# AI-driven Facial Authentication: Real versus AI-generated Image Detection

**Hongdezibieke Ayijiang*, Guest Student; CJ Chung, PhD, Professor**

College of Arts & Science, Lawrence Technological University
(*)College of Informational Technologies, Astana IT University

## INTRODUCTION

In the digital era, where the authenticity of online content is increasingly questionable, my project introduces an AI-based solution designed to distinguish real human faces from artificially generated or manipulated counterparts in images. This innovative project takes advantage of deep learning to specifically address people's growing concerns over deepfakes digital deception in media.

The main goal of my project is to develop a complex artificial intelligence model, which can carefully observe facial features and textures of the images. By detecting subtle differences that are often overlooked by the human eyes, the model aims to distinguish the real face from the face that has been changed by digital means. My approach utilizes advanced convolutional neural networks (CNNs) and some pre-trained models such as VGG16 and MobileNetV2. These techniques are very important for analyzing and learning complex patterns in facial images.

To ensure the robustness and versatility of the model, I collected a comprehensive data set. This data set contains all kinds of images, including real and digitally modified faces, which are derived from different demographic data and various lighting and background conditions. Training my AI in this compromise combination will enhance its accuracy and reliability across different scenarios.

Once successfully completed, this project will make a significant contribution to the field of digital forensics. It also lays a foundation for future research endeavors in AI-based media authentication, paving the way for more secure and trustworthy digital communication.

## DATA COLLECTION

Source of Data:
The data for this project was sourced from an extensive dataset available on **Kaggle[1]**, a platform known for hosting a wide variety of datasets that cater to the machine learning community. This dataset is specifically tailored for training models to differentiate between authentic and digitally altered facial imagery.
In addition to the Kaggle dataset, a set of personal images provided by my professor Chan-Jin Chung has been utilized for further real-world testing. This unique collection includes both authentic and digitally manipulated photographs, comprising a real-test case scenario to test the robustness of the model's predictive accuracy.

Dataset Composition:
The dataset consists of images classified into '**real**' and '**fake**' categories, encompassing various types of digital manipulations. It includes a diverse range of facial images across different demographics, providing a heterogeneous mix of age groups, ethnic backgrounds, and facial expressions.

Data Diversity:
The images are collected under various environmental conditions, featuring multiple lighting settings and background scenarios. This ensures that the model trained on this dataset can operate effectively in real-world conditions, which are often unpredictable and varied.

Preparation and Preprocessing:
Prior to usage, the images were preprocessed to maintain consistency in size and format. The dataset was divided into training(80%), validation(10%), and test sets(10%), ensuring a robust training phase while leaving a substantial portion of data untouched for model evaluation. This distribution allows for a comprehensive assessment of the model's performance and generalizability.


Figure 1. The Two Data Class Examples

| Dataset Info. | Total Data Size | "fake" labeled | "real" labeled | Training batch* |
|---|---|---|---|---|
| Training | 4000 | 2000 | 2000 | 32 |
| Validation | 500 | 250 | 250 | 10 |
| Testing | 500 | 250 | 250 | 10 |

Table 1. Dataset Information with image_dataset_from_directory

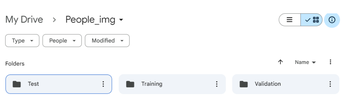



Figure 2. Dataset Environment (Google Drive)

## MODEL EXPERIMENTS

**Data Preparation**
Images are resized to 180x180 pixels for uniformity.
Batches of 32 images for training and 10 images for validation/testing.
Test data is unshuffled to maintain order for consistent evaluation.

**Data Augmentation**
To enhance the robustness of the model and prevent overfitting, a data augmentation strategy is implemented using Keras. This strategy includes random horizontal flips, rotations up to 10%, and zooming up to 20%. These transformations introduce variability in the training data, simulating different viewing conditions and angles, thereby enabling the model to generalize better on unseen images.

**Early Stopping:**
The training process incorporates an early stopping mechanism to optimize model performance and prevent overfitting. This technique monitors validation accuracy and halts training after three consecutive epochs without improvement to prevent overfitting.
**Different Models:**

**CNN Model :**
This model is a deep learning architecture specifically designed for image classification tasks. It consists of multiple layers of convolutional and pooling operations followed by fully connected layers for classification. CNNs are well-suited for tasks involving image analysis due to their ability to automatically learn hierarchical features from raw pixel data.

Architecture:
Input shape: (180, 180, 3)
Data Augmentation: Random horizontal flip, rotation, and zoom
Normalization: Rescaling pixel values to [0,1]
Convolutional layers with ReLU activation:
Increasing filter sizes (64, 128, 256,512) with kernel size 3
Max-pooling layers (2x2)
Flattening layer
Dropout regularization (rate: 0.5)
Fully connected layer with ReLU activation (512 neurons)
Output layer: Sigmoid activation for binary classification

Training Configuration:
Optimizer: RMSprop (learning rate: 1e-4)
Loss Function: Binary cross-entropy
Evaluation Metric: Accuracy

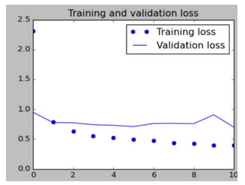
**2. VGG16 Fine-Tuned Model:**
I employed the widely-used VGG16 pre-trained model. Trained on the ImageNet dataset, it's known for its effectiveness. I fine-tuned it by keeping only the last four layers trainable, adapting it to my task of detecting fake and real faces. Data augmentation techniques like flips and rotations were used to diversify the dataset. With input images resized to 180x180 pixels and preprocessed accordingly, the model was trained to classify images into fake or real faces using a sigmoid activation function. With RMSprop optimization and binary cross-entropy loss, it achieved high accuracy in its classifications.

**3. MobileNetV2:**
MobileNetV2 is another powerful pre-trained model. I configured it to work with images sized 180x180 pixels and excluded its top layers. Then, I added some new layers to help classify images into two categories: fake and real faces. These new layers help the model learn from my specific dataset. After setting up the model, I trained it using the Adam optimizer and measured its performance using accuracy. This setup ensures that MobileNetV2 can accurately detect fake and real faces in my project.

## Experimental RESULTS

**Experiment I results:**
Based on the test data, the CNN model achieved an accuracy of 0.710. Originally trained for 30 epochs, the training process was halted at the 13th epoch due to the inclusion of an early stopping mechanism in the callback. This information is illustrated in Figure 3. Additionally, a confusion matrix was constructed to provide further insights into the model's performance, as depicted in Figure 4. For detailed information about the model architecture and configuration, refer to Table 2. The correctly predicted images as shown in Figure 5.("0" means fake, "1" means real)

**Experiment II results:**
During the training process of the VGG16 model, an early stopping mechanism was implemented to prevent overfitting. The training was conducted over 40 epochs, with each epoch taking approximately 22 to 24 seconds to complete. The model's accuracy gradually improved over the epochs(see Figure 6), reaching a peak accuracy of 71.40% at the end of training. This improvement is evident in the accuracy values reported for each epoch, as shown in the training log. Additionally, a ModelCheckpoint callback was used to save the best-performing model based on validation loss, ensuring that the model with the lowest validation loss was retained. Furthermore, a ReduceLROnPlateau callback was employed to dynamically adjust the learning rate if no improvement in validation loss was observed for a certain number of epochs. This setup helped optimize the training process and improve the model's generalization capabilities.



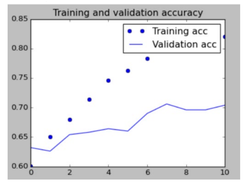




Figure 6. Graphs of Training Validation Loss/Accuracy of the Pre-trained model VGG16

**Experiment III results:**
The MobileNetV2 model achieved an accuracy of 63.20% on the test dataset. During training, an epoch-by-epoch breakdown reveals that the model's accuracy gradually improved over the course of 20 epochs. The training process utilized a learning rate scheduler, which dynamically adjusted the learning rate based on the epoch number to facilitate convergence. The model's training progress was monitored through a series of epochs, with each epoch taking approximately 16 to 24 seconds to complete. Early stopping was not employed in this scenario, allowing the training process to continue for the full 20 epochs. After training, the model was saved to disk for future use. Despite achieving a slightly lower accuracy compared to other models, MobileNetV2 demonstrates its effectiveness in balancing model complexity and performance.

| Model | Accuracy | Loss |
|---|---|---|
| CNN | 71.00% | 0.5620 |
| VGG16 | 71.40% | 0.7318 |
| MobileNetV2 | 63.20% | 0.8143 |

Table 2. Comparison of three models

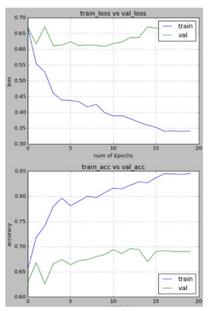



Figure 7. Graphs of Training Validation Loss/Accuracy of the MobileNetV2 model

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_4 (InputLayer) | [(None, 180, 180, 3)] | 0 |
| rescaling (Rescaling) | (None, 180, 180, 3) | 0 |
| conv2d (Conv2D) | (None, 178, 178, 64) | 1792 |
| max_pooling2d (MaxPooling2D) | (None, 89, 89, 64) | 0 |
| conv2d_1 (Conv2D) | (None, 87, 87, 128) | 73856 |
| max_pooling2d_1 (MaxPooling2D) | (None, 43, 43, 128) | 0 |
| conv2d_2 (Conv2D) | (None, 41, 41, 256) | 295168 |
| max_pooling2d_2 (MaxPooling2D) | (None, 20, 20, 256) | 0 |
| conv2d_3 (Conv2D) | (None, 18, 18, 512) | 1180160 |
| max_pooling2d_3 (MaxPooling2D) | (None, 9, 9, 512) | 0 |
| flatten_1 (Flatten) | (None, 41472) | 0 |
| dropout_6 (Dropout) | (None, 41472) | 0 |
| dense_9 (Dense) | (None, 512) | 21234176 |
| dense_10 (Dense) | (None, 1) | 513 |

Total params: 22785665 (86.92 MB)
Trainable params: 22785665 (86.92 MB)
Non-trainable params: 0 (0.00 Byte)

Table 2. Detail of the model with pure CNN

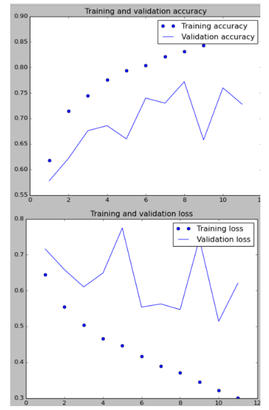



Figure 3. Graphs of Training Validation Loss/Accuracy of the CNN model

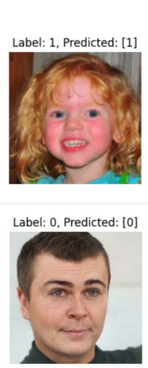



Figure 5. The correctly predicted images

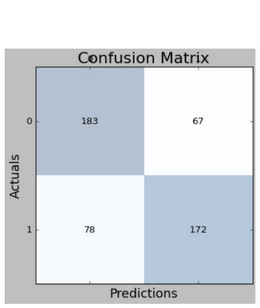



Figure 4. Confusion matrix for the CNN model.

## SUMMARY and Conclusion

**Pure CNN Performance:**
The project commenced with the development of a CNN model tailored for image classification tasks, specifically focused on distinguishing real faces from manipulated ones. Trained on a diverse dataset encompassing various demographic backgrounds and environmental conditions, the CNN model demonstrated promising performance metrics. Despite encountering challenges in discerning subtle differences between authentic and altered facial features, the model showcased fine accuracy levels.

**Comparison with VGG16 and MobileNetV2:**
Additionally, the project explored the efficacy of pre-trained models such as VGG16 and MobileNetV2 in achieving the desired classification outcomes. Through fine-tuning and adaptation to the task at hand, these models were evaluated alongside the custom CNN architecture. While each model exhibited distinct strengths and weaknesses, the comparative analysis provided valuable insights into their respective performance and suitability for the project's objectives.

**Addressing Validation Accuracy Discrepancies:**
Upon reviewing other completed projects, I found the discrepancy between training and validation accuracy across various models. While high training accuracy was often attained, validation accuracy typically hovered around fifty percent, indicative of potential overfitting and generalization issues. Despite this commonality among similar projects, I still made my efforts to optimize model architectures and training methodologies to improve validation accuracy and enhance model robustness. Therefore, I have hypothesized that human faces exhibit immense complexity and diversity, with each individual's facial features being distinct and influenced by factors such as age, ethnicity, gender, expression, and lighting conditions. This variability poses a significant challenge for facial recognition models. To accurately discern between genuine and fake faces, models must comprehend and capture the myriad features and subtle differences present in human faces. Consequently, it is conceivable that models may require a sufficient level of complexity. Additionally, for intricate facial datasets, models necessitate a larger volume of data for effective training. Extensive datasets encompassing various facial features can aid models in better generalizing to diverse scenarios, thereby enhancing accuracy.

In conclusion, my project represents a significant step towards combating digital deception in media through the development of AI-driven facial recognition solutions. While encountering challenges inherent to the dataset and model training process, the project underscores the importance of continuous refinement and adaptation in addressing contemporary digital forensics challenges. As the digital landscape evolves, the insights gleaned from this project pave the way for future advancements in AI-based media authentication, fostering a more secure and trustworthy digital communication ecosystem.

## REAL-WORLD TESTING

In a practical application scenario, Professor CJ Chung graciously provided eight personal photographs for testing purposes. These images, consisting of both genuine and AI generated faces, served as a benchmark to assess the real-world efficacy of our model. Impressively, the model's predictions were in perfect alignment with the provided labels, correctly identifying all the synthetic images as well as the authentic ones. This reinforces the model's potential for practical applications in distinguishing real from synthetic imagery. The complete results are illustrated in Figure 8 (0 means real, 1 means fake), demonstrating the model's robust detection capabilities. This outcome highlights the model's precision in detecting authentic versus digitally altered images and signifies a critical step towards enhancing its real-life application.

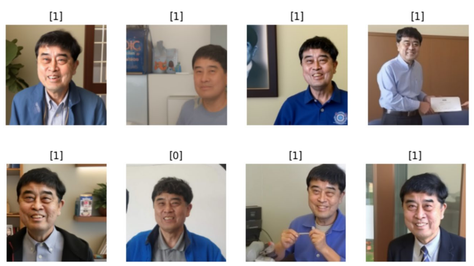



Figure 8. Predicted labels on Prof. Chung's real and synthetic images

## REFERENCES

1. VM7608. (2024). *Real vs. AI-Generated Faces Dataset* [Data set]. Kaggle. https://www.kaggle.com/datasets/philosopher0808/real-vs-ai-generated-faces-dataset
2. DEBASIS SAMAL. (2020). *Real vs. Fake Face Detection* [Code notebook]. Kaggle. https://www.kaggle.com/code/debasisdotcom/real-vs-fake-face-detection#MobileNetV2
3. Dharmaraj. (2022, June 1). Convolutional Neural Networks (CNN) — Architecture Explained. Medium. https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243
4. Jain, V. (2019, November 22). Everything you need to know about MobileNetV3. Medium. https://towardsdatascience.com/everything-you-need-to-know-about-mobilenetv3-and-its-comparison-with-previous-versions-a5d5e5a6eeaa
5. Boesch, G. (n.d.). VGG Very Deep Convolutional Networks (VGGNet) – What you need to know [Blog post]. viso.ai. https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/
6. OpenAI. (2023). ChatGPT [Computer software]. https://openai.com/ (Used for troubleshooting and checking the English grammar.)