République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Benyoucef BENKHEDDA-Alger1



Faculté des Sciences

Département des Mathématiques et Informatique

Projet de Fin d’Etude**s** pour l’obtention du diplôme de Licence en Informatique

Option : Systèmes Informatiques

Thème

Une approche d’apprentissage profond

Pour la détection d’objet

**Encadré par : Réalisé par :**

- Dr Aiouez Sabrina - Hamitouche Anis

- Belmadoui Mohamed Sabri

2020/2021

Remerciement

**Résume**

En vision par ordinateur on désigne par détection d’objet une méthode permettant de détecter la présence d’une instance ou d’une classe d’objets dans une image numérique.

Dans cette étude, nous avons élaboré une application mobile pour la détection et la reconnaissance des postures de la main de la langue des signes arabes basée sur le Deep Learning.

**Abstract**

En vision par ordinateur on désigne par détection d’objet une méthode permettant de détecter la présence d’une instance ou d’une classe d’objets dans une image numérique.

Dans cette étude, nous avons élaboré une application mobile pour la détection et la reconnaissance des postures de la main de la langue des signes arabes basée sur le Deep Learning.

# Table des matières:

[Table des matières: 4](#_Toc75263778)

[Introduction générale : 5](#_Toc75263779)

[Chapitre 1 : Détection d’objets dans les images : 6](#_Toc75263780)

[1. Introduction : 6](#_Toc75263781)

[2. Composants d’un système de détection d’objets : 6](#_Toc75263782)

[2.1 La localisation D’objets : 7](#_Toc75263783)

[2.2 La classification d’image : 7](#_Toc75263784)

[3. Problème liée à la détection d’objets : 7](#_Toc75263785)

[4. Domaine d’application : 7](#_Toc75263786)

[5. Reconnaissance des postures de la main de la LSA : 7](#_Toc75263787)

[6. Conclusion : 7](#_Toc75263788)

[Chapitre 2 : Réseaux de Neurones Convolutifs : 8](#_Toc75263789)

[1. Introduction : 8](#_Toc75263790)

[2. Les CNNs : 8](#_Toc75263791)

[2.1 Couche convolution : 8](#_Toc75263792)

[2.2 Couche pooling: 10](#_Toc75263793)

[2.3 Couche fully connected : 11](#_Toc75263794)

[3. Les architecture utilisées pour la détection d’objets : 12](#_Toc75263795)

[3.1 R-CNN : 12](#_Toc75263796)

[3.2 Fast R-CNN : 13](#_Toc75263797)

[3.3 Faster R-CNN : 15](#_Toc75263798)

[3.4 Yolo (you look only once): 17](#_Toc75263799)

[4. Conclusion : 18](#_Toc75263800)

[Chapitre 3 : Un système de vison pour la reconnaissance des signes de la LSA : 19](#_Toc75263801)

[1. Introduction : 19](#_Toc75263802)

[2. Présentation de l’architecture du système de détection des signes de LSA : 19](#_Toc75263803)

[3. Datasets : 19](#_Toc75263804)

[3.1 Construction des données : 19](#_Toc75263805)

[3.2 Prétraitements des données : 19](#_Toc75263806)

[3.3 Annotation des images : 19](#_Toc75263807)

[4. Mesure des performances : 19](#_Toc75263808)

[5. Outils de développement : 19](#_Toc75263809)

[6. Interfaces de l’application : 19](#_Toc75263810)

[7. Résultats expérimentaux : 19](#_Toc75263811)

[Conclusion : 20](#_Toc75263812)

[Bibliographie 21](#_Toc75263813)

# Introduction générale :

La détection d’objets dans des flux vidéo est une étape essentielle pour de nombreux algorithmes de vision par ordinateur. En effet, avec le développement des interfaces homme machine multimodales, on trouve aujourd’hui de plus en plus d'applications dans le domaine de la reconnaissance de la gestuelle de la main.

Dans ce cadre, la communauté des malentendants qui utilisent les gestes de la main pour communiquer trouvent beaucoup de difficultés au quotidiens pour se faire comprendre par les autres, qui ne maitrisent pas toujours la langue des signes. Ces gestes sont en effet utilisés pour épeler le nom des personnes, adresse et d'autres mots. Par conséquent, le développement de systèmes de traduction automatique entre la langue des signes et la langue parlée, pour éliminer les obstacles liés à l'intégration des personnes sourdes dans la société, est apparue comme un besoin urgent pour cette communauté.

Ces dernières années, les réseaux de neurones convolutionnels ont permis d’obtenir de performances élevés dans la reconnaissance visuelle.

L'objectif de notre projet et de développer une application mobile capable de reconnaitre les signes de la main, en temps réel. Nous nous intéresserons dans ce travail aux postures de la main représentant les lettres de l'alphabet. Pour cela nous avons construit une base d’images de trente catégories correspondant aux lettres de la langue des signes arabes. Le système développé utilise un réseau de neurones convolutionnels pour détecter la main dans les images données par une caméra pour ensuite reconnaître les signes exprimés par ces postures.

Ce mémoire est organisé en 3 chapitres :

* Dans le premier nous présenterons les différents composants d’un système de détection d’objets. Ensuite nous aborderons le domaine de la reconnaissance des postures de la main de la LSA.
* Le deuxième décrit les réseaux de neurones convolutifs et les diffèrent architectures utilisé pour la détection d’objets.
* Et dans le dernier chapitre on présente les étapes de réalisation de notre application de reconnaissance des postures de la main pour la LSA.
* Nous clôturons ce mémoire par une conclusion générale et quelques perspectives.

# Chapitre 1 : Détection d’objets dans les images :

## Introduction :

## Composants d’un système de détection d’objets :

La détection d’objet est une méthode base sur l’apprentissage profond **(Deep Learning)** vise à déterminer s’il existe des instance d’un objet d’une catégories donnée (telles que : des humains, animaux, voitures, arbres ………) dans une image. La détection d’objets s’appuie sur deux étapes intermédiaires : la **localisation d’objets** et **la classification d’image**.

Pour avoir une bonne méthode de détection d’objets, il est nécessaire d’avoir un algorithme solide de détection de régions et un bon algorithme de classification d’images, Cette méthode qui utilise le **DL** sert à résoudre des taches complexe et de haut niveau de la vision tel que la segmentation, la poursuite des objets, la détection des évènements, reconnaissance des activités, compréhension des scène, comptages des objets ….

On peut classer la détection d’objet en deux type : la détection d'instances spécifiques et détection d’une catégorie. Le premier type consiste à détecter un objet spécifique (par exemple : visage de Bill Gates, le monument Makam E'chahid, les cartoons tom and Jerry …). Par contre le but de l’autre type est de détecter une instance d’une catégorie d’objet prédéfinies (comme un humain ou une voiture ou chat …) [1].



Figure 1 Instance d’objet spécifique

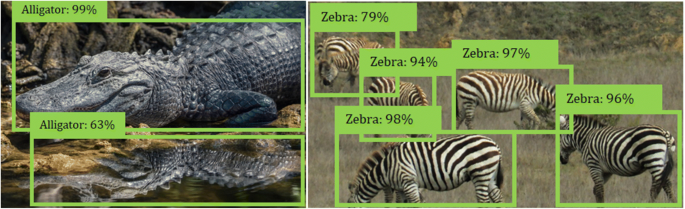
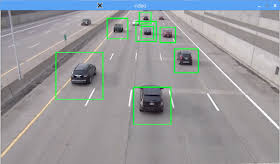
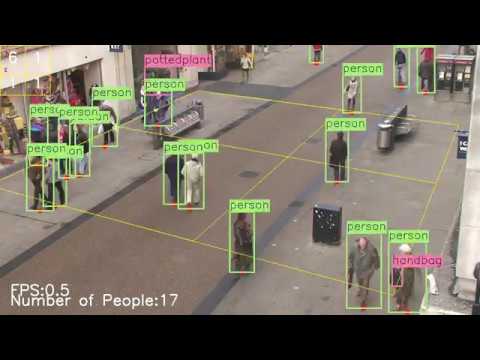


Figure 2 Détectiond’une catégorie

### La localisation D’objets :

La première étape de détection d’objet consiste à trouver la position d’un ou plusieurs objets dans une image et de dessiner leurs contours. Donc on génère des valeurs contenues qui constituent les cordonnées et les dimensions du cadre (**Bounding Box**) qui inclure l’objet unique détecté dans l’image.

### La classification d’image :

C’est la méthode qui consiste à prédire et attribuer à chaque instance une étiquette de classe correspondante. Classer les images en fonction des données qui sont des images d’entrée (images : inputs) avec leurs classes respectivement (objets visée : outputs). Il extrait toutes les **features** (caractéristiques) des images pour les utiliser lors de la prédiction. Pour les chiffres manuscrits, nous pourrions avoir dix classes, correspondant aux chiffres de 0 à 9.

La forme la plus simple de classification est lorsqu'il n'y a que deux classes, un problème que nous appelons classification binaire (binary classification). Par exemple, notre ensemble de données pourrait être constitué d'images d'animaux et de nos étiquettes pourrait être les classes {chat, chien}. Lorsque nous avons plus de deux classes possibles, nous appelons le problème de classification multiclasse (multiclass classification). Les exemples courants incluent la reconnaissance de caractères manuscrits {0, 1, 2, ... 9, a, b, c, ...}. [2]

## Problème liée à la détection d’objets :

## Domaine d’application :

## Reconnaissance des postures de la main de la LSA :

## Conclusion :

# Chapitre 2 : Réseaux de Neurones Convolutifs :

## Introduction :

## Les CNNs :

Les CNNs est un type particulier de réseaux de neurones artificiels basé sur le concept de **Deep Learning**, qui ont conçue pour le traitement d’image, la reconnaissance vocale et pour le traitement des langages naturelles (NLP), Ils sont apparus en 1980 par le chercheur LuCan Yann.et depuis ce temps les CNN ont révolutionner le domaine de reconnaissance des images jusqu’au ces dernières années ils ont pu atteindre des performances élevées sur certains taches visuel complexe de l’humain tel que la classification et la segmentation des images, la reconnaissance faciale, la reconnaissance des objets et etc. Donc ils ont remporté un grand succès dans le monde de vision par ordinateur. [3]

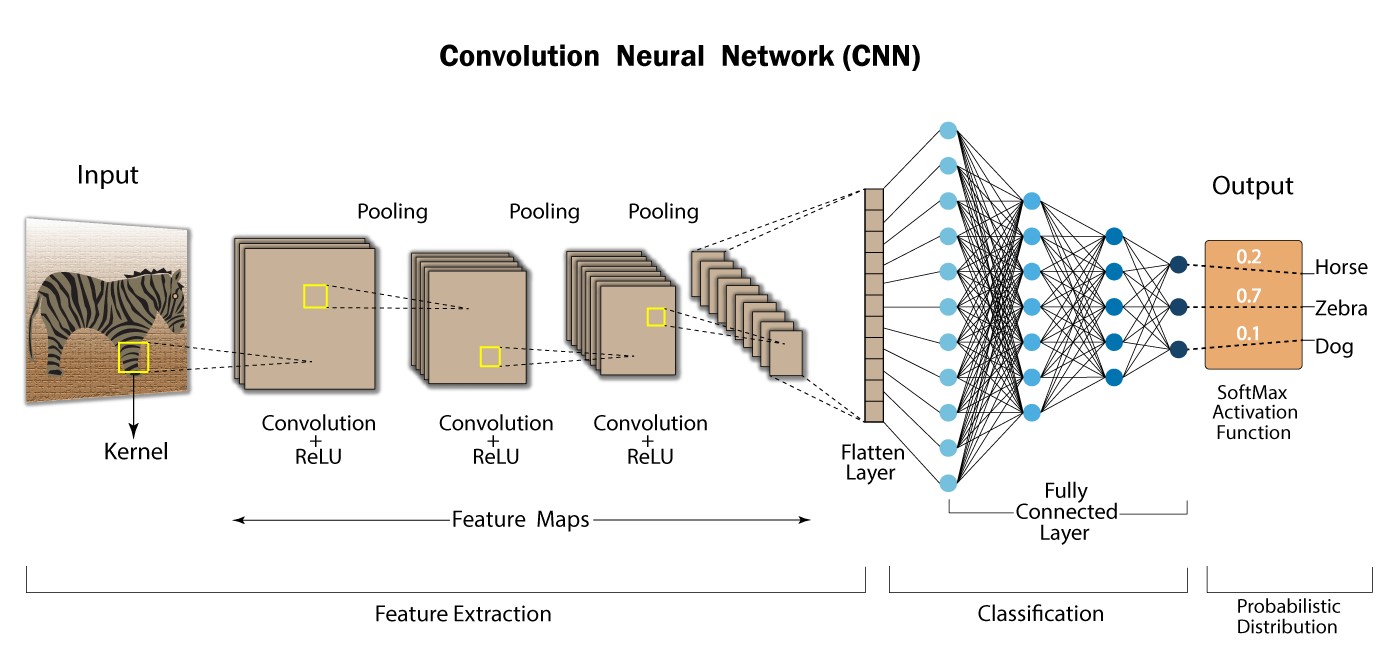


Figure 3 ARCHITECTURE CNN.

L’architecture CNN est une séquence de couche lier qui forme des réseaux hiérarchiques à plusieurs niveaux, elle est composée de trois types de couche (La couche de convolution ; La couche d’activation Relu ; La couche pooling ; Couche entièrement connecter (Fully connected))

Les deux premier qui ont rôle de l’extraction des caractéristiques dans une image, par contre la couche fully connected qui est responsable de la classification de ses **features.**

### Couche convolution :

Les couches convolution c’est les couches les plus important qui caractérise les CNN, elle consiste d’une suite des opérations linéaire et non linéaire.

#### Convolution:

Les couches convolutionnelles fait la particularité des réseaux CNN puisqu’elle fonctionne comme un extracteur de caractéristique, C’est est un type des fonctions linéaires, où on utilise un ensemble de filtres (**Kernel**) pour chaque couche convolution. Les filtres sont des vecteurs générés aléatoirement dans le réseau, constitués de poids et de biais. Les mêmes poids et biais sont partagés entre divers neurones dans CNN au lieu de poids et biais uniques pour chaque neurone. De nombreux filtres peuvent être générés où chaque filtre capture une caractéristique unique à partir de l'entrée [5]. Nous glissons (plus précisément, convolons) chaque filtre sur la largeur et la hauteur du volume d'entrée et calculons les produits scalaires entre les entrées du filtre et l'entrée à n'importe quelle position (**tensor**) puis on calcule la somme des valeurs obtenu pour avoir une valeur correspondant à la positionne actuelle de l’image, On applique cette procédure sur tous les entrées de la couche convolution pour former à la sortie ce qu’on appelle carte des caractéristiques (**feature map)** *(Voir figure 3)*. Les ConvNet aide à augmenter l'efficacité du modèle en réduisant le nombre de paramètres à apprendre par rapport aux réseaux de neurones. D'autre part, les hyper paramètres qui doivent être définis avant la taille des filtres, le nombre de filtres, le padding le pas. [6]

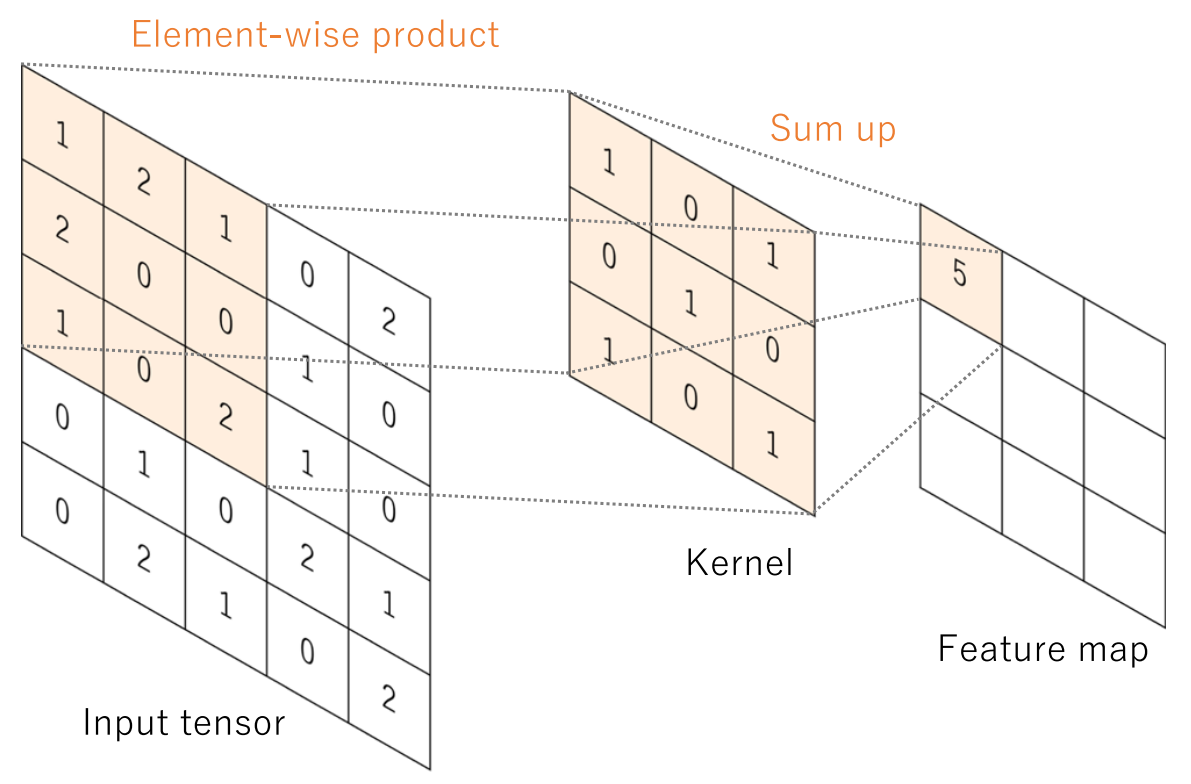


Figure 4 exemple d’une opération de convolution avec un filtre de 3 x 3, pas =1, padding =0.[6]

#### La fonction d’activation:

Tout sorties (feature map) d'une opération linéaire telle que la convolution sont ensuite passées par une fonction d'activation non linéaire ont été utilisées précédemment car ce sont des représentations mathématiques du comportement d'un neurone biologique, la fonction d'activation non linéaire la plus couramment utilisée actuellement est l'unité linéaire rectifié (ReLU), [3] elle consiste à remplacer toutes les valeurs négatives par zéro. [Fei-Fei Li, 2020].

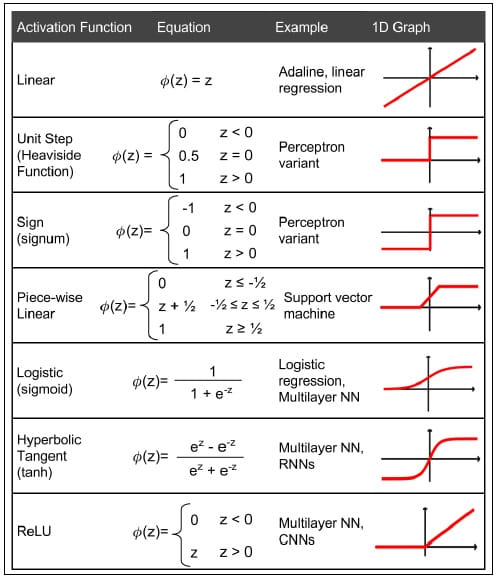


Figure 5 les Fonction d’activation les plus connus. [7]

### Couche pooling:

On trouve souvent Dans les CNN, les couche de convolution suivis des couches de mis en commun (pooling). Qui sert à réduire progressivement la dimension spatiale de la représentation, donc réduire la quantité de paramètres et les calculs dans le ConvNet. En utilisant le « max Pooling » et le « Average Pooling ». [6]

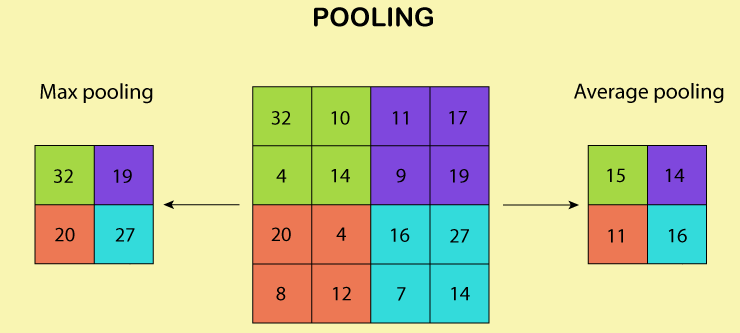


Figure 6 les résultats des opér*ations de Pooling (Avg/Max)*.[4]

#### Max pooling:

La forme la plus populaire d'opération de sous-échantillonnage (pooling) est la Max pooling, qui extrait les sous-échantillonnage des cartes d'entités d'entrée (feature map), en faisant sortie la valeur maximale de chaque carte caractéristique et supprime toutes les autres valeurs (figure 5). Un pooling max avec un filtre de taille 2×2 avec un pas de 2 est couramment utilisée dans la pratique. Cela sous-échantillonne la dimension dans le plan des cartes d'entités par un facteur de 2. Contrairement à la hauteur et la largeur, la dimension de profondeur des cartes d'entités reste inchangée. [6]

#### Average pooling:

Une autre opération de mise en commun à noter est le Average pooling. Qui effectue un type extrême de sous-échantillonnage, où une carte d'entités (feature map) avec la taille de H × L est sous-échantillonnée dans un tableau 1 × 1 en prenant simplement la moyenne de tous les éléments de chaque carte de caractéristiques, alors que la profondeur des cartes de caractéristiques est retenue. Cette opération n'est généralement appliquée qu'une seule fois avant les couches entièrement connectées. Les avantages de l'application de la mise en commun moyenne globale sont les suivants:

- réduit le nombre de paramètres prenables.

- active le CNN pour accepter des entrées de taille variable. [6]

### Couche fully connected :

## Les architecture utilisées pour la détection d’objets :

Dans cette partie, On va parler des modèles connus de détection d'objets: R-CNN, Fast R-CNN, Faster R-CNN, YoloV5. Ces modèles sont très liés et les nouvelles versions montrent une grande amélioration de la vitesse par rapport aux anciennes.

### R-CNN :

Les R-CNN (Region-Based Convolutional Neural Network) est une méthode proposée par Ross Girshick dont l’idée se compose de deux étapes [8]:

1. Utiliser un algorithme de la recherche sélective () qui sert à extraire environ 2k régions sur l’image avec les bounding box d’objet (“region of interest” ou “RoI”).
2. Et puis il extrait les caractéristiques avec CNN de chaque région indépendamment pour la classification.

Le fonctionnement de R-CNN peut être résumé comme suit [8]:

1. **Pré-entrainer** un réseau CNN aux tâches de classification d'images qui implique N classes.
2. Proposez des **region of interest** indépendantes de la catégorie par la recherche sélective (~ 2k régions par image). Ces régions peuvent contenir des objets cibles et elles sont de tailles différentes.
3. Les candidats de région sont **déformés** pour avoir une taille fixe compatible avec l’entrée de CNN.
4. Application de **fine-tuning** de ConvNet sur les régions de proposition déformées pour les classes N + 1; La classe supplémentaire fait référence à l'arrière-plan (aucun objet d'intérêt). Dans la phase de **fine-tuning**, nous devrions utiliser un taux d'apprentissage beaucoup plus faible et le mini-batch sur échantillon des cas positifs parce que la plupart des régions proposées ne sont que d'arrière-plan.
5. Étant donné chaque région d'image, une propagation vers l'avant à travers le CNN génère un vecteur de caractéristiques. Ce vecteur de caractéristiques est ensuite consommé par un **SVM binaire** formé pour **chaque classe** indépendamment.  
   Les échantillons positifs sont des régions proposées avec un seuil de chevauchement IoU (intersection sur union)> = 0,3, et les échantillons négatifs ne sont pas pertinents pour les autres.
6. Pour réduire les erreurs de localisation, un modèle de régression est entraîné pour corriger la fenêtre de détection prévue sur le décalage de correction de la boîte englobante à l'aide des fonctionnalités CNN.

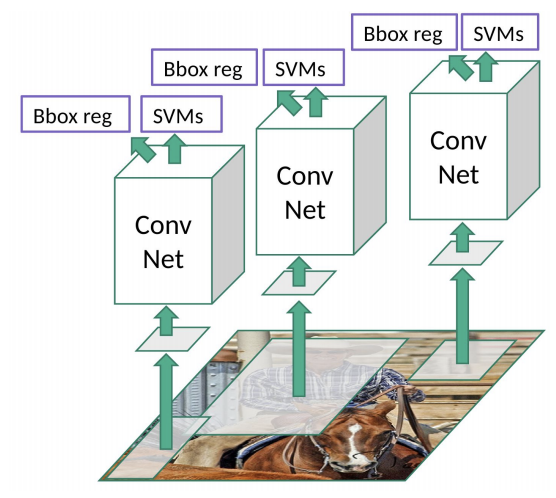


Figure 7 FONCTIONNEMENT de R-CNN. [9]

#### Les inconvénients Des R-CNN :

Jusqu'à présent, nous avons vu comment RCNN peut être utile pour la détection d'objets. Mais cette technique a ses propres limites. La formation d'un modèle RCNN est coûteuse et lente grâce aux étapes ci-dessous:

* Extraction de 2000 régions pour chaque image en fonction de la recherche sélective
* Extraction de caractéristiques à l'aide de CNN pour chaque région d'image. Supposons que nous ayons N images, alors le nombre de caractéristiques CNN sera N \* 2000
* L'ensemble du processus de détection d'objets à l'aide de R-CNN comporte trois modèles:
  1. CNN pour l'extraction de caractéristiques.
  2. Classificateur SVM linéaire pour identifier les classes des objets.
  3. Modèle de régression pour resserrer les boîtes englobantes (bBox).

Tous ces processus se combinent pour rendre RCNN très lent. Il faut environ 40 à 50 secondes pour faire des prédictions pour chaque nouvelle image, Donc nous avons une autre technique de détection d'objets qui corrige la plupart des limitations que nous avons vues dans R-CNN. [9]

### Fast R-CNN :

Le même créateur du modèle précédent (R-CNN) a résolu certains des inconvénients de R-CNN en construisant un algorithme de détection d'objet plus rapide appelé FR-CNN. En unifiant trois modèles indépendants dans un modèle unique qui extrait les entités des régions, les divise en différentes classes et renvoie simultanément les boîtes de délimitation des classes identifiées.

Au lieu d'extraire les vecteurs de caractéristiques CNN indépendamment pour chaque 2000 région, ce modèle les regrouper en un seul passage CNN sur toute l'image et les region proposals partagent cette matrice des features. Ensuite, la même matrice de features est utilisée pour la classification d'objets et le régresseur de boîte englobante. En conclusion, le partage des calculs accélère R-CNN. [8]

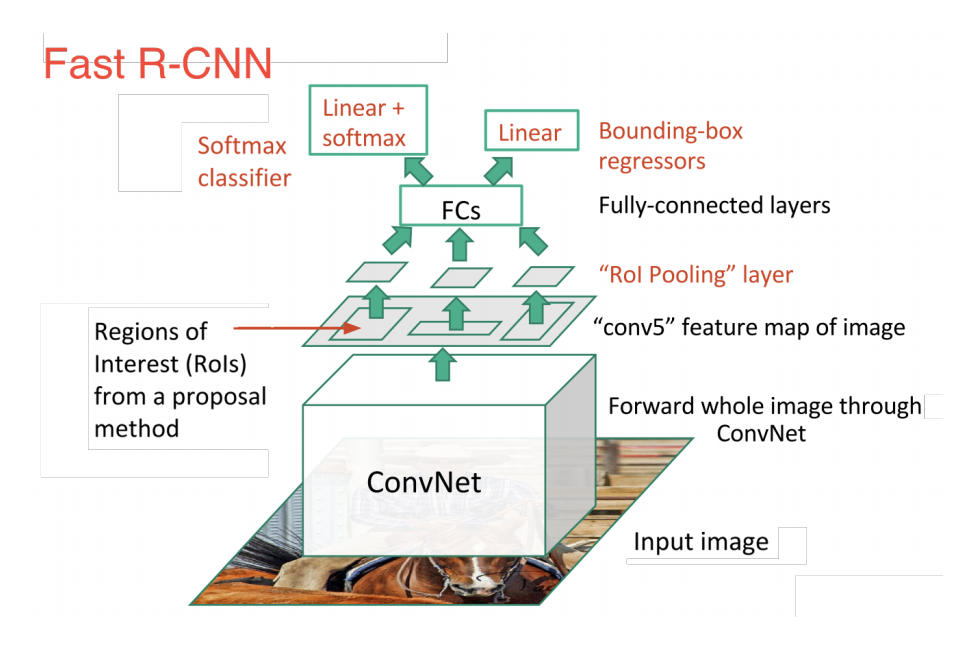


Figure 8 FONCTIONNEMENT de Fast R-CNN. [9]

Décomposons cela en étapes pour simplifier le concept:

1. Nous prenons une image comme entrée.
2. Cette image est transmise à un ConvNet qui à son tour génère les **region of interest**.
3. Une couche de **pooling RoI** est appliquée sur toutes ces régions pour les remodeler selon l'entrée du ConvNet. Ensuite, chaque région est transmise à un réseau entièrement connecté.
4. Une couche softmax est utilisée pour la classification des objets, Et une couche de régression linéaire est également utilisée parallèlement à la sortie pour les coordonnées de la boîte englobante pour les classes prédites.

#### RoI pooling :

Il s'agit d'un type diffèrent de max pooling pour convertir les entités de la région projetée de l'image de n'importe quelle taille, h x l, en une petite fenêtre fixe, H x W. La région d'entrée est divisée en grilles H x W, environ chaque sous-fenêtre de taille h / H x l / l. Ensuite, appliquez le max pooling dans chaque grille. [8]

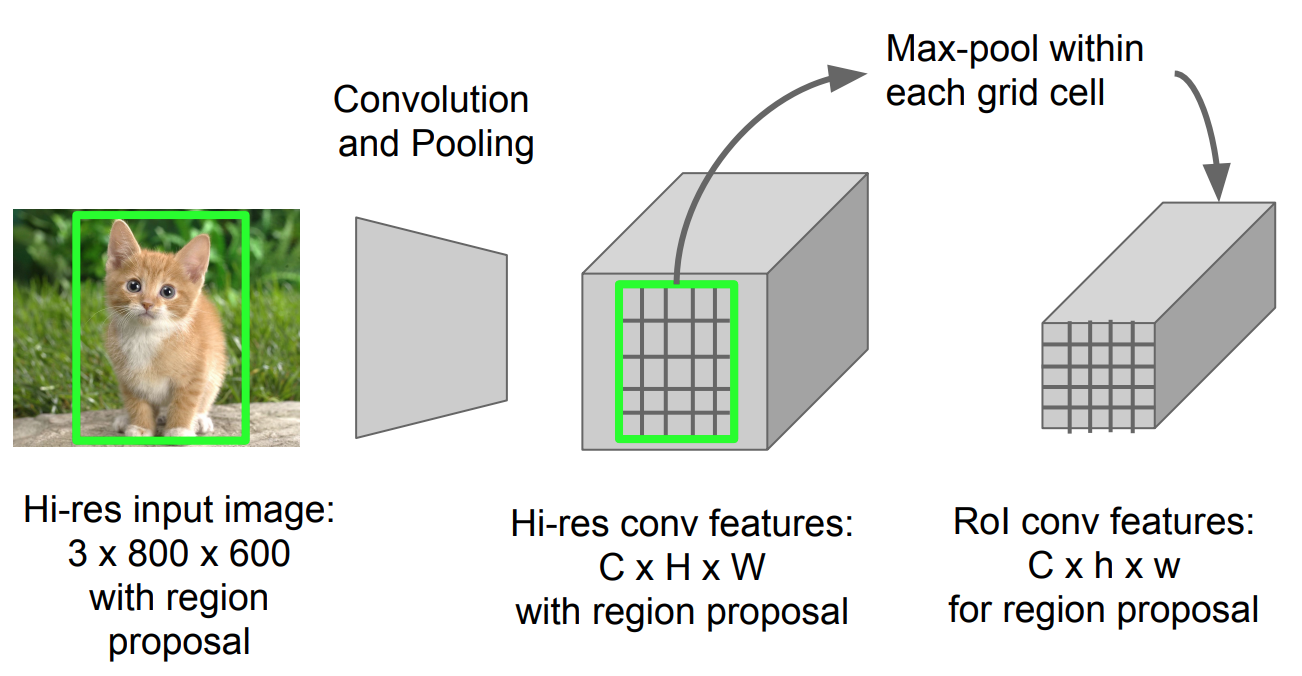


Figure 9 RoI pooling. [11]

#### Les inconvénients Des Fast R-CNN :

Le Fast R-CNN a certains problèmes. Il utilise les algorithmes da recherche sélective pour trouver les régions d'intérêt, ce qui est un processus lent dont Il faut environ 2 secondes par image pour détecter les objets. Mais lorsque nous utilisons des grands ensembles de données, donc le Fast R-CNN ne devient plus si rapide. [10]

### Faster R-CNN :

Faster R-CNN est la version mieux optimisée. La principale différence entre eux est que Fast R-CNN utilise la recherche sélective pour générer des régions de propositions, tandis que Faster RCNN utilise « Region Proposal Network ». RPN prend en entrée les feaetures maps d'image de CNN et génère un ensemble de region of interest. [10]

Voici des étapes d’une approche Faster R-CNN:

1. Nous prenons une image comme entrée et la transmettons au CNN qui renvoie la carte des caractéristiques de cette image.
2. Le réseau de RPN est appliqué sur ces cartes de caractéristiques. Cela renvoie les régions de propositions.
3. Une couche de pooling RoI est appliquée sur ces propositions pour les réforme une taille fixée.
4. Enfin, les propositions sont transmises à une couche entièrement connecté qui a une couche softmax et une couche de régression linéaire en son sommet, pour classer et sortir les cadres englobants des objets. [10]

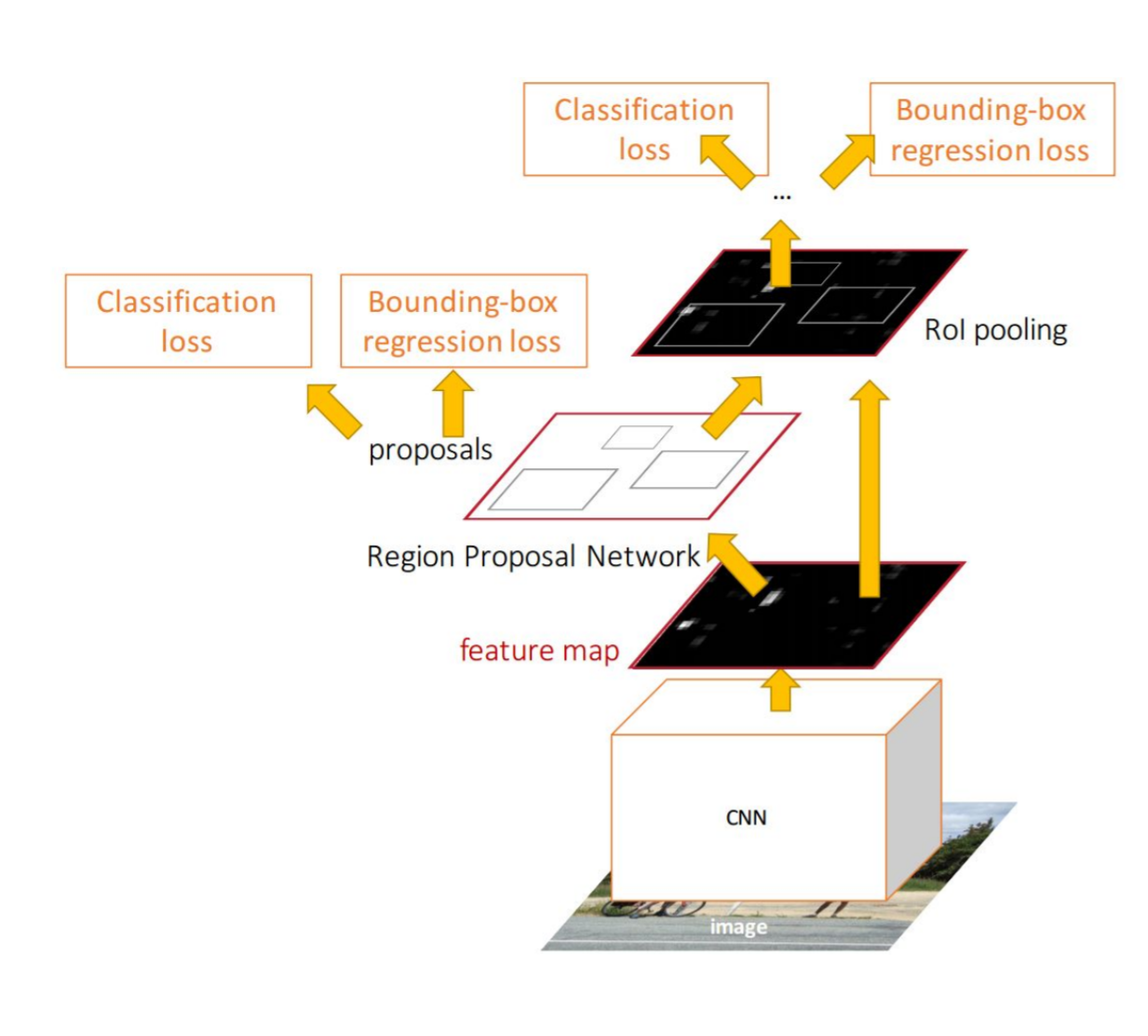


Figure 10 FONCTIONNEMENT de Faster R-CNN. [11]

### Yolo (you look only once):

Il existe quelques algorithmes différents pour la détection d'objets et ils peuvent être divisés en deux groupes:

* Algorithmes basés sur la classification. Ils sont mis en œuvre en deux étapes. Tout d'abord, ils sélectionnent les régions d'intérêt dans une image. Deuxièmement, ils classent ces régions à l'aide de CNN. Cette solution peut être lente car nous devons exécuter des prédictions pour chaque région sélectionnée.
* Algorithmes basés sur la régression - au lieu de sélectionner des parties intéressantes d'une image, ils prédisent les classes et les Bounding box pour l'image entière en une seule exécution de l'algorithme. Les exemples les plus connus de ce groupe sont les algorithmes de la famille YOLO (You Only Look Once). Et SSD (Détecteur Multibox Single Shot).

A l’apparitions de YOLO en 2016, Il est considéré comme le modèle le plus rapide utilisés dans la détection d’objets en temps réel. Étant donné que YOLO évite l'étape de proposition de région et ne prédit que sur un nombre limité de cadre englobante, il est capable de faire des prédictions très rapidement.

En résume les étapes qui compose le modèle YOLO :

* **Pré-entrainer** un réseau CNN sur la tâche de classification d'images.
* Diviser une image en S× S cellules. Si le centre d'un objet trouver dans une cellule, cette cellule est « responsable » de la détection de l'existence de cet objet. Chaque cellule prédit l'emplacement de B boîtes englobants, Un score de confidence, et une probabilité de classe d'objet conditionnée à l'existence d'un objet dans la boîte englobant.
  + Les **coordonnées de** Bounding Box sont définies par la quadruplé (centre X, centre y, largeur, hauteur).
  + Un **score de confidence** indique la probabilité que la cellule contienne un objet : Pr (objet exister) x IoU (pred, vrai); où Pr= probabilité et IoU= **intersection over union**.
  + Si la cellule contient un objet, elle prédit une **probabilité** que cet objet appartienne à chaque classe Ci, i = 1, …, K : Pr (l’objet appartient à la classe Ci | contient un objet). A ce stade, le modèle ne prédit qu'un seul ensemble de probabilités de classe par cellule, quel que soit le nombre des Bounding Box B.
  + Au total, une image contient S × S × B boîtes englobant, chaque cadre correspondant à **4** prédictions de localisation, **1** score de confidence (au total **5** valeur donc 5xB) et K probabilités conditionnelles pour la classification des objets. Les valeurs de prédiction totales pour une image sont S × S × (5B + K), qui est la forme du tenseur de la couche de convolution finale du modèle.
* La couche finale du CNN pré-entraîné est modifiée pour produire un tenseur de prédiction de taille S × S × (5B + K).

Pour faire un test de la base de données Pascal Voc sur YOLO, On utilise ses paramètres S=7, B=2. Pascal Voc content 20 classe donc K=20. Alors on a 7 x 7 x (5 x 2 + 20). Donc on produire 98 tenseur par le ConvNet au lieu de 2000 de la famille R-CNN.

## Conclusion :

# Chapitre 3 : Un système de vison pour la reconnaissance des signes de la LSA :

## Introduction :

## Présentation de l’architecture du système de détection des signes de LSA :

## Datasets :

### Construction des données :

### Prétraitements des données :

### Annotation des images :

## Mesure des performances :

## Outils de développement :

## Interfaces de l’application :

## Résultats expérimentaux :

# Conclusion :

# Bibliographie

|  |  |
| --- | --- |
| [1] | «Deep Learning for Generic Object Detection: A Survey,» International Journal of Computer Vision (2020) 128:261–318, 31 October 2019. |
| [2] | Z. C. L. M. L. a. A. J. S. Aston Zhang, Dive into Deep Learning, Jan 19, 2021. |
| [3] | Hands-On Machine Learning with Scikit-Learn and TensorFlow, O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.. |
| [4] | «CS231n Convolution Neural Networks for Visual Recognition,» [En ligne]. Available: https://cs231n.github.io/convolutional-networks/. |
| [5] | Togashi, Rikiya Yamashita & Mizuho Nishio & Richard Kinh Gian Do & Kaori, «Convolutional neural networks: an overview and application in radiology,» [En ligne]. Available: https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9. |
| [6] | «simplilearn,» [En ligne]. Available: https://www.simplilearn.com/ice9/free\_resources\_article\_thumb/list-of-activation-functions-used-with-perceptron.jpg. |
| [7] | «Convolution neural network deep learning,» [En ligne]. Available: https://developersbreach.com/convolution-neural-network-deep-learning/. |
| [8] | «Lil'Log,» [En ligne]. Available: https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html. |
| [9] | «towards Data Science,» [En ligne]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. |
| [10] | «Analytics Vidhya,» [En ligne]. Available: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/?utm\_source=blog&utm\_medium=computer-vision-implementing-mask-r-cnn-image-segmentation. |
| [11] | F.-F. L. &. A. K. &. J. Johnson, «cs231n,» [En ligne]. Available: http://cs231n.stanford.edu/slides/2016/winter1516\_lecture8.pdf. |