

UQTR



Université du Québec
à Trois-Rivières

Mémoire présenté à l'Université du Québec à Trois-rivières comme
exigence partielle de la **Maîtrise en Mathématiques et Informatiques
Appliquées**

INTELLIGENCE ARTIFICIELLE ET AU-DELÀ

Par

Oumar Mohamed Kane

Sous la direction de

Pr. Mhamed Mesfioui

Mars 2024

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

2.4.2.2. Couche de pooling	
2.4.2.3. Couche de correction
2.4.2.4. Couche entièrement connectée
2.4.2.5. La couche de perte
2.4.3. Quelques domaines d'applications
2.5. Exploit majeur réalisé : ChatGPT, le début d'une nouvelle ère
2.5.1. Introduction
2.5.2. La guerre contre les moteurs de recherches
2.5.3. Critiques et regards sur l'avenir de la société
Projet : Prévention de crime avec l'Intelligence Artificielle
3.1. Motivation
3.2. Vision par ordinateur ou Computer Vision
3.2.3. Débouché majeur : Midjourney AI Image Generator
3.3. Revue de littérature
3.3.1. Initiatives déjà prises : l'utilisation de l'Intelligence Artificielle par les forces de l'ordre.	
Recueil intelligent de données à des fins de prédiction
La Reconnaissance faciale
Robots physiques plus intelligents
3.3.2. Analyse et critiques de l'existant
3.3.2.1. La faille de la reconnaissance faciale
3.3.2.2. Antécédents douteux
3.3.2.3. Allées non explorées
Reconnaissance d'outils criminels dans un espace public
Détection de chute dans un espace public
Expression faciale
3.4. Méthodologie
3.4.1. Environnement de développement
3.4.1.1. Langage de programmation
3.4.1.2. Plateforme
3.4.2. Module 1 : Détection d'armes
3.4.2.1. Outils de développement
YOLO
Roboflow

3.4.2.2. Base de données
3.4.2.2. Entraînement du modèle
3.4.2.3. La matrice de confusion et résultats
3.4.2.4. Test conclusif
3.4.3. Module 2 : Détection de chutes
3.4.3.1. Procédure
3.4.3.2. Évaluation sur des données étrangères
3.4.3.3. Analyse des résultats et critiques
3.4.4. Détection de l'expression faciale
3.4.4.1. Bibliothèques
OpenCV
Deepface
3.4.4.2. Utilisation
3.4.4.3. Remarques
3.5. Conditions d'implémentation
3.5.1. Aspect technologique
3.5.2. Aspect légal
Responsabilité
Confidentialité et sécurité des données
Préjugés et discrimination
3.6. Conclusion
Conclusion générale
Références

Remerciements

Je remercie **Le Tout-Clément** 🙏 pour l'air dans mes poumons, pour ces doigts qui ont tenu la plume et pour l'immensurable bénédiction dont Il m'a fait grâce à mon su et insu.

Je remercie mon directeur de recherche **Pr. Mhamed Mesfioui** dont la guidance et le soutien me laissent sans mots. Je n'ai jamais cru qu'on pouvait rendre les mathématiques divertissant et **Pr Mesfioui** l'a fait en mélangeant défis et rires.

Mon travail a été financé en partie par l'octroi accordés au **Professeur Mesfioui** par le **Conseil de recherches en sciences naturelles et en génie du Canada**.

Je tiens aussi à exprimer ma sincère reconnaissance envers **l'Université du Québec à Trois-Rivières (UQTR)** pour avoir rendu cette recherche possible. Les nombreux financements généreusement accordés par l'université ont joué un rôle essentiel dans la réalisation de cette étude.

Ces deux soutiens financiers m'ont permis de consacrer mon temps et mes ressources à la recherche, en garantissant l'accès aux ressources nécessaires, aux données et à l'expertise requise.

Je tiens aussi à remercier le Directeur de comité de programme de cycles supérieurs, **Pr François Meunier** qui, et je crois parler pour tous les étudiants du département, a été un support indispensable au cours de toutes les sessions de notre programme.

Sans oublier les professeurs **Nadia Ghazzali** et **B. Amar Bensaber** du jury pour leurs exhaustives évaluations de ce présent; ces dernières incluant aussi celle du Professeur **Mesfioui** me seront une lanterne pour le restant de mon parcours académique.

Je remercie aussi mes parents et toutes les personnes qui m'ont soutenu et encouragé pour faire face à la grande tornade qu'a été ces deux dernières années.

Introduction

Il ne devient rien d'autre qu'un pléonasme de dire que la technologie évolue à une vitesse excessive, ce qui n'était que fiction il y a à peine une dizaine d'années se voit se réaliser et devenir une norme pour l'individu lambda. La transmission d'information ne se fait plus par le biais d'un pigeon, mais s'effectue par un simple clic sur la souris de votre ordinateur portable ou par un simple appui sur un bouton sur votre appareil tactile.

Mais elle est loin de s'arrêter à juste digitaliser toutes vos données, elle en est au point de pouvoir prédire vos futurs besoins et automatiser toutes vos activités de routines quotidiennes. Au fil de ces dix dernières années, les scientifiques se sont donné un et un seul objectif : « *Offrir les fonctions cognitives de l'Homme aux machines* » ; ainsi ce dernier donna naissance à la doctrine de l'intelligence artificielle (IA), une technologie qui permettrait à la machine d'émuler l'intelligence humaine résolvant des problèmes de différents niveaux de complexités.

Leur croyance en de telles perspectives, que certains considéreraient être des absurdités, n'a fait que croître avec l'amélioration des performances des processeurs qui d'après des études, ne cessent d'évoluer en termes de performances chaque année durant ces cinq dernières.

Pour procéder à un tel exploit, les scientifiques ont dû fragmenter cette doctrine en trois grands sous-domaines tels que le **Machine Learning** (l'apprentissage automatique), les **Neural Networks** (**Les réseaux neuronaux**) et le **Computer Vision** (la vision par ordinateur).

L'apprentissage automatique a comme objectif de permettre à l'ordinateur d'apprendre et de pouvoir prendre des décisions tout seul. Le réseau neuronal est une imitation de la structure cérébrale pour pouvoir reconnaître des patrons, et la vision par ordinateur permet à l'ordinateur de reconnaître les objets qu'il voit.

Ces deux dernières années en particulier se voient accueillir une percée technologique qui marque le début d'une nouvelle ère industrielle qui considère presque tout programme informatique dépourvu de l'intelligence artificielle comme obsolète, et comme on peut le supposer, une telle technologie ne peut être qu'accompagnée d'un nombre de questions; l'intelligence artificielle fait certes fureur mais offre-t-elle réellement assez de solutions au point d'en devenir une nécessité ? Quelles sont les risques qu'elle présente ? Quelle est l'importance de ces dernières par rapport aux bénéfices qu'elle apporte ? Finalement, quelles sont les demandes de l'IA d'une perspective technologique et légale ?

Ce mémoire est structuré de la façon suivante :

- **Une vue approfondie sur les sous-domaines de l'IA** : Nous élaborerons sur chacun des sous-domaines de l'IA en les définissant d'une manière plus exhaustive mathématiquement et informatiquement, en mettant en emphase leurs bienfaits et leurs limites et en présentant leurs débouchés majeurs.
- **L'usage à esient controversé de l'IA par la société** : Nous discuterons aussi de l'utilisation de l'intelligence artificielle par différents secteurs composant la société, l'opinion de cette dernière sur l'IA et sa perspective sur leur future ensemble.
- **Projet de prévention de crime avec l'Intelligence artificielle** : Cette partie représente la contribution principale de ce mémoire. Avec l' intention de prévenir ou réduire les fréquentes crimes qui ont lieu dans les espaces publiques, nous développerons trois modules notificateurs définis de la suivante sorte :
 - **Détection d'arme** : Nous ferons usage des dernières technologies de pointe de la **Vision Par Ordinateur (VPO)** pour détecter toute sorte d'arme dangereuses dans les endroits publiques grâce aux caméras de surveillance.
 - **Détection d'émotions** : Toujours grâce à la VPO, nous allons détecter les émotions sur les visages des individus dans les lieux publiques; des émotions pouvant indiquer un danger imminent comme la **peur** sur le visage d'un individu effrayé par des bandits ou la colère d'un homme vindicatif, etcetera.

Chapitre 1 : Machine Learning ou Apprentissage Machine

L'apprentissage automatique, ou Machine Learning en anglais est le sous-domaine de l'intelligence artificielle qui fait usage d'algorithmes pour apprendre à partir de données fournies et faire des prédictions ou prendre des décisions sans aucune programmation (instructions) explicites. Le but de l'apprentissage automatique est de construire des modèles qui pourront se généraliser en de données nouvelles et inconnues pour faire de précises prédictions et décisions.

Un modèle est une représentation mathématique d'un système ou d'un processus qui pourrait être utilisé pour faire des prédictions ou prendre des décisions selon une donnée d'entrée; il s'agit essentiellement d'une fonction qui associe une donnée d'entrée à une variable de sortie (aussi appelée variable cible).

Il y a principalement trois types d'apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement; dans l'apprentissage supervisé, le modèle est entraîné selon un ensemble de données étiquetées, dans ce cas là le résultat final désiré est déjà connu; dans l'apprentissage non supervisé, l'algorithme apprend d'un ensemble de données non étiquetées, cela est utilisé pour des tâches telles que le partitionnement de données et la réduction de dimensionnalité; et enfin, dans l'apprentissage par renforcement, l'algorithme apprend de ses interactions avec l'environnement, en utilisant tentatives et erreurs pour maximiser les signaux de récompenses.

En premier lieu, nous allons élaborer davantage sur les trois grands types d'apprentissage automatique qu'on vient de mentionner et en second et dernier lieu, nous énumérerons quelques algorithmes communément utilisés dans l'apprentissage automatique.

1.1. Les grands types d'apprentissage automatique

Il y a principalement trois types d'apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement. Bien qu'il y ait d'autres sous-types spécialisés d'apprentissage automatique, nous nous concentrerons sur les principaux.

1.1.1. L'apprentissage supervisé

Les algorithmes d'apprentissage supervisé construisent un modèle mathématique à partir d'un ensemble de données qui se compose de données d'entrée et des résultats désirés. Les données sont composées de spécimens d'entraînement. Chaque spécimen est constitué d'une ou plusieurs données d'entrées (input) et du résultat désiré correspondant (output) aussi appelé signal de supervision. Dans le modèle mathématique, ces spécimens sont représentés par des vecteurs qu'on dit vecteurs de caractéristiques et un ensemble de données est logiquement représenté par une matrice.

À travers une optimisation itérative effectuée par une **fonction objectif**, les algorithmes d'apprentissage supervisé apprennent une fonction qui pourra prédire des résultats précis pour une toute nouvelle entrée, c'est à dire une entrée qui ne figure pas dans son ensemble de données d'entraînement. La fonction objectif mesure la performance d'un modèle d'apprentissage automatique, elle prend en paramètres les prédictions du modèle et les étiquettes associées ou les résultats ciblés et retourne une valeur scalaire relatant le degré de précision des prédictions.¹

Parmi les types d'apprentissages supervisés, nous comptons l'apprentissage actif, la classification supervisée et la régression. L'apprentissage actif est un cas spécial de l'apprentissage automatique dans lequel l'algorithme apprend en enquêtant interactivement auprès de l'utilisateur ou une autre source d'information pour associer de nouvelles données aux sorties désirées. Les algorithmes de classification sont utilisés lorsque les sorties (résultats) sont restreints à un ensemble limité de valeurs; par exemple, pour une musique donnée, la classification déterminerait le genre, Pop, Rock, Jazz etc. Par ailleurs, l'algorithme de régression ne restreint pas ses résultats à un ensemble de valeurs mais plutôt une plage de valeurs numériques.

L'apprentissage de métriques est une spécialisation de l'apprentissage supervisé, qui est étroitement liée à la classification supervisée et à la régression, son but est d'apprendre à partir d'exemples en utilisant une fonction de similarité qui mesure le degré de similarité ou de parenté entre deux objets. Il est appliqué dans une variété de domaines tels que la reconnaissance faciale et les systèmes de recommandations.

1.1.2. L'apprentissage non supervisé

Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé (unsupervised learning en anglais) ne requiert pas l'étiquetage des données, il prend un ensemble de données qui ne comprend que des entrées puis tente d'identifier des patrons et structures significatives. En d'autres termes, il apprend de données qui ne sont ni étiquetées, ni classifiées et ni catégorisées. Au lieu de répondre aux rétroactions, l'algorithme d'apprentissage non supervisé tente d'identifier les

points communs des données et réagit selon la présence ou l'absence de ces points communs dans chaque nouvelle donnée. L'apprentissage non supervisé est centralement utilisé dans l'estimation de densité en statistique, par exemple pour chercher la densité d'une variable aléatoire. Cela dit, il est néanmoins utilisé dans d'autres domaines impliquant la synthèse et l'explication des caractéristiques des données.

Le type d'apprentissage non supervisé le plus commun est la classification non supervisée (ou clustering en anglais); Elle consiste à regrouper les points de données similaires selon une certaine mesure de similarité ou de distance. Un autre type commun d'apprentissage non supervisé est la réduction de dimensionnalité qui implique la réduction du nombre de variables dans un ensemble de données tout en préservant la plus importante information ou structure.

L'utilité de l'apprentissage non supervisé devient très apparente quand il s'agit de données très complexes ou très chères à étiqueter (coût de performance), ou dans le plus fréquent des cas où les étiquettes ne sont pas connus d'avance. ²

1.1.3. Apprentissage par renforcement

L'apprentissage par renforcement est un type d'apprentissage automatique qui consiste à entraîner un agent logiciel (ou agent autonome) à prendre des décisions dans un environnement donné afin de maximiser un signal de récompenses cumulatives. L'agent en question est un logiciel qui agit de manière autonome, il accomplit des tâches à la manière d'un automate.

L'agent interagit avec l'environnement en passant à l'action et en recevant des retours sous formes de récompenses ou de pénalités; il utilise ces retours pour mettre à jour sa politique et améliorer, au fil du temps, sa capacité de prendre des décisions. En raison de sa généralité, ce type d'apprentissage est étudié dans plusieurs autres disciplines, telles que la théorie des jeux, la théorie du contrôle, la recherche opérationnelle, la théorie de l'information, la statistique et aussi dans l'algorithme génétique. Dans un problème d'apprentissage par renforcement, un agent interagit avec l'environnement modélisé en processus de décision markovien, l'environnement est composé d'un ensemble d'états et un ensemble d'actions possibles; À chaque étape temporelle, l'agent choisit une action selon l'état actuel de l'environnement et une politique, qui est un ensemble de règles selon lesquelles une action est choisie. L'état de l'environnement se transforme avec une certaine probabilité selon l'action choisie par l'agent et ce dernier reçoit une récompense ou une pénalité associée au nouvel état.

1.2. Les algorithmes communs de l'apprentissage automatique

Il y a plusieurs algorithmes d'apprentissage automatique, chacun muni de ses points forts et faibles, dans cette section nous citerons en expliquant brièvement les algorithmes les plus communs de l'apprentissage automatique.

1.2.1. Régression linéaire

La régression linéaire est une méthode statistique utilisée pour modéliser la relation entre une variable dépendante et une ou plusieurs variables indépendantes. La variable dépendante est celle qui est prédite, tandis que la variable indépendante est celle dont on fait usage pour faire la prédiction; par exemple dans le scénario où on voudrait savoir si les personnes de grandes tailles pèsent plus lourd que celles de tailles inférieures, la hauteur de la personne est la variable indépendante et son poids est la variable dépendante. Dans la régression linéaire, la relation entre la variable dépendante Y et la variable indépendante X est présumée linéaire, cela veut dire qu'on considère que l'évolution de la variable indépendante est proportionnelle à celles des variables dépendantes. ³

Le but de régression linéaire est de trouver le meilleur ajustement de la droite tel qu'elle prédise la valeur de la variable dépendante en se basant des valeurs des variables indépendantes. La droite la mieux ajustée est celle qui minimise la somme des différences entre les valeurs prédites et les valeurs réelles de la variable dépendante.

Dans le contexte de l'apprentissage automatique, il s'agit d'un algorithme d'apprentissage supervisé utile dans plusieurs domaines telles que la finance, l'économie, les sciences sociales, l'ingénierie et autres pour analyser et modéliser la relation entre variables.

Ci-dessous se trouvent les formules pour une simple régression linéaire avec une seule variable indépendante :

L'équation de la droite de régression estimée est la suivante (voir **Fig 1**) :

$$\hat{y} = \hat{\alpha}x + \hat{\beta} ,$$

où :

- \hat{y} est la variable prédite, soit la variable dépendante
- x est la variable explicative, soit la variable indépendante
- $\hat{\alpha}$ est la pente estimée de la droite
- $\hat{\beta}$ est l'ordonnée à l'origine estimée.

La formule pour estimer les coefficients de régression est la suivante :

$$\hat{\alpha} = \frac{\sum_i x_i \hat{y}_i - \sum_i x_i \sum_i \hat{y}_i}{\sum_i x_i^2 - (\sum_i x_i)^2},$$

$$\hat{\beta} = \frac{1}{n} \sum_i \hat{y}_i - \hat{\alpha} \frac{1}{n} \sum_i x_i.$$

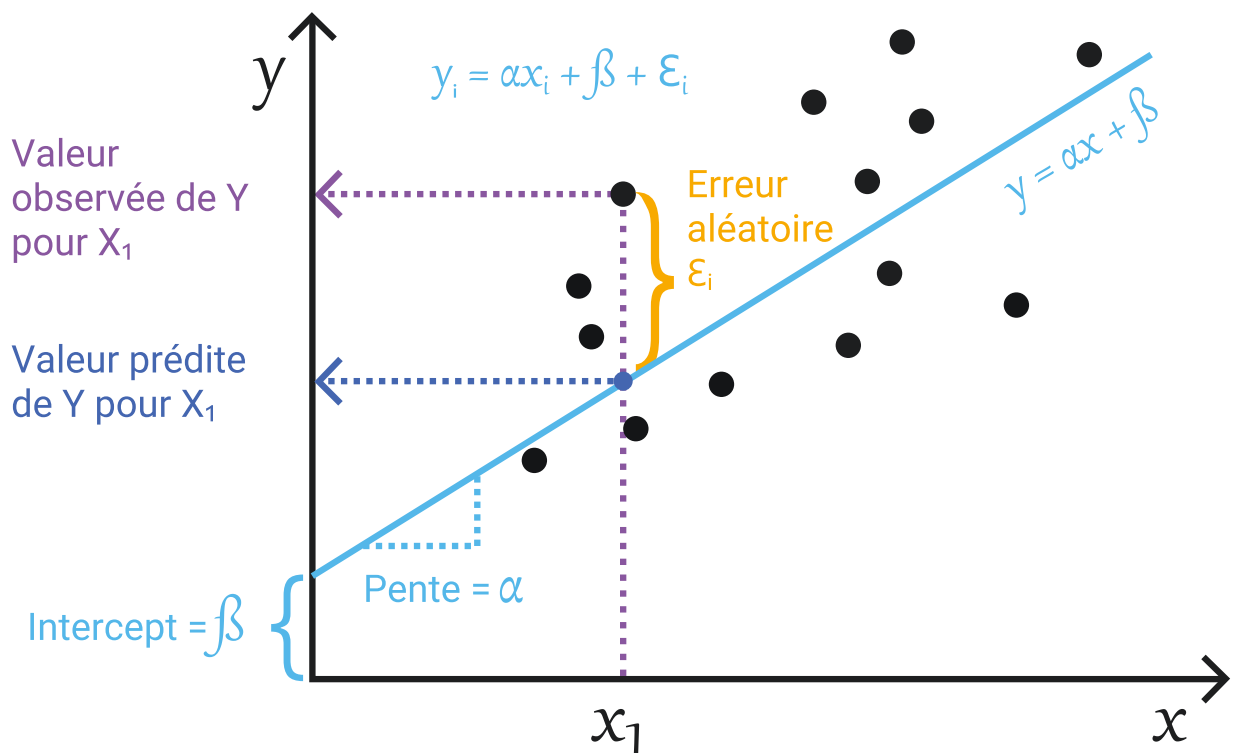


Fig 1 - Régression linéaire

1.2.2. Régression logistique

La régression logistique est un modèle statistique utilisé pour analyser et modéliser la relation entre une variable dépendante binaire et une ou plusieurs variables indépendantes. En d'autres termes, c'est un type d'analyse de régression utilisé quand la variable dépendante est catégorielle, c'est à dire limitée à deux possibles issues, et les variables indépendantes peuvent être continues ou catégorielles aussi. ⁴

Il s'agit d'un algorithme d'apprentissage supervisé utilisé dans des problèmes de classifications. Son but est de trouver la courbe, à la forme de S, la mieux ajustée pour décrire la relation entre une ou plusieurs variables d'entrées (variables indépendantes) et une seule variable de sortie (variable dépendante).

La fonction de cette courbe, appelé fonction sigmoïde ou fonction logistique, associe les valeurs d'entrée à un score compris entre 0 et 1 qui représente la probabilité prédite que la variable dépendante tombe dans une des deux catégories.

La fonction logistique est donnée par la formule suivante (voir **Fig 2**) :

$$p(z) = \frac{1}{1 + e^{(-z)}} ,$$

où :

- z est la combinaison linéaire des variables indépendantes et leurs coefficients :

$$z = \beta + \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_k x_k$$

- où β est l'ordonnée à l'origine ou la constante, et $\alpha_1, \alpha_2, \dots, \alpha_k$ sont les coefficients des variables indépendantes x_1, x_2, \dots, x_k .

- $p(z)$ est la probabilité expliquée précédemment

Le modèle de régression logistique estime les valeurs des coefficients $\alpha_1, \alpha_2, \dots, \alpha_k$ de sorte à minimiser la différence entre les probabilités prédites $p(z)$ et les valeurs réelles de la variable dépendante.

La régression logistique est un algorithme populaire car il est simple, interprétable et peut être entraîné efficacement dans de larges bases de données. Elle est utilisée dans une panoplie d'applications, telles que l'évaluation du risque de crédit, la détection des fraudes et le filtrage des spams, entre autres.

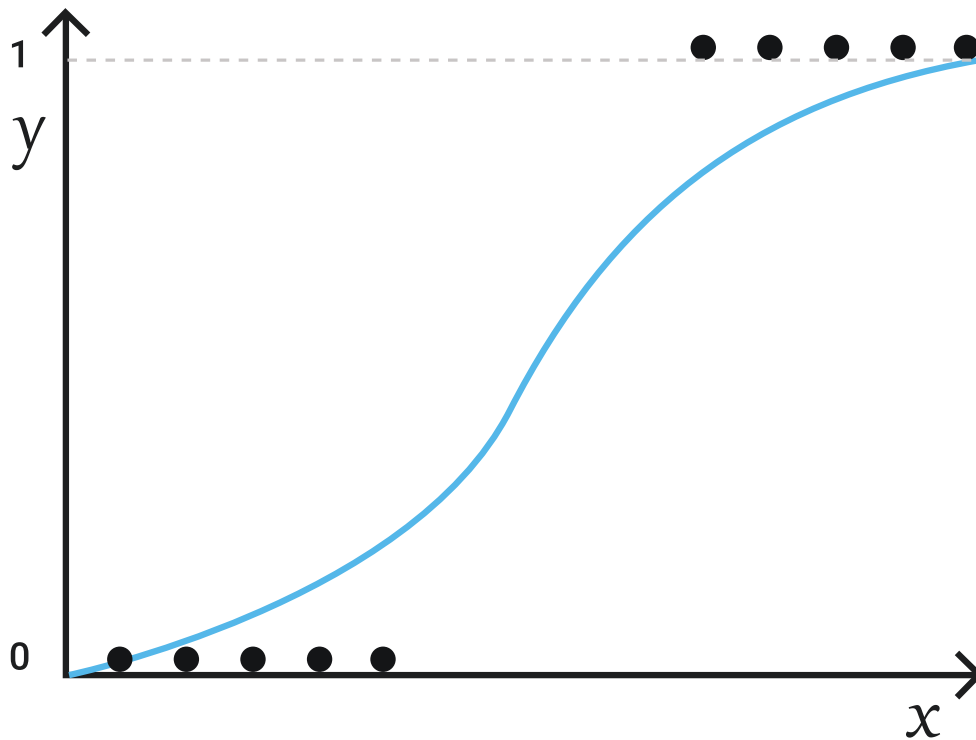


Fig 2 - Régression logistique

1.2.3. Arbre de décision

L'arbre de décision est un algorithme d'apprentissage automatique populaire qui est principalement utilisé dans les tâches de classification et de régression. Dans cet algorithme, la base de données est divisée en sous-ensembles selon les caractéristiques qui contribuent le plus à la prédiction. Ces sous-ensembles sont divisés récursivement selon les valeurs de différentes caractéristiques, et ce processus continue jusqu'à ce qu'un certain critère d'arrêt soit satisfait, tel qu'un nombre minimum d'échantillons dans un sous-ensemble ou l'atteinte d'un certain niveau de profondeur dans l'arbre.

Le nom arbre a été préféré parce que l'algorithme représente les données dans une structure hiérarchique semblable à celle d'un arbre (voir **Fig 3**); le nœud racine représente l'entièreté de la base de données et les nœuds subséquents représentent les sous-ensembles résultant de la division de la base de données en fonction des caractéristiques; chaque nœud interne correspond à une décision, un test sur une caractéristique et chaque branche émanant de ce nœud représente un

résultat de ce test. Chaque feuille (nœud non partitionné) correspond à un résultat final ou une prédiction à laquelle aboutit un processus de prise de décision.

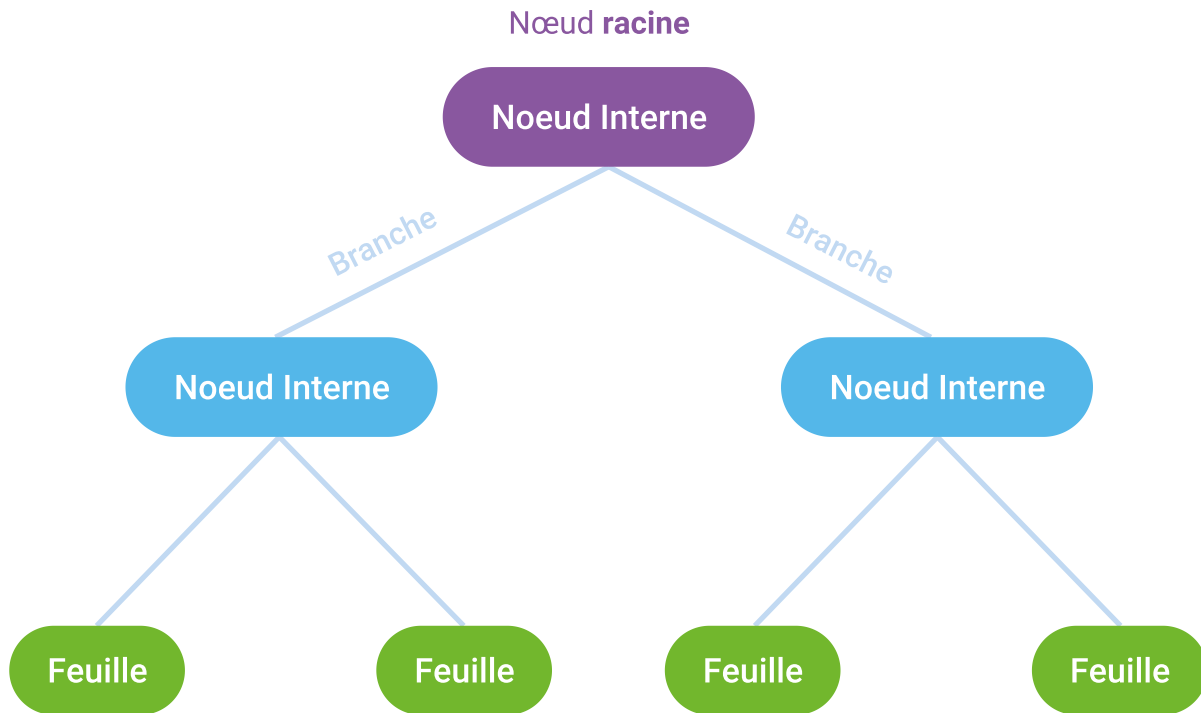


Fig 3 - Arbre de décision

La division des nœuds s'effectue en itérant sur les règles de décisions; une règle de décision est une condition (le test mentionné précédemment) pour prendre une décision. On commence par diviser le nœud racine en itérant sur toutes les règles de décisions, puis on compare le gain d'informations de toutes les itérations et on choisit la division qui nous permet le plus d'informations. On répète le même processus pour les nœuds subséquents en tâchant de ne plus considérer une règle de décision précédemment évaluée dans la même branche. La division s'arrête s'il n'y a plus de règles de décision à considérer, si l'index d'impureté du sous-ensemble est égale à 0 (groupe totalement homogène), ou si un critère d'arrêt, comme ceux mentionnés précédemment, est satisfait.

Par exemple, soit une base de données de patients avec leurs diagnoses correspondants (cancer ou pas cancer), l'algorithme évaluera différentes règles de décisions telles que "âge supérieur à 50", "taille de la tumeur supérieure à 5 cm" ou "antécédents familiaux de cancer". L'algorithme sélectionne la règle qui offrira la meilleure division, c'est-à-dire la règle qui séparera le plus efficacement les patients atteints de cancer et ceux qui ne le sont pas.

Toutefois, une inconvénience de cet algorithme est le fait qu'il tend à surapprendre les données, surtout si l'arbre est trop profond ou s'il y a un très grand nombre de caractéristiques. Le surapprentissage se produit lorsque le modèle est tellement bien entraîné qu'au lieu d'apprendre des données d'entraînement et en découvrir les patrons sous-jacents, il finit par les mémoriser et il en résulte une excellente performance sur les mêmes données (les données d'entraînement) et une très mauvaise performance sur de toutes nouvelles données d'entrées.

Le surapprentissage est presque certain lorsque le modèle est très complexe, parce-que dans ce cas, le modèle est muni de plusieurs paramètres et donc a une plus grande capacité de s'adapter aux données d'entraînement. Il sera plus apte à détecter ne serait-ce que les plus petites irrégularités dans les données d'entraînement, celles-là comprennent les valeurs aberrantes et les fluctuations aléatoires qui ne sont pas représentatives des patrons sous-jacents.

Une technique appelée élagage nous permet de remédier au surapprentissage; il s'agit de supprimer les branches et nœuds qui ne sont pas nécessaires ou qui ne contribuent pas à la performance de l'arbre, ce qui réduit la complexité de ce dernier et réduit le risque de mémoriser les données bruitées, et tout cela aide grandement sa capacité de généralisation.

L'algorithme de l'arbre de décision est populaire car il est simple, facile à comprendre et est capable de traiter à la fois les données catégorielles et continues. En plus de cela, il peut gérer les valeurs manquantes et les données bruitées.

1.2.4. L'Analyse en composantes principales

L'Analyse par composantes principales ou ACP est un algorithme populaire d'apprentissage non supervisé utilisé pour réduire la dimensionnalité de large base de données, en transformant un grand nombre de variables en un nombre largement inférieur tout en préservant une très bonne portion de l'information originale. Comme on pourrait le soupçonner, réduire le nombre de variables s'effectue au dépens de la précision ou de la véracité; cela dit, le compromis dans l'ACP c'est échanger un petit peu d'exactitude contre de la simplicité.

Les variables seront résumées en **composantes** représentant chacune un pourcentage de la variation des données.

Dans un premier temps, nous allons parler des notions prérequis pour conduire une ACP et dans un second, nous allons finalement décrire les étapes nécessaires.

1.2.4.1. Prérequis de l'ACP : L'écart-type et la variance estimées

Il s'agit là de deux notions requises pour une bonne compréhension de l'analyse en composante principale.

L'écart-type estimé est la variabilité moyenne dans un ensemble de données de taille n , il exprime la distance moyenne entre une observation aléatoire et la moyenne.

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} .$$

La variance estimée est la mesure de la variabilité, en d'autres termes elle mesure la dispersion dans un ensemble de données de taille n . La variance estimée est égale à l'écart-type estimé au carré.

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 .$$

1.2.4.2. Les étapes de l'ACP

La standardisation des données

La standardisation est importante dans l'ACP car elle permet de mettre toutes les variables à la même échelle, ce qui évite que les variables ayant des écarts types plus importants ne dominent l'analyse. En standardisant les variables, on permet également aux variables ayant des unités de mesure différentes d'être comparables. La formule pour standardiser est la suivante :

$$x_{sd} = \frac{x - \bar{x}}{S} .$$

Établir la matrice de covariance et/ou celle de la corrélation

Tandis que la variance mesure la variation d'une seule variable aléatoire, la covariance elle mesure combien deux variables varient ensemble, la formule pour la covariance estimée est la suivante pour une population :

$$S_{xy} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) .$$

Si les variables sont à des échelles différentes ou mesurées dans des unités différentes, la matrice de corrélation est généralement utilisée car elle normalise l'échelle des variables, permettant une comparaison équitable. La matrice de corrélation est essentiellement une version normalisée de la

matrice de covariance où chaque élément représente le coefficient de corrélation entre deux variables :

$$C_{xy} = \frac{S_{xy}}{S_x S_y}.$$

Les vecteurs propres et les valeurs propres

Un **vecteur propre** représente une nouvelle propriété qui diffère fortement entre les observations, mais tout en gardant une relation avec les anciennes propriétés, nous permettant ainsi de prédire ou reconstruire les caractéristiques originales aussi bien que possible. En d'autres termes, on cherche la variance maximale et l'erreur minimum (la distance de chaque projection qui nous permet de reconstruire la position des observations); ces deux sont atteints en même temps.

Géométriquement parlant, il s'agit de ces vecteurs qui ne changent pas de directions après une **transformation linéaire**; cette dernière n'impacte que leurs dilatations et le rapport de cette dilatation est ce qu'on appelle la **valeur propre**.

Soit A une matrice carrée de dimension $n \times n$, on dit λ est une valeur propre de A, s'il existe un vecteur non nul \vec{u} de dimension $n \times 1$ tel que :

$$A \cdot \vec{u} = \lambda \cdot \vec{u}.$$

Cette équation peut être formulée autrement :

$$A \cdot \vec{u} - \lambda I \cdot \vec{u} = 0 ;$$

$$(A - \lambda I) \vec{u} = 0 .$$

Et comme nous voulons une solution non nul pour \vec{u} , nous devons opérer selon le cas où $(A - \lambda I) = 0$, or pour que le produit d'une matrice avec un vecteur non nul soit égale à 0, il faut que la transformation associée à cette matrice écrase l'espace en une dimension inférieure et cet écrasement correspond à un déterminant nul :

$$\det(A - \lambda I) = 0 .$$

Il s'agit là du polynôme caractéristique correspondant à la matrice A.

Après avoir calculé les vecteurs propres et leurs valeurs propres correspondantes, il faut :

- Trier les valeurs propres et les vecteurs propres correspondants.

- Soit k la dimension souhaitée, choisir les k premiers vecteurs et former une matrice U de vecteurs propres.

Et finalement calculer la matrice F finale à dimension réduite :

$$X_s \cdot U = F ,$$

avec X_s l'ensemble des observations normalisées. Voilà comment se réduit la dimensionnalité d'un ensemble de donnée.

1.2.5. Classification Naïve bayésienne

La classification naïve bayésienne est un algorithme probabiliste d'apprentissage supervisé; elle est basée sur le théorème de Bayes et est communément utilisée dans l'analyse de textes. Les algorithmes de classification probabiliste prédisent la probabilité qu'un échantillon appartienne à une classe ou catégorie selon une caractéristique observée; il s'agit donc d'une probabilité conditionnelle, car on détermine la probabilité qu'un événement se produise sachant qu'un autre événement s'est produit.

L'algorithme présume que toutes les caractéristiques des données d'entrées sont indépendantes l'une de l'autre, d'où l'adjectif "naïve", cela veut dire que la présence ou l'absence d'une caractéristique n'affecte en rien la probabilité de la présence ou de l'absence d'une autre.

La classification naïve bayésienne opère en calculant la probabilité à priori de chaque classe (ou catégorie) selon sa fréquence dans les données d'entraînements, puis, pour chaque caractéristique, l'algorithme calcule sa probabilité pour une classe donnée, en utilisant la fréquence de cette caractéristique dans cette même classe, et finalement, il calcule la probabilité à posteriori de chaque classe étant données les caractéristiques en utilisant le théorème de Bayes.

Par exemple, pour entraîner une application anti-spam, nous avons une base de données d'entraînement de 12 courriels parmi lesquels 8 sont des courriels normaux et 4 sont des spams (aussi appelés pourriels ou messages frauduleux); on commence par calculer la probabilité à priori de chaque catégorie, soit $P(N)$ la probabilité qu'un courriel soit normal et $P(S)$ la probabilité qu'un courriel soit un spam, l'opération s'effectue comme suit :

$$P(N) = \frac{\text{nombre de courriels normaux}}{\text{le nombre total de courriels}} = \frac{8}{12} = 0.67 ;$$

$$P(S) = \frac{\text{nombre de spams}}{\text{le nombre total de courriels}} = \frac{4}{12} = 0.33 .$$

Les messages sont composés de mots, et les mots représentent les caractéristiques dans la notion générale de l'algorithme, et à chaque mot m , sa fréquence $freq$ dans chaque catégorie $C = N | S$ et sa probabilité étant donnée cette catégorie $P(m | C)$.

Prenons comme exemple le tableau suivant (voir **Tab 1**):

m	$freq_N(m)$	$P(m N)$	$freq_S(m)$	$P(m S)$
Cher	14	0.35	3	0.094
Famille	11	0.275	2	0.063
Ami	10	0.25	3	0.094
Gratuit	2	0.05	10	0.313
Argent	3	0.075	14	0.438

Tab 1 - Exemple de fréquences et probabilités de mots dans les courriels

Pour calculer la probabilité qu'un nouveau courriel soit normal ou un spam, nous allons faire référence à la table précédente pour procéder; supposons qu'on reçoit un nouveau message M disant : «Cher ami», pour prédire s'il s'agit d'un message normal ou d'un spam, nous devons déterminer, pour chaque catégorie C , la probabilité $P(C | M)$ qu'il fasse partie de cette dernière, or l'équation de cette probabilité est la suivante :

$$P(C | M) = P(C) \times \prod_{i=1}^n P(m_i | N) ,$$

donc la probabilité $P(N | M)$ que le courriel soit normal est égale à :

$$P(N | M) = 0.67 \times 0.35 \times 0.25 ,$$

$$\boxed{P(N | M) = 0.059} .$$

et la probabilité $P(S | M)$ que le courriel soit un spam est égale à :

$$P(S | M) = 0.33 \times 0.094 \times 0.094 ,$$

$$P(S | M) = 0.003 .$$

$P(N | M)$ étant supérieur à $P(S | M)$, on en conclut (prédit) que le courriel est un courriel normal.

Vous noterez que l'ordre des mots n'a aucune pertinence dans cet algorithme d'où sa naïveté, car comme vous le savez, dans toutes les langues humaines l'emplacement des mots dans une phrase a une très grande importance et établit le sens de la phrase; cela dit, en pratique, cet algorithme donne de très bon résultat et est fréquemment utilisé pour filtrer les messages.

Dans le cas où nous recevons un message composé d'un nouveau mot, c'est à dire un mot indisponible dans les messages d'une catégorie ou dans toute la base de données d'entraînement, nous aurons une fréquence qui est égale à 0, ce qui est peu convenable car une probabilité à posteriori sera égale à 0; la solution à ce dilemme serait d'ajouter +1 à toutes les fréquences de tous les mots.

En conclusion, la classification naïve bayésienne a plusieurs avantages tels que sa simplicité, son efficacité et sa capacité de gérer des données de très grandes dimensions; l'algorithme ne requiert pas une très grande base de données d'entraînement contrairement à un bon nombre d'algorithmes, ce qui le rend utile dans les situations où les données d'entraînements sont limitées.

Chapitre 2: Réseaux neuronaux ou Neural Networks

2.1. Introduction

Le réseau de neurones artificiels représente un sous-domaine de l'apprentissage automatique, il s'agit d'un système inspiré des réseaux neuronaux biologiques du cerveau humain dans le but de reproduire le comportement de ce dernier. C'est l'algorithme le plus approprié pour effectuer des tâches telles que la reconnaissance d'image, le traitement de langage naturel et la reconnaissance vocale.

Le réseau neuronal se compose d'une collection d'unités connectées ou nœuds appelés neurones artificiels; chaque connexion, appelée synapse comme celui d'un cerveau biologique, peut transmettre un signal d'un neurone à un autre. Un neurone artificiel reçoit un signal, le traite puis en transmet un autre à son tour vers d'autres neurones qui sont liés à lui. Chaque synapse est associée à un poids synaptique qui augmente ou réduit l'ampleur du signal. Le signal est un nombre réel et la sortie de chaque neurone est traitée en calculant la somme des signaux d'entrées pondérés puis en faisant passer cette dernière par une fonction d'activation ⁵. La somme peut être mathématiquement représentée comme suit :

$$\sum_{i=1}^n E_i \times P_i ;$$

où E_i représente une entrée et P_i le poids associé.

La fonction d'activation introduit la non-linéarité dans le résultat, c'est-à-dire la sortie du neurone, ce qui permet au réseau d'apprendre des relations complexes entre entrées et sorties. Différentes fonctions d'activation peuvent être utilisées, et la plus commune est la fonction sigmoïde. La fonction sigmoïde se représente graphiquement par une courbe de la forme de la lettre S qui associe toute valeur à un nombre compris entre 0 et 1. Sa formule est la suivante :

$$f(z) = \frac{1}{1 + e^{-z}}$$

Un neurone peut aussi avoir un seuil, de sorte qu'un signal pondéré ne sera transmis que s'il est supérieur ou égale au mentionné seuil.

Avant d'aller plus loin, nous allons tout d'abord expliquer d'une manière générale comment se déroule l'entraînement d'un réseau neuronal. En second lieu, pour une meilleure idée de son

fonctionnement et de ses cas d'utilisations, nous allons naviguer les intentions des premiers chercheurs qui ont rendu possible ce potentiel déterminant du futur de la technologie moderne. Troisièmement, nous ferons passer sous la loupe, l'un des plus communs types de réseaux neuronaux dont l'implémentation est devenue aujourd'hui le premier choix de toutes les applications d'intelligence artificielle, le réseau neuronal convolutif. En quatrième et dernier lieu, nous parlerons d'un débouché révolutionnaire des réseaux neuronaux, le prototype d'agent conversationnel **ChatGPT** qui fait ravage dans tous les secteurs constituant notre société et qui pose un sérieux risque à cette dernière.

2.2. Entraînement supervisé d'un réseau neuronal

Les réseaux neuronaux apprennent en traitant des exemples; chacun de ces exemples se compose d'une entrée et d'un résultat connus. L'entraînement supervisé est effectué en évaluant la différence entre la sortie du réseau ou prédiction et la sortie cible, c'est-à-dire le résultat correcte souhaité; cette différence est appelée erreur. Le réseau ajuste les poids et les seuils selon une règle d'apprentissage en utilisant la valeur de cette erreur pour améliorer les prédictions.

L'objectif est de minimiser la valeur d'erreur, et l'apprentissage est considéré complet lorsque le traitement de nouveaux exemples ne réduit plus considérablement la valeur d'erreur. Après quelques bonnes itérations, la prédiction devrait être de plus en plus similaire à celle du résultat souhaité. Si après un bon nombre d'itérations la valeur d'erreur reste élevée, le système doit être repensé, ainsi que la fonction de coût, c'est-à-dire la fonction qui mesure la différence entre la valeur prédite et la valeur cible.

Le système, comme précisé précédemment, apprend en traitant des exemples, et cela sans recevoir des règles spécifiques à la tâche en main; en prenant la reconnaissance d'image à titre d'exemple, pour faire apprendre le système à reconnaître un être humain, on lui fournit un ensemble d'images avec des êtres humains présents et d'autres sans, et chacune de ces images avec une étiquette "avec humain" ou "sans humain", et le système apprendra sans qu'on ait à préciser des informations supplémentaires telles que le fait qu'un être humain a deux bras ou se tient debout sur deux jambes, etc; au lieu de cela, il va générer automatiquement des caractéristiques d'identification à partir des exemples qu'on lui a fournis.

Cela dit, dans le cas non supervisé de l'apprentissage avec les réseaux de neurones artificiels (ANN), puisqu'il n'y a pas d'étiquettes avec lesquelles comparer les résultats, l'erreur ou la perte est calculée de différentes manières en fonction de la tâche ou de l'architecture spécifique du réseau de neurones.

2.3. Un peu d'histoire...

Contrairement à ce qui est écrit dans plusieurs journaux historiques relatant l'origine des réseaux de neurones artificiels, ces derniers ont vu le jour bien avant les travaux de McCulloch et Pitts ⁶. La régression linéaire, existant depuis plus de deux siècles, est une forme de réseaux de neurones à propagation avant (RNPA) nommée réseau linéaire. Le RNPA est le premier type de réseau de neurones artificiels conçu et il est le plus simple aussi car les connexions entre ses nœuds ne forment pas un cycle, c'est-à-dire que l'information n'avance que vers une seule direction, en avant, depuis les nœuds d'entrées, passant par les nœuds cachés (intermédiaires) s'il y en a, et finalement elle aboutit aux nœuds de sorties. La forme la plus simple du RNPA est un réseau linéaire qui consiste d'une seule couche de nœuds de sorties, les valeurs d'entrées sont directement alimentées aux neurones de sorties de la seule couche via une série de poids (voir Fig 4).

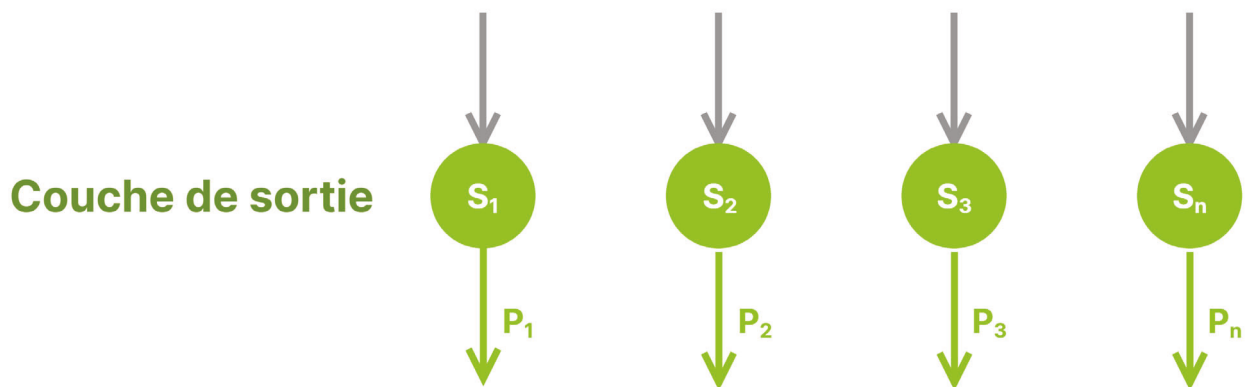


Fig 4 - La forme la plus simple de RNPA (S pour neurone de sortie et P pour poids)

Le réseau linéaire étant dépourvu d'une fonction d'activation, le traitement est donc décrit de manière plutôt simple: Chaque neurone calcule la somme pondérée des entrées qu'il reçoit et produit une valeur de sortie, à la suite de cela, l'erreur quadratique moyenne entre les valeurs calculées et les valeurs cibles est minimisée simplement en ajustant les poids. Autre que l'identité de la régression linéaire, cette méthode fut aussi connue sous l'appellation de la méthode des moindres carrés. Elle fut un moyen pour Adrien-Marie Legendre (1805) et Carl Friedrich Gauss (1795) de trouver un bon ajustement linéaire approximatif à un ensemble de points, dans leurs travaux sur la prédiction du mouvement planétaire. ⁷ ⁸

C'est en 1943, qu'un document de recherche publié par Warren McCulloch, un neurophysiologiste, et Walter Pitts un logicien, propose un modèle de neurones artificiels en démontrant comment ces derniers peuvent être utilisés pour effectuer des opérations logiques simples; le document intitulé

"A Logical Calculus of the Ideas Immanent in Nervous Activity" (Un calcul logique des idées immanentes dans l'activité nerveuse) prônait l'idée d'un modèle de calcul **sans apprentissage** pour les réseaux de neurones.⁶

En 1958, un psychologue nommé Frank Rosenblatt développa un type de réseaux de neurones artificiels appelé le perceptron qui était capable de reconnaître de simples patrons dans les données d'entrée.⁹ Certains disputent cette invention en l'attribuant à McCulloch et Walter Pitts, mais il n'y a aucun doute que la première implémentation fut effectuée par Rosenblatt en 1971, une machine qu'il a construite dans le laboratoire aéronautique de Cornell à New York.⁹ Le perceptron est un algorithme d'apprentissage supervisé de classificateurs binaires. L'algorithme du perceptron est aussi appelé **perceptron monocouche** pour aider la distinction avec le **perceptron multicouche**.

Quelques années plus tard, deux chercheurs, Marvin Minsky et Seymour Papert publièrent un livre nommée "Perceptron" dans lequel ils relatent de leur découverte sur les limitations du perceptron monocouche;¹⁰ Ils découvrent que le perceptron monocouche était incapable des problèmes non-linéaires comme le OU exclusif et ils notent que les machines à l'époque n'étaient pas assez puissantes pour traiter des réseaux de neurones pratiques.

Cette limitation fut supprimée grâce à l'application de la rétropropagation du gradient de l'erreur sur les perceptrons multicouches par Paul Werbos en 1974;¹¹ Cette méthode de rétropropagation a été mise au point douze années plus tard, en 1986, par David Rumelhart¹². Elle permet l'entraînement d'un réseau de neurones en ajustant les poids de chaque neurone de la dernière couche vers la première. Cette découverte a revitalisé l'intérêt des réseaux de neurones après qu'il fut perdu suite à la découverte de Minsky et Papert.

Le perceptron multicouche a joué un rôle crucial dans le développement de l'apprentissage profond, car il permettait trois ou plusieurs couches dans le réseau, d'où l'adjectif "profond". Il est composé d'au moins trois couches, une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie (voir **Fig 5**)

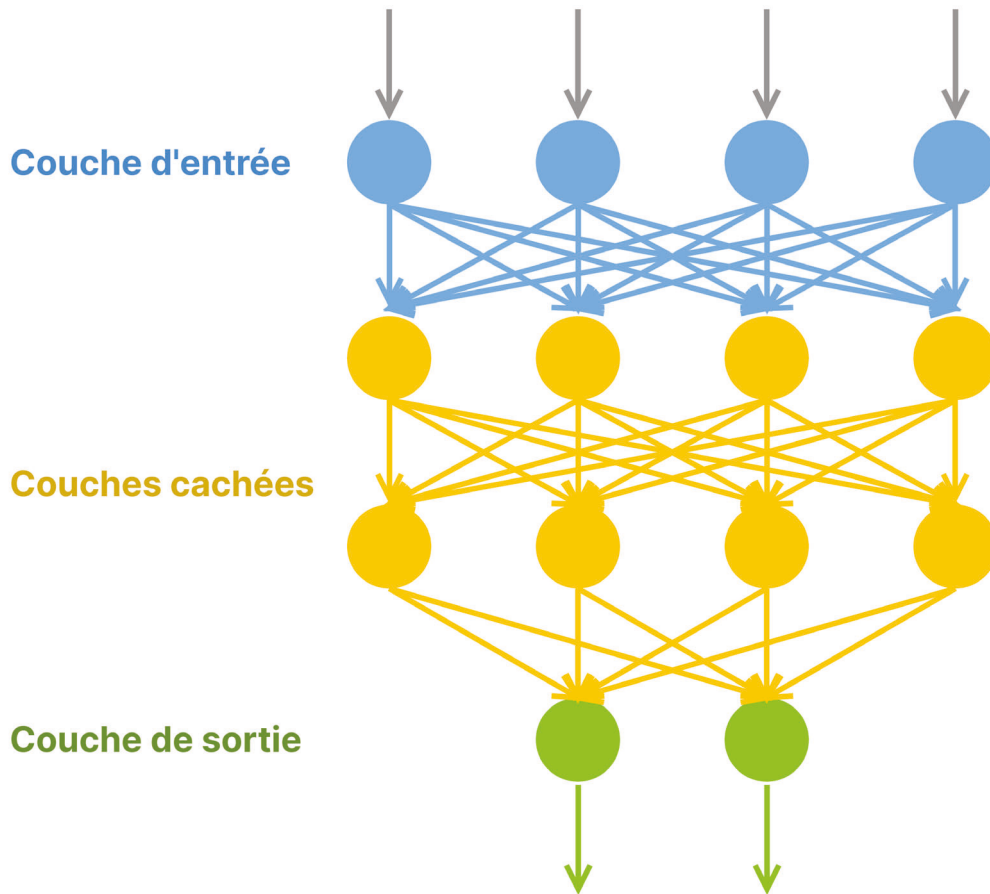


Fig 5 - Perceptron Multicouche (MLP)

Aujourd'hui les réseaux de neurones artificiels sont les composants clés dans plusieurs systèmes d'intelligence artificielles et l'usage se note dans nombreuses applications telles que les voitures autonomes, les diagnostics médicaux, et même dans les prévisions financières car ils permettent même la résolution de problèmes stochastiques. Le domaine continue d'évoluer par la découverte de nouvelles techniques et d'algorithmes jusqu'à présent.¹³

2.4. Les réseaux de neurones convolutifs

Le réseau neuronal convolutif est un algorithme d'apprentissage profond qui est largement utilisé pour analyser les données visuelles. Le réseau est spécialement conçu pour traiter les données en forme de grilles (matrices) telles que des images ou des vidéos, en tirant avantage d'une structure spéciale qui incorpore une ou plusieurs couches de convolutions. Dans ces dernières, le réseau

neuronal de convolution utilise l'opération mathématique du produit de convolution au lieu d'une multiplication matricielle générale.

Le réseau neuronal convolutif s'inspire du système visuel humain; dans ce dernier, les différentes couches de neurones sont responsables de la détection et du traitement des caractéristiques visuelles à différents niveaux de complexités. Les niveaux les plus élevés de neurones (les plus proches de la sortie du réseau) détectent les caractéristiques complexes telles que les contours et la texture, et les niveaux les plus profonds répondent aux objets et formes moins complexes.

Avant d'entrer dans le vif du sujet, nous parlerons, dans un premier temps, des notions mathématiques pré-requises, spécialement le produit de convolution qui représente l'opération mathématique majeure derrière l'algorithme du réseau neuronal convolutif, puis nous élaborerons davantage sur l'architecture des réseaux neuronaux convolutifs et finalement nous énumérerons les domaines communs où ils s'appliquent.

2.4.1. Notions mathématiques pré-requises

2.4.1.1. Le produit de convolution

Définition

Le produit de convolution est une opération mathématique effectuée sur deux fonctions f et g , qui sont définies sur un même domaine infini, pour produire une troisième fonction $f * g$ qui représente comment la forme d'une des fonctions est modifiée par l'autre. Il s'agit de l'intégrale du produit des deux fonctions après qu'une d'elles soit reflétée par rapport à l'axe des ordonnées et décalée. Grâce à sa caractéristique de commutativité, le choix de la fonction à refléter et à décaler ne change en aucun cas le résultat de l'intégrale. Cette dernière est évaluée pour toutes les valeurs de décalage, produisant ainsi la fonction de convolution, dont la formule est la suivante :

$$(f \star g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau .$$

L'usage du symbole t peut causer une confusion, t ne représente pas le temps, mais plutôt le montant du décalage. Le produit de convolution est décrit comme la surface en dessous de la fonction $f(\tau)$ pondérée par la fonction $g(-\tau)$ décalée d'un montant t . Au fur et à mesure que t change, la fonction de pondération $g(t - \tau)$ souligne les différentes parties de la fonction d'entrée $f(\tau)$; si t est positif, $g(t - \tau)$ équivaut à $g(-\tau)$ qui est décalée vers la droite sur l'axe τ , c'est à dire vers $+\infty$, d'un montant t , par contre si t est négatif, la fonction est décalée vers la gauche, c'est à dire vers $-\infty$ d'un montant $|t|$.

La commutativité du produit de convolution permet aussi cette écriture (voir Fig 6) :

$$(f \star g)(t) = \int_{-\infty}^{\infty} f(t - \tau)g(\tau) d\tau .$$

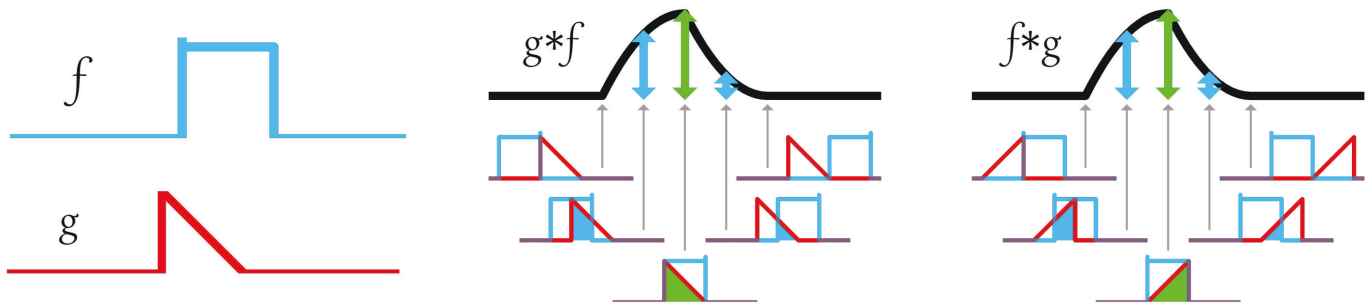


Fig 6 - Produit de convolution

Le produit de convolution bidimensionnel

Le produit de convolution bidimensionnel est simplement une extension du produit de convolution précédemment mentionné, pour opérer sur un domaine spatial bidimensionnelle, cela veut dire que le produit est effectué sur deux axes, l'horizontal et le vertical. Cette opération est conduite entre deux matrices et se résume par l'application d'un filtre, plus communément appelé noyau, sur une donnée d'entrée bidimensionnelle et sa formule est la suivante :

$$(f \star g)[m, n] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[i, j] \cdot g[m - i, n - j] ,$$

où :

- f et g sont deux matrices représentant respectivement un signal d'entrée et le noyau (filtre).
- $[m, n]$ représente la position dans la matrice résultante (signal de sortie), avec m représentant la colonne et n la ligne.
- i et j représente les indices des éléments des matrices opérandes (f et g).

Exemple :

Soient deux matrices f et g , f étant la matrice de données d'entrée et g le noyau :

$$f = \begin{bmatrix} 4 & 3 & 1 \\ 6 & 1 & 7 \\ 8 & 9 & 2 \end{bmatrix} \quad g = \begin{bmatrix} -1 & 3 & -2 \\ 0 & 0 & 1 \\ 1 & 2 & 1 \end{bmatrix}.$$

L'origine du noyau $g[0, 0]$ est le centre du noyau correspondant à 0, ce qui translate le premier élément du noyau à $g[-1, -1]$ qui correspond à -1 ; la matrice d'entrée par contre a comme premier élément $f[0, 0]$ qui correspond à 4. Cette translation est due au **rembourrage** effectué sur la matrice d'entrée, on considère des éléments additionnels, des zéros, pour avoisiner les éléments situés aux extrémités de la matrice

Dans un contexte visuel, le noyau est renversé horizontalement et verticalement.

Pour calculer les éléments de la matrice de sortie, on superpose le centre du noyau $g[0, 0]$ avec chaque élément de la matrice d'entrée f ; en projetant les éléments du noyau g sur f , on multiplie les correspondances ensuite on les additionne; les éléments du noyau qui ne correspondent à aucun élément de la matrice d'entrée sont multipliés par 0 (éléments remboursés).

À titre d'exemple, voici comment se calcule les deux premiers éléments (illustrés respectivement dans les figures 7 et 8) de la matrice de sortie :



Fig 7 - Calcul du premier élément de la matrice de sortie

$$\begin{aligned}
 (f \star g)[0,0] &= 1 \cdot 0 + 2 \cdot 0 + 1 \cdot 0 \\
 &+ 1 \cdot 0 + 0 \cdot 4 + 0 \cdot 3 \\
 &+ (-2) \cdot 0 + 3 \cdot 6 + -1 \cdot 1
 \end{aligned}$$

$$(f \star g)[0,0] = \boxed{17}.$$

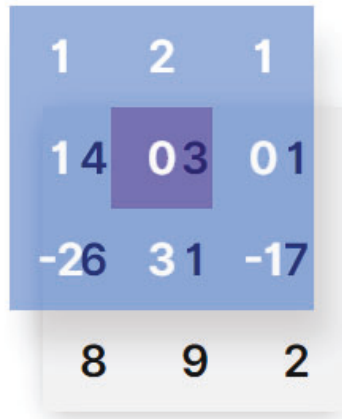


Fig 8 - Calcul du second élément de la matrice de sortie

$$\begin{aligned}
 (f \star g)[1,0] &= 1 \cdot 0 + 2 \cdot 0 + 1 \cdot 0 \\
 &+ 1 \cdot 4 + 0 \cdot 3 + 0 \cdot 1 \\
 &+ (-2) \cdot 6 + 3 \cdot 1 + -1 \cdot 7
 \end{aligned}$$

$$(f \star g)[1,0] = \boxed{-11}.$$

et ainsi de suite se calcule le reste des éléments de la matrice de sortie.

Il est possible d'effectuer le produit de convolution pour obtenir une matrice de sortie à une dimension inférieure à celle de la matrice d'entrée, il suffit de sauter l'étape du rembourrage et considérer le premier élément du noyau comme son origine.

2.4.1.2. Le produit scalaire de Frobenius

Le produit scalaire de Frobenius est une opération binaire effectuée sur deux matrices pour retourner un scalaire. Cette opération est le produit scalaire de deux matrices comme si elles étaient des vecteurs tant qu'elles satisfont les conditions des opérands d'un produit scalaire.

Les deux matrices doivent être de mêmes dimensions, elles ne sont pas nécessairement carrées; soient A et B deux matrices complexes de mêmes dimensions, leur produit scalaire de Frobenius se note $\langle A, B \rangle_F$ et la formule du produit est la suivante :

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1m} \\ A_{21} & A_{22} & \dots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nm} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1m} \\ B_{21} & B_{22} & \dots & B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ B_{n1} & B_{n2} & \dots & B_{nm} \end{bmatrix}$$

$$\langle A, B \rangle_F = \sum_{i=1}^m \sum_{j=1}^n \overline{A}_{i,j} \overline{B}_{i,j},$$

où \overline{A} et \overline{B} représentent les conjugués de A et B respectivement.

2.4.1.3. La fonction Unité Linéaire Rectifiée

La fonction Unité Linéaire Rectifiée (ReLU), aussi connue sous l'identité du redresseur dans le contexte des réseaux de neurones artificiels, est une fonction d'activation, c'est-à-dire la fonction qui définit la sortie d'un neurone en prenant les entrées de ce dernier en paramètre. L'utilisation d'une fonction d'activation introduit la non-linéarité au résultat et cela permet au réseau de modéliser des relations complexes entre les données.

La fonction ReLu est toute simple, elle retourne son paramètre si ce dernier est positif, ou 0 s'il est négatif :

$$f(x) = x^+ = \max(0, x) = \frac{x + |x|}{2} = \begin{cases} x & \text{si } x > 0, \\ 0 & \text{sinon,} \end{cases}$$

$$f'(x) = \begin{cases} 1 & \text{si } x > 0, \\ 0 & \text{si } x < 0. \end{cases}$$

2.4.2. Structure

Un réseau neuronal convolutif se compose d'une couche d'entrée, de plusieurs couches cachées, en d'autres termes des couches intermédiaires, et finalement d'une couche de sortie. Subséquemment, nous allons faire passer sous la loupe chacune de ces couches intermédiaires.

2.4.2.1. Couche de convolution

Le réseau neuronal convolutif comprend obligatoirement une ou plusieurs couches dans lesquelles s'effectue le produit de convolution, celles-ci se nomment convenablement couches de convolutions. Généralement, au moins une de ces couches effectue le produit scalaire entre le noyau de convolution et la matrice d'entrée de la mentionnée couche; ce produit scalaire est, pour la majorité des fois, le **produit scalaire de Frobenius**, et la fonction d'activation de la couche est la fonction **ReLU**. Toute matrice résultante d'une couche de convolution est appelée une carte de caractéristique (feature map en Anglais) et cette dernière devient la donnée d'entrée de la prochaine couche du réseau.

Le motif de connexions entre les neurones est aussi inspiré du cortex visuel des animaux. Chaque neurone cortical répond impulsivement à une zone particulière du champ de vision. Par analogie, chaque neurone dans une couche de convolution traite un fragment de la donnée d'entrée qu'on appelle champ récepteur.

Il est vrai qu'un perceptron multicouche pourrait apprendre des caractéristiques et classifier les données d'entrée, mais son architecture n'est guère convenable pour des données de larges dimensions telles que les images; dans un perceptron multicouche, chaque neurone d'une couche intermédiaire est connecté à tous les neurones de la couche précédente. Par conséquent, pour pouvoir traiter une image d'une résolution de 100 (largeur) \times 100 (hauteur), soient 10000 pixels, chaque neurone de la seconde couche du réseau aura 10000 connections (poids), et donc s'il y a 100 neurones dans la couche, cette dernière présentera 10000×100 poids, soient un million de poids, un nombre trop élevé. Par contre, dans un réseau de neurones convolutifs, la taille du champ récepteur est déterminée par la dimension du noyau, cela veut dire que si nous tenons un noyau d'une dimension 5×5 , soient 25 éléments, chaque neurone traitera 25 pixels et donc nous aurons besoin que de 40 neurones, au lieu d'un million de neurones dans notre exemple du perception.

Ce partage de poids (noyau) permet au réseau neuronal convolutif d'exploiter la structure spatiale et détecter les patrons locaux peu importe leurs emplacements dans l'image; cette propriété est appelée invariance par translation ou Shift-invariance en Anglais.

2.4.2.2. Couche de pooling

La couche de pooling est employée pour réduire les dimensions spatiales (hauteur et largeur) des cartes de caractéristiques résultant de la couche de convolution qui la précède. Cette procédure réduit la quantité de paramètres et de calcul dans le réseau; en effet, il est plus facile d'opérer sur une image d'une résolution moyenne qu'une de haute définition. Comme vous pouvez le déduire de la mention des cartes de caractéristiques, les couches de pooling sont installées entre les couches de convolutions successives.

L'objectif principal de la couche de pooling est d'extraire les caractéristiques les plus importantes des cartes de caractéristiques tout en égarant les moins significatives, ce qui permet la robustesse du réseau face aux variations dans les données d'entrées et la prévention du surapprentissage.

L'image d'entrée est divisée en rectangles (tuiles) de n côtés de sortes que ces derniers ne se chevauchent pas.

Il y a deux types de couches de pooling employés dans un réseau de neurones convolutifs :

- **Max pooling** : Il prend une région rectangulaire de la carte, communément une fenêtre de dimensions (2×2) ou (3×3) , et retourne la plus grande valeur de cette mentionnée région, puis nous avançons d'un pas équivalent à la dimension de la région (pour éviter le chevauchement), et nous effectuons la même opération sur la prochaine région; n'est retenue que la caractéristique la plus dominante (voir Fig 9).

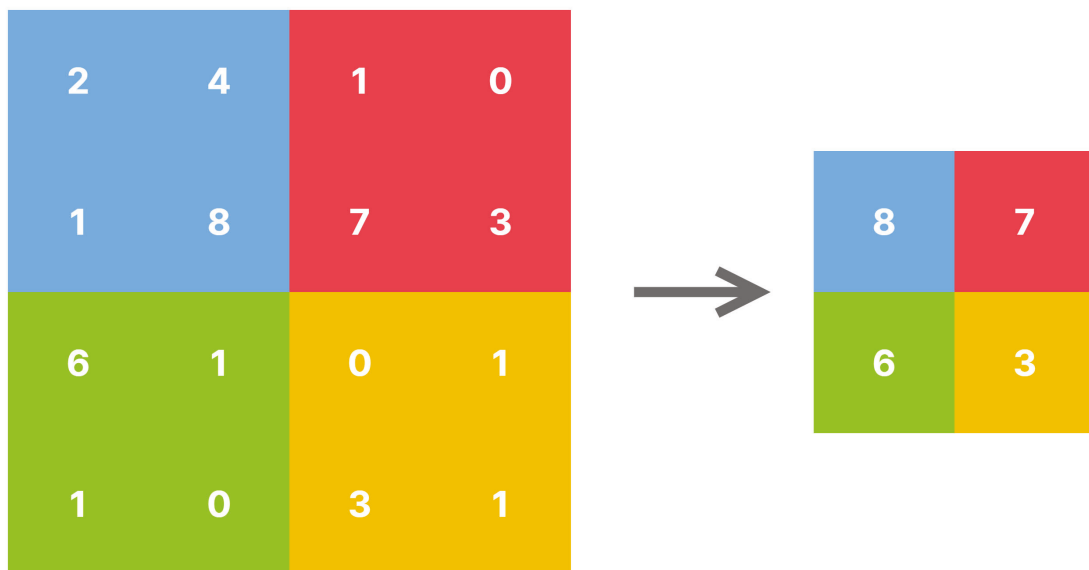


Fig 9 - Max pooling avec une fenêtre 2x2 et pas = 2

- **Average pooling** : Semblable au max pooling, sauf que la valeur moyenne est retournée au lieu de la valeur maximale. Il est généralement moins préférable d'utiliser ce type de pooling car il pourrait rendre floue l'image et perdre les caractéristiques les plus importantes. Cela dit, il s'avère être utile lorsque l'objectif est de capturer l'intensité générale des caractéristiques de la carte.

La portée de ces opérations se définit en deux catégories :

- le **pooling local** : L'opération est effectuée sur une région de la carte, ensuite la fenêtre de pooling est glissée d'un **pas** défini vers la prochaine région de la carte, et ainsi de suite.
- le **pooling global** : L'opération est effectuée sur toute la carte, c'est-à-dire que la dimension de la fenêtre est égale à celle de la carte, et donc il en résulte une seule valeur pour chaque carte.

2.4.2.3. Couche de correction

Cette couche est optionnelle mais très recommandée, elle permet une meilleure efficacité en introduisant la non-linéarité dans les signaux de sortie grâce à une fonction d'activation. Les suivantes sont les fonctions qui peuvent être utilisées :

- La fonction Unité Linéaire Rectifiée (ReLU) : $f(x) = \max(0, x)$,
- La tangente hyperbolique : $f(x) = \tanh(x)$,
- La tangente hyperbolique saturante : $f(x) = |\tanh(x)|$,
- La fonction sigmoïde : $f(x) = (1 + e^{-x})^{-1}$.

La fonction préférée est usuellement la fonction **Unité Linéaire Rectifiée (ReLU)** car elle est plus performante en offrant efficacité sans compromettre la précision.¹⁴

2.4.2.4. Couche entièrement connectée

La couche entièrement connectée, comme mentionnée précédemment, est une couche dans laquelle chaque neurone est connecté à tous les neurones de la couche précédente; il s'agit de la même couche qu'on observe dans un perceptron multicouche. Elle agit à cet emplacement comme classificateur et donne la prédiction finale selon les caractéristiques apprises. Elle prend en paramètre les cartes de caractéristiques aplaties résultantes de la couche précédente; la structure de cette couche lui permet d'effectuer un raisonnement de haut niveau, c'est-à-dire la reconnaissance de caractéristiques visuelles moins complexes, telles que visages, animaux, véhicules, etc.

2.4.2.5. La couche de perte

La couche de perte quantifie l'erreur de prédiction, c'est-à-dire la différence entre les résultats attendus et la prédiction du réseau grâce à une fonction objectif que nous avons mentionnée dans le chapitre précédent. Plusieurs fonctions objectif peuvent être utilisées selon la tâche spécifique, notamment :

- Pour une classification en classes multiples : la fonction **softmax** est préférée; elle mesure la dissimilarité entre la distribution de probabilités prédite et les véritables étiquettes encodées en **vecteurs binaires unitaires (one-hot encoded)** :

- La fonction **softmax** prend en paramètre un vecteur $z = (z_1, \dots, z_K)$ de K nombres réels et le normalise en une distribution de probabilité $\sigma(z)$ sur K choix :

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

et

$$\sigma(z) = (\sigma(z_1), \dots, \sigma(z_k)) .$$

Les éléments de $\sigma(z)$ sont compris entre 0 et 1 et la somme des éléments est égale à 1.

- La représentation en **vecteurs binaires unitaires (one-hot encoded)** est une technique de représentation utilisée en apprentissage automatique et en analyse de données pour convertir les variables catégorielles en vecteurs binaires; supposons avoir **3** catégories, pour chaque catégorie nous aurons un vecteur de **3** éléments dont un seul est non-nul et est égal à 1. Par exemple, pour trancher entre *animal*, *homme*, *voiture* et *plante*, nous aurions :

$$\text{animal} = [1,0,0,0] ,$$

$$\text{homme} = [0,1,0,0] ,$$

$$\text{voiture} = [0,0,1,0] ,$$

$$\text{plante} = [0,0,0,1] .$$

Toujours à titre illustratif, quand la fonction softmax retournera un vecteur $\sigma(z) = [0.15, 0.04, 0.8, 0.01]$, nous saurons qu'il s'agit d'une *voiture* grâce à la **position** de la plus grande probabilité, soit 0.8, dans le vecteur.

- Pour les tâches de régression où la sortie est une valeur continue : la **fonction d'erreur quadratique moyenne** est usuelle :

$$EQM = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 ,$$

avec $Y = (Y_1, \dots, Y_K)$ est le vecteur des valeurs observées et $\hat{Y} = (\hat{Y}_1, \dots, \hat{Y}_K)$ celui des valeurs prédites.

Après avoir calculé la perte dans cette dernière couche du réseau, l'algorithme de rétropropagation propage cette perte à rebours, c'est-à-dire de la dernière à la première couche, en ajustant les poids dans le but de minimiser l'erreur de prédiction et d'optimiser le réseau pour qu'il traite mieux les données nouvelles (généralisation).

2.4.3. Quelques domaines d'applications

Le réseau neuronal convolutif est conçu spécifiquement pour traiter les données multidimensionnelles telles que les pixels. Grâce à sa capacité d'apprendre les caractéristiques hiérarchiques d'une image, il se voit naturellement utilisé dans bon nombre de domaines dont :

- La **classification d'images** : Voici l'exercice ciblé dans la conception du réseau neuronal convolutif. L'objectif principal est de catégoriser l'image selon une liste de classes ou étiquettes prédéfinies. Cela englobe des tâches telles que la reconnaissance d'objets dans une image, identification d'un animal, etc.
- La **détection d'objets** : En plus de classifier l'image, on localise un objet en particulier en dessinant un cadre le délimitant.
- La **segmentation d'image** : L'objectif est d'étiqueter chaque pixel dans une image, la divisant efficacement en plusieurs segments (voir Fig 10).

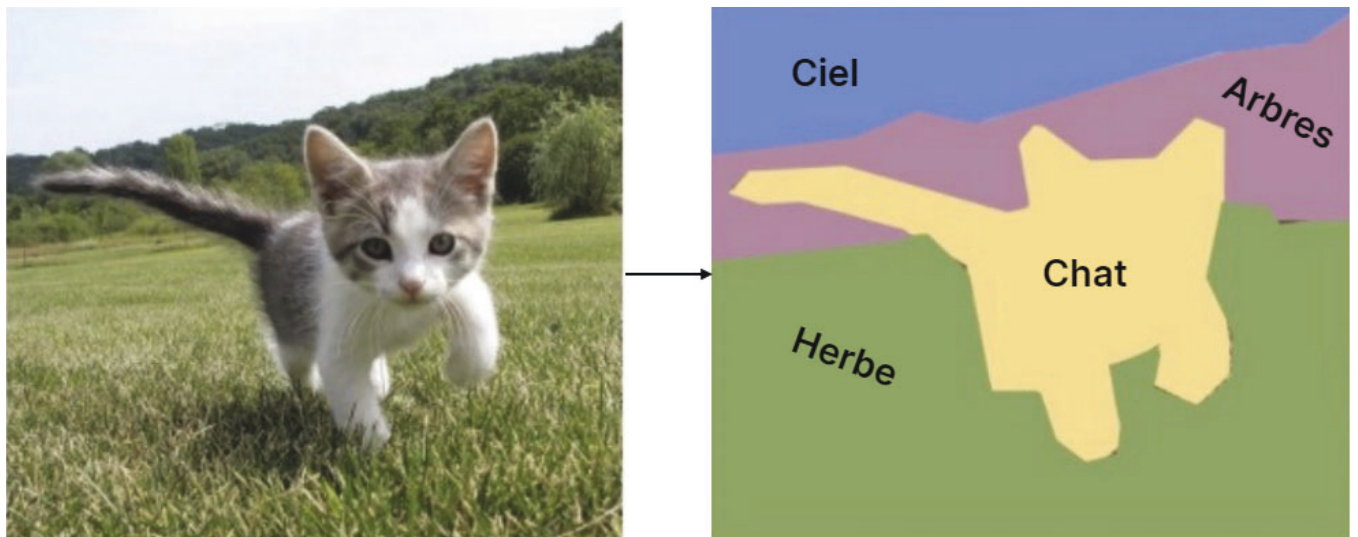


Fig 10 - Segmentation d'image

- La **génération d'image** : Un exemple de cette tâche serait la reproduction d'images de très haute définition à partir de leurs versions en faible qualité.
- **Traitement automatique de la langue naturelle (TALN)** : Bien qu'il soit principalement conçu pour traiter les images, le réseau neuronal convolutif peut être combiné avec le réseau de neurones récurrents pour traiter des données séquentielles dans les tâches de TALN telles que la classification de textes et l'analyse de sentiments ¹⁵.

Les applications du réseau neuronal convolutif ne sont point limitées à celles que nous venons de mentionner, mais la totalité ou la majorité implique une de ces tâches.

2.5. Exploit majeur réalisé : ChatGPT, le début d'une nouvelle ère

2.5.1. Introduction

ChatGPT, comme le programme se définit lui-même (voir **Fig 11**), est un modèle de langage développé par **OpenAI** en novembre 2022, basé sur l'architecture GPT-3.5. Son objectif principal est de comprendre et de générer du texte en fonction des informations qu'il a apprises pendant son entraînement. Il peut répondre à une variété de questions, fournir des explications, raconter des histoires, aider avec des problèmes techniques et bien plus encore.

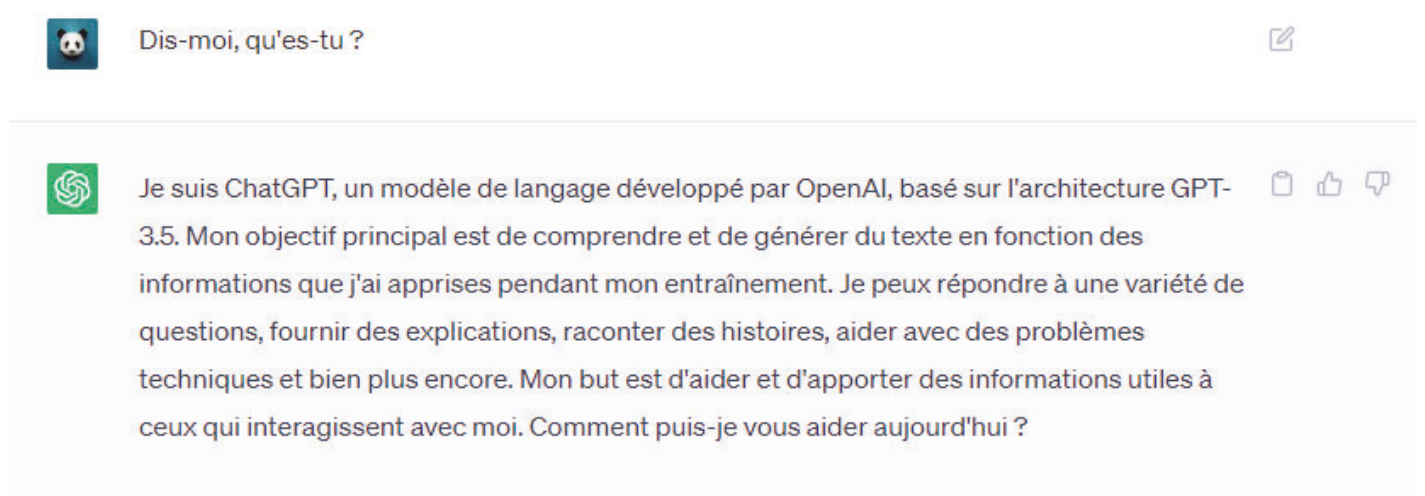


Fig 11 - ChatGPT se définissant lui même

En de termes plus avancés, ChatGPT est une extrapolation du grand modèle linguistique (Large Language Model ou **LLM** en anglais). Le grand modèle linguistique est un modèle résultant d'un apprentissage profond (Deep Learning) effectué par un réseau de neurones artificiels comptant un

nombre immense de paramètres, c'est-à-dire que les neurones d'une seule couche peuvent compter jusqu'à des dizaines de millions de poids entre eux. L'entraînement est effectué sur plusieurs processeurs graphiques (GPU) en un court laps de temps dû au traitement en parallèle d'une quantité énorme de textes non étiquetés composés de milliards de jetons (mots ou parts de mots). Grâce à cet entraînement rigoureux, le grand modèle linguistique est capable d'effectuer divers tâches de **traitement automatique de la langue naturelle** (TALN) notamment :

- La **Compréhension de la langue** : Le modèle est capable de lire et de comprendre un texte, répondre aux questions contextuelles, résumer le contenu et d'en extraire les informations pertinentes (voir **Fig 12**).
- La **Génération de la langue** : le modèle peut générer un texte à croire que ce dernier fut écrit par un homme et pas une machine; par exemple rédiger des histoires, des poèmes, des dissertations, etc (voir **Fig 13**).
- La **Traduction** : le modèle peut aussi traduire le texte d'une langue à une autre, en respectant les règles grammaticales de la langue cible tout en conservant le contexte.
- **L'analyse du sentiment** : Le modèle est capable de déterminer le sentiment ou le ton émotionnel du texte.
- Les **Jeux de langues** : Le modèle pourrait être entraîné pour jouer parfaitement aux jeux de textes tels que "20 questions", charades ou autres.

GPT signifie "Generative Pre-Trained Transformer" en anglais, cela se traduit par **Transformateur Pré-entraîné Génératif** et le titre se dissèque ainsi :

- **Pré-entraîné** : Le modèle est entraîné d'avance sur un océan de données recueillies de l'internet jusqu'à une date donnée (septembre 2021 pour GPT 3.5 par exemple), ce qui lui a permis d'apprendre les patrons textuelles, les grammaires, les relations contextuelles de divers langues, etc. Cet entraînement lui permet un très solide fondement pour comprendre et générer du texte concernant divers sujets (voir **Fig 14**).
- **Génératif** : indique la capacité du modèle à générer du texte cohérent et approprié à un contexte bien défini.
- **Transformateur** : Il s'agit d'une architecture basé sur le concept de l'auto-attention qui permet au modèle de peser l'importance des différents mots composant la donnée d'entrée, l'attention est dirigée vers les mots les plus pertinents du texte, et les mots impertinents sont ignorés; ainsi sont gérés les cas de dépendances à longues distances, c'est-à-dire la relation entre des

éléments (mots) séparés par un bon nombre d'autres éléments; par exemple dans la phrase " *l'UQTR*, l'université dans laquelle je me suis inscrit, est très *accueillante* ", les mots " *UQTR* " et " *accueillante* " présentent une relation entre eux, mais sont séparés par d'autres mots.



c'est quoi ce truc super long qu'on te demande d'écrire à la fin de tes études à l'université ?



Il semblerait que vous parliez du mémoire de fin d'études universitaires, également appelé "thèse" dans certains pays. Le mémoire de fin d'études est un document écrit de recherche approfondie qui doit être présenté à la fin d'un programme universitaire, généralement au niveau du master ou du doctorat.



Fig 12 - ChatGPT lisant entre les lignes



raconte-moi une très courte blague sur le mémoire de fin d'études



Bien sûr! Voici une blague courte sur le mémoire de fin d'études :



Pourquoi le mémoire de fin d'études est-il comme une montagne russe ?

Parce qu'il y a des hauts, des bas, beaucoup de cris, mais à la fin, on se sent soulagé d'être descendu de là !

Fig 13 - ChatGPT racontant une blague



Si j'écris une page par jour, mais chaque semaine, je me repose une journée, combien de pages écrirais-je entre le premier 1 juillet au 28 septembre ?



Pour calculer le nombre de pages que vous écrirez entre le 1er juillet et le 28 septembre en écrivant une page par jour, mais en prenant une journée de repos chaque semaine, nous devons diviser cette période en semaines et jours.

Premièrement, déterminons combien de jours il y a entre le 1er juillet et le 28 septembre :

Juillet a 31 jours.

Août a 31 jours.

Septembre a 28 jours (jusqu'au 28 septembre).

Total des jours entre le 1er juillet et le 28 septembre :

31 (jours en juillet) + 31 (jours en août) + 28 (jours en septembre) = 90 jours

Deuxièmement, déterminons combien de semaines complètes il y a dans cette période :

90 jours / 7 jours (une semaine) = 12 semaines et 6 jours

Cela signifie qu'il y a 12 semaines complètes, avec 6 jours supplémentaires.

Ensuite, calculons le nombre total de jours de repos que vous prendriez :

12 semaines * 1 jour de repos par semaine = 12 jours de repos

Maintenant, soustrayons ces jours de repos du total de jours d'écriture :

Total de jours d'écriture = 90 jours (total) - 12 jours (repos) = 78 jours d'écriture

Comme vous écrivez une page par jour, le nombre de pages que vous écrirez entre le 1er juillet et le 28 septembre sera de :

78 pages.



Fig 14 - ChatGPT raisonnant une opération mathématique

ChatGPT est tellement intelligent qu'il est devenu aujourd'hui un outil indispensable pour l'internaute lambda. En effet, il a une utilité particulière pour presque tous les groupes démographiques : les étudiants et les instructeurs l'utilisent pour chercher des explications claires sur différents sujets académiques, les créateurs de contenus en font usage pour générer des idées originales et créatives, les développeurs l'utilisent pour générer du code (templates et fonctions communes) et trouver des solutions à même de complexes problèmes, d'autres en font usage pour planifier leur emploi du temps et pour faire une tempête d'idée, etc.

Le programme a été mis au service du public le 30 novembre 2022; deux mois plus tard, il était devenu l'application détenant le record de la croissance la plus rapide du nombre de consommateurs, gagnant plus de 100 millions d'utilisateurs et contribuant à la montée en valeur de OpenAI qui valait désormais 29 milliards de dollars américains.

Comme vous pouvez certainement vous en douter, un programme capable de tels exploits doit sûrement présenter de flagrants risques dans plusieurs domaines. En effet, un nombre de questions se posent sur son sujet: *Qu'en sera-t-il du secteur éducatif ? Est-ce que la majorité des travaux se verront effectués par des machines plutôt que des Hommes ? Est-ce que ChatGPT aura une compétition, à quoi serviront les moteurs de recherches maintenant ?*

Premièrement nous parlerons des actions prises par les géants des moteurs de recherches en ligne tels que Google et Microsoft pour remédier à leurs positions défavorables à l'émergence d'une telle technologie, et en second lieu et dernier lieu, nous discuterons des critiques dirigés vers ChatGPT, et des regards portés sur l'impact de l'intelligence artificielle sur l'avenir des secteurs majeurs de la société, notamment le secteur éducatif et le secteur professionnel.

2.5.2. La guerre contre les moteurs de recherches

Il n'y a pas plus d'une année, le moyen préféré des internautes d'obtenir des réponses à leurs questions était de le rechercher sur Google, ou pour une minorité, Bing, Yahoo, Yandex, et bien d'autres moteurs de recherches. Aujourd'hui avec l'entrée en scène de ChatGPT, beaucoup de ces moteurs de recherches perdent de bons nombres d'utilisateurs chaque jour. En effet, dans plusieurs aspects, ChatGPT offre une meilleure alternative à ces derniers; Étant capable, en quelques sortes, de raisonner, il retourne une et une seule réponse unique et précise à chaque utilisateur, au lieu de compiler une liste de sites web pouvant contenir les mots clés composant la requête de l'utilisateur, comme le ferait un moteur de recherche. Pour une meilleur illustration de cette comparaison, nous

avons demandé à Google (voir Fig 15) et à ChatGPT (voir Fig 16) la question informatique suivante : « Avec HTML, comment garder deux éléments sur la même ligne ? » Voici les réponses retournées :



The image shows a Google search interface with the query 'avec html comment garder deux éléments sur la même' in the search bar. Below the search bar, three search results are displayed. The first result is from Mozilla Developer, titled 'Le positionnement - Apprendre le développement web | MDN', dated 21 July 2023. The second result is from Alsacrations, titled '[Résolu]Garder 3 blocs <div> sur la même ligne - Alsacreations', dated 2 September 2005. The third result is from Developpez.net, titled 'Sur la même ligne mettre du texte à gauche et à droite', dated 26 January 2005.

Google

avec html comment garder deux éléments sur la même X

Mozilla Developer
https://developer.mozilla.org › CSS_layout › Positioning

Le positionnement - Apprendre le développement web | MDN

21 juill. 2023 — Le positionnement permet de sortir les **éléments** du flux normal de la composition du document, **et** de les faire se comporter différemment, ...

https://developer.mozilla.org › ... › CSS Flow Layout

Disposition de bloc et en ligne avec le flux normal - CSS

20 juill. 2023 — Dans ce guide, nous verrons le comportement des **éléments** de bloc **et** des **éléments** en **ligne** lorsqu'ils sont placés dans le flux normal.

Alsacrations
https://forum.alsacreations.com › topic-4-6774-1-Reso...

[Résolu]Garder 3 blocs <div> sur la même ligne - Alsacreations

2 sept. 2005 · 15 réponses · 8 auteurs

salut. Il te suffit de laisser float:left pour tes trois blocs. Ils s'aligneront ainsi les uns à la suite des autres.

Developpez.net
https://www.developpez.net › forums › mise-page-css

Sur la même ligne mettre du texte à gauche et à droite

26 janv. 2005 — Metre sur la **même ligne** du texte aligné à gauche, **et** une autre partie à droite. Il faudrait un resultat du style ...

Fig 15 - Google répond à une question informatique (extrait)



Pour garder deux éléments sur the même ligne en HTML, vous pouvez utiliser les propriétés CSS appropriées. Voici quelques approches courantes pour réaliser cela :

1. Utilisation de l'élément `` : Vous pouvez utiliser l'élément `` pour encadrer les deux éléments que vous souhaitez garder sur la même ligne. L'élément `` est par défaut un élément en ligne, ce qui signifie qu'il ne force pas un saut de ligne après lui.

Exemple :

```
html Copy code  
  
<p>  
  Ceci est un paragraphe avec du texte <span>et un élément supplémentaire</span>  
</p>
```

1. Utilisation de propriétés CSS :

Vous pouvez utiliser les propriétés CSS `display: inline;` ou `display: inline-block;` pour les éléments que vous souhaitez garder sur la même ligne. Voici un exemple avec des divs :

```
html Copy code  
  
<div style="display: inline;">Élément 1</div>  
<div style="display: inline;">Élément 2</div>
```

Dans cet exemple, les divs contenant "Élément 1" et "Élément 2" seront affichés sur la même ligne

Fig 16 - ChatGPT répond à une question informatique (extrait)

Au lieu de compiler une liste de liens qui pourraient être pertinents à ma question comme l'a fait Google, ChatGPT répond directement en donnant plusieurs moyens d'effectuer la tâche demandée, et pour chaque méthode il donne un exemple d'un code embelli en expliquant étape par étape le fonctionnement de ce dernier.

Avec sa facilité d'utilisation, sa compréhension des nuances, et ses réponses du style conversationnel, ChatGPT pose une sérieuse menace à ces géants de moteurs de recherche, par conséquent, des réactions s'imposent. Le premier à répondre est Microsoft; grâce à son investissement sur OpenAI, Microsoft intègre la dernière version de GPT dans leur moteur de recherche Bing, GPT-4. Avec 1.7 billion de paramètres, le modèle GPT-4 est largement plus capable que son prédécesseur utilisé dans la version gratuite de ChatGPT. En mettant le web à la disposition d'un tel modèle, Bing devient le parfait compagnon de l'internaute. Google, au lieu d'acheter la compétition comme le fait Microsoft, crée une alternative à ChatGPT nommé Bard. Bard est tout

récent, il voit le jour le 21 Mars 2023, et comme on pourrait le soupçonner, il n'est pas totalement au point et ne rivalise pas encore avec ChatGPT et Bing; d'après ses bêta-testeurs, ses réponses sont pour la plupart du temps inexactes. Bard est accessible dans 238 pays mais le Canada ne figure pour l'instant pas parmi ceux-là. Cela dit, on peut s'attendre à une très féroce compétition dans les années à venir.

2.5.3. Critiques et regards sur l'avenir de la société

Comme on dit souvent, rien est parfait, et ChatGPT ne l'est sûrement pas. Au cours d'une évaluation extensive et étalée sur le calendrier, nous remarquons que ses réponses ne sont toujours pas correctes; ChatGPT tentant d'offrir une réponse unique à chacun de ses utilisateurs, finit parfois par donner des informations erronées, spécialement après une longue discussion car en répondant, il essaie de prendre en compte tous les sujets pertinents dans son échange avec l'utilisateur et de formuler une réponse qui joint tous les bouts. Cette chasse à l'originalité s'incruste même dans son service de références; lorsqu'on lui demande des références pour les informations qu'il fournit, il donne pour la plupart des fois des références inexistantes; soit il retourne de faux articles (peut-être parce-qu'il essaie de reformuler les titres), ou il les associe aux mauvais auteurs.

Faute à cette inexactitude et ce sourçage qui laisse à désirer, beaucoup d'organisations ferment leurs portes à ChatGPT par peur de propager de la mauvaise information; en décembre 2022, le site web de questions réponses Stack Overflow a banni l'utilisation de ChatGPT pour générer des réponses aux questions posées¹⁶. En janvier 2023, la conférence internationale sur l'apprentissage automatique (*International Conference on Machine Learning*) a banni l'usage non-documenté de ChatGPT ou d'autres applications LLM similaire pour générer du texte dans les papiers soumis¹⁷. En Mai 2023, Samsung fit de même en interdisant l'usage de ChatGPT à ses employés¹⁸. Ceux-là ne représentent que le début d'une longue liste d'entités qui ont banni l'usage de ChatGPT et d'outils similaires.

En mars 2023, plus d'un millier de chercheurs et leaders dans le domaine technologique ont pétitionné un halte au développement des systèmes avancés d'intelligence artificielle; il s'agit d'une lettre pressant les laboratoires de recherches dans le domaine de l'IA de prendre une pause de 6 mois, le temps de développer et d'implémenter des protocoles de sécurité avant d'accueillir des systèmes IA avancés qui pourraient être plus intelligents que GPT-4. La lettre relate que ce genre de systèmes pourraient poser une très grande menace à l'humanité, et que les chercheurs et experts devraient prendre un temps mort pour travailler ensemble pour s'assurer que leurs travaux s'alignent aux valeurs et éthiques humaines. La lettre fut signée par plus de 1800 personnalités incluant chercheurs, entrepreneurs, activistes et célébrités; À la surprise de tout le monde, parmi les

pétitionnaires se repère Elon Musk, fondateur de Tesla et co-fondateur de OpenAI, donnant des raisons plutôt implicites lors de son interview à la conférence technologique à Paris :

Nous devons minimiser la probabilité que quelque chose aille mal avec la super-intelligence digitale. Oui, je suis donc en faveur de l'instauration d'une régulation IA parce-que je pense que l'intelligence artificielle avancée est un risque au public, et toute chose qui est un risque au public doit se soumettre à une sorte d'arbitre, et cet arbitre est le modérateur.¹⁹

Les risques auxquels font allusion Musk et ses co-pétitionnaires demeurent un mystère, mais un bon nombre de risques sont apparents à l'issue de GPT. En effet, GPT-4 est d'une intelligence sans précédent; pour mettre en évidence cette intelligence, GPT-4 fut soumis à plusieurs tests dont certains qu'avait échoué son prédécesseur GPT-3.5, en particulier l'examen du barreau, l'examen auquel se soumet tous les aspirants avocats; non seulement le passe-t-il, mais il atteint un score battant ceux de 90% des avocats (voir **Fig 17**)²⁰.

Exam	GPT-4	GPT-4 (no vision)	GPT-3.5
Uniform Bar Exam (MBE+MEE+MPT)	298 / 400 (~90th)	298 / 400 (~90th)	213 / 400 (~10th)
LSAT	163 (~88th)	161 (~83rd)	149 (~40th)
SAT Evidence-Based Reading & Writing	710 / 800 (~93rd)	710 / 800 (~93rd)	670 / 800 (~87th)
SAT Math	700 / 800 (~89th)	690 / 800 (~89th)	590 / 800 (~70th)
Graduate Record Examination (GRE) Quantitative	163 / 170 (~80th)	157 / 170 (~62nd)	147 / 170 (~25th)
Graduate Record Examination (GRE) Verbal	169 / 170 (~99th)	165 / 170 (~96th)	154 / 170 (~63rd)
Graduate Record Examination (GRE) Writing	4 / 6 (~54th)	4 / 6 (~54th)	4 / 6 (~54th)
USABO Semifinal Exam 2020	87 / 150 (99th - 100th)	87 / 150 (99th - 100th)	43 / 150 (31st - 33rd)
USNCO Local Section Exam 2022	36 / 60	38 / 60	24 / 60
Medical Knowledge Self-Assessment Program	75 %	75 %	53 %
Codeforces Rating	392 (below 5th)	392 (below 5th)	260 (below 5th)
AP Art History	5 (86th - 100th)	5 (86th - 100th)	5 (86th - 100th)
AP Biology	5 (85th - 100th)	5 (85th - 100th)	4 (62nd - 85th)
AP Calculus BC	4 (43rd - 59th)	4 (43rd - 59th)	1 (0th - 7th)
AP Chemistry	4 (71st - 88th)	4 (71st - 88th)	2 (22nd - 46th)
AP English Language and Composition	2 (14th - 44th)	2 (14th - 44th)	2 (14th - 44th)
AP English Literature and Composition	2 (8th - 22nd)	2 (8th - 22nd)	2 (8th - 22nd)
AP Environmental Science	5 (91st - 100th)	5 (91st - 100th)	5 (91st - 100th)
AP Macroeconomics	5 (84th - 100th)	5 (84th - 100th)	2 (33rd - 48th)
AP Microeconomics	5 (82nd - 100th)	4 (60th - 82nd)	4 (60th - 82nd)
AP Physics 2	4 (66th - 84th)	4 (66th - 84th)	3 (30th - 66th)
AP Psychology	5 (83rd - 100th)	5 (83rd - 100th)	5 (83rd - 100th)
AP Statistics	5 (85th - 100th)	5 (85th - 100th)	3 (40th - 63rd)
AP US Government	5 (88th - 100th)	5 (88th - 100th)	4 (77th - 88th)
AP US History	5 (89th - 100th)	4 (74th - 89th)	4 (74th - 89th)
AP World History	4 (65th - 87th)	4 (65th - 87th)	4 (65th - 87th)
AMC 10	30 / 150 (6th - 12th)	36 / 150 (10th - 19th)	36 / 150 (10th - 19th)
AMC 12	60 / 150 (45th - 66th)	48 / 150 (19th - 40th)	30 / 150 (4th - 8th)
Introductory Sommelier (theory knowledge)	92 %	92 %	80 %
Certified Sommelier (theory knowledge)	86 %	86 %	58 %
Advanced Sommelier (theory knowledge)	77 %	77 %	46 %
Leetcode (easy)	31 / 41	31 / 41	12 / 41
Leetcode (medium)	21 / 80	21 / 80	8 / 80
Leetcode (hard)	3 / 45	3 / 45	0 / 45

Fig 17 - Une capture d'écran des 34 différents examens sur lesquels OpenAI a testé GPT4. JOHN

KOETSIER

Aujourd'hui, une large majorité des étudiants utilise ChatGPT pour traiter leurs devoirs à la maison quels qu'ils soient, ceci ouvre la porte au plagia et à la tricherie en général. Ce risque est maintenant le sujet de discussion de toutes les écoles, et certaines ont déjà pris action en bannissant l'utilisation de ChatGPT et d'outils similaires dans leurs lieux.^[21] Study.com, une plateforme éducationnelle en ligne utilisée par près de 30 millions d'étudiants et d'instituteurs, s'est assignée la tâche de conduire un sondage pour savoir les sentiments des enseignants et des étudiants à l'égard de l'émergence de ChatGPT; il a été demandé à 100 enseignants et 1000 étudiants en ligne (de divers pays) quels étaient leurs opinions de ChatGPT,^[22] et les résultats du sondage se résument ainsi :

- De la perception des enseignants où :
 - 82% des professeurs de collège (PC) ont entendu parler de ChatGPT, comparé 55% des enseignants d'écoles primaires (EEP).
 - 72% des PC qui sont au courant de ChatGPT sont inquiets de son impact sur la tricherie, comparé 58% des EEP.
 - 34% des enseignants sont pour le bannissement de ChatGPT, alors que 66% sont en faveur que les étudiants y aient accès.
- De la perception des étudiants où :
 - 9 étudiants sur 10 sont au courant de ChatGPT.
 - 89% des étudiants ont utilisé ChatGPT pour s'aider à traiter leurs travaux à la maison.
 - 48% des étudiants ont admis s'être servi de ChatGPT pour un examen à domicile ou quiz, 53% ont écrit une rédaction avec, et 22% ont généré un plan pour leurs articles avec.
 - À la surprise de tous, 72% des étudiants pensent que ChatGPT devrait être banni du réseau de leurs collègues.

Toutes ces données prouvent que ChatGPT est une réelle menace au secteur éducatif; les étudiants pourraient trop en dépendre ce qui pourrait conduire au plagia et à la tricherie. Parmi les actions prises pour tenter de remédier à cela, certains outils ont été conçus pour détecter les contenus générés par l'IA notamment [ContentDetector.AI](#), [GPTZero](#) et bien d'autres.

Pour le secteur professionnel, ChatGPT est un nuage noir à l'autre bout de l'horizon, et plusieurs groupes démographiques en sont de plus en plus rapprochés. Pour les littéraires tels que les journalistes, les auteurs de bouquins, les blogueurs et autres, la menace est belle et bien réelle; ils

se voient accordés un sursis par les détecteurs de textes IA qu'on vient de mentionner. Hélas, ChatGPT apprend par renforcement, il devient meilleur de jour en jour et apprend à générer du texte qui devient de plus en plus difficile à distinguer de l'écriture humaine; il est cru qu'il sera bientôt très difficile voire impossible même pour les détecteurs de contenu IA de faire la différence. Cette peur est aussi partagée par le département informatique, ChatGPT est capable d'écrire du code, et vient récemment de se munir de la faculté de tester son propre code; cela veut dire qu'il pourrait générer un code à la demande de l'utilisateur, le tester, et si le code échoue le test, il régénère jusqu'au code qui réussira. À l'issue de son émergence, Stack Overflow qui était le site de question-réponse favoris des développeurs observe un fort déclin d'activité de la part de ses utilisateurs²³ Les programmeurs gagnent beaucoup de temps avec ChatGPT qui leur retourne des bouts de codes contextualisés dans divers langages de programmation. Cela dit, ChatGPT est loin d'être qualifié pour totalement remplacer un développeur, et les sociétés ne sont pas non plus enthousiastes à l'idée de prendre cet énorme risque.

ChatGPT sème la terreur dans le monde avec son arsenal de fonctionnalités sans précédent dans le domaine de l'intelligence artificielle, et cela juste après 8 mois de sa date de publication; cette date marque une nouvelle ère, l'ère de la super-intelligence digitale (SID); et avec Google, Microsoft et bien d'autres compagnies investissant dans la SID, la peur du robot remplaçant l'homme semble de plus en plus justifiée.

Projet : Prévention de crime avec l'Intelligence Artificielle

3.1. Motivation

S'il y a un secteur qui ne connaît pas un moment de répit c'est bien celui des forces de l'ordre. Ce n'est point une exagération de dire que des milliers de meurtres se commettent par jour dans le monde. En 2021, Seulement au Canada, il y a eu plus 788 homicides commis, 29 de plus que l'année passée et 41% de ces homicides ont été commis par armes à feu ²⁴. Malheureusement, ce n'est pas la totalité de ces cas meurtres qui aboutissent à une résolution; d'après Statistique Canada, toujours dans la même année, plus de 263 cas de homicides parmi les mentionnés 788 ne sont toujours pas résolus, cela se traduit par un taux de résolution de 67 %, ce qui est inférieur au taux correspondant de 2020, et donc une mauvaise direction, un mauvais signe ²⁴.

La philosophie "***mieux vaut prévenir que guérir***" est une qui est très délicate à implémenter dans l'application de la loi; il est illégal et inconcevable de détenir quelqu'un sur une base aussi volatile que la suspicion, et il en résulte que la police ne puisse prendre action que suite à l'exécution du crime, aussi grave que ce dernier puisse être. Cette prise d'action est d'habitude très délayée car la police n'est notifiée, dans la majorité des cas, que par un appel téléphonique d'un civil qui aurait remarqué le corps peut-être des heures, voire des jours après l'occurrence du crime.

Cependant, à l'aide de cette récente percée de l'intelligence artificielle, il serait peut-être possible de faciliter la tâche des agents des forces de l'ordre tout autant dans la mesure préventive que notificative, ce qui aiderait à sauver beaucoup de vies et aussi à rendre justice à celles perdues. Ainsi se formule la motivation pour notre projet de prévention de crimes à l'aide de l'intelligence artificielle; il s'agit d'une application intégrant, grâce à la vision par ordinateur (en anglais Computer Vision), plusieurs modules permettant de reconnaître des scènes de crimes et peut-être en prédire. L'application adoptera l'architecture modulaire car cette dernière est évolutive, elle nous permettra d'ajouter des fonctionnalités (modules) indépendamment des autres, contrairement à l'architecture monolithique dont les services sont traditionnellement très couplés.

Dans un premier temps, nous parlerons brièvement de la **vision par ordinateur (VPO)** qui permet le réalisme de cet effort, une branche de l'intelligence artificielle qui permet à la machine une compréhension de haut niveau à partir d'images ou de vidéos numériques; pour donner une vue sur les possibilités offertes par la vision par ordinateur, nous présenterons un débouché majeur de cette branche de l'IA. Dans un second temps, nous discuterons de l'adoption de l'intelligence artificielle par les forces de l'ordre, des initiatives prises pour atteindre des objectifs similaires aux notre. En troisième lieu, nous procéderons à l'implémentation de notre application en théorisant

d'abord l'approche à laquelle nous adhérons, détaillant ainsi les concepts de bases et les outils utilisés pour développer ce projet tout en analysant les résultats. Finalement nous estimerons les conditions requises pour déployer ce programme dans une ville intelligente.

3.2. Vision par ordinateur ou Computer Vision

La vision par ordinateur (VPO) est une branche de l'intelligence artificielle se donnant l'objectif d'ajouter à l'ordinateur la capacité de comprendre et d'identifier des objets et personnes dans des images et vidéos numériques. C'est une application du réseau neuronal convolutif mentionnée dans le chapitre précédent.

Les tâches de la vision par ordinateur incluent les méthodes d'acquisition, de traitement, d'analyse et compréhension des images digitales, ainsi que l'extraction de données de hautes dimensions à partir du monde réel dans le but de produire des informations numériques et symboliques, c'est-à-dire dans la forme de décisions. La compréhension qu'on vient d'énumérer signifie la transformation d'images visuelles (par analogie, les données d'entrées de la rétine) en descriptions logiques du monde, prenant sens dans le lexique algorithmique.

La vision par ordinateur entraîne la machine à effectuer ces opérations sous délais fortement réduits, en utilisant des caméras, des données et algorithmes à l'instar de la rétine, des nerfs optiques et du cortex visuel.

La vision par ordinateur devient similaire à celle de l'être humain en l'imitant, à la différence que l'Homme détient l'avantage d'une vie entière d'entraînement à la perception de la profondeur, de mouvements et de reconnaissance d'objets, et aussi d'erreurs dans une image.

Il existe plusieurs applications de la VPO telles que la reconnaissance faciale, la reconnaissance de caractères, et la tendance à la date d'écriture de ce texte, le générateur d'art qu'on va tout de suite examiner.

3.2.3. Débouché majeur : Midjourney AI Image Generator

Midjourney AI Image Generator est une technologie dernier-cri combinant la vision par ordinateur et l'apprentissage automatique pour générer des images de hautes qualités en temps réel. Cette technologie est un type de synthèse d'images qui permet aux utilisateurs de créer des images personnalisées avec un haut degré de contrôle et de précision. Midjourney AI Image Generator a un large éventail d'applications dans divers secteurs, notamment le marketing et la publicité, le développement de jeux vidéo, l'art et le design numériques, ainsi que le cinéma et l'animation.

Midjourney AI Image Generator fonctionne en utilisant des techniques d'apprentissage en profondeur, telles que les réseaux contradictoires génératifs (GAN), pour former des réseaux de neurones sur de grands ensembles de données d'images. Ces réseaux peuvent ensuite générer de nouvelles images qui imitent le style et le contenu des données d'entraînement. Cette technologie peut produire des images très réalistes, difficiles à distinguer des photographies réelles ou des illustrations dessinées à la main.

En termes plus simples, il peut lire votre description d'une image (texte) et créer une image basée sur celle-ci.

Voici une image générée par Midjourney avec la description d'entrée suivante (voir Fig 18) :

"Un mélange de fruits secs, de graines et de chocolat noir pour un en-cas portable."



Fig 18 - Image réaliste générée par Midjourney avec la commande suivante : "Un mélange de fruits secs, de graines et de chocolat noir pour un en-cas portable."

Midjourney n'est pas le premier à faire cela. Nous avons déjà des modèles comme DALL-E 2 et Stable Diffusion de OpenAI, mais ce qui le distingue est sa capacité à adapter des styles artistiques qui peuvent être mélangés et assortis pour créer une image.

3.3. Revue de littérature

3.3.1. Initiatives déjà prises : l'utilisation de l'Intelligence Artificielle par les forces de l'ordre.

Il est peu de dire qu'on ne peut fermer nos yeux aux progrès très évidents de l'intelligence artificielle cette dernière décennie. Toutes les entités de tous les secteurs essaient d'intégrer l'intelligence artificielle dans leurs flux de travail, et cela n'épargne en rien les forces de l'ordre. La police tire profit de trois fonctions de l'intelligence artificielle pour se faciliter la tâche dans plusieurs aspects de leurs travaux.

Recueil intelligent de données à des fins de prédiction

L'IA est capable d'analyser de larges bases de données d'archives casiers judiciaires pour prédire un potentiel récidivisme criminel. Les chercheurs au **Research Triangle Institute (RTI)**, en partenariat avec le **Département Policier de Durham** et avec le bureau du shérif du comté d'Anne Arundel (Maryland), travaillent pour la création d'un outil automatisé de tri des services de mandats pour le référentiel de mandats de l'État de Caroline du Nord.²⁵

L'équipe soutenue par le **l'Institution de Justice Nationale** (National Institute of Justice (NIJ) en anglais) utilise des algorithmes pour analyser des ensembles de données comprenant plus de 340 000 enregistrements de mandats. Les algorithmes forment des arbres de décision et effectuent des analyses de survie pour déterminer le délai avant la prochaine occurrence d'un événement intéressant, et prédire le risque de récidive pour les fugitifs (si un mandat n'est pas reçu). Ce modèle aidera les praticiens à trier les mandats en cas d'arriérés. L'outil qui en résultera sera également géographiquement référencé afin que les praticiens puissent poursuivre des concentrations de fugitifs à haut risque - ainsi que d'autres qui ont des mandats actifs - afin d'optimiser les ressources.

²⁵

Le service de police de Chicago et l'Illinois Institute of Technology ont utilisé des algorithmes pour collecter des informations et former des groupes initiaux axés sur la construction de réseaux

sociaux et la réalisation d'analyses pour déterminer les individus potentiels à haut risque. Cette recherche soutenue par le NIJ (National Justice Institute) fait depuis lors partie de la stratégie de réduction de la violence du département de police de Chicago.²⁵

La Reconnaissance faciale

La reconnaissance faciale a été bien reçue et appliquée un peu partout dans le monde pour aider à identifier les criminels dans les espaces publics et dans les foules. Tout récemment, des officiers chinois ont arrêté avec succès une personne dans une foule à l'intérieur d'un stade; cet exploit serait presque impossible sans la technologie de l'intelligence artificielle. Dans la ville de Londres, on expérimente avec l'IA dans les gares de chemins de fer pour aider à augmenter le niveau de sécurité. Les États-Unis aussi se voient en besoin d'une telle technologie; en une seule année, celle de 2017, plus de 546568 personnes ont été déclarées disparues d'après *Statista* ²⁶. Pour remédier à cela, la reconnaissance faciale est donc utilisée par les forces de l'ordre pour scanner les endroits publics tels que les stades ou les aéroports afin de localiser les personnes portées disparues.

Bien qu'elle présente de nombreux avantages, la reconnaissance faciale est quand-même sujet aux dédains dans plusieurs coins du globe; les membres du Parlement européen ont convenu d'interdire la reconnaissance faciale en direct utilisant l'IA dans un projet de loi sur l'intelligence artificielle (IA) - la législation historique de l'UE catégorisant les différentes applications de l'IA en fonction de leur préjudice pour le public ²⁷.

Robots physiques plus intelligents

Les forces de l'ordre font de plus en plus usage des robots physiques pour effectuer une variété de tâches. La police fait recours à ces robots pour effectuer des tâches considérées dangereuses pour l'homme telle que la diffusion d'une bombe ou entrer dans un environnement dont les conditions sont inconnues qui pourraient notamment présenter un risque à la santé des agents. Ces mentionnés robots (voir **Fig 19**) sont maintenant équipés par les dernières technologies de l'Intelligence Artificielle afin de les offrir des fonctionnalités plus avancées, plus sophistiquées telles qu'une meilleure identification des objets et personnes, et aussi un automatisme allégeant le rôle de l'agent qui pilote le robot.



Fig 19 - Robot policier à Dubaï

Certaines villes font recours à ce genre de robots leur assignant un rôle d'officier supplémentaires dans les rues. En 2017, Dubaï introduit des robots physiques agissant comme policiers, en les plaçant dans presque tous les coins touristiques de la ville.²⁸

Ces robots conversaient dans six langues avec les touristes et ses fonctions étaient limitées à permettre au public de reporter un crime grâce à un écran tactile, et à transmettre, grâce aux caméras, des images en direct à la chambre des opérations où des personnes humaines surveillaient le flux.

3.3.2. Analyse et critiques de l'existant

3.3.2.1. La faille de la reconnaissance faciale

La reconnaissance faciale, comme on peut le déduire du nom, compare une image avec une autre pour préciser s'il s'agit de la même personne. Ces images pourraient provenir d'une photo ou vidéo numérique, mais aussi d'un flux en direct d'une caméra qui pourrait être téléverser directement

dans la base de données d'une station de police. Cependant, pour qu'un modèle soit très précis, il faudrait un **entraînement** sur un large nombre de données d'entrées; par analogie, il est requis d'avoir un très bon nombre de photos de la personne posant avec différentes expressions faciales pour que le programme de reconnaissance faciale soit précis, c'est-à-dire plusieurs photos de l'individu souriant, coléreux, triste etc. Le programme, tel qu'il est, ne donne pas une réponse binaire, c'est-à-dire un oui ou un non, à la comparaison mais il présente les meilleurs possibilités, les triant selon la probabilité ou pourcentage de ressemblance. Ensuite, il incombe au policier et au témoin oculaire de prendre la décision finale. Ainsi, nous avons deux fragilités devant nous, d'une main le manque **d'entraînement** de la machine et l'imprécision qui pourrait en résulter, et de l'autre l'erreur humaine du témoin oculaire et le très possible préjugé de ce dernier.

En effet, des moments intenses telles qu'une agression quelle qu'elle soit générant stress et anxiété pourrait être facteurs d'une perte temporaire de mémoire, menant à une mauvaise identification de la part d'un confus témoin oculaire. Ajouté à cela, les témoins oculaires peinent souvent à identifier les personnes d'une race autre que la leur; c'est un phénomène appelé "**erreur d'identification raciale**" ²⁹.

Parmi tous les défauts de la reconnaissance faciale que nous venons de souligner, nous mettons en relief celui d'un manque d'entraînement à haut niveau de spécificité, c'est-à-dire que pour avoir une **parfaite** précision, nous avons besoin d'un modèle pour chaque citoyen en recueillant une énorme quantité de photos pour chaque individu portant différentes expressions faciales, faute de quoi le programme peinera à distinguer deux personnes partageant une certaine ressemblance. Cette solution requerra une ressource énorme, prises de photos, une base de données à stockage et performance phénoménale, et une machine à pareille spécification, et pour mettre un point sur les i, cette énorme quantité de données concernant les citoyens ajoutera certainement beaucoup de propane sur le feu qui est analogue des troubles légaux que nous avons mentionnés précédemment.

3.3.2.2. Antécédents douteux

Nous avons précédemment mentionné l'utilisation de l'intelligence artificielle par la police pour prédire la récidivité d'un individu présentant un antécédent criminel enregistré dans la base de données de la police; cependant, est-ce vraiment une méthode fiable ?

Le nombre de fois qu'un individu se fait injustement arrêté par la police ne se compte plus sur les doigts. Certaines victimes de cet empiétement se voient lavées de cette offense et d'autres, non. Il y a une forte probabilité que plusieurs cas d'injustices figurent dans les bases de données policières et si le programme de prédiction par intelligence artificielle est entraîné sur ces données là, il

pourrait en résulter de très mauvaise prédiction pouvant mener à l'incarcération de plusieurs innocentes personnes.

3.3.2.3. Allées non explorées

Dans cette section, nous énumérerons les différentes possibilités offertes par l'intelligence artificielle qui n'ont pas été préférées. Nous expliquerons en détails leurs utilités et leurs fonctionnements que nous allons implémenter individuellement dans la section suivante.

Reconnaissance d'outils criminels dans un espace public

L'intelligence artificielle nous permet plus de fonctionnalités aujourd'hui qu'elle ne l'a jamais permis. La détection d'objets qui est déjà légèrement utilisée par la police peut être orientée vers un autre cas d'utilisation très important voire crucial. En effet, on pourrait utiliser un programme de détection d'objets pour identifier les armes à feu dans les endroits publics; ce programme sera installé dans les caméras de surveillances dans les endroits publics; il opérera sur un flux en direct et notifiera la station de police la plus proche suite à la détection d'une arme à feu. Une fois notifié, le policier pourrait décider s'il s'agit d'un événement criminel ou non.

Au delà des espaces publics, ce programme pourrait même trouver son utilité dans les institutions financières. Traditionnellement, pour notifier un danger dans une banque tel qu'une attaque à main armée (braquage), un agent de la banque devait discrètement appuyer sur un bouton d'urgence placé quelque part, usuellement sous le bureau d'accueil (ou comptoir d'accueil); malheureusement les braqueurs sont très conscients de cela et gardent un œil sur tous les employés de la banque afin d'éviter la notification de la police. Ce programme pourrait détecter automatiquement une arme à feu et le notifier instantanément à la station de police la plus proche, un laps de temps qui pourrait faire une énorme différence.

Détection de chute dans un espace public

Il est aussi possible d'utiliser l'intelligence artificielle pour reconnaître une action qui serait très probablement indicatrice d'un scène de crime, il s'agit de la chute d'un individu. Crime ou pas, lorsqu'une personne fait une chute dans un espace public et reste sur l'horizontale pour une certaine durée, il est très pertinent de notifier la police ou les urgences, car il s'agit certainement d'une gravité. En entraînant un modèle avec un nombre suffisant de données, nous pourrions ajouter à notre programme conceptuel, la fonctionnalité de détection de chute. Cette fonctionnalité appairée avec celle de la reconnaissance d'un outil criminel pourrait déterminer à une très haute précision une scène de crime, car lorsque le programme détecte, par exemple, une arme à feu et quelques instants plus tard un individu est par terre, cela s'agirait très certainement d'une attaque

et il sera primordial d'en notifier la police et, par convention, les urgences pour s'occuper de la victime.

Expression faciale

Pour une meilleure précision, on pourrait apparier avec les précédentes fonctionnalités, la détection de l'expression faciale. Lorsque la colère d'un individu se fait remarquer dans un espace public, cette personne est probablement à garder un œil sur; cela dit, on met en emphase un degré de colère très apparente dans le visage de l'individu. Une expression agressive associée au port d'une arme quelconque pourrait très certainement indiquer une action criminelle potentielle et en notifier la police pourrait très probablement sauver une ou plusieurs vies.

Une seconde émotion pourrait être aussi importante à observer que la première, il s'agit bel et bien de la peur. Lorsqu'un individu porte l'expression de la peur sur son visage dans un endroit public tel qu'une allée ou les stations de trains dans la nuit, cet individu se sent très certainement en danger et ce serait très important de porter cela à l'attention la police, invitant une surveillance humaine des lieux.

S'appuyant sur plus de 50 ans de recherche, le Dr Eckman a mis au point un système de codage des manifestations émotionnelles des individus impliquant des centaines de combinaisons de muscles faciaux, souvent corrélées scientifiquement à des états émotionnels distincts ³⁰. Il appelle chaque mouvement facial observable une « unité d'action » ou une UA. L'une de ses principales contributions à la recherche concerne les micro-expressions, de brefs éclairs d'émotions révélées sur le visage. Cette formation a été utilisée par des experts en détection de mensonges, des acteurs et des agents de la sécurité intérieure.

Science of People (La Science des Personnes en français) nous donne un bref résumé des travaux du Dr Eckman sur les différents indicateurs d'émotions observables sur le visage. ³⁰ Nous n'énumérerons que deux émotions, celles qui nous intéressent notamment la colère et la peur :

- **Colère (voir Fig 20) :**
 - Froncement et rapprochement des sourcils, souvent avec des lignes verticales entre eux
 - Tension dans les paupières inférieures
 - Yeux larges ou saillants
 - Lèvres pressées ensemble avec des coins tournés vers le bas
 - Dilatation des narines

- Saillie de la mâchoire inférieure



Fig 20 - La colère

- **Peur** (voir Fig 21):
 - Sourcils relevés ensemble, généralement en une ligne plate
 - Rides entre les sourcils
 - Paupières supérieures relevées avec des paupières inférieures tendues
 - La partie supérieure de l'œil montre du blanc



Fig 21 - Un peu d'humour... Kevin dans Home Alone

Grâce à l'apprentissage automatique appairé à la vision machine, il serait possible d'entraîner un modèle pouvant reconnaître ces patrons là sur le visage de l'individu au moyen d'une caméra de surveillance.

Nous tenterons d'implémenter chacune de ses fonctionnalités dans une application IA afin de remédier aux manques que nous venons de souligner de l'existant.

3.4. Méthodologie

Nous décrivons la méthodologie choisie pour implémenter individuellement les fonctionnalités mentionnées dans la section des *allées non explorées* lors de la revue de l'existant.

Nous avons préféré une architecture modulaire pour ce programme afin de permettre une testabilité simple des fonctionnalités et une évaluation approfondie de la performance des modules pour en déduire les ressources nécessaires.

3.4.1. Environnement de développement

Dans cette section de la méthodologie, nous allons décrire l'environnement élu pour mener à bien le développement du programme. Différents modules de ce programme requièrent différentes bibliothèques, par conséquent nous allons spécifier les bibliothèques nécessaires pour chaque module dans sa section.

3.4.1.1. Langage de programmation

Le langage de programmation préféré pour ce projet est **Python**. Python est un langage de programmation de haut niveau, interprété et très versatile. Python présente beaucoup d'avantages et voilà les aspects de Python qui nous intéressent :

- **Lisibilité** : La syntaxe de Python est conçue pour être très lisible, et cet objectif est atteint avec les indentations et une structure propre et uniforme.
- **Interprété** : Python est un langage interprété, cela veut dire qu'il est possible d'écrire et d'exécuter son code sans la nécessité d'une autre étape de compilation, ce qui permet un développement et un débogage très simple.
- **Bibliothèques** : Python supporte une large quantité de bibliothèques facilitant la tâche dans presque tous les domaines informatiques possibles, spécialement dans celui de l'intelligence artificielle.
- **Multiplateforme** : Python est disponible dans presque toutes les plateformes incluant Windows, Linux et macOS.

La facilité d'utilisation de Python, le soutien important de la communauté et la disponibilité de bibliothèques et de modèles prédéfinis en font un excellent choix pour le développement de l'IA. Il permet aux chercheurs et aux développeurs de se concentrer sur les algorithmes et les applications de l'IA plutôt que sur les complexités de la programmation de bas niveau. C'est pourquoi Python est le langage de choix pour les projets d'IA et d'apprentissage automatique.

3.4.1.2. Plateforme

L'entraînement d'un modèle est une opération qui requiert une ressource presque phénoménale que la performance d'un ordinateur lambda ne permet pas. C'est pourquoi la plateforme choisie pour développer notre projet est **Google Colaboratory**.

Google Colab, abréviation de Google Colaboratory, est une plateforme basée sur le cloud fournie par Google qui offre un accès gratuit à un environnement de cahier Jupyter avec des ressources

GPU (unité de traitement graphique) et TPU (unité de traitement de tenseurs). Elle est conçue pour permettre aux chercheurs, aux étudiants et aux data scientists d'écrire, d'exécuter et de partager du code et des documents. Google Colab est devenu un choix populaire pour diverses tâches en apprentissage automatique, en analyse de données, et plus encore en raison de sa commodité et de ses ressources rentables.

Voici quelques caractéristiques et aspects clés de Google Colab :

- **Cahiers Jupyter** : Google Colab permet aux utilisateurs de créer et d'exécuter des cahiers Jupyter. Les cahiers Jupyter sont des documents interactifs qui combinent du code, du texte et des éléments multimédias, facilitant ainsi l'écriture et le partage de code avec des explications et des visualisations.
- **Ressources GPU et TPU gratuites** : Google Colab offre un accès gratuit à des ressources GPU et TPU, essentielles pour l'entraînement de modèles d'apprentissage profond et d'autres tâches intensives en calcul.
- **Basé sur le cloud** : Vous n'avez pas besoin d'installer de logiciel sur votre machine locale ; tout est hébergé dans le cloud. Vous pouvez accéder à vos cahiers Colab de n'importe où avec une connexion Internet.
- **Pré-installation de bibliothèques** : Google Colab est pré-installé avec de nombreuses bibliothèques populaires et des outils pour l'analyse de données et l'apprentissage automatique, tels que TensorFlow, PyTorch, NumPy, Pandas, et bien d'autres.
- **Stockage** : Vous pouvez facilement télécharger et accéder à des données depuis Google Drive ou depuis votre machine locale.

Pour faire court, Google Colab nous permettra de développer sans utiliser les ressources de notre machine locale, mais plutôt celles qu'il offre qui sont de loin plus optimisées pour effectuer les tâches liées à l'intelligence artificielle.

NB: Les opérations que nous allons effectuer sont très coûteuses en termes de performance, nous recommandons d'utiliser **Google Colab** pour procéder et de sélectionner l'option **GPU T4** pour de meilleures performances, car autrement la majorité des scripts vont prendre énormément de temps à s'exécuter.

3.4.2. Module 1 : Détection d'armes

Pour implémenter cette fonctionnalité nous devons entraîner la machine à reconnaître une arme à feu lorsque cette dernière apparaît sur une image; nous rappelons qu'une vidéo n'est autre qu'une succession d'images, alors le principe est le même.

3.4.2.1. Outils de développement

La détection d'objets est une tâche populaire en vision par ordinateur. Il s'agit de localiser une région d'intérêt dans une image et de classer cette région comme un classificateur d'image typique. Une image peut inclure plusieurs régions d'intérêt pointant vers différents objets. Cela fait de la détection d'objets un problème plus avancé de classification d'images. Pour nous aider à relever ce défi, nous avons fait usage des outils suivants.

YOLO

YOLO (You Only Look Once) est un modèle de détection d'objets populaire connu pour sa rapidité et sa précision ³¹. Il a été introduit pour la première fois par Joseph Redmon et al. en 2016 et a depuis subi plusieurs itérations, la dernière en date étant YOLO v8 (voir Fig 22).



Fig 22 - YOLO

L'algorithme de YOLO prend une image comme donnée d'entrée et utilise un réseau profond de neurones convolutifs pour détecter les objets présents dans l'image. L'architecture du réseau de neurones convolutifs en question est comme suit (voir Fig 23):

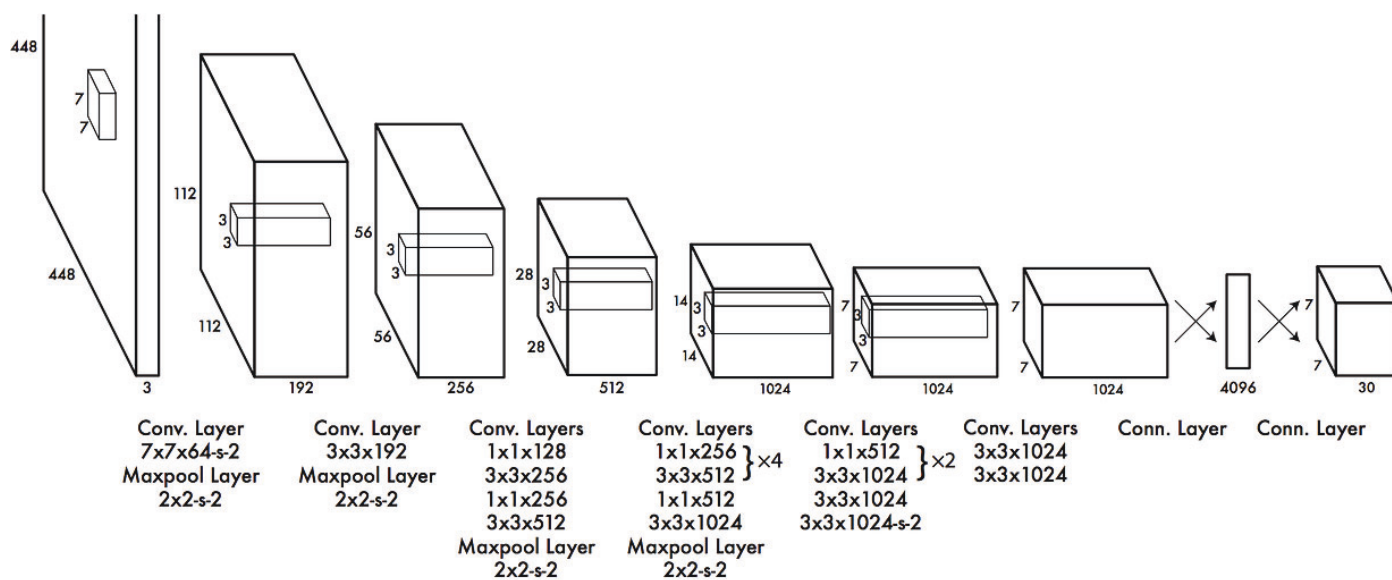


Fig 23 - Architecture du réseau neuronal convolutif de YOLO

Le réseau est composé de 24 couches de convolution suivies de deux couches entièrement connectées. Les 20 premières couches de convolution du modèle sont pré-entraînées à l'aide d'une large base de données d'images appelé ImageNet ^[32] en connectant un pool moyen temporaire et une couche entièrement connectée. Ensuite, ce modèle pré-entraîné est converti pour effectuer une détection puisque des recherches précédentes ont montré que l'ajout de couches de convolution et connectées à un réseau pré-entraîné améliore les performances. La couche finale entièrement connectée de YOLO prédit à la fois les probabilités de classe et les coordonnées du cadre de délimitation ^[33].

Roboflow

Roboflow (voir Fig 24) est une plateforme en ligne qui met à la disposition des développeurs un arsenal d'outils facilitant les tâches de vision par ordinateur. C'est tout d'abord un énorme dépôt de bases de données d'images annotées qu'on peut télécharger gratuitement, ce qui nous épargne la lourde tâche de prendre des dizaines de milliers d'images et de les annoter nous-même. L'annotation peut aussi être faite dans Roboflow, car il offre un outil permettant de visualiser les images fournies par l'utilisateur et permet à ce dernier de les annoter (ou étiqueter), et d'en publier la base de données.

Roboflow présente plusieurs autres fonctionnalités telles que la collaboration qui permet aux équipes de travailler facilement ensemble (par exemple se répartir les tâches d'annotations), et le déploiement.



Fig 24 - Roboflow

Après s'être inscrit Roboflow donne à l'utilisateur une clé API pour lui permettre d'intégrer Roboflow dans une application tierce. En installant la librairie Roboflow sur Python on peut accéder à toutes les bases de données publiques à partir de notre application, et la clé API nous donne l'autorisation d'effectuer cela.

3.4.2.2. Base de données

Nous enseignons à la machine à reconnaître un objet lorsqu'il apparaît sur image, mais avant de reconnaître, il faut connaître. Pour les êtres humains, il suffit qu'on nous indique une chose une ou deux fois pour pouvoir la reconnaître dans presque n'importe quelle circonstance; ce n'est malheureusement pas le cas pour la machine, elle requiert qu'on lui indique une dizaine de milliers de fois cette même chose dans différentes circonstances pour qu'elle puisse la reconnaître automatiquement.

L'indication qu'on souligne ici fait référence à l'apprentissage supervisé qu'on a expliqué en détails dans le tout premier chapitre. Il s'agit de prendre une énorme quantité d'images présentant l'objet intéressant dans différentes positions et dans différents environnements et de spécifier dans chacune d'elle la région d'intérêt, c'est-à-dire la région de l'image encadrant l'objet à détecter, cette action s'appelle annotation ou étiquetage.

Cette tâche d'annotation peut s'effectuer facilement avec **Roboflow**. La plateforme offre une interface intuitive permettant de sélectionner une région avec la souris et suggère une classe (étiquette) parmi celles que l'utilisateur a définies (voir **Fig 25**).

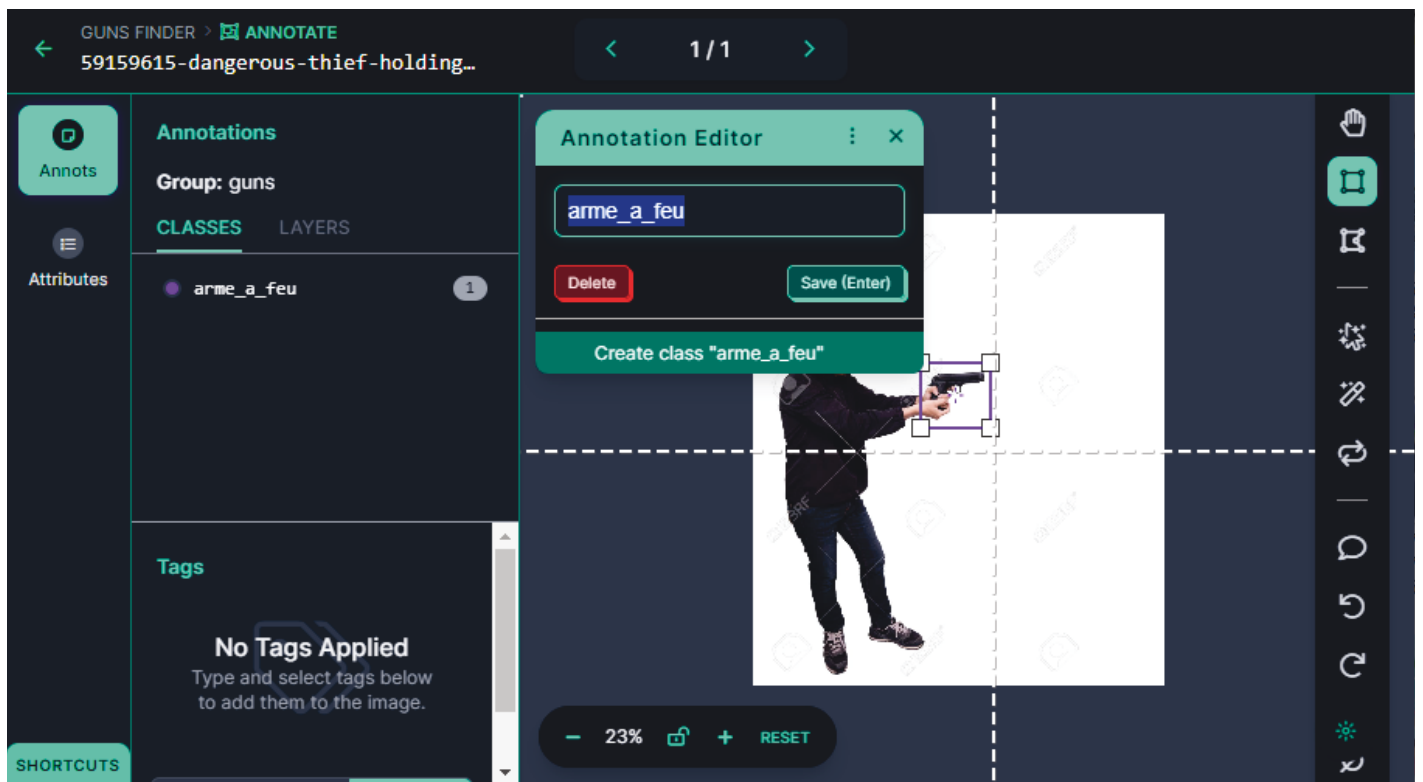


Fig 25 - Étiquetage avec Roboflow

La tâche d'annotation est une très lourde responsabilité, c'est pourquoi Roboflow présente aussi l'option d'inviter des coéquipiers pour aider à cette tâche. Une fois les images annotées, l'utilisateur peut publier sa base de données pour permettre aux autres chercheurs d'en faire usage, leur sauvant une énorme quantité de temps.

À titre illustratif, n'ayant ni une large collection d'images ni un large personnel, nous avons choisi une base de données publique pour développer notre module de détection d'arme; il s'agit d'une base de données d'images étiquetées de pistolets disponible [ici](#) ³⁴.

3.4.2.2. Entraînement du modèle

Nous allons utiliser YOLOv8 pour entraîner notre modèle, nous pouvons procéder à son installation grâce à la commande suivante qu'on entre dans une invite de commande ou dans notre cas, un cahier Jupyter dans Google Colab (notez le signe '!' devant la commande) :

```
# Doit avoir installé Python et Pip
!pip install ultralytics
```

Pour intégrer Roboflow dans notre application et avoir accès aux bases de données publiques, nous devons aussi installer la librairie `roboflow` avec la commande suivante :

```
!pip install roboflow
```

Accédons à la base de données sélectionné grâce au code suivant (les textes suivant le caractère `#` sont des commentaires sur le code de la ligne suivante :

```
# Commençons par importer la classe Roboflow depuis la librairie éponyme
from roboflow import Roboflow
# Créons une instance en saisissant notre clé API privée
rf = Roboflow(api_key="votre_clé_api_roboflow_ici")
# Accédons au projet contenant la base de données souhaitée
project = rf.workspace("abhishekarparth").project("pistol-y5ku1")
# Accédons à la base de données en spécifiant la version et le format
dataset = project.version(1).download("yolov8")
```

Suite au téléchargement de la base de données, on note dans l'explorateur de fichier un nouveau dossier nommé après la base de données (voir Fig 26) :

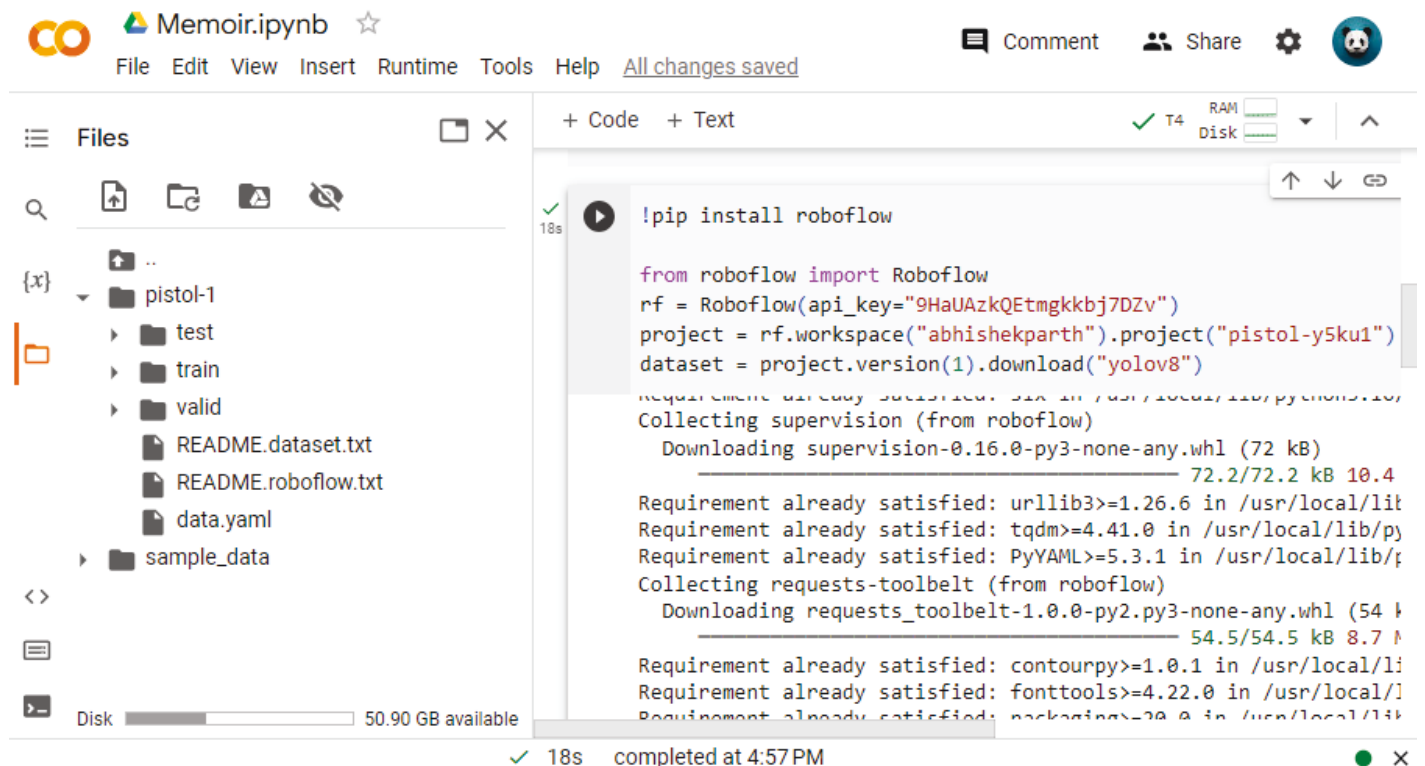


Fig 26 - Dossier contenant la base de données

En dehors des fichiers d'instructions (les fichiers textes README), on y retrouve trois dossiers et un fichier expliqué comme suit :

- `train/` : Dans ce dossier se trouve toutes les images et leurs annotations (étiquettes) qu'on utilisera pour entraîner le modèle.
- `val/` : Une portion des images se trouvant dans `train/` est copiée pour valider le modèle.
- `test/` : Sont placées ici des images choisies arbitrairement et différentes de celles utilisées pour l'entraînement du modèle. Elles sont utilisées pour tester notre modèle sur des données étrangères.
- `data.yaml` : un fichier parchemin `.yaml` spécifiant l'emplacement des trois dossiers mentionnés en haut et des une signature concernant la base de données :

(NB: enlever le nom du dossier racine pour conformer avec le système d'acheminement Google Colaboratory)

```
names:
- '0' # le nom de la classe, l'auteur a préféré parce qu'il n'y a pas d'autres
nc: 1 # nombre de classes
roboflow:
  license: MIT
  project: pistol-y5ku1
  url: https://universe.roboflow.com/abhishekarth/pistol-y5ku1/dataset/1
  version: 1
  workspace: abhishekarth
# emplacement du dossier 'test'
test: ../test/images
# emplacement du dossier 'train'
train: pistol-1/train/images # enlever "pistol-1" pour conformer avec Google Colab
# emplacement du dossier 'val'
val: pistol-1/valid/images # enlever "pistol-1" pour conformer avec Google Colab
```

Pour démarrer l'entraînement, on exécute YOLO en spécifiant les arguments suivants :

- `mode` : le mode d'exécution réglé sur `train` pour entraînement.
- `model` : le modèle de base parmi les quatre offerts par YOLO variant selon la performance et la précision; nous avons choisi `yo1ov8s` qui offre un parfait équilibre entre les deux.
- `data` : l'emplacement du fichier parchemin `data.yaml` de la base de données.

- `epoch` : une époque, c'est lorsque la base de données entière est transmise vers l'avant et vers l'arrière via le réseau neuronal une seule fois.
- `imgsz` : La taille des images d'entrée.

```
!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25
imgsz=416
```

L'entraînement va prendre un bon bout de temps dépendant du nombre d'époques (`epoch`) spécifiées et la performance de la machine sur laquelle l'opération est effectuée. Le résultat de l'entraînement est enregistré dans l'emplacement `runs/detect/train` (voir **Fig 27**) et deux modèles sont enregistrés : `best.pt` , le modèle détenant le meilleur résultat de validation, et `last.pt` celui de la dernière itération (pour continuer l'entraînement) sous `runs/detect/train/weights` :

The screenshot shows a JupyterLab environment. On the left, a file explorer displays a directory structure: `pistol-1` contains `runs`, `detect`, and `train`. Inside `train`, there is a `weights` folder containing `best.pt` and `last.pt`. The main workspace shows three code cells:

```
[2] !pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="9HaUAzkQEtmgk")
project = rf.workspace("abhishekp")
dataset = project.version(1).download("pistol")

[5] !yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=416

!yolo task=detect mode=val model=weights/best.pt data={dataset.location}/data.yaml epochs=10 imgsz=416
```

The output of the validation cell shows a model summary for Ultralytics YOLOv8.0.200 (Python 3.10.12) with 168 layers and 1 class. It indicates scanning for images in the `valid` directory. The speed is reported as 1.3ms preprocess and 28.1ms inference.

Fig 27 - Dossier contenant les modèles

3.4.2.3. La matrice de confusion et résultats

Avant de tester le modèle avec des données étrangères, c'est-à-dire des images qui n'ont pas été utilisées pour entraîner le modèle, nous pouvons l'évaluer avec des statistiques générées par le processus d'entraînement (voir **Fig 28**).

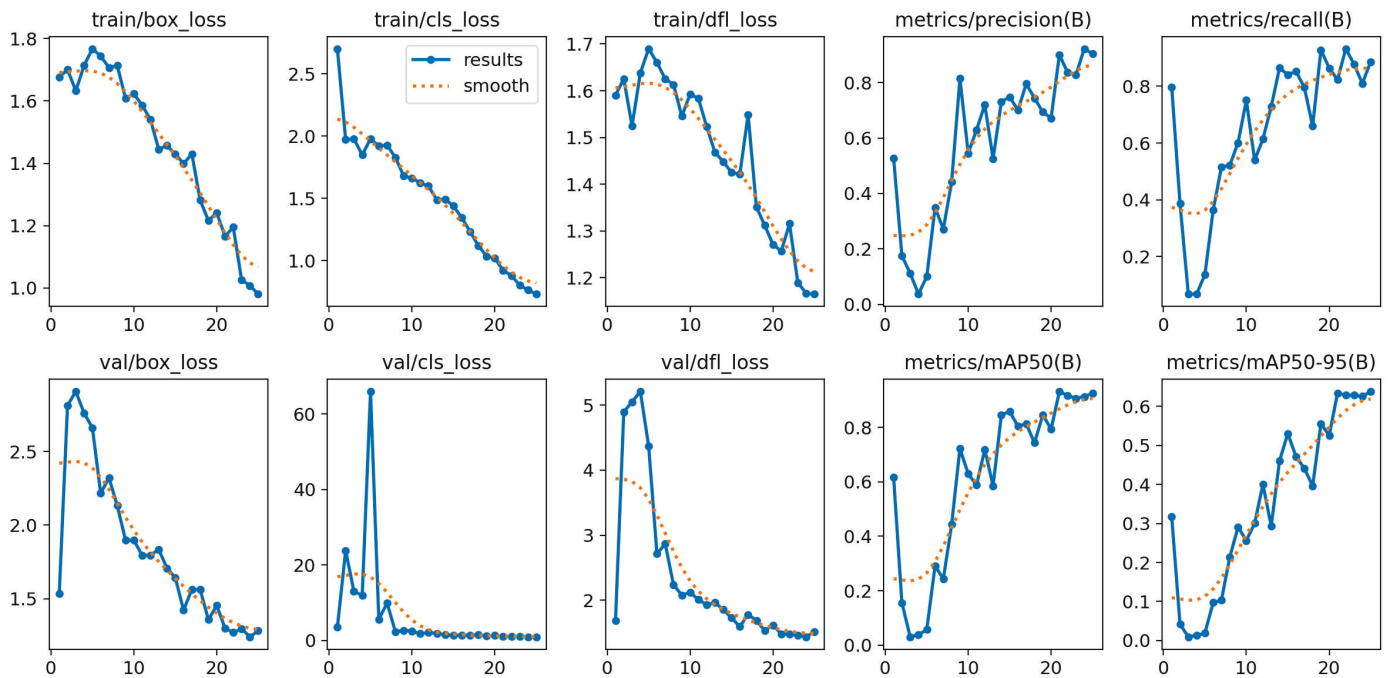


Fig 28 - Résultats

Nous sommes intéressés par `box_loss` et `cls_loss` de l'entraînement (`train`) et de la validation (`val`) représentant respectivement la perte en détection (délimitation) et en classification (prédiction de la classe correcte). Lorsque la courbe descend et donne l'impression de continuer à descendre, cela veut dire qu'en continuant l'entraînement on pourrait avoir de meilleurs résultats, une meilleure précision; lorsque la courbe donne l'impression de stagner, cela veut dire qu'on fatigue la machine en vain, on n'obtiendra très probablement (mais pas assurément) pas de meilleurs résultats.

Parmi les fichiers générés dans le dossier `train` nous pouvons repérer la matrice de confusion qui est également connue sous le nom de tableau de contingence. La matrice de confusion permet d'évaluer les performances d'un modèle de classification. Elle montre donc à quel point notre modèle peut être confus lorsqu'il fait des prédictions.

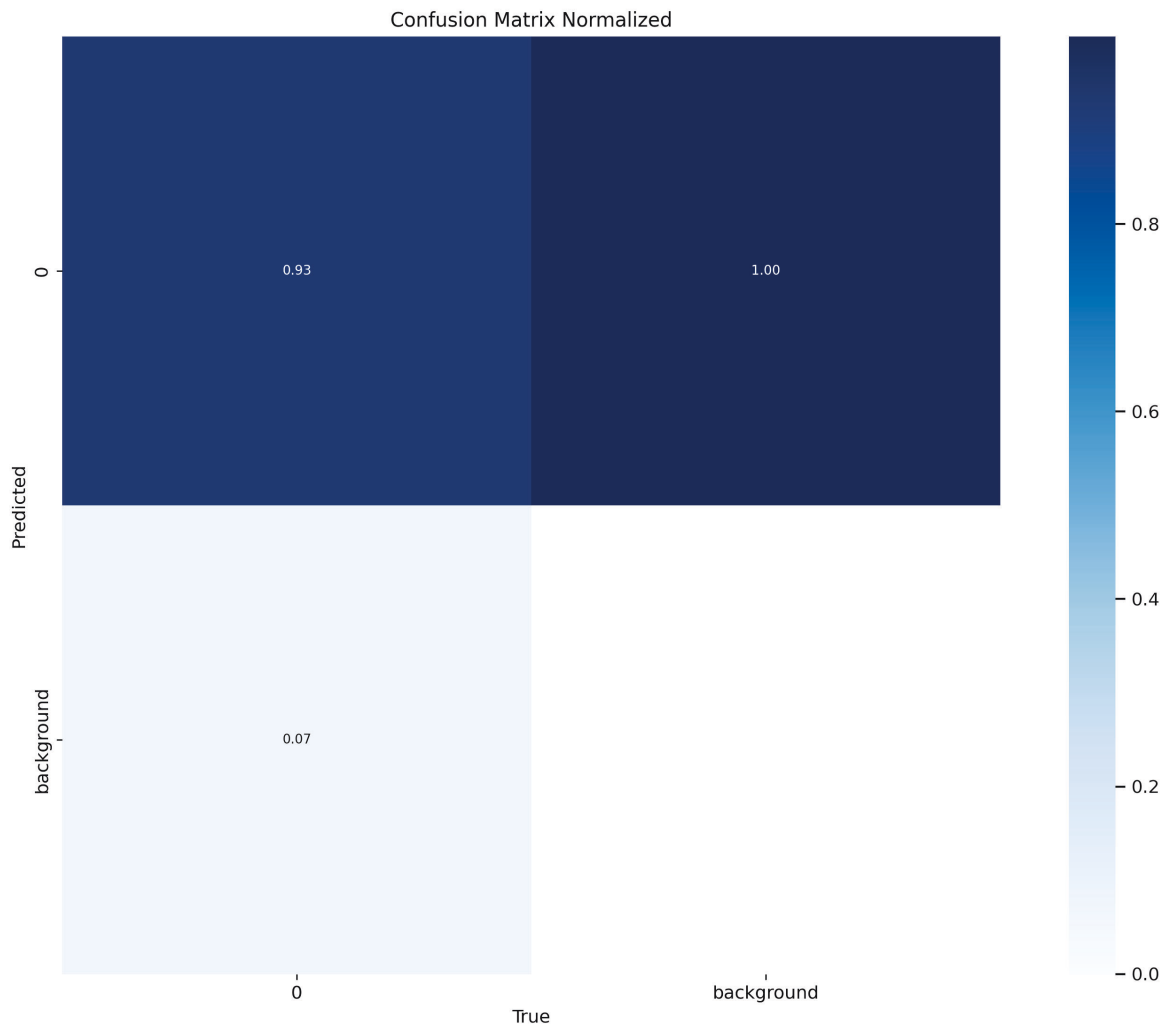


Fig 29 - Matrice de confusion

On lit sur cette matrice (voir Fig 29) que le pistolet est détecté 93% du temps, et pour 7% des fois il est confus avec l'arrière-plan, c'est-à-dire qu'il n'est pas détecté du tout.

3.4.2.4. Test conclusif

Pour la cerise sur le gâteau, nous allons finalement tester notre modèle sur une image étrangère. Nous allons créer un dossier dans l'explorateur de fichier correspondant et y placer nos images ou vidéos de test; dans notre cas, le dossier `/media`.

Voici notre image de test, une image aléatoire gratuite retrouvable sur Google Image (voir Fig 30) :



Fig 30 - Image de test, braquage

Pour effectuer une prédiction, nous allons démarrer YOLO en mode prédiction avec `mode=predict` en spécifiant le modèle avec l'option `model=chemin_vers_model`, le seuil de prédiction avec `conf`, et le chemin vers le dossier contenant nos images de test avec `source`, dans notre cas nous aurons :

```
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.25  
source=/content/media
```

Une fois que la commande a été exécutée et que la prédiction a été effectuée, nous retrouverons les images de prédictions dans l'emplacement `runs/detect/predict`. Voici la prédiction de notre image de test (voir Fig 31) :



Fig 31 - Prédiction sur l'image de test

Nous remarquons que le pistolet est détecté avec une probabilité ou confiance (`conf`) de 77%; considérant la qualité de l'image, la position et l'angle du pistolet sur l'image, nous sommes satisfaits du résultat.

Cette procédure peut être répétée avec une très large base de données contenant une centaine de milliers d'images présentant différentes armes pour une meilleure précision et une meilleure diversité de cette application.

3.4.3. Module 2 : Détection de chutes

Ce module de notre application détectera lorsqu'un individu fera une chute.

Ce module pourrait être développé suivant la procédure décrite lors du développement du premier module. En effet, tous les modules décrits dans ce projet peuvent être réalisés selon la procédure YOLO dans laquelle nous annotons les images présentant différentes circonstances (ex: **tombé** ou **pas tombé**) et ensuite nous entraînons un modèle capable d'effectuer la reconnaissance sur des images étrangères.

Nous avons découvert ce module existant dans un dépôt GitHub libre ³⁵ (open source) d'un jeune ingénieur en informatique nommé **Baruch Baksht**, et pour éviter de réinventer la roue, nous allons décrire la procédure qu'il a adoptée et ensuite nous testerons avec nos propres données.

3.4.3.1. Procédure

L'équipe de **Baksht** a d'abord commencé par suivre la procédure traditionnelle décrite en haut, mais ils n'ont trouvé que 500 images en ligne qui n'ont pas été suffisantes pour entraîner un modèle d'une haute précision; par conséquent, l'équipe a utilisé YOLO-Pose pour détecter conjointement les points clés du corps d'un individu sur une image numérique (ou vidéo).

YOLO-Pose qui est une méthode de détection de pose multi-personnes basée sur le framework de détection d'objets YOLO. Cette méthode utilise une approche sans heatmap (carte de chaleur) pour l'estimation de la pose 2D et se concentre sur l'élimination des post-traitements non normalisés qui sont actuellement utilisés. Elle utilise le même post-traitement que la détection d'objets.

Le programme se définit principalement ainsi :

- La fonction `get_pose_model()` retrouve le modèle à partir de YOLO-Pose qu'on peut télécharger [ici](#) :

```
def get_pose_model():
    device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
    print("device: ", device)
    weights = torch.load('yolov7-w6-pose.pt', map_location=device)
    model = weights['model']
    _ = model.float().eval()
    if torch.cuda.is_available(): # optimise si notre matériel présente CUDA
        model = model.half().to(device)
    return model, device
```

- Une fonction `get_pose` qui avec le modèle récupéré avec la fonction précédente va maintenant effectuer la détection de pose dans une image :

```

def get_pose(image, model, device):
    image = letterbox(image, 960, stride=64, auto=True)[0]
    image = transforms.ToTensor()(image)
    image = torch.tensor(np.array([image.numpy()]))
    if torch.cuda.is_available():
        image = image.half().to(device)
    with torch.no_grad():
        output, _ = model(image)
        output = non_max_suppression_kpt(output, 0.25, 0.65, nc=model.yaml['nc'],
nkpt=model.yaml['nkpt'],
                                     kpt_label=True)

    with torch.no_grad():
        output = output_to_keypoint(output)
    return image, output

```

- Et pour finir, la fonction `fall_detection()` prend en paramètre la pose de l'individu contenant les coordonnées des différents membres du corps et, par la suite, vérifie que les emplacements de ces derniers sont irréguliers ce qui signifierait qu'il y a chute :

```

def fall_detection(poses):
    for pose in poses:
        xmin, ymin = (pose[2] - pose[4] / 2), (pose[3] - pose[5] / 2)
        xmax, ymax = (pose[2] + pose[4] / 2), (pose[3] + pose[5] / 2)
        left_shoulder_y = pose[23]
        left_shoulder_x = pose[22]
        right_shoulder_y = pose[26]
        left_body_y = pose[41]
        left_body_x = pose[40]
        right_body_y = pose[44]
        len_factor = math.sqrt(((left_shoulder_y - left_body_y) ** 2 +
(left_shoulder_x - left_body_x) ** 2))
        left_foot_y = pose[53]
        right_foot_y = pose[56]
        dx = int(xmax) - int(xmin)
        dy = int(ymax) - int(ymin)
        difference = dy - dx
        if left_shoulder_y > left_foot_y - len_factor and left_body_y > left_foot_y -
(
            len_factor / 2) and left_shoulder_y > left_body_y - (len_factor / 2)
or (
            right_shoulder_y > right_foot_y - len_factor and right_body_y >
right_foot_y - (
            len_factor / 2) and right_shoulder_y > right_body_y - (len_factor /
2)) \
            or difference < 0:
            return True, (xmin, ymin, xmax, ymax)
    return False, None

```

Le reste des fonctions sont des fonctions de bases qui préparent les entrées et sorties des images et dessinent les contours.

3.4.3.2. Évaluation sur des données étrangères

Pour exécuter le programme localement, on commence d'abord par cloner le projet depuis le dépôt avec la commande suivante :

```
!git clone https://github.com/Y-B-Class-Projects/Human-Fall-Detection.git
```

Un dossier sera créé dans votre répertoire nommé `Human-Fall-Detection`, il contient le projet tel qu'il est dans le dépôt. On se repère dans le dossier avec l'instruction suivante :

```
cd /content/Human-Fall-Detection/
```

Ensuite, on installe toutes ses dépendances (toutes inscrites dans le fichier `requirements.txt`) avec la commande `pip` suivante :

```
!pip install -r requirements.txt
```

Téléchargeons le modèle YOLO-Pose officiel avec la commande suivante :

```
!wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7-w6-pose.pt
```

Ensuite, plaçons nos vidéos préférées sous `fall_dataset/videos` et finalement lançons le programme pour effectuer la détection avec la commande suivante :

```
!python main.py
```

Ce qui nous donne un résultat très satisfaisant prédisant correctement la totalité des images d'évaluation (voir **Fig 32**) :



Fig 32 - Détection de chute sur une vidéo personnelle

Après un bon nombre d'essais, nous remarquons que le programme détecte la chute à l'exact moment où elle est faite, et le modèle ne confond pas non plus l'action de s'asseoir et celle de faire une chute.

3.4.3.3. Analyse des résultats et critiques

L'équipe de **Baksht** a bien compensé son manque d'image par une solution futée qui principalement détecte les coordonnées des différentes conjonctures du corps humain et s'assure qu'elles sont placées à des distances relativement régulières entre elles, faute de quoi, le programme déclare qu'il y a effectivement une chute.

Après une intense évaluation, ils ont les résultats suivants :

Exactitude	Précision	Rappel	Score F1
81.18%	83.27%	83.58%	83.42%

Voici le contexte dans lequel se déclarent ces termes ci-dessus:

- **Précision** : La proportion de vraies prédictions positives effectuées par le modèle sur l'ensemble des prédictions positives effectuées (vraies et fausses).

$$\text{Précision} = \frac{\text{Vraies positives}}{\text{Vraies positives} + \text{Fausses positives}}$$

- **Exactitude** : La proportion de prédictions correctes effectuées (vraies positives et vraies négatives) par le modèle sur l'ensemble des prédictions effectuées.

$$\text{Exactitude} = \frac{\text{Vraies positives} + \text{Vraies négatives}}{\text{Nombre total de prédictions}}$$

- **Rappel** : La proportion de vraies prédictions positives effectuées par le modèle sur l'ensemble des instances positives réelles.

$$\text{Rappel} = \frac{\text{Vraies positives}}{\text{Vraies positives} + \text{Fausses négatives}}$$

- **Score F1** : La moyenne harmonique de la précision et du rappel. C'est une mesure qui combine à la fois la précision et le rappel pour fournir un score unique qui équilibre les deux métriques.

$$\text{Score F1} = 2 * \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Notez qu'à la date d'écriture du mémoire, YOLO-Pose est maintenant intégré avec YOLOv8 qu'on utilisé pour développer le premier module. Pose est une option dans YOLOv8 offrant différent niveaux de détection, et cette dernière est plus performante et précise, et s'effectue en deux lignes de code.

3.4.4. Détection de l'expression faciale

Ce module permet de détecter l'expression faciale d'une personne en prédisant l'émotion apparente sur le visage d'un individu dans une image. Une fois de plus, ce module peut être aussi effectué suivant la procédure YOLO

3.4.4.1. Bibliothèques

OpenCV

OpenCV (Open Computer Vision) est une très large bibliothèque open source qui met à la disposition du développeur un très grand nombre d'algorithmes de vision par ordinateur. Il joue un rôle majeur dans les opérations VPO en temps réel, qui sont très communes dans les systèmes actuels. En l'utilisant, on peut traiter des images et des vidéos pour identifier des objets, des visages ou même l'écriture manuscrite d'un humain.

Deepface

Deepface est une bibliothèque de reconnaissance faciale et d'analyse d'attributs faciaux légère pour Python. Elle est capable de reconnaître les visages humains dans les images numériques et d'analyser des attributs tels que l'âge, le genre, l'émotion et la race. Deepface est un cadre de reconnaissance faciale hybride qui enveloppe des modèles de pointe tels que VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, Dlib et SFace. La bibliothèque est développée par un groupe de chercheurs de chez Facebook.³⁶

3.4.4.2. Utilisation

Avant d'effectuer la détection, nous devons d'abord installer les bibliothèques requises selon les commandes suivantes :

```
!pip install opencv-python deepface
```

Ensuite, nous allons importer la bibliothèque `cv2` pour pouvoir charger l'image sur laquelle nous allons effectuer la détection, puis nous importerons `deepface` pour effectuer la détection :

```
import cv2
from deepface import DeepFace
# on charge l'image
img = cv2.imread('chemin_vers_image')
# et on effectue la détection avec la méthode analyze() du module Deepface
result = DeepFace.analyze(img, actions = ['emotion'])
```

La méthode `analyze()` de Deepface permet une analyse de haute précision des attributs faciaux d'un individu; parmi les options qu'elle offre, s'énumèrent la détection de l'âge, du genre, de la race et enfin de l'expression faciale (émotion); c'est pourquoi il est important de spécifier l'action qu'on souhaite effectuer grâce au paramètre `actions` qui prend un tableau de textes décrivant les actions souhaitées.

Notre variable `result` contient les résultats de la prédiction, présentant différents pourcentages de correspondance avec différentes émotions telles que la peur, la colère, la joie, la neutralité, le dégoût, la tristesse et la surprise.

Pour faire une démonstration, nous allons utiliser une photo aléatoire libre en ligne (voir **Fig 33**) :



Fig 33 - La peur

Et en lançant le programme avec cette image, nous obtenons le résultat suivant (explications en commentaires) :

```
[{
  'emotion': {
    'angry': 0.0007562022342958423, # 0% de correspondance avec la colère
    'disgust': 1.8804249643700717e-10, # 1.8% de correspondance avec le dégoût
    'fear': 99.45976134894856, # 99,46% de correspondance avec la peur
    'happy': 0.0047807766471241655, # 0% de correspondance avec la joie
    'sad': 3.518736800723614e-06, # 3.5% de correspondance avec la tristesse
    'surprise': 0.5346970485715299, # 0.5% de correspondance avec la surprise
    'neutral': 1.7967917857564861e-12 # et 1.7% de correspondance avec la neutralité
  },
  'dominant_emotion': 'fear', # l'émotion dominante est la peur
  'region': {'x': 181, 'y': 61, 'w': 164, 'h': 164} # le cadre délimitant le visage
}]
```

3.4.4.3. Remarques

Bien que la bibliothèque soit une technologie de dernier cri utilisé à Facebook, elle reste quand-même imparfaite; deepface peine des fois à détecter le visage de l'individu sur une image comme celle de la peur (voir Fig 21) .

Dans cette photo (voir Fig 21) où l'enfant place légèrement ses mains sur ses joues, Deepface refuse d'effectuer la détection d'émotion car il n'a pas reconnu un visage, nous donnant la réponse suivante :

```
ValueError: Face could not be detected.
Please confirm that the picture is a face photo
or consider to set enforce_detection param to False
```

Cependant en forçant la détection avec le paramètre `enforce_detection` de la méthode `analyse()` , nous recevons quand-même un résultat plus ou moins satisfaisant avec 70% de correspondance avec la peur :

```
[
  {
    "emotion": {
      "angry": 0.014971748498625875,
      "disgust": 2.988279702425089e-08,
      "fear": 69.55760354490323, # la peur
      "happy": 30.261299641287284,
      "sad": 0.02620306352990878,
      "surprise": 0.12154632998437642,
      "neutral": 0.018378474495787857,
    },
    "dominant_emotion": "fear", # l'émotion dominante est la peur
    "region": {"x": 0, "y": 0, "w": 1024, "h": 768},
  }
]
```

Même si le visage n'est pas détecté, Deepface arrive quand-même à reconnaître l'émotion, ce qui est l'objectif principal du module.

3.5. Conditions d'implémentation

Pour implémenter une application de la vision par ordinateur (ou toute application de l'apprentissage automatique) dans un environnement tel que l'espace public, il est nécessaire de prendre en compte deux grands paramètres que sont l'aspect technologique et l'aspect légal.

L'aspect **technologique** dictera le matériel nécessaire pour lancer le programme avec une très bonne performance et l'aspect **légal** étudiera les lois d'ordre public que pourrait effleurer notre programme.

3.5.1. Aspect technologique

Si vous avez déjà effectué des tâches VPO, ou n'importe quelle tâche d'apprentissage automatique d'ailleurs, vous savez sûrement à quel point elles sont exigeantes en termes de performance. En effet, l'entraînement des modèles est tellement coûteux que même les processeurs modernes peinent à en subvenir, et la tâche de prédiction non plus n'en fait pas exception. C'est pour cela que beaucoup de bibliothèques sont optimisés CUDA.

CUDA (acronyme anglais Compute Unified Device Architecture) est une interface de programmation d'applicative qui permet aux logiciels de faire usage de certains types de

processeurs graphiques (GPU) pour traiter les tâches à usage général. En de termes plus simples, en général, dans un ordinateur, presque toutes les instructions (tâches) sont traitées par le processeur central (CPU ou unité centrale de traitement) à part celles qui concernent vidéos ou applications tri-dimensionnelles qui sont par contre traitées par le processeur graphique de la carte graphique; certaines cartes graphiques, notamment celles qui sont activées CUDA, permettent de faire usage du processeur graphique pour traiter les instructions non visuelles, comme par exemple dans notre cas, l'entraînement d'un modèle qui, hyperboliquement parlant, dure une éternité pour s'accomplir usant un processeur central.

Tout récemment en 2016, Google, voyant l'intelligence artificielle digne d'un grand investissement pour le future, a créé un processeur spécial pour effectuer plus efficacement et avec efficience les tâches d'apprentissage automatiques profonds, le TPU. L'Unité de traitement tenseur (TPU en anglais pour Tensor Processing Unit) est un circuit intégré spécifique pour une application, il est spécialement conçu pour accélérer les systèmes d'intelligence artificielle par réseaux de neurones. Le TPU est plus efficace que le GPU et le CPU mais coûte plus cher.

La plateforme Google Colaboratory vous permet d'utiliser différents types de ressources pour effectuer vos tâches d'intelligence artificielle. Vous pouvez alterner entre le CPU, GPU et TPU pour avoir une bonne idée de la différence en termes de performance entre ces derniers.

Les GPU ont la capacité de diviser des problèmes complexes en milliers ou millions de tâches distinctes et de les résoudre toutes en même temps, tandis que les TPU ont été conçus spécifiquement pour les charges des réseaux neuronaux et ont la capacité de travailler plus rapidement que les GPU tout en utilisant moins de ressources.

3.5.2. Aspect légal

L'aspect légal concernant l'intelligence artificielle se fragmente en quatre sous-aspects qui causent chacune des inquiétudes à la société.

Responsabilité

Suite à une requête, l'IA peut prendre des décisions autonomes et prendre action ayant appris sur une énorme quantité de données. Lorsque les systèmes d'IA agissent et causent des dommages, déterminer la responsabilité devient une tâche complexe. Qui doit être tenu responsable des conséquences de ces actions ? Le développeur de l'IA, l'utilisateur ou l'intelligence artificielle elle-même ?

D'un point de vue juridique, lorsque les systèmes d'IA sont utilisés pour prendre des décisions, les contrats commerciaux doivent être adaptés pour souligner les obligations de responsabilité et

d'indemnisation, en cas de dommages, préjudices ou pertes découlant de ces décisions. La couverture d'assurance devrait être revue pour déterminer comment elle s'applique lorsque les pertes sont causées par les décisions prises par les systèmes d'IA.

Confidentialité et sécurité des données

La conformité aux réglementations sur la confidentialité des données, y compris la Loi sur la protection des renseignements personnels et les documents électroniques du Canada, reste essentielle, quelle que soit la technologie utilisée. La technologie de l'IA soulève des préoccupations uniques en matière de confidentialité. L'IA s'appuie fortement sur des ensembles de données massifs, impliquant souvent des informations personnelles sensibles. Lorsque les employés déploient des outils d'IA sur le lieu de travail, les politiques du lieu de travail doivent traiter de l'utilisation des informations personnelles et contrôler la manière dont les informations personnelles sont saisies, afin de garantir le respect des obligations en matière de sécurité et de confidentialité.

Préjugés et discrimination

Les systèmes d'IA peuvent, par inadvertance, perpétuer les préjugés présents dans leurs données d'apprentissage, ce qui entraîne des résultats discriminatoires. Des problèmes juridiques se posent lorsque l'IA est utilisée dans les décisions d'embauche, de prêt et de justice pénale, car elle peut avoir un impact disproportionné sur certains groupes. Il est essentiel d'établir des lignes directrices et des réglementations pour lutter contre les préjugés et la discrimination liés à l'IA afin de garantir l'équité et de prévenir les litiges juridiques. Amazon a dû se retirer de l'utilisation de l'IA pour prendre des décisions d'embauche lorsque le système s'est avéré clairement biaisé à l'encontre des candidates,³⁷ reflétant ainsi le biais inhérent à l'ensemble des données analysées par l'outil d'IA.

S'ajoute à cette liste, de nombreux autres problèmes légaux qui sont jusqu'à aujourd'hui un sujet de discussions pour les politiciens et agents du pouvoir législatif.

3.6. Conclusion

Juste après qu'on ait découvert que l'ordinateur pouvait acquérir la capacité de réfléchir pour lui-même et prendre des décisions informées d'une manière autonome, on se voit stupéfait par son acquisition du sens visuel. Eh oui, la machine peut maintenant voir et reconnaître ce qu'elle voit.

Cette nouvelle faculté se voit déjà utilisée à bon escient par les pouvoirs exécutifs du monde; du moyen-orient aux pays occidentaux, de l'est à l'ouest, les forces de l'ordre reconnaissent les

bienfaits que peuvent apporter la vision par ordinateur pour leur faciliter la tâche dans leur guerre contre le crime. La VPO contribue grandement à reconnaître les criminels, à reconnaître un crime et même à en prédire. Cette contribution est d'une grande importance, la VPO sauve déjà des vies et à rendre justice à des innocents.

L'utilisation de l'intelligence artificielle par les agents de sécurité publique reste cependant loin d'être parfaite; certaines fonctionnalités de l'IA qui sont très préférées comme celle de la reconnaissance faciale qui présente les failles de l'imprécision et de la préjudice, ou bien la fonctionnalité de la prédiction que la police utilise pour prédire les criminels susceptibles à récidiver, une fonctionnalité qui est sujet au risque de se baser sur des données loin d'être impeccables.

Ajouté à ce douteux et risqué usage de l'intelligence artificielle est un manque de recours à un bon potentiel de l'intelligence artificielle que nous avons jugé être indispensable pour une presque parfaite prédiction et prévention de crimes.

La reconnaissance d'outils criminels est un moyen à des fins heureuses que nous sommes ébahis de noter que la police ne profite pas de l'utilisation pour instaurer l'ordre dans les endroits publics. Pour mettre au service de la justice, nous avons décrit en détails la procédure pour développer cette technique grâce à un système VPO dernier cri appelé YOLO permettant d'annoter, d'entraîner et de prédire. En outre, nous suggérons deux autres techniques qui peuvent être développées de la même manière et qui sont presque tout aussi importantes : la détection de chute, qui appairée avec la détection d'arme indiquerait une très possible descente d'un individu, et la détection d'émotion, qui inviterait une surveillance sur un individu qui porte sur son visage une émotion dangereuse qu'est la colère ou une émotion alarmante qu'est la peur.

Toutefois, les applications d'apprentissage automatique sont très en demande en termes de ressources, que même la plupart des processeurs modernes peinent à effectuer des tâches d'un tel ordre. Pour remédier à cela, nous avons dû faire recours aux processeurs graphiques qui ne sont originellement produits que pour les vidéos et les applications en trois dimensions. La vision par ordinateur, plus que tout autre algorithme d'apprentissage automatique, est très coûteuse; la plupart des processeurs graphiques ne sont pas à la hauteur de telle tâche et ce qui conduit à la naissance de l'unité de traitement tenseur, un processeur qui est conçu spécifiquement pour les tâches d'apprentissage automatique.

La performance n'est cependant pas la seule inquiétude que nous cause l'implémentation de notre programme; le développement doit s'effectuer tout en se pliant aux lois du gouvernement qui accueille de tels logiciels. Ainsi, quelques questions se posent et restent sans réponses : Qui est à blâmer lors ce qu'un logiciel qui prend des décisions tout seul fait erreur ? Est-ce le développeur,

l'utilisateur ou l'IA elle-même ? Ensuite, quelles sont les actions légales à prendre pour permettre au programme de s'entraîner sur un si grand nombre de données sensibles qui s'avèrent nécessaires pour son bon fonctionnement ? et bien d'autres...

Opérer avec l'intelligence artificielle présente beaucoup de hauts et aussi quelques bas, mais l'opinion populaire reste toutefois la plus humanitaire, tous les moyens devraient être employés pour sauver des vies et rendre justice à celles perdues.

Conclusion générale

À moins que vous ayez vécu sous un caillou durant cette dernière quinzaine d'années, vous devez sûrement être au courant de l'invasion de l'intelligence artificielle. Aujourd'hui un logiciel est presque considéré obsolète s'il ne présente pas des fonctionnalités IA; on attend de toute application une quelconque sorte de prise de décisions autonomes. Eh oui, l'ordinateur qui ne servait qu'à suivre des instructions, est maintenant capable de générer ses propres instructions; grâce à cela, il s'est accaparé du sens de la vision et même de l'ouïe.

Grâce à l'apprentissage automatique, l'ordinateur devient capable d'apprendre de trois manières, soit supervisé par un homme, partiellement supervisé ou bien tout seul. Des efforts datant de plus d'un siècle qui ne furent point ménagés ni par les mathématiciens et ni par les informaticiens, pour reproduire artificiellement le système neuronal de l'homme et attribuer à la machine une faculté de raisonnement semblable à la sienne. Jusqu'à aujourd'hui le réseau neuronal reste un sujet de recherche dont le potentiel devient plus profond que jamais.

Avec l'arrivée des grands modèles de langage, les programmes d'intelligence artificielle sont maintenant capable de comprendre le contexte et le sentiment en adaptant le style conversationnel; ce qui a donné naissance à des technologies telles que ChatGPT qui est capable de proposer des réponses à toute sorte de questions, de compléter des phrases, de traduire des textes, d'écrire des articles et de tenir des conversations avec des humains sous n'importe quel ton.

En addition à sa capacité de raisonner, la machine est maintenant capable de voir et de reconnaître ce qu'elle voit. La vision par ordinateur réunit presque toutes les notions de l'apprentissage automatique pour copier le système visuel de l'homme pour reconnaître objets, personnes et paternes de la même façon que le fait un homme, à la différence que ce dernier bénéficie d'une vie entière d'apprentissage, et la machine ne sera aussi bonne à cette tâche que selon la qualité de son entraînement.

Toutes ces nouvelles cordes de l'arc, se voient progressivement utilisées par différents composants de la société pour améliorer le bon fonctionnement de cette dernière. Un composant majeur est la sécurité publique; dans plusieurs pays, la police tente son meilleur pour monter à bord du bateau de la technologie moderne et offrir le meilleur niveau de sécurité aux citoyens grâce à l'IA. En utilisant des applications IA telles que la reconnaissance faciale et la prédiction de données (pour prédire les récidivités), la police, sans doute, se dispense de l'erreur humaine, mais s'expose néanmoins à l'imperfection de l'IA. Pour remédier à cela, nous avons mis en relief certaines failles que présentent les technologies actuellement en usage, et nous avons proposé des solutions

appuyées par des technologies IA de pointe pour détecter les armes, les chutes et les expressions faciales.

Bien que l'intelligence artificielle soit bénéfique pour le monde, elle reste cependant un sujet de débats posés sur les plus grandes tables rondes opposant une fraction de la société à une autre. Les leaders de ce monde, quand-même impressionnés par cet exploit technologique, sont cependant réticents à l'idée de reposer les responsabilités de l'homme sur la machine.

Références

- 1 - Lark Editorial Team - [Objective Function](#).
- 2 - Google - [What is unsupervised learning?](#).
- 3 - [Everything you need to Know about Linear Regression](#).
- 4 - [Logistic Regression Tutorial for Beginners](#).
- 5 - Temi Babs (Coinmonks) - [The Mathematics of Neural Networks](#).
- 6 - Warren S. McCulloch & Walter Pitts (1943) - "A logical calculus of the ideas immanent in nervous activity" | [SpringerLink](#).
- 7 - Mansfield Merriman (1877) - "A List of Writings Relating to the Method of Least Squares".
- 8 - Stephen M. Stigler (1981) - "Gauss and the Invention of Least Squares".
- 9 - Hugh L. Carrey (1971). "Tribute to Dr. Frank Rosenblatt" (PDF).
- 10 - Marvin Minsky; seymour Papert (1969). *Perceptrons: An Introduction to Computational Geometry*.
- 11 - Paul Werbos (1974). [Beyond regression : new tools for prediction and analysis in the behavioral sciences](#).
- 12 - David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams (1986) , "Learning representations by back-propagating errors".
- 13 - Nida Shahid - [Applications of artificial neural networks in health care organizational decision-making: A scoping review](#).
- 14 - Devron - [Why is ReLU preferred over linear activation functions?](#).
- 15 - X Wang - [Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts](#).
- 16 - James Vincent (5 décembre 2022). "AI-generated answers temporarily banned on coding Q&A site Stack Overflow". *The Verge*.

- 17 - James Vincent (5 janvier 2023). "Top AI conference bans use of ChatGPT and AI language tools to write academic papers". *The Verge*.
- 18 - Rachel Curry (13 juin 2023). "Samsung among companies starting to draft ChatGPT policies for workers". *CNBC*.
- 19 - Elon Musk: Advanced AI Is a Risk to the Public - YouTube *Bloomerg*.
- 20 - John Koetsier (14 mars 2023) GPT-4 Beats 90% Of Lawyers Trying To Pass The Bar.
- 21 - Bethany Biron (9 Janvier 2023) New York City schools ban use of ChatGPT — becoming first US district to block AI technology as concern over cheating and plagiarism mounts.
- 22 - Productive Teaching Tool or Innovative Cheating? (2023) | Study.com.
- 23 Forest Knight - Stack Overflow is dead (2023) - YouTube.
-
- 24 - Homicide in Canada, 2021.
- 25 - Christopher Rigano - "Using Artificial Intelligence To Address Criminal Justice Needs" *NIJ Journal* 280, January 2019
- 26 - Number of missing person files U.S. 1990-2022 - Statista
- 27 - Imran Rahman-Jones & Liv McMahon - AI facial recognition: Campaigners and MPs call for ban (2023)
- 28 - Reuters Staff - Robocop joins Dubai police to fight real life crime | Reuters
- 29 - Valena Elizabeth Beety, What the Brain Saw: The Case of Trayvon Martin and the Need for Eyewitness Identification Reform, 90 *DENV. U. L. REV.* 331, 341 (2012).
- 30 - Vanessa Van Edwards - *The Definitive Guide to Reading Microexpressions (Facial Expressions)* (2020).
- 31 - YOLO (You Only Look Once)
- 32 - ImageNet.
- 33 - Rohit Kundu - YOLO Algorithm for Object Detection Explained [+Examples].

- 34 - [AbhishekParth - pistol Object Detection Dataset | Roboflow.](#)
- 35 - [GitHub - Y-B-Class-Projects/Human-Fall-Detection: Human Falling Detection.](#)
- 36 - [DeepFace: Closing the Gap to Human-Level Performance in Face Verification.](#)
- 37 - [Jeffrey Dastin - Amazon ditches AI recruiting tool that didn't like women - National | Globalnews.ca](#)