Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 1 (2024) 1 - 9 e-ISSN: 2580-0760

Image Preprocessing Approaches Toward Better Learning Performance with CNN

Dhimas Tribuana[1], Hazriani[2*], Abdul Latief Arda[3]
[1,2,3]Departement of Computer System, Handayani University Makassar, Makassar, Indonesia
[1]MNC Bank, Cabang Makassar, Indonesia
[1]d.tribuana@gmail.com, [2]hazriani@handayani.ac.id, [3]abdullatief@handayani.ac.id

Abstract

Convolutional neural networks (CNNs) are at the forefront of computer vision, relying heavily on the quality of input data determined by the preprocessing method. An excessive preprocessing approach will result in poor learning performance. This study critically examines the impact of advanced image preprocessing techniques on convolutional neural networks (CNNs) in facial recognition. Emphasizing the importance of data quality, we explore various preprocessing approaches, including noise reduction, histogram equalization, and image hashing. Our methodology involves feature visualization to improve facial feature discernment, training convergence analysis, and real-time model testing. The results demonstrate significant improvements in model performance with the preprocessed data set: average precision, recall, precision, and F1 score enhancements of 4.17%, 3.45%, 3.45%, and 3.81%, respectively. Furthermore, real-time testing shows a 21% performance increase and a 1.41% reduction in computing time. This study not only underscores the effectiveness of preprocessing in boosting CNN capabilities, but also opens avenues for future research in applying these methods to diverse image types and exploring various CNN architectures for a complete understanding.

Keywords: *convolutional network; deep learning; face recognition; advanced preprocessing; classification*

How to Cite: D. Tribuana, Hazriani, and A. L. Arda, "Image Preprocessing Approaches Toward Better Learning Performance with CNN", *J. RESTI (Rekayasa Sist. Teknol. Inf.),* vol. 8, no. 1, pp. 1 - 9, Jan. 2024.
DOI: *https://doi.org/10.29207/resti.v8i1.5417*

1. Introduction

In recent years, the field of computer vision has undergone remarkable advances, largely attributed to the efficacy of convolutional neural networks (CNNs) in various image analysis tasks. These tasks range from image classification and object detection to image segmentation and style transfer [1] –[3]. However, the performance of CNNs is highly dependent on the quality and preparation of the input data. As image data sets become more diverse and complex, the role of image pre-processing techniques becomes increasingly vital to extract meaningful features and patterns [4] - [6].

Image preprocessing encompasses a spectrum of operations and transformations applied to raw image data prior to its use in CNN model training [7]. These operations serve multiple purposes, including noise reduction, contrast enhancement, feature extraction, and data enhancement [8]. Properly designed and implemented preprocessing steps can significantly impact CNN performance, leading to improved accuracy, faster convergence, and enhanced generalization [9].

Image data, in its raw form, can be noisy, inconsistent, and contain artifacts that hinder the learning process of CNNs. Noise, introduced during image acquisition or transmission, can obscure important features and mislead the network [10]. Uneven lighting conditions and varying contrasts can also impede accurate feature extraction, adversely affecting the model's ability to discriminate between different classes.

Additionally, the presence of irrelevant or redundant information can increase model complexity and potentially lead to overfitting. Image preprocessing techniques address these challenges by applying a sequence of transformations that improve the quality and relevance of the data. These techniques can be broadly categorized into noise reduction methods, contrast enhancement techniques, feature extraction approaches, and data enhancement strategies. Each

Received: 06-09-2023 | Accepted: 07-01-2024 | Published Online: 13-01-2024

category addresses specific aspects of the input data, contributing to the overall improvement in CNN performance [11].

The background of this research accentuates the intricacies of the problem at hand. Raw image data often contains noise, irregularities, and artifacts that hinder CNN learning, affecting feature extraction, accuracy, and generalization. Uneven lighting, contrast issues, and irrelevant data further compound these challenges, affecting the network's ability to distinguish between different classes.

The novelty of our research lies in its unique approach to image pre-processing for CNNs in facial recognition. We integrate a distinctive combination of preprocessing techniques, including noise reduction, histogram equalization, background removal, and image hashing, which have not been collectively explored in prior studies. Our comprehensive evaluation framework, which includes feature visualization, training convergence analysis, and real-time model testing, offers a detailed understanding of their impacts.

This research is imperative, as it not only identifies the limitations of raw image data, but also pioneers advanced solutions to enhance CNN performance. Building on previous related research, our study integrates novel preprocessing methods, demonstrating substantial improvements in accuracy and real-world applicability. By addressing existing gaps in preprocessing techniques and showcasing tangible results, this research contributes to a broader understanding of optimizing CNN models, paving the way for enhanced facial recognition systems.

In our review of the research, we found a first-of-its-kind face recognition system that uses an innovative mix of hybrid features, such as the pyramid gradient histogram (PHOG), the Edge Histogram Descriptor (EHD), and the Local Binary Pattern (LBP), to pull out different characteristics from high-quality facial images [12]–[14]. Notable contributions include a creative way to improve search results using computational neural networks, a group of useful hybrid features, and a detailed performance analysis that shows that the improvement method works within the limits of the database. The system incorporates an automated evaluation function that uses the Cuckoo Search Optimization technique, strategically selecting hybrid features to efficiently extract vital information, thus substantively improving the computational efficiency and precision of face recognition systems [15].

Data preprocessing is the process of preparing data for processing using machine learning or deep learning. Before heading to the processing stage, the raw data will first be processed. Some common preprocessing techniques are as follows: grayscale conversion, edge detection [16], gaussian smoothing [17], contrast enhancement, binarization [18], facial cropping, image resizing [19], face alignment, etc. Image preprocessing reduces the processing time and increases the likelihood of a perfect match. Face images are pre-processed to meet feature extraction requirements [20] - [22].

In another review of the literature, a pseudorandom pixel placement method is proposed for face recognition preprocessing, with the aim of improving accuracy. The results show that the use of the method improves accuracy by 63.76% compared to without it, indicating its potential to improve face recognition performance [23]. A study evaluated color-to-grayscale algorithms in edge detection, revealing that lightness performs better than other methods in grayscale images [24].

Collected images with high relative similarity to ensure the accuracy of the deep neural network's predictions. To avoid bias, the dataset needs to be cleaned to remove similar images. The application of a hash function to each image calculates a numerical representation of the image. Images with the same hash are considered duplicates [25]. The combination of contrast-limited Adaptive Histogram Equalization (CLAHE) and Discrete Wavelet Transform (DWT) effectively enhances low-frequency components while keeping high-frequency components unchanged. The Discrete wavelet transform enhanced contrast-limited adaptive histogram equalization (DWT E-CLAHE) is computationally simple and faster than gradient-based methods and feature descriptors, making it an effective illumination pre-processing method for face recognition [26].

2. Research Methods

Our research methodology adopts a systematic approach to improve image quality for an advanced face recognition system. In the initial phase, Python code is implemented to meticulously select the best images based on criteria such as blurriness, exposure, and facial detectability. Subsequent stages involve targeted image selection through bespoke scripts that use image hashing to eliminate duplicates, address exposure variations with adaptive brightness adjustments, systematic resizing using OpenCV, and precise facial alignment through the Dlib library. These steps collectively contribute to the refinement of the dataset, highlighting the importance of high-quality images. The methodology also includes noise reduction techniques to improve the overall quality of the dataset. Hardware and software specifications, along with pre-processing results, set the stage for subsequent training and evaluation phases of the face recognition system.

In the first stages, we used a systematic approach to enhance the quality of the images for subsequent analysis. The Python code implemented aimed to select superior images by assessing their quality based on various criteria. Images showing blurriness, underexposure, overexposure, or lack detectable faces were excluded from the final selection. To achieve this, the code used functions to evaluate image characteristics, such as blurriness, exposure levels, and the presence of faces, using Haar cascades. The images that met established quality standards were then retained, while those that were short-lived were ignored. This meticulous image selection process was integral to constructing an improved dataset, ensuring that only high-quality images were considered for subsequent stages of the face recognition system's development.

In the subsequent phase of our methodology, we focused on refining the image dataset through a targeted image selection process, facilitated by a custom Python script. This script harnessed the concept of image hashing, specifically the 'average hash' technique from the 'image hash' library, to generate unique hash keys for each image. These keys served as fingerprints for image identification, enabling the systematic detection and elimination of duplicate images. By meticulously comparing hash keys, the script efficiently traversed the dataset, selectively copying distinct images to an output directory. The integration of the 'tqdm' library introduced progress bars for transparent tracking of image processing, and the script adeptly managed exceptions, ensuring robust execution. This process ensured the enhanced quality and uniqueness of the dataset, laying a strong foundation for subsequent stages of the face recognition system's development.

In summary, our second methodological stage exemplified an intricate approach to image selection, utilizing advanced techniques to curate a refined data set. Using image hashing, progress bars, and error handling, the custom Python script was able to improve the quality of the dataset. This showed how important clear, high-quality images are in improving the performance of the face recognition system as a whole.

In the subsequent stage, we engaged a customized Python script to address varying image exposure levels within the dataset, optimizing image quality for further analysis. This script capitalized on techniques such as brightness adjustment to harmonize exposure discrepancies. The script systematically processed the images, evaluating their exposure characteristics to determine whether they were under or overexposed.

For underexposed images, the brightness was increased using a tailored factor, while overexposed images underwent a brightness reduction. These adaptive adjustments are intended to correct exposure imbalances and enhance image quality. In the cases where no adjustment was needed, the images were directly copied to the output directory. By applying these image enhancement strategies, the script demonstrated its capability for standardizing exposure levels, resulting in an improved dataset for subsequent stages of the face recognition system's development.

In the subsequent stage of our methodology, a custom Python script was used to systematically resize images, aligning them with predetermined dimensions to enhance consistency and optimize subsequent analysis. The script utilized OpenCV to meticulously process images within the dataset, offering the flexibility to define a specific width or height for resizing. By determining the aspect ratio and applying the appropriate resizing technique, the script ensured that the images were within the designated dimensions, facilitating uniformity and simplifying further processing. The resulting resized images were thoughtfully aligned to a standardized scale, thereby fortifying the quality and readiness of the dataset for integration into the face recognition system. This stage underscored the importance of homogenizing the image dimensions to facilitate robust and consistent performance throughout the training and evaluation phases of the system.

In the subsequent stage, our methodology focused on precise facial alignment to further optimize the dataset for the face recognition system. This phase involved employing the Dlib library for face detection and alignment. The script iterated through the dataset, detecting faces within each image, and subsequently generating aligned face images. Using the frontal face detector model and shape predictor, the script located facial landmarks to determine the accurate facial region. A mask was applied to isolate the face, allowing the extraction and alignment of the facial region. The aligned images were then saved to the designated output directory. This stage of alignment significantly contributed to improving the consistency and quality of the dataset, ensuring that facial features were accurately centered and aligned for improved performance within the face recognition system.

In the subsequent stage, our methodology focused on enhancing the quality of the dataset by minimizing image noise through a customized noise reduction process. Using a custom Python script, the images within the dataset were subjected to a comprehensive noise reduction technique. The script systematically processed each image, applying Gaussian blur to mitigate noise and enhance image clarity. This operation was designed to alleviate imperfections and irregularities in the image, resulting in a cleaner and more refined dataset.

By executing this noise reduction process, the script effectively optimized the overall quality of the dataset, thus paving the way for increased accuracy and reliability within the face recognition system's subsequent training and evaluation phases. The hardware and software requirements used in this research consist of an AMD Ryzen 9 5900HS CPU, 16.0 GB of RAM, the Windows 11 Home Single Language Version 22H2 Operating System, Anaconda Navigator 2.4.2, Python 3.9, Tensorflow 2.13, and OpenCV 4.8.0.74. The result of the complete preprocessing can be seen in Figure 1, which is on the left the original image and on the right the result.

Figure 1 visually represents the impact of preprocessing on image quality. On the left is the original image, and on the right is the result after preprocessing. The comparison highlights the improvements achieved through the preprocessing steps, showcasing the significance of each enhancement.

Figure 1. Image comparison on original and pre-processing data sets

For evaluation, two distinct datasets were meticulously prepared for testing. The first dataset was subjected to advanced preprocessing techniques elucidated earlier, including stages such as image enhancement, quality control, alignment, and noise reduction. On the other hand, the second dataset underwent standard preprocessing methods, including grayscale conversion, resizing, data normalization, and data augmentation.

Subsequently, both data sets were trained using an identical convolutional neural network (CNN) architecture. The architecture featured multiple convolutional layers, followed by maximum-pooling and dropout layers for regularization. Batch normalization was integrated to improve convergence, while fully connected layers closed the network, culminating in softmax activation for classification [27].

The datasets were appropriately processed and formatted, with facial images normalized and their corresponding identities encoded with a hot ring. The datasets were then divided into training and testing sets using a standard 80-20 split. This comprehensive evaluation strategy allowed for direct comparison of the impact of advanced preprocessing techniques versus conventional methods on the performance of the trained CNN model in the context of face recognition.

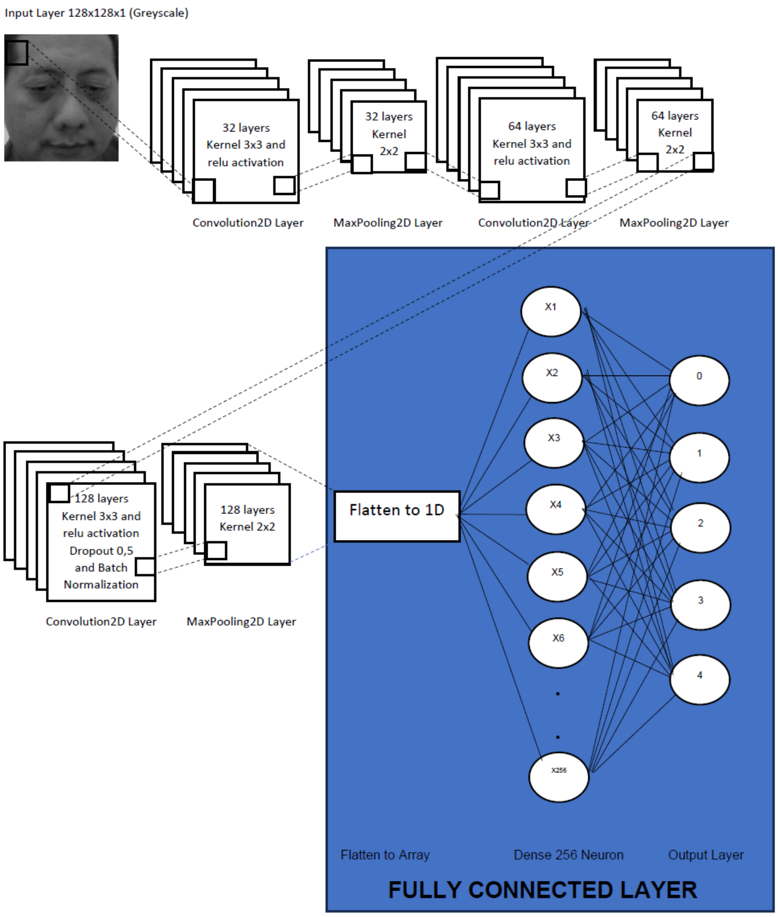


Figure 2. CNN Architecture Used

Figure 2 provides a visual representation of the CNN architecture used in the study. It consists of three convolutional layers with 32, 64, and 128 layers, respectively, each with a 3x3 kernel size and ReLU activation. Dropout and Batch Normalization are applied to prevent overfitting. The fully connected layer includes 256 neurons, and the output layer is equipped with softmax activation. This figure serves as a reference for the architecture used during training.

3. Results and Discussions

A comprehensive evaluation strategy was employed to compare the impact of advanced preprocessing techniques with conventional methods on the performance of convolutional neural network (CNN) models for face recognition. The assessment encompassed several dimensions. The assessment encompassed several dimensions, allowing for a comprehensive understanding of the multifaceted effects of preprocessing on the model's capabilities.

3.1. Accuracy and Performance

First, accuracy and performance metrics, such as accuracy, precision, recall, and F1 score, of CNN models trained on the original and preprocessed datasets were examined. These metrics shed light on the influence of preprocessing techniques on the models' classification capabilities and overall effectiveness. We can see the results of the training on the two datasets in Table 1.

Table 1. Performance of Training Results

Dataset	Epoch	Accuracy	Precision	Recall	F1 Score
Original	30	0,97	0,98	0,98	0,98
	40	0,93	0,94	0,94	0,93
	50	0,98	0,98	0,98	0,98
Preprocessed	30	1,00	1,00	1,00	1,00
	40	1,00	1,00	1,00	1,00
	50	1,00	1,00	1,00	1,00

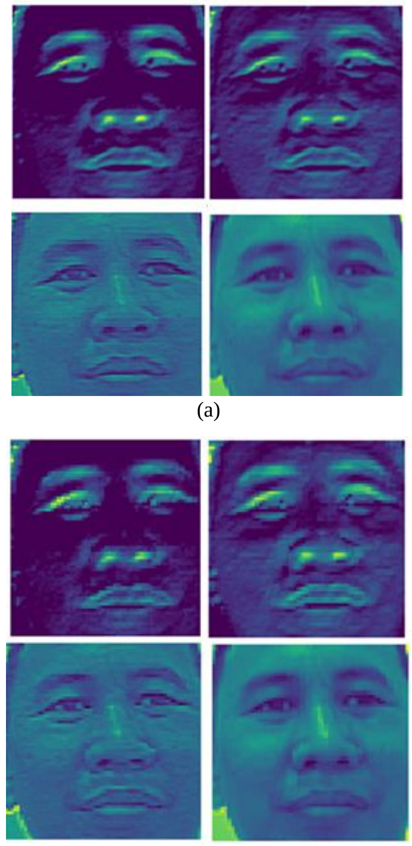
The training results of the two datasets can be compared, and it can be concluded that the training results of the preprocessed dataset show results in precision, recall, precision, and F1 score consistently at epochs 30, 40, and 50, giving better results. The results indicate that the preprocessed dataset shows an average accuracy improvement of 4.17%, an average recall improvement of 3.45%, an average precision improvement of 3.45%, and an average F1 score improvement of 3.81%.

Our study's findings on the enhanced performance of CNN models due to preprocessing techniques align with recent research in different domains. For instance, in the field of agricultural product classification, a study titled 'Performance Analysis of Deep Learning CNN Models for Variety Classification in Hazelnut' demonstrated that custom-designed CNN models, with appropriate preprocessing, could outperform pretrained models like VGG16 and ResNet50. This was evident in their improved accuracy, precision, recall, and F1 score. This evidence further validates the impact of preprocessing in improving the efficiency of CNN models in various applications [28].

3.2. Feature Visualization

Additionally, feature visualization was carried out, revealing the extracted features of different CNN layers for both datasets. When looking at how different and separate these features were, it was possible to find out if preprocessing made it easier for the network to distinguish complex facial features, which would affect the accuracy of recognition. Figures 3 and 4 show the results of the feature extraction on the Conv2D layer (a) and the Max_Pooling2D layer (b) for the original dataset and the pre-processed dataset.

The results of the extraction of the features of the two datasets can be compared and it can be concluded that the results of the extraction of the features from the preprocessed data set show better results, as can be seen in Figures 3 (a) and (b), with only the features included in the area of the extracted face landmarks. The extraction results also appear clearer and brighter. In Figures 4 (a) and (b), which are the results of the extraction of features from the original dataset, it can be seen that other parts do not include landmarks of the face, such as hair and ears. The results of the extraction appear to be vaguer and blurrier. This will complicate the feature extraction in the next layer, which will be processed with lower pixels.



(a)

(b)

Figure 3. Feature extraction on the preprocessed data set (a) Conv2D layer (b) Max_Pooling2D layer

In Figures 3 (a) and (b), the exclusive focus on the features within the extracted face landmark area is evident for the preprocessed data set. The extraction results exhibit clarity and brightness, which shows the effectiveness of the pre-processing techniques.

(a)

(b)

Figure 4. Feature Extraction in the Original Dataset (a) Conv2D layer (b) Max_Pooling2D layer

On the contrary, in Figures 4 (a) and (b), which represent the extraction of features from the original dataset, certain parts lack facial landmarks, including regions such as hair and ears. This results in vagueness and blurriness in the extraction, which introduces complications for subsequent feature extraction layers, especially those processing with lower pixel information. The comparison underscores the advantageous impact of preprocessing on isolating and enhancing crucial facial features for improved recognition accuracy in subsequent stages of the neural network.

In our study, noise reduction and histogram equalization methods are used for contrast normalization, and background removal is implemented to improve feature clarity, which is crucial to improve CNN performance in facial recognition. Similar preprocessing approaches have been used effectively in medical imaging. For example, a study using chest radiograph images with pre-processing algorithms demonstrates significant improvements in CNN accuracy for COVID-19 detection from chest radiograph images through similar pre-processing techniques, including noise reduction and histogram equalization [29].

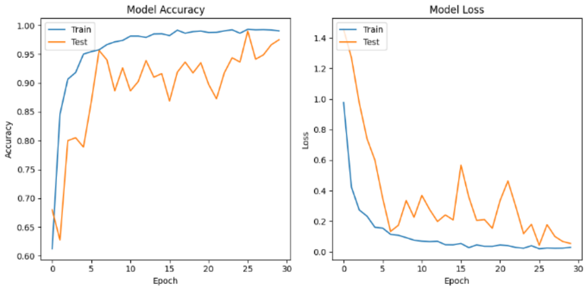
3.3. Training Convergence

The convergence rate during training was another aspect under analysis. Monitoring and contrasting the convergence rate and validation loss illuminated whether preprocessing techniques expedited and optimized the learning process, hinting at potential performance enhancements. Table 2 shows the training results for the original dataset and the preprocessed dataset. It can be seen from the training results of the two datasets that the validity accuracy for the preprocessed dataset is constantly higher at 30, 40, and 50 epochs. The validity loss for the preprocessed dataset is constantly lower for all epochs.

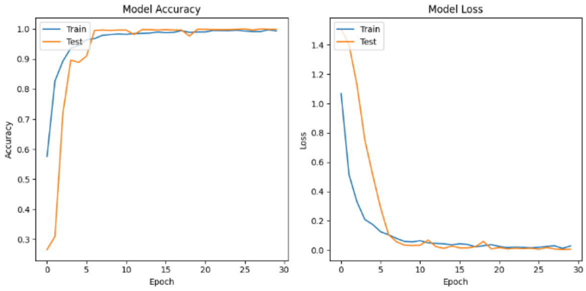
Table 2. Validity accuracy and validity loss

Dataset	Epoch	Validity Accuracy (Highest)	Validity Loss (Lowest)	Average Validity Accuracy	Average Validity Loss
Original	30	0,9900	0,0433	0,9950	0,0244
	40	0,9950	0,0281		
	50	1,0000	0,0017		
Pre-processed	30	1,0000	0,0041	0,9996	0,0044
	40	0,9988	0,0077		
	50	1,0000	0,0013		

To be able to see more thoroughly, we can see figures 5, 6, and 7, which show the accuracy and loss during the training process. Figures 5 (a) 6 (a), and 7 (a) show the accuracy and loss of the original dataset, while Figures 5 (b) 6 (b), and 7 (b) show the preprocessing dataset. The results indicate that the preprocessing data set shows an average improvement of 0.46% in the accuracy of the validity, accompanied by an average drop of 82.08% in the loss of validity. Figure 5 illustrates the accuracy of the model and the loss of the model at 30 epochs for both the original and the preprocessed datasets. In the case of the original dataset, the validation accuracy and validation loss graphs exhibit significant fluctuations, indicative of model instability and training failure, leading to overfitting. In contrast, the preprocessed data set displays a stable trend in both model accuracy and model loss graphs, with validation accuracy and validation loss maintaining consistency. This suggests the effectiveness of pre-processing in enhancing model stability and mitigating overfitting issues, contributing to a more reliable and optimized training process.

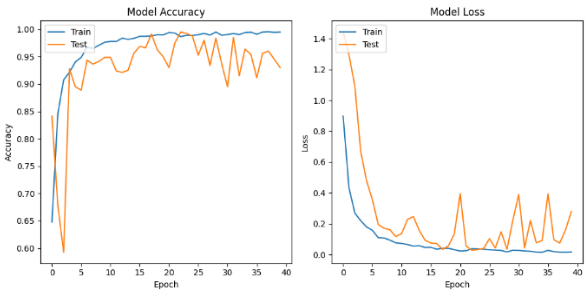


(a)

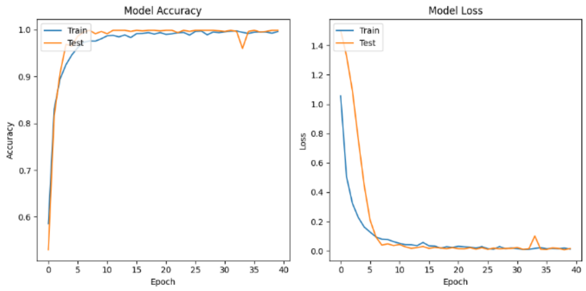


(b)

Figure 5. Model Accuracy and Model Loss at 30 epochs. (a) Dataset Original (b) Dataset Preprocessed



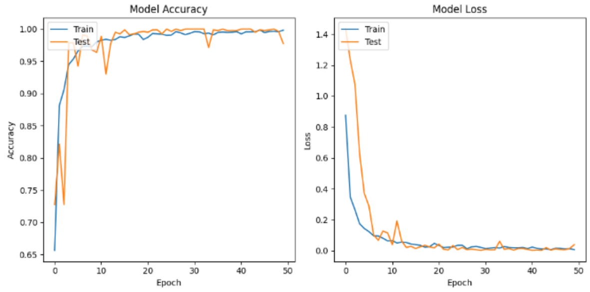
(a)



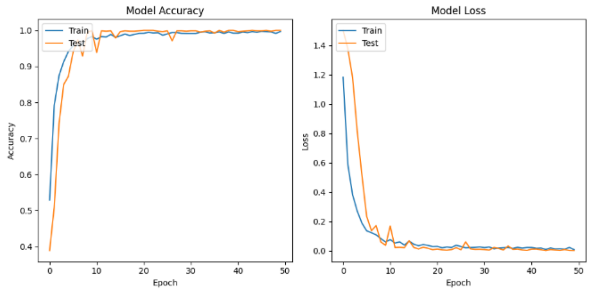
(b)

Figure 6. Model Accuracy and Model Loss at 40 epochs. (a) Dataset Original (b) Dataset Preprocessed

Figure 6 shows the accuracy of the model and the loss of the model at 40 epochs for the original and pre-processed data sets. For the original data set, both the validation accuracy and the validation loss graphs exhibit notable fluctuations and signs of overfitting. These observations suggest model instability and training failure, which affects the efficacy of the data set. In contrast, the preprocessed data set demonstrates consistent and stable trends in both the model accuracy and the model loss graphs, with the validation accuracy and the validation loss maintaining stability. This further underscores the positive impact of preprocessing in stabilizing the model, mitigating overfitting, and enhancing the overall effectiveness of the training process.



(a)



(b)

Figure 7. Model Accuracy and Model Loss at 50 epochs. (a) Original Dataset; (b) Preprocessed Dataset

Figure 7, illustrates the model accuracy and model loss at 50 epochs for the original and preprocessed datasets, the graph for the original dataset displays occasional fluctuations in both validation accuracy and validation loss. This continued variability indicates ongoing instability in the model. In contrast, the preprocessed dataset exhibits a stable trend in both model accuracy and model loss graphs, with validation accuracy and validation loss consistently maintaining stability. This further emphasizes the effectiveness of pre-processing to ensure model stability, reduce overfitting, and improve the robustness of the training process.

From the accuracy and loss graphs in Figures 5, 6, and 7, the preprocessed dataset's graphs are consistently more stable, both at epochs 30, 40, and 50. The more stable the graph lines, it indicates the more optimal the training process. This shows that the models generated from preprocessed datasets have a more optimal training process.

When examining the impact of different epoch numbers on CNN training, our study aligns with recent findings in medical image analysis. The chapter 'Transfer Learning Using Convolutional Neural Network Architectures for Brain Tumor Classification from MRI Images' highlights that varying the number of training epochs (25, 50, and 90) significantly affects CNN models' performance and training time. This supports our observation that adjusting the epoch numbers in our models, trained on original and pre-processed datasets, critically influences the convergence and accuracy of the training. Such findings underscore the importance of epoch selection in optimizing CNN training efficiency [30].

3.4. Real-time Model Testing

Additionally, real-time model testing was performed, pitting the original model against the preprocessed model in real-world scenarios. This involved evaluating the models on a stream of video frames and categorizing recognized faces based on an average recognition score from the last 20 frames in the same class. Summarizing the results in log files provided insight into the practical applicability and performance of the models. Table 3 shows a recap of the results of log-file calculations from real-time model testing of the original dataset and the preprocessed dataset.

Table 3. Real-time model testing result

Class	Original			Pre-processed		
	Known	Unknown	Comp Time (s)	Known	Unknown	Comp Time (s)
0	74%	26%	0,0638	93%	7%	0,0630
1	78%	22%	0,0631	89%	11%	0,0620
2	79%	21%	0,0619	100%	0%	0,0611
3	83%	17%	0,0621	98%	2%	0,0613

It can be seen from the data contained in Table 3 that in the preprocessed data set, there is a constant improvement in the percentage of recognized faces. Similarly, in the unrecognized percentage, there is a constant decline. This shows that the pre-processing process has been going well. The results indicate that the pre-processing data set exhibits an average performance improvement of 21% and an average reduction in computing time of 1.41%.

In our exploration of real-time model testing, a significant reference is a study by Saponara, Elhanashi, and Gagliardi (2021) in the Journal of Real-Time Image Processing. They successfully implemented the YOLOv2 CNN architecture for real-time fire and smoke detection in surveillance systems. This study exemplifies the ability of CNNs to perform efficiently in real-time applications, aligning with our findings on the practical applicability and performance of CNN models in similar real-time scenarios [31].

4. Conclusions

The comprehensive evaluation presented in this study meticulously examines the impact of advanced preprocessing techniques on convolutional neural network (CNN) models in the domain of facial recognition. Assessment spans key dimensions, including precision, uniqueness of features, training convergence, and real-time capabilities, offering a holistic view of the effectiveness of preprocessing techniques. The research rigorously compared the performance of CNN models on both original and preprocessed datasets, employing various evaluation strategies. In particular, the evaluation encompassed accuracy metrics, precision, recall, and F1 scores at different epochs, shedding light on the consistent improvements observed in the pre-processed data set. The results revealed an average improvement in accuracy of 4.17%, recall of 3.45%, precision of 3.45%, and an improvement in the average F1 score of 3.81% in epochs 30, 40, and 50. Visualizing characteristics played a crucial role in elucidating the impact of pre-processing on facial features extraction. The visualizations distinctly showcased the superiority of the pre-processed dataset in isolating and improving crucial facial features, contributing to improved recognition accuracy in subsequent CNN layers. These findings underscore the advantageous impact of pre-processing on feature extraction, emphasizing clarity and brightness in the extracted facial landmarks area. Training convergence was a key facet under scrutiny, revealing that the preprocessed dataset consistently outperformed the original dataset in terms of accuracy of validity and loss of validity in epochs 30, 40, and 50. Stability and lower validity loss in the preprocessed dataset indicated improved model stability, reduced overfitting, and a more reliable and optimized training process overall. Real-time model testing further substantiated the effectiveness of preprocessing, demonstrating a constant improvement in the percentage of recognized faces, a decline in unrecognized faces, and notable enhancements in average performance, coupled with a reduction in average computing time. These results affirm the successful implementation of pre-processing techniques and their positive impact on real-world applicability. In conclusion, this study highlights the significant potential of advanced preprocessing techniques to enhance CNN models for facial recognition. However, for future investigations, it is recommended to extend the application of these techniques to different types of images beyond facial recognition. Additionally, exploring various CNN architectural models, such as VGG, MobileNet, FaceNet, and AlexNet, among others, would contribute to a more comprehensive understanding of the broader applicability and nuances of pre-processing techniques in diverse contexts.

Acknowledgments

The author declares that there is no conflict of interest. This research did not receive a specific grant from any funding agency in the public, commercial, or non-profit sectors.

References

[1] S. and Z. A. Razzak Muhammad Imran and Naz, “Deep Learning for Medical Image Processing: Overview, Challenges and the Future,” in Classification in BioApps: Automation of Decision Making, A. S. and B. S. Dey Nilanjan and Ashour, Ed., Cham: Springer International Publishing, 2018, pp. 323–350. https://doi.org/10.1007/978-3-319-65981-7_12

[2] H. P. Wei, Y. Y. Deng, F. Tang, X. J. Pan, and W. M. Dong, “A Comparative Study of CNN- and Transformer-Based Visual Style Transfer,” J Comput Sci Technol, vol. 37, no. 3, pp. 601–614, Jun. 2022. https://doi.org/10.1007/s11390-022-2140-7

[3] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, “A review of convolutional neural network applied to fruit image processing,” Applied Sciences (Switzerland), vol. 10, no. 10.

MDPI AG, May 01, 2020. https://doi.org/10.3390/app10103443
[4] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction," Journal of Applied Science and Technology Trends, vol. 1, no. 2, pp. 56–70, May 2020. https://doi.org/10.38094/jastt1224
[5] S. Wei, S. Ji, and M. Lu, "Toward Automatic Building Footprint Delineation From Aerial Images Using CNN and Regularization," IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 3, pp. 2178–2189, 2020. https://doi.org/10.1109/TGRS.2019.2954461
[6] J. Waleed, S. Albawi, H. Q. Flayyih, and A. Alkhayyat, "An Effective and Accurate CNN Model for Detecting Tomato Leaves Diseases," in 2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA), 2021, pp. 33–37. https://doi.org/10.1109/IICETA51758.2021.9717816
[7] Y. Sonawane, D. Khachane, and O. Khaire, "Classification of Plant Leaf Diseases Using Machine Learning and Image Preprocessing Techniques with Smart Agriculture," International Journal of Scientific Research in Engineering and Management (IJSREM), vol. 06, no. 06, pp. 1–5, 2022. https://doi.org/10.55041/IJSREM13932
[8] F. A. Zeiser et al., "Segmentation of Masses on Mammograms Using Data Augmentation and Deep Learning," J Digit Imaging, vol. 33, no. 4, pp. 858–868, 2020. https://doi.org/10.1007/s10278-020-00330-4
[9] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," J Big Data, vol. 8, no. 1, Dec. 2021. https://doi.org/10.1186/s40537-021-00444-8
[10] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," Global Transitions Proceedings, vol. 3, no. 1, pp. 91–99, Jun. 2022. https://doi.org/10.1016/j.gltp.2022.04.020
[11] H. Chougrad, H. Zouaki, and O. Alheyane, "Deep Convolutional Neural Networks for breast cancer screening," Comput Methods Programs Biomed, vol. 157, pp. 19–30, 2018. https://doi.org /10.1016/j.cmpb.2018.01.011
[12] J. Liu, Y. Feng, and H. Wang, "Facial Expression Recognition Using Pose-Guided Face Alignment and Discriminative Features Based on Deep Learning," IEEE Access, vol. 9, pp. 69267–69277, 2021. https://doi.org/10.1109/ACCESS.2021.3078258
[13] J. Liu, H. Wang, and Y. Feng, "An End-to-End Deep Model with Discriminative Facial Features for Facial Expression Recognition," IEEE Access, vol. 9, pp. 12158–12166, 2021. https://doi.org/10.1109/ACCESS.2021.3051403
[14] S. J. Elias et al., "Face recognition attendance system using local binary pattern (LBP)," Bulletin of Electrical Engineering and Informatics, vol. 8, no. 1, pp. 239–245, Mar. 2019. https://doi.org/10.11591/eei.v8i1.1439
[15] M. O. Oloyede, G. P. Hancke, and H. C. Myburgh, "Improving face recognition systems using a new image enhancement technique, hybrid features and the convolutional neural network," IEEE Access, vol. 6, pp. 75181–75191, 2018. https://doi.org/10.1109/ACCESS.2018.2883748
[16] R. Ullah et al., "A Real-Time Framework for Human Face Detection and Recognition in CCTV Images," Math Probl Eng, vol. 2022, 2022. https://doi.org/10.1155/2022/3276704
[17] J. Zeng, X. Qiu, and S. Shi, "Image processing effects on the deep face recognition system," Mathematical Biosciences and Engineering, vol. 18, no. 2, pp. 1187–1200, 2021. https://doi.org/10.3934/MBE.2021064
[18] L. Rai, Z. Wang, A. Rodrigo, Z. Deng, and H. Liu, "Software development framework for real-time face detection and recognition in mobile devices," International Journal of Interactive Mobile Technologies, vol. 14, no. 4, pp. 103–120, 2020. https://doi.org/10.3991/IJIM.V14I04.12077
[19] B. Qin, L. Liang, J. Wu, Q. Quan, Z. Wang, and D. Li, "Automatic identification of Down syndrome using facial images with deep convolutional neural network," Diagnostics, vol. 10, no. 7, Jul. 2020. https://doi.org/10.3390/diagnostics10070487
[20] T. Hussain et al., "Internet of Things with Deep Learning-Based Face Recognition Approach for Authentication in Control Medical Systems," Comput Math Methods Med, vol. 2022, 2022. https://doi.org/10.1155/2022/5137513
[21] D. S. AbdELminaam, A. M. Almansori, M. Taha, and E. Badr, "A deep facial recognition system using computational intelligent algorithms," PLoS One, vol. 15, no. 12 December, Dec. 2020. https://doi.org/10.1371/journal.pone.0242269
[22] X. Liu, Y. Zou, H. Kuang, and X. Ma, "Face image age estimation based on data augmentation and lightweight convolutional neural network," Symmetry (Basel), vol. 12, no. 1, 2020. https://doi.org/10.3390/SYM12010146
[23] M. S. Hadis, J. Akita, M. Toda, and Nurzaenab, "The Impact of Preprocessing on Face Recognition using Pseudorandom Pixel Placement," in 2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP), 2022, pp. 1–5. https://doi.org/10.1109/IWSSIP55020.2022.9854474
[24] I. Ahmad, I. Moon, and S. J. Shin, "Color-to-grayscale algorithms effect on edge detection — A comparative study," in 2018 International Conference on Electronics, Information, and Communication (ICEIC), 2018, pp. 1–4https://doi.org/10.23919/ELINFOCOM.2018.8330719
[25] Shaik, G. Azeem, and E. C. Jones, "Image Classification Using Deep Learning Architecture-EfficientNET for COVID-19 Medicine Consumption," International Supply Chain Technology Journal, vol. 7, no. 07, 2021.https://doi.org/10.20545/isctj.v07.i07.01
[26] T. Ayyavoo and J. J. Suseela, "Illumination pre-processing method for face recognition using 2D DWT and CLAHE," in IET Biometrics, Institution of Engineering and Technology, Jul. 2018, pp. 380–390. https://doi.org/10.1049/iet-bmt.2016.0092
[27] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," J Big Data, vol. 8, no. 1, Dec. 2021. https://doi.org/10.1186/s40537-021-00444-8
[28] A. Taner, Y. B. Öztekin, and H. Duran, "Performance analysis of deep learning cnn models for variety classification in Hazelnut," Sustainability (Switzerland), vol. 13, no. 12, Jun. 2021. https://doi.org/10.3390/su13126527
[29] M. Heidari, S. Mirniaharikandehei, A. Z. Khuzani, G. Danala, Y. Qiu, and B. Zheng, "Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms," Int J Med Inform, vol. 144, Dec. 2020. https://doi.org/10.1016/j.ijmedinf.2020.104284
[30] R. Chelghoum, A. Ikhlef, A. Hameurlaine, and S. Jacquir, "Transfer learning using convolutional neural network architectures for brain tumour classification from MRI images," in IFIP Advances in Information and Communication Technology, Springer, 2020, pp. 189–200. https://doi.org/10.1007/978-3-030-49161-1_17
[31] S. Saponara, A. Elhanashi, and A. Gagliardi, "Real-time video fire/smoke detection based on CNN in antifire surveillance systems," J Real-Time Image Process, vol. 18, no. 3, pp. 889–900, Jun. 2021. https://doi.org/10.1007/s11554-020-01044-0