FEDERATEUR

RAPPORT: MINI-PROJET "RASKALNY"

Rania Swessi , Mohamed TLILI ,Wafa nouri

Etudiants en MPDS 2

Faculté des Sciences de Bizerte

Université de Carthage

November 2021

Contents

1 Introduction générale

Sauver la planète grâce à l'intelligence artificielle ? Les débats sur la protection de l'environnement et l'urgence climatique sont omniprésents et au cœur de nos préoccupations. Alors que le compte à rebours a déjà démarré et que nous commençons à vivre à crédit sur notre planète, l'IA est une piste envisagée par beaucoup d'institutions. L'heure est donc à la recherche de solutions pour résoudre ces défis de transition environnementale. Ainsi, les outils de collecte, d'analyse de données et d'aide à la décision auront un rôle central à jouer dans cette lutte contre le réchauffement climatique.

2 Chapitre 1 : Présentation générale et état de l'art

Ce chapitre introductif à comme objectif de mettre notre travail dans son contexte général. Tout d'abord, nous commençons de faire une présentation de l'organisme

d'accueil. Ensuite, nous présentons le sujet en détaillant la problématique et la solution proposée.

2.1 Contexte général du projet

Dans cette partie nous présentons d'abord le cadre du projet.

2.1.1 Domaine du projet

intelligence artificielle

L'intelligence artificielle ou IA s'applique à tous les secteurs d'activité : transports, santé, énergie, industrie, logistique, finance ou encore commerce. Cloud, véhicule autonome, compteurs intelligents... utilisent tous des algorithmes performants pour fournir des réponses efficaces, fiables et personnalisées aux utilisateurs. Associant matériels et logiciels, l'intelligence artificielle mobilise des connaissances multidisciplinaires : électronique (collecte de données, réseaux de neurones), informatique (traitement de données, apprentissage profond), mathématiques (modèles d'analyse des données) ou sciences humaines et sociales pour analyser l'impact sociétal induit par ces nouveaux usages. L'essentiel sur les enjeux industriels et sociétaux majeurs de l'intelligence artificielle.

Science des données Le sujet de ce projet tutoré s'inscrit dans le domaine très vaste de la science des données, en anglais data science. Cette discipline récente s'appuie sur des outils mathématiques, statistiques et informatiques afin de traiter et d'exploiter au mieux la grande quantité d'informations dont la société moderne est submergée. Plus précisément nous avons été placés face à des problèmes d'analyse de données, d'apprentissage automatique et de prédiction, le tout avec l'aide de l'outil informatique. Nous allons détailler dans ce rapport comment nous avons fait face à ces défis et quelles méthodes nous avons employées.

Deep Learning

- Définition simple de Deep Learning : Le deep learning ou apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.
- Fonctionnement du deep Learning Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de «couches» de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit.

2.1.2 Cadre du projet

Notre sujet intitulé « conception et réalisation d'une Poubelle intelligente » . Ce stade s'inscrit dans le cadre d'un mini-projet inclus dans le module "Projet Fédérateur" en formation Master Professionnel en sciences des données "DATA SCIENCE" à la faculté des sciences de Bizerte.

2.2 Problématique

La Pollution représente un énorme problème dans notre pays, et malheureusement la Tunisie est la quatrième consommatrice de produits en plastique au monde et la troisième pays d'Afrique en termes de pollution environnementale après l'Egypte et l'Algérie.et qui produit environenviron 2,2 millions de tonnes de déchets de ménages et environ 53 000 t d'emballage par an. Ces chiffres fournis par l'AnGEd sont pour l'année 2007.

L'omniprésence du plastique dans notre environnement représente un danger sanitaire pour l'homme. En effet, ce produit est en continuelle interaction avec l'environnement humain et finit par s'infiltrer dans le corps humain, par ingestion, inhalation ou contact direct. Ainsi, de plus en plus de microfibres et microparticules plastiques sont retrouvées dans les tissus humains et le système sanguin.

Les effets sur la santé peuvent être divers : impacts sur le système immunitaire et le système respiratoire, perturbations endocriniennes, baisse de la fertilité, hausse des risques de cancers... Ces effets existent à chaque étape du cycle de vie du plastique et démultiplient donc les conséquences sur la santé. Néanmoins, l'impact de la combinaison de ces effets est encore mal connu et ne peut donc pas être appréhendé correctement.

Chaque année, vous absorbez potentiellement des dizaines de milliers de particules microplastiques(morceaux de moins de 5 millimètres de long)

Ainsi, Le plastique bénéficie d'une durée de vie longue et ne disparaît jamais totalement du milieu dans lequel il est jeté. Il se retrouve abandonné dans nos villes, en pleine nature et jusqu'au fond des océans où plus des deux tiers de nos déchets finissent par atterrir. L'équivalent de plusieurs millions de tonnes par an.

A noter toutefois que ce secteur de collecte et de gestion des déchets est encore sous-exploité en Tunisie. Des particuliers comme Abdelkader Missaoui, fondateur de l'Eco-Gad ont certes essayé d'alléger le problème de prolifération des déchets en triant les déchets recyclables. Ce promoteur diplômé en sciences de la géologie et de l'environnement croit que tout se crée et se transforme. Il est parti avec 1 000 dinars mais ses centres de tri créés en 2014 se trouvent maintenant dans plusieurs gouvernorats. De même, Tunisie recyclage, l'association créée en 2012 s'occupe aussi de ramassage de déchets triés au dépôt de la banlieuenord de Tunis. Elle couvre environ 80 foyers par semaine. Et sa collaboration avec l'Agence allemande de coopération internationale lui a permis d'envisager la création d'une centaine d'emplois. Ils sont plus de 8.000 en Tunisie selon les estimations officielles, mais difficiles à dénombrer car ils travaillent sans reconnaissance légale ni couverture sociale et sont en situation de précarité.
Ces chiffonniers se chargent de la collecte des matières recyclables : objets en plastique, en aluminium ou en verre (bouteilles et canettes notamment) , papier et carton. Invisibles aux yeux de la société mais souvent stigmatisés, ils sont devenus les « barbéchas » (« berbech » en dialecte tunisien), ceux qui farfouillent dans les poubelles pour récupérer du plastique et le vendre à des collecteurs, ceux-ci se chargeant de le revendre à des usines de recyclage dont la production est ensuite exportée vers la France, l'Allemagne, ou même la Chine ou le Brésil. Les barbéchas font un travail indispensable mais ils restent peu intégrés à l'économie légale et souvent exposés à des risques multiples. Ils doivent affronter diverses maladies du fait de leur contact permanent avec les poubelles et la saleté, mais aussi la fatigue pour ceux qui font encore leurs collectes à pied avec une brouette, le harcèlement pour les femmes, sans parler des regards stigmatisants et méprisants de leurs concitoyens. Rien que dans la cité d'Ettadhamen, près de 800 barbéchas font vivre des familles entières. Selon ces barbéchas eux-mêmes, ils seraient plus de 3.000.

Trois causes de la pollution plastique en Tunisie:

- L'inconscience
- Pas d'exploitation suffisante du plastique
- Difficulté de processus de collecte

Alors comment pour faire face à ce problème à fin de faciliter la collecte du plastique et renforcer en même temps la conscience dans notre pays et donc augmenter l'exploitation et le recyclage de ces déchets. C'est au delà notre idée est venu, et qu'on va l'expliquer tout de suite dans la section suivante.

à partir de ces besoins et problématiques on remarque un processus manquant qui va faire une mise en relation entre entre le processus de collecte et le processus de recyclage et a fin de faciliter ce mécanisme on propose notre solution "RaskelnY".

2.3 Étude de l'existant

Dans cette partie, nous allons présenter l'existant dans l'organisme d'accueil et nous enchainons par la suite par l'étude d'autres solutions existantes.

2.4 Solution proposée

Afin de faire un pas contre ce problème dont il souffre chaque citoyen Tunisien, Notre équipe à décidé de réaliser une solution intelligente à l'objectif d'encourager les citoyens tunisiens et les rappelle de jeter les déchets dans son bon emplacement, et qui aide pour le tri du plastique sans une intervention humaine, pour faciliter son recyclage , aider les collecteurs et renforcer la conscience dans notre pays et donc résoudre les trois causes principale de la pollution en Tunisie.

Comment et Quelle produit on va proposer ? Notre mission consiste à développer une poubelle intelligente qui répond efficacement aux besoins de la soucis, respecte les exigences de domaine.

Raskelny c'est une poubelle intelligente qui lancera une voie robotique lors d'une détection d'une personne afin de l'encourager de jeter les déchets s'il a, et qui font le tri de ces derniers automatiquement (en deux catégories plastiques et non plastiques) grâce à un algorithme de classification (Deep Learning)de détection d'objets.

2.5 Étapes de résolution: Détection des objets grace à une capture d'image ?

Un analyste qui tente de classer les caractéristiques d'une image, utilise les éléments de l'interprétation visuelle (discutés à la section 4.2) pour identifier des groupes homogènes de pixels qui représentent des classes intéressantes de surfaces. La classification numérique des images utilise l'information spectrale contenue dans les valeurs d'une ou de plusieurs bandes spectrales pour classifier chaque pixel individuellement.
Ce type de classification est appelé reconnaissance de regroupements spectraux. Les deux façons de procéder (manuelle ou automatique) ont pour but d'assigner une classe particulière ou thème (par exemple : eau, forêt de conifères, maïs, blé, etc.) à chacun des pixels d'une image. La "nouvelle" image qui représente la classification est composée d'une mosaïque de pixels qui appartiennent chacun à un thème particulier. Cette image est essentiellement une représentation thématique de l'image originale.

Lorsqu'on parle de classes, il faut faire la distinction entre des classes d'information et des classes spectrales. Les classes d'information sont des catégories d'intérêt que l'analyste tente d'identifier dans les images, comme différents types de cultures, de forêts ou d'espèce d'arbres, différents types de caractéristiques géologiques ou de roches, etc. Les classes spectrales sont des groupes de pixels qui ont les mêmes caractéristiques (ou presque) en ce qui a trait à leur valeur d'intensité dans les différentes bandes spectrales des données. L'objectif ultime de la classification est de faire la correspondance entre les classes spectrales et les classes d'information. Il est rare qu'une correspondance directe soit possible entre ces deux types de classes. Des classes spectrales bien définies peuvent apparaître parfois sans qu'elles correspondent nécessairement à des classes d'information intéressantes pour l'analyse. D'un autre côté, une classe d'information très large (par exemple la forêt) peut contenir plusieurs sous-classes spectrales avec des variations spectrales définies. En utilisant l'exemple de la forêt, les sous-classes spectrales peuvent être causées par des variations dans l'âge, l'espèce, la densité des arbres ou simplement par les effets d'ombrage ou des variations dans l'illumination. L'analyste a le rôle de déterminer de l'utilité des différentes classes spectrales et de valider leur correspondance à des classes d'informations utiles.

2.6 Conclusion

Tout au long de ce chapitre nous avons pris une vue plus globale sur la problématique de notre projet et la solution que nous proposons pour sa résolution.

3 Chapitre 1 : Travail réalisé

3.1 Environnement de travail

Dans cette partie, nous décrivons l'ensemble des logiciels ou bien langages utilisés pour développer notre application.

En voyant le sujet, nous pensions utiliser le langage de programmation : Python. Et nous avons choisi ce langage parce que sont utilisation est plus judicieuse grâce à la multitude de bibliothèques présentes pour le data science et précisément Machine Learning. Nous avons décidé d'utiliser la version 2.7 et non la 3.5 (plus récente) afin que les scriptes soit compatible avec l'environnement de travail de chacun.

Comme un environnemt pour le Machine Learning et qui est un service hébergé de notebooks Jupyter choisi à parce que il ne nécessite aucune configuration et permet d'accéder gratuitement à des ressources informatiques, dont des GPU.

Le logiciel open source Arduino (IDE),choisi à cause de la simplicité et la facilite de l'écriture de code et son téléchargement sur la carte. et qui est compatible avec n'importe quelle carte Arduino.

3.1.1 Les données

Une base de données multimédia est un type de base de données consacré au stockage et à l'organisation de données multimédia : images. Leur apparition et leur développement récents sont une tentative de réponse non seulement à un besoin de stockage de tels documents, mais également à un besoin de possibilités accrues pour leur traitement. Les images contiennent de nombreuses informations importantes. Si elles sont faciles à déceler pour nos yeux habitués, elles représentent un vrai challenge en analyse de données. L'ensemble de ces techniques est connu sous le nom d'« image processing » ou traitement d'image. Dans, vous découvrirez les algorithmes, techniques et outils classiques permettant de traiter les données sous forme d'images.

3.1.2 La structure

La première étape de la modélisation du problème a été de formaliser la structure de données. Notre base de données contienne des images de format "JPG". La façon dont les fichiers ont été découpés n'étant pas très intuitive nous avons divisé les données en deux parties plus lisibles et plus facilement exploitables

Nous avons utilisé ImageDataGenerator pour créer mes ensembles de données de formation, de validation et de test. Pour cela, je dois stocker les images de train, valides et de test dans trois fichiers distincts afin de transmettre ces chemins de fichiers en tant que paramètres à ImageDataGenerator. Nous avons également utilisé le tuner keras pour trouver le taux d'apprentissage optimal

Nous avons déclaré un répertoire dans lequel je stockerai les images train, valides et test. Nous avons déclaré trois autres directoires dans le nouveau répertoire pour le même.

3.2 Collecte des Données

Pour la phase de collection de donnée, Nous avons choisir la méthode de collecte par recherche, et de recueillir nos image manuellement selon notre besoin de base.Et puisque on va faire une classification en deux type de déchets :plastique re-cyclable(bouteilles), et déchets non re-cyclable catégorisé en deux:

-Des images représentant des bouteilles en plastique en différentes formes
-Des images représentant n'importe quel type de déchets: papier, canette, cigarette,carton,...

3.3 Pré-traitement et préparation de données

La sélection et la préparation ainsi que le pré traitement des données sont une phase essentielle de l'apprentissage automatique qui consiste à choisir scrupuleusement les données d'entraînement de manière à ce qu'une fois entraîné le modèle se comporte de manière attendue.

Une fois que nous avons correctement identifié et localisé nos données, on les mettre en forme pour qu'elles puissent être utilisées pour entraîner notre modèle.
Les tâches de préparation des données le nettoyage (« cleansing »), l'agrégation, l'augmentation, l'étiquetage (labels), la normalisation et la transformation ainsi que toutes les activités autour des données structurées, non structurées et semi-structurées.
Les opérations suivi lors de cette étape sont les suivantes :

- séparer les données train et test et validation en 3 répertoires
- Re-dimension des images (64,64,3)
- Normaliser les formats des différentes données.
- Supprimer les données incorrectes(image de taille 0, images inutiles).
- Mise en forme des images(Reshape, Convert To array)

La préparation des données prend beaucoup de temps. Des études menées auprès des Data Scientists estiment que cette étape peut prendre jusqu'à 80 % du temps d'un projet de machine learning. Mais comme dit l'expression : « Garbage in, garbage out ». Dit autrement, la qualité des modèles dépendra de celle des données. Le temps consacré à ces tâches de préparation et de nettoyage des jeux de données en vaut toujours la peine.

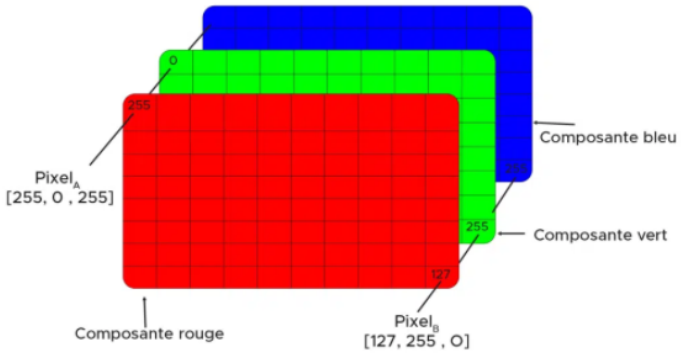


Figure 1: Illustration d'une image RVB

3.3.1 Entraînement du model

Déterminer les attributs du modèle

Cette phase nécessite de choisir le bon modèle, de l'entraîner, de régler ses hyperparamètres, de le valider, puis de passer à la phase de test et d'optimisation. Pour cela, les actions suivantes sont nécessaires :

- Choix des algorithmes en fonction de l'objectif de l'apprentissage et de ses besoins en données.
- Configuration et régler les hyperparamètres pour optimiser les performances, et choix d'une méthode d'itération pour arriver aux meilleurs hyperparamètres.
- Identification les attributs qui fourniront les meilleurs résultats.
- on détermine si l'explicabilité ou l'interprétabilité du modèle est nécessaire.
- Utilisation des plusieurs algorithmes (apprentissage ensembliste) pour améliorer les performances.

1. Détection d'objets expliquée : R-CNN
   La détection d'objets consiste en deux tâches distinctes qui sont la classification et la localisation. R-CNN signifie Region-based Convolutional Neural Network. Le concept clé derrière la série R-CNN est la proposition de région. Les propositions de région sont utilisées pour localiser des objets dans une image. Dans les blogs suivants, j'ai décidé d'écrire sur les différentes approches et architectures utilisées dans la détection d'objets. Par conséquent, je suis heureux de commencer ce voyage avec des détecteurs d'objets basés sur R-CNN.
2. SPP-Net
   SPP-Net est une architecture neuronale convolutive qui utilise la mise en commun de pyramides spatiales pour supprimer la contrainte de taille fixe du réseau. Plus précisément, nous ajoutons une couche SPP au-dessus de la dernière couche convolutive. La couche SPP regroupe les caractéristiques et génère des sorties de longueur fixe, qui sont ensuite introduites dans les couches entièrement connectées (ou d'autres classificateurs). En d'autres termes, nous effectuons une agrégation d'informations à un stade plus profond de la hiérarchie du réseau (entre les couches convolutives et les couches entièrement connectées) pour éviter le recadrage ou la déformation au début.
3. Fast-RCNN
   Fast R-CNN s'appuie sur des travaux antérieurs pour classer efficacement les propositions d'objets à l'aide de réseaux convolutifs profonds. Par rapport aux travaux précédents, Fast R-CNN utilise plusieurs innovations pour améliorer la vitesse de formation et de test tout en augmentant la précision de détection. Fast R-CNN entraîne le réseau VGG16 très profond 9 fois plus vite que R-CNN, est 213 fois plus rapide au moment du test et atteint un mAP plus élevé sur PASCAL VOC 2012. Par rapport à SPPnet, Fast R-CNN entraîne VGG16 3 fois plus vite, teste 10 fois plus rapide et plus précis. Fast R-CNN est implémenté en Python
4. VGG
   VGG signifie Visual Geometry Group ; il s'agit d'une architecture standard de réseau neuronal convolutif (CNN) à plusieurs couches. Le "profond" fait référence au nombre de couches avec VGG-16 ou VGG-19 composé de 16 et 19 couches convolutives.
   L'architecture VGG est à la base de modèles de reconnaissance d'objets révolutionnaires. Développé comme un réseau de neurones profonds, le VGGNet surpasse également les références sur de nombreuses tâches et ensembles de données au-delà d'ImageNet. De plus, c'est encore aujourd'hui l'une des architectures de reconnaissance d'images les plus populaires .
5. MLP
   Un perceptron multicouche (MLP) est une classe de réseau de neurones artificiels (RNA) à action anticipée . Le terme MLP est utilisé de manière ambiguë, parfois de manière vague pour désigner tout ANN à action directe, parfois strictement pour désigner des réseaux composés de plusieurs couches de perceptrons (avec activation de seuil) ; voir § Terminologie . Les perceptrons multicouches sont parfois appelés familièrement réseaux de neurones « vanille », en particulier lorsqu'ils ont une seule couche cachée.

Un MLP se compose d'au moins trois couches de nœuds : une couche d' entrée , une couche cachée et une couche de sortie . À l'exception des nœuds d'entrée, chaque nœud est un neurone qui utilise une fonction d'activation non linéaire . MLP utilise une technique d' apprentissage supervisé appelée rétropropagation pour la formation. [2] [3] Ses couches multiples et son activation non linéaire distinguent le MLP d'un perceptron linéaire . Il peut distinguer des données qui ne sont pas linéairement séparables .

6. CNN
   Un convolutional neural network (CNN) est un type de réseau neuronal artificiel utilisé dans la reconnaissance et le traitement d'images et spécifiquement conçu pour traiter les données de pixels.
   Les convolutional neural network sont de puissants systèmes de traitement d'images, d'intelligence artificielle (IA) qui utilisent un apprentissage approfondi (deep learning) pour effectuer des tâches à la fois génératives et descriptives, souvent à l'aide de Machine Vision qui inclut la reconnaissance d'images et de vidéos, ainsi que des systèmes de recommandation et le traitement du langage naturel (NLP).

- Test des différentes versions de notre modèle pour en vérifier les performances.
- Identification les exigences qu'il faudra respecter pour la mise en production et le déploiement du modèle.

  Un réseau neuronal convolutif, ou CNN en abrégé, comporte trois types de couches principales :

  – La couche convolutionnelle (CONV) : Elle est responsable de l'exécution de l'opération de convolution. L'élément impliqué dans l'exécution de l'opération de convolution est appelé le noyau/filtre (matrice). Le noyau effectue des décalages horizontaux et verticaux jusqu'à ce que l'image entière soit traversée. Son fonctionnement est similaire à la technique de détection des contours.
  – Couche de mise en commun (POOL) : Cette couche est responsable de la réduction de la dimensionnalité. Elle permet de diminuer la puissance de calcul nécessaire au traitement des données. Il existe deux types de mise en commun : La mise en commun maximale et la mise en commun moyenne. La mise en commun maximale renvoie la valeur maximale de la zone couverte par le noyau sur l'image. La mise en commun moyenne renvoie la moyenne de toutes les valeurs de la partie de l'image couverte par le noyau.
  – Couche entièrement connectée (FC) : La couche entièrement connectée (FC) est présente à la fin des architectures CNN. Elle est similaire à une couche traditionnelle de réseaux de neurones et permet, après application d'une fonction d'activation, de renvoyer la sortie attendue par le réseau, par exemple une classification.

  L'ajout de fonctions non linéaires (« RELU ») au sein des réseaux ou d'architectures spécifiques permettent de répondre à des problématiques plus complexes. Les exemples sont nombreux : DenseNet, U-Net, VGG…

3.4 Modélisation du problème

Dans cette partie, nous allons nous focaliser sur un des algorithmes les plus performants du Deep Learning, les Convolutional Neural Network ou CNN : Réseaux de neurones convolutifs en français, ce sont des modèles de programmation puissants permettant notamment la reconnaissance d'images en attribuant automatiquement à chaque image fournie en entrée, une étiquette correspondant à sa classe d'appartenance.

3.4.1 Architecture d'un Convolutional Neural Network-CNN

3.4.2 Convolutional Neural Network-CNN

Un convolutional neural network (CNN) est un type de réseau neuronal artificiel utilisé dans la reconnaissance et le traitement d'images et spécifiquement conçu pour traiter les données de pixels.

Les convolutional neural network sont de puissants systèmes de traitement d'images, d'intelligence artificielle (IA) qui utilisent un apprentissage approfondi (deep learning) pour effectuer des tâches à

la fois génératives et descriptives, souvent à l'aide de Machine Vision qui inclut la reconnaissance d'images et de vidéos, ainsi que des systèmes de recommandation et le traitement du langage naturel (NLP).

Un réseau de neurones est un système de matériel et/ou de logiciel conçu d'après le fonctionnement des neurones du cerveau humain. Les réseaux neuronaux traditionnels ne sont pas idéaux pour le traitement des images et doivent être alimentés en images à résolution réduite. Les « neurones » dit CNN sont disposés de manière à ressembler davantage à ceux du lobe frontal, la zone responsable du traitement des stimuli visuels chez l'homme et des autres animaux. Les couches de neurones sont disposées de manière à couvrir l'ensemble du champ visuel, évitant ainsi le problème du traitement d'images par morceaux des réseaux neuronaux traditionnels.

Un CNN utilise un système semblable à un perceptron multicouche qui a été conçu pour des besoins de traitement réduits. Les couches d'un CNN se composent d'une couche d'entrée, d'une couche de sortie et d'une couche cachée qui comprend plusieurs couches convolutionnelles, des couches de regroupement, des couches entièrement connectées et des couches de normalisation. La suppression des limitations et l'augmentation de l'efficacité pour le traitement des images aboutissent à un système beaucoup plus efficace, plus simple à former, et spécialisé pour le traitement des images et le traitement du langage naturel.

3.4.3 Architecture

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

3.4.4 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 :

- Dans un premier temps, on définit la taille de la fenêtre de filtre située en haut à gauche.
- 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.
- À chaque portion d'image rencontrée, un calcul de convolution s'effectue permettant d'obtenir en sortie une carte d'activation ou feature map qui indique où est localisées les features dans l'image : plus la feature map est élevée, plus la portion de l'image balayée ressemble à la feature.

3.4.5 Le Max pooling

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.) 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 4×4 représentant notre entrée initiale et un filtre d'une fenêtre de taille 2×2 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.

Illustrons le processus :

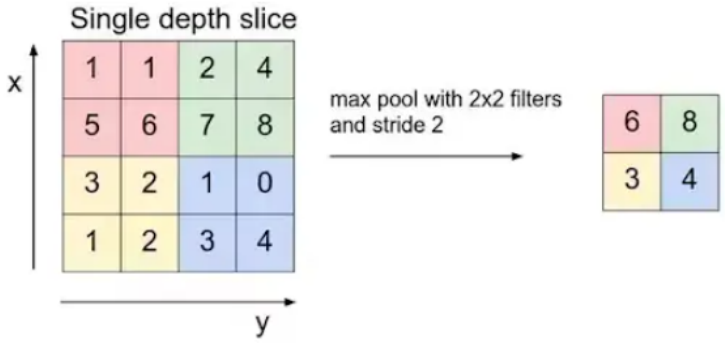


Figure 2: Processus de Max-Pooling

La fenêtre de filtre se déplace de deux pixels vers la droite (stride/pas = 2) et récupère à chaque pas "l'argmax" correspondant à la valeur la plus grande parmi les 4 valeurs de pixels.

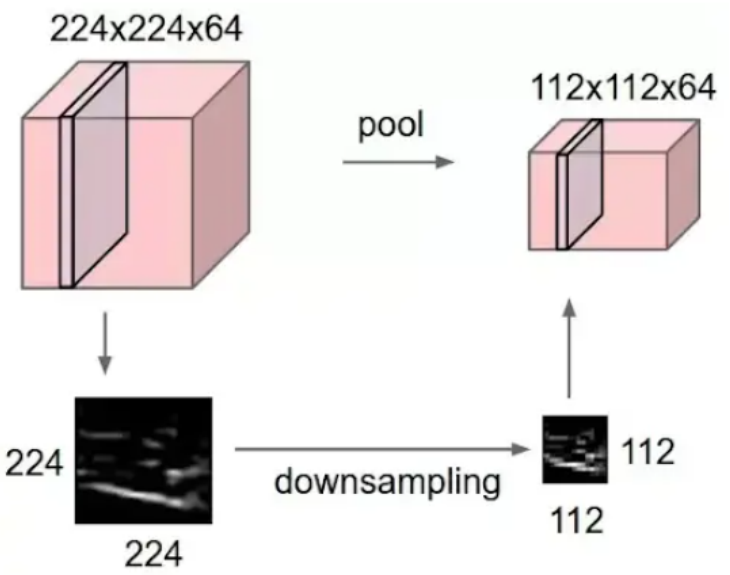


Figure 3: Exemple d'effet du Max-Pooling

On comprend mieux l'utilité de la partie convolutive d'un CNN : contrairement à un modèle MLP classique, l'ajout en amont de la partie convolutive permet d'obtenir en sortie une "carte de caractéristiques" ou "code CNN" (matrice de pixels située à droite dans l'exemple) dont les dimensions sont plus petites que celles de l'image initiale ce qui va avoir l'avantage de diminuer grandement le nombre de paramètres à calculer dans le modèle.

Exemple d'une architecture CNN et sa sortie Après la partie convolutive d'un CNN, vient la partie classification. Cette partie classification, commun à tous les modèles de réseaux de neurones, correspondant à un modèle perceptron multicouche (MLP).

Son objectif est d'attribuer à chaque échantillon de données une étiquette décrivant sa classe d'appartenance.

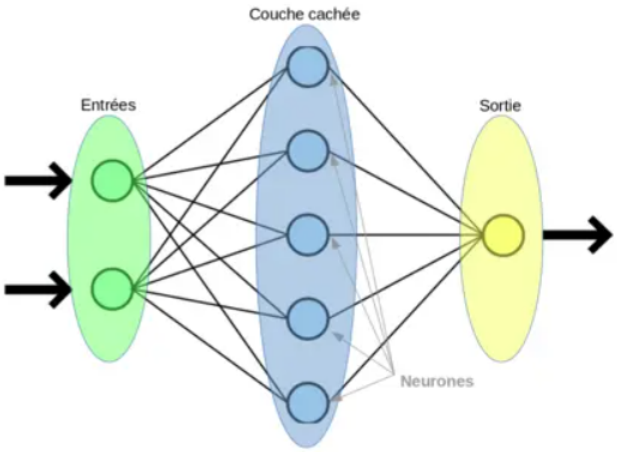


Figure 4: Représentation d'un perceptron multicouche

L'algorithme que les perceptrons utilisent pour mettre à jour leurs poids (ou coefficients de réseaux) s'appelle la rétropropagation du gradient de l'erreur, célèbre algorithme de descente de gradient que nous verrons plus en détail par la suite .

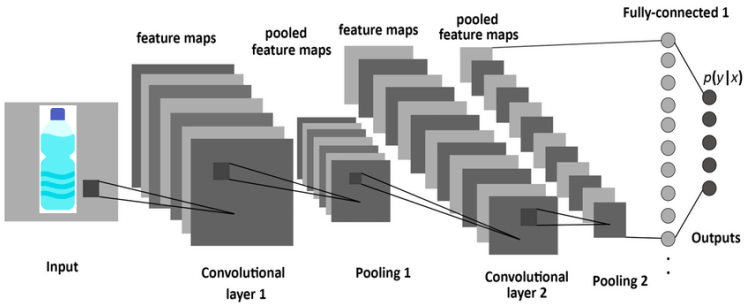


Figure 5: d'architecture de notre CNN

Généralement, l'architecture d'un Convolutional Neural Network est sensiblement la même :

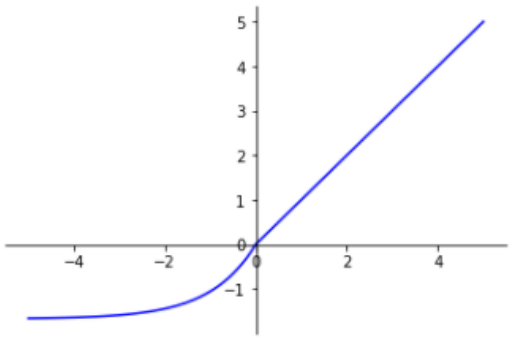
–
– Couche de convolution (CONV) : Le rôle de cette première couche est d'analyser les images fournies en entrée et de détecter la présence d'un ensemble de features. On obtient en sortie de cette couche un ensemble de features maps (cf plus haut : à quoi sert la convolution ?).
– Couche de Pooling (POOL) : La couche de Pooling est une opération généralement appliquée entre deux couches de convolution. Celle-ci reçoit en entrée les features maps formées en sortie de la couche de convolution et son rôle est de réduire la taille des images, tout en préservant leurs caractéristiques les plus essentielles. Parmi les plus utilisés, on retrouve le max-pooling mentionné précédemment ou encore l'average pooling dont l'opération consiste à conserver à chaque pas, la valeur moyenne de la fenêtre de filtre.

Finalement, on obtient en sortie de cette couche de Pooling, le même nombre de feature maps qu'en entrée mais considérablement compressées.

- La couche d'activation ReLU (Rectified Linear Units) : Cette couche remplace toutes les valeurs négatives reçues en entrées par des zéros. L'intérêt de ces couches d'activation est de rendre le modèle non linéaire et de ce fait plus complexe.

  – Fonctions d'activation

    * couches intermédiaire
      ELU
      La fonction Exponential Linear Unit (ELU) est une amélioration de ReLU car elle permet d'avoir des valeurs lisses lorsque x < 0.
      Lorsque x < 0, ELU a des valeurs négatives différents de 0 (ce qui n'est pas le cas de ReLU). Cela permet de rapprocher la moyenne de la fonction de zéro.
      Une moyenne plus proches de zéro permet un apprentissage plus rapide car cela rapprochent le gradient calculé du gradient naturel (un concept qui mérite un article entier).
      Effectivement, plus x diminue, plus ELU saturent à une valeur négative. Cette saturation implique qu'ELU a une petite dérivée ce qui diminue la variation du résultat et donc l'information qui est propagée vers la couche suivante.
      fonction ELU(x) = $\begin{cases} if x > 0 : x \\ if x < 0 : \alpha * (exp(x)\check{}1); avec : \alpha > 0 \end{cases}$



Fonction ELU – Exponential Linear Unit

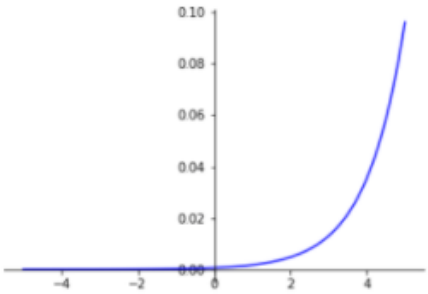
  – Couches de sortie:
    Softmax
    La fonction Softmax permet elle de transformer un vecteur réel en vecteur de probabilité.

    On l'utilise souvent dans la couche finale d'un modèle de classification, notamment pour les problèmes multiclasse.

    Dans la fonction Softmax, chaque vecteur est traité indépendamment. L'argument axis définit l'axe d'entrée sur lequel la fonction est appliquée.

    fonction$_S oftmax(x) = exp(x)/tf.reduce_sum(exp(x))$

$$\text{fonction\_Softmax}(x) = \exp(x) / \text{sum}(\exp(x_i))$$



Fonction d'activation Softmax

code: tf.keras.activations.softmax(x, axis=-1)
axis : Nombre entier, axe le long duquel la normalisation softmax est appliquée.

3.4.6 Mise en application du CNN

Implémentation d'un CNN pré-entraîné sur Python : Pour des usages pratiques et étant donné la complexité de création de performants CNN "fait à la main", nous allons utiliser des réseaux pré-entraînés disponibles dans le module Torchvision. Voyons comment celui-ci peut être implémenté sur Python :

Mise en application d'un CNN : Identification d'images quelconques à partir du dataset : images étiquetées

Notre dataset est une base de données de plus de dix millions d'images étiquetées produit par l'organisation du même nom, à destination des travaux de recherche en vision par ordinateur.

Voici un extrait de ce volumineux dataset :

Étape 0 : Tout d'abord, importons toutes les librairies qui nous serons nécessaires pour la suite Étape 1 : Entrainement du modèle pré-entrainé VGG16 Étape 2 : Importation des 3 images à prédire Étape 3 : Prétraitement des images Étape 4 : Prédiction du modèle

3.4.7 Évaluer les performances du modèl

Dans un contexte d'IA, l'évaluation comprend l'évaluation des métriques du modèle, la matrice de confusion, les KPI, les métriques de performance, la mesure de la qualité du modèle, et une validation finale de sa capacité à atteindre les objectifs métiers qui ont été fixés.

Au cours du processus d'évaluation du modèle, vous devez procéder comme suit :

- Évaluer les modèles en utilisant une approche et un jeu de données de validation.

Tableau comparatif entre les algorithmes d'apprentissage profond dans le cadre De détection d'objets

RCNN	SPP-Net	Fast-RCNN	VGG	MLP	CNN
-certains transformation sur les images est requise (chaque région de l'image doit avec la même taille)on utilisant des technique tel que le recadrage /déformation qui présente aussi des inconvénient comme la diminution de la position par cette modification -plus lent que d'autre algorithmes	-il n'est pas besoin d'une transformation d'entrées -beaucoup plus difficile à former -la couche conv calcule une fois la carte des caractéristiques de taille fixe pour l'ensemble de l'image.	-un algorithme complexe -plus rapide que RCNN -La formation dans cet algorithme n'est pas une tâche facile (calcul du dérivé,...)	- nécessite un prétraitement spécifique(calcul du valeur du RGB moyenne) sur l'ensemble d'apprentissage. -problème de précision bloqué (elle ne change pas beaucoup à cause de l'espace de paramètre énorme) -pas de technique comme batch normalisation pour améliorer notre modèle	-considère chaque neurone indépendant et affecte un poid différent à chaque signal entrant -	-diminution du nombre de paramètre à calculer dans le modèle -amélioration de performance ,réduire l'empreinte mémoire(grâce à l'utilisation d'un poid unique associés aux signaux entrant) -peu de pré-traitement (évaluer ses propres filtres tout seul:apprentissage sans supervision) -l'absence de paramétrage initial -les objets ou bien les régions dan une image 'ont pas besoin d'être localisé le plus précisément.

Tableau comparatif entre les algorithmes

- Déterminer les valeurs de la matrice de confusion dans le cadre des problèmes de classification.
- Identifier des méthodes de validation croisée si cette approche k-fold est utilisée.
- Affiner les hyperparamètres pour optimiser la performance.
- Comparer le modèle de machine learning au modèle de base (ou heuristique).

3.4.8 Conclusion

Finalement, le principe de fonctionnement d'un CNN est assez facile à comprendre, mais paradoxalement, la mise en place d'un tel procédé pour classifier des images demeure très complexe étant donné le nombre considérable de paramètres à définir : nombre, taille, déplacement des filtres, choix de la méthode de pooling, choix du nombre de couches de neurones, nombre de neurones par couches, etc.

Pour pallier à cet obstacle, Python à travers le module Torchvision, offre la possibilité d'exploiter des modèles CNN pré-entraînés performants tels que VGG16, Resnet101, etc.
Nous avons défini dans cet article le fonctionnement et l'architecture des Convolutional Neural Network, en nous concentrant sur sa spécificité: la partie convolutive.
Il nous reste encore à appréhender une étape de la classification : la rétropropagation du gradient de l'erreur, célèbre algorithme de descente de gradient.

3.5 Valeurs ajoutées

3.5.1 Obstacles (les erreurs):

- **HDF5** Le format HDF5 (pour Hierarchical Data Format, V5) est un format de type conteneur de fichier. Ce format est donc assimilable à une arborescence de dossiers/fichiers, le tout contenu dans un fichier.
  Le projet est alors renommé et le format HDF5 fait son apparition. Cependant, sa structure complexe et certaines limitations de fonctionnement (taille de fichier maximale à 2 Go) vont entraîner la création du format HDF5, simplifiant sa structure et passant outre les limitations bloquantes.
  Les possibilités qu'offre le format HDF pour effectuer des calculs complexes sur de grandes masses de données, grâce à une indexation efficace, sont à la base de son succès. De plus, en Python, l'utilisation de ce format est très simple grâce au package h5py.
  Probléme de convertion en HDF5 Dans notre travail on a rencontré certain difficultés pour convertir notre base en HDF5,et pour ne pas bloquer on a orienté vers la solution de faire entrer notre base comme des images en JPG ou PNG et de faire le pré-traitement nécessaire. Ces fichiers proviennent du challenge PICMUS et la seule
- **donnees manqantes**
  Il est important d'identifier les données manquantes dans un jeu de données avant d'appliquer un algorithme de Machine Learning (ML). En effet, beaucoup de ces derniers reposent sur des méthodes statistiques qui supposent recevoir un jeu de données complet en entrée. Faute de quoi, l'algorithme ML

risque de fournir un mauvais modèle prédictif ou pire, ne pas fonctionner tout simplement. Ainsi, traiter les données manquantes est une phase nécessaire pour tout projet de Data Science.

- **Image de taille 0**
  Le titre decrit la situation : nous avons crié notre dataset sans probleme, et lorsque nous souhaitons uploader une image pour illustrer le produit, on obtient des erreurs, pour cela on a fait le traitement nécessaire pour supprimer les images inutiles ou les images vides(taille 0).

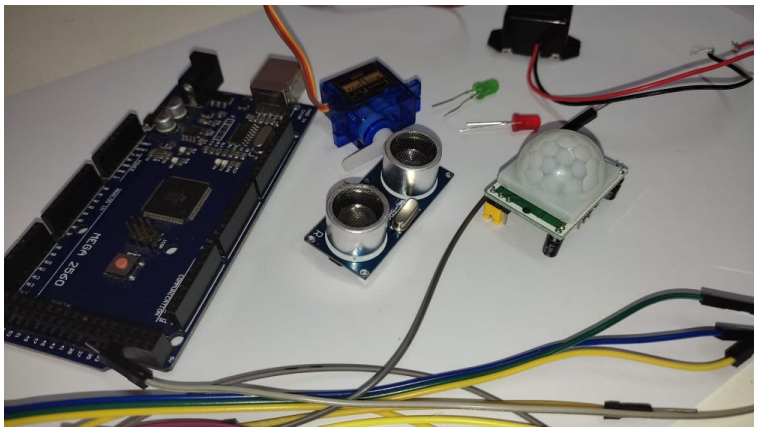
3.6 Réalisation

La réalisation de notre projet se devise en deux partie principale :

3.6.1 Partie intelligence artificielle :

Cette partie consiste à réaliser un modèle d'essai avec l'utilisation de l'intelligence artificielle dans le but de l'automatisation d'ouverture de la porte lorsque la poubelle détecte un passager grâce à un servomoteur et lancer un voix robotique (ici dans notre premièr essai nous avons utilisé un buzzer passif).

Les matérielles utiliser dans cette partie :



matériels utilisés

- 1- Carte Arduino Méga 2560:

  La carte Arduino Mega 2560 est basée sur un ATMega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs. Elle est idéale pour des applications exigeant des caractéristiques plus complètes que la Uno.

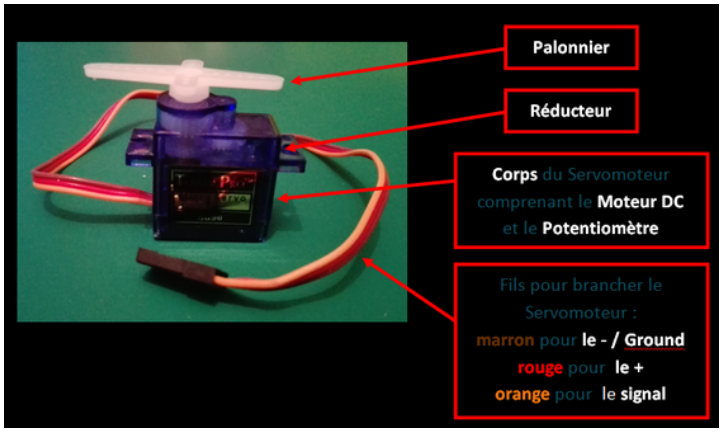
Carte Arduino Méga 2560

- 2- Capteur ultrason HC-S04 :
  Le capteur HC-SR04 est un capteur à ultrason low cost. Ce capteur fonctionne avec une tension

d'alimentation de 5 volts, dispose d'un angle de mesure de 15° environ et permet de faire des mesures de distance entre 2 centimètres et 4 mètres avec une précision de 3mm (en théorie, dans la pratique ce n'est pas tout à fait exact.

Capteur Ultrason HC-S04

- 3- Servomoteur : Un servomoteur est un type de moteur particulier. Sa fonction principale consiste à assurer la production d'un mouvement afin de répondre à une commande externe. Comme l'étymologie du terme le laisse deviner (""servomoteur"" vient du latin ""servus"", ""esclave""), le servomoteur désigne simultanément un système asservi et un actionneur utilisé pour le déclenchement d'une action. Il embarque dans le même boîtier l'électronique et la mécanique. Un servomoteur est capable de maintenir une position prédéterminée dans les instructions.

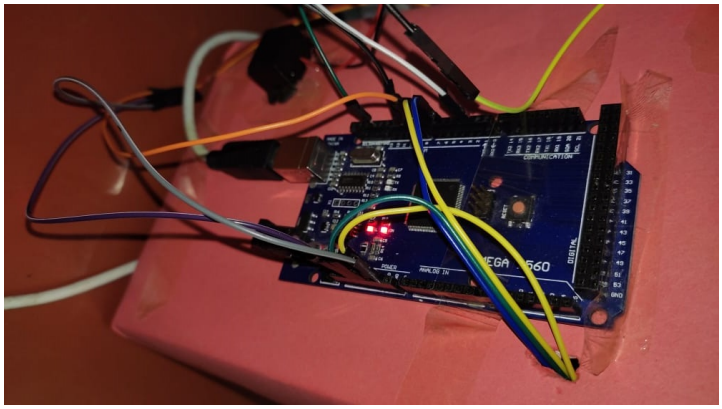


Servomoteur

- 4- Buzzer passif:
le buzzer a pour rôle de lancer un petit vibreur utilisé pour vérifier la continuité électrique des circuits dans la réalisation de notre poubelle nous le utiliser au lieu de voix robotique a cause de manque de matérielle .

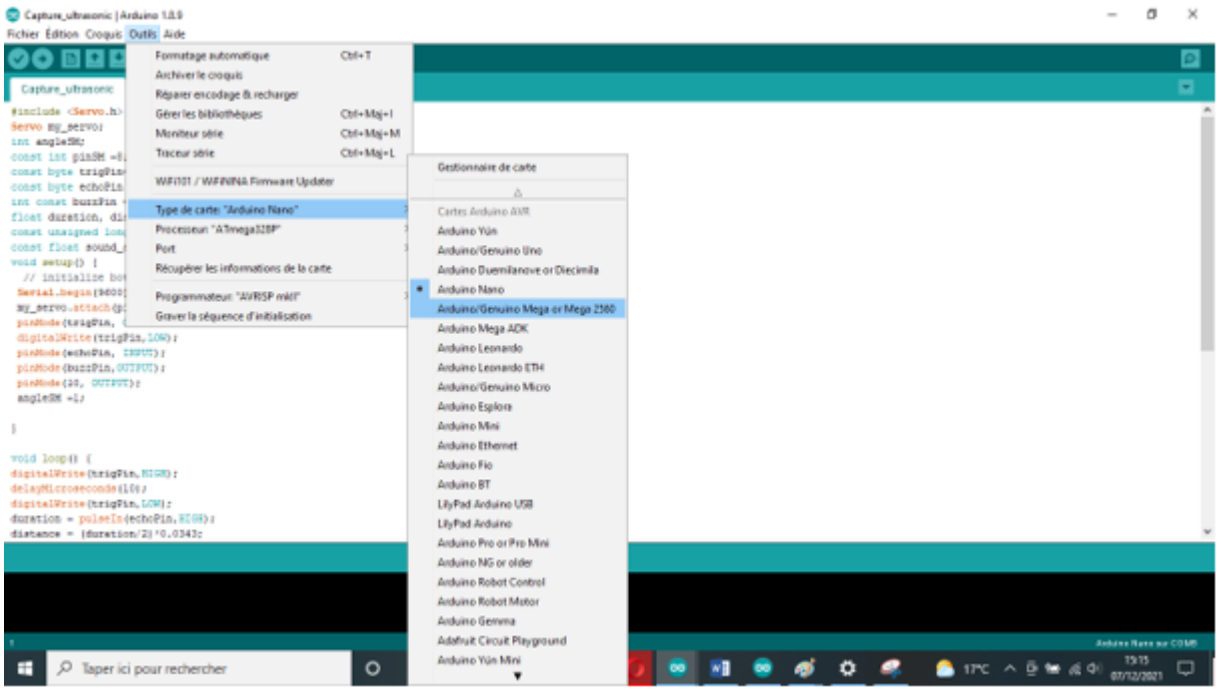
Buzzer passif 12V DC

- Branchement de matériels:
  la photo se dessous représente le branchement nécessaire:



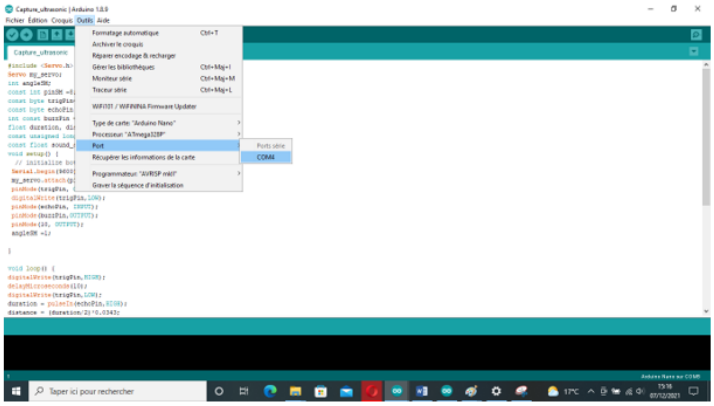
Branchement des matérielle

- Implementation du code arduino:
  Pour programmer la carte arduino méga 2650 nous utilisons l'environnement de développement Arduino IDE tout d'abord nous allons configurer l'environnement de façon qu'il sera compatible avec notre carte méga premièrement nous choisissons le type de la carte cliquez sur l'anglet Outils/Type de carte: et choisissez Arduino/Genuino Mega or Mega2560



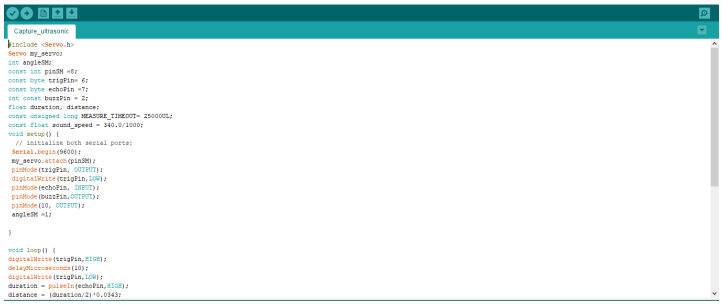
Choisir le type de la carte

Aprés la choix de carte nous choisissons aussi le port



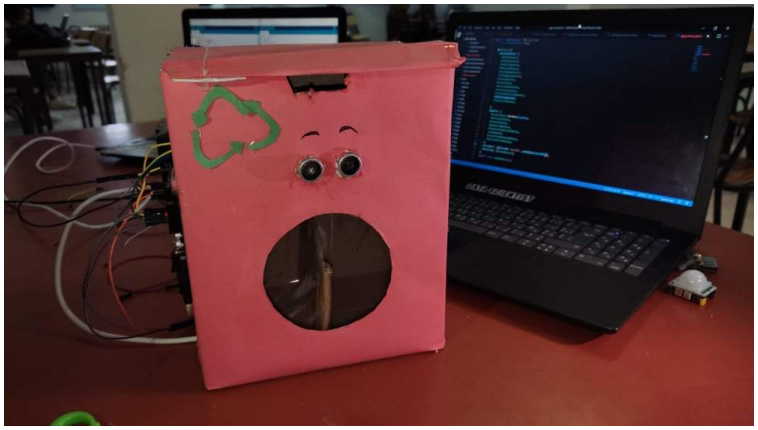
Choix de port

Notre environnement est prêt pour développer un code arduino



code Arduino

La résultat final Partie IA



Modele d'essai en carton

3.6.2 Partie déploiement de modèle Deep Learning:

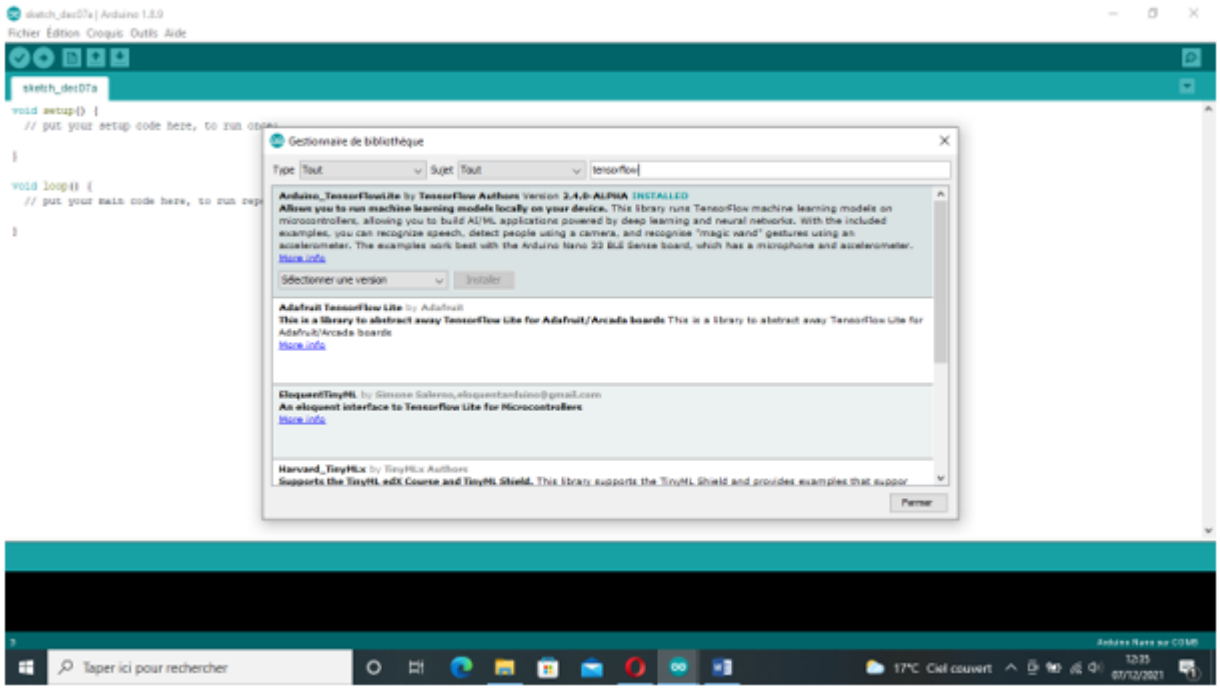
Après l'optimisation de modèle CNN nous arrivons maintenant à la phase de déployer notre modèle sur une carte électronique en utilisons la technologie de tinyMl et tensorflow for Microcontrôleurs. Mais avant approfondie dans le technique de travail laissons nous explique s'est quoi le tinyMl : Tiny Machine Learning est une technique d'apprentissage automatique qui intègre des des applications d'apprentissage automatique réduits et optimisées qui nécessitent des solutions «full-stack» (matériel, système, logiciel et applications) y compris des architectures, technique, outils et approches d'apprentissage automatique capable d'effectuer des analyses sur les appareils à la périphérique du cloud. TinyML peut être implémenté dans des systèmes à faible consommation d'énergie, tels que des capteurs ou des microcontrôleurs

Tensorflow for microcontrôleurs:
TensorFlow Lite for Microcontrollers est conçu pour exécuter des modèles de machine learning sur des microcontrôleurs et d'autres appareils ne disposant que de quelques kilo-octets de mémoire. Le composant d'exécution principal tient sur 16 Ko sur un processeur Arm Cortex M3 et peut exécuter de nombreux modèles de base. Aucune compatibilité avec le système d'exploitation n'est nécessaire. Il ne nécessite, en outre, aucune bibliothèque C ou C++ standard, ni d'allocation de mémoire dynamique. **Les cartes de développement suivantes sont acceptées à deployer le modéle :**

- Arduino Nano 33 BLE Sense
- SparkFun Edge
- STM32F746 Discovery kit
- Adafruit EdgeBadge
- Kit Adafruit TensorFlow Lite for Microcontrollers
- Adafruit Circuit Playground Bluefruit
- Espressif ESP32-DevKitC
- Espressif ESP-EYE
- Wio Terminal : ATSAMD51
- Carte de développement Himax WE-I Plus EVB Endpoint AI
- Plate-forme de développement de logiciels Synopsys DesignWare ARC EM
- Sony Spresense
-

les étapes nécessaire pour déployer un modelé ML sur une carte arduino Nano 33 BLE Sense
nous utilisons arduino ide comme outils de développement après entraîner notre modèle nous Convertissons au format TensorFlow Lite à l'aide du convertisseur TensorFlow Lite. Effectuez une conversion en tableau d'octets C à l'aide d'outils standards en vue du stockage dans une mémoire de programme en lecture seule sur l'appareil.

et dans l'environnement arduino nous téléchargons la bibliothéque tensorflow



Téléchargement de bibliothèque tensorflow

4 Conclusion

L'invention d'une poubelle de tri sélectif, qui trie automatiquement les déchets vient d'apparaître. Plus besoin de trier soi-même ses déchets, la poubelle intelligente s'en charge. Cette invention de poubelle connectée pourrait remplacer les bacs de recyclage.