1 Problem Overview

I aim to develop a binary classifier capable of distinguishing between images of *fields* and *roads*. The available dataset comprises 45 Field images, 108 Road images, and 10 unannotated images designated for testing the trained model. The 10 unannotated images (consisting of 4 Field images and 6 Road images) can be manually annotated without difficulty. One significant challenge we face is the limited size of the proposed dataset, which may not be sufficient for training state-of-the-art classifiers with convolutional layers (such as ResNet, VGG, EfficientNet, etc.). Additionally, there is a data imbalance issue that needs to be addressed to prevent the model from becoming biased toward the majority class.

2 Proposed Approach

In the following section, I outline my strategy for addressing the challenges previously mentioned:

Data Scarcity I applied data augmentation techniques to introduce variability into the dataset, enhancing the model's capacity to generalize to unseen data effectively. I also tried freezing the convolutional layers (use pre-trained models on ImageNet). The underlying idea is to allocate the model's capacity to learn the weights of the dense layers.

Data Imbalance I pursued two methods. First, I applied weighted loss function that assigns higher weights to classification errors associated with the underrepresented class, which in our case is "Field." This technique helps the model give more importance to correctly classifying the minority class. Then, I employed an oversampling technique specifically focused on increasing the number of images from the underrepresented class. This approach contributes to a more balanced training dataset.

3 Experimental Details

Architecture I considered two options: ResNet 18 and MobileNet (a lightweight model).

Learning loss Since there are two classes, I opted for a model with a scalar value for the output rather than R^2 vector. I used binary cross entropy loss (instead of the standard cross entropy loss designed for multi-class problems). A sigmoid function is applied to scale model output between 0 and 1. This provides flexibility in setting the decision boundary between Field and Road samples by adjusting the threshold value.

Hyperparameters Selection I partitioned the annotated data into two subsets: a training set (comprising 80% of the data) and a validation set (comprising the remaining 20%). Importantly, I ensured that both sets maintained the same class ratios as the original dataset. The validation set was utilized to fine-tune hyperparameters and select the optimal configuration, primarily based on the best metric values achieved on this validation set.
I set the number of epochs to 20 and weight decay to 0.001. I tried two optimizers: SGD and

Adam, tuned the learning rate (lr) $\in$ {0.01,0.001} and batch size (BS) $\in$ {8,16}. I followed these steps to reduce the number of tests:

Table 1: First step: tests to select the optimizer and the learning rate

Test ID	Model	Optimizer	lr	BS
0	ResNet18	SGD	0.01	8
1	ResNet18	SGD	0.001	8
2	ResNet18	adam	0.01	8
3	ResNet18	adam	0.001	8

Second step (test 4): select the best configuration in the previous step and vary BS $\in$ {8,16}.
Third step (test 5): select the best configuration in the previous step and check results for MobileNet.

Remark I could have continued tuning hyperparameters such as learning rate or batch size, I recognized that this might be computationally intensive, especially given my limited computational resources (CPU only).

Evaluation Metric To assess the model's performance, I chose the F1-score as my primary evaluation metric instead of accuracy. This decision was driven by the presence of data imbalance, making the F1-score a more robust and informative metric for the binary classification task.

Experimental Results We represent results of the conducted experiments in table 2.

Table 2: F1-score evaluated on validation and testing sets.

Test ID	F1-score (validation set)	F1-score (testing set)
0	0.77	0.73
1	0.77	0.73
2	0.77	0.73
3	0.77	0.73
4	0.75	0.77
5	**0.87**	0.71

The different hyperparameters in tests 1,2,3 and,4 yielded comparable results on validation and testing set. Consequently, I opted to utilize the configuration from test 0 as the basis for test 4, where the batch size (BS) was increased from 8 to 16. During this adjustment, there was a slight decrease in the F1-score on the validation set.
Subsequently, for test 5, I reverted to a batch size of 8 and introduced a change in the model architecture by replacing ResNet18 with MobileNet.
Interestingly, test 5 exhibited improved results on the validation set compared to test 0.

However, it's important to note that there was a larger F1-score difference between the validation and testing sets in test 5. This discrepancy may indicate a potential challenge in generalizing to new, unseen data. Consequently, I have decided to proceed with the model configuration from test 0 for further evaluations.

Throughout these tests, I consistently implemented oversampling, data augmentation, and updated all the weights of the trained model. These choices were driven by the data analysis, which revealed imbalanced data and a restricted dataset for model training. However, it's important to note that due to the limited dataset size, I was unable to conduct a comprehensive study to assess the potential significant improvements brought about by these additional techniques.

On the other hand, I adjusted the threshold to 0.4, which serves as the threshold for distinguishing between negative and positive samples, instead of the conventional threshold of 0.5. Notably, the best F1-score achieved on the validation set aligns with this threshold setting, as shown in table 3.

Table 3: Evaluation of the selected model in test 0 using different thresholds

Threshold	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65
F1-score (Validation set)	0.71	0.67	0.73	**0.77**	0.6	0.48	0.45	0.36	0.3
F1-score (Testing set)	0.67	0.67	0.77	0.73	0.6	0.25	0	0	0