drones

MDPI

Article

Real-Time Search and Rescue with Drones: A Deep Learning Approach for Small-Object Detection Based on YOLO

Francesco Ciccone and Alessandro Ceruti *

Department of Industrial Engineering—DIN, University of Bologna, Viale Risorgimento 2, 40132 Bologna, Italy; francesco.ciccone2@unibo.it

* Correspondence: alessandro.ceruti@unibo.it

Abstract

Unmanned aerial vehicles are increasingly used in civil Search and Rescue operations due to their rapid deployment and wide-area coverage capabilities. However, detecting missing persons from aerial imagery remains challenging due to small object sizes, cluttered backgrounds, and limited onboard computational resources, especially when managed by civil agencies. In this work, we present a comprehensive methodology for optimizing YOLO-based object detection models for real-time Search and Rescue scenarios. A two-stage transfer learning strategy was employed using VisDrone for general aerial object detection and Heridal for Search and Rescue-specific fine-tuning. We explored various architectural modifications, including enhanced feature fusion (FPN, BiFPN, PB-FPN), additional detection heads (P2), and modules such as CBAM, Transformers, and deconvolution, analyzing their impact on performance and computational efficiency. The best-performing configuration (YOLOv5s-PBfpn-Deconv) achieved a mAP@50 of 0.802 on the Heridal dataset while maintaining real-time inference on embedded hardware (Jetson Nano). Further tests at different flight altitudes and explainability analyses using EigenCAM confirmed the robustness and interpretability of the model in real-world conditions. The proposed solution offers a viable framework for deploying lightweight, interpretable AI systems for UAV-based Search and Rescue operations managed by civil protection authorities. Limitations and future directions include the integration of multimodal sensors and adaptation to broader environmental conditions.

Keywords: search and rescue; unmanned aerial vehicles; small-object detection; YOLO; model optimization; real-time inference

Academic Editors: Kristian Amadori and Christopher Jouannet

Received: 11 June 2025
Revised: 10 July 2025
Accepted: 18 July 2025
Published: 22 July 2025

Citation: Ciccone, F.; Ceruti, A. Real-Time Search and Rescue with Drones: A Deep Learning Approach for Small-Object Detection Based on YOLO. *Drones* **2025**, *9*, 514. https://doi.org/10.3390/drones9080514

1. Introduction

Climate change has led to a sharp increase in the frequency and intensity of natural disasters such as wildfires, landslides, and floods, posing serious challenges for Search and Rescue operations [1,2] managed by civil authorities. Rapid localization of people in life-threatening conditions is critical to saving lives, particularly when individuals are trapped, injured, or isolated in inaccessible areas due to environmental disasters or health issues. Aerial platforms such as helicopters and unmanned aerial vehicles (UAVs) have become central tools in Search and Rescue missions, enabling fast coverage of vast territories and high-resolution data acquisition [3–5].

Despite these advantages, manually analyzing aerial imagery remains slow, error-prone, and cognitively demanding for human operators; moreover, it requires a complex training that is quite difficult to ensure in the case of people who should carry out several

tasks in an organization. Fatigue, attentional drift, and the overwhelming number of uninformative frames can compromise mission effectiveness. These limitations highlight the need for intelligent vision systems to detect people from aerial images in real time. In recent years, deep learning-based object detectors, particularly the YOLO (You Only Look Once) family, have shown promising performance in this context [6].

Unmanned aerial vehicles (UAVs), or drones, are increasingly employed in Search and Rescue missions in the civil field because of their ability to survey large and complex areas rapidly. In both wilderness and urban scenarios, their effectiveness hinges on the capacity to detect small, sparsely located human figures from aerial imagery; a task that remains highly challenging due to occlusions, scale variation, cluttered backgrounds, and limited onboard computational resources. Recent surveys such as Zhang et al. [7] provide a comprehensive review of aerial person detection (APD), identifying key challenges related to target size, sparsity, and background complexity. Their introduction of the VTSaR dataset, which includes multimodal data (visible/IR), reinforces the growing need for robust and efficient models tailored to drone-based applications. In this direction, PS-YOLO [8] demonstrates how lightweight and efficient designs can maintain accuracy while remaining deployable on resource-constrained UAVs. Also, from an economical point of view, which is important for civil applications, the hardware resources needed are more than affordable.

Further advances include Yeom (2024) [9], who integrates YOLOv5 with Kalman filters for multi-target tracking in thermal imagery, revealing both the potential and limitations of thermal-based Search and Rescue, particularly concerning resolution loss and scene complexity. Similarly, Kucukayan and Karacan (2024) [10] improve indoor human detection by embedding attention mechanisms into YOLO-based models (YOLO-IHD), highlighting the need for context-specific architectural adaptation. Outside of vision, Calabrò and Marchetti (2024) [11] introduce Transponder, a Wi-Fi-based localization system that integrates with UAVs for non-GSM areas, exemplifying how multimodal systems can enhance Search and Rescue capabilities.

Nevertheless, the gap between academic performance and field validation remains significant. Manzini and Murphy (2023) [12] report in a real Search and Rescue that models like EfficientDet, despite strong performance on benchmarks, failed in deployment due to high false positives and unreliable predictions.

One of the root causes of these limitations is the lack of well-suited datasets. Generic datasets such as COCO [13] or ImageNet [14] are inadequate for aerial person detection. VisDrone [15] provides partial support but lacks domain specificity. Newer datasets, like Heridal [16], offer high-resolution UAV footage focused on Search and Rescue use cases, helping to address issues such as small-object sparsity, high-class imbalance, and large-scale background clutter. Nonetheless, further methodological improvements are needed.

Detecting small and sparse targets in cluttered aerial scenes presents specific architectural challenges. The spatial resolution of deeper convolutional layers is often too low to capture tiny objects effectively. Models must, therefore, integrate multi-scale feature fusion techniques such as FPN [17], BiFPN [18], or PB-FPN [19] and benefit from attention mechanisms like CBAM [20] and RFCBAM [21], which allow the network to focus on informative regions. Data augmentation strategies, such as rotation, mosaic, and copy–paste, also play a critical role in expanding the diversity and frequency of positive examples [22,23]. Other optimizations, such as anchor box tuning, oversampling, and the use of transformer modules [24], further contribute to performance gains in low-data or small-object contexts.

Recent developments in UAV-based Search and Rescue have increasingly leveraged lightweight object detection frameworks within realistic operational contexts. Notably, Dumenčić et al. (2025) [25] present an experimental validation of a YOLO-powered UAV detection system deployed in a wilderness environment, demonstrating how the integration

of ergodic search strategies with onboard inference can robustly detect human targets under real-world conditions. Similarly, Fu et al. (2024) [26] introduce DLSW-YOLOv8n, a customized small-object detection model optimized for maritime SAR imagery captured by UAVs; the work illustrates that careful architectural design (e.g., deformable large kernel and SPD-Conv layers) can significantly enhance detection accuracy in challenging maritime environments. Complementing this, Weng et al. (2025) [27] propose YOLO-SRMX, a lightweight real-time detector tailored to infrared UAV imagery, showcasing effective small-object detection while preserving onboard efficiency. These studies collectively underscore the practical importance of balancing accuracy, model size, and deployment feasibility in SAR applications, reinforcing the relevance of our architectural modifications and performance evaluations.

This work presents a comprehensive methodology to adapt and optimize the YOLOv5 object detection model for real-time aerial Search and Rescue missions in the civil domain. We propose architectural modifications such as enhanced multi-scale feature fusion and lightweight detection heads to improve small-object sensitivity while preserving computational efficiency. By fine-tuning the model through transfer learning on the VisDrone and Heridal datasets, we substantially boost mAP over the YOLOv5 baseline. The generalizability of our approach is further demonstrated by successfully porting the methodology to YOLOv8, showcasing its scalability and robustness across architectures and mission profiles.

The main contributions of this work are as follows:

1. The design of a lightweight and accurate object detection model for small-object recognition in aerial Search and Rescue imagery based on YOLOv5s;
2. The integration of advanced architectural modules (e.g., feature fusion and attention mechanisms) to enhance detection performance;
3. A comprehensive evaluation of benchmark UAV datasets, including VisDrone and Heridal, demonstrating superior accuracy and real-time suitability;
4. The validation of the proposed methodology on YOLOv8, confirming its generalizability across architectures.
5. A discussion of practical deployment aspects, including real-world constraints, dataset handling, and embedded implementation for UAV-based Search and Rescue platforms, focusing attention on civil applications, where cost should be kept as low as possible and the usability for operators must be considered.

2. Materials and Methods

To address the specific challenges of Search and Rescue operations and enable practical deployment of object detection systems on embedded aerial platforms, we developed a comprehensive methodology focused on the algorithmic optimization and hardware implementation of YOLO-based models. Our approach begins with selecting and adapting YOLOv5 and YOLOv8 architectures, targeting the detection of small, sparsely distributed human figures in complex aerial scenes.

The methodology involved three tightly integrated components (Figure 1):

1. Dataset selection and preparation, applying pre-processing steps such as resizing, bounding box selection, and splitting the dataset into train, validation, and test sets and applying data augmentation techniques like rotation, flipping, mosaic, copy-paste of small objects;
2. Architectural customization of the base model, including the integration of enhanced feature fusion layers, additional detection heads, and attention modules to improve small-object sensitivity without significantly increasing computational complexity;

3. transfer learning and dataset selection using aerial-specific datasets such as Heridal and VisDrone to ensure the model is exposed to realistic Search and Rescue-like conditions during training and validation;
4. Embedded deployment optimization, involving evaluating and adapting the trained models for execution on constrained hardware platforms (NVIDIA Jetson Nano), with particular attention to inference time, memory usage, and energy efficiency.

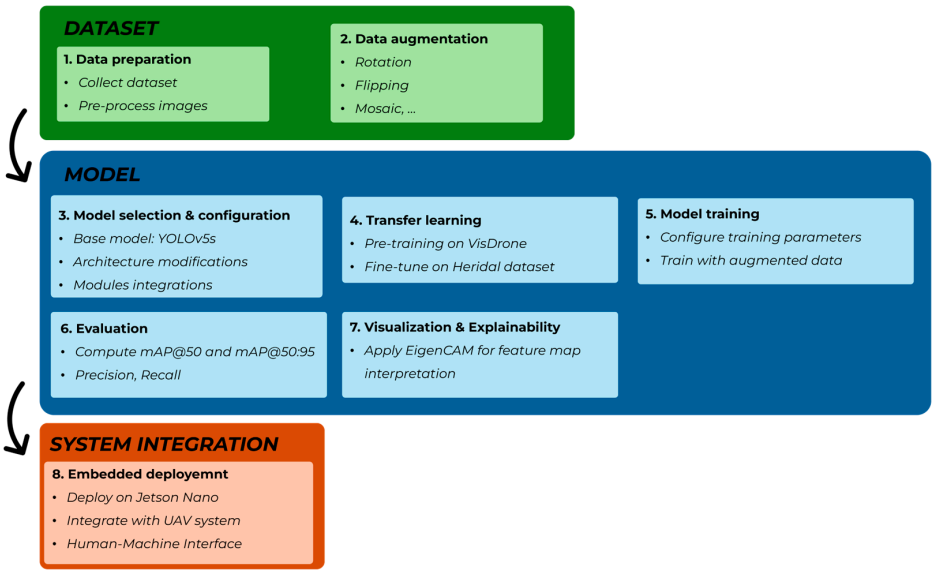


Figure 1. The workflow of the proposed Search and Rescue object detection approach. The process starts with dataset preparation and augmentation, followed by model selection and architectural modifications. After transfer learning and training, the model is evaluated using standard metrics. The final optimized model is deployed on embedded hardware for UAV-based inference.

The architectural changes were systematically tested across multiple configurations, including variations in input resolution, depth, and width multipliers, to assess the trade-off between performance (in terms of precision and recall) and computational load. Additionally, the pipeline incorporates pre- and post-processing steps suitable for onboard execution, such as image resizing strategies, lightweight data augmentation, and non-maximum suppression tuning.

By integrating model-level optimization with hardware-aware constraints, this methodology aims to bridge the gap between high-performing object detectors in controlled environments and their real-world applicability in time-critical Search and Rescue missions conducted by UAVs.

2.1. Dataset

To train a model capable of detecting missing persons in aerial Search and Rescue operations after environmental disasters or emergencies, it is essential to use datasets that reflect the characteristics of real-world scenarios. A typical Search and Rescue civil context involves open-field environments as forests, agricultural fields, or mountainous terrain, where people may appear occluded, lying down, or occupying only a few pixels due to altitude and resolution.

For this purpose, we selected the Heridal dataset, explicitly designed for Search and Rescue missions. It includes various scenes with diverse lighting conditions, heterogeneous terrain types, and multiple human poses. All images were captured from altitudes between 40 and 65 m, resulting in tiny human instances. Each frame included at least one person, making the dataset particularly useful but also highly challenging due to the presence of false positives from natural artifacts and cluttered backgrounds.

We adopted a transfer learning approach to address the scarcity of positive samples and improve generalization. The first training step was conducted using the VisDrone dataset, which contains drone-acquired images and offers many small-object instances. Transfer learning was then applied to Heridal. A further experiment using the Stanford Drone Dataset [28] as an intermediate fine-tuning step was performed, but its benefits were marginal and did not justify the increased training time. We thus adopted a two-stage training strategy: VisDrone pretraining followed by Heridal fine-tuning. Figure 2 shows examples of the VisDrone and Heridal datasets in urban and forestal scenarios.

(a) **(b)**

Figure 2. (**a**) Examples from the VisDrone dataset; (**b**) examples from the Heridal dataset.

2.2. Model Architecture and Modifications

YOLO is widely recognized as a state-of-the-art object detection framework for real-time applications. Although newer versions like YOLOv9 [29], YOLOv10 [30], YOLOv11 [31], and YOLOv12 [32] have been released recently, in this study, we selected YOLOv5 for several reasons. First, YOLOv5 provides a mature and stable implementation with extensive documentation and an active developer community, facilitating the integration of custom modules such as additional detection heads, feature fusion layers, and attention mechanisms. Second, YOLOv5 has established compatibility with NVIDIA TensorRT and ONNX export pipelines, which is essential for efficient deployment on embedded systems like the Jetson Nano. Third, preliminary experiments conducted as part of this work demonstrated that applying the same architectural modifications to YOLOv8 yielded comparable detection accuracy (mAP@50 and mAP@50:95) to YOLOv5 on our aerial Search and Rescue dataset while requiring a higher computational burden. YOLOv5 allows for architectural definition, supporting changes in model depth, width, number of classes, and specific modules in the backbone, neck, and head. YOLOv5s, the smallest full-scale version, was selected for its balance between accuracy and efficiency. It employs a CSPDarknet53 backbone, an SPPF (Spatial Pyramid Pooling-Fast) module, a PANet neck, and a YOLOv3 anchor-based detection head. Figure 3 shows the authors' representation of YOLOv5 architecture.

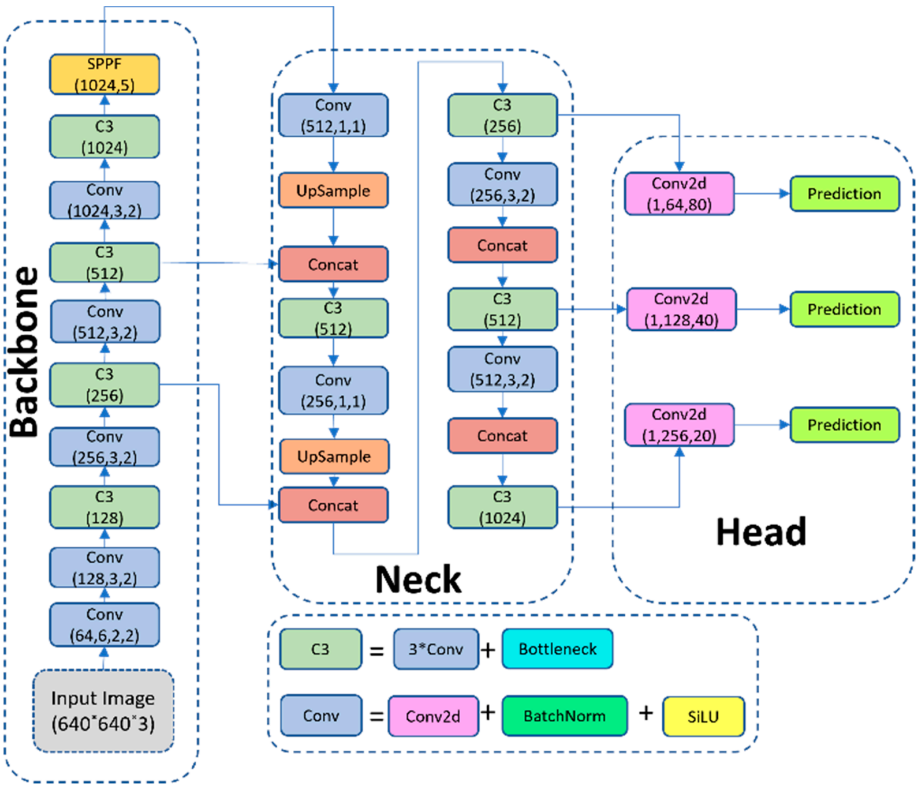


Figure 3. The default Yolov5 network structure.

Because of the application of the convolutions to the input image, as the number of layers increases, the image's resolution decreases. If we denote with P_i the feature level at layer "i" (Figure 4), the corresponding resolution will be $1/2^i$ of the input image [18]. Considering an input resolution of 640 × 640, the resolution of the fourth feature level P_4 will be $\left(640/2^4 = 40\right)$ 40 × 40. Even if the deeper layers are rich in semantic features, the shallow layers dominate higher resolutions; in this context, combining the different feature layers (adequately resized) to enhance small-object detection is crucial.

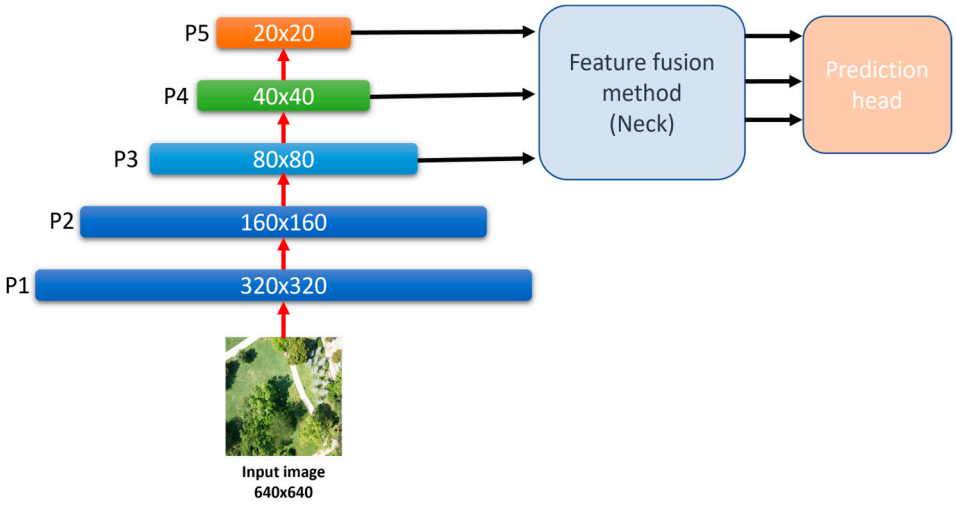


Figure 4. Different feature levels in YOLOv5.

We investigated the impact of different feature fusion strategies on small-object detection by replacing PANet with FPN, BiFPN, and PB-FPN [19]. FPN and PB-FPN yielded superior results on the Heridal dataset, likely due to their top-down fusion paths, which better preserve high-resolution features from shallow layers. Conversely, BiFPN and PANet,

which use bottom-up fusion, underperformed. Figure 5 shows the schemes of the feature fusion methods used.

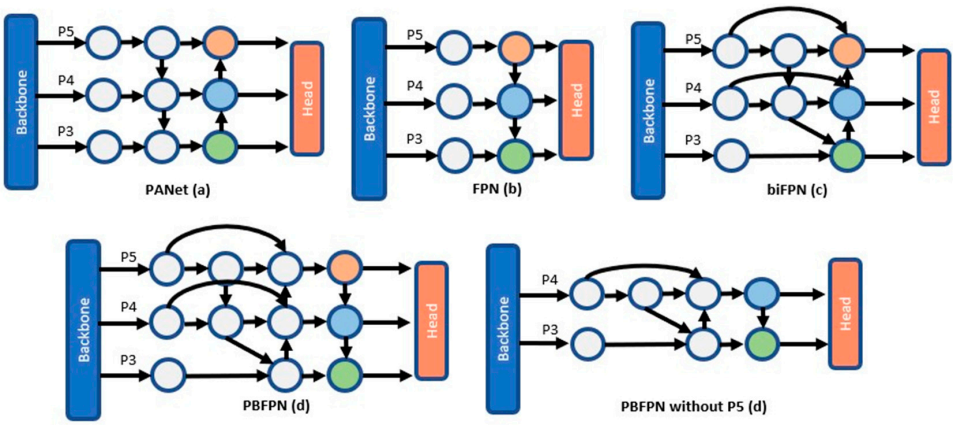


Figure 5. Different feature fusion methods are compared.

Additionally, we experimented with adding extra detection heads to improve sensitivity to small objects and analyzed the integration of Transformer Prediction Heads (TPH) and CBAM (Convolutional Block Attention Module), inspired by TPH-YOLOv5 [33]. While TPH-YOLOv5 did not significantly improve Heridal, using CBAM and Transformer blocks separately showed more promising results in detecting occluded or tiny targets.

2.3. Training Strategy and Hyperparameters Tuning

The training process adopted in this study was designed to maximize both generalizations across diverse aerial scenarios and specialization for search and rescue environments after environmental disasters. The approach was centered on a two-step transfer learning strategy, consisting of a pretraining phase on the VisDrone dataset, targeted at general small-object detection from aerial views, followed by fine-tuning on the Heridal dataset, which contains realistic Search and Rescue-specific imagery. This curriculum-based learning approach led to a 7.82% improvement in mAP@50 over models trained directly on Heridal, as demonstrated in Section 3.

Training was conducted using Stochastic Gradient Descent (SGD) with the following parameters: an initial learning rate (lr0) of 0.01, momentum of 0.937, and weight decay of 0.0005. A OneCycleLR scheduler was employed, with a final learning rate factor (lrf) of 0.1. The initial phase included a 3-epoch warmup, with linearly increasing momentum and bias learning rates (warmup momentum = 0.8, warmup bias_lr = 0.1) to ensure stable convergence.

All experiments were executed at 640 × 640 input resolution, with a batch size of 16 and early stopping after 50 epochs of stagnant validation performance. Training proceeded for a maximum of 300 epochs per configuration. A 70/20/10 train/validation/test split was used across all datasets, with consistent seed initialization to ensure reproducibility.

YOLOv5 supports a wide range of "bag of freebies" options, training improvements that do not increase inference cost [34]. Several of these techniques were leveraged, including the following:

- CutMix [35] and MixUp [36] for advanced data augmentation that encourages generalization by blending samples and their labels;
- Modified IoU-based loss functions, including GIoU, DIoU, and CIoU [37,38], to improve bounding box regression accuracy;
- Customization of the configuration files to adjust hyperparameters such as learning rate, batch size, mosaic probability, and anchor box scaling.

Additional augmentation strategies included HSV augmentation (hue: 0.015, saturation: 0.7, value: 0.4), image translation (±10%), scaling (±10%), left–right flipping (50%), and segment-level augmentations such as copy–paste (10%) and mosaic (100%). While techniques like perspective distortion and up–down flipping were disabled (perspective = 0.0, flipud = 0.0) due to limited applicability in aerial imagery, these carefully tuned augmentations reduced overfitting and improved performance in complex cases.

Anchor boxes were re-optimized for the Heridal dataset using the built-in K-means clustering with a genetic algorithm, adapting to the real-world distribution of object sizes and improving localization, particularly for small, partially occluded figures.

Finally, loss weighting was manually calibrated to reflect the detection task's priorities: box loss = 0.05, objectness = 0.7, and class loss = 0.3, with all positive class/object weights set to 1.0. The training IoU threshold (iou_t) was set to 0.20 to allow the model to consider even loosely overlapping predictions, which is particularly beneficial in low-resolution, cluttered scenes typical of Search and Rescue imagery in scenarios of interest for civil protection.

This comprehensive training pipeline enabled robust convergence, minimized overfitting (as later discussed), and yielded models capable of generalizing well across datasets while remaining computationally efficient for deployment on embedded UAV systems; all the details of the training phase hyperparameters are collected in Table 1.

Table 1. Detailed hyperparameter values were used during the training phase.

Category	Parameter	Value	Description
Optimizer	Optimizer	SGD	Stochastic Gradient Descent
	lr0	0.01	Initial learning rate
	lrf	0.1	Final learning rate (OneCycleLR multiplier)
	momentum	0.937	SGD momentum
	weight_decay	0.0005	Weight decay
Image Augmentation	hsv_h	0.015	HSV hue variation
	hsv_s	0.7	HSV saturation variation
	hsv_v	0.4	HSV value variation
	translate	0.1	Translation
	scale	0.9	Scaling (±gain)
	fliplr	0.5	Horizontal flip (probability)
	mosaic	1.0	Mosaic augmentation (probability)
	mixup	0.1	MixUp augmentation (probability)
	copy_paste	0.1	Segment copy-paste (probability)
Training Strategy	epochs	400	Maximum number of epochs
	batch_size	32	Batch size
	early_stopping	50	Early stopping patience (no improvement)
	image_size	640 × 640	Input resolution
	anchor_auto	Enabled	Anchor box recalculation via K-means + GA
	split	70/20/10	Train/val/test dataset split
	seed	Fixed	Reproducibility
Transfer Learning	pretrain_dataset	VisDrone	Generic small-object pretraining (only "person" class)
	finetune_dataset	Heridal	Search and Rescue specific fine-tuning

2.4. Embedded Deployment Setup

We implemented and tested the entire system on an embedded UAV platform to validate the feasibility of deploying person detection models for real-time Search and Rescue operations. The objective was to enable onboard AI inference for detecting missing persons from aerial imagery and provide real-time communication with a ground station through a Human–Machine Interface (HMI). This section describes in detail the hardware

and software setup, the communication infrastructure, and the functionalities integrated into the GUI for field deployment.

2.4.1. UAV and Embedded Hardware Setup

The embedded system was integrated into a Holybro X500 V2 quadrotor platform equipped with a Pixhawk 6C flight controller. This drone was selected for its payload capacity, flight stability, and modular design, making it well suited for testing embedded AI systems in Search and Rescue civil scenarios. The Holybro X500 V2 features a carbon-fiber frame with a 500 mm wheelbase, providing a maximum payload capacity of approximately 1.5 kg. The Pixhawk 6C was configured with PX4 firmware (v1.13) to enable MAVLink telemetry and reliable autonomous flight control.

The onboard AI inference was performed by an NVIDIA Jetson Nano (4 GB), selected for its compact form factor, low power consumption, and CUDA-capable 128-core GPU, enabling deep learning inference directly onboard the UAV. The Jetson Nano was powered by a dedicated 5 V power bank battery. The system ran Ubuntu 18.04 with JetPack 4.6 and CUDA 10.2. The Jetson Nano was connected to a USB camera capable of capturing 1080p video at 30 FPS. Video acquisition was implemented using the V4L2 driver and OpenCV (v4.5).

Wireless connectivity was provided by a TP-Link Archer T2U Plus Wi-Fi module configured in Access Point mode, creating a dedicated peer-to-peer Wi-Fi network between the Jetson Nano and the ground station.

The ground station was a laptop running Windows 10, executing a custom Python (v3.9) application capable of receiving the video stream, displaying detection results in real time, and sending commands to the Jetson Nano to start or stop inference, adjust model parameters, or control the data flow.

All components were securely mounted on the UAV in a manner that preserved flight stability and minimized electromagnetic interference. The hardware setup is summarized in Table 2.

Table 2. Hardware specifications.

Component	Specification
Drone	Holybro X500 V2 (500 mm wheelbase) with Pixhawk 6C, PX4 v1.13
Computing Unit	NVIDIA Jetson Nano (4 GB RAM, 128-core GPU, JetPack 4.6)
Wi-Fi Module	TP-Link Archer T2U Plus
Camera	1080p ELP USB Camera, 30 FPS

2.4.2. Human–Machine Interface (HMI)

To interact with the onboard AI system, a custom Graphical User Interface (GUI) was developed in Python and deployed on the ground station. The GUI was designed to offer task-specific modules for Search and Rescue, focusing on person detection from aerial video. The GUI was developed to support firefighters or civil protection agencies: the interface prioritizes usability for non-technical operators while supporting advanced AI functionalities such as:

- Establishing connections with the drone and Jetson Nano;
- Streaming real-time video from the onboard camera;
- Sending commands to trigger AI inference;
- Receiving and visualizing detection results, including bounding boxes and metadata;
- Saving frames or exporting inference outputs for post-mission analysis.

The GUI also integrates a real-time telemetry monitor that displays essential drone parameters such as GPS location, altitude, heading, and battery status.

2.4.3. Communication Architecture

The system employs a hybrid communication framework based on the following protocols:

- TCP is used for reliable transmission of control commands (e.g., start video, stop video, send image, shut down) and to request or retrieve inference results from the Jetson Nano;
- UDP handles real-time video and telemetry streaming due to its low-latency characteristics, which are crucial for maintaining situational awareness during flight.

This architecture ensures a reliable and responsive communication channel between the Jetson and the GUI, balancing speed and reliability based on the type of data being exchanged. Figure 6 shows the general architecture of the hardware and communication setup.

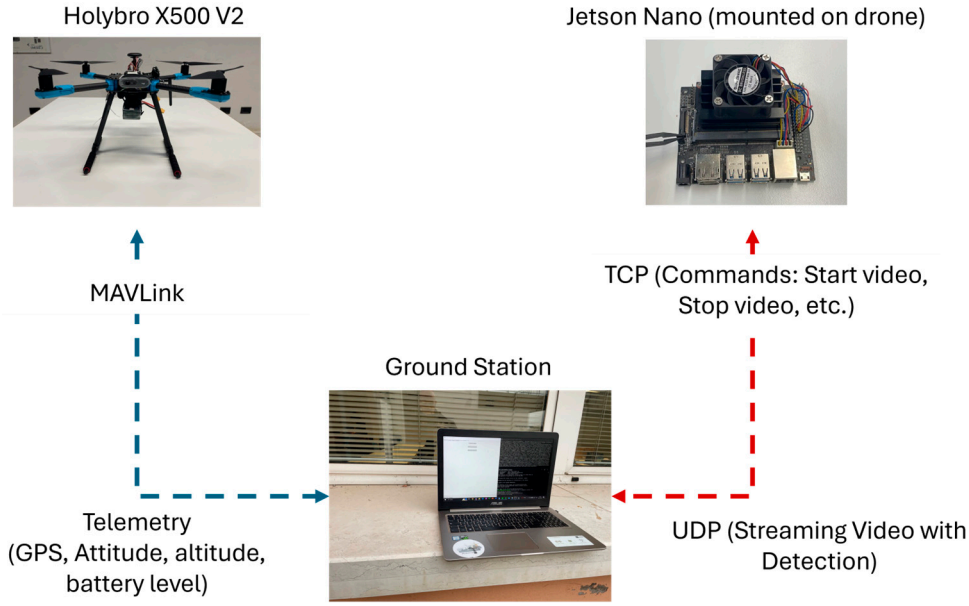


Figure 6. Hardware and communication setup.

2.4.4. Person Detection Workflow

The person detection workflow on the embedded UAV platform was structured to provide a reliable and responsive user experience, even under operational constraints typical of Search and Rescue civil missions. The process begins with the initialization phase, during which the ground station operator launches the GUI and establishes connections with both the UAV via telemetry and the Jetson Nano via TCP/UDP sockets. Once these connections are active, the system displays real-time video from the drone's onboard camera and relevant telemetry data, such as altitude, heading, and battery status. After initialization, the operator can activate the onboard inference system by directly selecting and launching the person detection model from the GUI. The Jetson Nano then applies the object detection model, identifies any instances of persons in the scene, and generates bounding boxes around detected individuals. The resulting frame, with overlaid detection information, is transmitted back to the ground station, where the GUI immediately displays this output to the operator and logs relevant metadata such as the number of detections, timestamp, and frame index, thereby ensuring traceability and facilitating post-mission review.

An optional feature allows the operator to save selected frames locally on the ground station. When enabled, this functionality stores inference results for subsequent analysis or reporting, enhancing the system's documentation and mission debriefing usability.

2.4.5. Performance and Optimization

The embedded system was optimized for reliable performance under resource constraints to meet the real-time requirements of aerial Search and Rescue applications for civil applications. A key goal was maintaining an inference rate of at least one to two frames per second, a threshold deemed sufficient given the nature of UAV motion in Search and Rescue contexts, which typically involves hovering or slow aerial scanning.

The architecture was implemented with multithreading to prevent system bottlenecks, where video streaming and inference were handled on separate execution threads. Multithreading ensured that the streaming process remained uninterrupted by the time-consuming operations associated with model inference.

Finally, to reduce redundant computations, the system supported on-demand inference triggering. In cases where continuous detection was not required, the operator could request person detection only when necessary, such as after identifying a potential target in the live video feed. This feature was handy in static or low-activity scenes, where continuous inference would waste computational resources without significantly improving mission outcomes.

2.5. Evaluation Metrics

- Model performance was evaluated using standard object detection metrics:

$$\text{Precision} = TP/(TP + FP) \qquad TP = \text{True Positive, } FP = \text{False Positive}$$

$$\text{Recall} = TP/(TP + FN) \qquad FN = \text{False Negative}$$

The training followed a two-step transfer learning process: first, training on VisDrone for general small-object recognition from aerial images, then fine-tuning on Heridal to adapt to Search and Rescue-specific features. This approach provided a 7.82% boost in mAP@50 compared to training on Heridal from scratch.

The mean Average Precision (mAP) is the area under the precision–recall curve. A true positive is considered if a target is detected with an intersection over union (IoU) of at least 0.5, and a detection is considered positive if the predicted bounding box overlaps with the ground truth by at least IoU = 0.5. When evaluated with an IoU threshold of 0.5, it is referred to as mAP@50. This means that a detection is considered correct if the predicted bounding box overlaps with the ground truth by at least 50%.

The mAP@50:95 is computed as the average of mAP values at different IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05 (i.e., IoU thresholds of 0.50, 0.55, 0.60,. . ., 0.95). This stricter metric provides a more comprehensive evaluation of the model's localization accuracy, as it penalizes predictions that only roughly overlap with the ground truth.

Results are reported in Section 3. The training strategy using VisDrone pretraining followed by Heridal fine-tuning led to consistently higher performance in both precision and recall. Results from experiments involving different fusion layers and attention mechanisms are compared and analyzed in Section 3.

3. Results

This section presents the experimental results obtained by training and evaluating various configurations of YOLOv5 and YOLOv8 for person detection in aerial imagery, specifically targeting Search and Rescue applications. The performance was assessed using metrics including mAP@50, mAP@50:95, precision, recall, parameter count, and computational cost in GFlops. All results are contextualized concerning deployment feasibility on embedded systems such as the NVIDIA Jetson Nano.

Initial experiments focused on evaluating the impact of different dataset combinations on model performance (Table 3).

Table 3. Results obtained applying different transfer learning strategies (H = Heridal, V = VisDrone, and S = Stanford drone dataset). The best result is highlighted in bold.

Dataset	mAP@50	mAP@50:95	Precision	Recall	Parameters	GFlops
H	0.665	0.281	0.675	0.633	7.013 M	15.8
V + H	**0.738**	**0.343**	**0.799**	**0.670**	**7.013 M**	**15.8**
S + H	0.700	0.313	0.746	0.649	7.013 M	15.8
V + S + H	0.679	0.297	0.682	0.680	7.013 M	15.8

Training the baseline YOLOv5s model on the Heridal dataset alone resulted in a mAP@50 of 0.665. Pretraining the same model on VisDrone before fine-tuning Heridal substantially improved it, reaching a mAP@50 of 0.738. A similar strategy using the Stanford Drone Dataset (S + H) achieved 0.700, while the combined use of VisDrone, Stanford, and Heridal (V + S + H) slightly underperformed, yielding 0.679. These results confirm the effectiveness of VisDrone pretraining and highlight potential overfitting or domain shift when too many heterogeneous sources are combined.

The second set of experiments examined the effect of different feature fusion strategies in the neck of the YOLOv5s model (Table 4).

Table 4. Result comparison of changing feature fusion methods. The best result is highlighted in bold.

Model	mAP@50	mAP@50:95	Precision	Recall	Parameters	GFlops
YOLOv5s	0.738	0.343	0.799	0.670	7.013 M	15.8
YOLOv5s-FPN	0.783	0.344	0.788	0.736	7.022 M	15.9
YOLOv5s-BiFPN	0.750	0.337	0.758	0.699	7.087 M	16.2
YOLOv5s-PBFPN	**0.801**	**0.386**	**0.864**	**0.738**	**5.493 M**	**16.5**

Replacing the default PANet with FPN improved the mAP@50 to 0.783, while BiFPN reached 0.750. The best result was obtained with the PB-FPN configuration, which achieved 0.801 mAP@50 and 0.386 mAP@50:95, with fewer parameters (5.493 M) and comparable computational cost. This suggests that lightweight fusion structures with strong top-down pathways are particularly beneficial for small-object detection in cluttered aerial scenes.

Additional improvements were tested by modifying the detection head structure (Table 5).

Table 5. The results obtained include new detection heads in the model's architecture. The best result is highlighted in bold.

Model	mAP@50	mAP@50:95	Precision	Recall	Parameters	GFlops
YOLOv5s-P2	0.781	0.360	0.845	0.711	5.384 M	17.3
YOLOv5s-P1	0.641	0.283	0.693	0.581	5.473 M	21.8
YOLOv5s-FPNP2	0.750	0.333	0.808	0.665	6.020 M	16.4
YOLOv5s-BiFPNP2	0.701	0.321	0.749	0.665	5.464 M	17.7
YOLOv5s-PBFPNP2	**0.783**	**0.373**	**0.822**	**0.710**	**5.625 M**	**20.5**

Adding a detection head at the P2 scale (YOLOv5s-P2) resulted in a mAP@50 of 0.781, indicating that enhancing the model's sensitivity to higher-resolution features is advantageous when detecting small targets. Combining P2 with FPN or PB-FPN maintained high performance, with YOLOv5s-PBFPNP2 reaching 0.783 mAP@50 and 0.373 mAP@50:95. Conversely, placing detection heads too early (P1) degraded perfor-

mance significantly (mAP@50 = 0.641), confirming that extremely shallow layers lack sufficient semantic context.

We evaluated various modular additions to the best-performing models to further explore architectural enhancements. TPH-YOLOv5, which includes transformer-based heads and CBAM modules, resulted in 0.735 mAP@50. However, applying these components separately proved more effective. In particular, YOLOv5s-P2-Deconvolution achieved 0.779 mAP@50, while the addition of PB-FPN and deconvolution layers (YOLOv5s-PBfpn-Deconvolution) reached the best overall result in this group: 0.802 mAP@50 and 0.388 mAP@50:95. Models combining PB-FPN, CBAM, and deconvolution also performed well, slightly sacrificing recall for precision. These findings underline the critical balance between architectural complexity and small-object detection capability. The results are shown in Table 6.

Table 6. Results obtained by adding different modules to YOLOv5s. The best result is highlighted in bold.

Model	mAP@50	mAP@50:95	Precision	Recall	Parameters	GFlops
YOLOv5s-TPH	0.735	0.326	0.750	0.686	5.430 M	21.5
YOLOv5s-P2-Transformer	0.758	0.332	0.805	0.700	5.384 M	17.0
YOLOv5s-P2-CBAM	0.752	0.344	0.825	0.658	5.395 M	17.3
YOLOv5s-P2-Trans.-CBAM	0.716	0.310	0.711	0.687	6.017 M	17.3
YOLOv5s-BiFPN-T.-CBAM	0.610	0.249	0.650	0.614	9.177 M	23.1
YOLOv5s-P2-CBAM-Deconvolution	0.770	0.353	0.780	0.710	5.048 M	17.7
YOLOv5s-P2-Deconvolution	0.779	0.363	0.835	0.691	5.388 M	17.4
YOLOv5s-PBFPN-P2-Deconvolution	0.722	0.366	0.788	0.723	5.639 M	20.8
YOLOv5s-PBFPN-Deconvolution	**0.802**	**0.388**	**0.842**	**0.743**	**5.490 M**	**16.4**
YOLOv5s-PBFPN-CBAM-Deconvolution	0.769	0.359	0.826	0.710	5.504 M	16.5

A visual representation of the training performance for the best configuration (YOLOv5s-PBfpn-Deconvolution) is provided in the training curve graphs (Figure 7), which illustrate rapid convergence and stable validation metrics over epochs.

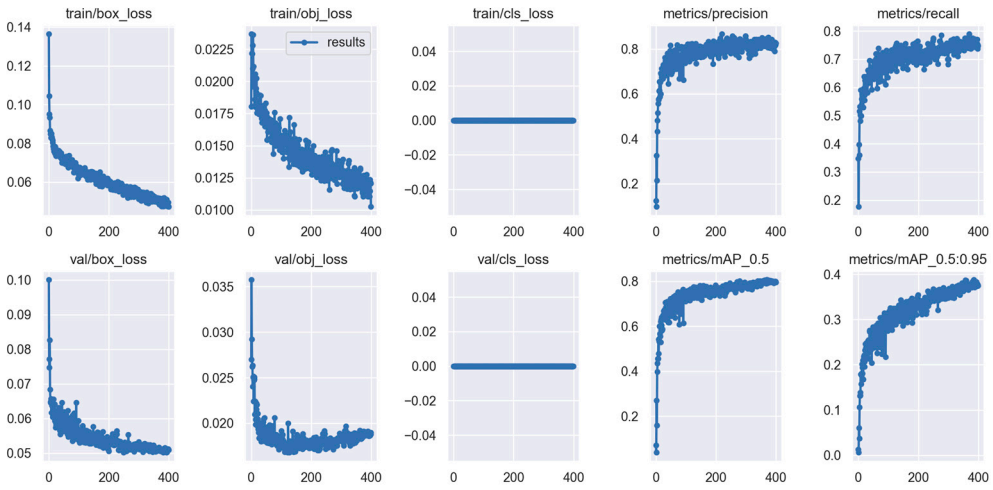


Figure 7. A visual representation of the training and validation performance for the best configuration YOLOv5s-PBFPN-Deconvolution.

Moreover, Figure 8 shows an example of prediction conducted by the trained model YOLOv5s-PBFPN-Deconvolution.

Figure 8. (**a**) An image example from the Heridal dataset test set; (**b**) predicted output by the model.

The study was then extended to the newer YOLOv8 architecture, which integrates an anchor-free design and a more refined backbone. The results are shown in Table 7.

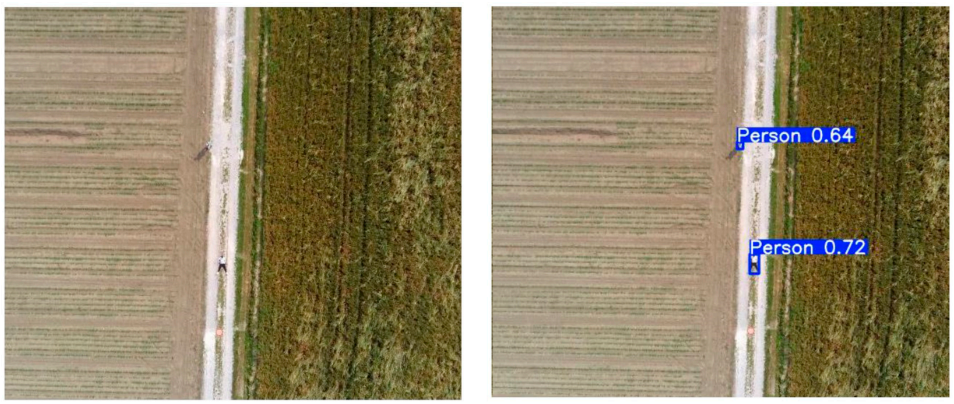
Table 7. Result comparison of YOLOv8 architectural changes. The best results are highlighted in bold.

Model	mAP@50	mAP@50:95	Precision	Recall	Parameters	GFlops
YOLOv8s	0.781	0.387	0.801	0.734	11.126 M	28.4
YOLOv8s-P2	0.788	0.393	0.832	0.716	10.626 M	36.6
YOLOv8s-PBFPN	**0.807**	**0.421**	**0.825**	**0.739**	**12.195 M**	**41.7**
YOLOv8s-PBFPN-P2	0.792	0.418	0.853	0.726	25.662 M	236.5

The baseline YOLOv8s achieved 0.781 mAP@50 and 0.387 mAP@50:95. Adding a P2 detection head slightly improved the performance to 0.788 and 0.393, respectively. PB-FPN integration yielded 0.807 mAP@50 and 0.421 mAP@50:95, the best results among all tested configurations. However, the variant combining both PB-FPN and P2 (YOLOv8s-PBfpnP2), while slightly less accurate (0.792 mAP@50), suffered from a significant increase in model size and complexity, reaching over 25 million parameters and 236.8 GFlops. Such a model exceeds the capabilities of current embedded systems like the Jetson Nano and is thus better suited for offline or cloud-based analysis.

Using UAV-acquired imagery, a complementary experiment evaluated the model's inference performance at different flight altitudes, specifically at 10 m, 20 m, and 50 m. Interestingly, the results revealed a counterintuitive trend: detection accuracy improved with increasing altitude. This behavior can be explained by the fact that the Heridal dataset itself was collected at similar altitudes, meaning the model was implicitly optimized to detect targets under these specific acquisition conditions; YOLOv5 (and v8) uses k-means algorithm to compute the best anchor boxes relative to the dataset, so if the dataset contains only very small objects, the model will learn to detect only very small objects. Performance naturally improved as the test images aligned closely with the training distribution in terms of perspective, object scale, and visual context. Moreover, the broader field of view at higher altitudes may have enhanced the model's ability to capture contextual cues that help

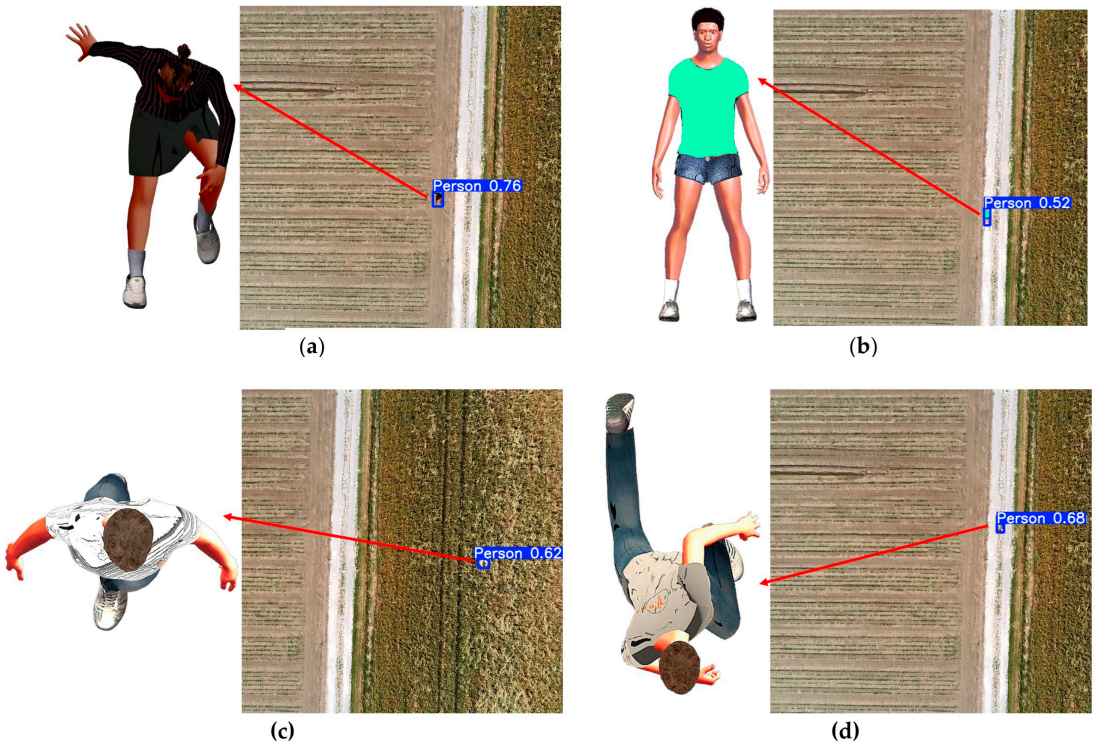
recognize small objects, such as human figures. Figure 9 illustrates detection outputs at 50 m tested altitude, confirming that the feature fusion technique (e.g., PB-FPN) contributed to robust detection across this range of image scales.



(**a**) (**b**)

Figure 9. (**a**) Original image taken at 50 m; (**b**) YOLOv8s-PBFPN output.

More tests have been conducted using YOLOv8s-PBFPN to highlight its detection abilities in scenarios where missing people must be found. The same background as the collected video was used. However, it has been possible to create different humanoid figures with different clothes, positions, gender, and skin color using "MakeHuman" open-source software v1.3.0. Figure 10 shows the figures used with the respective detections: an exhausted woman sitting on the ground, a runner lying on the ground after a heart attack, a man wearing a casual white shirt and jeans walking, and another man running away from hazardous zones are the four virtual mannequins implemented for this test.



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Different examples were obtained using MakeHuman software to create new examples. The images have been used as a new test for YOLOv8s-PBFPN: (**a**) women sitting on the ground; (**b**) woman lying down; (**c**) man walking; (**d**) man running.

Overall, the results demonstrate that careful adaptation of the detection model through feature fusion, head placement, and modular enhancements can significantly improve the detection of small objects like missing persons in aerial Search and Rescue imagery. The optimal configurations achieve a strong balance between accuracy and efficiency, making them viable for real-time deployment on edge devices embedded in UAV systems.

To evaluate the interpretability and robustness of the proposed detection system, we conducted an explainability analysis using EigenCAM [39], a gradient-based class activation mapping method that highlights the regions of the input image most influential in the model's prediction. This technique is beneficial for understanding the internal mechanisms of convolutional neural networks (CNNs) when applied to aerial imagery, where object size and background context play a critical role.

These visual explanations confirmed that the improved architectures focused on relevant regions of the input image. An example image (Figure 11) illustrates how the model's attention aligns with true positive detections, further validating the system's reliability in operational contexts.

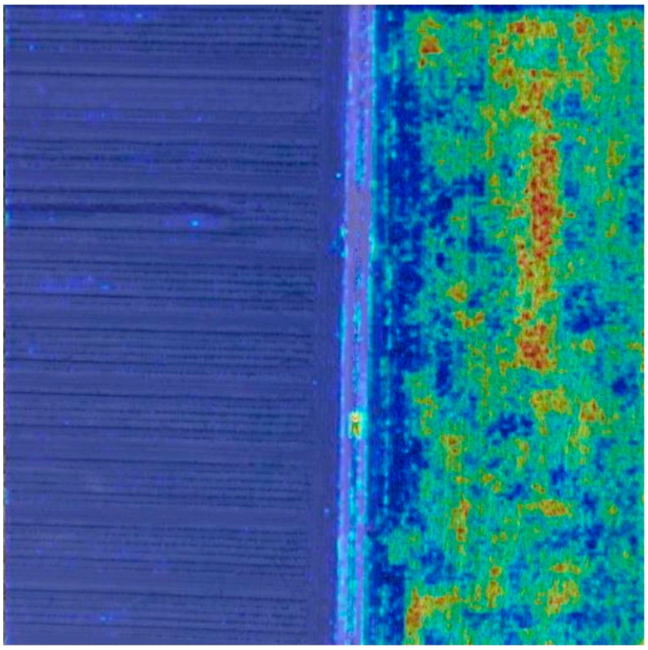


Figure 11. The result of the application of the EigenCAM visualization tool. The image shows how the model focuses its attention on the right part of the image containing vegetation. The model also focuses on the persons correctly.

Finally, to assess the feasibility of deploying the proposed model on embedded hardware, inference experiments were conducted on the NVIDIA Jetson Nano platform. The model was evaluated using representative video sequences containing aerial imagery with small objects to simulate operational conditions.

The evaluation focused on measuring inference latency and confirming that the model could maintain acceptable detection performance on resource-constrained hardware. No additional optimizations, such as INT8 quantization or TensorRT acceleration, were applied at this stage.

The average inference time per frame at 640 $\times$ 640 input resolution was approximately 450 ms, corresponding to an effective processing speed of about 2 frames per second. This latency includes image preprocessing, model inference, and output post-processing. While not yet optimized for real-time operation, these results demonstrate that the model is functional on the Jetson Nano and provides a baseline for future improvements.

Visual inspections confirmed that the model was able to detect persons across different scales, poses, and backgrounds, consistent with the results obtained during desktop evaluation. However, a quantitative measurement of the detection rate under field conditions remains to be conducted.

4. Discussion

This study aimed to develop and optimize real-time object detection models for aerial Search and Rescue civil missions, focusing on improving the accuracy of small-object detection under resource-constrained conditions typical of UAV-based deployments by civil protection agencies. The experimental results provided a comprehensive understanding of how different architectural choices, training strategies, and environmental factors impact model performance, interpretability, and deployability.

The first set of experiments highlighted the importance of dataset composition in training robust detection models. Models trained exclusively on the Heridal dataset exhibited decent performance, but pretraining on VisDrone, followed by fine-tuning on Heridal, yielded a significant boost in detection accuracy, particularly in mAP@50 and precision. This suggests that cross-domain transfer learning from general aerial imagery enhances feature generalization, even when the target dataset contains more specific Search and Rescue scenarios. On the other hand, pretraining on the Stanford Drone Dataset produced more modest gains, possibly due to differences in perspective, scene complexity, and person distribution. Notably, the combination of all three datasets (VisDrone + Stanford Drone Dataset + Heridal) decreased performance, which may be attributed to domain drift and increased intra-dataset variance, reinforcing the idea that more data is not always better without proper domain alignment.

The study of different feature fusion strategies provided additional insight into the backbone-neural encoding capabilities of the models. Replacing the default PANet neck in YOLOv5s with alternative structures such as FPN, BiFPN, and PB-FPN demonstrated that top-down fusion methods with high-resolution preservation are particularly beneficial for small-object detection. Among them, PB-FPN consistently achieved the best performance, suggesting that lightweight, information-rich, and semantically aware feature aggregation can substantially enhance localization precision without increasing model complexity.

In terms of detection head configuration, adding a prediction layer at the P2 scale significantly improved detection metrics across the board. The P2 layer allows the model to utilize high-resolution, shallow features that are particularly informative for detecting small instances, such as humans, in aerial scenes. However, detection heads placed at very early layers (P1) led to underwhelming results, confirming that a minimal semantic representation is required, even when leveraging high spatial resolution.

The evaluation of modular enhancements revealed nuanced trade-offs between architectural complexity and detection efficacy. The integration of Transformer-based modules provided a moderate improvement in recall, likely due to their ability to capture long-range dependencies and contextual relationships. The addition of CBAM increased precision in most configurations by guiding the model's attention toward salient spatial and channel-wise features. Among all tested modules, deconvolution layers offered the most consistent gains in both mAP and precision, particularly when combined with PB-FPN and P2 heads. These findings support the hypothesis that enhancing the spatial resolution in the later stages of the model pipeline is a key factor for improving small-object detection in low-resolution UAV imagery.

The training curves for the best-performing configurations showed smooth convergence and limited overfitting, even in scenarios with deep feature hierarchies. Using well-structured pretraining and careful dataset balancing contributed to stable generalization

across epochs. Models that incorporated multiple architectural innovations (e.g., PB-FPN + P2 + deconvolution) reached higher validation accuracy without exhibiting divergence or performance decay, confirming the robustness of the proposed training pipeline.

A comparative analysis between YOLOv5 and YOLOv8 revealed the inherent trade-off between detection performance and computational feasibility. YOLOv8s and its variants achieved higher mAP@50:95 scores than YOLOv5-based models, benefiting from an anchor-free architecture and a more efficient backbone. However, this gain came at the cost of increased parameter count and GFlops, making several YOLOv8 configurations impractical for real-time inference on edge devices such as the Jetson Nano. Notably, YOLOv5s models with optimized architecture and lightweight modules achieved comparable accuracy at a fraction of the computational cost, reinforcing the suitability of YOLOv5 for embedded Search and Rescue applications in civil scenarios.

An additional set of experiments evaluated the model's performance at different flight altitudes (10 m, 20 m, 50 m). Contrary to common expectations, detection accuracy increased with altitude. This effect can be attributed to the Heridal dataset being collected within the same altitude range, resulting in training data that closely matched the test conditions. Moreover, the higher altitude provides a broader contextual view, which may help the model distinguish humans from background clutter better. This highlights the importance of dataset alignment with deployment conditions and suggests that models trained on a specific altitude distribution may struggle to generalize outside that range.

The practical implications for UAV deployment were explored through a Jetson Nano-based embedded system, where person detection models were executed onboard with real-time transmission of inference results to a ground station. The system was tested under laboratory and field conditions, confirming that the best-performing YOLOv5s configuration achieved a stable inference rate of 1–2 FPS, which is acceptable for UAVs performing stationary or low-speed search operations. Modular, multithreaded communication protocols (TCP/UDP) and a task-specific HMI interface enhanced system reliability and usability, allowing civil protection agencies to use them.

Explainability analyses using EigenCAM added a crucial layer of transparency to the system. One observation was that the model consistently activated more strongly around individuals lying down than those standing upright. This difference may be partially explained by the vertical aspect ratio and the limited footprint of a standing person in top-down views, which leads to weaker gradient signals during training and reduced visual prominence in activation maps. It is important to note that this does not necessarily imply a detection failure but rather a reduced interpretability signal in CAM outputs, a limitation known in the literature for narrow or low-contrast targets.

More intriguingly, the activation maps showed a clear bias in spatial attention toward areas containing vegetation instead of regions with cultivated soil or artificial textures. This was evident in multiple Heridal test images, where the right-hand side (vegetation) triggered significantly stronger activations, even without ground-truth objects, than the left-hand side (open field). This suggests that the model may have implicitly learned to associate natural textures with the presence of humans, potentially due to an unbalanced distribution in the training dataset.

The Heridal dataset includes numerous scenes where persons are embedded within vegetated environments, such as forests or overgrown rural paths. As a result, the network may have developed a spurious correlation, effectively learning that "*vegetation implies a higher probability of a person being present*." While such a heuristic might be effective within distribution, it introduces the risk of background-induced false positives or blind spots in non-vegetated regions during inference in unseen environments.

From a broader XAI perspective, this phenomenon aligns with concerns regarding background context bias in deep learning models. CNN-based detectors can latch onto dominant low-level features (e.g., color, texture, edge density) that co-occur with positive samples, even when these features are not causally linked to the target object. Although these patterns may improve local accuracy during training, they compromise generalization and interpretability, particularly in safety-critical scenarios like Search and Rescue missions, where people must be detected in environmental crises or accidents.

To validate whether this behavior stems from genuine spatial correlation or is an artifact of explainability methods like EigenCAM, future work should include feature ablation studies and counterfactual visual explanations (e.g., by systematically masking background textures or altering terrain types). Such techniques would clarify whether attention is semantically grounded or driven by texture bias.

These findings underscore the importance of dataset curation and bias analysis in critical applications such as the typical Search and Rescue operations occurring after accidents, disasters, or medical emergencies.

Nonetheless, the study has some limitations. All experiments were conducted using RGB imagery without incorporating other sensor modalities such as thermal or multispectral data, which could be advantageous in low-visibility conditions. Additionally, while the inference system was validated on the Jetson Nano, future work should explore deployment on other embedded platforms (e.g., Coral Dev Board, Jetson Orin Nano) to assess energy efficiency and throughput across different hardware configurations. The performance of the detection models should also be evaluated under more diverse environmental conditions, such as urban areas, maritime settings, or nighttime operations.

5. Conclusions

This study proposed and validated a complete methodology for real-time aerial detection of missing persons using optimized YOLO-based architectures deployed on embedded UAV platforms. The system has been developed to support civil protection activities, such as the detection of people lost or disoriented, blocked by flooding, escaping wildfires, or disabled by health issues. By systematically evaluating a wide range of architectural modifications, including enhanced feature fusion strategies, lightweight detection heads, and modular attention mechanisms, we demonstrated that substantial gains in detection accuracy for small objects can be achieved without compromising computational efficiency. Among the tested configurations, the integration of PB-FPN with a P2 detection head and a lightweight deconvolution module yielded the best trade-off between performance and resource usage, making it well suited for real-time deployment on low-power hardware such as the Jetson Nano.

Through extensive training on Search and Rescue-relevant datasets (Heridal) combined with a multi-step transfer learning strategy, the proposed models consistently increased mAP and recall metrics, even in scenarios with cluttered backgrounds and low-resolution targets. Our results also revealed that detection performance can be sensitive to the flight altitude of the UAV, suggesting the need for altitude-aware training or augmentation when targeting broader operational ranges.

Furthermore, the incorporation of explainability methods, specifically EigenCAM, enabled a deeper understanding of the model's internal decision-making processes. This interpretability analysis uncovered strengths (e.g., attention on partially occluded persons) and biases (e.g., over-reliance on vegetated regions), underscoring the need for dataset diversity and careful validation when deploying AI in safety-critical contexts like Search and Rescue.

The embedded deployment tests confirmed the feasibility of running the detection pipeline onboard in near real time, demonstrating the system's applicability in field scenarios where wireless bandwidth, processing power, and operator attention are constrained.

In future work, we plan to extend the system to support multimodal detection (e.g., thermal, IR), increase its robustness to diverse terrains and lighting conditions, and integrate higher-level reasoning modules to support mission-level decision-making, integrating also procedures developed by civil protection authorities. Additionally, we intend to evaluate and compare newer YOLO versions (e.g., YOLOv8, YOLOv10, YOLOv11, YOLOv12) to assess their suitability for embedded deployment and potential accuracy improvements in Search and Rescue civil scenarios. Ultimately, the methodology outlined in this work lays the foundation for building reliable, interpretable, and efficient UAV-based perception systems tailored for real-world Search and Rescue operations managed by civil protection authorities.

Author Contributions: Conceptualization, F.C. and A.C.; methodology, F.C.; software, F.C.; validation, F.C. and A.C.; formal analysis, A.C.; data curation, F.C.; writing—original draft preparation, F.C.; writing—review and editing, A.C.; visualization, F.C.; supervision, A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SAR	Search and Rescue
UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once
APD	Aerial Person Detection
COCO	Common Object in Context.
FPN	Feature Pyramid Network
BiFPN	Bidirectional Feature Pyramid Network
CBAM	Convolutional Block Attention Module
mAP	Mean Average Precision
SPPF	Spatial Pyramid Pooling Fast
TPH-YOLO	Transformer Prediction Head–YOLO
SGD	Stochastic Gradient Descent
IoU	Intersection over Unit
HSV	Hue Saturation Value
HMI	Human–Machine Interface
GUI	Graphical User Interface
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
CAM	Classification Activation Map
CNN	Convolutional Neural Network

References

1. Ford, J.; Clark, D. Preparing for the Impacts of Climate Change Along Canada's Arctic Coast: The Importance of Search and Rescue. *Mar. Policy* **2019**, *108*, 103662. [CrossRef]
2. Lionello, P.; Abrantes, F.; Gacic, M.; Planton, S.; Trigo, R.; Ulbrich, U. The Climate of the Mediterranean Region: Research Progress and Climate Change Impacts. *Reg. Environ. Chang.* **2014**, *14*, 1679–1684. [CrossRef]
3. Ho, Y.-H.; Tsai, Y.-J. Open Collaborative Platform for Multi-Drones to Support Search and Rescue Operations. *Drones* **2022**, *6*, 132. [CrossRef]
4. Khan Mohd, T.; Nguyen, V.; Hoang, T.; Zeyede, P.M.; Abdisa, B. Performance and Efficiency Assessment of Drone in Search and Rescue Operation. In Proceedings of the Artificial Intelligence and Machine Learning, Toronto, ON, Canada, 23 July 2022; pp. 1–16. [CrossRef]
5. Yoo, J.; Goerlandt, F.; Chircop, A. Unmanned Remotely Operated Search and Rescue Ships in the Canadian Arctic: Exploring the Opportunities, Risk Dimensions and Governance Implications. In *Governance of Arctic Shipping*; Chircop, A., Goerlandt, F., Aporta, C., Pelot, R., Eds.; Springer Polar Sciences; Springer International Publishing: Cham, Switzerland, 2020; pp. 83–103, ISBN 978-3-030-44974-2. [CrossRef]
6. Sary, I.P.; Andromeda, S.; Armin, E.U. Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection Using Aerial Images. *Ultim. Comput.* **2023**, *15*, 8–13. [CrossRef]
7. Zhang, X.; Feng, Y.; Wang, N.; Lu, G.; Mei, S. Aerial Person Detection for Search and Rescue: Survey and Benchmarks. *J. Remote Sens.* **2025**, *5*, 0474. [CrossRef]
8. Zhong, H.; Zhang, Y.; Shi, Z.; Zhang, Y.; Zhao, L. PS-YOLO: A Lighter and Faster Network for UAV Object Detection. *Remote Sens.* **2025**, *17*, 1641. [CrossRef]
9. Yeom, S. Thermal Image Tracking for Search and Rescue Missions with a Drone. *Drones* **2024**, *8*, 53. [CrossRef]
10. Kucukayan, G.; Karacan, H. YOLO-IHD: Improved Real-Time Human Detection System for Indoor Drones. *Sensors* **2024**, *24*, 922. [CrossRef] [PubMed]
11. Calabrò, A.; Marchetti, E. Transponder: Support for Localizing Distressed People through a Flying Drone Network. *Drones* **2024**, *8*, 465. [CrossRef]
12. Manzini, T.; Murphy, R. Open Problems in Computer Vision for Wilderness SAR and the Search for Patricia Wu-Murad. *arXiv* **2023**. [CrossRef]
13. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2014; Volume 8693, pp. 740–755, ISBN 978-3-319-10601-4. [CrossRef]
14. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Kai, L.; Li, F.-F. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–26 June 2009; pp. 248–255. [CrossRef]
15. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 213–226. [CrossRef]
16. Božić-Štulić, D.; Marušić, Ž.; Gotovac, S. Deep Learning Approach in Aerial Imagery for Supporting Land Search and Rescue Missions. *Int. J. Comput. Vis.* **2019**, *127*, 1256–1278. [CrossRef]
17. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
18. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10778–10787. [CrossRef]
19. Liu, H.; Sun, F.; Gu, J.; Deng, L. SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. *Sensors* **2022**, *22*, 5817. [CrossRef] [PubMed]
20. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11211, pp. 3–19, ISBN 978-3-030-01233-5. [CrossRef]
21. Li, Y.; Li, Q.; Pan, J.; Zhou, Y.; Zhu, H.; Wei, H.; Liu, C. SOD-YOLO: Small-Object-Detection Algorithm Based on Improved YOLOv8 for UAV Images. *Remote Sens.* **2024**, *16*, 3057. [CrossRef]
22. Hao, W.; Zhili, S. Improved Mosaic: Algorithms for More Complex Images. *J. Phys. Conf. Ser.* **2020**, *1684*, 012094. [CrossRef]
23. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for Small Object Detection. In Proceedings of the 9th International Conference on Advances in Computing and Information Technology (ACITY 2019), Ceske Budejovice, Czech Republic, 21 December 2019; pp. 119–133. [CrossRef]

24. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**. [CrossRef]
25. Dumenčić, S.; Lanča, L.; Jakac, K.; Ivić, S. Experimental Validation of UAV Search and Detection System in Real Wilderness Environment. *Drones* **2025**, *9*, 473. [CrossRef]
26. Fu, Z.; Xiao, Y.; Tao, F.; Si, P.; Zhu, L. DLSW-YOLOv8n: A Novel Small Maritime Search and Rescue Object Detection Framework for UAV Images with Deformable Large Kernel Net. *Drones* **2024**, *8*, 310. [CrossRef]
27. Weng, S.; Wang, H.; Wang, J.; Xu, C.; Zhang, E. YOLO-SRMX:A Lightweight Model for Real-Time Object Detection on Unmanned Aerial Vehicles. *Remote Sens.* **2025**, *17*, 2313. [CrossRef]
28. Robicquet, A.; Sadeghian, A.; Alahi, A.; Savarese, S. Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes. In *Computer Vision–ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9912, pp. 549–565, ISBN 978-3-319-46483-1. [CrossRef]
29. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*; Springer: Cham, Switzerland, 2024. [CrossRef]
30. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**. [CrossRef]
31. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**. [CrossRef]
32. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv* **2025**. [CrossRef]
33. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. *arXiv* **2021**. [CrossRef] [PubMed]
34. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**. [CrossRef]
35. Yun, S.; Han, D.; Chun, S.; Oh, S.J.; Yoo, Y.; Choe, J. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6022–6031. [CrossRef]
36. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. Mixup: Beyond Empirical Risk Minimization. *arXiv* **2017**. [CrossRef]
37. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 658–666. [CrossRef]
38. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2019. [CrossRef]
39. Muhammad, M.B.; Yeasin, M. Eigen-CAM: Class Activation Map Using Principal Components. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Rome, Italy, 30 June–5 July 2020; pp. 1–7. [CrossRef]