AML challege: Aerial Imagery

Giuseppe Antonio Orlando
Politecnico di Torino
EURECOM
orlando@eurecom.fr

Simone Varriale
Politecnico di Torino
EURECOM
varriale@eurecom.fr

Gabriele Sanmartino
Politecnico di Torino
EURECOM
sanmarti@eurecom.fr

I. Introduction

The goal of this project is the identification of a specific type of cactus in aerial imagery; the dataset contains a large number of 32 x 32 thumbnail images containing aerial photos of a columnar cactus (Neobuxbaumia tetetzo). This kind of task is useful to asses the impact of climate change on Earth's flora and founa and it is useful to quantify how human activities are impacting protected natural aerias. For this reason, it is important to be able to recognize the vegetation inside the protected areas.

II. Dataset

The provided dataset for this task consists of labeled train images and unlabeled test images. Some examples of these images are shown in Figure 1. During model selection and evaluation, only the labeled dataset will be used; the unlabeled one will be adoperated in Section VI. The first analysis performed on the dataset is the one related to classes imbalance, which lead to the discovery of an uneven number of samples for the two classes. The training data is composed of 75% (13136) of images depicting a cactus, the true class for the task. Methods to address this imbalance will be covered in section III.

Fig. 1: Examples of images in the dataset.

III. Data pre processing

To enhance performance, various methods of data preprocessing are exploited in this section to address the problem of class imbalance, to scale pixel values and to define a new way to utilize the pixels of our images. In order to assess the effectiveness of these techniques the following simple deep convolutional model has been used.

Layer Name	Output Size
Conv2d	$16X32X32$
MaxPool2d	$16X16X16$
Conv2d	$32X16X16$
MaxPool2d	$32X8X8$
Linear	128
Linear	2

TABLE I: Simple model architecture

A. Oversampling and Undersampling

The initial observed issue was the class imbalance within our dataset: 13136 images depict a cactus, the true class for the task, while only 4364 do not depict a cactus. To deal with this issue, two techniques are explored.

1) Oversampling: this technique involves increasing the number of instances in the minority class by either duplicating existing instances or generating synthetic data points. By doing so, the number of images in the false class has been increased gradually, to finally match the number of samples in the majority class. This procedure can help improve its ability to correctly classify those instances. In order to increase the number of samples, some data augmentation techniques as random flips and random rotations are applied.

2) Undersampling: in contrast, undersampling involves reducing the number of instances in the majority class to balance the class distribution. This is typically done by randomly removing instances from the majority class until the desired balance is achieved. Undersampling helps prevent the model from being biased towards the majority class and encourages it to learn from the minority class as well.

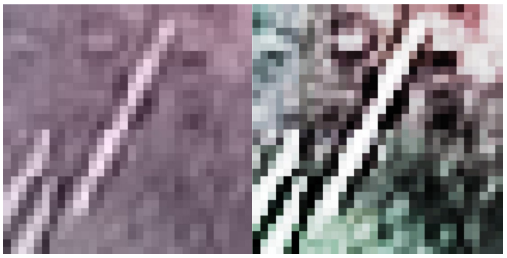
B. Normalization

Normalization, as a preprocessing technique in the context of image data, refers to scaling the pixel values

of an image to a standard range. The purpose of normalization is to ensure that each pixel value falls within a consistent range, this is done by scaling the pixel values to a common range or by subtracting the mean and dividing by the standard deviation. This will help the convergence of training algorithms, prevent vanishing or exploding gradients, and make the optimization process more stable. Additionally, normalization can sometimes enhance the interpretability of the learned features by ensuring that all features are treated equally during training. In this task, normalization is applied per channel given the multichannel nature of the RGB data.

C. Histogram Equalization

Histogram equalization has been chosen alongside the previous techniques since it enhances the contrast of an image by adjusting the intensity distribution of its pixels, spreading out these values so that the full dynamic range of pixel values is utilized. Histogram equalization is particularly effective for images with low contrast or uneven lighting conditions as it can help reveal details in both dark and bright areas of the image. In Figure 3 there is an example of what this technique does to the distribution of the pixel values.



(a) Original Image (b) Equalized Image

Fig. 2: Effect of histograms equalization

Figure 2 shows instead the changes on the image after equalization has been applied.

D. Results

Table II reports the results, using different metrics, obtained for each different pre-processing technique. Instead of determining the goodness of a data preprocessing technique only looking at the model accuracy, oher metrics, like AUC and F1-Score, are studied in order to actually understand the ability of our model to separate the two classes. The best results are achieved with Normalization and Histogram Equalization both reaching almost perfect results. Oversampling technique does not help the classification task, instead it degrades it when trying to match the samples from the majority class. After training on the augmented data from the minory class, the model then fails to generalize on validation data, showing symptoms of overfitting. Undersampling, instead, will get decent performance, but not as good as the other techniques, probably due to the lower number of samples on which the network is trained. Due to this reason, it will be considered alongside the best methods to see what happens with a pretrained model that needs less training data. The above techniques will be used during model evaluation phase to understand the performances of the candidate model.

IV. Model Selection

ResNet18 [1] has been picked as the backbone portion of our final model due to its effectivness in feature extraction and its stability thanks to residual connections.

On top of that, its good optimization properties and small number of paramters when compared to bigger models (e.g. VGG) have helped its choice. The architecture of the backbone is shown in Figure 4

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Fig. 4: ResNet18 architecture

The weights of a version of the network, trained on ImageNet1K, have been loaded considering the extensive variety of ”natural” classes within that dataset.

Different heads will be attached to the backbone in order to check which one will be best suited for the task and finally the performances of the model will be evaluated.

As a starting point, only one fully connected layer is attached to the backbone without additional complexity. In subsequent steps, more sophisticated approaches are explored by incrementally adding fully connected layers and techniques, like dropout regularization and maximum pooling, to enhance the model’s robustness and performance. The best model is chosen considering the best AUC value scored on the validation dataset; the

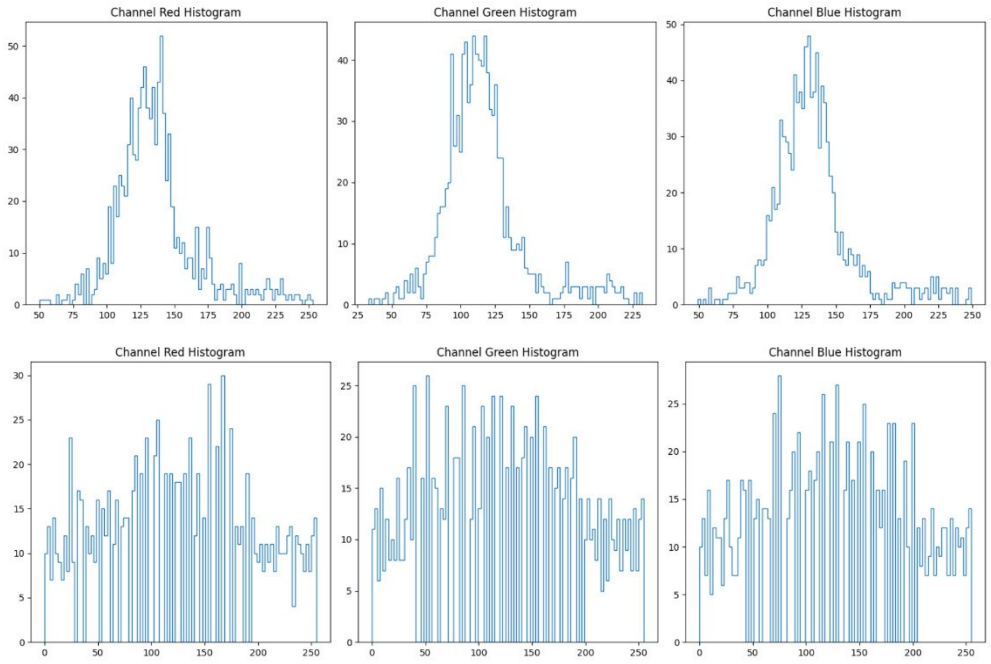


Fig. 3: Examples of histograms

Technique	Accuracy	Precision	Recall	F1Score	AUC
No technique	0.9851	0.9949	0.9850	0.9899	0.9973
Normalization	0.9894	**0.9973**	0.9886	0.9929	**0.9997**
Histogram Eq.	**0.9900**	0.9912	**0.9954**	**0.9933**	0.9985
Undersampling	0.9760	0.9877	0.9801	0.9839	0.9962
Oversampling	0.7483	0.7483	1.0000	0.8560	0.5000
Normalization + Histogram Eq.	0.9871	0.9962	0.9867	0.9914	0.9990
Undersampling + Normalization	0.9837	0.9946	0.9836	0.9891	0.9984
Undersampling + Normalization + Histogram Eq.	0.9880	0.9965	0.9874	0.9919	0.9986

TABLE II: Performances with pre-processing techniques

usage of the AUC metric is justified by its ability to describe how well a model is able to discriminate the two classes, given the present imbalance. The loss used for training is the cross entropy loss.

Model	Accuracy	Precision	Recall	F1Score	AUC
1-FC	0.9857	0.9822	**0.9992**	0.9906	0.9998
2-FC	**0.9951**	**0.9958**	0.9977	**0.9968**	**0.9998**
3-FC	0.9946	0.9943	0.9985	0.9964	0.9996

TABLE III: Performances of ResNet18 without pre-processing

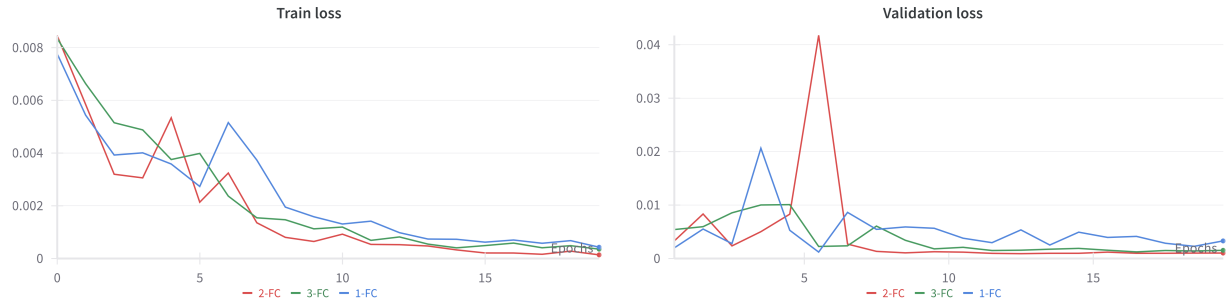
Although the results are close to perfect for all the heads, the model with two fully connected layers seems to be the best performing option. This architecture, as a consequence, will be chosen as the final one. Figure 5 depicts plots of the validation and training loss of the model with the different heads during the training phase.

After having chosen the best version of the model, a comparison between different pre-processing techniques has been made, utilizing the previously picked architecture. As shown in Table IV the best results are achieved with the Normalization technique.

V. Model Performance

From Section IV, the candidate model is ResNet with two fully connected layer applying Normalization pre-processing technique. The performances are now assessed on the test dataset. To mitigate the risk of obtaining results based on chance, multiple seeds are utilized in this phase. This ensures the model is tested on various test sets, allowing for the construction of a confidence interval for the scores obtained. The deviation of the values is evaluated to understand how stable our model is. The results of this analysis are reported in Figure 6.

The biggest deviation can be observed on precision metrics, while the AUC metric has the smallest one. Overall, the results on the test set exhibit a slight decrease when compared to the results obtained on the validation set, but they remain close to one for all



(a) Training loss

(b) Validation loss

Fig. 5: Training and Validation loss

Technique	Accuracy	Precision	Recall	F1Score	AUC
No technique	0.9937	0.9948	0.9970	0.9959	0.9998
Normalization	**0.9986**	0.9991	**0.9990**	**0.9991**	**0.9999**
Histogram Eq.	0.9909	0.9948	0.9933	0.9940	0.9991
Undersampling	0.9966	**0.9993**	0.9963	0.9978	0.9998

TABLE IV: Performances of ResNet with 2-FC

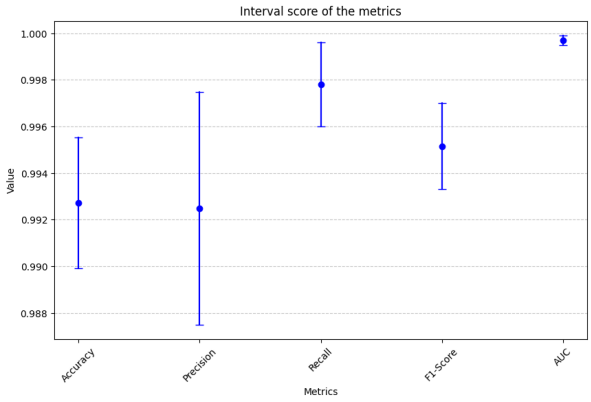


Fig. 6: Interval score for the best model

the metrics. This shows that the model has generalized well.

VI. Experiments

This section is dedicated to labeling the test samples, for which true labels are not provided, in order for them to be used during training time to try to improve model performances. These samples will be referred to as test samples. To label the test dataset, feature extraction techniques are exploited and the same models used throughout the analysis are employed. The fully connected layers at the end have been removed in order to operate exclusively using the convolutional backbones. The goal is to extract all the features from the image, which were useful for classification in the previous task, then evaluate the mean of each feature for the two classes, in order to obtain an averaged feature summary of them. To classify the test samples, the features are extracted and the euclidean distances between the data and the two means are computed. The label of the "closest" class has then been assigned.

A. Evaluation of the mean

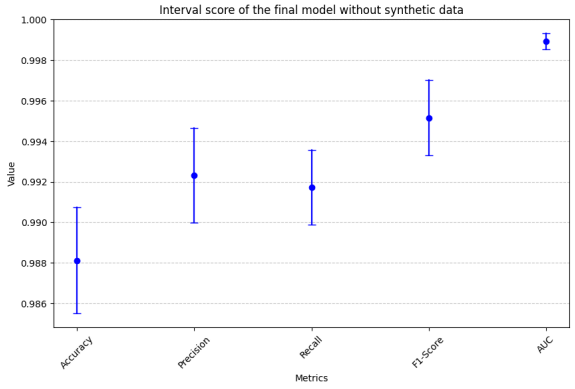
To find the feature means of each class, two feature extraction backbones are used:

- ResNet18
- Custom model showed in Table I

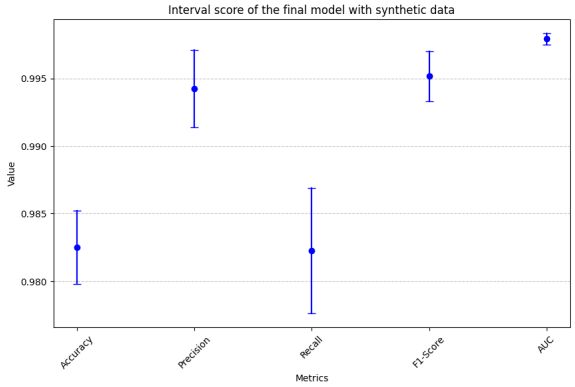
ResNet18 has been used in its pretrained version, while the second model has been retrained on a bigger portion of the labeled set leaving just 10% out for validation purposes. After evaluating the mean of the feature vector for the two classes, an investigation has been performed to actually understand the goodness of this technique: the distance from the two means has been evaluated for each of the training sample and the corresponding class association has been compared to the true label. Out of the 15750 samples, 2366 were assigned to the wrong class using the custom model. When a wrong classification happened, the absolute differences of the distances from the two means has an average value of 26.664. The latter will be used as a confidence score to assign the correct label: when the difference of the distances is greater than the confidence score the classification "should" be correct. Using ResNet has instead lead to 4205 samples assigned to the wrong class, with a confidence score of 2.702. The better perfomances of the custom model are mostly due to the training performed on the actual images. As a consequence, the custom model has been chosen to continue the analysis.

B. *Classifying test dataset and Retraining*

The classification has been performed taking into account the ”closest” class and the value of the confidence score defined before. Exclusively the samples that scored better than the threshold will be used for the following step. The new data is employed to expand the training set and retrain the model; checks if this retraining has helped to improve performances will be run. Results of the training set boost are showed in Figure 7



(a) No synthetic data



(b) With syntethic data

Fig. 7: Effects of systethic labels injection

As depicted in the above plots, a minor drop in accuracy and recall is present. A decrease in the recall score suggests that the model sometimes neglects the presence of a cactus, since the number of false negatives has increased; this might be due to some wrongly labeled samples in the synthetic data. Nonetheless, the precision score has increased. This means that the model is more accurate when predicting the absence of a cactus, since the number of false postives has decreased. The preference for high recall or high precision depends on the application. Since there are no specific guidelines for the costs of misclassification, the approach can be considered reasonable.

VII. Conclusion

In this aerial imagery challenge various preprocessing techniques and model were explored to find the best solution to improve classification.

The preprocessing techniques included normalization, histogram equalization, oversampling, and undersampling. Normalization and histogram equalization yielded the best results, improving the model's performance while oversampling did not provide significant improvements and led to overfitting issues.

For model selection phase, different fully connected layer compositions were testes and metrics such as accuracy, precision, recall, F1-score, and AUC were used to evaluate the model's performance.

Further experiments were conducted to classify unlabeled test samples and retrain the model using the classified samples, trying to improve the classifier. Unfortunately, the score were similar on average.

Future research could explore other combination of preprocessing methods or exploit some unsupervised algorithms in order to obtain clusters for the unlabeled dataset starting from the extracted features.

References

[1] @mische2015deep, title=Deep Residual Learning for Image Recognition, author=Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, year=2015, eprint=1512.03385, archivePrefix=arXiv, primaryClass=cs.CV