

Détecteur des Plaques d'Immatriculation (DPI) : Application dans un parking

Ilyas MOUTAWWAKIL : 8471

Épreuve de TIPE

Session 2019

Plan de l'exposé

- 1 Introduction au système : DPI
- 2 La théorie des réseaux de neurones artificiels
- 3 La pratique des réseaux de neurones artificiels
- 4 Réalisation et implémentation
- 5 Conclusion

Introduction au système : DPI

Problème

- Le contrôle d'accès aux espaces privés.



Fig. 1.1 – Exemple d'application du système

Introduction au système : DPI

Solution

- Un processus simple et dynamique assurant une detection rapide et précise des plaques !



Fig. 1.2 – Le processus de reconnaissance optimal

Introduction au système : DPI

Analyse Fonctionnelle

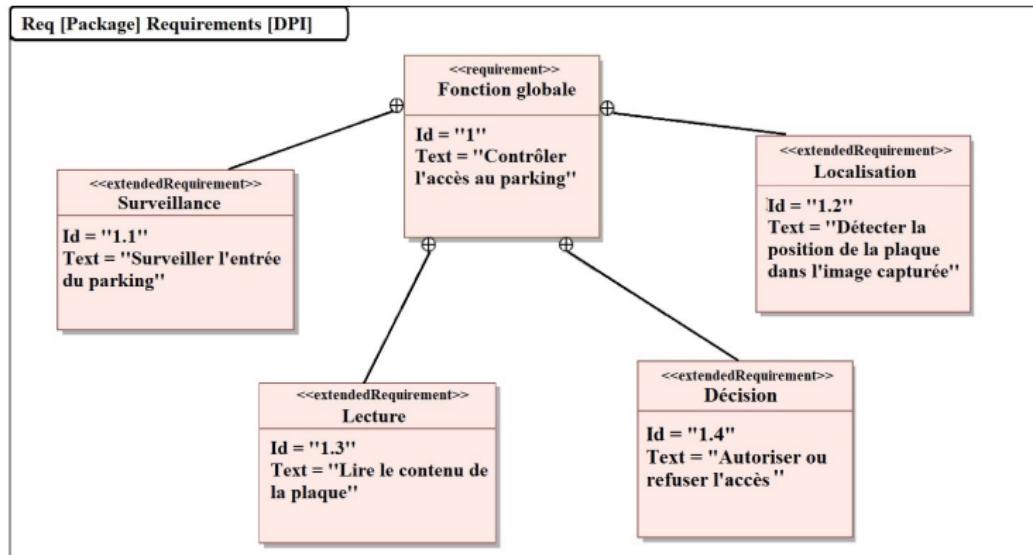


Fig. 1.3 – Diagramme d'exigences

Introduction au système : DPI

Une approche

- Développer des algorithmes basés sur les nouvelles techniques d'apprentissage profond pour la localisation et la reconnaissance.

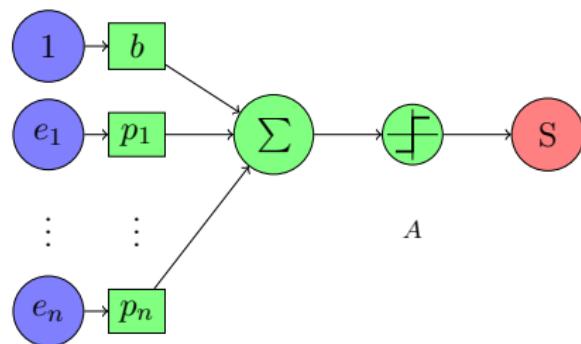


Fig. 1.4 – Les trois bibliothèques essentielles

Réseau de neurones artificiels

Perceptron

- Le perceptron : Une fonction à paramètres ajustables.



$$S = A\left(\sum_{i=1}^n (e_i \cdot p_i) + b\right)$$

- La fonction d'activation A.
 - Heaviside :

$$A(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$

- Sigmoid :

$$A(x) = \frac{1}{1 + \exp(-x)}$$

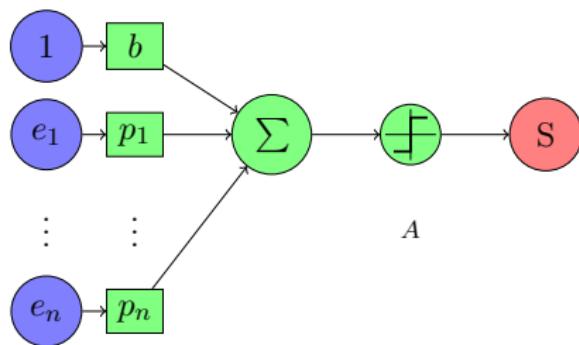
- RELU :

$$A(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

Réseau de neurones artificiels

Perceptron

- Le perceptron : Une fonction à paramètres ajustables.



$$S = A \left(\sum_{i=1}^n (e_i \cdot p_i) + b \right)$$

- La fonction d'activation A.
 - Heaviside :

$$A(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$

- Sigmoid :

$$A(x) = \frac{1}{1 + \exp(-x)}$$

- RELU :

$$A(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

Réseau de neurones artificiels

Réseau profond

- Le théorème d'approximation universelle.
- Pourquoi un réseau multi-couches (Profond) ?

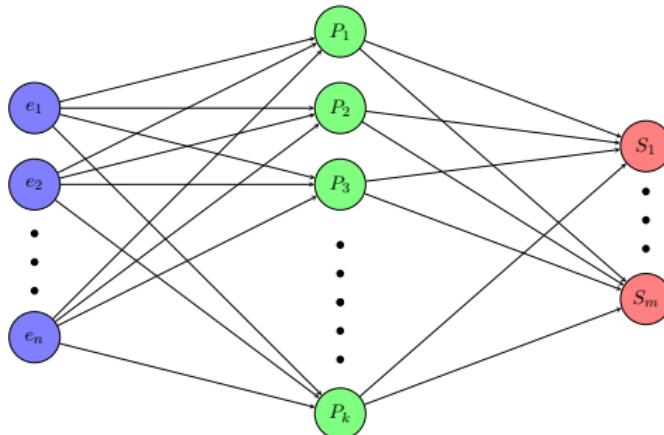


Fig. 2.1 – Réseau mono-couche

Réseau de neurones artificiels

Réseau profond

- Le théorème d'approximation universelle.
- Pourquoi un réseau multi-couches (Profond) ?

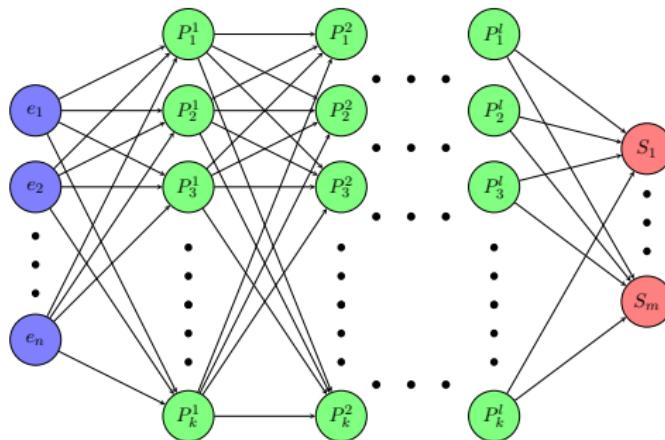


Fig. 2.2 – Réseau multi-couche

Algorithme d'apprentissage

Modélisation du problème

- Base de données annotées : $(x_1, y_1), \dots, (x_n, y_n)$.
- Fonction paramétrée : $f_P : \mathcal{X} \rightarrow \mathcal{Y}$.
- Fonctions de coût : $\mathcal{L}_i : \mathcal{P} \rightarrow \mathbb{R}$; $\mathcal{L}(P) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(P)$.

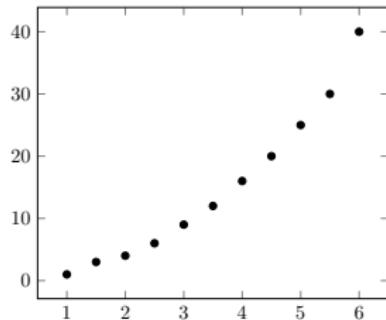


Fig. 2.3 – Problème de régression

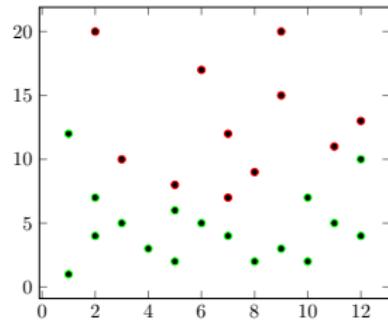


Fig. 2.5 – Problème de classification

Algorithme d'apprentissage

Modélisation du problème

- Base de données annotées : $(x_1, y_1), \dots, (x_n, y_n)$.
- Fonction paramétrée : $f_P : \mathcal{X} \rightarrow \mathcal{Y}$.
- Fonctions de coût : $\mathcal{L}_i : \mathcal{P} \rightarrow \mathbb{R}$; $\mathcal{L}(P) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(P)$.

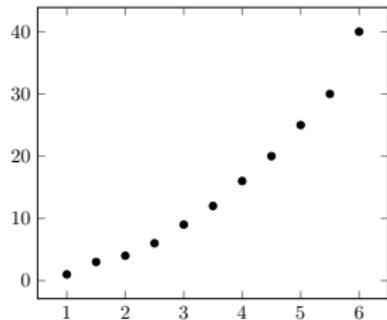


Fig. 2.3 – Problème de régression

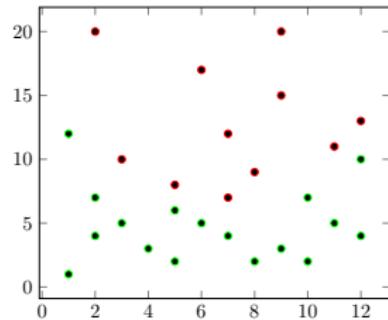


Fig. 2.5 – Problème de classification

Algorithme d'apprentissage

Modélisation du problème

- Base de données annotées : $(x_1, y_1), \dots, (x_n, y_n)$.
- Fonction paramétrée : $f_P : \mathcal{X} \rightarrow \mathcal{Y}$.
- Fonctions de coût : $\mathcal{L}_i : \mathcal{P} \rightarrow \mathbb{R}$; $\mathcal{L}(P) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(P)$.

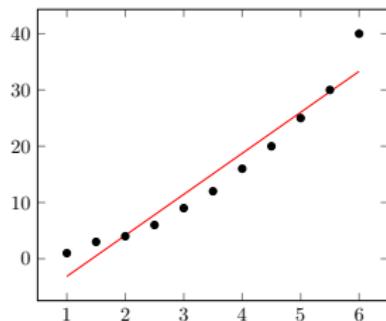


Fig. 2.4 – Fonction de régression

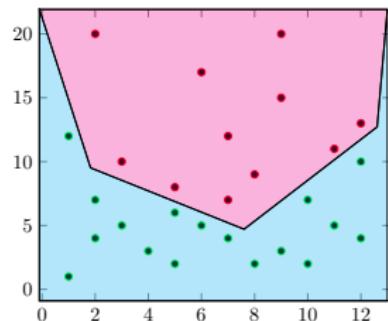


Fig. 2.6 – Frontière de décision

Algorithme d'apprentissage

La minimisation d'une fonction

- \mathcal{L} est une fonction définie sur l'espace de poids.
- Objectif : Trouver la valeur de P^* qui minimise \mathcal{L} .
- Méthode : Descente de gradient

$$P^{(i+1)} = P^{(i)} - \eta_i \nabla \mathcal{L}(P^{(i)})$$

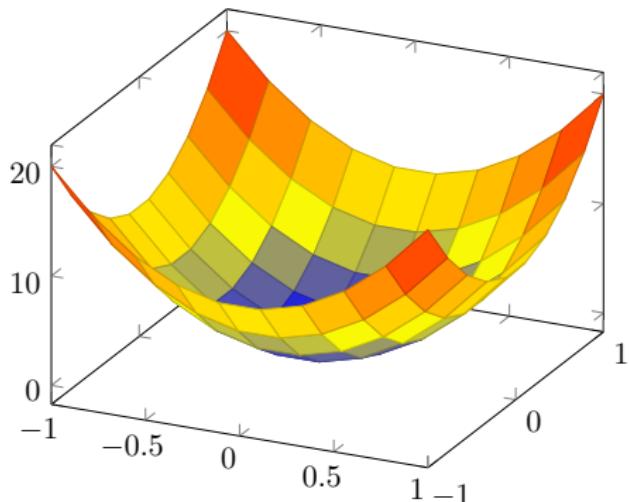


Fig. 2.7 – La forme de \mathcal{L} : cas de deux paramètres

Algorithme d'apprentissage

La minimisation d'une fonction

- \mathcal{L} est une fonction définie sur l'espace de poids.
- Objectif : Trouver la valeur de P^* qui minimise \mathcal{L} .
- Méthode : Descente de gradient

$$P^{(i+1)} = P^{(i)} - \eta_i \nabla \mathcal{L}(P^{(i)})$$

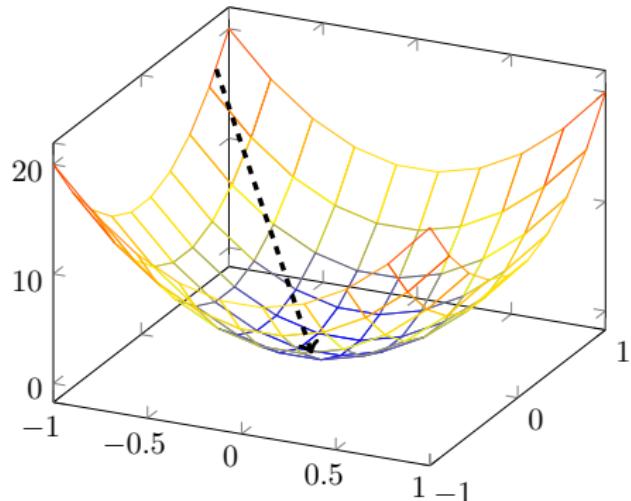


Fig. 2.8 – Le minimum de \mathcal{L} : cas de deux paramètres

Algorithme d'apprentissage

La minimisation d'une fonction

- \mathcal{L} est une fonction définie sur l'espace de poids.
- Objectif : Trouver la valeur de P^* qui minimise \mathcal{L} .
- Méthode : Descente de gradient

$$P^{(i+1)} = P^{(i)} - \eta_i \nabla \mathcal{L}(P^{(i)})$$

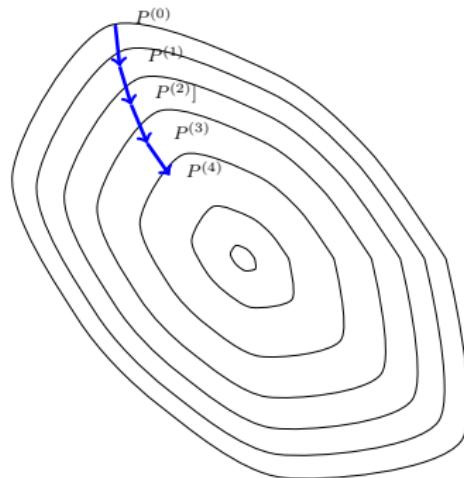


Fig. 2.9 – Le chemin de descente

Création de la base de données

Données réelles

- Chercher des images sur Internet.
 - Google et quelques sites web.
- Prendre des photos de la rue.
 - Parkings publics.
- Écrire un programme d'annotation.
- Annoter les images collectées.



Fig. 3.1 – Exemples d'images collectées

Création de la base de données

Données réelles

- Chercher des images sur Internet.
 - Google et quelques sites web.
- Prendre des photos de la rue.
 - Parkings publics.
- Écrire un programme d'annotation.
- Annoter les images collectées.



Fig. 3.2 – Exemples d'images annotées

Création de la base de données

Données synthétisées

- Analyser les caractéristiques des plaques marocaines.
- Trouver une base de données d'images sans plaques.
- Écrire un programme qui génère des données annotées.
- Analyser et optimiser la qualité des données générées.



Fig. 3.3 – Exemples de plaques marocaines

Création de la base de données

Données synthétisées

- Analyser les caractéristiques des plaques marocaines.
- Trouver une base de données d'images sans plaques.
- Écrire un programme qui génère des données annotées.
- Analyser et optimiser la qualité des données générées.



Fig. 3.4 – Exemples d'images sans plaques

Création de la base de données

Données synthétisées

- Analyser les caractéristiques des plaques marocaines.
- Trouver une base de données d'images sans plaques.
- Écrire un programme qui génère des données annotées.
- Analyser et optimiser la qualité des données générées.

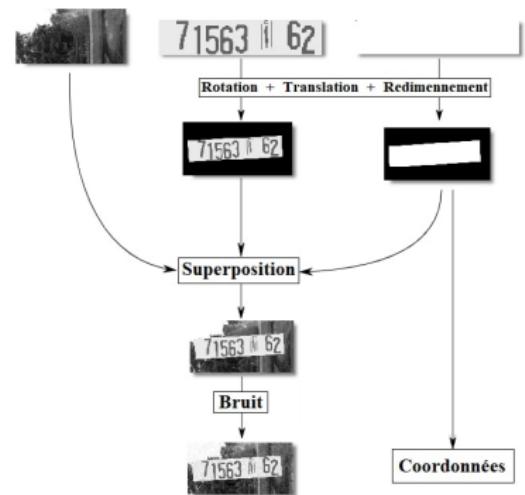


Fig. 3.5 – Le processus des génération de données annotées

Création de la base de données

Données synthétisées

- Analyser les caractéristiques des plaques marocaines.
- Trouver une base de données d'images sans plaques.
- Écrire un programme qui génère des données annotées.
- Analyser et optimiser la qualité des données générées.

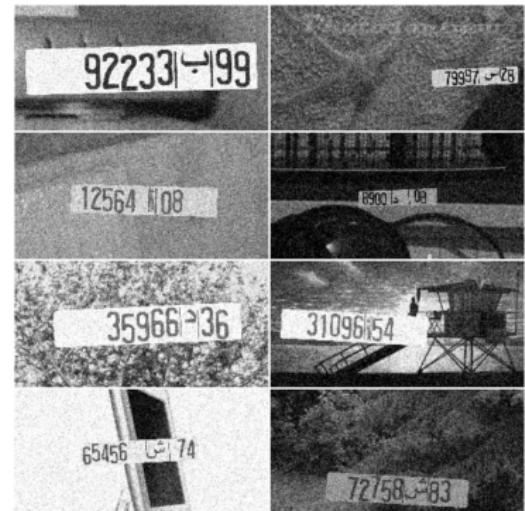


Fig. 3.6 – Exemples des données générées

Création de la base de données

Comparaison entre bases de données

Type de données	Moyenne	Ecart type	Coefficient de variation
Générées	126	32.71	3.86
Réelles	97.84	24.79	3.94

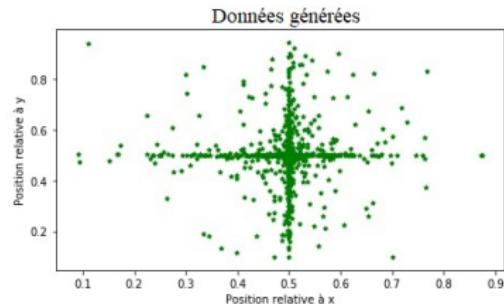
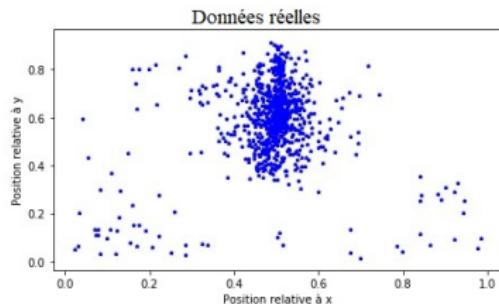


Fig. 3.7 – Dispersion des centres des plaques

Création du réseau de neurones

Un problème de régression ?

- Un model simple de prédiction.
- Prédictions érronées.
- Problème de "Surapprentissage" .

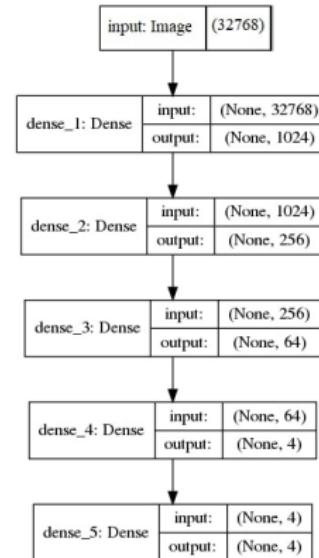


Fig. 3.8 – Model de régression

Création du réseau de neurones

Un problème de régression ?

- Un model simple de prédiction.
- Prédictions érronées.
- Problème de "Surapprentissage" .

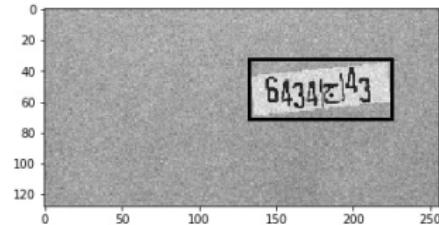


Fig. 3.9 – Resultats des prédictions du réseau

Création du réseau de neurones

Un problème de classification ?

- Problème : Dimensionnalité élevée du problème .
- Idée : Extraction de caractéristiques.
- Solution : Réseaux de neurones convolutionnels.
- Inconvenient : Traitement de chaque région dans l'image.

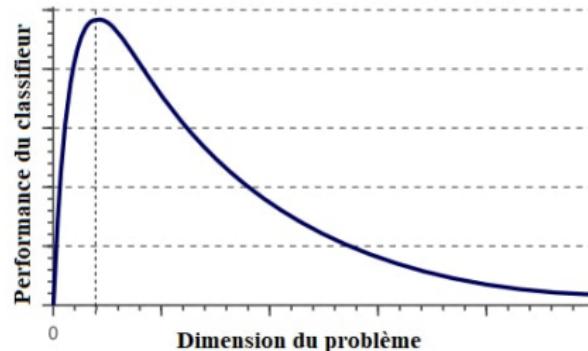


Fig. 3.10 – Malédiction de la dimensionnalité

Création du réseau de neurones

Un problème de classification ?

- Problème : Dimensionnalité élevée du problème .
- Idée : Extraction de caractéristiques.
- Solution : Réseaux de neurones convolutionnels.
- Inconvenient : Traitement de chaque région dans l'image.

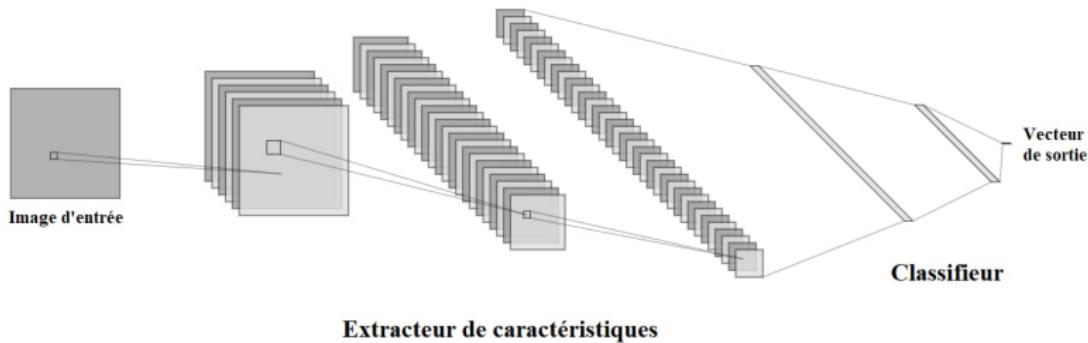


Fig. 3.11 – Principe de l'extraction des caractéristiques

Création du réseau de neurones

Un problème de classification ?

- Problème : Dimensionnalité élevée du problème .
- Idée : Extraction de caractéristiques.
- Solution : Réseaux de neurones convolutionnels.
- Inconvénient : Traitement de chaque région dans l'image.

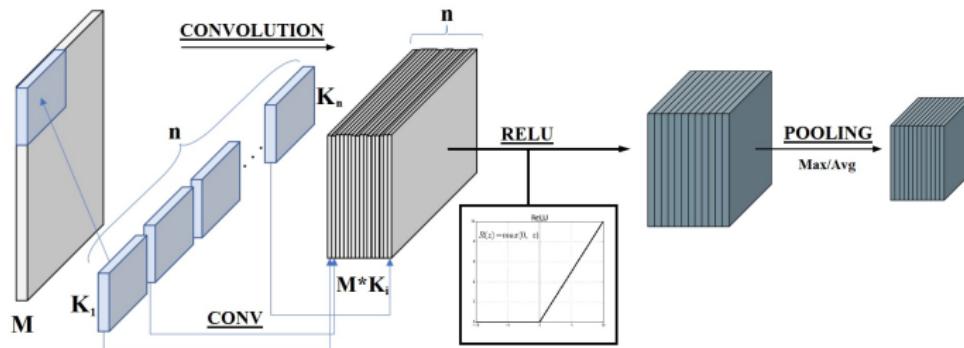


Fig. 3.12 – Principe du bloc de convolution

Création du réseau de neurones

Un problème de classification ?

- Problème : Dimensionnalité élevée du problème .
- Idée : Extraction de caractéristiques.
- Solution : Réseaux de neurones convolutionnels.
- Inconvenient : Traitement de chaque région dans l'image.

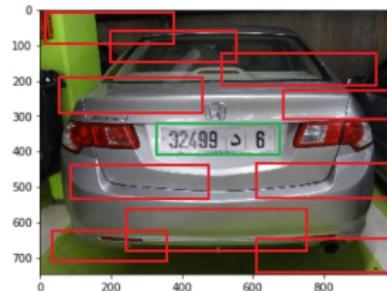


Fig. 3.13 – Détection par fenêtres glissantes

Création du réseau de neurones

Solution Optimale

- "YOLO" : Une transformation de l'espace du problème
 - Diviser l'image en grilles
 - Localiser celle contenant le centre de l'objet.
 - Déterminer le vecteur de sortie.



Fig. 3.14 – Exemple d'image à traiter

Création du réseau de neurones

Solution Optimale

- "YOLO" : Une transformation de l'espace du problème
 - Diviser l'image en grilles
 - Localiser celle contenant le centre de l'objet.
 - Déterminer le vecteur de sortie.



Fig. 3.15 – Première étape

Création du réseau de neurones

Solution Optimale

- "YOLO" : Une transformation de l'espace du problème
 - Diviser l'image en grilles
 - Localiser celle contenant le centre de l'objet.
 - Déterminer le vecteur de sortie.



Fig. 3.16 – Deuxième étape

Création du réseau de neurones

Solution Optimale

- "YOLO" : Une transformation de l'espace du problème
 - Diviser l'image en grilles
 - Localiser celle contenant le centre de l'objet.
 - Déterminer le vecteur de sortie.

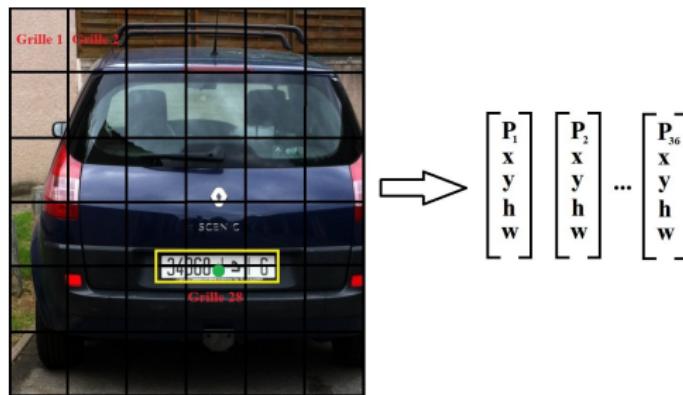


Fig. 3.17 – Troisième étape

Création du réseau de neurones

Variété de solutions

- Differents extracteurs de caractéristiques :
 - DarkNet : Structure originale.
 - MobileNet : Développée par Google.
 - SqueezeNet : Développée par Stanford.

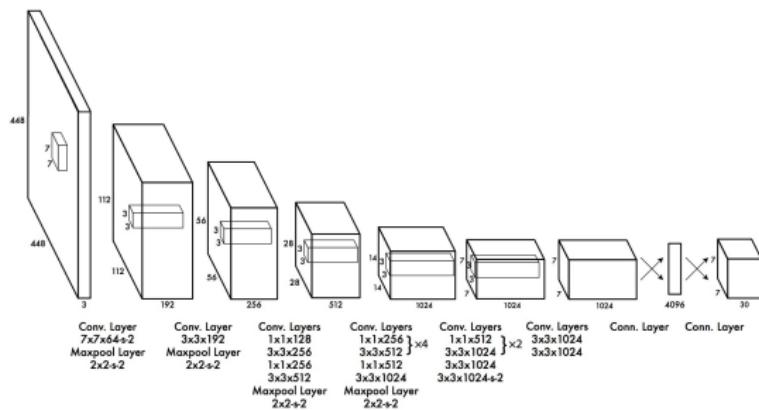


Fig. 3.18 – Structure globale de DarkNet

Création du réseau de neurones

Variété de solutions

- Differents extracteurs de caractéristiques :
 - DarkNet : Structure originale.
 - MobileNet : Développée par Google.
 - SqueezeNet : Développée par Stanford.

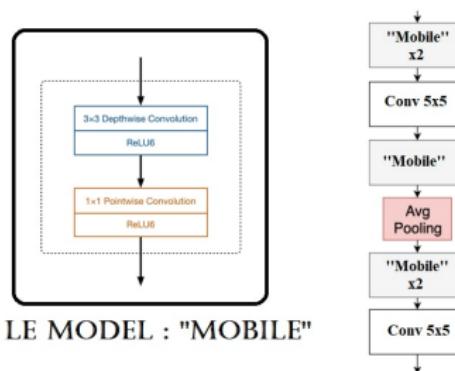


Fig. 3.19 – Unité principale de MobileNet

Création du réseau de neurones

Variété de solutions

- Differents extracteurs de caractéristiques :
 - DarkNet : Structure originale.
 - MobileNet : Développée par Google.
 - SqueezeNet : Développée par Stanford.

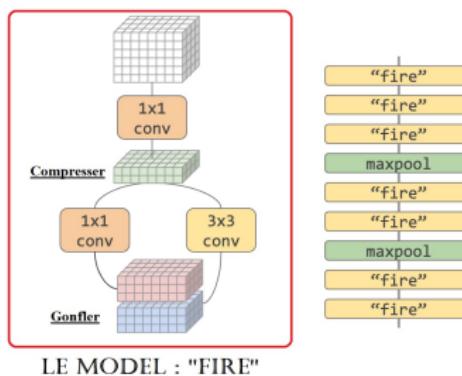


Fig. 3.20 – Unité principale de SqueezeNet

Création du réseau de neurones

Comparaison entre structures

Classifieur	Nombre de paramètres	Temps de traitement	Temps d'entraînement
DarkNet	50,983,561	0.5 s	Plus que 20 jours
MobileNet	3,284,214	0.09 s	6 jours et 3 heures
SqueezeNet	750,198	0.02 s	1 jours et 14 heures

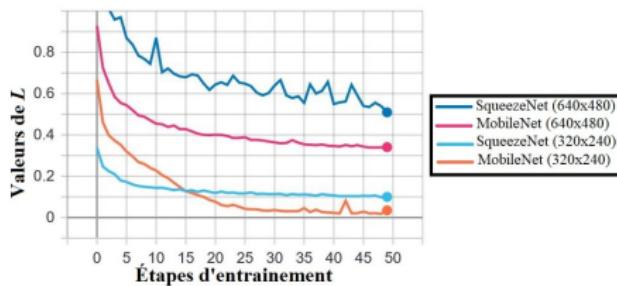


Fig. 3.21 – Resultats d'entraînement

Création du réseau de neurones

Évaluation de la structure retenue : SqueezeNet

- Sur les images :
 - Bons résultats.
 - Bonus : Généralisation.
- Sur un flux vidéo :
 - Détection en temps réel.
 - Bonus : Multi-détection.



Fig. 3.22 – Exemples de détections dans des cas différents

Création du réseau de neurones

Évaluation de la structure retournée : SqueezeNet

- Sur les images :
 - Bons résultats.
 - Bonus : Généralisation.
- Sur un flux vidéo :
 - Détection en temps réel.
 - Bonus : Multi-détection.

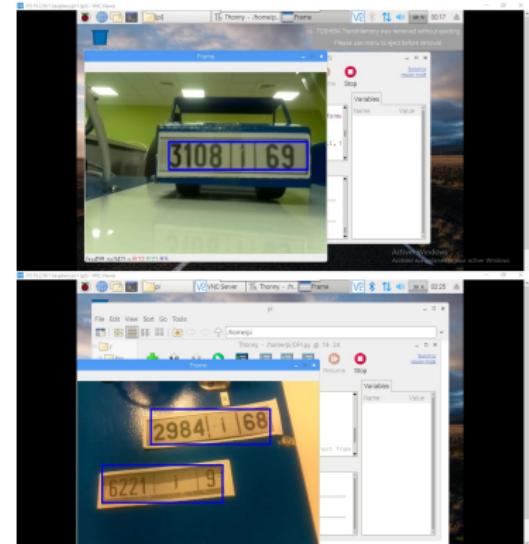


Fig. 3.23 – Resultats avec Raspberry

Un système embarqué

Circuit de commande

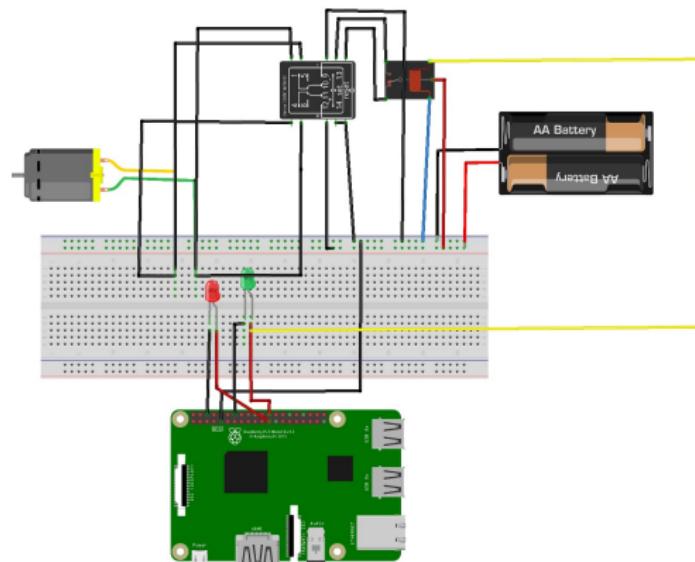


Fig. 4.1 – Montage de commande

Un système embarqué

Experimentation finale

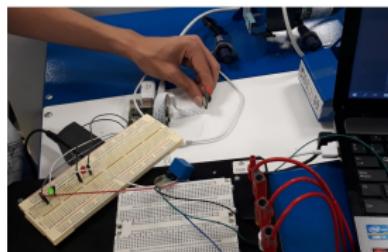
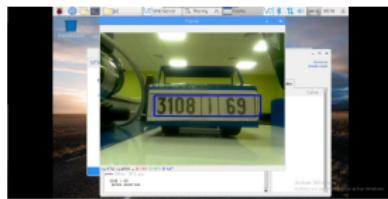


Fig. 4.2 – Essai sur une plaque autorisée

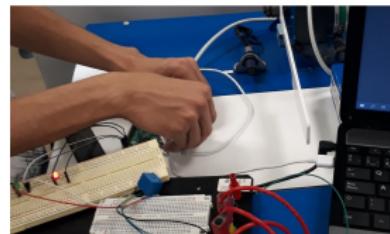


Fig. 4.3 – Essai sur une plaque non autorisée

Conclusion

- Les réseaux de neurones sont puissants
 - Etat de l'art dans plusieurs domaines ;
 - Peuvent encore être développés ;
- Pourtant ...
 - Ils sont très compliqués à implémenter ;
 - Difficulté de les entraîner ;

Merci pour
votre
attention !