**Theoretical description of the problem**

Neural networks are used to solve complex tasks that require analytical calculations similar to those of the human brain. The most common applications of neural networks are:

Classification-distribution of data by parameters. For example, you can enter a set of people and decide who should get a loan and who should not. This work can be done by a neural network, analyzing information such as age, solvency, credit history, and so on.

Prediction-the ability to predict the next step. For example, the rise or fall of a stock based on the situation in the stock market.

Recognition is currently the most widely used neural network. Used in Google when you search for a photo or in phone cameras when it detects the position of your face and highlights it, and more.

**Description of the model**

We apply a set of augmentations with random order and customizable probabilities for each of the transformations

It is unlikely to perform reliable validation with such an imbalance of classes and a small number of samples, so we use all the marked samples to train the model. This should at least slightly improve the efficiency of the classification, especially for small classes

To compare the degree of influence of augmentation on target metrics, create 2 datasets (one with augmentation, the other without)

Creating a neural network with large dropouts

Adding random noise to uploaded images from train

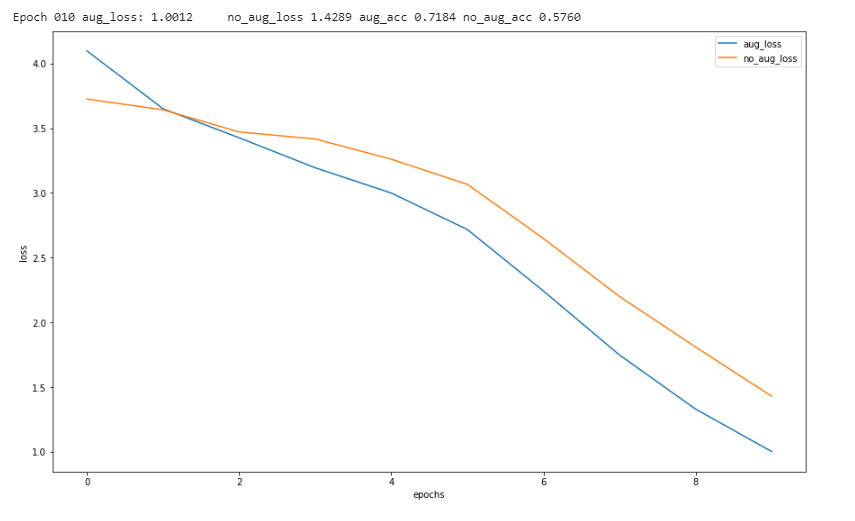
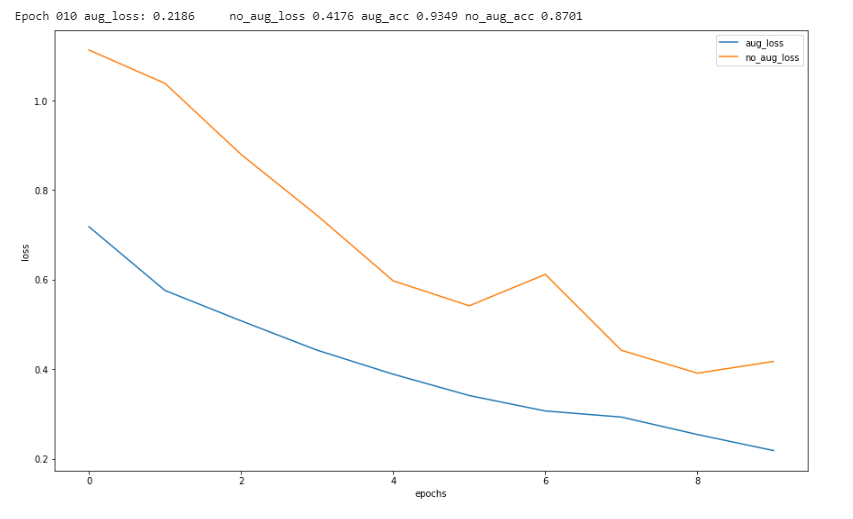
We use weighted random sampler inside data loader to deal with class imbalance

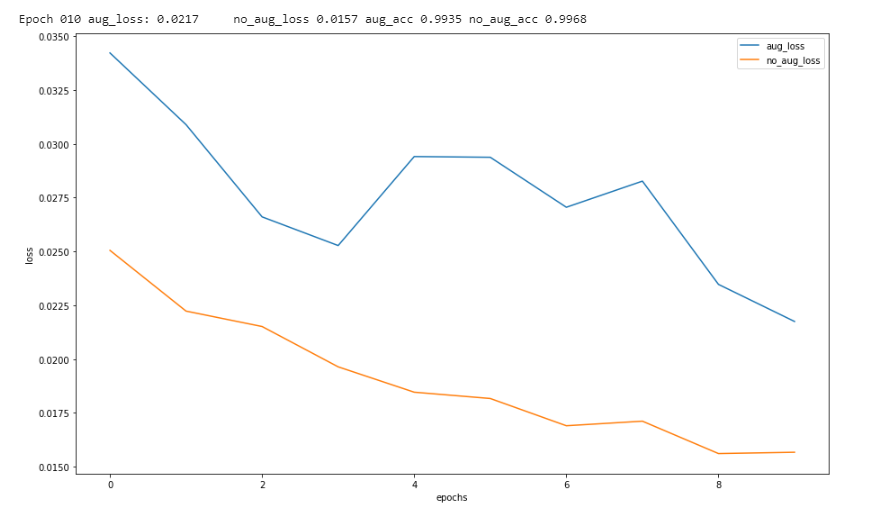
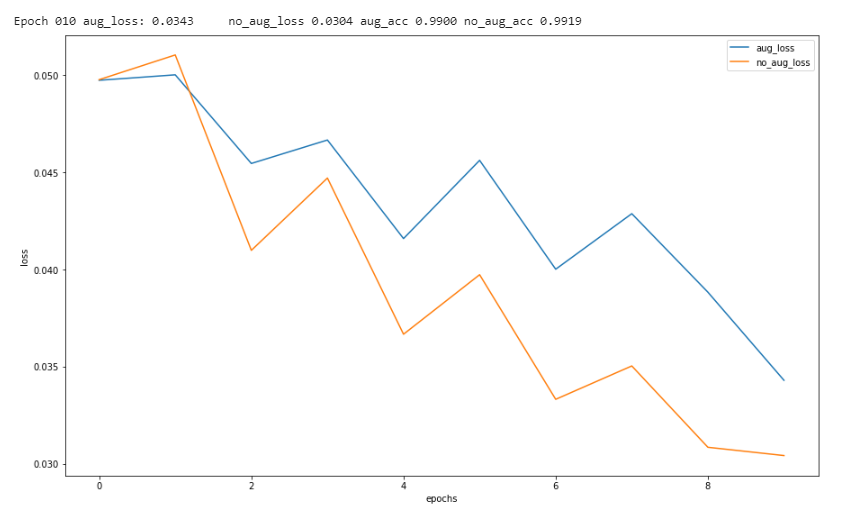
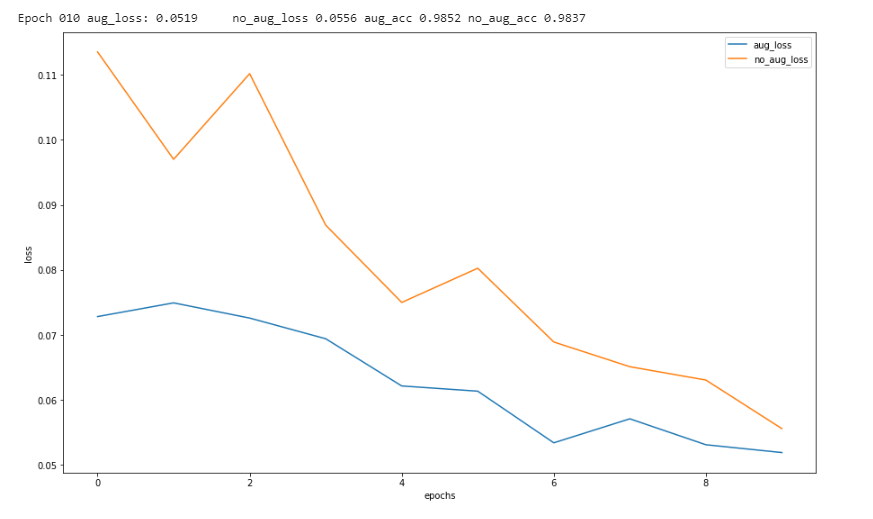
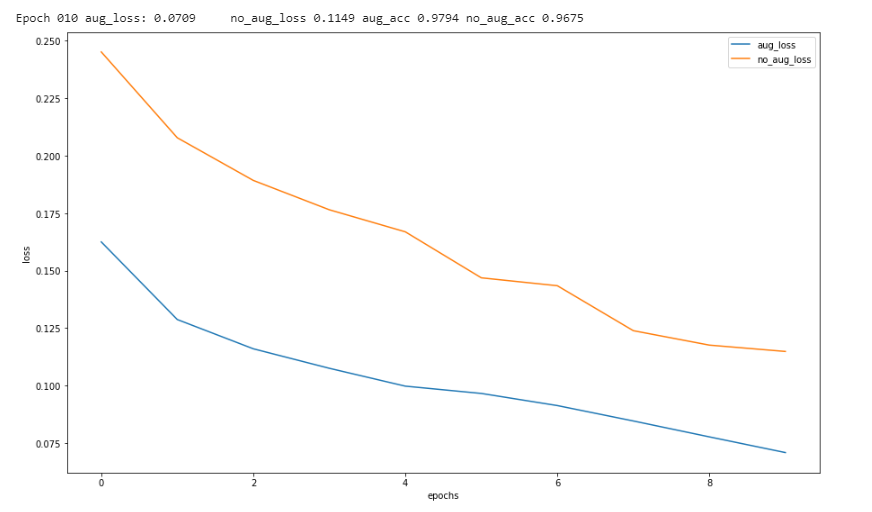
The Butch size must be a multiple of the number of classes (i.e.: 42; 84; 126; 168, etc.)

Changing the batch size and learning rate every 10 epochs

Periodically, we save intermediate States of the model weights (they may be useful to roll back training)

**Target metrics and learning curve**

****

****