SmartVision Engine: Adaptive Detection System with Automated Retraining Cycle

AUTHORS: JONATHAN MOROCHO

JORGE PIZARRO

1. SUMMARY

The furniture retail sector faces the challenge of personalizing recommendations based on the customer's real context. Current systems lack the ability to automatically interpret the user's environment to suggest complementary products. Therefore, an architecture based on Deep Learning is proposed that uses YOLOv8 for the detection of objects (Sofa, Rug, Pillows) and implements a continuous learning cycle (Active Learning) through user feedback. Performance was evaluated using Mean Average Precision (mAP) with MLflow to track metrics and versioning.

2. PROPOSED METHOD

The implemented solution consists of a **REST API** developed in **FastAPI** that orchestrates the model lifecycle. The retraining request is received from the web application, triggering the relearning cycle where the model receives user corrections and uses a subset of the base dataset (50 images) to perform retraining. Once finished, the system dynamically updates the model in production using a **Hot-Swap** mechanism without stopping the service.

Figure 1 illustrates the flow of data from the system. The user interacts with the REST API to get predictions. When an error is detected, the user sends the correction, which is stored in the *Feedback Dataset*. The retraining process (retrain_service) combines a subset of the original dataset with the new feedback data, trains a new version of the YOLOv8 model, registers it in MLflow, and dynamically updates the model in production without stopping the service.

3. TECHNOLOGIES AND LIBRARIES

The system uses a modern MLOps stack for high performance and scalability:

Technology / Library	Justification
Ultralytics (YOLOv8)	**Main framework for training, facilitating Transfer Learning and Fine-Tuning.**
FastAPI / Uvicorn	**High-performance web framework for API orchestration and model loading.**
MLflow / SQLite	**Experiment monitoring, model registration, and local metadata storage.**
PyTorch / CUDA	**Deep Learning engine and hardware acceleration for retraining.**

Technology / Library	Justification
OpenCV	Real-time image processing and bounding box manipulation.
HTML5/CSS3/JS	Frontend technologies for the interactive user interface.

Figure 1. Flowchart

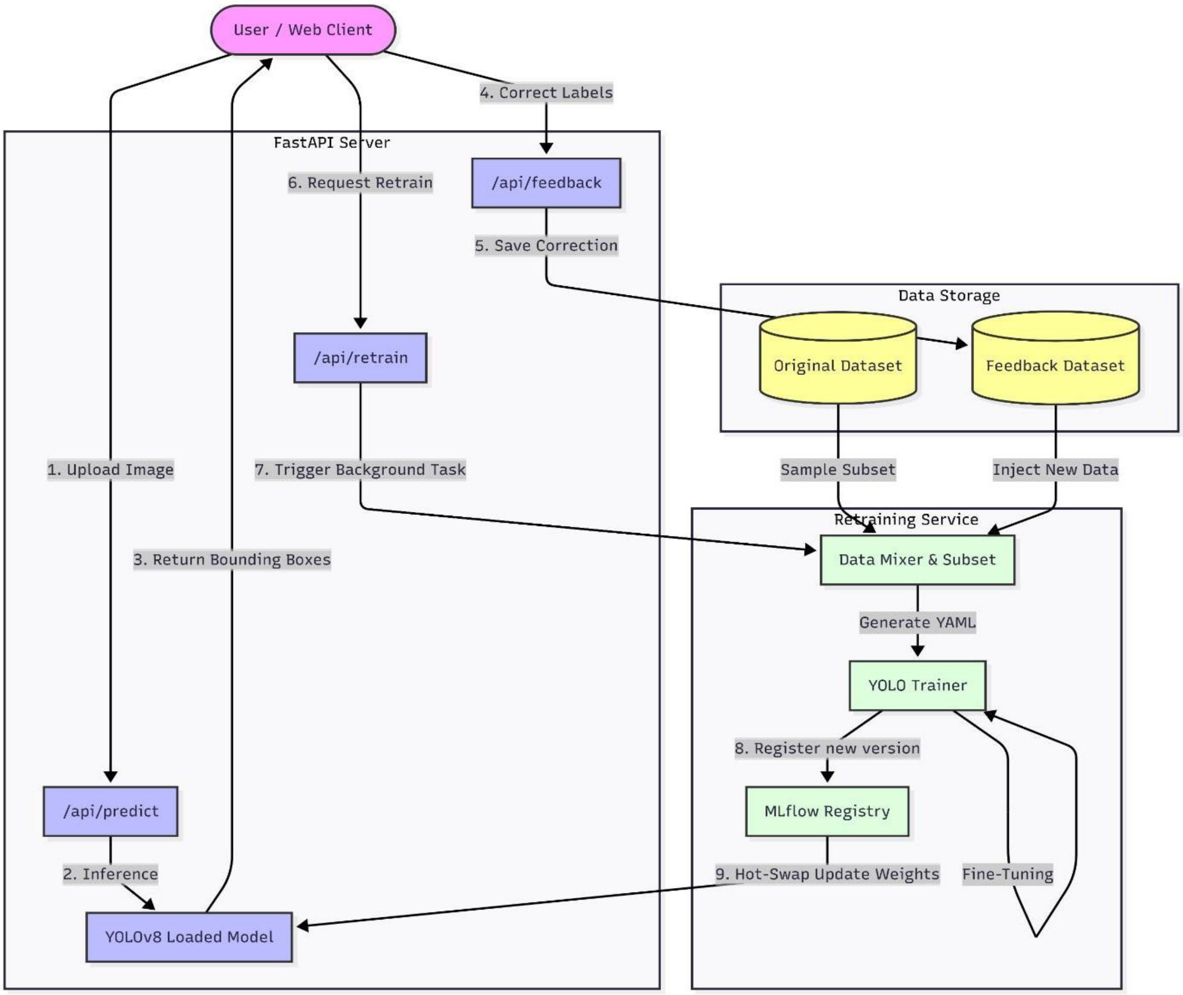


Table 1. Parameters of the proposed method

Name	Description
Class	Class ID detected
Box coordinates	Position of the box in the image

4. EXPERIMENT DESIGN

This section details the characteristics of the data and the configuration of the experiments performed to validate the adaptability of the system.

1) **Dataset characteristics**

The " Living Room Computer Vision Model " dataset hosted in the Roboflow Universe was used. This dataset presents significant challenges such as occlusions (tables covering sofas) and light variability. Because only 3 classes will be taken into account, the analysis is performed only for those classes.

Table 2. Dataset Distribution

Class	Description	Approx.
Sofa	Main element, great variability in shape and color.	~3000
Rug	Carpets, usually occluded by other elements.	~3000
Pillows	Cushions, small objects that are present multiple times in a single image.	~9000

2) **Method optimization parameters**

Two experimental scenarios were designed to demonstrate the evolution of the system:

- **Scenario A:** Simulated training with scarce data (50 images) and few periods, which provided an average base that allows visualizing learning in the future use of the application.
- **Scenario B (Production Retrain):** Re-training using the dataset, the same base (50 images) was used in addition to simulated correction data (taken from validation), applying augmentation techniques such as *Mosaic* to improve robustness.

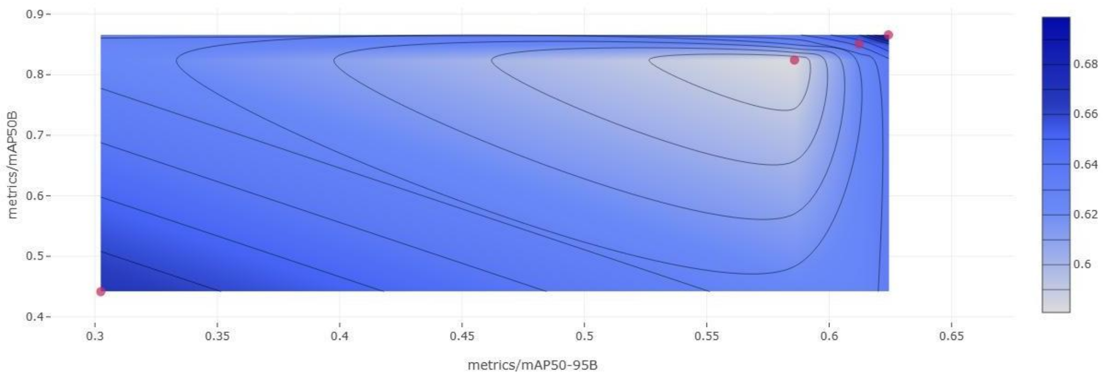
Table 3. Training Hyperparameters

Hyperparameter	Base Model	Retrained Model
Epochs	5	2
Batch size	8	16
Image Size	640	640
Learning Rate	-	$1e-4$
Data Augmentation	-	Mosaic 0.5

5. RESULTS AND DISCUSSION

After the retraining cycle, a significant improvement in generalization was observed. The Base Model presented an mAP of 44%, failing to detect objects in complex selected images. After the refinement phase, the **Refined Model (v4)** showed rapid convergence thanks to **Transfer Learning**, effectively doubling its mAP value. Visual tests confirm the model correctly distinguishes objects and adapts to new user-provided examples.

Figure 1. Evolution of mAP50-G5



To test the model's learning, it was tested against a new image using version 1 and then version 4.

Image 1. Version 1 Prediction

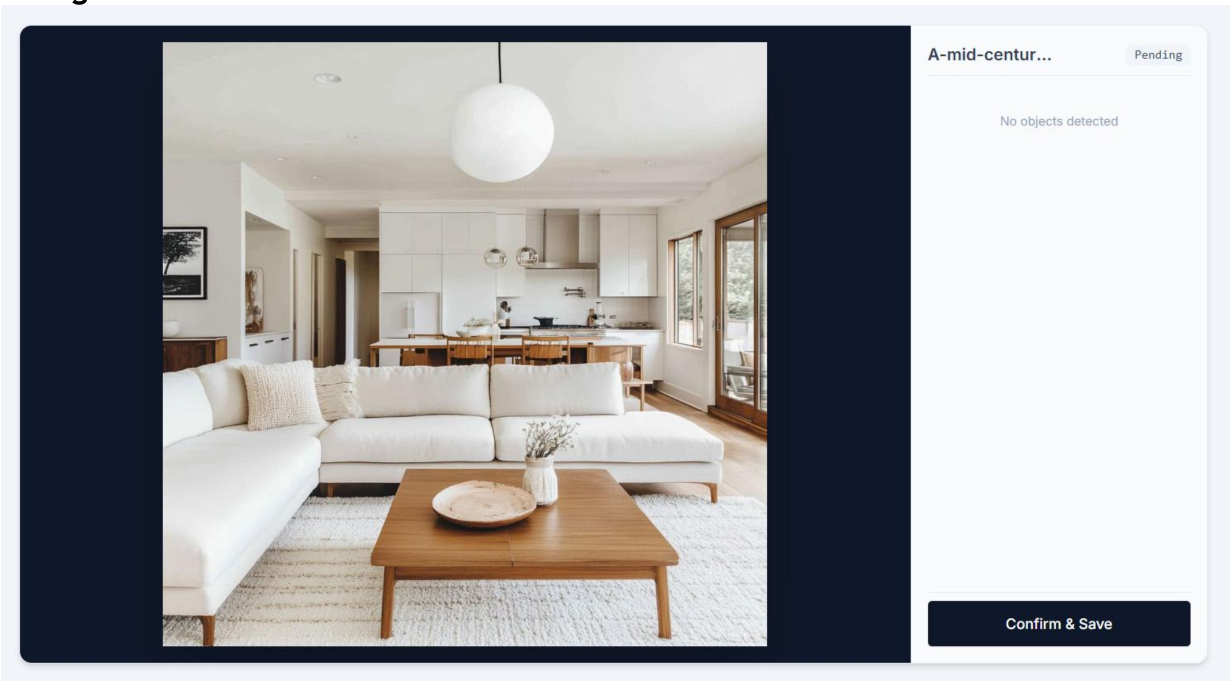
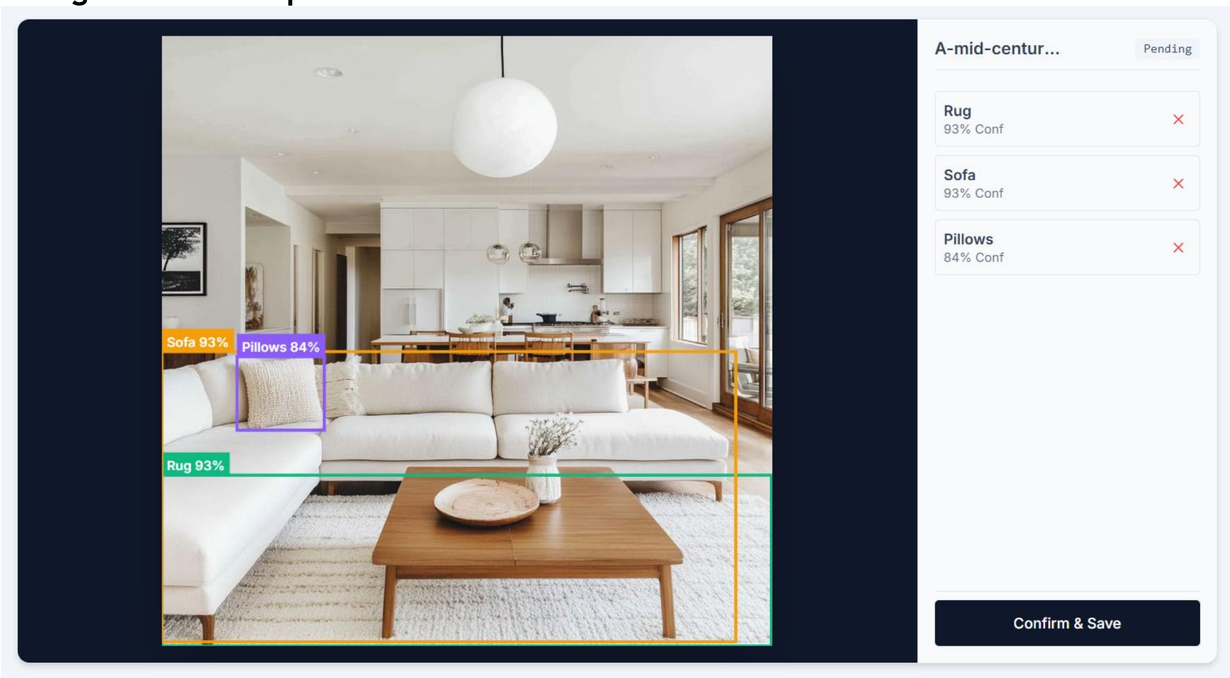


Image 2. Version 2 prediction.



Evidence demonstrates that the model has developed a superior capacity for object discrimination, achieving optimal adaptation to new testing scenarios. The findings indicate that the strategic optimization of hyperparameters was fundamental to the evolution of the system's learning; specifically, the implementation of a low learning rate and the Mosaic augmentation technique significantly enhanced the model's robustness. These adjustments effectively mitigated the risk of overfitting, ensuring a smoother and more horizontal learning curve while doubling the model's overall accuracy.

5. CONCLUSIONS

Implementing an MLOps pipeline integrated with an end-user application can dramatically reduce the gap between model development and its actual utility. It was shown that YOLOv8 is able to adapt to new domains (user-specific images) with a reduced dataset if the appropriate hyperparameter techniques are applied. As future work, it is recommended to implement a comparison system to detect objects similar to those detected.

BIBLIOGRAPHIC REFERENCES

1. Roboflow Universe. (2024). Living Room Object Detection Dataset. Retrieved from https://universe.roboflow.com/living-room/living-room-hn7cw

2. Zaharia, M.A., Chen, A., Davidson, A., Ghodsi, A., Hong, S.A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., Xie, F., C Zumar, C. (2018). Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.*, *41*, 39-45.