

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي



جامعة مولاي طاهر سعيدة
كلية التكنولوجيا
قسم الاعلام الالي

Mémoire de Master

Spécialité : MICR

Thème

Transfert Learning pour la
reconnaissance du langage
des signes

Présenté par :

OUICI AMIRA IBTISSEM.
CHIBANI DJIHANE.

Dirigé par :

Dr. RAHMANI MOHAMED EL_HADI.



Promotion 2021 - 2022

ملخص

فكرة هذا المشروع تكمن في استعمال الذكاء الاصطناعي في الحل لبناء نظام ذكي مبني على أسس التعلم العميق و استخدام مختلف أدوات هذا الأخير لعلاج مشكل الأشخاص غير القادرين على التحدث و إنشاء نظام يحل لغة الإشارة و لعلاج هذا المشكل نمر بعدة اجزاء

1. استعمال بيانات مجمعة و منظمة لايادي اشخاص لتحليل كلمات لغة الإشارة
2. استعمال نوع Inception v3 من keras لتشفير الصورة و تحويلها إلى كتابة يستطيع من خلالها كل من الشخص العادي و الغير القادر على الكالم التواصل.

Abstract

The idea of this project resides in the use of artificial intelligence for the purpose to develop an intelligent system based on the bases of deep learning and using different tools of this last one for training the language of sings for hard hearing people (people unable to talk) and create a system that analyses the language of sings. in order to treat this problem, we should pass through many steps to resolve it.

- 1) utilize existing data and aggregate different people to analyze words of the language of signs.
- 2) utilize the inception type v3 Keras to code the image and convert it to writing in which an ordinary person, as well as people without a voice, can communicate.

Résumé

L'idée de ce projet réside dans l'utilisation de l'intelligence artificielle dans le but de développer un système intelligent basé sur les bases du deep Learning et utiliser les différents outils de ce dernier en traitant de la langue des signes pour les malentendants (des personnes incapables de parler) et de créer un système qui analyse la langue des signes et pour traiter ce problème nous passons par plusieurs parties :

1. Utiliser les données existantes et agrégées de différentes personnes pour analyser les mots de la langue des signes.
2. Utiliser le type Inception v3 de keras pour coder l'image et la convertir en une écriture à travers laquelle la personne ordinaire et la personne sans voix peuvent communiquer.

REMERCIEMENT :

Nous rendons grâce à Dieu tout puissant pour ses grâces, le souffle de vie, la santé, l'intelligence et le courage qu'il nous a donné et qui nous a accompagné tout le long de cette année d'étude jusqu'à l'aboutissement de ce travail. La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

Je voudrais tout d'abord adresser toute ma gratitude à mon encadreur, DR RAHMANI MOHAMED ELHADI Pour m'avoir encadré et dirigé sur la voie du succès avec ses recommandations, sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion. Je tiens à remercier profondément les membres du jury qui m'ont fait l'honneur de juger ce travail. Merci à Dr BOUARARA HADJ AHMED. Je tiens aussi à remercier l'ensemble de la direction du département Mathématiques et Informatique de l'université DR MOULAY TAHAR SAIDA pour m'avoir donné l'opportunité d'effectuer ma formation de master au sein du département.

Je dédie ce travail,

À mon grand-père, autant de phrases et d'expressions éloquentes sans elles ne sauraient exprimer ma gratitude et ma reconnaissance., tu as su m'inculquer le sens de la responsabilité, de l'optimisme et de la confiance en soi face aux difficultés de la vie., les meilleurs pour la fin, je remercie chaleureusement ma Grand-Mère, mes parents et ma famille : ADEL, KARIM, WISSAL pour leurs encouragements, leurs amours inconditionnels et leurs précieux conseils. J'aurais bien aimé que vous soyez à côté de moi le jour de ma soutenance. Je remercie également ma grande sœur ANISSA d'avoir été un grand soutien inébranlable. Mes derniers remerciements vont à ma chère amie CHAIMAA BENTADJ, tu n'as cessé de me soutenir et de m'encourager durant toutes les années de mes études, tu as toujours été présente à mes côtés pour me consoler quand il fallait.

REMERCIEMENT :

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je voudrais tout d'abord adresser toute ma reconnaissance au directeur de ce mémoire, monsieur Elhadi rahmani, pour son patience, son disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je désire aussi remercier tout les professeurs de département de l'informatique, qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires.

Je tiens à remercier spécialement Mes parents, pour tout.

Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de ma démarche.

Enfin, merci a Moi ...

CHIBANI DJIHENE

SOMMAIRE :

Introduction	10
1. L'histoire de la langue des signes:.....	10
2. Les premières langues des signes	11
3. Les philosophes grecs:.....	11
3. Enseignants en Italie et en Espagne	12
4. La révolution de la langue des signes française	13
5. Signe fleurissant dans le Nouveau Monde	14
6. Technologies d'intelligence artificielle pour la langue des signes:.....	15
abstraite.....	15
Introduction	15
7. dernières recherches sur la langue des signes.....	19
Conclusion	22
Deep Learning	24
Introduction	24
1. L'IA dans le monde professionnel.....	24
2. Domaine du Deep Learning.....	25
3. Deep Learning et IA.....	25
4. Le Deep Learning comme solution dans le e-commerce.....	26
5. L'histoire du Deep Learning	26
6. Les algorithmes de Deep Learning.....	27
7. Transfert Learning.....	28
Définition 28	
Neurones artificiels.....	32
Architecture.....	32
9. L'apprentissage	34
APPRENTISSAGE SUPERVISÉ.....	34
CLASSIFICATION.....	34

APPRENTISSAGE NON SUPERVISÉ	34
10. Applications des réseaux de neurones de classification.....	35
11. Les réseaux de neurones convolutifs.....	35
Architecture d'un Convolutional Neural Network-CNN.....	36
Partie convolutive	37
Couche de convolution.....	37
Méthode de sous échantillonnage : le Max-Pooling.....	39
Couche d'échantillonnage (Pooling).....	39
Conclusion.....	41
Introduction.....	43
Environnement du travail.....	43
Python.....	43
La version actuelle de Python.....	44
Keras.....	44
Le fonctionnement du keras.....	45
Les avantages du Keras.....	46
La destination du keras et Tensorflow_keras dans les projets.....	46
NumPy.....	47
Pandas.....	47
Matplotlib Visualisation avec Python.....	47
1.6.OS.....	48
1.7. TensorFlow	48
2. À propos de l'ensemble de données:.....	51
3. L'approche proposée.....	51
Inception.....	51
Pourquoi Inception V3	52
Description du modèle	53
4. Ensemble de données ImageNet	55
4. Pipeline d'entrée	56
Étape de prétraitement.....	60
5. Résultats obtenus	62
6. Etude comparative	66
Conclusion	68

LISTE DES TABLEAUX

Tableau 1.resultat obtenue.....	61
Tableau 2.Etude comparative.....	66

LISTE DES FIGURES :

Figure 1. langage de signe	17
Figure 2. DETECTION AUTOMATIQUE D'UNE FRACTURE A L'AIDE DU DEEP LEARNING.....	25
Figure 3. Approche traditionnelle vs Approche de transfert Learning	29
Figure 4. Exemple de modèle de réseau de neurones utilisés pour la classification d'images	30
Figure 5. neurone	31
Figure 6. neurone	32
Figure 7. le réseau de neurones à un seul niveau	33
Figure 8. Le réseau de neurones à multi niveaux	33
Figure 9. Architecture d'un réseau de neurones convolutifs	36
Figure 10. Schéma représentant l'architecture d'un CNN.....	37
Figure 11. schéma du parcours de la fenêtre de filtre sur l'image.....	37
Figure 12. Exemple d'une opération de Pooling taille 2*2	40
Figure 13. tensorflow	49
Figure 14. inception v3	53
Figure 15. shema générale inception v3	53
Figure 16. pipeline d'entrée	56
Figure 17. TPU.....	56
Figure 18. performance inception v3.....	58
Figure 19. performance inception v3.....	58
Figure 20. performance inception v3	59
Figure 21. performance inception v3.....	59
Figure 22. la précision de l'apprentissage.....	62
Figure 23. valeurs d'entropie	63
Figure 24. valeurs de la précision.....	64

Introduction générale :

Certains des principaux problèmes rencontrés par une personne incapable de parler sont qu'elle ne peut pas exprimer ses émotions aussi librement dans ce monde. Utilisez ces systèmes de reconnaissance vocale et de recherche vocale dans les Smartphones. Les résultats audio ne peuvent pas être récupérés. Ils ne peuvent pas utiliser (l'intelligence artificielle/le majordome personnel) comme l'assistance Google, ou le SIRI d'Apple, etc., car toutes ces applications sont basées sur le contrôle vocal. Il y'a un besoin pour de telles plateformes pour ce genre de personnes. La langue des signes américaine (ASL) est une langue complète et complexe qui utilise des signes faits en déplaçant les mains combinés avec des expressions faciales et des postures du corps. C'est la langue de prédilection de nombreux Nord-Américains qui ne peuvent pas parler et c'est l'une des diverses alternatives de communication utilisées par les personnes sourdes ou malentendantes. Alors que la langue des signes est très essentielle pour les personnes sourdes-muettes, communiquer à la fois avec des personnes normales et avec elles-mêmes, reçoit toujours moins d'attention de la part des personnes normales. L'importance de la langue des signes a eu tendance à être ignorée, à moins qu'il y'ait des sujets de préoccupation avec les personnes sourdes-muettes. Une des solutions pour parler avec les sourds-muets est d'utiliser les mécanismes de la langue des signes. Le geste de la main est l'une des méthodes utilisées en langue des signes pour la communication non verbale. Il est le plus couramment utilisé par les personnes sourdes et muettes qui ont des troubles de l'ouïe ou de la parole pour communiquer entre elles ou avec des personnes normales. Divers systèmes de langue des signes ont été développés par de nombreux fabricants à travers le monde, mais ils ne sont ni flexibles ni rentables pour les utilisateurs finaux, Langage de signe est créer est pour les gens qui sont pas capable a parlé il suffit de faires des gestes pour nous comprendre mais ce n'est pas donnée a tout le

monde c'est juste les personnes qui connues la langue de signe qui peuvent les contacter, dans ce cas il y a un problème entre les personnes qui ne sont pas capables de parler et qui sont capables. Les statistiques disent qu'une personne sur cent présente une limitation importante ou complète de l'ouïe dans le monde, alors que 13 % de la population est malentendante. Dans le premier chapitre, on a défini le langage des signes ensuite on a parlé de son historique, son histoire ainsi que comment utiliser l'intelligence artificielle afin de résoudre cette problématique. Dans le deuxième chapitre, on a présenté une phase de l'intelligence artificielle qui est le deep Learning (apprentissage en profondeur), les réseaux de neurones convolutifs. Dans le troisième chapitre, on a cité les différents outils, le modèle afin de réaliser ce travail, engendrant ainsi les résultats obtenus.

CHAPITRE 01

L langage des signes

Introduction :

La reconnaissance de la langue des signes est une percée pour aider les personnes sourdes-muettes et fait l'objet de recherches depuis de nombreuses années. Malheureusement, chaque recherche a ses propres limites et ne peut toujours pas être utilisée commercialement. Certaines des recherches sont connues pour réussir à reconnaître la langue des signes, mais nécessitent un coût élevé pour être commercialisées. De nos jours, les chercheurs ont accordé plus d'attention au développement de la reconnaissance de la langue des signes qui peut être utilisée commercialement. Les chercheurs mènent leurs recherches de différentes manières. Cela commence par les méthodes d'acquisition de données. La méthode d'acquisition de données varie en raison du coût nécessaire pour un bon appareil, mais une méthode bon marché est nécessaire pour que le système de reconnaissance de la langue des signes soit commercialisé. Les méthodes utilisées pour développer la reconnaissance de la langue des signes varient également d'un chercheur à l'autre. Chaque méthode a sa propre force par rapport aux autres méthodes et les chercheurs utilisent toujours différentes méthodes pour développer leur propre reconnaissance de la langue des signes. Chaque méthode a également ses propres limites par rapport aux autres méthodes. Le but de cet article est de passer en revue les approches de reconnaissance de la langue des signes et de trouver la meilleure méthode qui a été utilisée par les chercheurs. Par conséquent, d'autres chercheurs peuvent obtenir plus d'informations sur les méthodes utilisées et pourraient développer de meilleurs systèmes d'application de la langue des signes à l'avenir.[1]

Mots clés: reconnaissance de la langue des signes application, de la langue des signes entrée de données sourd mettre en sourdine

1. L'histoire de la langue des signes:

Les langues des signes existent depuis bien plus longtemps que la plupart des gens ne le pensent. Saviez-vous qu'ils existaient dans la Grèce antique ? Et même avant l'histoire enregistrée? Pour vous donner une idée de l'incroyable et prolifique langue des signes, plongeons dans la longue et colorée histoire de la naissance des signes, et de l'ASL en particulier.

2. Les premières langues des signes

Personne ne sait *exactement* quand la langue des signes est apparue pour la première fois, mais de nombreuses sources s'accordent à dire que l'utilisation des mains pour communiquer existe depuis aussi longtemps que la langue parlée. Et ces premiers systèmes de signature étaient le résultat direct du fait que les humains avaient besoin d'une nouvelle façon d'interagir. Les chercheurs pensent que les chasseurs des plaines ouvertes utilisaient des signes pour communiquer entre eux à de grandes distances. En raison de l'absence d'obstruction visuelle dans un environnement de plaines, le signe était le moyen le plus évident de communiquer sans effrayer les animaux qu'ils chassaient.

Les anciens Amérindiens des Grandes Plaines ont également développé un système de signature complexe. On ne sait pas exactement à quoi servait le système, mais de nombreuses théories différentes existent. Un signe populaire est que le signe a rendu possible le commerce intertribal. Pour surmonter les barrières linguistiques, les indigènes ont développé un système standardisé de gestes de la main pour négocier avec les tribus qui ne parlaient pas leur langue, y compris les expéditionnaires européens. Plusieurs récits du débarquement de Christophe Colomb dans les Amériques affirment que les indigènes communiquaient avec son équipage par signe.

3. Les philosophes grecs:

Il est impossible de savoir exactement quand et où la première personne sourde a essayé le signe, mais nous savons que la première trace écrite de la langue des signes provient de la Grèce antique. Au Ve siècle av. J.-C., le philosophe Platon écrivit le dialogue *Cratyle*. Il y enregistra Socrate disant : « Si nous n'avions ni voix ni langue, et que nous voulions pourtant nous manifester les choses les uns aux autres, ne devrions-nous pas, comme ceux qui sont actuellement muets, nous efforcer de signifier ce que nous voulons dire par nos mains, notre tête et d'autres parties du corps ? Apparemment, les anciens Grecs qui ne pouvaient pas parler avaient en effet une langue des signes rudimentaire pour vaquer à leurs occupations quotidiennes.

Plus tard, l'élève de Platon, Aristote, est devenu la première personne à enregistrer une affirmation sur les personnes sourdes - et malheureusement, ce n'était pas une bonne affirmation. Il croyait que pouvoir entendre la parole était le seul moyen d'apprendre. Ainsi, selon Aristote, il était totalement impossible d'éduquer les sourds. Même s'il n'y avait pas la moindre preuve factuelle pour étayer son affirmation, la théorie d'Aristote s'est imposée et a été largement crue pendant les 2000 années suivantes à travers le monde.

Et les résultats n'étaient pas jolis. À cette époque de l'histoire, les personnes sourdes étaient considérées comme des êtres humains inférieurs qui ne pouvaient légalement détenir des biens. Ils ne pouvaient pas se marier car la société avait peur que la surdité soit un trait héréditaire qui serait transmis aux générations futures. Les personnes sourdes se sont souvent vu refuser la citoyenneté et même les droits religieux. Et si la surdité était considérée comme un handicap honteux, toute forme de signe était ostracisée et découragée, rendant presque impossible pour ces personnes de communiquer librement.

Les érudits de cette période croyaient sincèrement que les personnes sourdes ne pouvaient pas apprendre, mais certains enseignants ont quand même essayé. En 685

après JC, l'archevêque d'York, John Beverly, a appris à parler à un garçon sourd . Mais au lieu de voir cet accomplissement comme la preuve qu'Aristote avait tort, les penseurs de l'époque considéraient cet acte comme divin. L'archevêque a ensuite été canonisé pour avoir accompli le miracle, mais les gens croyaient toujours que la seule façon de « vaincre » la surdité était de parler la même langue que la population générale.

3. Enseignants en Italie et en Espagne

Au XVI^e siècle , les philosophes et les enseignants ont finalement commencé à remettre en question l'affirmation d'Aristote selon laquelle les personnes qui ne pouvaient pas entendre ne pouvaient pas être éduquées. Un médecin et mathématicien italien nommé Girolamo Cardano (également connu sous le nom de Gerolamo ou Geronimo) a été la première voix à contester l'hypothèse de longue date d'Aristote. Cardano a affirmé que l'ouïe n'était pas nécessaire pour qu'une personne comprenne des idées et a même commencé à développer son propre code de gestes de la main. Il croyait que l'on pouvait utiliser des mots écrits assortis de symboles de ce qu'ils représentaient pour communiquer avec les étudiants sourds. Bien que son code n'ait jamais été largement adopté, il a utilisé ses méthodes pour enseigner à son propre fils sourd. Et les théories de Cardano ont grandement influencé d'autres dirigeants et penseurs de l'époque.

À peu près à la même époque que Cardano (vers 1570), un moine espagnol du nom de Pedro Ponce de Leon a commencé à éduquer ses propres étudiants sourds , les fils de nobles espagnols. Parce qu'ils étaient sourds, ces jeunes hommes n'avaient pas le droit d'hériter de biens. Leon leur a appris à lire, écrire et parler afin qu'ils puissent revendiquer la fortune familiale qui leur appartenait de droit. Et ses efforts ont été couronnés de succès.

Cardano et Leon ont inspiré un autre moine espagnol nommé Juan Pablo de Bonet à faire le plus grand pas dans l'histoire de la langue des signes. Après avoir développé ses propres méthodes d'éducation des élèves sourds, Bonet publie le premier livre sur la langue des signes en 1620. Il y inclut son propre alphabet manuel de formes de mains représentant des sons. Ce fut le premier système publié d'orthographe digitale dans l'histoire.

Même si ces premiers systèmes ont été conçus pour apprendre aux personnes sourdes à parler d'autres langues, le livre de Bonet était toujours un jalon révolutionnaire dans le développement de la langue des signes en tant que forme de communication officiellement reconnue. Son livre a suscité un intérêt dans toute l'Europe pour l'éducation des étudiants sourds, mais ce n'est qu'au milieu des années 1700 que la prochaine réalisation révolutionnaire dans le développement de la langue des signes a eu lieu.

4. La révolution de la langue des signes française

L'abbé Charles Michel de l'Eppe était un prêtre catholique français qui a étudié la théologie et le droit. Après avoir été ordonné, l'Eppe a consacré sa vie à s'occuper des Français pauvres, et c'est au cours de ce service dans les bidonvilles de Paris qu'il a rencontré deux sœurs jumelles, toutes deux sourdes. Un autre ecclésiastique avait été

le tutorat des filles mais est décédé très soudainement. L'Eppe est intervenu en tant que nouveau professeur, une décision qui a commencé sa mission de toute une vie de servir et d'éduquer la population sourde de France.

La communauté sourde française utilisait déjà une langue des signes commune à Paris, une langue qui s'était développée de manière organique au fil des siècles. L'Eppe a complété ce système de l'ancienne langue des signes française en créant une série de signaux manuels pour remplacer les sons de l'alphabet . Alors qu'il enseignait aux jumeaux, L'Eppe a découvert une percée dans l'éducation des sourds : que les sourds apprennent visuellement les mêmes choses que les autres apprennent en entendant. Les personnes sourdes et muettes avaient déjà une langue qui était tout aussi puissante et expressive que le français parlé, et la clé pour les éduquer était de les entraîner à communiquer avec leurs mains plutôt qu'avec leur voix.

En 1760 , l'Eppe fonde la première école publique gratuite de langue des signes au monde, financée par son propre héritage. L'école s'appelait Institution Nationale des Sourds-Muets à Paris (The Royal Institution of Deaf-Mutes). Alors que le système des signes français et les méthodes d'enseignement de l'Eppe continuaient à se développer, des sourds de toute la France affluaient vers son école. Même les responsables d'autres pays ont commencé à s'en apercevoir. L'empereur d'Autriche et l'impératrice de Russie ont tous deux envoyé des enseignants pour apprendre le style d'enseignement de l'Eppe , et son influence a finalement conduit à la création de vingt et une écoles au total en France et dans de nombreux autres pays.

Bien sûr, l'Eppe n'était pas le seul professeur de langue des signes influent de cette période. En Angleterre ,Thomas Braidwood créait la Braidwood'sAcademy for the Deafand Dumb à peu près au même moment où l'école de l'Eppe ouvrait en France. Même si l'école de Braidwood n'éduquait que les enfants sourds de familles plus riches, Braidwood a joué un rôle central dans le développement du style britannique de la langue des signes en conjonction avec l'enseignement de la parole à ses élèves.

Mais tous les enseignants de l'époque n'acceptaient pas de signe. Samuel Heinicke a fondé la première école allemande pour sourds en 1778, mais contrairement à l'Eppe, Heinicke était un orateur convaincu. Il croyait à tort que la fonction première de l'éducation des enfants sourds devrait être de développer leurs compétences en langage parlé afin qu'ils puissent s'intégrer pleinement dans la société entendante. C'est le seul domaine où l'influence de l'Eppe s'est démarquée parmi les autres enseignants européens.

L'Eppe a vraiment été le premier enseignant "manualiste", le premier leader de l'éducation des sourds qui a réalisé que la langue des signes était la façon dont les sourds *devaient* communiquer, et pas seulement comme un véhicule pour les aider à parler les langues orales. En plus de perpétuer l'importance du signe, la formation unique de l'Eppe en théologie et en droit a également fait de lui un allié précieux pour les droits des sourds à la fois dans la religion et dans la salle d'audience. Il a été l'une des premières personnes de l'histoire à affirmer publiquement que les personnes sourdes méritaient d'être traitées comme des êtres humains pleinement fonctionnels avec quelque chose de significatif à apporter à la société, même si elles parlaient une langue différente. Il n'est pas étonnant qu'aujourd'hui l'Eppe soit connu comme le « père des sourds ».

5. *Signe fleurissant dans le Nouveau Monde:*

En 1620, exactement la même année que Bonet publiait le premier livre sur la langue des signes, le Mayflower débarqua à Cape Cod, Massachusetts. Ce fut la première de nombreuses vagues de colons naviguant vers le Nouveau Monde à la recherche d'une vie meilleure. Parmi les nombreux immigrants qui ont sauté sur des navires à destination de l'Amérique, il y avait un groupe de colons originaires du comté de Kent en Angleterre. Ils sont arrivés dans le Massachusetts dans les années 1690, apportant avec eux un défaut génétique unique qui provoque une surdité héréditaire. Le résultat a été la langue des signes de Martha's Vineyard.

Martha's Vineyard est une île au large de la côte sud-est du Massachusetts, et c'est un lieu historiquement important pour le développement de l'enseignement. Grâce à la composition génétique unique des ancêtres de l'île, Martha's Vineyard abritait la plus grande population sourde du continent. Ces insulaires sourds ont créé l'un des systèmes de signes les plus développés de l'histoire, peut-être basé sur une langue utilisée dans le Kent. Tout le monde sur Martha's Vineyard connaissait les panneaux et les utilisait régulièrement pour communiquer avec ses amis, sa famille et ses voisins. Parce que même les membres entendants de cette communauté utilisaient la langue des signes, Martha's Vineyard était l'une des rares sociétés au monde où les personnes sourdes et entendants étaient pleinement intégrées ensemble dans toutes les activités sociales, civiques et religieuses au milieu des années 1700.

Mis à part le célèbre signe Martha's Vineyard, d'autres versions de la langue des signes ont commencé à apparaître partout en Amérique partout où des populations sourdes existaient. Au XIXe siècle, les données du recensement de l'église rapportaient qu'environ 800 enfants sourds vivaient aux États-Unis. Mais la langue des signes était encore loin d'être normalisée et même loin d'être reconnue comme langue officielle. Ce n'est que des décennies plus tard que l'éducation des sourds américains fera un pas en avant important et décisif.[2]

6. *Technologies d'intelligence artificielle pour la langue des signes:*

abstraite:

Les technologies de l'IA peuvent jouer un rôle important dans la suppression des barrières de communication des personnes sourdes ou malentendantes avec d'autres communautés, contribuant ainsi de manière significative à leur inclusion sociale. Les progrès récents des technologies de détection et des algorithmes d'IA ont ouvert la voie au développement de diverses applications visant à répondre aux besoins des communautés sourdes et malentendantes. À cette fin, cette enquête vise à fournir un examen complet des méthodes de pointe en matière de capture, de reconnaissance, de traduction et de représentation de la langue des signes, en soulignant leurs avantages et leurs limites. En outre, l'enquête présente un certain nombre d'applications, tout en discutant des principaux défis dans le domaine des technologies de la langue des signes.

Mots-clés : reconnaissance de la langue des signes, représentation de la langue des signes, capture de la langue des signes, applications

Introduction:

La langue des signes (SL) est le principal moyen de communication entre les personnes malentendantes et les autres communautés et elle s'exprime par des caractéristiques manuelles (c'est-à-dire les mouvements du corps et des mains) et non manuelles (c'est-à-dire les expressions faciales). Ces caractéristiques sont combinées pour former des énoncés qui transmettent le sens des mots ou des phrases [3]. Être capable de saisir et de comprendre la relation entre les énoncés et les mots est crucial pour la communauté sourde afin de nous guider vers une ère où la traduction entre les énoncés et les mots peut être réalisée automatiquement [4]. La communauté des chercheurs a depuis longtemps identifié la nécessité de développer des technologies de langue des signes pour faciliter la communication et l'inclusion sociale des personnes malentendantes. Bien que le développement de telles technologies puisse être très difficile en raison de l'existence de nombreuses langues des signes et du manque de grands ensembles de données annotées, les progrès récents de l'IA et de l'apprentissage automatique ont joué un rôle important dans l'automatisation et l'amélioration de ces technologies.

Les technologies de la langue des signes couvrent un large spectre, allant de la captation des signes à leur représentation réaliste afin de faciliter la communication entre personnes malentendantes, ainsi que la communication entre personnes malentendantes et personnes parlantes. Plus précisément, la capture de la langue des signes implique l'extraction précise des expressions du corps, de la main et de la bouche à l'aide de dispositifs de détection appropriés dans des configurations sans marqueur ou basées sur des marqueurs. La précision des technologies de capture de la langue des signes est actuellement limitée par la capacité de résolution et de discrimination des capteurs et par le fait que les occlusions et les mouvements rapides de la main posent des défis importants à la capture précise des signes. La reconnaissance du langage des signes (SLR) implique le développement de puissants algorithmes d'apprentissage automatique pour classer de manière robuste les articulations humaines en signes isolés ou en phrases continues. Les limites actuelles du SLR résident dans le manque de grands ensembles de données annotées qui affectent grandement la précision et la capacité de généralisation des méthodes SLR, ainsi que la difficulté à identifier les limites des signes dans les scénarios SLR continus.

D'autre part, la traduction en langue des signes (SLT) implique la traduction entre différentes langues des signes, ainsi que la traduction entre les langues des signes et les langues parlées. Les méthodes SLT utilisent des algorithmes d'apprentissage automatique basés sur des séquences et visent à combler le fossé de communication entre les personnes qui signent ou parlent différentes langues. Les difficultés en SLT résident dans le manque d'ensembles de données multilingues en langue des signes,

ainsi que dans les imprécisions des méthodes SLR, étant donné que la reconnaissance de la glose (réalisée par les méthodes SLR) est l'étape initiale des méthodes SLT. Enfin, la représentation en langue des signes implique la représentation et la reproduction exactes des signes à l'aide d'avatars réalistes ou d'approches vidéo signées. Actuellement,

Les technologies de la langue des signes sont connectées d'une manière qui s'influencent mutuellement. L'extraction précise des mouvements de la main et du corps ainsi que des expressions faciales joue un rôle crucial dans le succès des algorithmes d'apprentissage automatique qui sont responsables de la reconnaissance robuste des signes. De plus, la reconnaissance précise de la langue des signes affecte de manière significative les performances des méthodes de traduction et de représentation de la langue des signes. Les percées dans les appareils sensoriels et l'IA ont ouvert la voie au développement d'applications en langue des signes qui peuvent énormément faciliter la vie quotidienne des personnes malentendantes.

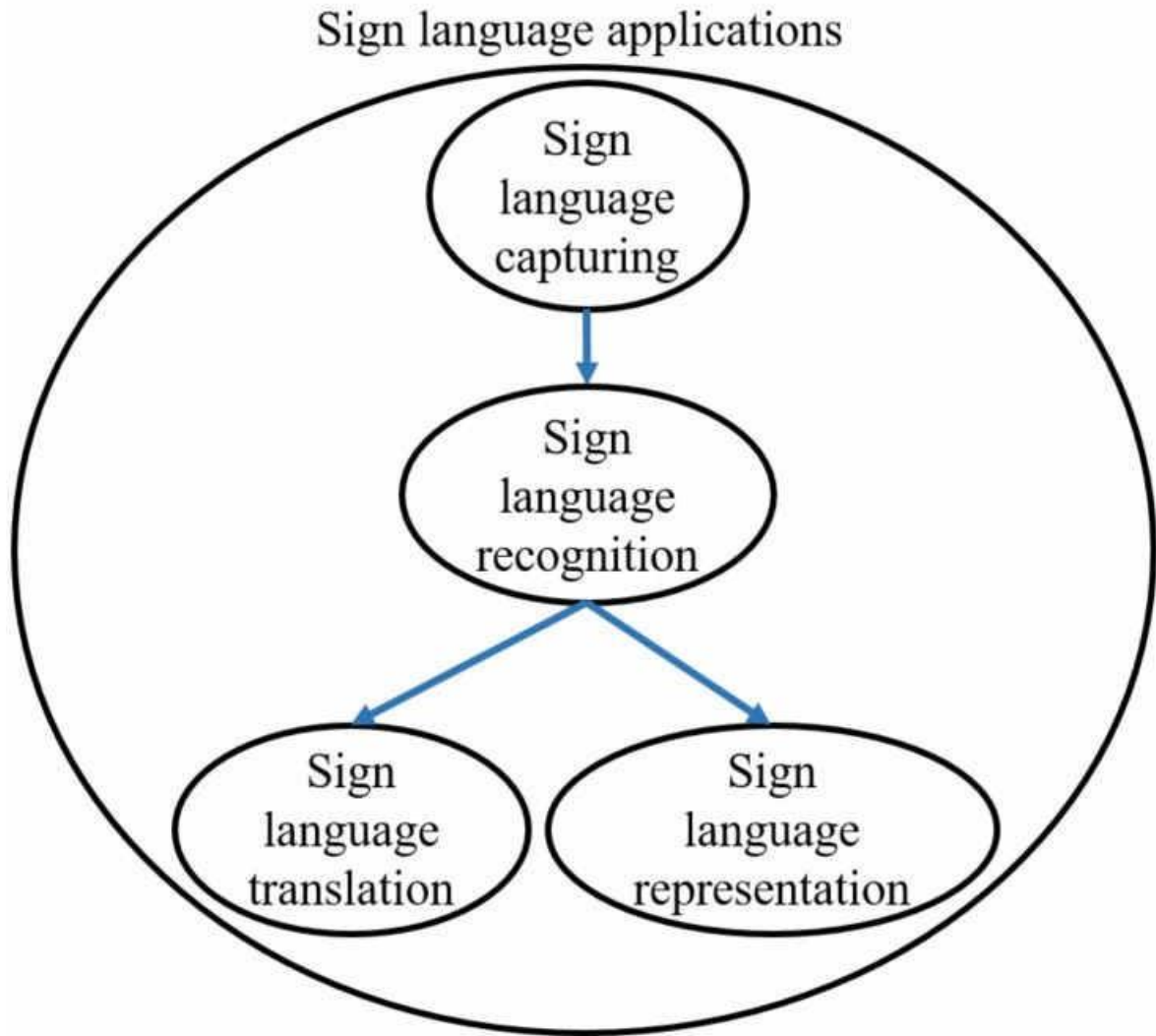


Figure 1. langage de signe

Les revues de littérature précédentes se concentrent principalement sur des technologies spécifiques de la langue des signes, telles que la reconnaissance de la langue des signes par vidéo et par capteur [5, 6, 7, 8,9] et la traduction de la langue des signes [10,11]. Dernièrement, avec le développement d'applications en langue des signes, il existe également des revues qui ont présenté des systèmes de langue des signes pour faciliter l'enseignement et l'apprentissage des personnes malentendantes, ainsi que dans les systèmes d'interprétation de la voix et du texte [12,13]. Cependant, il n'existe pas de revue systématique qui présente toutes les technologies de la langue des signes et leurs relations les unes avec les autres. Cette revue vise à combler cette lacune en présentant les avancées de l'IA dans toutes les technologies de la langue des signes, allant de la capture et la reconnaissance à la traduction et la représentation et conclut en décrivant les applications récentes de la langue des signes qui peuvent considérablement faciliter la communication entre personnes malentendantes et parlantes. L'objectif principal de cette revue est de démontrer l'importance d'utiliser

les technologies de l'IA en langue des signes pour faciliter la communication des personnes sourdes et malentendantes avec les autres communautés. En outre, cette revue vise à familiariser les chercheurs avec l'état de l'art dans toutes les technologies de la langue des signes et à proposer de futures orientations de recherche qui peuvent faciliter le développement d'approches encore plus précises pouvant conduire à des produits grand public pour la communauté sourde. Plus précisément, les objectifs de cette revue peuvent être résumés comme suit :

Un aperçu complet de l'utilisation des technologies d'IA dans diverses tâches en langue des signes (c'est-à-dire la capture, la reconnaissance, la traduction et la représentation), ainsi que leur importance pour leur domaine, est fourni.

Les avantages et les limites des technologies modernes de la langue des signes et les relations entre elles sont discutées et explorées.

Des orientations futures possibles dans le développement des technologies d'IA pour la langue des signes sont suggérées pour faciliter les futurs chercheurs dans le domaine

7. dernières recherches sur la langue des signes:

Nicolas Adaloglou[14] et al. ont mené une étude comparative expérimentale pour la reconnaissance du langage dans les méthodes basées sur la vision par ordinateur. Reconnaissance de texte de scène et parole, deux de nouveaux critères d'entraînement à la séquence ont été introduits pour cette tâche. De plus, beaucoup de pré-formation régimes ont été discutés en profondeur. Pour la langue des signes grecque, un nouveau jeu de données RVB+D est créé. Selon les auteurs, il s'agit du premier ensemble de données en langue des signes où le niveau de brillance des annotations et des phrases sont fournies pour une capture vidéo. Une analyse approfondie de la langue des signes basée sur le réseau de neurones profond la plus caractéristique Les architectures de modèles de reconnaissance ont été menées dans l'article. Après de nombreuses expériences dans trois ensembles de données accessibles au public, une évaluation comparative du signe le plus représentatif L'architecture de reconnaissance du langage est présentée. Parallèlement à cette évaluation, une nouvelle L'ensemble de données RVB + D à grande échelle disponible est également introduit pour la langue des signes grecque, adapté à SLR. Deux variantes de la classification temporelle connexionniste (CTC) connues d'une autre application champs, EntropyRegularization CTC & Stimulated CTC, sont évalués pour CSLR et il est remarqué que la combinaison de ces deux aborde deux questions importantes. Premièrement, les limites ambiguës des gloses adjacentes et d'autre part, des dépendances intra-gloses. En outre, un programme de pré-formation a été fourni, dans lequel l'apprentissage par transfert à partir d'un ensemble de données isolé proximal peut être une bonne initialisation pour la formation en reconnaissance continue de la langue des signes. La principale conclusion des travaux est que, bien que Les architectures basées sur le réseau de neurones convolutifs 3D sont plus efficaces dans les signes isolés Reconnaissance du langage, modèles basés sur le réseau neuronal convolutif 2D avec un intermédiaire par représentation brillante obtenir des résultats supérieurs dans la majorité des signes continus Ensembles de données de reconnaissance de la langue.

Amit Moryossef [15] et al. proposent un modèle de détection de la langue des signes en temps réel et léger pour besoins de la vidéoconférence. Les caractéristiques du flux optique sont extraites à partir de l'estimation de la pose humaine. Ils ont utilisé un classificateur linéaire pour montrer que ces attributs sont significatifs avec une précision de 80%. précision, lorsqu'ils sont vérifiés sur le corpus DGS. Des améliorations allant jusqu'à 91% de précision sont observées lorsqu'un modèle répétitif est utilisé directement sur l'entrée. Pour les applications de vidéo conférence une application de démonstration est décrite dans le navigateur pour la détection de la langue des signes. Contrairement à la reconnaissance du langage des signes, l'objectif de la détection du langage des signes est de détecter si et quand quelque chose est

signé. quelque chose est en train d'être signé. Cet article propose une "représentation simple du flux optique humain", qui est donnée en entrée d'un réseau neuronal sensible au temps pour effectuer une classification binaire par trame, afin de déterminer si le langage des signes a été signé ou non. pour déterminer si l'utilisateur fait des signes ou non. Ceci est pour les vidéos qui sont basées sur l'estimation de la pose.

Les auteurs rassemblent différentes entrées possibles, telles que l'estimation de la pose partielle, l'estimation de la posture du corps entier et les boîtes de délimitation, puis ils les analysent. Les auteurs rassemblent différentes entrées possibles, telles que l'estimation de la pose partielle, l'estimation de la posture du corps entier et les boîtes de délimitation, et déterminent leur temps d'acquisition en tenant compte de l'application en temps réel visée. Les systèmes de base indiquent qu'il est possible d'obtenir une précision de 79,9 % à 84,3 % sur l'ensemble de tests lorsqu'ils sont utilisés en tant que systèmes de base. 84,3% sur l'ensemble de test lorsqu'il est utilisé comme un classificateur linéaire avec un nombre spécifié de cadres de contexte. de cadres contextuels. Néanmoins, tous les modèles récurrents obtiennent de meilleures performances que les systèmes de base, pour lesquels la précision obtenue se situe entre 87,7 % et 84,3 %. pour lesquels la précision obtenue se situe entre 87,7% et 91,5% sur l'ensemble de test. Il existe de nombreuses restrictions à la perspective. La première étant qu'elle dépend du système d'estimation de la posture pour fonctionner en temps réel sur l'appareil du signataire. Cela représente un défi, car l'estimation de la posture la plus moderne est réalisée sur une seule image sur un GPU avec OpenPose peut prendre jusqu'à 300 ms. En outre, le modèle peut ne pas être modèle pourrait ne pas être en mesure de faire la différence entre une gestuelle occasionnelle et une signature. Un autre inconvénient est que les sons à haute fréquence peut avoir un effet négatif sur les animaux domestiques à l'ouïe sensible.

MohitPatil [16] et. al. présentent un système de reconnaissance des gestes et des poses de la langue des signes indienne basé sur une grille. langue des signes indienne. Ce système offre à la fois : détection en temps réel et précision. Les techniques comme la stabilisation des objets et la couleur de la peau, la détection des visages, sont utilisées. La segmentation est également utilisée pour le suivi et la détection des mains. L'article décrit un système permettant de classer avec succès les 33 poses de main dans la langue des signes indienne. SignLanguage. Les gestes d'une seule main sont uniquement pris en considération dans cette recherche. Le système proposé Le système présenté dans cet article est capable de détecter précisément les mouvements de la main d'un utilisateur en utilisant des méthodes telles que Extraction de la couleur de la peau, stabilisation de l'objet, extraction de la main et élimination du visage. Le système classe les 28 poses de main (8 chiffres et 20 lettres) en langue des signes indienne avec une précision de 96,4 %. avec une précision de 96,4 %. Le système classe également 10 gestes avec une précision moyenne de 98,23 %. L'approche utilisée est une chaîne de modèles de Markov cachés pour chaque signe et un modèle kNN qui classe chaque pose de main. chaque pose de main. Les résultats montrent

que le système est capable de détecter les postures de la main et les signes de la langue des signes indienne avec précision. et des signes en langue des signes indienne avec précision et en temps réel. La méthode présentée est commune et peut être étendue à d'autres signes à deux mains et à une seule main. La procédure présentée dans cette étude peut être utilisée dans d'autres langues, à condition que le jeu de données remplisse les conditions préalables actuelles du système. la condition préalable actuelle du système.

L'article de Jie Huang et. al.[17] se concentre sur la décomposition de la langue des signes continue Reconnaissance (SLR) dans des mondes isolés uniques et résolution de ces problèmes à l'aide de segmentation. Les problèmes rencontrés sont divers tels que l'étiquetage incorrect des fragments, détection incorrecte de divers mouvements. Pour résoudre ce problème Hiérarchique Les réseaux d'attention (HAN) sont utilisés, qui sont des extensions de la mémoire à long court terme (LSTM).

Le processus adopté comprend le passage de la vidéo entière à travers les HAN et l'acquisition du sortie mot à mot. La plus grande crainte de la segmentation temporelle est que la reconnaissance inexacte de les mouvements de transition entre les gestes de la main peuvent conduire à des résultats de sortie radicalement différents.

Pour la génération de description vidéo, un LSTM à deux couches couplé à un CNN offre des résultats prometteurs résultats.

L'une des étapes les plus importantes est la représentation des caractéristiques vidéo qui comprend le haut du corps mouvements principalement des gestes de la main. Ceux-ci sont lus à l'aide de capteurs RGB-D de style Kinect qui fournir un modèle 3D pratique. Pour la reconnaissance gestuelle, une combinaison de R-CNN est utilisée et un des données vidéo y sont transmises qui sont traitées image par image pour divers gestes.

Dans cet article, les auteurs, Shobhit Sinha et. al.[18] utilisent le réseau de neurones convolutifs pour concevoir un modèle robuste capable de comprendre 29 caractères de la langue des signes américaine composé de 3 caractères spéciaux et de 26 alphabets. Les auteurs ont en outre hébergé le modèle sur une interface vidéo instantanée qui donne des prédictions donne instantanément les données correspondantes Alphabets anglais à l'écran, tout comme les sous-titres. Ce système est considéré comme un traducteur de Langue des signes américaine vers l'anglais

pour les alphabets. Les auteurs ont conceptualisé cette méthode dans cet article et ont exploré quelques utilisations qui peuvent être exécuté.

Conclusion :

Dans ce chapitre on a présenté un obstacle qui menace notre société (la langue des signes américaine) et comment la résoudre en utilisant l'intelligence artificielle.

CHAPITRE 02

Deep Learning

Introduction :

Ces dernières années, deep Learning et plus particulièrement les réseaux de neurones à convolution(CNN) sont apparut spécialement pour résoudre les problèmes rencontrés dans machine Learning. Le CNN est l'une des structures réseau les plus représentatives de la technologie d'apprentissage en profondeur et a connu un grand succès dans le domaine du traitement et de la reconnaissance d'images. Ce chapitre porte sur l'étude du deep Learning Ce chapitre s'articule autour d'IA, ainsi que le transfert Learning.[19]

L'intelligence artificielle (IA) est en plein développement ces dernières années. Comprendre les dernières avancées dans ce domaine revient à étudier deux concepts très populaires tels que l'apprentissage automatique ou Machine Learning (ML) et l'apprentissage profond ou Deep Learning (DL).[20]

Le Deep Learning est une méthode d'IA issue du concept de machine Learning. Cette méthode d'apprentissage dit profond se base plus spécifiquement sur la notion de réseau de neurones artificiel.

De structure non linéaire, un réseau de neurones artificiel se présente sous la forme d'un réseau d'ensembles d'unités d'exécution d'information (représentant les neurones) superposées en couches et liés entre eux via des connecteurs (les synapses). Partant de-là, il traite l'information à travers des modèles de propagation d'activations de ces unités.[21]

1.L'IA dans le monde professionnel :

Presque toutes les industries sont affectées par l'IA. Le Machine Learning et le Deep Learning y jouent un grand rôle.

Que vous soyez un professionnel de la santé ou un avocat, il est possible qu'un jour un modèle hautement autonome vous assiste ou même vous remplace.

Dans les **métiers de la santé**, il existe déjà des applications pour **automatiquement diagnostiquer un patient**.



Figure 2. DETECTION AUTOMATIQUE D'UNE FRACTURE A L'AIDE DU DEEP LEARNING

2. Domaine du Deep Learning :

Le deep Learning s'emploie sur pour innombrable contextes et éventualité d'usage dans imitation, La reconnaissance d'image La reconnaissance vocale, La robotique, La cyber sécurité, La bio_informatique Quelqu'un peut régulièrement rappeler les technologies dédiées de la direction assistée et de l'aide au évaluation sur le domaine thérapeutique.[21]

3. Deep Learning et IA :

Le Deep Learning peut être considéré comme une nouvelle étape dans l'évolution de l'intelligence artificielle. A ses origines, cette dernière se contentait de répondre à des règles prédéterminées sur la base d'un modèle cognitiviste. L'intervention d'un programmeur demeurait alors indispensable pour perfectionner le système ou y intégrer d'autres fonctionnalités ou de nouvelles règles.[21]

A l'instar de la machine Learning statistique, le Deep Learning rend l'IA autonome en lui permettant d'intégrer par elle-même de nouvelles règles. L'amélioration exponentielle de la puissance de calcul et le développement d'applications connexes

permettent à l'apprentissage profond de générer des couches de neurones toujours plus complexes et denses.[21]

4. Le Deep Learning comme solution dans le e-commerce

Il est évident que le secteur du commerce électronique génère de grandes quantités de données. Les entreprises, les commerçants et les distributeurs sont conscients que les solutions de Big Data pour gérer leurs opérations rendront leur activité plus précieuse. Malgré l'irruption de toutes ces solutions innovantes, le Big Data peut représenter une bénédiction ou une malédiction, selon la façon dont il est utilisé et appliqué.

La révolution de l'intelligence artificielle vise à faciliter la gestion de cette énorme quantité de données, grâce à des technologies intelligentes comme le Deep Learning. Elle est essentielle, car elle fournit des éléments pour une meilleure analyse des données.

Dans un cas pratique, l'analyse par l'IA permet à une boutique en ligne de proposer plus facilement des produits intéressants à ses clients, de mettre en évidence leurs préférences et de leur apporter une attention personnalisée. Pour ce faire, le Deep Learning automatise ce que l'on appelle l'analyse prédictive. Grâce à cette dernière, les clients peuvent recevoir des suggestions lors d'un achat.

L'apprentissage profond définit un style lorsqu'il s'agit de faire du commerce électronique. En effet, il ne s'agit pas de créer des sites en ligne qui attirent de grandes proportions d'acheteurs. L'objectif est d'envoyer des **messages clairs et individualisés** à chacun d'eux.

Le Big Data est soumis à une analyse approfondie grâce au Deep Learning, ce qui conduit à faciliter le processus d'achat des clients. Les algorithmes d'apprentissage en profondeur aident l'entreprise à obtenir une meilleure expérience et à garder une trace de ceux qui ont visité son site.

L'apprentissage profond arrive pour faciliter l'expansion du commerce électronique. Les ventes en ligne sont stimulées par les tendances technologiques telles que les chatbots.

D'une certaine manière, le Deep Learning redéfinit le commerce en ligne et on n'est qu'aux balbutiements. Par conséquent, ceux qui l'adoptent auront plus d'avantages.

5. L'histoire du Deep Learning :

Contrairement à ce qu'on pourrait penser, le Deep Learning n'est rien apparu seul joliment et de repartir pour pas absolument de commencé de repartir d'une germe une germe lequel s'est développée et lequel n'a cessation pour mûrir à concevoir le Deep Learning.

Le Deep Learning est un concept nouveau qui émerge depuis les années 2000. Si le Deep Learning lui est nouveau, ce n'est pas le cas des réseaux de neurones artificiels, concept sur lequel se base le Deep Learning.

On entend parler du premier neurone artificiel en 1943 lorsque Warren McCulloch et Walter Pitts publient leur premier modèle mathématique et informatique du neurone biologique: le neurone formel.

Le neurone formel est directement inspiré du neurone biologique.

Un neurone biologique se compose d'un corps cellulaire qui comprend le noyau du neurone et la plupart des éléments complexes de la cellule. De nombreux prolongements appelés dendrites et un très long prolongement appelé axone. À son extrémité, l'axone se décompose en plusieurs ramifications que l'on appelle télodendrons qui se terminent par de minuscules structures appelés synapses et qui sont directement reliés à des dendrites ou directement au corps cellulaire d'autres neurones.[2] Un neurone reçoit des signaux électriques par le biais des dendrites et lorsque que le neurone reçoit suffisamment de signaux en un temps donné (quelques millisecondes) alors il déclenche ses propres signaux.[22]

Le fonctionnement d'un neurone est relativement simple mais lorsqu'un neurone est connecté à des milliers d'autres et qu'il y a des milliards de neurones, cela crée des réseaux capable de résoudre des problèmes extrêmement complexes.[22] Warren Mc Culloch et Walter Pitts propose le premier neurone artificiel (1943), qui est un modèle très simplifié du neurone biologique. Ce neurone artificiel possède une ou plusieurs entrées et une sortie binaires. [22]

6. Les algorithmes de Deep Learning :

Après avoir été préalablement entraîné sur une base d'exemples, un réseau de neurones artificiel pourra typiquement être utilisé dans la reconnaissance d'images, pour identifier un chat par exemple. Chaque couche du réseau sera dédié à l'identification d'une caractéristique de l'animal : la silhouette, la tête, les deux oreilles, les quatre pattes, le pelage, etc. Les réseaux de neurones artificiels sont également très utilisés dans le traitement automatique du langage ou Natural languageprocessing (NLP). [21]

Depuis 2012, les algorithmes à base de Deep Learning (apprentissage profond) semblent prêts à résoudre bien des problèmes : reconnaître des visages comme le propose DeepFace, ou encore la recherche de cellules cancéreuses. Pourtant, les fondements de ces méthodes ne sont pas si récents : le Deep Learning a été formalisé en 2007 à partir des nouvelles architectures de réseaux de neurones dont les précurseurs sont McCulloch et Pitts en 1943. Suivront de nombreux développements comme le perceptron, les réseaux de neurones convolutifs de Yann Le Cun et Yoshua Bengio en 1998 et les réseaux de neurones profonds qui en découlent en 2012 et ouvrent la voie à de nombreux champs d'application comme la vision, le traitement du langage ou la reconnaissance de la parole.

Pourquoi maintenant ? Parce que ces nouvelles techniques de machine Learning profitent de données massives (big data) que l'on est désormais capables d'analyser ainsi que de capacités de calcul phénoménales notamment grâce aux processeurs graphiques. Preuve que chaque domaine irrigue les autres, c'est pour pouvoir utiliser

les immenses promesses du DeepLearning que Google a mis au point les accélérateurs TPU. [23]

7. Transfert Learning :

De nombreux modèles de classification d'images ont été introduits pour aider à résoudre le problème primordial de la précision de la reconnaissance. La classification des images est l'un des problèmes fondamentaux dans le domaine de la vision par ordinateur avec une grande variété d'applications pratiques. Les exemples incluent : la reconnaissance d'objets pour la manipulation robotique, la détection de piétons ou d'obstacles pour les véhicules autonomes, entre autres. Une grande attention a été associée à l'apprentissage automatique, en particulier aux réseaux de neurones tels que les concours de classification d'images gagnants du réseau de neurones convolutifs (CNN). Ce travail propose l'étude et l'investigation d'un tel modèle d'architecture CNN (c'est-à-dire Inception-v3) pour déterminer s'il fonctionnerait mieux en termes de précision et d'efficacité avec de nouveaux ensembles de données d'images via l'apprentissage par transfert. Le modèle recyclé est évalué et les résultats sont comparés à certaines approches de pointe.

Définition :

Commençons d'abord par définir ce terme de plus en plus utilisé en *Data Science*. *Le Transfer Learning, ou apprentissage par transfert en français, désigne l'ensemble des méthodes qui permettent de transférer les connaissances acquises à partir de la résolution de problèmes donnés pour traiter un autre problème.*

Le *Transfer Learning* a connu un grand succès avec l'essor du *Deep Learning*. En effet, bien souvent, les modèles utilisés dans ce domaine nécessitent des **temps de calcul élevés** et des **ressources importantes**. Or, en utilisant des modèles pré-entraînés comme point de départ, le *Transfer Learning* permet de développer rapidement des modèles performants et résoudre efficacement des problèmes complexes en *Computer Vision* ou *Natural Language Processing, NLP*.

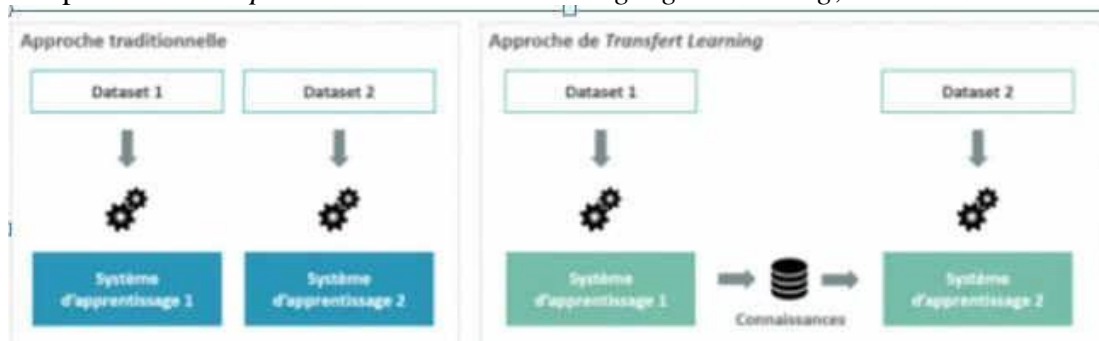


Figure 3. Approche traditionnelle vs Approche de transfert Learning

Intuitivement, le Transfer Learning s'inspire fortement du processus avec lequel nous apprenons.

Prenons l'exemple de quelqu'un qui maîtrise la guitare et souhaite apprendre à jouer au piano. Il pourra capitaliser sur ses connaissances en musique pour apprendre à jouer un nouvel instrument. De la même manière, un modèle de reconnaissance de voiture pourra être très rapidement réadapté à la reconnaissance de camions. Comment le Transfer Learning est mis en œuvre concrètement pour la résolution de problématiques de Deep Learning ?

Maintenant que nous avons défini le Transfer Learning, intéressons-nous à son application aux problématiques de Deep Learning, domaine dans lequel il connaît aujourd'hui un grand succès.

L'utilisation des méthodes de Transfer Learning en Deep Learning consiste principalement à exploiter des réseaux de neurones pré-entraînés. Généralement, ces modèles correspondent à des algorithmes très performants qui ont été développés et entraînés sur de larges bases de données et qui sont aujourd'hui librement partagés.

Dans ce contexte, on peut distinguer 2 types de stratégies :

1. L'utilisation de modèles pré-entraînés comme extracteurs de features :

L'architecture des modèles de Deep Learning se présente très souvent sous la forme d'un empilement de couches de neurones. Ces couches apprennent différentes caractéristiques en fonction du niveau où elles se situent. La dernière couche (généralement une couche entièrement connectée, dans le cas de l'apprentissage supervisé) est utilisée pour obtenir la sortie finale. La figure ci-dessous illustre l'architecture d'un modèle de Deep Learning utilisé pour la détection de chat/chien. Plus la couche se situe en profondeur, plus elle permet d'extraire des features spécifiques.

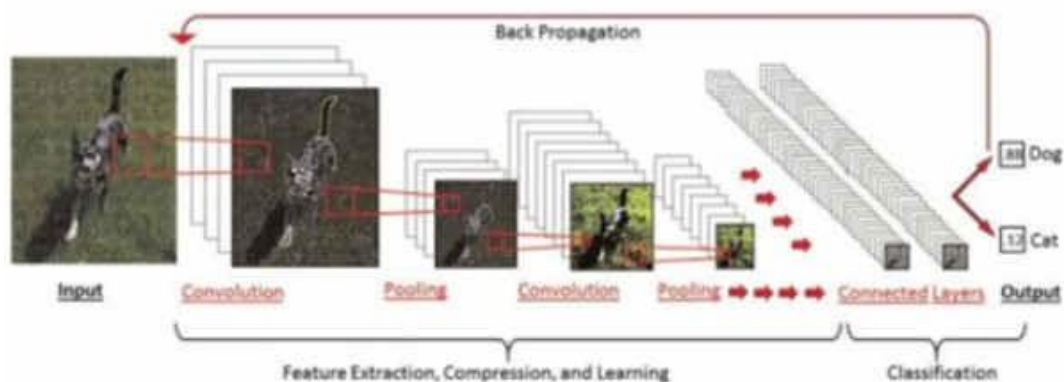


Figure 4. Exemple de modèle de réseau de neurones utilisés pour la classification d'images

L'idée est donc de réutiliser un réseau pré-entraîné sans sa couche finale. Ce nouveau réseau fonctionne alors comme un extracteur de features fixes pour la réalisation d'autres tâches.

Afin d'illustrer cette stratégie, prenons le cas où nous souhaitons créer un modèle capable d'identifier l'espèce d'une fleur à partir de son image. Il est alors possible d'utiliser les premières couches du modèle de réseau de neurones convolutifs

AlexNet, initialement entraîné sur la base d'image ImageNet pour la classification d'images.

8. Les réseaux de neurones :

Les réseaux de neurones ont été développés comme un modèle mathématique générique afin de modéliser les neurones biologiques. Ils comportent un certain nombre d'éléments de traitement d'information appelés neurones.[4] Chaque neurone a son propre état interne interprété par la fonction d'activation. Il envoie son activation aux autres neurones sous forme de signaux. La connexion entre les neurones est réalisée via des liens orientés et pondérés.[4]

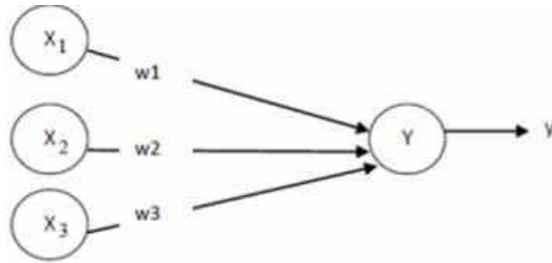


Figure 5. neurone artificiel avec une seule sortie

Le neurone Y reçoit les entrées de X_1 , X_2 et X_3 qui ont comme valeurs de sortie x_1 , x_2 et x_3 . Les poids des liens de connexion de $X_1, 2$ et X_3 sont w_1, w_2 et w_3 . La valeur d'entrée de neurone Y est : $y = w_1x_1 + w_2x_2 + w_3x_3$. Le signal de sortie y est déterminée par la fonction d'activation (y). Les réseaux de neurones sont caractérisés par l'architecture (l'organisation des neurones), l'apprentissage (méthode de détermination des poids de connexions), et par leur fonction d'activation. [24]

Deep Learning et réseaux de neurones : biologiques ou artificiels ? Même combat

Avant de s'attaquer au fonctionnement précis des réseaux de neurones, on s'est dit qu'il serait intéressant de faire un parallèle avec les neurones biologiques.

(Pas de panique, un article dédié aux réseaux de neurones utilisés en Deep Learning suivra)

Neurone Biologique : Structure et rôle

Le système nerveux est composé de milliards de cellules : c'est un réseau de neurones biologiques. En effet, les neurones ne sont pas indépendants les uns des autres, ils établissent entre eux des liaisons et forment des réseaux plus ou moins complexes.

Le neurone biologique est composé de trois parties principales :

- Le corps cellulaire composé du centre de contrôle traitant les informations reçues par les dendrites.
- Les dendrites sont les principaux fils conducteurs par lesquels transitent l'information venue de l'extérieur.

- L'axone est fil conducteur qui conduit le signal de sortie du corps cellulaire vers d'autres neurones.
 - Quant aux **synapses**, elles font **effet de liaison et de pondération entre neurones** et permettent donc aux neurones de communiquer entre eux.

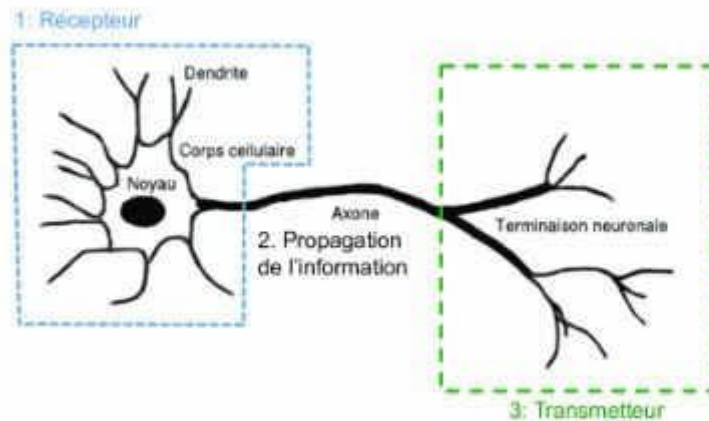


Figure 5. neurone

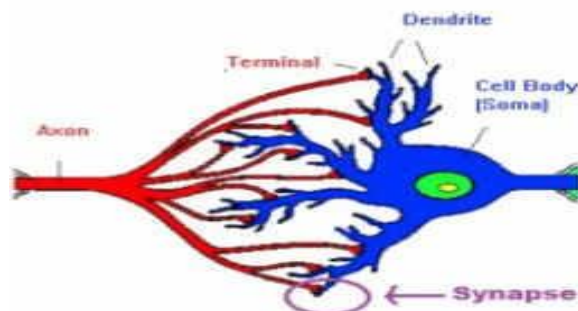


Figure 6. neurone

Neurones artificiels :

Les neurones biologiques disposent d'un centre de contrôle (appelé cellule somatique) faisant la somme des informations recueillies par les dendrites. Ensuite, le centre de contrôle retourne un potentiel d'action en suivant les règles suivantes :

- Si la somme en entrée ne dépasse pas le seuil d'excitation : pas de message nerveux via l'axone.
- Si la somme en entrée dépasse le seuil d'excitation : un message nerveux est émis via l'axone (c'est l'idée, mais en réalité un peu plus compliqué).

Procédons à une comparaison simple des étapes principales de l'algorithme du perceptron avec les éléments constitutifs des neurones biologiques. Ce choix

d'algorithme se justifie car il se rapproche au mieux du fonctionnement des neurones biologiques :

- Les synapses/dendrites : pondération de chaque élément en entrée $w_i x_i$
- Corps cellulaires : application d'une fonction d'activation f à la somme des entrées pondérées
- Axone : sortie de notre modèle

Architecture :

Les réseaux de neurones sont souvent classifiés en deux architectures : les réseaux de neurone avec un seul niveau et multi-niveaux. Le nombre de niveaux est calculé sans prendre en considération les unités.[24]

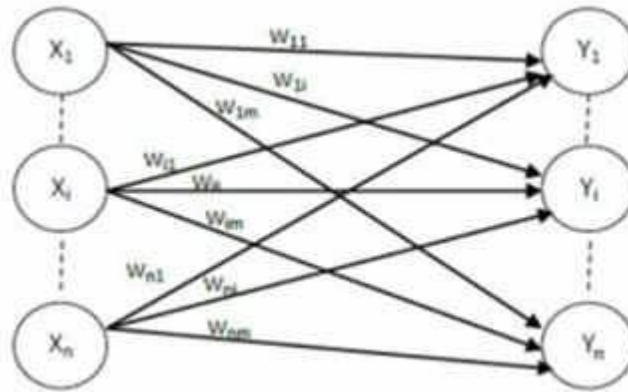


Figure 7. le réseau de neurones à un seul niveau

Les neurones de la couche d'entrée doivent uniquement passer et distribuer les entrées et ne pas effectuer de calcul. Ainsi, la seule vraie couche de neurones est celle de droite. Chacune des entrées $X_1, X_2 \dots X_n$ est connectée à chaque neurone de la couche de sortie à travers le poids de lien. Comme chaque valeur des sorties $Y_1, Y_2 \dots Y_n$ est calculée à partir du même ensemble de valeurs d'entrée, chaque sortie est modifiée en fonction des poids de liens.[24]

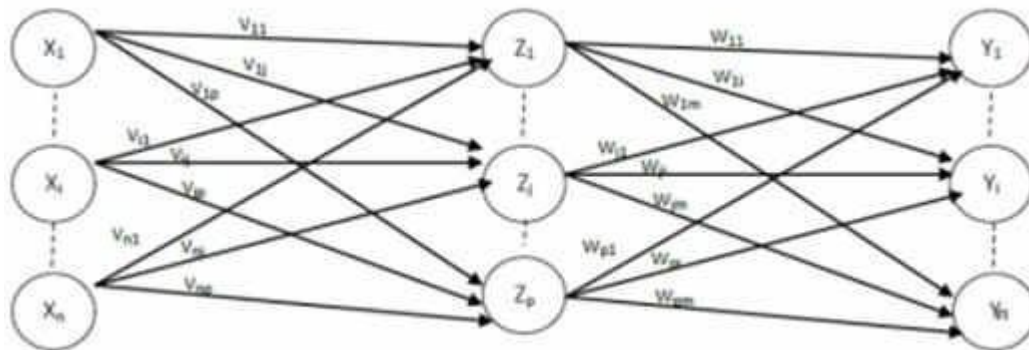


Figure 8. Le réseau de neurones à multi niveaux

La figure 5 montre le réseau de neurones multi-niveaux ce dernier se distingue du réseau d'un seul niveau en ayant une ou plusieurs couches masquées. Dans cette structure, les nœuds d'entrée transmettent les informations aux unités dans la première couche masquée, puis les sorties de la première couche masquée sont passées à la couche suivante, et ainsi de suite. [24]

Le réseau multi-niveaux peut également être considéré comme une cascade de groupes de réseaux d'un seul niveau. Le niveau de complexité se traduit par le nombre de réseaux monocouche qui sont combinés dans ce type de réseau. Le concepteur d'un réseau de neurones devrait considérer combien de couches cachées sont requises, selon la complexité du calcul souhaité. [24]

9.L'apprentissage :

La méthode de paramétrage des poids (apprentissage) est une caractéristique importante pour distinguer différents types de réseaux de neurones. Deux modes d'apprentissage existent : l'apprentissage supervisé, et l'apprentissage non supervisé. [24]

APPRENTISSAGE SUPERVISÉ :

Ce cadre de machine Learning part du fait que les données historiques (ou exemples) sont annotées.

Prenons le cas de la reconnaissance d'objets : un problème supervisé correspond au cas où le label « voiture » est bien associé, en base, à des photos de voitures, le label « chat » à des photos de chat, etc. L'algorithme apprend ainsi à partir de milliers ou de millions d'exemples étiquetés : il cherche la relation qui permet de relier les images aux labels. Après avoir classifié correctement les exemples, il peut ensuite généraliser ce classement à de nouvelles données : classifier correctement des images de voiture ou de chat qu'il n'a jamais vues durant la phase d'apprentissage. C'est ce qu'on appelle la capacité de généralisation.[28] Dans un cadre business, on parle souvent d'analyse prédictive. Parmi les exemples d'applications, citons la classification d'email en spam ou non selon le contenu du message, son expéditeur, son sujet..., le diagnostic médical selon les symptômes, etc.[25]

CLASSIFICATION :

Cette méthode d'analyse de données regroupe des algorithmes d'apprentissage supervisé adapté aux données qualitatives. L'objectif est d'apprendre (autrement dit de trouver) la relation qui lie une variable d'intérêt, de type qualitative, aux autres variables observées, éventuellement dans un but de prédiction. On utilise la classification lorsque la variable d'intérêt est qualitative, c'est à dire qu'elle prend ses valeurs dans un espace qui ne possède pas de métrique naturelle. Par exemple on peut essayer de prédire le genre littéraire d'un livre ; cette variable est discrète (genre « policier », genre « science-fiction », etc.) et il n'y a aucune relation entre les genres, il est difficile de définir une distance entre eux. Les algorithmes de classification les plus simples sont la régression logistique, le k-nearestneighbour (méthode des k plus proches voisins)... ; les plus complexes sont les réseaux de neurones, les support

vector machine, les mixture model (modèles de mélange), le Bayesian classifier (classifieur Bayésien), etc.[25]

APPRENTISSAGE NON SUPERVISÉ :

A la différence de l'apprentissage supervisé, le contexte non supervisé est celui où l'algorithme doit opérer à partir d'exemples non annotés.

Il doit faire émerger automatiquement les catégories à associer aux données qu'on lui soumet pour reconnaître qu'un chat est un chat, une voiture, une voiture comme sont capables de le faire les animaux et les humains. Le problème d'apprentissage non supervisé le plus fréquent est la segmentation (ou clustering) où l'on essaie de séparer les données en groupes (catégorie, classe, cluster...) : regrouper des images de voitures, de chats, etc. Beaucoup d'espairs sont portés sur la détection d'anomalies pour la maintenance prédictive, lacybersécurité, mais aussi le dépistage précoce de maladies, etc.

De manière générale, l'algorithme cherche à maximiser d'une part l'homogénéité des données au sein des groupes de données et à former des groupes aussi distincts que possible : selon le contexte, on choisit d'utiliser tel ou tel algorithme pour classer les données par exemple selon leur densité ou leur gradient de densité. Dans le cas de la détection d'anomalies, c'est plutôt le caractère extrême ou atypique des valeurs ou d'un pattern dans les données qui est recherché.

La métrique sous-jacente joue un rôle clé pour déterminer ce qui est la norme et ce qui s'en éloigne.[25]

10.Applications des réseaux de neurones de classification :

La classification est l'une des tâches de prise de décision les plus fréquentes de l'activité humaine. Un problème de classification se produit lorsqu'un objet doit être affecté à un groupe ou une classe prédéfini en fonction d'un nombre d'attributs observés liés à cet objet [26]. Afin de trouver une solution à ce problème, des chercheurs dans différents domaines ont eu recours aux réseaux de neurones. De nombreuses études telles que [56, 173, 194] ont permis d'effectuer plusieurs comparaisons de performances entre classificateurs neuronaux et classificateurs classiques. D'après ces références, les classificateurs basés sur les réseaux de neurones ont donné des résultats prometteurs dans la classification et la reconnaissance. C'est pour cela que les chercheurs les ont appliqués avec succès à diverses tâches de classification du monde réel dans les domaines de l'industrie, des affaires et de la science [238]. Commençons par le domaine des affaires, un cas spécifique a été évalué concerne la prévision de la faillite. L'hypothèse commune à tous les modèles de prévision de la faillite est que les états financiers tiennent compte des facteurs économiques fondamentaux et des caractéristiques d'une entreprise. [25]

11. Les réseaux de neurones convolutifs :

Les réseaux de neurones convolutifs sont directement inspirés du cortex visuel des vertébrés. Un réseau de neurones à convolution, appelé aussi convnet (pour «Convolutional Network»), ou encore CNN (pour «Convolutional Neural Network») se représente en général sous la forme suivante:

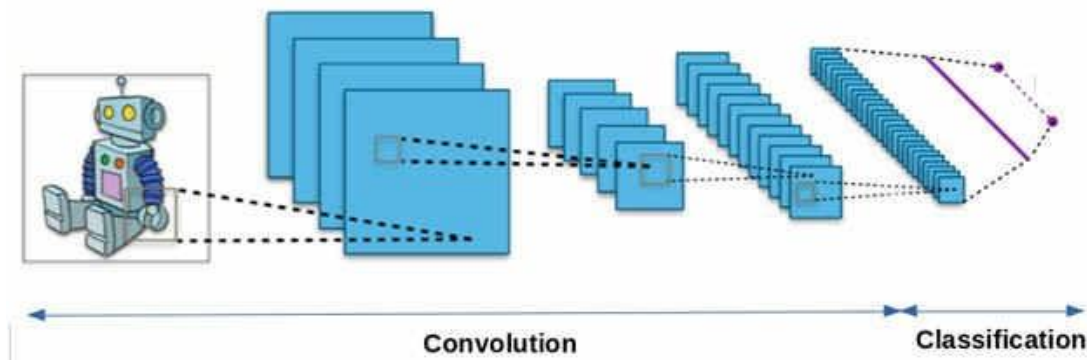


Figure 9. Architecture d'un réseau de neurones convolutifs

Architecture d'un Convolutional Neural Network-CNN :

Les CNN désignent une sous-catégorie de réseaux de neurones et sont à ce jour un des modèles de classification d'images réputés être les plus performant

Leur mode de fonctionnement est à première vue simple : l'utilisateur fournit en entrée une image sous la forme d'une matrice de pixels.[26]

Celle-ci dispose de 3 dimensions :

- Deux dimensions pour une image en niveaux de gris.
- Une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert, Bleu).

Contrairement à un modèle MLP (Multi Layers Perceptron) classique qui ne contient qu'une partie classification, l'architecture du Convolutional Neural Network dispose en amont d'une partie convolutive et comporte par conséquent deux parties bien distinctes :

- Une partie convolutive : Son objectif final est d'extraire des caractéristiques propres à chaque image en les compressant de façon à réduire leur taille initiale. En résumé, l'image fournie en entrée passe à travers une succession de filtres, créant par la même occasion de nouvelles images appelées cartes de convolutions. Enfin, les cartes de convolutions obtenues sont concaténées dans un vecteur de caractéristiques appelé code CNN.
- Une partie classification : Le code CNN obtenu en sortie de la partie convolutive est fourni en entrée dans une deuxième partie, constituée de couches entièrement connectées appelées perceptron multicouche (MLP)

pour Multi Layers Perceptron). Le rôle de cette partie est de combiner les caractéristiques du code CNN afin de classer l'image.[26]

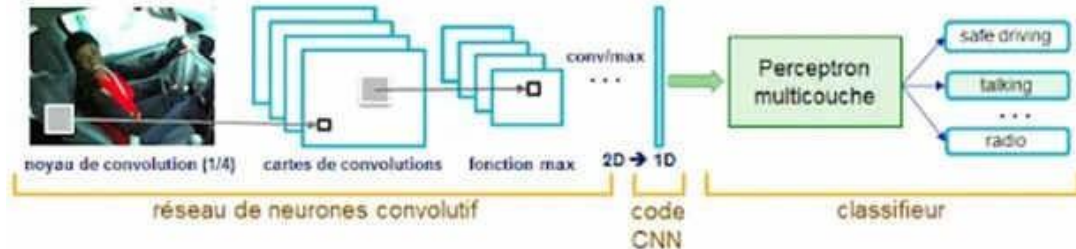


Figure 10. Schéma représentant l'architecture d'un CNN

Partie convolutive :

Tout d'abord, à quoi sert la convolution ?

La **convolution** est une opération mathématique simple généralement utilisée pour le **traitement et la reconnaissance d'images**. Sur une image, son effet s'assimile à un filtrage dont voici le fonctionnement :

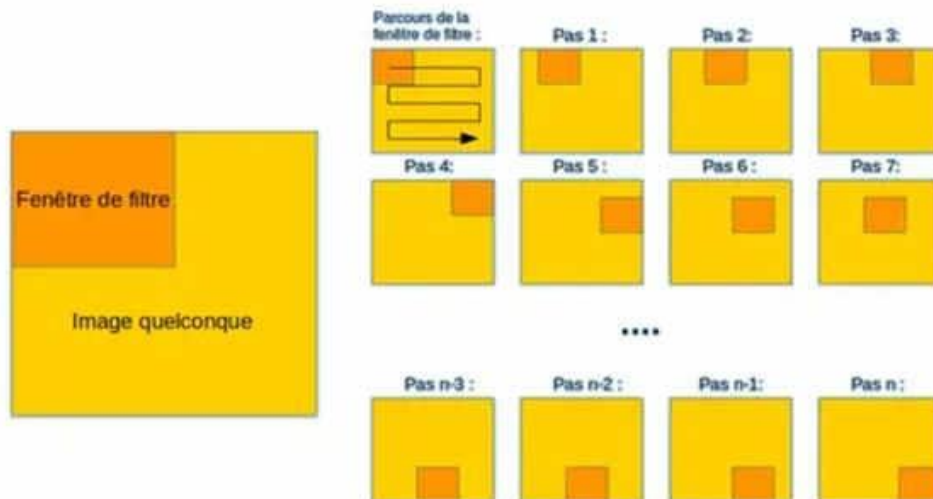


Figure 11. schéma du parcours de la fenêtre de filtre sur l'image

Couche de convolution :

Un réseau de neurones convolutifs [94, 144] est un réseau de neurones qui utilise une opération mathématique qui s'appelle convolution ou produit de convolution. Il s'agit d'une opération linéaire. Chaque réseau de neurones convolutifs contient au moins une couche de convolution. Soient f et g deux fonctions définies sur \mathbb{R} , le produit de convolution entre f et g est généralement noté $f * g$ et il est défini par l'équation suivante : $s(x) = (f * g)(x) = \int_{-\infty}^{+\infty} f(t)g(x - t)dt$ (3.18)

Dans un réseau de neurones convolutifs, le premier argument f est assimilé à l'entrée, le deuxième argument g est assimilé au noyau de convolution et la sortie est assimilée à la carte de caractéristiques. En réalité, les réseaux de neurones convolutifs utilisent une convolution discrète qui est définie par : $s(x) = (f * g)(x) = \sum_{t} f(t)g(x - t)$

$X_{a=-\infty}^{\infty} f(t)g(x-a)$ (3.19) En apprentissage automatique, l'entrée est toujours un tableau de données multidimensionnel, et le noyau est toujours un tableau de paramètres multidimensionnel qui seront adaptés par l'algorithme d'apprentissage.

Ces tableaux sont référencés par des tenseurs. [6]

La convolution est toujours utilisée avec une dimension supérieure à 1. La convolution la plus utilisée est une convolution 2D. Dans ce cas, pour une entrée image I et pour un noyau K , la convolution discrète s'écrit : $S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i-m, j-n)K(m, n)$ (3.20) D'après cette formule, l'image I et le noyau de convolution

K sont parcourus dans des sens opposés.

En effet, quand m augmente, les indices de I décroissent et les indices de K augmentent. La seule raison d'utiliser la convolution est sa propriété de commutativité. Cependant, cette propriété n'est pas importante dans l'implémentation du réseau de neurones, car les autres fonctions avec lesquelles la convolution est utilisée ne sont pas commutatives. C'est pour cette raison que la plupart des bibliothèques de réseaux de neurones implémentent la fonction de la corrélation croisée mais ils l'appellent convolution (3.23). Cet opérateur est identique à la convolution sauf qu'il perd la propriété de commutativité. $S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i+m, j+n)K(m, n)$

(3.21) Une couche de convolution est caractérisée par : — les dimensions des noyaux de convolution, généralement une convolution à une dimension égale à 2 avec des noyaux carrés. — le nombre des filtres de convolution C , c'est le nombre de cartes d'activations, ou cartes de caractéristiques, en sortie de la couche. Ces cartes sont représentées sous la forme de tenseurs de dimension 3 (H, W, C) avec H la hauteur des cartes, W la largeur et C le nombre de canaux. — le pas de convolution -ou stride- s . C'est le pas de décalage du noyau de convolution à chaque calcul. Par exemple dans le cas de deux dimensions $s = (s_1, s_2) = (1, 1)$, le noyau de convolution sera déplacé d'un seul pixel vers la droite pour chaque mouvement horizontal du noyau et d'un seul pixel vers le bas pour chaque mouvement vertical du noyau lors de la création de la carte de caractéristiques. — le padding p . C'est le paramètre permettant de dépasser la taille de l'image pour appliquer la convolution en ajoutant des pixels autour de l'image. Dans la suite, et afin de faciliter l'analyse et simplifier la visualisation, nous utilisons une convolution 2D ($N = 2$), avec une entrée carrée $h = w$, un noyau de convolution carré de dimension k , un pas s , et un padding p

1. Dans un premier temps, on définit la taille de la fenêtre de filtre située en haut à gauche.

2. La fenêtre de filtre, représentant la feature, se déplace progressivement de la gauche vers la droite d'un certain nombre de cases défini au préalable (le pas) jusqu'à arriver au bout de l'image.
3. À chaque portion d'image rencontrée, un calcul de convolution s'effectue permettant d'obtenir en sortie une carte d'activation ou featuremap qui indique où est localisées les features dans l'image : plus la featuremap est élevée, plus la portion de l'image balayée ressemble à la feature.

Méthode de sous échantillonnage : le Max-Pooling

Le Max-Pooling est un processus de discrétisation basé sur des échantillons. Son objectif est de sous-échantillonner une représentation d'entrée (image, matrice de sortie de couche cachée, etc.) [27] en réduisant sa dimension. De plus, son intérêt est qu'il réduit le coût de calcul en réduisant le nombre de paramètres à apprendre et fournit une invariance par petites translations (si une petite translation ne modifie pas le maximum de la région balayée, le maximum de chaque région restera le même et donc la nouvelle matrice créée restera identique).

Pour rendre plus concret l'action du Max-Pooling, voici un exemple : imaginons que nous avons une matrice 4x4 représentant notre entrée initiale et un filtre d'une fenêtre de taille 2x2 que nous appliquerons sur notre entrée. Pour chacune des régions balayées par le filtre, le max-pooling prendra le maximum, créant ainsi par la même occasion une nouvelle matrice de sortie où chaque élément correspondra aux maximums de chaque région rencontrée.[26]

Couche d'échantillonnage (Pooling) :

Semblable à la couche de convolution, la couche d'échantillonnage est chargée de réduire la taille spatiale des cartes de caractéristiques, mais elle conserve les informations les plus importantes. Il existe différents types d'échantillonnage dont l'échantillonnage maximum -ou Max Pooling-, l'échantillonnage moyen -ou Average Pooling-, etc. L'échantillonnage consiste à appliquer un noyau de taille $n \times n$ sur la carte d'activation en le faisant glisser avec un pas préalablement défini (le pas est généralement égal à la taille du noyau n pour éviter le phénomène de chevauchement). Le Max Pooling renvoie la valeur maximale de la partie de l'image couverte par le noyau. Au lieu de prendre le maximum, nous pourrions prendre la moyenne de tous les éléments couverts par le noyau, cela est assuré par le Average Pooling. Le Max Pooling permet de supprimer le bruit. D'autre part, le Average Pooling effectue simplement une réduction de la dimensionnalité en tant que mécanisme de suppression de bruit. Par conséquent, en pratique, le Max Pooling est beaucoup plus utilisé que le Average Pooling puisqu'il fonctionne mieux. La forme la plus courante est une couche de Pooling avec des noyaux de taille 2×2 appliqués avec un pas de 2 réduisant ainsi la dimension de l'entrée par 2, en éliminant 75% des activations et en laissant la dimension de profondeur inchangée. Dans la Figure 3.8, le

nombre des activations passe de 16 à 4 en appliquant le pooling. Plus généralement, la couche de pooling : — accepte un tenseur de taille $W1 \times H1 \times D1$ — nécessite deux hyperparamètres : la taille F et le pas s avec $F < W1$ et $F < H1$ 47

produit un tenseur de taille $W2 \times H2 \times D2$ où : $\cdot W2 = (W1 - F) / s + 1 \cdot H2 = (H1 - F) / s + 1 \cdot D2 = D1$ — introduit zéro paramètre puisqu'il calcule une fonction fixe de l'entrée.[25]

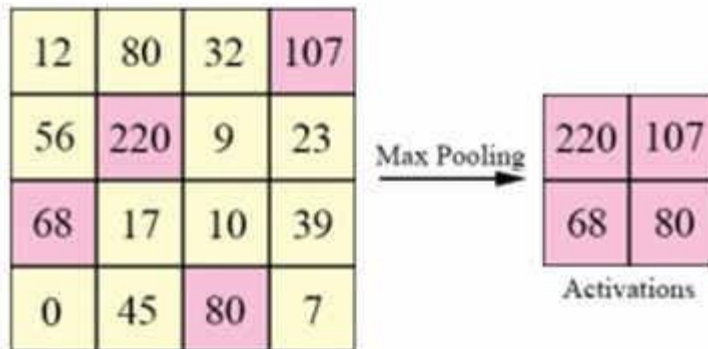


Figure 12. Exemple d'une opération de Pooling taille 2*2

Complètement connecté » implique que chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante. Leurs activations peuvent donc être calculées avec une multiplication matricielle suivie d'un offset de biais. La sortie des couches de convolution et d'échantillonnage représente des caractéristiques de haut niveau de l'image d'entrée. L'objectif de la couche complètement connectée est d'utiliser ces caractéristiques pour classer l'image d'entrée du réseau en différentes classes en fonction de la base de données d'apprentissage. Outre la classification, l'ajout d'une couche complètement connectée est généralement un moyen peu coûteux pour faire apprendre des combinaisons non-linéaires des caractéristiques. La plupart des cartes de caractéristiques des couches de convolution et d'échantillonnage peuvent être utiles pour la tâche de classification, mais une combinaison de ces fonctionnalités peut être encore meilleure.

Les couches entièrement connectées sont généralement suivies d'un Dropout [214]. Ce dernier agit sur les poids de ces couches afin de désactiver un certain nombre de neurones pour réduire le nombre de paramètres. Cela permet de contrôler le sur-apprentissage qui peut être causé par un nombre important de paramètres. La somme des probabilités en sortie de la couche entièrement connectée est $\sum_i y_i = 1$. Cela est garanti en utilisant Softmax comme fonction d'activation dans la couche en sortie de la couche entièrement connectée. La fonction Softmax prend un vecteur de scores arbitraires à valeurs réelles et le réduit à un vecteur de valeurs comprises entre zéro et un qui égal à un. Elle s'écrit sous la forme : [25]

$$\hat{y}_i = \text{softmax}(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Conclusion :

Dans ce chapitre nous avons présenté ce qu'est l'intelligence artificielle, en particulier l'apprentissage en profondeur, comme nous avons focalisé sur le réseau de neurones convolutifs et ses caractéristiques.

CHAPITRE 03

E

xpérimentation &

Résultat

Introduction :

L'intelligence artificielle est aujourd'hui essentielle dans le monde numérique, que ce soit pour le développement et l'utilisation de jeux vidéo (ou d'autres applications) ou pour les services Web, les appareils et les machines. Les neural network ou « réseaux neuronaux » en français, constituent l'un des principaux domaines de recherche s'intéressant aux caractéristiques de base de l'IA. Keras simplifie considérablement l'implémentation de ces réseaux. Découvrez ce qu'offre cette **bibliothèque open source** et dans quelle mesure elle facilite la mise en place de réseaux neuronaux.

L'**intelligence artificielle** (IA) s'attache à résoudre des problèmes qui relèvent d'activités humaines de nature variée (perception, prise de décision, planification, diagnostic, interprétation de données, compréhension du langage, conception). Ces problèmes nécessitent de mettre en jeu une grande quantité de données et de connaissances, soit exploitées directement, soit codées sous différentes formes (distributions de probabilités, poids synaptiques, etc.). Nous présentons les différents modèles développés depuis le début de l'IA, ainsi que les grands domaines d'application. Les aspects éthiques liés à l'utilisation de systèmes d'IA sont également étudiés.[30]

1. Environnement du travail :

Python :

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.

Les principales utilisations de Python par les développeurs sont :

- la programmation d'applications
- la création de services web
- la génération de code
- la métaprogrammation.

Techniquement, ce langage servira surtout pour le scripting et l'automatisation (interaction avec les navigateurs web).

1.1.2. La version actuelle de Python :

On différencie deux versions : Python 2 et Python 3. Python 2, l'ancienne version propose des mises à jour jusqu'en 2020. Python 3 est la version actuelle. Son interpréteur est plus efficace, ainsi que son contrôle de concurrence.

Keras :

Keras est une bibliothèque open source écrite en Python (sous licence MIT) basée principalement sur les travaux du développeur de Google François Chollet dans le cadre du projet ONEIROS (**O**pen-ended**N**euro-**E**lectronic**I**ntelligent **R**obot **O**perating **S**ystem). Une première version du logiciel multiplateforme a été publiée le 28 mars 2015. Le but de cette bibliothèque est de permettre la constitution rapide de réseaux neuronaux. Dans ce cadre, Keras ne fonctionne pas comme un framework propre mais

comme une **interface de programmation applicative** (API) pour l'accès et la programmation de différents **frameworks d'apprentissage automatique**. Theano, Microsoft Cognitive Toolkit (anciennement CNTK) et TensorFlow font notamment partie des frameworks pris en charge par Keras. C'est une API d'apprentissage en profondeur écrite en Python, s'exécutant sur la plate-forme d'apprentissage automatique TensorFlow. Il a été développé dans le but de permettre une expérimentation rapide. *Pouvoir passer de l'idée au résultat le plus rapidement possible est la clé d'une bonne recherche.*

Kéras est :

- **Simple** -- mais pas simpliste. Keras réduit *la charge cognitive* du développeur pour vous permettre de vous concentrer sur les parties du problème qui comptent vraiment.
- **Flexible** -- Keras adopte le principe de *divulcation progressive de la complexité* : les flux de travail simples doivent être rapides et faciles, tandis que les flux de travail arbitrairement avancés doivent être *possibles* via un chemin clair qui s'appuie sur ce que vous avez déjà appris.
- **Puissant** - Keras offre des performances et une évolutivité de pointe : il est utilisé par des organisations et des entreprises telles que la NASA, YouTube ou

Waymo

Le fonctionnement du keras :

Keras est une bibliothèque agissant au niveau du modèle : elle **met à disposition des modules** permettant de développer des modèles de deeplearning (apprentissage profond) complexes. Contrairement aux frameworks indépendants, ce logiciel open source ne s'occupe pas des opérations « lowlevel » et utilise à cet effet les bibliothèques de frameworks d'apprentissage automatique associés qui tiennent pratiquement lieu de moteurs back-end pour Keras. Conformément au **principe de modularité**, les couches désirées pour le réseau neuronal à mettre en place sont connectées. Pour autant, il n'est pas nécessaire de comprendre l'infrastructure réelle du framework choisi et l'utilisateur de Keras n'a pas à la démarrer directement.

Comme indiqué précédemment, Keras repose essentiellement sur les trois outils TensorFlow, Theano et Microsoft Cognitive Toolkit, qui disposent d'ores et déjà d'**interfaces prêtes à l'emploi** permettant un **accès rapide et intuitif à l'infrastructure concernée**. Vous n'avez donc pas à choisir un framework et pouvez

aisément passer d'une infrastructure à l'autre. Il est également possible de choisir une infrastructure alternative ne faisant pas partie des trois solutions mentionnées précédemment. Pour cela, il suffit que cette infrastructure soit indiquée dans le **fichier de configuration** (*keras.json*) et dispose des trois fonctions « placeholder », « variable » et « fonction ».

Les avantages du Keras :

Keras facilite considérablement la constitution de réseaux neuronaux et se révèle un complément exceptionnel aux outils existants dans ce domaine. La **convivialité** joue un rôle prépondérant dans ce cadre : Keras fonctionne comme une interface conçue explicitement pour l'homme et, de façon secondaire, pour les machines. Dans les cas d'application les plus importants, les **actions des utilisateurs** sont **réduites à un minimum**. Si des **erreurs** devaient malgré tout survenir, un **feed-back** correspondant aidera à la résolution. Il est donc relativement simple d'apprendre et d'utiliser Keras pour une productivité accrue. Keras étant lié à des frameworks d'apprentissage profond, cette simplicité ne s'accompagne pas de restrictions fonctionnelles : vous pouvez intégrer toutes les fonctionnalités désirées via des **interfaces sur mesure** (ou les activer en cas de besoin).

Vue d'ensemble des autres avantages de Keras :

- **Les modèles élaborés sont largement compatibles avec les plateformes** : les modèles créés avec Keras peuvent être mis à disposition sur différentes plateformes en toute simplicité. Par défaut, Keras supporte par exemple les configurations iOS (Apple CoreML), Android (KerasTensorFlow Android Runtime), Google Cloud et Raspberry Pi.
- **Le support de multiples moteurs back-end** : Keras vous permet non seulement de choisir librement l'infrastructure souhaitée mais aussi d'en combiner plusieurs. D'autre part, vous pouvez à tout moment transférer un modèle créé sur une autre infrastructure.
- **Un support multi-GPU exceptionnel** : Keras permet de répartir facilement la puissance de calcul nécessaire aux processus de deeplearning élaborés sur plusieurs puces ou cartes graphiques.
- **La capacité de développement de grandes entreprises** : la maintenance et le développement de Keras sont assurés par les principaux acteurs du secteur. Google, Amazon AWS, Microsoft, Apple et Nvidia participent notamment au projet.

La destination du keras et Tensorflow keras dans les projets :

Keras est aujourd'hui utilisé dans différents projets du **domaine de l'IA** en tant qu'interface universelle pour les plateformes d'apprentissage automatique les plus diverses. Dès le milieu de l'année 2018, cette bibliothèque enregistre **plus de 250 000 utilisateurs individuels**, un nombre largement dépassé avec la reprise de la bibliothèque dans le logiciel TensorFlow. La possibilité de choisir librement le framework sous-jacent, la licence libre et l'indépendance en termes de plateforme font de Keras une solution polyvalente parfaite pour les applications de réseaux neuronaux professionnels, que ce soit dans le secteur industriel ou dans le domaine de la recherche. Des entreprises de renom comme **Netflix, Uber et Yelp**, mais aussi des organisations comme la **NASA** ou le **CERN** utilisent par exemple Keras ou le duo TensorFlow-Keras dans leurs projets.[31]

NumPy :

est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.[33]

Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

NumPy est la base de SciPy, regroupement de bibliothèques Python autour du calcul scientifique

Pandas :

est un package Python open source qui est le plus largement utilisé pour la science des données/l'analyse des données et les tâches d'apprentissage automatique. Il est construit au-dessus d'un autre package nommé NumPy, qui prend en charge les tableaux multidimensionnels. En tant que l'un des packages de gestion de données les plus populaires, Pandas fonctionne bien avec de nombreux autres modules de science des données au sein de l'écosystème Python et est généralement inclus dans chaque distribution Python, de celles fournies avec votre système d'exploitation aux distributions de fournisseurs commerciaux comme ActivePython d'ActiveState.

Matplotlib Visualisation avec Python :

Matplotlib est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python. Matplotlib rend les choses faciles et les choses difficiles possibles.[35]

- Créer des tracés de qualité de publication .

- Créez des figures interactives qui peuvent zoomer, faire un panoramique, mettre à jour.
- Personnalisez le style visuel et la mise en page .
- Exportez vers de nombreux formats de fichiers .
- Intégrer dans JupyterLab et les interfaces utilisateur graphiques .
- Utilisez un large éventail de packages tiers construits sur Matplotlib.

OS :

fait partie des modules utilitaires standard de **Python**. Ce module fournit un moyen portable d'utiliser les fonctionnalités dépendant du système d'exploitation. Les modules `* os *` et `* os.path *` incluent de nombreuses fonctions pour interagir avec le système de fichiers.

TensorFlow :

TensorFlow est une plate-forme Open Source de bout en bout dédiée au machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie

Création de modèles facilitée Construisez et entraînez facilement des modèles de machine learning à l'aide d'API intuitives de haut niveau comme Keras avec exécution eager, ce qui permet une itération immédiate des modèles et un débogage facile.

Une production de machine learning fiable en tout lieu.

Entraînez et déployez facilement des modèles dans le cloud, sur site, dans le navigateur ou sur des appareils, quel que soit le langage utilisé.

Des expérimentations probantes pour la recherche.

Une architecture simple et flexible pour amener plus rapidement les nouvelles idées du concept au code, puis à des modèles de pointe et enfin à la publication.

Créé par l'équipe Google Brain en 2011, sous la forme d'un système propriétaire dédié au réseaux de neurones de Deep Learning, TensorFlow s'appelait à l'origine DistBelief. Par la suite, le code source de DistBelief a été modifié et cet outil est

devenu une bibliothèque basée application. En 2015, il a été renommé TensorFlow et Google l'a rendu open source. Depuis lors, il a subi plus de 21000 modifications par la communication et est passé en version 1.0 en février 2017.

Pour faire simple, TensorFlow est une bibliothèque de Machine Learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissage expérimentales et de les transformer en logiciels.

On peut le concevoir comme un système de programmation dans lequel les calculs sont représentés sous forme de graphiques. Les noeuds du graphique représentent les opérations mathématiques, et les bordures représentent des flèches de données multidimensionnelles communiquées entre elles : les tensors.

TensorFlow regroupe un grand nombre de modèles et d'algorithmes de Machine Learning et de Deep Learning. Son API front-end de développement d'applications repose sur le langage de programmation Python, tandis que l'exécution de ces applications s'effectue en C++ haute-performance.

Cette bibliothèque permet notamment d'entraîner et d'exécuter des réseaux de neurones pour la classification de chiffres écrits à la main, la reconnaissance d'image, les plongements de mots, les réseaux de neurones récurrents, les modèles sequence-to-sequence pour la traduction automatique, ou encore le traitement naturel du langage.

TensorFlow permet aux développeurs de **créer des graphiques de dataflow (dataflow graphs)**, à savoir des structures qui décrivent la façon dont les données se déplacent sur un graphique ou une série de noeuds de traitement.

Chaque noeud du graphique représente une opération mathématique, et chaque connexion entre ces noeuds est une **flèche de données multidimensionnelle : un tensor**.

Les noeuds et les tensors sont des objets Python, et les applications TensorFlow sont elles-mêmes des applications Python. Cependant, les opérations mathématiques en

elles-mêmes ne sont pas effectuées en Python. Les bibliothèques de transformation disponibles sur TensorFlow sont **écrites en C++ haute-performance**. Ainsi, le Python se contente de diriger le trafic entre les éléments et fournit des abstractions de programmation de haut niveau pour les connecter entre eux.

Les applications TensorFlow peuvent être exécutées sur une machine locale, un cluster sur le Cloud, des appareils mobiles iOS ou Android, ou même des CPU et des GPU. Sur la Google Cloud Platform, il est possible d'exécuter TensorFlow **sur la TPU (TensorFlow Processing Unit)** pour profiter d'une importante accélération.

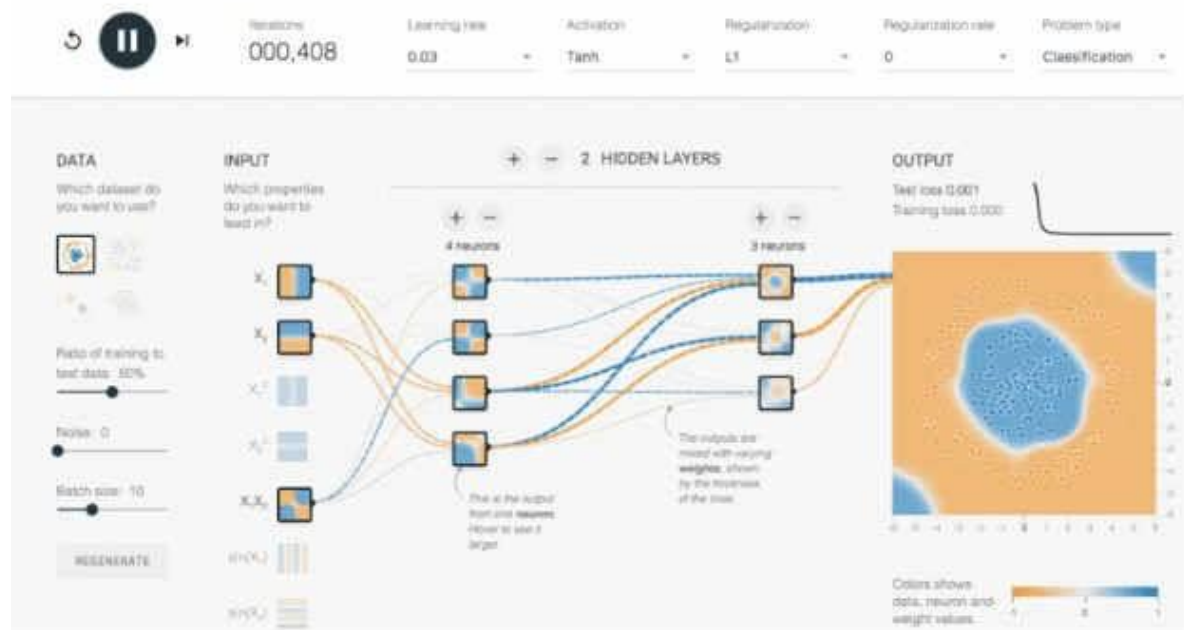


Figure 13.tensorflow

Les avantages de TensorFlow sont multiples. Tout d'abord, **grâce à l'abstraction**, ce framework facilite l'implémentation d'algorithmes et permet au développeur de se focaliser sur la logique générale d'une application.

En outre, **le mode »eagerexecution«** permet d'évaluer et de modifier toutes les opérations de graph de façon séparée et transparente plutôt que d'avoir à construire l'intégralité du graphique en tant qu'objet opaque unique et de tout évaluer en une fois. De même, **la suite de visualisation TensorBoard** permet d'inspecter la façon dont les graphiques fonctionnent par le biais d'un tableau de bord interactif basé web.

En outre, **grâce au soutien de Google**, le déploiement et l'utilisation de TensorFlow sont facilités. En plus du TPU proposé sur Google Cloud pour accélérer les performances, le géant californien propose aussi un hub online permettant de partager les modèles créés avec TensorFlow ainsi que des versions mobiles et web du framework. Un autre avantage est son caractère open source, personnalisable, et modulaire.

Parmi ses inconvénients, on peut citer le fait qu'un même modèle entraîné sur deux systèmes différents **présentera parfois des variations** même si les données avec lesquelles il a été nourri sont exactement les mêmes. En outre, TensorFlow **ne prend pas en charge OpenCL**. Enfin, il **requiert des connaissances** en calculus avancé et en algèbre linéaire ainsi qu'une importante compréhension du Machine Learning.

2. À propos de l'ensemble de données:

L'ensemble de données est une collection d'images d'alphabets de la langue des signes américaine, séparés en 29 dossiers qui représentent les différentes classes.[31]

L'ensemble de données d'apprentissage contient 87 000 images de 200 x 200 pixels. Il existe 29 classes, dont 26 pour les lettres AZ et 3 classes pour *SPACE*, *DELETE* et *NOTHING*.

Ces 3 classes sont très utiles dans les applications en temps réel et la classification.

L'ensemble de données de test ne contient que 29 images, pour encourager l'utilisation d'images de test du monde réel.

cette base de données et sur la plateforme kaggle sous le nom "ASL ALPHABET"

3. L'approche proposée :

Dans ce travail, nous avons proposé l'utilisation de la technique d'apprentissage par transfert pour la reconnaissance de la langue des signes américaine qui se résume à l'adaptation du modèle préalablement formé à notre problème, afin qu'il puisse détecter nos objets souhaités. Nous avons choisi dans cette étude le modèle inception V3 [33]. Dans cette section, nous décrivons la création V3.

Inception :

L'un des réseaux les plus complexes est Inception qui a été construit par Google. Il a été conçu de façon à améliorer les performances des réseaux de neurones profonds à la fois en termes de rapidité et de précision. Son évolution a conduit à la création de plusieurs versions du réseau. Les versions populaires sont les suivantes : GoogLeNet ou Inception-v1 [219], Inception-v2 [220], Inception-v3 [220], et Inception-v4 [218]. Chaque version est une amélioration itérative par rapport à la précédente. La compréhension des mises à niveau peut nous aider à créer des classificateurs personnalisés optimisés en vitesse et en précision. L'architecture de ce réseau se base essentiellement sur des modules d'inceptions, d'où son nom. Chaque version de ce réseau a abouti à la construction d'un nouveau module. Inception-v1 a présenté le premier module d'inception. Celui-ci a été créé suite à 3 problèmes. Premièrement, la difficulté de transmettre les mises à jour de gradient à travers un réseau trop profond.

Deuxièmement, le fait d'empiler naïvement de grandes opérations de convolution, augmente le temps de calcul. Et troisièmement, vu que l'emplacement des informations dans une image est très varié, il devient difficile de choisir la bonne taille du noyau pour l'opération de convolution. Un noyau plus gros est préférable pour les informations distribuées plus globalement, et un noyau plus petit est préféré pour les informations distribuées plus localement. Pour cela, un module d'inception a été mis en place permettant de tester plusieurs configurations de convolution avec des filtres de tailles multiples (1×1 , 3×3 , 5×5) sur le même niveau à travers des convolutions en parallèle. Ensuite, une couche de pooling est appliquée. Pour rendre le réseau moins coûteux en termes de temps de calcul, les auteurs limitent le nombre de canaux d'entrée en ajoutant une convolution supplémentaire de 1×1 avant les convolutions 3×3 et 5×5 . Cependant, cette couche est ajoutée après la couche de pooling. Aussi, ils ont factorisé les convolutions avec des filtres de 5×5 en deux convolutions de 3×3 , car les couches de 5×5 sont 2,78 fois plus coûteuses que les couches de 3×3 . Les sorties sont ensuite concaténées et envoyées au module suivant. Cette dernière modification a été introduite lors de la construction du réseau Inception-v2, d'où la construction du module d'inception A (Figure 3.12a). Un autre type de factorisation consistait à remplacer les convolutions de taille de filtre $n \times n$ en une combinaison de convolutions $1 \times n$ et $n \times 1$ ce qui limite la complexité de calcul. En fixant n à 7, le module d'inception B (Figure 3.12b) prend place lors de la construction de Inception-v3. Le fait d'empiler les filtres les uns après les autres réduit excessivement les dimensions et donc il y aura une perte d'informations. Dans ce cas une autre technique a été introduite afin de remplacer un filtre par deux autres en parallèle.

Pourquoi Inception V3 :

Néanmoins, cette technique implique une augmentation de la consommation de ressources, car il y a plus de paramètres calculés à ce stade que s'il n'y avait qu'un seul filtre. Pour réduire la consommation de ressources qu'implique l'utilisation de plusieurs filtres, une astuce mathématique est utilisée pour réduire la dimension de l'image analysée par les filtres 3×3 et 5×5 , en ajoutant une étape de filtrage 1×1 avant. De cette manière, la convolution de taille 1×1 effectue une opération de mise en commun sur les valeurs d'un pixel dans l'espace dimensionnel de l'image. Par exemple, si l'image d'entrée a 3 canaux (RVB), alors après 1×1 convolution, l'image de sortie aura 1 canal. La réduction de dimension permet de conserver l'information, tout en réduisant le coût des ressources lors de la prochaine étape de convolution avec un filtre de taille 3×3 . En imagerie médicale, l'architecture Inception est souvent utilisée pour la classification d'anomalies, notamment pour définir le stade de la maladie (ex : cancer du poumon, cancer du rein, inflammation du complexe ostioméatal). Des dérivés de cette architecture ont été rapidement développés, comme Xception ou Inception-ResNet. Tous deux obtiennent de bons résultats dans la classification des images médicales (par exemple, classification des images CT des poumons pour la détection du covid-19 ou des nodules pulmonaires malins). Xception

remplace les modules Inception par des modules de convolution séparables en profondeur et ajoute des liens résiduels. Ce type d'approche réduit considérablement l'utilisation des ressources lors du calcul matriciel sans modifier le nombre de paramètres. InceptionResNet intègre un lien résiduel dans les modules Inception (voir schéma ci-dessous), et améliore significativement les performances du modèle en classification par rapport aux versions précédentes de l'architecture inception.

le modèle est créé à l'aide de l'[API Estimator](#).

L'API simplifie considérablement la création de modèles en encapsulant la plupart des fonctions de bas niveau. Vous pouvez ainsi vous concentrer sur le développement de modèles, et non sur le fonctionnement interne du matériel sous-jacent qui exécute des tâches.

Description du modèle :

Inception v3 est un modèle de reconnaissance d'images capable d'obtenir une précision significative. Il s'agit de l'aboutissement de nombreuses idées développées par plusieurs chercheurs au fil des années. Il est basé sur l'article original "Rethinking the Inception Architecture for Computer Vision" (Repenser l'architecture Inception pour la vision par ordinateur) de C. Szegedy et al.[37]

Le modèle est constitué d'une combinaison de composants de base symétriques et asymétriques, dont les suivants :

- Convolutions
- Pooling moyen
- Pooling maximal
- concatenate
- Abandons
- Couches entièrement connectées

La perte est calculée à l'aide de Softmax.

L'image suivante présente une vue d'ensemble du modèle :

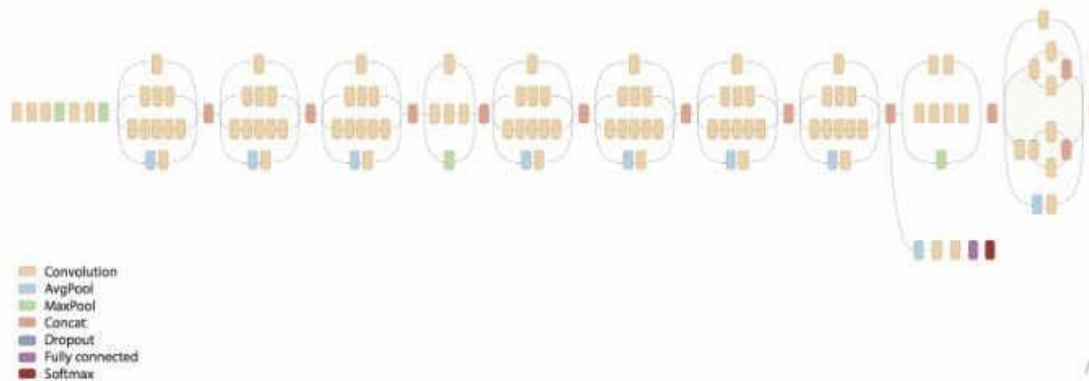


Figure 14. inception v3

Inception v3 est un modèle de reconnaissance d'images qui a prouvé une précision supérieure à 78,1% sur l'ensemble de données ImageNet. Ce modèle est l'aboutissement de nombreuses idées développées par plusieurs chercheurs au fil des ans. Il est basé sur l'article original Rethinking the Inception Architecture for Computer Vision (Repenser l'architecture Inception pour la vision par ordinateur) de Szegedy, et. al.

Le modèle lui-même est constitué de composants de base symétriques et asymétriques, y compris les convolutions, le regroupement moyen, le pooling maximal, les concaténations, les pertes d'énergie et les couches entièrement connectées. La normalisation des lots est utilisée de manière intensive dans le modèle et appliquée aux entrées d'activation. La perte est calculée à l'aide de Softmax.

La capture d'écran suivante montre le schéma général du modèle:

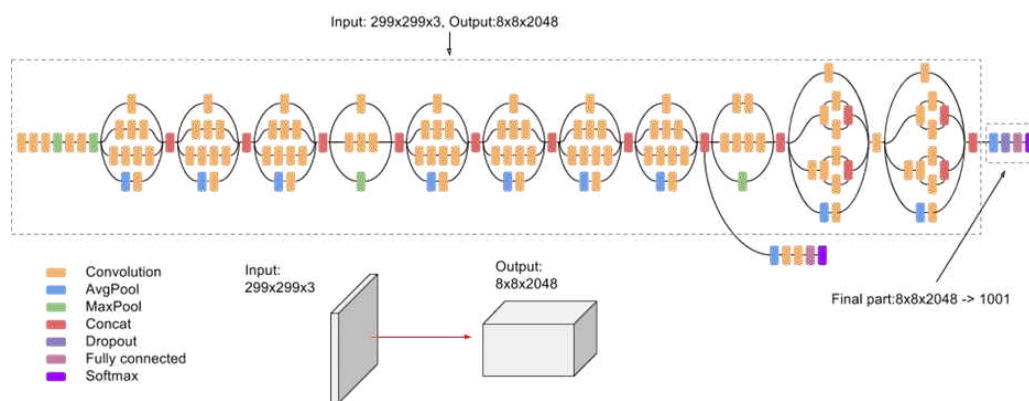


Figure 15. shemagénéraleinception v3

La version TPU de Inception v3 est codée à l'aide de [TPUEstimator](#), une API conçue pour faciliter le développement et faire en sorte que vous puissiez vous concentrer sur les modèles eux-mêmes plutôt que sur les détails du matériel sous-jacent. L'API effectue en arrière-plan l'essentiel du travail de bas niveau nécessaire à l'exécution de

modèles sur les TPU, tout en automatisant des fonctions courantes telles que l'enregistrement et la restauration des points de contrôle.

L'API Estimator fait respecter la séparation des sections de code relatives au modèle et aux données.

```
def model_fn(features, labels, mode, params):
    ...
    return tpu_estimator.TPUEstimatorSpec(mode=mode, loss=loss, train_op=train_op)

def input_fn(params):
    def parser(serialized_example):
        ...
        return image, label

    ...

    images, labels = dataset.make_one_shot_iterator().get_next()
    return images, labels
```

Les deux fonctions principales fournies par l'API sont `train()` et `evaluate()` utilisées pour l'entraînement et l'évaluation, comme indiqué dans le code suivant:

```
def main(UNUSED_argv):
    ...
    run_config = tpu_config.RunConfig(
        master=FLAGS.master,
        model_dir=FLAGS.model_dir,
        session_config=tf.ConfigProto(
            allow_soft_placement=True, log_device_placement=True),
        tpu_config=tpu_config.TPUCConfig(FLAGS.iterations, FLAGS.num_shards,))

    estimator = tpu_estimator.TPUEstimator(
        model_fn=model_fn,
        use_tpu=FLAGS.use_tpu,
        train_batch_size=FLAGS.batch_size,
        eval_batch_size=FLAGS.batch_size,
        config=run_config)

    estimator.train(input_fn=input_fn, max_steps=FLAGS.train_steps)

    eval_results = inception_classifier.evaluate(
        input_fn=imagenet_eval.input_fn, steps=eval_steps)
```

4. Ensemble de données ImageNet

Avant de pouvoir utiliser le modèle pour reconnaître des images, il doit être entraîné à l'aide d'un grand ensemble d'images étiquetées. [ImageNet](#) est un ensemble de données commun à utiliser.

ImageNet contient plus de dix millions d'URL d'images avec libellés. Un million d'images possèdent également des cadres de délimitation spécifiant un emplacement plus précis pour les objets étiquetés.

Pour ce modèle, l'ensemble de données ImageNet est composé de 1 331 167 images partagées en deux ensembles de données d'entraînement et d'évaluation, contenant respectivement 1 281 167 et 50 000 images.

La séparation entre ensembles de données d'entraînement et d'évaluation est intentionnelle. Seules les images de l'ensemble de données d'entraînement sont utilisées pour entraîner le modèle, et seules les images de l'ensemble de données d'évaluation sont utilisées pour évaluer la justesse du modèle.

Le modèle s'attend à ce que les images soient enregistrées sous forme de fichiers TFRecords. Pour convertir des images à partir de fichiers JPEG bruts au format TFRecord, utilisez le script Open Source de traitement par lots : [download and preprocess imagenet.sh](#). Le script doit produire une série de fichiers (aussi bien pour l'entraînement que pour la validation) au format suivant :

```

${DATA_DIR}/train-00000-of-01024
${DATA_DIR}/train-00001-of-01024
...
${DATA_DIR}/train-01023-of-01024
and
${DATA_DIR}/validation-00000-of-00128
${DATA_DIR}/validation-00001-of-00128
...
${DATA_DIR}/validation-00127-of-00128
```

où DATA_DIR est l'emplacement hébergeant l'ensemble de données, par exemple : DATA_DIR=\$HOME/imagenet-data

4. Pipeline d'entrée :

Chaque appareil Cloud TPU possède huit cœurs et est connecté à un hôte (processeur). Les tranches plus conséquentes possèdent plusieurs hôtes. D'autres configurations plus conséquentes interagissent avec plusieurs hôtes. Ainsi, la v2-256 communique avec 16 hôtes.

Les hôtes récupèrent les données depuis le système de fichiers ou la mémoire locale, effectuent tout prétraitement requis sur les données, puis transfèrent les données prétraitées vers les cœurs de TPU. Nous considérons individuellement chacune de ces trois phases de traitement des données effectuées par l'hôte et nous les désignons par les termes : 1) *stockage*, 2) *prétraitement*, 3) *transfert*. La figure suivante illustre une vue d'ensemble du diagramme:

Transfert Learning pour la reconnaissance du langage des signes

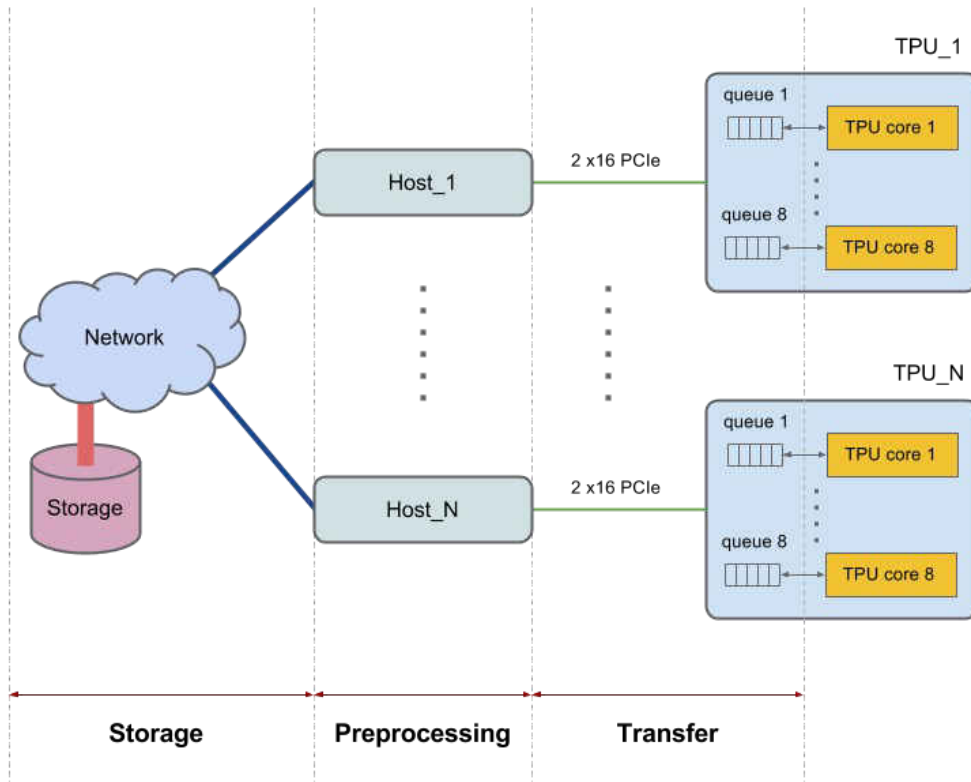


Figure 16.pipeline d'entrée

Pour optimiser les performances, le système doit être équilibré. Si le processeur hôte prend plus de temps que le TPU pour terminer les trois phases de gestion des données, l'exécution est alors liée à l'hôte. Les deux cas sont présentés dans le schéma suivant:

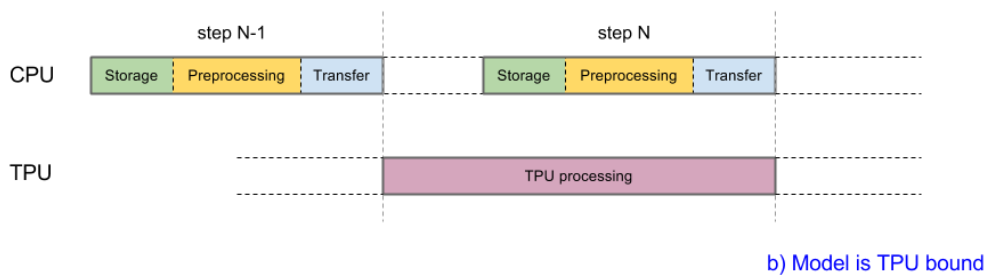
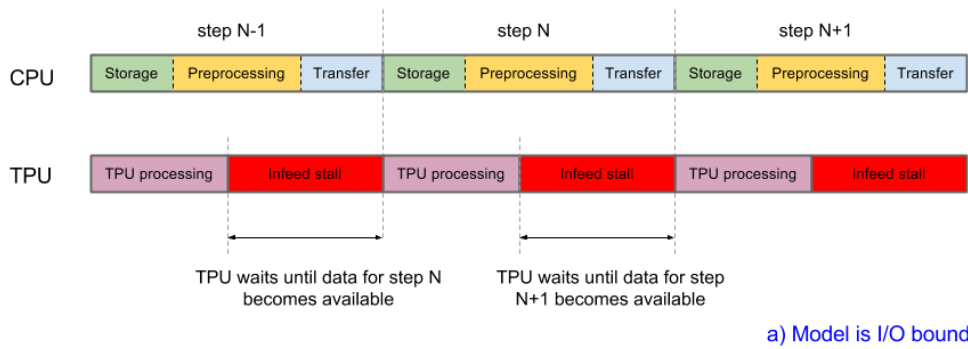


Figure 17.TPU

L'implémentation actuelle d'Inception v3 est à la limite d'une limite d'entrées. Les images sont extraites du système de fichiers, décodées, puis prétraitées. Différents types d'étapes de prétraitement sont disponibles, allant de la phase modérée à la phase complexe. Si nous utilisons les étapes de prétraitement les plus complexes, le pipeline d'entraînement est lié à la phase de prétraitement. Vous pouvez obtenir une précision supérieure à 78,1% à l'aide d'une étape de prétraitement modérément complexe qui maintient le modèle lié au TPU.

Le modèle utilise `tf.data.Dataset` pour gérer le traitement du pipeline d'entrée. Pour en savoir plus sur l'optimisation des pipelines d'entrée, consultez le [guide relatif aux performances des ensembles de données](#).

Bien que vous puissiez définir une fonction et la transmettre à l'API Estimator, la classe `InputPipeline` encapsule toutes les fonctionnalités requises.

L'API Estimator facilite l'utilisation de cette classe. Il suffit de la transmettre au paramètre `input_fn` des fonctions `train()` et `evaluate()`, comme indiqué dans l'extrait de code suivant:

```
def main(UNUSED_argv):
    ...

    inception_classifier = tpu_estimator.TPUEstimator(
        model_fn=inception_model_fn,
        use_tpu=FLAGS.use_tpu,
        config=run_config,
        params=params,
        train_batch_size=FLAGS.train_batch_size,
        eval_batch_size=eval_batch_size,
        batch_axis=(batch_axis, 0))
    ...

    for cycle in range(FLAGS.train_steps // FLAGS.train_steps_per_eval):
        tf.logging.info('Starting training cycle %d.' % cycle)
        inception_classifier.train(
            input_fn=InputPipeline(True), steps=FLAGS.train_steps_per_eval)

        tf.logging.info('Starting evaluation cycle %d.' % cycle)
        eval_results = inception_classifier.evaluate(
            input_fn=InputPipeline(False), steps=eval_steps, hooks=eval_hooks)
        tf.logging.info('Evaluation results: %s' % eval_results)
```

La section concernant le **stockage** commence par la création d'un ensemble de données et comprend la lecture des fichiers TFRecord à partir de l'espace de stockage (à l'aide de `tf.data.TFRecordDataset`). Les fonctions spéciales `repeat()` et `shuffle()` sont utilisées si nécessaire. La fonction `tf.contrib.data.parallel_interleave()` mappe la fonction `prefetch_dataset()` sur son entrée pour produire des ensembles de données imbriqués et renvoie leurs éléments entrelacés. Elle récupère les éléments des ensembles de données imbriqués `cycle_length` en parallèle, ce qui augmente le débit. L'argument `sloppy` assouplit la contrainte exigeant que les sorties soient produites dans un ordre déterministe et permet à la mise en œuvre de sauter les ensembles de données imbriqués dont les éléments ne sont pas immédiatement disponibles sur demande.

La section concernant le **prétraitement** appelle `dataset.map(parser)`, qui à son tour appelle la fonction d'analyseur dans laquelle les images sont prétraitées. Les détails de la phase de prétraitement sont abordés à la section suivante.

La section concernant le **transfert** (à la fin de la fonction) comporte la ligne return images, labels. TPUEstimator prend les valeurs renvoyées et les transmet automatiquement à l'appareil.

La figure suivante montre un exemple de trace des performances d'Inception v3 dans Cloud TPU. Le temps de calcul des TPU, en ignorant les blocages d'ingestion, est d'environ 815 ms.

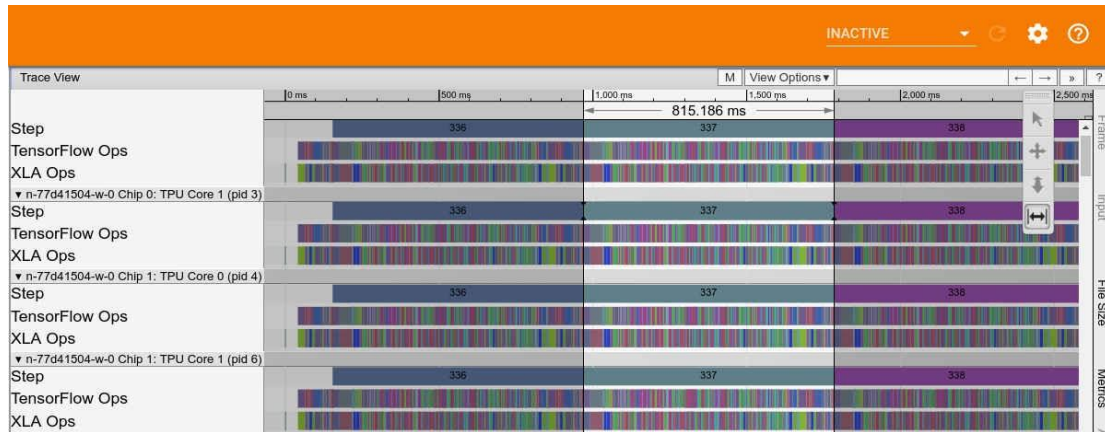


Figure 18. performance inception v3

Le **stockage** de l'hôte est écrit dans la trace et illustré dans la capture d'écran suivante:

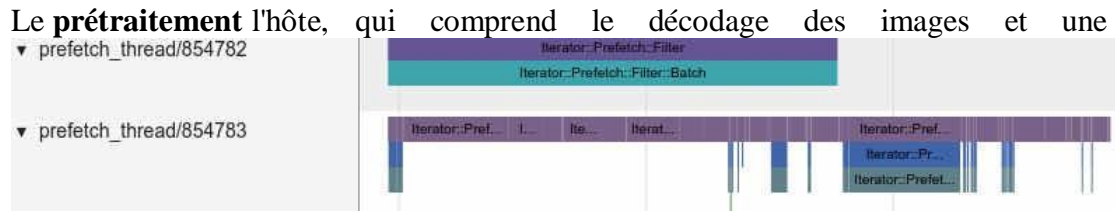


Figure 19. performance inception v3

série de fonctions de distorsion d'images, est illustré dans la capture d'écran suivante:

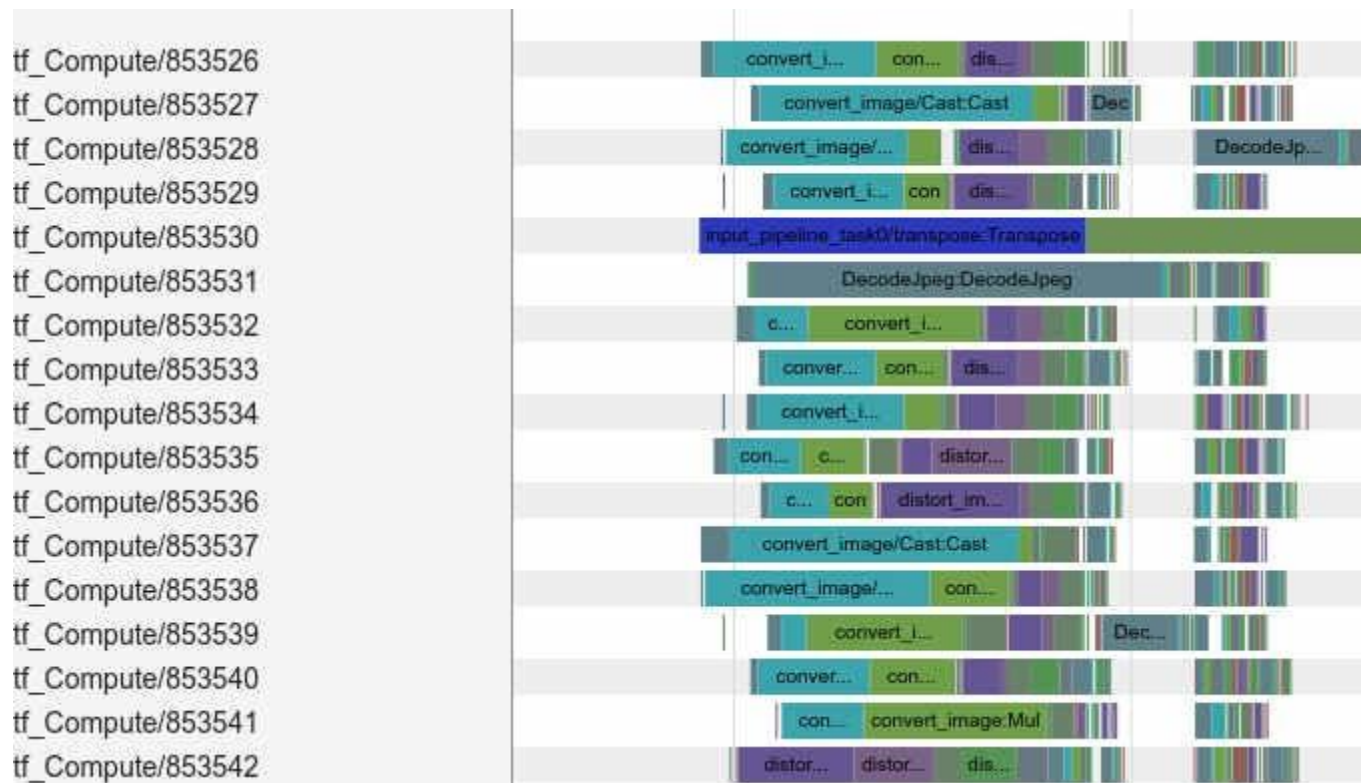


Figure 20. performance inception v3

L'hôte/le **transfert** TPU est illustré dans la capture d'écran suivante:

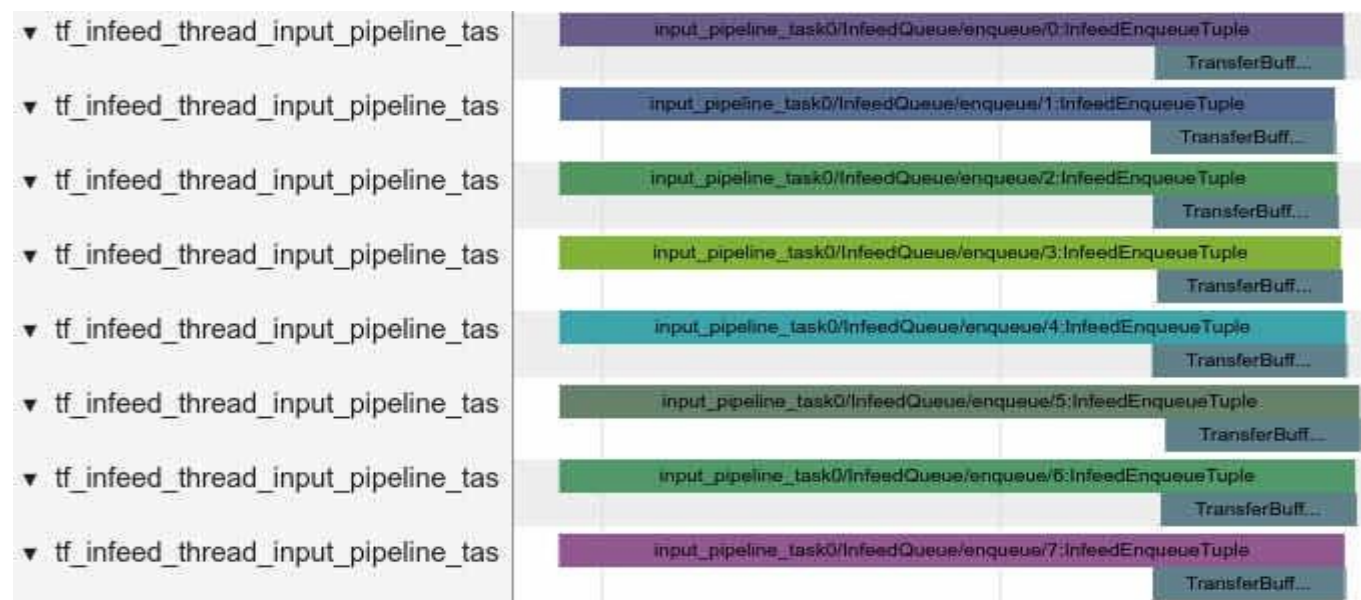


Figure 21. performance inception v3

Étape de prétraitement :

Le prétraitement des images est un élément essentiel du système. Il peut influencer sur la justesse maximale atteinte par le modèle pendant l'entraînement. Les images doivent

être au moins décodées et redimensionnées pour s'adapter au modèle. Pour Inception, les images doivent être au format 299 x 299 x 3 pixels.[37]

Toutefois, le simple décodage et le redimensionnement ne suffisent pas pour obtenir une bonne précision. L'ensemble de données d'entraînement ImageNet contient 1 281 167 images. Une passe effectuée sur l'ensemble des images d'apprentissage correspond à une époque. Pendant l'entraînement, le modèle nécessite plusieurs passages dans l'ensemble de données d'entraînement pour améliorer ses capacités de reconnaissance d'images. Pour entraîner Inception v3 avec une justesse suffisante, utilisez entre 140 et 200 époques en fonction de la taille globale du lot.[37]

Il est utile de modifier en continu les images avant de les transmettre au modèle afin qu'une image particulière soit légèrement différente à chaque époque. La meilleure façon d'effectuer ce prétraitement des images est autant d'art que de science. Une étape de prétraitement bien conçue peut considérablement améliorer les fonctionnalités de reconnaissance d'un modèle. Une étape de prétraitement trop simple peut entraîner un plafond artificiel sur la précision que le même modèle peut atteindre pendant l'entraînement.

Inception v3 propose des options pour la phase de prétraitement, allant de la simplicité et du calcul à un prix relativement faible à celui d'une tâche relativement coûteuse.

.Les fichiers **vgg_preprocessing.py** et **inception_preprocessing.py** illustrent deux scénarios de prétraitement distincts.

Le fichier **vgg_preprocessing.py** définit une étape de prétraitement qui a été utilisée avec succès pour entraîner resnet avec une justesse de 75 %, mais qui produit des résultats sous-optimaux lorsqu'elle est appliquée à Inception v3.

Le fichier **inception_preprocessing.py** contient une étape de prétraitement qui a été utilisée pour entraîner Inception v3 avec une précision comprise entre 78,1 et 78,5% sur des TPU.

Le prétraitement diffère selon que le modèle est en cours d'entraînement ou utilisé pour l'inférence ou l'évaluation.

Au moment de l'évaluation, le prétraitement est simple: recadrez une région centrale de l'image, puis redimensionnez-la à la taille par défaut de 299 x 299.

Pour Inception v3, les tests montrent que RMSProp donne les meilleurs résultats en termes de justesse maximale et de temps requis pour l'atteindre, suivi de près par Momentum. Ainsi, RMSprop est défini comme l'optimiseur par défaut. Les paramètres utilisés sont : decay $\alpha = 0,9$, momentum $\beta = 0,9$ et $\epsilon = 1,0$.

5. Résultats obtenus :

Tableau 1. résultat obtenue

Les expérimentations de ces résultats dans un PC i5 4 ram.

(tableau 1. résultat obtenue)	Training Accuracy	Cross Entropy	Validation Accuracy	Test Accuracy
100	99	0.98	96	76.4
200	100	0.53	96	83
500	100	0.41	96	81.3
1000	100	0.24	97	89.7
1500	100	0.08	97	91
2000	100	0.08	97	91.4
2500	100	0.06	97	92.3
4000	100	0.02	98	98.6

Tableau 1. résultats obtenus

Comme on le voit dans le tableau 1, le nombre d'itérations a affecté les résultats obtenus de manière à ce que si nous ajoutons plus d'itérations, le réseau de neurones construit de meilleurs modèles. Et cela est clair en termes de trois premières mesures :

- En termes de Training Accuracy : Cette mesure est la précision d'application du modèle sur les données d'apprentissage, elle est utilisée pour évaluer le modèle lors des étapes de rétropropagation afin d'améliorer le modèle. Dans le tableau 1, nous avons cité la meilleure précision d'apprentissage obtenue, comme on le voit, le modèle est parfait puisqu'il classe correctement 99% à 100% des données d'apprentissage. La figure 1 montre une mise à jour détaillée de la précision de la formation à chaque itération ou (a), (b), (c), (d), (e), (f), (g) et (h) présentent 100, 200, 500, 1000, 1500, 2000, 2500 et 4000

itérations respectivement.

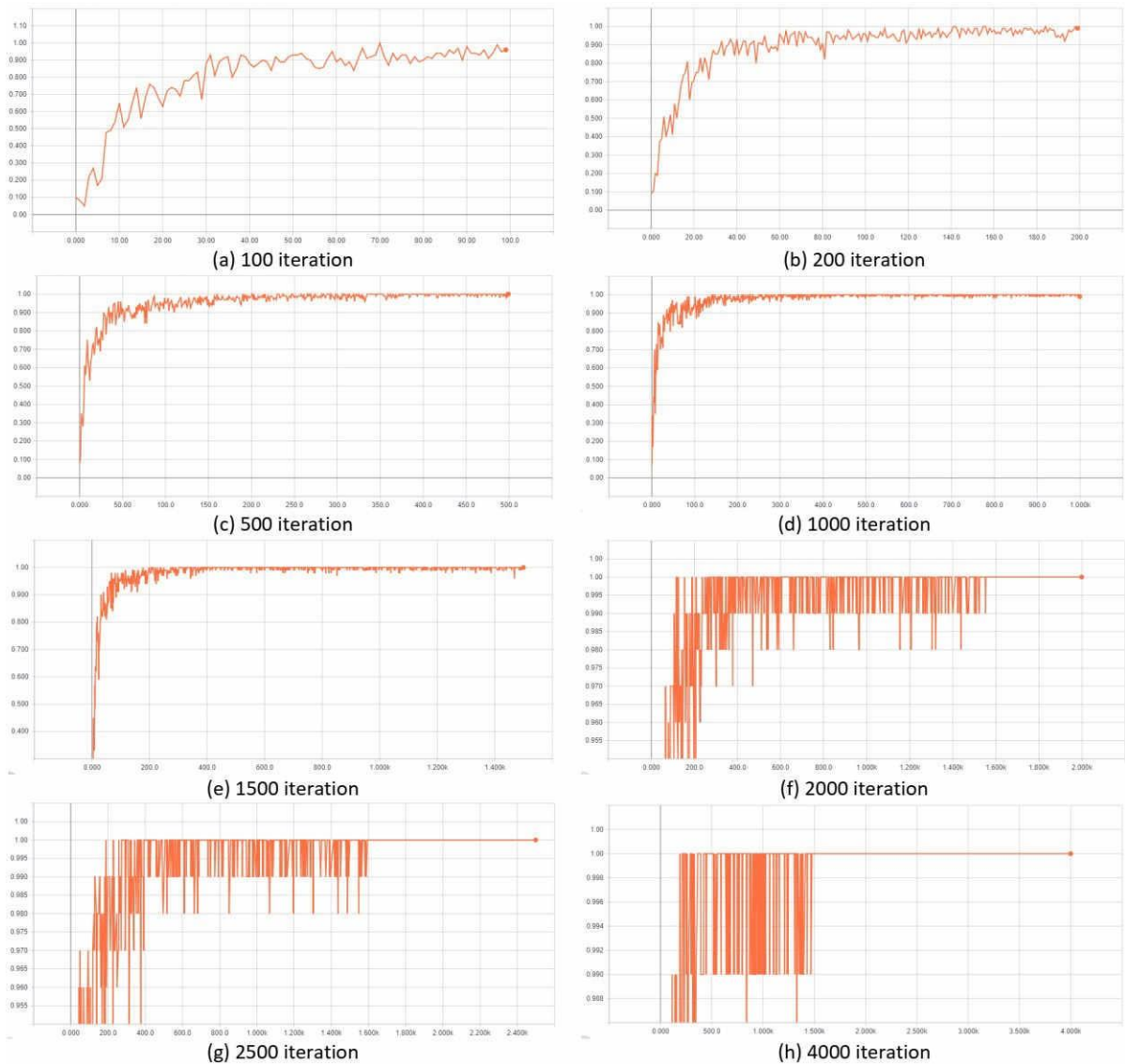


Figure 22. la précision de l'apprentissage

-Pour tous les cas, on voit sur la figure 22 comment la précision de l'apprentissage commence avec de petites valeurs (environ 5%), puis elle s'améliore itération après itération pour atteindre 100%.

-En termes de Cross Entropy : Lorsque nous utilisons la perte d'entropie croisée lors de la formation de réseaux de neurones, nous calculons en fait la fonction de score a chaque fois lorsque nous calculons des gradients pour les poids dans le réseau. Ainsi, l'objectif est de minimiser cette mesure, comme le montre le tableau 1, lorsque nous avons ajouté plus d'itérations, le réseau de neurones a minimisé l'entropie croisée, ce qui signifie que nous avons obtenu de meilleurs modèles. La figure 2 montre une mise à jour détaillée de l'entropie croisée au cours de chaque itération ou (a), (b), (c), (d), (e), (f), (g) et (h) présentent 100, 200, 500, 1000, 1500, 2000, 2500 et 4000 itérations respectivement.

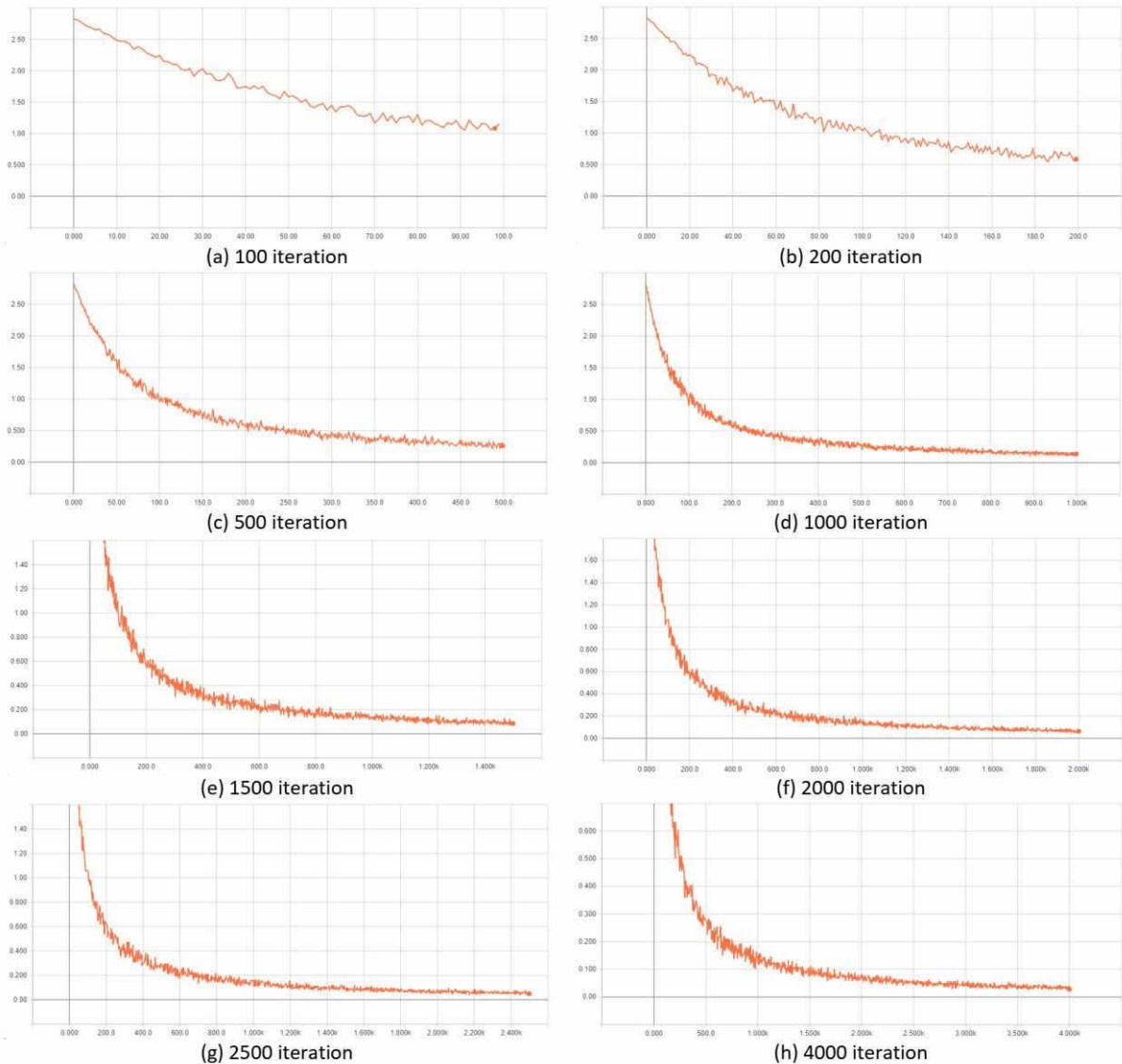


Figure 23. valeurs d'entropie

Pour tous les cas, on voit sur la figure 23 comment l'entropie commence par des valeurs élevées (environ 1), puis elle s'améliore itération après itération pour converger vers 0. En détaillant les figures, on voit que plus on ajoute d'itérations,

plus l'entropie croisée devient plus faible ce qui signifie que nous avons réduit la perte d'informations en améliorant le modèle.

- En termes de Validation Accuracy : Cette mesure est la précision d'application du modèle sur les données de validation, en tant que précision d'apprentissage, elle est également utilisée pour évaluer le modèle lors des étapes de rétropropagation afin d'améliorer le modèle. Dans le tableau 1, nous avons cité la meilleure précision de validation obtenue, comme on le voit, le modèle est parfait puisqu'il classe correctement 95% à 100% des données d'apprentissage. La figure 23 montre une mise à jour détaillée de la précision de la validation au cours de chaque itération ou (a), (b), (c), (d), (e), (f), (g) et (h) présentent 100, 200, 500, 1000, 1500, 2000, 2500 et 4000 itérations respectivement.

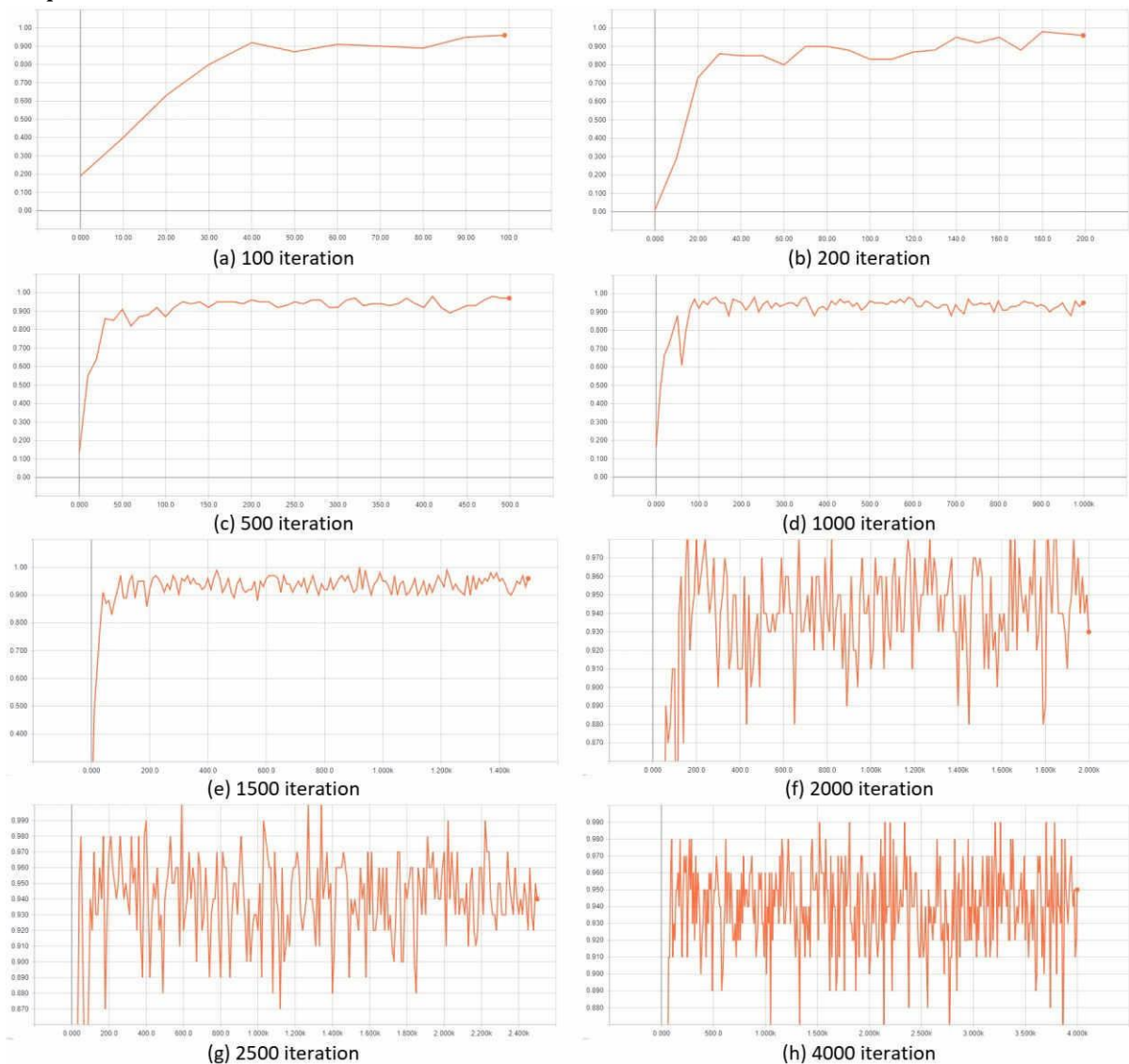
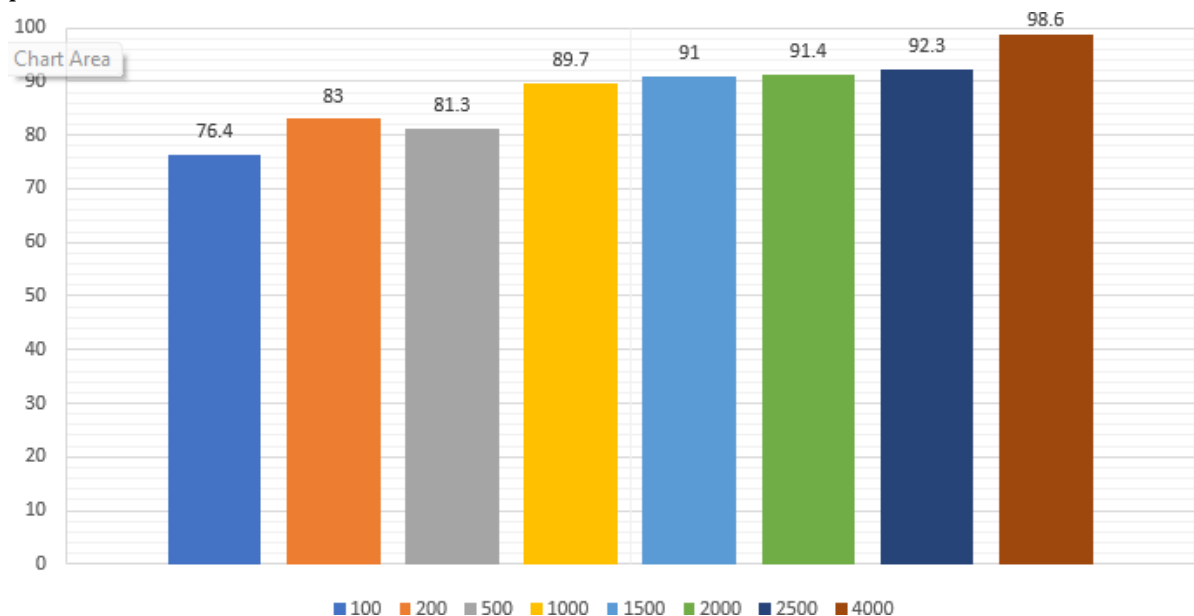


Figure 24. valeurs de la précision

- Pour tous les cas, on voit sur la figure 24 comment la précision de la validation commence par de petites valeurs (environ 5%), puis elle s'améliore itération après itération pour atteindre 100%. De plus, comme la précision de validation est inférieure à la précision de la formation, nous avons validé que notre modèle ne souffrait pas de problème de sous-ajustement.
- En termes de Test Accuracy : Cette mesure est la précision de l'application du modèle final sur les données de test, elle est utilisée pour évaluer la prédiction de nouvelles images. La figure 4 montre la comparaison de la précision du test en fonction du nombre d'itérations.



Dans notre cas, le modèle reconnaissait 76,4% des images de test lorsque nous construisions un modèle en 100 itérations, alors qu'il était amélioré lorsque nous augmentions le nombre d'itérations à 200 itérations pour reconnaître 83% des grains de pollen, puis il s'améliorait mieux et mieux jusqu'à ce qu'il atteigne 98,6 % pendant que nous avons entraîné le modèle en 4000 itérations.

6. Etude comparative :

Jeu de données utilisé L'ensemble de données ouvert donné à Kaggle appelé reconnaissance de la langue des signes américaine, qui contient un ensemble de 29*29 images standard American SignLanguage (ASL). Les données contiennent un total de 27 455 cas. Un exemple d'ensemble d'images peut être observé. Résultats obtenus Le tableau 2 montre la comparaison avec quelques travaux antérieurs.

Tableau 2. Etude comparative

(tableau2.etude comaparative)	descreption	accuracy
Das.A et al	transferlearningusinginception v3 on custom dataset	90.0%
Alashhab , S, et al	transferlearning in multiple arcitecture like VGGNET, RESNET,etc on custom 5 classes of hand gestures .	99.45%
kania K et al	transferlearningusingwideresidual Network with data augmentation on ASLs	93.3%
Gracia and Viesca	CNN ON 24 aslswithgooglenettransferlearning	70%
bousbai, k , Merrah , M.	compare custom CNN model and transferlearningusingMobileNET V2 on ASLs	97.06%
KIN YUN LUM YEH HUANN GOH , YI BIN LEE	CUSTOM CNN ON 24 ASLs using phone camera	95%
ouici, a , I, chibani D 2021	transferlearningusinginception V3 ON 29 CLASSES OF ASLs	98.6%

Tableau2. Etude comparative

Pour avoir une comparaison équitable, seuls les travaux mettant en œuvre l'apprentissage par transfert ont été comparés. Dans la comparaison, il n'y avait qu'une seule étude réalisée par Alashhab. Régler. Al. avaient une précision légèrement meilleure que ce travail, mais ils ne travaillaient pas avec l'ASL et il n'y avait que 5 classes de signes de la main utilisés. En dehors de cela, ce travail a montré de meilleurs résultats que les travaux précédents. Pour une architecture d'apprentissage par transfert MobileNetV2 similaire pour l'ASL, il existe également une amélioration marginale par rapport au travail effectué par Bousbai, K et Merah, M. ,Kin Yun Lum *, Yeh Huann Goh, Yi Bin Lee, 2020 ont utilisé l'apprentissage par transfert en utilisant MobileNetV2 sur 29 classes d'ASL et sa précision de 98,67%.

Conclusion :

Dans ce chapitre, nous présentons le logiciel utilisé lors du développement de ce système, nous avons testé et expérimenté les réseaux profonds Inception-V3 dans la base de données qui a été utilisée en considérant les 29 classes. Dans ce cas, nous avons obtenu des résultats encourageants.

On a remarqué que la validation accuracy est toujours inférieure à la training accuracy, ce qui signifie qu'on n'a pas le problème de sur-apprentissage.

CONCLUSION GÉNÉRALE :

Cette thèse avait pour but d'explorer un domaine de recherche qui passionne de nombreux chercheurs, et surtout être en avance sur une tendance qui est le langage des signes et l'Intelligence Artificielle.

Pour ce travail, nous avons utilisé le réseau de neurones convolutifs comme méthode, et ce choix de méthode se justifie par la simplicité et l'efficacité de ses méthodes. On a proposé comme un modèle Inception V3. Les résultats que nous avons récupérés confirment l'efficacité de notre stratégie. Ce projet est encore ouvert à d'autres améliorations.

BIBLIOGRAPHIE :

- [1] 2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 octobre 2017, Bali, Indonésie SignLanguage Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output Suharjitoa*, Ricky Andersonb, Fanny Wiryanab, Meita Chandra Aristab, GedePutraKusumaa.
- [2] site internet : goreact, histoire de la langue des signes
- [3] Kendon A. Geste : l'action visible comme énoncé. La presse de l'Université de Cambridge; Cambridge, Royaume-Uni : 2004.
- [4] Stefanidis K., Konstantinidis D., Kalvourtzis A., Dimitropoulos K., Daras P. Progrès récents en imagerie, modélisation et reconstruction 3D. IGI Global ; Hershey, PA, États-Unis : 2020. Technologies et applications 3D en langue des signes ; p. 50–78.
- [5] Kudrinko K., Flavin E., Zhu X., Li Q. Reconnaissance de la langue des signes basée sur des capteurs portables : un examen complet. IEEE Rév. Biomédical. Ing. 2020 ;14 : 82–97. doi : 10.1109/RBME.2020.3019769. [PubMed]
- [6] Nimisha K., Jacob A. Un bref examen des tendances récentes en matière de reconnaissance de la langue des signes ; Actes de la conférence internationale IEEE 2020 sur la communication et le traitement du signal (ICCSP); Virtuel. 16-18 mars 2021 ; pp. 186–190.
- [7] Safeel M., Sukumar T., Shashank K., Arman M., Shashidhar R., Puneeth S. Techniques de reconnaissance de la langue des signes—Un examen ; Actes de la Conférence internationale IEEE 2020 pour l'innovation technologique (INOCON); Bangalore, Inde. 19-21 novembre 2021 ; p. 1–9.
- [8] Sun Z. Avancées en informatique, communication et sciences informatiques. Springer ; Singapour : 2021. Une enquête sur la reconnaissance dynamique de la langue des signes ; pp. 1015-1022.
- [9] Wadhawan A., Kumar P. Systèmes de reconnaissance de la langue des signes : une revue systématique de la littérature sur une décennie. Cambre. Calcul. Méthodes Ing. 2021 ; 28 : 785–813. doi : 10.1007/s11831-019-09384-2.

[10] Ayadi K., ElHadj YO, Ferchichi A. Traduction automatique de l'arabe vers la langue des signes arabe : un examen ; Actes de la Conférence internationale conjointe IEEE 2018 JCCO sur les TIC dans l'éducation et la formation, Conférence internationale sur l'informatique en arabe et Conférence internationale sur la géoinformatique (JCCO : TICET-ICCA-GECO) ; Tunisie, Afrique du Nord. 9-11 novembre 2018 ; p. 1–5.

[11] Grover Y., Aggarwal R., Sharma D., Gupta PK Systèmes de traduction en langue des signes pour les malentendants et les malentendants : un examen ; Actes de la conférence internationale IEEE 2021 sur les pratiques innovantes en technologie et gestion (ICIPTM) ; Noida, Inde. 17-19 février 2021 ; p. 10–14.

[12] Falvo V., Scatalon LP, Barbosa EF Le rôle de la technologie dans l'enseignement et l'apprentissage des langues des signes : une cartographie systématique ; Actes de la conférence IEEE Frontiers in Education (FIE) 2020 ; Upsala, Suède. 21-24 octobre 2020 ; p. 1–9.

[13] Rastgoo R., Kiani K., Escalera S. Reconnaissance de la langue des signes : une enquête approfondie. *Système expert. Appl.* 2020 ; **164** :113794. doi : [14]1016/j.eswa.2020.113794.

[15] Nicholas Adologlou, et. al., “A Comprehensive Study on SignLanguage Recognition Methods”, *Methods.* arXiv 2020, arXiv:2007.12530., 2020 [2] Amit Moryoseff, et. al., “Real time SignLanguage Detection using Human Pose Estimation”, *Proceedings of the European Conference on Computer Vision (ECCV)*, SignLanguage Recognition, Translation and Production (SLRTP) Workshop, 2020

[16] Mohit Patil, et. al., “Indian SignLanguage Recognition”, *International Journal of Scientific Research & Engineering Trends*, 2020

[17] L'article de Jie Huang et. al. [17], signlanguage recognition, generation and transition

[18] Shobhit Sinha et. al. Conference on Advances in Computing and Data Sciences ICACDS 2019: Advances in Computing and Data Sciences pp 22–31 Cite as Real Time Prediction of American SignLanguage Using Convolutional Neural Network

[21] <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501333-deep-learning-definition-et-principes-de-l-apprentissage-profond/.80/02/202212:12>

[22] <https://www.natural-solutions.eu/blog/la-reconnaissance-d'image-avec-les-reseaux-de-neurones-convolutifs.10/02/202210:03>

[23] <https://dataanalyticspost.com/deep-learning-vieille-histoire-reserve-surprises/.10/02/202215:12>

[24] <https://tel.archives-ouvertes.fr/tel-03022934/document.11/02/202211:30>

[20] <https://tel.archives-ouvertes.fr/tel-03022934/document.10/03/202212:40>

- [56] S. P. Curram and J. Mingers. Neural Networks, DecisionTree Induction and Discriminant Analysis : an EmpiricalComparison. Journal of the OperationalResearch Society, 45(4) :440–450, Apr. 1994.
- [173] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, editors. Machine Learning, Neural and Statistical Classification. Ellis Horwood, UpperSaddle River, NJ, USA, 1994.
- [194] C. Robert, C. Guilpin, and A. Limoge. Comparisonbetweenconventional and neural network classifiers for rat sleep-wake stage discrimination. Neuropsychobiology, 35(4) :221–225, 1997.
- [238] B. Widrow, D. E. Rumelhart, and M. A. Lehr. Neural Networks : Applications in Industry, Business and Science. Commun. ACM, 37(3) :93–105, Mar. 1994.
- [26] G. P. Zhang. Neural networks for classification : a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 30(4) :451–462, 2000.
- [27] <https://datascientest.com/convolutional-neural-network.11/02/202215:30>
- [26] <https://datascientest.com/fonctionnement-des-reseaux-neurones.13/02/202216:00>
- [25] <https://datascientest.com/transfer-learning.12/04/202214:20>
- [28] <https://datascientest.com/deep-learning-definition.20/02/202215:30>
- [19] <file:///C:/Users/nsc/Desktop/M%C3%A9moire-Ghediri-Sakri.pdf>.
- [30] <https://www.sciencedirect.com/science/article/abs/pii/S0181551221005842.13/03/20227:15>
- [31] <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-que-keras/.13/03/202210:30>
- [32] <https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/12/04/20229:30>
- [33] <https://fr.wikipedia.org/wiki/NumPy11/05/202217:50>
- [34] <https://www.tensorflow.org/?hl=fr21/05/202218:16>
- [35] <https://matplotlib.org/21/05/202219:15>
- [36] <https://www.kaggle.com/datasets/datamunge/sign-language-mnist/metadata21/05/202220:15>
- [37] <https://cloud.google.com/tpu/docs/inception-v3-advanced22/05/202220:40>

