computers

MDPI

Article

Real-Time Detection of Face Mask Usage Using Convolutional Neural Networks

Athanasios Kanavos [1,*], Orestis Papadimitriou [1], Khalil Al-Hussaeni [2,*], Manolis Maragoudakis [3] and Ioannis Karamitsos [4]

[1] Department of Information and Communication Systems Engineering, University of the Aegean, 83200 Samos, Greece; icsdd20016@icsd.aegean.gr
[2] Computing Sciences Department, Rochester Institute of Technology, Dubai 341055, United Arab Emirates
[3] Department of Informatics, Ionian University, 49100 Corfu, Greece; mmarag@ionio.gr
[4] Graduate and Research Department, Rochester Institute of Technology, Dubai 341055, United Arab Emirates; ixkcad1@rit.edu
* Correspondence: icsdd20017@icsd.aegean.gr (A.K.); kxacad@rit.edu (K.A.-H.)

Abstract: The widespread adoption of face masks has been a crucial strategy in mitigating the spread of infectious diseases, particularly in communal settings. However, ensuring compliance with mask-wearing directives remains a significant challenge due to inconsistencies in usage and the difficulty in monitoring adherence in real time. This paper addresses these challenges by leveraging advanced deep learning techniques within computer vision to develop a real-time mask detection system. We have designed a sophisticated convolutional neural network (CNN) model, trained on a diverse and comprehensive dataset that includes various environmental conditions and mask-wearing behaviors. Our model demonstrates a high degree of accuracy in detecting proper mask usage, thereby significantly enhancing the ability of organizations and public health authorities to enforce mask-wearing rules effectively. The key contributions of this research include the development of a robust real-time monitoring system that can be integrated into existing surveillance infrastructures to improve public health safety measures during ongoing and future health crises. Furthermore, this study lays the groundwork for future advancements in automated compliance monitoring systems, extending their applicability to other areas of public health and safety.

Keywords: face mask detection; convolutional neural networks (CNNs); advanced CNN techniques; deep transfer learning; computer vision

check for updates

Citation: Kanavos, A.; Papadimitriou, O.; Al-Hussaeni, K.; Maragoudakis, M.; Karamitsos, I. Real-Time Detection of Face Mask Usage Using Convolutional Neural Networks. *Computers* **2024**, *13*, 182. https://doi.org/10.3390/computers13070182

Academic Editors: Aditya Kumar Sahu, Amine Khaldi and Jatindra Kumar Dash

Received: 5 June 2024
Revised: 8 July 2024
Accepted: 13 July 2024
Published: 22 July 2024

1. Introduction

Following public health rules like wearing masks and maintaining distance from others is mandated by many organizations to mitigate the spread of airborne diseases. However, adherence to these measures varies, sometimes intentionally and sometimes inadvertently, complicating assessments of their effectiveness. The World Health Organization underscores the importance of masks as part of a comprehensive approach to halt disease transmission and promote public health, recommending that mask-wearing become a normative behavior in public settings [1].

Many governments, businesses, and organizations are keen on deploying systems that can monitor compliance continuously. However, this is complicated by the fact that individuals may initially follow the rules by wearing masks but might later remove them or wear them incorrectly [2]. These monitoring systems must operate in real time and with high accuracy since even brief periods without proper mask usage can present significant health risks. These challenges are exacerbated by the dynamic nature of human behavior, where individuals may comply initially but later deviate from health guidelines. Furthermore, traditional methods of monitoring compliance are inadequate for real-time analysis and intervention, which are critical in high-risk settings like airports and hospitals.

One of the significant challenges in implementing real-time monitoring systems is the need for effective integration of hardware and software to achieve low latency and high throughput. Utilizing edge devices equipped with hardware accelerators such as GPUs or TPUs can help process data efficiently on-site, reducing delays between data capture and analysis. This is particularly critical in high-traffic environments like shopping malls, airports, and public transportation hubs [3].

Moreover, optimizing neural network models through techniques such as model pruning and quantization can decrease computational demands and memory usage, facilitating their deployment on resource-constrained devices without compromising performance. Additionally, addressing ethical considerations is crucial when deploying surveillance technologies [4]. Ensuring data privacy and adhering to regulations like the General Data Protection Regulation (GDPR) are vital for maintaining public trust and preventing data misuse [5].

Utilizing deep learning, a powerful subset of machine learning known for its prowess in image recognition, anomaly detection, and language understanding could be the key to overcoming these challenges [6]. By integrating deep learning with computer vision—a branch of AI focused on interpreting visual information—we can develop advanced systems capable of determining whether individuals are wearing masks correctly [7]. Computer vision analyzes images to detect whether a face mask is present and assesses its wear correctly. Training models extensively with a diverse dataset improves their ability to recognize and classify images accurately based on mask usage [8].

Furthermore, combining advanced data analysis and machine learning techniques with these monitoring systems can enhance public health management by predicting potential outbreak sites and compliance lapses, allowing for proactive responses. For instance, machine learning algorithms can identify correlations between specific times, locations, and non-compliance rates, providing valuable insights for resource allocation and public health messaging [9].

Additionally, integrating these monitoring systems with Internet of Things (IoT) and edge-computing technologies can significantly improve their performance and scalability. IoT devices enable the deployment of sensors and cameras across various settings, while edge computing facilitates local data processing, reducing latency and bandwidth demands. This synergy enhances system responsiveness and ensures data privacy and security [10].

In the early part of 2020, the World Health Organization declared the outbreak of COVID-19 a global pandemic. This virus has posed a severe threat to worldwide health, exacerbated by emerging variants. To combat this crisis, technologies for automatically detecting proper mask usage are crucial [11]. However, there has been a gap in research, particularly in recognizing faces with masks. This paper aims to address this gap by utilizing a comprehensive dataset for detecting masks and recognizing masked faces. The dataset, which includes images of 226 individuals representing diverse demographics and various mask orientations, fills a critical gap in standardized data for masked face recognition [12,13]. Utilizing this dataset not only contributes to technological advancements in health rule compliance but is also pivotal in combating the ongoing COVID-19 pandemic.

This paper introduces an advanced approach to real-time face mask detection using a novel convolutional neural network (CNN) architecture, tailored specifically to address public health safety measures during health crises. We developed a sophisticated CNN model optimized for accuracy and efficiency, trained on a comprehensive dataset that captures a wide array of mask-wearing scenarios, including different mask types, wearer positions, and background variations. By employing transfer learning, we enhanced the model's generalization capabilities across new, unseen data, further augmented by data augmentation techniques that introduced artificial variability to better simulate real-world conditions. The model's effectiveness was rigorously validated through extensive testing on a publicly accessible dataset, where it demonstrated superior performance in detecting mask usage accurately compared to existing methods. This holistic approach not only advances the field of computer vision in public health applications but also provides a robust tool for enhancing compliance with mask-wearing protocols, thereby contributing to the control of disease spread in communal settings.

The organization of this paper is as follows: Section 2 reviews the related work, setting the stage by discussing existing methodologies and advancements in the field of mask detection using deep learning techniques, and highlighting the gaps that our research aims to fill. In Section 3, the foundations of our methodology are laid out, detailing the deep learning concepts and theoretical underpinnings that support our approach, including the fundamentals of convolutional neural networks. The model architecture, described in Section 4, elaborates on the specific CNN models designed for this study, explaining their configurations and the rationale behind their structures. Section 5 presents the evaluation of these models, including the methodologies for training, testing, and the metrics used to assess their performance. In Section 6, a comparative analysis and discussion are provided, where our models' outcomes are benchmarked against existing solutions, showcasing their efficacy and improvements over prior work. Section 7 explores an ablation study that systematically investigates the impact of various architectural components on the model's performance, providing a deeper understanding of their contributions. Finally, Section 8 concludes by summarizing this study's findings and outlining future research directions that could extend and enhance the proposed solutions.

2. Related Work

During the COVID-19 pandemic, the necessity for technologies capable of detecting face masks and recognizing faces with masks has been underscored, facing numerous practical challenges. This surge in technological development has led to diverse research efforts, which typically fall into three main categories: traditional machine learning (ML) methods, deep learning (DL) techniques, and hybrid approaches that combine elements of both. These efforts aim not only to address immediate needs but also to innovate on the robustness and efficiency of recognition systems in public health scenarios.

Traditional ML methods, while overshadowed by more sophisticated techniques, are still employed in some studies due to their simplicity and lower computational demands. Systems alerting when healthcare workers fail to wear masks have been developed using Viola-Jones for face detection and Gentle AdaBoost for mask detection [14]. Additionally, comparisons between traditional ML classifiers such as KNN and SVM with DL models like MobileNet have demonstrated the latter's superior effectiveness in mask detection scenarios [15]. These traditional approaches provide a valuable baseline for evaluating the advanced capabilities of newer models.

Deep learning has become the dominant approach due to its robustness in handling complex image processing tasks [16]. The InceptionV3 model, for instance, has been utilized to differentiate between masked and unmasked faces using the Simulated Masked Face dataset [17]. Furthermore, systems based on SSDMNV2, which combine a single-shot multibox detector with MobileNetV2, have been created for enhanced accuracy in classification [18]. Additionally, real-time systems using VGG-16, and three-stage cascaded CNN architectures, although demanding significant computational resources, illustrate the significant advances in the field [19,20].

Hybrid approaches aim to leverage the strengths of both traditional and modern methods, providing a balanced solution for complex detection tasks [21–23]. Models combining ResNet50 with SVM and other machine learning algorithms have been developed to improve decision-making processes [24]. The HybridFaceMaskNet, which integrates deep learning, handcrafted feature extraction, and traditional ML classifiers, has been proposed to efficiently detect face masks, showing that a combination of approaches can enhance detection accuracy [25].

Recent applications of these technologies have demonstrated their utility in real-world settings, emphasizing practical deployments over theoretical models. Methods that simplify the facial recognition process by using features extracted via Haar Cascade have achieved high accuracy rates, streamlining detection through deep neural networks [26].

Additionally, the YOLOv3 architecture, known for its efficiency in object detection, has been effectively applied to mask detection, showcasing impressive performance in live video feeds [27].

Furthermore, the integration with existing surveillance systems has been a focus area, extending the application of these technologies to a wider array of public environments. Models based on MobileNetV2 have been used to monitor mask usage in public areas using data from various surveillance sources, achieving high accuracy rates [28]. Similarly, facial recognition and detection systems utilizing a global pooling block with a pre-trained MobileNet to prevent overfitting have demonstrated how advanced pooling strategies can enhance the recognition process [29].

Emerging technologies continue to evolve, incorporating advanced computational methods to improve the efficacy and efficiency of mask detection systems. Real-time deep learning methods for classifying facial expressions using architectures like VGG-16 have been crucial in aiding the enforcement of mask regulations during the pandemic [30]. Moreover, principal component analysis has been used to distinguish between masked and unmasked individuals, enhancing facial recognition capabilities even under the constraints of mask-wearing [31]. Additionally, novel approaches using CNNs to determine head orientation have significantly improved recognition accuracy for individuals wearing masks, offering promising directions for future research in ensuring compliance with health guidelines [32–36].

Recent advancements in facial image processing, particularly in the context of facial expression recognition with face mask occlusion, have demonstrated innovative approaches to handling partially occluded faces. Notable among these is the work presented in [37], which enhances CNN architectures to better recognize facial expressions even when masks obscure part of the face. Similarly, [38] introduces an adaptive dual-attention mechanism that adjusts to occlusions by focusing on unoccluded regions of the face, and while these studies focus primarily on facial expression recognition, our framework distinguishes itself by specifically targeting mask detection and compliance. Our approach not only identifies the presence of masks but also ensures that they are worn correctly, addressing public health compliance rather than emotional expression. This difference underscores the unique application and technical adaptation of our CNN models to meet the specific demands of public health safety measures in the context of ongoing health crises.

This section collectively advances our understanding and capabilities in face mask detection and recognition, contributing to public health safety measures during the ongoing global health crisis.

3. Methodology Foundations

3.1. Convolutional Neural Networks

Convolutional neural networks (CNNs) are a pivotal element in the field of deep learning, designed to efficiently process spatial hierarchies in image data by recognizing patterns at various scales and complexities. This capability is facilitated by a rigorous training phase where the network learns to identify and enhance important features from different areas of an image, thereby improving its post-training performance significantly, especially in complex image-based applications like medical diagnostics.

The architecture of CNNs is recognized for its ability to automate feature extraction, which is achieved through the strategic arrangement of convolutional, pooling, and fully connected layers. These layers work in concert to effectively classify data with high precision. By adding multiple fully connected layers, CNNs can refine the feature extraction process, thus simplifying the representation of image data and enhancing the model's interpretive performance [39].

Structured similarly to multilayer perceptrons, CNNs consist of an input layer, multiple hidden layers, and an output layer. The key component, the convolutional layer, employs specialized operations to extract salient features from the input image. Downsampling techniques within these layers enhance computational efficiency, reinforcing the

CNN's capability to interpret complex visual data with minimal manual preprocessing, a major step forward for automated medical analysis and diagnostics.

3.2. Tensorflow

Developed by Google, TensorFlow is an expansive open-source framework tailored for executing complex mathematical computations, fundamental to constructing and training deep learning models. Its capacity to manage dataflow graphs, which detail data transformations through various computational phases, is critical for operational efficiency. Nodes in these graphs represent mathematical operations on tensors, and edges illustrate the flow of data between these operations.

TensorFlow's design allows it to excel on a variety of computational platforms, encompassing mobile devices and extensive distributed systems, utilizing CPUs and GPUs. This versatility makes it particularly suited for the demands of training large-scale deep learning models used in tasks such as image recognition.

In the domain of image recognition, TensorFlow excels due to its efficient management of convolutional and pooling layers, crucial for high-level image classification tasks. The framework supports a layered architecture similar to a multilayer perceptron, enhancing the hierarchical processing of image data which is vital for effective feature extraction and classification.

Moreover, TensorFlow's capabilities extend to mobile and edge computing with TensorFlow Lite, and to large-scale production environments with TensorFlow Extended (TFX), which provides tools for deploying machine learning solutions at scale [40].

3.3. Keras

Keras is a high-level, Python-based, open-source interface designed for the streamlined creation and training of deep learning models, particularly within the TensorFlow ecosystem. It simplifies the development process by providing a more abstract and user-friendly layer of operations, which allows developers to focus more on designing and implementing neural networks without getting bogged down by the intricate details of underlying tensor manipulations.

Keras facilitates model construction through its Sequential API, a method where models are built by stacking layers linearly. This architecture is particularly effective for standard deep learning models as each layer is designed to accept a single tensor as input and output another tensor, creating a clear and efficient pipeline for model building. By abstracting away many of the lower-level operations, Keras enables developers to experiment more freely with deep learning, significantly speeding up the development of sophisticated models without compromising on performance or flexibility [41].

In medical image analysis, Keras is often employed to quickly prototype CNNs that can handle complex image datasets. For instance, layers such as convolutional layers, pooling layers, and fully connected layers can be easily stacked to recognize and classify various pathological features from medical scans, demonstrating Keras's utility in rapidly deploying models that are both robust and accurate.

3.4. Convolutional Layers

Convolutional layers form the backbone of CNNs, optimizing the automatic extraction of spatial features such as edges and textures from images. These layers apply a kernel or filter across the image, calculating the dot product of the filter with the image pixels at each position to produce a feature map that indicates the presence and intensity of features.

The convolution operation in CNNs can be mathematically expressed as

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(i+m, j+n) \cdot K(m,n) \tag{1}$$

where $S(i,j)$ is the output feature map, I is the input image, K is the kernel or filter, (i,j) are the coordinates on the feature map, (m,n) are the coordinates in the kernel, and $\cdot$ denotes the convolution operation.

Each element $S(i,j)$ of the output feature map is the sum of the element-wise product of the kernel K and the portion of the input image I over which the kernel is currently positioned.

For grayscale images, the input matrix I will have a single layer. In contrast, color images typically consist of three layers (RGB), with the convolution operation often performed separately on each layer.

The kernel is a smaller matrix relative to the input image, with dimensions typically 3 × 3 or 5 × 5. It contains weights that are learned during the training process and is designed to detect specific types of features from the input image. As the kernel strides over the input image, it performs element-wise multiplication followed by a sum, producing the output feature map where each element represents the presence and intensity of a feature detected at a specific location.

The dimensions of the output feature map (W_{out}, H_{out}) are determined by the size of the input (W_{in}, H_{in}), the filter size (F), the stride (S), and the padding (P) using the following equations:

$$W_{out} = \frac{W_{in} - F + 2P}{S} + 1 \tag{2}$$

$$H_{out} = \frac{H_{in} - F + 2P}{S} + 1 \tag{3}$$

where W_{out} and H_{out} are the width and height of the output feature map, W_{in} and H_{in} are the width and height of the input, F is the filter size, S is the stride, and P is the padding.

3.5. Pooling Layers

Pooling layers are critical in CNNs for reducing the dimensionality of feature maps, thereby lowering computational requirements and enhancing the model's ability to generalize. These layers consolidate the essential information in feature maps by summarizing feature presence in patches, thus making the network more robust to variations in the input.

Pooling layers decrease the size of the feature maps, which reduces the number of parameters and computations required in the network. This simplification allows the network to focus on the most significant features, helping to ensure that the model remains computationally efficient and less prone to overfitting. Additionally, by summarizing the presence of features in patches of the feature map, pooling enhances the network's robustness to minor variations and translations in the input image.

There are several types of pooling techniques, including max pooling, average pooling, and global pooling. In this study, we focus on max pooling, which is the most commonly used form of pooling in deep learning applications. Max pooling operates by selecting the maximum value from a set of values within a defined window (or patch) on the feature map and forwarding this value to the next layer. This technique effectively captures the most pronounced feature in each patch, which is particularly useful for features like edges and textures that are critical in image recognition tasks.

The operation of max pooling can be mathematically expressed as follows:

$$P_{max}(i,j) = \max_{a=0}^{n-1} \max_{b=0}^{n-1} F(i \cdot s + a, j \cdot s + b) \tag{4}$$

where $P_{max}(i,j)$ is the output of the pooling operation at position (i,j), F is the feature map, $n \times n$ is the size of the pooling window, and s is the stride of the pooling window. Variables a and b iterate over the window dimensions, and this operation is applied independently across each position of the feature map to reduce its dimensions.

The size of the pooling window and the stride determine the degree of reduction in the feature map dimensions. A commonly used configuration in many CNN architectures is a 2 × 2 window with a stride of 2. This setup reduces both the height and width of the feature map by half, significantly lowering the spatial resolution but preserving the most critical feature information.

Pooling layers, by reducing the number of parameters, not only saves computational resources but also help in making the detection of features invariant to scale and orientation changes, which is a desirable property in many vision-based applications.

3.6. Batch Normalization

Batch normalization (BN) has become a cornerstone technique in deep learning, particularly valued for enhancing the stability and efficiency of neural network training. It is especially beneficial for deep networks, helping to accelerate the training phase and improve the overall performance and accuracy of the model. Despite its widespread use and observable benefits, the exact mechanisms and theoretical underpinnings of BN continue to be subjects of ongoing research and debate [42].

The principal advantage of batch normalization is its effectiveness in combating the problem of internal covariate shift. This phenomenon occurs when the distributions of each layer's inputs change during training, which can slow down the training process and lead to unstable convergence behaviors. BN tackles this by normalizing the inputs of each layer to ensure they have a consistent mean and variance, as follows:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{5}$$

where x_i is the input to a layer, μ_B and σ_B^2 are the mean and variance calculated over the batch, and ϵ is a small constant added for numerical stability. This normalization allows each layer to learn on a more stable distribution of inputs, facilitating a smoother and faster training process.

By standardizing the inputs in this way, BN enables higher learning rates to be used without the risk of instabilities typically induced by unfavorable initial parameter choices or extreme value ranges. This can significantly speed up the convergence of the training process. Furthermore, BN helps to prevent the network from reaching saturation points—states where changes in input produce minimal or no change in output—which can impede learning. It maintains activation functions within their non-saturating regions, thereby enhancing the sensitivity and responsiveness of the network during training.

Additionally, BN serves a regularization function, reducing the network's dependency on dropout. It allows each layer to utilize more of its input features effectively, promoting more efficient learning dynamics. This regularization effect, while not a substitute for dropout entirely, provides a complementary mechanism that can lead to more robust generalization in some cases.

Overall, batch normalization has proven to be an effective method for improving the training stability and performance of neural networks, contributing to faster convergence rates and more consistent training outcomes. Its integration into modern neural architectures is indicative of its crucial role in advancing the field of deep learning [43].

3.7. Dropout

In the domain of large-scale machine learning, particularly in deep neural networks, overfitting is a pervasive challenge. Overfitting occurs when a model performs exceptionally well on training data but poorly on unseen data, a problem exacerbated by the complex architectures and large parameter sets characteristic of deep networks. Dropout is a regularization technique specifically designed to prevent this issue by randomly disabling certain neurons and their connections during the training phase, thus reducing the risk of interdependent neuron behavior.

The mechanism of dropout involves randomly selecting a subset of neurons in each training iteration and temporarily removing them along with all their incoming and outgoing connections. This process creates a "thinned" network, where the surviving neurons must adapt to the absence of their dropped counterparts. Mathematically, if a neuron's output is represented by x, then during training, dropout is applied by multiplying x by a random variable d drawn from a Bernoulli distribution, as follows:

$$x' = d \cdot x \tag{6}$$

where d is 1 with probability p (the retention probability), and 0 with probability $1 - p$. This operation is performed independently for each neuron, resulting in different network architectures in each training iteration.

During training, this random thinning of the network ensures that no single set of neurons can co-adapt too strongly, since they may be dropped out in subsequent iterations. Instead, the network learns more robust features that are useful in conjunction with many different random subsets of the other neurons. At inference time, all neurons are used, but their outputs are scaled down by a factor equivalent to the retention probability p, compensating for the larger number of active units compared to the training phase.

Dropout has been empirically shown to significantly improve the generalization of neural networks, particularly in scenarios where the training data are limited and the network is large and complex. Unlike traditional regularization methods, which might involve constraining the magnitude of weights directly, dropout regularizes the model by enhancing the diversity of the internal representations learned during training. This diversity ensures that the model does not rely too heavily on any single or small group of features, leading to better performance on unseen datasets [44].

4. Model Architecture

In this research, a tailored convolutional neural network (CNN) was constructed specifically to meet the demands of the classification challenges presented by the dataset. This CNN architecture is meticulously designed to process and classify input images efficiently into their designated categories. It includes multiple layers of convolution and pooling that synergistically extract and compress spatial features from the images. Subsequently, the architecture employs fully connected layers that interpret these features to render the final classification decisions.

The architecture of the CNN initiates with an input layer that receives the image data. Following this, several convolutional layers equipped with spatially sensitive filters are applied to perform robust feature extraction. Each convolution operation is complemented by batch normalization, which plays a critical role in stabilizing the learning process by normalizing the inputs to each layer. Interspersed with these convolutional layers, pooling layers serve to reduce the dimensionality of the feature maps, which simplifies the computational demands and sharpens the model's focus on pivotal features.

Post feature extraction and reduction, the data undergo a flattening process to prepare for dense neural network analysis. This section of the network, containing multiple fully connected layers, is where the deep interpretation of the extracted features occurs, culminating in the classification output.

The diversity in CNN model designs is explored through four distinct architectural configurations, each engineered to evaluate different structural impacts on the model's performance. These variations are visually depicted in Figure 1, illustrating the detailed layer configurations and operations within each proposed model.

These models are initially assessed using a variety of evaluation techniques to identify optimal configurations, and subsequently, their performance is compared when applied to a uniform structural framework. The distinctive features of these architectures are summarized in Table 1, detailing the sequence and operations of layers within each model.

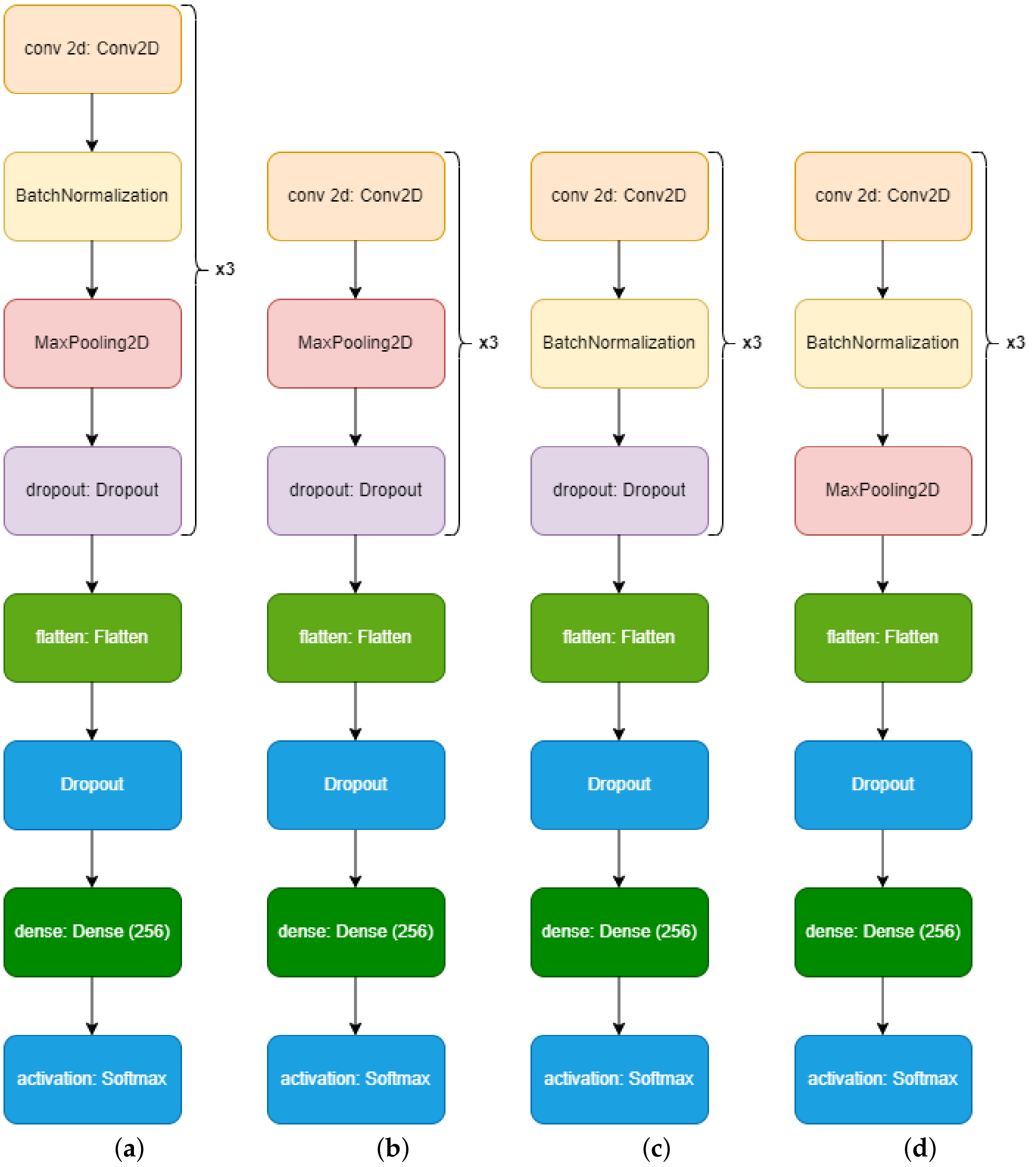


Figure 1. Visual representations of the proposed CNN architectures where each diagram delineates the arrangement and operations of layers within the models. (**a**) First proposed CNN architecture. (**b**) Second proposed CNN architecture. (**c**) Third proposed CNN architecture. (**d**) Fourth proposed CNN architecture.

The diversity in architectural configurations is designed to assess the impact of layer depth and sequence on the accuracy and speed of mask detection. For instance, architectures with more BatchNorm layers are hypothesized to enhance generalization across varied lighting conditions in mask detection scenarios.

Integral to all four architectures are layers including the following:

- Input() that initializes a symbolic tensor named “images” to hold the image data.
- Conv2D() which constructs a convolution kernel that is convolved with the layer input to produce a tensor of outputs. Conv2D() is pivotal for feature extraction in our CNN. By convolving with the layer input, it highlights essential features such as the edges and shapes of masks on faces, which are crucial for accurate mask detection.
- Batch Normalization() normalizes the output of the previous layer at each batch, applying a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.

- MaxPooling2D() performs downsampling by dividing the input into rectangular pooling regions and computing the maximum of each region.
- Flatten() transforms the formatted data into a 1D array for input into the next layer.
- Dropout() randomly omits individual connections between layers during training, which helps prevent overfitting.
- Dense() fully connected layer that processes the network's learned features from the convolutional layers.
- Softmax() applies the softmax function to the input, normalizing the output distribution over predicted output classes.

This section elaborates on the sophisticated structuring of CNN models designed to enhance classification accuracy, detailing the functionality and integration of various layers within the architectures to achieve optimal performance in image categorization tasks.

Table 1. Detailed configurations of the proposed CNN architectures.

Architecture	Layer Sequence and Operations
1st	(Conv2D → BatchNorm → MaxPooling2D → Dropout) ×3 → (Flatten → Dropout → Dense) → Softmax
2nd	(Conv2D → MaxPooling2D → Dropout) ×3 → (Flatten → Dropout → Dense) → Softmax
3rd	(Conv2D → BatchNorm → Dropout) ×3 → (Flatten → Dropout → Dense) → Softmax
4th	(Conv2D → BatchNorm → MaxPooling2D) ×3 → (Flatten → Dropout → Dense) → Softmax

5. Evaluation

5.1. Dataset

The dataset utilized for the Face Mask Detection Classification task is a comprehensive collection of nearly 12,000 images, sourced from a publicly available repository on Kaggle (https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset, accessed on 16 July 2024). The images are distributed across two primary categories: 'With Mask' and 'Without Mask', ensuring the dataset addresses the binary classification nature of the task effectively.

Each image in the dataset is a high-resolution file in JPEG format, meticulously annotated to indicate whether a mask is present or absent. This rich dataset is organized into distinct sets for training, testing, and validation purposes, facilitating a systematic approach to model training and performance evaluation.

Each image was selected based on its clarity and relevance to common real-world scenarios, ensuring a practical focus. The annotation process involved multiple reviewers to confirm the presence or absence of masks, reducing subjective bias and enhancing the dataset's accuracy.

The dataset includes a balanced representation of both categories, with 5000 images for training and 400 images for validation per category. Additionally, an extended set of images enhances the dataset's diversity, aiding in the development of robust machine learning models capable of recognizing masked and unmasked faces under various conditions.

The distribution of images across different subsets—training, testing, and validation—is detailed in Table 2. This table provides an overview of the number of images available for each category within each subset, supporting a comprehensive evaluation of the model's performance across varied operational scenarios.

Table 2. Distribution of class instances.

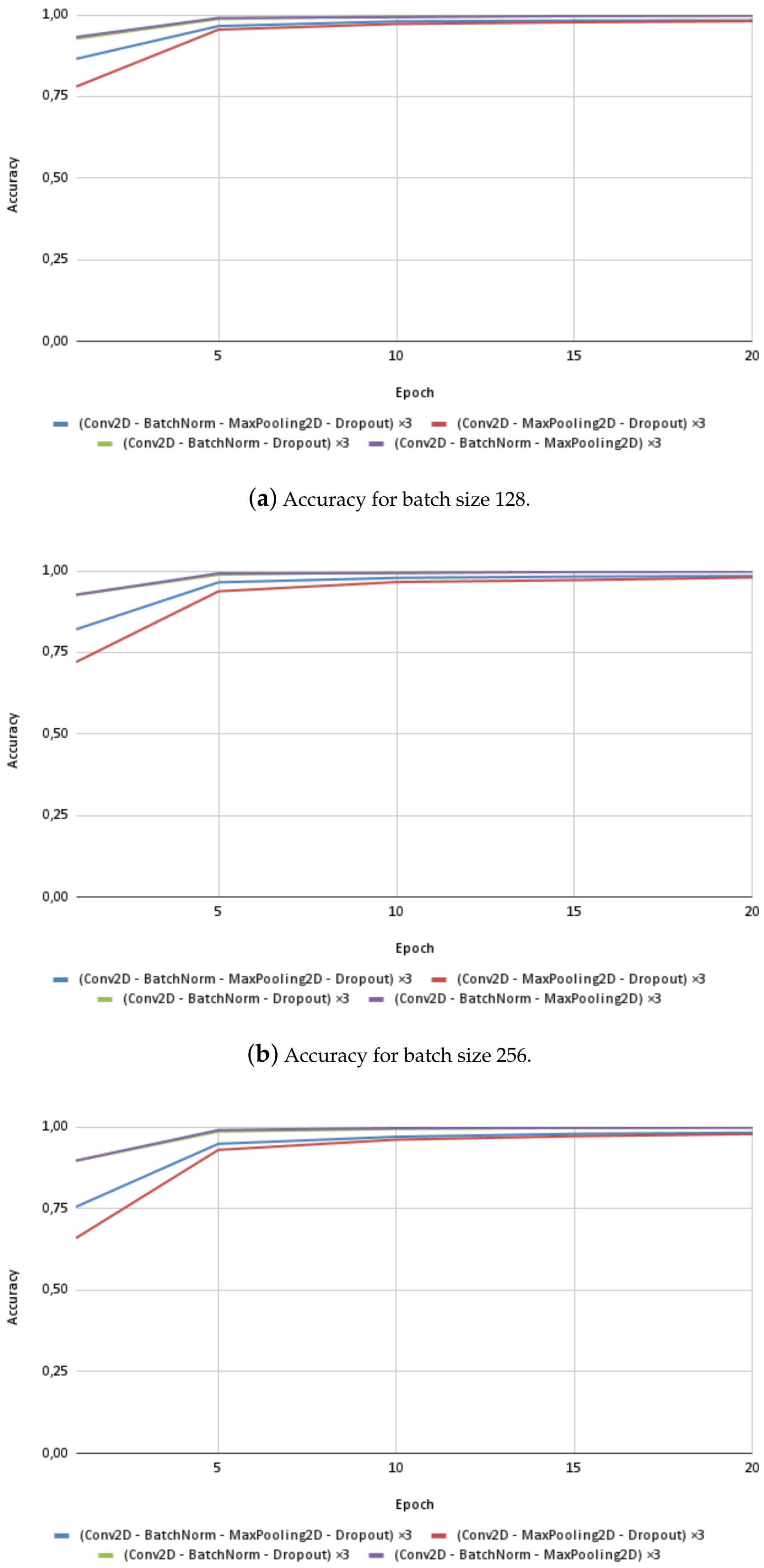
Face Mask	Test	Train	Validation
With Mask	483	5000	400
Without Mask	509	5000	400
Total	992	10,000	800

5.2. Results and Analysis

This subsection evaluates the performance of four proposed CNN architectures across varying epochs and batch sizes, focusing on key metrics such as loss, accuracy, and computational time. Detailed results are tabulated in Table 3 and graphically represented in Figures 2–4.

Table 3. Experimental evaluation for four architectures.

Epochs	Loss	Accuracy	Time	Loss	Accuracy	Time	Loss	Accuracy	Time
1st: (Conv2D - BatchNorm - MaxPooling2D - Dropout) ×3									
	Batch Size = 128			**Batch Size = 256**			**Batch Size = 512**		
1	0.3777	0.8653	55	0.5296	0.8209	57	0.7437	0.7551	61
5	0.0958	0.9659	52	0.1120	0.9650	56	0.1514	0.9483	55
10	0.0594	0.9803	51	0.0642	0.9786	54	0.0918	0.9700	57
15	0.0529	0.9826	52	0.0523	0.9826	56	0.0662	0.9789	55
20	0.0464	0.9835	52	0.0432	0.9844	54	0.0519	0.9824	56
2nd: (Conv2D - MaxPooling2D - Dropout) ×3									
	Batch Size = 128			**Batch Size = 256**			**Batch Size = 512**		
1	0.4709	0.7803	42	0.5569	0.7206	42	0.6344	0.6592	54
5	0.1342	0.9550	39	0.1756	0.9375	41	0.1965	0.9297	42
10	0.0848	0.9722	38	0.1066	0.9659	40	0.1171	0.9610	43
15	0.0649	0.9774	39	0.0847	0.9718	42	0.0919	0.9716	50
20	0.0590	0.9808	39	0.0614	0.9799	39	0.0673	0.9781	40
3rd: (Conv2D - BatchNorm - Dropout) ×3									
	Batch Size = 128			**Batch Size = 256**			**Batch Size = 512**		
1	0.2997	0.9272	170	0.2706	0.9264	169	0.3704	0.8961	192
5	0.0386	0.9877	163	0.0338	0.9885	161	0.0415	0.9860	196
10	0.0100	0.9967	161	0.0126	0.9958	161	0.0174	0.9937	188
15	0.0091	0.9972	160	0.0074	0.9972	163	0.0098	0.9970	189
20	0.0041	0.9980	163	0.0029	0.9989	165	0.0052	0.9980	190
4th: (Conv2D - BatchNorm - MaxPooling2D) ×3									
	Batch Size = 128			**Batch Size = 256**			**Batch Size = 512**		
1	0.1835	0.9323	64	0.2078	0.9272	64	0.2743	0.8967	63
5	0.0271	0.9907	59	0.0261	0.9925	63	0.0308	0.9904	61
10	0.0200	0.9930	61	0.0216	0.9934	61	0.0117	0.9967	62
15	0.0093	0.9967	62	0.0088	0.9971	60	0.0117	0.9969	62
20	0.0117	0.9978	61	0.0055	0.9981	62	0.0054	0.9983	64

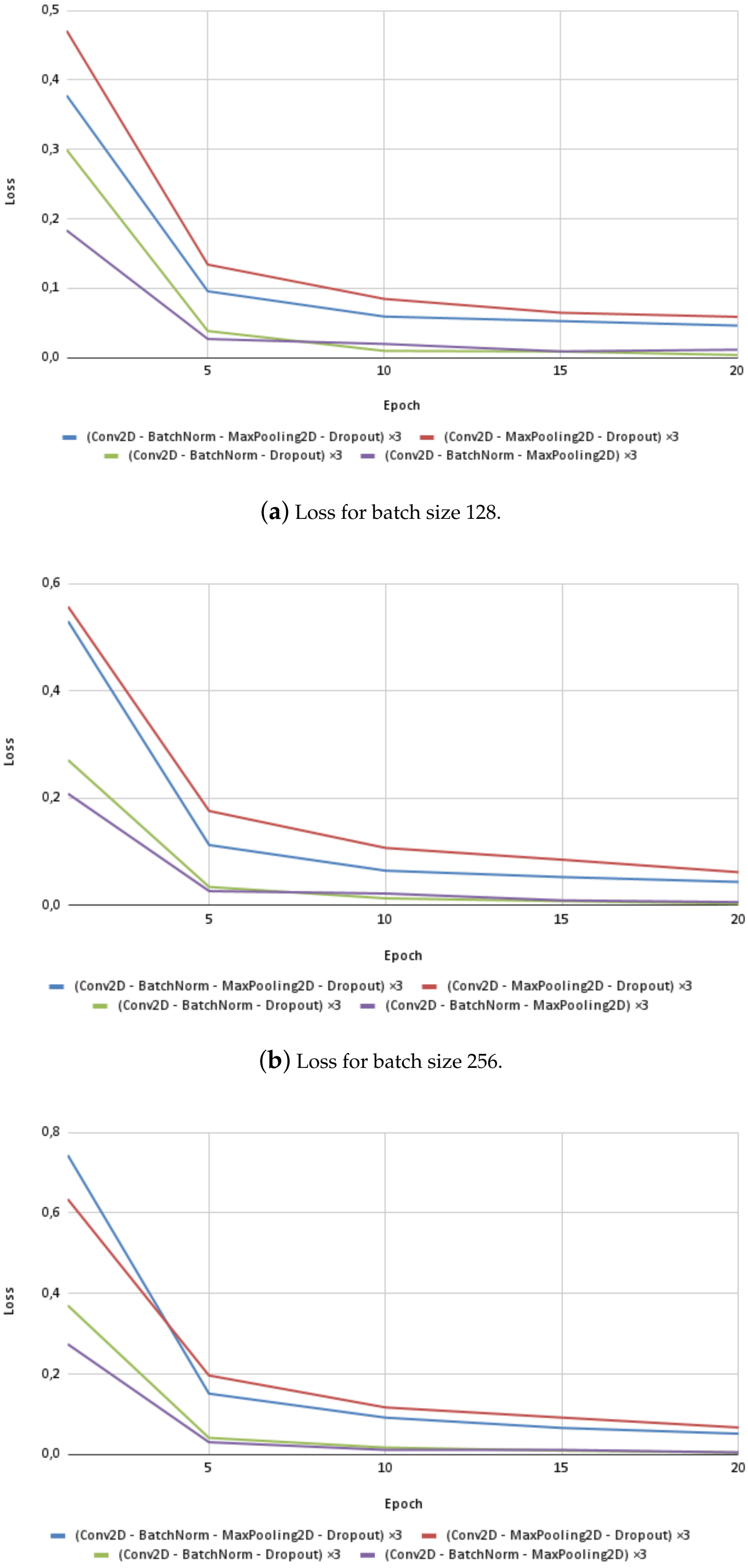


(**a**) Accuracy for batch size 128.

(**b**) Accuracy for batch size 256.

(**c**) Accuracy for batch size 512.

Figure 2. Accuracy trajectories for different batch sizes across the four proposed models.

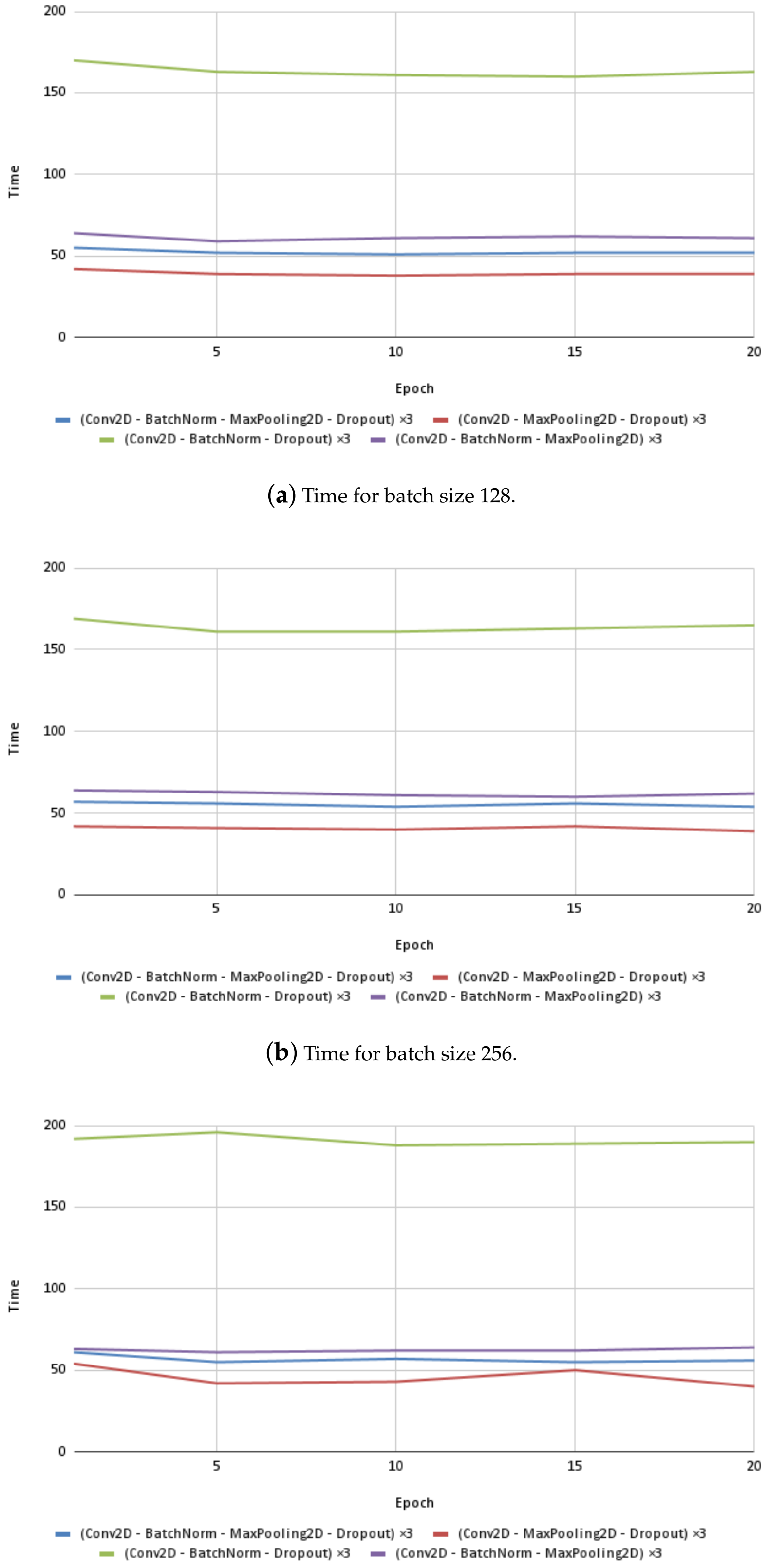


(**a**) Loss for batch size 128.

(**b**) Loss for batch size 256.

(**c**) Loss for batch size 512.

Figure 3. Loss curves for different batch sizes across the four proposed models.



(**a**) Time for batch size 128.

(**b**) Time for batch size 256.

(**c**) Time for batch size 512.

Figure 4. Computational time for different batch sizes for the four proposed models.

Each architecture was evaluated using batch sizes ranging from 128 to 512, documenting performance metrics at key milestones (1, 5, 10, 15, and 20 epochs). The outcomes indicate that smaller batch sizes typically facilitate quicker learning, though they may increase the risk of overfitting. Conversely, larger batch sizes often result in more stable but slower learning curves.

The performance of each architecture across different batch sizes is visually represented in the figures below. These graphs illustrate the trajectory of loss, accuracy, and computational time, providing insights into the scalability and efficiency of each model.

5.2.1. First Architecture: Enhanced Feature Extraction and Regularization

The first architecture, integrating Conv2D, BatchNorm, MaxPooling2D, and dropout layers three times, shows robust learning capabilities. At a batch size of 128, it swiftly reduces loss from 0.3777 to 0.0464 within 20 epochs, achieving an accuracy of 98.35%. This setup demonstrates the effectiveness of BatchNorm in stabilizing parameter updates throughout training, thus facilitating faster convergence. Additionally, the dropout layers help in reducing overfitting by randomly deactivating neurons during training, which enhances the generalization capability of the network.

This architecture's consistent performance across different batch sizes and training epochs highlights its suitability for applications requiring reliable and rapid processing, such as real-time image classification systems. The rapid improvement in loss and accuracy, particularly in the early epochs, underscores the architecture's efficiency in adapting to the data.

5.2.2. Second Architecture: Streamlined Efficiency

The second architecture, consisting of a simpler sequence of Conv2D, MaxPooling2D, and dropout layers repeated three times, emphasizes efficiency and faster computational times. This model is particularly advantageous for scenarios with limited computational resources, showing significant improvement in training duration per epoch. For instance, at a batch size of 128, it reduces the average training time per epoch by about 3 s compared to the first architecture while maintaining high accuracy levels, peaking at 98.08% by the 20th epoch.

Despite its streamlined design, this architecture effectively captures and classifies features, demonstrating its potential for deployment in environments where both speed and accuracy are critical. Its performance suggests that removing BatchNorm does not drastically impact the learning capabilities, provided that other regularization techniques like dropout are effectively utilized.

5.2.3. Third Architecture: High Stability and Accuracy

The third architecture, featuring repeated sequences of Conv2D, BatchNorm, and dropout, excels in stability and high accuracy across all batch sizes. It maintains an accuracy above 99.67% by the 20th epoch for the smallest batch size, showcasing excellent resilience against overfitting and superior adaptability to varying training conditions. The inclusion of BatchNorm after each convolutional layer ensures consistent normalization of activations, which reduces internal covariate shift and accelerates the training process.

This architecture is particularly effective for tasks that require precise and reliable outcomes, such as medical image analysis, where high accuracy and model stability are paramount. Its ability to perform consistently well across various training configurations makes it a robust choice for critical applications.

5.2.4. Fourth Architecture: Optimal Convergence and Performance

The fourth architecture, with a repetitive setup of Conv2D, BatchNorm, and MaxPooling2D layers, is designed to achieve optimal convergence rates and maintain high performance standards. This model achieves the best balance between accuracy and computational efficiency, showcasing the lowest loss rates and highest accuracies consistently across epochs and batch sizes. For instance, at a batch size of 128, it reaches an accuracy of nearly 99.78% by the 20th epoch, with minimal fluctuations in performance metrics.

This architecture's strong performance underlines its effectiveness for high-stake applications where both precision and efficient processing are required. Its scalable design ensures that performance does not degrade with increased batch sizes, making it ideal for large-scale deployment.

Each architecture's performance is meticulously analyzed, providing valuable insights into how different configurations and batch sizes affect the learning dynamics and overall effectiveness of the models. This detailed examination aids in understanding each model's strengths and potential areas for improvement, guiding future refinements and deployments in various applications.

6. Comparative Analysis and Discussion

This section undertakes a rigorous comparative analysis to gauge the performance of our proposed models against existing alternatives in the field of face mask detection, focusing particularly on classification accuracy. The outcomes, summarized in Table 4, highlight the effectiveness of our third architecture, which achieved significant accuracy enhancements.

Table 4. Comparative analysis with other studies.

Study	Accuracy (%)
Pham-Hoang-Nam et al. [45]	94.59%
Abirami et al. [46]	99.54%
Mohan et al. [47]	99.79%
Aydemir et al. [48]	100%
Fallaha et al. [49]	100%
Proposed Method (3rd architecture)	99.89%

Our third architecture, utilizing a sequence of Conv2D, BatchNorm, and dropout layers repeated three times, attained an impressive accuracy of 99.89%. This model not only surpassed the performance of several prior studies, such as [45] at 94.59% and [46] at 99.54%, but also closely approached the perfect scores reported in [48,49].

The slight discrepancy between our model's performance and the perfect scores could be attributed to several factors, including differences in dataset complexity, model generalization capabilities, and possibly the overfitting of models in other studies where perfect scores were achieved. While a perfect score is desirable, it often raises concerns about the model's ability to generalize across unobserved data. Thus, our model's slightly lower score may actually reflect a better balance between accuracy and generalizability.

Furthermore, this discussion section examines the limitations and potential failure cases of our model. Despite its high accuracy, the model may still encounter challenges in environments with extreme variations in lighting, occlusions, or highly unconventional mask types not represented in the training data. These conditions could affect the model's ability to detect masks accurately, leading to potential false negatives or positives. Recognizing these limitations is crucial for ongoing improvements and for setting realistic expectations for the model's deployment in diverse real-world scenarios.

Moreover, the architecture's robustness is underscored by its ability to significantly outperform earlier approaches under similar evaluation conditions, suggesting that our enhancements in model design—particularly the integration of BatchNorm and dropout—have effectively augmented its capability to handle varied and complex image scenarios more efficiently.

The analysis also highlights the critical role of architecture configuration in achieving high accuracy. The incorporation of BatchNorm helps in stabilizing the learning process by normalizing the inputs to each layer, thus facilitating faster and more stable convergence. Similarly, dropout prevents over-dependence on specific neurons, enhancing the model's robustness and preventing overfitting.

Given these results, our model not only establishes new benchmarks in face mask detection accuracy but also offers insights into the architectural features that contribute to high-performance deep learning models. This understanding is crucial for future research and development in the field, suggesting that similar architectural strategies could be beneficially applied to other complex image classification tasks.

The superior performance of our model compared to those achieving near-perfect scores also invites further investigation into the trade-offs between accuracy and other critical performance metrics like model interpretability, computational efficiency, and real-time processing capabilities. Such comprehensive evaluations are essential for the practical deployment of deep learning models in real-world applications, where multiple factors influence the ultimate utility of the technology.

7. Ablation Study

To validate the contributions of specific components within our convolutional neural network (CNN) model, an ablation study was conducted. This study systematically assessed the impact of removing or altering key layers and configurations on model performance, focusing on accuracy, computational efficiency, and generalization capabilities.

Our ablation study involved creating several variants of the original CNN architecture. Each variant was modified by either removing or adjusting layers such as batch normalization, dropout, and different settings of convolutional layers. This study was designed to quantify the impact of these components on the model's performance in terms of classification accuracy, training time, and robustness.

The results of the ablation study are summarized in Table 5, which shows the performance metrics for each model variant compared to the best performing full model configuration from Section 5.

Table 5. Ablation study results.

Model Variant	Accuracy (%)	Training Time (s)	Comments
Full Model (3rd Architecture, Batch Size 256)	99.89	165	Baseline
No BatchNorm	97.50	150	Faster but less accurate
No Dropout	98.75	165	Slight decrease in accuracy, more overfitting
Reduced Conv Layers	98.00	145	Less complex, faster

The ablation study provided several key insights, as follow:

- **Batch Normalization:** Removing batch normalization resulted in a noticeable decrease in accuracy by approximately 2.39%, confirming its role in stabilizing the learning process and improving convergence.
- **Dropout:** Models without dropout layers showed only a slight decrease in accuracy but were more prone to overfitting, demonstrating the importance of dropout in enhancing generalization.
- **Convolutional Layers:** Reducing the number of convolutional layers led to quicker training times but at the cost of reduced accuracy, highlighting the trade-off between model complexity and performance.

These findings validate the necessity of each examined component in our model architecture, with each playing a critical role in achieving the balance between efficiency and accuracy.

Based on the results, several adjustments can be recommended to enhance model performance, as follow:

- Incorporate batch normalization consistently to ensure model stability across different training scenarios.
- Utilize dropout strategically to prevent overfitting, especially when expanding the model to larger datasets.
- Optimize the number of convolutional layers to balance computational demands with performance needs, particularly for deployment in resource-constrained environments.

The ablation study underscores the importance of each component in our CNN architecture, providing a robust foundation for further refinement and ensuring that our model is well-suited for practical deployment in mask detection tasks.

8. Conclusions and Future Work

This research has effectively demonstrated the considerable capabilities of convolutional neural networks (CNNs) in the detection and classification of face masks, a critical component in managing public health, particularly during global health crises such as the COVID-19 pandemic. Our findings highlight the robustness and precision of CNN models in distinguishing between masked and unmasked faces, a task that has significant implications for public safety and disease prevention.

The high accuracy rates achieved by our CNN models underscore their potential to significantly enhance current surveillance and monitoring systems. These systems are vital for enforcing public health policies and ensuring compliance with safety regulations, which in turn helps in curbing the spread of infectious diseases. The ability of our models to accurately identify compliance in real time can aid public health officials and policymakers in making informed decisions that protect community health.

Our study not only reaffirms the efficacy of CNNs in complex image recognition tasks but also sets a benchmark for future applications in public health surveillance. The success of our CNN architectures in achieving high classification accuracy establishes a strong case for the broader application of deep learning technologies in public health initiatives. Moreover, the adaptability and scalability of our proposed models suggest their potential deployment in various other domains requiring similar surveillance measures, such as environmental monitoring, security, and beyond.

The research outcomes contribute valuable insights into the design and implementation of neural networks, particularly in how layer configurations and training strategies can be optimized for specific tasks. This work lays a substantial groundwork for the integration of machine learning technologies into public health systems, offering a scalable tool for enhancing disease prevention strategies through automated compliance monitoring.

Looking forward, the promising results from this study open several avenues for further research and development. There is a clear opportunity to extend this work by exploring the detection capabilities of CNNs under more varied and challenging scenarios, such as different lighting conditions, angles, or obscured faces. Enhancing the model's ability to accurately identify face masks in such conditions would greatly increase its utility in real-world settings.

Additionally, future work could explore the integration of this technology with other biometric recognition systems to develop a more comprehensive monitoring solution. Such systems could offer multi-faceted benefits, from enhancing security protocols to improving personalized health tracking and compliance.

Furthermore, advancing the interpretability of these CNN models is crucial for their acceptance and trust among users, particularly in sensitive applications like public health. Efforts to make the models' decision-making processes more transparent and understandable to users could facilitate wider adoption and acceptance, especially in regulatory environments.

Finally, extending our models to accommodate real-time processing without significant resource expenditure remains a critical challenge. Optimizing the models to reduce their computational demands while maintaining high accuracy would allow for deployment on a larger scale, including in mobile and edge-computing devices, thereby broadening their applicability.

In conclusion, our research not only highlights the effectiveness of CNNs in face mask detection but also opens up expansive possibilities for their application in enhancing public health and safety. Future research directions, aimed at overcoming current limitations and expanding capabilities, promise to propel this technology to the forefront of public health tools, paving the way for smarter, more reliable public health management systems.

Author Contributions: Conceptualization, A.K., O.P. and M.M.; data curation, A.K.; formal analysis, A.K., O.P., M.M. and I.K.; funding acquisition, K.A.-H.; investigation, A.K., O.P., K.A.-H., M.M. and I.K.; methodology, A.K., O.P., K.A.-H. and I.K.; project administration, A.K. and O.P.; resources, A.K. and K.A.-H.; software, A.K. and O.P.; supervision, O.P., M.M. and I.K.; validation, A.K. and I.K.; visualization, A.K. and M.M.; writing—original draft, A.K.; writing—review and editing, A.K., O.P., K.A.-H., M.M. and I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported in part by Research Fund 2023-24-1004 from the Rochester Institute of Technology—Dubai (RIT-Dubai).

Data Availability Statement: The dataset utilized in this research study is publicly available on Kaggle at https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset, accessed on 16 July 2024.

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

References

1. Leung, N.H.; Chu, D.K.; Shiu, E.Y.; Chan, K.H.; McDevitt, J.J.; Hau, B.J.; Yen, H.L.; Li, Y.; Ip, D.K.; Peiris, J.; et al. Respiratory Virus Shedding in Exhaled Breath and Efficacy of Face Masks. *Nat. Med.* **2020**, *26*, 676–680. [CrossRef] [PubMed]
2. Teboulbi, S.; Messaoud, S.; Hajjaji, M.A.; Mtibaa, A. Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention. *Sci. Program.* **2021**, *2021*, 8340779:1–8340779:21. [CrossRef]
3. Bhamare, D.; Suryawanshi, P. Review on Reliable Pattern Recognition with Machine Learning Techniques. *Fuzzy Inf. Eng.* **2018**, *10*, 362–377. [CrossRef]
4. Cai, Z.; He, Z.; Guan, X.; Li, Y. Collective Data-Sanitization for Preventing Sensitive Information Inference Attacks in Social Networks. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 577–590. [CrossRef]
5. Zheng, X.; Cai, Z. Privacy-Preserved Data Sharing Towards Multiple Parties in Industrial IoTs. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 968–979. [CrossRef]
6. Suresh, K.; Palangappa, M.; Bhuvan, S. Face Mask Detection by Using Optimistic Convolutional Neural Network. In Proceedings of the 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 20–22 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1084–1089.
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification With Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*. Available online: http://www.cs.utoronto.ca/~kriz/imagenet_classification_with_deep_convolutional.pdf (accessed on 16 July 2024). [CrossRef]
8. Kaur, G.; Sinha, R.; Tiwari, P.K.; Yadav, S.K.; Pandey, P.; Raj, R.; Vashisth, A.; Rakhra, M. Face Mask Recognition System using CNN Model. *Neurosci. Inform.* **2022**, *2*, 100035. [CrossRef]
9. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
10. Talahua, J.S.; Buele, J.; Calvopiña, P.; Varela-Aldás, J. Facial Recognition System for People With and Without Face Mask in Times of the COVID-19 Pandemic. *Sustainability* **2021**, *13*, 6900. [CrossRef]
11. Kumaraswamy, M.; Shaji, J.; Sowmya, V.; Paul, P. General Awareness Regarding Face Masks to Combat COVID-19: A Comprehensive Review. *Int. J. Pharm. Sci. Rev. Res.* **2021**, *67*, 24–29.
12. Katz, R.; Vaught, A.; Simmens, S.J. Local Decision Making for Implementing Social Distancing in Response to Outbreaks. *Public Health Rep.* **2019**, *134*, 150–154. [CrossRef]
13. Zhao, Z.Q.; Zheng, P.; Xu, S.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef]
14. Nieto-Rodríguez, A.; Mucientes, M.; Brea, V.M. System for Medical Mask Detection in the Operating Room Through Facial Attributes. In Proceedings of the Pattern Recognition and Image Analysis—7th Iberian Conference, IbPRIA 2015, Santiago de Compostela, Spain, 17–19 June 2015; Proceedings; Lecture Notes in Computer Science; Paredes, R., Cardoso, J.S., Pardo, X.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9117, pp. 138–145.
15. Vijitkunsawat, W.; Chantngarm, P. Study of the Performance of Machine Learning Algorithms for Face Mask Detection. In Proceedings of the 5th International Conference on Information Technology (InCIT), Chonburi, Thailand, 21–22 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 39–43.

16. Savvopoulos, A.; Kanavos, A.; Mylonas, P.; Sioutas, S. LSTM Accelerator for Convolutional Object Identification. *Algorithms* **2018**, *11*, 157. [CrossRef]
17. Chowdary, M.K.; Nguyen, T.N.; Hemanth, D.J. Deep Learning-Based Facial Emotion Recognition for Human–Computer Interaction Applications. *Neural Comput. Appl.* **2023**, *35*, 23311–23328. [CrossRef]
18. Nagrath, P.; Jain, R.; Madan, A.; Arora, R.; Kataria, P.; Hemanth, J. SSDMNV2: A Real Time DNN-Based Face Mask Detection System Using Single Shot Multibox Detector and MobileNetV2. *Sustain. Cities Soc.* **2021**, *66*, 102692. [CrossRef]
19. Bu, W.; Xiao, J.; Zhou, C.; Yang, M.; Peng, C. A Cascade Framework for Masked Face Detection. In Proceedings of the International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), Ningbo, China, 19–21 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 458–462.
20. Militante, S.V.; Dionisio, N.V. Real-Time Facemask Recognition With Alarm System Using Deep Learning. In Proceedings of the 11th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 8 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 106–110.
21. Kanavos, A.; Papadimitriou, O.; Kaponis, A.; Maragoudakis, M. Enhancing Disease Diagnosis: A CNN-Based Approach for Automated White Blood Cell Classification. In Proceedings of the IEEE International Conference on Big Data (BigData), Sorrento, Italy, 15–18 December 2023; pp. 4606–4613.
22. Kanavos, A.; Kolovos, E.; Papadimitriou, O.; Maragoudakis, M. Breast Cancer Classification of Histopathological Images using Deep Convolutional Neural Networks. In Proceedings of the 7th IEEE South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Ioannina, Greece, 23–25 September 2022; pp. 1–6.
23. Vernikou, S.; Lyras, A.; Kanavos, A. Multiclass sentiment analysis on COVID-19-related tweets using deep learning models. *Neural Comput. Appl.* **2022**, *34*, 19615–19627. [CrossRef]
24. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. Fighting Against COVID-19: A Novel Deep Learning Model Based on YOLO-v2 With ResNet-50 for Medical Face Mask Detection. *Sustain. Cities Soc.* **2021**, *65*, 102600. [CrossRef]
25. Bhattacharya, I. HybridFaceMaskNet: A Novel Face-Mask Detection Framework Using Hybrid Approach. 2021. Available online: https://www.researchsquare.com/article/rs-476241/v1 (accessed on 16 July 2024).
26. Gupta, U.; Wu, C.J.; Wang, X.; Naumov, M.; Reagen, B.; Brooks, D.; Cottel, B.; Hazelwood, K.M.; Hempstead, M.; Jia, B.; et al. The Architectural Implications of Facebook's DNN-Based Personalized Recommendation. In Proceedings of the International Symposium on High Performance Computer Architecture (HPCA), San Diego, CA, USA, 22–26 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 488–501.
27. Bhuiyan, M.R.; Khushbu, S.A.; Islam, M.S. A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety With YOLOv3. In Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
28. Sanjaya, S.A.; Rakhmawan, S.A. Face Mask Detection Using MobileNetV2 in the Era of COVID-19 Pandemic. In Proceedings of the International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 26–27 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
29. Venkateswarlu, I.B.; Kakarla, J.; Prakash, S. Face Mask Detection Using MobileNet and Global Pooling Block. In Proceedings of the 4th Conference on Information & Communication Technology (CICT), Chennai, India, 3–5 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
30. Hussain, S.A.; Balushi, A.S.A.A. A Real Time Face Emotion Classification and Recognition Using Deep Learning Model. *J. Phys. Conf. Ser.* **2020**, *1432*, 012087. [CrossRef]
31. Ejaz, M.S.; Islam, M.R.; Sifatullah, M.; Sarker, A. Implementation of Principal Component Analysis on Masked and Non-Masked Face Recognition. In Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 3–5 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
32. Chachere, A.; Dongre, S. Real Time Face Mask Detection by using CNN. In Proceedings of the 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 22–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1325–1329.
33. Chavda, A.; Dsouza, J.; Badgujar, S.; Damani, A. Multi-Stage CNN Architecture for Face Mask Detection. In Proceedings of the 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2–4 April 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
34. Islam, M.S.; Moon, E.H.; Shaikat, M.A.; Alam, M.J. A Novel Approach to Detect Face Mask using CNN. In Proceedings of the 3rd International Conference on Intelligent Sustainable Systems (ICISS), Thoothukudi, India, 3–5 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 800–806.
35. Li, L.; Mu, X.; Li, S.; Peng, H. A Review of Face Recognition Technology. *IEEE Access* **2020**, *8*, 139110–139120. [CrossRef]
36. Saranya, G.; Sarkar, D.; Ghosh, S.; Basu, L.; Kumaran, K.; Ananthi, N. Face Mask Detection using CNN. In Proceedings of the 10th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 18–19 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 426–431.
37. Chen, Y.; Liu, S. Deep Partial Occlusion Facial Expression Recognition via Improved CNN. In Proceedings of the 15th International Symposium on Advances in Visual Computing (ISVC), San Diego, CA, USA, 5–7 October 2020; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12509, pp. 451–462.

38. Ge, C.; Peng, G.; Zhu, W.; Fan, Z.; Zhu, X.; Hu, B. Overcoming Occlusion for Robust Facial Expression Recognition using Adaptive Dual-Attention Net. In Proceedings of the 6th International Conference on Information Technologies and Electrical Engineering (ICITEE), Hunan, China, 3–5 November 2023; ACM: New York, NY, USA, 2023; pp. 368–373.
39. Desai, M.; Shah, M. An Anatomization on Breast Cancer Detection and Diagnosis Employing Multi-Layer Perceptron Neural Network (MLP) and Convolutional Neural Network (CNN). *Clin. eHealth* **2021**, *4*, 1–11. [CrossRef]
40. Smilkov, D.; Thorat, N.; Assogba, Y.; Yuan, A.; Kreeger, N.; Yu, P.; Zhang, K.; Cai, S.; Nielsen, E.; Soergel, D.; et al. TensorFlow.js: Machine Learning for the Web and Beyond. *arXiv* **2019**, arXiv:1901.05350.
41. Manaswi, N.K. Understanding and Working With Keras. In *Deep Learning with Applications Using Python*; Apress: Berkeley, CA, USA, 2018; pp. 31–43.
42. Bjorck, N.; Gomes, C.P.; Selman, B.; Weinberger, K.Q. Understanding Batch Normalization. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. Available online: https://proceedings.neurips.cc/paper/2018/hash/36072923bfc3cf47745d704feb489480-Abstract.html (accessed on 16 July 2024).
43. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
44. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks From Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
45. Pham-Hoang-Nam, A.; Le-Thi-Tuong, V.; Phung-Khanh, L.; Ly-Tu, N. Densely Populated Regions Face Masks Localization and Classification Using Deep Learning Models. In *Annals of Computer Science and Information Systems, Proceedings of the Sixth International Conference on Research in Intelligent and Computing, Thu Dau Mot, Vietnam, 3–4 June 2021*; Solanki, V.K., Quang, N.H., Eds.; Polish Information Processing Society: Warsaw, Poland, 2021; Volume 27, pp. 71–76.
46. Abirami, T.; Priakanth, P.; Madhuvanthi, T. Effective Face Mask and Social Distance Detection With Alert System for COVID-19 Using YOLOv5 Model. In *Advances in Parallel Computing Algorithms, Tools and Paradigms*; IOS Press: Amsterdam, The Netherlands, 2022; pp. 80–85.
47. Mohan, P.; Paul, A.J.; Chirania, A. A Tiny CNN Architecture for Medical Face Mask Detection for Resource-Constrained Endpoints. *arXiv* **2020**, arXiv:2011.14858.
48. Aydemir, E.; Yalcinkaya, M.A.; Barua, P.D.; Baygin, M.; Faust, O.; Dogan, S.; Chakraborty, S.; Tuncer, T.; Acharya, U.R. Hybrid Deep Feature Generation for Appropriate Face Mask Use Detection. *Int. J. Environ. Res. Public Health* **2022**, *19*, 1939. [CrossRef]
49. Fallaha, A.; Ashour, M.T.E.; Krayem, D.; Alqaraleh, S. Face Mask Detection Using Deep Learning for Public Places. 2020. Available online: https://www.set-science.com/manage/uploads/ISAS-WINTER-2022_0089/SETSCI_ISAS-WINTER-2022_0089_004.pdf (accessed on 16 July 2024).

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