Summer Student Project Report 2024

Advancements on Aerial Robotic Systems and Sensor Technologies for Future HEP Detectors

Student:
Chan Chi Kit[1]

Supervisors:
Francesco Mazzei[2,3]
Paolo Francesco Scaramuzzino[2]
Corrado Gargiulo[2]

[1] The Hong Kong University of Science and Technology (HKUST), Hong Kong, China
[2] CERN, European Organization for Nuclear Research, Geneva, Switzerland
[3] University of Rome Tor Vergata, Rome, Italy

Abstract

The European Organization for Nuclear Research (CERN) has pioneered the development of a robotic Lighter-than-Air (LtA) Unmanned Aerial Vehicle (UAV), specifically a blimp, aimed at mitigating radiation exposure risks for human personnel and optimizing detector uptime. This document elucidates the evolution of a blimp detection model based on deep learning algorithms and its integration within a localization system based on external cameras. Furthermore, it explores the integration process of a hybrid pixel detector for real-time radiation monitoring within the blimp payload.

Contents

1 Introduction

The deployment of robotic systems in environments featuring harsh conditions, such as underground particle detector caverns, is being investigated at the European Organization for Nuclear Research (CERN) [1].With the future High Energy Physics (HEP) experiment projects currently under discussion at CERN, there will be higher radiation levels inside detector caverns, making human presence potentially dangerous even long after the beam run has been shut down. [2]. In fact, nowadays, in the event of a suspected malfunction, it is necessary to dump the beam and wait for radiation levels to decrease below the thresholds that endanger human health before operators can access the cavern and verify the triggered alarm. In the future, this process could become prohibitive. Therefore, the deployment of robotic systems inside the cavern could offer several benefits, such as reducing human exposure to radiation, enabling continuous environmental monitoring during the beam run, and allowing for timely verification of triggered alarms and seamless detection of false alarms, thereby maximizing beam run time.

To this end, the CERN Experimental Physics (EP) Detector Technologies (DT) department, as part of the EP R&D program [3], is developing an indoor blimp in collaboration with an external company, Windreiter. This blimp is designed to collect three-dimensional spatial data of the cavern environment, including radiation and magnetic fields, temperature, and humidity, as well as to enable visual inspection of the detector [1]. The project aims to develop an autonomous system capable of flying and monitoring the caverns of future experiments without human intervention.

To operate a vehicle remotely or autonomously, it is necessary to know its real-time position. These vehicles are usually equipped with on-board navigation systems based on inertial odometry. Such systems, however, suffer from drift, preventing them to be accurate enough to enable safe navigation. Reliable outdoor navigation is facilitated by the Global Navigation Satellite System (GNSS) that, through the use of Real-Time Kinematic (RTK) positioning, allows to reach the required accuracy. Navigation in GNSS-denied areas, as well as indoor navigation, especially in cluttered environments like the cavern, pose several challenges. One valid solution enabling relative localization consists of expanding the on-board suite of navigation sensors with a 3-D LiDAR (Light Detection And Ranging) used in combination with a SLAM (Simultaneous Localization and Mapping) algorithm. This solution is better suited for navigating unknown environments. Since the cavern is a known environment with an available 3D map (though it may contain some dynamic obstacles), this approach isn't ideal. Additionally, using it on the blimp is not feasible because the extra hardware would likely exceed the maximum payload capacity, and the computational demands of running a SLAM algorithm would be too high for the on-board computer. In a similar fashion, relative navigation is possible also through computer vision or visual-inertial odometry that uses information coming from the on-board camera(s) to correct the drift of the inertial odometry. For a known environment as the cavern and due to the inherent limitations of the blimp, it is more convenient to adopt an external indoor positioning system. Several solutions exist encompassing systems based on Radio Frequency (RF) (e.g., Wi-Fi, Bluetooth, ZigBee, RF identification (RFID), Ultra-Wide Band (UWB)), Infrared (IR), Visible Light Communication (VLC)/Optical, Ultrasonic, and Magnetic. The majority of these systems are unable to provide the required accuracy, require the vehicle to be in the line-of-sight with a very limited range or are affected by measurement noise due to the presence of many obstacles and obstructions causing reflections. To achieve the desired accuracy, an indoor positioning system based on external cameras installed inside the cavern was chosen. This system requires a blimp detection model and a triangulation algorithm to compute the blimp's absolute 3D position by combining object-detection data from multiple camera views.

This report first presents the methods and results for the development of a blimp localization system based on computer vision. An object detection model is developed using the You Only Look Once version 8 (YOLO v8) deep learning algorithm. This model can be used to locate unnamed aerial vehicles in an indoor environment. Then, the report focuses on the integration of a hybrid pixel detector for real-time radiation monitoring within the blimp payload, providing valuable information on performing data acquisition and processing through micro-controllers, such as a Raspberry Pi.

2 Methodology

2.1 Blimp localization system

To navigate with the blimp inside a detector cavern it is necessary to know its real-time position. To this end, an indoor positioning system is necessary. To achieve the desired accuracy, an indoor positioning system based on external cameras installed inside the cavern was selected. This solution does not require any additional hardware installations on the blimp nor computationally expensive algorithms to run on the on-board computer. In turn, it is necessary to develop a blimp detection model. This task was conducted using YOLO version 8, training the model on a vast collection of blimp photos.

2.1.1 Theory and Principle

Running the blimp detection model involves real-time video capture from a webcam, where each video frame undergoes meticulous analysis. The model dynamically identifies the blimp within each frame, delineating a bounding box that encapsulates the blimp's spatial extent. Furthermore, it computationally derives the x and y coordinates corresponding to the center of this bounding box within the camera's field of view.

By extrapolating the position of the identified box as discerned through a single camera's perspective, a framework is established to extrapolate the three-dimensional (3D) location of the blimp's centre of volume employing two or more cameras. This multi-camera setup enables the triangulation of the blimp's precise spatial coordinates within the cavern, facilitating comprehensive tracking and localization.

Fig. 1 shows the workflow adopted to obtain a reliable blimp localization system together with the tools used in each step of the process..

2.1.2 You Only Look Once (YOLO)

You Only Look Once (YOLO) is a deep learning algorithm widely used for object detection models introduced by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi [4]. Unlike traditional methods that repurpose classifiers for detection tasks, YOLO frames object detection as a regression problem, predicting bounding boxes and associated class probabilities directly from full images in a single evaluation.

Figure 1: Workflow of blimp localization system training.

With its exceptional speed and accuracy balance, the YOLO algorithm has distinguished itself as a tool for quickly and accurately identifying objects in photos [5]. The *base YOLO* model processes images in real-time at 45 frames per second, while the faster variant, *Fast YOLO*, achieves 155 frames per second while doubling the mean average precision of other real-time detectors. YOLO's streamlined approach eliminates the need for complex pipelines, enabling rapid processing with minimal latency [4].

To predict each bounding box, or object on the image, the YOLO algorithm adopts characteristics from the entire image. Additionally, it simultaneously predicts all bounding boxes for a picture in all of the classes. The image is divided into an $S \times S$ grid, and predictions are made for each grid cell regarding B bounding boxes, their corresponding confidence, and the probability of the C class. The tensor encoded with these predictions is $S \times S \times (B \times 5 + C)$ [4]. This procedure is shown in Fig. 2.

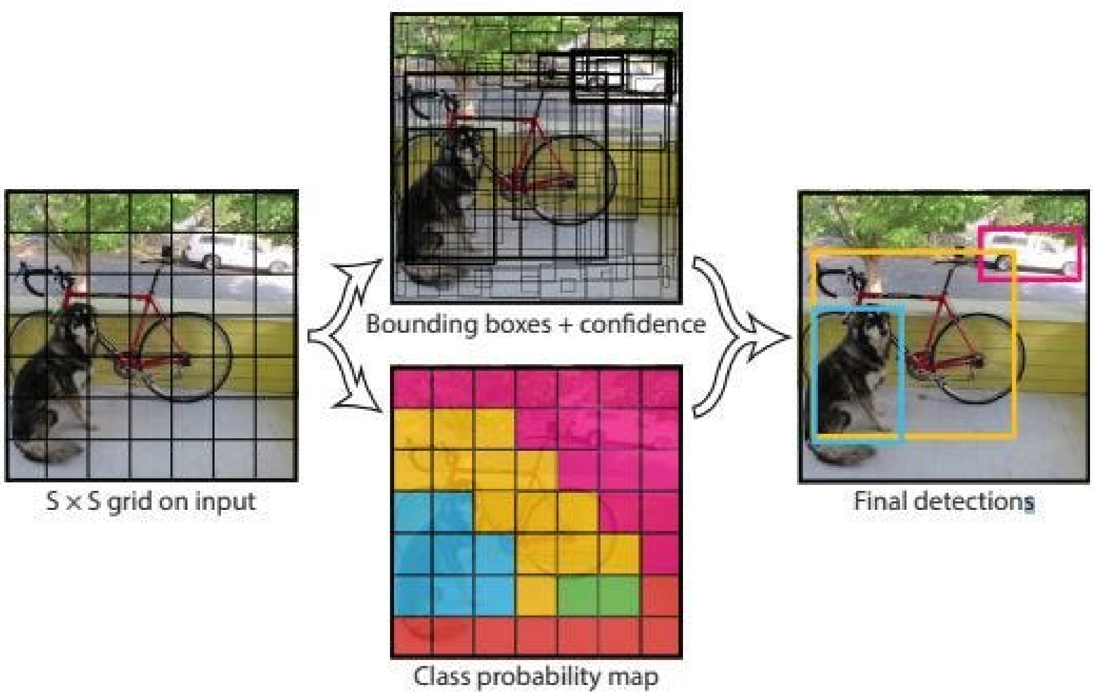


Figure 2: Working principle of YOLO [4].

2.1.3 YOLO v8 performance

YOLO version 8 was published in January 2023, producing five model weights: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) and YOLOv8x (extra-large). Numerous vision tasks, including object identification, segmentation, pose estimation, tracking, and classification, are supported by YOLOv8 [5].

YOLOv8 has a similar structure to YOLOv5, but with some modifications to the CSPLayer—which is now known as the C2f module. To increase detection accuracy, the C2f module (cross-stage partial bottleneck with two convolutions) combines contextual information with high-level features. The models also process classification, and regression tasks separately using an anchor-free model with a decoupled head. This arrangement enhances the accuracy of the model overall and lets each branch concentrate on what it can do best [5].

YOLO v8 predicts results with reduced latency and can get the same accuracy with less parameters when compared with its predecessors in the YOLO family [6]. Extensive data are displayed in Fig. 3.

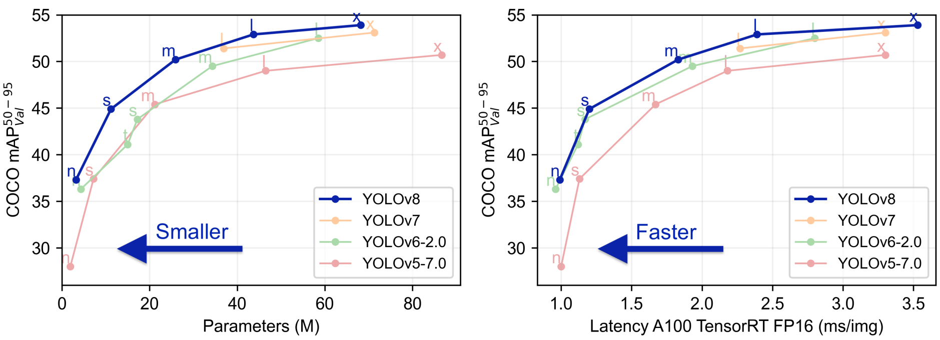


Figure 3: Comparison of different versions of YOLO [6].

2.1.4 Object Detection Metrics

The strengths and shortcomings of an object detection model can be determined by examining its performance metrics. They serve as quantitative indicators that are used to evaluate the algorithm's accuracy. To be more precise, these metrics assess how well a video or image frame is detected, located, and classified. We can compare and improve the performance of various models used for object identification and picture classification by using these metrics [7]. Before introducing these metrics, it is important to define the confusion matrix, that embeds the following concepts:

- **True Positive (TP)**: The model correctly predicts a positive outcome when the actual outcome is indeed positive.
- **True Negative (TN)**: The model correctly predicts a negative outcome when the actual outcome is negative.
- **False Positive (FP)**: The model predicts a positive outcome when the actual outcome is negative.
- **False Negative (FN)**: The model predicts a negative outcome when the actual outcome is positive.

The metrics used to define the performance of an object detection model are:

- **Precision**: The percentage of correct predictions (how accurate the model is).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{1}$$

- **Recall**: How well the model finds all positive.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2}$$

- **F1 score**: A measure that takes both precision and recall in account.

$$\text{F1} = \frac{\text{precision} \times \text{recall}}{\text{precision+recall}} \tag{3}$$

- **Intersection over Union (IoU)**: How much the predicted bounding box overlaps the ground truth box.

$$\text{IoU} = \frac{\text{area of overlap}}{\text{area of union}} \tag{4}$$

2.1.5 Integration of the Object Detection Model in the Blimp Localization System

The blimp detection model provides the x-y position in the local camera system of the center of the bounding box encompassing the detected object. The information from two or more cameras are required to localize the blimp in the absolute reference system (Fig. 4). To this end, a Python script combining the information coming from the application of the detection model to two or multiple camera views to triangulate the blimp was developed. The triangulation algorithm embedded in the OpenCV package [8] was used, requiring as input also the 3×4 projection matrix of each camera (containing information such as the field of view, the position and the orientation of the camera) that allows to switch from the local camera system to the absolute reference system.

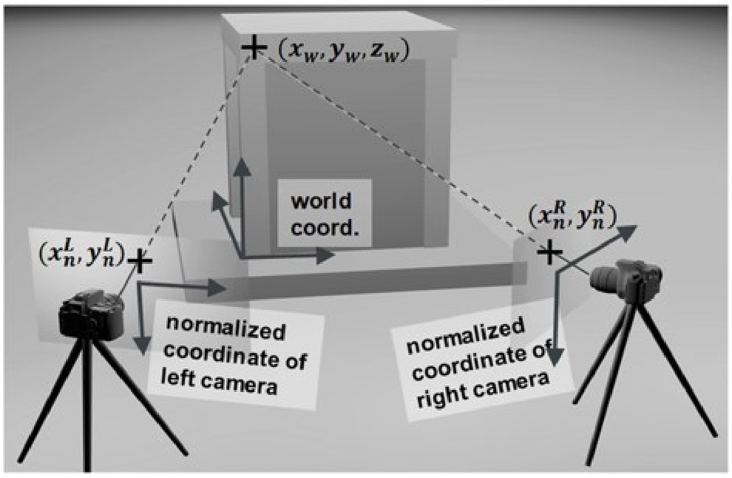
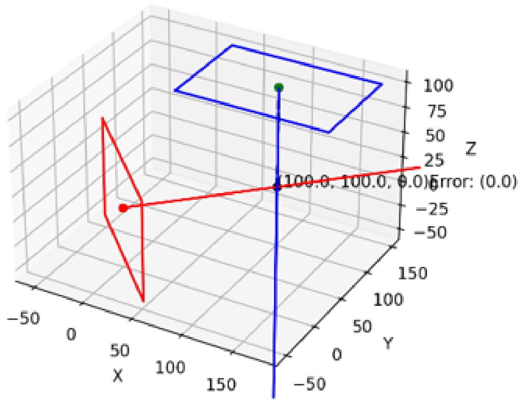


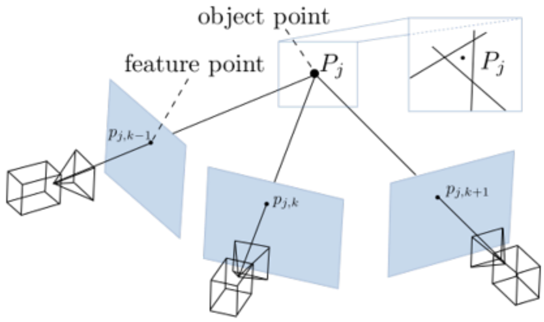
Figure 4: Example of triangulation system with two cameras [9].

If the projection lines from the cameras detecting the blimpintersect, the point of intersection is identified as the blimp position (Fig. 5a).

However, due to potential noise and errors in the detection model, the projection lines may not intersect precisely. In such cases, the triangulation will identify the point equidistant from the projection lines as the blimp position, as shown in Fig. 5b.



(a) Direct interception.



(b) Indirect interception [10]

Figure 5: Object localization under different interception scenatios.

2.2 Onboard radiation monitoring with MiniPIX TPX3 detector

Measuring the radiation dose is crucial when mapping a detector cavern's 3D environment since it indicates which regions will have high radiation concentrations that could endanger individuals. The blimp could also reduce radiation exposure and sensor damage by using the collected spatial data to stay clear of high radiation locations during beam run. This can only be done *a posteriori*, otherwise we would need a simulated background radiation field of the detector to enable trajectory optimization minimizing radiation exposure.

2.2.1 Theory and Principle

The blimp will be equipped with a radiation detector to enable cavern inspection and environmental monitoring.. While maintaining the ability to process and transfer data, the entire configuration must be as lightweight as possible. To this end, a hybrid semiconductor pixel detector, MiniPix TPX3, was used. The MiniPIX TPX3 detector is a miniature radiation camera from the company ADVACAM [11] with a Timepix3 chip for particle tracking and imaging. The detector weighs 41g and is powered and transfer data via a micro USB 2.0 port [11]. The chip has a pitch of 55 μm and has 256×256 square pixels [12]. It can operate in both Time-of-Arrival (ToA) and Time-over-Threshold (ToT) modes. Our available sensor material is silicon (Si) with a thickness of 500 μm.

For the purpose of facilitating communication, a microcontroller that is both sufficiently light for the blimp, that features a maximum payload of 200 grams, and capable of processing data onboard and storing them locally and/or transferring them via WiFi or Bluetooth is required.

2.2.2 Hybrid Silicon pixel detectors

In the inner detection layers of the four larger experiments (ATLAS, CMS, ALICE and LHCb) at the CERN Large Hadron Collider (LHC), silicon pixel detectors are widely used. These detectors are used to identify and record the path of particles produced in high-energy collisions [13].

A silicon sensor consists of an aluminium layer on top, an n-type implant just below the top surface, a silicon sensor chip, and p-type implant pixels toward the bottom of the sensor. These are the components that make up a hybrid silicon pixel detector sensor part. With a high voltage (positive) connection to the aluminium surface and a negative connection to the p-type silicon layer, they combine to produce an array of silicon diodes that operate in reverse bias mode. Via a layer of solder bumps, a hybrid readout ASIC (Application Specific Integrated Circuit) is attached to the sensor component. Fig. 6 displays the schematic cross-section of a hybrid silicon pixel detector.

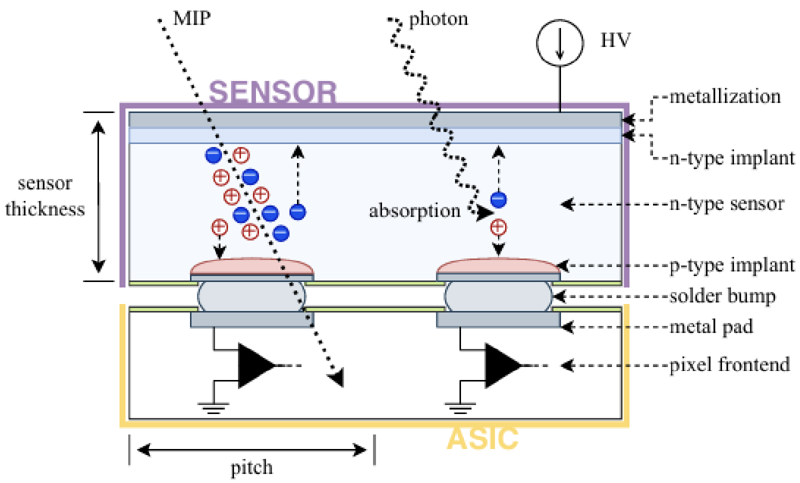


Figure 6: Schematic cross-section of a hybrid pixel detector [13].

When highly energetic particles pass through the sensor, charge carriers are created along or at the end of the particle trajectory, depending on the type of particle. These charged carriers will move towards the p-type or n-type implant of the sensor, creating a current which would be detected by the readout ASIC.

2.2.3 Radiation monitoring

The sensor has to be first calibrated [14], ensuring each pixel of the detector will have the same ToT with the same type of particles. This can be done by threshold equalization using the Pixet Pro Software equalization tool. Equalization tool location and threshold equalization results are shown in Fig. 7 [15].

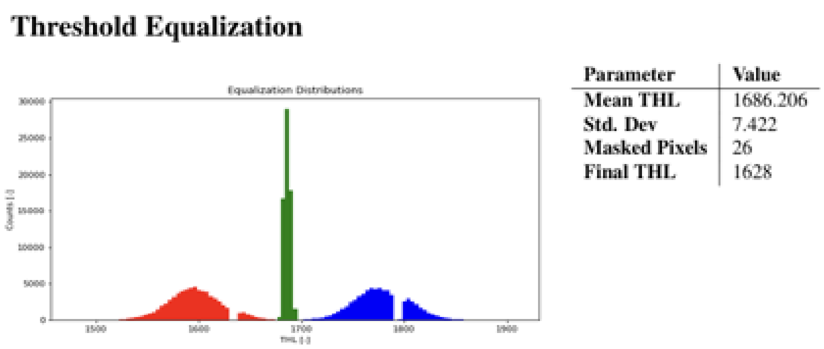
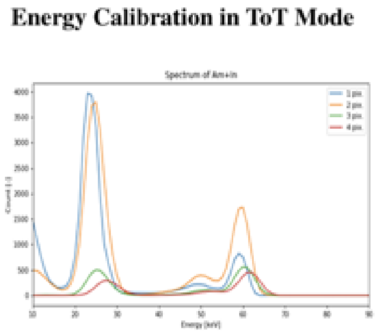


Figure 7: Threshold Equalization [15].

One can then use known radiation sources, such as Fe, In and Am to calibrate the ToT to energy graph by estimating the coefficients a, b, c and t in the surrogate function [16]:

$$f(x) = ax + b - \frac{c}{x - t} \tag{5}$$

The graph of energy calibration in ToT mode and the estimated coefficients of the surrogate function of Eq. (5) are shown in Fig. 8.



Source	Energy	1	2	3	4
Fe	5.725	5.124	8.106		
In	24.047	23.628	24.696	25.356	27.419
Am	59.499	59.378	59.748	60.546	61.734

Coefficient	Mean value
A	2.100
B	15.903
C	23.483
T	1.967

Figure 8: Energy calibration in ToT mode and parameter values [15].

The surrogate function of Eq. (5) represents the dependency of ToT on particle energy, as shown in Fig. 9. Using the inverse of this function, the energy of each incoming particle can be estimated. As each particle would have different effects on the human body, clustering is required to identify the type of particles collected by the detector [17]. By multiplying the quality factor with the absorbed dose for each type of particle and summing the entire energy over a predetermined period, the effective dose rate can be computed.

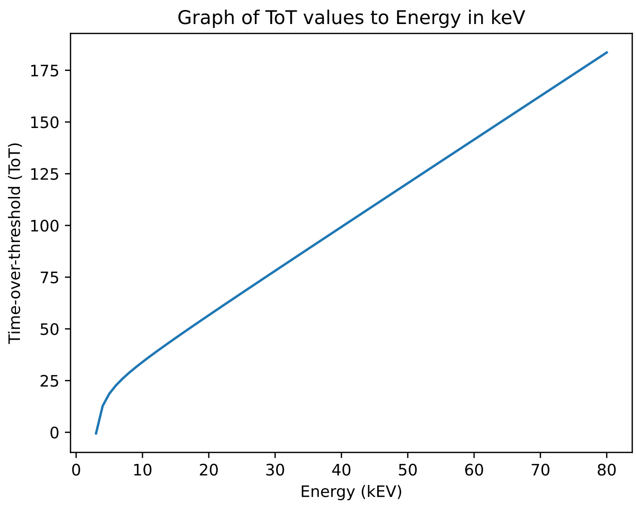


Figure 9: Graph of ToT value to Energy in keV.

3 Results and discussions

3.1 Blimp localization system

A total of six blimp detection models were trained. The differences between the datasets used to train each model will be discussed, and then the performance of the different blimp detection models will be compared.

All the code and training materials used to generate the Blimp detection model can be found on the following GitHub page: `https://github.com/Kelvinchan324/CERN_blimp_dectection`.

3.1.1 Dataset preparation

Before training the models, photos of the blimp were annotated by the developer using labelling tools, such as Label Studio, LabelImg and Roboflow platforms to provide the ground truth. Then, each dataset of photos was divided into three groups. Each subset of photos is used in a different phase of the model generation that comprises three phases: training, validation and testing. Tab. 1 shows the number of photos used in each dataset to generate the model.

Table 1: Number of annotated photos used in different training models

Object	Model name	Training	Validation	Testing	Total
Generic blimps	`20240731-r`	5595	69	230	5894
	`20240731-py`	5595	69	230	5894
Specific blimp	`20240802-py`	12	1	1	14
	`20240806-py`	184	30	10	224
	`20240807-py`	287	70	24	381
	`20240808-py`	600	100	54	754

“`r`”: Trained with Roboflow training platform.
“`py`”: Trained locally using a Python script.

Models are named after their training dates and training platforms: models trained with Roboflow training platform have “`r`” as suffix, whereas those trained locally using a Python script have “`py`” as suffix. Models were all trained with the YOLO v8 algorithm despite the use of different training platforms.

For models “`20240731-r`” and “`20240731-py`”, generic blimp images are used. Examples of annotated generic blimps are shown in Fig. 10.

To train the model with images specifically related to our blimp prototype, which is being developed in collaboration with Windreiter, and not just with generic airship images, we used photos of the blimp taken during a test flight conducted in the ALICE cavern at CERN, as well as images captured during test flights within Windreiter’s facilities. This imageswere used for models “`20240802-py`”, “`20240806-py`”, “`20240807-py`” and “`20240808-py`”. Examples of annotated specific blimp are shown in Fig. 11.

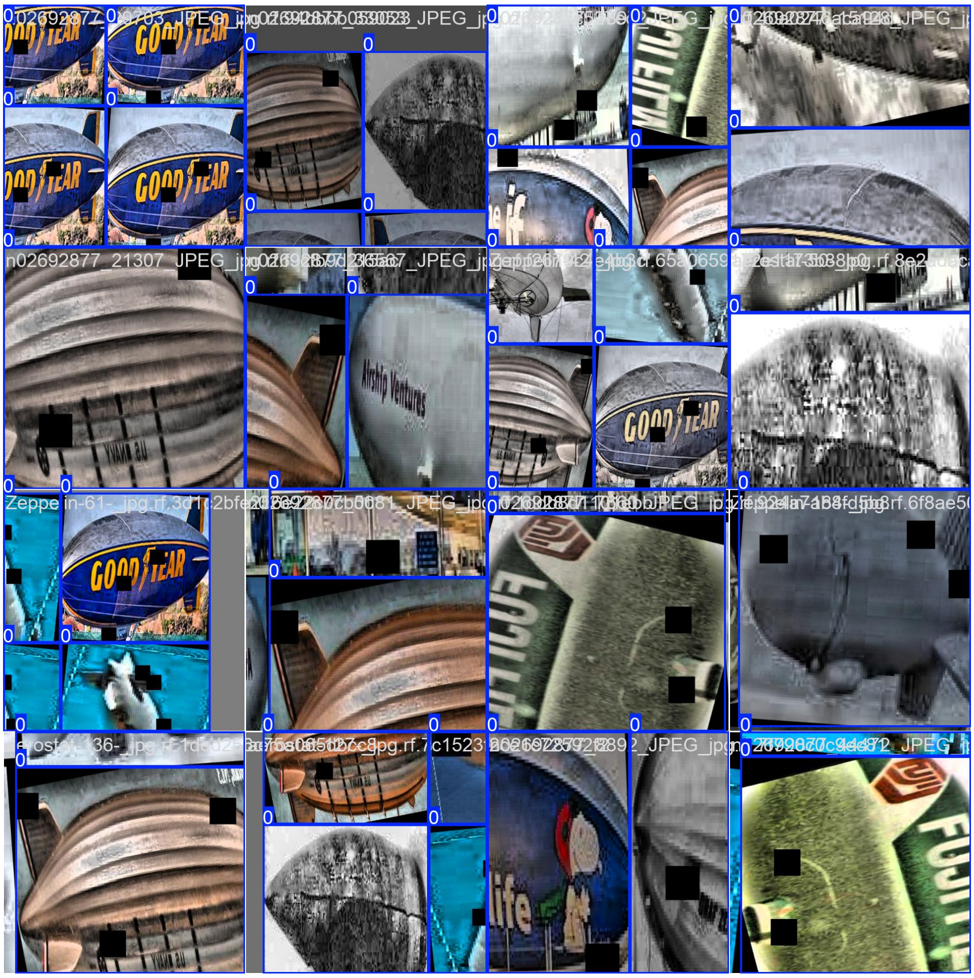


Figure 10: Annotated dataset with generic blimp images from online sources.

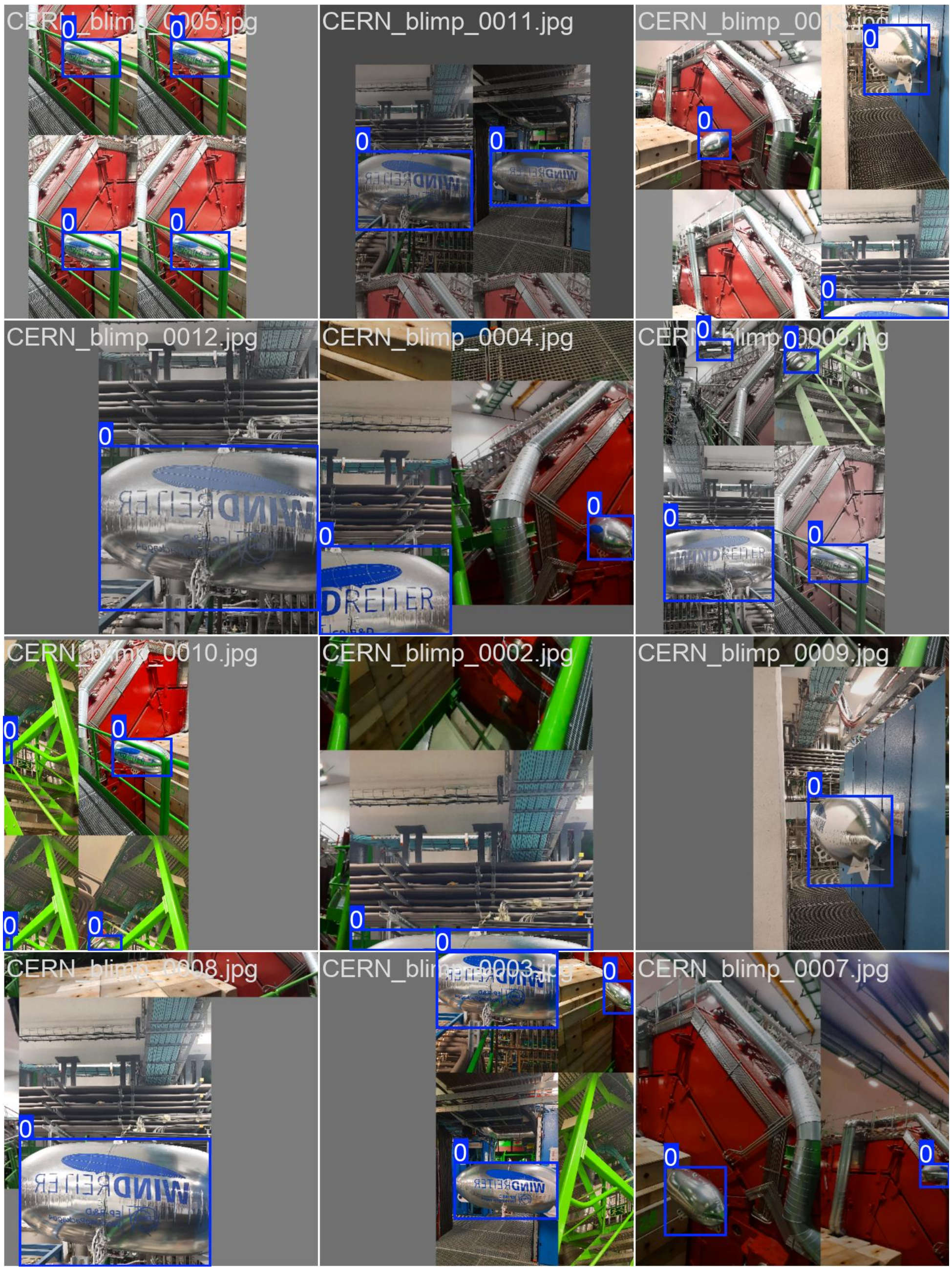


Figure 11: Annotated dataset with specific blimp images from CERN Alice cavern and Windreiter testing area.

3.1.2 Results

Models “20240731-r” and “20240731-py” share the same dataset, but the former was trained with Roboflow training platform, whereas the latter was trained locally with a Python script. Despite being trained on a different platform, YOLO v8 deep learning algorithm was used in both cases. Therefore, it is reasonable to expect similar results from these two models. Furthermore, the Roboflow model can be used to verify that the Python script does not contain any bugs.

Model “20240731-r” Despite being trained on a large dataset with around 6000 photos and exhibiting excellent performance metrics (mAP = 99.5%, precision = 99.9%, and recall = 100%), this model performs poorly. Indeed, it identifies any item that appears in a video or on a camera as a blimp. An example of a “20240731-r” model prediction result is shown in Fig. 12, where several blimps are detected even though there is no blimp whatsoever.

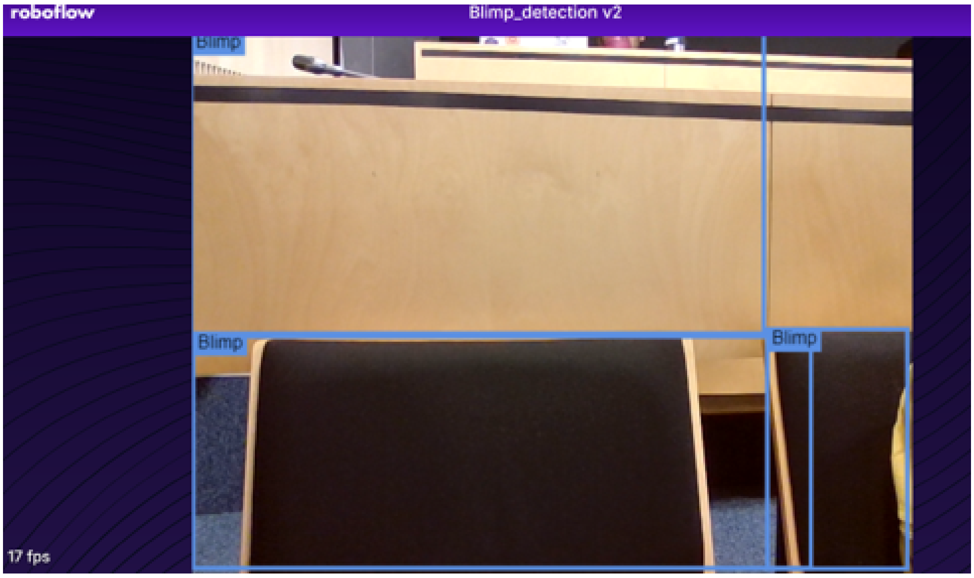


Figure 12: Example prediction result of model “20240731-r”: Chair and wall detected as a blimp.

Model “20240731-py” Similarly, model “20240731-py” exhibits excellent performance metrics, with a very high (close to 1.0) Precision, Recall and F1 score. The Confusion Matrix shows that all testing images were correctly predicted. The metrics graphs are shown in Fig. 23 in Appendix A.. However, the detection results were nearly identical to those from model “20240731-r”. Also in this case, several blimps are detected even though there is no blimp whatsoever as shown in Fig. 13.

Model “20240802-py” Considering that earlier models, which used a huge dataset of 6000 photos of generic blimps, performed poorly, a dataset of 14 unique blimp photos captured within the CERN ALICE detector cavern was used in this model. The “20240802-py” model exhibits a low F1 score, and low recall, but high precision (Fig. 24) when it comes to blimp detection from particular angles. The results from Fig. 14 and 15 show that the blimp detection occasionally loses track and has a low confidence

Figure 13: Example prediction result of model “20240731-py”: Chair and wall detected as a blimp.

rate (between 0.3 and 0.5). In this model, the blimp in the testing image (only one testing image) was predicted as the background. This is probably caused by the small dataset size.

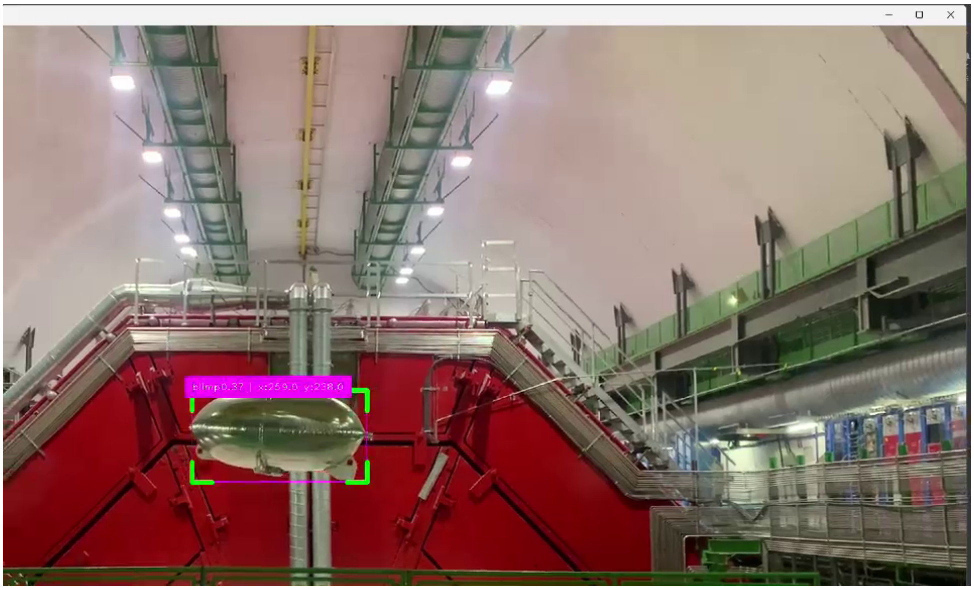


Figure 14: Example prediction result of model “20240802-py”: blimp recognised.

Model “20240806-py” Given that the blimp was successfully detected using a rather limited dataset in the model “20240802-py”, the dataset size of model “20240806-py” was increased by 16 times, with 224 photos in total. Using videos captured inside the ALICE cave in 2022, the model can now accurately, with an average confidence rate of 0.9, identify the blimp without losing track (Fig. 16 and 17). Additionally, it can identify the most recent blimp (2024) from footage captured in Windreiter’s testing zone. Still, the model cannot identify the blimp when there is dark lighting inside or when it is

Figure 15: Example prediction result of model “20240802-py”: model loses track of the blimp 0.2 seconds after Fig. 14.

only partially visible. This is likely related to training photosthat are predominantly showing only the entire blimp shape in the ALICE cavern under bright lighting. The model exhibits a high precision value and an acceptable recall and F1 values, as shown in Fig. 25.

Figure 16: Example prediction result of model “20240806-py”: blimp front view recognised.

Model “20240807-py” Throughout the development process, the blimp prototype underwent several changes with respect to the initial version. Windreiter is providing videos to show the evolution of the blimp development. Model “20240807-py” was trained using a dataset that included both blimp images from the ALICE Cavern and the latest blimp images from Windreiter’s testing area. With a high confidence rate for both

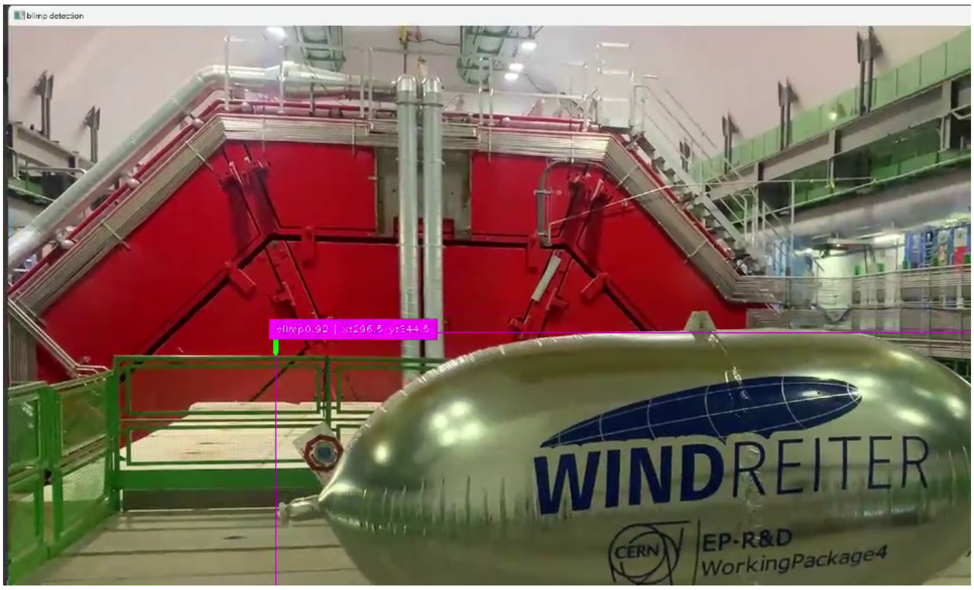


Figure 17: Example prediction result of model “20240806-py”: blimp side view recognised.

the new blimp from 2024 and the blimp taken from 2022, the model “20240807-py” can now identify the latest blimp design without any issues (18). The model exhibits high values for precision, recall, and F1 score, as shown in Fig. 26.

Model “20240808-py” The dataset size of model “20240807-py” was doubled by model “20240808-py”. This model can identify partially visible blimps and blimps in low-light conditions more accurately (higher recall), as shown in Fig. 19. The blimp’s confidence rate is roughly 0.9, which is comparable to that of model “20240807-py”. It performs exceptionally well and exhibits high performance metrics (Fig. 27).

3.2 Onboard radiation monitoring with MiniPIX TPX3 detector

The MiniPIX TPX3 detector needs to be integrated with the blimp onboard electronics to be used for radiation monitoring. The circuit schematic diagram of the blimp onboard electronics was studied to understand how to enable data flow from the detector to the on-board computer.

3.2.1 Blimp control communications

The circuit schematic diagram of the blimp onboard electronics is shown in Fig. 20. The main control board is an Arduino Nano 33 IoT, and each of the four propellers are connected to two DRV8835 motor controllers. The SD logger uses SPI protocol for data transfer, while the majority of sensors, such as the BN0055 inertial measurement unit (IMU), the VL53L4CXV00H/1 time of flight (TOF) laser sensor, and the LIS3MDLTR magnetic sensor, uses I2C protocol. The Arduino Nano’s pins are therefore all already in use. The detector communicates via USB and due to the amount of data transmitted, the cleanest solution is to connect it to a Raspberry Pi, that will be powered through the Arduino.

Figure 18: Example prediction result of model “20240807-py”: recognised blimp with different angles in same video input.

Figure 19: Example prediction result of model “20240808-py”: partial-displayed blimps are also recognized.

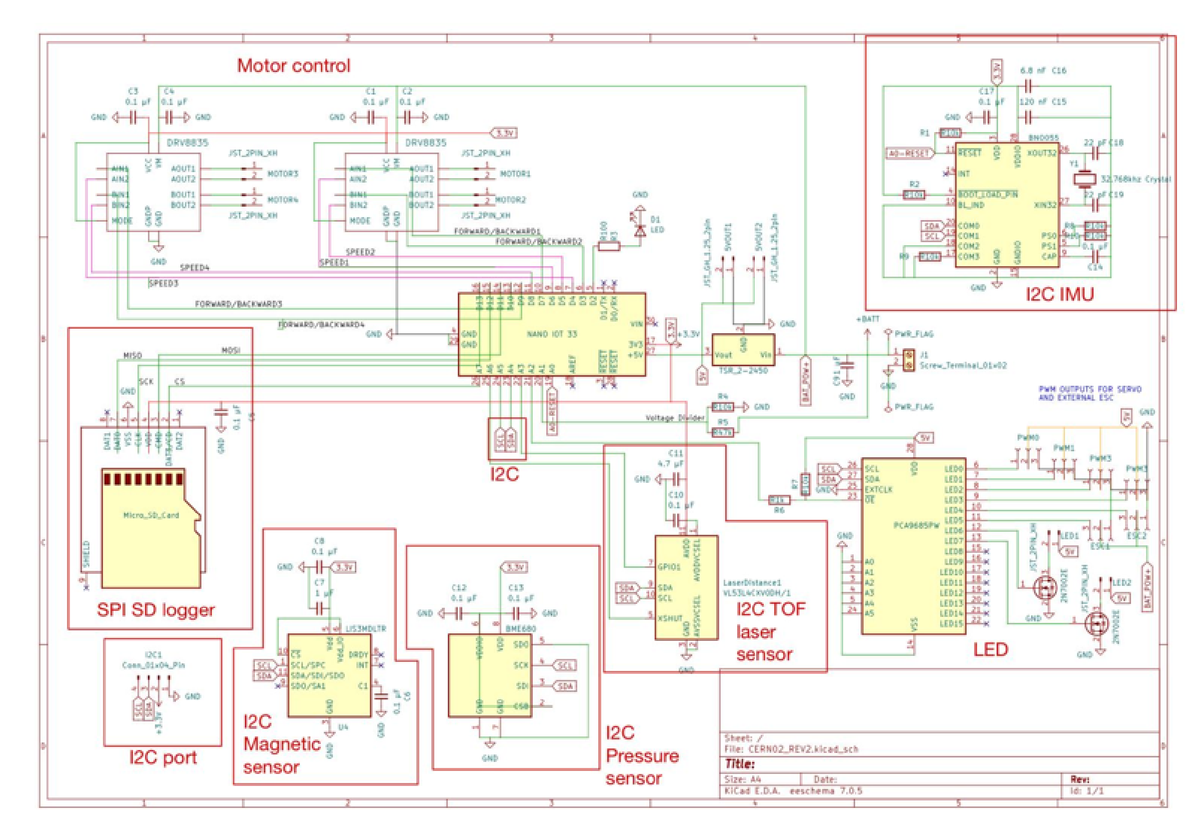


Figure 20: Schematic of the original blimp electronics.

3.2.2 Detector Readout and data acquisition

ADVACAM provides the detector readout and data acquisition for Windows, Mac and Linux operating systems. There are also API interface readouts for ARM32 and ARM64 architecture units. The "Pixet Pro" software was first tested on a Windows personal computer. With the "Pixet Pro" GUI software (Fig. 21), it is easy to perform complex measurements such as DAC Scans and Equalization tool for Threshold equalization and Clustering.

There are two API versions, one is based on C++, while the other on Python. Both versions exist for Windows, whereas only the Python version exists for ARM architecture. Both versions were successfully tested on Windows, enabling read-out in both frame and data-driven mode.

A Raspberry Pi 4B was used to test the Python version of the API for ARM architecture. The Raspberry Pi was used to test both data-driven and simple frame measurements, and both methods worked well. In frame mode, ToA and ToT subframes were saved as figures (Fig. 22) and ToA and ToT data from data-driven mode were stored in a text file (Listing 1). A procedure to convert ToA and ToT counter values to time and energy, respectively, still needs to be implemented.

Listing 1: Example of data-driven mode output file.

```
Index   Matrix Index    ToA ToT FToA    Overflow
0   19838   8135    1   16  0
1   19840   8135    1   15  0
```

```
2   19586   8136    1    29   0
3   19577   8135    1    12   0
4   19578   8135    1    16   0
5   19836   8135    1    16   0
6   18559   8134    46   2    0
7   18561   8135    1    15   0
8   18552   8135    1    10   0
9   18554   8135    1    17   0
10  19324   8135    38   17   0
11  18562   8136    1    29   0
12  17534   8135    1    12   0
13  17792   8135    1    14   0
14  18041   8135    1    11   0
15  17786   8135    1    13   0
16  17788   8135    1    14   0
...
```

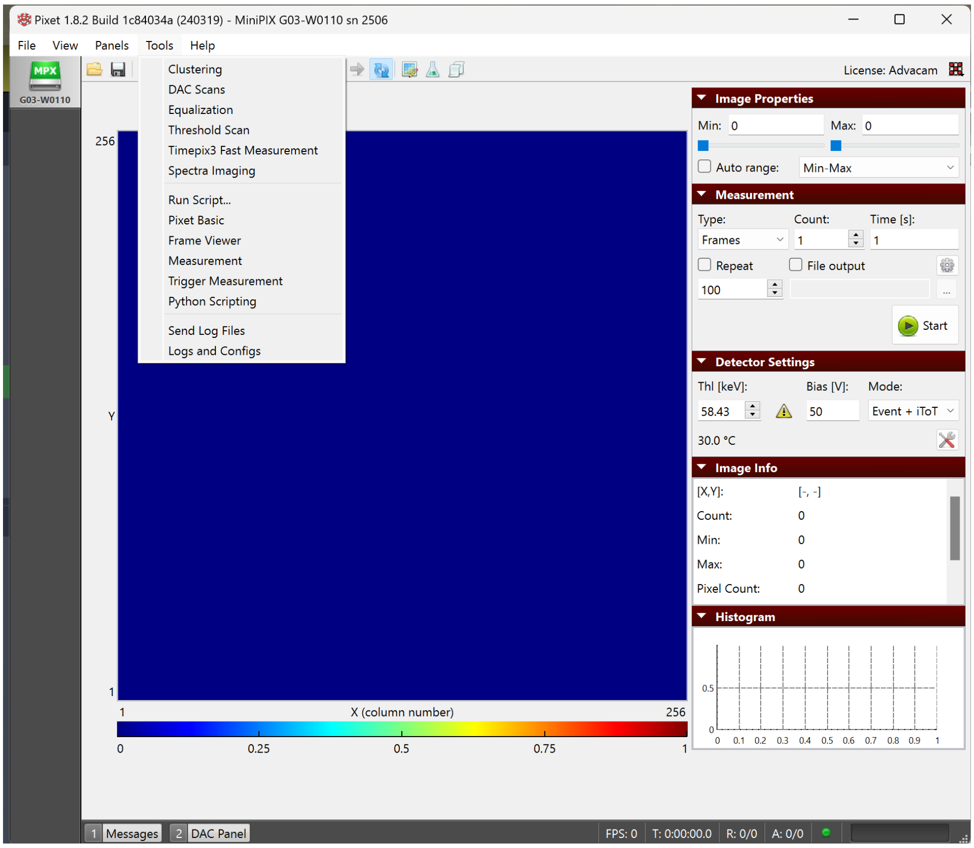
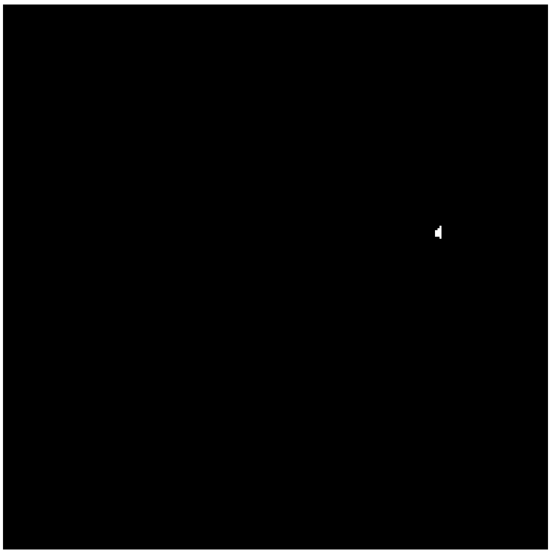
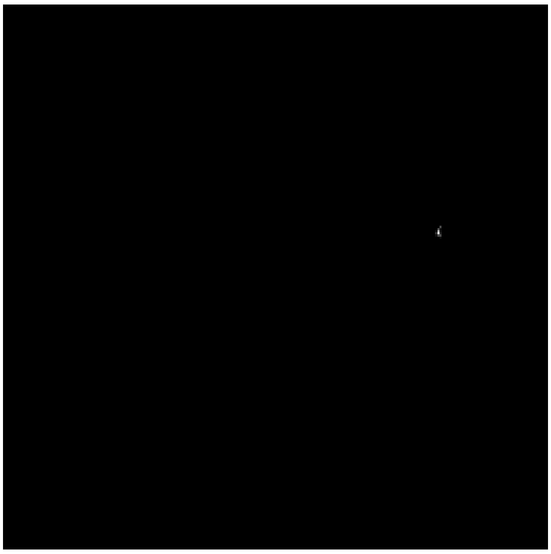


Figure 21: Photo of GUI "Pixet Pro" window with various tools such as DAC Scans, Equalization and Clustering.



(a) Time of arrival subframe.



(b) Time-over-threshold subframe.

Figure 22: Example of frame mode outputs.

4 Conclusion

4.1 Blimp localization system

Using YOLO for object detection, the model efficiently identifies the blimp in video frames with different backgrounds, minimizing on-board hardware needs. Models trained on large, generic datasets struggled with false positives. However, by refining the dataset with specific blimp images taken from CERN ALICE Cavern and increasing the sample size, significant improvements were achieved. The localization system uses triangulation with multiple camera views to accurately calculate the blimp's 3D position.

4.2 Onboard Radiation monitoring with MiniPIX TPX3 detector

The MiniPIX TPX3 detector was used to calculate the effective dose to safeguard both human personnel and equipment by identifying high-radiation areas. The system's setup involves the use of a Raspberry Pi for data acquisition from the detector due to the limitations of the blimp on-board computer, based on an Arduino board, to handle the MiniPIX data flow. Procedures to compute the dose rate from ToT measurements are being developed.

5 Future developments

5.1 Blimp localization system

Future work on the localization system will involve the development of the triangulation system. This will enable the calculation of the 3D position of the blimp inside the environment based on images from multiple cameras in which the YOLO model has correctly identified and classified the blimp.

To assess the accuracy of the localization system, there is the need to test the detection model with the blimp in the real operational environment. This can be first done in simulation inside a virtual environment of the cavern. In this case the accuracy of the system can be easily computed as the exact absolute position of the blimp is known and can be compared with that estimated by the system. To repeat this assessment in a real-world scenario, it is necessary to compare the absolute position estimated with the detection model with that estimated from another localization system with known accuracy, such as the PhaseSpace motion capture.

With the photo capture script available on the GitHub page, it is possible to automatically gather blimp images that can be used to enhance the blimp detection model by expanding the dataset size and to guarantee high performance in real operational situations. Indeed, enriching the dataset with images of the blimp under various lighting conditions and with different orientations would also be beneficial for the detection model.

Simultaneous detection from several camera viewpoints could improve the accuracy of the localization system, as it will help to reduce the discrepancy between the blimp center and the bounding box center

5.2 Onboard radiation monitoring with MiniPIX TPX3

Initial MiniPIX read-out was performed testing the Pixet Pro API on a Raspberry Pi 4B. Due to the blimp weight limitations, the MiniPIX will be integrated within the blimp on-board electronics using a Rasperry Pi Zero.

Furthermore, there is the need to implement a script, that will run on the Raspberry, to compute the dose rate from ToT measurements and stream the data to the ground control station and/or log them locally. Although a preliminary procedure to achieve this goal was described in Sec. 2.2.3, several refinements are still required. The procedure also needs to be validated against measurements acquired with dosimeters and the stability of the communication with the ground control station needs to be verified.

6 Acknowledgements

I would like to thank my supervisors, Francesco Mazzei, Paolo Francesco Scaramuzzino and Luca Bernardi, for their help on my projects. I would also like to thank CERN and the Hong Kong University of Science and Technology for their support throughout the summer.

References

[1] F. Mazzei, L. Teofili, et al. Mission analysis, dynamics and robust control of an indoor blimp in a CERN detector magnetic environment. *Frontiers in Robotics and AI*, 10, 2023.

[2] A. Abada, M. Abbrescia, et al. FCC-Hh: the hadron collider: future circular collider conceptual design report volume 3. Technical report, Future Circular Collider, Geneva, 2018.

[3] EP-R&D Programme, `https://ep-dep.web.cern.ch/rd-experimental-technologies`.

[4] J. Redmon, S. Divvala, et al. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[5] Juan Terven, Diana-Margarita Córdova-Esparza, et al. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.

[6] Ultralytics. *YOLOv8*, July 2024. Ultralytics YOLO Docs.

[7] Y. Kniazieva. Object Detection: Key metrics for Computer Vision model performance, August 2024.

[8] OpenCV. Triangulation, 2024. Open Source Computer Vision.

[9] Y.-S. Yang. Measurement of dynamic responses from large structural tests by analyzing non-synchronized videos. *Sensors*, 19:3520, 2019.

[10] Indirect interception `https://imagine.enpc.fr/~moulonp/openMVG/coreFeatures.html`.

[11] ADVACAM. MiniPIX TPX3 - miniaturized low-power radiation camera, 2024. `https://advacam.com/camera/minipix-tpx3/`.

[12] T. Poikela, J. Plosila, et al. Timepix3: a 65k channel hybrid pixel readout chip with simultaneous toa/tot and sparse readout. *Journal of Instrumentation*, 9(05):C05013–C05013, 2014.

[13] Justus Braach, Eric Buschmann, Dominik Dannheim, and Peter Svihra. CERN Summer Student 2024 Lab Exercise: Characterisation of Silicon Pixel Detectors, 2024.

[14] J. Jakubek. Precise energy calibration of pixel detector working in time-over-threshold mode. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 633:S262–S266, 2011.

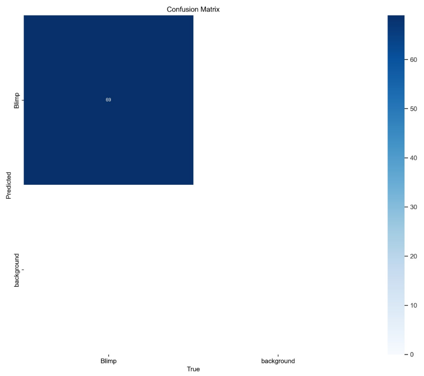
[15] ADVACAM. Detector G03-W0110 - Calibration Protocol, June 2024. Unpublished document.

[16] M. Platkevic, J. Jakubek, et al. Evaluation of local radiation damage in silicon sensor via charge collection mapping with the timepix read-out chip. *Journal of Instrumentation*, 8(4):C04001, 2013.

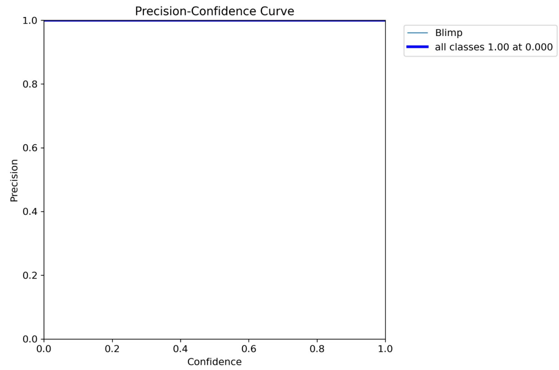
[17] D. Turecek, L. Pinsky, et al. Small dosimeter based on timepix device for international space station. *Journal of Instrumentation*, 6(12):C12037–C12037, 2011.

A Performance Metrics

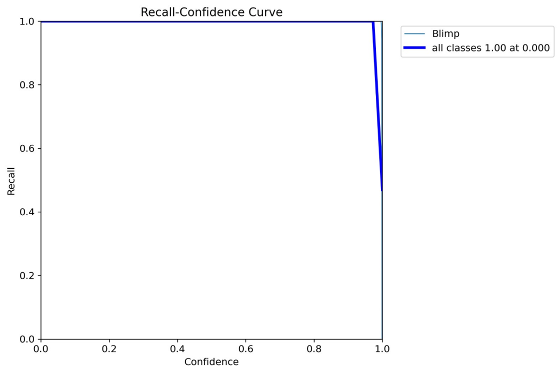
A.1 Model “20240731-py”



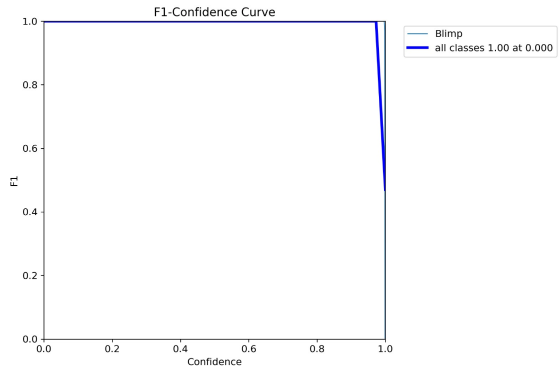
(a) Confusion Matrix.



(b) Precision-Confidence Curve.



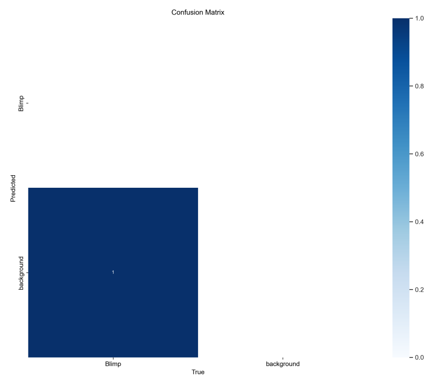
(c) Recall-Confidence Curve.



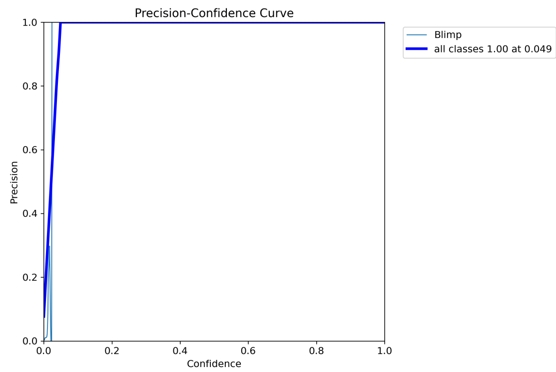
(d) F1-Confidence Curve.

Figure 23: Model “20240731-py” performance metrics.

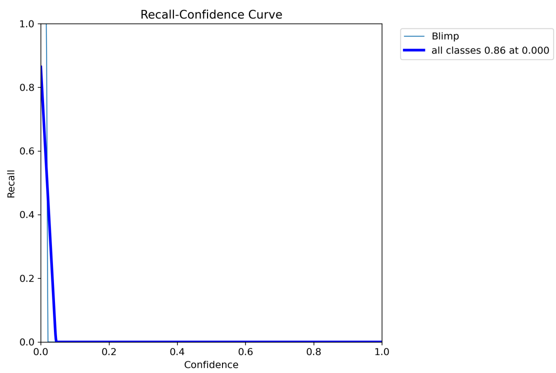
A.2 Model "20240802-py"



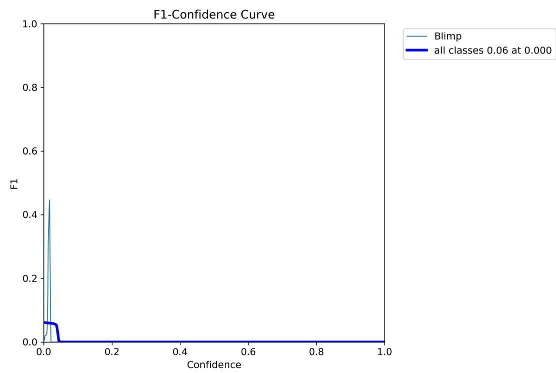
(a) Confusion Matrix.



(b) Precision-Confidence Curve.



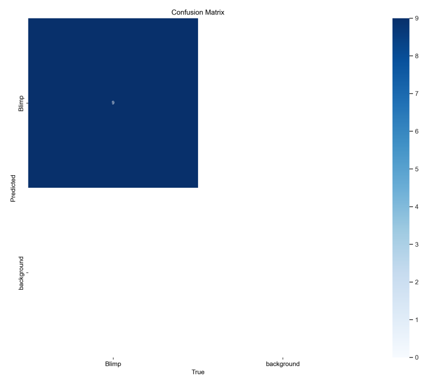
(c) Recall-Confidence Curve.



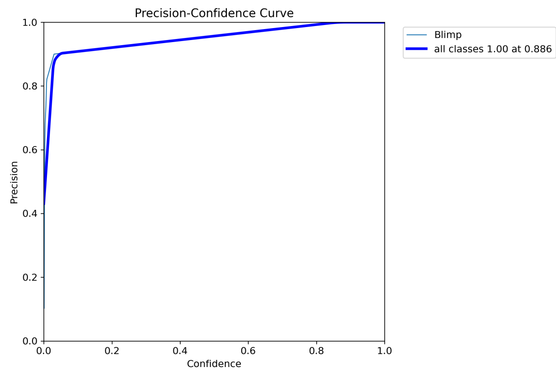
(d) F1-Confidence Curve.

Figure 24: Model "20240802-py" performance metrics.

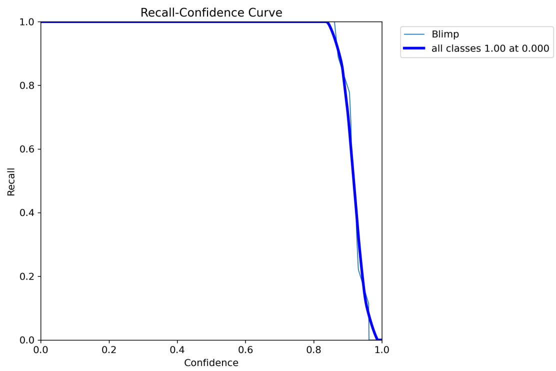
A.3 Model "`20240806-py`"



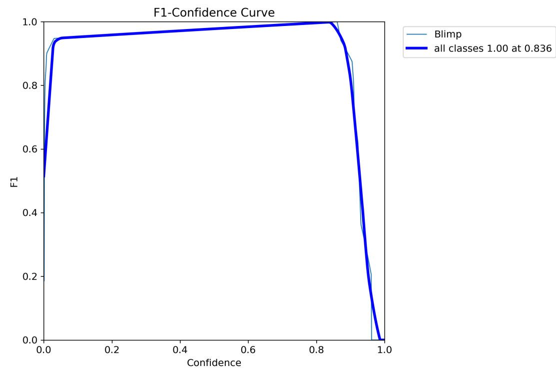
(a) Confusion Matrix.



(b) Precision-Confidence Curve.



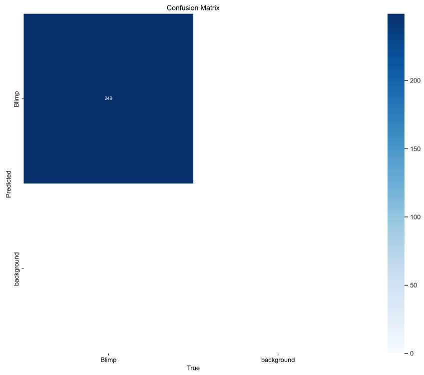
(c) Recall-Confidence Curve.



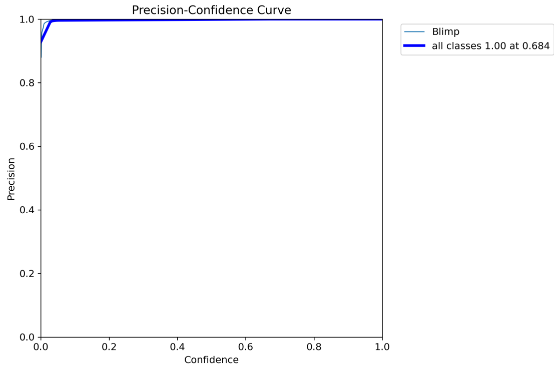
(d) F1-Confidence Curve.

Figure 25: Model "`20240806-py`" performance metrics.

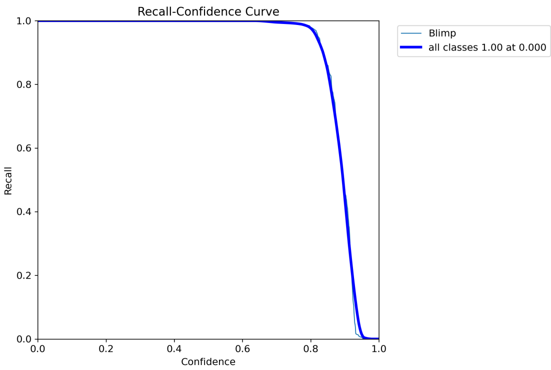
A.4 Model “20240807-py”



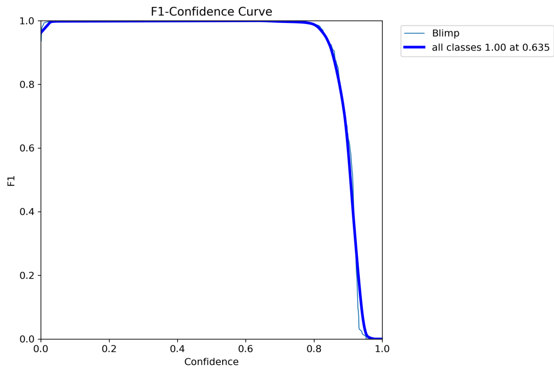
(a) Confusion Matrix.



(b) Precision-Confidence Curve.



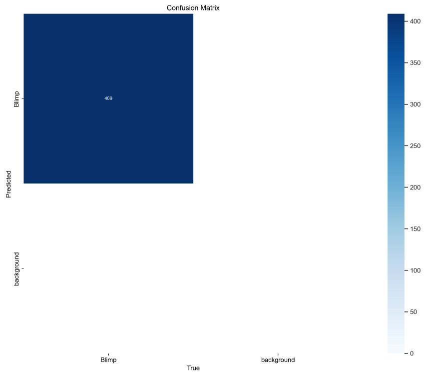
(c) Recall-Confidence Curve.



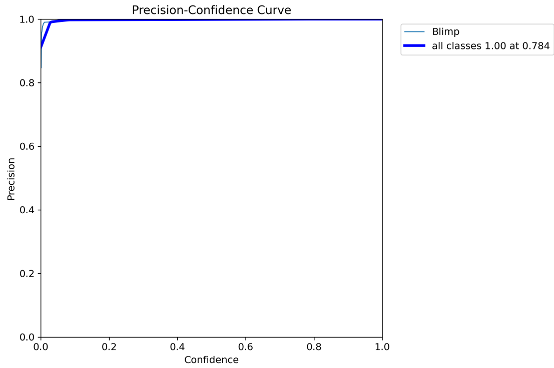
(d) F1-Confidence Curve.

Figure 26: Model “20240807-py” performance metrics.

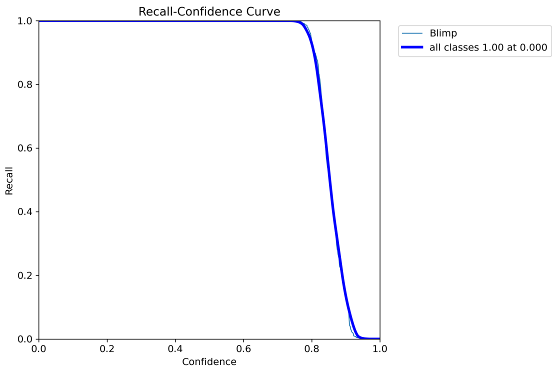
A.5 Model “`20240808-py`”



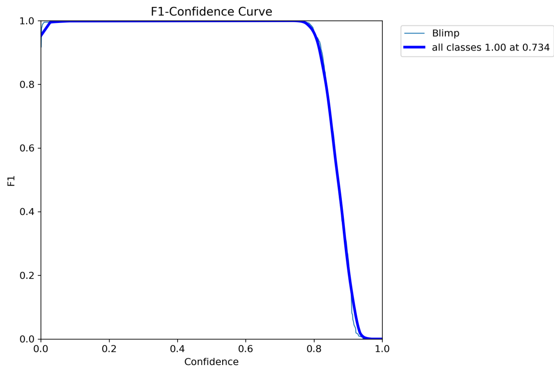
(a) Confusion Matrix.



(b) Precision-Confidence Curve.



(c) Recall-Confidence Curve.



(d) F1-Confidence Curve.

Figure 27: Model “`20240808-py`” performance metrics.