POLITECNICO
MILANO 1863

AN2DL - Second Challenge Report
Team Name: iFormaggini

Federico Pinto, Mattia Gotti, Michele Milani, Stefano Pedretti

federicopinto02, stealthygotti, michelemilani2, stefanopedretti

273427, 271034, 276185, 273086

December 16, 2025

1 Introduction

In this second challenge, we were given a collection of images. Each image comes from a low-magnification WSI of human tissue. The objective of this task is to perform classification — that is, to assign each image to one of four possible classes corresponding to molecular subtypes identifying a potential disease within the tissue. To be evaluated we are expected to submit:

1. `.csv` file with test predictions, over which the F1-score will be evaluated and posted on Kaggle leaderboard
2. This work report

2 Problem Analysis

The dataset contains 1272 images of different sizes, each paired with a binary mask. These masks identify the regions most likely to contain the diseased tissue. Classes to be predicted were 4:

1. `Luminal A`
2. `Luminal B`
3. `HER2(+)`
4. `Triple Negative`

It was written that it was guaranteed that the pictures were taken in an *absolutely sterile environment*. That was a tricky thing because the images were full of Shrek's pictures and boogers :D; Initial assumptions we have made are mainly two:

- We assume that the training set offers sufficient **morphological variability** across the four classes. Due to the nature of histology, rigorous color normalization (e.g., H&E staining standardization) is used to mitigate processing artifacts and improve model generalization. We hypothesize that standardizing the chromatic distribution allows the model to focus on structural cellular features rather than being biased by the specific scanning conditions or staining intensity of different laboratories. The binary masks are auxiliary information to focus the model's attention on disease areas, reducing background noise.

- **Validation and Evaluation** No official split is provided, but we've reserved a portion (20%) of the training data as a *stratified* internal validation set. This stratification is crucial to maintain the same class distribution as the original dataset, preventing the validation metrics from being ruined by potential class imbalances. The model will be primarily optimized for the F1-Score (macro-averaged), as required by the challenge metric.

3 Method

Our final method is defined by three key areas: the model architecture, the data preprocessing pipeline

that supports it and the training strategy.

Model Architecture: We designed a custom Network to process both the RGB image and its corresponding mask simultaneously [2], leveraging the nature of the input data. The network consists of two parallel branches:

- **Image backbone:** A `ConvNeXt tiny` encoder processes the raw RGB images to extract rich visual features.
- **Mask backbone:** A parallel backbone processes the single-channel segmentation masks, allowing the model to explicitly leverage the provided spatial annotations without mixing them directly with color channels in the input level.

The features extracted from both streams are concatenated (feature fusion) and passed to a custom classifier head. This head is composed of a `BatchNorm1d` layer to normalize the fused features, followed by a `Dropout` layer for regularization. The features then pass through a linear layer activated by `ReLU`, a second `Dropout` layer and a final linear layer mapping to the 4 output classes.

Data Preprocessing: All images and masks are resized to a fixed resolution of `384x384` pixels (it's the best trade of between resolution and efficiency that we have found). The RGB images are normalized using the standard ImageNet mean and standard deviation. To improve generalization and robustness to variations, we applied a strong data augmentation pipeline:

- **Geometric Transformations:**
  we applied random *horizontal flip* and *vertical flip*, as well as random 90-degree rotations.
- **Color Augmentation:**
  we utilized `ColorJitter` applied with a probability of 0.8 to the images to reduce sensitivity to lighting conditions.

Training and Regularization Strategy: To have a balanced representation of classes across validation sets and to reliably estimate model performance, we implemented a **Stratified K-Fold** cross-validation strategy[1]. The training process utilized the following techniques:

- **Loss Function:** we used `CrossEntropyLoss`[1] combined with computed `class_weights` to address the significant class imbalance. The last class, `triple negative` was pretty rare. Additionally, we applied **label smoothing** overconfidence in predictions.
- **Optimizer & Scheduler:** we employed[1] the **Lion** optimizer[2] because we found out it has better convergence speed and generalization capabilities compared to standard AdamW. A `CosineAnnealingWarmRestarts` scheduler was used to adaptively adjust the learning rate during training.
- **Regularization:** beyond standard Dropout, we implemented **CutMix** augmentation[2]. This technique cuts and pastes patches among training images and mixes their labels, effectively regularizing the model and encouraging it to focus on less discriminative parts of the image.

4 Experiments

Our experimental process was iterative. We began by establishing baseline performance with simpler architectures. Each model was then tested to identify key weaknesses, like rapid overfitting, underfitting and poor coherence between kaggle and notebook F1 score. Our development consisted of a systematically fix of these weaknesses. We experimented with different architectural models and different hyperparameter to see which model perform better for our problem. Our best results are summarized in Table 1.

5 Results

The relative performance of the F1-score with and the relative Kaggle submission evaluation are shown in the table. The models we've found more interesting are in **bold** style.
The main unexpected outcomes across the experiments we have done are mainly two:

1. The dimension of a model not always implies a better performance. An example is the difference between F1 score in `ConvNeXt`

[1] "Employed"

Table 1: Log of notable experiments. Evolution of Notebook (NB) and Kaggle Leaderboard (LB) scores.

Model/Technique	Configuration Details	NB F1	LB F1
ConvNeXt Tiny	Res. 224 (Distorted), Base Aug, Class Wgt	0.2937	0.3245
ConvNeXt Tiny+Tiling	Res. 512 → Crop 224, Bal. Sampler	0.2493	0.2192
ConvNeXt Tiny+High-Res	Res. 512, SqPad, Grad Accum (BS=4)	0.3030	0.2316
ConvNeXt Tiny+Masks	Res. 224 (Reflected), SqPad, BS 32, No Warmup	0.3266	0.2592
ConvNeXt Tiny+ResNet	Dynamic Masks in train/test	0.3843	0.3007
ConvNeXt Tiny+ResNet	**Still with Lion, Weighted Loss**	**0.3721**	**0.3649**
ConvNeXt Small+ResNet	With CutMix	0.3640	0.3569
ConvNeXt Tiny+ResNet	**With CutMix**	**0.3578**	**0.3652**
Best Folds only	Inference on fold 2 only	0.4112	0.3206
Ensemble on 3 best Folds	Not weighted	0.3912	0.3796
Weighted Ensemble	**Same 3 folds, weighted**	”	**0.3978**

`Tiny+ResNet` and `ConvNeXt Small+ResNet` in the table. We expected to gain a lot in the F1 score but, in reality, the model with too many parameters is too much. By having too much weights and a small training data it learns easily the data causing a rapid *overfitting*.

2. The second unexpected result is the improvement in the F1 score when only some folds are selected for generating predictions and also when the ensemble of the three best folds is *weighted*, compared to an *unweighted* approach. These experiments have been done on the same model and folds. The reason could be that in some folds models carry an higher noise, by correctly weighting them we can obtain a greater balance.

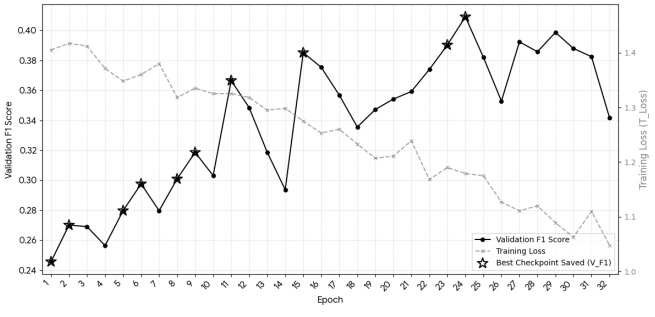

Figure 1: Val. F1-Score and Train Loss oscillation example

6 Discussion

Strengths and effective strategies are multiple in these experiments. The most significant performance gains in single models came from **regularization** and optimization techniques rather than purely architectural changes. As shown in the table, they have successfully narrowed the generalization gap. The inclusion of *CutMix*[2] proved to be a robust strategy for **generalization**. The ensemble confirms that the errors across different folds were sufficiently uncorrelated, allowing the ensemble to smooth out predictions and boost the F1 score significantly. The worst weakness is that the initial introduction of `masks` and `ResNet` components (without strong regularization) led to severe overfitting. Contrary to the intuition that higher resolution yields better feature extraction, the `Tiny+Tiling` and `Tiny+High-Res` experiments performed poorly compared to the baseline.

7 Conclusion

In this challenge, we developed a classification pipeline to categorize histological images into four molecular subtypes. Our work addressed the constraints of a small, imbalanced dataset. Our main contributions include:

- Dual branch architecture
- Advanced optimization and regularization
- Strategic ensembling

Our experiments highlighted limitations that offer directions for future improvement. Future work should focus on highly efficient, lightweight architectures or *transfer learning* from domain-specific medical datasets to handle the **data scarcity** better. Future research could also investigate other multi-scale approaches to capture fine-grained cellular details without introducing noise or losing context.

References

[1] Eugenio Lomurno. *AN2DL exercice notebook*.

[2] Eugenio Lomurno. *LogBook challenge AN2DL*.