Review

Lightweight Deep Learning Models for Face Mask Detection in Real-Time Edge Environments: A Review and Future Research Directions

Saim Rasheed *

Posted Date: 7 January 2026

doi: 10.20944/preprints202601.0505.v1

Keywords: face mask detection; lightweight deep learning architectures; hybrid deep learning models; edge computing; convolutional neural networks; YOLO-based detectors

 doi:10.20944/preprints202601.0505.v1

Review

Lightweight Deep Learning Models for Face Mask Detection in Real-Time Edge Environments: A Review and Future Research Directions

Saim Rasheed

Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University; srahmed@kau.edu.sa

Abstract

Automated face mask detection remains an important component of hygiene compliance, occupational safety, and public health monitoring, even in post-pandemic environments where real-time, non-intrusive surveillance is required. Traditional deep learning models offer strong recognition performance but are often impractical for deployment on embedded and edge devices due to their computational complexity. Recent research has therefore emphasized lightweight and hybrid architectures that maintain high detection accuracy while reducing model size, inference latency, and energy consumption. This review provides an architecture-centered examination of face mask detection systems, analyzing conventional convolutional models, lightweight convolutional networks such as the MobileNet family, and hybrid frameworks that integrate efficient backbones with optimized detection heads. A comparative performance analysis highlights key trade-offs between accuracy and computational efficiency, emphasizing the constraints of real-world and edge-oriented deployments. Open challenges, including improper mask detection, domain adaptation, model compression, and extending detection systems toward broader compliance-monitoring applications, are discussed to outline a forward-looking research agenda. This work consolidates current understanding of architectural strategies for mask detection and offers guidance for developing scalable, robust, and real-time deep learning solutions suitable for embedded and mobile platforms.

Keywords: face mask detection; lightweight deep learning architectures; hybrid deep learning models; edge computing; convolutional neural networks; YOLO-based detectors

1. Introduction

The widespread proliferation of airborne infectious diseases such as COVID-19 highlighted the importance of face coverings as an effective measure for reducing person-to-person transmission in both clinical and community environments [1]. However, the relevance of automated mask detection extends well beyond pandemic response. In many occupational settings, including healthcare, pharmaceutical laboratories, hospitality services, manufacturing plants, and food-handling industries, mask-wearing remains a mandatory hygiene and safety requirement [2]. In such environments, inconsistent or improper mask usage can compromise safety, reduce operational compliance, and increase the risk of contamination. Automated, computer-vision-based monitoring systems therefore provide meaningful value by detecting mask-wearing violations and triggering real-time alerts to reduce reliance on manual supervision [2,3].

Deep learning-based face mask detection has consequently emerged as a scalable, non-intrusive solution for compliance monitoring across public and professional domains [2,3]. As artificial intelligence (AI) increasingly shifts toward on-device and edge-computing environments, research emphasis has expanded from merely achieving high accuracy to achieving high accuracy with real-time efficiency under hardware limitations [4,5]. This shift reframes the central research question

 doi:10.20944/preprints202601.0505.v1

from “How do we detect masks accurately?” to “How do we detect masks accurately and efficiently enough for deployment on constrained edge platforms?”

Traditional CNN architectures such as VGGNet and ResNet deliver strong feature-learning capabilities but are computationally intensive and often unsuitable for deployment on low-power devices. Lightweight architectures, including the MobileNet family and other efficient convolutional models, address these challenges by reducing parameter count, lowering latency, and enabling deployment on embedded systems [3]. More recently, hybrid architectures that integrate lightweight feature extractors with optimized one-stage detectors (e.g., SSD-, YOLO-, or attention-enhanced variants) have demonstrated promising improvements in balancing accuracy, speed, and resource usage [5,6]. These advancements suggest that robust and efficient mask detection is feasible even within strict real-time constraints.

Problem Statement and Rationale: Despite considerable progress, practical deployment of mask detection systems continues to face two major challenges:
(1) maintaining high recognition reliability under real-world variability such as occlusion, lighting changes, and diverse mask types, and
(2) operating efficiently on hardware-limited platforms where computational resources, memory, and energy budgets are restricted [4,5].

Lightweight and hybrid architectures offer promising solutions, yet the rapid diversification of efficient model designs has created uncertainty regarding which architectural choices best balance accuracy, inference speed, and deployment feasibility. To address this gap, this review synthesizes contemporary architectural approaches for mask detection, compares their performance characteristics, and identifies research opportunities to guide future development [6–9].

Existing surveys and review efforts remain limited in several respects. Many emphasize pandemic-driven solutions without considering broader compliance applications beyond COVID-19 [7–9]. Others focus heavily on accuracy while giving insufficient attention to practical factors such as latency, model size, generalization capacity, or the constraints of embedded deployment [4,5,8]. Issues related to improper mask detection, domain shift, and long-term sustainability remain largely unresolved, and emerging methods for compression, distillation, and hardware-aware optimization are underexplored in current literature. This review aims to address these gaps by providing an architecture-centric analysis followed by a performance comparison and a forward-looking discussion of open challenges.

This review aims to:

a) Examine conventional, lightweight, and hybrid deep learning architectures used for face mask detection;
b) Compare their performance with respect to accuracy, inference efficiency, and deployment suitability;
c) Analyze the core challenges affecting real-world deployment, including improper mask detection, domain shift, and computational constraints;
d) Outline future research directions focused on model compression, knowledge distillation, domain adaptation, and expanding applications beyond mask detection.

The remainder of this paper is organized as follows. Section 2 presents the methodology adopted for conducting the review. Section 3 analyzes the major architectural families employed in face mask detection systems. Section 4 provides a comparative evaluation of model performance, focusing on accuracy and efficiency considerations. Section 5 discusses the open challenges and outlines potential directions for future research. Finally, Section 6 concludes the paper by summarizing the key findings and contributions.

2. Methodology

The methodology of this review was designed to rigorously analyze lightweight and hybrid deep learning approaches for face mask detection, with particular emphasis on their feasibility in

 doi:10.20944/preprints202601.0505.v1

real-time and resource-constrained deployment environments. The assessment focused not only on reported performance metrics such as accuracy and inference speed, but also on architectural efficiency, evaluation settings, hardware compatibility, and scalability. Rather than following a rigid systematic protocol, the selection and evaluation process were guided by conceptual relevance and alignment with the research objectives defined earlier. Each included study was examined through a multi-dimensional analytical framework considering the problem addressed, the proposed architecture, optimization strategies, dataset characteristics, experimental setup, reported strengths and limitations, and potential research gaps.

2.1. Literature Search Strategy

The literature search was conducted using peer-reviewed digital libraries including ScienceDirect, IEEE Xplore, SpringerLink, ACM Digital Library, and MDPI. Publications from 2020 to 2025 were considered to capture developments aligned with post-pandemic transitions and emerging edge-based AI deployment trends. Search terms included combinations of "face mask detection," "lightweight deep learning," "real-time inference," "edge computing," "MobileNet," "YOLO," "improper mask detection," "embedded deployment," and "deep learning optimization." Only journal articles and reputable conference proceedings indexed in SCI, SCIE, or Scopus were shortlisted. Reference tracing was additionally employed to capture studies with architectural innovation or deployment-oriented experimentation.

2.2. Inclusion and Exclusion Criteria

Studies were included if they presented a deep learning–based method for face mask detection (binary or multi-class) and provided quantitative evaluation using publicly available or well-described datasets. Priority was given to research addressing real-time inference, constrained-resource deployment, architectural efficiency, or model-level optimization. Studies without empirical experimentation, those focused exclusively on cloud-based inference without discussion of deployment feasibility, and works based solely on traditional image processing were excluded. Non-peer-reviewed preprints and duplicates were also omitted to ensure reliability.

2.3 Screening and Selection Approach

Screening was conducted in multiple stages. Initial filtering based on titles and abstracts ensured alignment with lightweight architectures and deployment-aware mask detection. Full-text screening was then used to assess architectural contributions, optimization strategies, dataset suitability for the stated task, and real-time implementation potential. Articles that described mask detection without addressing computational or deployment constraints were retained only as secondary background sources rather than core analytical studies.

2.4. Data Extraction and Categorization

For each selected study, key information was extracted to support thematic and architectural analysis. This included the research problem, architectural design, enhancement strategies, dataset and evaluation conditions, training setup, performance metrics, model complexity (e.g., parameter count), inference performance (e.g., FPS), and any deployment-specific details such as hardware specifications or optimization pipelines. Studies were categorized into conventional CNNs, lightweight convolutional models, and hybrid architectures. This structured categorization enabled comparative analysis of architectural trends, efficiency trade-offs, and practical deployment implications discussed in later sections. To ensure transparency and consistency in the comparative review, the essential characteristics of each included study were extracted and organized into a structured summary. This overview captures the goals, methodological designs, experimental settings, and principal findings of the selected works. **Table 1** presents these extracted details and

 doi:10.20944/preprints202601.0505.v1

provides a foundational context for the architectural and performance analyses discussed in subsequent sections.

Table 1. Overview of Included Studies Based on Extracted Goals, Methods, Experimental Settings, and Results.

Ref	Experiment	Goal	Materials	Methods	Results	Conclusion
[10]	Object detection	To develop a Raspberry Pi 4–based SSDLite MobileNetV3 Small device capable of detecting correct and incorrect cloth masks, correct and incorrect medical masks, and cases where no mask is worn or the face is obscured.	Raspberry Pi 4 Model B 4Gb, Raspberry Pi 4 Model B Cam V.1, monitor, push button non-momentary switch, fan, diode 1N4001, 3 resistor 470 Ohm, transistor 2n2222	1. Trained SSDLite MobilenetV3 Small model with fine-tuning and without fine-tuning. 2. Compared the detection performance of SSDLite MobilenetV3 Small with other models like SSDLite MobilenetV3 Large, SSDLite MobilenetV2, SSD MobilenetV2, SSDLite Mobiledets, and SSDMNV2. 3. Evaluated the detection, FPS, and power consumption of the models.	The SSDLite MobilenetV3 Small model with fine-tuning had the highest FPS compared to other models, but could not detect the incorrect use of masks accurately. The overall accuracy of the SSDLite MobilenetV3 Small model was 70%.	The SSDLite MobilenetV3 Small model offers faster detection than others but is less effective than SSDLite MobilenetV2 in identifying incorrect mask usage. The tool also faces limitations.
	Object detection model comparison	To compare the performance of different object detection models including SSDLite MobilenetV3 Small, SSDLite MobilenetV3 Large, SSDLite MobilenetV2, SSD MobilenetV2, SSDLite Mobiledets, and SSDMNV2 for face mask detection on Raspberry Pi 4.	Raspberry Pi 4 Model B 4Gb, Raspberry Pi 4 Model B Cam V.1, dataset of face images with and without masks	1. Trained the different object detection models on the face mask dataset. 2. Evaluated the detection accuracy, FPS, and power consumption of the models on the Raspberry Pi 4.	The SSDLite MobilenetV2 model with fine-tuning had the best detection performance, able to detect all the test cases correctly. The SSDLite MobilenetV3 Small model had the highest FPS but struggled to detect incorrect mask usage.	The SSDLite MobilenetV2 model is the most suitable for face mask detection on Raspberry Pi 4 among the models tested, providing good accuracy and detection speed.
[11]	Empirical study	To systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance.	Convolutional Neural Networks (ConvNets)	Systematically studied scaling up ConvNets by adjusting network depth, width, and resolution.	Scaling up any dimension of network width, depth, or resolution improves accuracy, but the accuracy gain diminishes for bigger models. It is critical to balance all dimensions of	Carefully balancing network width, depth, and resolution is an important but missing piece, preventing us from better accuracy and efficiency.

					network width, depth, and resolution during ConvNet scaling.	
	Methodology development	To propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient.	Convolutional Neural Networks (ConvNets)	Proposed a compound scaling method that uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.	The proposed compound scaling method can achieve better accuracy and efficiency compared to conventional single-dimension scaling methods.	The compound scaling method enables scaling up a baseline ConvNet to any target resource constraints in a more principled way, while maintaining model efficiency.
	Neural architecture search and model scaling	To use neural architecture search to design a new baseline network and scale it up to obtain a family of models, called EfficientNets, which achieve much better accuracy and efficiency than previous ConvNets.	Convolutional Neural Networks (ConvNets)	Used neural architecture search to develop a new baseline network called EfficientNet-B0, and then applied the proposed compound scaling method to scale it up and obtain a family of EfficientNet models.	The scaled EfficientNet models significantly outperform other ConvNets in terms of accuracy and efficiency. EfficientNet-B7 achieves state-of-the-art 84.4% top-1 accuracy on ImageNet, while being 8.4x smaller and 6.1x faster on inference than the best existing ConvNet.	The EfficientNet models, developed using the proposed compound scaling method, achieve much better accuracy and efficiency than previous ConvNets.
[12]	Machine learning algorithm through image classification using MobileNetV2	To develop a face mask detection model that can be used by authorities to make mitigation, evaluation, prevention, and action planning against COVID-19.	1,916 images of people wearing masks, 1,930 images of people not wearing masks, image size of 224x224 pixels	1) Collect data, 2) Pre-process data (resize images, convert to array, pre-process using MobileNetV2, perform hot encoding on labels), 3) Split data into 75% training and 25% testing, 4) Construct training image generator for augmentation, build base model with MobileNetV2, add model parameters, compile model, train model, save model, 5) Test model on testing set and evaluate performance metrics (precision, recall, F1-score, accuracy)	The built model can detect people wearing and not wearing face masks with an accuracy of 96.85%.	The face mask detection model developed in this study can be used by authorities to monitor and evaluate the implementation of face mask wearing policies, and help with mitigation, prevention, and action planning against COVID-19.

	Application of the face mask detection model to real-world data	To apply the developed face mask detection model to images from 25 cities in Indonesia and analyze the percentage of people wearing face masks in each city.	Images from various sources (public place CCTV, shops, traffic cameras) in 25 cities in Indonesia, selected based on data availability	Apply the trained face mask detection model to the images from the 25 cities, calculate the percentage of people wearing and not wearing face masks in each city.	The percentage of people not wearing face masks ranged from 64.14% (Surabaya) to 82.76% (Jambi).	Face mask usage differs across cities, with some showing notably lower compliance. This helps authorities target interventions and allocate resources to areas with the weakest mask-wearing.
	Correlation analysis	To evaluate the validity of the face mask wearing percentage data by correlating it with the COVID-19 vigilance index.	Percentage of people wearing face masks in the 25 cities, COVID-19 vigilance index data	Conduct a bivariate correlation analysis between the percentage of people wearing face masks in the cities and the COVID-19 vigilance index.	The percentage of people wearing face masks and the COVID-19 vigilance index have a strong, negative, and significant correlation of -0.62.	The model's mask-wearing data aligns with the COVID-19 vigilance index, showing that cities with lower mask-wearing rates require higher vigilance against transmission.
[3]	Face mask detection	To propose a novel face mask detection framework FMD-Yolo to monitor whether people wear masks in a right way in public, which is an effective way to block the virus transmission.	Im-Res2Net-101 feature extractor, enhanced path aggregation network En-PAN, localization loss, Matrix NMS method	The feature extractor employs Im-Res2Net-101 which combines Res2Net module and deep residual network, where utilization of hierarchical convolutional structure, deformable convolution and non-local mechanisms enables thorough information extraction from the input. An enhanced path aggregation network En-PAN is applied for feature fusion, where high-level semantic information and low-level details are sufficiently merged. Localization loss is designed and adopted	The proposed FMD-Yolo has achieved the best precision AP50 of 92.0% and 88.4% on the two datasets, and AP75 at IoU=0.75 has improved 5.5% and 3.9% respectively compared with the second one.	The results demonstrate the superiority of FMD-Yolo in face mask detection with both theoretical values and practical significance.

				in model training phase, and Matrix NMS method is used in the inference stage.		
[13]	Algorithm development	To propose a novel object detector, lightweight FMD through You Only Look Once (LFMD-YOLO), which can achieve an excellent balance of precision and speed.	C3E(CSP Bottleneck with 3 convolutions-ECA) module, MECAPF(max-pooling ECA pyramid-fast) module, new backbone network combining C3E and MECAPF modules, weighted bidirectional feature pyramid network based on the C3E module (E-BiFPN), detection heads, intersection over union (IoU)	1. Designed the C3E(CSP Bottleneck with 3 convolutions-ECA) module and MECAPF(max-pooling ECA pyramid-fast) module based on the effective attention mechanism (ECA) to enrich channel information. 2. Proposed a new backbone network combining C3E and MECAPF modules. 3. Designed the weighted bidirectional feature pyramid network based on the C3E module (E-BiFPN) as the feature fusion neck, making full use of multi-scale features to mine more local information and enhancing the representation of small objects of face masks. 4. Further enhanced the model performance by adding detection heads and improving intersection over union (IoU).	The proposed LFMD-YOLO achieves higher detection accuracy with mAPs of 68.7% and 60.1%, respectively, while having lower parameters and GFLOPs.	The proposed LFMD-YOLO can achieve an excellent balance of precision and speed for lightweight face mask detection.
[14]	Deep learning-based face mask detection	To develop a deep learning-based system for real-time face mask detection to enhance public health monitoring in environments where mask compliance is critical.	Convolutional Neural Network (CNN) built with TensorFlow and Keras, diverse input images, Google Colab, Google Drive.	Utilize a CNN model to effectively classify individuals as mask-wearing or non-mask-wearing. Apply data preprocessing and augmentation techniques to improve model robustness and generalizability. Leverage cloud-based resources for efficient model training and deployment.	The system achieved high training and validation accuracy, consistent loss reduction, and strong real-time detection. It remained reliable despite minor validation fluctuations, demonstrating resilience and suitability for varied environments.	The DL–based system detects mask usage in real time. Data augmentation improves generalization, allowing reliable performance across varied scenarios and image conditions.

[15]	Face mask detection system development	To develop a rapid real-time face mask detection system (RRFMDS) for effective COVID-19 monitoring	Single-shot multi-box detector based on ResNet-10, fine-tuned MobileNetV2, custom dataset of 14,535 images with 5000 incorrect masks, 4789 with masks, and 4746 without masks	Used single-shot multi-box detector for face detection and fine-tuned MobileNetV2 for face mask classification. Trained the system on the custom dataset.	The system can detect all three classes (incorrect masks, with mask and without mask faces) with an average accuracy of 99.15% and 97.81% on training and testing data respectively. The system takes on average 0.14201142 s to process a single frame.	The proposed RRFMDS system is a lightweight and efficient approach for real-time face mask detection from video data. It outperforms existing state-of-the-art models in terms of accuracy and processing speed.

 doi:10.20944/preprints202601.0505.v1

The studies summarized in **Table 1** reveal significant variation in architectural choices, dataset configurations, evaluation procedures, and deployment assumptions. This diversity underscores the need for an architecture-centric viewpoint when interpreting performance results, as models differ not only in structural complexity but also in experimental conditions and optimization strategies. The extracted patterns also highlight gaps, such as inconsistent reporting of inference metrics, limited multi-class improper mask annotations, and scarce evaluation under real-world deployment constraints that motivate the architectural comparison and performance analysis presented in upcoming sections. These observations further justify the focus of this review on understanding how architectural families influence efficiency, robustness, and readiness for edge deployment.

3. Architectural Landscape of Face Mask Detection Models

Deep learning–based face mask detection systems rely fundamentally on the architectural design of their underlying neural networks. Architectural choices determine not only a model's feature-extraction capacity and recognition accuracy but also its computational footprint, memory usage, inference speed, and overall suitability for deployment on resource-constrained edge environments. As mask detection has transitioned from high-performance computing systems toward embedded and real-time monitoring platforms, the efficiency and scalability of these architectures have become just as important as their classification accuracy.

This section provides an architecture-oriented review of face mask detection models, examining how different architectural families address the competing demands of accuracy and efficiency. The discussion progresses from conventional convolutional neural networks, which offer strong representational power but high computational cost, to lightweight convolutional models specifically optimized for mobile and embedded inference. The section then evaluates hybrid architectures that integrate lightweight backbones with streamlined detection heads or specialized optimization strategies to achieve real-time performance without substantial loss of accuracy. Together, these architectural categories reflect the evolution of face mask detection from computationally expensive methods toward edge-ready solutions suitable for widespread deployment.

3.1. Conventional CNN-Based Approaches

Early work on automated face mask detection relied heavily on conventional convolutional neural network (CNN) architectures, which formed the foundation of modern deep learning in computer vision. These models, originally developed for large-scale image classification and object detection tasks, were adopted due to their strong representational capacity and readily available pre-trained weights. Architectures such as **AlexNet**, **VGGNet**, **Inception**, **ResNet**, **DenseNet**, **Xception**, and the **R-CNN family** provided the initial backbone for mask detection pipelines during the early phase of the COVID-19 pandemic.

The evolution of conventional CNNs reflects a gradual improvement in depth, efficiency, and stability. **LeNet-5** introduced the first successful convolutional network structure for digit recognition, demonstrating the feasibility of learned hierarchical features [16]. **AlexNet** catalyzed the deep learning revolution by winning the ImageNet 2012 challenge, leveraging ReLU activations and GPU training to drastically outperform traditional hand-crafted descriptors [17]. **VGG16/VGG19** refined network depth with small, uniform 3×3 convolutions, producing strong yet computationally heavy models [18]. The **Inception family** introduced multi-branch processing to improve efficiency while preserving expressive power [19], while **Inception-ResNet** blended residual learning with inception modules to enable deeper and more stable optimization [20].

Several major CNN families subsequently introduced important structural innovations. **ResNet**, with its identity skip connections, addressed the vanishing-gradient problem and enabled successful training of networks exceeding 100 layers [21]. **DenseNet** extended this concept by connecting each layer to all subsequent layers, improving feature reuse and reducing parameter growth [22]. **Xception** reorganized convolutions into fully depthwise separable form, a precursor to later lightweight

 doi:10.20944/preprints202601.0505.v1

models such as **MobileNet** [23]. More recent backbone families such as **RegNet** attempted to design regular, scalable architectures via design-space exploration, though these were not widely adopted for mask detection because of computational demands [24].

Beyond classification backbones, conventional object detection frameworks also played a significant role. The **R-CNN family** including **R-CNN, Fast R-CNN**, and **Faster R-CNN**, extended CNNs to region-based object detection using two-stage processing [25–27]. **Mask R-CNN** further introduced an instance segmentation branch, enabling pixel-level analysis of facial regions or mask boundaries [28]. These models were frequently adopted in early COVID-19 surveillance systems, especially in industrial or controlled environments requiring both face detection and classification.

In the context of face-mask detection, conventional CNN backbones were widely adopted in early studies implementing either face-classification pipelines or end-to-end object-detection frameworks. A typical classification workflow included: (i) frame acquisition from CCTV or mobile cameras, (ii) pre-processing such as resizing and normalization, (iii) face detection using Haar cascades, HOG-based detectors, or CNN-based region detectors, and (iv) classification of each cropped face using a CNN such as VGG16, ResNet50, DenseNet121 or InceptionV3. Also, conventional CNNs and detection frameworks were employed early on to evaluate the feasibility of automated mask compliance. For instance, **MaskedFace-Net**, a large publicly available dataset containing over 137,000 images with correct and incorrect mask wearing annotations, was used to train and test classifiers based on VGG, ResNet, or DenseNet backbones; results demonstrated that these models could reliably distinguish between masked and unmasked faces under controlled image conditions [29].

Conventional CNN backbones have been employed in both face-classification and object-detection pipelines for mask detection. For instance, the **MaskedFace-Net** dataset, containing over 137,000 images with masks correctly or incorrectly worn, has served to train classification models using architectures such as VGG, ResNet or DenseNet, showing that conventional CNNs can reliably distinguish between masked and unmasked faces under controlled conditions [29]. On the detection side, Jiang et al. proposed **SE-YOLOv3**, a YOLOv3-based detector enhanced with a Squeeze-and-Excitation module, trained on the **PWMFD** dataset of 9,205 images. Their results indicated high detection accuracy along with real-time inference capability, demonstrating that conventional CNN backbones remain effective when combined with modern detection heads [30].

However, performance of conventional CNN-based detectors degrades under real-world variability. A recent survey on masked-face recognition and detection notes that occlusion (e.g., by hands or non-mask objects), non-frontal poses, diverse mask designs, and inconsistent lighting conditions significantly reduce the reliability of classification and detection models based on conventional CNNs [31]. Even large annotated datasets like MaskedFace-Net, though extensive but often contain mostly frontal or simply synthesized mask images, limiting their representativeness for unconstrained environments [29].

Overall, while conventional CNN architectures continue to provide a valuable baseline for face-mask detection tasks, their robustness and generalization remain limited in challenging, real-world scenarios. The observed deficiencies under occlusion, pose variation, and diverse mask usage conditions motivate the shift toward lightweight CNNs, optimized detection heads, and hybrid architectures, which strive to balance accuracy with computational efficiency and deployment feasibility in edge or surveillance environments.

A comparative overview of major conventional CNN architectures, highlighting their innovations, parameter scales and deployment implications, is presented in **Table 2**.

 doi:10.20944/preprints202601.0505.v1

Table 2. Comparative Overview of Major Conventional CNN Architectures.

Architecture	Year	Key Innovation	Parameter Count	Strengths	Limitations	Original Source
LeNet-5	1998	Early CNN architecture (conv + pooling)	~60K	Simple, stable	Too shallow for modern tasks	[16]
AlexNet	2012	ReLU, dropout, GPU training	~60M	Started modern deep learning	Heavy; not edge-friendly	[17]
VGG16/VGG19	2014	Deep stacks of 3×3 conv layers	~138M	Strong features	Extremely large & slow	[18]
Inception-v1	2015	Multi-branch convolutions	~6.8M	Efficient, flexible	Complex structure	[19]
Inception-ResNet	2017	Residual + inception blocks	23–55M	Very accurate	Heavy	[20]
ResNet (18–101)	2016	Skip connections	11–44M	Deep & stable	Still heavy for edge	[21]
DenseNet121	2017	Dense connectivity	~8M	High feature reuse	Slow inference	[22]
Xception	2017	Depthwise separable conv	~22M	Good efficiency	Not lightweight enough	[23]
Faster R-CNN	2015	Two-stage region detector	Backbone-dependent	Accurate	Slow without GPU	[27]
Mask R-CNN	2017	Adds segmentation branch	Backbone-dependent	Detects improper masks	Heavy for edge	[28]
RegNet	2020	Regular network design space	10–50M	Strong accuracy	Rarely used in mask detection	[24] [Radosavovic2020]

Despite their strengths, these networks exhibit important limitations for real-time deployment and motivated the rise of lightweight CNNs (e.g., MobileNet, ShuffleNet) and hybrid architectures designed for edge-optimized deployment while retaining adequate accuracy.

 doi:10.20944/preprints202601.0505.v1

3.2. Lightweight Convolutional Models

Lightweight convolutional neural networks are designed to provide competitive recognition accuracy while significantly reducing memory footprint, parameter count and computational cost. Instead of relying on wide and deep stacks of standard convolutions, these models adopt mechanisms such as depth-wise separable convolutions, inverted residual bottlenecks, grouped pointwise convolutions with channel shuffling, or squeeze–expand fire modules to minimize the number of multiplications and parameters. As a result, they are well suited for deployment on mobile and embedded platforms, including edge devices used in camera-based mask monitoring systems.

Among these architectures, the **MobileNet family** is arguably the most influential. The original MobileNet formulation introduced a streamlined architecture based on depth-wise separable convolutions and global width and resolution multipliers, enabling a tunable trade-off between latency and accuracy for mobile vision tasks [32]. MobileNetV2 extended this idea by incorporating inverted residual blocks with linear bottlenecks, improving accuracy across multiple benchmarks while preserving efficiency [33]. Later variants such as MobileNetV3 [34] further refined block design and activation functions, but MobileNetV2 remains the most widely adopted backbone in the face mask detection literature due to its balance between representational power and computational cost.

Other lightweight architectures follow different optimization strategies. **EfficientNet** introduced compound scaling of depth, width and input resolution, combined with an architecture discovered via neural architecture search, producing a family of models that deliver state-of-the-art accuracy with substantially fewer parameters than conventional CNNs [11]. **ShuffleNet** instead reduces computational cost by using pointwise group convolutions and channel shuffle, allowing efficient information mixing while maintaining high throughput on ARM-based mobile devices [35]. **SqueezeNet** targets extreme parameter reduction by replacing many standard convolution layers with fire modules (1×1 squeeze layers followed by 1×1 and 3×3 expand layers), achieving AlexNet-level ImageNet accuracy with roughly 50× fewer parameters [36]. These architectures form the algorithmic foundation upon which many recent lightweight mask detection systems are built.

In the specific context of face mask detection, **MobileNetV2** is the most frequently used lightweight backbone. Several studies employ MobileNetV2, either as a feature extractor within an object-detection pipeline or as the main classifier in a transfer-learning setting. For example, one real-time system (SSDMNV2) integrates MobileNetV2 into a single-shot detection framework and reports effective two-class (mask/no-mask) detection using OpenCV, TensorFlow and Keras on live video streams [37]. Other works fine-tune MobileNetV2 on custom mask datasets and then attach classical machine-learning classifiers (such as SVM or decision trees) on top of deep features, demonstrating that MobileNetV2-based embeddings can be reused across a variety of deployment strategies [38]. Additional implementations exploit MobileNetV2 in embedded or IoT-oriented systems to enforce mask-wearing compliance, emphasizing its ability to achieve real-time inference on constrained hardware while maintaining high recognition accuracy [39].

EfficientNet-based models have also been explored for mask detection, typically as stronger yet still relatively compact alternatives to MobileNet. One study employing **EfficientNet-B0** for binary mask classification reports an accuracy of around 99–99.7% on a two-class problem, outperforming several heavier backbones while remaining deployable in practical systems [40]. In [41] the authors employ **EfficientNet-B0** as the feature extractor backbone and a large-margin piecewise-linear (LMPL) classifier on top of deep features. The method achieved 99.53% and 99.64% accuracies for the two tasks respectively, outperforming conventional end-to-end CNN models and classical image classification approaches. The authors argue that their solution achieves both high accuracy and efficiency, making it suitable for real-world deployment in scenarios like biometric authentication or public-health compliance checks. Other works using EfficientNet variants for mask identification similarly highlight their favorable accuracy–efficiency trade-off compared with legacy CNNs, though their computational requirements are generally higher than those of MobileNetV2 on extremely low-power devices [42].

 doi:10.20944/preprints202601.0505.v1

Architectures derived from SqueezeNet provide another lightweight option. SqueezeNet itself was designed as an extremely compact classification network, but face mask detection has inspired specialized derivatives such as **SqueezeMaskNet**, which integrates fire modules with channel-attention mechanisms to support four-way classification (correctly worn mask, mask covering only the mouth, mask not covering, and no mask) while running in real time on edge GPUs [43]. Comparative evaluations indicate that SqueezeMaskNet achieves accuracy in the mid- to high-90% range while sustaining high frame rates on Jetson-class hardware, making it particularly suitable for embedded applications where classical CNNs would be too heavy.

Finally, several hybrid lightweight architectures combine these backbones with efficient detection heads. For instance, MobileNetV2 features can be paired with SSD-style detection layers or YOLO derivatives, and EfficientNet or SqueezeNet variants can be integrated into multi-stage pipelines that first detect faces and then classify mask usage [37]. Survey studies on mask detection also confirm that MobileNetV2-based solutions dominate the lightweight category, with EfficientNet and SqueezeNet-derived models appearing as strong but less commonly used alternatives [8]. Overall, the literature suggests that MobileNetV2 is currently the most popular lightweight model for face mask detection, while EfficientNet and SqueezeMaskNet offer accuracy gains in applications where slightly higher computational budgets are acceptable. A structured comparison of these lightweight architectures is presented in **Table 3** to highlight their respective design strategies, computational efficiency and suitability for real-time face-mask detection

Table 3. Comparative Summary of Lightweight Architectures Applied in Face-Mask Detection Systems.

Model Type	Key Architectural Concept	Approx. Parameters / Complexity	Typical Usage in Mask Detection
MobileNetV2	Depthwise separable convolutions with inverted residual bottlenecks	~3.4M parameters (α=1.0)	Most widely adopted lightweight backbone; real-time mask/no-mask or 3-class classification on embedded devices.
EfficientNet-B0	Compound scaling of depth, width, and resolution	~5.3M parameters	Used in high-accuracy systems (e.g., EfficientMask-Net); suitable for improper mask detection with slightly higher computational needs.
ShuffleNet	Grouped 1×1 convolution with channel shuffle	~2.3M parameters (1.0×)	Limited adoption; tested in low-resource conditions but less consistent than MobileNet.
SqueezeNet / SqueezeMaskNet	Fire module (1×1 squeeze + expand) with attention extensions	~1.2M (SqueezeNet), ~1.5M (SqueezeMaskNet)	Designed for real-time multi-class classification; high FPS on Jetson-class edge hardware.

 doi:10.20944/preprints202601.0505.v1

EfficientMask-Net (2022)	EfficientNet-B0 backbone with large-margin piecewise-linear classifier (LMPL)	~5.3M parameters	Achieves up to 99.6% accuracy; offers detailed detection of improper mask positioning (nose/chin uncovered).
Hybrid CNN–YOLO variants (e.g., MobileNetV2 + YOLO)	Lightweight backbone with optimised detection head	Varies (<8M total)	Used for real-time detection + localisation in surveillance and compliance monitoring; effective for streaming environments.

Lightweight architectures have significantly improved the feasibility of real-time face-mask detection on embedded platforms by minimizing computational overhead without substantially compromising recognition accuracy. Among these models, MobileNetV2 remains the most commonly deployed due to its favorable balance between architectural simplicity and inference efficiency, particularly when used in transfer-learning pipelines. EfficientNet-based implementations, such as EfficientMask-Net, demonstrate superior accuracy, especially for improper mask detection, though their computational demands are marginally higher. SqueezeMaskNet, derived from SqueezeNet with channel-attention enhancements, offers exceptional frame rates on edge GPUs and supports four-class decision schemes, making it suitable for rapid compliance verification. ShuffleNet is rarely preferred due to inconsistent performance, whereas hybrid configurations integrating MobileNet or EfficientNet with YOLO-like detection heads provide strong localization capabilities for surveillance applications. Overall, MobileNetV2 is most widely adopted, while EfficientNet derivatives lead in accuracy-sensitive tasks, and SqueezeMaskNet offers the best real-time responsiveness in constrained environments.

3.3. Hybrid Architectures

Hybrid architecture represents an advanced design philosophy that extends beyond standalone conventional CNNs or lightweight backbones by combining multiple complementary modules, typically a feature extraction network and a detection or classification head that are used to handle both spatial localization and mask classification effectively. Unlike the approaches in Sections 3.1 and 3.2, hybrid architectures integrate networks such as **MobileNetV2, EfficientNet, or SqueezeNet** with detection frameworks like **YOLO, SSD, or Faster R-CNN**, or with classical ML classifiers. These combinations are not random; they arise from deliberate architectural reasoning to meet real-time deployment demands, especially in public health monitoring during COVID-19. Researchers have demonstrated that hybridization improves robustness, speed, and accuracy under real-world constraints, such as occlusion, inconsistent lighting and crowd density (e.g., YOLOv2+ResNet50 [44], MobileNetV2+SSD [45], YOLOv5 with attention mechanisms [46], or CNN+SVM hybrids [47]).

Lightweight hybrid architectures are crucial for face mask detection, especially in the context of the COVID-19 pandemic, where real-time monitoring in public spaces is essential for public health compliance. These architectures enable deployment on edge devices, which are often resource-constrained, by reducing computational costs and memory requirements while maintaining high detection accuracy. The integration of lightweight models such as **MobileNetV2**, and the use of heavier feature extractors like **VGG19** within hybrid pipelines, allows for efficient feature extraction and classification, making them suitable for real-time applications in environments with limited hardware capabilities [48–52].

 doi:10.20944/preprints202601.0505.v1

This motivation aligns with findings from multiple hybrid studies, where integrating lightweight backbones with advanced detection heads significantly improved end-to-end performance for mask localization and classification in constrained settings (e.g., YOLOv3-based hybrids [53], ensemble hybrid detectors [2], and smart-city hybrid systems[54]).

The design of lightweight hybrid architectures typically involves combining the strengths of different model types, such as convolutional neural networks (CNNs) and transformer models, to optimize both efficiency and accuracy. Techniques like depth-wise separable convolutions, inverted residual structures, and parallel hybrid architectures are commonly employed to minimize the number of parameters and computational operations. These principles ensure that models can operate effectively on mobile and edge devices without sacrificing performance [48,49,51,55].

Such design principles are evident in real hybrid implementations. For example, **YOLOv5** combined with coordinate attention blocks improves detection precision while maintaining fast inference[46], and **CNN+ML hybrids** (deep features + SVM) exploit efficiency while preserving discriminative power [47]. These works demonstrate that hybridization follows structured engineering choices rather than arbitrary pairing of networks.

Developers of lightweight hybrid architecture must balance the trade-off between computational efficiency and detection accuracy. While lightweight models are optimized for speed and resource usage, they may experience accuracy degradation under challenging conditions such as illumination changes or occlusions. Hybrid models attempt to mitigate these issues by leveraging complementary strengths of different architectures, but some loss in robustness may still occur in extreme scenarios [56–58].

This trade-off is well-documented in comparison studies, where hybrid detectors show better occlusion-robustness than pure classifiers. For instance, **YOLOv2-ResNet50** hybrids handle masked and partially occluded faces more effectively than standalone CNNs [44], and multi-detector ensembles enhance detection under crowd conditions [2]. Nevertheless, no hybrid fully eliminates robustness issues, especially under severe occlusions or rapid motion.

To further enhance the performance of lightweight hybrid models, various optimization techniques are applied. These include model compression, quantization, pruning, and knowledge distillation, all aimed at reducing model size and computational demands. Additionally, architectural innovations such as ghost modules and Bi-FPN (Bidirectional Feature Pyramid Networks) are explored to improve feature extraction and detection capabilities without increasing resource consumption [51,57].

These optimization strategies also appear in hybrid architectures that rely on TensorRT acceleration, auto-labelling modules, and attention-based feature refinement, as seen in improved YOLOv5 hybrids [46]. Similarly, system-level hybrids incorporate IoT and edge intelligence to compensate for on-device constraints [54], demonstrating that optimization extends beyond the neural architecture itself.

Several case studies demonstrate the successful deployment of lightweight hybrid architectures for face mask detection. For example, systems combining MobileNetV2 and VGG19 have been implemented for real-time surveillance, achieving high accuracy and robustness in drone-based monitoring and public space compliance checks. Other models, such as SSD with MobileNetV2, have been used for automated face mask compliance monitoring, highlighting the flexibility and effectiveness of these architectures in diverse real-world scenarios [50–52,59].

Hybrid detectors based on **MobileNetV2 + SSD** [45], **YOLOv3 variants** [53], and multi-component ensembles [2] consistently outperform standalone models in deployment scenarios requiring person-level localization, mask-wearing classification and real-time operation. Additionally, smart-city systems integrating detection models with sensor networks, edge-computing nodes and cloud-based analytics highlight the broader potential of hybrid architectures in scalable public-health monitoring [54]. These examples reinforce that hybrid architecture serves as a bridge between computational feasibility and large-scale real-world applicability**. Table 4**

 doi:10.20944/preprints202601.0505.v1

summarizes the main hybrid architecture employed in face-mask detection, highlighting their backbones, detection heads and reported strengths.

Table 4. Comparative Overview of Hybrid Architectures Used in Face-Mask Detection.

Hybrid Architecture	Backbone Type	Detection/Classification Head	Key Idea	Reported Strengths	Representative Study
YOLOv2–ResNet50	ResNet50 (heavy backbone)	YOLOv2 one-stage detector	Combine high-level semantic features with fast one-stage detection	High accuracy in medical mask detection; good robustness	Loey et al. (2021) [44]
YOLOv5 + Coordinate Attention	YOLOv5 backbone	Attention-enhanced detection head	Spatial refinement + auto-labelling	Strong mAP improvement; suitable for embedded devices	Pham et al. (2023) [46]
MobileNetV2 + SSD	MobileNetV2 (lightweight)	SSD one-stage detector	Lightweight backbone with efficient localizations	Real-time mask detection on edge devices	Balaji & Gowri (2021) [45]
CNN Feature Extractor + SVM/ML Classifier	VGG19, ResNet, MobileNet	SVM / KNN / RF classifiers	Deep features + classical ML	Good performance on small datasets; simpler deployment	Loey et al. (2021) [47]
YOLOv3-Based Hybrid Detector	CSPDarknet-style backbone	YOLOv3 detection head	Full detector tailored to mask usage	Real-time performance with strong localization	Jiang et al. (2021)[30]
Smart-City System-Level Hybrid	CNN/YOLO backbone	IoT + Edge-tier inference pipeline	Combines DL, transfer learning, and IoT	Scalable deployment across large environments	Himeur et al. (2023) [54]

 doi:10.20944/preprints202601.0505.v1

In summary, hybrid architecture provides a principled balance between speed, accuracy and deployment feasibility, making them the dominant and most practical design strategy for real-time face-mask detection in embedded and resource-constrained environments.

4. Comparative Performance Analysis

The architectures discussed in Section 3, conventional CNNs (e.g., VGG, ResNet, DenseNet), lightweight models (e.g., MobileNetV2-based designs), and hybrid/attention-enhanced YOLO variants, are typically evaluated along two axes: recognition quality (accuracy-oriented metrics) and operational efficiency (latency, throughput, and resource usage). This section synthesizes reported results from recent face-mask detection studies to compare these families more explicitly.

4.1. Evaluation Metrics

Performance evaluation of face-mask detection systems relies on well-established metrics from classification and object-detection literature. These metrics quantify correctness, robustness, and discriminative capability of the models under various conditions (binary vs. multi-class classification, detection vs. recognition, occlusion, class imbalance, etc.). The primary metrics include Accuracy, Precision, Recall (Sensitivity) and F1-Score.

Let the outcomes for a mask-classification or detection system be described using the standard confusion-matrix counts:

- TP (True Positive): correctly predicted positive instances
- FP (False Positive): incorrect positive predictions
- FN (False Negative): missed positive instances
- TN (True Negative): correctly predicted negative instances

Where *positive* may refer to "mask," "no-mask," or "improper mask," depending on the class definition.

4.1.1. Accuracy

Accuracy measures the proportion of correctly predicted instances among all predictions.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \tag{1}$$

It is widely used in early face-mask classification (e.g., mask vs. no-mask), especially on curated datasets such as MaskedFace-Net. However, accuracy can be misleading in imbalanced datasets where one class dominates (e.g., many more masked than improperly masked samples). Therefore, accuracy should always be interpreted along with precision and recall.

4.1.2. Precision

Precision indicates the proportion of correct positive predictions out of all predicted positives.

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{2}$$

High precision means the model makes few false alarms (e.g., rarely labels someone as "no mask" incorrectly). For multi-class mask compliance (proper / improper / no mask), precision is computed per class, then averaged macro- or weighted-wise.

4.1.3. Recall (Sensitivity)

Recall represents the proportion of actual positive instances that are correctly detected.

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{3}$$

High recall means the model makes few misses, which is important for detecting *improperly worn masks* or *no-mask cases* in high-risk zones.

 doi:10.20944/preprints202601.0505.v1

For mask detection systems deployed in public spaces, recall is often considered more critical than precision, since missing non-compliant subjects can compromise public safety.

4.1.4. F1-Score

The F1-score is the harmonic mean of precision and recall:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{4}$$

It provides a balanced measure, particularly when the dataset is imbalanced. F1-score is widely reported in both conventional CNN mask classifiers and lightweight/hybrid models such as SSD-MobileNetV2, SE-YOLOv3, and YOLOv4-tiny derivatives.

Several deep-learning–based face mask detection studies [60–63] have reported accuracy, precision, recall, and F1-score as their primary evaluation metrics, typically derived from confusion matrices built over test sets or cross-validation folds. For example, Hussain et al. developed a MobileNetV2-based transfer learning system for binary mask classification and evaluated it on two datasets (one in-house, one public). They compared a generic DCNN with MobileNetV2 and reported per-model accuracy, precision, recall, and F1-score, showing that MobileNetV2 achieved up to 99% accuracy with closely matched precision, recall, and F1, thereby justifying model selection based on balanced performance across all four metrics rather than accuracy alone [60]. Similarly, Umer et al. proposed a custom CNN with a four-stage image-processing pipeline for face mask detection and evaluated it on their real-image RILFD dataset and two public datasets (MAFA and MOXA), comparing their model against YOLOv3 and Faster R-CNN; they reported detailed precision, recall, accuracy, and F1-scores across datasets, illustrating how these metrics jointly reveal robustness under real-world variations in lighting, mask type, and occlusion.

Ensemble and hybrid classification approaches also rely heavily on these four measures to demonstrate robustness under different data distributions. Bania et al. proposed an ensemble of ResNet50, Inception-v3, and VGG-16 for real-time mask detection and reported class-wise precision, recall, and F1, as well as macro-averaged scores; their best configuration achieved F1-scores around 0.997, indicating that both false positives and false negatives were extremely rare on their test sets [64]. Similarly, Habeeb et al. focused on incorrect facemask-wearing detection (e.g., mask under nose or chin) and explicitly emphasized that high recall was required to avoid missing non-compliant cases, reporting accuracy of 99.4%, precision of 99.4%, recall of 98.6%, and F1-score of 99.0% on their multi-class dataset [65]. These works highlight that *precision and recall are often more informative than accuracy* when the cost of missing violators is high.

Several architectures that jointly address face-mask detection and masked face recognition use the same four metrics to evaluate both tasks. Ullah et al. introduced DeepMaskNet, an end-to-end framework trained on their MDMFR dataset for detecting mask usage and recognizing identities under mask occlusion. They report accuracy, precision, recall, and F1-score for both detection and recognition modules, showing that the model maintains high recall and F1 even under variations in pose, illumination, mask type, and occlusion, thereby demonstrating that the chosen metrics are sensitive enough to capture performance degradation under real-world conditions [5]. More recent works on parallel or multi-branch CNNs for mask detection similarly present confusion matrices and derive all four metrics to compare variants with different backbones or feature-fusion strategies [48].

In object-detection–oriented approaches (e.g., YOLO or SSD variants), accuracy, precision, recall, and F1-score are frequently complemented by detection-specific metrics such as Average Precision (AP) and mean Average Precision (mAP), which jointly evaluate both localisation and classification performance. For instance, Kumar et al. employed these metrics in their ETL-YOLOv4 framework, showing improvements in mAP across multiple IoU thresholds when enhancing the YOLOv4 backbone for mask-related detection tasks. Beyond this, several recent studies have demonstrated how mAP, along with precision–recall curves, provides deeper insights into robustness under real-world variability [66]. Sheikh and Zafar evaluated their RRFMDS real-time detection system using accuracy, precision, recall, F1-score, and mAP, demonstrating that balanced behavior across these

 doi:10.20944/preprints202601.0505.v1

metrics is essential for consistent detection in crowded surveillance settings [15]. Similarly, Umer et al. compared their CNN-based detector against YOLOv3 and Faster R-CNN using accuracy, precision, recall, F1-score and AP values on multiple datasets, showing how detection-oriented metrics reveal performance gaps in scenarios involving occlusion, lighting variation, and mask-wearing diversity [62]. Collectively, these findings indicate that AP (Average Precision), mAP (Mean Average Precision) and precision-recall analysis play a crucial role when evaluating detection-centric mask-monitoring systems, especially for models deployed in unconstrained or real-time environments.

Survey and review papers on masked-face detection and recognition reinforce the central role of these four metrics. Hosny et al. systematically analyze deep-learning–based masked face detection methods and observe that almost all recent works report at least accuracy and F1-score, with many also giving precision and recall to capture trade-offs between false alarms and missed detections [67]. Further, several studies and surveys note that additional metrics such as **specificity** (true-negative rate), **Receiver Operating Characteristic (ROC) curves**, **Area Under the ROC Curve (AUC)**, and **Intersection over Union (IoU)**, are increasingly employed alongside accuracy, precision, recall, and F1-score to provide a more comprehensive picture of model performance, especially for real-time, safety-critical deployments and highly imbalanced datasets [1,6–9,68–73]. **Table 5** summarizes which evaluation metrics are reported across a representative subset of face mask detection studies.

Table 5. Evaluation Metrics Reported Across Selected Face-Mask Detection Studies.

Study (Ref.)	Accuracy	Precision	Recall	F1-Score	AP	mAP	ROC / AUC	Use Case / Interpretation in Mask Detection
[2] Sethi et al., 2021	✓	✓	✓	✓				Binary classifier; strong balanced metrics on curated datasets
[3] Wu et al., 2022 (FMD-YOLO)	✓	✓	✓	✓	✓	✓		YOLO detection; AP/mAP used for bounding-box evaluation
[5] Ullah et al., 2022 (DeepMaskNet)	✓	✓	✓	✓				Detection + masked-face recognition; reports full metric suite
[18] VGG (Simonyan & Zisserman)	(✓ ImageNet)							Backbone for early mask-classification pipelines
[21] ResNet (He et al.)	(✓ ImageNet)							Backbone widely reused in mask detection & compliance tasks

 doi:10.20944/preprints202601.0505.v1

[25] R-CNN (Girshick et al.)					✓	✓		Basis for two-stage detectors adapted for mask detection
[27] Faster R-CNN (Ren et al.)					✓	✓		Used in early mask detectors assessing region-level AP/mAP
[33] MobileNetV2 (Sandler et al.)	✓	✓	✓	✓				Lightweight backbone for fast mask/no-mask classification
[37] Nagrath et al., 2021 (SSDMNV2)	✓	✓	✓	✓				SSD + MobileNetV2; used in real-time mask detection systems
[44] Loey et al., 2021 (YOLOv2–ResNet50)	✓	✓	✓	✓	✓	✓		Hybrid YOLO-based medical mask detector
[64] Bania et al., 2023 (Ensemble TL)	✓	✓	✓	✓			✓	Ensemble ResNet50/Inception /VGG; includes ROC curve & AUC ≈ 0.99
[73] Benítez-Baltazar et al., 2021 (IoT Mask Detection)	✓	✓	✓	✓			✓	IoT access-control system; explicitly reports ROC curve & AUC ≈ 0.96

Notably, only a few recent works explicitly report ROC curves and AUC values for face-mask detection, such as ensemble-based detectors and IoT access-control systems, whereas most studies rely primarily on accuracy, precision, recall, F1-score, and mAP.

4.2. Trade-Offs Between Accuracy and Efficiency

The trade-off between accuracy and computational efficiency is a central challenge in the development and deployment of deep learning models for face-mask detection. Heavyweight architectures, such as **VGGNet** [18], **ResNet** [21], **DenseNet** [22], and Inception variants [19,20], typically achieve high accuracy on curated datasets but incur substantial computational and memory demands. Their high parameter counts, slow inference speeds on embedded processors, and significant GPU memory requirements make them less suitable for real-time surveillance or IoT applications with strict latency and power constraints. Consequently, practical deployments

 doi:10.20944/preprints202601.0505.v1

increasingly favor architectures that aim to balance representational capacity with computational efficiency, especially in resource-constrained public health monitoring scenarios.

4.2.1. Impact of Model Size and Architecture on Accuracy and Efficiency

Lightweight models, including MobileNet, EfficientNet, ShuffleNet, GhostNet, YOLOv3-tiny, and YOLOv4-tiny, reduce parameter count and inference latency by design, enabling fast deployment on embedded or mobile devices. For example, YOLOv4-tiny contains approximately six million parameters, nearly one-tenth of YOLOv4, and offers significantly higher detection speed, making it more suitable for real-time applications while still incurring a reduction in detection accuracy compared to the full model [74,75]. Similar observations are reported across lightweight design attempts: decreasing the dimensionality of feature maps can reduce inference time but often results in higher false detections. For instance, LSNet reduces feature dimensionality by 35% but increases false-detection rates by 41%, whereas models like SwinNet (198M parameters) achieve higher accuracy but remain computationally prohibitive for real-time edge deployment [37].

Some lightweight models nonetheless demonstrate strong balanced performance. MobileNetV2 frequently achieves accuracy around 92–93% with F1-scores above 0.90 in real-time scenarios [62,75]. Similarly, an ultra-lightweight model with only 0.12M parameters achieves competitive accuracy (95.41% for binary and 95.54% for three-class classification), showing that significant parameter reduction is possible without proportional loss in performance [76].

4.2.2. Comparative Performance of Lightweight and Heavyweight Models

Heavyweight models such as Mask R-CNN with FPN backbones, DenseNet201, or ConvNeXt-T variants achieve high accuracy (often approaching 99%) but are unsuitable for real-time edge deployment due to their size and slow inference speeds [77–79]. For example, Mask R-CNN ConvNeXt-T FPN provides state-of-the-art accuracy but is impractical for hardware-limited environments [80].

Hybrid and ensemble approaches offer a middle ground. Combining lightweight and heavyweight components can improve accuracy while preserving efficiency. An ensemble of single- and two-stage detectors achieved approximately 98.2% accuracy with an average inference time of 0.05 seconds, demonstrating that such hybrid strategies can meet both accuracy and latency requirements in real-time deployments [2]. Many studies emphasize that model selection for deployment should reflect hardware limitations and operational scenarios, balancing accuracy with speed, power consumption, and available compute resources [74,75,80,81].

To better contextualize these performance trends, it is necessary to compare the underlying architectural families that shape each model's computational behavior. Differences in parameter count, convolutional design, memory footprint, and inference throughput directly determine whether architecture can meet real-time requirements on edge devices or embedded systems. A structural comparison of the major architecture families used in face-mask detection is presented in **Table 6**, highlighting differences in parameter size, inference speed, memory consumption, and suitability for edge deployment.

 doi:10.20944/preprints202601.0505.v1

Table 6. comparison of the major architecture families used in face-mask detection.

Architecture Family	Representative Models	Parameter Scale	FPS on Edge Devices (Jetson / RPi / low-power GPU)	Memory / Deployment Characteristics	Suitability for Real-Time Mask Detection
Conventional CNN Backbones	VGG16/19, ResNet50 [18,21], DenseNet121 [22], InceptionV3 [19], classical transfer-learning approaches	High (8M–140M+)	Low–Moderate (<10–15 FPS without optimization)	Require GPU-class memory; heavy compute	High accuracy under controlled datasets but generally not suitable for real-time edge deployment
Two-Stage Detectors (R-CNN Family)	R-CNN [25], Fast R-CNN [26], Faster R-CNN [27], Mask R-CNN [28]	High + region proposal overhead	Low (<5–10 FPS on Jetson; often <5 FPS on RPi)	Large VRAM usage; very slow on CPUs	Excellent detection accuracy, but too slow for practical edge-device mask monitoring
Single-Stage Detectors (Heavy Backbones)	YOLOv2–ResNet50 [44], ETL-YOLOv4 [66], drone-based YOLO [59]	Moderate–High (40M–60M+)	Moderate (10–30 FPS on Jetson Xavier; <15 FPS on Nano/RPi)	Need GPU acceleration; moderate memory	Suitable for edge devices only with optimization; strong accuracy but mixed speed
Lightweight CNN Backbones (Classification)	MobileNetV1/V2/V3 [32–34], EfficientNet-B0 [11], ShuffleNet [35], SqueezeNet [36]; mask-detection works [37–39,42]	Low (1M–5M range)	High (30–60 FPS on Jetson Nano; usable on RPi)	Very small footprint; easy to quantize and prune; CPU-friendly	Excellent for fast mask classification once faces are detected; ideal for edge and mobile deployment
Lightweight Single-Stage Detectors	SSD-MobileNetV2 (SSDMNV2) [37,45], EfficientMask-Net [41], YOLOv4-tiny / YOLOv5-s variants, SqueezeMaskNet [43]	Low–Moderate (2M–10M)	High (25–90 FPS depending on platform)	Optimized for low memory; fits into IoT/embedded systems	Best trade-off between accuracy and speed; preferred choice for real-time mask detection on edge devices
Hybrid & Attention-	SqueezeMaskNet with attention [43],	Low–Moderate	High (25–60 FPS with	Slightly heavier than	Very promising direction:

 doi:10.20944/preprints202601.0505.v1

Enhanced Architectures	YOLOv5+CoordAttention [46], hybrid MobileNetV2 + detection head [37], IoT-optimized deep learning [50,54].	(slightly higher due to attention modules)	optimized pipelines)	lightweight CNNs but still edge deployable	improved robustness (occlusion, clutter) while remaining efficient
Extreme Lightweight / Frugal / Deployment-Engineered Models	Pruned & quantized SqueezeNet/SqueezeMaskNet [43], frugal object detectors [57], augmentation-resilient edge detectors [56]	Very Low (<1M–3M)	Very High (60+ FPS even on modest devices)	Minimal memory; optimized for microcontrollers, NPUs, or minimal-GPU boards	Ideal for massive IoT, smart-city nodes, or hundreds of camera feeds with strict power limits; slight accuracy trade-off

This contextual foundation supports the subsequent analysis of accuracy–efficiency patterns across representative models. The strategies discussed next build upon these architectural characteristics to further enhance model performance under resource constraints. By understanding the inherent computational profiles of each architecture family, it becomes clearer how transfer learning, data augmentation, and model-level modifications can be applied to strengthen accuracy without compromising real-time efficiency.

Strategies to Improve the Trade-Off

Transfer Learning: Utilizing pre-trained models and transfer learning can enhance the accuracy of lightweight models without significantly increasing computational demands. For example, transfer learning with MobileNetV2 and ResNet50 has been shown to improve detection accuracy in real-time applications [2,9,62].

Data Augmentation and Balanced Datasets: Techniques such as random over-sampling and data augmentation help improve model accuracy by addressing class imbalance, as seen in studies that reduced imbalance ratios and achieved high accuracy with efficient models [2,76].

Model Modifications: Modifying backbone networks, activation functions, and loss functions, as done in YOLOv4 and its variants, can help maintain good accuracy while improving speed and reducing resource consumption [9,74].

The comparative relationships between accuracy and computational efficiency across representative models are summarized in **Table 7** providing a consolidated view of the trade-offs discussed in this section.

Table 7. Summary Table: Accuracy vs. Efficiency in Selected Models.

Model	Parameters (approx.)	Accuracy (%)	Speed/Resource Use	Notes
YOLOv4-tiny	~6M	Lower than YOLOv4	Fast, low resource	1/10th parameters of YOLOv4 [74]

 doi:10.20944/preprints202601.0505.v1

MobileNetV2	Lightweight	~92.6	Real-time, embedded devices	Robust for real-time use [37,62,75]
DenseNet201	Heavyweight	99	Slower, high resource	Highest accuracy in comparison [78]
Mask R-CNN ConvNeXt-T	Heavyweight	Highest	Not suitable for real-time	Best accuracy, poor efficiency [80]
Custom Lightweight Net	0.12M	~95.5	Highly efficient	Up to 496x parameter reduction [76]
Ensemble (Single+Two)	-	98.2	0.05s/image	High accuracy and speed [2]

Overall, while conventional CNNs deliver high accuracy, their computational cost limits deployment on real-time, hardware-constrained platforms. Lightweight architectures offer high efficiency but may lack robustness in complex scenes. Hybrid and ensemble frameworks consistently offer the most practical balance between accuracy and speed, making them suitable for real-time monitoring in smart-city, IoT, and embedded environments. Thus, the optimal choice depends on the deployment context: accuracy-critical applications may tolerate heavier models, whereas large-scale real-time monitoring benefits most from efficient lightweight or hybrid architectures.

5. Future Research Directions

Despite the significant progress achieved through conventional, lightweight, and hybrid architectures, several open challenges continue to limit the robustness, generalizability, and scalability of face-mask detection systems. These challenges present rich opportunities for future research, particularly in improving multi-class capability, domain adaptation, model compression, and extending mask-detection frameworks toward broader applications within public-health monitoring and intelligent surveillance.

5.1. Improper Mask Detection and Multi-Class Analysis

Most existing models are optimized for binary mask detection (mask vs. no mask), yet improper mask wearing remains a major real-world compliance issue. Identifying masks worn below the nose, under the chin, partially covering the mouth, or loosely attached requires fine-grained, region-aware feature extraction that many lightweight systems struggle with. Conventional CNNs offer strong discriminative capacity but are computationally expensive, while lightweight architectures may misclassify subtle misuse cases due to reduced representational depth. Hybrid detectors improve robustness, but their performance still degrades when improper mask patterns exhibit high variability across individuals, mask materials, or face shapes.

 doi:10.20944/preprints202601.0505.v1

Future work must therefore address multi-class imbalance, generate or augment datasets with diverse improper-wear scenarios, and incorporate region-specific constraints or anatomical priors. Approaches such as fine-grained visual reasoning, dense part-based attention, or landmark-guided mask positioning analysis could substantially improve improper-mask detection accuracy.

5.2. Domain Adaptation and Real-World Variability

A persistent challenge across all model families is domain shift, the discrepancy between curated training datasets and real-world deployment conditions. Surveillance environments introduce uncontrolled variables including illumination changes, shadows, motion blur, camera angle variations, diverse mask designs, facial accessories, occlusions (hands, hair, objects), and crowd density. Models trained purely on curated datasets such as MaskedFace-Net or PWMFD often suffer significant accuracy drops when deployed in unseen domains, demonstrating insufficient generalization.

Future research must explore domain-adaptation strategies, such as:

- **Unsupervised domain adaptation (UDA)** for aligning feature distributions across environments.
- **Self-supervised representation learning** to reduce dependency on labels
- **Cross-dataset training pipelines** that incorporate heterogeneous noise, mask materials, and cultural variations
- **Synthetic domain randomization** to simulate low-quality or occluded footage

Enhancing robustness under real-world variability is critical for scalable deployment across transportation hubs, hospitals, campuses, and crowded public spaces.

5.3. Expanding Applications Beyond Mask Detection

As mask usage declines post-pandemic, face-mask detection systems should evolve toward broader public-safety, healthcare, and compliance-monitoring applications. Since many architectural foundations are adaptable such as lightweight CNNs, attention-enhanced detectors, and hybrid YOLO variants, researchers can repurpose these systems for:

- **PPE compliance monitoring** (helmets, gloves, lab coats)
- **Human behavior analysis** (face-touching detection, cough detection, proximity violations)
- **Health screening** (visible respiratory cues, temperature screening integration)
- **Access-control and identity verification under occlusion**
- **Crowd analytics and anomaly detection** for smart-city infrastructure

Furthermore, integrating mask detection with multimodal sensing (audio, thermal imaging, RFID) can enable holistic public-health monitoring systems. The shift from task-specific detectors to generalizable compliance-monitoring frameworks represents a major future research opportunity.

6. Conclusion

This review analyzed the architectural evolution of deep learning–based face mask detection systems, tracing the progression from conventional CNNs to lightweight and hybrid architectures designed for real-time deployment on resource-constrained platforms. While traditional models such as VGGNet, ResNet, and DenseNet offer strong feature extraction capabilities, their high computational and memory demands limit their practicality for embedded and IoT applications. Lightweight architectures, including MobileNet, EfficientNet, ShuffleNet, and other efficient convolutional families, significantly reduce inference latency and parameter count, enabling practical deployment without severe accuracy degradation. Hybrid models, which integrate lightweight backbones with optimized detection heads or multi-branch processing strategies, further advance the accuracy–efficiency balance and represent the most promising direction for scalable real-time compliance monitoring.

 doi:10.20944/preprints202601.0505.v1

The comparative performance analysis demonstrated that no single architecture simultaneously maximizes accuracy, robustness, and efficiency; rather, each model family offers strengths tuned to particular deployment requirements. Open challenges remain in improper-mask detection, domain adaptation to real-world variability, and the need for compression-oriented techniques such as pruning and knowledge distillation. Future research must therefore focus on developing architectures that generalize robustly across environments, operate reliably under multi-class conditions, and support energy-efficient computation on embedded devices. Additionally, as mask usage transitions from pandemic-driven contexts to broader occupational safety applications, the underlying architectural principles reviewed here provide a foundation for more general compliance-monitoring systems.

Overall, this review contributes an architecture-centered perspective on face mask detection, clarifying design principles, identifying limitations, and outlining research directions to support the development of efficient, reliable, and scalable monitoring systems suitable for modern edge-computing environments

References

1. M. Liang et al., "Efficacy of face mask in preventing respiratory virus transmission: A systematic review and meta-analysis," *Travel Med Infect Dis*, vol. 36, p. 101751, Jul. 2020, doi: 10.1016/J.TMAID.2020.101751.
2. S. Sethi, M. Kathuria, and T. Kaushik, "Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread," *J Biomed Inform*, vol. 120, Aug. 2021, doi: 10.1016/j.jbi.2021.103848.
3. P. Wu, H. Li, N. Zeng, and F. Li, "FMD-Yolo: An efficient face mask detection method for COVID-19 prevention and control in public," *Image Vis Comput*, vol. 117, p. 104341, Jan. 2022, doi: 10.1016/J.IMAVIS.2021.104341.
4. D. Kolosov, V. Kelefouras, P. Kourtessis, and I. Mporas, "Anatomy of Deep Learning Image Classification and Object Detection on Commercial Edge Devices: A Case Study on Face Mask Detection," *IEEE Access*, vol. 10, p. 109167, Oct. 2022, doi: 10.1109/ACCESS.2022.3214214.
5. N. Ullah, A. Javed, M. Ali Ghazanfar, A. Alsufyani, and S. Bourouis, "A novel DeepMaskNet model for face mask detection and masked facial recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 9905–9914, Nov. 2022, doi: 10.1016/J.JKSUCI.2021.12.017.
6. S. F. Abbas, S. H. Shaker, and Firas. A. Abdullatif, "Face Mask Detection Based on Deep Learning: A Review," *Journal of Soft Computing and Computer Applications*, vol. 1, no. 1, Jun. 2024, doi: 10.70403/3008-1084.1006.
7. F. Amer, M. Ali, and M. S. H. Al-Tamimi, "Face mask detection methods and techniques: A review," *Int. J. Nonlinear Anal. Appl*, vol. 13, pp. 2008–6822, 2022, doi: 10.22075/ijnaa.2022.6166.
8. Vibhuti, N. Jindal, H. Singh, and P. S. Rana, "Face mask detection in COVID-19: a strategic review," *Multimedia Tools and Applications 2022 81:28*, vol. 81, no. 28, pp. 40013–40042, May 2022, doi: 10.1007/S11042-022-12999-6.
9. R. Alturki, M. Alharbi, F. AlAnzi, and S. Albahli, "Deep learning techniques for detecting and recognizing face masks: A survey," *Front Public Health*, vol. 10, p. 955332, Sep. 2022, doi: 10.3389/FPUBH.2022.955332.
10. N. Anggraini, S. H. Ramadhani, L. K. Wardhani, N. Hakiem, I. M. Shofi, and M. T. Rosyadi, "Development of Face Mask Detection using SSDLite MobilenetV3 Small on Raspberry Pi 4," in *2022 5th International Conference on Computer and Informatics Engineering, IC2IE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 209–214. doi: 10.1109/IC2IE56416.2022.9970078.
11. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, May 2019, Accessed: Dec. 01, 2025. [Online]. Available: https://arxiv.org/pdf/1905.11946
12. S. A. Sanjaya and S. A. Rakhmawan, "Face Mask Detection Using MobileNetV2 in the Era of COVID-19 Pandemic," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi: 10.1109/ICDABI51230.2020.9325631.

13. Y. Shao, J. Ning, H. Shao, D. Zhang, H. Chu, and Z. Ren, "Lightweight face mask detection algorithm with attention mechanism," *Eng Appl Artif Intell*, vol. 137, p. 109077, Nov. 2024, doi: 10.1016/J.ENGAPPAI.2024.109077.
14. R. Dodda, C. Raghavendra, U. Raghavendra Swamy, C. N. Azmera, M. Sreenu, and S. Nimmala, "Real-Time Face Mask Detection Using Deep Learning: Enhancing Public Health and Safety," *E3S Web of Conferences*, vol. 616, p. 02013, Feb. 2025, doi: 10.1051/E3SCONF/202561602013.
15. B. ul haque Sheikh and A. Zafar, "RRFMDS: Rapid Real-Time Face Mask Detection System for Effective COVID-19 Monitoring," *SN Comput Sci*, vol. 4, no. 3, May 2023, doi: 10.1007/s42979-023-01738-9.
16. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.
17. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386;CSUBTYPE:STRING:MAGAZINE;PAGE:STRING:ARTICLE/CHAPTER.
18. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2014, Accessed: Dec. 04, 2025. [Online]. Available: https://arxiv.org/pdf/1409.1556
19. C. Szegedy et al., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 07-12-June-2015, pp. 1–9, Jun. 2015, doi: 10.1109/CVPR.2015.7298594.
20. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 4278–4284, Feb. 2016, doi: 10.1609/aaai.v31i1.11231.
21. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, Dec. 2016, doi: 10.1109/CVPR.2016.90.
22. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, Nov. 2017, doi: 10.1109/CVPR.2017.243.
23. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 1800–1807, Oct. 2016, doi: 10.1109/CVPR.2017.195.
24. I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 10425–10433, 2020, doi: 10.1109/CVPR42600.2020.01044.
25. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Columbus: IEEE Computer Society, Sep. 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81.
26. R. Girshick, "Fast R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Sep. 2015, pp. 1440–1448. Accessed: Dec. 04, 2025. [Online]. Available: doi: 10.1109/ICCV.2015.169
27. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, Jun. 2015, doi: 10.1109/TPAMI.2016.2577031.
28. K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, vol. 2017-October, pp. 2980–2988, Oct. 2017, doi: 10.1109/ICCV.2017.322.
29. A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19," *Smart Health (Amst)*, vol. 19, p. 100144, Mar. 2020, doi: 10.1016/J.SMHL.2020.100144.
30. X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, "Real-Time Face Mask Detection Method Based on YOLOv3," *Electronics 2021, Vol. 10, Page 837*, vol. 10, no. 7, p. 837, Apr. 2021, doi: 10.3390/ELECTRONICS10070837.
31. M. Mahmoud, M. S. E. Kasem, and H. S. Kang, "A Comprehensive Survey of Masked Faces: Recognition, Detection, and Unmasking," *Applied Sciences 2024, Vol. 14, Page 8781*, vol. 14, no. 19, p. 8781, Sep. 2024, doi: 10.3390/APP14198781.

 doi:10.20944/preprints202601.0505.v1

32. A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, Accessed: Dec. 01, 2025. [Online]. Available: https://arxiv.org/pdf/1704.04861
33. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Jan. 2018, doi: 10.1109/CVPR.2018.00474.
34. A. Howard et al., "Searching for mobileNetV3," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1314–1324, Oct. 2019, doi: 10.1109/ICCV.2019.00140.
35. X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6848–6856, Jun. 2018, doi: 10.1109/CVPR.2018.00716.
36. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," Feb. 2016, Accessed: Dec. 01, 2025. [Online]. Available: https://arxiv.org/pdf/1602.07360
37. P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustain Cities Soc*, vol. 66, p. 102692, Mar. 2021, doi: 10.1016/J.SCS.2020.102692.
38. A. H. I. Al-Rammahi, "Face mask recognition system using MobileNetV2 with optimization function," *Applied Artificial Intelligence*, vol. 36, no. 1, Dec. 2022, doi: 10.1080/08839514.2022.2145638;PAGE:STRING:ARTICLE/CHAPTER.
39. Fadly, T. B. Kurniawan, D. A. Dewi, M. Z. Zakaria, and P. A. A. B. Hisham, "Deep Learning Based Face Mask Detection System Using MobileNetV2 for Enhanced Health Protocol Compliance," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 2067–2078, Nov. 2024, doi: 10.47738/JADS.V5I4.476.
40. M. Sharma, H. Gunwant, P. Saggar, L. Gupta, and D. Gupta, "EfficientNet-B0 Model for Face Mask Detection Based on Social Information Retrieval," *International Journal of Information System Modeling and Design*, vol. 13, no. 7, Jan. 2022, doi: 10.4018/IJISMD.313444.
41. N. Azouji, A. Sami, and M. Taheri, "EfficientMask-Net for face authentication in the era of COVID-19 pandemic," *Signal, Image and Video Processing 2022 16:7*, vol. 16, no. 7, pp. 1991–1999, Apr. 2022, doi: 10.1007/S11760-022-02160-Z.
42. B. B. Chakma, M. A. Masud, T. Ahamed, and M. H. Tusher, "IDENTIFICATION OF FACE MASK USING CONVOLUTIONAL NEURAL NETWORK-BASED EFFICIENTNET MODEL," *Khulna University Studies*, pp. 531–538, Nov. 2022, doi: 10.53808/KUS.2022.ICSTEM4IR.0096-SE.
43. G. Benitez-Garcia, L. Prudente-Tixteco, J. Olivares-Mercado, and H. Takahashi, "SqueezeMaskNet: Real-Time Mask-Wearing Recognition for Edge Devices," *Big Data and Cognitive Computing 2025, Vol. 9, Page 10*, vol. 9, no. 1, p. 10, Jan. 2025, doi: 10.3390/BDCC9010010.
44. M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," *Sustain Cities Soc*, vol. 65, p. 102600, Feb. 2021, doi: 10.1016/J.SCS.2020.102600.
45. B. Karthikeyan and S. Gowri, "A Real-Time Face Mask Detection Using SSD and MobileNetV2," *Proceedings of the 2021 4th International Conference on Computing and Communications Technologies, ICCCT 2021*, pp. 144–148, 2021, doi: 10.1109/ICCCT53315.2021.9711784.
46. T. N. Pham, V. H. Nguyen, and J. H. Huh, "Integration of improved YOLOv5 for face mask detector and auto-labeling to generate dataset for fighting against COVID-19," *The Journal of Supercomputing 2023 79:8*, vol. 79, no. 8, pp. 8966–8992, Jan. 2023, doi: 10.1007/S11227-022-04979-2.
47. M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement*, vol. 167, p. 108288, Jan. 2021, doi: 10.1016/J.MEASUREMENT.2020.108288.
48. T. Tabassum et al., "A Parallel Convolutional Neural Network for Accurate Face Mask Detection in the Fight Against COVID-19," *Biomedical Materials & Devices 2025*, pp. 1–11, Jun. 2025, doi: 10.1007/S44174-025-00390-6.

 doi:10.20944/preprints202601.0505.v1

49. S. B. U. Haque, "A fuzzy-based frame transformation to mitigate the impact of adversarial attacks in deep learning-based real-time video surveillance systems," *Appl Soft Comput*, vol. 167, p. 112440, Dec. 2024, doi: 10.1016/J.ASOC.2024.112440.
50. P. Dubey, P. Dubey, C. Iwendi, C. N. Biamba, and D. D. Rao, "Enhanced IoT-Based Face Mask Detection Framework Using Optimized Deep Learning Models: A Hybrid Approach with Adaptive Algorithms," *IEEE Access*, vol. 13, pp. 17325–17339, 2025, doi: 10.1109/ACCESS.2025.3532764.
51. D. Parikh, A. Karthikeyan, V. Ravi, M. Shibu, R. Singh, and R. S. Sofana, "IoT and ML-driven framework for managing infectious disease risks in communal spaces: a post-COVID perspective," *Front Public Health*, vol. 13, p. 1552515, May 2025, doi: 10.3389/FPUBH.2025.1552515/BIBTEX.
52. C. D. Truong, S. Mishra, N. Q. Long, and L. A. Ngoc, "Efficient Face Mask Detection for Banking Information Systems," *Creative Approaches Towards Development of Computing and Multidisciplinary IT Solutions for Society*, pp. 435–454, Jan. 2024, doi: 10.1002/9781394272303.CH28.
53. X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, "Real-time face mask detection method based on yolov3," *Electronics (Switzerland)*, vol. 10, no. 7, 2021, doi: 10.3390/electronics10070837.
54. Y. Himeur, S. Al-Maadeed, I. Varlamis, N. Al-Maadeed, K. Abualsaud, and A. Mohamed, "Face Mask Detection in Smart Cities Using Deep and Transfer Learning: Lessons Learned from the COVID-19 Pandemic," *Systems 2023, Vol. 11, Page 107*, vol. 11, no. 2, p. 107, Feb. 2023, doi: 10.3390/SYSTEMS11020107.
55. A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel, "EdgeFace: Efficient Face Recognition Model for Edge Devices," *IEEE Trans Biom Behav Identity Sci*, vol. 6, no. 2, pp. 158–168, Apr. 2024, doi: 10.1109/TBIOM.2024.3352164.
56. T. N. Anh and V. D. Nguyen, "MAPBoost: augmentation-resilient real-time object detection for edge deployment: Augmentation-resilient lightweight detection," *J Real Time Image Process*, vol. 23, no. 1, Jan. 2026, doi: 10.1007/S11554-025-01805-9.
57. A. Hamdi, H. Noura, J. Azar, and G. Pujolle, "Frugal Object Detection Models: Solutions, Challenges and Future Directions," *21st International Wireless Communications and Mobile Computing Conference, IWCMC 2025*, pp. 1694–1701, 2025, doi: 10.1109/IWCMC65282.2025.11059526.
58. J. Qian, S. Mu, H. Lu, and S. Xu, "Two-stage model re-optimization and application in face recognition," *Neurocomputing*, vol. 651, p. 130805, Oct. 2025, doi: 10.1016/J.NEUCOM.2025.130805.
59. S. A. Mostafa et al., "A YOLO-based deep learning model for Real-Time face mask detection via drone surveillance in public spaces," *Inf Sci (N Y)*, vol. 676, p. 120865, Aug. 2024, doi: 10.1016/J.INS.2024.120865.
60. D. Hussain, M. Ismail, I. Hussain, R. Alroobaea, S. Hussain, and S. S. Ullah, "Face Mask Detection Using Deep Convolutional Neural Network and MobileNetV2-Based Transfer Learning," *Wirel Commun Mob Comput*, vol. 2022, 2022, doi: 10.1155/2022/1536318.
61. I. Hagui, A. Msolli, A. Helali, and H. Fredj, "Face Mask Detection using CNN: A Fusion of Cryptography and Blockchain," *Engineering, Technology and Applied Science Research*, vol. 14, no. 5, pp. 17156–17161, Oct. 2024, doi: 10.48084/etasr.7827.
62. M. Umer et al., "Face mask detection using deep convolutional neural network and multi-stage image processing," *Image Vis Comput*, vol. 133, p. 104657, May 2023, doi: 10.1016/J.IMAVIS.2023.104657.
63. J. V. B. Benifa et al., "FMDNet: An Efficient System for Face Mask Detection Based on Lightweight Model during COVID-19 Pandemic in Public Areas," *Sensors*, vol. 23, no. 13, Jul. 2023, doi: 10.3390/s23136090.
64. R. K. Bania, "Ensemble of deep transfer learning models for real-time automatic detection of face mask," *Multimed Tools Appl*, vol. 82, no. 16, p. 1, Jul. 2023, doi: 10.1007/S11042-023-14408-Y.
65. Z. Q. Habeeb and I. Al-Zaydi, "Incorrect facemask-wearing detection using image processing and deep learning," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 2212–2219, Aug. 2023, doi: 10.11591/eei.v12i4.4786.
66. A. Kumar, A. Kalia, and A. Kalia, "ETL-YOLO v4: A face mask detection algorithm in era of COVID-19 pandemic," *Optik (Stuttg)*, vol. 259, p. 169051, Jun. 2022, doi: 10.1016/J.IJLEO.2022.169051.
67. K. M. Hosny, N. AbdElFattah Ibrahim, E. R. Mohamed, and H. M. Hamza, "Artificial intelligence-based masked face detection: A survey," *Intelligent Systems with Applications*, vol. 22, p. 200391, Jun. 2024, doi: 10.1016/J.ISWA.2024.200391.

68. M. Mahmoud, M. S. E. Kasem, and H. S. Kang, "A Comprehensive Survey of Masked Faces: Recognition, Detection, and Unmasking," *Applied Sciences 2024, Vol. 14, Page 8781*, vol. 14, no. 19, p. 8781, Sep. 2024, doi: 10.3390/APP14198781.
69. E. Mbunge, S. Simelane, S. G. Fashoto, B. Akinnuwesi, and A. S. Metfula, "Application of deep learning and machine learning models to detect COVID-19 face masks - A review," *Sustainable Operations and Computers*, vol. 2, pp. 235–245, Jan. 2021, doi: 10.1016/J.SUSOC.2021.08.001.
70. A. O. Mulani and T. M. Kulkarni, "Face Mask Detection System Using Deep Learning: A Comprehensive Survey," *Communications in Computer and Information Science*, vol. 2439 CCIS, pp. 25–33, 2025, doi: 10.1007/978-3-031-88759-8_3.
71. R. Jayaswal and M. Dixit, "AI-based face mask detection system: a straightforward proposition to fight with Covid-19 situation," *Multimedia Tools and Applications 2022 82:9*, vol. 82, no. 9, pp. 13241–13273, Sep. 2022, doi: 10.1007/S11042-022-13697-Z.
72. A. M. Vukicevic, M. Petrovic, P. Milosevic, A. Peulic, K. Jovanovic, and A. Novakovic, "A systematic review of computer vision-based personal protective equipment compliance in industry practice: advancements, challenges and future directions," *Artificial Intelligence Review 2024 57:12*, vol. 57, no. 12, pp. 319-, Oct. 2024, doi: 10.1007/S10462-024-10978-X.
73. V. H. Benitez-Baltazar et al., "Autonomic Face Mask Detection with Deep Learning: an IoT Application," *Revista mexicana de ingeniería biomédica*, vol. 42, no. 2, pp. 160–170, May 2021, doi: 10.17488/RMIB.42.2.13.
74. Z. Han, H. Huang, Q. Fan, Y. Li, Y. Li, and X. Chen, "SMD-YOLO: An efficient and lightweight detection method for mask wearing status during the COVID-19 pandemic," *Comput Methods Programs Biomed*, vol. 221, p. 106888, Jun. 2022, doi: 10.1016/J.CMPB.2022.106888.
75. A. K. Biswas and K. Roy, "A comparative study on 'face mask detection' using machine learning and deep learning algorithms," *Artificial Intelligence in e-Health Framework, Volume 1: AI, Classification, Wearable Devices, and Computer-Aided Diagnosis*, vol. 1, pp. 193–200, Jan. 2025, doi: 10.1016/B978-0-443-13816-4.00010-3.
76. U. Masud, M. Siddiqui, M. Sadiq, and S. Masood, "SCS-Net: An efficient and practical approach towards Face Mask Detection," *Procedia Comput Sci*, vol. 218, pp. 1878–1887, Jan. 2023, doi: 10.1016/J.PROCS.2023.01.165.
77. E. J. Y. Koh, E. Amini, G. J. McLachlan, and N. Beaton, "Utilising convolutional neural networks to perform fast automated modal mineralogy analysis for thin-section optical microscopy," *Miner Eng*, vol. 173, p. 107230, Nov. 2021, doi: 10.1016/J.MINENG.2021.107230.
78. M. P. Sahoo, M. Sridevi, and R. Sridhar, "Covid prevention based on identification of incorrect position of face-mask," *Procedia Comput Sci*, vol. 235, pp. 1222–1234, Jan. 2024, doi: 10.1016/J.PROCS.2024.04.116.
79. M. Koklu, I. Cinar, and Y. S. Taspinar, "CNN-based bi-directional and directional long-short term memory network for determination of face mask," *Biomed Signal Process Control*, vol. 71, p. 103216, Jan. 2022, doi: 10.1016/J.BSPC.2021.103216.
80. J. Wang, S. Yuan, T. Lu, H. Zhao, and Y. Zhao, "Fusing YOLOv5s-MediaPipe-HRV to classify engagement in E-learning: From the perspective of external observations and internal factors," *Knowl Based Syst*, vol. 305, p. 112670, Dec. 2024, doi: 10.1016/J.KNOSYS.2024.112670.
81. B. Kuriakose, R. Shrestha, and F. E. Sandnes, "DeepNAVI: A deep learning based smartphone navigation assistant for people with visual impairments," *Expert Syst Appl*, vol. 212, p. 118720, Feb. 2023, doi: 10.1016/J.ESWA.2022.118720.