International Journal on Robotics, Automation and Sciences

Scratch Train for Lightweight Models for Face Mask Detection

Kai Liang Lew[*1], Lazaroo Shane, Chean Khim Toa and Tetuko Kurniawan[*2]

***Abstract* – Automated systems for detecting face mask use in public became urgent during the COVID-19 pandemic. Most existing mask detection research fine-tunes ImageNet pre-trained backbones on relatively small datasets of masks. This approach raises concerns about model performance in situations with limited computational resources or when external pre-trained weights are not accessible. Additionally, there is a limited comparative analysis of recent lightweight architectures under consistent training conditions for mask detection tasks. This paper evaluates four state-of-the-art lightweight architectures for binary mask detection, including RepViT, ShuffleNetV2, EdgeNeXt Small, and EfficientFormer. These models were trained from scratch using identical training protocols on two datasets containing 7,553 and 11,792 RGB images, respectively. Performance was assessed using standardised metrics, including accuracy, precision, recall, and F1-score. Results showed that EdgeNeXt Small achieved the highest accuracy with 0.980 on Dataset 1. RepViT achieved the highest accuracy with 0.944 on Dataset 2. ShuffleNetV2 achieved the fastest inference time, with 0.51 milliseconds on Dataset 1 and 1.19 milliseconds on Dataset 2. It was the smallest model with 1.26 million parameters across all models. RepViT and EdgeNeXt Small achieved higher accuracy than ShuffleNetV2 but required greater computational resources. EfficientFormer underperformed across all evaluation metrics. These findings indicate that extremely lightweight CNNs can excel at mask detection when trained from scratch. The scope is limited to binary classification and workstation-level profiling. On-device measurements and multi-seed variation are not included.**

Keywords—Lightweight, Classification, COVID-19, Mask Detection, Convolutional Neural Network, Deep Learning, Transformer-CNN hybrid

I. INTRODUCTION

The COVID-19 pandemic created an urgent need for automated systems that can detect whether people are wearing protective face masks [1], [2], [3]. With mask mandates enforced across many countries, particularly in public and crowded areas, computer vision researchers have focused intensively on developing face mask detection systems to support public health monitoring [4], [5], [6]. These systems typically perform classification or detection to identify the presence of a mask [7], [8].

Most existing mask detection work fine-tunes ImageNet-pretrained backbones on relatively small mask datasets. This dependency on pretrained weights raises questions about model performance in scenarios where computational resources are constrained and external pretrained weights are unavailable [9], [10], [11]. Additionally, there is a limited comparative analysis of recent lightweight architectures under consistent training conditions for mask detection tasks.

[*1]Corresponding Author 1 email: lewkailiang@gmail.com ORCID: 0000-0002-0376-2970
[*2]Corresponding Author 2 email: tkurniaw@ippt.pan.pl ORCID: 0000-0002-6356-1134
Kai Liang Lew is with Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia (e-mail: lewkailiang@gmail.com).
Lazaroo Shane is with Product Test Engineering, Infineon Technologies, Melaka (email: ShaneLemuel.Lazaroo@infineon.com).
Chean Khim Toa is with School of Computing and Data Science, Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, 43900 Sepang, Selangor. (e-mail: cheankhim.toa@xmu.edu.my).
Tetuko Kurniawan is with Institute of Fundamental Technological Research, Polish Academy of Sciences, Pawinskiego 5B, 02-106 Warsaw, Poland (email: tkurniaw@ippt.pan.pl)

International Journal on Robotics, Automation and Sciences (2026) 8, 1:87-95
https://doi.org/10.33093/ijoras.2026.8.1.9
Manuscript received: 12 Jun 2025 | Revised: 30 Jul 2025 | Accepted: 24 Sep 2025 |
Published: : 31 Mar 2026

Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

Many face mask classifiers utilise popular models, such as ShuffleNetV2, MobileNet, EfficientFormer, InceptionV3, EdgeNeXt Small or ResNet50, by fine-tuning them on mask datasets [12]. However, this body of work lacks a systematic comparison of these architectures when trained from scratch under identical conditions.

The first objective is to evaluate the performance of four state-of-the-art (SOTA) lightweight architectures for binary mask classification when trained from scratch using identical training protocols. This enables a standardised evaluation approach.

The second objective is to analyse computational efficiency through inference time measurement and parameter analysis across diverse hardware constraints.

The main research question can be stated as follows: Which lightweight vision architecture achieves the best balance between classification accuracy and computational efficiency for mask detection when trained from scratch on limited data?

The first contribution is to provide a comprehensive comparison of EfficientFormer, ShuffleNetV2, EdgeNeXt, and RepViT architectures in the context of mask detection.

The second contribution is providing a comprehensive efficiency analysis, including inference time benchmarking and parameter footprint assessment, under identical evaluation conditions.

This study provides a standardised comparison from scratch of modern lightweight architectures under one training protocol. The contribution is practical and focuses on a fair and reproducible setup that isolates the effect of training from scratch on small binary mask datasets. Architectural novelty is not claimed.

The remainder of this paper is organised as follows. The Literature Review examines related work on face mask detection systems and lightweight architectures. The Methodology section describes the rationale for architecture selection and the standardised training protocols implemented for fair comparison. The Experiments, Results, and Discussion section outlines the dataset and evaluation metrics and presents comprehensive performance comparisons across both datasets. It also analyses the efficiency and accuracy tradeoffs among the four architectures with detailed confusion matrix analysis. Finally, the Conclusion synthesises the key findings and identifies promising directions for future research.

II. Literature Review

A. Face Mask Detection Systems

Classification approaches assume a cropped face image as input and classify it as "with mask" or "without mask." Recent studies have achieved remarkable accuracy improvements using deep learning techniques. Dewi et al. [1] implemented YOLOv8 for face mask detection, achieving SOTA performance with real-time capabilities. Their system processes both the Face Mask Dataset (FMD) and Medical Mask Dataset (MMD) with high precision. Verma et al. [13] developed an automated face mask detection system using ResNet152V2 with the Haar cascade classifier, achieving 99.65% training accuracy and 99.63% validation accuracy.

Object detection approaches aim to locate faces in images and videos and classify each as masked or unmasked. Mostafa et al. [14] proposed a YOLO-based deep learning C-Mask model for real-time face mask detection via drone surveillance, addressing three categories such as wearing a mask, incorrect wearing, and no mask. Their system demonstrated robust performance in crowded public spaces with mobile camera feeds. Multiple studies have employed Single Shot Detector (SSD) frameworks with lightweight backbones for real-time mask detection on embedded devices. These studies generally found that existing object detection models could be adapted to detect masked faces with only minor drops in accuracy compared to general face detection.

Recent advancements include the development of specialised datasets addressing bias issues. Kantarcı et al. [15] presented the Bias-Aware Face Mask Detection (BAFMD) dataset, which contains a larger number of images from underrepresented racial and age groups to mitigate bias issues in face mask detection. Suryawanshi et al. [16] introduced a comprehensive Face Mask Wearing Image Dataset with 24,916 images categorised into correct and incorrect mask-wearing across different mask types and demographics. The results showed that their models achieved high accuracy in mask and no-mask classification, even with models deployable on edge devices.

B. Lightweight Architectures

Recent studies have shown ShuffleNetV2's effectiveness in various vision tasks [17]. Ullah et al. demonstrated that ShuffleNetV2 maintains competitive performance while significantly reducing computational requirements compared to heavier architectures. MobileNetV2 and V3 are other popular architectures optimised for mobile .

Vision transformers (ViT) have gained popularity for image recognition, but vanilla ViTs are too computationally intensive [18]. This led to research on hybrid models that combine CNN and Transformer ideas. Maurício et al. [20] provided a comprehensive comparison between Vision Transformers and CNNs for face recognition tasks, evaluating models on five diverse datasets, including Labelled Faces in the Wild and VGG Face 2. MobileViT [19] embedded transformer-like attention blocks into a CNN backbone, achieving higher accuracy than MobileNet with some increase in latency. EdgeViT and EfficientViT are variants for edge devices [20], [21].

EfficientFormer [22] is a pure transformer architecture that was optimised to match MobileNet-level latency. The authors identified and removed inefficient design elements in ViTs and used latency-driven slimming to produce EfficientFormer variants that are extremely fast.

Comprehensive comparisons of recent models under consistent training conditions remain limited, even though there have been advances in lightweight

architectures. Chen et al. [23] demonstrated the potential of lightweight CNNs with their IR-Shuffle unit design, achieving 98.65% accuracy with a model size of only 1.45 MB and 5ms faster inference than MobileFaceNet. Individual studies have shown EfficientFormer, EdgeNeXt, and RepViT to be effective on general computer vision benchmarks. However, their comparative performance, specifically when trained from scratch on specialised datasets with limited computational resources, has not been systematically assessed.

III. METHODOLOGY

A. Architecture Selection and Design

This paper evaluates four SOTA lightweight architectures chosen for their potential in resource-constrained face mask detection applications. The selected models represent different design philosophies while maintaining computational efficiency.

EfficientFormer-L1 [22] serves as the transformer-based representative, implementing a pure vision transformer architecture optimised for mobile-level latency. The model employs dimension-consistent design principles and latency-driven optimisation, achieving 79.2% ImageNet top-1 accuracy. It has 12.3M parameters while maintaining transformer capabilities for global feature learning.

EdgeNeXt [24] is a hybrid model combining CNN and Transformer architectures. It is built based on ConvNeXt-style blocks and enhanced with an efficient split depth-wise transpose attention (SDTA) mechanism. This design choice results in linear computational complexity with respect to the number of patches, thereby avoiding the quadratic computational cost typically associated with standard multi-head self-attention (MHSA).

RepViT [25] is a pure CNN model that was redesigned by incorporating innovations from Transformer architectures. The concept involves integrating architectural elements from Vision Transformers into a MobileNet-style CNN and then applying reparameterisation techniques to simplify the model during inference. The model achieves approximately 78-79% ImageNet accuracy with 5.1M parameters and sub-millisecond inference times.

ShuffleNetV2 provides a baseline for lightweight CNNs, implementing channel shuffling and efficient design architecture. With approximately 1.26 million parameters, it embodies established lightweight CNN design principles and serves as a benchmark for newer architectures.

All architectures are integrated into a unified framework using a GenericClassifier wrapper that maintains consistent input and output interfaces while preserving each model's unique architectural characteristics. The classifier automatically detects and removes the original classification head from each backbone model, replacing it with a task-specific linear layer for binary classification. Feature dimensions are determined dynamically through forward pass inference with dummy inputs, ensuring compatibility across diverse architecture designs.

B. Training from Scratch Protocol

A standardised training protocol is implemented to ensure fair comparison across architectures, which removes the influence of pre-trained weights and focuses solely on each model's learning capacity within the specific domain.

To determine optimal training hyperparameters, a comprehensive ablation study was conducted testing eighteen configurations combining three learning rates, 1×10^{-5}, 1×10^{-4}, and 1×10^{-3}, two optimisers, namely Adam and AdamW, and three batch sizes, such as 16, 32, and 64. Images are resized to 224×224 pixels to maintain compatibility across all architectures while providing sufficient resolution for face mask detection. The data augmentation process helps the model generalise better by including random horizontal flipping (with a 0.5 probability), random vertical flipping (with a 0.3 probability), and random rotations of up to ±15 degrees. Additional augmentation includes ColorJitter (brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1), RandomGrayscale (p=0.1), and RandomApply with GaussianBlur (p=0.1). GaussianNoise (σ=0.1) and 10% label noise are applied on the training split only. All images are normalised using ImageNet statistics (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]) to maintain consistency with standard computer vision practices, despite training from scratch.

TABLE 1. Algorithm Table on Face Mask Classification Framework

Step	Action	Description
1	Initialize	Define all datasets, models, and hyperparameters to test. Enable resume mode to skip completed experiments.
2	Loop Through Datasets	Process two face mask datasets sequentially: Face Mask Dataset and Face Mask Detection Dataset
3	Loop Through Configurations	For each dataset, iterate through 72 combinations of:- 4 models- 3 learning rates- 2 optimisers- 3 batch sizes.
4	Run One Experiment	Execute full ML pipeline for each configuration:
	a. Prepare Data	Load images, apply augmentations (rotation, flipping, colour jittering), add noise (10% label corruption, Gaussian σ=0.1), split into 60/20/20 train/validation/test sets.
	b. Train Model	Train on the training set with early stopping based on validation performance. Monitor GPU power and memory usage.
	c. Evaluate Model	Test the final model on the held-out test set to compute accuracy, F1 Score, precision, recall, and other relevant metrics.
	d. Test Generalisation	Evaluate the trained model on other datasets to assess cross-dataset transfer capability.
	e. Log Everything	Save all metrics, plots, and model files to the MLflow tracking server.
5	Repeat & Finish	Continue through all 144 experiments (72 per dataset) and

Step	Action	Description
		generate the final statistical analysis.

Based on ablation results showing optimal average performance across all architectures, all models are trained for 100 epochs using the AdamW optimiser with an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-5}. OneCycleLR scheduling is employed, with a maximum learning rate of 2×10^{-3} with 20 times the initial learning rate, to facilitate efficient convergence. The batch size is set to 32. Early stopping with a patience of 10 epochs based on validation accuracy was implemented to prevent overfitting.

The loss function is Cross-entropy loss. Gradient norms are monitored throughout training. The training process includes automatic model checkpointing based on validation accuracy to preserve best-performing weights.

Experiments are conducted on an NVIDIA GeForce RTX 2080 Ti GPU. All experiments are tracked using MLflow with comprehensive logging of hyperparameters, metrics, and artefacts. Table 1 provides an overview of the end-to-end training and evaluation pipeline.

IV. Experiments, Results and Discussion

A. Dataset

Dataset 1 assembled by Ashish Jangra comprises 11,792 RGB images, specifically designed for mask detection [26]. The dataset originally splited into 10,000 images for training, 800 images for validation, and 992 images for the test dataset. The perfectly balanced training set and controlled validation set provide ideal conditions for comparing architectural performance without class imbalance effects. Images were collected from multiple sources, including Google images and existing datasets, to ensure diversity in the training data.

Dataset 2 compiled by Omkar Gurav contains 7,553 RGB images, divided into 3,725 "with-mask" and 3,828 "without-mask" classes [27]. The dataset provides a binary classification with a slight bias toward the without-mask class. Images are stored in two class-labelled folders and vary in resolution. Preprocessing is required to standardise input dimensions. This dataset was splited into training, validation, and test sets using a 60:20:20 ratio with a fixed random seed of 42. Table 2 shows the number of training, validation and test sets for each dataset.

TABLE 2. The number of training, validation and test sets for each dataset.

Dataset	Training Set	Validation Set	Test Set
Dataset 1	10000	800	992
Dataset 2	4531	1510	1512

B. Evaluation Metrics

Model performance evaluation utilised four evaluation metrics to assess the model's performance in classifying images with and without masks. Accuracy measures overall classification correctness by dividing correctly predicted instances by total test samples. While accuracy provides overall correctness, it can mislead the interpretation when the classes are imbalanced. Precision calculates positive prediction reliability through the ratio of true positives to all positive predictions made by the model. It avoids unnecessary intervention. Recall determines detection coverage by dividing the number of true positives by the total number of actual positive instances in the dataset. It ensures no instances are missed. The F1-score provides a balanced assessment of a model's performance by considering both precision and recall, effectively acting as their harmonic mean. It helps identify models that perform well on both metrics rather than excelling at only one. Accuracy can hide class-specific behaviour. Precision, recall and F1 score are required. Confusion matrices are included.

C. Results and Discussion

Dataset 1 Result & Discussion

The model has been trained on dataset 1. Table 3 shows the results of the model. All the models are trained with configuration with a learning rate of 0.0001, AdamW optimiser and batch size of 32.

TABLE 3. Result of the model that was trained on dataset 1

Model	Accuracy	Precision	Recall	F1-Score	Parameters (M)
ShuffleNetV2	0.978	0.978	0.978	0.978	1.26
RepViT	0.972	0.972	0.972	0.972	4.72
EdgeNeXt Small	0.980	0.980	0.980	0.980	5.28
EfficientFormer	0.923	0.923	0.923	0.923	11.39

EdgeNeXt Small achieved the highest accuracy of 0.980, followed by ShuffleNetV2 with an accuracy of 0.978 and RepViT with an accuracy of 0.972. The EfficientFormer performed the worst with an accuracy of 0.923. All models show identical values across precision, recall and f1-score. EfficientFormer required more data to train from scratch due to the transformer-style blocks.

The parameter of the model shows the tradeoff between model size and accuracy. ShuffleNetV2 has the lowest number of parameters with 1.26M, while EfficientFormer has the largest number of parameters with 11.39M. EdgeNeXt Small has slightly higher accuracy compared to ShuffleNetV2, while the parameter of EdgeNeXt Small is larger than that of ShuffleNetV2. This comparison shows that the small model remains competitive. Figure 1 shows the ShuffleNetV2 confusion matrix for dataset 1.

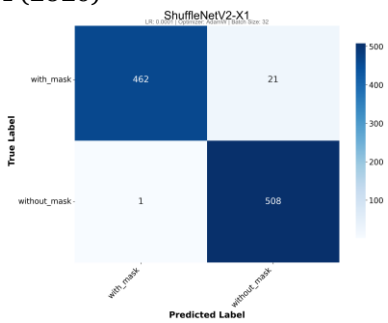

FIGURE 1. ShuffleNetV2 Confusion Matrix for dataset 1.

ShuffleNetV2 correctly classified 462 masked faces and 508 unmasked faces. It classified a total of 22 errors. Figure 2 shows the RepViT confusion matrix for dataset 1.

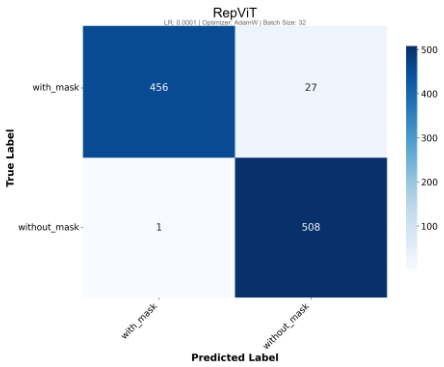

FIGURE 2. RepViT Confusion Matrix for dataset 1.

RepViT-M0.9 classified 456 true positives, 508 true negatives, 27 false negatives, and one false positive. Thus, it has a total of 28 wrong classifications. Figure 3 shows the EdgeNeXt confusion matrix for dataset 1.

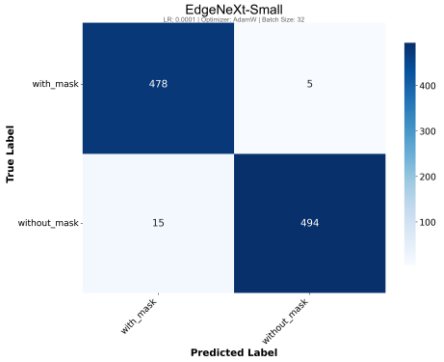

FIGURE 3. EdgeNeXt Confusion Matrix for dataset 1.

EdgeNeXt Small correctly classified 478 masked and 494 unmasked examples but made five false negatives and 15 false positives. It has a total of 20 errors. Figure 4 shows the EfficientFormer confusion matrix for dataset 1.

EfficientFormer classifies with 415 true positives, 501 true negatives, 68 false negatives, and eight false positives. It has a total of 76 misclassified classes.

Throughout all four confusion matrices, the EdgeNeXt Small has the fewest errors compared to the others, while EfficientFormer has the most misclassified classes.

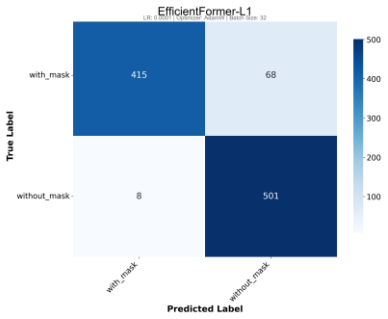

FIGURE 4. EfficientFormer Confusion Matrix for dataset 1.

The efficiency gap between ShuffleNetV2 and RepViT models highlights the significance of extreme parameter economy, particularly when working with small datasets. Based on the results, both have the potential to work with limited data and resources while maintaining good accuracy in classification.

EdgeNeXt Small ranked first in accuracy but had 20 classification errors, indicating that its hybrid attention layers and slightly larger capacity do not improve the scratch-trained performance on this dataset.

EfficientFormer has the worst accuracy, with 76 misclassifications. The amount of misclassification indicates a data hunger issue, likely due to transformer-style blocks. When trained from scratch on fewer than 8,000 images, global self-attention mechanisms do not generalise as robustly as convolutional filters. Therefore, EfficientFormer is unable to compete with RepViT and ShuffleNetV2. Table 4 shows the training and inference efficiency metrics on dataset 1.

TABLE 4. Training and Inference Efficiency Metrics on dataset 1

Model	Train Time (hours)	Energy (Wh)	Inference Time (ms)	GFLOPs	VRAM (GB)
EfficientFormer	5.53	54.32	1.28	1.32	2.38
RepViT M0.9	6.24	55.76	1.27	0.85	2.24
EdgeNeXt Small	5.64	37.45	1.26	0.97	1.85
ShuffleNetV2 X1	5.44	35.78	0.51	0.15	0.73

The ShuffleNetV2 has the fastest inference time with 0.51 ms, while the EfficientFormer has the slowest inference time with 1.28 ms. The ShuffleNetV2 is at 0.15 GFLOPs and the EfficientFormer is at 1.32 GFLOPs. The VRAM also follows the same pattern. The ShuffleNetV2 uses 0.73 GB of VRAM while the EfficientFormer uses 2.38 GB of VRAM. The training time ranges from 5.44 to 6.24 hours across all models, with RepViT as the slowest.

TABLE 5. Ablation Study on ShuffleNetV2 All Configurations on dataset 1

Learning Rate	Optimizer	Batch Size	Accuracy	Precision	Recall	F1-Score
0.00001	Adam	16	0.877	0.901	0.877	0.875
0.00001	Adam	32	0.849	0.881	0.849	0.845
0.00001	Adam	64	0.645	0.783	0.645	0.591
0.00001	AdamW	16	0.768	0.840	0.768	0.753

0.00001	AdamW	32	0.769	0.841	0.769	0.755
0.00001	AdamW	64	0.637	0.784	0.637	0.578
0.0001	AdamW	16	0.981	0.982	0.981	0.981
0.0001	AdamW	32	0.978	0.979	0.978	0.978
0.0001	AdamW	64	0.987	0.987	0.987	0.987
0.0001	Adam	16	0.971	0.972	0.971	0.971
0.0001	Adam	32	0.958	0.961	0.958	0.958
0.0001	Adam	64	0.953	0.957	0.953	0.953
0.001	AdamW	16	0.983	0.983	0.983	0.983
0.001	AdamW	32	0.968	0.968	0.968	0.968
0.001	AdamW	64	0.984	0.984	0.984	0.984
0.001	Adam	16	0.967	0.969	0.967	0.967
0.001	Adam	32	0.971	0.972	0.971	0.971
0.001	Adam	64	0.978	0.978	0.978	0.978

Energy used is the lowest for ShuffleNetV2 with 35.78 Wh, and the second lowest is the EdgeNeXt Small with 37.45 Wh. The RepViT consumes the highest energy with 55.76 Wh. All the metrics indicate that ShuffleNetV2 has the best latency and the lowest computational cost. Table 5 presents the ablation study on ShuffleNetV2 for dataset 1, covering all configurations.

The best configuration on dataset 1 uses AdamW with a learning rate of 0.0001 and a batch size of 64. ShuffleNetV2 achieved the highest accuracy with 0.987, with matching precision, recall and f1-score. The performance drops when the learning rate is 0.00001, AdamW and batch size of 64, with an accuracy of 0.637. AdamW generally outperformed Adam across the same learning rates and batch sizes.

Dataset 2 Result & Discussion

All the models have been trained on dataset 2. All models are trained with a configuration, using a learning rate of 0.0001, the AdamW optimiser, and a batch size of 32. Table 6 shows the results of the model.

TABLE 6. Result of the model that was trained on dataset 2

Model	Accuracy	Precision	Recall	F1-Score	Parameters (M)	VRAM (GB)
RepViT	0.944	0.944	0.944	0.944	4.72	2.24
EdgeNeXt Small	0.903	0.903	0.903	0.903	5.28	1.85
ShuffleNetV2	0.880	0.880	0.880	0.879	1.26	0.73
EfficientFormer	0.818	0.818	0.818	0.817	11.39	2.38

RepViT achieved the highest accuracy of 0.944, followed by EdgeNeXt Small with an accuracy of 0.903 and ShuffleNetV2 with an accuracy of 0.880. EfficientFormer performed the worst with an accuracy of 0.818. All models show identical values across precision, recall and f1-score. The parameter of the model shows the tradeoff between the model size and accuracy. Figure 5 shows the ShuffleNetV2 confusion matrix for dataset 2.

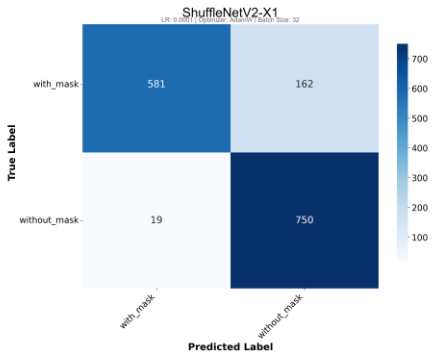

FIGURE 5. ShuffleNetV2 Confusion Matrix for dataset 2

ShuffleNetV2 correctly identified 581 images with masks and 750 images without masks. The model incorrectly classifies 181 images. Figure 6 shows the RepViT confusion matrix for dataset 2.

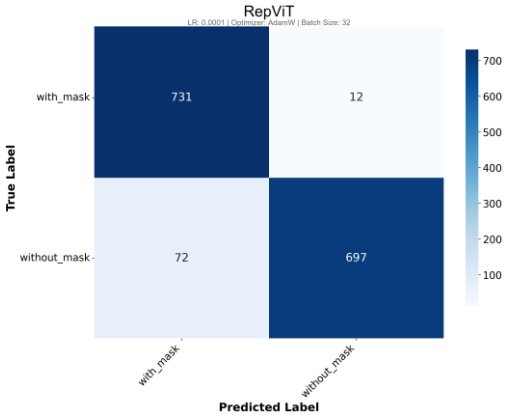

FIGURE 6. RepViT Confusion Matrix for dataset 2.

RepViT correctly classified 731 images with masks and 697 images without masks. It makes 84 incorrect predictions. Figure 7 shows the EdgeNeXt confusion matrix for dataset 2.

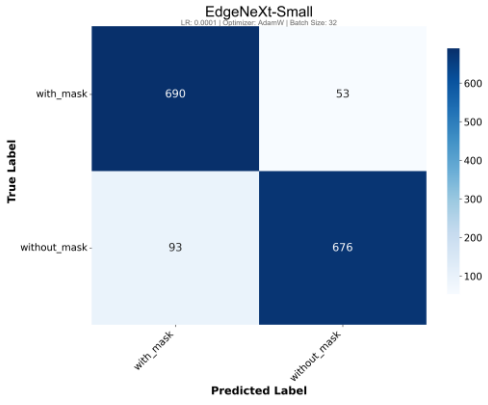

FIGURE 7. EdgeNeXt Small Confusion Matrix for dataset 2

EdgeNeXt Small correctly classified 690 images with masks and 676 images without masks. It makes 146 incorrect predictions. Figure 8 shows the EfficientFormer confusion matrix for dataset 2.

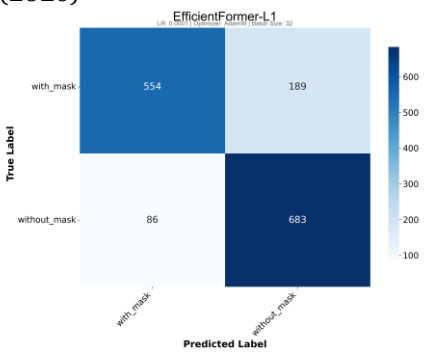

FIGURE 8. EfficientFormer Confusion Matrix for dataset 2

EfficientFormer showed the weakest performance, with 554 correct classifications with masks and 683 correct classifications without masks. This model has a total of 275 wrongly classified classes.

ShuffleNetV2 maintains competitive performance while having computational efficiency advantages. The model has a balanced error distribution, indicating unbiased classification behaviour across both classes.

EfficientFormer shows the most errors, with an equal number of mistakes in both categories. The error pattern suggests that the model had difficulty classifying the feature. These results validate the tradeoff analysis between efficiency and accuracy. Table 7 shows the training and inference efficiency metrics on dataset 2.

TABLE 7. Training and Inference Efficiency Metrics on dataset 2

Model	Training Time (hours)	Energy (Wh)	Inference Time (ms)	GFLOPs
EfficientFormer	2.43	102.4	1.33	1.32
RepViT	2.78	104.4	1.27	0.85
EdgeNeXt Small	2.49	115.9	1.32	0.97
ShuffleNetV2	2.43	43.3	1.19	0.15

The ShuffleNetV2 has the fastest inference time with 1.19 ms, while the EfficientFormer has the slowest inference time with 1.33 ms. The RepViT is at 1.27 ms and the EdgeNeXt Small is at 1.32 ms. The ShuffleNetV2 is at 0.15 GFLOPs and the EfficientFormer is at 1.32 GFLOPs. Training time is 2.43 to 2.78 across all the models. The ShuffleNetV2 and EfficientFormer have the same training time, with 2.43 hours. The longest training time is RepViT with 2.78 hours. Energy consumption is the lowest for ShuffleNetV2 with 43.3 Wh, while EdgeNeXt Small is the highest with 115.9 Wh. These metrics show that ShuffleNetV2 has the best latency and the lowest computation cost on this dataset. Table 8 presents the ablation study on ShuffleNetV2 for dataset 2, covering all configurations.

TABLE 8. Ablation Study on ShuffleNetV2 All Configurations on dataset 2

Learning Rate	Optimizer	Batch Size	Accuracy	Precision	Recall	F1-Score
0.00001	Adam	16	0.700	0.805	0.700	0.669
0.00001	Adam	32	0.838	0.867	0.838	0.834
0.00001	Adam	64	0.655	0.777	0.655	0.610
0.00001	AdamW	16	0.695	0.796	0.695	0.664
0.00001	AdamW	32	0.809	0.844	0.809	0.803
0.00001	AdamW	64	0.636	0.778	0.636	0.578
0.0001	AdamW	16	0.972	0.972	0.972	0.972
0.0001	AdamW	32	0.880	0.894	0.880	0.879
0.0001	AdamW	64	0.946	0.947	0.946	0.946
0.0001	Adam	16	0.964	0.965	0.964	0.964
0.0001	Adam	32	0.896	0.905	0.896	0.895
0.0001	Adam	64	0.951	0.951	0.951	0.951
0.001	AdamW	16	0.960	0.960	0.960	0.960
0.001	AdamW	32	0.947	0.948	0.947	0.947
0.001	AdamW	64	0.940	0.945	0.940	0.940
0.001	Adam	16	0.927	0.932	0.927	0.926
0.001	Adam	32	0.944	0.945	0.944	0.944
0.001	Adam	64	0.802	0.855	0.802	0.795

The optimal configuration for dataset 2 utilises AdamW with a learning rate of 0.0001 and a batch size of 16. ShuffleNetV2 achieved the highest accuracy with 0.972, with matching precision, recall and f1-score. The performance drops when the learning rate is 0.00001 across all the settings. The high learning rate of 0.001 is sensitive to large batch sizes. AdamW generally outperformed Adam across the same learning rates and batch sizes.

Limitation

This paper has several limitations that should be considered when interpreting its results. Results use a single seed. Statistical variation, such as mean and standard deviation, across multiple runs is not included due to computational limits. Real-time video streams and on-device profiling are not evaluated. These are planned as follow-up experiments. An augmentation ablation is not included. A fixed light transform policy is used to keep the comparison controlled.

The evaluation focused on binary classification using only two datasets with controlled imaging conditions. The experimental design also relied on single-run results without statistical significance testing. The small performance differences observed among the top-performing models would benefit from multiple experimental runs to strengthen comparative conclusions.

The computational efficiency analysis relied on a single hardware setup. Inference time, parameters, VRAM and energy are reported for this workstation. However, on-device measurements may differ. Furthermore, the paper lacked cross-dataset validation to check how well models generalise, and it did not compare with transfer learning methods to see the benefits of training from scratch.

These limitations mean that while the paper offers valuable insights for picking the right model for mask detection, the results should only be understood within the specific conditions used in this evaluation.

V. CONCLUSION

This paper provided a comprehensive evaluation of four lightweight architectures for face mask detection. This reveals that training from scratch can achieve excellent performance even with limited data. Across both datasets, the results showed the accuracy and efficiency tradeoffs rather than a single winner. On dataset 1, EdgeNeXt Small achieved the highest accuracy with 0.980. ShuffleNetV2 is the second highest with an accuracy of 0.978, followed by RepViT with an accuracy of 0.972 and EfficientFormer with an accuracy of 0.923. On dataset 2, RepViT achieved the highest accuracy with 0.944. EdgeNeXt Small is the second highest with an accuracy of 0.903, followed by ShuffleNetV2 with an accuracy of 0.880 and EfficientFormer with an accuracy of 0.818.

In terms of efficiency, ShuffleNetV2 is the leader. It processes an image with 0.51 ms in Dataset 1 and 1.19 ms in Dataset 2. It has 1.26 M parameters, which is the smallest model across all models. The results show that increased architectural complexity does not guarantee better performance in specialised tasks such as mask detection. EfficientFormer's poor performance across all metrics highlights the data requirements of transformer-based architectures when training from scratch on limited datasets.

Several research directions warrant investigation to extend these findings. Evaluating these architectures on more diverse datasets with varying lighting conditions, ethnic backgrounds, and mask types would provide insights into model robustness and generalisation capability. Exploring the impact of different data augmentation strategies and training techniques could potentially enhance performance for larger models, such as RepViT and EfficientFormer. Investigating quantisation and pruning techniques specifically for mask detection could further reduce model sizes and inference times while maintaining accuracy. Examining cross-domain transfer learning between different mask datasets could reveal whether models trained on one dataset generalise effectively to others [28].

ACKNOWLEDGMENT

We would like to thank the peer reviewers for their assistance in reviewing the paper.

FUNDING STATEMENT

There are no funding agencies supporting the research work.

AUTHOR CONTRIBUTIONS

Kai Liang Lew: Conceptualisation, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;

Lazaroo Shane: Writing – Review & Editing;

Chean Khim Toa: Writing – Review & Editing;

Tetuko Kurniawan: Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Ethical approval was not applicable to this research since it did not involve human participants, animals, or sensitive data.

REFERENCES

[1] C. Dewi, D. Manongga, Hendry, E. Mailoa and K. D. Hartomo, "Deep Learning and YOLOv8 Utilized in an Accurate Face Mask Detection System," *Big Data Cognitive Computing*, vol. 8, no. 1, p. 9, 2024. DOI: https://doi.org/10.3390/bdcc8010009

[2] R.A.S. Naseri, A. Kurnaz and H.M. Farhan, "Optimised face detector-based intelligent face mask detection model in IoT using deep learning approach," *Applied Soft Computing*, vol. 134, p. 109933, 2023. DOI: https://doi.org/10.1016/j.asoc.2022.109933

[3] B.U.H. Sheikh and A. Zafar, "RRFMDS: Rapid Real-Time Face Mask Detection System for Effective COVID-19 Monitoring," *SN Computer Science*, vol. 4, no. 3, p. 288, 2023. DOI: https://doi.org/10.1007/s42979-023-01738-9

[4] Vibhuti, N. Jindal, H. Singh and P.S. Rana, "Face mask detection in COVID-19: a strategic review," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 40013–40042, 2022. DOI: https://doi.org/10.1007/s11042-022-12999-6

[5] 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*, vol. 11, no. 2, p. 107, 2023. DOI: https://doi.org/10.3390/systems11020107

[6] 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, p. 6090, 2023. DOI: https://doi.org/10.3390/s23136090

[7] H. Wang, Y. Gu and H. Li, "Research on Face Detection and Recognition with Face Mask Based on FaceNet," *Proceedings of the 2022 4th International Conference on Robotics, Intelligent Control and Artificial Intelligence*, pp. 618–623, 2023. DOI: https://doi.org/10.1145/3584376.3584485

[8] H. Goyal, K. Sidana, C. Singh, A. Jain and S. Jindal, "A real time face mask detection system using convolutional neural network," *Multimedia Tools and Applications*, vol. 81, no. 11, pp. 14999–15015, 2022. DOI: https://doi.org/10.1007/s11042-022-12166-x

[9] A. Panda, D. Panigrahi, S. Mitra, S. Mittal and S. Rahimi, "Transfer Learning Applied to Computer Vision Problems: Survey on Current Progress, Limitations, and Opportunities," *arXiv*, 2024. DOI: https://doi.org/10.48550/arXiv.2409.07736

[10] Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan and Y. Gu, "A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations," *Expert Systems with Applications*, vol. 242, p. 122807, 2024. DOI: https://doi.org/10.1016/j.eswa.2023.122807

[11] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M.A. Azim, "Transfer learning: a friendly introduction," *Journal of Big Data*, vol. 9, no. 1, p. 102, 2022. DOI: https://doi.org/10.1186/s40537-022-00652-w

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv*, 2014. DOI: https://doi.org/10.48550/arXiv.1409.1556

[13] S. Verma, "An automated face mask detection system using transfer learning based neural network to preventing viral infection," *Expert Systems*, vol. 41, no. 3, p. e13507, 2024. DOI: https://doi.org/10.1111/exsy.13507

[14] S.A. Mostafa, "A YOLO-based deep learning model for Real-Time face mask detection via drone surveillance in public spaces," *Information Sciences*, vol. 676, p. 120865, 2024. DOI: https://doi.org/10.1016/j.ins.2024.120865

[15] A. Kantarcı, F. Ofli, M. Imran and H.K. Ekenel, "Bias-Aware Face Mask Detection Dataset," *Multimedia Tools and Applications*, 2024.

DOI: https://doi.org/10.1007/s11042-024-20226-7
[16] Y. Suryawanshi, V. Meshram, V. Meshram, K. Patil, and P. Chumchu, "Face mask wearing image dataset: A comprehensive benchmark for image-based face mask detection models.," *Data in Brief*, vol. 51, p. 109755, 2023. DOI: https://doi.org/10.1016/j.dib.2023.109755
[17] N. Ma, X. Zhang, H. Zheng and J. Sun, "ShuffleNetV2: Practical Guidelines for Efficient CNN Architecture Design," *arXiv*, 2018. DOI: https://doi.org/10.48550/arXiv.1807.11164
[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv*, 2020. DOI: https://doi.org/10.48550/arXiv.2010.11929
[19] S. Mehta and M. Rastegari, "MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer," *arXiv*, 2021. DOI: https://doi.org/10.48550/arXiv.2110.02178
[20] M. Rodrigo, C. Cuevas and N. García, "Comprehensive comparison between vision transformers and convolutional neural networks for face recognition tasks.," *Scientific Reports*, vol. 14, no. 1, p. 21392, 2024. DOI: https://doi.org/10.1038/s41598-024-72254-w
[21] H. Cai, J. Li, M. Hu, C. Gan and S. Han, "EfficientViT: Multi-Scale Linear Attention for High-Resolution Dense Prediction," *arXiv*, 2022. DOI: https://doi.org/10.48550/arXiv.2205.14756
[22] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang and J. Ren, "EfficientFormer: Vision Transformers at MobileNet Speed," *arXiv*, 2022. DOI: https://doi.org/10.48550/arXiv.2206.01191
[23] Z. Chen, J. Chen, G. Ding, and H. Huang, "A lightweight CNN-based algorithm and implementation on embedded system for real-time face recognition," *Multimedia Systems*, vol. 29, no. 1, pp. 129–138, 2023. DOI: https://doi.org/10.1007/s00530-022-00973-z
[24] M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S.W. Zamir, R.M. Anwer and F.S. Khan, "EdgeNeXt: Efficiently Amalgamated CNN-Transformer Architecture for Mobile Vision Applications," *arXiv*, 2022. DOI: https://doi.org/10.48550/arXiv.2206.10589
[25] A. Wang, H. Chen, Z. Lin, J. Han and G. Ding, "RepViT: Revisiting Mobile CNN From ViT Perspective," *arXiv*, 2023. DOI: https://doi.org/10.48550/arXiv.2307.09283
[26] A. Jangra, "Face Mask Detection \~12K Images Dataset," 2020. URL: https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset (accessed: 12 June 2025)
[27] O. Gurav, "Face Mask Detection Dataset," *Kaggle*, 2020. URL: https://www.kaggle.com/datasets/omkargurav/face-mask-dataset (accessed: 9 June 2025)
[28] K.L. Lew, K.S. Sim and Z. Ting, "Deep Learning Approach EEG Signal Classification," *International Journal on Informatics Visualization*, vol. 8, no. 3–2, pp. 1693–1702, 2024. DOI: https://doi.org/10.62527/joiv.8.3-2.2959