M2IIA : PROJET TUTEURÉ

Université de Bourgogne, UFR Sciences et Techniques, Département I.E.M.

Rapport de Mémoire

Auteur/Autrice
Benjamin DESCHAMPS-FRILLEY

Titre du projet

Romain RAFFIN
Marco LA CASCIA
Liliana LO PRESTI

2023-24

Table des matières

1.1 Introduction 5
1.1.1 Contexte 5
1.1.2 Solution existante 6
1.1.3 Technologie utilisé 8
1.2 Application 9
1.2.1 première esquisse 10
1.2.2 Labelisation 11
1.2.3 arbre N-aire 12
1.2.4 Selection 13
1.3 Probème Rencontré 16
1.4 Piste d'amélioration 18
1.5 conclusion 20

Résume Dans le cadre de mon double diplôme, alliant un Master Image et Intelligence Artificielle et une formation d'ingénieur à ESIREM, j'ai eu l'opportunité de réaliser mon stage de fin d'études à l'Université de Palerme (UNIPA) en Italie. Ce stage s'inscrit dans une démarche de mobilité internationale, obligatoire pour l'obtention de mon diplôme d'ingénieur, ce qui m'a permis d'enrichir mon expérience professionnelle et personnelle dans un contexte académique et culturel différent.

Durant ce stage, j'ai été intégré le laboratoire de l'université UNIPA dont la spécialité est le traitement d'image et la reconnaissance de texte. La mission principale qui m'a été confiée consistait à développer un système de reconnaissance d'objets inconnus. Ce développement faire partie d'un projet d'un robot autonome capable d'interagir avec son environnement de manière intelligente et adaptative. Ce défi technologique m'a permis de mettre en pratique mes compétences en intelligence artificielle tout en contribuant à un projet de pointe.

Ce rapport est structuré autour du développement de ce système de reconnaissance d'objets. Il présente d'abord le contexte théorique et technique du projet, puis décrit en détail les étapes de conception, de mise en œuvre et de validation du système. Enfin, une analyse critique de mon travail ainsi qu'une réflexion sur les perspectives d'amélioration et d'intégration dans le projet global seront proposées. On donne un résumé du travail effectué, la problématique, l'axe proposé pour la solution qui sera mise en œuvre.

Mots-clés

1.1 Introduction

1.1.1 Contexte

Le développement de robots autonomes est un domaine en pleine expansion, avec des applications qui vont de la logistique à la sécurité en passant par la médecine. Dans ce cadre, la capacité d'un robot à identifier et à reconnaître des objets dans son environnement est cruciale pour assurer son efficacité et sa sécurité. C'est dans ce contexte que s'inscrit notre projet, visant à concevoir un système de reconnaissance d'objets intégré à un robot autonome.

Objectifs du projet Le projet avait pour but principal de permettre à un robot de reconnaître et de classifier les objets présents dans son environnement. Mon rôle spécifique consistait à développer un module capable d'identifier des objets précédemment inconnus et d'enrichir une base de données d'images pour améliorer les performances du robot au fil du temps. Ce processus d'apprentissage continu devait permettre au robot de devenir de plus en plus précis dans ses reconnaissances d'objets.

Approche collaborative Le projet a été mené en étroite collaboration avec un autre membre de l'équipe, chargé de la détection d'objets. Tandis que mon coéquipier se concentrait sur l'identification initiale des objets, j'étais responsable de l'analyse des objets détectés et de la création d'une base de données d'images. Cette approche collaborative nous a permis de combiner nos compétences pour développer un système robuste et efficace.

1.1.2 Solution existante

Pour entraîner des modèles d'apprentissage automatique, il est courant d'utiliser des jeux de données d'images bien établis qui offrent des ensembles d'images annotées dans diverses classes. Voici quelques-uns des jeux de données les plus utilisés :

ImageNet : Ce jeu de données est l'un des plus complets et largement utilisés dans le domaine de la vision par ordinateur. Il contient plus de 1 000 classes, couvrant une vaste gamme d'objets du quotidien. ImageNet a été crucial dans le développement et l'évaluation de nombreux modèles de classification d'images. Il est particulièrement apprécié pour sa richesse en termes de diversité des images et de profondeur des classes.

CIFAR-100 : Ce jeu de données contient 100 classes d'objets, souvent utilisé pour des tâches de classification d'images à faible résolution. CIFAR-100 est apprécié pour sa simplicité et sa capacité à permettre l'expérimentation rapide sur des modèles d'apprentissage profond.

Fashion MNIST : Ce jeu de données est centré sur les vêtements et accessoires, avec 10 classes distinctes. Fashion MNIST est souvent utilisé comme un substitut plus moderne et plus difficile à l'ancienne base de données MNIST de chiffres manuscrits. Il permet d'explorer les performances des modèles dans un domaine spécifique de l'imagerie.

COCO (Common Objects in Context) : COCO est un jeu de données destiné à des tâches complexes telles que la détection d'objets, la segmentation d'images et la légende d'images. Il comprend 80 classes d'objets et est largement utilisé pour des applications nécessitant une compréhension fine des objets dans des contextes variés.

Pascal VOC : Ce jeu de données est couramment utilisé pour la détection d'objets et la segmentation d'images. Avec 20 classes d'objets, Pascal VOC fournit des annotations détaillées pour les tâches de reconnaissance visuelle, bien que le nombre de classes soit relativement limité par rapport à d'autres jeux de données.

Limitations des Jeux de Données Existants Malgré leur utilité, ces jeux de données présentent plusieurs limitations qui peuvent poser des défis pour certains projets d'apprentissage automatique :

Spécificité de Domaine : Certains jeux de données, comme Fashion MNIST, sont extrêmement spécifiques à un domaine particulier. Fashion MNIST se concentre uniquement sur les vêtements et accessoires, ce qui limite sa pertinence pour les applications qui nécessitent une reconnaissance d'objets dans des contextes variés. Cette spécialisation rend le jeu de données moins utile pour des tâches de classification d'objets plus générales.

Variété d'Objets Limité : Des jeux de données comme Pascal VOC, avec ses 20 classes, peuvent offrir une couverture limitée des objets. Bien que cela puisse suffire pour des applications spécifiques, la restriction en termes de variété d'objets reconnus peut limiter la capacité du modèle à généraliser à de nouveaux contextes ou à des objets non inclus dans ces classes.

Couverture Inadéquate pour des Applications Spécifiques : Même les jeux de données plus généralistes comme ImageNet, bien qu'ils couvrent une large gamme d'objets,

peuvent ne pas inclure certaines catégories d'objets spécifiques nécessaires pour des applications particulières. La richesse de la base de données en termes de diversité des objets ne garantit pas que chaque catégorie d'objet d'intérêt pour des projets spécifiques soit bien représentée.

Qualité et Variabilité des Données : Certains jeux de données peuvent souffrir de problèmes de qualité et de variabilité des images. Par exemple, les images peuvent varier considérablement en termes de résolution, d'éclairage et de contexte, ce qui peut compliquer la tâche d'entraînement de modèles robustes. Les images mal annotées ou les annotations incohérentes peuvent également affecter la qualité de l'apprentissage.

Ces limitations soulignent l'importance de créer de nouveaux ensembles de données plus diversifiés et représentatifs, adaptés aux besoins spécifiques de certains projets d'apprentissage automatique. En développant des jeux de données qui répondent aux exigences particulières des applications visées, nous pouvons améliorer la précision et la robustesse des modèles d'intelligence artificielle, tout en surmontant les contraintes imposées par les ensembles de données existants. La création de nouvelles bases de données permet d'affiner les capacités des modèles en fonction des objectifs spécifiques du projet, tout en offrant des solutions adaptées aux défis particuliers rencontrés dans des contextes variés.

1.1.3 Technologie utilisé

D'accord ! Voici une présentation simplifiée en se concentrant sur Google Vision API et Google Search API :

Google Vision API est un service de Google qui permet d'analyser des images et de détecter des objets, du texte, des visages, et même des émotions. Par exemple, en envoyant une image à cette API, vous pouvez obtenir une liste des objets reconnus dans l'image, comme des chats, des voitures, ou du texte imprimé. C'est très utile pour des applications comme la reconnaissance d'images, l'analyse de contenu visuel, ou l'extraction d'informations à partir d'images.

Google Search API permet d'intégrer les capacités de recherche de Google dans une application. Par exemple, vous pouvez effectuer des recherches sur le web directement depuis votre application, récupérer les résultats, et les afficher à l'utilisateur. Cette API est pratique pour créer des applications qui nécessitent l'accès aux vastes informations disponibles sur le web, comme un moteur de recherche personnalisé ou un système de recommandation.

Spacy est une bibliothèque Python open-source spécialisée dans le traitement automatique du langage naturel (NLP). Elle est conçue pour être rapide et efficace dans des tâches comme l'analyse syntaxique, la reconnaissance d'entités nommées (NER), la lemmatisation, et bien plus encore. SpaCy est largement utilisé dans des applications où l'analyse et la compréhension du texte sont essentielles, telles que les chatbots, l'extraction d'informations et l'analyse de sentiments.

Sklearn est une bibliothèque Python très populaire pour le machine learning. Elle fournit un ensemble complet d'outils pour l'apprentissage supervisé et non supervisé, comprenant des algorithmes de classification, régression, clustering, réduction de dimension, et plus encore. Scikit-learn est reconnu pour sa facilité d'utilisation, sa documentation exhaustive et son intégration fluide avec d'autres bibliothèques Python comme NumPy et SciPy.

MatplotLib est une bibliothèque Python utilisée pour la visualisation de données. Elle permet de créer une large gamme de graphiques, comme des courbes, des histogrammes, des diagrammes en barres, et plus encore. Matplotlib est très flexible et est souvent utilisé en conjonction avec d'autres bibliothèques comme Pandas pour visualiser des données de manière intuitive. Elle est particulièrement appréciée pour la création de visualisations scientifiques et statistiques.

1.2 Application

Je devais donc utiliser un autre système de classification pour mon image.

Nous avons donc choisi google qui dispose de la base d'image la plus grande ainsi qu'un très grand nombre de classes (le nombre est un connus mais c'est plusieurs milliers de classes).

1.2.1 première esquisse

Lorsque nous utilisons Google pour obtenir les classes associées à une image, nous recevons souvent une multitude d'informations. Il est important de comprendre qu'une seule image peut contenir plusieurs objets, chacun pouvant être décrit par différents termes ou synonymes. Pour gérer cette diversité, nous pouvons choisir le nombre de mots que nous souhaitons faire apparaître dans les résultats.

Lorsque nous effectuons une recherche, les résultats sont présentés sous forme d'une liste de labels ou de classes attribués à l'image en question. Cependant, plusieurs problèmes peuvent se poser :

Inexactitude des labels : Certains labels peuvent ne pas faire référence à l'objet central de l'image. Par exemple, si l'objet principal est une voiture, un label comme "automotive parking light" (feu de stationnement automobile) pourrait apparaître, bien qu'il ne soit pas directement lié à la voiture elle-même.

Synonymes multiples : Un même objet peut être représenté par plusieurs termes différents. Dans le cas d'une voiture, nous pourrions voir des labels comme "Car" et "Vehicle", qui désignent tous deux le même objet mais sous des appellations différentes. Cela peut compliquer l'analyse, surtout si nous cherchons à regrouper ou à filtrer les résultats de manière cohérente.

Labels non pertinents : Le premier label de la liste peut parfois être totalement hors sujet par rapport à l'objet central de l'image. Cela peut être dû à des erreurs dans l'algorithme de reconnaissance d'image ou à des éléments secondaires dans l'image qui sont mal interprétés comme étant plus importants.

```
Automotive parking light
Automotive side-view mirror
Car
Vehicle
Plant
Hood
Tire
Tire
Automotive side marker light
Motor vehicle
Automotive lighting
Wheel
Automotive tire
Automotive mirror
Bumper
Alloy wheel
```

FIGURE 1.1 – affichage d'une image avec ces différent label

Pour résoudre ces problèmes, il est nécessaire de filtrer les labels afin de s'assurer qu'ils correspondent bien à l'objet central de l'image. Cela peut impliquer de supprimer les labels qui ne sont pas pertinents, de regrouper les synonymes sous une même classe, et de réorganiser la liste des labels pour que les plus pertinents soient mis en avant. En effectuant ces ajustements, nous pouvons obtenir une représentation plus précise et utile de l'objet principal de l'image.

1.2.2 Labelisation

Lors de la labélisation d'une image par Google, le système génère une liste de labels triés de manière décroissante en fonction de plusieurs caractéristiques spécifiques. Ces labels fournissent des informations précieuses sur les objets identifiés dans l'image, ainsi que sur la confiance du modèle quant à ces identifications. Voici une explication détaillée des informations fournies pour chaque label :

```
label_annotations {
  mid: "/m/0k4j"
  description: "Car"
  score: 0.951050878
  topicality: 0.951050878
}
```

FIGURE 1.2 – Exemple d'annotations de labels par Google

Chaque label contient les quatre informations suivantes :

— **mid : "/m/01yrx" :** Le "MID" (Machine ID) est un identifiant unique qui correspond à une entité dans la base de connaissances de Google, connue sous le nom de Google Knowledge Graph. Ici, "/m/01yrx" correspond à l'entité "Cat" (Chat en français).

— **description : "Cat" :** C'est la description textuelle de l'entité identifiée par le modèle. Dans ce cas, l'objet reconnu est un "Cat" (un chat).

— **score : 0.941844702 :** Ce score représente la confiance du modèle dans l'identification de l'objet. Il est exprimé sous forme de probabilité (entre 0 et 1). Un score de 0,9418 indique que le modèle est très confiant que l'objet est bien un chat.

— **topicality : 0.941844702 :** La topicalité mesure la pertinence du label par rapport à l'image entière. Un score élevé (ici, identique au score de confiance) indique que l'objet (chat) est non seulement présent mais aussi pertinent dans le contexte de l'image.

1.2.3 arbre N-aire

Arbre N-aire est une structure de données hiérarchique où chaque noeud peut avoir un nombre variable d'enfants, permettant ainsi de modéliser des relations complexes avec une grande flexibilité. Ce type d'arbre est particulièrement adapté pour des applications où les branches doivent contenir un nombre non fixe d'éléments, comme dans les systèmes de gestion de fichiers, les représentations de données arborescentes, ou les systèmes de classification hiérarchique. Parmi ses principaux avantages, l'arbre n-aire offre une modularité exceptionnelle, permettant d'ajouter ou de supprimer des sous-arbres sans affecter la structure globale. Sa capacité à gérer de manière intuitive une hiérarchie complexe facilite la navigation et l'accès aux données, tout en restant scalable pour s'adapter à des ensembles de données en expansion. L'optimisation de ce type d'arbre peut également répondre aux besoins spécifiques de performance, en garantissant une structure à la fois robuste et adaptable.

L'arbre que je voulais concevoir était a 3 ramification. Notre image initiale serait la racine de l'arbre ensuite nous aurions les labels représentant une branche suivi de photo internet télécharger a partie de leur labels mère. et enfin en position de feuilles nous avons les labels de nos nouvelles Images. Voici un plan de l'arbre.

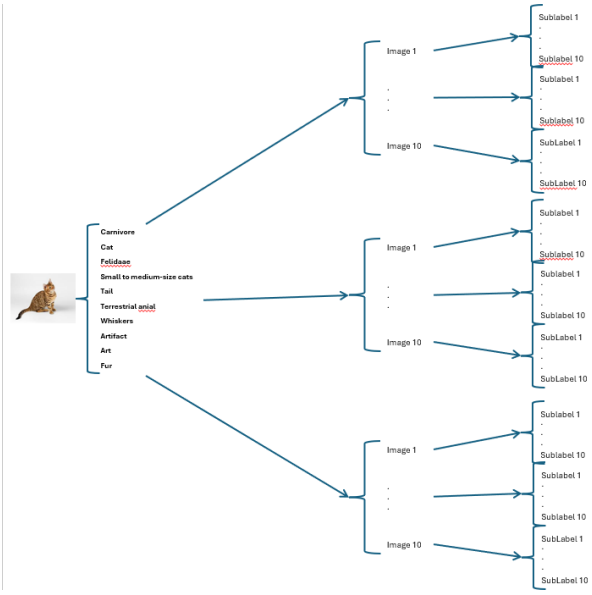

FIGURE 1.3 – Caption

1.2.4 Selection

La sélection des images dans notre arbre est une étape cruciale pour garantir des résultats de qualité. Étant donné que nous téléchargeons des images provenant d'internet et que nous récupérons souvent les premières images qui apparaissent lors de la recherche d'un label spécifique, il est possible de se retrouver avec une grande variété de types d'images. Ces images peuvent inclure des couvertures de livres, des affiches publicitaires, des dessins ou des images représentant des objets très différents de ceux que nous recherchons.

L'objectif principal de la sélection est de s'assurer que chaque image téléchargée correspond bien à l'objet ou au concept représenté par le mot-clé ou le label. De plus, comme nous avons déjà téléchargé les images, il est nécessaire d'effectuer un tri pour déterminer quelles images doivent être conservées et lesquelles doivent être supprimées. Ce processus de sélection se déroule en trois étapes distinctes :

Sélection Basée sur la Liste des Labels de l'Image Initiale

La première étape de la sélection consiste à analyser la liste des labels associés à l'image initiale. Comme mentionné précédemment, Google nous fournit une liste de mots, chacun accompagné d'un score qui représente la confiance du modèle dans l'identification de l'objet. Cependant, il arrive fréquemment que les premiers mots de cette liste ne correspondent pas toujours à l'objet central de l'image.

Étant donné que nous construisons un arbre à partir de cette liste de labels, il est crucial de filtrer les labels qui ne sont pas pertinents pour l'objet central. Cela permettrait de gagner un temps précieux, car la complexité d'un arbre augmente de manière exponentielle avec chaque label supplémentaire. Réduire le nombre de labels non pertinents avant la construction de l'arbre peut considérablement optimiser le processus.

Pour effectuer ce tri, nous allons utiliser *Spacy*, une bibliothèque avancée de traitement du langage naturel (NLP). Spacy offre des capacités robustes pour la comparaison de mots et l'évaluation de leur similarité. Voici comment nous allons procéder :

Utilisation de Spacy pour la Sélection des Labels Spacy dispose de plusieurs modèles préentraînés sur des textes anglais (langues des labels donnés par google). Ces modèles sont capables de comprendre les relations sémantiques entre les mots. Dans notre cas. Cela nous permet de bénéficier d'une compréhension sémantique approfondie des mots sans avoir à entraîner un modèle de zéro.

Spacy dispose de nombreuse fonction de calcul permettant de connaitre le score de similarité basé sur leurs représentations vectorielles (word embeddings). En utilisant ce score de similarité, nous pouvons comparer chaque label fourni par Google avec un ensemble de mots-clés prédéfinis qui représentent au mieux l'objet central que nous cherchons à identifier. - Par exemple, si l'objet central attendu est une "voiture", nous pouvons comparer chaque label avec des mots comme "car", "vehicle", "automobile", etc. Les labels qui présentent une forte similarité avec ces mots-clés seront conservés, tandis que ceux qui sont éloignés en termes de signification seront écartés. Une fois la comparaison effectuée, nous construisons un tableau de similarité qui répertorie les scores de similarité entre chaque label et les mots-clés de référence. Ce tableau nous permet de visualiser rapidement quels labels sont les plus pertinents par rapport à l'objet central et de décider lesquels conserver pour la construction de l'arbre. Les labels

ayant des scores de similarité élevés sont prioritaires pour la génération de l'arbre, tandis que les autres peuvent être éliminés ou relégués au second plan.

Images

La deuxième étape de la sélection se déroule lors du téléchargement des images. Une fois l'image téléchargée, nous procédons à deux analyses distinctes pour évaluer sa pertinence.

Vérification de la Présence du Label Recherché

La première analyse consiste à vérifier si le label que nous avons recherché est bien présent dans les labels associés à l'image téléchargée. Cela garantit que l'image correspond au concept ou à l'objet central que nous souhaitons identifier. Si le label recherché n'est pas détecté dans les labels de l'image, celle-ci est considérée comme non pertinente et peut être immédiatement écartée.

Comparaison des Labels avec l'Image de Base

La deuxième analyse implique une comparaison plus approfondie entre les labels de l'image téléchargée et ceux de l'image de base. Cette comparaison permet de filtrer les images qui ne ressemblent pas suffisamment à l'image de référence.

Calcul du Pourcentage de Correspondance : Pour chaque image téléchargée, nous comparons ses labels avec ceux de l'image de base. Nous évaluons le pourcentage de labels qui correspondent. Si au moins 60 % des labels de l'image téléchargée sont similaires à ceux de l'image de base, l'image est considérée comme pertinente et est conservée.

Élimination des Images Non Pertinentes : Les images dont le pourcentage de correspondance est inférieur à 60 % sont jugées trop dissemblables de l'image de base. Ces images ne sont pas représentatives de l'objet central que nous recherchons et sont donc supprimées de la liste.

Objectifs de cette Étape Cette double vérification, à la fois de la présence du label recherché et de la correspondance avec l'image de base, permet d'affiner la sélection des images. Seules les images qui reflètent réellement l'objet ou le concept recherché sont conservées, ce qui améliore la qualité des données utilisées pour la construction de l'arbre. En éliminant les images non pertinentes à ce stade, nous optimisons le processus global, réduisant ainsi le bruit et les erreurs dans les étapes ultérieures de l'analyse.

Groupe d'image

La troisième et dernière étape de la sélection se focalise sur le groupe d'images téléchargées en fonction d'un label particulier de l'image initiale. Cette étape est essentielle pour affiner davantage notre sélection en regroupant les images similaires et en éliminant celles qui ne sont pas pertinentes. Le processus se déroule en deux phases distinctes :

1. Extraction des Features et Analyse des Similarités

Dans un premier temps, nous allons extraire les caractéristiques (features) des images afin de visualiser les similarités entre elles. Pour ce faire, nous utilisons un modèle de réseau de neurones convolutif (CNN) pré-entraîné, en l'occurrence **ResNet-50**.

- **ResNet-50** : ResNet-50 est un modèle de réseau de neurones convolutif très performant, pré-entraîné sur de vastes bases de données comme ImageNet. Il est capable d'extraire des features complexes et riches des images, ce qui permet de capturer les nuances visuelles essentielles pour la comparaison entre images.

- **Extraction des Features** : Chaque image est passée à travers le modèle ResNet-50 pour extraire un vecteur de caractéristiques. Ces vecteurs représentent les aspects visuels importants de chaque image, tels que les contours, les textures, et les motifs, qui peuvent être utilisés pour évaluer la similarité entre les images.

- **Évaluation des Similarités** : Une fois les vecteurs de caractéristiques obtenus, nous comparons les images entre elles pour évaluer leur degré de similarité. Cette étape permet de déterminer quelles images sont visuellement proches les unes des autres et lesquelles se distinguent par leurs caractéristiques uniques.

2. Clustering Hiérarchique et Gestion Dynamique des Clusters

Une fois les similarités entre les images établies, nous passons à la phase de **clustering hiérarchique**. Cette technique permet de regrouper les images similaires en clusters, ce qui simplifie l'organisation et l'analyse des images.

- **Clustering Hiérarchique** : Cette méthode de clustering organise les images en une hiérarchie de clusters imbriqués. Les images les plus similaires sont d'abord regroupées, puis ces groupes sont à leur tour regroupés en clusters plus larges. Le résultat est une structure arborescente qui montre les relations de similarité à différents niveaux de granularité.

- **Gestion Dynamique des Clusters** : Contrairement à un clustering statique, la gestion des clusters ici est dynamique. Cela signifie que la taille et le nombre de clusters peuvent être ajustés en fonction des besoins spécifiques de l'analyse. Par exemple, si un cluster contient trop d'images disparates, il peut être divisé en sous-clusters pour affiner la sélection. À l'inverse, des clusters très homogènes peuvent être fusionnés pour réduire le nombre total de groupes.

- **Sélection Finale des Images** : Après le clustering, nous évaluons chaque cluster pour déterminer quelles images sont les plus représentatives du label initial. Les clusters qui montrent une forte cohérence interne, c'est-à-dire où les images sont très similaires entre elles, sont conservés. Les clusters contenant des images moins pertinentes ou trop variées sont éliminés.

Objectifs de cette Étape

L'objectif principal de cette étape est de regrouper efficacement les images qui sont visuellement similaires tout en éliminant celles qui ne correspondent pas à l'objet ou au concept central. En utilisant ResNet-50 pour l'extraction des features et le clustering hiérarchique pour l'organisation des images, nous parvenons à affiner encore davantage notre sélection. Cela permet de conserver uniquement les images les plus pertinentes pour l'analyse finale, tout en optimisant le processus de sélection pour être plus efficace et précis.

1.3 Probème Rencontré

Au cours des différentes phases de développement de mon projet, j'ai rencontré plusieurs défis majeurs.

de Connaissances en Intelligence Artificielle et Traitement d'Images Le premier défi a été mon manque de connaissances approfondies en intelligence artificielle et en traitement d'images. En effet, bien que j'aie suivi quelques cours sur ces sujets lors de ma formation à l'ESIREM, ils étaient limités à seulement deux jours par semaine en deuxième année de master. Cette exposition limitée ne m'a pas permis de devenir un expert dans ces domaines complexes.

Face à ces lacunes, j'ai dû investir beaucoup de temps dans l'auto-formation. J'ai passé de nombreuses heures à lire des articles scientifiques et à explorer les recherches existantes liées à mon projet. Il a fallu que je me familiarise avec différents modèles d'intelligence artificielle, que je comprenne leurs mécanismes, leurs avantages et leurs inconvénients, pour pouvoir les appliquer correctement. Cette phase d'apprentissage, combinée aux tests pratiques que j'ai réalisés pour valider mes choix, m'a demandé un investissement considérable en temps et en effort.

Tendance à Vouloir Tout Concevoir de A à Z Le deuxième problème auquel j'ai été confronté a été mon état d'esprit, fortement influencé par ma formation d'ingénieur, qui m'a poussé à vouloir tout concevoir moi-même, de A à Z. Cette approche, bien que formatrice, s'est avérée être un piège dans certains aspects du projet.

Prenons par exemple la partie "Sélection Basée sur la Liste des Labels de l'Image Initiale". Mon objectif était de filtrer les mots qui ne faisaient pas partie du cluster pertinent. Au lieu de m'appuyer sur des outils existants pour effectuer ce tri, j'ai décidé de coder une solution entièrement personnalisée. J'ai commencé par utiliser Spacy pour calculer la similarité entre chaque mot, puis j'ai entrepris des recherches sur les lois de probabilité pour calculer la médiane et l'écart-type. Ensuite, j'ai ajouté des paramètres pour ne conserver que les mots ayant le plus de similarité. Bien que cette approche ait été techniquement correcte, elle m'a demandé un temps considérable pour arriver à un résultat que j'aurais pu obtenir plus rapidement en utilisant des méthodes de clustering établies.

Un autre exemple est la partie concernant les images, où j'ai mentionné qu'il fallait un minimum de 60 % des labels pertinents pour qu'une image soit conservée. Là encore, au lieu de m'appuyer sur une méthode éprouvée, j'ai choisi de créer une fonction personnalisée pour cette tâche. Cela m'a permis de mieux comprendre les mécanismes sous-jacents, mais cela a également allongé le temps de développement.

Réflexion et Leçons Apprises En rétrospective, bien que cette approche de tout vouloir concevoir moi-même m'ait permis de renforcer mes compétences en programmation et en compréhension théorique, elle a également mis en lumière l'importance de savoir quand s'appuyer sur des outils et des frameworks existants pour gagner en efficacité. Il est essentiel de trouver un équilibre entre l'apprentissage et l'efficacité, surtout dans un domaine aussi vaste et complexe que l'intelligence artificielle.

Ces expériences m'ont appris que, bien qu'il soit important de comprendre les bases et de savoir créer des solutions personnalisées, il est tout aussi crucial de reconnaître les limites du

temps disponible et d'utiliser les ressources existantes de manière judicieuse. Cela permet non seulement d'accélérer le développement, mais aussi de concentrer ses efforts sur les aspects du projet où l'innovation et la personnalisation apportent une réelle valeur ajoutée.

1.4 Piste d'amélioration

Voies d'Amélioration du Programme

Au cours de mon travail, plusieurs pistes d'amélioration se sont dégagées pour optimiser le programme, tant au niveau de la performance que de la précision des résultats.

1. Optimisation du Temps de Traitement

L'un des principaux axes d'amélioration concerne le temps de traitement, particulièrement en ce qui concerne le téléchargement des images. Actuellement, le programme peut prendre beaucoup de temps pour télécharger un grand nombre d'images, surtout si la connexion Internet est instable. Ce problème pourrait devenir critique si notre système devait être déployé dans un environnement inconnu où il rencontrerait un grand nombre de nouveaux objets.

Pour remédier à cela, plusieurs solutions pourraient être envisagées : - **Optimisation de la gestion des téléchargements** : Implémenter des techniques de téléchargement parallèle ou asynchrone pourrait réduire considérablement le temps nécessaire pour récupérer les images. De plus, mettre en place un système de cache ou de préchargement intelligent permettrait de minimiser les latences en anticipant les besoins en images. - **Réduction du nombre d'images téléchargées** : En affinant les critères de sélection avant le téléchargement, on pourrait limiter le nombre d'images nécessaires tout en conservant celles qui sont les plus pertinentes. Cela pourrait inclure des pré-filtres basés sur des métadonnées ou l'utilisation de requêtes plus spécifiques pour obtenir des images de meilleure qualité dès le départ.

2. Intégration des Clusters des Labels et des Images

Une autre amélioration potentielle serait de développer une nouvelle fonctionnalité dans le programme qui établirait un lien direct entre les clusters de labels et les clusters d'images après la phase de sélection.

- **Correspondance entre Clusters** : Une fois que toutes les sélections sont effectuées, il serait intéressant d'analyser les relations entre les clusters de labels et les clusters d'images. Cette analyse pourrait révéler des corrélations entre les descriptions textuelles (labels) et les caractéristiques visuelles (images), permettant ainsi de mieux comprendre la représentation de l'objet dans différentes perspectives. - **Amélioration de la Cohérence** : En établissant ces liens, nous pourrions améliorer la cohérence entre les labels et les images, assurant ainsi que les images sélectionnées sont bien alignées avec les concepts représentés par les labels. Cela pourrait également permettre de détecter des anomalies ou des incohérences dans la sélection, conduisant à un filtrage encore plus précis.

3. Évaluation et Validation des Résultats

Enfin, une amélioration essentielle concerne l'évaluation des résultats obtenus. Actuellement, il existe peu de moyens pour déterminer si le programme produit de bons résultats, ce qui complique la validation automatique du processus.

- **Automatisation de l'Évaluation** : Une des priorités serait de développer des outils d'évaluation automatisés. Par exemple, on pourrait implémenter des métriques de performance qui comparent les images finales sélectionnées avec un jeu de données de référence. Cela permettrait d'obtenir une mesure objective de la fiabilité du programme sans nécessiter une vérification manuelle à chaque exécution. - **Comparaison avec d'autres Méthodes** : Il serait également bénéfique de mettre en place un système de benchmarking pour comparer les résultats obtenus avec ceux d'autres méthodes ou algorithmes similaires. Cela pourrait inclure l'intégration de techniques d'évaluation standardisées pour le traitement d'images ou l'intelligence artificielle,

afin de positionner notre approche par rapport à l'état de l'art.

- **Validation Croisée** : L'ajout d'une validation croisée entre les différentes étapes du programme pourrait aider à identifier les points faibles du processus. Par exemple, les résultats obtenus après chaque phase (sélection des labels, sélection des images, clustering) pourraient être réévalués et comparés pour s'assurer de leur cohérence et de leur précision.

Conclusion

En résumé, les axes d'amélioration identifiés portent principalement sur l'optimisation du temps de traitement, l'amélioration de la cohérence entre les labels et les images, et le renforcement des méthodes d'évaluation et de validation des résultats. Ces améliorations non seulement augmenteraient l'efficacité du programme, mais elles permettraient également de garantir une plus grande fiabilité et une meilleure qualité des résultats obtenus. Ces développements futurs seraient cruciaux pour faire face aux défis potentiels liés à la détection d'objets dans des environnements nouveaux et inconnus, tout en assurant une évaluation plus rigoureuse de la performance du programme.

1.5 conclusion

Conclusions générale En conclusion, ce projet m'a offert l'opportunité d'explorer et de manipuler une variété de systèmes et de technologies, enrichissant ainsi mes compétences dans des domaines clés de l'intelligence artificielle. J'ai pu me familiariser avec des outils avancés tels que Spacy pour la comparaison de mots, ce qui m'a permis d'approfondir ma compréhension du traitement du langage naturel. De plus, j'ai eu l'occasion de travailler avec des réseaux de neurones convolutifs (CNN) pour la comparaison d'images, ce qui m'a donné un aperçu précieux des techniques de vision par ordinateur. Par ailleurs, j'ai utilisé les API de Google pour enrichir le projet, ce qui m'a permis d'intégrer des fonctionnalités de labelisation d'image et de recherche internet.

Conclusions Personnel

Ce stage à l'étranger a été une expérience extrêmement enrichissante pour moi, tant sur le plan personnel que professionnel. La découverte d'un nouveau pays, d'une nouvelle ville, et le fait de travailler sur un projet innovant ont été très bénéfiques. Au départ, l'idée de me plonger dans un projet d'intelligence artificielle en étant entièrement seul dessus me faisait un peu peur, surtout que mon introduction à l'IA s'est faite seulement à travers les cours du master. Cependant, le fait d'être dans une université, entouré de professeurs accessibles et d'étudiants expérimentés, a rendu cette expérience beaucoup plus rassurante et stimulante.
En effet travailler dans une université, plutôt que dans une entreprise, m'a offert un environnement d'apprentissage idéal. Dans une entreprise, l'apprentissage est souvent perçu comme un investissement en temps qui, inévitablement, a un coût financier. Ici, dans un laboratoire de recherche, j'ai pus me concentrer sur l'apprentissage nouvelle chose. Les professeurs étaient toujours disponibles pour des discussions approfondies, me permettant d'explorer les concepts complexes de l'IA à mon rythme. Par exemple dans mes problèmes rencontrer j'ai présenté le faites que je voulais tout faire de a à z plusieurs fois mes responsables m'ont dit qu'ils me faisait confiance sur mon developpement et lorsque je commencais a m'éloigner du sujet il venait me remettre dans le chemins et en me disant qu'il fallait plutôt que je m'interrese a tel ou tel concept sans me presser.
Dans le laboratoire, j'étais entouré d'étudiants qui avaient plusieurs années d'expérience en intelligence artificielle. Leur présence a été un atout pour moi car déjà nous venions tous d'horizons différents et nous avons pus parler des cultures de chaqu'un mais aussi sur le technique ou ils ont pus m'aider
Un des principe intéressant du laboratoire de recherche est la présentation de projet.Chaque semaine, un étudiant présentait son projet ainsi que les modèles qu'il utilisait, ce qui permet a chaque personnes d'avoir un domaine sur lequel il se spécialise que comme l'image, la vidéo, les document texte.Et d'approfondir ces connaissant générale en découvrant différents systèmes. Une parties question était prévu a la fin de chaque oral ou toutes les questions étaient les bienvenues. cela pouvait donc pour les pllus début de comprendre le systèmes en générale et pour les plus expérimenté de remettre en question la personnes sur son projet.