

Rapport du laboratoire 2

IAA - Artificial Intelligence for Autonomous Systems

« Following a line »



Kylian Manzini

Loïc Brasey

Dataset

Mise en place

Nous avons fait le choix de créer un dataset de A à Z en recréant l'environnement dans lequel le drone devra voler. Nous avons donc utilisé des lignes blanches au sol pour représenter la trajectoire. Nous avons donc pris le drone et, grâce à un script Python, les images venant directement du drone ont été enregistrées. Voici trois exemples d'images capturées venant de notre dataset de 114 images :



Les images de notre dataset ont été prises dans 4 endroits différents : Le corridor côté fenêtre, le corridor côté intérieur, la salle de classe côté fenêtre et la salle de classe côté corridor. Ces différents endroits permettent d'avoir plusieurs environnements avec des effets différents, comme des réflexions des néons sur le sol ou une forte luminosité dans l'image.

Avantage / Inconvénient

Créer son dataset de cette façon présente de nombreux avantages mais également des inconvénients :

Avantages :

- Les images sont représentatives des conditions réelles dans lesquelles se trouvera le drone.
- Les performances de notre modèle se rapprocheront très probablement des performances lors de l'évaluation du laboratoire par les enseignants.

Inconvénient :

- Il faut annoter toutes les images manuellement. (Nous avons utilisé [ce site web](#) extrêmement pratique pour l'annotation).
- Il est nécessaire de filtrer les images prises par le drone.
- Nous n'avons pas de dataset très grand.
- Le modèle ne sera pas très généraliste et se limitera à des environnements ressemblant à ceux du dataset. (Environnement semblable à la HEIG-VD)

Architecture du modèle

Vue d'ensemble

Nous avons choisi de faire un modèle basé sur la régression. Son but est de générer 2 points représentant le segment de ligne le plus proche.

Nous utilisons un split entre des données d'entraînement et des données de test.

Le réseau de neurone que nous avons créé est composé de multiples étapes :

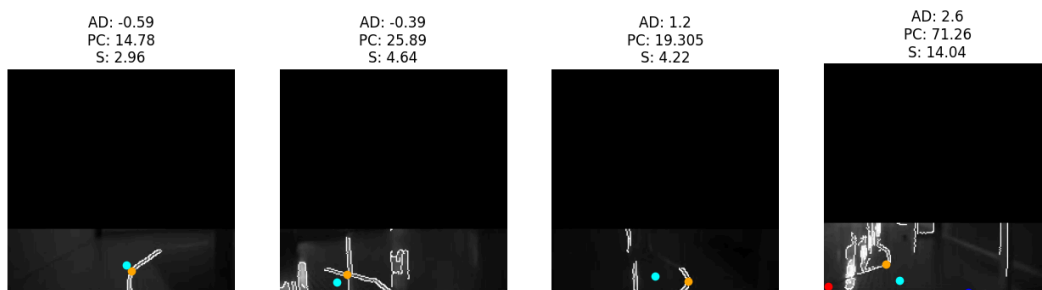
- | | |
|----------------|--------------|
| 1. Convolution | 6. MaxPool2D |
| 2. Relu | 7. Linear |
| 3. MaxPool2D | 8. Relu |
| 4. Convolution | 9. Sigmoid |
| 5. Relu | |

Hyperparamètres

Lorsque notre réseau de neurones semblait cohérent, nous avons essayé de modifier les hyperparamètres de façon empirique jusqu'à trouver un modèle qui semblait être précis selon un score.

Evaluation du modèle

Nous avons évalué les résultats de la prédiction sur le set de test via 2 métriques : l'écart entre l'emplacement des points réel et prédite, ainsi que l'angle de la ligne réel et prédite. Nous sommes partis du principe que l'angle de la ligne prédite était important à prendre en compte afin de diriger au mieux le drone.



Exemple de prédictions avec leurs valeurs évaluatrices. AD est le Angle Difference et PC est le Points Closeness. S représente le score attribué à la prédiction, 0 étant le score parfait

Preprocessing

Nous avons choisi de faire une phase de preprocessing de l'image avant de l'utiliser dans le modèle.

En premier lieu, les 150 pixels du haut de l'image sont mis en noir. Cela vient du fait que l'information importante qui permettra la décision de direction du drone se trouve dans la partie basse de l'image et que cette partie supérieure n'apporte rien mis à part des artefacts et fausses pistes pour la décision.

Ensuite, un passage de traitement est utilisé pour créer une image des bord (edge detection). Cette image sera mélangée avec l'image de base afin de mettre en valeur la ligne blanche en bas.

Toutes les images passeront par ce processus avant d'être utilisées par le modèle.