

ÉLABORATION D'UNE INTERFACE INTUITIVE : CLASSIFICATION D'IMAGES VIA CNN DANS UNE APPLICATION WEB

Réalisée par :

EL MAHRAOUI AMAL

FILIERE : IDDL

Demandé par :

PR. ZANNOU ABDERRAHIM

MODULE : INTELLIGENCE ARTIFICIELLE



Un rapport de projet de fin de module présenté, dans le cadre du module d'Intelligence Artificielle. Ce projet vise à développer une application web utilisant Flask pour la classification d'images à l'aide du modèle CNN
“(Convolutional Neural Network)”

24 Mars 2024

RÉSUMÉ

Ce rapport présente une étude approfondie sur le développement et l'implémentation d'un modèle de réseau de neurones convolutif (CNN) pour la classification d'images de l'ensemble de données Fashion MNIST. En se concentrant sur les étapes clés de conception du CNN, notamment le prétraitement des données, la configuration du réseau et les stratégies d'apprentissage, cette étude vise à démontrer l'efficacité du CNN dans la classification précise des images de vêtements. Les résultats détaillés de l'évaluation du modèle mettent en évidence sa performance remarquable dans la catégorisation précise des articles vestimentaires. Des recommandations sont également fournies pour améliorer les performances du modèle, notamment l'exploration de techniques d'augmentation de données et l'optimisation des hyperparamètres. Enfin, une application web Flask est développée pour offrir une interface utilisateur intuitive et pratique, démontrant ainsi l'applicabilité réelle et l'utilité de l'approche CNN dans le domaine de la classification d'images.

Mots clés : Réseau de neurones convolutif (CNN); Classification d'images; Fashion MNIST.

Sommaire

	Page
Page de garde	i
Résumé	i
Sommaire	ii
List des Figures	iii
List des Abbreviations	iv
Acknowledgment	v
 CHAPTER I	
INTRODUCTION	1
1.1 Contexte du projet et travail demandé	2
1.2 Les réseaux de neurones artificiels	2
1.2.1 Définition	2
1.2.2 Types des reseaux de neurones	2
1.3 Convolutional neural net work	3
1.3.1 Définition	3
1.3.2 Architecture d'un réseau de neurones convolutif	3
1.4 Conclusion	6
 CHAPTER II	
Classification	7
2.1 Introduction	7
2.2 Définition	7
2.3 Etapes d'une classification	7
2.4 Type de classification	8
2.4.1 Binaire	8
2.4.2 multi-class	8
2.5 Principe de fonctionnent de la classification	9
2.6 Conclusion	9
 CHAPTER III	
Implémentation	10
3.1 Environnement de réalisation	10
3.1.1 Jupyter-Notebook	10
3.1.2 Le langage Python	11
3.1.3 Biliothèques Python	11
3.2 Réalisation et résultats	12
3.2.1 Introduction	12
3.2.2 Les etapes suivies	13
 APPENDICES	18

TABLE DES FIGURES

N° figure	Description	Page
1	Architecture du modèle CNN	3
2	Architecture du modèle CNN	4
3	Principe de Max Pooling	5
4	ReLu – Unité linéaire rectifiée	5
5	Couche fully connected	6
1	Classification binaire vs classification multi-class	8
1	Jupyter-notebook	10
2	Python	11
3	Bibliothèques Python	11
4	Fashion MNIST Dataset	12
5	sous graphiques avec leurs étiquettes	14
6	Construction du modèle	15
7	Compilation du modèle	15
8	Cross entropy Loss vs Classification accuracy	16
9	Matrice de confusion	17
10	Prédiction	17
11	Interface web-1	18
12	Interface web-2	19
13	Informations sur les données	19
14	Visualisation des données	20
15	Prédiction	20

LIST DES ABBREVIATIONS

CNN	Convolutional Neural Network
MNIST	Modified National Institute of Standards and Technology

INTRODUCTION GÉNÉRAL

Ces dernières années, l'intelligence artificielle et plus particulièrement le deep learning ont gagné en popularité, notamment grâce à l'émergence des algorithmes convolutifs (CNN) qui ont permis de résoudre efficacement des problèmes complexes en apprentissage automatique. Les CNN sont devenus des structures de réseau représentatives dans le domaine du traitement et de la reconnaissance d'images.

La technologie du deep learning, incluant les CNN, a rencontré un franc succès dans le domaine du traitement et de la reconnaissance d'images en raison de sa capacité à apprendre automatiquement des caractéristiques et à exploiter des modèles complexes. Les performances des CNN ont souvent surpassé celles des méthodes traditionnelles, ouvrant de nouvelles perspectives dans le domaine de la vision par ordinateur.

Le premier chapitre détaillera l'architecture des réseaux convolutifs CNN ainsi que la composition de leurs différentes couches. Le deuxième chapitre proposera une définition de la classification, en abordant les types de classification et les critères associés.

Quant au dernier chapitre, il sera dédié à la partie pratique, à savoir la classification d'images du jeu de données Fashion MNIST à l'aide d'un CNN, en utilisant le langage Python et le framework Flask.

CHAPITRE I

INTRODUCTION

Les réseaux de neurones convolutifs sont spécifiquement conçus pour traiter des données structurées en grille, telles que les images, qui présentent des dépendances fortes entre leurs éléments. Le Deep Learning s'est avéré être un outil puissant au cours des dernières décennies, en grande partie grâce à sa capacité à gérer des volumes massifs de données. L'introduction de couches cachées a révolutionné les approches traditionnelles, notamment dans le domaine de la reconnaissance de formes. Parmi les architectures de réseaux de neurones profonds les plus influentes, les réseaux de neurones convolutifs (également connus sous le nom de CNN ou ConvNet) occupent une place prépondérante, particulièrement dans les applications de vision par ordinateur.

Depuis les années 1950, les premiers jours de l'intelligence artificielle, les chercheurs ont rencontré des défis considérables dans le développement de systèmes capables de comprendre les données visuelles. Au fil du temps, ce domaine a évolué pour devenir ce que l'on connaît aujourd'hui sous le nom de vision par ordinateur. En 2012, un jalon majeur a été franchi dans ce domaine lorsque des chercheurs de l'Université de Toronto ont développé un modèle d'IA surpassant de loin les meilleurs algorithmes de reconnaissance d'images disponibles à l'époque.

1.1 Contexte du projet et travail demandé

Le projet réalisé dans la matière 'Intelligence artificielle' consiste à créer un classificateur basé sur un réseau de neurones convolutif(CNN). Le but est de développer un modèle capable de reconnaître ces différentes catégories d'objets en utilisant des techniques d'apprentissage automatique.

1.2 Les réseaux de neurones artificiels

1.2.1 Définition

Un réseau de neurones artificiels est un système conçu à l'origine en s'inspirant du fonctionnement des neurones biologiques, mais qui s'est ensuite rapproché des méthodes statistiques. Ces réseaux sont composés de processeurs élémentaires fortement connectés, qui fonctionnent en parallèle. Chaque processeur élémentaire, ou neurone artificiel, effectue des calculs pour générer une sortie spécifique en se basant sur les informations qu'il reçoit [1].

1.2.2 Types des reseaux de neurones

Il existe différents types de réseaux de neurones, et ils sont classés en fonction du nombre de nœuds cachés du modèle ou encore du nombre d'entrées et de sorties de chaque nœud. La propagation des informations entre les différents neurones peut varier et dépend du type de réseaux de neurones.

Les réseaux de neurones à propagation avant (feed-forward) : C'est la variante la plus simple, où l'information se déplace uniquement dans une direction. Elle traverse directement les nœuds de traitement (couches cachées) à partir de l'entrée, puis se dirige vers les sorties. Dans ce type de réseau, il n'y a pas de cycles ou de boucles.

Les réseaux de neurones récurrents : Ce mode d'apprentissage est un peu plus complexe. Ces réseaux comportent au moins un cycle dans leur structure, ce qui leur permet de sauvegarder les résultats produits par les nœuds de traitement et de les utiliser pour nourrir le modèle. Les réseaux de neurones récurrents sont utilisés dans des domaines tels que la reconnaissance automatique de formes et la traduction automatique de la parole.

Les réseaux de neurones convolutifs : Inspirés par le fonctionnement du cortex visuel des animaux, ces réseaux traitent les données en les faisant passer à travers plusieurs couches,

de la couche d'entrée à la couche de sortie. Les réseaux de neurones convolutifs sont utilisés dans des domaines tels que la reconnaissance faciale, la reconnaissance d'images et de vidéos, ainsi que le traitement du langage naturel.

1.3 Convolutional neural net work

1.3.1 Définition

Dans le domaine de l'apprentissage en profondeur, les réseaux de neurones convolutifs (CNN/ConvNet) sont une classe de réseaux de neurones profonds largement utilisés pour l'analyse d'images visuelles. Contrairement à ce que l'on pourrait penser généralement lorsqu'on évoque les réseaux de neurones, les CNN ne se contentent pas d'utiliser des multiplicateurs matriciels. Ils font appel à une technique spéciale appelée "bypass" ou "skip connection".

La convolution, en mathématiques, est une opération arithmétique qui permet de combiner deux fonctions pour en produire une troisième, exprimant ainsi la manière dont la forme de l'une est modifiée par l'autre. Dans le contexte des CNN, la convolution est utilisée pour extraire des caractéristiques des images en appliquant des filtres spécifiques sur des régions locales. Cela permet aux CNN de capturer des motifs et des informations significatives au sein des images[3].

1.3.2 Architecture d'un réseau de neurones convolutif

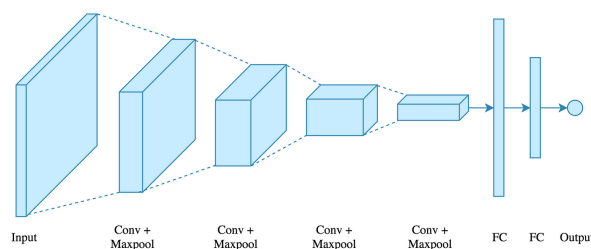


FIGURE 1 – Architecture du modèle CNN

Un réseau de neurones convolutif n'est pas seulement un réseau neuronal profond avec de nombreuses couches cachées. Il s'agit plutôt d'un réseau profond qui simule le fonctionnement du cortex visuel du cerveau pour reconnaître et classifier des images ou des vidéos, et pour découvrir un objet ou même une partie dans une image [2]. Le concept et le fonctionnement des réseaux de neurones convolutifs est différent des autres réseaux de neurones, en

effet un réseau neuronal convolutif comporte deux parties distinctes avec une entrée dans laquelle une image en forme de matrice de pixels bidimensionnelle (avec 2 dimensions, noir et blanc), ou une image couleur avec 3 dimensions (couleurs : rouge, vert et bleu) ou une image multidimensionnelle (image satellitaire) .

La première partie d'un réseau de neurones convolutif est la partie convolutionnelle qui sert à extraire les caractéristiques de l'image. Ensuite, l'image passe par le fichier de séquence de filtre, ou le noyau d'enroulement, ce qui conduit à la création d'une nouvelle image appelée cartes de convolution. Généralement, les filtres intermédiaires réduisent la résolution de l'image.

Ensuite, les cartes des caractéristiques sont aplaties dans un vecteur de caractéristiques pour former les données d'entrée de la partie de couche entièrement connectée. Le rôle principal de cette couche (complètement connectée) est de combiner les caractéristiques contenues dans le vecteur de son entrée pour la classification des images [2].

La couche convolutive (CONV)

traite les données en utilisant des filtres ou des champs récepteurs pour détecter des motifs et des caractéristiques spécifiques dans les données d'entrée. Elle applique des opérations de convolution pour extraire les informations pertinentes[3].

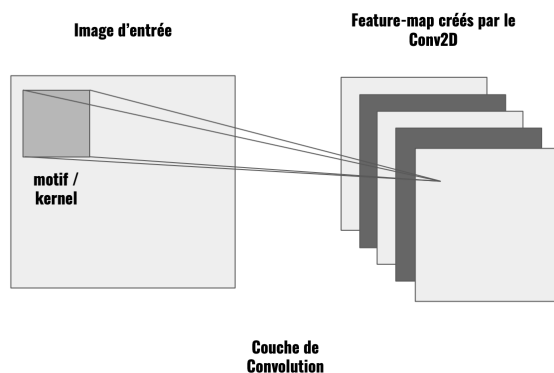


FIGURE 2 – Architecture du modèle CNN

La couche de regroupement (POOL)

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling.

L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes. Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale[4].

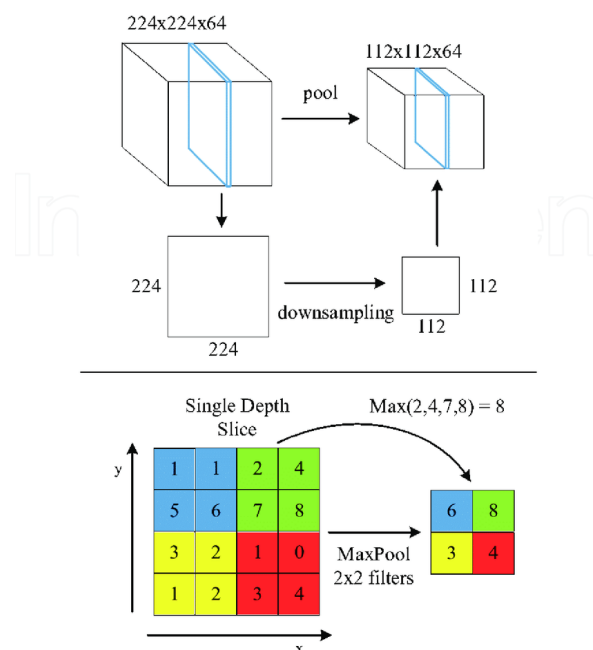


FIGURE 3 – Principe de Max Pooling

La couche de correction (ReLU)

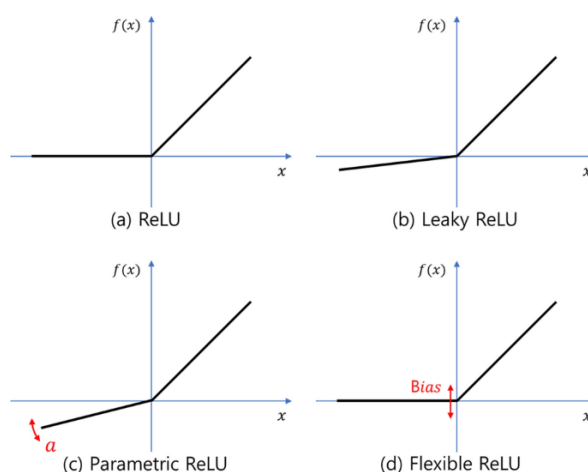


FIGURE 4 – ReLu – Unité linéaire rectifiée

ReLU est une fonction qui doit être appliquée à chaque pixel d'une image après convolution, et remplace chaque valeur négative par un 0. Si cette fonction n'est pas appliquée, la fonction créée sera linéaire et le problème XOR persiste puisque dans la couche de convolution, aucune fonction d'activation n'est appliquée.

La couche entièrement connectée (FC)

est une couche de type perceptron, où chaque neurone est connecté à tous les neurones de la couche précédente. Elle agit en combinant les caractéristiques apprises précédemment pour effectuer des prédictions ou des classifications sur les données d'entrée.

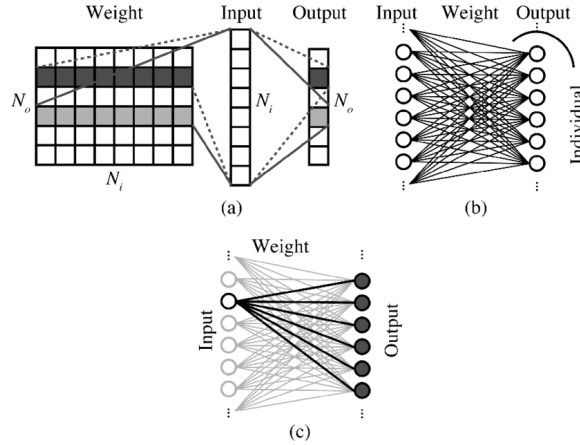


FIGURE 5 – Couche fully connected

La couche de perte (LOSS)

utilisée pour évaluer l'écart entre les prédictions du modèle et les valeurs réelles. Elle mesure l'erreur du modèle et est utilisée pour ajuster les poids et les paramètres du réseau lors de l'apprentissage, afin d'optimiser les performances du modèle[5].

1.4 Conclusion

Les réseaux de neurones, en particulier les CNN, sont des outils puissants pour la classification et la reconnaissance de motifs dans les données. Leur architecture profonde leur permet d'apprendre des représentations complexes des données, ce qui en fait des outils précieux dans de nombreux domaines.

CHAPITRE II

Classification

2.1 Introduction

La classification est le processus de classification des éléments en groupes spécifiques, et dans le contexte de l'apprentissage automatique, cette classification est effectuée par un ordinateur. Pensez à quel point ce serait génial si votre ordinateur pouvait faire la différence entre vous et un étranger, ou distinguer une pomme de terre d'une tomate, ou décider si une certaine performance méritait un A ou un F. Soudain, le concept devient beaucoup plus intéressant. Dans l'apprentissage automatique supervisé, la classification est une tâche essentielle qui implique l'utilisation de données étiquetées pour l'apprentissage. Dans les domaines de l'apprentissage automatique et des statistiques, la classification fait référence au problème consistant à attribuer une nouvelle observation à l'une de plusieurs classes ou sous-populations prédéfinies. Cette détermination est basée sur un ensemble de données de d'entraînement, où les observations sont connues pour appartenir à des catégories spécifiques.

2.2 Définition

La classification est une discipline largement utilisée dans de nombreux domaines. Elle est souvent connue sous différents termes tels que classification, segmentation et regroupement. Pour donner une définition précise de la classification, il est nécessaire de comprendre ses racines, qui dérivent du verbe "classer", englobant plus qu'un seul domaine. En mathématiques, la classification est la catégorisation des objets. Elle consiste à attribuer une classe à chaque objet ou individu à classer, sur la base de données d'apprentissage. Les méthodes d'apprentissage sont couramment utilisées pour accomplir cette tâche.

2.3 Etapes d'une classification

Sélection des données à utiliser.

- Calcul de la similarité entre les n individus en se basant sur les données initiales.
- Choix d'un algorithme de classification et exécution de celui-ci.
- Interprétation des résultats obtenus.
- Évaluation de la qualité de la classification.
- Description des classes obtenues.

2.4 Type de classification

2.4.1 Binaire

la classification binaire consiste à catégoriser les données en deux classes ou catégories distinctes. C'est la forme de classification la plus simple, où le but est d'affecter chaque point de données à l'une des deux classes prédéfinies. Par exemple, déterminer si un e-mail est un spam ou non, classer une transaction comme frauduleuse ou légitime, ou prédire si un patient a ou non une certaine condition médicale.

2.4.2 multi-class

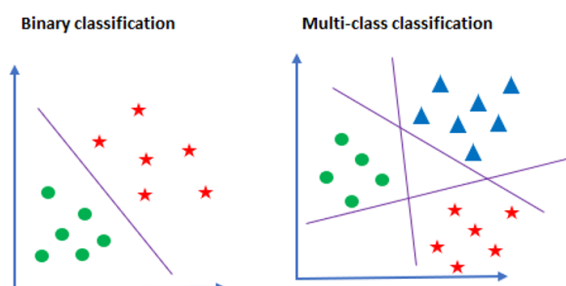


FIGURE 1 – Classification binaire vs classification multi-class

la classification multi classe implique la catégorisation des données en plus de deux classes ou catégories. Dans ce type de classification, l'objectif est d'affecter chaque point de données à l'une de plusieurs classes prédéfinies. Par exemple, classer des images en différents types d'animaux (chat, chien, oiseau, etc.), reconnaître des chiffres manuscrits (0-9) ou identifier le genre d'une chanson (rock, pop, jazz, etc.). Les problèmes de classification multi classe peuvent être résolus à l'aide de divers algorithmes tels que les arbres de décision, la régression logistique, les machines à vecteurs de support ou les modèles d'apprentissage en profondeur comme les réseaux de neurones profond. Ces deux types de classification fournissent une base pour organiser et analyser les données en fonction de leurs catégories ou classes distinctes

2.5 Principe de fonctionnement de la classification

L'objectif principal est de construire un modèle en apprenant les relations entre les données d'entrée et les étiquettes de sortie correspondantes. Cette phase de formation permet au modèle de comprendre les caractéristiques de chaque classe et de développer un processus de prise de décision basé sur les exemples étiquetés fournis. Une fois le modèle formé, il peut être utilisé pour faire des prédictions pour de nouvelles instances, même si leurs étiquettes n'étaient pas initialement connues. Le but ultime de la classification est de généraliser les connaissances acquises à partir des données d'entraînement et de faire des prédictions précises pour les données non étiquetées, élargissant ainsi la capacité du modèle au-delà de l'ensemble de données étiqueté fourni.

2.6 Conclusion

Dans ce chapitre, nous avons discuté des concepts de classification, en explorant les divers types de classifications. Le dernier chapitre se concentrera sur l'application d'un classificateur profond (CNN) à un cas concret, celui de la base de données Fashion MNIST, dans le but de reconnaître et de classer les images.

CHAPITRE III

Implémentation

3.1 Environnement de réalisation

3.1.1 Jupyter-Notebook



FIGURE 1 – Jupyter-notebook

Jupyter notebook est une application web qui permet de stocker et exécuter du code écrit en Python, ainsi que de visualiser les résultats de l'exécution de ce code sous forme de graphiques, tableaux, etc. Il permet également de créer et de partager des documents interactifs qui combinent du code, des visualisations, des textes de documentation et des équations. C'est un outil populaire pour les scientifiques, les ingénieurs, les data scientists et les développeurs qui travaillent avec Python, car il permet une collaboration facile, une documentation interactive et une visualisation de données.



FIGURE 2 – Python

3.1.2 Le langage Python

Python est un langage de programmation populaire pour sa flexibilité et sa facilité d'utilisation. Il est utilisé dans nombreux domaines, notamment le développement web, le développement d'applications financières et l'automatisation de tâches. Il est également largement utilisé dans le domaine de l'apprentissage automatique, de l'apprentissage en profondeur et de l'intelligence artificielle en raison de sa facilité d'utilisation et de sa grande communauté de développeurs. Il est facilement intégrable à d'autres bibliothèques et outils pour le Machine Learning et le Deep Learning. Ce qui en fait un choix populaire pour les projets de ces domaines.

3.1.3 Bibliothèques Python



FIGURE 3 – Bibliothèques Python

NumPy est une bibliothèque Python pour la manipulation de tableaux multidimensionnels et

des fonctions mathématiques opérant sur ces tableaux.

Pandas est une bibliothèque open-source pour Python qui offre des structures de données et des outils d'analyse de données performants et faciles à utiliser. Elle est largement utilisée dans le domaine de la science des données et de l'analyse statistique pour manipuler et analyser des données tabulaires.

Matplotlib est une bibliothèque Python pour créer des graphiques et des visualisations de données.

Keras est une API de réseau de neurones écrite en Python qui facilite la création de modèles de Deep Learning. Il est exécuté sur des Framework tels que TensorFlow ou Theano.

TensorFlow est une bibliothèque open source utilisée pour créer des modèles de Deep Learning et pour effectuer des tâches d'apprentissage automatique complexes. Il fournit une boîte à outils pour résoudre des problèmes mathématiques complexes avec aisance et simplicité. Il peut être utilisé avec keras pour faciliter l'implémentation des modèles de Deep Learning.

3.2 Réalisation et résultats

3.2.1 Introduction

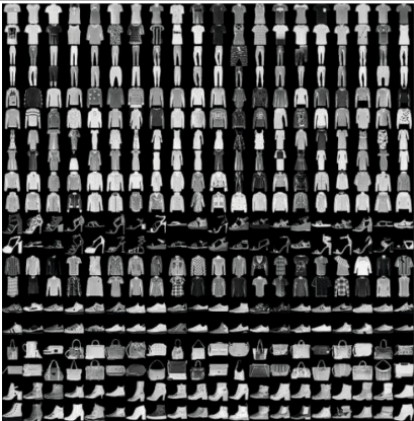
Label	Description	Examples
0	T-shirt/top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

FIGURE 4 – Fashion MNIST Dataset

L'ensemble d'entraînement Fashion se compose de 70 000 images divisées en 60 000 échantillons d'entraînement de 10 000 échantillons de test. Chaque échantillon de données est une image en niveaux de gris de 28x28 pixels, associée à une étiquette appartenant à l'une des 10 classes suivantes :

Les 10 classes sont les suivantes : 0 - T-shirt/haut 1 - Pantalon 2 - Pull 3 - Robe 4 - Manteau 5 - Sandale 6 - Chemise 7 - Basket 8 - Sac 9 - Bottine

Chaque image mesure 28 pixels de hauteur et 28 pixels de largeur, pour un total de 784 pixels au total. Chaque pixel a une seule valeur de pixel associée, indiquant la luminosité ou l'obscurité de ce pixel, les nombres plus élevés signifiant plus foncé. Cette valeur de pixel est un entier compris entre 0 et 255.

3.2.2 Les étapes suivies

Etape 1 : Import des données Cette étape consiste à importer les bibliothèques nécessaires à notre projet. Cela comprend généralement des bibliothèques de traitement de données telles que Pandas pour la manipulation des données, NumPy pour le calcul numérique, et éventuellement des bibliothèques d'apprentissage automatique comme Scikit-learn ou TensorFlow. Importer ces bibliothèques est essentiel car elles fournissent les outils et les fonctions nécessaires pour charger, nettoyer et préparer vos données, ainsi que pour construire et entraîner des modèles d'apprentissage automatique.

Etape 2 : Import des données La deuxième étape est le téléchargement des données pertinentes pour notre projet. Cela peut impliquer le téléchargement de fichiers de données à partir de sources en ligne, des sites Web, ou l'importation de données à partir de fichiers locaux sur le système.

Etape 3 : Visualisation des données

Dans l'étape de visualisation des données, nous explorons la structure et les caractéristiques de nos ensembles de données afin de mieux comprendre la nature des informations qu'ils contiennent.

Cette étape est essentielle pour identifier les tendances, les schémas et les relations entre les variables, ce qui nous permet de prendre des décisions éclairées lors de l'analyse et de la modélisation. Nous avons également inclus une visualisation qui présente un ensemble d'images avec leurs étiquettes associées, une représentation graphique de la distribution des classes dans nos ensembles de données d'entraînement et de test.

Etape 4 : Construction du modèle La construction du réseau neuronal nécessite la confi-



FIGURE 5 – sous graphiques avec leurs étiquettes

guration des couches du modèle, puis la compilation du modèle.

1- Configuration des couches Le bloc de construction de base d'un réseau neuronal est la couche. Les couches extraient des représentations des données qui leur sont fournies. Espérons que ces représentations sont significatives pour le problème en cours.

La plupart de l'apprentissage profond consiste à enchaîner des couches simples. La plupart des couches, telles que `tf.keras.layers.Dense`, ont des paramètres qui sont appris lors de l'entraînement.

La première couche de ce réseau, **`keras.layers.Flatten`**, transforme le format des images d'un tableau bidimensionnel (de 28 par 28 pixels) en un tableau unidimensionnel (de $28 * 28 = 784$ pixels). On peut considérer cette couche comme déroulant les lignes de pixels dans l'image et les alignant. Cette couche n'a pas de paramètres à apprendre; elle ne fait que reformater les données.

Après avoir aplati les pixels, le réseau se compose d'une séquence de deux couche **`keras.layers.Dense`**. Ce sont des couches neuronales connectées de manière dense, ou entièrement connectées. La première couche Dense a 128 nœuds (ou neurones). La deuxième (et dernière)

```

cnn_model = Sequential()

cnn_model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(28,28,1), activation='relu', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(28,28,1), activation='relu', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D(pool_size=(2, 2)))
cnn_model.add(Dropout(0.2))

cnn_model.add(Conv2D(filters=64, kernel_size=(3, 3), input_shape=(28,28,1), activation='relu', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(Conv2D(filters=64, kernel_size=(3, 3), input_shape=(28,28,1), activation='relu', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D(pool_size=(2, 2)))
cnn_model.add(Dropout(0.2))

cnn_model.add(Flatten())
cnn_model.add(Dense(units=128, activation='relu'))
cnn_model.add(Dropout(0.2))
cnn_model.add(Dense(units=10, activation='softmax'))

```

FIGURE 6 – Construction du modèle

couche est une couche softmax à 10 nœuds qui renvoie un tableau de 10 scores de probabilité qui s'additionnent à 1. Chaque nœud contient un score qui indique la probabilité que l'image actuelle appartienne à l'une des 10 classes.

2- Compilation du modèle : Avant que le modèle ne soit prêt pour l'entraînement, il a besoin de quelques réglages supplémentaires. Ceux-ci sont ajoutés lors de l'étape de compilation du modèle : - Fonction de perte (Loss function) — Cela mesure à quel point le modèle est précis pendant l'entraînement. Vous voulez minimiser cette fonction pour "diriger" le modèle dans la bonne direction. - Optimiseur (Optimizer) — C'est ainsi que le modèle est mis à jour en fonction des données qu'il voit et de sa fonction de perte. Métriques (Metrics) — Utilisées pour surveiller les étapes d'entraînement et de test. L'exemple suivant utilise la précision, la fraction des images correctement classées.

```

METRICS = [
    'accuracy',
    tf.keras.metrics.Precision(name='precision'),
    tf.keras.metrics.Recall(name='recall')
]

cnn_model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

```

FIGURE 7 – Compilation du modèle

3- Entraînement du modèle : Nous divisons nos données d'entraînement en ensembles d'entraînement et de validation, afin d'entraîner notre modèle et de le valider en utilisant l'en-

semble de données de validation pour éviter le surajustement avant de tester le modèle sur les ensembles de données de test, qui représentent des données du monde réel pour notre modèle. Pour commencer l'entraînement, appelons la méthode `model.fit`. Cette méthode ajuste le modèle aux données d'entraînement

4- Evaluation du modèle : -Perte de cross-entropy et la précision : Une perte de cross-entropy plus faible indique que le modèle est meilleur à prédire les étiquettes correctes. D'autre part, la précision de classification mesure la proportion d'images correctement classées par le modèle sur l'ensemble de test. Une précision plus élevée indique que le modèle est plus précis dans ses prédictions. En combinant ces deux mesures, on peut obtenir une évaluation complète de la performance du modèle CNN sur l'ensemble de test, ce qui est crucial pour évaluer son efficacité dans la classification des données.

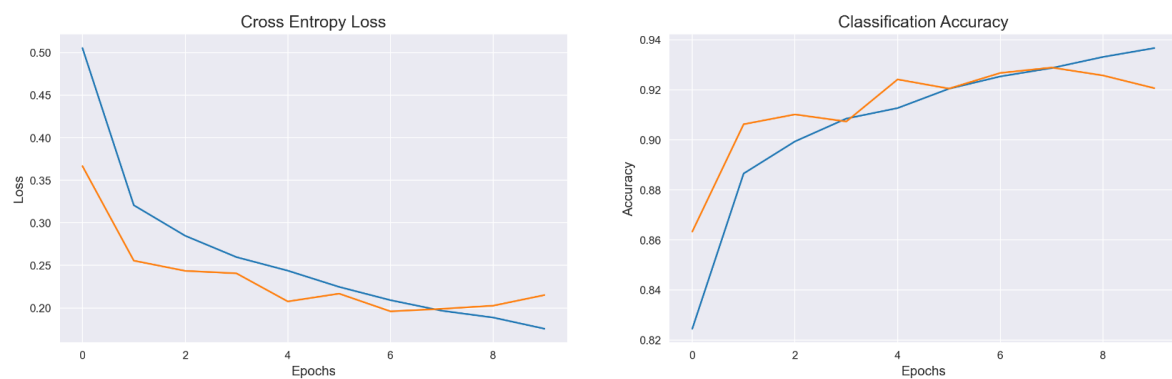


FIGURE 8 – Cross entropy Loss vs Classification accuracy

-Matrice de confusion : La matrice de confusion est un outil essentiel pour évaluer la performance d'un modèle de classification, y compris les modèles CNN. Cette matrice présente un tableau qui compare les prédictions du modèle avec les étiquettes réelles des données d'évaluation. Elle répertorie les prédictions correctes ainsi que les erreurs de classification pour chaque classe.

En utilisant la matrice de confusion, on peut analyser où le modèle se trompe le plus fréquemment et s'il y a des classes spécifiques pour lesquelles il a des difficultés. Cela permet d'identifier les lacunes du modèle et de déterminer les ajustements nécessaires pour améliorer sa performance. En résumé, la matrice de confusion fournit une évaluation détaillée et visuelle de la capacité du modèle à classer correctement les données.

		Confusion Matrix									
True Label	T-shirt top	874	0	14	14	2	2	91	0	3	0
	Trouser	0	978	1	17	1	0	2	0	1	0
	Pullover	14	2	905	11	19	0	49	0	0	0
	Dress	20	3	8	907	14	0	47	0	1	0
	Coat	2	0	94	29	767	0	104	0	4	0
	Sandal	0	0	0	0	0	960	0	6	0	4
	Shirt	105	0	65	14	44	0	767	0	5	0
	Sneaker	0	0	0	0	0	35	0	951	1	13
	Bag	1	0	0	4	0	6	5	2	982	0
	Ankle boot	0	0	0	0	0	15	0	37	0	948
		Predicted Label									

FIGURE 9 – Matrice de confusion

- **Prédiction** : Après avoir entraîné un modèle sur un ensemble de données, l'étape de prédiction consiste à utiliser ce modèle pour faire des prédictions sur de nouvelles données non vues auparavant.

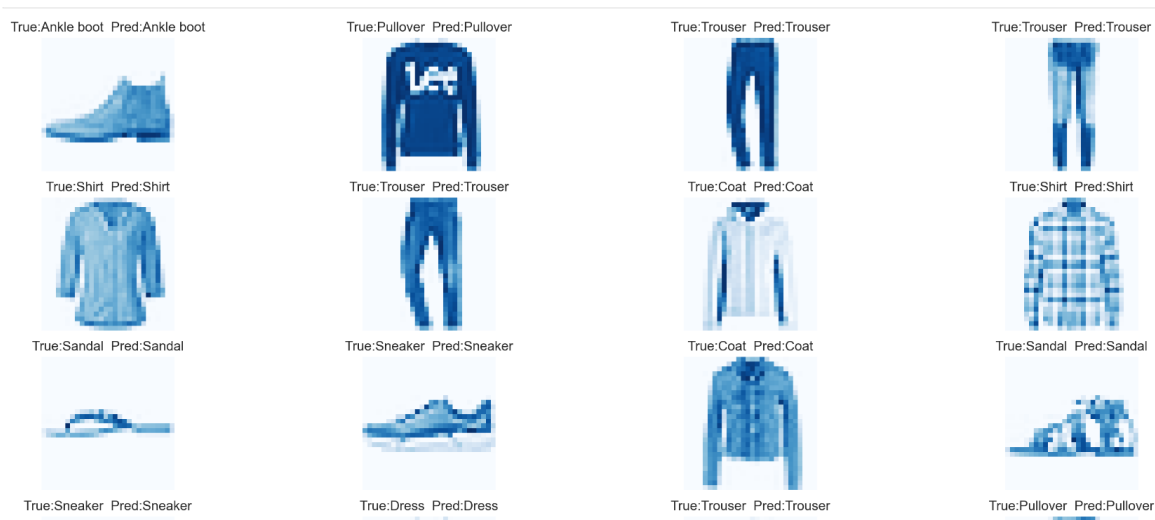


FIGURE 10 – Prédiction

ANNEXE

Dataset

Le jeu de données a été récupéré à partir de Kaggle.



FIGURE 11 – Interface web-1

[1]-[5]

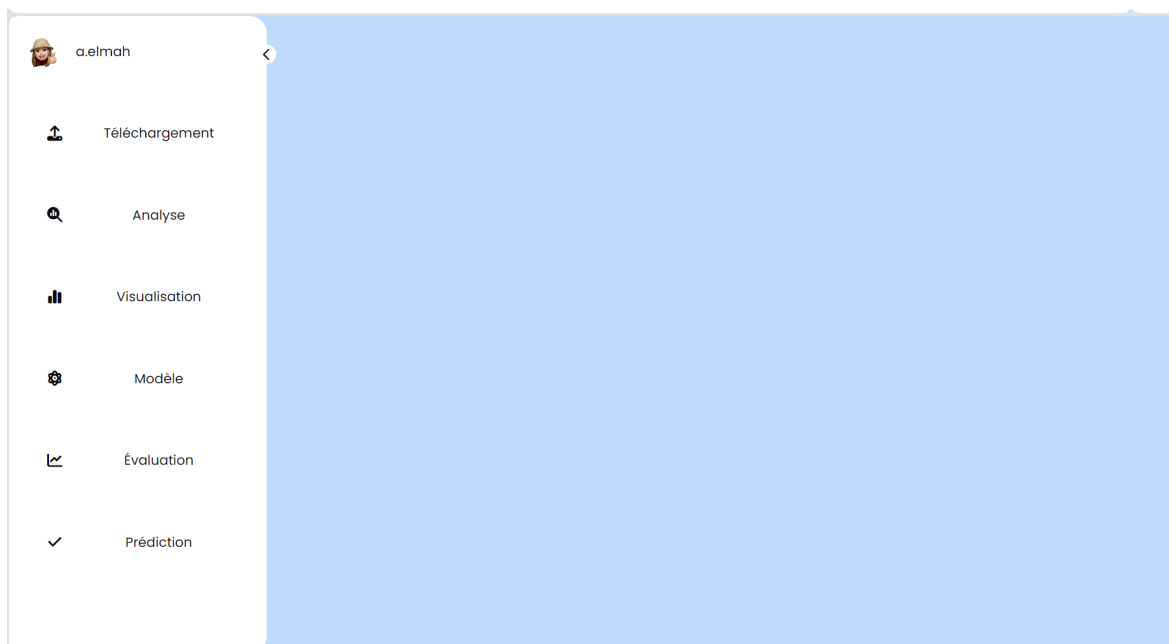


FIGURE 12 – Interface web-2

Informations sur le Dataset

- Ensemble des données

L'ensemble d'entraînement Fashion se compose de 70 000 images divisées en 60 000 échantillons d'entraînement et 10 000 échantillons de test. Chaque échantillon de données est une image en niveaux de gris de 28x28 pixels, associée à une étiquette appartenant à l'une des 10 classes.

- Les classes:

Les 10 classes sont les suivantes :

- 0 => T-shirt/haut
- 1 => Pantalon
- 2 => Pull
- 3 => Robe
- 4 => Manteau
- 5 => Sandale
- 6 => Chemise
- 7 => Basket
- 8 => Sac
- 9 => Bottine

- Les images

- Chaque image mesure 28 pixels de hauteur et 28 pixels de largeur, pour un total de 784 pixels au total. Chaque pixel a une seule valeur de pixel associée, indiquant la luminosité ou l'obscurité de ce pixel, les nombres plus élevés signifiant plus foncé. Cette valeur de pixel est un entier compris entre 0 et 255.
- Nombre total d'enregistrements : 70 000
- Nombre total de colonnes : 785

FIGURE 13 – Informations sur les données



FIGURE 14 – Visualisation des données

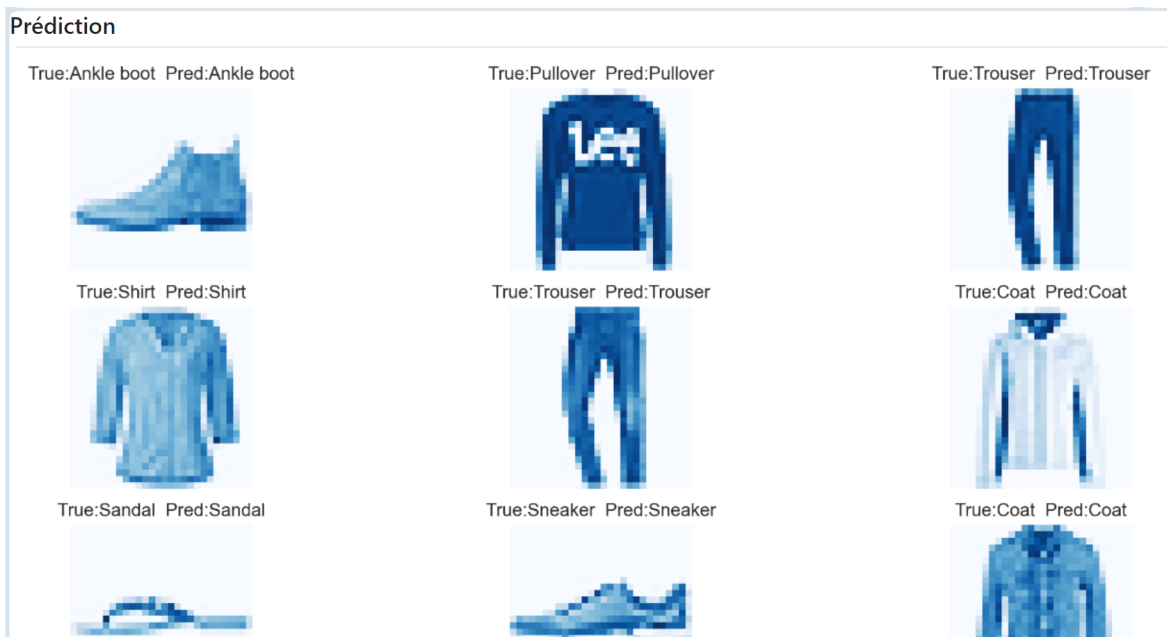


FIGURE 15 – Prédiction

Bibliographie

- [1] M. OUNISSI et Z. I. HARNANE, « Modélisation et classification avec Deep Learning : Application à la détection du Coronavirus Covid-19, » Thèse de doctorat, Thesis, Université (ou institution), 2020.
- [2] OPENCLASSROOMS. « Classez et segmentez des données visuelles. » Consulté le date. (année), adresse : <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn>.
- [3] L. GASMI, « Deep Learning for Face Recognition, » Ph.D. dissertation, Thesis, Université (ou institution), (adresse de l'université ou de l'institution), 2020.
- [4] M. MANDAL. « Convolutional Neural Networks (CNN). » Published On May 1, 2021 and Last Modified On April 28th, 2023. (2021), adresse : <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networkscnn/>.
- [5] S. MADELEINE. « Classification des Images Médicales : Comprendre le Réseau de Neurones Convolutifs (CNN). » Consulté le date. (février 2021), adresse : <https://www.imaios.com/fr/ressources/blog/classification-des-images-medicales-comprendre-le-reseau-de-neurones-convolutifs-cnn>.