Fachhochschule Nordwestschweiz

Where is Santa?

Mini-Challenge DLBS: Object detection with YOLOv8

Autumn Semester 2025/26

Katarina Fatur and Sofia Alcaide Agúndez

Table of contents

1. Assignment

1.1. Problem statement

For generations, children around the world have tried to catch a glimpse of Santa Claus on Christmas Eve. Nowadays, with cameras in every pocket, it seems easier than ever. However, in practice, picking out the real Santa is not trivial: a red coat and a white beard might belong to Santa, Grinch in disguise, Gandalf on holiday, or just someone in an ugly Christmas sweater.

In this mini-challenge an object detection problem is studied: automatically localising Santa Claus in images using deep learning based detectors for Christmas-themed media. The detector must work on mixed-style data and remain robust despite intra-class variation.

The following research questions will be explored:

1. How robust is the detector to intra-class variability (poses, occlusions, real vs cartoon Santas) and to domain shift and inter-class similarity (other red/white objects)?
2. How to regularize and tune the hyperparameters to improve performance under these data constraints?
3. We want to catch as many Santas as possible: how to optimize for high recall without sacrificing precision?

Hypotheses:

H1: *On an unseen Santa test set, a fine-tuned YOLOv8 model achieves at least recall ≥ 0.7 for IoU ≥ 0.5.*
H2: *The main failures come from intra-class variability, domain shift, and confusion with visually similar non-Santa objects.*

1.2. Data

A Santa dataset was created by combining several publicly available “Santa Claus” projects from Roboflow, some images from the internet and even some pictures taken by us. The resulting dataset contains 950 images, 700 for training (73.7%), 155 for validation (16.3%) and 95 for test (10.0%). We have a single target class, *Santa*, annotated with bounding boxes. The images cover a range of real photographs, cartoons, drawings, featuring Santa, Grinch in disguise, and also Santa-lookalikes. The training dataset includes a substantial fraction of background-only images.
The data are split into separate training, validation and test sets without image overlap, but the total number of images is still quite small, so preventing overfitting and good generalisation are important aspects to address.

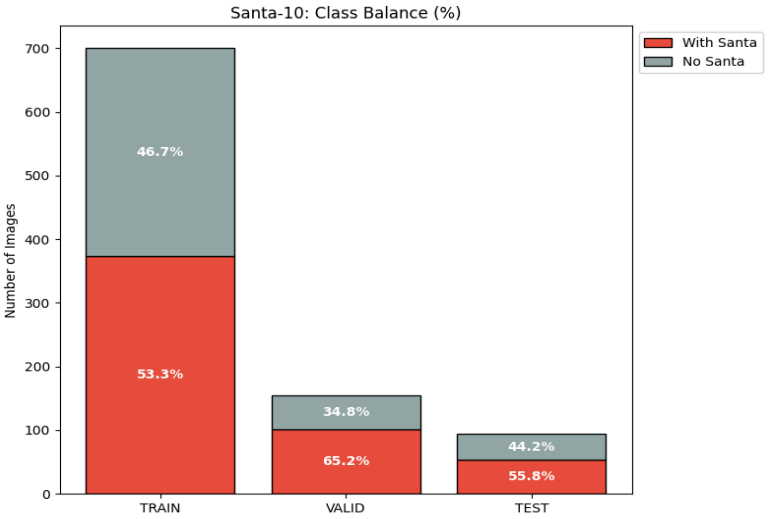

Figure 1: Dataset split distribution

1.3. Methods

The approach of this project is based on a YOLOv8 model. We experimented with nano and small YOLOv8 versions (~3M and ~11M respectively). is a compact one-stage detector that trains efficiently in Google Colab while still offering enough capacity for this task, which is important given the relatively small size of the dataset.

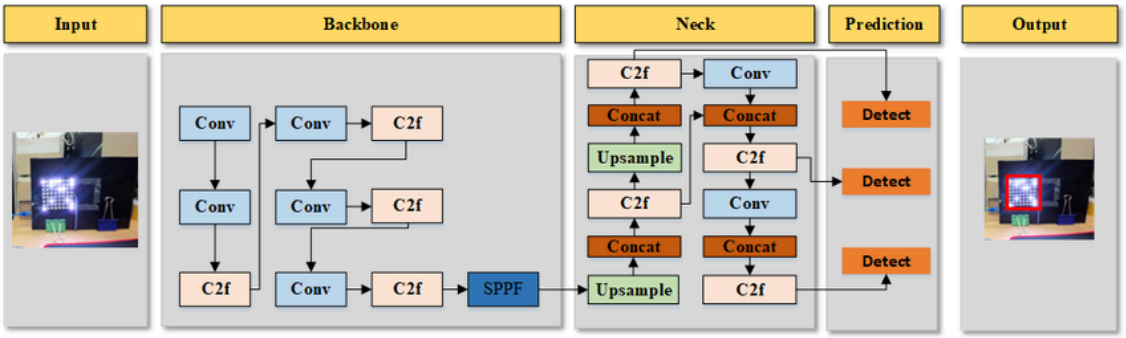

Figure 2: Object Detection with YOLO, Source

All images and labels exported from Roboflow are converted to the YOLO format and organized in the required folder structure. Bounding box annotations are cleaned so that only the Santa class remains. The dataset is then split into training, validation and test sets without image overlap. The test set is kept completely separate and is only used once at the end of the project to obtain an unbiased estimate of the final performance.

As a starting point, a baseline YOLOv8 configuration is trained with settings close to the Ultralytics defaults and fixed input resolution. During training, the evolution of the loss and several detection metrics are monitored. Because the task is single-class object detection, IoU ≥ 0.5 and mAP at 0.5 are used as main indicators of localisation quality, complemented by precision and recall. Recall is considered especially important, since missing a Santa instance is more problematic for the use case than producing a small number of false positives.

After establishing the baseline, a series of controlled experiments is carried out. Three overfitting configurations are defined to systematically reduce regularisation and make it easier for the model to memorise the training data. Run 1 uses the full training set with many epochs, relatively large batches, no data augmentation, no early stopping and no explicit regularization such as weight decay or dropout, with all layers of YOLOv8 kept trainable. Run 2, follows a similar idea but further increases the batch size and freezes the early backbone layers, so that mainly the later layers and detection head are adapted. Finally, Run 3 trains on a smaller subset of the data and freezes an even larger part of the backbone, focusing the learning on the final layers. Together these three settings explore different ways of encouraging overfitting: by changing the effective dataset size, by altering the amount of trainable capacity and by removing regularization components of the training procedure.

In a subsequent regularization phase, the focus shifts from fitting the training set to improving generalisation on the validations set. Stronger data augmentation is introduced, early stopping based on the validation loss is activated and different values for earning rate and batch size are explored. A final tuning step investigates small variations around the most promising configuration. The model that shows the most stable and best behaviour on the validation set is selected as the final Santa detector and is then evaluated once on the test set.

	epochs	batch	lr0	weight decay	augmentation	early stop / patience	freeze
Baseline 1	200	84	0.01	0.0	False	no	none
Baseline 2	250	32	0.05	0.0	False	no	19
Overfit 3	250	15	0.01	0.0	False	no	9
Overfit 4	250	15	0.01	0.0	False	no	20
Regularized 1	180	16	0.01	0.005	True	20	none
Regularized 2	150	16	0.001	0.008	True	25	none
Regularized 3	180	16	0.01	0.00025	True	20	none
Tuning	50	16	0.0001	0.0005	False	10	none

Table 1: Training parameters

Table 1 summarizes the exact training parameters used for the baselines, the overfitting runs, and the regularization and tuning runs. This makes the experiments directly reproducible and clarifies what “smaller batches” and “long training” mean quantitatively.

2. Results

This section summarises the main quantitative findings of the Santa detection experiments. The project focuses on single-class object detection of Santa Claus using a pretrained YOLOv8 model

fine tuned on a small custom dataset of around 950 images taken mainly from several public Roboflow projects and from the internet, as explained in the previous section. The data are split into training, validation and test sets without overlap, and only the Santa class is kept in the annotations. The chosen metrics are IoU ≥ 0.5, mAP50, mAP50-95, precision and recall, with a particular emphasis on recall because missing a Santa instance is more critical than a few false positives.

A baseline YOLOv8 configuration is first trained to obtain a reference level of performance. Then, several overfitting experiments with reduced regularization are run to show that the model can fit the training data, which also leads to a visible gap between training and validation performance. In a second phase, different regularization strategies and small hyperparameter changes are explored, including stronger data augmentation, early stopping based on the validation loss and adjusted batch sizes and learning rates. The final regularised and tuned configuration clearly improves over the baseline in terms of recall and mAP50 on both validation and test sets, while keeping precision at a reasonable level. The most relevant numerical results and example detections are presented in Section 2.1.

2.1. Most relevant results

The main results are focused on three key configurations: the baseline model, an overfitting setup and the final regularised and tuned model. Also, a set of example images that illustrate typical successes and failures is included. Together, these results address the research questions by showing how well YOLOv8 performs on the Santa dataset and how the training choices influence recall, precision and localisation quality.

Baseline performance

The baseline 1 configuration provides a first reference for the Santa detection task. Table 2 summarizes the training outcomes of the baseline. The model achieves a strong best mAP50 of 0.903 and best mAP50-95 of 0.641 (both at epoch 176), indicating that even without further optimization the detector already learns a fairly robust notion of the Santa class. The final checkpoint at 200 epochs reaches a mAP50 of 0.886 and mAP50-95 of 0.651, suggesting that performance largely plateaus after 175 to 180 epochs, with only marginal changes after. Precision peaks at 0.916 (epoch 175), meaning that when the baseline predicts a Santa it is usually correct. While the summary reports a “best recall” of 1.000 at epoch 0, this value is likely not representative of the baseline’s overall behavior (and can be unstable depending on early-epoch evaluation), so we primarily use the mAP and precision trends as the most reliable indicators of baseline performance. Overall, this baseline provides a strong reference point, but it still motivates later regularization/tuning to improve robustness and the precision–recall trade-off for the intended use case.

Metric	Value	Epoch
Best mAP50	0.903	176
Best mAP50-95	0.641	176
Final mAP50	0.886	200
Final mAP50-95	0.651	200
Best Precision	0.916	175
Best Recall	1.000	0

Table 2: Baseline results for epoch 16

Overfitting behaviour

To verify that the model has enough capacity to fit the dataset, we trained an intentionally under-regularized configuration (overfit_run_4). Figure 2 shows a clear generalization gap: while the training loss decreases steadily with more epochs, the validation loss remains substantially higher and does not follow the same downward trend. This indicates that continued training mainly improves fit to the training data rather than learning features that generalize.

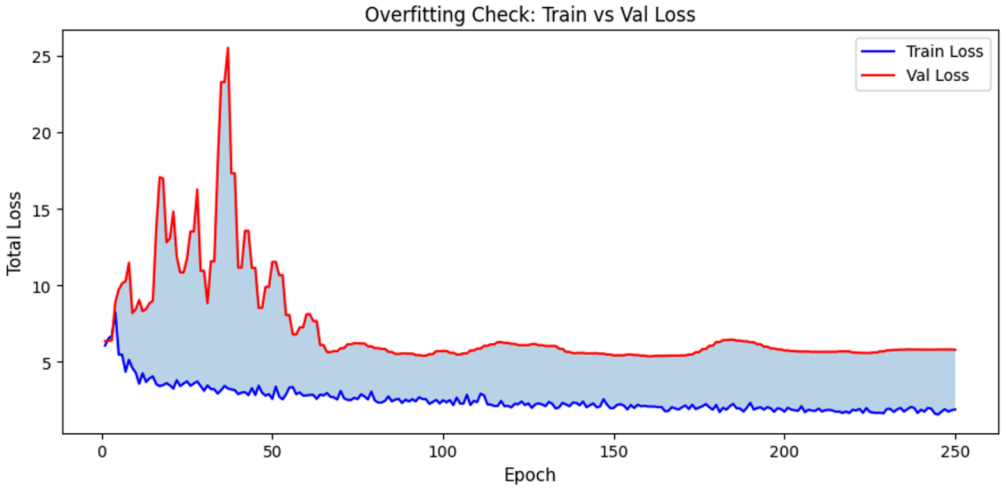

Figure 3: Train vs Val Loss Overfitting Run 4

Importantly, the validation detection scores shown in Figure 3 confirm this: mAP50 and mAP50-95 oscillate and plateau instead of improving consistently (best mAP50 = 0.644 and best mAP50-95 = 0.362 at epoch 66; final mAP50 = 0.572 and final mAP50-95 = 0.328 at epoch 250).

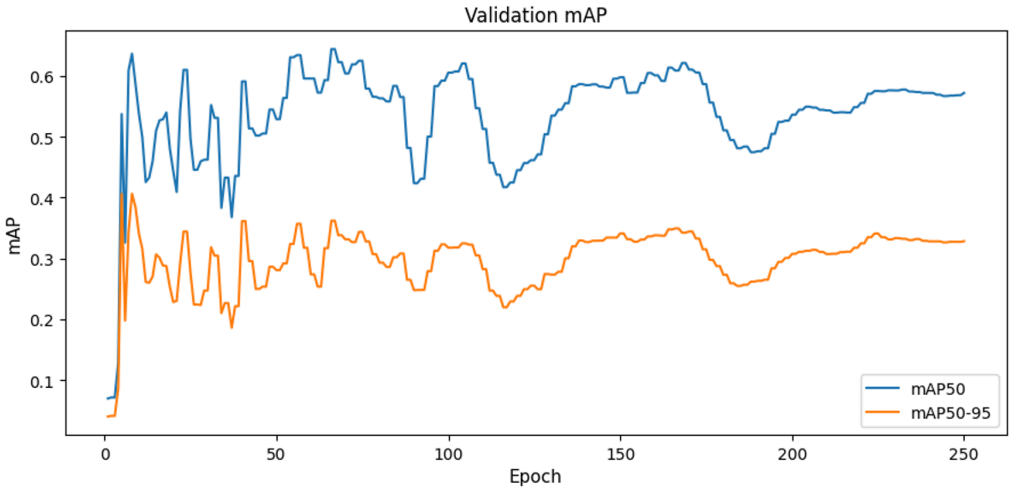

Figure 4: Validation mAP Overfitting Run 4

Overall, this experiment demonstrates that the task is prone to overfitting under reduced regularization and motivates the next section, where augmentation and early stopping are applied to improve generalization.

Regularized and tuned model

After verifying that the detector can easily overfit when regularization is minimized, we trained three regularized configurations using data augmentation and early stopping, while keeping the backbone pretrained. Here, we show the two most representative regularization runs. Figure 4 shows the results of the regularized run 1, where validation performance improves compared to the overfitting setting but remains relatively unstable: the best validation mAP50 reaches 0.805 and mAP50-95 reaches 0.470 at epoch 106, while the final checkpoint drops to mAp50 = 0.589 and mAP50-95 = 0.320, suggesting that training beyond the peak does not reliably improve generalization. Consistently with this, the best precision reaches 0.798 (epoch 110) and the best recall goes up to 0.829 (epoch 97), indicating progress but yet still clearly below what we achieve with stronger regularization in run 2.

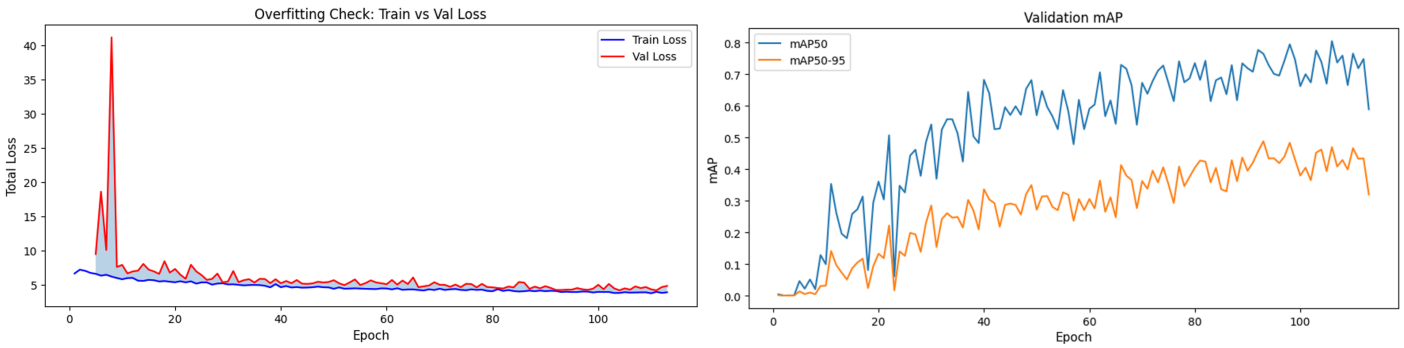

Figure 5: Regularized Run 1 Results

In contrast, Figure 5 shows the desired effect of regularization: training and validation losses follow a more consistent trend and the generalization gap is reduced. More importantly, the validation scores stabilize at high level, with best **mAP50 = 0.899** and **mAP50-95 = 0.683** at epoch 56.

Precision and recall are both high and well balanced (best **precision = 0.901**, best **recall = 0.924**), which is particularly important for our use case where missing Santa instances is more costly than producing a small number of extra detections.

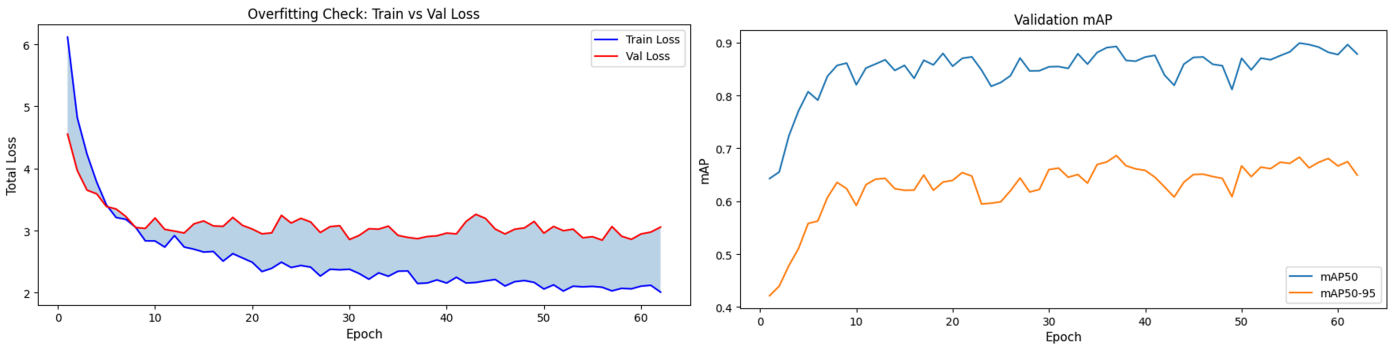

Figure 6: Regularized Run 2 Results

Overall, the regularized run 2 clearly outperforms run 1 across all key validation metrics (mAP50, mAP50-95, precision and recall) and shows more stable validation behaviour. We also evaluated an additional regularized configuration (Regularized run 3 in Table 1), which did not perform better than run 2.

Finally, we performed an additional tuning attempt, by adjusting weight decay and increasing the box loss weight. This tuned run achieved best mAP50 = 0.844 and mAP50-95 = 0.561, with precision = 0.873 and recall = 0.838. While this confirms that the model remains robust under alternative hyperparameter choices, it does not outperform the best regularized configuration (run 2) in any of the main validation metrics.

Qualitative examples

Figure 7 shows sample predictions produced by the overfitting model (best_overfit_3) on ten images from the small dataset. The model detects clear, canonical Santa appearances with relatively high confidence (predictions 1,2 and 8), but it fails on more challenging cases and frequently outputs no detection even when Santa is present (predictions 3 and 5). This illustrates limited robustness: the model has learned a narrow set of “easy” Santa cues and does not generalize reliably to small instances, cluttered scenes, or unusual depictions. The presence of several images that only have background also highlights why a detector trained under reduced regularization can become inconsistent on diverse inputs.

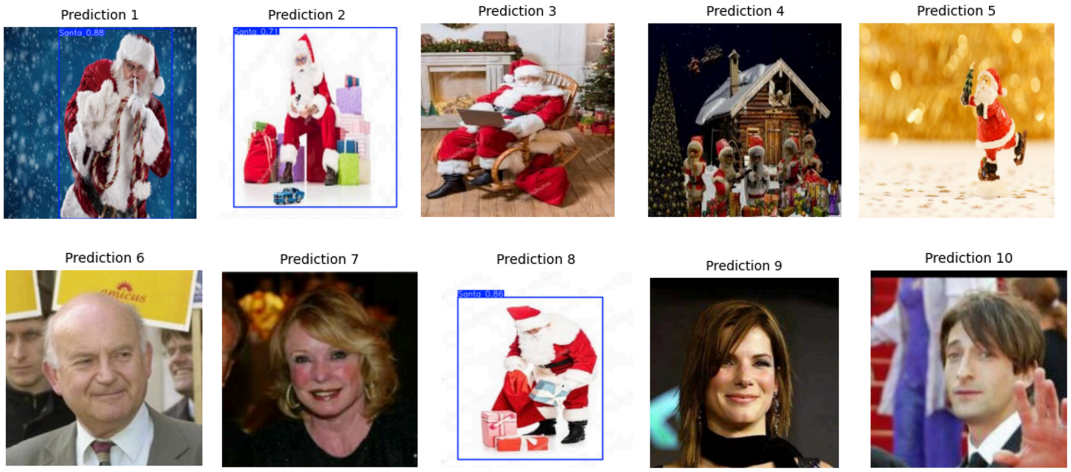

Figure 7: Prediction overfit run 3

Figure 8 compares ground truth labels (left) with the predictions of our best regularized model (right). In contrast to the overfitting run, the regularized model detects a broader range of Santa instances more consistently, including smaller or partially occluded Santas, and produces tighter boxes with high confidence for typical cases. It also behaves more appropriately on images that only have background, reducing incorrect detections compared to the overfitting configuration. Overall, the qualitative behaviour matches the quantitative results: adding augmentation and early stopping improves generalization and makes predictions more stable across the diverse visual styles in our dataset.

Figure 8: Labels and predictions regularized run 2

3. Discussion

The tuned model meets the goal for high-recall Santa detection with mAP50 of 0.924 on the test set. label_smoothing=0.1 and lrf=0.01 haven't completely smoothed the precision cliff observed in baseline runs (*Figure 9*), but the curve now shows a "stepped" or more gradual decline. This indicates that the model is no longer binary (perfect vs. guessing) but has developed a more nuanced understanding of "Santa-like" features. and the model is good at identifying Santas (high precision) while still finding most of them (high recall).

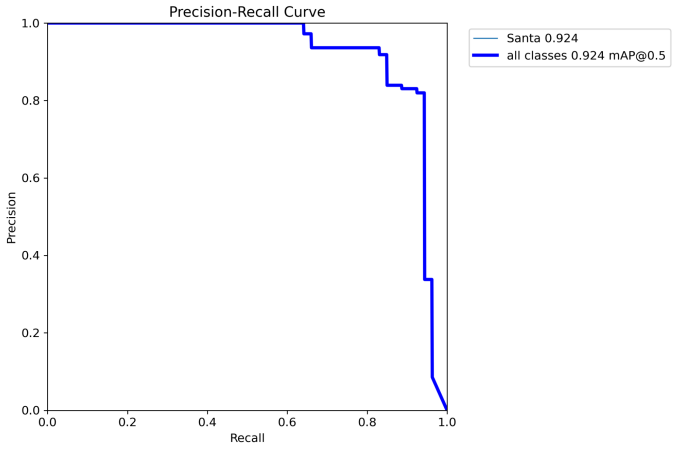

Figure 9: Precision-Recall Curve of a fine-tuned YOLOv8

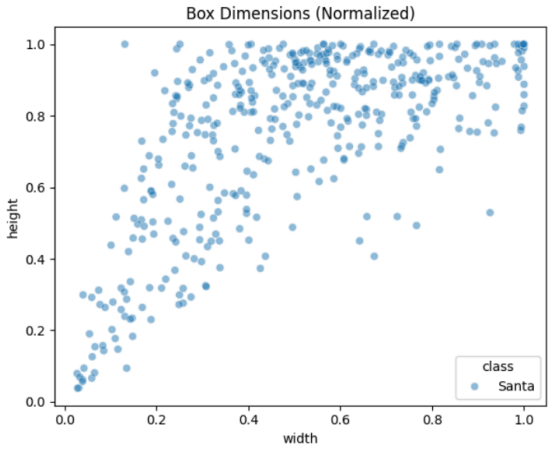

Figure 10: Box Dimensions (Normalized)

Because we have a small dataset, the model eventually runs out of learned patterns. When forced to achieve near-total recall, it is forced to accept lower-confidence detections that are often False Positives. There are also a few hard examples in the dataset that would need to be represented more often to naturally increase the model's confidence in them (like the small object bias).

The model handles partial varying poses, occlusions, and transitions between real-life and cartoon styles well. This is due to data augmentation, which increased data diversity to overcome the limitations of a small dataset. In addition, implementing dropout=0.1 forced the network to generalize by preventing reliance on specific neuronal pathways. The model sometimes struggles with smaller, distant Santas. The Box Dimensions scatter plot (Figure 10) provides an explanation: the dataset contains few examples of small bounding boxes. However, here the problem might be also the lack of recognizable gestalt - in a distance Santa might be just a white-red smudge. An issue is also the case where Grinch is mixed up with Santa. This occurs because the model likely over-prioritizes Santa's clothes (red/white textures) as a primary feature. The current pipeline does not sensitivize the model to this precise distinction.

The transition from baseline to optimized models yielded a cleaner Confusion Matrix, significantly reducing False Positives (only 9 in the test set, Figure 12). Our strategy utilized L2 Regularization which prevented overfitting by penalizing large weights. With Box Gain & close_mosaic=20 the model prioritized bounding box precision, acting as the solution for higher than desired DFL loss.

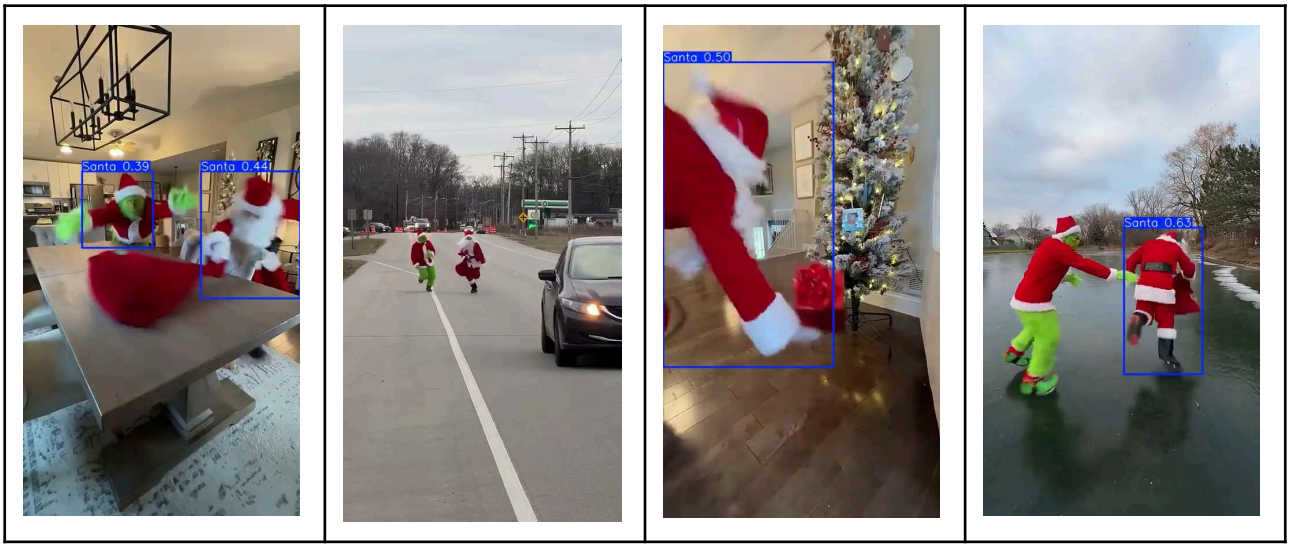

Figure 11: Grinch-Santa confusion, distant Santa figure, good recognition of Santa's parts and at occlusion

With early Stopping (patience=15) we ensured that training stopped before memorization could ruin generalization and with warmup (10 epochs) we prevented early gradient instability on the small dataset.

The F1-Confidence Curve shows a broad plateau, peaking at 0.88. This is a major advantage for the use case, as it allows for stable performance across various confidence thresholds (0.2 to 0.6), rather than requiring a precise "magic number" to achieve high recall without sacrificing precision.

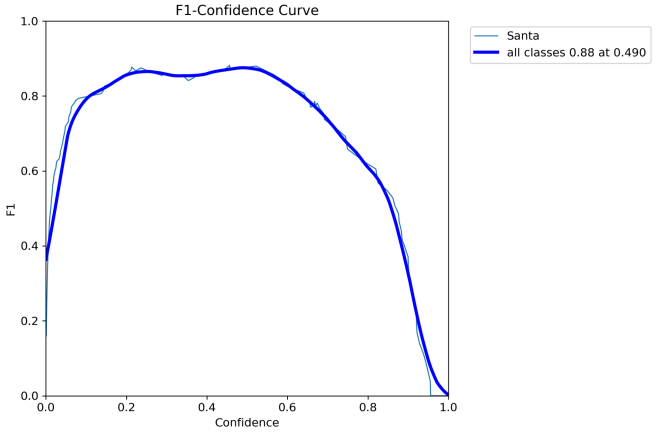

Figure 12: F1 Curve for tuned model performance on test set

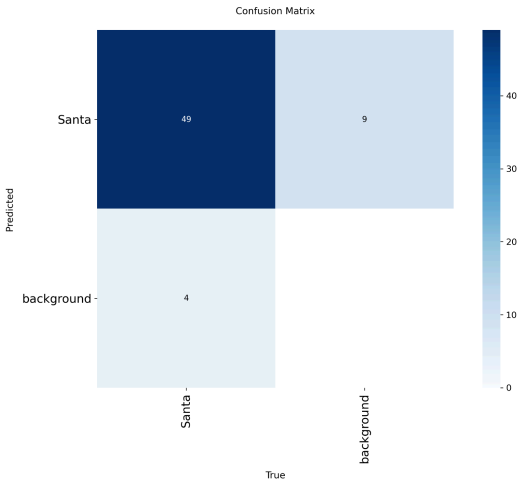

Figure 13: Confusion matrix for test set

Can the research question be answered with the chosen models and data? Partially. The results show that a well-regularized YOLOv8 model can handle style and pose variability with minimal data. However, the chosen data cannot currently support small-object detection or adversarial discrimination (Grinch). The optimization successfully addressed the "precision cliff" and localization errors, but true robustness will require expanding the dataset to include background images and smaller box dimensions to avoid generalizing beyond the data.

When we experimented with the nano and small versions of the YOLOv8, we assumed that nano's limited capacity will force it to prioritize the most dominant features of Santa, effectively acting as a

form of architectural regularization. But YOLOv8 small proved much harder to overfit and also showed promise distinguishing the finer details of the face to avoid the Grinch confusion better, especially if we would enlarge datasets to support that extra learning capacity.

4. Reflection

The project showed that a compact detector such as YOLOv8 can be adapted to a specific object detection problem even with limited data, but only if training is structured carefully. The biggest technical win was that regularization choices (augmentation, early stopping, weight decay and dropout) improved stability and produced a model that achieves high test performance while keeping false positives low. At the same time, the work made the limits of our data very explicit: small or distant Santas remain difficult, and the model can confuse visually similar red/white distractors such as the Grinch, suggesting that the detector relies heavily on texture/color cues than truly discriminative features. Next time, we would prioritize dataset design over further hyperparameter tuning: add more small-object examples, improve annotation consistency, and include explicit “distractor” labels (like the Grinch) and more hard negative background images. A useful assignment improvement would be to encourage or require a distractor/negative set, because it forces a more robust technical discussion on generalization and failure modes rather than only optimizing headline metrics. Because, after all, the issue with any missed Santa is not a lack of model "intelligence" (parameters), but a lack of data and a decently good performance on the small dataset was possible only due to the one-class target and its very recognizable gestalt.

5. References

Kapoor, Sayash, et al. “REFORMS: Consensus-Based Recommendations for Machine-Learning-Based Science.” *REFORMS: Consensus-Based Recommendations for Machine-Learning-Based Science*, 2024, reforms.cs.princeton.edu/.

Karpathy, Andrej. “A Recipe for Training Neural Networks.” *Karpathy.github.io*, 25 Apr. 2019, karpathy.github.io/2019/04/25/recipe/. Accessed 30 Nov. 2025.

Lones, Michael A. “How to Avoid Machine Learning Pitfalls: A Guide for Academic Researchers.” *CoRR*, vol. abs/2108.02497, no. 5, Aug. 2021, arxiv.org/abs/2108.02497. Accessed 4 Dec. 2025