*Undergraduate Conference on Intelligent Computing and Systems (UCICS 2025)*
*26-27 February, 2025; Varendra University, Rajshahi, Bangladesh*

# Deep Learning-based Classification of Real and Fake Face Images Using Convolutional Neural Networks

Shahriar Siddique Arjon
*Dept. of Computer Science and Engineering*
*Pabna University of Science and Technology*
Pabna-6600, Bangladesh
Email: aorjon123@gmail.com

Tamanna Yasmin
*Dept. of Computer Science and Engineering*
*Pabna University of Science and Technology*
Pabna-6600, Bangladesh
Email: tamannayasmin2632001@gmail.com

Ankur Kumar Mondol
*Dept. of Computer Science and Engineering*
*Pabna University of Science and Technology*
Pabna-6600, Bangladesh
Email: ankurmondol2109@gmail.com

Pabitra Kumar Biswas
*Dept. of Computer Science and Engineering*
*Pabna University of Science and Technology*
Pabna-6600, Bangladesh
Email: pk605532@gmail.com

Shabbir Mahmood
*Dept. of Computer Science and Engineering*
*Pabna University of Science and Technology*
Pabna-6600, Bangladesh
Email:shabbir.cse@pust.ac.bd

Dr. Md. Abdur Rahim
*Dept. of Computer Science and Engineering*
*Pabna University of Science and Technology*
Pabna-6600, Bangladesh
Email:rahim@pust.ac.bd

***Abstract*—Fake image detection, aiming to differentiate between real and fake images, a crucial task in combating image manipulation and ensuring the reliability of visual content across various applications. This research introduces a method for binary image classification based on convolutional neural networks (CNNs), which is tailored to improve the precision of fake image detection.In digital forensics, facial recognition software, content moderation, social networking platforms, and e-commerce, this technique can be used to verify images and guard against unauthorized alterations. After training, we assess the model with a classification report and a confusion matrix, which offer insights into its accuracy, recall, F1-score, and overall performance in identifying fake images. The model shows its practical usefulness by detecting subtle image alterations, achieving a validation accuracy of 80.60%. This algorithm, which is based on CNN technology, provides effective detection of altered images, thus safeguarding the integrity of visual material in a variety of fields.**

***K*eywords - *Deep Learning*, *CNN*, *Data Augmentation*, *Image Normalization*,*Adam Optimizer***

## I. Introduction

Using a CNN-based method, this work tackles the problem of differentiating between real and fake face images. With the help of a strong preprocessing pipeline that includes picture scaling and normalization to guarantee constant input quality, the suggested model makes use of CNNs' capacity to extract and learn features from visual data. Rotation, flipping, and zooming were used to increase model performance and generalization in a labeled dataset of actual and generated facial images. During training and validation, the model was assessed using metrics like accuracy and loss, which showed how well it could detect fake images. [1].

Classifying fake images has become essential for identity verification and digital forensics due to the growing use of deepfake technology. Deepfakes are become more sophisticated and pervasive, affecting domains including security, content verification, and social media. In an age of sophisticated synthetic media, our study emphasizes CNNs' capacity to counter this escalating threat and supports the larger endeavor to preserve the authenticity of digital information. [2]

This is how the work is arranged: Section V outlines the methodology, Section VI assesses performance, Section VII addresses difficulties and future directions, Section III analyzes relevant literature, Section IV explains data preparation, and Section II deals with fake image identification.

## II. Problem Statement

Image synthesis technology's quick development has sparked worries about digital information authenticity, security, and privacy. When realistic fake images are misused, accurate identification is required while maintaining consistent performance across a variety of datasets with different lighting conditions, resolutions, and content.

## III. Literature Review

In this section, we discuss previous research on variety of deep learning algorithms. In order to identify and monitor tool wear in metal cutting, Thomas Bergs, et al. [3] concentrated on deep learning. An FCN with a 0.7 IoU score detected worn areas in microscopic pictures, demonstrating automated tool wear analysis using machine tool microscopes, while a CNN obtained 95.6% accuracy in classifying tool types.Deep learning models were used by Olsen, et al. [4] to categorize images of 16 different types of weeds.Their effectiveness in agricultural applications and adaptability across a variety of datasets were demonstrated by their average classification accuracies of 95.1% and 95.7%.Using the CICIDS2018 and Edge_IIoT datasets, Vanlalruata Hnamte, et al. [5] created a deep learning-based network intrusion detection system (NIDS). Although more research is required for greater generalization, the model's 100% and 99.64% accuracy, respectively, showed that

deep learning has the capacity to detect cyber threats.Deep learning and machine learning were utilized by Rahman, et al. [6] to enhance the detection of acute lymphoblastic leukemia (ALL). DenseNet201 reported 99.5% accuracy, while their CNN model obtained an impressive 99.84% accuracy. To make sure that these findings are reliable, more research is required.Using CNNs for feature extraction and optimized K-Nearest Neighbors and SVM for classification, Rimsha Rafique, et al. [1] proposed an automated deepfake picture classification approach that achieved 89.5% accuracy with ResNet. To guarantee robustness, more validation is required.

## IV. DATASET DESCRIPTION AND PREPROCESSING

### A. Dataset Description

The 2,041 images in the dataset for this study, which came from Kaggle [7], include 1,081 real and 960 fake face images in various file types, including.png,.jpg, and.jpeg. The labels training_real (genuine) and training_fake (synthetic) were applied to every image.This dataset is perfect for evaluating CNNs' capacity to discriminate between real and fake faces because it has been carefully selected to reflect a variety of facial appearances.Fig.1 represents real and fake images.

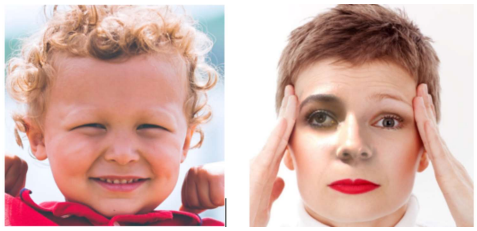

Fig. 1. Representing Real and Fake Images

### B. Data Preprocessing

By scaling all pictures to 224x224 pixels, transforming them from BGR (OpenCV format) to RGB for TensorFlow compatibility, and normalizing pixel values to a 0–1 range for uniformity and quicker convergence, the preprocessing pipeline made sure that the CNN would receive consistent and compatible input. To increase the diversity of training data, data augmentation techniques such as magnification (0.2 factor), width/height shifts (10%), random rotations (up to 30°), and horizontal flips were used. The dataset was divided into subsets for testing and training, with testing data being used to assess performance and training data being used to train the model. For photographs of real faces, labels were encoded as 0, and for images of fake faces, they were encoded as 1.The data distribution table and Hyparameters are shown in Table I,II.

TABLE I
DATA DISTRIBUTION TABLE

| Class | Number of Images | Percentage |
|---|---|---|
| Real Images | 1081 | 52.97% |
| Fake Images | 960 | 47.03% |
| **Total** | **2041** | **100%** |

TABLE II
MODEL HYPERPARAMETERS AND EXPERIMENTAL SETUP

| Parameter/Setup | Details |
|---|---|
| Image Size | 224x224 pixels |
| Number of Classes | 2 (Real, Fake) |
| Loss Function | Sparse Categorical Crossentropy |
| Optimizer | Adam Optimizer |
| Learning Rate | 0.000001 |
| Activation Function | ReLU , Softmax |
| Epochs | 500 |
| Augmentation | Rotation (30°), Zoom (0.2), Width and Height Shift (0.1, 0.1), Flip |
| Dropout Rate | 0.4 |
| Train-Test Split | Independent loading for train and test data |
| Framework Used | TensorFlow/Keras |
| Hardware | GPU-enabled machine (if available), CPU fallback |
| Software | Python, Keras, OpenCV, NumPy, Matplotlib, Seaborn |

## V. METHODOLOGY

CNNs are used in this method to differentiate between real and fake faces. In order to reduce spatial dimensions, the design incorporates max-pooling, ReLU activation, and convolutional layers with 32 3x3 filters. Additional features are extracted by a second set of convolutional layers with 32 and 64 filters. To avoid overfitting, dropout is introduced after the third convolutional block. With a final softmax output layer for binary classification, the model employs dense layers for classification. Sparse categorical cross-entropy loss, the Adam optimizer, and a learning rate of 0.000001 are used in its training.Model resilience is improved by data augmentation methods including flipping, zooming, and rotation.The workflow diagram is displayed in Fig. 2.

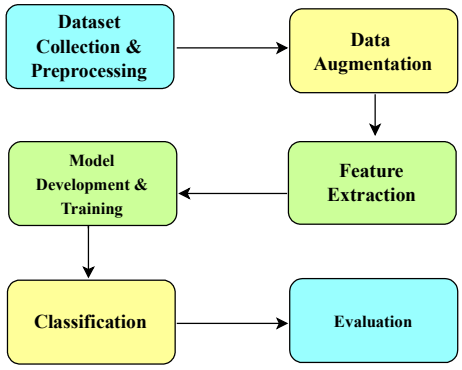


Fig. 2. Workflow of methodological steps

### A. Convolutional Neural Network

This technique uses CNNs to differentiate between real and fake facial images. Preprocessing involves scaling, normalizing to a 0–1 range, and applying effects like rotation, flipping, and zooming to images. In the CNN architecture, dropout

and max-pooling are employed to reduce overfitting, and convolutional layers are utilized to extract features. A dense layer with softmax activation and full linking is used for binary classification.The Adam optimizer and sparse categorical cross-entropy loss are used to train the model. Performance is evaluated using loss and accuracy curves, which display accuracy in training and validation.Fig. 3 depicts the proposed CNN's architecture.

**Convolution :**

$$I_{\text{out}}(i,j) = \sum_{m} \sum_{n} I_{\text{in}}(i+m, j+n) \cdot K(m,n) \tag{1}$$

**Max Pooling :**

$$I_{\text{pool}}(i,j) = \max_{m,n}(I_{\text{in}}(i+m, j+n)) \tag{2}$$

**Fully Connected :**

$$y = W \cdot x + b \tag{3}$$

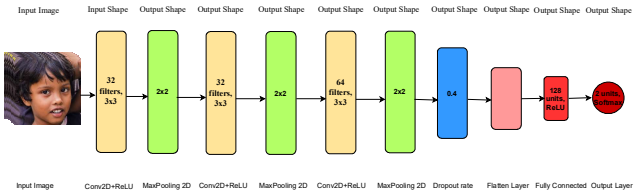


Fig. 3. Proposed CNN Architecture

TABLE III
LAYER DETAILS OF THE CNN ARCHITECTURE.

| Layer | Output Shape |
|---|---|
| Input Layer | (224, 224, 3) |
| Conv2D Layer 1 (Convolutional) | (224, 224, 32) |
| MaxPool2D Layer 1 (Pooling) | (112, 112, 32) |
| Conv2D Layer 2 (Convolutional) | (112, 112, 32) |
| MaxPool2D Layer 2 (Pooling) | (56, 56, 32) |
| Conv2D Layer 3 (Convolutional) | (56, 56, 64) |
| MaxPool2D Layer 3 (Pooling) | (28, 28, 64) |
| Dropout Layer (Regularization) | - |
| Flatten Layer | (50176,) |
| Dense Layer 1 (Fully Connected) | (128,) |
| Dense Layer 2 (Output Layer- Fully Connected) | (2,) |

## VI. RESULTS AND PERFORMANCE ANALYSIS

After 500 epochs of training, the model consistently performed well in identifying real and fake facial images.Highest training accuracy reached 80.52% in the 495th epoch, whereas training loss varied between 45.37% and 48.03%. While validation accuracy reached its highest point at 80.60% in the 498th epoch, validation loss stayed constant between 45.33% and 45.64%. The model's good generalization and appropriateness for real face categorization tasks are demonstrated by the small difference between training and validation metrics.The training and validation summary is shown in Table IV.

TABLE IV
MODEL TRAINING AND VALIDATION SUMMARY

| Epoch No. | Loss (%) | | Accuracy (%) | |
|---|---|---|---|---|
| | Training | Validation | Training | Validation |
| 491 | 48.03% | 45.61% | 76.91% | 80.06% |
| 492 | 47.28% | 45.64% | 77.92% | 80.21% |
| 493 | 46.88% | 45.62% | 78.99% | 80.21% |
| 494 | 47.40% | 45.48% | 77.70% | 80.16% |
| 495 | 45.37% | 45.44% | **80.52%** | 80.21% |
| 496 | 46.39% | 45.48% | 79.66% | 80.16% |
| 497 | 46.95% | 45.37% | 78.50% | 80.45% |
| 498 | 45.90% | 45.33% | 78.96% | **80.60%** |
| 499 | 46.31% | 45.36% | 78.95% | 80.11% |
| 500 | 46.05% | 45.39% | 80.06% | 80.40% |

The evolution of accuracy and validation accuracy throughout training epochs is depicted graphically in Fig. 4, 5.

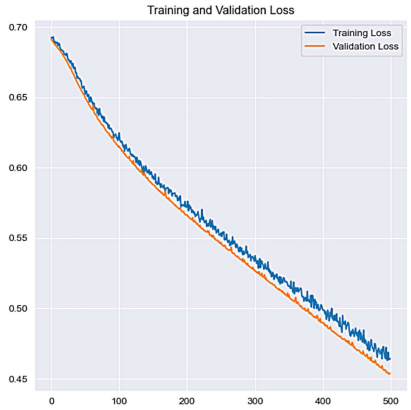


Fig. 4. Loss Curve

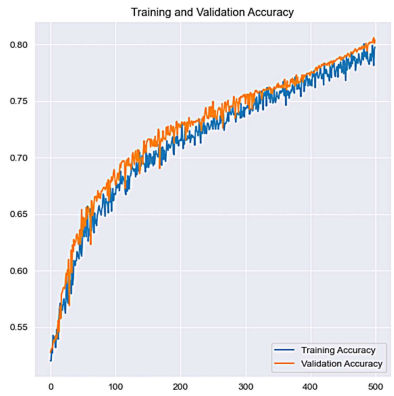


Fig. 5. Accuracy Curve

The model demonstrated good classification performance with an accuracy of 80.60%. The confusion matrix analysis shows that the model performs well in differentiating between real and fake facial images. 783 cases were accurately classified as testing_fake (TN) and 858 occurrences as testing_real (TP). However, there is still room for improvement in terms of reducing misclassification. Incorrectly, the model identified 177 cases of testing_real as testing_fake (FN) and 223 cases of testing_fake as testing_real (FP). These results show that although the model is dependable, it might perform even better

and have fewer occurrences of incorrect classification if it were improved in its capacity to distinguish between the two groups.The confusion matrix is shown in Fig. 6.

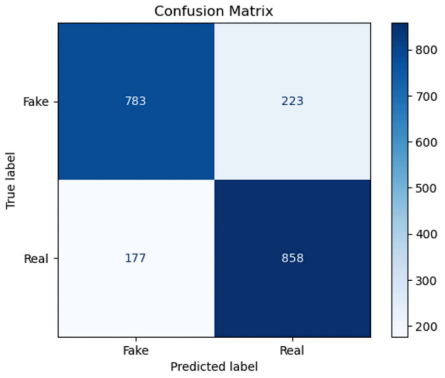


Fig. 6. Confusion Matrix

An exhaustive evaluation of the model's functionality for the testing_real and testing_fake classes is given in the classification report. The model's accuracy for the testing_real class was 83%, meaning that 83% of the images that were predicted to be testing_real were really recognized. With an F1-score of 81%, this class's recall was 79%, which indicates that 79% of all real testing_real images were correctly predicted. The dataset (support) contained 1,081 testing_real images in total. Likewise, the model's precision for the testing_fake class was 78%, which indicates that 78% of the testing_fake instances that were predicted were accurate. 82% of the real testing_fake occurrences were successfully detected by the model, according to the recall for testing_fake. This class's F1-score was 80%, indicating a well-rounded performance. A total of 960 testing_fake images were included in the dataset. The report's summary is displayed in Table V.

TABLE V
OVERVIEW OF THE CLASSIFICATION REPORT

| Class | Accuracy | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|---|
| testing_real | **80.60%** | 83% | 79% | 81% | 1,081 |
| testing_fake | **80.60%** | 78% | 82% | 80% | 960 |

TABLE VI
PERFORMANCE COMPARISON OF CLASSIFICATION MODELS

| Description | Accuracy/Performance |
|---|---|
| Res50_Attn_DSCE [8] | 58.63% |
| MobileVNet [9] | 62.5% |
| InceptionV3 [9] | 72.5% |
| **Proposed Model** | **80.60%** |

Table VI compares the accuracy of many models from several recent articles.These are the most recent studies that have been shown to be similar, however there hasn't been much work done on this uncommon dataset. The table displays the performance of several models in a certain job by comparing their accuracy. The accuracy of Res50_Attn_DSCE is 58.63% [8], while MobileVNet comes in second with 62.5% [9]. At 72.5% [9], InceptionV3 exhibits a noticeable improvement. With an astounding accuracy of 80.60%, the suggested model surpasses all others, demonstrating its greater efficacy and promise for the specified use case.

## VII. CONCLUSION

This study utilized a dataset of 2,041 images, including 1,081 real face images and 960 fake ones. With 80.52% accuracy on the training set and 80.60% accuracy on the validation set, the model performs satisfactorily. Given that the model performs similarly on the training and validation sets, this suggests that it is generalizing well to new, unexplored data. The relatively small gap between training and validation accuracy suggests minimal overfitting, which is a positive sign for model stability.

Despite the good performance, the learning rate is set very low at 0.000001, which might slow down the training process. It is worth experimenting with higher learning rates to potentially accelerate learning without compromising accuracy. Training the model for 500 epochs could lead to overfitting if early stopping is not implemented, so reducing the number of epochs methods might further optimize the results.

Data augmentation could be expanded to include additional transformations to improve the model's robustness. For future work, increasing the complexity of the model, optimizing hyperparameters, and applying advanced techniques like transfer learning or ensembling could further enhance accuracy. Furthermore, assessing the model's performance on a bigger, more varied dataset may shed more light on its capacity for generalization.

## REFERENCES


[1] R. Rafique, R. Gantassi, R. Amin, J. Frnda, A. Mustapha, and A. H. Alshehri, "Deep fake detection and classification using error-level analysis and deep learning," *Scientific Reports*, vol. 13, no. 1, p. 7422, 2023.

[2] H. S. Shad, M. M. Rizvee, N. T. Roza, S. A. Hoq, M. Monirujjaman Khan, A. Singh, A. Zaguia, and S. Bourouis, "[retracted] comparative analysis of deepfake image detection method using convolutional neural network," *Computational intelligence and neuroscience*, vol. 2021, no. 1, p. 3111676, 2021.

[3] T. Bergs, C. Holst, P. Gupta, and T. Augspurger, "Digital image processing with deep learning for automated cutting tool wear detection," *Procedia Manufacturing*, vol. 48, pp. 947–958, 2020.

[4] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, *et al.*, "Deepweeds: A multiclass weed species image dataset for deep learning," *Scientific reports*, vol. 9, no. 1, p. 2058, 2019.

[5] V. Hnamte and J. Hussain, "Dcnnbilstm: An efficient hybrid deep learning-based intrusion detection system," *Telematics and Informatics Reports*, vol. 10, p. 100053, 2023.

[6] W. Rahman, M. G. G. Faruque, K. Roksana, A. S. Sadi, M. M. Rahman, and M. M. Azad, "Multiclass blood cancer classification using deep cnn with optimized features," *Array*, vol. 18, p. 100292, 2023.

[7] CipLab, "Real and fake face image detection dataset." https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection, 2019. Accessed: 2024-12-19.

[8] Y. Zhang, B. Hu, W. Zhang, M. M. Hasan, and H. Liu, "Improved detection of forged and generated facial images based on resnet-50," in *2024 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 199–205, IEEE, 2024.

[9] A. Anirudh and S. Chakraborty, "Hidden and face-like object detection using deep learning techniques-an empirical study," in *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2022.