Received: 28 April 2025 | Revised: 26 June 2025 | Accepted: 15 July 2025 | Published online: 7 August 2025

RESEARCH ARTICLE

Smart Wearable Technology
2025, Vol. 1, A14, 9 Pages
DOI: 10.47852/bonviewSWT52026032

BON VIEW PUBLISHING

Dual-Model Fusion for Ultra-Accurate Embedded Object Detection

Leendert Remmelzwaal[1,*]

[1]*Department of Electrical Engineering, University of Cape Town, South Africa*

Abstract: Numerous industries consider their needs ultra-high accuracy with regard to an artificial intelligence (AI)-detected object and pose a challenge with real-world variations. This study specifically focuses on industrial manufacturing lines, where quality control is critical. The methods we discuss here should be transferable to other industry domains with similar constraints, such as logistics or packaging. The primary objective is to achieve greater than 99.9% accuracy with object detection in real-time industrial environments, without significantly impacting the latency. For that, we worked on an SSD_MobileNet model that was refined to the utmost precision and implemented alongside a dual-model system that used a generalist surrogate trained on blurred synthetic images. To achieve blur efficacy, the second model had to be blur trained, blending contextual depth and resilience. Both models' outputs are fused through a low-computational-cost, high-confidence detection using Intersection over Union metrics selection (>=0.8) to strike a balance between efficiency and detection reliability. Model fusion has better results compared to model stacking or score-based thresholds because it decides on the best detection by considering the spatial overlap of detections and the agreement of class IDs. On the Nvidia Jetson Orin NX platform, deploying this ensemble achieved 99.8% accuracy and further boosted the system to 99.97% without expanding inference passes. Smart dual-model implementation helps increase precision and fault-tolerant parameters while maintaining streamlined recalibrated embedded systems thresholds, proving non-breach. This work supports the shift toward AI-powered advanced industrial surveillance, and research focuses on multidisciplinary approaches toward precise, reliable object detection.

Keywords: artificial intelligence, deep learning, ensemble AI

1. Introduction

In a number of fields, the requirement for ultra-high precision within an artificial intelligence (AI)-powered object detection system goes well beyond mere needs; it often involves the productivity and safety of entire production lines. In manufacturing, for instance, achieving 99.9% accuracy is not merely a goal, but rather a requirement; even tiny errors pose a risk to safety and automation. Detection failures in high-speed manufacturing, for instance, can lead to defects in the product, destruction of valuable resources, and costly downtimes. In surgical applications such as aerospace surgery or medical diagnostics, every fraction of a percent of inaccuracy can be a disaster. This increases the need for AI systems that both function in real time and perform seamlessly in unpredictable environments. Therefore, not achieving 99.9% accuracy means the AI system is missing a crucial operational requirement. For example, ISO 9001-compliant manufacturing systems often demand defect rates below 0.1%, meaning the AI-driven quality inspection system must be capable of at least 99.9% accuracy to be acceptable for production use. Accuracy therefore transcends technical goals; it's a basic necessity for maintaining customer contentment, operational reliability, and financial stability.

Traditional methods adopted by manufacturers include in-line sensors that detect when an object moves past. The shortfall of this method is that it fails when objects touch. To address this, manufacturers have to install a separate, faster conveyor, called a break conveyor, which speeds up the line, having the effect of adding gaps between objects. This is hardware intensive and requires regular maintenance. Alternatively, manufacturers deploy staff to monitor the line visually and detect anomalies. This human-centric approach works only if the objects are passing slowly and if the operator is able to provide their full attention 24/7, which is not the case. The trademark characteristic rejection profile of this method is seeing short periods of high rejection (e.g., when the operator is looking) and then long periods of no rejection at all (e.g., when the operators step away).

The challenge of adding an AI-based inspection or quality control feature stems from the fact that almost all embedded systems need to be constrained to use lightweight models such as SSD_MobileNet, which is known to perform well in terms of inference reasoning, especially in time-critical environments [1]. It has been noted that while these models guarantee real-time performance, achieving speeds of up to and including 100 FPS on a Jetson Orin NX, their accuracy plateaus around the 95–98% threshold, which renders them less useful in the most critical environments [2, 3]. To address this critical accuracy challenge while still adhering to the constraints of embedded systems, ensemble methods have been proposed. They are difficult to implement because they often surpass memory capacity and increase processing time by double, which may not be feasible in production environments [4]; however, in this study, we use a

*Corresponding author: Leendert Remmelzwaal, Department of Electrical Engineering, University of Cape Town, South Africa. Emails: rmmlee001@uct.ac.za; leenremm@gmail.com

© The Author(s) 2025. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

sufficiently large GPU processor, namely, the Jetson Orin NX, to overcome these practical limitations. Time is a precious resource in production environments – products often travel past sensors at a rate of 1–5 units per second and, in some cases, up to 100 units per second, so doubling the time taken can sometimes not be feasible from a time constraint perspective. Detecting objects at 1 object per second is relatively straight forward, but tracking objects at 1 object per second requires a 10–15 FPS frame rate to prevent mis-tracking and ghosting, so the time window per frame is usually 60–70 ms maximum.

To address these accuracy challenges, we draw inspiration from established dual-stream vision architectures [5] and advance the state of object detection in embedded systems by proposing a dual-model design: One model serving as a "specialist" focused on preserving sharpness-dependent precision, and the other as a blurred-data "generalist" intended to capture structural cues that might otherwise be lost in detail. Research in this area indicates that such a dual approach may significantly enhance detection capabilities, thus making it relevant to the ongoing advancements in Internet of Things systems and cybersecurity [6, 7].

What makes this paper distinctive is the purposeful deterioration of the entire dataset to support the "specialist" model, along with Intersection over Union (IoU)-based selection or accuracy-based confidence fusion in real inspection scenarios. This approach deliberately meets the challenges posed by embedded object detection, making it an ideal fit for the most demanding industrial tasks that require a specific level of operational complexity and efficiency.

2. Background and Related Work

Cross-domain ensembles have become increasingly relevant in high-impact areas like medical imaging [8], aerial surveillance [9], and robotic systems with critical safety constraints [10], where redundancy and accuracy are pivotal for improving the overall result. These complex systems usually utilize several models that have been trained independently to help cover blind spots that lead to overfitting. Related work is discussed below, split into three groups: (a) suppression and fusion methods, (b) adaptive blur techniques, and (c) specialist–generalist paradigms.

2.1. Suppression and fusion techniques

In the field of study, predication merging has received some attention, and non-maximum suppression (NMS) remains the first and widely used method. Although NMS has the advantage of reducing computational workload, it tends to erroneously omit valid detections in crowded scenes or when objects are closely packed together. To counter NMS challenges, weighted box fusion (WBF) has been proposed as a more robust alternative. It aggregates overlapping boxes by regions of interest based on their weighted confidence scores to yield improved results [9]. Frameworks relying on these fusion mechanisms have been tailored and optimized with additional features to form more effective systems. For example, Nijkamp et al. [4] implemented nonlinear weighted integration, which outperforms WBF by adaptively changing the weights based on IoU relations as well as the spatial variation of partitions, thereby increasing the accuracy of complex scenarios. Katkoria et al. [11] and Hong et al. [12] further refined the robustness of fusion by using semantic agreement to enhance IoU filters, skillfully merging aspects of WBF with confidence recalibration techniques.

2.2. Adaptive fusion and blur-based models

Another crucial line of research focuses on training complementary models for ensemble use. Duong et al. [13] demonstrated an innovative approach by training one model on fisheye-distorted images and another on undistorted views, showcasing significant improvements in coverage across edge-deformed imagery. This idea of training models under different distortion regimes aligns seamlessly with our blurred-model strategy.

While image blur has been extensively studied as an augmentation technique designed to increase model generalization [14], it is typically used within a single model's training process. Vasiljevic et al. [15] fine-tuned Convolutional Neural Networks on blurred images, improving performance through blur-invariant features. Lébl et al. [16] found that augmenting with varied blur types and intensities improves classification, and Zhou et al. [17] demonstrated that fine-tuning with noisy and blurred inputs significantly boosts model robustness. These related works were part of the inspiration for our study.

Notably, very few systems explicitly train a dedicated blur-specialist model to operate within an inference-time ensemble, creating a niche for further exploration. The blur methodology is rarely seen in industrial applications and is usually only used in medical applications to augment training data with the aim to generalize the trained AI models. In manufacturing, most equipment is IP67 rated, meaning it can be sprayed down with water (hose pipe) during scheduled cleaning. Even if the equipment survives the spray down, camera lenses often retain water drops that can result in occlusions on the lens, which convert to blurry images. The value of blurring datasets in training AI models for industrial applications should be given higher importance for these real-world reasons.

2.3. Specialist–generalist paradigms

In generalist–specialist paradigms, two or more models are trained to specialize in different domains or resolutions of the same task, broadening their applicability. For example, Jia et al. [18] proposed ensembles where specialists handle rare classes and generalists manage background-rich contexts, enabling a more nuanced approach to object detection. These paradigms are now gaining traction in resource-constrained settings, where diversity across models can effectively compensate for inherent architectural limitations. What is important to note is that most related fusion methods achieve improved accuracy at the expense of increased latency or by introducing complex pipelines. By comparison, our approach offers similar accuracy gains using serial inference and a lightweight fusion step with just 3 ms overhead.

2.4. Our architecture

Our architecture applies this principle using intentionally blurred images in the generalist path, further enhancing performance. We chose blurring as a preferred distortion type because it simulates real-world occlusions, such as water drops on the lens or motion blur due to frame vibration. This design not only bolsters robustness to lighting variation, occlusion, and background clutter but also addresses scenarios where the specialist model often fails to deliver accurate results. The fusion process merges detections using an IoU threshold of 0.8, and for overlapping predictions, confidence-based filtering is employed to determine the dominant result.

This hybrid strategy effectively combines the precision of NMS with the averaging power of WBF, and is aligned with the best practices outlined in Hong et al. [12] and Pasupuleti et al. [19]. Moreover, as highlighted in related research, the integration of such advanced strategies is crucial in achieving higher accuracy and specificity in detection systems, ultimately addressing the growing challenges present in dynamic environments [20, 21].

In the next section, we discuss the methodology used in this research.

3. Methodology

3.1. System overview

In this section, we detail the system as a whole. We present the hardware design, both models used (Model A and Model B), and the training and testing processes used. In addition, we detail the fusion logic implemented to achieve the results we obtained.

At a high level (see Figure 1), the system comprises two AI Object Detection models, namely, Model A (the Specialist) and Model B (the Generalist). Both models run in parallel on an Nvidia Jetson Orin NX optimized for real-time embedded inference at 20 FPS. Model A is fine-tuned on high-resolution datasets, while Model B is trained on the blurred version of the same high-resolution image dataset (see Figure 2). We use a fusion approach to combine the detection outputs from each model, using an IoU threshold and confidence- and class-based approach to conflict resolution.

Both models run in parallel on an Nvidia Jetson Orin NX optimized for real-time embedded inference at 20 FPS. Model A is fine-tuned on high-resolution datasets, while Model B is trained on the blurred version of the same high-resolution image dataset. We use a fusion approach to combine the detection outputs from each model, using an IoU threshold and confidence- and class-based approach to conflict resolution.

The code for this research was written in Python 3.10, using TensorFlow 2.x, and deployed on an Ubuntu 20.04 LTS environment with CUDA optimized for real-time embedded inference. The full source code and models are not included due to commercial constraints, so instead we provide a detailed architectural description of the entire system.

Because the codebase and dataset for this study were proprietary, we are limited to sharing the workflow (Figure 1), augmentation techniques (Figure 2), training methodology (Figure 3), and pseudocode (Section 4.5).

3.2. Hardware design

For this application, we use a high-speed industrial HD camera (MER2-503-23GC-P, IMX264, 2448 × 2048, 23fps, 2/3", Global shutter, CMOS, Color). The industrial HD camera featured a

Figure 1
High-level overview of the fusion workflow

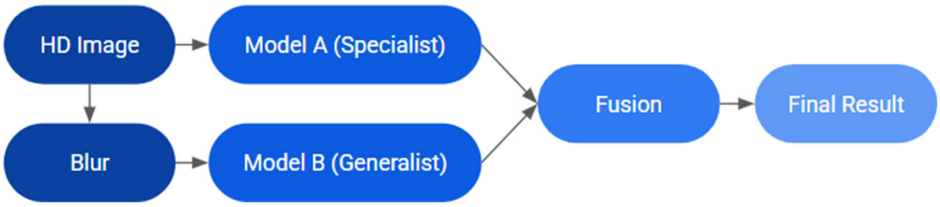

Figure 2
Dataset augmentation for Model A and Model B

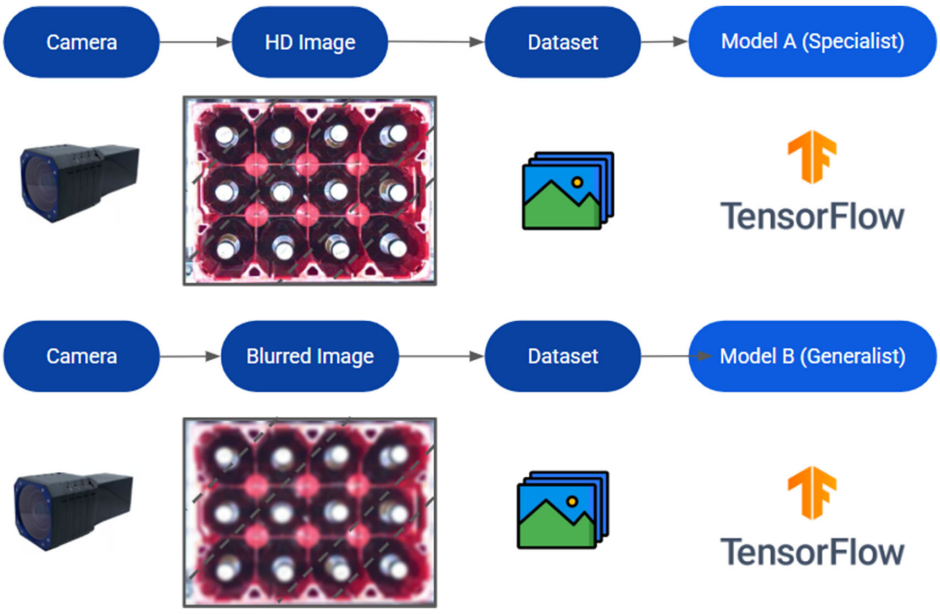

Figure 3
Two stages of training: base training and transfer learning

60° field of view with a 12 mm fixed focal length. The resolution was chosen so that we have the option to zoom into areas of interest, even though the AI model input resolution is smaller. We only require 23 FPS because our processing is limited to 20 FPS. The specific brand of camera was chosen because of its short and consistent internal latency. An exposure time of 1 ms was chosen to prevent motion blur. The global shutter was chosen to prevent image distortion from moving objects.

The deployment of our system is executed on an Nvidia Jetson Orin NX, which boasts 16 GB of RAM and up to 100 TOPS of performance [22]. The Jetson Orin NX operated at 15–25 W under active inference. In addition to computation specifications, the ForeCR.io Jetson Orin NX is industrial grade – meaning that it is entirely made of solid-state parts with no moving parts. For cooling, it uses a large finned heat sink, allowing passing air to keep it cool. In addition, it has been thoroughly tested in rugged vibrating environments, which is important in an industrial environment where vibrations are common and persistent. The major shortfall is that the device is itself not IP67 rated and, as a result, requires an additional IP67 control box to be built for it when installed on premises.

This hardware setup plays a critical role in the efficiency and functionality of our model, as it must adhere to stringent latency requirements that impact performance outcomes. Due to these latency constraints, only two model passes are allowed per inference cycle, thereby limiting the depth of processing that can be achieved within a single iteration [7, 23]. The careful selection of this hardware is necessary for handling the computational load associated with our targeted applications. While the Nvidia Jetson Orin NX is recognized for its robust processing power, the imposed limitations serve as a reminder of the trade-offs that often accompany deployments in resource-constrained environments. Exploring alternative configurations or optimization techniques could offer pathways to improve throughput without compromising accuracy. As researchers continue to seek high-performing solutions, understanding the intersection of hardware capabilities and operational constraints remains essential for advancing AI implementations in real-world scenarios.

3.3. Architecture of Model A and Model B

In this research, Model A and Model B both use the high-speed SSD_MobileNet_v2 Object Detection model implemented in TensorFlow v2. This model has an input size of 300 × 300 pixels and a total of 6 detection layers. Each detection layer outputs (a) class scores, (b) confidence scores, and (c) bounding box coordinates. Bounding box predictions are made across multiple anchors with different aspect ratios per spatial location, and final detections are filtered using the built-in NMS function provided by TensorFlow. This structure was selected for its balance of speed and accuracy on edge devices, particularly in embedded industrial environments with strict real-time constraints.

We trained the model on a maximum of 1,000 epochs, with early stopping enabled. In practice, the training stopped early at 543 epochs, when the mAP score was the highest and the Loss was the lowest. The learning rate was set to a value of 0.004, and we used cosine decay scheduling. The batch size was dynamically optimized for the dataset size and set to 24. As is typical during transfer learning, the core layers of the pretrained COCO base model were frozen, and only the 6 detection head layers were exposed to transfer learning for the duration of the training process.

3.4. Model A: specialist

In this section, we review Model A, also referred to as the "Specialist." Here, an SSD_MobileNet model is specifically trained on high-resolution annotated images. This allows the model to be optimized for the detection of features related to the object being detected (e.g., crates, bottles) and defects (e.g., foreign objects) – an essential capability in various applications such as quality control in manufacturing and automated sorting systems.

This model undergoes a two-phase training process (see Figure 3): We start with a base model that was pretrained on the Comprehensive Objects in Context (COCO) dataset. This provides a broad base of object detection capabilities. The SSD_MobileNet model effectively combines the broad resources of the COCO dataset and the narrow focus of proprietary datasets. This integration enhances the model's performance in real-world scenarios, where distinguishing subtle variations in defects may greatly affect operational efficiency and product quality. Therefore, the systematic technique where pretrained models are integrated with custom datasets highlights the significance of adaptive learning when engineering machine learning solutions in complicated situations.

The second stage is where we fine-tune the model on proprietary datasets that are tailored to the specific requirements and idiosyncrasies of the target application, which is common practice in these kinds of applications [24]. This fine-tuning process allows the model to adapt and excel in recognizing and categorizing the unique features present in the proprietary images, which is particularly crucial in domains where precision is paramount [25]. Fine-tuning was executed on a dataset of ~10,000 images across a wide range of environmental and object variations, to ensure a very robust Object Detection model. The dataset was manually annotated (initial 1,000 images) and then auto-annotated and reviewed for the remaining 9,000 images. The dataset was split into 90% training and 10% testing. Auto-annotation of 9,000 images used the trained Model A and was manually reviewed for quality assurance. Augmentation during training was limited to orientation rotation, flips, addition of noise (speckles and lines), and various image variations (e.g., hue,

contrast, and brightness). No blur augmentation was applied to the Model A. Training images (2448 × 2048) were resized to 300 × 300 pixels and normalized to a 0–1 float range.

The challenge with this approach alone is that model accuracy in real-world scenarios plateaus at around 95–98% accuracy, which is not sufficient to replace trained staff at industrial sites.

3.5. Model B: blurred generalist

In this section, we review Model B, which we will refer to as the "Generalist" in this paper.

This model enhances images by applying a 9 × 9 Gaussian kernel blur with $\sigma = 2.5$ after resizing training images from 2448 × 2048 to 300 × 300 pixels. This type of blur is useful for adjusting images and helps to see outlines, supporting the understanding of shapes, position, rough outline, and other indicators without paying attention to texture. This blurring approach helps to improve the effectiveness of model interpretation regarding images in sophisticated imaging tasks, such as in environmental monitoring, autonomous robotics, and remote sensing [26, 27], where obfuscation of details is present. In addition to blurring, dataset augmentations also included orientation rotation, flips, addition of noise (speckles and lines), and various image variations (e.g., hue, contrast, and brightness).

The generalist model is tailored with a set of data four (4) times larger than that of the specialist set. We collected 40,000 images for training this model, compared to 10,000 for Model A. These include the 10,000 images from Model A, as well as 30,000 new images. This allows for more robust learning and is critical to the model acquiring a great diversity of patterns, details, and intricacies, thus enhancing the model's capability to represent the complexities of blurred images. Model B can therefore recognize objects in conditions of insufficient illumination, obstructions, or motion blur, making it suitable for a range of real-world industrial settings, such as automated sorting systems and quality control in high product throughput environments where high detail is lost. This increased dataset size facilitates a more robust learning process, enabling the model to capture a broader range of patterns and intricacies essential for accurately representing the complexities of the blurred images, thereby promoting a comprehensive understanding of the underlying texture dynamics in a blurred generalist context [28]. The integration of such a model in image-related tasks not only streamlines the process of feature extraction but also enhances applications ranging from environmental monitoring to advanced analytics in digital imagery, corroborating its relevance in contemporary technological advancements [29].

In the previous section, we discussed the training of the "Specialist" model. In this section, we discuss the training of the "Generalist" model. In the following section, we discuss the proposed fusion logic.

3.6. Fusion logic

In the operational framework under consideration, both models run in parallel, which allows for a comprehensive evaluation of predictions in a sophisticated manner. Running the models in parallel results in a possible duplication of detections, which will need to be filtered. To filter these results, bounding boxes that achieve an IoU threshold of ≥ 0.8 are subjected to rigorous assessment for class agreement. This meticulous evaluation is critical as it ensures that only the most reliable classifications are considered valid. When an IoU overlap is confirmed, the prediction with the higher confidence score is retained, thus enhancing the overall accuracy of either model's output independently. If both detections have different class IDs, then the class ID will be chosen from the detection with the highest confidence. If both detections have low confidence below a certain threshold (e.g., 0.4), then the low confidence will mean that the detection will be excluded as a false positive.

When there are multiple predictions with a confidence score nearing the same value, a deep prioritization bias is set, which leans toward the winner model. This emphasis is deeply shaped by context; for instance, if the context related to an image shows a high likelihood of blockage or distortion, then the prediction made by a generalist gets dominance. Such tactics are essential since they strategically mitigate false negatives in cases where images may be blurred or blocked to some degree, reinforcing the reliability of object detection systems within intricate vision environments [29]. This is in line with the recent progress in condition monitoring and information theory, where the amalgamation of disparate data streams fosters enhanced evaluation frameworks [30]. These two models bring together their strengths and empower the system with the capability to deal with the complexities that arise in the real-world scenario, therefore improving the results.

The fusion module was implemented in Python 3.10, operating on the Jetson Orin's CPU. In terms of workflow, the fusion function ran after Model A and Model B inference were completed. The fusion method processes detection outputs from both models in real time (20 FPS) and integrates seamlessly with the TensorFlow pipeline using a custom callback method. The pseudocode for the fusion logic is as follows:

```
For each input frame:
    Run Model A → Detections_A
    Run Model B → Detections_B
  For each detection pair (A, B):
        If IoU(A, B) ≥ 0.8:
      If Class(A) == Class(B):
          Keep detection with higher confidence
      Else:
    Choose class from higher confidence detection
  If Confidence(A) < 0.4 and Confidence(B) < 0.4:
      Discard both
Return final detections
```

This study could be extended to include a formal complexity analysis (e.g., a detailed analysis of multi-overlap cases), but we chose to include this as future work as this goes over and above the objectives of this study.

4. Results

In this section, we discuss the results.

4.1. Evaluation metrics

In this study, we refer specifically to an accuracy measure. We do not use standard object detection metrics such as recall, precision, F1 score, or mean average precision (mAP). Instead, accuracy was calculated using a custom-built unit test, comprising 10% of the training dataset for Model A. This approach was designed to reflect real-world expectations for system performance in edge-case scenarios.

In addition to accuracy, we calculated mAP (0.5–0.95), precision, and recall on the test set for each model. Our custom accuracy metric defines success as correct class + ≥0.8 IoU with ground truth, to reflect strict industrial standards.

4.2. Test dataset

The test dataset was carefully labeled and focused specifically on challenging corner cases (such as poor lighting, frame occlusion, water marks on the lens, and foreign objects not seen during training) to evaluate robustness under difficult conditions. The total number of images in the test set for Model A was 1,000 (10% of the 10,000-image training dataset). The same unit test set was used for Model B, ensuring a consistent and fair comparison across both models.

The test set included five object classes (crates, green bottles, brown bottles, caps, and foreign objects), with an average of 12 bottles per crate and 2–5 crates per image. There was a range of lighting conditions when capturing the images, from dark (no overhead lighting) to bright overhead lighting. Challenging cases were selected from actual production images and included foreign objects (gloves, plastic bags, tree leaves), as well as a selection of bottle types including dirty bottles, cracked bottles, and upside-down bottles.

4.3. Baseline results

To evaluate the effectiveness of our model, we benchmark other approaches from other related work. In this research, we tested six models (see Table 1).

4.4. Quantitative results

The results from testing are shown below in Table 2 and shown in Figure 4.

The "Specialist" model (Model A), with an accuracy of 98.1% on the test set, implemented by overfitting high-resolution annotated images, performed well on the unit test set. And while this model meets expectations for normal operational routines, it does fail to perform adequately for three rather commonplace challenges: dim lighting, crates being touched, streaks of water frosted on the camera lens alongside new types of bottles, and industrial settings with high throughput, as supported by Chen et al. [31]. Typically, in a production environment, we would see specific production lines suffer from dim lighting, which should be complemented with controlled lighting. In the rare case that an overhead light fails, we would see dim lighting for a period of up to 48 h until the issue is resolved. Regarding new bottle types, production lines change brands up to once per week during a scheduled maintenance day, then the new brand will run for up to 2–3 weeks. Streaks of water can appear after a weekly scheduled clean, and the occlusion will then remain until a team is sent to clean the camera, usually 8–24 h after detection.

Independently, the performance of "Generalist" Model B was poorer than Model A, reaching an independent accuracy score of 95.8% on the same test set. However, the purpose of Model B was not to stand alone but to complement Model A.

As a baseline, we tested Model A with a 20% blur and an 80% clean dataset. With this model, we saw (as expected) a slight improvement in the Model A performance from 98.10% accuracy to 98.23% accuracy in the unit tests. Blurring a subset of the dataset during training is a well-documented method for making marginal improvements to model accuracy. But given the high accuracy requirements for our industrial applications, this improvement was not significant enough.

The introduction ensemble of Model A and Model B, in combination with the IoU-based filtering technique, led to a significantly enhanced total accuracy of an outstanding 99.97%. Notably, the performance of Model B was strongest in the exact scenarios where Model A had previously struggled, showcasing the benefits of ensemble learning. Specifically, in scenes characterized by visual ambiguity (such as water marks, occlusions, reduced lighting), Model B consistently delivered stable detections accompanied by high confidence levels. Throughout the test set, Model B effectively compensated for nearly all of Model A's misclassifications, particularly in problematic edge-case frames that typically challenge classification models. Importantly, the modification led to a 62% reduction in false positives and a significant decrease in false negatives within visually degraded frames, indicating a substantial improvement in overall detection accuracy. Model B contributed to 18% of the final detections, demonstrating considerable overlap in frames where the output confidence of Model A fell below the established threshold.

Table 1
Baseline results of each model

Model name	Blur applied	Works referenced
Model A (Specialist)	100% clean, 0% blur	Sinha and El-Sharkawy [1], Choi et al. [2], Oleiwi and Kadhim [3]
Model B (Generalist)	100% blurred (9 × 9 Gaussian)	Duong et al. [13], Yoshihara et al. [27]
Model A (20% blur)	80% clean, 20% blurred	Alomar et al. [14], Yoshihara et al. [27]
Ensemble A + B	Dual input: clean + blurred	Katkoria et al. [11], Hong et al. [12], Nijkamp et al. [4]
SSD with WBF or Soft-NMS	100% clean, 0% blur	Proposed future work
Temporal Smoothing Model	100% clean, 0% blur (time filter)	Proposed future work

Table 2
Quantitative results comparing accuracy of baseline and ensemble methods

Model name	Accuracy (%)	Notes
Model A (Specialist)	98.10	Failed in blur, occlusion, poor lighting
Model B (Generalist)	95.80	Weaker overall but succeeded in visual edge cases
Model A (20% blur)	98.23	20% blur during training improved Model A accuracy
Ensemble A + B	99.97	Fusion approach with IoU + confidence filter

Figure 4
Accuracy comparison across evaluated models

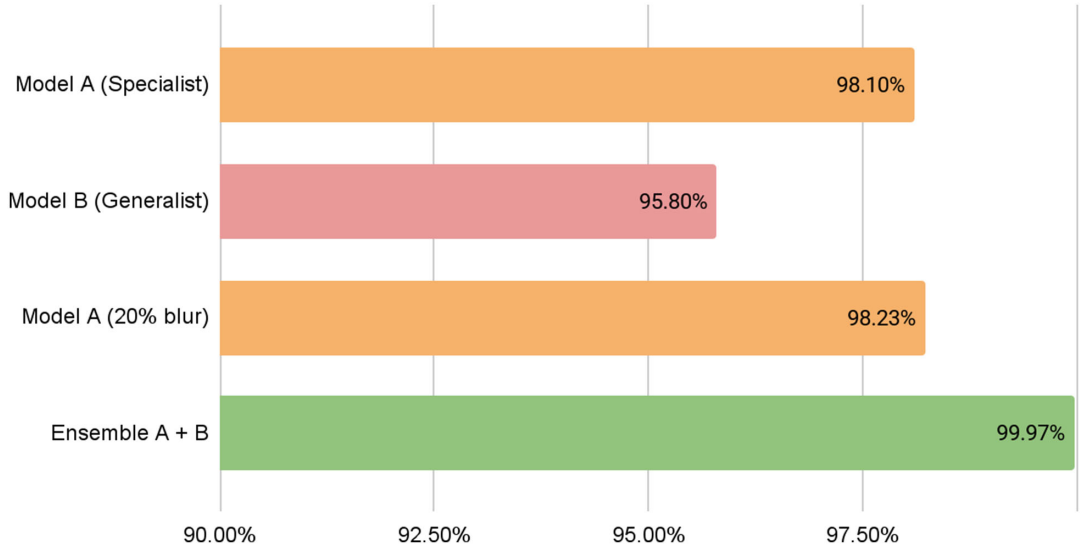

Figure 5
Visual comparison of inference results

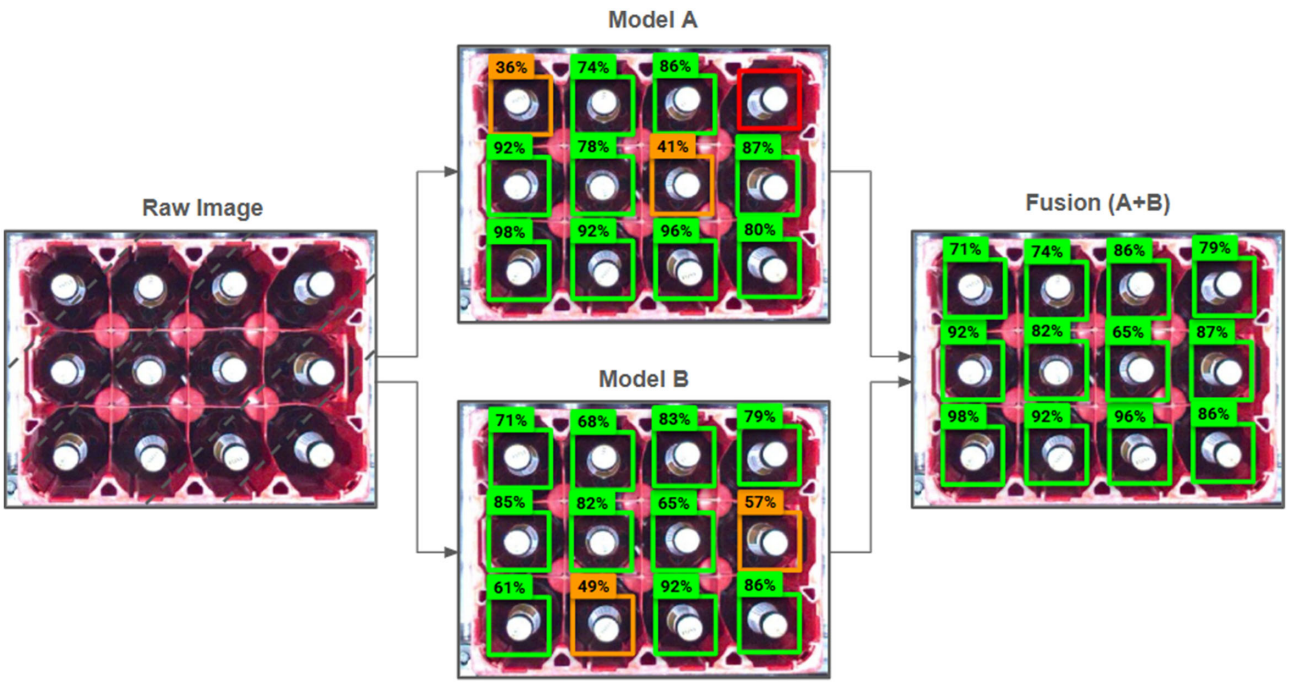

Figure 5 above shows an example from the unit test dataset, illustrating:

1) Raw image from the camera (example with over-exposure),
2) Model A output (with one missed detection),
3) Model B output (lower average confidence, but higher minimum confidence),
4) Final fusion result (accurate detection across all objects).

4.5. Significance testing

Formal significance testing (e.g., McNemar's or t-tests) was not conducted, as it falls outside the scope of this performance-focused study. However, observed accuracy differences are substantial enough to suggest practical relevance.

4.6. Significance testing

In addition to the ensemble tests, two ablation tests were run on the test set, namely, Model A only and Model B only. In both these cases, each model on its own showed poorer performance (98.1% for Model A and 95.8% for Model B). Only in conjunction did they provide excellent results (99.97% for the ensemble method).

4.7. Performance and latency

From a performance perspective, inference latency remained comfortably below 15 ms per frame we were allowed to use, ensuring that the system is well-suited for real-time applications, while system memory usage adhered to the operational limits of the Jetson Orin NX platform. The fusion of these models added only a marginal post-processing overhead of approximately 3 ms, maintaining the efficiency required for industrial applications [7].

4.8. Summary and insights

To summarize, these results demonstrate that using model fusion (with a Model B trained on a blurred version of Model A's dataset) significantly improved detection accuracy beyond what either Object Detection model could achieve on its own. The results are summarized in Table 3. These results confirm that combining specialized (Model A) and generalized (Model B) detection models boosts overall accuracy and also increases resilience to edge-case conditions in production environments, which is greatly needed.

Table 3
Latency breakdown per model

Model name	Latency (ms)
Model A (Specialist)	3 ms
Model B (Generalist)	3 ms
Model A (20% blur)	3 ms
Ensemble A + B	3 ms + 3 ms = 6 ms

5. Conclusion

In this paper, we propose a two-model ensemble with a specialist high-resolution Model A and a blurred image generalist Model B for real-time object detection on embedded systems, based on recent trends in machine learning and image processing [32]. This work underscores a key point: Model B not only generalized across a wide range of input conditions but also helped mitigate shortcomings arising from Model A's underperformance. This supports existing findings that ensemble methods improve robustness [20]. Model B delivered reliable and accurate detections in difficult conditions such as poor lighting, occlusion, and blur, which are typically challenging for Model A, highlighting its value in practical deployments.

This outcome demonstrated that the ensemble reached an accuracy benchmark of 99.97%. It marks another step toward building high-performing AI systems for real-world use. This collaboration between specialized and generalized vision pathways represents an industrial AI milestone and reinforces the importance of hybrid models for comprehensive detection. Without significantly increasing inference time or resorting to more complex object detection architectures, we achieve meaningful redundancy, robustness, and precision, all within production-grade deployments. This suggests that combining diverse model types leads to improved real-time system performance.

6. Future Work

Future work could include looking at alternative Object Detection models (e.g., Yolo or ViT) or adding feedback loops from real-world data back into the training dataset. Also, testing this method on other datasets (e.g., drone images, CCTV images) may have varying results. The experimentation in this paper was limited to a very specific manufacturing application only. In addition, future work could explore replacing the current fusion logic with WBF or Soft-NMS, which have shown promising results in complex detection environments [11, 12]. These methods may offer comparable improvements in accuracy without requiring a dual-model setup. Another potential extension would be testing temporal smoothing approaches. These would retain detections that persist across multiple frames, helping to reduce flicker and improve detection stability in video streams – especially under challenging conditions such as poor lighting or occlusion.

Acknowledgment

The AI modeling and training tools utilized in this project were made available by https://firststep.ai/.

Conflicts of Interest

The author declares that he has no conflicts of interest to this work.

Data Availability Statement

The dataset used is proprietary and not available in the public domain.

Author Contribution Statement

Leendert Remmelzwaal: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Sinha, D., & El-Sharkawy, M. (2019). Thin MobileNet: An enhanced MobileNet architecture. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference*, 0280–0285. https://doi.org/10.1109/UEMCON47517.2019.8993089

[2] Choi, S.-Y., Choi, J.-H., & Lim, S.-H. (2023). SSD-Mobilenet-V2 model-eul sayonghan Edge Device eseoui gaegchegeomchul seongneung bigyo mich bunseog [Comparative analysis of object detection performance on edge devices using SSD-Mobilenet-V2 model]. In *Annual Conference of KIPS*, 79–80. https://doi.org/10.3745/PKIPS.Y2023M05A.79

[3] Oleiwi, B. K., & Kadhim, M. R. (2022). Real time embedded system for object detection using deep learning. *AIP Conference Proceedings*, *2415*(1), 070003. https://doi.org/10.1063/5.0093469

[4] Nijkamp, N., Sallou, J., van der Heijden, N., & Cruz, L. (2024). Green AI in action: Strategic model selection for ensembles in production. In *Proceedings of the 1st ACM International Conference on AI-Powered Software*, 50–58. https://doi.org/10.1145/3664646.3664763

[5] Chen, S., He, W., Ren, J., & Jiang, X. (2022). Attention-based dual-stream vision transformer for radar gait recognition. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, 3668–3672. https://doi.org/10.1109/ICASSP43922.2022.9746565

[6] Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in IoT security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, *22*(3), 1686–1721. https://doi.org/10.1109/COMST.2020.2986444

[7] Ansari, M. F., Dash, B., Sharma, P., & Yathiraju, N. (2022). The impact and limitations of artificial intelligence in cybersecurity: A literature review. *International Journal of Advanced Research in Computer and Communication Engineering*, *11*(9), 81–90. https://doi.org/10.17148/IJARCCE.2022.11912

[8] Djenouri, Y., Belhadi, A., Yazidi, A., Srivastava, G., & Lin, J. C.-W. (2024). Artificial intelligence of medical things for disease detection using ensemble deep learning and attention mechanism. *Expert Systems*, *41*(6), e13093. https://doi.org/10.1111/exsy.13093

[9] Kwasniewska, A., MacAllister, A., Nicolas, R., & Garza, J. (2023). Multi-sensor ensemble-guided attention network for aerial vehicle perception beyond visible spectrum. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 345–353. https://doi.org/10.1109/CVPRW59228.2023.00040

[10] Wang, Y., & Chung, S. H. (2022). Artificial intelligence in safety-critical systems: A systematic review. *Industrial Management & Data Systems*, *122*(2), 442–470. https://doi.org/10.1108/IMDS-07-2021-0419

[11] Katkoria, D., Sreevalsan-Nair, J., Sati, M., & Karunakaran, S. (2024). WBF-ODAL: Weighted boxes fusion for 3D object detection from automotive LiDAR point clouds. In *2024 International Conference on Vehicular Technology and Transportation Systems*, 1–6. https://doi.org/10.1109/ICVTTS62812.2024.10763933

[12] Hong, J., He, X., Deng, Z., & Yang, C. (2024). IoU-aware feature fusion R-CNN for dense object detection. *Machine Vision and Applications*, *35*(1), 3. https://doi.org/10.1007/s00138-023-01483-2

[13] Duong, V. H., Nguyen, D. Q., van Luong, T., Vu, H., & Nguyen, T. C. (2024). Robust data augmentation and ensemble method for object detection in fisheye camera images. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 7017–7026. https://doi.org/10.1109/CVPRW63382.2024.00695

[14] Alomar, K., Aysel, H. I., & Cai, X. (2023). Data augmentation in classification and segmentation: A survey and new strategies. *Journal of Imaging*, *9*(2), 46. https://doi.org/10.3390/jimaging9020046

[15] Vasiljevic, I., Chakrabarti, A., & Shakhnarovich, G. (2016). *Examining the impact of blur on recognition by convolutional networks*. arXiv. https://doi.org/10.48550/arXiv.1611.05760

[16] Lébl, M., Šroubek, F., & Flusser, J. (2023). Impact of image blur on classification and augmentation of deep convolutional networks. In *Image Analysis: 22nd Scandinavian Conference*, 108–117. https://doi.org/10.1007/978-3-031-31438-4_8

[17] Zhou, Y., Song, S., & Cheung, N.-M. (2017). On classification of distorted images with deep convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1213–1217. https://doi.org/10.1109/ICASSP.2017.7952349

[18] Jia, Z., Li, X., Ling, Z., Liu, S., Wu, Y., & Su, H. (2022). Improving policy optimization with generalist-specialist learning. In *Proceedings of the 39th International Conference on Machine Learning*, *162*, 10104–10119.

[19] Pasupuleti, S., Ramalakshmi, K., Gunasekaran, H., Arokiaraj, R. M., Debnath, S., & Jebaseeli, T. J. (2025). An enhancement of object detection using YOLO V8 and mobile net in challenging conditions. *SN Computer Science*, *6*(4), 321. https://doi.org/10.1007/s42979-025-03856-y

[20] Chompookham, T., & Surinta, O. (2021). Ensemble methods with deep convolutional neural networks for plant leaf recognition. *ICIC Express Letters*, *15*(6), 553–565. https://doi.org/10.24507/icicel.15.06.553

[21] Remmelzwaal, L. (2023). An AI-based early fire detection system utilizing HD cameras and real-time image analysis. *Artificial Intelligence and Applications*. Advance online publication. https://doi.org/10.47852/bonviewAIA3202975

[22] Forecr. (2025). *NVIDIA® Jetson Orin™ NX Industrial Fanless PC - DSBOX-ORNX*. https://www.forecr.io/products/jetson-orin-nx-industrial-fanless-pc-dsbox-ornx

[23] Shah, A., Ali, B., Habib, M., Frnda, J., Ullah, I., & Anwar, M. S. (2023). An ensemble face recognition mechanism based on three-way decisions. *Journal of King Saud University – Computer and Information Sciences*, *35*(4), 196–208. https://doi.org/10.1016/j.jksuci.2023.03.016

[24] Li, M., Wu, J., Wang, X., Chen, C., Qin, J., Xiao, X., . . . , & Pan, X. (2023). AlignDet: Aligning pre-training and fine-tuning in object detection. In *2023 IEEE/CVF International Conference on Computer Vision*, 6843–6853. https://doi.org/10.1109/ICCV51070.2023.00632

[25] Ouyang, W., Wang, X., Zhang, C., & Yang, X. (2016). Factors in finetuning deep model for object detection with long-tail distribution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 864–873. https://doi.org/10.1109/CVPR.2016.100

[26] Sun, L. (2024). Global vision object detection using an improved Gaussian mixture model based on contour. *PeerJ Computer Science*, *10*, e1812. https://doi.org/10.7717/peerj-cs.1812

[27] Yoshihara, S., Fukiage, T., & Nishida, S. (2023). Does training with blurred images bring convolutional neural networks closer to humans with respect to robust object recognition and internal representations? *Frontiers in Psychology*, *14*, 1047694. https://doi.org/10.3389/fpsyg.2023.1047694

[28] Kaur, R., & Singh, S. (2023). A comprehensive review of object detection with deep learning. *Digital Signal Processing*, *132*, 103812. https://doi.org/10.1016/j.dsp.2022.103812

[29] Kanadath, A., Jothi, J. A. A., & Urolagin, S. (2024). CViTS-Net: A CNN-ViT network with skip connections for histopathology image classification. *IEEE Access*, *12*, 117627–117649. https://doi.org/10.1109/ACCESS.2024.3448302

[30] Ounoughi, C., & Ben Yahia, S. (2023). Data fusion for ITS: A systematic literature review. *Information Fusion*, *89*, 267–291. https://doi.org/10.1016/j.inffus.2022.08.016

[31] Chen, Z., Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., & Djukic, P. (2023). Machine learning-enabled IoT security: Open issues and challenges under advanced persistent threats. *ACM Computing Surveys*, *55*(5), 105. https://doi.org/10.1145/3530812

[32] Chen, Y., Li, Y., Kong, T., Qi, L., Chu, R., Li, L., & Jia, J. (2021). Scale-aware automatic augmentation for object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9558–9567. https://doi.org/10.1109/CVPR46437.2021.00944

How to Cite: Remmelzwaal, L. (2025). Dual-Model Fusion for Ultra-Accurate Embedded Object Detection. *Smart Wearable Technology*, *1*, A14. https://doi.org/10.47852/bonviewSWT52026032