# Real-Fake Face Detection based on Joint Multi-Layer CNN Structure and Data Augmentation

Weiting Bian
*School of Material and Engineering, South China University of Technology, Guangzhou, China*

Keywords: Convolutional Neural Network, Real-False Face Recognition, Fake Face Images.

Abstract: Nowadays, with the advancement of technology, various image editing and image generation tools have emerged, leading to the generation of fake face images. This has caused many issues, such as fraud and false information. Therefore, it is highly meaningful to use more effective methods to identify real and fake faces. The topic of this study is real-false face recognition grounded on the Convolutional Neural Network (CNN) model. CNN structure is utilized, consisting of data augmentation, resize, scale, convolution, pooling, and fully-connected layers (FC). Initially, both training and validation losses are relatively high for the training results, but as training iterations progress, the losses gradually decrease. Meanwhile, the accuracy of the model gradually improves after several rounds of iterations, ultimately reaching 90% on the training and validation sets. After being evaluated on an independent test dataset, the model achieved a 15.90% loss with a 93.63% accuracy. The model achieves high accuracy in predicting real and fake faces, demonstrating good performance and practicality. Lastly, effective recognition of real and fake faces can help people identify false information, avoiding panic, financial losses, and rumors to some extent. It plays a significant role in social stability.

## 1 INTRODUCTION

With the emergence of technologies such as photo editing and Artificial Intelligence (AI)-generated images, an increasing number of exquisite pictures have been presented to the public. However, the misuse of these technologies also presents a worrying growth trend. Many images of fake faces are produced. The main issues caused by fake faces include fake information, online hoaxes, and financial fraud. Disturbingly, through some software applications or tools, it is possible to create deep fake images without any programming techniques or relevant background information (Suganthi et al 2022). Therefore, the recognition of real and fake faces is a very important topic. This topic aims to develop advanced algorithms and methods to distinguish between real faces and false faces generated by various forgery techniques.

Earlier attempts at this task involved traditional machine-learning models and relied on handcrafted features (Li and Lyu 2018. While effective to some extent, these methods lacked the robustness and scalability needed in the age of deepfakes. In the past few years, many achievements have been made in identifying processed images using deepfakes. Through advanced deep learning methods, it is possible to superimpose someone's face onto another one's face to create an image (Suganthi et al 2022). For instance, by utilizing Generative Adversarial Neural Networks (GANs), a deep learning algorithm that is grounded in the idea of automatic decoders and encoders, false images or videos can be detected (Yadav et al 2019). Yang, Li & Lyu introduced a model in 2019 that detects deep fake through discrepancies in head pose (Yang et al 2019). Jagdale and his team introduced a novel algorithm for video super-resolution (NA-VSR) that processes videos by breaking them down into individual frames (Jagdale et al 2019). Given the limitations of current technologies, which include reduced accuracy and extended processing times, Mohamed and colleagues in 2021 suggested a method to detect facial images manipulated through deep fake techniques the combination of fisher face and the Local Binary Pattern Histogram (LBPH) technique (Suganthi et al 2022). Recent works have begun leveraging the Convolutional Neural Network (CNN) for this very purpose. For instance, Nguyen and his team members suggested a system in research that uses CNNs to detect and classify deepfakes with a high degree of



Bian, W.
Real-Fake Face Detection Based on Joint Multi-Layer CNN Structure and Data Augmentation.
DOI: 10.5220/0012800300003885

In *Proceedings of the 1st International Conference on Data Analysis and Machine Learning (DAML 2023)*, pages 25-29
ISBN: 978-989-758-705-4

accuracy (Nguyen et al 2019). Also, Singh, Shanmugam, and Awasthi used face recognition based on CNN to detect fake accounts on social media (Singh et al 2021). What's more, to distinguish genuine from fake images within the domain of ocular biometrics, techniques like Squeeze Net, Dense Convolutional Network (DenseNet), Residual Neural Network (ResNet), and Light CNN have been employed (Nguyen et al 2020).

The main purpose of this study is to detect real and fake faces using a deep learning model. Specifically, a model with several convolutional and pooling layers is employed for image feature extraction. The dataset is then divided into validation, training, and test sets. to educate and evaluate the model. The results show that the model's predictions match the actual results perfectly, confirming the accuracy and reliability of the model. The experimental results demonstrate that by extracting features from the images and applying data augmentation techniques, the model can better distinguish between real and fake faces. This is significant for protecting personal privacy and preventing the spread of false information. To sum up, this study provides an effective approach to real and fake face detection using a deep learning model, addressing the challenges of face recognition and information security in today's society. Through thorough training and testing, the model achieved high accuracy in predicting real and fake faces, demonstrating good performance and practicality.

# 2 METHODOLOGY

## 2.1 Dataset Description and Preprocessing

The dataset originates from the Kaggle platform and is titled "real-and-fake-face-detection" (Dataset). As the name suggests, the dataset comprises images of both real and fake faces. Inside the parent directory, the file of fake images contains high-quality photoshopped face images generated by experts. And the other file contains photos of real human faces. The fake photos are divided into three groups, which are easy, mid, and hard. However, these groups are separated subjectively, so using them as explicit categories is not recommended. The objective is to educate a model in differentiating authentic and fabricated facial images. For preprocessing, the images are resized to a uniform size of 256x256 pixels and are batched in sets of 32. Additionally, random flips (both horizontal and vertical) and rotations are applied for data augmentation purposes to enhance the model's generalization capabilities. The sample is shown in figure 1.


Figure 1: Images from the real-and-fake-face-detection dataset (Picture credit: Original).

## 2.2 Proposed Approach

Within the paradigm of deep learning, CNNs have risen as a foremost technique for image classification tasks. The present research strives to harness the prowess of CNNs to discern between authentic and fabricated facial images. The method is divided into several parts, including data loading, dataset splitting, data augmentation, model building, model compilation, model training, model evaluation, and prediction demonstration.

In the process, facial images are systematically sourced from a designated directory, identifying inherent class labels, as shown in Fig. 2. This dataset is then segmented into distinct training, validation, and testing cohorts using algorithmic randomization. To enhance model generalization and counter overfitting, the image datasets are augmented, involving random geometrical changes like flips and rotations. A detailed CNN architecture is designed, including defined input tensors, convolution layers, pooling points, and densely connected layers. For training preparation, the model is equipped with the Adam optimization algorithm, For training preparation, the model is equipped with the Adam optimization algorithm, using sparse categorical cross-entropy for the loss calculation and utilizing accuracy as the evaluation metric. The model undergoes intensive training epochs using the training data while periodically gauging its performance on the validation dataset. After training, the model's accuracy is evaluated on a previously unseen test dataset. Lastly, a select group of images are used as samples, where the model projects their corresponding classes, marking them with related confidence levels.

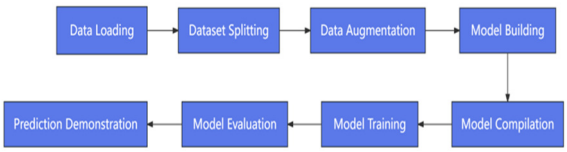


Figure 2: The pipeline of the processing (Picture credit: Original).

### 2.2.1 CNNs

The model used in the process is CNNs. It was first introduced by Fukushima in 1998. In the field of deep learning, CNNs are one of the most important architectures. They have made outstanding contributions in various fields, especially in fields such as computer vision and computational linguistics. CNNs have gained considerable attention from the industrial and academic sectors in the past few years (Li et al 2021). It is composed of neurons, each with a learnable weight and bias. Multiple hidden layers, an input layer, and an output layer are contained by it. Various normalization layers, a convolutional layer, a pooling layer, and a fully connected layer (FC) constitute the hidden layer. By applying convolution operations, the convolution layer can thus combine two groups of data. It mimics the feedback of individual neurons when visual stimuli are generated. When a layer of neural clusters outputs, by correlating their outputs with individual neurons, the dimensionality can be reduced through pooling layers. Through the FC layer, input images can be classified, which is also the main function of the FC layer. Moreover, every neuron in a given layer is linked to all neurons in the subsequent layer. The FC layer follows the convolutional layer. There can be a subsampling layer between these two layers.

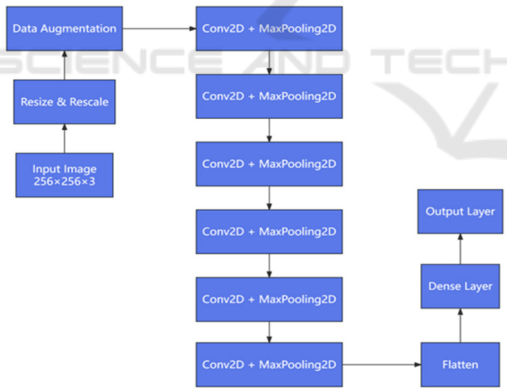


Figure 3: The pipeline of the processing (Picture credit: Original).

In the experiment, no predefined or pre-trained CNN models are employed, such as VGG16, ResNet50, etc. Instead, a simple CNN architecture is defined from scratch. The model is structured in the following manner: In the process, firstly, the input data undergoes resizing and normalization. Then data augmentation is applied, which includes random horizontal and vertical flips, as well as random rotations. And six convolutional layers are present, with each succeeded by a max-pooling layer. After that, the data is then flattened to be fed into FC layers. Finally, there are two dense (FC) layers, with the last output layer meant for classification. The process is shown in Fig. 3.

### 2.2.2 Loss Function

The loss function used is Sparse Categorical Cross-entropy. The Sparse Categorical Cross-entropy loss is an optimization function used for multi-class classification problems. The formula for cross-entropy loss is denoted as:

$$H(y,p) = -\sum iyi \cdot \log(pi) \qquad (1)$$

where y represents the true probability distribution, p denotes the estimated probability distribution, and i represents the index of the class. This cross-entropy loss function can be used when multiple label classes are present. The labels are expected to be provided as integers. In machine learning, a loss function called Sparse Categorical Cross-entropy is widely used for classifying. The cross-entropy loss is determined by comparing the predicted class probabilities with the actual class labels, especially when the loss function deals with sparse target labels (integer labels) rather than one-hot encoded labels. This loss function is frequently employed in neural network training. Cross-entropy quantifies the divergence between the true distribution and the predicted probability distribution. In the context of classification, the true distribution is often represented as a one-hot encoded vector, assigning a value of 1 to the correct class and a value of 0 to every other class. In the case of Sparse Categorical Cross-entropy, the true labels are integers representing classes, while the predicted labels are probability distributions over all classes. Since the labels are provided as integers and not one-hot vectors, the true distribution for a particular sample can be inferred from its integer label.

## 2.3 Implementation Details

The system employs Kaggle's Python environment, which offers a multitude of convenient data analytics libraries tailored for data scientists. It operates using a Docker image provided by Kaggle and activates the Tensorflow framework to leverage GPU computations. This ensures rapid and efficient performance when dealing with sizable datasets and deep learning models. In the process of data augmentation, the images are flipped either horizontally, vertically, or both. This ensures that the model is invariant to the orientation of the face. Moreover, images are rotated by a specified factor, making the model robust against slight tilts and rotations in the input images. The use of these

augmentation techniques not only helps in increasing the effective size of the dataset but also ensures that the model generalizes well and is less likely to overfit the training data. In this system, the following hyperparameters are defined: The resolution of the images being processed is established at 256x256 pixels. The model operates with a batch size of 32, and the entire dataset undergoes 50 iterations during the training process, with each iteration termed as an epoch. The chosen optimizer for this system is the Adam optimizer, renowned for its adaptive learning rate mechanism and its efficient handling of gradient descent, especially in high-dimensional spaces.

# 3 RESULTS AND DISCUSSION

This chapter mainly provides a detailed evaluation and examination of the training outcomes of the aforementioned deep learning model. The content will be mainly divided into three parts, namely the loss of the model, the change in validation accuracy and training accuracy, and the evaluation of the generalization ability of the model.

## 3.1 Loss Value Analysis

As shown in Fig. 4, the model's training and validation loss values both show a downward trend with the increase in the number of epochs. Initially, both training and validation losses are relatively high, but as the number of training iterations increased, the losses gradually decrease, indicating that the model has a better fit for the data.

The main reason for this phenomenon is that the model is constantly adjusting its weights through forward and backward propagation, resulting in a gradual reduction in the difference between the predicted output and the real label. The reduced loss value means that the predictive accuracy of the model is improving, which is very beneficial for image classification tasks.

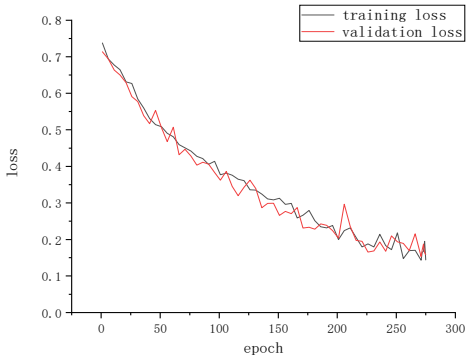


Figure 4: The training and validation losses (Original).

## 3.2 The Performance of the Various Epochs

As shown in Fig. 5, as the training epochs progress, the model's performance in terms of accuracy on both the validation set and the training set shows an upward trend. At the beginning, the accuracy of the model is around 50 %, which is similar to random guessing. However, after several iterations, the accuracy of the model gradually improves, ultimately exceeding 90 % on the training set and 90 % on the validation set. This trend indicates that the model gradually captures the characteristics of the data during the learning process and can classify it more accurately. In addition, the small difference between training accuracy and validation accuracy also indicates that the model does not have significant overfitting, as techniques such as data augmentation are used to improve the model's generalization ability.


Figure 5: The training and validation accuracy (Original).

## 3.3 Generalization Performance Evaluation

Upon evaluation on an independent test dataset, the model achieves a loss value of 0.1590 and attains an accuracy rate of 93.63%. Compared to the results on the validation set, the model's results on the testing dataset is remarkably similar, suggesting that the information learned by the model during training and validation can be effectively transferred to unseen data. This high accuracy further attests to the model's excellent generalization capabilities. Possible reasons might include the use of appropriate data augmentation, regularization techniques, and the choice of network architecture. In conclusion, through the experimental analysis in this chapter, it can be observed that the model's loss progressively diminishes during training, the accuracy continually rises, and the performance on both validation and test

data remains stable, reflecting a commendable generalization capability. These experimental outcomes validate the efficacy and feasibility of the methods employed, offering a potent tool for genuine and counterfeit facial image classification.

## 4 CONCLUSION

The subject of this study is real and false face recognition. A deep learning-driven model is introduced to analyze and differentiate between authentic and fake human faces. With the rapid proliferation of digital media and deepfake technologies, determining the authenticity of facial images has emerged as an imperative in numerous applications ranging from security to entertainment. A comprehensive deep learning model is proposed in the experiment to analyze and differentiate between real and fake facial images. This model uses a multilayer CNN structure that includes data augmentation, resizing, scale, convolution, pooling, and fully-connected layers. The initial stages involve preprocessing the images using resizing and rescaling techniques to ensure uniformity. Following this, data augmentation strategies, such as random flipping and rotation, are employed to augment the dataset and provide robustness to the model. The main model comprises multiple convolutional and pooling layers to extract intricate features from facial images, culminating in dense layers that classify the images into real or fake. Numerous experiments have been carried out on the model to evaluate the proposed methods during the process. During training, after 275 epochs, the model achieves approximately 94. 18 % accuracy on the training dataset and approximately 93. 63% accuracy on the validation dataset. The model also exhibits excellent performance in individual testing sets, achieving a 93. 63 % accuracy rate, highlighting its efficacy and robustness in distinguishing between real and fake facial images. In future research, enhancing the robustness and adaptability of the model is considered. Given the continuously evolving deepfake generation methods, the model needs to be better equipped to counter these sophisticated forgery techniques. Moreover, to ensure the model's effectiveness in real-world applications, it must maintain high accuracy and reliability even when confronted with various facial obstructions and diverse facial expressions. Therefore, the next phase of research will focus on analyzing the model's performance across these varied scenarios and exploring how to optimize its responsiveness in the face of more complex situations. This will necessitate not only a deep dive into the model's architecture and training strategies but also a consideration of more comprehensive data augmentation techniques to train the model to better understand and address these challenges.

## REFERENCES

S. T. Suganthi, et al, “Deep learning model for deep fake face recognition and detection,” PeerJ Computer Science, vol. 8, 2022, p. e881.

Y. Li, S. Lyu, “Exposing deepfake videos by detecting face warping artifacts,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, vol. 2018, pp. 46-52.

Yadav, Digvijay, and Sakina, “Deepfake: A survey on facial forgery technique using generative adversarial network,” 2019 International conference on intelligent computing and control systems (ICCS), IEEE, vol. 2019.

X. Yang, Y. Li, and S. Lyu, “Exposing deep fakes using inconsistent head poses,” ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, vol. 2019.

Jagdale, Rohita, and S. Shah, “A novel algorithm for video super-resolution,” Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, vol. 1, 2019.

Nguyen, H. Huy, Y. Junichi, and E. Isao, “Capsule-forensics: Using capsule networks to detect forged images and videos,” ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, vol. 2019.

Singh, Vernika, Raju Shanmugam, and Saatvik Awasthi. "Preventing fake accounts on social media using face recognition based on convolutional neural network." Sustainable Communication Networks and Application: Proceedings of ICSCN 2020. vol. 2021.

Nguyen, H. Mark, and R. Derakhshani, “Eyebrow recognition for identifying deepfake videos,” 2020 international conference of the biometrics special interest group (BIOSIG). IEEE, vol. 2020.

Dataset,https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection.

Z. Li, et al, “A survey of convolutional neural networks: analysis, applications, and prospects,” IEEE transactions on neural networks and learning systems, 2021.