State-of-the art methodologies to deal with imbalanced data:

1. Sampling to yield balanced data, under-sampling/over-sampling etc.
2. Synthetic data generation to yield balanced data
3. Cost-sensitive learning, assigning costs of misclassification to instances via a cost matrix

<https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/69/5173046/5128907/5128907-fig-7-source-large.gif>

With this framework at hand, cost sensitivity can be introduced to neural networks in four ways [65]: first, cost-sensitive modifications can be applied to the probabilistic estimate; second, the neural network outputs can be made cost-sensitive; third, cost-sensitive modifications can be applied to the learning rate η; and fourth, the error minimization function can be adapted to account for expected costs.

<https://doi.org/10.1109/TKDE.2008.239>

<https://doi.org/10.1016/j.media.2016.05.004>

Two-phase training:

First phase, train on equiprobable classes. Then fine tune (the last layers) with focus on underrepresented classes.

https://doi.org/10.1016/j.jag.2022.102690

Oversampling methods increase computational cost and may be more prone to overfitting due to the inclusion of duplicated data. On the other hand, undersampling methods can discard important data for learning, reducing accuracy in the prediction.

Approaches are also based on constraints during training, such as restricting the number of pixels contributing to the loss function during backpropagation at random ([Bansal et al., 2016](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "b0010)), based on the k highest loss of the pixels ([Wu et al., 2016](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "b0195)) or hard samples ([Dong et al., 2019](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "b0065)). [Huang et al. (2016)](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "b0090) reduced the effect of class imbalance by enforcing inter-cluster and inter-class margins in standard deep learning frameworks. These margins can be applied through quintuplet instance sampling and the associated triple-header hinge loss. [Ren et al. (2018)](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "b0165) proposed a meta-learning framework that assigns weights to training examples based on their gradient directions to reduce class imbalance and corrupted label problems. Recently, focal loss ([Lin et al., 2020](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "b0120)) was proposed to penalize hard samples assuming that they belong to the minority class. However, this does not happen when minority classes are well defined and may not have their participation in training effectively. A survey on deep learning with class imbalance can be found in [Johnson and Khoshgoftaar (2019)](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "b0105).

To improve these issues, we propose to weight the contribution of each pixel based on its labeled class importance and uncertainty of its labeling as shown in [Fig. 1](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "f0005). A weight for each pixel  is used in the loss function according to Eq. [3](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "e0015).(3)Unlike other approaches (e.g., focal loss ([Lin et al., 2020](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub#b0120))), the weight  of the pixel *x*is calculated by considering two important characteristics as shown in Eq. [4](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "e0020). The first part  considers class imbalance, where  is the class labeled for pixel *x*. The second part  considers the labeling uncertainty of the pixel *x*. Both parts are described in detail in the sections below.(4)

The first characteristic takes the unbalance of classes into account. To determine the weight of each class c, we use the training set according to Eq. [5](https://www.sciencedirect.com/science/article/pii/S0303243422000162?via%3Dihub" \l "e0025). The lower the number of pixels in a given class, the higher the weight so that CNN layer filters fit evenly. When  equals 1 for all classes, training is performed as traditionally. It is important to note that this weight is the same for all pixels in the same class c.(5)where m is the number of pixels of all training images, C is the number of classes, and is the number of pixels that belong to class c.

https://ars.els-cdn.com/content/image/1-s2.0-S0303243422000162-gr1.jpg

<https://doi.org/10.3390/rs13163220>

more loss functions here

<https://doi.org/10.1109/ICCV.2017.324>

Focal Loss

Why is mIoU working?

<https://doi.org/10.6094/UNIFR/150065>