

Cours 6INF911

Mini-Projet : Remise avant le 17 décembre 2025

Détection et classification de panneaux de signalisation : GTSDDB database

Contexte

La base de données GTSDDB – German Traffic Sign Detection Benchmark contient environ 900 images couleur (format PPM), chacune pouvant contenir un nombre de panneaux de signalisation. Elle est utilisée comme référence pour les tâches de détection de panneaux dans le contexte de la conduite autonome et des systèmes d'aide à la conduite.

Dans ce TP, vous utiliserez GTSDDB pour :

- Construire un jeu de données de classification d'images de panneaux à partir des annotations de détection.
- Mettre en œuvre et comparer :
 - un MLP (réseau multicouche),
 - un CNN (réseau de neurones convolutif),
 - une approche de transfert de connaissances (transfer learning) à partir d'un réseau pré-entraîné. Vous pouvez utiliser Python(Keras) ou MATLAB (Deep Learning Toolbox).

Données: GTSDDB

Vous pouvez télécharger la base de données **GTSDDB** nécessaire à ce travail à partir du lien suivant :

<https://erda.ku.dk/public/archives/ff17dc924eba88d5d01a807357d6614c/published-archive.html>

(Public Archive : *ff17dc924eba88d5d01a807357d6614c*)

Au bas de cette page, téléchargez puis décompressez le fichier : [TrainIJCNN2013.zip](#).

Pour ce travail, récupérez :

- **Toutes les images** dont les noms vont de 00000.ppm à 00599.ppm (soit 600 images).
- Le **fichier d'annotations** gt.txt.
Chaque ligne de ce fichier contient :
 - le nom du fichier image,
 - les coordonnées du panneau de signalisation dans l'image (x_min, y_min, x_max, y_max),
 - l'identifiant de la classe correspondante.

Note : Lors de la remise de votre projet, ne pas inclure les fichiers images (Moodle limite la taille des fichiers déposés)

Travail demandé

Partie 1 – Exploration et préparation de la base (10 %)

1. Visualisation et inspection rapide :
 - a) Afficher quelques images GTSDb avec leurs boîtes englobantes (dessiner le rectangle autour du panneau).
 - b) Commenter : taille des panneaux, variations de point de vue, luminosité, occlusions, etc.
2. Extraction de imagettes de panneaux :
 - a) À partir des annotations (voir gt.txt), extraire automatiquement des imagettes (sous-images) à partir de l'image de chaque. Redimensionner les imagettes à une taille fixée (par exemple $32 \times 32 \times 3$ ou $64 \times 64 \times 3$).
3. Filtrage des classes
 - a) Calculer le nombre d'images par classes.
 - b) Pour simplifier, sélectionner seulement les classes les plus fréquentes avec plus que 20 imagettes.
 - c) Diviser vos imagettes en : Entraînement (~70 %), Validation (~15 %) et Test (~15 %). Décrire brièvement la méthode utilisée pour le découpage de vos données.

Partie 2 – Classification avec un MLP (25 %)

1. Préparation des données
 - a) Normalisation les pixels
 - b) (Optionnel) Data augmentation modérée : petites rotations, translations, légères variations de luminosité.
 - c) Vecteurs Input et vecteurs Targets
2. Conception du MLP
 - a) Proposer une architecture minimalement raisonnable, par ex. : 1 à 3 couches cachées, une couche de sortie softmax (classification).
3. Entraînement
 - a) Choisir une fonction de coût (cross-entropy), un optimiseur (SGD, Adam...), un nombre d'époques.
 - b) Tracer les courbes loss / accuracy pour l'entraînement et la validation.
4. Évaluation
 - a) Donner les performances sur le jeu de train et test.
 - b) Afficher les matrices de confusion.
 - c) Commenter les classes les plus confondues et les limites du MLP sur des images.

Partie 3 – Classification avec couches CNN (25 %)

1. Préparation des données (en principe les mêmes données pour le MLP)
 - a) Utiliser les images redimensionnées ($32 \times 32 \times 3$ ou $64 \times 64 \times 3$) sans les aplatir.
 - b) Vérifier que les tenseurs ont la bonne forme (N, H, W, C ou équivalent).
 - c) (Optionnel) Data augmentation modérée : petites rotations, translations, légères variations de luminosité.
2. Conception d'un réseau avec couches CNN
 - Exemple de structure minimale :
 - Conv2D + ReLU + MaxPooling
 - Conv2D + ReLU + MaxPooling
 - (option) Conv2D + ReLU
 - Projection (Flatten)
 - Dense + ReLU + Dropout
 - Dense (nb_classes) + Softmax
3. Entraînement
 - Utiliser le même découpage train/validation/test que pour le MLP.
 - Choisir fonction de coût, optimiseur, nombre d'époques.
 - Tracer les courbes loss / accuracy pour train et validation.
4. Évaluation
 - Donner les matrices de confusion.
 - Comparer avec le MLP :

Partie 4 – Transfert de connaissances (Transfer Learning) (20 %)

On souhaite maintenant utiliser un CNN pré-entraîné et adapter ce modèle à la classification de panneaux GTSDb.

1. Mise en place du transfert de connaissances
 - a) Choisir un modèle pré-entraîné (resnet18, mobilenet_v2, googlenet, VGG16, etc. selon votre environnement).
 - b) Indiquer les adaptations apportées pour utiliser le réseau pré-entraîné
2. Entraîner ce modèle pré-entraîné sur les mêmes images de panneaux en utilisant les mêmes données que MLP et CNN.
3. Comparer les performances, le nombre d'époques nécessaires pour atteindre une bonne performance, la stabilité de l'apprentissage par rapport au CNN entraîné sans transfert de connaissance.

Partie 5 – Localisation (10%)

On s'intéresse à l'apprentissage d'un réseau de neurones qui sera capable de délimiter les formes des objets dans une image (codée en binaire). Le réseau doit être invariant à la translation, la rotation et le changement d'échelle des objets. Le problème peut être ramené dans un premier temps à déterminer le périmètre d'objets simples comme un rectangle ou un disque dans une image. Pour se faire, on vous demande :

1. Génération des données : créer automatiquement des images binaires contenant une seule forme (par exemple un carré, un disque, un triangle), afin de construire une base d'apprentissage convenable.
2. Apprentissage et test du réseau : entraîner le réseau à localiser le contour de la forme, puis évaluer ses performances sur un jeu de test généré avec les mêmes variations (translation, rotation, échelle).
3. Extension du scénario : une fois le cas simple maîtrisé (une seule forme par image), générer des images contenant plusieurs objets de même forme et vérifier la capacité du réseau à détecter toutes les occurrences des formes présentes.

Partie 6 –Synthèse et comparaison globale (10 %)

Dans le rapport, rédiger une section synthèse qui compare les différentes approches :

- a) MLP vs CNN
- b) CNN vs CNN pré-entraîné (transfert de connaissances)
- c) Conclusion générale

Barème proposé (indicatif)

- Partie 1 – Préparation et compréhension des données :	10 %
- Partie 2 – MLP (implémentation + résultats + analyse) :	25 %
- Partie 3 – CNN from sans transfert de connaissances :	25 %
- Partie 4 – Transfert de connaissance :	20 %
- Partie 5 – Synthèse, qualité du rapport et clarté du code :	10 %
- Partie 6 – Synthèse, qualité du rapport et clarté du code :	10 %
Total :	100 %

N'oubliez pas de sauvegarder les poids de vos réseaux et les remettre avec votre rapport, script et un fichier Readme.