

## Réseau de neurones et apprentissage profond

### TP5 : Localisation de plaque d'immatriculation

L'objectif de ce TP est de concevoir et d'entraîner un réseau de neurones capable de localiser une plaque d'immatriculation dans une image (Figure 1).



Figure 1: Boites englobantes tracées à partir des localisations obtenues suite à l'entraînement du réseau de neurones.

#### I. Jeux de données

Comme dans les TP précédents, un sous ensemble de la base de données CCPD est scindé en deux parties, l'une pour l'entraînement, la seconde pour la validation.

Afin de faciliter le chargement de ces données, la classe “CCPD\_SubDataset” (héritage de la classe abstraite “torch.utils.data.Dataset”) vous est fournie. Cette classe permet de charger le dataset, ce qui inclus le pré-traitement des images (dimensions et format) et l'extraction de la vérité terrain depuis le nom de fichier de chaque image. Dans le cadre de ce TP, la vérité terrain correspond à 4 valeurs, le centre, la largeur et la hauteur de la boite englobante de la plaque d'immatriculation visible dans l'image.

Après avoir charger un dataset avec cette classe, il est possible de créer un itérable Python à l'aide de la classe “torch.utils.data.DataLoader” afin pouvoir itérer, batcher et paralléliser l'accès aux données de ce dataset.

#### Travail à réaliser

1. Instancier la classe “CCPD\_SubDataset” afin charger le split d'entraînement et le split de validation.
2. Créer un itérable pour chacun de ces splits.
3. Écrire une fonction permettant d'afficher un batch de données sous la forme d'une grille d'images avec leurs boites englobantes associées (ex. Figure 1 pour un batch de 9).

## II. Architecture du réseau

Le réseau de neurones à implémenter possède cette architecture :

$INPUT \rightarrow [CONV \rightarrow POOL \rightarrow RELU]_{\times 5} \rightarrow [FC \rightarrow RELU]_{\times 2} \rightarrow FC \rightarrow OUTPUT$

où

- $INPUT$  est l'image RGB,
- $CONV$  est une couche de convolution,
- $POOL$  est une couche de pooling (max  $\times 4$  puis avg  $\times 1$ ),
- $RELU$  est une couche d'activation (Unité Linéaire Rectifiée  $\times 7$ , puis Sigmoïde  $\times 1$ ),
- $FC$  est une couche entièrement connectée,
- $OUTPUT$  est la boîte englobante (coordonnées de son centre, sa largeur et sa hauteur).

### Travail à réaliser

1. Implémenter ce réseau de neurones.
2. Discuter le dimensionnement des différentes couches.

## III. Entrainement et évaluations

### Travail à réaliser

1. Entrainer ce réseau de neurones avec l'optimiseur Adam et une fonction de perte L2 (erreur quadratique moyenne (MSE)).
2. En parallèle, évaluer l'apprentissage sur le split de validation.
3. Tracer les fonctions de pertes.
4. Évaluer les performances de votre modèle en calculant l'intersection sur l'union (IoU).