1. Provide your team background and organization description (if applicable).

I am an employee at VitaDX, a start-up based in Paris specializing in cytological analysis for bladder cancer diagnosis. At VitaDX, we analyze whole slide images of voided urine cytology to determine whether a patient has cancer, distinguishing between high-grade and low-grade cases.

I completed this challenge on behalf of VitaDX, working on it individually while benefiting from input and support from my colleagues. My background includes a PhD in computer science from the University of Technology of Troyes. My two colleagues who provided assistance during this challenge are also PhD holders in applied mathematics. Together, we form the team responsible for the artificial intelligence and computer vision at VitaDX.

2. Explain why you participated in the Cytologia challenge.

We discovered the Cytologia challenge through the Health Data Hub website, and it immediately caught our attention. As a company specializing in cytology, VitaDX found this challenge highly relevant to our expertise and objectives. The dataset stood out due to its quantity, quality, and variety, which surpassed what is typically available in public datasets.

In particular, the large number of classes and the detection aspect presented a unique challenge that aligns with real-world hematological analysis. It provided an opportunity to apply our expertise in cytological analysis to a new domain, strengthening our capabilities in detecting and classifying complex patterns in medical data.

By participating, we aimed to benchmark our methods against a challenging dataset, refine our AI approaches, and gain valuable insights applicable to our work at VitaDX. This challenge also allowed us to explore new methodologies that could have a broader impact on medical imaging.

3. Describe how you built your winning model and elaborate on the technical and modeling choices you made.

When building the model, we focused on three key considerations: maximizing performance (IoU and F1-score), minimizing inference time and maintaining simplicity in the model design. Initially, we planned to use a two-step approach: a small YOLO detector to localize bounding boxes of white blood cells, followed by cropping the detected regions for classification. However, the results were mixed, and we ultimately opted for a single YOLO model to handle both detection and classification of the 23 classes. This decision simplified the pipeline and reduced inference time.

Rather than relying on a brute-force approach with extensive trial and error across a large catalog of models and an exhaustive hyperparameter grid to maximize performance metrics on this specific dataset, we prioritized simplicity and ease of analysis. We believe our approach is more practical and focused on solving real-world problems by emphasizing explainability and generalizability, rather than solely aiming to win the challenge.

Datasets

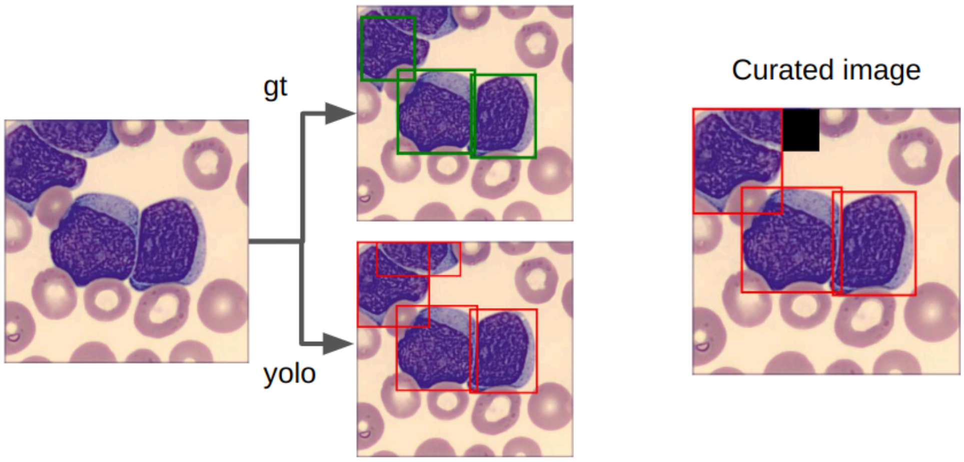
To improve performance, we adopted a data-driven approach. We trained an initial YOLO model to curate the dataset and address ambiguous ground truth annotations. After testing different data curation techniques, we generated two specialized datasets:

1. **Dataset 1**: Ground truth boxes with low IoU scores relative to the predictions were masked.
2. **Dataset 2**: These same ground truth boxes were retained.

In both datasets, YOLO detections with high confidence but no matching ground truth boxes were masked to minimize the impact of misannotated data.

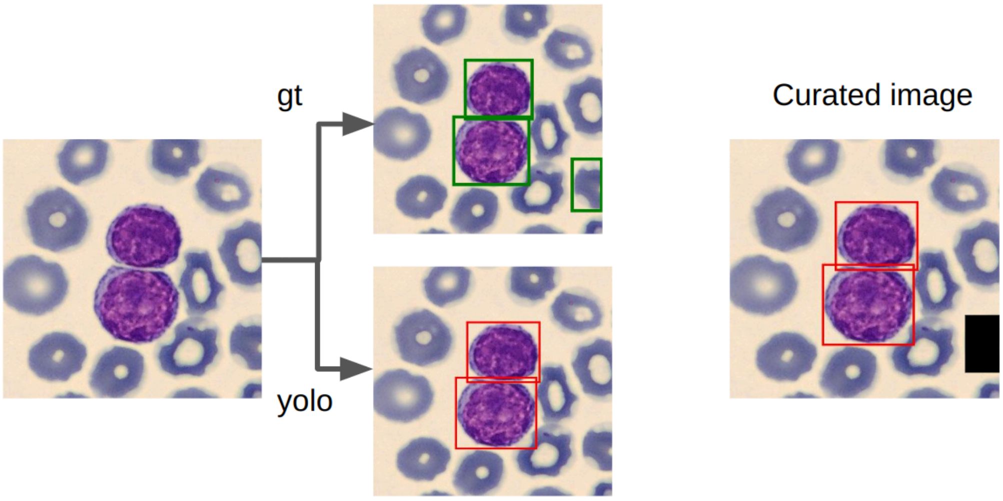
The curation we applied on the 2 datasets :

WBC in the border is not annotated, boxes are more precise with yolo



The curation we applied only on the first dataset :

A red blood cell in the background is annotated as a white blood cell



Training and inference

During training, we used the automatic optimizer from Ultralytics with a batch size of 64 for 250 epochs. Classical data augmentation techniques such as random flips, rotations, and HSV adjustments along with mosaic data augmentation were applied to improve generalization.

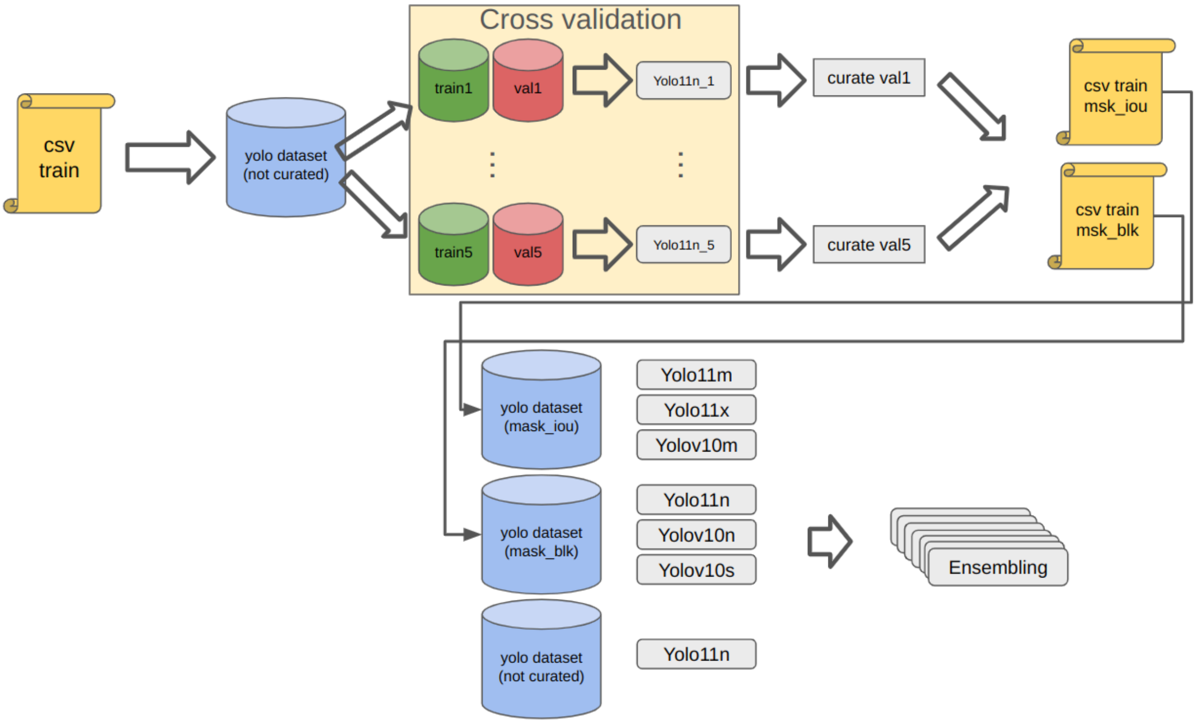
For the final submission, we employed an ensemble of 7 models to maximize performance. To ensure diversity:

1. **Dataset Diversity**: 3 models were trained on Dataset 1, 3 on Dataset 2, and 1 on the uncurated dataset.
2. **Model Architecture**: The ensemble included YOLO11 and YOLO10 models of varying sizes (e.g., YOLO11n, YOLO11m) to introduce architectural variability.

We combined the outputs using a custom **weighted box fusion algorithm**. The ensemble achieved a combined score of 0.9385 on public learderboard, outperforming individual models.

In a challenge setting, ensembling—combining predictions from multiple models—is often essential for competitive performance, which is why we employed it. This strategy typically enhances performance by leveraging the strengths of different models. However, instead of extensively exploring a wide range of models and combinations, which can reduce explainability and increase inference time, we chose to focus on data curation and optimizing model training. While this approach may have resulted in a slight loss in performance compared to the best possible combination of numerous models, we believe it is more impactful and practical for real-world applications.

This figure summarizes the training process :



<u>Inference time</u>

Despite using 7 models, inference remained efficient at **70 ms per image on an RTX 4060 laptop GPU**, thanks to optimized implementations and batch processing. Interestingly, a single YOLO11n model achieved close performance (0.9320 combined score) while being **10 times faster (7ms)**, demonstrating its practicality for real-world applications. YOLO11n can also be easily run on a CPU, with an inference time of approximately **30 ms** on an **Intel® Core™ Ultra 7 155H** processor.

For detailed technical steps, our methodology is fully documented in the **Readme.md** of our GitHub repository, and the Python scripts are thoroughly commented.

4. What GPU/CPU/RAM resources you used to build your model

To build our model, we used two different machines with the following specifications:

1. **Laptop Configuration**:
    - GPU: NVIDIA RTX 4060 (8GB VRAM)
    - CPU: Intel(R) Core(TM) Ultra 7 155H
    - RAM: 32GB
2. **Desktop Configuration**:
    - GPU: NVIDIA RTX 4080 (16GB VRAM)
    - CPU: AMD Ryzen 7 7700X (8-core processor)
    - RAM: 64GB

We primarily used the desktop with the RTX 4080 for training due to its larger VRAM, which allowed us to process bigger batch sizes and achieve faster training speeds. The laptop was used for smaller-scale experiments, debugging, and inference testing to ensure portability and efficiency.

5. **Do you have any positive feedback or improvement opportunities for the Trustii.io platform?**

Positive Feedback

1. **Evaluation Code Availability**: Providing the code used for evaluation was helpful, and such transparency is highly appreciated.
2. **Score Transparency**: While the evaluation was provided, having a breakdown of the scores (IoU and F1-score separately) would have been even more insightful. This would enable participants to track their progress on both localization and classification aspects individually.
3. **Reactivity and Support**: The responsiveness of the Trustii team to participant queries was remarkable. It fostered a sense of engagement and trust throughout the challenge.
4. **Clear and Accessible Challenge Design**: The data page was well-organized, and the overall challenge was structured to be accessible, catering to participants with varying levels of expertise in data science.

Suggestions for Improvement

1. **Forum Features**: It would be helpful if the forum displayed the date of the last message posted alongside the creation date. This would allow participants to quickly see if there are unread updates without opening each post.
2. **Platform Speed**: The platform felt somewhat slow when navigating between pages. While not a major issue, smoother navigation would enhance the overall user experience.

3. **Private Leaderboard Data Access**: Having access to the images used for scoring on the private leaderboard could, either intentionally or unintentionally, lead to overfitting or misuse. Restricting this access might ensure a more robust evaluation process.
4. **Real-World Task Representation**: For a detection task, revealing the number of cells present in each image seems unrealistic and may inadvertently guide models. In practical scenarios, this information isn't typically available, so excluding it could make the challenge more aligned with real-world conditions.