1. **Partie DeepLearning:**

Les dossiers:

* **PatternRoi**: contient la version initiale du dataset de M.Chabaine.
* **DATASET(test)**: une 50aine des images prises par l’application mobile pour tester les modèles.
* **Dataset, Dataset1 et Datasett2**: 3 versions des hiérarchies différentes pour partitionner les images.
* **original\_denoised:** contient le nouveau dataset après rehaussement, train: 70%

validation, test:15%

* **results**: pour chaque architecture DL, on sauvegarde quelques exécutions dans les dossiers correspondants selon le modèle DL

Note: Pour chaque exécution, il existe 3 images qui ont le même numéro (dans le nom), 2 courbes pour l’accuracy et le loss et une 3ème image pour l’architecture du modèle, exemple: 2.png et 2\_.png et 22.png sont pour la même exécution.

* **RESULTS\_DENOISED\_DATASET**: l’équivalent du dossier résultats, mais pour le dataset augmenté.

Les notebooks google colab:

* **Data\_preprocess**: Fait quelques modifications sur le dataset original et corrections des noms: pour organisation.
* **Split\_Dataset**: contient une fonction pour partitionner les fichiers en 3 directories: train, validation et test selon des coefficients à entrer dans les paramètres.
* **EAD\_Training**: permet de visualiser quelques échantillons, ImageDataGenerator et propose des modèles CNN pour comparer les résultats.

Les meilleurs résultats sont pour Resnet18.

* **DL\_Models\_augmentedD:** commence par faire un rehaussement et augmentation du dataset, et propose 5 modèles CNN: 3 Resnet+ Aquasight\_modifié+ ex nihilo qui a donné la meilleure performance.

1. **Partie IOT et Mobile:**

**Dossier IOT\_ARDUINO:** contient un seul fichier : nodeMcu \_to\_ SpreadSheet.ino pour

calculer la valeur de la turbidité et envoyer les données à une feuille de calcul Google SpreadSheet.

* **Lien pour 2 videos pour la partie iot et mobile réalisés fin Mars 2022:** [**lien\_drive**](https://drive.google.com/drive/folders/1fuPdaSyfiPWKxf6oqIQ-bt6we1j-bK6X?usp=sharing)
* **Lien googleSpreadSheet:** [**lien**](https://docs.google.com/spreadsheets/d/1Q6bY-KrpubYifVJcrrWpAxJ6Rtdz-Ie2SZ1RCbcTvmY/edit#gid=0)
* **Lien github du code de la partie mobile (flutter):** [**lien**](https://github.com/KhelifiBilel/Turbidity_Classification)

Pour installer l’application mobile à partir du code, on installe tout d’abord les packages nécessaires avec la commande: **flutter pub get**

et on compile avec la commande **flutter run**

**Note**: L’application mobile est connectée à google firestore.

Pour suivre le chargement des données au cloud, on se connecte à ce compte Google: **mail:** [**pfa.turbidity@gmail.com**](mailto:pfa.turbidity@gmail.com)

**mot de passe: pfaturbidity01**