

## VERSION FRANCAISE

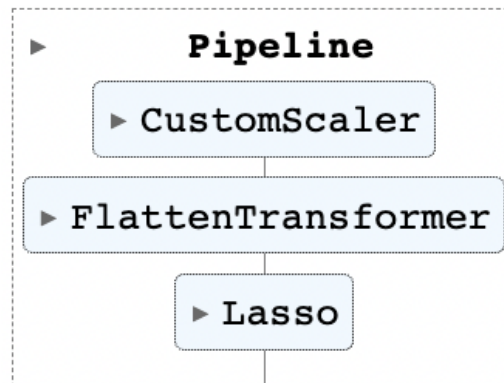
**Sujet:** Estimation de la quantité de biomasse d'une zone à travers les images satellitaires.

**Objectif:** Conception d'un modèle d'IA pour la prédiction de la biomasse d'une zone.

**Baseline:** Pour réaliser la baseline nous avons décidé de mettre en place deux modèles.

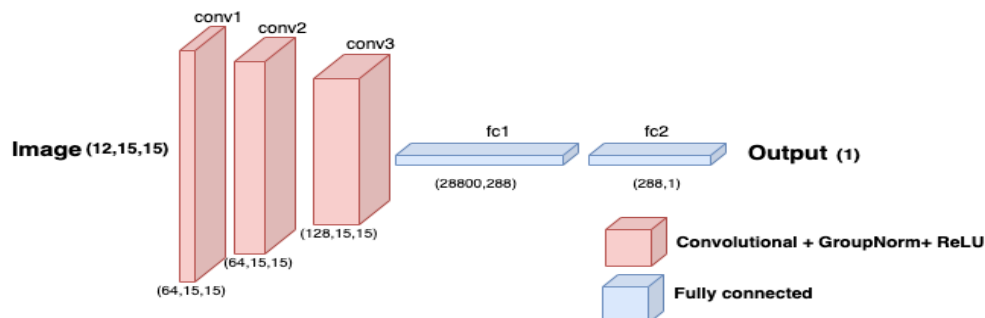
### - Premier modèle

Pour la mise en place d'un tel modèle, nous avons utilisé un pipeline d'entraînement basé sur la librairie Sklearn. Dans ce pipeline on note trois niveaux de transformation des données. Le premier niveau consiste à normaliser les images dans chaque bande spectrale. Ensuite le second niveau transforme chaque image en vecteur (flatten). Ainsi, chaque image passe de (15,15,12) en un vecteur de  $15 \times 15 \times 12$  éléments ou chaque élément est en réalité une valeur de pixels. Enfin le dernier niveau fait appel au modèle choisi pour faire l'entraînement. Pour aller plus loin, nous avons réalisé une **grisearch** en intégrant différents modèles avec différents hyperparamètres dans le pipeline. Le modèle retenu à la fin était : **Lasso** et comme hyperparamètre **alpha: 0.1**.



### - Deuxième modèle

Le deuxième modèle réalisé est basé sur les réseaux de neurones convolutionnels (CNN) et utilise la librairie Pytorch. L'architecture réalisée est un bloc de 3 convolutions. Les images sont standardisées avant de passer au premier bloc de convolution. Ensuite, à la sortie de chaque convolution on applique une normalisation (GroupNorm) et une fonction d'activation ReLU avant son entrée dans la convolution suivante. Une fois les convolutions terminées, on transforme les résultats en vecteur (flatten) qu'on passe dans un réseau entièrement connecté.



**Résultats** : La métrique utilisée est la RMSE

- Premier modèle : 61.566107
- Deuxième modèle : 255.79233
-

## ENGLISH VERSION

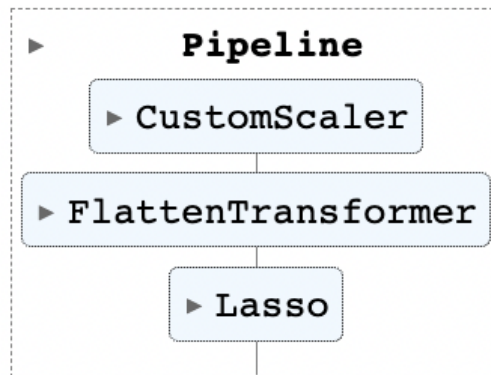
**Subject:** Biomass estimation of an area through satellite images.

**Objective:** Design an AI model to predict the biomass of an area.

**Baseline:** To achieve the baseline we decided to implement two models.

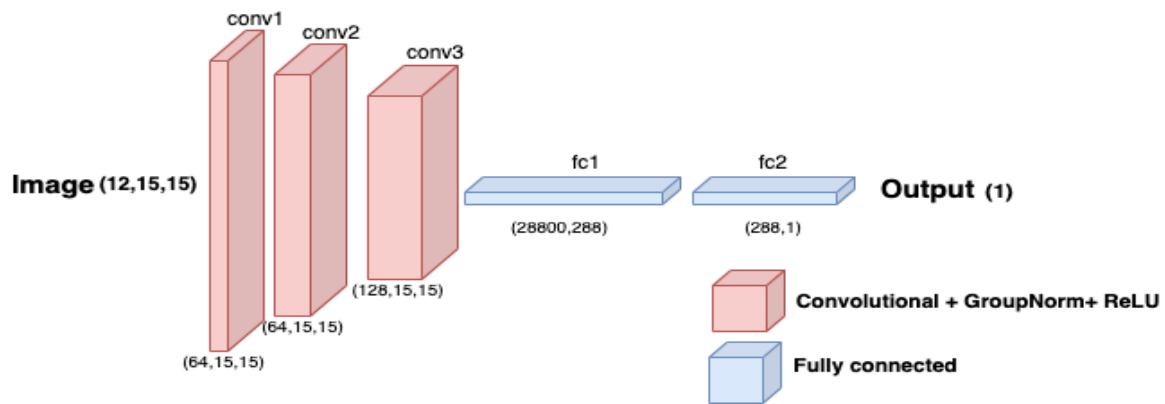
- First model

For the implementation of such a model, we used a training pipeline based on the Sklearn library. In this pipeline we note three levels of data transformation. The first level consists in normalizing the images in each spectral band. Then the second level transforms each image into a vector (flatten). Thus, each image passes from (15,15,12) to a vector of  $15 \times 15 \times 12$  elements where each element is actually a pixel value. Finally, the last level uses the model chosen to do the training. To go further, we performed a *grisearch* by integrating different models with different hyperparameters in the pipeline. The model selected at the end was: **Lasso** and as hyperparameter **alpha: 0.1**.



- Second model

The second model realized is based on convolutional neural networks (CNN) and uses the Pytorch library. The realized architecture is a block of 3 convolutions. The images are standardized before going to the first convolution block. Then, at the output of each convolution we apply a normalization (GroupNorm) and a ReLU activation function before entering the next convolution. Once the convolutions are completed, the results are transformed into a vector (flatten) that is passed into a fully connected network.



**Results:** The metric used is the RMSE

- First model : 52.93983
- Second model : 255.79233