# Computer Vision

Afin de faire concorder le simulateur et la réalité, réduire l’influence du bruit et simplifier le modèle en réduisant le nombre de paramètres en entrée, nous avons implémenté une partie Computer Vision pour prétraiter l’image d’entrée du modèle.

## Choix

### Correction de l’effet fish-eye

La caméra utilisée sur la voiture crée un effet fish-eye. Il est donc nécessaire de le corriger pour réutiliser les résultats du simulateur.

Cela se fait en calibrant la caméra à l’aide d’une cible de calibration (un échiquier) nous permettant d’obtenir la matrice des déformations dues aux lentilles.

### Filtres appliqués à l’image

Dans les prises de vue réelles, de nombreuses sources de bruit sont présentes, qu’il est nécessaire de corriger. Nous pouvons par exemple citer les variations de luminosité et les reflets sur la piste qu’il est nécessaire d’éliminer afin de détecter correctement les limites de la piste.

Nous avons choisi de combiner plusieurs filtres (inRange, Canny, birdview) afin de donner des informations complémentaires au réseau.

### Calcul de l’écart au centre de piste

Une primitive importante pour connaître notre position sur la piste et la décision de direction et vitesse à prendre est l’écart au centre de piste. Cette valeur est directement connue dans le simulateur, mais doit être calculée lors des roulements sur piste réelle.

Pour cela, nous utilisons un second réseau de neurones de petite taille entraîné sur simulateur, prenant en entrée une image légèrement filtrée (uniquement un inRange).

## Expériences

A chaque amélioration de notre preprocessing, nous faisions des tests des résultats sur piste réelle. Nous pouvions ainsi nous assurer que nos changements donnaient des résultats concrets et cohérents avec ce que nous souhaitions.

A chaque expérience, nous avons suivi le protocole suivant :

**Connexion d’une manette**

**Ajout des nouvelles parties du preprocessing**

**Conduite à la manette avec enregistrement des images**

**Conduite à la manette avec enregistrement des images**

**Vérification des images récupérées et des résultats associés**

**Toutes les modifications testées ?**

Pour optimiser notre conduite, nous connectons une manette de ps4 pour mieux conduire qu’avec la manette fournie par défaut ou le contrôleur web. Lors de la course nous essayons d’être le plus fluide possible et de ne pas trop sortir des lignes. Les images récupérées sont automatiquement enregistrées dans un nouveau répertoire.

## Observations

Etant donné que nous avons testé nos corrections individuellement, nous pouvons discuter de leur résultat sans les corréler.

### Correction du fish-eye



La correction de l’effet fish-eye de la caméra marche correctement. Nous retrouvons une image non déformée de la route, qui permet de transférer de manière plus correcte les modèles entre simulateur et réel.

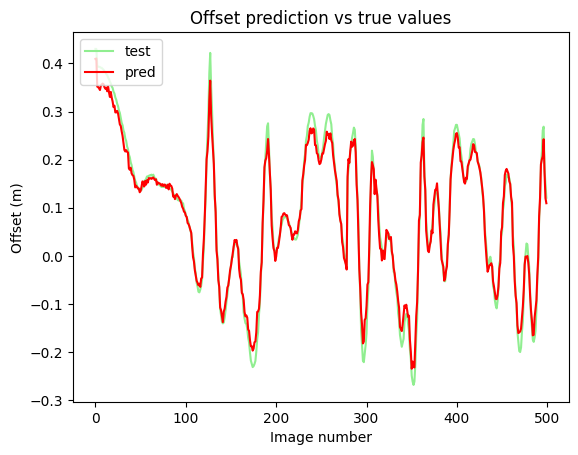
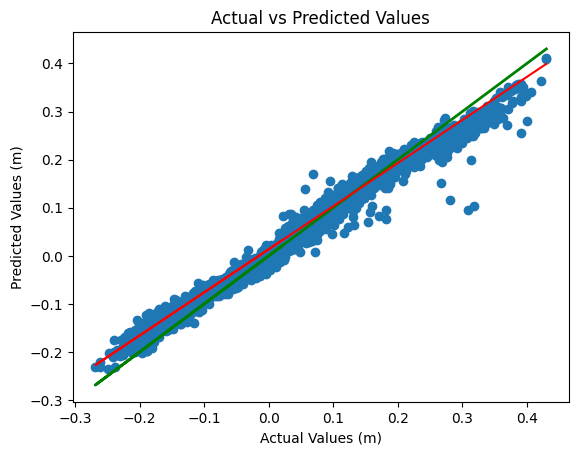
### Filtres appliqués à l’image

A black and blue background with red lines

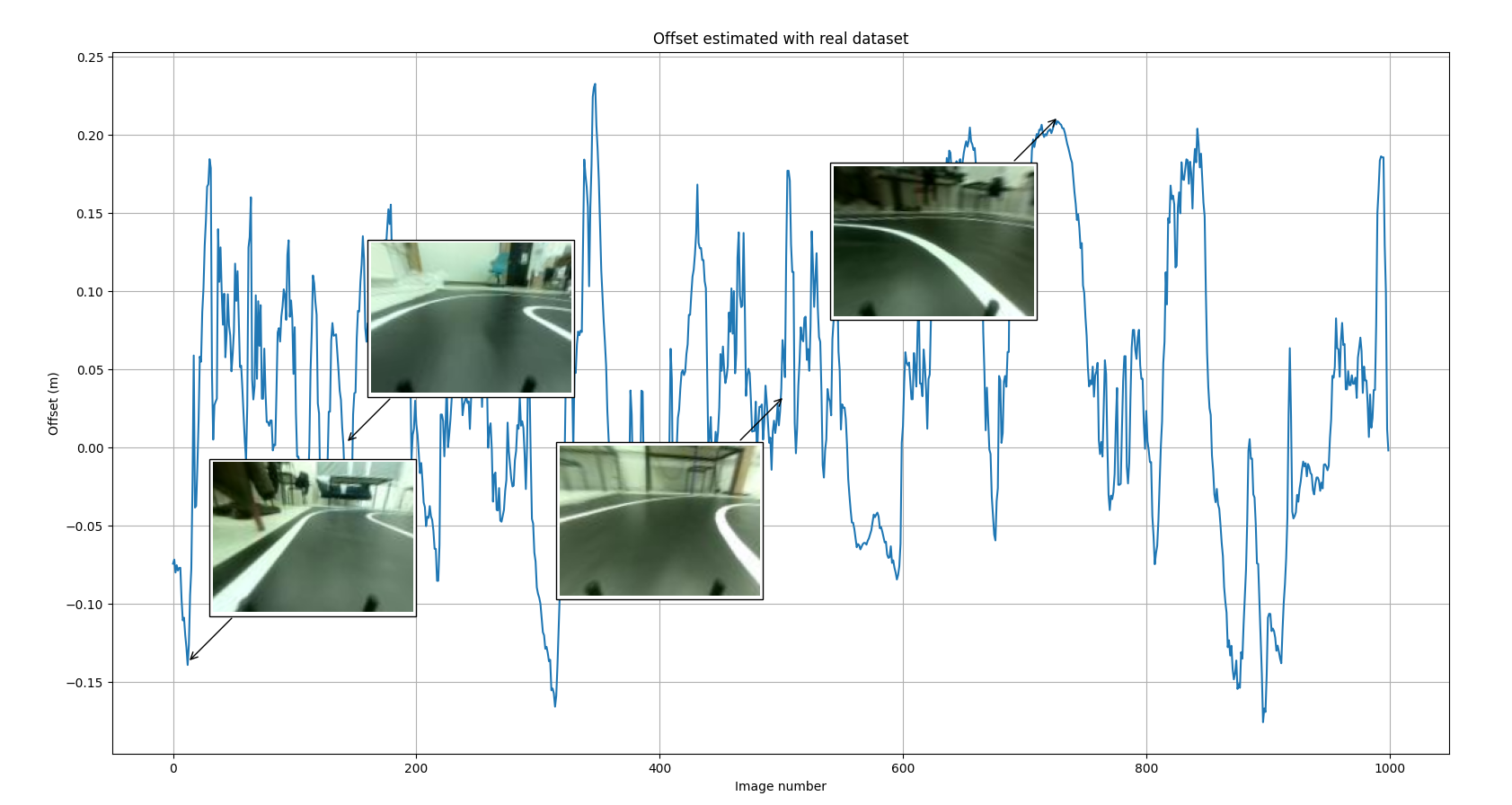
Description automatically generated

La combinaison des trois filtres sur les canaux RGB de l’image nous permet de distinguer correctement les lignes de la piste. Cependant, lors de notre présentation au forum, nous avons pu constater que les reflets sur la piste et sur le sol pouvaient parfois ne pas être filtrés. Après une adaptation des seuils des différents filtres, nous avons pu améliorer le résultat au forum. Nous en retenons cependant que les valeurs utilisées doivent être testées dans différentes conditions pour ne pas se spécialiser dans une unique situation (dans notre cas fenêtres fermées et lumières allumées).

### Calcul de l’écart au centre de piste



Nous avons d’abord utilisé le réseau de neurones en simulateur. On retrouve ici deux représentations des résultats prédits et réels, à gauche une représentation temporelle, et à droite une représentation comparative. On constate que les valeurs prédites sont cohérentes avec les valeurs cherchées, à l’exception des valeurs extrêmes, où il a tendance à légèrement réduire les valeurs. Le réseau de neurones fournit donc un résultat que nous considérons comme satisfaisant en simulateur.



Nous avons ensuite testé le réseau sur un dataset d’images réelles corrigées par l’effet fish-eye. Le résultat ci-dessus nous montre une très bonne corrélation entre les valeurs obtenues (la piste ayant une largeur de 40cm) et la position réelle de la voiture. Il faut cependant être attentif, puisque certaines valeurs sont erronées (voir 3e image ci-dessus, ou la voiture est clairement à droite de la piste mais le CTE pratiquement nul).

## Conclusion et pistes

Finalement, les objectifs de la partie CV du projet ont été partiellement remplis. Nous avons réussi à gommer les différences entre le simulateur et la réalité à l’aide des différents filtres appliqués, et nous avons réussi à déterminer l’une des primitives de la piste : l’écart au centre de voie. Cependant, nous n’avons pas réussi à reconstruire le rayon de courbure de la piste ainsi que les angles proches et lointains que nous souhaitions obtenir au départ. Une idée pour obtenir ces différentes valeurs serait d’entraîner des réseaux de neurones, mais nous n’avions plus le temps nécessaire pour labelliser l’ensemble des images requises pour l'entraînement de tels réseaux.