zant

October 4-5

TINY_HACK

Turin

The first Embedded AI Vision hackathon

Focus AI

MuseINO

Gruppo 11

Crescenzo De Palma, Francesco Maestri, Stefano Bisignano

with the support of:

ARDUINO TOOLBOX Datapizza

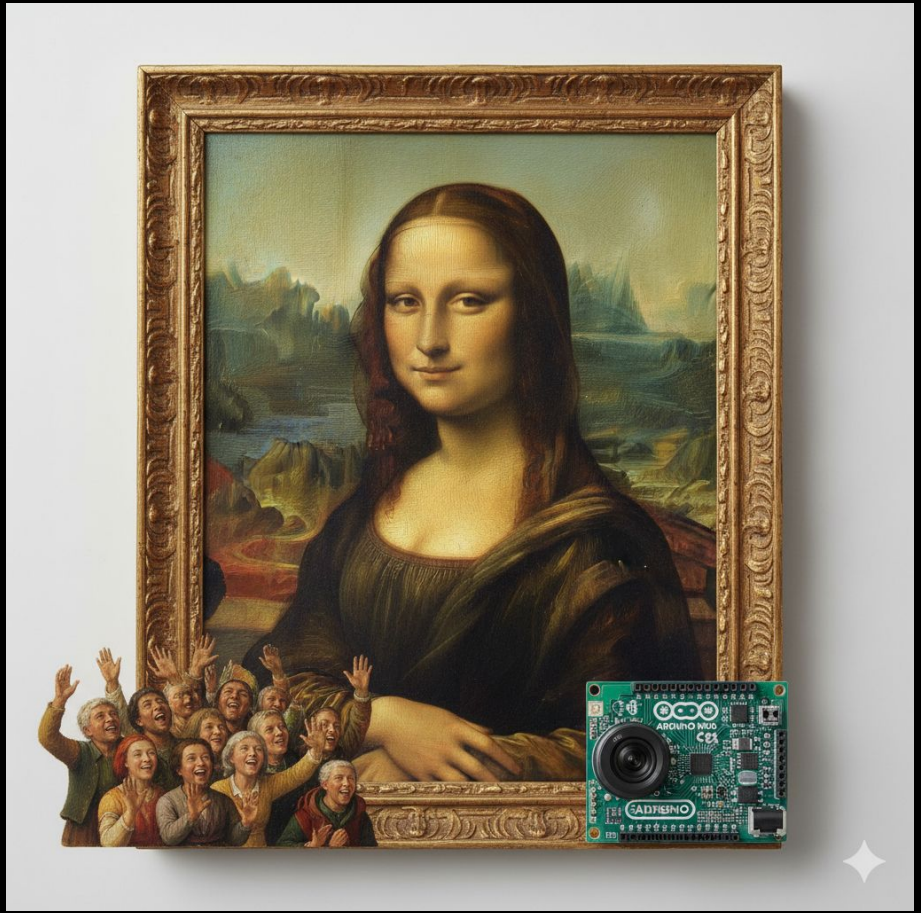
1. Project Overview

👀 Goal: the pipeline counts the number of people passing in front of an artwork in a museum and performs surveillance through a proximity sensor. All of this is performed real-time.

🎯 Use case in brief: measure user engagement and ensure security for an artwork in a museum or gallery.

1. Project Overview

🖼️ Visual:



TINY_HACK

2. Dataset & Model

📊 **Dataset**:

- Adapted COCO dataset to two classes: “person, no_person”.
- Trimmed the dataset to avoid model underfitting and rebalanced over the two classes.
- Light geometric augmentation used during training,such as horizontal_flip, zoom and rotation.

2. Dataset & Model

🧠 **Model architecture**:

- fai-cls-n-coco with 2 classification outputs.
- Trained from scratch on FocoosAI web platform.
- We experimented discarding the last classification layers to use the final feature map as a “detection heat map”.
  - The final resolution was too low to allow any type of reasoning at that stage, but it was worth trying 🤖.

2. Dataset & Model

⚖️ **Trade-offs**:

- Despite being very tiny, the model performs an inference in 1.9s.
- The application would require a tinier model or a faster processor.
- The input resolution, kept at 96x96 is necessary to meet the memory constraints.
- The obvious tradeoff is in classification accuracy.
- The usage of a classification cannot ensure precise people counting.

💡 **Reasoning**:

- Everything happens on device, granting data-privacy and an easy setup.

3. Deployment Pipeline

🔧 Pipeline steps

- Init and data acquisition (image + proximity).
- Model performs classification.
- Serial transmission
- Data merge and processing (safety proximity monitoring + people counting)
- Data visualization and dashboard

⚡ **Performance**: 0.5 frames/second

📦 **Submission files**: .ino, .onnx, model_info.json, webpages, mobile app -> all shared on repo ✅

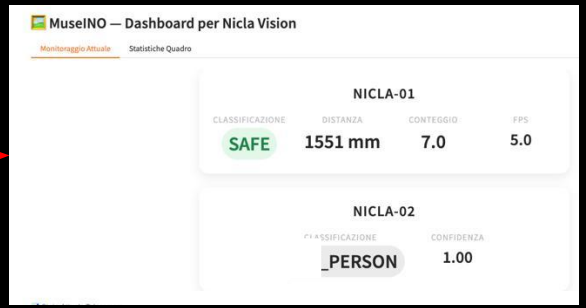
TINY_HACK

4. Demo & User Experience

🎥 **Demo flow**: devices → classification + proximity → output

🌐 **Integration**: connection via serial to backend

👤 **UX**: outputs proximity from camera laser + classification (person, no_person)



5. Impact & Next Steps

🌟 **Innovation & originality**:

- The solution fits in a context where a statistical insight is beneficial, but data privacy is crucial.
- It is an all-in-one solution that requires a minimal setup and connects with a remote backend.
- This type of analysis has a broad range of applications, wherever knowledge of the interest for a specific article/element is required (grocery, retail).

5. Impact & Next Steps

🚀 **Future potential**:

- The project would benefit from a specifically crafted detection model + tracking algorithm, that is optimized for this specific task and can grant exact people counting.
- COCO was used for people classification, a more pedestrian/crowd dataset oriented could be a good addition to the already employed dataset.
- Wifi communication instead of serial.

That’s a wrap! 🌮

Thanks