

Face Mask Classification Based on the Convolutional Neural Network

Youyang Qu *

Department of Postal Engineering, Beijing University of Posts and Telecommunications, Beijing, China

* Corresponding Author Email: quyouyang@ldy.edu.rs

Abstract. The worldwide emergence of the COVID-19 pandemic has instigated widespread anxiety. In an effort to curb the virus's transmission, the adoption of mask-wearing has steadily become more prevalent. Numerous public facilities now require individuals to wear masks upon entry and exit. Within this framework, the effective and economical identification of mask-wearing individuals has become a growing priority, and deep learning has demonstrated its capability to address this need. Utilizing a dataset consisting of 440 provided images, the current experiment employed a Convolutional Neural Network (CNN) model to determine the presence or absence of masks on individuals. The experimental process encompassed dataset segmentation, CNN model construction, model training, and performance evaluation with the objective of attaining the best-performing model. The research revealed that, within a specific range, an increase in the number of layers corresponded to an enhancement in the CNN's performance. Remarkably, at 13 layers, the CNN achieved its highest accuracy, reaching 93.18%, with a recorded lowest loss value of 0.1608. Despite this, the CNN model's accuracy decreased as a result of an increase in the convolutional kernel size, which was accompanied by variations in the loss value. After multiple iterations of experimentation, the optimal convolutional kernel size was determined to be 3x3. In the quest for a more effective means of detecting mask compliance in public spaces, this study underscores the potential of deep learning techniques, particularly CNNs, and highlights the significance of carefully configuring model architecture for optimal results.

Keywords: Deep learning, CNN, face mask classification.

1. Introduction

COVID-19, identified in 2019, rapidly became a global pandemic by 2020. Originating as an unknown illness, it quickly revealed itself as a highly contagious disease targeting the human respiratory system. This virus is mostly spread through the air, where it becomes encapsulated in microscopic droplets and is released when an infected person talks, sneezes, or coughs. These droplets can infect close contacts, making person-to-person transmission especially prevalent in close-knit settings. Furthermore, another alarming characteristic of this virus is its ability to survive on various surfaces for extended periods. This means that areas or objects frequented by an infected individual can act as potential transmission points, thereby amplifying the risk [1]. Given its potent combination of rapid transmission and the capacity to cause severe respiratory distress, it's no surprise that the global populace was consumed by a wave of fear and uncertainty.

With this newfound knowledge and the urgent need for preventive measures, public health advisories across the globe began mandating mask usage in public spaces. The challenge, however, lay in monitoring and ensuring compliance. Traditional methods involving human surveillance were not only labor-intensive but also presented a risk of exposure. This predicament opened the door for technology-driven solutions, with Artificial Intelligence (AI) being at the forefront.

Deep learning, a subfield of AI, has seen exponential advancements in recent years, proving its mettle across various applications [2, 3]. Historically, tasks such as facial recognition relied on manually curated image features. However, with the advent of deep learning, these tedious processes became obsolete. The technology enabled the automatic extraction and learning of features, thereby increasing efficiency and accuracy [4]. Within the realm of deep learning, Convolutional Neural Networks (CNN) emerged as the gold standard for image-related tasks. Their architecture, designed

to process data with a grid-like topology, is particularly adept at handling images. CNNs can autonomously identify and extract intricate patterns from images, making them ideal for the task at hand [5]. Awdhutrao et al suggested that the purpose of facial recognition can be achieved by utilizing techniques such as Principal Component Analysis (PCA) and K-Nearest Neighbor (KNN) for the extraction and comparison of facial features [6]. In addition, Wang et al. has employed ABASNet and H-Softmax to achieve facial recognition [7], with an accuracy rate that can reach 99.5%. While their impressive performance is undeniable, in today's prevalent circumstances, numerous people might choose to don masks. In these situations where masks are worn, the capability of facial recognition models to differentiate and anticipate might face challenges. Therefore, in the aspect of recognizing whether a face is wearing a mask, the robust feature extraction capability of CNN is highly suitable for addressing such issues. This paper delves into the utilization of CNN for the detection of face masks on individuals. This study begins by outlining the fundamental architecture and principles of CNNs, followed by a detailed exploration of how they can be tailored for the specific task of mask detection. Through rigorous testing, the model's performance was evaluated, pinpointing areas of excellence and those that require improvement. Building on these insights, this study subsequently ventures into optimization techniques, aiming to refine the model for even greater accuracy. The culmination of this research provides a comparative analysis, juxtaposing the optimized CNN model against its initial iteration and traditional methods.

2. Method

2.1. Dataset Preparation and Preprocessing

Before conducting the experiment, the dataset was prepared in advance. A high-quality dataset can reduce experimental errors and improve the efficiency of experimentation. The dataset file serves as the basis for this paper's dataset [8]. The file contains the with_mask and without_mask folders, each containing 220 images, representing images of faces with masks and faces without masks, respectively (Table 1). In addition, each face image is in PNG format with RGB color channel. Specific examples of the images are shown in Fig. 1 and Fig. 2, with the first image located in the with_mask folder and the second image located in the without_mask folder.

Table 1. The detailed information related to the collected dataset.

Label	Training	Validation	Testing	Total
with_mask	44	132	44	220
without_mask	44	132	44	220

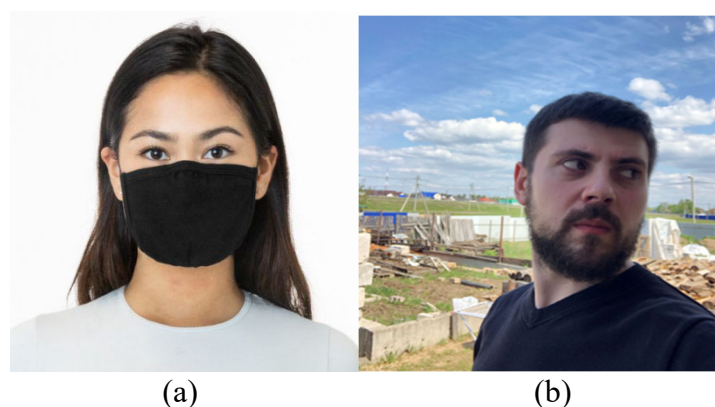


Figure 1. (a) sample data with mask (b) sample data without mask [8].

After obtaining a suitable dataset for this experiment, the paper will proceed to discuss the data preprocessing. In this experiment, three files, namely, the training set, validation set, and test set, have been created to store images. The distribution of images among these three sets is in the ratio of 3:1:1, and there is no overlap between the images in these sets. Given the relatively small number of images

in the dataset, TensorFlow's ImageDataGenerator has been employed to perform data augmentation by applying techniques such as rotation, translation, and scaling to enhance dataset diversity. The objective is to augment the training dataset and ensure comprehensive model training. Subsequently, a data generator loads the image data and feeds it into the model for training and evaluation. In this experiment, due to the varying image sizes in the dataset, the data generator is used to resize the images to 150×150 . Furthermore, binary classification is applied in this experiment as the classification mode. To prevent the model training from being too short or too long, the experiment sets the number of images per batch for training to 20.

2.2. CNN

In the 1980s and 1990s, CNNs began to be researched by people. Compared to ordinary neural networks, CNNs have an added layer structure. Additionally, CNN models, when being trained, encompass more specialized layer types, such as convolutional layers and pooling layers, which can only be found in CNNs. Consequently, CNNs are superior to ordinary neural networks in the area of image recognition [9]. In this work, after data preprocessing, it will be transmitted to the CNN for training and validation, and the model's performance will be evaluated through a test dataset. For CNNs, convolutional layers and pooling layers are the most critical components. Convolutional layers perform point-wise multiplication operations on different regions of the input data using a sliding window of convolutional kernels [10], which can be viewed as a filtering process. Through this process, convolutional layers can detect various features. After multiple iterations of training, the weights of the convolutional kernels are optimized, thereby improving the model's performance. Pooling layers, on the other hand, reduce the data's dimensionality and computation while preserving essential information [11], thus reducing the likelihood of model overfitting. The initial CNN structure for this experiment is shown in Fig. 2, displaying only convolutional and pooling layers (this figure was created using the online drawing tool NN-SVG).

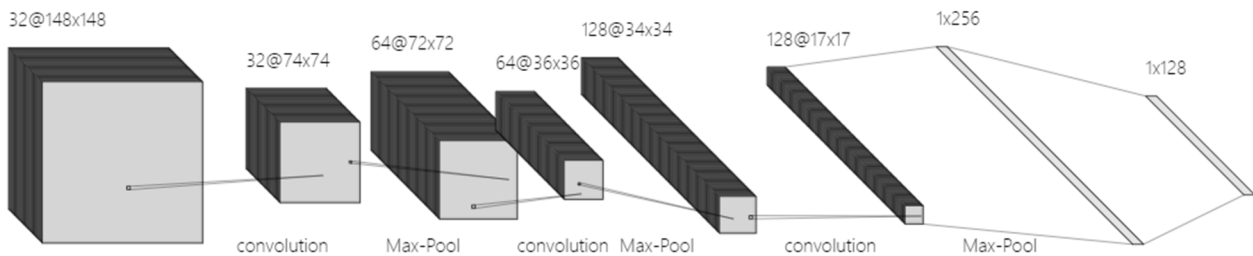


Figure 2. The structure of the CNN established in this study (Photo credit: Original).

In addition, this experiment includes two Dense layers and one Dropout layer (the Flatten layer is not counted). Therefore, the initial number of convolutional layers is 9, with a 3×3 kernel size. Regarding the activation functions, this experiment primarily utilizes ReLU and sigmoid. Two activation functions are as follows.

$$f(x) = \max(0, x). \quad (1)$$

$$f(x) = \frac{1}{1+e^{-x}}. \quad (2)$$

ReLU is employed in both the convolutional layers and fully connected layers because of its ability to effectively model data and enhance performance. In this study, as there are only two categories, which are wearing masks and not wearing masks, sigmoid is used in the final Dense layer to address the binary classification problem.

2.3. Implementation Details

In addition to the aforementioned information, this section will provide further details. The experiment was conducted using the open-source TensorFlow framework. In order to ensure

compatibility with the Python 3.6 environment, TensorFlow CPU version 1.14 was ultimately selected and installed. To prevent excessively long training times, the number of epochs for this experiment was set to 10. Regarding the model's parameter configuration, given that the problem addressed in this experiment pertains to binary classification, the `binary_crossentropy` loss function was chosen. The mathematical formula for binary cross-entropy loss is as follows (where x represents the true label, which can be 1 or 0; and y represents the model's predicted probability, ranging from 0 to 1):

$$f(x, y) = -[x \log y + (1 - x) \log(1 - y)]. \quad (3)$$

The loss function computes the difference between the model's output and the actual labels, which causes the entire model training process to be an iterative reduction of the loss function. Additionally, the optimization algorithm employed in this experiment is the Adam optimizer. Furthermore, in this experiment, the model's evaluation metrics consist of accuracy and loss value. A certain level of analysis is carried out to gauge the effect of parameter variations on the model.

3. Results and Discussion

In this experiment, the initial CNN model was configured with 9 layers, and the convolution kernel size was set to 3×3 . In order to make sure that other factors wouldn't affect the experimental results, the approach of controlling variables was used to look into how the quantity of layers and the size of the kernel affected the model's performance. The quantity of layers was independently set to 9, 11, and 13 while the other parameters stayed the same for examining the layer's influence count on the CNN. The accuracy and loss figures of the model are currently shown in Fig. 3.

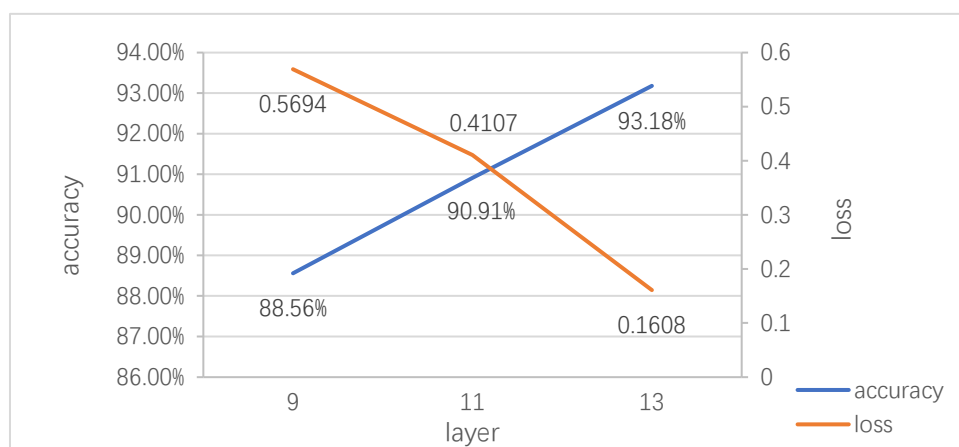


Figure 3. Different Layer Architectures and Their Impact on CNN Models (Photo credit: Original)

The convolution kernel size was then adjusted to 2×2 , 3×3 , 4×4 , and 5×5 in this experiment while the other parameters remained constant in order to examine the effect of the convolution kernel size on CNN. The experimental results are shown in Fig. 4.

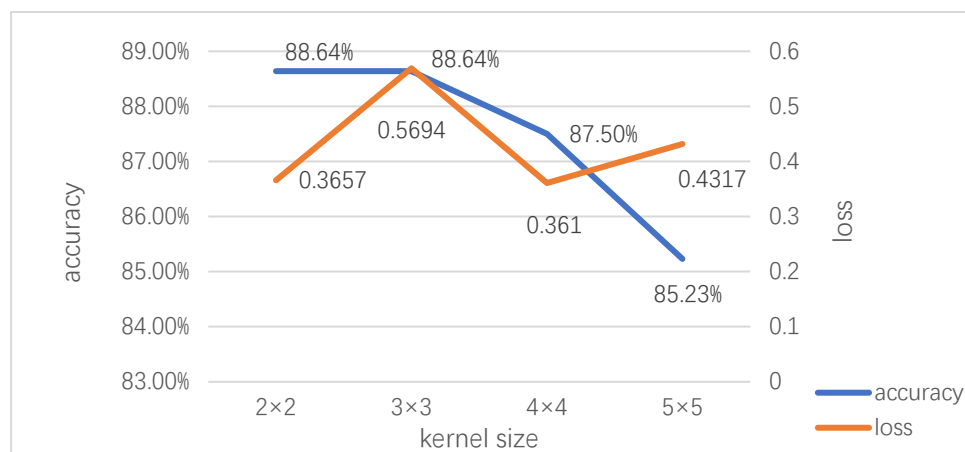


Figure 4. The impact of Convolutional Kernel Size on CNN Models (Photo credit: Original)

Observing the provided images, it can be observed that the accuracy of the CNN model gradually improves as the number of layers increases. Simultaneously, the model's loss gradually decreases. However, in the course of altering the convolution kernel size, differing results are evident. With the continuous increase in the size of the convolution kernel, the accuracy of the CNN model consistently declines, while the loss fluctuates within the range of 0.36 to 0.57.

Based on an analysis of the aforementioned experimental results, the following conclusions are summarized. In terms of CNN, increasing the number of layers allows the model to delve deeper into learning abstract features from the data, which is conducive to enhancing accuracy. The decrease in loss also reflects that the CNN model, with an accumulation of layers within a certain range, fits the data better [12]. On the other hand, enlarging the convolution kernel size may cause the CNN model to reduce its ability to extract fine or local features. Furthermore, an excessively large convolution kernel might introduce noise, diverting the model's attention towards irrelevant features, thereby reducing accuracy. The loss values fluctuate, indicating that noise may have an impact on loss values and that the size of the convolution kernel may not have a major effect on the model's performance [13]. Consequently, it takes several trials to determine the ideal convolution kernel size. This article confirms that 3×3 is the ideal convolution kernel size.

4. Conclusion

Consequently, this research indicates that augmenting the model's accuracy and reducing loss can be achieved by incorporating additional convolutional layers within a defined range. However, an escalation in the count of convolutional layers leads to a decline in model accuracy, accompanied by fluctuations in loss values. This trend might be attributed to the fact that as the convolutional kernel size increases, its efficacy in feature extraction diminishes, potentially introducing noise into the model. The best number of convolutional layers, according to the experiments, is 13. However, a 3×3 convolution kernel size turns out to produce better model results. This article offers insightful information on the variables affecting CNN's effectiveness in picture classification jobs. For CNN models, parameter design is important in real-world applications. Consequently, there will be limitations in this experiment's detailed design for future research. Only the impact of specific parameters on CNN was examined in this article. In real-world scenarios, there may be more complicated situations, so it's crucial to examine the combined impact of multiple parameters on CNN in future experiments.

References

- [1] Lakhera, A., Jain, P., Gajjar, R., & Patel, M. Face Mask Detection for Preventing the Spread of Covid-19 using Knowledge Distillation. In International Conference on Computational Performance Evaluation (ComPE) (pp. 815-819). Shillong, India, 2021. DOI: 10.1109/ComPE53109. 2021. 9752237.

- [2] Alonso, M. N., Batres-Estrada, G., & Moulin, A. Deep learning in finance: Prediction of stock returns with long short-term memory networks. *Big data and machine learning in quantitative investment*, 1, 2018, 251 - 277.
- [3] Qiu, Y., Yang, Y., Lin, Z., Chen, P., Luo, Y., & Huang, W. Improved denoising autoencoder for maritime image denoising and semantic segmentation of USV. *China Communications*, 17 (3), 2020, 46 - 57.
- [4] Ejaz, M. S., & Islam, M. R. Masked Face Recognition Using Convolutional Neural Network. In *International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1 - 6). Dhaka, Bangladesh, 2019. DOI: 10.1109/STI47673. 2019. 9068044.
- [5] Han, J. Algorithm design and application research of English picture recognition intelligent system based on convolutional neural network. In *IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)* (pp. 685-690). Jilin, China, 2023. DOI: 10.1109/ICCECT57938. 2023. 10140637.
- [6] Sambhe, A. A., & Deorankar, A. V. Face Detection and Recognition System. In *4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (pp. 1175-1179). Greater Noida, India, 2022, DOI: 10.1109/ICAC3N56670.2022.10074142.
- [7] Wang, F., Xie, F., Shen, S., Huang, L., Sun, R., & Yang, J. L. A Novel Multiface Recognition Method with Short Training Time and Lightweight Based on ABASNet and H-Softmax. *IEEE Access*, 8, 175370 - 175384, 2020. DOI: 10.1109/ACCESS. 2020. 3026421.
- [8] Makwana, D., & Wadhwani, B. Face Mask Detection [Data set]. Kaggle, 2020, <https://www.kaggle.com/datasets/dhruvmak/face-mask-detection>.
- [9] Dai, D. An Introduction of CNN: Models and Training on Neural Network Models. In *International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)* (pp. 135-138), 2021. Shanghai, China. DOI: 10.1109/ICBAR55169. 2021. 00037.
- [10] Amin, M., & Adiono, T. Area Optimized CNN Architecture Using Folding Approach. In *International Conference on Electrical Engineering and Informatics (ICEEI)* (pp. 206 - 209), 2019. Bandung, Indonesia. DOI: 10.1109/ICEEI47359. 2019. 8988879.
- [11] More, Y., Dumbre, K., & Shiragapur, B. Horizontal Max Pooling a Novel Approach for Noise Reduction in Max Pooling for Better Feature Detect. In *International Conference on Emerging Smart Computing and Informatics (ESCI)* (pp. 1-5). Pune, India, 2023. DOI: 10.1109/ESCI56872. 2023. 10099648.
- [12] Benkendorf, D. J., & Hawkins, C. P. Effects of sample size and network depth on a deep learning approach to species distribution modeling. *Ecological Informatics*, 60, 2020, 101137. <https://doi.org/10.1016/j.ecoinf.2020.101137>.
- [13] Song, H., Kim, M., Park, D., Shin, Y., & Lee, J. -G. Learning from Noisy Labels with Deep Neural Networks: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34 (11), 8135 - 8153, 2023. <https://doi.org/10.1109/TNNLS. 2022. 3152527>.