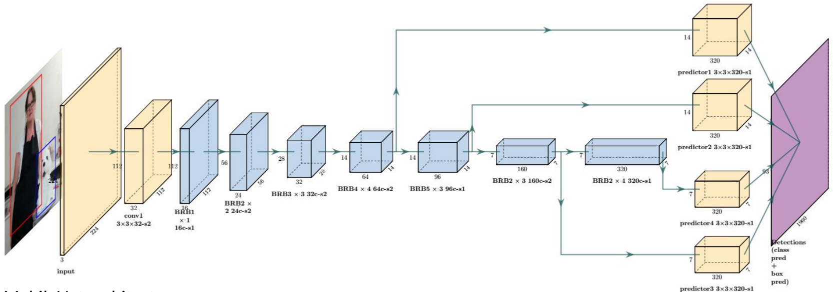
Schneider Electric European Hackathon

Zero deforestation mission

Álvaro Campillos Delgado
Sergio Sánchez Vallés

Transfer Learning approach



MobileNet architecture

Due to the computational costs of creating and training a new model from scratch, we have used transfer learning in order to take advantage of the predictive capacity of pretrained models and refine them to make predictions of our dataset.

We have trained and tested the following neural network architectures initializing their weights to the ones they used in ImageNet, freezing all the layers and adding our own on top:

- **MobileNetV2**.
- **ResNet50.**
- **EfficientNet.**

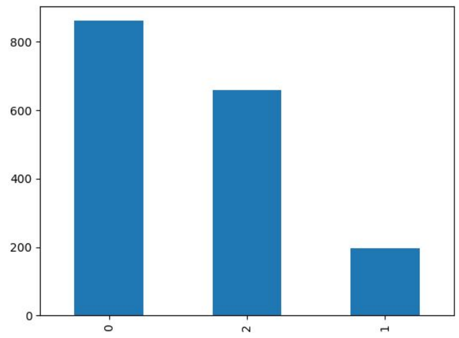
We have finally selected MobileNetV2 because we achieved better results and shorter training times.

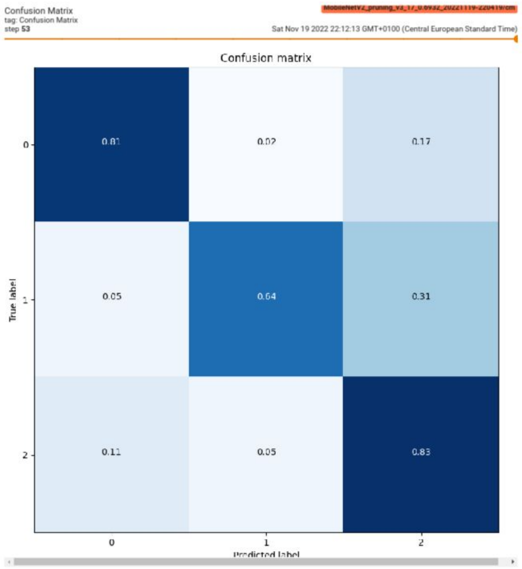
Data preprocessing and model training

When doing exploratory data analysis on the CSV files, we found that the three existing classes were **heavily unbalanced**. We weighed the loss function based on their frequency on the dataset to make the model make better predictions for the less frequent classes.

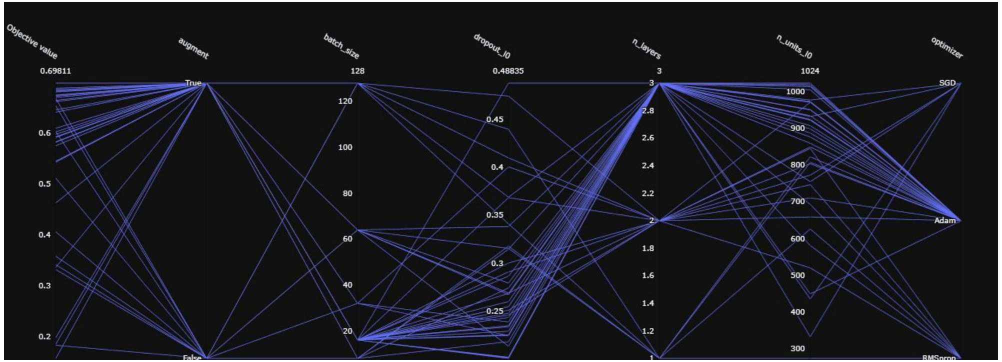
We also used **data augmentation** to increase the dataset size. We used the following transformations: *horizontal flip*, *vertical flip* and *90 degrees rotations*. The final dataset was made of the original images and the transformed ones.

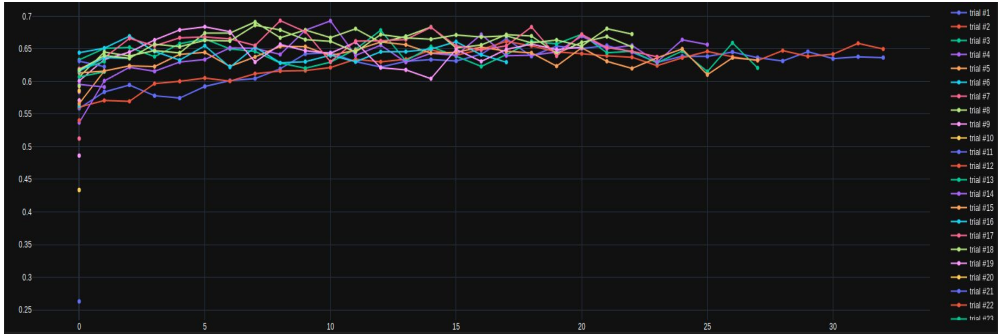
We used ***TensorBoard*** to track the models' training process, checking the loss function evolution and its classification capacity through a live confusion matrix.





Hyperparameter tuning: Optuna





After several experiments with Optuna, we adjusted the hyperparameters bounds. We considered the following:

Layers	[1 - 6]
Units	[256 - 1024]
Optimizers	{Adam, RMSProp, SGD}
Learning rate	[0.00001 - 0.001]
Batch size	{8, 16, 32, 64,128}
Dropout	[0.2 - 0.5]

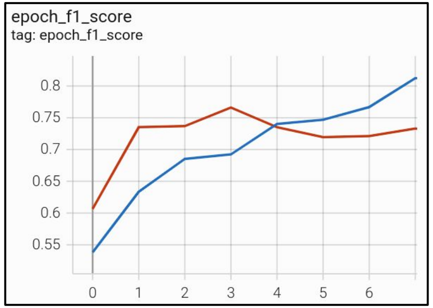
Optimization methods:

- **TPESampler** (Tree-Structured Parzen Estimator)
- **SuccessiveHalvingPruner** (Asynchronous Successive Halving)

Final Model

- We saved the best model from the transfer learning with hyperparameter tuning using Optuna, scoring a F1 Macro Score of **0.6932**

- For the last part, we went further and carry out **fine-tuning**. The last model is unfrozen fully, and retrained with the new data at a very slow learning rate with a few epochs. This has the potential to lead to large improvements by progressively adapting the pretrained features to the new input …



Leading us to a final F1 Macro Score of **0.7661**