the image preprocessing is necessary :

- Enhanced Model Performance :By preprocessing images, image analysis algorithms perform better. By mitigating noise, inconsistencies, and outliers present in images, we create a cleaner and more reliable dataset for analysis. This results in improved accuracy of image processing tasks, reduced risk of overfitting to irrelevant details, and enhanced ability to generalize patterns across images.
- Reliable Insights and Interpretability :When images are preprocessed, they yield more reliable insights and make image analysis algorithms more interpretable. When images are cleaned, enhanced, and standardized through preprocessing techniques, underlying patterns and structures become more discernible. This facilitates better decision-making in image interpretation and aids in understanding the factors influencing the algorithm's predictions.
- Robustness to Real-World Scenarios :Preprocessing images contributes to the robustness of image analysis algorithms in real-world scenarios. Often, real-world images have variations in lighting, perspective, and quality, which require preprocessing techniques. By preparing images through preprocessing, we ensure that algorithms can cope with diverse environmental conditions encountered during deployment, leading to more reliable and consistent performance[?].

2.2.3 Exploring Image Preprocessing Methods

There are several techniques used in image preprocessing

1. Thresholding : Thresholding is a method that transforms a grayscale image into a binary image (black and white) by selecting a threshold value. Pixels darker than the threshold are assigned

to black, while pixels lighter than the threshold are assigned to white. This technique is effective for images characterized by high contrast and uniform lighting conditions.[?].

20

Chapitre 2 Deuxième chapitre

Figure 2.1 –

Thresholding image[?]

2. Resizing : Resizing images to a consistent size is important for machine learning algorithms to work properly. You'll want all your images to be the same height and width, usually a small size like 28x28 or 64x64 pixels[?].

Figure 2.2 – Resizing image[?]

3. Grayscaling : Converting color images to grayscale can simplify your image data and reduce computational needs for some algorithms[?]. (b)
4. Binarization : Binarization converts grayscale images to black and white by thresholding[?].(c)
5. Noise reduction : Techniques like Gaussian blurring, median blurring, and bila teral filtering can reduce noise and smooth images.

Normalizing pixel values to a standard range like 0 to 1 or -1 to 1 helps algorithms work better[?].(e)

Figure 2.3 – (a) Original

Image. (b) Converted to Grayscale. (c)

Binarized image. (d)
Thinning and Skeletonization are done. (e) Noise Removed[?]

6. Thinning and Skeletonization : This step is performed for the handwritten text, as different writers use different stroke widths to write. This step makes the width of strokes uniform.(d)
7. Skew Correction : While scanning or taking a picture of any document, it is possible that the scanned or captured image might be slightly skewed sometimes. For the better performance of the OCR, it is good to determine the skewness in image and correct it[?].

Figure 2.4 – Skew Correction image[?]

8. Lines Straightening : When the lines are curvy as in the case of the above image, it may result in OCR issues and can cause issues with line segmentation and text re arrangement. Hence, detecting the curved lines and straightening them will improve our OCR results[?].

Figure 2.5 – Lines Straighteningimage[?]

2.3 Deep Learning Approaches for Form Recognition

2.3.1 Introduction to Deep Learning

– What is Deep Learning ?

Deep learning is the branch of machine learning which is based on artificial neural network architecture. An artificial neural network or ANN uses layers of interconnected nodes called neurons that work together to process and learn from the input data.

In a fully connected Deep neural network, there is an input layer and one or more hidden layers connected one after the other. Each neuron receives input from the previous layer neurons or the input layer. The output of one neuron becomes the input to other neurons in the next layer of the network, and this process continues until the final layer produces the output of the network. The layers of the neural network transform the input data through a series of nonlinear transformations, allowing the network to learn complex representations of the input data[?].

– The using of Deep Learning :

Deep learning can be used for supervised, unsupervised as well as reinforcement machine learning. it uses a variety of ways to process these[?].

– Supervised Learning : Neural networks learn from labeled datasets to make predictions or classifications, minimizing errors through backpropagation. Com mon tasks include image classification, sentiment analysis, and language trans lation using algorithms like Convolutional and Recurrent Neural Networks.

– Unsupervised Learning : Neural networks uncover patterns or cluster un labeled data. Autoencoders and generative models are utilized for tasks such as clustering, dimensionality reduction, and anomaly detection.

– Reinforcement Learning : Agents learn to maximize rewards by interacting with environments. Deep reinforcement learning algorithms like Deep Q Net works and Deep Deterministic Policy

Gradient (DDPG) excel in tasks such as robotics and game playing.

— A Comparative Analysis : AI, Machine Learning, and Deep Learning : 23

Aspect	Artificial Intelligence (AI)	Machine Learning (ML)	Deep Learning (DL)
Definition	A broad field encompassing the simulation of human intelligence by machines.	A subset of AI that enables systems to learn from data.	A subset of ML focused on neural networks with many layers.
Scope	Includes ML, DL, and other techniques like rule-based systems.	Involves algorithms that improve over time with data.	Involves complex neural networks for high-level feature extraction.
Techniques Used	Rule-based systems, search algorithms, ML, DL, expert systems.	Regression, classification, clustering, decision trees.	Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Transformers.
Data Dependency	Can operate with less data or predefined rules.	Requires large datasets to improve performance.	Requires massive amounts of labeled data for training.

Computational Power	Varies widely ; generally less intensive than DL.	Moderate computational requirements.	High computational power required (GPUs, TPUs).
Training Time	Generally shorter than DL but varies.	Moderate training time.	Can be very long due to complex networks and large data sets.
Applications	Robotics, game playing, expert systems, natural language processing (NLP).	Spam detection, recommendation systems, image recognition, predictive analytics.	Advanced image and speech recognition

Table 2.1 – Comparative Analysis of AI, Machine Learning, and Deep Learning[?]

24

Chapitre 2 Deuxième chapitre

2.3.2 Neural Networks Basics

1. Introduction to Neural Networks :The term "deep learning" refers to a class of neural network models characterized by their multilayered architecture, consisting of interconnected units called neurons. This chapter aims to deliver a precise and exhaustive introduction to the functioning of these neurons and all their components[?].
2. Neuron Structure : Neural networks consist of layers of similar neurons. Most have at least an input layer and an output layer. The program presents the input pattern to the input layer. Then the output pattern is returned from the output layer. What happens between the input and an output layer is a black box. By black box, we mean that you do not know exactly why a neural network outputs what it does. At this point, we are not yet concerned with the internal structure of

the neural network, or the black box.
Many different architectures define the
interaction between the input and output layer[?].

(a) Input Layer :Receives the input values[?].

(b) Hidden Layer(s) :A set of neurons between the input and output layers. There
can

be a single or multiple layers.Usually,
it has one neuron, and its output
ranges between 0 and 1, that is, greater than 0 and less than 1.

(c) Output Layer :Usually, it has one neuron, and its output ranges between 0 and
1, that is, greater than 0 and less than

1[?].

(d) Weights :The weights of the neuron allow you to adjust the slope or shape of
the activation function[?].

(e)

Bias :Programmers add bias neurons to
neural networks to help them learn
patterns. Bias neurons function like an input neuron that always produces the
value of 1. Because the bias neurons have a constant output of 1, they are

not connected to the previous layer.
The value of 1, which is called the bias
activation, can be

set to values other than 1. However,1
is the most common
bias activation. Not all neural networks have bias neurons[?].

25

Figure 2.6 –

Lines Straighteningimage[?]

Y : The final value of the node.

W : Represents the weights between the nodes in the previous layer and the output node.

X : Represents the values of the nodes of the previous layer. B : Represents bias, which

is an additional value present for each neuron. Bias is essentially a weight without an input term. It's useful for having an extra bit of adjustability which is not dependent on the previous

layer.

H : Represents the intermediate node value. This is not the final value of the

node.

f() : Called an Activation Function, it is something we can choose. We will go through its importance later.

3. Activation Functions : In neural network programming, activation or transfer functions establish bounds for the output of neurons. Neural networks can use many different activation functions. We will discuss the most common activation functions in this section.

Choosing an activation function for your neural network is an important considera tion because it can affect how you must format input data [?].

— Linear Activation Function : The most basic activation function is the linear function because it does

not change the neuron output at all. Equation 2.1 shows how the program typically implements a linear activation function :

$$f(x) = x \quad (2.1)$$

— Hyperbolic Tangent Activation Function : The hyperbolic tangent function is also a very common activation function for neural networks that must output values in the range between -1 and 1. This activation function is

simply the hyperbolic tangent (tanh) function2.2 :

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

— Sigmoid Activation Function

: The sigmoid or logistic activation function is a very common choice for feedforward neural networks that need to output only positive numbers2.3 :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

— The Softmax Function : Softmax is usually found in the output layer of a neural network. The softmax function is used in classification neural networks.

The neuron

that has the highest value claims the input as a member of its class. Because it is a preferable method, the softmax activation function forces the output of the neural network to represent the probability that the input falls into each of the classes 2.4 :

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{2.4}$$

— The ReLU Function : Introduced in 2000 by Teh Hinton, the rectified linear unit (ReLU) has seen very rapid adoption over the past few years. Prior to the ReLU activation function, the hyperbolic tangent was generally accepted as the activation function of choice. Most current research now recommends the ReLU due to superior training results. As a result, most neural networks should utilize the ReLU on hidden layers and either softmax or linear on the output layer. Equation 2.5 shows the very simple ReLU function :

$$f(x) = \max(0, x) \tag{2.5}$$

chapitre

4. Forward Propagation : Forward propagation is the process in a neural network where the input data is passed through the network's layers to generate an output. Forward propagation is essential for making predictions in neural networks. It calculates the output of the network for a given input based on the current values of the weights and biases. The output is then compared to the actual target value to calculate the loss, which is used to update the weights and biases during the training process.
5. Backpropagation : Backpropagation is one of the most common methods for training a neural network. Rumelhart, Hinton, Williams (1986) introduced backpropagation, and it remains popular today. Programmers frequently train deep neural networks with backpropagation because it scales really well when run on graphical processing units (GPUs). To understand this algorithm for neural networks, we must examine how to train it as well as how it processes a pattern. Classic backpropagation has been extended and modified to give rise to many different training algorithms. In this chapter, we will discuss the most commonly used training algorithms for neural networks. We begin with classic backpropagation and then end the chapter with stochastic gradient descent (SGD).

2.3.3 Deep Learning Architectures

Deep learning architectures offer promising solutions. By leveraging neural networks, these architectures can handle complex object detection tasks efficiently. They allow for better training and yield superior results compared to traditional methods.

In the following sections, we delve into various deep learning architectures tailored for object detection, exploring their designs and functionalities to address the challenges posed by real-world scenarios.[?].

1. Convolutional Neural Networks (CNNs) :onvolutional networks (,), also known asLeCun 1989 convolutional neural networks or CNNs, are a specialized kind of neural network for processing data that has a known, grid-like topology. Examples include time-series data, which can be thought of as a 1D grid taking samples at regular time intervals, and image data, which can be thought of as a 2D grid of pixels. Convolutional networks have been tremendously successful in practical applications. The name “convolutional neural network” indicates that the network

28

Chapitre 2 Deuxième chapitre

employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers [?].

Figure 2.7

– CNN [?]

(a) Input : The input to the CNN is an image with dimensions 28 × 28 and one channel (grayscale).

(b) Conv 1 Convolution (5 x 5) kernel valid padding : In this phase, the image is convolved with a 5×5 kernel, producing feature maps. "Valid padding"

means that the convolution operation is performed only where the input and the filter fully overlap, resulting in an output feature map size of

24 × 24 with n1 channels.

(c) Max-Pooling (2 x 2) : After convolution, max-pooling is applied

with a 2 × 2 filter, reducing the spatial dimensions of the feature maps by half. This results in feature maps of

size 12 × 12 with n1 channels.

(d) Conv 2 Convolution (5 x 5) kernel valid padding : Another convolutional layer is applied similarly to the first one, resulting in a feature map size of 8 × 8 with n2 channels.

(e) Max-Pooling (2 x 2) : Max-pooling is applied again, reducing the spatial dimensions to 4 × 4 with n2 channels.

(f) Flattening : The feature maps are flattened into a one-dimensional vector, which serves as the input to the fully connected layers.

(g) fc 3 Fully-Connected Neural Network ReLU activation : The flattened vector is fed into a fully connected layer (fc 3) with ReLU activation. This layer performs transformations to learn complex patterns from the flattened features.

(h) fc 4 Fully-Connected Neural Network (with dropout) : Another fully connected layer (fc 4) is employed, which further learns complex relationships in the data. Dropout, a regularization technique, is applied to prevent overfit ting by randomly dropping some connections during training.

(i) n3 units (Output) : Finally, the output layer consists of n3 units, representing the predicted classes or values. This phase generates the final output of the CNN model.

2. Feedforward Neural Networks : Deep feedforward networks, also known as feed forward neural networks or multilayer perceptrons (MLPs), are fundamental models in deep learning. The goal of a feedforward network is to approximate a function f ∗ . For instance, in a classifier, y = f ∗ (x) maps an input x to a category y. A feed forward network defines a mapping y = f (x; θ) and learns the parameters θ that yield the best function approximation [?].

3. Recurrent Neural Networks (RNNs) : Recurrent neural networks (RNNs) are a family of neural networks designed for processing sequential data . Similar to

how convolutional networks are specialized

for processing grids of values, such as images, RNNs are specialized for processing sequences of values x(1), . . . , x(τ). Just as convolutional networks can scale to images with large dimensions and variable sizes, RNNs can scale to much longer sequences than would be practical for networks without sequence-based specialization. Most recurrent networks can also handle se quences of variable length[?].

4. Generative Adversarial Networks (GANs)GANs are at the forefront of dis ruptions in DL and have been an active research topic recently. In a nutshell, a GAN allows a network to learn from images that represent a real-world entity (say, a cat or dog ; when we simply develop a DL model to classify between a cat and a dog) and then generate a new image using the same features it has learned in

30

Chapitre 2 Deuxième chapitre

the process ; that is, it can generate a new image of a cat that looks (almost) au thentic and is completely different from the set of images you provided for training. We can simplify the entire explanation for GAN into one simple task (i.e., image generation). If the training time and the sample images provided during train are sufficiently large, it can learn a network that can generate new images that are not identical to the ones you provided while training ; it generates new images[?].

2.3.4 Data Preprocessing for Deep Learning

Data preprocessing is a crucial step in preparing data

for deep learning models. It involves various techniques to clean, transform, and enhance the raw data to improve the model’s performance. Here’s an overview

of key steps in data preprocessing for

deep
learning :

1. Data Cleaning :Data cleaning involves fixing systematic problems or errors in messy data. The most useful data cleaning involves deep domain expertise and could involve identifying and addressing specific observations that may be incorrect.
   There are many reasons data may have incorrect values, such as being mistyped, corrupted, duplicated, and so on. Domain expertise may allow obviously erroneous observations to be identified as they are different from what is expected, such as a person's height of 200 feet[?].
2. Photo

Resize :The photos will have to be reshaped prior to modeling so that all images have the same shape. This is often a small square image. There are many ways to achieve this, although the most common is a simple resize operation that will stretch and deform the aspect ratio of each image and force it into the new shape. We could load all photos and look at the distribution of the photo widths and heights, then design a new photo size that best reflects what we are most likely to see in practice. Smaller inputs mean a model that is faster to train, and typically this concern dominates the choice of image size. In this case, we will follow this approach and choose a fixed size of 200 × 200 pixels[?].
3. Image augmentation : Once we have collected the images, many times we do not have big enough dataset to train the algorithm. It also allows us to add the generalization ability to the network and prevent it from overfitting. For Neural Networks, we require more and more data, and image augmentation can help in

artificially increasing

the training dataset by creating versions of images. It enhances
the ability of the models as the augmented images provide different

variations along with the original training dataset[?].

2.3.5 Training Deep Learning Models

1. Loss Functions : Loss is a measure of a model's accuracy, representing the diffe rence between actual and predicted values. The function used to calculate this loss is called the loss function. Different loss functions can yield different values for the same loss, affecting the respective model's performance[?].
2. Optimization Algorithms : Here are concise definitions for some common opti mization algorithms :
   (a) Genetic Algorithms :

Figure 2.8 – Life cycle of a Genetic Algorithm[?]

Genetic Algorithms (GAs) are searching processes based on the principles of natural selection and genetics.

Typically, a simple GA consists of three opera tions : Selection, Genetic Operation, and Replacement. Initially, a population is generated randomly. The fitness values of all chromosomes are evaluated by calculating the objective function in a decoded form (phenotype). A group of chromosomes (parents) is selected to generate offspring via genetic operations.

The offspring's fitness is evaluated similarly to their parents. The chromosomes in the current population are then replaced by their offspring based on a cer tain replacement strategy. This GA cycle repeats until a desired termination

criterion is reached, such as a predefined number of generations. If successful,
the best chromosome in the final population becomes a highly evolved solution to the problem [?].

Figure 2.9 – Steps of a Genetic Algorithm[?]

(b) Ant Colony Optimization : Ant System is the first Ant Colony Optimization (ACO) algorithm proposed. Its main characteristic is that, at each iteration, the pheromone values are updated by all the m ants that have built a solution in the iteration itself. The pheromone τij , associated with the edge joining cities i and j, is updated as follows [?] :

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \tau_{ij}^{k} , \quad (2.6)$$

where ρ is the evaporation rate, m is the number of ants, and τ_{ij}^{k} is the quantity of pheromone laid on edge (i, j) by ant k :

$$\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ used edge } (i, j) \text{ in its tour,} \\ 0 & \text{otherwise,} \end{cases} \quad (2.7)$$

where Q is a constant, and L_k is the length of the tour constructed by ant k.
In the construction of a solution, ants select the next city to visit through a stochastic mechanism. When ant k is in city i and has so far constructed the

partial solution s^p , the probability of going to city j is given by :

$$\frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum \tau^{\alpha} \cdot \eta^{\beta}} \text{ if } j \in N(s^p),$$

$$p^k_{ij} = \begin{cases} \dfrac{\quad ij}{\sum_{l \in N(s^p)} \quad il \; il} & \\ 0 & \text{otherwise,} \end{cases} \qquad (2.8)$$

where $N(s^p)$ is the set of feasible components, i.e., edges (i, l) where l is a city not yet visited by ant k. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij}, which is given by :

$$\eta_{ij} = \frac{1}{d_{ij}}, \qquad (2.9)$$

where d_{ij} is the distance between cities i and j.

(c) Fuzzy Logic

: Fuzzy logic is a branch of mathematics that allows for approxi mate reasoning rather than fixed and exact reasoning. In a fuzzy logic control system, three main modules are involved [?] :

- Fuzzification module : Converts crisp inputs into fuzzy sets.
- Inference module : Applies a set of rules to the fuzzy sets to derive fuzzy outputs.
- Defuzzification module : Converts the fuzzy outputs back into crisp values.

3. Regularization : is a technique that reduces overfitting, which occurs when neural networks attempt to memorize training data, rather than learn from it. Humans are capable of overfitting as well. Before we examine the ways that a machine acciden tally overfits,we will first explore how humans can suffer from it[?].
4. Hyperparameter Tuning :Hyper-parameters, are the numerous settings for mo dels such as neural networks. Activation functions, hidden neuron counts, layer structure, convolution, max-pooling and dropout are all examples of neural network hyper-parameters. Finding the optimal set of hyper-parameters can seem a daunting task, and, indeed, it is one of the most time-consuming tasks for the AI programmer. However, do not fear, we will provide you with a summary of the current research on neural network architecture in this chapter. We will also show you how to conduct experiments to help determine the optimal architecture for your own networks[?].

2.3.6 Evaluation and Performance Metrics

1. Accuracy :is another measurement defined as the proportion of true instances retrieved, both positive and negative, among all instances retrieved. Accuracy is a weighted arithmetic mean of precision and inverse precision. Accuracy can also be high but precision low, meaning the system performs well but the results produced are slightly spread, compare this with hitting the bulls eye meaning both high accuracy and high precision, see Formula 2.10 [?].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

2. Precision, Recall, F1-Score :Two metrics used for measuring the performance of a retrieval system are precision and recall. Precision measures the number of correct instances retrieved divided by all retrieved instances, see Formula 2.11. Recall measures the number of correct instances retrieved divided by all correct instances, see Formula 2.12. Instances can be entities in a text, or a whole document in a document collection (corpus), that were retrieved. A confusion matrix, see Table 2.11 is often used for explaining the different entities.
Here follow the definitions of precision and recall, see Formulas2.11 and 2.12 respectively [?].

$$Precision = \frac{TP}{TP + FP} \quad (2.11)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.12)$$

The F-score is defined as the weighted average of both precision and recall depending on the weight function β, see Formula 2.13. The F1-score means the harmonic mean between precision and recall, see Formula 2.14, when it is written F-score it usually means F1-score. The F-score is also called the F-measure. The F1-score can have different indices giving different weights to precision and recall [?].

$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{\beta^2 \times P + R} \quad (2.13)$$

With β = 1, the standard F-score is obtained, which is equivalent to the F1-score :

$$F1 = F = \frac{2 \times P \times R}{P + R} \quad (2.14)$$

35

3. ROC Curve and AUC :A ROC curve is a two-dimensional plot that illustrates how well a classifier system works as the discrimination cut-off value is changed over the range of the predictor variable. The x axis or independent variable is the false positive rate for the predictive test. The y axis or dependent variable is the true positive rate for the predictive test. Each point in ROC space is a true positive false positive data pair for a discrimination cut-off value of the predictive test. If the probability distributions for the true positive and false positive are both known, a ROC curve can be plotted from the cumulative distribution function [?].
4. Confusion MatrixA neural network trained for the MNIST data set should be able to take a handwritten digit and predict what digit was actually written. Some digits are more easily confused for others. Any classification neural network has the possibility of misclassifying data. A confusion matrix can measure these misclassifi cations [?]. You can create a confusion matrix with the following steps [?] :
   (a) Separate the dataset into training and validation sets.
   (b) Train a neural network on the training set.

(c) Initialize the confusion

matrix with all zeros.
(d) Loop over every element in the validation set.
(e) For every element, increase

the cell corresponding to the expected class (row)
and the predicted class (column) in the confusion matrix.
(f) Report the confusion

matrix.

2.3.7 Deep Learning for Computer Vision

Computer vision is a field of study focused on the problem of helping computers

to see.

1. Image Classification : Predict the type or class of an object in an image [?].
   – Input : An image with a single object, such as a photograph.
   – Output : A class label (e.g., one or more integers that are mapped to class labels).
2. Object Localization : Locate the presence of objects in an image and indicate their location with a bounding box [?].
   – Input : An image with one or more objects, such as a photograph.

   – Output : One or more bounding boxes (e.g., defined by a point, width, and height).
3. Object Detection : Locate the presence of objects with a bounding box and types or classes of the located objects in an image [?].
   – Input : An image with one or more objects, such as a photograph.
   – Output : One or more bounding boxes (e.g., defined by a point, width, and height), and a class label for each bounding box.
4. Image Segmentation : Image segmentation is a computer vision technique that

partitions a digital image into discrete groups of pixels—image segments—to inform object detection and related tasks. By parsing an image's complex visual data into specifically shaped segments, image segmentation enables faster, more advanced image processing [?].

5. Face Recognition

: Face recognition is nothing new. We are born with a natural capability to differentiate and recognize faces. It is a trivial task for us. We can recognize people we know in any kind of background, different lights, hair color.. [?].

2.4 Conclusion

Form recognition combines traditional image processing and advanced deep learning techniques to enhance document processing accuracy. Preprocessing improves input data quality, while deep learning offers robust solutions for complex tasks. This chapter outlined the essential methods, from preprocessing to deep learning models, demonstrating their integration for efficient form recognition. Future advancements will continue to refine these approaches, further automating and optimizing document processing workflows.

37

Chapitre 3
Model Development and Performance Evaluation

3.1 Introduction

The purpose of this chapter is to present our Convolutional Neural Network (CNN) models, their performance, datasets, and a novel

preprocessing technique. We describe the structure of the CNN models and discuss the use of the MNIST dataset, enhanced through a

skeletonization preprocessing step. In this preprocessing, handwritten digits are converted into skeletal forms, which can improve the accuracy of recognition. We compare model performance using Genetic Algorithm (GA) and Particle Swarm Optimization

(PSO). Performance metrics such as accuracy, precision, recall, and F1-score demonstrate significant improvements with our preprocessing method. A comprehensive overview of our models, datasets, and proposed preprocessing technique is provided in this chapter.

3.2 Setup and Training Procedure

For our handwritten digit recognition project, we used the Google Colab platform, leveraging only the CPU for our experiments. We implemented two approaches to generate convolutional neural network (CNN) architectures : one model using a genetic algorithm and another model using the particle swarm optimization (PSO) algorithm. Each approach was run for 10 iterations. Each generated CNN architecture was trained for 20 epochs, using the Adam optimizer and a batch size of 128. These experiments were designed

to develop a CNN architecture that could correctly classify handwritten digits from our dataset.

3.3 Dataset

In our work, we use the MNIST dataset, which consists of 70,000 28x28 black-andwhite images of handwritten digits extracted from two NIST databases. The training dataset consists of 60,000 images, while the testing dataset comprises 10,000 images. The dataset contains 10 classes, one for each digit from zero to nine. Additionally, we use 20% of the training data as validation data.

The following table provides the distribution of images across the classes in the training
and test sets :

Classes	Training image	Test image
0	5923	980
1	6742	1135
2	5958	1032

3	6131	1010
4	5842	982
5	5421	892
6	5918	958
7	6265	1028
8	5851	974
9	5949	1009
Total	60 000	10 000

Table 3.1 – Distribution

of images in the MNIST dataset across training and test sets

The performance of different algorithms and models in the field

of image classification can be evaluated by using the MNIST dataset, which is a widely used benchmark. The uniform distribution of images across classes ensures that the dataset provides a comprehensive evaluation of a model's ability to recognize handwritten digits. Our experimentation and analysis of CNN models optimized using Genetic Algorithm and PSO is

Figure 3.1 – Random samples of each image class in dataset

based on this dataset.

3.3.1 Our Proposed Preprocessing for MNIST dataset

As part of our project, we applied a preprocessing technique to the MNIST dataset in order to enhance the dataset used for training. Specifically, we used a skeletonization function from Python to transform the handwritten digits into their thinned, skeletal forms. This preprocessing step modifies the original dataset to a thin-line representation, potentially improving the model's ability to recognize and differentiate between different digits.

Purpose and Rationale :

The skeletonization process is designed to strip away extraneous pixel data, leaving only the essential structure of each digit. This could prove especially advantageous

for
the field of machine learning models, as it reduces noise and focuses on the

most critical
features of the input data. By converting each digit to its skeleton form, we aim to make it easier for the CNN model to identify and classify digits based on their core shapes and patterns. Furthermore, this technology improves the model's ability to detect thin numbers. The preprocessed version makes the numbers look smaller and more concise.
This may result in the loss of some pixels, resulting in thinning or even disappearance of
parts of numbers. With this transformation, the model must learn and recognize numbers even when they are reduced to their basic structure. As a result, the model will be more durable and accurate in classifying accurate and complex numbers.

40

Chapter 3 Model Development and Performance Evaluation

Implementation Details :

We implemented the skeletonization using the skimage library in Python, which provides robust tools for image processing. The skeletonization function works by iteratively thinning the image until only the skeletal structure remains. Here's a brief outline of the

steps involved :

– Image Loading : Load the original MNIST images.
– Thresholding : Convert the images to binary

format where the pixel values are either
0 or 1.
– Skeletonization : Apply the skeletonization algorithm to produce the thinned, ske letal version of each digit.
– Dataset Expansion : Increase the dataset size by augmenting the skeletonized images with slight variations to improve model robustness.

Dataset Overview :

After preprocessing, the new dataset contains 140,000 images, 120,000 for training and 20,000 for testing. This expansion and transformation of the dataset aim to provide a more robust training set, helping the model to generalize better. Additionally, we use

20% of the 120,000 training images as validation data.
Visual Comparison :
Below are examples of the dataset before and after the proposed preprocessing :
Figure 3.2 – Example of the dataset before and after the proposed preprocessing
41

Chapter 3 Model Development and Performance Evaluation

3.4 CNN models presentation

In this work, we utilized two optimization algorithms : Genetic Algorithm

and Particle Swarm Optimization (PSO). These algorithms were employed to optimize the performance of our

model, ensuring efficient and effective training. We present these algorithms in detail to highlight their roles in improving the model's performance.

3.4.1 Genetic Algorithm (GA)

The concept of genetic algorithms is based on natural selection, and

is used to find optimal solutions to complex problems. Here's a precis of the way it works :

– Population Initialization : We start by generating an initial population of hyper parameters randomly. A unique combination

of hyperparameter values characterizes every individual in the population.
– Fitness Evaluation : Each individual in the population is evaluated using a fitness function. Our fitness measure is the accuracy of the CNN model using the test data.
– Parent Selection : The individuals in the population with the highest accuracy are selected to become the parents of the next generation.
– Crossover : The selected parents are combined to create a

new generation of indivi duals. Crossover involves taking the average of the hyperparameters of two parents to produce a child.

- Mutation : A small percentage of the new population is subjected to mutation, where minor random changes are made to the hyperparameters. This helps maintain genetic diversity within the population and avoids local minima.
- Iterations : The steps of selection, crossover, and mutation are repeated over seve ral generations. With each generation, the population evolves towards the optimal combination of hyperparameters.
- Final Result : After a certain number of generations, the algorithm returns the best individual from the last generation, which is the combination of hyperparameters that yielded the fine outcomes in phrases of accuracy.

42

Chapter 3 Model Development and Performance Evaluation

3.4.2 The Particle Swarm Optimization Algorithm (PSO)

In PSO, the optimal population size is determined by the social behavior of birds flocking or fish schooling. Here's a summary of how it works :

- Initialization : A swarm of particles is initialized, with each particle representing a potential solution (learning rate, regularization parameter, and dropout rates).
- Objective Function : In our case, the CNN model's negative accuracy on the test data is used to evaluate each particle's position. The goal is to minimize this

objective function, thereby maximizing the model accuracy.

– Velocity and Position Update : Every particle adjusts its position in the search space by updating its velocity. The update is influenced by its own best-known position and the best-known positions of its neighbors. This process mimics the social aspect of swarming behavior.

This step is implicitly handled by the ParticleSwarm() optimizer in the Optim library, which updates the particles' positions and velocities based on the objective function evaluations.

– Iterative Optimization : Iteratively updating positions and velocities, the particles gradually achieve the optimal hyperparameter set. During each iteration, the fitness of the new positions is evaluated, and the particles' personal best and the global best positions are updated accordingly.

– Convergence : After a specified number of iterations, the algorithm converges on the set of hyperparameters that yield the highest accuracy. The best hyperparameters found by the swarm are then used to train the final model.

3.4.3 Models Architecture

CNN models in this study consist of several convolutions, pooling, and fully connected layers. The architecture of the model is designed to efficiently extract features from the input images and perform classification. Below is a detailed description of each layer in the model :

– Input Layer : 28x28 grayscale images.

Figure 3.3 – CNN architecture optimezed by GA

- First Convolutional Layer : 6 filters, each of size 5x5, followed by ReLU activa tion.
- First MaxPooling Layer : 2x2 pooling size.
- Second Convolutional Layer : 16 filters, each of size 5x5, followed by ReLU activation.
- Second MaxPooling Layer : 2x2 pooling size.
- Flatten Layer : Converts the 2D matrix information to a vector.
- First Dense Layer : 120 units, followed by ReLU activation.
- Dropout Layer :In our models, the Dropout rate typically ranges between 0 and 0.5, depending on the hyperparameters of each specific model.
- Second Dense Layer : 84 units, followed by ReLU activation.
- Second Dropout Layer : In our models, the Dropout rate typically ranges between 0 and 0.5, depending on the hyperparameters of each specific model.
- Output Layer : 10 units (one for each class)

This model consists of 44,426 parameters, all of which are trained to recognize handwritten numbers.

3.4.4 Hyperparameters for each model

We present in this section the hyperparameters obtained from GA and PSO for our CNN models. The key hyperparameters identified include learning rate, weight decay, and dropout.

CNN models with traditional MNIST dataset

1. CNN model optimized by GA

   Here, we present the CNN model hyperparameters optimized by genetic algorithm. The table below (table 3.2)lists the hyperparameters, their respective ranges, and the best values identified during the optimization process.

Hyperparameter	Range	Best Value
Learning Rate	[0.0001 : 0.001]	0.0004
Weight Decay	[0 : 0.001]	0
Dropout 1	[0 : 0.5]	0.1
Dropout 2	[0 : 0.5]	0.1

Table 3.2 – Optimized Hyperparameters with the Genetic Algorithm and Their Best Values

2. CNN model optimized by PSO

   Here, we present the CNN model hyperparameters optimized by PSO algorithm. The table below (table 3.3)lists the hyperparameters, their respective ranges, and the best values identified during the optimization process.

Hyperparameter	Range	Best Value
Learning Rate	[0.0001 : 0.001]	0.0001
Weight Decay	[0 : 0.001]	0

Dropout 1	[0 : 0.5]	0.2
Dropout 2	[0 : 0.5]	0.1

Table 3.3 – Optimized Hyperparameters with PSO and Their Best Values

CNN models with enhansed MNIST dataset

1. CNN model optimized by GA

Here, we present the CNN model hyperparameters optimized by GA. The table

below (table 3.4)lists the hyperparameters, their respective ranges, and the best values identified during the optimization process.

Hyperparameter	Range	Best Value
Learning Rate	[0.0001 : 0.001]	0.0003
Weight Decay	[0 : 0.001]	0
Dropout 1	[0 : 0.5]	0.14
Dropout 2	[0 : 0.5]	0.1

Table 3.4 – Optimized Hyperparameters with the Genetic algorithm and Their Best Values

2. CNN model optimized by PSO

Here, we present the CNN model hyperparameters optimized by PSO algorithm. The table below (table 3.5)lists the hyperparameters, their respective ranges, and the best values identified during the optimization process.

Hyperparameter	Range	Best Value
Learning Rate	[0.0001 : 0.001]	0.0001
Weight Decay	[0 : 0.001]	0
Dropout 1	[0 : 0.5]	0.2
Dropout 2	[0 : 0.5]	0.15

Table 3.5 – Optimized Hyperparameters with the PSO algorithm and Their Best Values

3.5 Experimental material and platforms

3.5.1 Julia

Julia is an open source high-level, high-performance dynamic programming language designed at MIT for large-scale, partial-differential equation simulations and distributed
linear algebra.
Julia's ability to support scientific computing makes it a good choice for designing machine learning models and AI simulations.

Compared to other platforms, Julia is known for being easy to use. Additionally, it is acknowledged for its speed comparable to C, dynamic nature akin to Ruby, general capabilities like Python, statistical friendliness similar to R, powerful performance in linear algebra like Matlab, and natural proficiency in string processing like Perl.

3.5.2 google colab

Google Colab is a cloud-based platform provided by Google for collaborative coding, where users can write, execute, and share code. It offers access to various computing resources such as CPU, GPU, and TPU. Users can run machine learning models, analyze data, and perform computational tasks using Jupyter notebooks without the need for local setup or

installation.

3.5.3 Flux

Flux.jl is a powerful and flexible machine learning library in Julia, known for its simplicity

and high performance. It offers an intuitive API for model building and training, utilizing Julia’s JIT compilation for efficient execution. Flux.jl integrates well with other Julia packages like CUDA.jl for GPU acceleration, making it suitable for both research and production.

3.5.4 Plots

The Plots package in Julia is a versatile, high-level plotting tool designed to work seamlessly with multiple plotting backends. Its goal is to provide powerful functionality while remaining intuitive, allowing users to create sophisticated visualizations with minimal code.

3.5.5 Images

Julia’s Images package offers a complete framework for image processing and computer vision tasks. It offers a rich set of tools for loading, manipulating, and analyzing images with ease and efficiency. The package supports a wide variety of image formats and integrates seamlessly with Julia’s ecosystem for scientific computing. Images.jl is highly extensible, allowing users to implement custom algorithms and pipelines for their specific needs. Both academic research and practical applications in image analysis can benefit from its powerful combination of performance and flexibility.

3.6 Results and discussion

3.6.1 CNN models with traditional MNIST dataset

We discuss the results achieved using these hyperparameters, including the accuracy and loss curves, the ROC curve, the confusion matrix, and the classification report.

CNN model with GA

First of all, we achieved an impressive test accuracy of 98.92%. The results clearly indicate that the CNN model trained with optimized hyperparameters is effective.

To provide a comprehensive evaluation of the model's performance, we present the following results :

1. Accuracy and Loss Curves

In the figure 3.4 , both training and test sets have accuracy curves over 20 epochs.

The training accuracy represented by the blue line, while the red line represents the test accuracy. There is a slight dip in accuracy at first, which is normal as the model learns and adjusts its weights. The training accuracy quickly improves and stabilizes around 99.8%, indicating that the model is learning well from the training data. The test accuracy shows some fluctuations but generally remains high, around 98.9%, demonstrating good generalization to unseen data.

48

Figure 3.4 – Accuracy Curve of the Genetic Algorithm

Figure 3.5 – Loss Curve of the Genetic Algorithm

Figure 3.5

shows the loss curves for both the training and validation sets over 20 epochs. The blue line represents the training loss, while the red line represents the

validation loss. Initially, both training and test loss are relatively high, with the validation loss showing more variability. As training progresses,

the training loss decreases steadily, reaching a low value of 0.004 at the end of the 20 epochs. The validation loss

also decreases but shows more fluctuations compared to the training loss, ending at a value of 0.05. Overall, the model demonstrates strong performance, with both training and test losses significantly reduced by the end of the training period.

2. Confusion Matrix

The confusion matrix for the model is shown

in Figure 3.6.
The diagonal elements of the confusion matrix represent the instances that were
Figure 3.6 – Confusion matrix
correctly classified by the model.
The off-diagonal elements represent the misclassified instances, providing insight into specific classes where the model may have difficulty distinguishing between similar digits.
In general,

the confusion matrix substantiates the high performance of the CNN model optimized with the genetic

algorithm in accurately classifying handwritten digits.

3. ROC curve

The provided ROC curve graphically represents

the performance of a classification model, with the x-axis denoting the False Positive Rate (FPR) and the y-axis representing the True Positive Rate (TPR). This blue

ROC curve illustrates a high TPR over various thresholds and a low FPR over lower thresholds, as indicated by a vertical ascent at the beginning and a horizontal run at the top. This signifies exceptional model performance, close to the top left corner of the plot, which is characteristic of a nearly perfect classifier. The dashed diagonal line serves as a ba seline, representing random guessing where TPR equals FPR. In comparison to a

random guess classifier, the model shows excellent discriminatory power because the ROC curve lies near the top left corner.Show figure 3.7 for a visual representation of the ROC curve, which clearly demonstrates the model's high performance and excellent ability to distinguish between positive and negative classes.

Figure 3.7 – ROC Curve of the Genetic Algorithm

4. Classification Report

The classification report gives a comprehensive analysis of performance metrics for every class in the digit recognition

task. These metrics include precision, recall, and F1-score, which are essential for evaluating the accuracy and effectiveness of the model. The figure below shown the results :

Figure 3.8 – Classification report of the Genetic Algorithm

The model performs

exceptionally well across all classes, as indicated by the overall precision, recall, and F1-score values of 0.9892. Specifically, the presicion is 0.9893.

The report indicates that the model maintains a high level of performance consistently across all digit classes, with slight variations. For instance, class 4 has the highest precision (0.9990), while class 9 has the lowest precision (0.9783). Nevertheless, all classes exhibit strong performance metrics, demonstrating the model's robustness and accuracy in recognizing handwritten digits.

CNN model with PSO

First of all, we achieved an impressive test accuracy of 98.98%. It is clear from these results

that the CNN model trained with the optimized hyperparameters is effective.

For the purpose of providing a comprehensive evaluation of the performance of the model, we present the following results :

1. Accuracy and Loss Curves

The graph provided shows the accuracy of a CNN model optimized using the

PSO
algorithm over 20 epochs, with separate lines representing training and test accuracy. The training accuracy starts at approximately 98% and consistently increases, reaching around 99.8% by the 20th epoch. The model improves its performance with each epoch as it learns from the training data. The validation accuracy also shows a positive trend, starting near 97.7% and reaching approximately

98.95% by the
20th epoch. The results demonstrate that the PSO algorithm effectively optimizes the

CNN model, achieving high accuracy on both training and validation datasets.
The final accuracies of 99.8% for training and 98.95% for validation data reflect the
model's strong learning. Show figure 3.9 to visualize the accuracy trends.

Figure 3.9 – Accuracy curve of PSO algorithm

Initially, both training and validation loss values start relatively high, around 0.05
and 0.07 respectively. By the 20th epoch, the training loss has decreased steadily, reaching approximately 0.005, indicating that the model is effectively minimizing error. The validation loss also shows a significant reduction, reaching around 0.03 by the 20th epoch. Overall, the graph shows that the PSO algorithm effectively reduces both training and validation losses, with the final values indicating a strong
learning capability and good generalization performance of the CNN model. Show figure 3.10

Figure 3.10 – Loss curve of PSO algorithm

2. Confusion Matrix

The confusion matrix shows the model's performance in classifying handwritten digits from the MNIST dataset when optimized using the Particle Swarm Opti mization (PSO) algorithm. Classification accuracy is high across all classes, with diagonal values close to 100%. For instance, the correct classifications for digit ’0’ are 977 out of 980, and for digit ’1’, they are 1130 out of 1135. Misclassifications are minimal, as indicated by the low off-diagonal values. For example, digit ‘3’ is misclassified as digit ‘5’ only 4 times, and digit ‘8’ as digit ‘3’ 3 times.

Some specific misclassification trends are observed, such as digit ‘5’ being confused with digit ‘3’ 9 times and with digit ‘6’ 1 time. Digit ‘4’ is misclassified as ‘9’ 14 times, which is slightly higher compared to other classes. Despite these minor misclassifications, the overall performance is highly consistent, with the majority of predictions being correct.Show figure 3.11.

Figure 3.11 – Confusion Matrix of PSO algorithm

3.

ROC Curve

The ROC curve for the PSO-optimized model demonstrates excellent performance, with the curve

closely following the left-hand border and the top border

of the plot,

indicating a high true positive rate and a low false positive rate.

Its near-perfect classification ability is evident in an AUC (area under the ROC curve) close to 1.0, demonstrating high precision, recall, and overall performance.Show figure 3.12.

Figure 3.12 – ROC Curve of PSO algorithm

4. Classification Report

The classification report for the PSO-optimized CNN model demonstrates exceptio nal performance in classifying handwritten digits from the MNIST dataset. In all classes, the model achieves high precision, recall, and F1-scores, with overall metrics of 0.9898. This indicates the model's accuracy in making correct predictions and its effectiveness in identifying positive instances. The consistently high scores across all digit classes reflect the model's robustness and reliability in distinguishing bet ween different handwritten digits with minimal errors. Figure 3.13 provides detailed values for each class.

Figure 3.13 – Classification Report of PSO algorithm

3.6.2 Comparision of results

To compare the performance of the two CNN models, one optimized using a genetic algorithm (GA) and the other using particle swarm optimization (PSO), we examine the provided metrics and the accompanying bar chart. As depicted in the chart, both optimization techniques yield extremely high accuracy, precision, recall, and F1 scores. Specifically, the accuracy for the GA-optimized model is 98.92%, while the PSO-optimized model achieves a slightly higher accuracy of 98.98%.

According to the confusion matrices for both models, the GA model underperforms in some classes compared to the PSO model, which has fewer misclassifications. For instance, in class 2, the GA model misclassifies more samples compared to the PSO model. This trend is consistent across other classes, indicating that the PSO model might be better at distinguishing between different digits, especially in challenging cases.

The precision, recall, and F1 scores for both models are also extremely close, with the PSO model having marginally better scores across these metrics. This suggests that while both models perform exceptionally well, the PSO model has a slight edge in terms of generalizing better to the test data.

Overall, the comparison highlights that while both optimization algorithms are effective, PSO provides a marginal improvement in performance metrics. The provided image further visualizes this comparison, showing the detailed breakdown of these metrics for each model. Here is figure 3.14 presenting the detailed breakdown of these metrics :

Figure 3.14 – Comparison of CNN Model Performance with different optimization algorithms

3.6.3 CNN models with enhanced MNIST dataset

CNN model with GA

1. Accuracy and Loss Curves

Graph showing in figure 3.15 CNN accuracy over 20 epochs for training and vali dation datasets. The training accuracy, depicted by the blue line, starts at approxi mately 97.66%, steadily increasing to around 99.5% by the 20th epoch, indicating that the model is effectively learning from the training data. The validation accu racy, shown by the orange line, begins at about 96.6% steadily increasing to around 98.4%. This demonstrates that the model is not only learning well from the training data but also generalizing effectively to the validation data.

Figure 3.15 – Accuracy curve of Genetec Algorithm

In Figure 3.16, we show the CNN's loss values over 20 epochs for both the training and validation datasets. The training loss, represented by the blue line, starts at a low value and continues to decrease, with minor fluctuations, ending around 0.008 at the 20th epoch.

Based on this consistent decline, the model is effectively minimizing the training data error. The

validation loss, shown by the orange line, begins at a hi gher value (0.1), reaching approximately 0.06 by the 20th epoch. Overall, the model demonstrates strong performance, with both training and test losses significantly reduced by the end of the training period.

Figure 3.16 – Loss curve of Genetec Algorithm

2. Confusion matrix

Overall, the confusion matrix of this model indicates that the model is performing well, with most samples correctly classified. However, there are classes, particularly classes 3, 5, and 9, where classification errors are slightly more pronounced. This could indicate similarities between these classes or shared features that make the distinction more challenging for the model.The figure 3.17 represents the detailed confusion matrix.

Figure 3.17 – Confusion matrix of Genetec Algorithm

3. ROC curve

Based on the ROC curve(show figure 3.18), the blue line shows

the performance of

the classification model, while the dashed line shows the results of random guessing.

It plots the True Positive Rate (TPR) towards the False Positive Rate (FPR). The curve begins at (0,0)

, sharply rises to (0,1), then runs parallel to the y-axis to (1,1), indicating perfect performance where TPR equals

1 and FPR equals 0. The Area Under the Curve (AUC) measures the model's ability to differentiate between positive and negative classes, with 1 representing perfect classification.

As the ROC curve closely hugs the top-left corner, its AUC is nearly 1, indicating near-perfect performance. The dashed line acts as a baseline for random guessing, which the blue curve significantly surpasses. According to the ROC curve, the classification model performs exceptionally well, nearly perfectly differentiating positive from negative classes with a high level of accuracy.

Figure 3.18 – ROC curve of Genetec Algorithm

4. Classification report

In the classification report represented by the figure 3.19, each class has its precision, recall, and F1-score calculated, reflecting how well the model predicts each specific class. For most classes, the precision and recall values are very high, typically above

0.98, indicating that the model is highly accurate in its

predictions and correctly
identifies true positives while minimizing false positives and false negatives.
Class 0 through Class 9

show minor variations in their scores,
with Class 5 and Class
9 having slightly lower precision and recall compared to the other classes. Despite
these minor discrepancies, the overall precision, recall, and F1-score for the model
stand at 0.9845, showcasing the model's robust performance and generalization ca
59

Chapter 3 Model Development and Performance Evaluation
pabilities across the dataset. This high overall performance metric suggests that
the
model is reliable and performs consistently well across different classes.
Figure 3.19 – Classification report of Genetec Algorithm

CNN model with PSO

1. Accuracy and Loss Curves

The graph illustrates the accuracy of a CNN model

over 20 epochs, depicting both
training and validation accuracy. The training accuracy, represented by the blue
line,
starts around 97.9% and shows a steep increase during the first few epochs, leveling
off around 99.5% by the 20th epoch, suggesting effective learning from the training
data. The orange line, which represents the validation accuracy, begins close to
97.6% and gradually increases to around 98.6% by the 20th epoch. Despite showing
minor fluctuations, this suggests some variability in the model's performance when
dealing with unseen data. These accuracy curves demonstrate that the CNN model
achieves high accuracy on both training and validation datasets. The final
accuracies
of 99.5% for training and 98.6% for validation data reflect the model's strong

learning and generalization capabilities. Show figure 3.20 to visualize the model's learning process and performance trends over the epochs.

Figure 3.20 – Accuracy curve of PSO algorithm

The "Loss vs Epoch" graph (figure 3.21) shows the training and validation loss of a CNN model over 20 epochs. The training loss, depicted by the blue line, decreases significantly from approximately 0.06 at the first epoch to around 0.01 by the 20th epoch, indicating effective learning

and reduction of errors on the training data. The validation loss, represented by the orange

line, also decreases initially but shows some fluctuations between epochs 10 and 20, settling at approximately 0.045 by the final epoch. These fluctuations suggest variability in the model's performance on unseen data. Despite this, the overall trend of decreasing validation loss indicates that the model is generalizing well, although not as smoothly as on the training data. The final epoch shows a validation loss of 0.045 and a training loss of 0.01.

Figure 3.21 – Loss curve of PSO algorithm

2. Confusion matrix

The confusion matrix shows the model's performance in classifying handwritten digits from a dataset. Classification accuracy is high across all classes, with most values along the diagonal indicating correct classifications. For instance, digit '0' is correctly classified 1952 times out of 1960, digit '1' is correctly classified 2247 times out of 2270, and digit '2' is correctly classified 2028 times out of 2064. Misclassifica tions are minimal, with off-diagonal values indicating errors. For example, digit '4' is misclassified as digit '9' 13 times. Overall, the model shows strong performance, with the majority of predictions being accurate and only a few misclassifications present. Show figure 3.22 to visualize the distribution of predictions

Figure 3.22 – Confusion matrix of PSO

algorithm

3. ROC curve

A high true positive rate and a low false positive rate are evidenced by the ROC curve of the PSO-optimized model, which is closely tracing the top and left

borders

of the plot. Its near-perfect classification capability is evident with an AUC (Area

Under the ROC Curve) approaching 1.0.Show figure 3.23

Figure 3.23 – ROC curve of PSO algorithm

4. Classification report

We note from the classification report shown in

the figure 3.24, the precision values

ranging from 0.9669 to 0.9951, indicate the proportion of

true positive predictions

among all positive predictions made for each class. Recall values, ranging from 0.9757

to 0.9959,

reflect the proportion of true

positives cor- rectly identified out of all actual

positives for each class. The

F1-scores, combining precision and recall, range from 0.9781 to 0.9931, providing a balanced measure of the model's accuracy for each class. The overall metrics show an excellent perfor- mance with precision, recall, and F1-score all at 0.9852. Figure 3.25 presents the detailed breakdown of these metrics for each class.

Figure 3.24 – Classification report of PSO algorithm

63

Chapter 3 Model Development and Performance Evaluation

3.6.4 Comparision of results

As shown in the provided results and chart, there is a subtle but notable difference between

the two models optimized with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The PSO-optimized CNN model slightly

outperforms the GA-optimized model across all evaluated metrics. Specifically, the PSO model achieves an accuracy of 98.52%, marginally

higher than the GA model's 98.46%. Similarly, the precision, recall, and F1 score for the

PSO model all stand at 98.5%, compared to 98.4% for the GA model.

These consistent improvements suggest that the PSO model handles certain classes better, as evidenced by fewer misclassifications in the confusion matrices. The bar graph visually corroborates this, showing nearly identical performance with the PSO model having a slight edge in all metrics. As a result, while both optimization techniques are effective, the PSO-optimized model shows marginally superior performance, suggesting that PSO might be a more effective optimization algorithm for this particular CNN model and dataset.Show figure 3.25.

Figure 3.25 – Comparison of CNN Model Performance with Genetic Algorithm and Particle Swarm Optimization

3.7 Comparision of models

On the chart (figure 3.26), four models are compared based on their accuracy and loss : PSO+MNIST, GA+MNIST, PSO+newMNIST, and GA+newMNIST. The first two models use the standard MNIST dataset, while the latter two use an enhanced dataset with preprocessing through skeletonization, resulting in a new dataset of 140,000 samples.

Observing the chart, we see that PSO+MNIST has an accuracy of 0.9900 and a loss of 0.0371, GA+MNIST has an accuracy of 0.9892 and a loss of 0.0478, PSO+newMNIST has an accuracy of 0.9851 and a loss of 0.0518, and GA+newMNIST has an accuracy of 0.9812 and a loss of 0.0783. The models trained on the enhanced dataset (PSO+newMNIST and GA+newMNIST) show a slight decrease in accuracy but are trained on a more diverse set of shapes due to the preprocessing. This means they can recognize a wider variety of patterns and are more robust to variations in the data. In practical terms, although the losses are slightly higher, these models are better because they generalize more effectively and recognize a broader range of patterns. This makes them particularly suited to applications where varied pattern recognition is crucial.

In conclusion, despite a slight increase in loss, the PSO+newMNIST and GA+newMNIST models are better because they benefit from the diversity and richness of the enhanced dataset, improving their generalization ability and performance in varied real-world scenarios.

Figure 3.26 – Comparision of models

3.8 Comparision with other works

The following table (3.6) presents a comparison with other works in the field

Reference	dataset	model	accuracy
[?]	MNIST	CNN+KNN	98.80
[?]	MNIST	CNN	99.2
[?]	MNIST	CNN+gabor	98.78
[?]	MNIST	CNN	98.16
[?]	MNIST	MLP	97.32
[?]	MNIST	CNN	98.86
Our proposed method	MNIST	CNN+GA	98.92
	MNIST	CNN+PSO	98.98
	MNIST+preprocessing	CNN+GA	98.45
	MNIST+preprocessing	CNN+PSO	98.51

Table 3.6 – Comparision with other works

3.9 Conclusion

The conclusion of this chapter highlights the successful implementation and evaluation of CNN models. By comparing Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), we demonstrate the effectiveness of our approach and the significant potential of these optimization methods to enhance CNN-based recognition systems. This comprehensive overview provides valuable insights into the benefits of our preprocessing method and the strengths of different optimization techniques in the context of handwritten digit recognition.Furthermore, for our platform presented in the upcoming chapter, we have chosen the CNN model optimized with PSO and the new dataset.

Chapitre 4

Interfaces and Test

4.1 Introduction

In this chapter, we will discuss the final part, which represents the implementation of our project, based on the mechanisms mentioned above in the Third chapter(CNN model optimized with PSO and the new dataset). This chapter consists of two parts : the

first

presents the environment of our program, as well as the results of the tests

that were

conducted.

4.2 Platform Overview

In the following illustration, we show how our system predicts an image by showing each layer it must pass through 4.1. Beginning with image acquisition followed by preprocessing steps such as binarization and skeletonization. The preprocessed image undergoes segmentation into lines, words, and digits, which are then subjected to prediction and location identification. Finally, the prediction results and digit locations are concatenated to generate the final recognized output.

Figure 4.1 – General diagram of our platform system

4.3 Development Tools

VS Code :Visual Studio Code is a source code editor and an integrated development environment (IDE) of Microsoft. It is open-source and cross-platform, meaning it runs on Windows, Linux and Mac. It was designed for web developers, but it supports many other programming languages such as C++, C#, Python, Java, etc. It offers many features

Test
like syntax highlighting, auto-completion, error highlighting, code navigation, debugging,
versioning, integration with Git, and many more. It is also extensible using a wide variety
of extensions developed by the community, allowing developers to customize the editor according to their needs.

4.4 Configuration Used in the implementation :

The configuration of the hardware used in our implementation is :

– ASUS VivoBook Core i7-8550U CPU @ 1.80GHz 1.99 GHz.
– RAM size 16 GB.
– 500GB Hard Drive Size.
– Windows 64-bit Exploitation System.

4.5 Server-Side Development

To ensure that our platform works effectively, we use a range of techniques and technologies,

including

4.5.1 RESTful API

A REST API (also known as RESTful API) is an application programming interface (API

or web API) that conforms to the constraints of REST architectural style and allows
for interaction with RESTful web services.
We use RESTful APIs to integrate all the components of our application. Our platform utilizes multiple programming languages, including Julia, Python, and JavaScript. RESTful APIs provide a standardized and efficient way to enable communication and data exchange between these diverse components. This approach allows us to maintain flexibility and scalability while ensuring seamless interoperability across different parts of our
system.

4.5.2 RESTful APIs in Julia

This language is fast, dynamic, and fits the needs of a wide variety of platforms, paradigms, and programming paradigms. Visit its website for details on its science, vi
69

Chapitre 4 Interfaces and Test

sualization, data science, and machine learning domains, as well as events and the open

source ecosystem.
In our application, Julia is employed primarily for server-side functionalities, model management, image preprocessing, and prediction tasks. Specifically, we use :

1. Genie.jl :
   We utilize Genie.jl to ensure that our data transfers correctly and efficiently when communicating with our Python API. By employing RESTful APIs, we facilitate seamless and reliable data exchange between Genie.jl and Python.
2. BSON : Used to load our pre-trained machine learning model (our CNNs model) stored in a BSON file.
3. Images and related libraries :These are used for loading, converting, and pre processing images to prepare them for prediction.
4. Flux : This library facilitates the prediction process using the loaded model.
5. JSON : Used for encoding the prediction results into JSON format, making it easy to return responses from the server.

4.5.3 RESTful APIs in Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python is utilized extensively in our FastAPI application for various tasks, including :

1. FastApi is a modern, fast (high-performance), web framework for building APIs

with Python based on standard Python type hints.

2. Image Processing

: Python's OpenCV library (cv2) is employed for image proces sing tasks such as resizing, rotating, converting to grayscale, and applying various filters. These operations are crucial for extracting text from images and enhancing image quality.

3. HTTP Requests : Python's requests library is used to make HTTP requests to an external server endpoint (julia server) to obtain predictions for processed images.

70

Chapitre 4 Interfaces and Test

4. Server Configuration and Deployment : Python is used to configure the server settings and start the FastAPI application using the uvicorn ASGI server. Additionally, the application is designed to run on the specified host and port for deployment.

4.6 Client-Side Development

In our application, the client-side development focuses on creating an intuitive, responsive, and dynamic user interface. We leverage modern frontend technologies to ensure a seamless user experience. Below are the key technologies and techniques we employ :

4.6.1 React.js for the Frontend

1. Introduction to React.js : React.js is a popular JavaScript library for building user interfaces, particularly for single-page applications where data dynamically changes over time.React.js allows developers to create large web applications that can update and render efficiently in response to data changes.
   In our application, we use React.js to build an intuitive and responsive user interface.
   Specifically, React.js enables us to Handle API Responses and Display Prediction Results.
2. Integration with RESTful APIs Making HTTP calls to the backend server and changing the user interface based on the result is how a React app consumes a RESTful API. The fundamental procedures for using a RESTful API in React are as follows :
   – Install an HTTP request library, such as Axios.
   – Specify the RESTful API endpoint URLs that you want to use.
   – In your React project, create a component that will make the API call. This can be a class-based component or a hook-based functional component.
   – Make a request to the relevant endpoint using the HTTP request library, handing in any necessary parameters or data.

–

Process the API response by changing the state of the component with the obtained data.

If an error occurs, respond correctly to the error status.

– Render the changed state in the user interface of the component.

4.6.2 Tailwind Css

Tailwind CSS is a CSS framework that provides a set of utility classes, enabling developers to create a wide range of styles without writing custom CSS. Instead of defining CSS rules, you apply utility classes directly to your HTML elements. This approach streamlines the development process and offers great flexibility for customization.
In our application, we use Tailwind CSS to Rapidly Develop Styles,Customize Easily,Reduce CSS Overhead.

4.7 Platform components

The image below 4.2 shows the interface of the OCR Platform. It consists of various components that play a specific role in facilitating OCR (Optical Character Recognition) and enhancing user interaction.

Figure 4.2 – Platfotm Components

1. Header
   – Title : “THE OCR PLATFORM”
2. Options Panel (Left Sidebar)
   – Crop all : A button to crop all selected images.
   – Right rotation : A button to rotate the selected area to the right with 8 ◦ .

- Left rotation : A button to rotate the selected area to the left with 8 ◦ .
- Smart Erase : A button to intelligently erase parts where the user select in the image.

3. Main Workspace (Center)
   - Image Upload Area : A central area where users can drop images or click to upload them. This is where the image to be processed will be displayed.
   - Predict it : A button to run the OCR prediction on the selected area from the image.
   - Show Output Image : A button to run the image after the preprocessing and the segmentation.
   - Fetch Predictions : A button to fetch text predictions from the OCR process.
4. Filters Panel (Right Sidebar)
   - Remove Shadows : A button to remove shadows from the image.
   - Lighten : A slider to adjust the lightness of the image.
   - Magic Color : A slider to enhance the colors in the image.
   - Binarize :Convert the image to black and white by specifying a brightness level cutoff.
5. Bottom Action Bar
   - Save the coordinates : A button to save the coordinates of the processed image area in .txt file.

4.8 Test and Implementation

We design our platform to be simple and easy to use and in this section we will explain :

the steps to get the prediction of an image with handwritten numbers.The flowchart below 4.3 illustrates the HDR process : starting from opening the platform, optionally adjusting image settings, selecting the area, predicting, and finally fetching predictions or displaying the output image.

73

Chapitre 4 Interfaces and Test

Figure 4.3 – Steps to the HDR

Step 1 : Open the HDR PlatformLoad the image with handwritten num bers into the HDR platform. You will see the initial screen as shown in Figure 4.4

Figure 4.4 – First step to the HDR

74

Chapitre 4 Interfaces and Test

Step 2 : Adjust Image Settings (Optional)

–

Use the filter options on the right to remove shadows, lighten the image, adjust the

magic color, or echo if needed (Figure 4.5) in our test we use the Binrize slider to make the image black and white.

Figure 4.5 – Second step to the HDR

Step 3 : Select the Area

– Draw a box around the area containing the handwritten numbers to select it. This area should encompass all the digits you want the OCR to recognize.In our test we used Crop all button to select the full image from the options on the left (Crop all, Right rotation, Left rotation, Smart Erase) they are used if any adjustments to the selected area or image are needed.e (Figure 4.6).

75

Figure 4.6 – Theerd step to the HDR

Step 4 : Fetch Predictions and Show Output Image

– Click the "Predict it" button to process the selected area and click the "Fetch Prediction" button to get the HDR predictions for the handwritten num bers,and click on the "Show Output Image" button to show the image after the segmentation and the preprocessing(Figure 4.7).

Figure 4.7 – Forth step to the HDR

4.9 Conclusion

This chapter discusses the implementation of our project in detail, detailing the various components and technologies used. A brief overview of the platform was presented, as well as a discussion of the development tools and configurations employed, as well as a clarification of its operational mechanisms. As a result of the tests conducted, our implementation has proven to be effective and robust, demonstrating that our project has practical applications.

In future work, we propose the development of a new feature that will allow users to input a template that recognizes only variable characters and integrates them into PDF documents. As a result of this enhancement, our platform will be able to provide even greater functionality and applicability.

General Conclusion

As a conclusion, this report presents a comprehensive evaluation of different Convolutional Neural

Network (CNN) models for the recognition of handwritten digits using the MNIST dataset. Results demonstrate that optimization and preprocessing techniques can improve the performance of models. On the standard MNIST dataset, the CNN model using Genetic Algorithm (GA) achieved 98.92% accuracy, while the Particle Swarm Optimization (PSO) model achieved 98.98% accuracy.

Moreover, our novel preprocessing technique, involving skeletonization, resulted in 98.45% accuracy for CNN models with GA, and 98.51% accuracy for CNN models with PSO. Although traditional MNIST dataset models produce slightly higher accuracy scores, enhanced MNIST dataset models with skeletonization are considered superior due to the diversity and richness of the enhanced dataset. As a result of this improvement, they are now more capable of generalizing and performing in a variety of real-world situations.

Integrated analyses of GA and PSO demonstrate that both have potential in optimizing CNN-based recognition systems, with PSO demonstrating a marginal advantage. The incorporation of skeletonization as a preprocessing step, despite its benefits, suggests that further refinement and additional preprocessing strategies could lead to even greater improvements.

Looking ahead, future work on this application should focus on several key areas for the purpose of improving its robustness and versatility. Among the important directions is the implementation of new preprocessing functions, such as rotations and scalings, to enrich the dataset and improve the model's generalization capabilities. Additionally, expanding

the dataset to include letters and other characters will broaden the application's scope

and utility.On the client side, users will be able to enter a template that will recognize only
variable characters and incorporate them into a PDF document. As an example, when a postal

check is numbered, the user will scan only the variable columns, simplifying the process of numbering the check

As well, addressing the diversity of handwriting styles remains a key challenge. Future
iterations should aim to incorporate samples from a wider range of individuals to better capture the variability in handwriting. The goal is to develop a method for seamlessly integrating new handwriting samples into the training dataset, ensuring that the model can accurately recognize characters written in different styles.

By further developing these aspects, the application can be further developed to support a wider range of handwritten characters and styles. This will ultimately lead to a more robust and adaptable recognition system that can effectively handle real-world handwriting variations.

79