Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871

Vol. 7, No. 1, February 2026, Page. 210-225
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.4461

Comparative Analysis of Face Mask Detection using Lightweight CNN and Bag of Visual Word-based Classifier for Real-Time Surveillance

Ika Candradewi*[1], Bakhtiar Aldino Ardi S[2], Agus Harjoko[3] and Andi Dharmawan[4]

[1,2,3,4]Electronics and Instrumentation, Universitas Gadjah Mada, Indonesia

Email: [1]ika.candradewi@ugm.ac.id

Received : Mar 2, 2025; Revised : Sep 16, 2025; Accepted : Sep 26, 2025; Published : Feb 15, 2026

Abstract

Face mask detection has become increasingly important across various sectors, including healthcare, food processing industries, and public safety, to ensure adherence to health and hygiene protocols and minimize the risks of contamination. Manual supervision of mask usage is often inefficient, labor-intensive, and prone to inconsistency. To address this challenge, this study proposes an automated face mask detection system utilizing computer vision technology, designed for real-time monitoring on resource-limited edge devices, such as the Raspberry Pi 4 with a Google Coral USB Accelerator.

The system integrates Multi-task Cascaded Convolutional Neural Networks (MTCNN) for face detection and a modified lightweight Convolutional Neural Network (CNN) for binary mask classification. Deployed via a web-based platform, it captures images of non-compliant individuals and triggers alerts. To evaluate model performance, the modified CNN is compared with the Bag of Visual Words (BoVW) method using SVM and MLP classifiers on two datasets: the 12k-Face Mask Dataset (Kaggle) and a newly proposed dataset. The CNN model demonstrated higher classification performance than both BoVW-SVM and BoVW-MLP, with AUC improvements of 49% and 43% on the proposed and 12k-Face Mask datasets, respectively.

This study contributes to the advancement of computer vision-based public health monitoring by offering a robust, scalable, and real-time face mask detection system. The findings highlight the practical advantages of deep learning approaches over traditional feature extraction techniques, supporting the development of intelligent, automated surveillance systems and policy enforcement in high-risk environments, which will facilitate future advancements in AI-driven public safety solutions.

Keywords : *Bag of Visual Word, Convolutional Neural Network, Face Mask Detection, Multi-layer Perceptron*

This work is an open-access article licensed under a Creative Commons Attribution 4.0 International License.

1. INTRODUCTION

The need for reliable face mask detection systems extends beyond pandemic response, as they play a crucial role in maintaining hygiene and safety standards in sectors such as healthcare and food processing. Manual supervision of compliance is often inefficient and inconsistent, motivating automated computer vision solutions for real-time monitoring on edge devices. Prior studies have explored both one-stage and two-stage approaches. However, many existing solutions face challenges related to real-time processing, occlusions, lighting variations, and computational limitations on edge devices. This study addresses these limitations by proposing an optimized, real-time deep learning-based face mask detection system designed for deployment on a low-power Raspberry Pi 4 device with a Google Coral USB Accelerator. A comprehensive survey [1] of deep learning-based detection systems, highlighting variations in datasets and methodological differences. while Tiny YOLO v4-SPP [2] improved small-object detection (mAP 64.31%, AP 84.42%) for surveillance. YOLOv3 offers faster inference (0.045 s) than Faster R-CNN but at lower precision, whereas Faster R-CNN achieves higher accuracy (mAP@0.5 = 62) with slower inference (0.15 s) [3]. Variants integrating Faster R-CNN with InceptionV2 [4] achieving an overall accuracy of 97.32% for simple scenes and 91.13% for complex scenes. [5] online attendance system integrating face recognition with mask detection, achieving 81.8% accuracy for face recognition and 80% for mask detection. Single Shot Object Detection (SSD) with MobileNet V2 [6], yielding 85%-95% accuracy in Raspberry Pi. They have been widely explored for real-time object detection, including face mask applications. While one-stage detectors offer speed, two-stage detectors often achieve better accuracy but at a higher computational cost. While one-stage detectors offer speed, two-stage detectors often achieve better accuracy but at a higher computational cost, such as Viola-Jones for face detection [7], MTCNN [8][9], have been employed to isolate facial regions before classification.

On the classification side, conventional approaches employed handcrafted features—such as BoVW with SVM [10], SVM for gender classification from face [11], and PCA with NN [12],—which achieve reasonable results in controlled datasets but are less reliable under challenging conditions. On the other hand, deep learning-based classifiers, especially CNNs, offer better generalization through automatic feature extraction. Other research has utilized the HGL method for head pose classification [13]. A four-stage detection method using a pinhole camera model [14] reached 90% accuracy, while SR network integration with MTCNN [15] improved accuracy to 98.7%. [16] introduced a Novel Framework combining Proposal, Classification, Regression, and a decouple spatial attention module, achieving 81.2 % for mean Average Precision (mAP). Another study [17] developed a KNN-CNN-based face-mask detection system, leveraging Locally Linear Embedding (LLE) for feature extraction, and achieved a precision of 76.4%. Deep learning-based approaches, particularly Convolutional Neural Networks (CNNs), have emerged as the dominant paradigm due to their ability to learn hierarchical feature representations directly from raw data and achieve higher accuracy and robustness for real-world mask detection tasks [18][19][20]. Modified CNNs such as VGG16 [21] even with small training datasets and a modified VGG19 with a similarity model [22] and YOLOv2 with ResNet-50 [23] have achieved 80–85% accuracy, while evaluation strategies [24] emphasize the importance of robust metrics. Still, high-performing CNNs are computationally demanding, limiting deployment on low-power hardware.

To address these constraints, lightweight architectures optimized for mobile and embedded applications. A representative multimodal design couples masked face recognition with non-contact temperature screening on a Raspberry Pi 4 using a FLIR Lepton sensor and a FaceNet backbone trained with simulated masks; in a small real-world trial (17 subjects) it achieved 94.1% masked-face recognition and documented failure cases (e.g., heavy bangs, mask–background color similarity, photochromic lenses), underscoring the value—and limits—of integrated sensing at the edge [25]. Early pipelines typically decouple face detection and mask classification. One representative design detects and crops faces (e.g., via MTCNN) and then applies a shallow CNN trained broadly on synthetic faces, reporting 98.47% test accuracy on a two-class (mask/no-mask) dataset; however, it relies heavily on GAN-generated faces, does not evaluate incorrect-wear cases, and offers no edge-hardware throughput measurements or adverse-condition tests it proposes for future work [26]. A two-stage alternative combines SSD (ResNet-10) for face detection with a MobileNetV2 classifier (“SSDMNV2”), targeting embedded deployment; it attains 0.9264 accuracy and 0.93 F1 on a mixed real/synthetic set and argues for real-time operation on devices such as Jetson Nano or Raspberry Pi, though systematic on-device FPS/latency and cross-dataset generalization are not rigorously quantified [27].

Moving from two-stage to one-stage, YOLO-based detectors reduce latency while adding task-specific inductive bias. A YOLOv3 variant with squeeze-and-excitation attention (SE-YOLOv3) introduces the PWMFD dataset (9,205 images; proper, improper, and no-mask classes) and reports an 8.6% mAP gain over vanilla YOLOv3 at comparable speed—promising for edge scenarios, albeit with dataset imbalance (few “improper” examples) that can limit robustness in the wild [28]. Beyond detection, a mobile system integrates real-time masked-face recognition with lightweight CNNs, demonstrating an on-device app but primarily validating its performance on a small 12-subject set (90.40% validation accuracy), which constrains its external validity for broader deployments [29]. and recent CNN classification studies report accuracies up to 99.89% with pruning/quantization for edge use, though evaluations remain binary and lack constrained-hardware benchmarks, yet still evaluates a binary mask/no-mask task and does not benchmark on constrained hardware—leaving real-time, multi-class compliance checks on edge devices as an open, practical gap [30].

Unlike previous works, this study proposes a modified lightweight CNN architecture optimized for real-time deployment on low-power edge devices. The objective of the proposed system is to integrate MTCNN for efficient face detection and a reduced-depth CNN for mask classification, thereby balancing computational efficiency with classification accuracy. Performance is evaluated using both the publicly available 12k-Face Mask Dataset and a custom low-resolution dataset designed to replicate real-world conditions. The objective of this research is to develop a robust, real-time, and scalable face mask detection system capable of running efficiently on edge devices, and to benchmark its performance against BoVW-SVM and BoVW-MLP classifiers. This research contributes to the advancement of computer vision-based surveillance by demonstrating that a lightweight CNN can achieve high accuracy while remaining computationally efficient, thereby enabling practical deployment in public health, industrial safety, and medical applications.

2. METHOD

The proposed workflow is illustrated in figure 1(a). Live video is acquired on a Raspberry Pi 4 with a Coral USB Accelerator, where frames are resized and normalized before face detection. Detected faces are localized using MTCNN, and each cropped region is forwarded to the classification stage for mask/no-mask prediction using either the baseline BoVW method or the proposed CNN model.

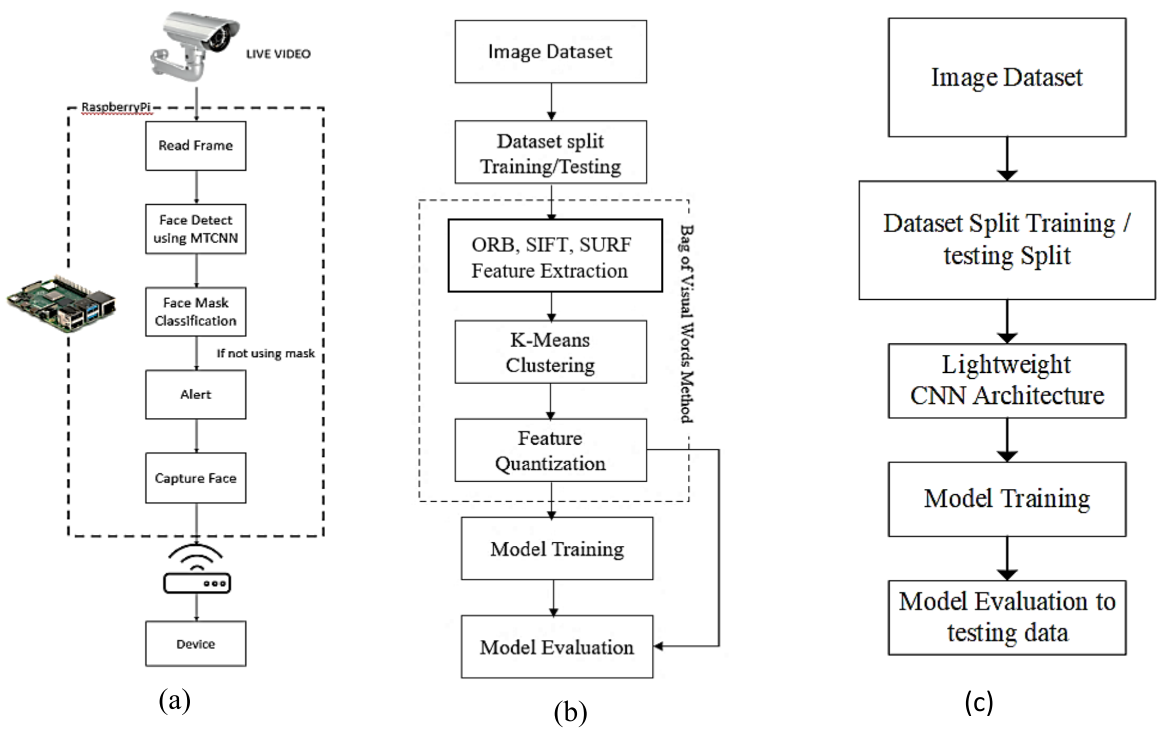


Figure 1. (a) Proposed System Diagram (b) Face Mask Classification Training Stage for BoVW with SVM and/or MLP classifiet Mask Detection (c) Face Mask Classification Training Stage for CNN

The BoVW training process is shown in Figure 1(b). The dataset is divided into training and testing subsets, features are extracted with ORB, SIFT, or SURF, clustered using K-Means to construct a visual vocabulary, and quantized into histograms that serve as inputs for SVM or MLP classifiers.

As illustrated in Figure 1(c), this stage is replaced in the proposed system by a lightweight CNN trained on both a custom dataset and the 12k-Face Mask Dataset, providing robustness to illumination, pose, and mask variations. The overall system integrates a real-time monitoring and alert mechanism that captures violator images, triggers alarms, and logs incidents. Performance is evaluated using accuracy, precision, recall, F1-score, and AUC, demonstrating that the CNN-based approach outperforms BoVW–SVM and BoVW–MLP while maintaining real-time feasibility on edge devices.

2.1. Proposed Dataset

In our experiments, we use two different datasets which shown in Table 1. We evaluate on two datasets. (i) Proposed dataset: 3,039 images captured with a 13-MP smartphone at 1:1 scale; per subject we collect two images (with/without mask) spanning four mask types (N95, cloth, scuba, medical). For video-based evaluation, public crowd scenes were recorded with the smartphone mounted at ~160 cm; the daytime set (11:00, Maguwoharjo Stadium, Yogyakarta) and nighttime set (19:00, Malioboro Street, Yogyakarta) were decomposed into 135 frames and manually annotated in PASCAL VOC format using LabelImg. (ii) External dataset: the 12k Face Mask Dataset from Kaggle (URL: https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset).

Table 1. Model Size and Parameter

Datasets	Training		Testing	
	with_mask	no_mask	with_mask	no_mask
Proposed Dataset	888	2151	67	148
12k-Face Mask Dataset	4252	5000	483	509

The custom dataset we created and collected as shown in Figure 2 (a) has lower resolution and quality than the 12k face mask dataset as shown in Figure 2 (b), which has much better resolution and quality to test the system's reliability in detecting image conditions that differ significantly in quality.

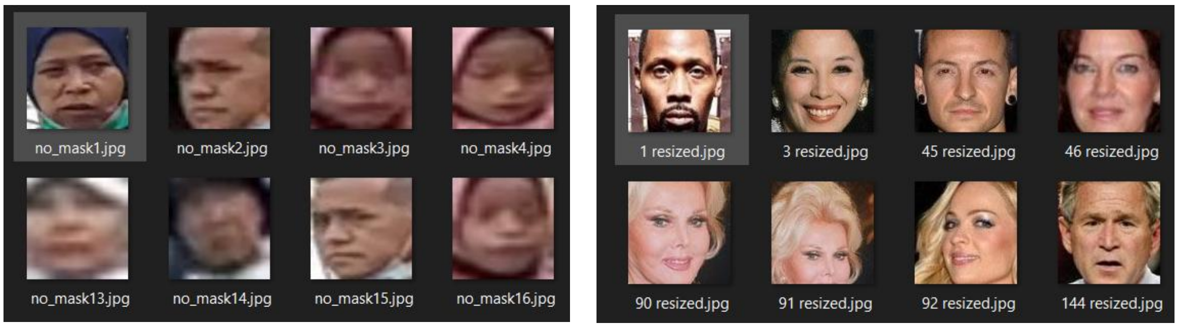


Figure 2. (a) Proposed Dataset ; (b) 12k – Face Mask Dataset

2.2. Face Detection using MTCNN

We employ a pre-trained Multi-task Cascaded Convolutional Network (MTCNN) for real-time face detection. MTCNN processes an image pyramid with a three-stage cascade—P-Net, R-Net, and O-Net. P-Net proposes candidate windows and coarse bounding-box offsets; non-maximum suppression (NMS) prunes overlaps while preserving high-recall hypotheses. R-Net filters false positives and refines localization (with NMS applied again). O-Net outputs the final face boxes together with five facial landmarks (eye centers, nose tip, mouth corners). Using these landmarks, we apply a similaritytransform alignment (rotation/scale correction) to canonicalize facial geometry, reducing pose and scale variation and improving downstream mask classification [8][9] .

2.3. Feature extraction using Bag of Visual Words (BOVW)

The Bag of Visual Words (BoVW) represents images as clusters of visual words, enhancing robustness to lighting, scale, and rotation. It uses vector quantization (VQ) with K-means clustering to group keypoints into structured vocabularies [31][32], while a codebook aids in feature matching for classificatio. BoVW follows five steps: (a) feature extraction, (b) quantization, (c) encoding, (d) histogram representation, and (e) classification. ORB (Oriented FAST and Rotated BRIEF) [33] extracts key local features for processing.

After feature extraction, the feature quantization phase clusters these descriptors into K groups using an unsupervised learning algorithm such as K-Means clustering. Each cluster center represents a visual word, forming a dictionary V of size K . Given a set of feature descriptors $\mathrm{F} = \{\mathrm{f}_1, \mathrm{f}_2, \ldots, \mathrm{F}_N\}$ from all training images, the visual vocabulary is defined as: $\mathrm{V} = \{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_K\}$ where v_j represents the j-th cluster center (visual word) obtained from K-Means clustering. Each feature descriptor is then assigned to the nearest cluster center by minimizing the Euclidean distance (eq.1).

$$\mathrm{c}_i = \arg\min_j \|\mathrm{f}_i - \mathrm{v}_j\| \tag{1}$$

where c_i the index of the closest visual word to v_j to f_i .

Following feature quantization, the feature encoding phase maps all extracted feature descriptors to their nearest visual words. For an image with N extracted features, its histogram representation is written as $\mathrm{H} = [h_1, h_2, \ldots, h_\mathrm{K}]$, where h_j represents the number of feature descriptors assigned to visual word v_j . This histogram serves as the final feature descriptor for classification. A normalized histogram representation accounts for varying numbers of key points across different images to improve interpretability. The normalized frequency of each visual word in an image is computed as eq.2.

$$h_j = \frac{1}{N} \sum_{i-1}^{N} 1(\mathrm{c}_i = j) \tag{2}$$

where : h_j is the normalized frequency of the visual word v_j, N is the total number of descriptors in the image, $1\ (C_i = j)$ is an indicator function that equals one if the feature belongs to a cluster. Otherwise, 0. Thus, the final BoVW descriptor becomes a histogram as in eq.3.

$$H = \left[\frac{h_1}{N}, \frac{h_2}{N}, \ldots, \frac{h_k}{N}\right] \tag{3}$$

Equation 3 is a fixed-length feature vector for classification once images are represented as BoVW histograms. Each image in the training dataset is represented by a histogram codeword, generated by detecting keypoints and matching them with a BoVW codebook [34]. This unique representation helps classify images based on visual word distributions. The extracted features serve as input for SVM and MLP classifiers, which learn decision boundaries to distinguish object categories.

2.4. Facemask Classification with SVM classifiers

The fundamental idea behind SVM is to find the optimal hyperplane that best separates data points of different classes with the maximum margin, ensuring better generalization to unseen data. In a binary classification problem, given a dataset of labeled training examples (x_i, y_i), where x_i represents a feature vector, and yi ∈ $\{-1,1\}$ denotes the class labels. SVM aims to construct a decision boundary that maximizes the margin between the two classes. The decision boundary is defined as a hyperplane: $\omega^T x + b = 0$, where ω is the weight vector (normal to the hyperplane), x is the input feature vector, and b is the bias term. The margin is the distance between the hyperplane and each class's nearest data points (support vectors) [11].

The optimization objective of SVM is to maximize this margin while ensuring correct classification, which is mathematically formulated as eq. 4.

$$\min_{\omega,b}\frac{1}{2}\|\omega\|^2 \text{ subject to the constraint : } y_i(\omega^T x_i + b) \geq 1, \forall i \tag{4}$$

where the constraint ensures that all correctly classified points lie on or beyond their respective margins. When data is not linearly separable, SVM introduces a soft margin approach by incorporating slack variables ξi to allow for some misclassification as eq. 5.

$$\min_{\omega,b,\xi}\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\xi_i \text{ subject to: } y_i(\omega^T x_i + b) \geq 1 - \xi_i\,,\ \xi_i \geq 0 \tag{5}$$

where C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing classification errors. For non-linearly separable data, SVM employs the kernel trick, which maps input data into a higher-dimensional space where a linear separator can be applied. The kernel functions shown in eq. 6 – 9.

Linear $K(x, x_i) = x.x_i$; (6)
Polynomial $K(x, x_i) = [\gamma * (x.x_i) + coef]^a$; (7)
Radial Basis Function (RBF) $K(x, x_i) = exp(-\gamma * \|x - x_i\|^2)$; (8)
Sigmoid: $K(x, x_i) = \tanh(\gamma(x.x_i) + coef)$. (9)

These kernels enable SVM to efficiently classify complex datasets by projecting them into higher-dimensional spaces without explicitly computing the transformation, thus significantly reducing computational cost.

2.5. Facemask Classification with MLP classifiers

A Multi-Layer Perceptron (MLP)[35] is a feedforward neural network composed of an input layer, one or more hidden layers, and an output layer. Each neuron processes a weighted sum of inputs followed by a non-linear activation function, enabling the model to learn complex patterns. The forward propagation process can be expressed as eq. 10.

$$\begin{gathered} z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)} \\ a^{(l)} = \sigma(z^{(l)}) \end{gathered} \tag{10}$$

where $W^{(l)}$ and $b^{(l)}$ are the weights and biases, $a^{(l-1)}$ is the previous layer activation, and $\sigma(.)$ is the activation function (e.g., ReLU, Sigmoid, or Tanh). The process progresses through multiple layers to the output, where Softmax is used for classification and Linear for regression. MLP training relies on optimization algorithms that adjust weights and biases based on the loss function. Adam (Adaptive Moment Estimation), an effective optimizer, combines SGD's efficiency with momentum and adaptive learning rates for improved performance. Adam optimization updates parameters using eq. 11.

$$\begin{gathered} m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2){g_t}^2 \\ \widehat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\ \theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}}\widehat{m}_t \end{gathered} \tag{11}$$

Where g_t is the gradient of the loss function, m_t and v_t represent the first moment (mean of gradients) and the second moment (uncentered variance of gradients). β1 and β2 are exponential decay rates (typically 0.9 and 0.999), and α is the learning rate. ϵ (epsilon) is a small constant to prevent division by zero. Adam optimization ensures adaptive learning rates, fast convergence, and efficient training, outperforming SGD in handling sparse gradients. In MLP, it prevents vanishing gradients and improves performance in image recognition, NLP, and structured data. Optimal results were achieved with 100 hidden layers, batch size 100, and 274 iterations.

2.6. Facemask Classification with proposed Convolutional Neural Network (CNN)

This work dispenses with BoVW and operates directly on raw 100×100×3 face crops. The proposed network—inspired by VGG-16 but heavily simplified for binary classification—comprises three stacked convolutional blocks (each with 3×3 kernels and 100 filters) followed by max-pooling; dropout is inserted after the early and late blocks to mitigate overfitting. The convolutional stack feeds two fully connected layers (1024 units → 2 logits), yielding a mask/no-mask decision.

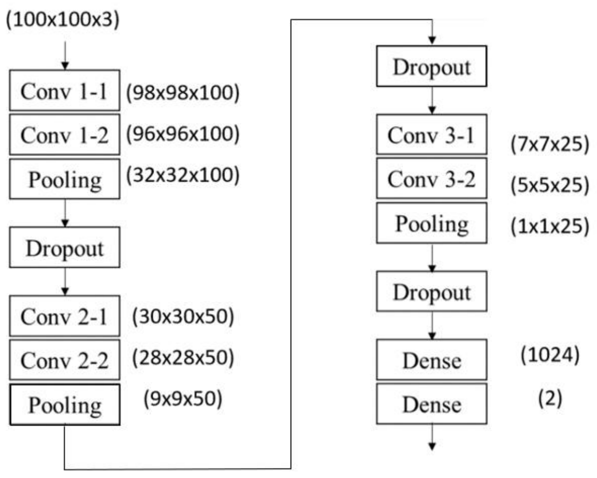


Figure. 3. The Proposed CNN Model

The proposed architecture is inspired by VGG-16 [36] but employs significant simplifications tailored for efficiency and effectiveness in binary classification tasks. It uses stacked convolutional layers to progressively extract hierarchical visual features, enhancing the network's capacity to learn complex patterns effectively. As illustrated in Figure 3, this reduced-depth design minimizes parameters and memory while preserving representational capacity via stacked 3×3 kernels (VGG-style receptive field with far fewer weights). The input size (100×100) and filter count (100/layer) were selected via fine-tuning to balance accuracy and compute, enabling faster training and deployment on low-spec edge hardware without pretraining. Compared to VGG-16, this model is more compact, consisting of fewer convolutional blocks, thus reducing computational complexity, accelerating training, and minimizing overfitting risks.

3. RESULT

In the test pipeline (Figure 2). Faces are localized and landmark-aligned using MTCNN, then cropped and resized to 100×100 for both datasets. In Figure 2(b) (BoVW branch), ORB descriptors from the aligned crop are vector-quantized with K-Means (k=100) and pooled into term-frequency histograms that feed SVM/MLP. In Figure 2(c), the same normalized 100×100 crop is used as raw input to the lightweight CNN.

3.1. Face Detection Model using MTCNN

Detection performance is assessed using the object detection metrics, which compute TP, FP, and AP per class under the PASCAL IoU protocol (threshold = 0.5). A True Positive (TP) is defined as a correct detection with IoU ≥ 0.5, a False Positive (FP) as an incorrect detection with IoU < 0.5, and a

False Negative (FN) when a ground-truth face is missed. True Negatives (TN) are not considered, since background regions correctly left undetected are not informative in dense detection contexts. Detection performance is assessed using the object detection metrics, which compute TP, FP, and AP for each class under the IoU protocol (threshold = 0.5).

Table 2. Evaluation results of the MTCNN face detector

Metric	Daytime	Nighttime
Detected Faces	1,151	730
True Positive	958 (83.23%)	584 (80.00%)
True Negative	0	0
False Positive	166 (14.42%)	107 (14.65%)
False Negative	27 (2.35%)	39 (5.35%)
Accuracy	83.23%	80.00%

As presented in Table 2, the MTCNN detector demonstrates a recall-oriented behavior with stable precision. Performance degrades under nighttime conditions due to higher false negatives, confirming sensitivity to low illumination rather than over-detection. False positives remain non-trivial across both settings, reflecting a liberal proposal policy intended to maximize coverage and ensure sufficient face crops for downstream classification. Within this evaluation framework, precision, recall, and F1-score provide more meaningful indicators than raw accuracy. For practical deployment on edge devices, precision can be improved with minimal recall loss by modestly raising stage thresholds, tightening NMS, enforcing temporal persistence across frames, applying lightweight low-light normalization, and performing hard-negative mining using representative background distractors.

3.2. Real time Face Mask Detection Result using Lightweight CNN

Table 3. Face Mask Detection Systems

	Daytime	*Nightime*
no_mask AP	89,61%	82,92%
with_mask AP	77,07%	64,09%
MAP	83,33%	73,5%
Ground Truth no_mask	443	348
Ground Truth with_mask	681	382
Detected no_mask	553	418
Detected with_mask	598	277
Ground Truth Face	1124	730
Detected Face	1151	695

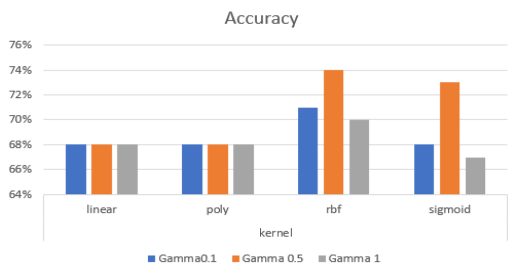
As shown in Table 3, the system achieves higher mAP under daytime conditions compared to nighttime, reflecting the impact of low illumination and sensor noise, which suggests that improved camera hardware would enhance performance in dark environments. Across both scenarios, the no_mask class consistently attains higher average precision than the with_mask class. This disparity arises because masks occlude critical facial regions, leaving fewer visible features and thereby limiting the effectiveness of face detection and subsequent classification.

4. DISCUSSION

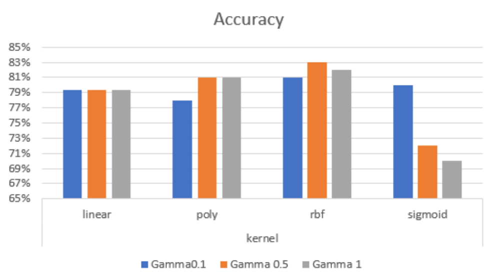
In the training Stage datas are split 80/20 for training/testing. When multiple faces appear, the highest-confidence (or largest) detection is retained; no-face frames invoke a fallback routine. All models undergo hyperparameter tuning, with fine-tuning applied to BOVW-SVM, BOVW-MLP and CNN. Performance is reported on the held-out set using accuracy, precision, recall, and F1-score.

4.1. Quantitative Analysis for facemask classification model

Figures 4 (a) and 4 (b) show that SVM performed best with the RBF kernel (Gamma = 0.5) for both datasets,and Figure 5 shows data visualization in each kernel. From both figure we can conclude that RBF effectively handles non-linearly separable data. In contrast, CNN automatically learns hierarchical features, making it more adaptable to mask types, lighting variations, and facial orientations, leading to higher classification accuracy and robustness. Meanwhile, BoVW-SVM and BoVW-MLP rely on handcrafted features, limiting their ability to handle non-uniform masks, varied angles, and background clutter, making them less suitable for real-world conditions.



(a)



(b)

Figure.4. SVM Accuracy Performance in each kernel with different Gamma on (a)Proposed Dataset; (b) Face Mask Dataset

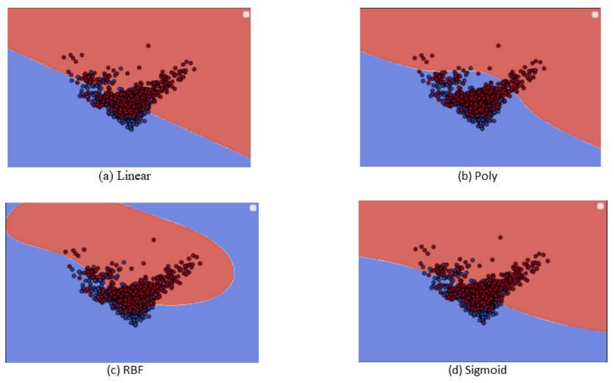


Figure. 5. SVM data visualization on each kernel

Using the Proposed Dataset and the 12k-Face Mask Dataset from Kaggle, dataset quality and diversity played a crucial role in model performance. CNN performed well on both datasets, with slightly better results on the 12k dataset, likely due to its higher resolution and balanced samples. While SVM's accuracy improved with a larger dataset, its AUC remained low, indicating weak feature discrimination. MLP struggled with the increased complexity of the 12k dataset, suggesting poor adaptability to high-dimensional variations. This highlights the importance of high-quality and balanced datasets for reliable face mask detection, as models trained on low-resolution or unbalanced data may fail in real-world scenarios with variable lighting, angles, and occlusions.

Table 4 shown the CNN model consistently achieved the highest accuracy across both datasets, showcasing deep learning's advantage in feature representation. The accuracy and F1-score gap between CNN and BoVW-based models further supports this. CNN's high recall (0.935–0.982) minimizes false negatives, crucial for public health safety. This demonstrates CNN's effectiveness over BoVW methods, which fail to generalize under lighting variations, angles, and occlusions due to their reliance on handcrafted features.

The experimental results demonstrate the comparative performance of three different classifiers—SVM (RBF kernel), MLP, and a modified CNN—for face mask detection using the Proposed Dataset and the 12k-Face Mask Dataset. The performance metrics, including accuracy, precision, recall, F1-score, and AUC, provide valuable insights into each model's strengths and weaknesses. The F1-score remains high (94.90% and 97.60%, respectively), indicating a strong balance between precision and recall. This result suggests that CNN is highly effective in learning feature representations for distinguishing between masked and unmasked faces.

Table 4. Different model performance

Model		Proposed Dataset	12k-Face Mask Dataset
BoVW- SVM (RBF kernel)	Accuracy	74.41 %	82.96 %
	Precision	73.80 %	82.30 %
	Recall	97.28 %	85.00 %
	F1-Score	83.90 %	83.60 %
BoVW-MLP	Accuracy	71.62 %	51.97 %
	Precision	98.60 %	59.65 %
	Recall	71.20 %	52.79 %
	F1-Score	82.70 %	56.01 %
CNN (without BOVW)	Accuracy	**94.67 %**	**97.40 %**
	Precision	96.30 %	97.10 %
	Recall	93.50 %	98.20 %
	F1-Score	**94.90 %**	**97.60 %**

The SVM model benefits from the larger 12k dataset, improving its accuracy from 74.41% (Proposed Dataset) to 82.96%. However, the F1-score remains almost the same (~0.839 and 0.836), suggesting that while the dataset size improves classification accuracy, it does not significantly enhance its overall ability to balance false positives and false negatives.

The MLP classifier exhibits a notable drop in performance when applied to the 12k dataset. Its accuracy decreases from 71.62% (Proposed Dataset) to 51.97% (12k-Face Mask Dataset), and the F1-score declines from 0.827 to 0.5601. This result suggests that MLP struggles with the increased complexity and diversity of the larger dataset, potentially due to limitations in its feature extraction capability compared to CNN.

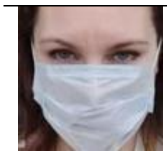
The AUC values in Table 5 highlight each model's ability to distinguish masked and unmasked faces. CNN achieves the highest AUC scores (0.97 for the Proposed Dataset, 0.98 for the 12k-Face Mask Dataset), reinforcing its strong classification capability, while SVM and MLP (AUC < 0.60) struggle to distinguish masked from unmasked faces. This makes CNN the most reliable model for mask detection. SVM shows the lowest AUC scores (0.46 and 0.51), performing only slightly better than random guessing, aligning with its lower accuracy and F1-score. MLP performs slightly better (0.50 and 0.59), but remains significantly weaker than CNN.

Table 5. AUC Score

Model	Proposed Dataset	12k-Face Mask Dataset
SVM (RBF kernel)	0.46	0.51
MLP	0.50	0.59
CNN (without BOVW)	0.97	0.98

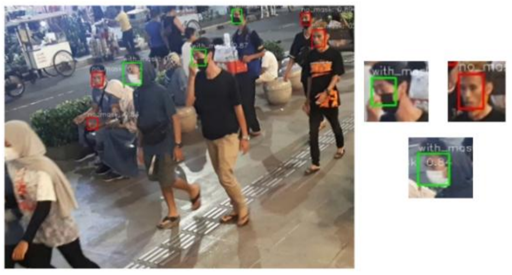
The large performance gap between CNN and BoVW-based SVM/MLP reinforces CNN’s hierarchical learning enables better generalization across datasets, whereas SVM and MLP struggle with handcrafted features, limiting their effectiveness. The sample outputs in Table 6 (Image Testing) are consistent with these findings: the three models agree on clear cases, but CNN maintains stronger performance on challenging instances. Overall, the results substantiate that learned deep features (CNN) are more reliable than BoVW-SVM/MLP in real-world conditions with variable lighting, viewing angles, occlusions, and non-uniform masks.

Table 6. Image Testing

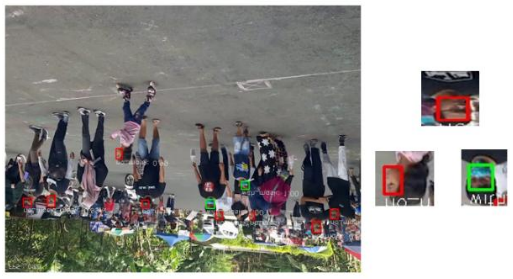
Image	Model	Result (class)
	SVM	No_mask
	MLP	No_mask
	CNN	No_mask
	SVM	With_mask
	MLP	With_mask
	CNN	With_mask
	SVM	No_mask
	MLP	No_mask
	CNN	No_mask

4.2. Qualitative Analysis

The system reliably localizes many faces in crowded scenes under both night scene and day scene conditions. Predicted classes are shown as green boxes for with_mask and red boxes for no_mask. At night, reduced illumination and motion blur lead to more unstable detections, whereas daylight improves box sharpness and score consistency. In dense crowds, overlaps between faces can cause proposals to be suppressed by NMS or missed entirely, especially for very small or oblique faces near the frame boundaries as shown in figure 6.



(a)



(b)

Figure 6. Sample of face mask detection result on (a) night scene (b) day scene

Figure 7 highlights the principal failure modes: false positives, where background textures, signage, or object edges are mistaken for faces; false negatives, when faces are not detected due to occlusions (hands, scarves/hoods), small scale, or low contrast; and misclassification, where masked faces are labeled no_mask (or vice versa) in borderline cases—e.g., dark masks blending with hair/headscarves, partially exposed noses, or strong side poses. These patterns indicate a recall-oriented detector that occasionally over-proposes boxes, which is beneficial for coverage but can propagate spurious crops to the downstream classifier.

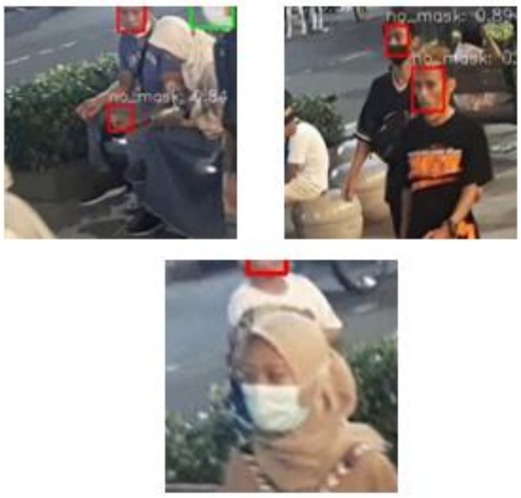


Figure 7. Failure Detection and classification of facemask

For evaluation and error auditing, each frame logs the predicted class, confidence score, and bounding-box coordinates (Xmin, Ymin, Xmax, Ymax). Logs are saved as text files indexed by frame order, enabling reproducible computation of aggregate metric and straightforward re-inspection of borderline cases. Practically, precision can be improved—while preserving recall—by modestly raising stage thresholds and tightening NMS, enforcing temporal consistency/tracking to filter transient false positives, applying lightweight low-light and occlusion augmentations, and performing hard-negative mining on frequent background distractors. Introducing an explicit “incorrectly worn mask” class can also reduce ambiguity in cases with partially exposed nose/mouth.

4.3. Computational and Applicability

We implemented a trained CNN model with the proposed dataset and the MTCNN for the face detection system. We tested the detection system on a PC and edge device equipped with a Coral USB Accelerator. The system is based on Web Applications using Flask Framework. When a person is detected not using the mask, an alarm will alert the security for further action, and the person's face is captured and then displayed on the website. The result shows on Table 7 indicate the system can run in acceptable fps but still need an improvement, albeit with some computational constraints on lower-powered edge devices.

As shown in Table 7, recent studies adopt a range of lightweight deep learning strategies to balance accuracy and computational efficiency on edge devices. The approach in [26] achieves very high accuracy (>98%) with a compact CNN, sustaining real-time throughput (~28 FPS) on CPU-based systems, while [28] leverages MobileNetV2 within an SSD framework to reach over 92% accuracy but does not explicitly quantify device-level runtime. Mobile implementations such as [29] demonstrate strong validation accuracy on Android smartphones, highlighting feasibility for consumer-grade hardware, though without detailed latency reports. Similarly, [30] emphasizes compression strategies

Vol. 7, No. 1, February 2026, Page. 210-225
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.4461

(pruning and quantization) to reduce model size, yet lacks concrete runtime evaluation on embedded platforms. Beyond pure mask detection, works like [6] extend functionality to include social distancing, reporting mean average precision around 91% at real-time speeds on Raspberry Pi 4B.

In comparison, this work integrates MTCNN with a lightweight CNN to deliver robust detection and classification across diverse datasets. The proposed CNN surpasses BoVW baselines with F1-scores up to 97.6% and maintains stable detection accuracy (mAP 83.3% daytime, 73.5% nighttime). Importantly, unlike most prior studies, this work provides explicit throughput benchmarks—21 FPS on CPU and 15 FPS on Raspberry Pi 4 with Coral USB—demonstrating practical real-time deployment under constrained hardware. These results highlight the effectiveness of architectural simplification and face-localization integration in achieving both high recognition performance and operational feasibility on edge devices.

Table 7. Comparison of face mask detection method (performance, runtime, device)

Ref.	Method (brief)	Performance (Acc / P / R / F1 / mAP)	Runtime evidence	Edge device(s)
[6]	SSD + MobileNetV2 (mask + distancing)	Prec/Rec ≈ 0.91/0.91	~28 FPS	Raspberry Pi 4 Model-B
[26]	Lightweight CNN (crop→classify)	Acc 98.47%; Mask F1 0.99; No-mask F1 0.98	~28 FPS (CPU)	Intel Core i3 PC (eval); Jetson Nano (target)
[28]	SSDMNV2 (SSD + MobileNetV2)	Acc 92.64%, F1 0.93	Real-time stated; FPS not reported	Jetson Nano / Raspberry Pi (target)
[29]	Mobile masked-face recognition (MobileNet, TF-Lite)	Val Acc 90.40% (lite); up to 99.89% on larger set	On-device real-time claim; FPS not reported	Android smartphone (on-device)
[30]	Compact CNN variants (classification)	Acc up to 98.44%	Training time only; inference FPS not reported	Recommends GPU/TPU edge/IoT (no device test)
Proposed method	MTCNN → lightweight CNN, Flask web app	Acc/P/R/F1: 94.67/96.30/93.50/94.90% (proposed set); 97.40/97.10/98.20/97.60% (12k-Face). Detector — mAP: 83.33% (day), 73.50% (night)	21 FPS (Intel i7-8750H CPU); 15 FPS (RPi 4B + Coral)	PC (CPU-only); Raspberry Pi 4B + Coral (eval)

5. CONCLUSION

This study proposes a modified CNN architecture for real-time face mask detection, outperforming BoVW-SVM and BoVW-MLP in accuracy, F1-score, and AUC. Despite its simplified design, CNN achieves AUC improvements of 49% (proposed dataset) and 43% (12k-Face Mask Dataset), effectively distinguishing masked and unmasked faces across various conditions. It performs well in both low- and high-resolution images, ensuring robustness in real-world scenarios. However, the current implementation reaches only 15 FPS on edge device, indicating that it is not yet fully real-time, and further optimization is required. Limitations also include computational constraints, sensitivity to occlusions, reliance on dataset quality, and and inability to detect improperly worn masks.

Future research should address this work by incorporating improper mask-wearning detection using pose estimation and facial landmarks analysis. Additional direction include further optimization for ultra-low-power devices, expanding dataset diversity to cover broader environmental conditions, and integrating the system into comprehensive, AI-driven public safety platform. Overall, the proposed lightweight CNN-based solution demonstrates practically, scalability, and reliability, while providing tangible contribution to the development of efficient, deep learning-based surveillance systems optimized for deployment on resource-constrained edge devices. The study hightlight how architectural optimization can achieve trade-off between computational efficiency and classification accuracy, visual recognition systems within the filed of Informatics and Computer Science.

CONFLICT OF INTEREST

The authors declares that there is no conflict of interest between the authors or with research object in this paper.

ACKNOWLEDGEMENT

This work was supported by the Universitas Gadjah Mada (UGM) Research Capacity Enhancement Grant for Young Faculty (*Hibah Peningkatan Kapasitas Peneliti Dosen Muda*), No. 2258/UN1.P.III/DIT-LIT/PT/2021.

REFERENCES

[1] A. Sharma, R. Gautam, and J. Singh, “Deep learning for face mask detection: a survey,” *Multimed. Tools Appl.*, vol. 82, no. 22, pp. 34321–34361, Sep. 2023, doi: 10.1007/s11042-023-14686-6.
[2] A. Kumar, A. Kalia, A. Sharma, and M. Kaushal, “A hybrid tiny YOLO v4-SPP module based improved face mask detection vision system,” *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 6, pp. 6783–6796, Jun. 2023, doi: 10.1007/s12652-021-03541-x.
[3] S. Singh, U. Ahuja, M. Kumar, K. Kumar, and M. Sachdeva, “Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment,” *Multimed. Tools Appl.*, vol. 80, no. 13, pp. 19753–19768, May 2021, doi: 10.1007/s11042-021-10711-8.
[4] B. Wang, Y. Zhao, and C. L. P. Chen, “Hybrid Transfer Learning and Broad Learning System for Wearing Mask Detection in the COVID-19 Era,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021, doi: 10.1109/TIM.2021.3069844.
[5] M. H. M. Kamil, N. Zaini, L. Mazalan, and A. H. Ahamad, “Online attendance system based on facial recognition with face mask detection,” *Multimed. Tools Appl.*, vol. 82, no. 22, pp. 34437–34457, Sep. 2023, doi: 10.1007/s11042-023-14842-y.
[6] S. Yadav, “Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 7, pp. 1368–1375, Jul. 2020, doi: 10.22214/ijraset.2020.30560.
[7] A. H. Alyousef, “Implementing Face Detector using Viola-Jones Method,” *SSRG Int. J. Electr. Electron. Eng.*, vol. 10, no. 7, 2023, doi: 10.14445/23488379/IJEEE-V10I7P113.
[8] C. Zhang and Z. Zhang, “Improving multiview face detection with multi-task deep convolutional neural networks,” in *IEEE Winter Conference on Applications of Computer Vision*, Mar. 2014, pp. 1036–1041. doi: 10.1109/WACV.2014.6835990.
[9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.
[10] B. Qin and D. Li, “Identifying Facemask-Wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19,” *Sensors*, vol. 20, no. 18, p. 5236, Sep. 2020, doi: 10.3390/s20185236.
[11] I. Candradewi, B. Nurcahyo Prastowo, and D. Lathief, “GENDER CLASSIFICATION FROM FACIAL IMAGES USING SUPPORT,” *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 10, pp. 2684–

2692, May 2019, doi: https://www.jatit.org/volumes/Vol97No10/4Vol97No10.pdf.

[12] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, "Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition," in *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019*, May 2019, pp. 1–5. doi: 10.1109/ICASERT.2019.8934543.

[13] S. Li *et al.*, "Multi-angle Head Pose Classification when Wearing the Mask for Face Recognition under the COVID-19 Coronavirus Epidemic," in *2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*, May 2020, pp. 1–5. doi: 10.1109/HPBDIS49115.2020.9130585.

[14] G. Deore, R. Bodhula, V. Udpikar, and V. More, "Study of masked face detection approach in video analytics," in *2016 Conference on Advances in Signal Processing (CASP)*, Jun. 2016, pp. 196–200. doi: 10.1109/CASP.2016.7746164.

[15] M. S. Ejaz and M. R. Islam, "Masked Face Recognition Using Convolutional Neural Network," in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, Dec. 2019, vol. 0, pp. 1–6. doi: 10.1109/STI47673.2019.9068044.

[16] C. V. Bhargavi, G. Mani, N. Cherukuri, C. Prasad, A. Krishna, and C. Z. Basha, "A Novel Framework for Facemask Detection Using R-Convolution Neural Network," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, Sep. 2021, pp. 958–962. doi: 10.1109/ICIRCA51532.2021.9544775.

[17] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting Masked Faces in the Wild with LLE-CNNs," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, vol. 2017-Janua, pp. 426–434. doi: 10.1109/CVPR.2017.53.

[18] N. Ullah, A. Javed, M. Ali Ghazanfar, A. Alsufyani, and S. Bourouis, "A novel DeepMaskNet model for face mask detection and masked facial recognition," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9905–9914, Nov. 2022, doi: 10.1016/j.jksuci.2021.12.017.

[19] Q. Mudassar Ilyas and M. Ahmad, "An Enhanced Deep Learning Model for Automatic Face Mask Detection," *Intell. Autom. Soft Comput.*, vol. 31, no. 1, pp. 241–254, 2022, doi: 10.32604/iasc.2022.018042.

[20] A. Kumar, A. Kalia, K. Verma, A. Sharma, and M. Kaushal, "Scaling up face masks detection with YOLO on a novel dataset," *Optik (Stuttg).*, vol. 239, p. 166744, Aug. 2021, doi: 10.1016/j.ijleo.2021.166744.

[21] C. Vimal and N. Shirivastava, "Face and Face-mask Detection System using VGG-16 Architecture based on Convolutional Neural Network," *Int. J. Comput. Appl.*, vol. 183, no. 50, pp. 16–21, Feb. 2022, doi: 10.5120/ijca2022921700.

[22] M. Mobaraki *et al.*, "Masked Face Recognition Using Convolutional Neural Networks and Similarity Analysis," in *2023 24th International Conference on Digital Signal Processing (DSP)*, Jun. 2023, vol. 2023-June, pp. 1–5. doi: 10.1109/DSP58604.2023.10167977.

[23] 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.

[24] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. Da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electron.*, vol. 10, no. 3, pp. 1–28, Jan. 2021, doi: 10.3390/electronics10030279.

[25] T.-H. Tsai, J.-X. Lu, X.-Y. Chou, and C.-Y. Wang, "Joint Masked Face Recognition and Temperature Measurement System Using Convolutional Neural Networks," *Sensors*, vol. 23, no. 6, p. 2901, Mar. 2023, doi: 10.3390/s23062901.

[26] H. Farman, T. Khan, Z. Khan, S. Habib, M. Islam, and A. Ammar, "Real-Time Face Mask Detection to Ensure COVID-19 Precautionary Measures in the Developing Countries," *Appl. Sci.*, vol. 12, no. 8, p. 3879, Apr. 2022, doi: 10.3390/app12083879.

[27] A. Kanavos, O. Papadimitriou, K. Al-Hussaeni, M. Maragoudakis, and I. Karamitsos, "Real-Time Detection of Face Mask Usage Using Convolutional Neural Networks," *Computers*, vol. 13, no. 7, p. 182, Jul. 2024, doi: 10.3390/computers13070182.

[28] 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,"

Vol. 7, No. 1, February 2026, Page. 210-225
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.4461

Sustain. Cities Soc., vol. 66, p. 102692, Mar. 2021, doi: 10.1016/j.scs.2020.102692.
[29] B. Kocacinar, B. Tas, F. P. Akbulut, C. Catal, and D. Mishra, “A Real-Time CNN-Based Lightweight Mobile Masked Face Recognition System,” *IEEE Access*, vol. 10, pp. 63496–63507, 2022, doi: 10.1109/ACCESS.2022.3182055.
[30] X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, “Real-Time Face Mask Detection Method Based on YOLOv3,” *Electronics*, vol. 10, no. 7, p. 837, Apr. 2021, doi: 10.3390/electronics10070837.
[31] M. Aly, M. Munich, and P. Perona, “BAG OF WORDS FOR LARGE SCALE OBJECT RECOGNITION - Properties and Benchmark,” in *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2011, pp. 299–306. doi: 10.5220/0003311402990306.
[32] K. Budiarta, D. M. Wiharta, and K. O. Saputra, “Balinese Mask Characters Classification using Bag of Visual Words Model,” *J. Electr. Electron. Informatics*, vol. 5, no. 1, p. 25, Feb. 2021, doi: 10.24843/jeei.2021.v05.i01.p05.
[33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544.
[34] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, “Evaluating bag-of-visual-words representations in scene classification,” in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, Sep. 2007, pp. 197–206. doi: 10.1145/1290082.1290111.
[35] A. N. Liyantoko, I. Candradewi, and A. Harjoko, “Klasifikasi Sel Darah Putih dan Sel Limfoblas Menggunakan Metode Multilayer Perceptron Backpropagation,” *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 9, no. 2, p. 173, Oct. 2019, doi: 10.22146/ijeis.49943.
[36] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” *Proc. - 3rd IAPR Asian Conf. Pattern Recognition, ACPR 2015*, pp. 730–734, 2016, doi: 10.1109/ACPR.2015.7486599.