electronics

MDPI

Article

Noise-Resilient Masked Face Detection Using Quantized DnCNN and YOLO

Rockhyun Choi [1,2], Hyunki Lee [1], Bong-seok Kim [3], Sangdong Kim [3,4] and Min Young Kim [2,*]

1 Division of Intelligent Robot, ICT Research Institute, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, Republic of Korea; choimosi@dgist.ac.kr (R.C.); hklee@dgist.ac.kr (H.L.)
2 School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea
3 Division of Mobility Technology, ICT Research Institute, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, Republic of Korea; remnant@dgist.ac.kr (B.-s.K.); kimsd728@dgist.ac.kr (S.K.)
4 Interdisciplinary Engineering, and Department of Advanced Technology of Daegu Gyeongbuk Institute of Science and Technology, Daegu 42988, Republic of Korea
* Correspondence: minykim@knu.ac.kr

Abstract

This study presents a noise-resilient masked-face detection framework optimized for the NVIDIA Jetson AGX Orin, which improves detection precision by approximately 30% under severe Gaussian noise (variance 0.10) while reducing denoising latency by over 42% and increasing end-to-end throughput by more than 30%. The proposed system integrates a lightweight DnCNN-based denoising stage with the YOLOv11 detector, employing Quantize-Dequantize (QDQ)-based INT8 post-training quantization and a parallel CPU–GPU execution pipeline to maximize edge efficiency. The experimental results demonstrate that denoising preprocessing substantially restores detection accuracy under low signal quality. Furthermore, comparative evaluations confirm that 8-bit quantization achieves a favorable accuracy–efficiency trade-off with only minor precision degradation relative to 16-bit inference, proving the framework's robustness and practicality for real-time, resource-constrained edge AI applications.

Keywords: noise reduction; DnCNN; object detection; YOLO; edge AI

1. Introduction

Reliable face detection in real-world environments remains challenging due to various image degradations such as illumination changes, motion blur, sensor interference, and low-resolution imaging. These degradations distort facial texture and significantly reduce the performance of conventional detectors, especially when facial regions are partially occluded. Prior studies have shown that noise is one of the most influential factors that deteriorate face-related tasks, particularly under unconstrained conditions where imaging quality cannot be guaranteed [1]. Low-quality or compressed images often fail to preserve key discriminative features, making downstream recognition and detection less reliable [2]. Such degradations are common in many practical deployments such as surveillance systems and low-cost camera platforms [3].

In addition to these challenges, face occlusion caused by mask-wearing has become increasingly relevant in various scenarios. Masks are often worn in medical facilities, industrial environments, public transportation, and crowded indoor locations where health

Check for updates

Academic Editor: Yue Wu

Received: 5 December 2025
Revised: 23 December 2025
Accepted: 25 December 2025
Published: 29 December 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

or safety considerations are required; the COVID-19 period is a representative example that highlighted the widespread impact of mask usage. The presence of masks further complicates the detection process by covering key facial regions. Previous works have attempted to address masked face detection; however, noise contamination continues to be a major obstacle for achieving stable performance under real-world conditions [4].

Deep learning-based image restoration models have recently shown strong capability in suppressing complex and spatially varying noise. Among them, the Denoising Convolutional Neural Network (DnCNN) has demonstrated high effectiveness due to its residual learning framework and batch normalization mechanism, outperforming classical filtering-based approaches [5]. Nonetheless, integrating such denoising networks with modern object detectors such as YOLOv11 [6] imposes substantial computational overhead. As a result, real-time deployment on resource-constrained edge-AI devices becomes difficult without additional optimization.

To overcome these limitations, this work proposes a noise-resilient masked face detection framework that combines DnCNN-based denoising with YOLOv11 detection, enhanced through neural network quantization techniques. Quantization is widely adopted to reduce memory usage and computational complexity for edge deployment [7]. Furthermore, recent findings indicate that quantization can provide a beneficial regularization effect, improving robustness under noisy conditions [8–10]. Motivated by these insights, this work focuses on both noise resilience and computational efficiency by unifying quantized denoising and high-accuracy object detection within a single edge-friendly architecture.

The key contributions of this paper are summarized as follows:

- End-to-End Noise-Resilient Detection Pipeline for Edge Deployment: We propose an end-to-end denoising–detection pipeline that integrates a lightweight DnCNN-based denoiser with the YOLOv11 detector, enabling robust masked-face detection under noisy imaging conditions on resource-constrained edge devices.
- Systematic Evaluation across Desktop and Embedded Platforms: We perform a comprehensive and controlled evaluation of the proposed pipeline on both a desktop workstation and the NVIDIA Jetson AGX Orin, focusing on detection robustness, quantization stability, and real-time feasibility under multiple noise levels.
- Practical INT8 Deployment via QDQ-Based Post-Training Quantization: We demonstrate that ONNX-compliant QDQ-based post-training quantization enables efficient INT8 acceleration of the denoising stage with minimal accuracy degradation, supporting practical deployment in latency-tolerant edge scenarios.
- Parallelized Edge-AI Execution Pipeline: We implement a parallelized CPU–GPU execution pipeline that overlaps preprocessing, denoising, and detection, significantly improving hardware utilization and increasing end-to-end throughput on the Jetson AGX Orin.
- Comprehensive Validation across Noise Levels and Hardware Settings: Extensive experiments on the FMLD dataset across noise variances from 0.01 to 0.10 confirm consistent detection improvements and demonstrate the robustness and deployability of the proposed system in real-world edge environments.

The remainder of this paper is organized as follows. Section 2 reviews related work on denoising, masked face detection, and model quantization. Section 3 details the architecture and methodology of the proposed model. Section 4 presents experimental results, and Section 5 concludes the paper.

2. Related Work

2.1. Image Denoising and Restoration

Image denoising and restoration techniques have traditionally relied on filtering-based approaches due to their computational efficiency and simplicity. Representative methods include BM3D [11], wavelet-based filtering [12], and multiscale or low-rank formulations for structured noise suppression [13–15]. However, these approaches depend on handcrafted transforms and manually designed priors, which limits their generalization to diverse noise characteristics and often leads to over-smoothing or loss of fine details. These limitations motivate the adoption of more expressive, data-driven denoising models.

Deep learning-based methods have thus emerged as powerful alternatives, offering the ability to learn adaptive representations that surpass the capabilities of traditional filtering-based algorithms. Convolutional Neural Network (CNN) models such as DnCNN [5] leverage residual learning and batch normalization to perform robust blind denoising without requiring explicit noise-level information. Transformer-based architectures like SwinIR [16] further enhance restoration quality by modeling long-range dependencies through self-attention mechanisms, achieving state-of-the-art performance across diverse benchmarks. Despite these advantages, deep learning-based denoisers often suffer from substantial computational and memory costs, particularly Transformer-based designs. This limits their suitability for latency-sensitive or resource-constrained environments, such as edge devices or real-time applications. Consequently, lightweight CNN-based models such as DnCNN and FFDNet [17] have been explored to strike a more practical balance between efficiency and performance. Notably, DnCNN functions as a blind denoiser without requiring a noise-level map [18], making it more adaptable to dynamic real-world conditions where noise variance is unknown. Nevertheless, even these CNN-based models can impose non-negligible latency depending on the platform and precision used. These observations motivate the need for further complexity reduction, especially through quantization or model simplification, to enable efficient yet robust denoising pipelines suitable for real-time deployment. To balance representativeness and experimental feasibility under edge deployment constraints, this study restricts the denoising comparison to three representative models: DnCNN as a blind CNN-based denoiser, FFDNet as a lightweight non-blind model, and SwinIR as a Transformer-based state-of-the-art approach.

Several studies have investigated YOLO-based object detection under challenging imaging conditions. Li et al. proposed a YOLO-based ship detection framework for thermal infrared images captured under complex backgrounds, demonstrating the applicability of YOLO detectors in degraded sensing environments [19]. Rodríguez-Rodríguez et al. systematically analyzed the impact of noise and brightness variations on modern object detectors, including YOLO, highlighting robustness degradation induced by input perturbations [20]. More recent works have explored architectural or preprocessing modifications to improve YOLO robustness under adverse conditions, such as DiffuYOLO for small-object detection in remote sensing imagery [21] and Dark-YOLO for low-light object detection [22]. In contrast to these studies, which primarily emphasize detection accuracy under clean or moderately degraded conditions, this work focuses on system-level robustness under severe noise and low-precision inference in resource-constrained edge environments.

2.2. Masked Face Detection Under Adverse Conditions

Environments where mask-wearing is unavoidable—such as medical facilities, industrial sites, and pandemic situations like COVID-19—have increased the demand for robust face detection systems capable of handling occlusions. Benchmark datasets such as MAFA and the FMLD dataset [23,24] were introduced to address this challenge. Although state-of-the-art object detectors, including YOLOv10 [25] and the recently released

YOLOv11 [6], have improved occlusion robustness through advanced feature fusion modules (e.g., PANet, BiFPN), their performance degrades sharply when visual degradations are combined—such as a masked face in a noisy, low-light environment [4].

Most existing studies focus on either denoising or masked face detection in isolation. There is limited research on integrated frameworks that simultaneously address occlusion and sensor noise. This study bridges this gap by proposing a unified pipeline that enhances the input quality for YOLOv11 via a lightweight denoiser, ensuring robust detection even under severe noise conditions.

2.3. Quantization for Efficient and Robust Edge Deployment

Deploying deep neural networks on resource-constrained edge devices (e.g., NVIDIA Jetson series) requires rigorous optimization. Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) are essential techniques that reduce model size and inference latency by converting 32-bit floating-point weights to lower-precision formats such as INT8 [7].

Beyond computational efficiency, recent theoretical and empirical studies suggest that quantization can enhance model robustness. Research indicates that the discrete nature of quantized weights can act as a form of implicit regularization, filtering out high-frequency noise perturbations and preventing overfitting to noisy labels [8,9]. For instance, Wang et al. [10] demonstrated that quantization consistency regularization improves generalization in varying domains. Motivated by these findings, this study explores how low-bit quantization (up to 8-bit) of the DnCNN module not only accelerates inference but also contributes to stable detection performance by suppressing minor noise artifacts.

3. Proposed Method

3.1. System Architecture: Initialization, Validation, and Edge Deployment

This subsection provides an overview of the end-to-end architecture of the proposed noise-robust masked face detection framework. The system is organized into three sequential stages—Initialization, Validation, and Edge Deployment—that together define the full operational pipeline from training to real-time inference on embedded hardware. Figure 1 summarizes this three-stage workflow; panels (a),(b), and (c) correspond to the Initialization, Validation, and Edge Deployment stages, respectively.

As shown in Figure 1a, the initialization step trains the YOLOv11 detector using both masked and unmasked images from the FMLD dataset to establish a baseline for masked-face detection. During this stage, Gaussian noise is added to create noise-augmented datasets that are later used to assess the benefit of denoising within the pipeline. A pre-trained DnCNN model is applied to generate reference denoised outputs. The DnCNN is intentionally not retrained, ensuring that the subsequent evaluation isolates the contribution of denoising itself and avoids dataset leakage or noise-specific overfitting.

Figure 1b illustrates the validation step, where noisy FMLD validation images are restored using 8-bit and 16-bit quantized versions of the DnCNN model. The denoised images are then processed by YOLOv11, which predicts both masked and unmasked face classes. The examples shown in the figure represent typical outcomes, where YOLOv11 correctly labels unmasked subjects as “face” and masked subjects as “mask” after the denoising stage. This step measures how quantization and denoising jointly influence detection robustness prior to deployment.

The complete denoising–detection pipeline is executed on the NVIDIA Jetson AGX Orin, as depicted in Figure 1c. Incoming images, including both masked and unmasked cases, are preprocessed on the CPU and sent to a TensorRT-based DnCNN engine for patch-wise denoising. The restored patches are reassembled into full-resolution frames

before YOLOv11 inference. A queue-based, asynchronous CPU–GPU execution structure enables the DnCNN and YOLOv11 engines to run in overlapping streams, allowing the system to achieve low-latency, real-time performance on the embedded platform.

Finally, Figure 1d outlines the visualization analysis framework, which is extensively discussed in Section 4. This panel serves as a conceptual preview of the qualitative evaluation, illustrating how the pipeline is analyzed in terms of intermediate feature preservation and detection robustness under diverse corruption scenarios. The specific visualization results corresponding to this framework are detailed later in Section 4.3

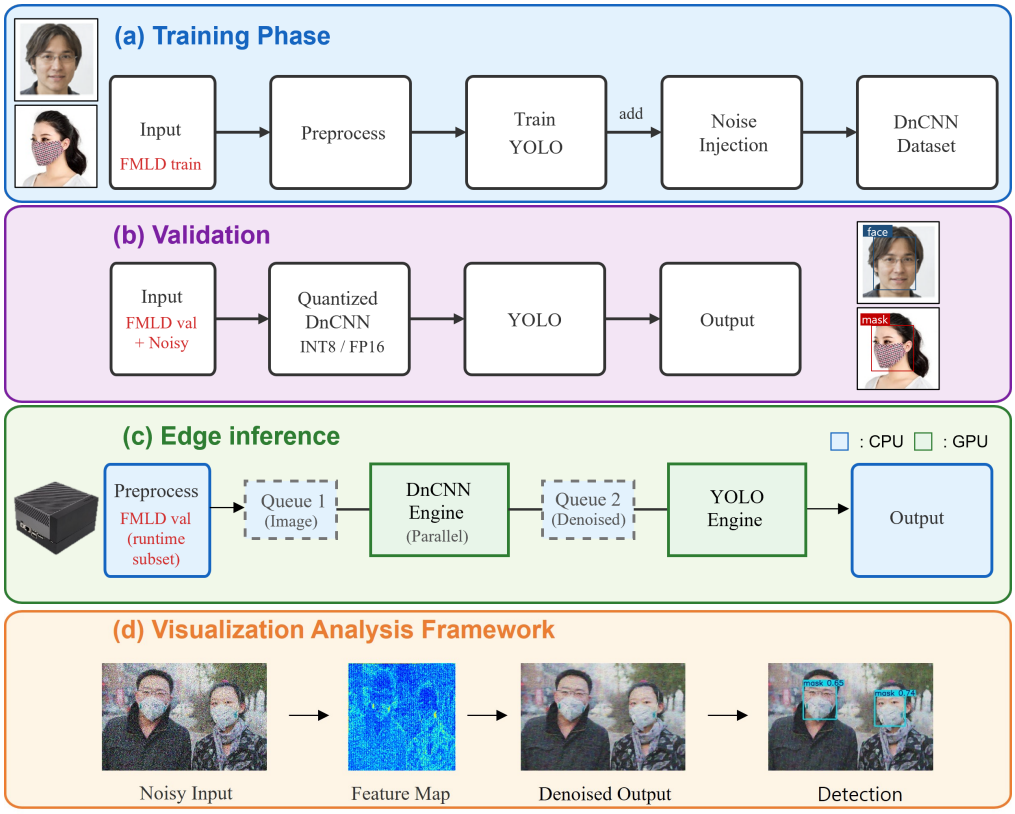


Figure 1. Overall architecture of the proposed three-stage noise-robust detection pipeline: (**a**) training phase with noise-injected data for YOLO learning, (**b**) validation step using quantized DnCNN and YOLO, (**c**) edge inference pipeline with parallel DnCNN–YOLO execution on embedded devices, (**d**) visualization analysis framework illustrating noisy inputs, intermediate feature maps, denoised outputs, and final detection results.

The following subsections describe the denoising strategy, noise construction process, quantization method, frame-reconstruction pipeline, and Jetson-based execution architecture in detail.

3.2. Denoising Strategy Using DnCNN

This subsection describes the denoising module used in the proposed pipeline. A pretrained DnCNN model is employed as a fixed restoration backbone to suppress Gaussian noise before YOLOv11 detection. The denoising process operates on each RGB channel independently, and the channels are recombined to produce a restored full-resolution image. Channel-wise inference prevents parameter growth that would occur in a joint three-channel model and reduces GPU memory usage. This decomposition also improves CPU–GPU parallelization on Jetson devices, allowing independent patch streams to be scheduled concurrently without cross-channel synchronization overhead.

3.2.1. Comparison with Other Denoising Models

To determine an appropriate denoising module for our masked-face detection pipeline, we compared three representative deep-learning–based denoisers: DnCNN, FFDNet, and SwinIR. Table 1 summarizes their quantitative performance under Gaussian noise with different σ values. SwinIR achieves the highest PSNR and SSIM across most noise levels due to its Transformer-based architecture; however, its extremely low inference speed (0.17 FPS) renders it unsuitable for real-time or edge-device applications.

Table 1. Quantitative comparison of denoising models under Gaussian noise with different σ values. Best performance per noise level is highlighted in bold.

Algorithm	Noise Level	PSNR (dB)	SSIM
DnCNN [5]	$\sigma = 15$	36.72	0.945
FFDNet [17]	$\sigma = 15$	35.62	0.936
SwinIR [16]	$\sigma = 15$	**37.69**	**0.955**
DnCNN [5]	$\sigma = 25$	34.70	0.920
FFDNet [17]	$\sigma = 25$	34.69	0.925
SwinIR [16]	$\sigma = 25$	**35.45**	**0.935**
DnCNN [5]	$\sigma = 50$	31.14	0.868
FFDNet [17]	$\sigma = 50$	31.55	0.880
SwinIR [16]	$\sigma = 50$	**32.41**	**0.897**

FFDNet provides the fastest throughput (64.91 FPS), but its usability is fundamentally limited by its non-blind design. DnCNN, in contrast, offers a balanced trade-off between restoration quality and computational efficiency, making it a practical candidate for deployment-focused pipelines.

Although SwinIR demonstrates superior restoration accuracy, its computational cost makes it unsuitable for embedded use. Meanwhile, FFDNet exhibits impressive runtime performance but requires a noise-level map as an additional input, restricting its applicability in real-world environments where noise intensity cannot be estimated.

Table 2 reports the inference speed comparison. The denoising benchmarks reported in Tables 1 and 2 follow the standard evaluation pipeline provided by the KAIR image restoration toolbox [26], which is widely used for reproducible comparison of CNN-based denoising models.

Table 2. Runtime performance of denoising models at $\sigma = 15$.

Algorithm	FPS
DnCNN [5]	17.02
FFDNet [17]	64.91
SwinIR [16]	0.17

A decisive factor in selecting DnCNN is its structural suitability for uncontrolled real-world settings. The following properties highlight its advantages:

- **Structural Difference (DnCNN vs. FFDNet):** FFDNet is a non-blind denoiser that requires a noise level σ as an additional input channel. This dependency is impractical in dynamic scenes where the noise level varies unpredictably and cannot be measured in advance.
- **Blind Denoising Capability:** DnCNN operates as a blind denoiser, removing noise without any external knowledge of σ. Its residual-learning structure allows it to handle

diverse and unknown degradation patterns, ensuring stable preprocessing across a wide range of conditions.

Following the residual-learning formulation of DnCNN [5], the adopted denoising network consists of a sequence of convolutional layers with batch normalization and ReLU activation. In this work, the same principle is applied in a channel-wise manner to facilitate parallel execution on edge devices. Let $I_c \in \mathbb{R}^{H \times W}$ denote the noisy input of the c-th color channel, where $c \in \{R, G, B\}$. For the l-th layer ($1 \leq l < L$), the intermediate feature map is computed as

$$F_c^{(l)} = \sigma\Big(\mathrm{BN}\Big(W^{(l)} * F_c^{(l-1)} + b^{(l)}\Big)\Big), \tag{1}$$

where $*$ denotes the convolution operator, $W^{(l)}$ and $b^{(l)}$ are the learnable kernels and biases, $\mathrm{BN}(\cdot)$ denotes batch normalization, and $\sigma(\cdot)$ represents the ReLU activation function. The input feature map is given by $F_c^{(0)} = I_c$. The final layer predicts the noise residual without a nonlinear activation,

$$\hat{n}_c = W^{(L)} * F_c^{(L-1)} + b^{(L)}. \tag{2}$$

The denoised output is obtained by residual subtraction,

$$\hat{I}_c = I_c - \hat{n}_c. \tag{3}$$

This channel-wise formulation enables independent denoising of each color component, reducing model complexity and facilitating parallel execution on edge devices.

Given these considerations, DnCNN provides a rational trade-off between accuracy, computational cost, and practical deployability. Accordingly, our pipeline adopts a MATLAB-pretrained DnCNN model [27]. The model was obtained using MATLAB R2024b (MathWorks, Natick, MA, USA) with the Image Processing Toolbox, providing a stable and well-validated implementation without the need for additional training.

3.2.2. Noise Construction and Parameter Definition

Additive Gaussian noise is adopted in this work not as a comprehensive model of real-world degradation, but as a controlled baseline that enables explicit parameterization of noise strength and direct correspondence with widely used denoising benchmarks. To model realistic degradation, Gaussian noise is added to RGB images normalized to the $[0, 1]$ range. Let $I = (R, G, B)$ denote a clean pixel and let $n = (n_R, n_G, n_B)$ denote an independent noise vector. The noisy pixel is generated according to

$$I_{\text{noisy}} = I + n, \tag{4}$$

where each component of n is sampled from $\mathcal{N}(0, \sigma_{\text{inj}}^2)$. The term σ_{inj}^2 represents the variance of the injected noise used in our system-level robustness evaluation.

In contrast, denoiser benchmarks such as DnCNN, FFDNet, and SwinIR typically express noise intensity using the standard deviation σ on a $[0, 255]$ scale. To relate the injected variance to this benchmark notation, the equivalent standard deviation is given by

$$\sigma_{\text{eq}} = 255\sqrt{\sigma_{\text{inj}}^2}, \tag{5}$$

consistent with standard practice in denoising studies [5,28]. This conversion clarifies how our injected noise levels correspond to common benchmark settings (e.g., $\sigma = 25$ or 50).

The preprocessing pipeline used to generate noisy and restored images is shown in Figure 2. The procedure consists of (a) additive noise injection, (b) RGB channel splitting,

(c) independent DnCNN inference per channel, and (d) channel merging to form a restored image. Each input image is first corrupted using a controlled noise model and then processed in a channel-wise manner. The RGB channels are separated and independently fed into the DnCNN model trained to predict the residual noise component. The DnCNN processes each channel independently to estimate and suppress the noise component, and the restored channels are subsequently merged to form a denoised image. This channel-wise residual learning strategy allows the denoiser to effectively suppress high-frequency noise while preserving structural image features that are critical for downstream detection. By operating independently on each channel, the pipeline avoids cross-channel interference and maintains color consistency under severe noise conditions.

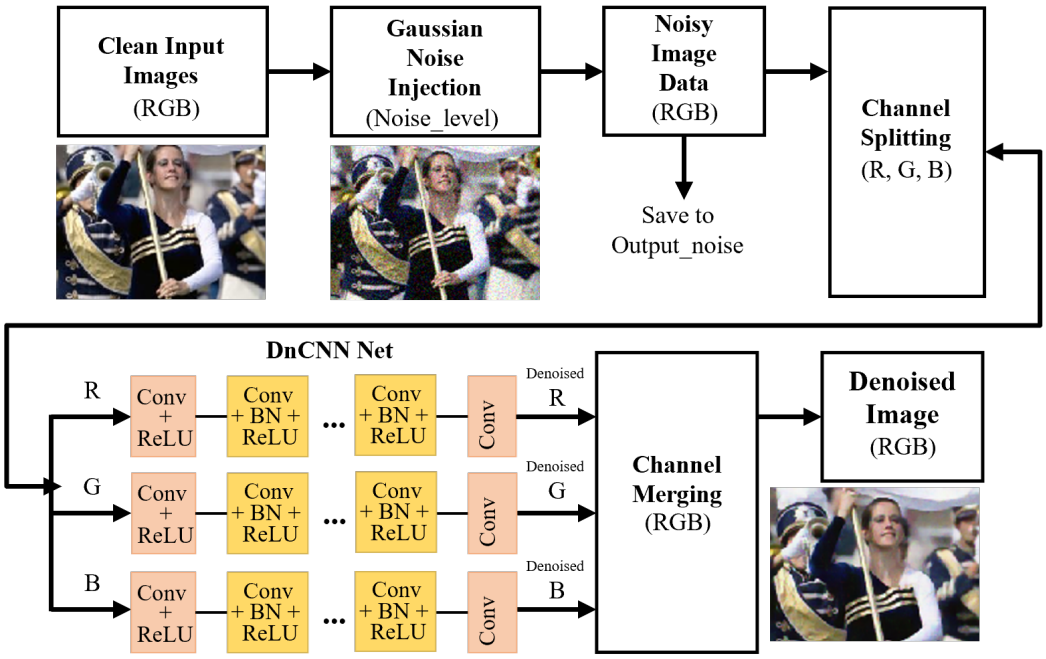


Figure 2. Architecture of the Channel-wise DnCNN Denoising Pipeline.

For evaluation, three injected noise variances are considered, $\sigma^2_{\text{inj}} \in \{0.01, 0.05, 0.10\}$, corresponding to increasing levels of degradation. The largest setting corresponds to $\sigma_{\text{eq}} \approx 80$, capturing severe sensor noise and compression artifacts often encountered in surveillance imagery, and aligns with extended evaluation protocols such as FFDNet [17].

3.3. Quantized DnCNN

To enable efficient execution on edge devices, we convert the DnCNN denoiser into low-precision formats using post-training quantization (PTQ). Quantization reduces the precision of weights and activations, lowering memory usage and enabling fast INT8 inference while preserving robust denoising capability.

3.3.1. Post-Training Quantization

Post-training quantization (PTQ) converts a pretrained floating-point model into an integer representation without additional training. Following the symmetric linear quantization scheme of Jacob et al. [7], a real-valued tensor x is quantized using a scale factor s as

$$x_q = \text{round}\left(\frac{x}{s}\right), \tag{6}$$

and the corresponding dequantized approximation is obtained by

$$x \approx s\, x_q. \tag{7}$$

The scale factor s is typically computed from the dynamic range of x using max-abs scaling for signed INT8, ensuring that the representable integer range covers the majority of the activation or weight distribution. This quantization scheme substantially reduces memory bandwidth and enables efficient INT8 inference on embedded devices such as the NVIDIA Jetson AGX Orin.

Although PTQ introduces quantization noise due to discrepancies between floating-point and integer arithmetic, DnCNN remains stable under INT8 conversion. Its residual-learning architecture inherently mitigates small perturbations in feature representations, allowing the quantized DnCNN to maintain effective denoising performance in our experiments.

3.3.2. QDQ-Based Post-Training Quantization (TensorRT INT8)

For deployment on Jetson AGX Orin, we employ TensorRT's QDQ-based PTQ pipeline [29], which inserts Quantize (Q) and Dequantize (DQ) nodes around each operator to form a hardware-optimized INT8 computation graph compliant with the ONNX Quantization Specification [30]. Scale factors for weights and activations are obtained through PTQ calibration using representative samples, with no retraining or QAT involved. The resulting QDQ INT8 engine achieves significant latency reduction while maintaining consistent denoising performance. The quantized DnCNN outputs are directly fed into YOLOv11, forming a lightweight two-stage pipeline for robust masked-face detection on edge devices.

3.4. YOLOv11-Based Masked Face Detection with Frame Reconstruction

In the proposed framework, YOLOv11 serves as the downstream detector that consumes the restored output from the quantized DnCNN (Q-DnCNN) module. Unlike standard detection pipelines that process raw input frames directly, our system incorporates an intermediate reconstruction mechanism to bridge the patch-based denoiser and the full-frame detector.

Frame Reconstruction and Input Processing: Since the Q-DnCNN module processes the input stream in localized patches to maximize GPU parallelization efficiency (see Section 3.5), the denoised patches must be spatially reassembled before detection. As illustrated in Figure 3, the CPU-based reconstruction module stitches the asynchronous stream of denoised patches into a coherent full-resolution frame. Subsequently, this restored frame is resized to the standard input resolution of 640×640 pixels required by the YOLOv11 architecture. This decoupled design ensures that the detector operates on globally consistent spatial features, which is critical for recognizing masked faces across varying scales and aspect ratios.

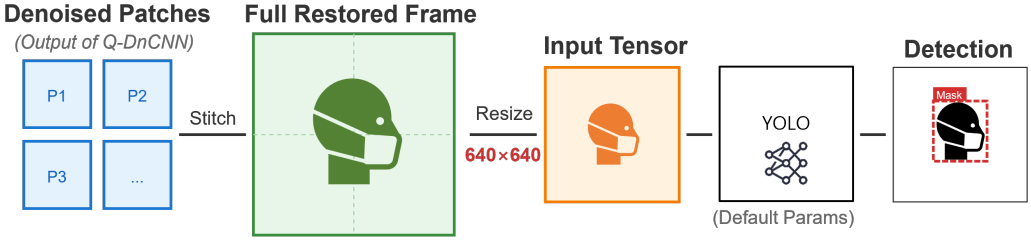


Figure 3. Frame Reconstruction and Detection Pipeline.

Training Configuration: To isolate the performance gains attributed solely to the proposed denoising preprocessing, we trained the YOLOv11 model using the default hyperparameters provided by the official Ultralytics repository. The model was trained on the clean FMLD training set without any additional architectural modifications. Using default parameters (e.g., SGD optimizer, initial learning rate of 0.01, and momentum of 0.937) ensures that the reported improvements in robustness (Section 4) result directly from the superior quality of the Q-DnCNN-enhanced input, rather than from extensive hyperparameter tuning or detector-specific optimizations.

Rationale for Choosing YOLOv11: While various lightweight detectors exist, YOLOv11 was selected for its superior trade-off between detection accuracy and computational efficiency on edge hardware. Recent comparative studies on YOLO architectures indicate that newer iterations, such as YOLOv11, not only achieve higher mAP but also exhibit improved inherent robustness against input perturbations and adversarial distortions compared to predecessors and other lightweight models [25,31]. This characteristic is particularly critical for our framework, where the detector must operate reliably on denoised outputs that may still contain residual artifacts.

3.5. Edge Device Implementation on Jetson AGX Orin

This subsection describes the edge-device implementation of the proposed framework on the NVIDIA Jetson AGX Orin platform. Since both DnCNN denoising and YOLOv11 detection are computationally intensive operations, achieving real-time performance requires a pipelined architecture that leverages multi-threaded CPU processing together with asynchronous GPU execution. Figure 4 illustrates the overall pipeline architecture adopted in this study.

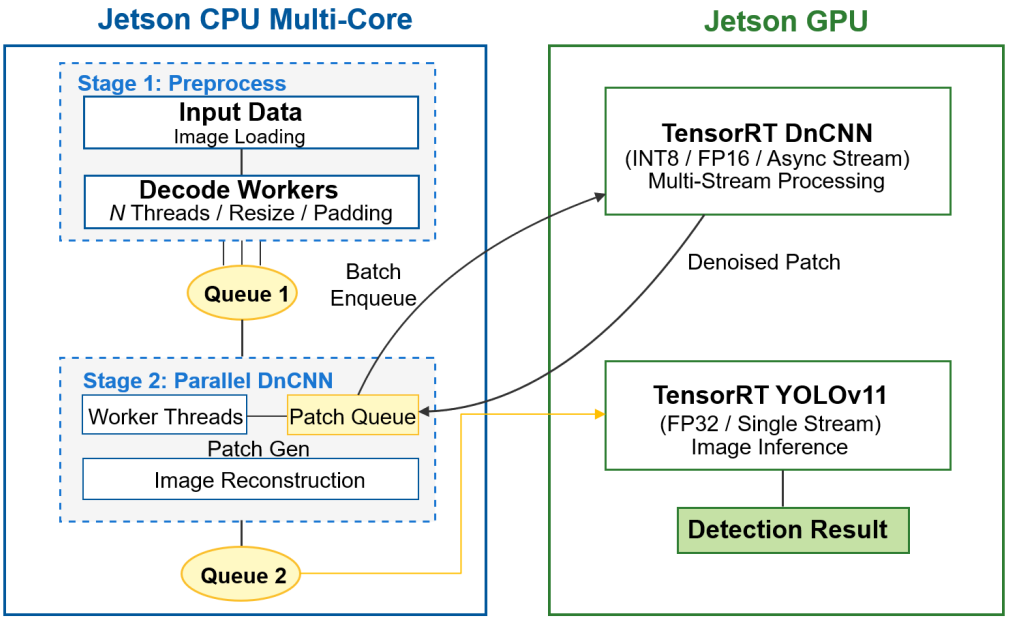


Figure 4. Pipeline Architecture: Multi-threaded CPU + Async GPU Processing on Jetson Orin AGX.

The overall design consists of two cooperating subsystems—a CPU-side preprocessing pipeline and a GPU-side inference pipeline—connected through decoupled FIFO queues. On the CPU side, incoming images are loaded and decoded by multiple worker threads, which perform resizing, normalization, and partitioning into patches. The preprocessed patches are then pushed into a patch queue that feeds the GPU-based denoising stage.

System-level parallelism is achieved by decoupling CPU-side preprocessing, result aggregation, and frame reconstruction from GPU-based inference. While the GPU exe-

cutes denoising and detection kernels, the CPU concurrently handles input acquisition, basic preprocessing, and reconstruction of denoised patches, enabling pipeline overlap across successive frames. The use of FIFO queues decouples stage execution timing and avoids frame-level synchronization barriers, allowing each stage to progress independently and efficiently.

Memory usage is managed at the frame level, and intermediate feature maps are not persistently stored. Instead, denoised patches are immediately accumulated into frame-level buffers and discarded after reconstruction, which limits memory overhead and avoids unnecessary data transfers. As a result, CPU–GPU communication overhead is amortized across the asynchronous pipeline and does not dominate the overall end-to-end latency, making the implementation suitable for real-time edge deployment under constrained computational and memory resources.

On the GPU side, two TensorRT engines operate in a pipelined manner. The DnCNN engine executes denoising using FP16 or INT8 precision and supports asynchronous inference, allowing multiple patches to be processed efficiently. Once denoising is completed, the reconstructed frame is transferred back to the CPU and pushed to the next queue for detection. The YOLOv11 engine retrieves denoised frames from this queue and performs single-stream object detection.

This hybrid execution model enables overlapping data transfer, denoising, reconstruction, and detection on the embedded platform. For example, while the GPU performs YOLOv11 inference on frame i, it can concurrently apply DnCNN denoising to patches of frame $i + 1$, ensuring high hardware utilization and reduced latency. Low-level kernel scheduling and driver-specific optimizations are intentionally abstracted, as the focus of this work is on system-level execution behavior and deployability rather than hardware-specific micro-optimizations.

3.6. Overall System Operation

Figure 5 summarizes the end-to-end operational flow of the proposed denoising–detection pipeline during deployment. Incoming frames are sequentially processed through normalization, patch-wise denoising, frame reconstruction, and masked-face detection.

Figure 5. End-to-end operation flow of the proposed denoising–detection pipeline.

In addition to quantitative performance metrics, qualitative visualization analyses are employed to examine the effects of denoising and quantization on intermediate feature representations and detection outputs. These visual comparisons are presented later in Section 4.3 to provide intuitive insights into the behavior of the proposed pipeline under noisy conditions.

From a system-level perspective, the pipeline is designed to maintain continuous throughput by allowing preprocessing and inference stages to proceed without strict frame-level synchronization. While denoising and detection are executed on different stages of the pipeline, the overall system behavior is governed by the availability of restored frames rather than individual module latency. This execution flow enables stable and predictable performance under severe noise conditions, ensuring that detection accuracy is preserved without introducing excessive end-to-end delay. As a result, the proposed framework

achieves a practical balance between robustness and real-time feasibility on embedded edge platforms.

4. Experiment Results

This section evaluates the proposed noise-resilient detection framework across two computational environments: a desktop workstation and the NVIDIA Jetson AGX Orin edge platform. Section 4.1 introduces the evaluation metrics used throughout the experiments. Section 4.2 describes the dataset preparation and baseline experimental settings. Section 4.3 analyzes noise robustness on the desktop platform under controlled Gaussian and real-world degradation conditions. Section 4.4 investigates quantization stability and detection accuracy on the Jetson AGX Orin under severe noise. Finally, Section 4.5 assesses real-time performance, throughput, and energy efficiency on the embedded platform.

4.1. Evaluation Metrics

Detection performance is evaluated using standard object detection metrics, including Precision, Recall, mAP@0.5, and mAP@0.5:0.95, which are widely adopted in object detection benchmarks. Precision and Recall are defined as

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \tag{8}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively. Mean Average Precision (mAP) is computed as the average of class-wise Average Precision (AP), where AP corresponds to the area under the precision–recall curve for each class. The mAP@0.5 metric evaluates detection performance at an Intersection-over-Union (IoU) threshold of 0.5, while mAP@0.5:0.95 further assesses robustness by averaging AP values across multiple IoU thresholds from 0.5 to 0.95 with a step size of 0.05.

4.2. Dataset Preparation and Settings

This subsection describes the baseline configuration used to assess the fundamental detection capability of YOLOv11 before introducing noise or quantization effects. All experiments in Sections 4.2 and 4.3 were performed on a desktop workstation equipped with an NVIDIA RTX 3090 GPU and an AMD Ryzen 9 processor, using the full FMLD validation set (7148 images). These baseline results serve as a reference point for the subsequent robustness and edge-deployment analyses presented in Sections 4.4 and 4.5.

FMLD [24], which integrates the MAFA and Wider Face datasets [4], contains three annotation categories that reflect real mask-wearing conditions: masked face, incorrectly masked face, and unmasked face. FMLD was selected for this study because it provides a realistic and challenging benchmark for masked face detection under adverse conditions. By integrating the MAFA and WIDER Face datasets, FMLD captures diverse real-world variations in face scale, pose, occlusion, illumination, and mask placement, including correctly worn masks, incorrectly worn masks, and unmasked faces. These characteristics make the dataset particularly suitable for evaluating noise robustness and detection stability in practical surveillance scenarios.

Table 3 summarizes the number of images and annotated instances after applying an automated bounding box correction step. To ensure training stability and evaluation reliability, we applied an automated quality-control pipeline that filters out only corrupted images and structurally invalid annotations (e.g., coordinates exceeding normalized bounds or missing label files). This procedure follows standard practices for object detection dataset preparation [32] and is recommended in official YOLO documentation to prevent parsing errors. Importantly, this step is a technical quality-assurance measure that does not alter the semantic content or class distribution of the dataset. This refinement resulted

in a minor reduction from 12,688 to 12,675 validation instances. Because these removed samples constitute a negligible proportion of the dataset, their impact on model evaluation is minimal, while the improved annotation consistency enhances the reliability of downstream detection.

Table 3. Instance Counts Before and After Bounding Box Filtering.

Dataset	Images	Instances	Masked Face	Incorrectly Masked Face	Unmasked Face
FMLD (Updated)	34,781	50,384	24,603	1204	24,576
Validation (Updated)	7148	12,675	7423	324	4928
Totals	41,934	63,059	32,026	1528	29,505

Figure 6 illustrates the training and validation box-loss curves. Both losses decrease rapidly in the initial epochs (0–20), indicating that the model quickly acquires the core localization features needed for face and mask detection. After approximately 50 epochs, the validation curve stabilizes, showing no oscillatory behavior that would signal poor generalization. The continuous decline of the training loss, contrasted with the near-flat validation loss, suggests a limited overfitting tendency; however, its magnitude is small and does not meaningfully affect the downstream experiments. These trends confirm that the YOLOv11 detector was trained stably and provides a reliable baseline for evaluating the effectiveness of the proposed denoising–detection pipeline. The training and validation losses exhibit similar convergence patterns, and no increasing gap is observed as training progresses, suggesting that overfitting is effectively controlled under the current training protocol.

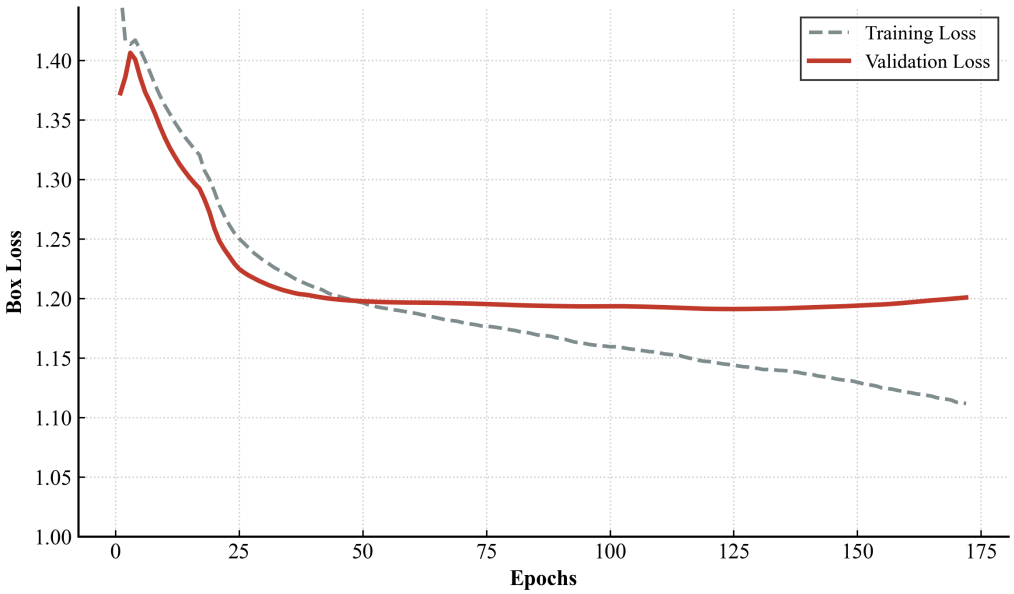


Figure 6. Training and Validation Box Loss Curves.

The normalized confusion matrix for the three-class prediction task is shown in Figure 7. The detector achieves strong classification performance for the *mask* (0.97) and *face* (0.83) categories, while performance for the *incorrectly masked face* category is notably lower (true positive rate of 0.65). Unlike a standard closed-set classifier, the columns of this detection-oriented confusion matrix do not sum to one because false negatives arising from missed detections (bounding box not generated) are accumulated outside the matrix. As a result, the matrix more accurately reflects the detector's localization behavior under partially occluded or irregular mask-placement conditions.

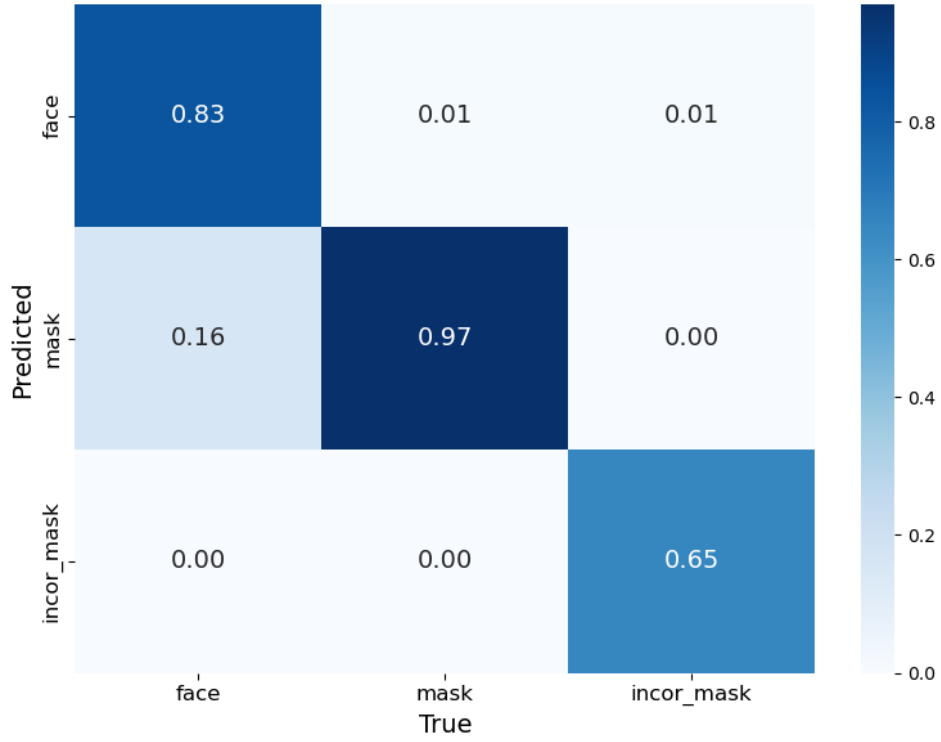


Figure 7. Normalized Confusion Matrix.

Performance per class is detailed in Table 4. The overall mAP@0.5 reaches 0.859 across the full validation set, demonstrating strong baseline capability. The *mask* class exhibits the highest performance (mAP@0.5 = 0.978, recall = 0.955), confirming that properly worn masks are consistently detected. Conversely, the *incorrectly masked face* class yields a lower mAP@0.5 of 0.720, reflecting the difficulty of detecting subtle variations in mask misuse. Because these statistics are computed over all 7,148 validation images, they carry sufficient statistical significance and constitute a robust reference point for evaluating noise robustness in later sections.

Table 4. Face Mask Detection Performance of YOLOv11 by Class.

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Face	0.817	0.824	0.880	0.623
Mask	0.940	0.955	0.978	0.654
Incorrect Mask	0.796	0.670	0.720	0.443
Average	**0.851**	**0.816**	**0.859**	**0.573**

4.3. Noise Robustness on Desktop

This subsection evaluates how synthetic Gaussian noise affects the detection performance of YOLOv11 and examines the extent to which DnCNN preprocessing— in both full-precision and quantized forms—recovers accuracy. All experiments here were conducted exclusively on the desktop workstation described in Section 4.1, ensuring that the analysis isolates the intrinsic noise sensitivity of the detector without hardware-dependent effects.

Table 5 reports Precision, Recall, mAP@0.5, and mAP@0.5:0.95 under three noise levels ($\sigma^2_{inj} \in \{0.01, 0.05, 0.10\}$). Gaussian noise was injected using the Albumentations library, a widely adopted and reproducible data augmentation framework for computer vision experiments [33]. For the real-world distortion scenarios (Motion Blur, Low Illumination, and JPEG Compression) presented in Table 5, we adopted the severity levels

L1, L3, and L5 following the corruption benchmark protocol established by Hendrycks and Dietterich [34] Across all metrics, noise substantially degrades performance. Under severe noise ($\sigma^2_{inj} = 0.10$), precision drops from 0.851 to 0.577, recall from 0.816 to 0.214, and mAP@0.5 from 0.859 to 0.262. These results confirm that YOLOv11 is highly vulnerable to high-frequency perturbations, leading to missed detections and unstable localization.

Table 5. Performance comparison of YOLOv11 with the proposed Q-DnCNN framework. The table compares the Baseline (Noise only), Full-Precision (FP32), and Quantized models (16-bit, 8-bit) to analyze the trade-off between precision and detection accuracy (Desktop).

Distortion	Intensity	Method	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Baseline	–	Original YOLOv11	0.851	0.816	0.859	0.573
Primary Target: Gaussian Noise						
Gaussian Noise	$\sigma^2 = 0.01$	Noise only	0.830	0.712	0.793	0.510
		DnCNN (FP32)	0.842	**0.800**	0.844	**0.619**
		Q-DnCNN (FP16)	**0.849**	0.793	**0.846**	0.561
		Q-DnCNN (INT8)	0.841	0.780	0.760	0.554
	$\sigma^2 = 0.05$	Noise only	0.742	0.416	0.489	0.295
		DnCNN (FP32)	**0.831**	**0.733**	**0.812**	**0.527**
		Q-DnCNN (FP16)	0.819	0.722	0.801	0.517
		Q-DnCNN (INT8)	0.761	0.700	0.768	0.492
	$\sigma^2 = 0.10$	Noise only	0.577	0.214	0.262	0.151
		DnCNN (FP32)	**0.751**	**0.655**	**0.723**	**0.456**
		Q-DnCNN (FP16)	0.732	0.637	0.704	0.441
		Q-DnCNN (INT8)	0.708	0.598	0.659	0.414
Verification: Real-world Distortions						
Motion Blur	L1	Noise only	0.806	0.594	0.676	0.426
	L3	Noise only	0.677	0.348	0.406	0.235
	L5	Noise only	0.578	0.208	0.241	0.130
Low Illumination	L1	Noise only	0.850	0.815	0.857	0.572
	L3	Noise only	0.831	0.797	0.838	0.545
	L5	Noise only	0.802	0.710	0.761	0.470
JPEG Compression	L1	Noise only	0.846	0.815	0.855	0.572
	L3	Noise only	0.843	0.811	0.854	0.570
	L5	Noise only	0.843	0.807	0.850	0.567

Applying DnCNN effectively restores performance across all noise levels. For $\sigma^2_{inj} = 0.10$, the denoiser increases precision from 0.577 to 0.751, recall from 0.214 to 0.655, and mAP@0.5 from 0.262 to 0.723—recovering more than 60% of the performance lost due to noise. This substantial improvement indicates that DnCNN successfully suppresses noise-induced distortions and restores the structural cues required for stable detection.

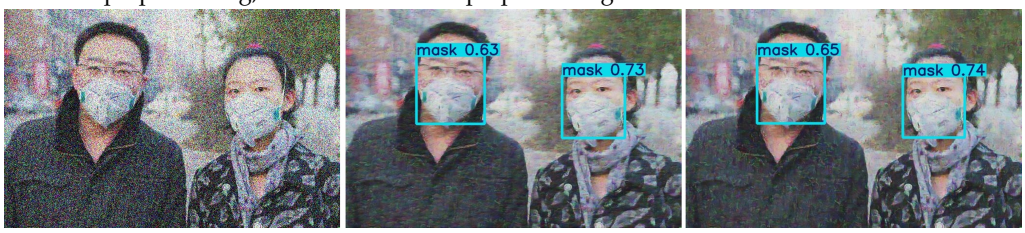
An additional observation arises under mild noise ($\sigma^2_{inj} = 0.01$): the quantized 16-bit and 8-bit DnCNN variants achieve precision values (0.849 and 0.841) comparable to or slightly higher than the full-precision model (0.842). A similar trend is observed for mAP@0.5 (0.846 and 0.843 vs. 0.844). This counterintuitive behavior suggests that reduced bit precision may act as an implicit regularizer. By constraining the representational dynamic range, quantization suppresses minor fluctuations in the activation space and stabilizes inference under weak perturbations, consistent with prior studies on quantization regularization [8]. Recent work further shows that quantization reduces overfitting in noisy environments [9] and enhances consistency in low-precision networks [10], which aligns with the observed trend that Q-DnCNN matches or slightly exceeds floating-point performance at low noise levels.

As noise severity increases, quantized models exhibit a gradual performance drop relative to full-precision DnCNN, as expected from the reduced numerical resolution. However, even at $\sigma^2_{inj} = 0.10$, the 16-bit and 8-bit variants preserve a meaningful portion of the denoising benefit (mAP@0.5 = 0.704 and 0.659), demonstrating that quantization does not undermine the fundamental noise-removal capability of DnCNN.

To complement the quantitative results in Table 5, Figure 8 presents qualitative visualizations illustrating how denoising preprocessing stabilizes feature representations and detection outputs under severe noise. In particular, backbone feature maps are visualized to provide insight into how high-frequency noise affects intermediate representations and how denoising preprocessing modulates these responses prior to detection. The detection results in Figure 8c reflect the differences observed in the preceding restoration and feature map visualizations, illustrating how changes in intermediate representations are manifested at the final detection stage. This qualitative comparison provides visual context for the quantitative performance trends reported in Table 5, without replacing the metric-based evaluation.

(**a**) Visual comparison of image restoration results, including the noisy input, the FP16 DnCNN output, and the INT8 DnCNN output.

(**b**) Visualization of YOLOv11 backbone feature maps corresponding to the noisy input, FP16 DnCNN preprocessing, and INT8 DnCNN preprocessing.

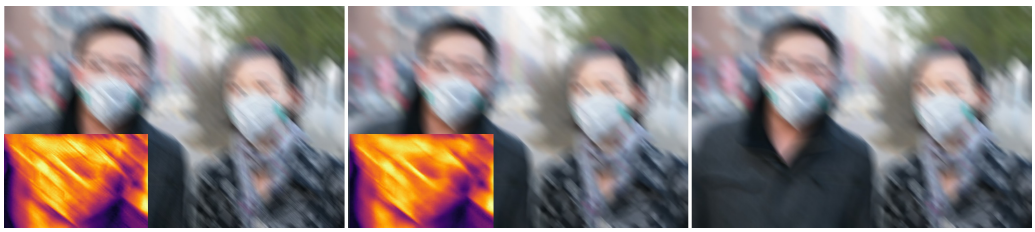


(**c**) Final masked-face detection results produced by YOLOv11 for each input condition.

Figure 8. Qualitative comparison of the proposed denoising–detection pipeline under severe Gaussian noise ($\sigma^2 = 0.10$).

To further validate the practical applicability of the proposed framework, we extended the qualitative analysis to real-world degradation scenarios beyond Gaussian noise. Figure 9 presents representative denoising and detection results under severe motion blur, JPEG compression, and low illumination conditions. While the visual restoration of motion-blurred images (Figure 9a) remains inherently challenging due to the design of DnCNN for

additive noise removal, the proposed pipeline consistently preserves essential structural cues required for detection.



(**a**) Robustness against severe motion blur (Level 5).

(**b**) Robustness against JPEG compression artifacts (Level 5).

(**c**) Robustness against low-illumination conditions (Level 5).

Figure 9. Qualitative robustness evaluation under severe real-world degradations (Severity Level 5) using CLAHE-enhanced ROI heatmaps. From left to right: degraded input, INT8-quantized DnCNN output, and YOLO detection result. Enlarged insets show magnified mask regions.

To support a clearer interpretation of this structural preservation, enhanced ROI heatmaps are provided in Figure 9. Specifically, contrast-limited adaptive histogram equalization (CLAHE) is applied to cropped mask regions to amplify local contrast, thereby revealing noise distribution patterns and fine structural details that are less discernible in raw RGB images. These visualizations indicate that the INT8-quantized denoising retains critical ROI structures across all degradation types, enabling stable bounding box localization. Consequently, the detector successfully localizes masked faces under all tested scenarios, reinforcing the robustness of the proposed quantized inference pipeline against diverse real-world environmental distortions.

Overall, these desktop observation results provide a hardware-neutral assessment of noise robustness and establish a controlled baseline for interpreting the quantized inference behavior on the Jetson AGX Orin in Section 4.4.

4.4. Quantization Stability on Jetson AGX Orin

This subsection investigates how reduced numerical precision affects denoising robustness and downstream detection accuracy when deploying the proposed pipeline on the NVIDIA Jetson AGX Orin. In contrast to the desktop evaluation in Section 4.2, which analyzes noise robustness under controlled conditions, this section focuses on the stability of detection accuracy across FP16 and INT8 DnCNN variants on an embedded platform.

Unless otherwise specified, all detection accuracy metrics (Precision, Recall, mAP@0.5, and mAP@0.5:0.95) reported in this subsection are computed over the full FMLD validation set (7148 images). However, to enable controlled and repeatable runtime profiling on the

embedded platform, runtime-related measurements (FPS, power consumption, energy efficiency) are evaluated using a fixed, class-balanced subset of 100 validation images.

Table 6 reports detection performance under severe Gaussian noise ($\sigma^2_{inj} = 0.10$), together with the corresponding throughput for reference. YOLO-only inference exhibits a severe degradation in robustness: although precision remains high (0.8921), recall collapses to 0.1427 due to missed detections, resulting in a substantial drop in localization accuracy (mAP@0.5 = 0.5223). This phenomenon is consistent with the noise sensitivity trends observed in Section 4.2 and highlights the necessity of denoising for stable detection on edge hardware.

Table 6. Combined quantitative comparison of detection performance and runtime efficiency on Jetson AGX Orin under severe noise ($\sigma^2_{inj} = 0.10$). The proposed INT8 Parallel pipeline achieves a favorable trade-off between accuracy and speed.

Model/Setting	Precision	Recall	mAP@0.5	mAP@0.5:0.95	FPS
YOLO (Clean)	0.8649	0.8031	0.8426	0.5893	14.58
YOLO (Noise)	0.8921	0.1427	0.5223	0.3238	26.09
DnCNN FP16 (Serial)	0.8387	0.4269	0.6427	0.4369	4.42
DnCNN FP16 (Parallel)	0.8465	**0.4274**	**0.6464**	0.4397	5.82
DnCNN INT8 (Serial)	0.8384	0.4172	0.6367	0.4364	6.24
DnCNN INT8 (Parallel) (Ours)	**0.8471**	0.4154	0.6402	**0.4407**	**7.66**

When DnCNN preprocessing is applied, detection performance is substantially restored. The FP16 variant achieves an mAP@0.5 of 0.6464, effectively mitigating the noise impact. Notably, the INT8 model maintains competitive accuracy (mAP@0.5 = 0.6402), with less than 1% degradation relative to FP16, demonstrating that aggressive quantization does not compromise denoising robustness on the embedded platform.

This behavior is consistent with prior observations on quantization-induced regularization effects. By constraining the dynamic range of activations, quantization can suppress minor perturbations and promote more stable feature representations [8]. Related studies further report that quantization reduces overfitting in noisy or perturbed environments [9] and improves in-distribution consistency in low-precision networks [10]. In addition, integer-arithmetic inference has been shown to preserve semantic fidelity with minimal degradation in detection backbones [7]. While this effect is not claimed as a novel theoretical contribution, the empirical stability observed here aligns well with these prior findings.

Overall, the results demonstrate that INT8 quantization preserves the robustness of DnCNN-based denoising while maintaining detection accuracy comparable to higher-precision models.

4.5. Real-Time Edge Efficiency on Jetson AGX Orin

This subsection evaluates whether the proposed denoising–detection pipeline can operate effectively in real time on the NVIDIA Jetson AGX Orin. While Section 4.3 analyzed detection accuracy stability under different quantization levels, the present analysis focuses on throughput, power consumption, and energy efficiency, which are critical metrics for power- and resource-constrained embedded systems.

Table 7 summarizes the throughput, power usage, and energy efficiency (FPS/W) for each pipeline configuration. A subset of 100 validation images was used to measure stable power metrics. Notably, the baseline detector under noise achieves 26.09 FPS, effectively satisfying standard high-speed real-time requirements ($\geq$20 FPS). However, as discussed in Section 4.3, this speed gain comes at the cost of a severe collapse in detection performance, making the noise-only condition impractical.

Table 7. Performance comparison of runtime throughput, power consumption, and energy efficiency on Jetson AGX Orin. Energy metrics are measured over the validation subset (100 frames). The proposed INT8 Parallel pipeline demonstrates the best balance, achieving 1.222 FPS/W.

Pipeline	FPS	Avg Power (W)	Max Power (W)	Energy (Wh)	FPS/W
YOLO-only (Clean)	14.58	5.59	6.99	0.010	2.671
YOLO-only (Noise)	26.09	6.77	7.19	0.007	4.000
DnCNN FP16 (Parallel)	5.82	5.79	6.08	0.028	0.982
DnCNN INT8 (Parallel)	**7.66**	**5.93**	**6.18**	**0.023**	**1.222**

To restore accuracy within a reasonable computational budget, our pipeline leverages parallel CPU–GPU execution and INT8 quantization. As shown in Table 7, introducing DnCNN restores robustness but incurs computational overhead. However, the proposed **INT8 quantization significantly mitigates this burden**. The **INT8 Parallel pipeline achieves 7.66 FPS**, representing a **31.6% throughput improvement** over the FP16 Parallel baseline (5.82 FPS). This confirms that integer-arithmetic inference effectively accelerates the denoising workload on the Jetson edge platform.

Furthermore, the energy efficiency analysis highlights the benefits of quantization. While the YOLO-only baseline exhibits high throughput-to-power efficiency (4.00 FPS/W), it fails to detect targets under noise. Among the denoising pipelines, the proposed INT8 parallel model achieves an efficiency of **1.222 FPS/W**, representing a **24.4% improvement** over the FP16 implementation (0.982 FPS/W). This indicates that INT8 quantization not only accelerates inference speed but also effectively reduces the energy cost per frame, enhancing the sustainability of edge surveillance systems.

It should be noted that real-time object detection does not universally require 30 FPS operation; depending on the application, frame processing rates of approximately 3–10 FPS can be sufficient when end-to-end latency remains bounded, as reported in prior studies [35]. By balancing throughput (7.7 FPS), accuracy (mAP retention), and energy efficiency (1.222 FPS/W), the proposed INT8+Parallel framework offers the most favorable configuration for robust real-time deployment on the Jetson AGX Orin.

5. Discussion

This section discusses the empirical observations, limitations, and system-level implications of the proposed noise-resilient masked-face detection framework.

5.1. Quantization Effects and Interpretation

An important empirical observation from our experiments is that low-bit quantized DnCNN models occasionally exhibit performance comparable to, or slightly exceeding, their full-precision counterparts under mild noise conditions. Similar behaviors have been reported in prior studies on quantized inference, where reduced numerical precision constrains activation dynamics and limits sensitivity to small perturbations. It is emphasized that this phenomenon is reported here as an empirical observation at the system level, rather than as a claimed noise suppression mechanism or theoretical contribution. The primary contribution of this work lies in the system-level integration and validation of quantized denoising for robust detection, rather than in proposing a new regularization principle.

It should be noted that the observed robustness gain under quantization should not be interpreted as a definitive noise suppression mechanism. Rather, quantization constrains the numerical dynamic range of activations, which may indirectly stabilize inference under mild noise conditions by reducing sensitivity to small perturbations. This explanation is provided as a retrospective and empirical interpretation rather than a claimed theoretical contribution, and alternative interpretations—such as effective model capacity reduc-

tion—cannot be excluded. Accordingly, the reported behavior should be interpreted within this bounded empirical context, rather than as evidence of a causal regularization effect.

Beyond the empirical observations, a plausible interpretation of the improved performance of quantized models under low-noise conditions can be discussed from the perspective of numerical stability and activation distribution compression. Low-bit quantization effectively limits the dynamic range of intermediate activations, which suppresses minor fluctuations caused by residual noise and prevents excessive amplification of such variations through successive layers. When the input noise level is moderate, this implicit constraint can stabilize feature propagation and lead to more consistent inference behavior. As noise becomes more severe, however, such compression may also attenuate semantically meaningful features, diminishing its beneficial effect. This suggests that the apparent robustness gain of quantized inference is most pronounced within a practical noise regime and may reflect a form of survivorship bias in the observed operating range, rather than a universal robustness property under extreme corruption conditions.

5.2. Noise Modeling and Robustness Scope

In this study, additive Gaussian noise was adopted as a controlled baseline to enable reproducible robustness analysis and direct comparison with standard denoising benchmarks. Gaussian corruption provides a well-established proxy for high-frequency sensor noise and compression artifacts, allowing systematic evaluation of noise-induced performance degradation.

To reflect realistic operating conditions, noise variance was incrementally increased to analyze robustness across distinct degradation regimes. Qualitative and quantitative results indicate that up to $\sigma^2_{\text{inj}} = 0.10$, the proposed pipeline maintains structurally consistent facial representations sufficient for reliable detection, even under visually severe corruption. This range therefore represents the upper bound of stable operation for edge-based vision sensors, where perceptual recognition remains possible despite significant noise contamination.

Although real-world degradations often involve mixed effects such as motion blur, low illumination, and compression artifacts, the controlled Gaussian setting allows isolation of noise-related failure mechanisms. Complementary qualitative evaluations under such non-Gaussian conditions further suggest that the proposed denoising–detection pipeline preserves essential structural cues beyond the strict Gaussian assumption, supporting its practical applicability.

5.3. Performance Boundaries and Failure Modes

Figure 10 illustrates the progressive degradation of detection performance as noise severity increases beyond the stable operating range, using an 8-bit quantized DnCNN followed by YOLO-based detection.

At $\sigma^2_{\text{inj}} = 0.10$ (Figure 10a), detection remains stable and confident for both masked faces. Although the input exhibits heavy grain noise, core facial structures such as contours and mask boundaries are sufficiently preserved by the denoising stage, allowing the detector to operate within a practical "safe zone." This noise level therefore defines the effective operational limit of the proposed pipeline.

As noise increases to $\sigma^2_{\text{inj}} = 0.15$ (Figure 10b), the system enters a transition regime where denoising begins to introduce over-smoothing effects. While detections are still produced, confidence scores become unstable, particularly for faces with weaker contrast or partial occlusion. For example, the confidence of the left masked face decreases from 0.63 to 0.37, indicating the onset of structural attenuation that directly impacts detection reliability.

(**a**) Limit ($\sigma^2_{\text{inj}} = 0.10$): Safe Zone. The system successfully detects both masked faces with high confidence, representing the effective operational limit.

(**b**) Failure Start ($\sigma^2_{\text{inj}} = 0.15$): Over-smoothing. As noise increases, the denoised output exhibits excessive smoothing, causing a significant drop in detection confidence (e.g., left face: 0.63 $\rightarrow$ 0.37), indicating the onset of instability.

(**c**) Total Failure ($\sigma^2_{\text{inj}} = 0.20$): Feature Collapse. Under extreme corruption, the structural features of the left face are lost (Feature Collapse), resulting in a missed detection.

Figure 10. Qualitative visualization of system failure boundaries and performance degradation under extreme noise conditions.

Under extreme corruption at $\sigma^2_{\text{inj}} = 0.20$ (Figure 10c), denoising is no longer able to recover semantically meaningful facial features. Patch-wise residual estimation collapses fine-scale structures, resulting in feature collapse and eventual missed detections. This failure mode is characterized not by erroneous classifications, but by the absence of detectable bounding boxes, reflecting a fundamental loss of discriminative information.

These observations indicate that performance degradation beyond $\sigma^2_{\text{inj}} = 0.10$ arises from intrinsic limitations of image restoration under extreme noise, rather than from deficiencies in the proposed framework. The transition from noise-robust operation to feature-collapse-driven failure defines a practical system boundary, beyond which preprocessing and detection are no longer effective. Systematic extension to adaptive noise modeling or cross-patch contextual restoration is therefore identified as an important direction for future work.

5.4. System-Level Implications for Edge Deployment

From a system perspective, the proposed framework is directly applicable to a range of edge-based vision systems, including intelligent surveillance cameras, access-control terminals, healthcare monitoring platforms, and perception modules for human–robot interaction. In such systems, robustness to image degradation, low-latency inference, and limited computational resources are critical constraints. The proposed noise-aware and quantization-friendly design explicitly addresses these constraints through lightweight preprocessing, low-bit inference, and parallel CPU–GPU execution.

The current evaluation focuses on a single-stream, fixed-resolution scenario, which reflects a common operational mode for embedded vision sensors. Specifically, the patch size was determined through preliminary profiling to identify the optimal operating point that balances GPU compute density with memory transfer latency. We observed that the selected patch dimension maximizes the throughput of the asynchronous CPU–GPU pipeline on the Jetson AGX Orin. Deviating from this optimal size (e.g., larger patches) resulted in memory bandwidth saturation and increased single-inference latency, which disrupted the continuous flow of the asynchronous pipeline. Therefore, the patch size was treated as a fixed hardware-aware design parameter to ensure stable real-time performance, rather than a tunable hyperparameter. Extensions to dynamic sizing or multi-stream scheduling are considered valuable directions for future investigation.

Power consumption was evaluated using the NVIDIA Jetson AGX Orin's built-in monitoring interfaces during runtime execution, rather than external power measurement instrumentation. The reported average and peak power values in Table 7 therefore reflect system-level readings collected under fixed power mode and workload conditions, and are intended to provide a relative comparison of energy efficiency across pipeline configurations rather than absolute power characterization. While such measurements may be affected by platform-specific variability, they are sufficient for comparing the relative efficiency of FP16 and INT8 pipelines under identical experimental settings.

A direct speed–accuracy comparison with other lightweight denoising models on the same edge platform was not conducted in this study. While such an evaluation would further strengthen the real-time performance analysis, the selection of DnCNN was motivated by its well-established balance between restoration quality, architectural simplicity, and blind denoising capability, as demonstrated in the desktop benchmarks. In contrast, alternative models such as FFDNet require explicit noise-level estimation, and transformer-based models incur substantial computational overhead that limits their suitability for embedded deployment. Comprehensive edge-level comparisons across denoising architectures are therefore identified as an important direction for future work.

6. Conclusions

This study proposed a noise-resilient masked-face detection framework that integrates DnCNN-based image denoising with the YOLOv11 detector, together with low-bit quantization and an optimized edge-device execution pipeline. Experimental results demonstrated that high-frequency noise severely degrades masked-face detection performance, and that lightweight residual denoising prior to detection substantially improves robustness under moderate to severe degradation.

Comprehensive evaluations showed that 16-bit and 8-bit quantized denoisers preserve most of the denoising benefit while significantly reducing computational cost. Edge deployment experiments on the NVIDIA Jetson AGX Orin further confirmed that quantization and parallel CPU–GPU execution enable near–real-time operation under noisy conditions, providing a practical foundation for deployable edge-AI systems.

Future work will focus on extending the proposed framework to continuous video pipelines, multi-stream scenarios, and broader real-world degradation models. Furthermore, we plan to integrate the framework into concrete edge applications such as intelligent surveillance cameras, access-control terminals, and healthcare monitoring platforms. Exploring its applicability to multi-task facial analysis—including facial landmark detection, identity recognition, and physiological signal estimation—represents another promising direction for future research.

Author Contributions: Conceptualization, R.C. and M.Y.K.; Methodology, R.C., S.K. and M.Y.K.; Software, R.C. and S.K.; Validation, R.C., B.-s.K. and H.L.; Formal analysis, R.C. and B.-s.K.; Investigation, R.C. and B.-s.K.; Resources, M.Y.K. and H.L.; Technical support, H.L.; Data curation, R.C.; Writing—original draft preparation, R.C.; Writing—review and editing, M.Y.K., B.-s.K., S.K. and H.L.; Visualization, R.C.; Supervision, M.Y.K.; Project administration, M.Y.K.; Funding acquisition, M.Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the DGIST R&D Program of the Ministry of Science and ICT (25-IT-02). This research was supported by the Regional Innovation System & Education (RISE) Glocal 30 program through the Daegu RISE Center, funded by the Ministry of Education (MOE) and the Daegu, Republic of Korea (2025-RISE-03-001).

Data Availability Statement: The Face Mask Label Dataset (FMLD) used in this study is publicly available at https://github.com/borutb-fri/FMLD. The pretrained DnCNN model is provided by MathWorks (https://www.mathworks.com/help/images/ref/denoisingnetwork.html). Noise injection and denoising benchmarks were implemented using publicly available toolboxes, including the KAIR image restoration framework (https://github.com/cszn/KAIR) and the Albumentations library (https://github.com/albumentations-team/albumentations, all accessed on 4 December 2025). The source code for the proposed edge-deployment pipeline is available from the corresponding author upon reasonable request due to hardware-specific dependencies on NVIDIA Jetson platforms.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive and insightful comments, which greatly improved the quality of this manuscript. We also express our gratitude to the Intelligent Robotics Research Division at DGIST for providing technical support and experimental resources. Finally, the first author would like to acknowledge Mercury and Depence for their continuous encouragement and support during the preparation of this study.

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

References

1. Chen, L.; Wang, H.; Wang, X.; Gao, J.; Deng, W. The Devil of Face Recognition Is in the Noise. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 765–782.
2. Esmaeilzehi, A.; Ahmad, M.O.; Swamy, M.N.S. SRNHARB: A deep light-weight image super resolution network using hybrid activation residual blocks. *Signal Processing: Image Commun.* **2021**, *99*, 116509. [CrossRef]
3. Guo, Y.; Zhang, L.; Hu, Y.; He, X.; Gao, J. A Dataset and Benchmark for Large-Scale Face Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4873–4882.
4. Batagelj, B.; Peer, P.; Struc, V.; Dobrisek, S. How to Correctly Detect Face-Masks for COVID-19 from Visual Information? *Appl. Sci.* **2021**, *11*, 2070. [CrossRef]
5. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]
6. He, L.; Zhou, Y.; Liu, L.; Ma, J. Research and Application of YOLOv11-Based Object Segmentation in Intelligent Construction-Site Recognition. *Buildings* **2024**, *14*, 3777. [CrossRef]
7. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713.
8. Nagel, S.; Shao, R.; Gouk, H.; Hospedales, T.M. QReg: On Regularization Effects of Quantization. *arXiv* **2022**, arXiv:2206.12372. [CrossRef]
9. Xu, Y.; Wang, Z.; Li, Z.; Xu, Y.; Tao, D. Fighting Overfitting with Quantization for Deep Neural Networks on Noisy Labels. *arXiv* **2023**, arXiv:2303.11803.
10. Wang, R.; Tang, Y.; Gong, C.; Liu, Y. In-Distribution Consistency Regularization for QAT Generalization. *arXiv* **2024**, arXiv:2402.13497.
11. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image Denoising by Sparse 3D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [CrossRef] [PubMed]
12. Pande-Chhetri, R.; Abd-Elrahman, A. De-Striping Hyperspectral Imagery Using Wavelet Transform and Adaptive Frequency Domain Filtering. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 620–636. [CrossRef]

13. Cui, H.; Jia, P.; Zhang, G.; Jiang, Y.-H.; Li, L.-T.; Wang, J.-Y.; Hao, X.-Y. Multiscale Intensity Propagation to Remove Multiplicative Stripe Noise From Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 2308–2323. [CrossRef]
14. Chang, Y.; Yan, L.; Liu, L.; Fang, H.; Zhong, S. Infrared Aerothermal Nonuniform Correction via Deep Multiscale Residual Network. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1120–1124. [CrossRef]
15. Sun, W.; Ren, K.; Meng, X.; Yang, G.; Xiao, C.; Peng, J.; Huang, J. MLR-DBPFN: A Multi-scale Low Rank Deep Back Projection Fusion Network for Anti-noise Hyperspectral and Multispectral Image Fusion. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5522914. [CrossRef]
16. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 1833–1844.
17. Zhang, K.; Zuo, W.; Zhang, L. FFDNet: Toward a Fast and Flexible CNN-Based Image Denoiser. *IEEE Trans. Image Process.* **2018**, *27*, 4608–4622. [CrossRef] [PubMed]
18. Guo, S.; Li, Q.; Zuo, W. Toward Convolutional Blind Denoising of Real Photographs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1712–1722.
19. Li, L.; Jiang, L.; Zhang, J.; Wang, S.; Chen, F. A Complete YOLO-Based Ship Detection Method for Thermal Infrared Remote Sensing Images under Complex Backgrounds. *Remote Sensing* **2022**, *14*, 1534. [CrossRef]
20. Rodríguez-Rodríguez, J.A.; López-Rubio, E.; Ángel-Ruiz, J.A.; Molina-Cabello, M.A. The Impact of Noise and Brightness on Object Detection Methods. *Sensors* **2024**, *24*, 821. [CrossRef] [PubMed]
21. Li, J.; Zhang, S.; Zhang, X.; Wang, A. DiffuYOLO: A Novel Method for Small Vehicle Detection in Remote Sensing Based on Diffusion Models. *Alex. Eng. J.* **2025**, *114*, 485–496. [CrossRef]
22. Liu, Y.; Li, S.; Zhou, L.; Liu, H.; Li, Z. Dark-YOLO: A Low-Light Object Detection Algorithm Integrating Multiple Attention Mechanisms. *Appl. Sci.* **2025**, *15*, 5170. [CrossRef]
23. Ge, S.; Li, J.; Ye, Q.; Luo, Z. Detecting Masked Faces with LLE-CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2682–2690.
24. Batagelj, B. Face Mask Label Dataset (FMLD). GitHub Repository. 2024. Available online: https://github.com/borutb-fri/FMLD (accessed on 3 September 2025).
25. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. In Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 10–16 December 2024; Curran Associates, Inc.: Red Hook, NY, USA, 2024; Volume 37.
26. Zhang, K.; Zuo, W.; Zhang, L. KAIR: Image Restoration Toolbox for Reproducible Research. GitHub Repository. 2021. Available online: https://github.com/cszn/KAIR (accessed on 3 September 2025).
27. MathWorks. Denoising Network (MATLAB R2024b, Image Processing Toolbox). Available online: https://www.mathworks.com/help/images/ref/denoisingnetwork.html (accessed on 12 February 2025).
28. Ke, R. Deep Variation Prior: Joint Image Denoising and Noise Variance Estimation Without Clean Data. *IEEE Trans. Image Process.* **2024**, *33*, 2908–2923. [CrossRef] [PubMed]
29. NVIDIA Corporation. *TensorRT Developer Guide: Quantization and Calibration*; NVIDIA: Santa Clara, CA, USA, 2023.
30. ONNX Working Group. ONNX Quantization Specification (Q/DQ Format). 2022. Available online: https://github.com/onnx/onnx/blob/main/docs/Operators.md (accessed on 17 December 2025).
31. Apostolidis, K.D.; Papakostas, G.A. Delving into YOLO Object Detection Models: Insights into Adversarial Robustness. *Electronics* **2025**, *14*, 1624. [CrossRef]
32. Ultralytics. Detection Datasets. Available online: https://docs.ultralytics.com/datasets/detect/ (accessed on 17 December 2025).
33. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. GitHub Repository. 2020. Available online: https://github.com/albumentations-team/albumentations (accessed on 3 September 2025).
34. Hendrycks, D.; Dietterich, T. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019. Available online: https://openreview.net/forum?id=HJz6tiCqYm (accessed on 3 September 2025).
35. Lee, J.; Hwang, K.-I. YOLO with Adaptive Frame Control for Real-Time Object Detection Applications. *Multimed. Tools Appl.* **2022**, *81*, 36375–36396. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.