

ACADIA

Auto-encodeur / Auto-encoder

Guide de l'utilisateur 

User's guide 

Projet de 5e année d'étude d'ingénieur à l'INSA Hauts-de-France

5th year engineering project at INSA Hauts-de-France



Groupe de travail / Working Group :

Antoine CHEVALLET - Charles DIJON - Etienne HOARAU - Arnaud LEFEBVRE

Professeurs encadrants / Supervising teachers :

David BUECHE - François-Xavier COUDOUX

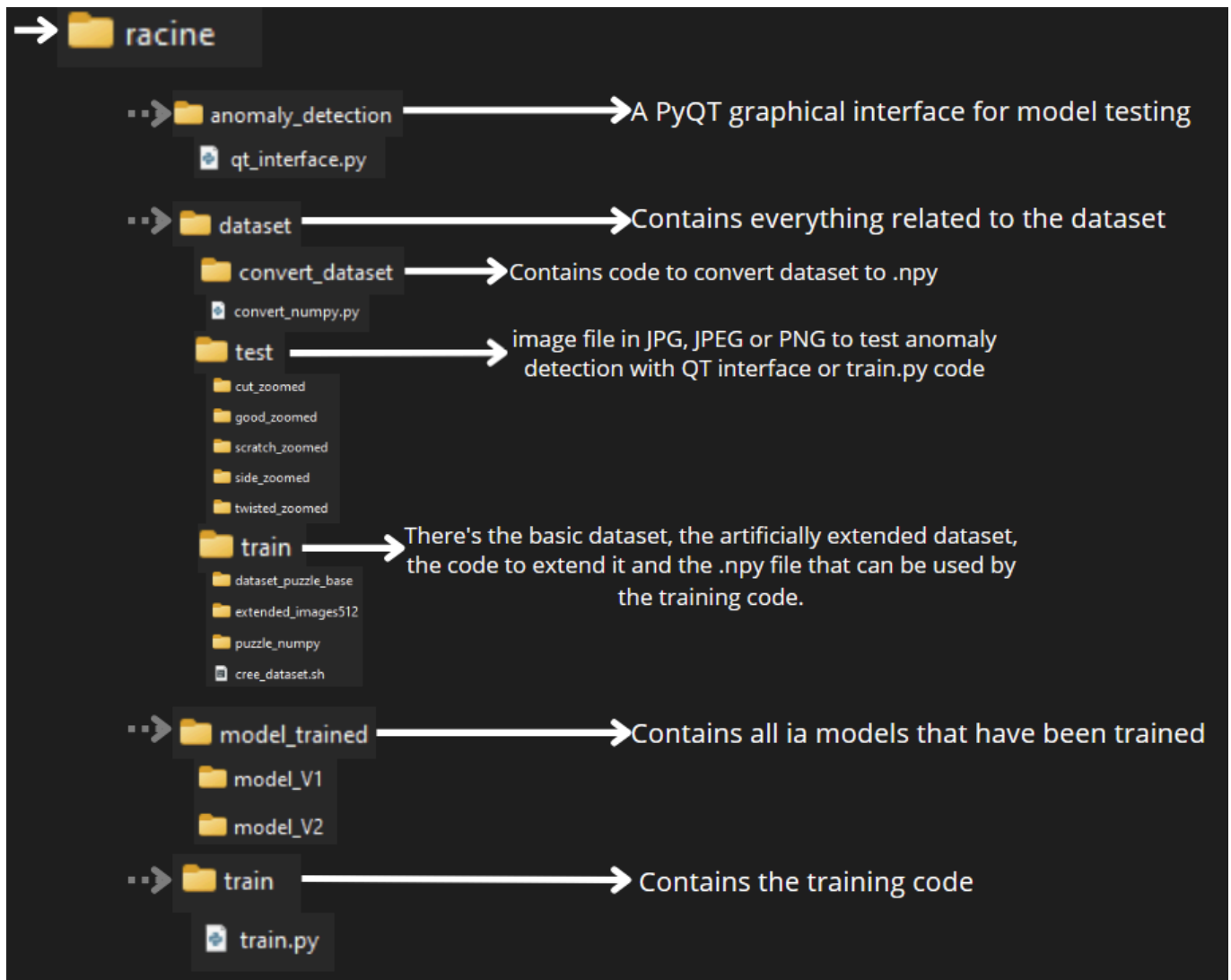
09/2024 - 02/2025

I. Programme et dataset 	3
A. Organisation du code.....	3
B. Utilisation du dataset.....	3
C. Interface QT.....	4
II. Software code and dataset 	5
A. Code organisation.....	5
B. Using a dataset.....	5
C. Qt Interface.....	6

I. Programme et dataset

Notre programme sert à entraîner un autoencodeur. Il est ensuite possible de tester la capacité du modèle à reconstruire l'image et à détecter les anomalies avec une interface QT.

A. Organisation du code



B. Utilisation du dataset

Pour utiliser notre code il y a 4 étapes:

- Importer le dataset voulue dans dataset/train
- Convertir le dataset en .NPY avec le code dataset/convert_dataset/convert_numpy.py
- Lancer le programme train/train.py
- Lancer l'interface anomaly_detection/qt_interface.py

Si vous n'avez pas les ressources nécessaires pour entraîner votre IA. Nous avons adapté le code pour Kaggle : <https://www.kaggle.com/code/arnaudlefebvre7/base-model> avec

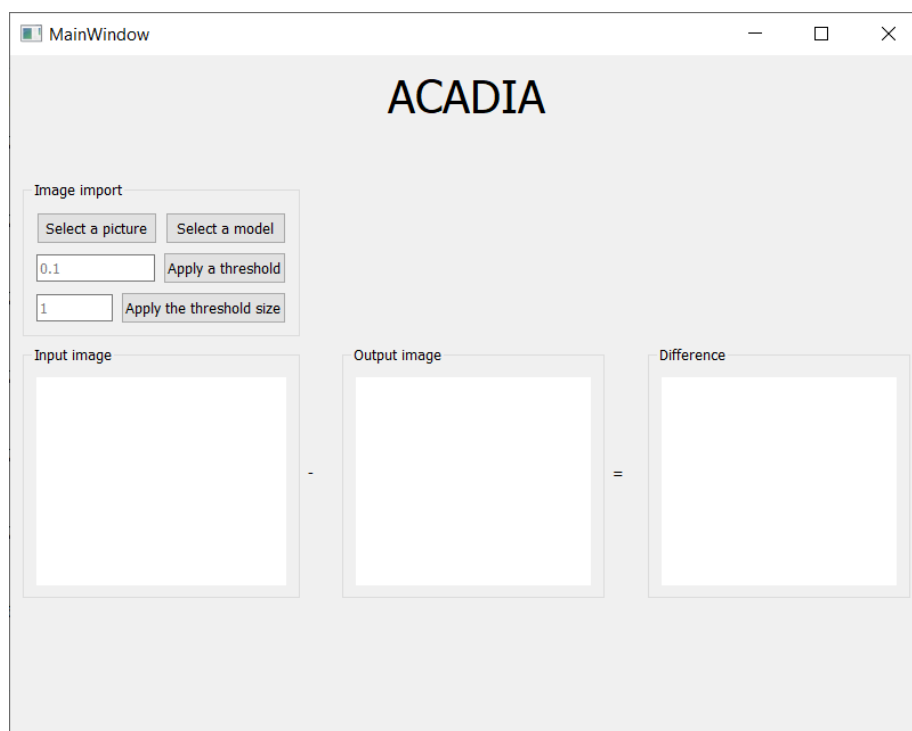
notre dataset de puzzle déjà importé. Il suffira alors d'importer le modèle depuis Kaggle pour utiliser l'interface QT.

Il existe le code `cree_dataset.sh`, un code FFMPEG qui permet d'artificiellement agrandir le dataset avec des rotations, des zooms, l'ajout de bruit gaussien et de symétrie axiale. `Extend_data.py` fait la même chose mais en python. Cependant FFMPEG est plus rapide, il est donc à prioriser pour des images nombreuses et volumineuses. Un code `zoom_images.sh` est disponible pour automatiquement zoomer toutes les images.

C. Interface QT

L'interface graphique Qt est constituée de 3 images centrales permettant de visualiser l'image d'entrée, l'image reconstruite et la différence entre les deux pour visualiser l'anomalie.

L'image de l'anomalie peut être ajustée à l'aide de deux paramètres : un seuil, qui est la première valeur avec 0.1 de base, sachant que les valeurs des pixels sont normalisées entre 0 et 1, et la taille de voisinage autour d'un pixel sur lequel le seuil est calculé

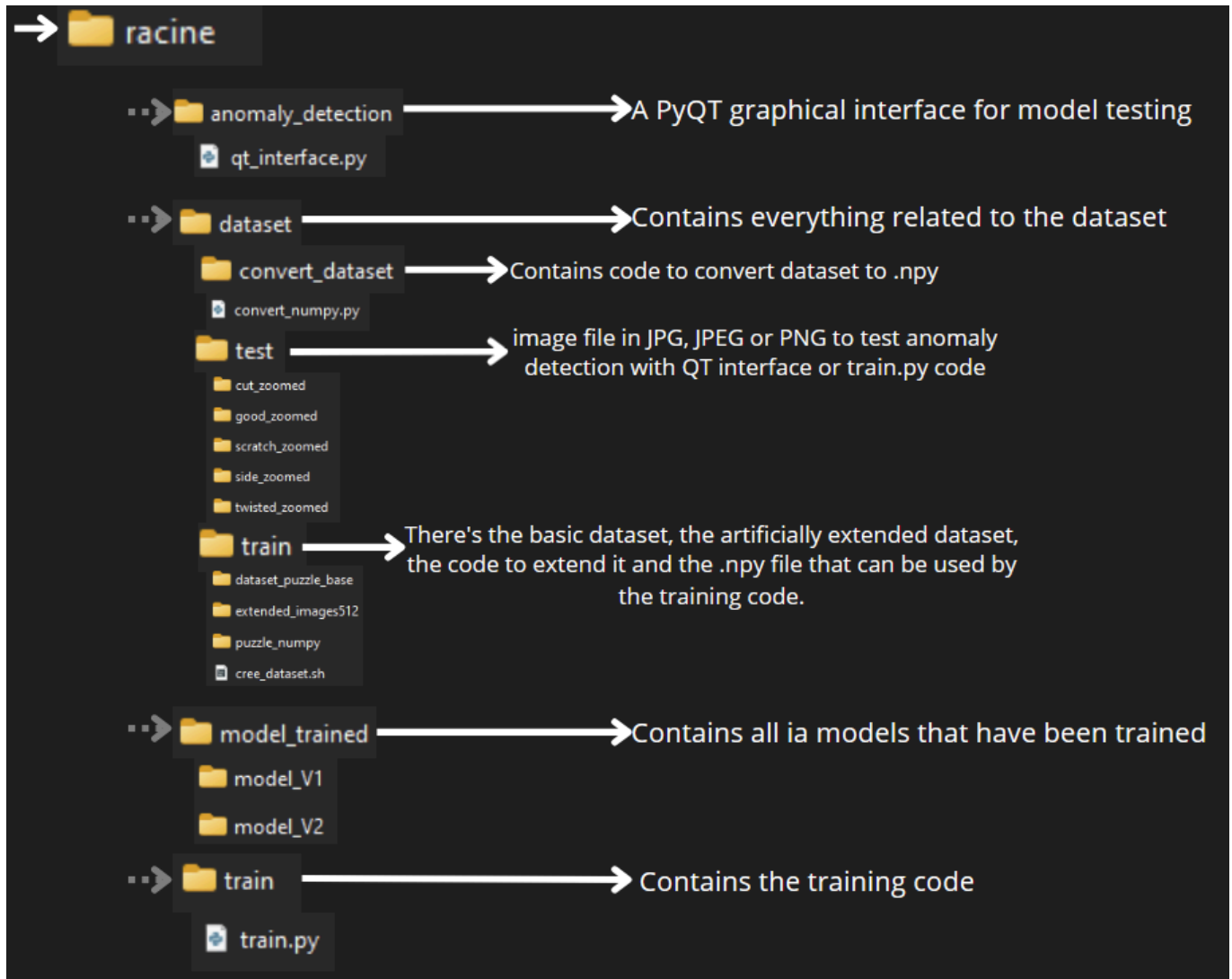


Les deux boutons restants permettent de sélectionner un modèle et une image d'entrée pour ce même modèle. Attention, les images ne se mettront à jour qu'après avoir sélectionné une image d'entrée, et ce même si vous ne voulez pas la changer

II. Software code and dataset

Our program is used to train an autoencoder. It is then possible to test the model's ability to reconstruct the image and detect anomalies with a QT interface.

A. Code organisation



B. Using a dataset

To use our code there are 4 steps:

- Import the desired dataset into dataset/train
- Convert dataset to . NPY with the code dataset/convert_dataset/convert_numpy.py
- Run the train/train.py program
- Launch the anomaly_detection/qt_interface.py interface

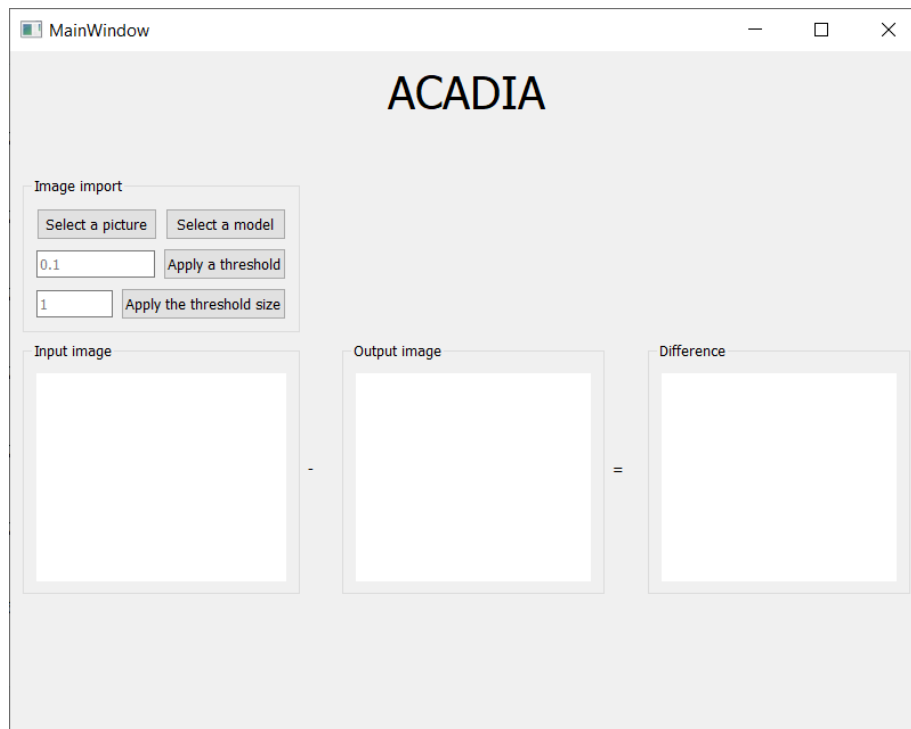
If you do not have the resources to train your AI. We adapted the code for Kaggle: <https://www.kaggle.com/code/arnaudlefebvre7/base-model> with our already imported puzzle dataset. It will then be enough to import the model from Kaggle to use the QT interface.

There is the `cree_dataset.sh` code, a FFMPEG code, which allows to artificially enlarge the dataset with rotations, zooms, adding gaussian noise and axial symmetry. `Extend_data.py` does the same thing but in python. However FFMPEG is faster, so it is to be prioritized for many and large images. A code `zoom_images.sh` is available to automatically zoom all images.

C. Qt Interface

The Qt GUI is made up of 3 central images to view the input image, reconstructed image and difference between the two to view the anomaly.

The image of the anomaly can be adjusted using two parameters: a threshold, which is the first value with 0.1 base, knowing that the pixel values are normalized between 0 and 1, and the size of neighborhood around a pixel on which the threshold is calculated



The remaining two buttons allow you to select a template and an input image for that same template. Warning, images will only update after you select an input image, even if you do not want to change it