Object Localization And Classification with an imbalanced dataset (IIT-AR-13K)

Gabriele Tazza-1679839, Luca Urban-1651869,
Marco Di Ciò-1889875, Shadi Andishmand-1919010
our github

Abstract
For this project we explored the problem of class imbalancing in image classification and object localization. We had an imbalanced dataset of real document images with 5 classes: natural image, table, signature, figure and logo. We tried to understand which are the best techniques to overcome the problem of class imbalancing. We analyzed some solutions like oversampling, data augmentation and focal loss. The results show the best techniques are different with respect to the quality of the images. The original dimension of the images was about (800, 1500), in the first case we rescaled them at (224, 224), in the second we used a dimension of (600, 600).

Introduction

In dataset like IIIT-AR-13K the presence of dominant classes can affect the performance of the training because the loss and the gradient are too dependent on them. In our case we want to see how oversampling, data augmentation and focal loss can affect this problem.

Dataset and metrics

The dataset we've used to solve the class imbalance problem is the IIIT-AR-13K. This dataset was made to make a multi-object detection and classification, but we wanted to explore the object localization and classification problem. So we extracted the images with only one element into it and we obtain a new dataset with about 8 thousand images in which the categories are divided in this way:

Classes	Before	After
Natural images	12,5%	15,9%
Tables	70%	70,3%
Signatures	2,6%	1,6%
Figures	12,5%	10,9%
Logos	2,4%	1,3%

The new dataset emphasizes even more the class imbalance problem because the percentages for the **Signatures** and the **Logos** are lower than the original dataset with respect to the dominant class (**Tables**).
The metrics we've used to measure the quality of the models and the proposed techniques are the **accuracy** for the classification problem and the **F1** score with an **IoU threshold** (that is a **Jaccard Index = Intersection over Union**) to measure the quality of the bounded boxes together with the accuracy for the localization and classification part.

Proposed methods explained

What we expected is that the balancing of the dataset through oversampling is insufficient because of the absence of diversity in the oversampled observations and it can lead to overfitting. To solve this problem of absence of diversity the best method is data augmentation which allows to create new synthetic observations starting from the oversampled ones. The last method, Focal Loss, should penalize the dominant classes in order to avoid the huge dependence of loss and gradient form them.

1. **Oversampling:** we decided to do an extreme balancing of the dataset by oversample all the classes until reaching the same number of observations for each class. In this way we assure that every batch is balanced.

2. **Oversampling + Augmentation:** we performed augmentation by randomly applying changes to batches. These changes involve random values of saturation, contrast, brightness, hue and gray scale.

3. **Focal Loss:** like many other cases of classification we had an imbalanced dataset which has an effect on the model which is using simple cross entropy. To solve this problem we used Focal loss which is improving cross entropy with assigning more weights to hard examples(misclassified) and down weights to the easy ones(correctly classified) and increasing the importance of misclassified examples which in our case are “logo” and “signature”.

   $$FL(pt) = -\alpha(1-pt)^{\gamma} log(pt)$$

   γ controls the shape of the curve. The higher the value of γ, the lower the loss for well-classified examples, so we could turn the attention of the model more towards ‘hard-to-classify examples. Having higher γ extends the range in which an example receives low loss. Alpha can give high weights to the rare class and small weights to the dominating or common class. [4]

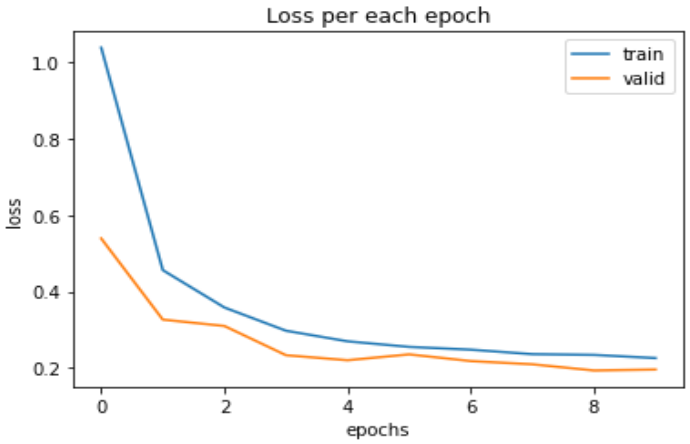
Experimental Results

For each method we performed an hyperparameters tuning on learning rate, batch size, momentum and for focal loss: alpha and gamma, in order to choose the best possible model for each method.

Image dimension → 224, 224

- *Baseline: learning rate: 0.001, momentum: 0.09, batch size: 32, num epochs: 10*

	IoU_th=0.5	IoU_th=0.55	IoU_th=0.6	IoU_th=0.65	IoU_th=0.7	IoU_th=0.75	IoU_th=0.8	IoU_th=0.85	IoU_th=0.9	IoU_th=0.95
natural_image	0.98182	0.98182	0.98182	0.98182	0.97916	0.97382	0.94624	0.87032	0.70627	0.44445
table	0.99557	0.99557	0.99558	0.99558	0.99494	0.99302	0.99239	0.9866	0.95732	0.70789
signature	0.92308	0.92308	0.92308	0.92308	0.89473	0.83334	0.64516	0.32001	0.09091	0
figure	0.95964	0.95964	0.95964	0.95964	0.95964	0.95495	0.93088	0.88461	0.68181	0.28149
logo	0.7	0.63158	0.55556	0.375	0.26666	0.26666	0.14286	0	0	0

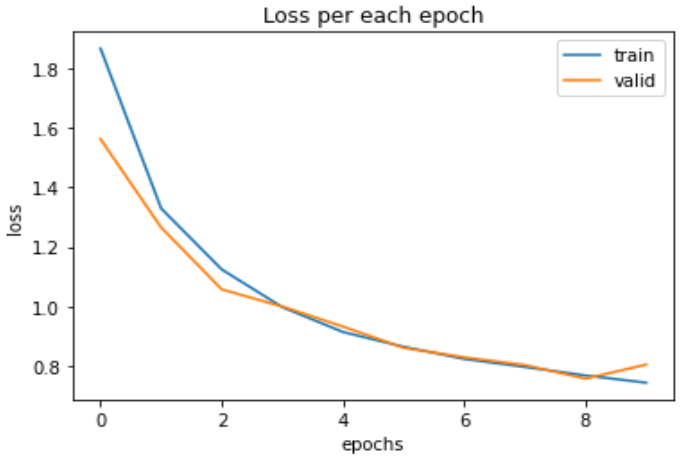


Class	Accuracy
Natural image	0.96429
Table	0.99244
Signature	0.90476
Figure	0.92241
Logo	0.61538

The baseline is runned on the original dataset without any balancing. The results show the problem of class imbalance for accuracy mainly on logos. The **F1 score** with **IoU threshold** reflects the same problem.

- *Over sampling: learning rate: 0.0001, momentum: 0.02, batch size: 64, num epochs: 10*

	IoU_th=0.5	IoU_th=0.55	IoU_th=0.6	IoU_th=0.65	IoU_th=0.7	IoU_th=0.75	IoU_th=0.8	IoU_th=0.85	IoU_th=0.9	IoU_th=0.95
natural_image	0.93452	0.91515	0.89506	0.8635	0.81848	0.75695	0.61776	0.48945	0.3578	0.19193
table	0.92492	0.91326	0.89638	0.87003	0.81667	0.73964	0.60214	0.42805	0.20253	0.02777
signature	0.83871	0.83871	0.83871	0.8	0.66667	0.56	0.28571	0.10527	0	0
figure	0.86458	0.8022	0.69461	0.55629	0.49655	0.3609	0.28347	0.12069	0.01817	0
logo	0.57143	0.57143	0.46154	0.46154	0.46154	0.46154	0.18182	0	0	0

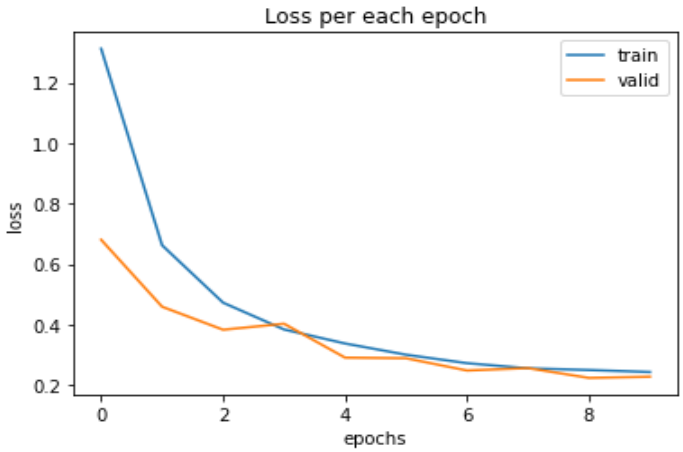


Class	Accuracy
Natural image	0.94898
Table	0.85516
Signature	0.66667
Figure	0.87069
Logo	0.69231

The extreme balancing of the dataset alone is insufficient as we expected, the results show worse performance in **F1** and **accuracy** score for each class. This is probably due to the few diversity in the balanced dataset, we also expected a lot of overfitting caused by the oversampling which is not present in the results.

- *Over sampling + Augmentation: 0.0001, momentum: 0.02, batch size: 64, num epochs: 10*

	IoU_th=0.5	IoU_th=0.55	IoU_th=0.6	IoU_th=0.65	IoU_th=0.7	IoU_th=0.75	IoU_th=0.8	IoU_th=0.85	IoU_th=0.9	IoU_th=0.95
natural_image	0.99487	0.99487	0.99229	0.99229	0.98969	0.97916	0.94624	0.89577	0.75555	0.42571
table	0.99494	0.9943	0.99239	0.98918	0.9866	0.98205	0.9695	0.93779	0.82458	0.36961
signature	0.92308	0.92308	0.89473	0.89473	0.89473	0.89473	0.72727	0.6875	0.32001	0
figure	0.95495	0.95495	0.95023	0.94545	0.94064	0.92593	0.88995	0.81633	0.59394	0.20154
logo	0.7619	0.7619	0.7619	0.63158	0.63158	0.55556	0.47059	0.47059	0.14285	0

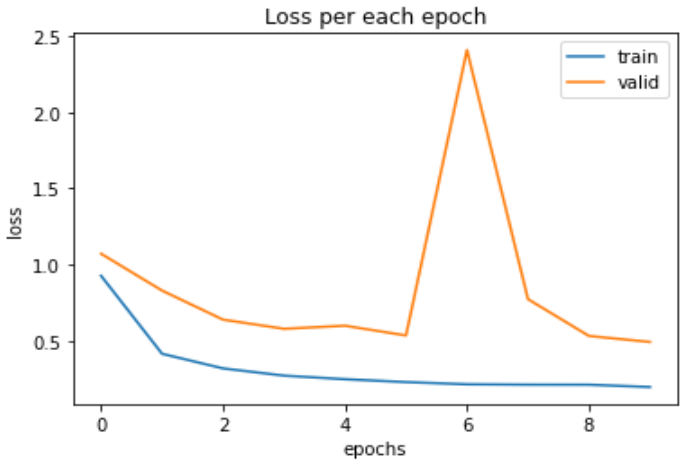


Class	Accuracy
Natural image	0.98469
Table	0.98741
Signature	0.95238
Figure	0.90517
Logo	0.76923

The **augmentation** as we expected performed a lot better than **just balancing** the dataset because of the added diversity of the observations. The results are better than the baseline in both **accuracy** and **F1** score mainly for **logos** and **signatures** (less present classes) but the problem of class imbalancing is still very clear for **logos**.

- *Focal Loss: 0.001, momentum: 0.09, batch size: 32, gamma:1, alpha: 0.8, num epochs: 10*

	IoU_th=0.5	IoU_th=0.55	IoU_th=0.6	IoU_th=0.65	IoU_th=0.7	IoU_th=0.75	IoU_th=0.8	IoU_th=0.85	IoU_th=0.9	IoU_th=0.95
natural_image	0.99229	0.99229	0.98969	0.98182	0.98182	0.97916	0.93767	0.8538	0.73139	0.46875
table	0.99748	0.99748	0.99621	0.99621	0.99621	0.99493	0.9943	0.98724	0.94765	0.67667
signature	0.95	0.95	0.92308	0.92308	0.92308	0.92308	0.89473	0.64516	0.25	0
figure	0.95022	0.95022	0.95023	0.95023	0.95023	0.94064	0.90047	0.79793	0.65896	0.30657
logo	0.7619	0.7619	0.63158	0.55556	0.55556	0.55556	0.375	0	0	0



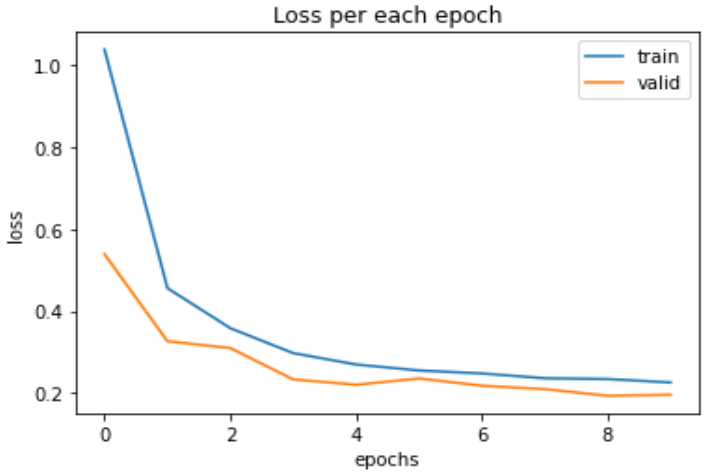
Class	Accuracy
Natural image	0.98469
Table	0.99622
Signature	0.90476
Figure	0.90517
Logo	0.76923

The results for the Focal Loss are not as we expected, the **validation loss** is always very high with some strange behaviour during the learning process. The **accuracies** are higher on the less present classes(**logos** and **signature**). About **F1** the result seems to be a little better than the baseline when the **IoU threshold** increases, this means that the bounding boxes regression is working better than in the baseline.

Image dimension → 600, 600

- *Baseline : learning rate: 0.001, momentum: 0.09, batch size: 16, num epochs: 10*

	IoU_th=0.5	IoU_th=0.55	IoU_th=0.6	IoU_th=0.65	IoU_th=0.7	IoU_th=0.75	IoU_th=0.8	IoU_th=0.85	IoU_th=0.9	IoU_th=0.95
natural_image	0.98182	0.98182	0.98182	0.98182	0.97916	0.97382	0.94624	0.87032	0.70627	0.44445
table	0.99557	0.99557	0.99558	0.99558	0.99494	0.99302	0.99239	0.9866	0.95732	0.70789
signature	0.92308	0.92308	0.92308	0.92308	0.89473	0.83334	0.64516	0.32001	0.09091	0
figure	0.95964	0.95964	0.95964	0.95964	0.95964	0.95495	0.93088	0.88461	0.68181	0.28149
logo	0.7	0.63158	0.55556	0.375	0.26666	0.26666	0.14286	0	0	0

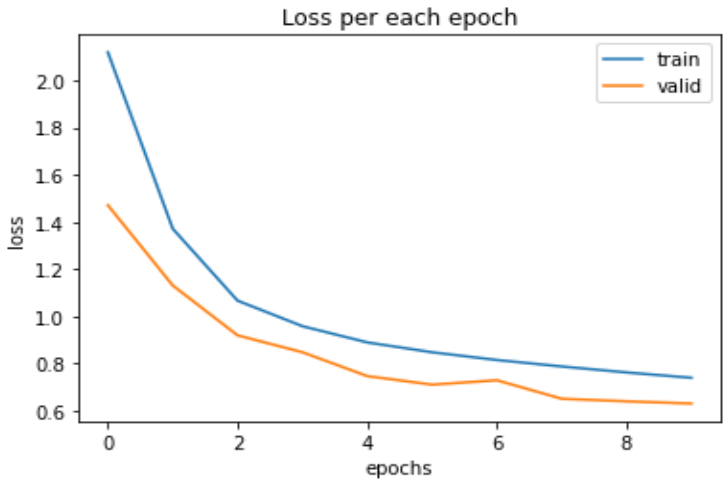


Class	Accuracy
Natural image	0.98469
Table	0.9937
Signature	0.90476
Figure	0.93966
Logo	0.69231

The baseline with images of dimension (600, 600) shows the same behaviour of the previous one(dimension 224, 224) and of course the problem of accuracy for the less present class is very clear.

- *over sampling: learning rate 0.0001, momentum: 0.02, batch size: 16, num epochs: 10*

	IoU_th=0.5	IoU_th=0.55	IoU_th=0.6	IoU_th=0.65	IoU_th=0.7	IoU_th=0.75	IoU_th=0.8	IoU_th=0.85	IoU_th=0.9	IoU_th=0.95
natural_image	0.91843	0.91843	0.91843	0.91843	0.91843	0.9052	0.85623	0.67408	0.50833	0.33489
table	0.92288	0.92288	0.92288	0.92289	0.92152	0.91534	0.88335	0.79063	0.58706	0.15585
signature	0.875	0.875	0.83871	0.8	0.75862	0.66667	0.56	0.28571	0.2	0
figure	0.92611	0.92611	0.92611	0.92611	0.91542	0.84656	0.71006	0.47552	0.19835	0.03604
logo	0.66667	0.66667	0.66667	0.57143	0.57143	0.33333	0.33333	0.18182	0	0

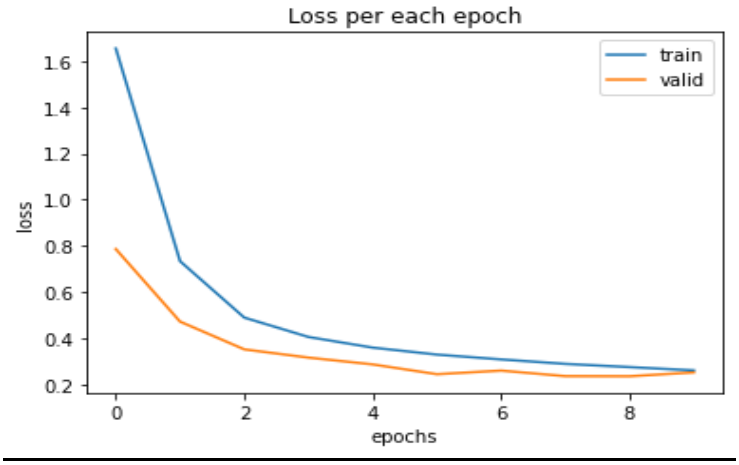


Class	Accuracy
Natural image	0.87755
Table	0.82746
Signature	0.80952
Figure	0.75862
Logo	0.46154

As expected the **accuracy** and the **F1 score** for the different classes for the balanced dataset with 600 px images using the **oversampling** are lower than the **baseline** model. This probably happens because we're passing over and over the same images to the model and it tends to go on **overfitting** both on the **localization** (bounding boxes) and the **classification**. In fact the **loss value** for this model is three times higher than the baseline.

- *over sampling + augmentation: learning rate 0.0001, momentum: 0.02, batch size: 20, num epochs: 10*

	IoU_th=0.5	IoU_th=0.55	IoU_th=0.6	IoU_th=0.65	IoU_th=0.7	IoU_th=0.75	IoU_th=0.8	IoU_th=0.85	IoU_th=0.9	IoU_th=0.95
natural_image	0.86047	0.85714	0.85714	0.8538	0.84024	0.82985	0.78638	0.66666	0.432	0.14218
table	0.99747	0.99747	0.99747	0.99747	0.99684	0.99175	0.97615	0.9263	0.78871	0.34928
signature	0.97561	0.97561	0.97561	0.95	0.92308	0.89473	0.8	0.6875	0.5	0
figure	0.97345	0.97345	0.97345	0.96889	0.96428	0.93578	0.87923	0.74594	0.43243	0.17322
logo	0.81818	0.81818	0.81818	0.76191	0.7	0.7	0.63158	0.26667	0.26667	0.14285

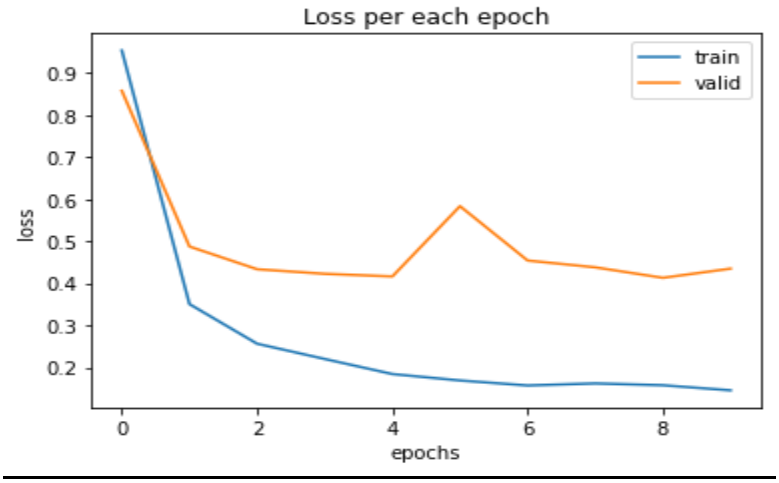


Class	Accuracy
Natural image	0.9898
Table	0.99496
Signature	0.95238
Figure	0.94828
Logo	0.69231

In this case, the oversampling + augmentation doesn't perform better than the baseline on the most underbalanced class (**logos**) but it goes well on the **signatures** in terms of accuracies and also in the localization part because the **F1 scores** are higher than the baseline for all the categories except the **natural images**. This model is better than the simple oversampling of the dataset also because the **validation loss** isn't increasing so much like in the simple oversampling method.

- *focal loss: 0.001, momentum: 0.02, batch size: 16, gamma:1, alpha: 0.6, num epochs: 10*

	IoU_th=0.5	IoU_th=0.55	IoU_th=0.6	IoU_th=0.65	IoU_th=0.7	IoU_th=0.75	IoU_th=0.8	IoU_th=0.85	IoU_th=0.9	IoU_th=0.95
natural_image	0.98969	0.98969	0.98969	0.98969	0.98182	0.9765	0.92603	0.81571	0.68896	0.42571
table	0.99874	0.99874	0.99874	0.99874	0.99874	0.99684	0.98854	0.96883	0.9292	0.76228
signature	0.97561	0.97561	0.97561	0.97561	0.95	0.86486	0.6	0.17392	0	0
figure	0.97798	0.97798	0.97798	0.97798	0.96889	0.95495	0.85714	0.72528	0.61904	0.43243
logo	0.91666	0.91666	0.81818	0.7619	0.7	0.47058	0.14285	0.14285	0	0



Class	Accuracy
Natural image	0.97959
Table	0.99748
Signature	0.95238
Figure	0.9569
Logo	0.84615

The results of the focal loss in this case are unexpected, because with these types of images (600 px) it performs really well on the localization and classification tasks, in fact the **accuracies** and the **F1 scores** are the highest for the underbalanced classes (**logos** and **signatures**) and it remains stable for the other classes. Also in this case the loss value for the validation dataset is very high compared to the baseline, this can be given by the penalization that the focal loss is giving to the oversample classes.

Conclusions and future work

All the experiments we've done show that the problem of **class imbalancing** is very tricky and it involves a lot of factors. The main thing we didn't expect is how much the dimension of the image has an impact on focal loss and data augmentation. The baseline is basically not affected by the increasing of the dimension and the simple balancing performs very badly in both cases as expected.
The **best results** we obtained for both F1 and accuracies is the Focal Loss with images of dimension 600, 600. We are surprised by the little improvement we had from data augmentation for higher dimensional images. We thought the low impact in the case of (224, 224) was due to the little change which data augmentation can give to very low dimensional images but also with high dimensional images the results are not satisfying. This can be solved by using a **more complex augmentation** but all the experiments we did, don't suggest this hypothesis.
The Focal Loss with high dimensional images seems to be the best path to follow in cases like ours.
More possible future works can be done by finding the best proportions to balance the dataset (**not just perfect balancing**) and to explore how very **high dimensional images** can improve or affect the results (we couldn't explore this path since the limit of computational power we had).

References

[1] Dataset: http://cvit.iiit.ac.in/usodi/iiitar13k.php
[2] Dataset paper: http://cdn.iiit.ac.in/cdn/cvit.iiit.ac.in/usodi/img/projects/detection/iiit-ar-13/pdf/IIIT-AR-13K.pdf
[3] Focal Loss paper: https://arxiv.org/pdf/1708.02002.pdf
[4] Focal Loss formula: https://amaarora.github.io/2020/06/29/FocalLoss.html

Role of each member in the work

For the sake of simplicity we are going to **divide the work in 6 parts:**

- **part 1**: code for work with a custom dataset on pytorch(extracting informations from xml files, and writing classes for the custom dataset)
  → ***50% Luca, 50% Gabriele***
- **part 2:** code for the baseline(train resnet50, classification header, regression header, train and validation loops)
  → ***50% Luca, 50% Gabriele***
- **part 3:** code for the metrics(IoU, accuracies and F1 score)
  → ***50% Luca, 50% Gabriele***
- **part 4:** code for the hyper-parameter tuning(grid search loop)
  → ***50% Marco, 50% Shadi***
- **part 5:** code for balancing + augmentation(pytorch transformation, and albumentation)
  → ***30% Marco, 30% Shadi, 30% Gabriele, 10% Luca***
- **part 6:** code for focal loss
  → ***95% Shadi, 5% Gabriele***