Université d’Ottawa
Faculté de genie

uOttawa

L’Université canadienne
Canada’s university

University of Ottawa
Faculty of Engineering

ELG5901 Electrical Engineering Project

Final Report

Student name: Hassan,Hadir	ID: 30038989
Student name: Mahmoud,Nada	ID: 300389903
Student name: Attia,Nada	ID: 300389906
Student name: Beshay,Sandy	ID: 300389917

Graduate Program: AI and DataScience

Semester to Register:2023, Winter

Project Title: DEEP FAKE IMAGE DETECTION

Table of Contents:

Table of Figures:

Acronyms

Abbreviation	Meaning
CNN	Convolutional neural network
DFDC	Deepfake Detection Challenge
API	Application Programming Interface

1. *Introduction*

1.1 Problem Definition

The problem we aim to address is the rapid proliferation of deepfake images and the potential harm they can cause. Deepfake images have the potential to deceive, manipulate, and mislead individuals, leading to negative consequences in various domains, including politics, journalism, and personal relationships. Our project aims to develop an effective deepfake image detection system to mitigate these risks and promote trust and authenticity in visual media.

1.2 Background

In order to understand the problem of deepfake image detection and develop an effective solution, we have conducted extensive research and investigation into relevant technologies, academic papers, and industry references. Our literature review has provided us with a comprehensive understanding of the current state of the art in deepfake detection algorithms, image analysis techniques, and machine learning models. During our research, we came across several key papers from both academic and industry sources that have contributed significantly to the field of deepfake image detection. These papers include:

Academic Papers:

1. *"Comparative Analysis of Deepfake Image Detection Method Using Convolutional Neural Network" by Hasin Shahed Shad et al [1]. This paper presents a comparative analysis of different convolutional neural network (CNN) models for deepfake image detection. It provides insights into the performance and effectiveness of various CNN architectures in detecting deepfake images. By studying this paper, we gained knowledge about the strengths and limitations of different CNN models and make informed decisions regarding the choice of architecture for your own deepfake detection system.*
2. *"FaceForensics++: Learning to Detect Manipulated Facial Images" by Andreas Rössler et al [2]. This paper introduces the FaceForensics++ dataset and provides insights into the detection of manipulated facial images. It offers valuable information about detecting facial manipulations, which are commonly employed in deepfake generation. By studying this paper,we can gain insights into the techniques and methodologies used for detecting manipulated facial images, which can enhance the accuracy and robustness of your own deepfake detection system.*

Industry References:

1. *Deep Fake Image Detection" by Omkar Salpekar [3] presents techniques for detecting deepfake images using a 2-phase learning architecture. This paper proposes a specific approach and methodology for deepfake detection, including feature extraction and classification techniques.*

2. *"The Deepfake Detection Challenge (DFDC) Dataset" by Brian Dolhansky et al [4]. This paper introduces the DFDC dataset, which is a benchmark dataset for deepfake detection. It highlights the challenges associated with deepfake detection and provides a valuable resource for training and evaluating deepfake detection models.*

1.3 Project Context

External Systems, Third-Party Interfaces, APIs, and Tools:
During the completion of your project, you interacted with the following external systems, third-party interfaces, APIs, and tools:

1. *Streamlit Library: We used the Streamlit library, an open-source Python framework, to create a web page for your project. Streamlit enabled you to develop an interactive interface for the demonstration of your fake image detection system [5].*
2. *Kaggle: We obtained a dataset from Kaggle, specifically the "140k Real and Fake Faces" dataset. This dataset provided a collection of real and deepfake images, which you used for training and evaluating your deep learning models [6].*
3. *Deep Learning Models: We utilized various deep learning models, including VGG16, MobileNet, and ResNet, for your project. These models were accessed through frameworks such as TensorFlow or Keras, which provided the necessary interfaces and APIs for model development and evaluation.*

Individuals/Organizations:
To complete your project successfully, you needed to interact with the following individuals or organizations outside of your project team:
Wakeb Company (Sponsor): Wakeb Company served as your project sponsor. We collaborated with representatives from Wakeb Company, such as project managers or technical advisors, who provided support, guidance, and resources throughout the project. They were involved in defining project requirements, providing access to necessary resources, and offering domain-specific knowledge. The availability and accessibility of Wakeb Company were confirmed, ensuring their continued support throughout the project.

2. Design Overview

2.1 Requirements:

- *Dataset: The project requires a dataset of fake and real images for training and evaluation purposes. The "140k Real and Fake Faces".*
- *Image Preprocessing: The images in the dataset need to be preprocessed. This includes resizing the images.*
- *Training and Evaluation: The models need to be trained using the prepared dataset and evaluated based on test accuracy and loss. The goal is to achieve high accuracy and low loss values.*
- *Hyperparameter Tuning: The project involves tuning the hyperparameters of the model to optimize its performance. This includes adjusting the batch size, learning rate, and using techniques like early stopping.*

- *Evaluation and Visualization: The project requires evaluating the trained models using learning curves, confusion matrices, and other visualization techniques to analyze their performance.*
- *Web Demo: An interactive web demo for fake image detection. This requirement involves implementing a user-friendly web interface for users to upload and detect fake images.*

2.2 Detailed Design

- *Data Preparation: The dataset is downloaded and partitioned into parts, with approximately 10,000 images per partition. The images are then resized to 128x128 pixels.*
- *Model Selection: we use various deep learning algorithms, specifically CNNs (Convolutional Neural Networks) such as VGG16 , MobileNetV2 and ResNet50, for modeling and detection of deepfake images. The MobileNetV2 model is chosen as the champion model. It achieves the highest test accuracy and the lowest test loss among the tested models.*
- *Hyperparameter Tuning: The hyperparameters, such as batch size, learning rate, and early stopping, are adjusted to improve the model's performance.*
- *Implementation: The models are trained using different batch sizes (16, 32, 8, 4, and 2) with early stopping and learning rate reduction.*

2.3 Implementation

The implementation phase involves detailed steps to accomplish each aspect of the project, adhering to the requirements and design choices.

1. Data Preparation:

- *Dataset Download: Obtain the "140k Real and Fake Faces" .*
- *Data Partitioning: Divide the dataset into subsets, each containing around 10,000 images.*
- *Image Resizing: Standardize image dimensions by resizing them to 128x128 pixels.*

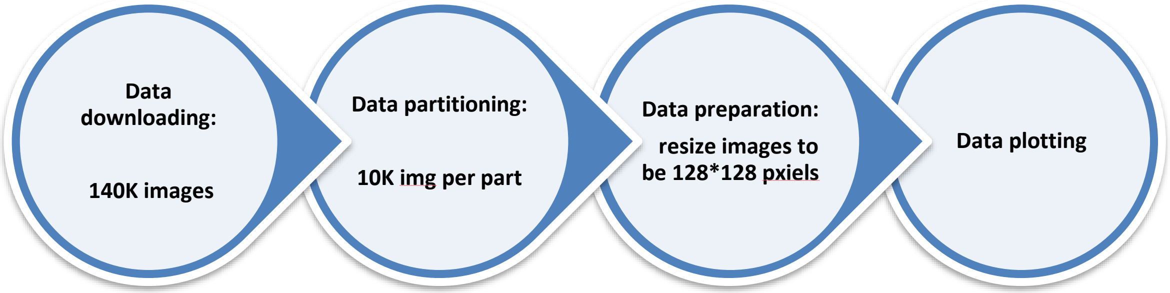


Figure 2: *Data Preparation*

2. Model Training:

- *Model Selection: Choose the MobileNetV2 architecture based on its superior performance in terms of test accuracy and low test loss.*
- *Experimentation: Train the model using various batch sizes (16, 32, 8, 4, 2) to observe the impact on performance.*

3. Hyperparameter Tuning:

- *Batch Size Adjustment: Experiment with different batch sizes during training (e.g., 16, 32, 8, 4, 2).*
- *Learning Rate Optimization: Adjust the learning rate for optimal model convergence.*
- *Early Stopping: Implement early stopping techniques to prevent overfitting.*

4. Evaluation and Visualization:

- *Learning Curves: Plot learning curves to visualize training and validation performance over epochs.*
- *Confusion Matrices: Generate confusion matrices to evaluate the model's ability to classify real and fake images.*

When we trained the MobileNetV2 model with a small-scale dataset, we obtained the following results:

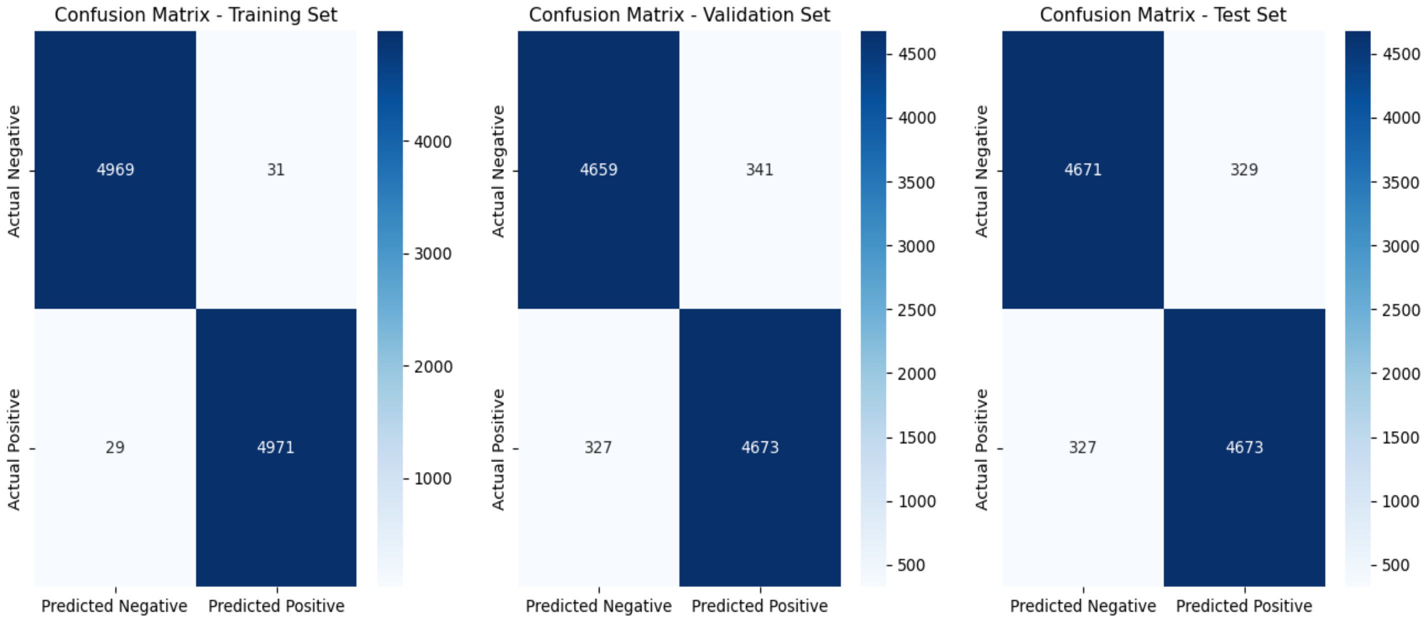


Figure 2: *Confusion Matrix of MobileNetV2 with a small-scale dataset*

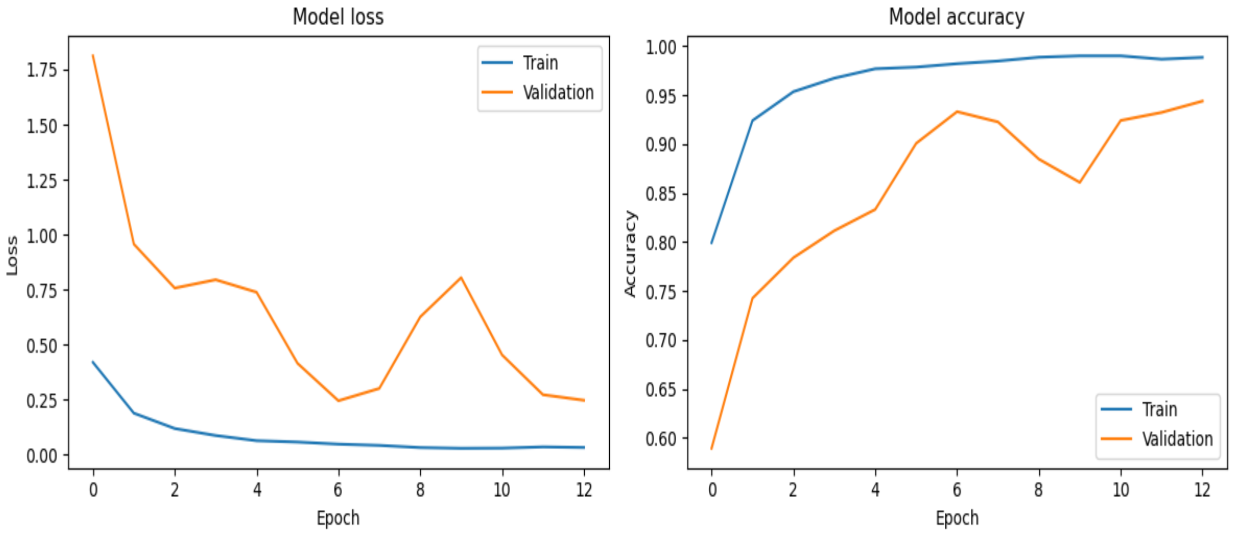


Figure 3: *Learning Curve of MobileNetV2 with a small-scale dataset*

And when we trained the MobileNetV2 model with a large-scale dataset, we obtained the following results:

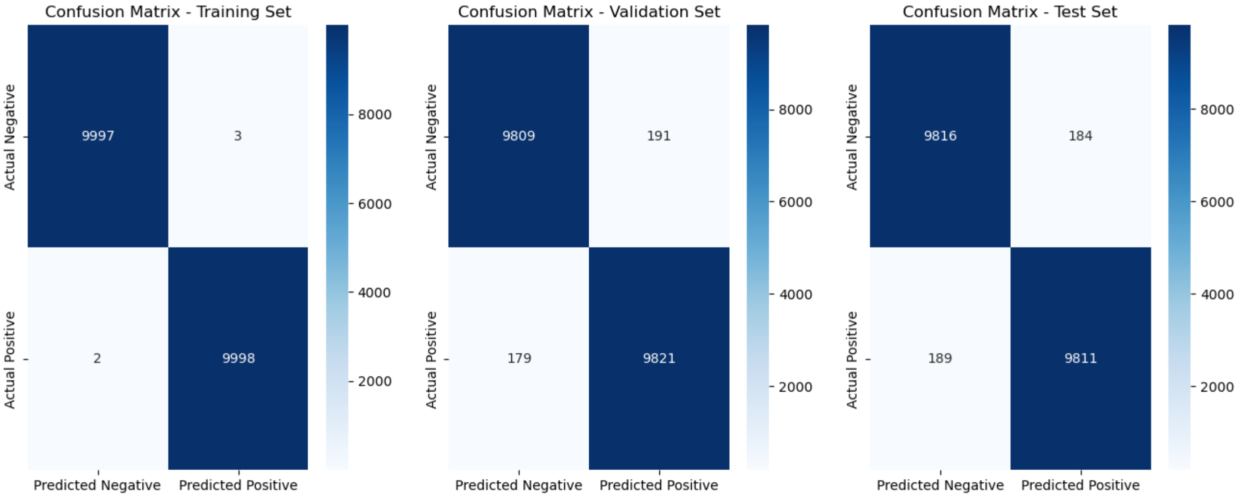


Figure 4: *Confusion Matrix of MobileNetV2 with a large-scale dataset*

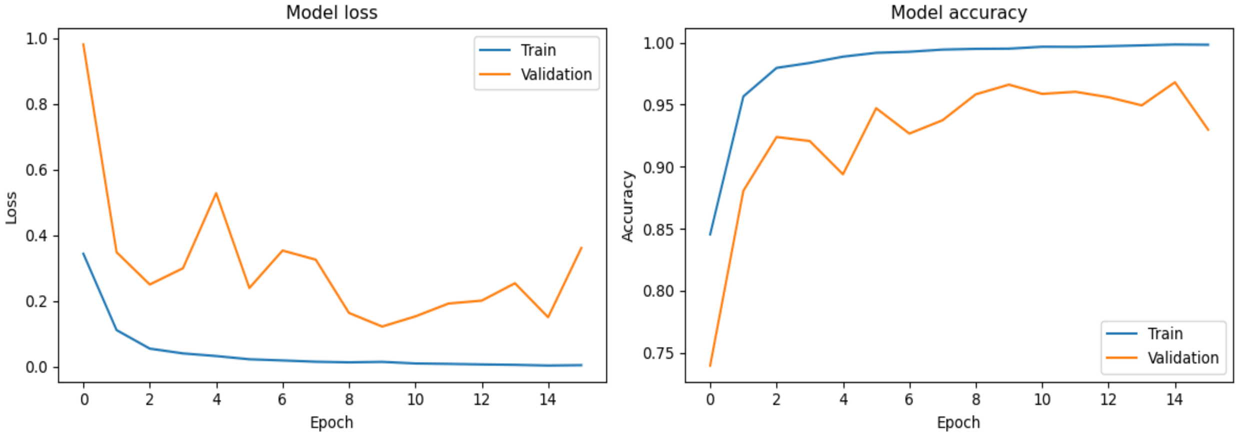


Figure 5: *Learning Curve of MobileNetV2 with a large-scale dataset*

5. Web Demo:

- *User Interface Development: Create an interactive web interface allowing users to upload images for fake image detection By StreamLit.*
- *Real-Time Prediction: Implement functionality to provide real-time predictions on the authenticity of uploaded images.*

2.4 Testing

2.4.1 Data Plan

The "140k Real and Fake Faces" dataset from Kaggle was used for this project, comprising 140,000 diverse real and fake facial images. This dataset was selected due to its large size, balanced composition of real and fake images, and high quality sourced from a reputable provider.

The images were partitioned into 10,000 image batches to facilitate organized training and testing. The full dataset was available throughout the project lifecycle to enable consistent model evaluation.

As preprocessing, all images were resized to 128x128 pixels to standardize the data and meet model input requirements. The dataset was also split 80/10/10 into train, validation, and test sets for model fitting and performance measurement.

2.4.2 Validation & Verification

A comprehensive validation and verification process was followed to ensure the deepfake detection model met the required design specifications and performance objectives:

- *Multiple deep learning model architectures including VGG16, MobileNetV2, and ResNet50 were evaluated on a subset of data to select the best performing model as the base architecture.*
- *Hyperparameter tuning was then conducted on the chosen MobileNetV2 model to optimize key parameters like batch size, number of epochs, activation functions, and learning rate scheduling.*

- *The model was trained on the majority of the 140k image dataset. Model skill was quantified on the smaller validation set at regular intervals during training to tune hyperparameters and minimize overfitting.*
- *Key skill metrics tracked included accuracy, confusion matrix, and loss function. Logging of training and validation losses identified overfitting.*
- *The optimized model was finally evaluated on the held-out test set (10% of data) to get an unbiased estimate of model skill at identifying real vs fake images. The final model achieved a test accuracy of 98.14% and test loss of 0.08, meeting the target criteria.*
- *Additional testing was done by passing individual real and fake images through the trained model and visually inspecting the outputs. An interactive web application was created to facilitate this qualitative testing.*

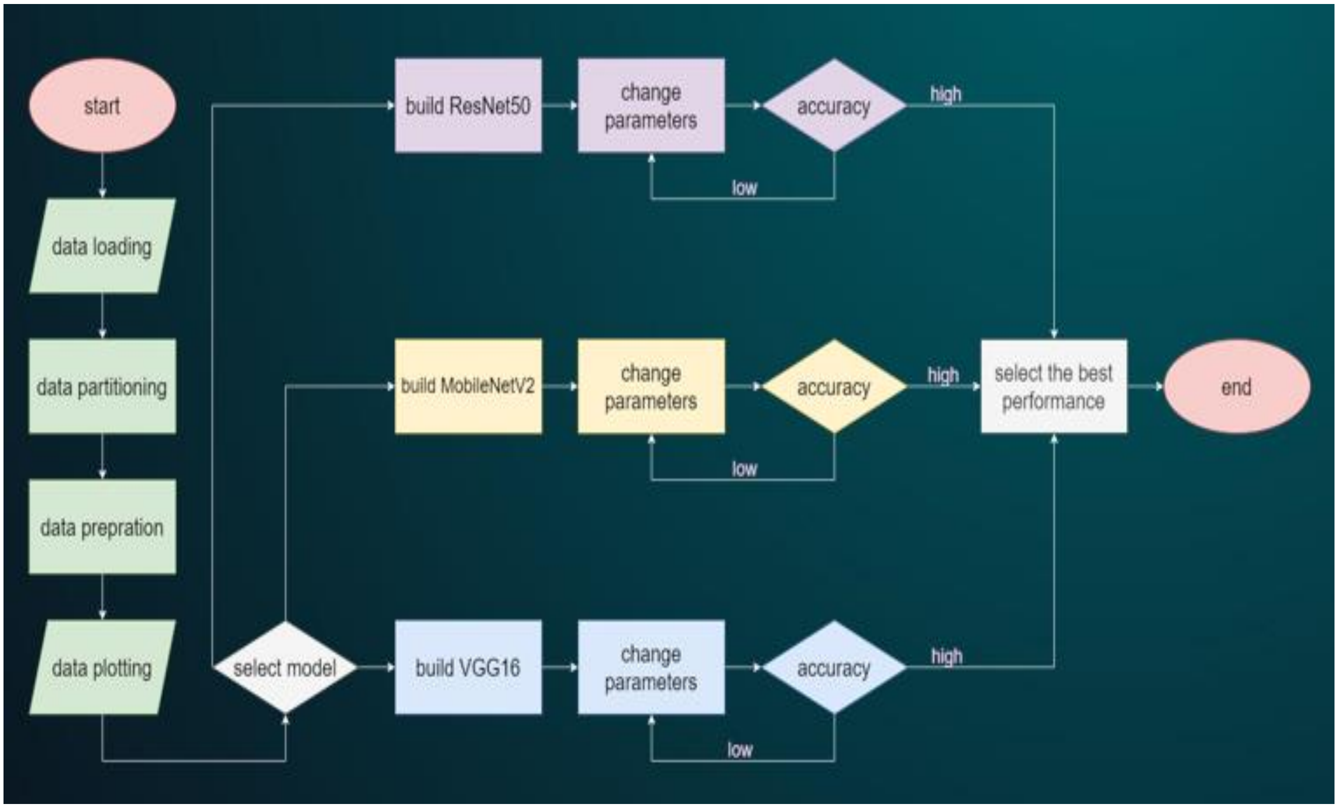


Figure 6: Project walkthrough

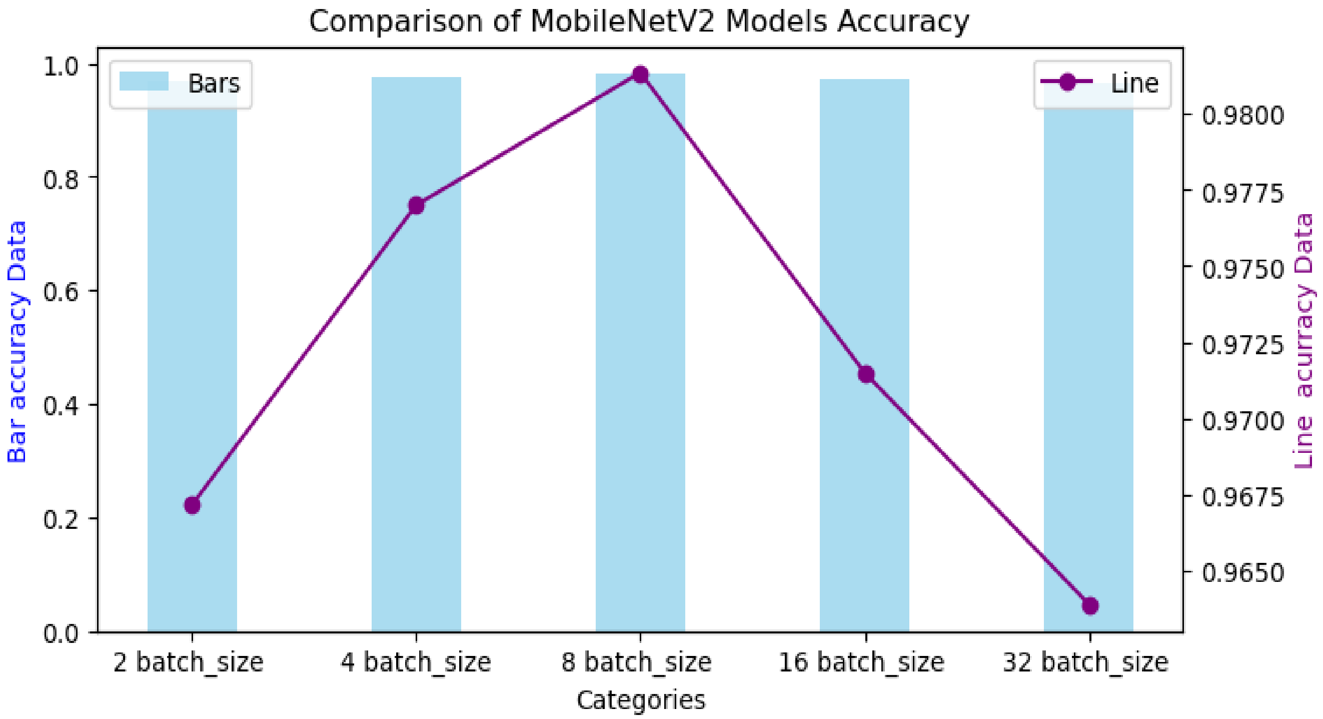
3. overall Results and Analysis:

Project Results:

The project successfully achieved its primary objective of developing an efficient and accurate deepfake detection system. The comparative study between VGG16, ResNet50, and MobileNetV2 provided valuable insights, with MobileNetV2 emerging as the most effective model in terms of accuracy and computational efficiency. The implementation of a web demo further enhanced the practical applicability of the project.

Positive Aspects:

- *High Accuracy with MobileNetV2: Achieving a test accuracy of 98.14% with MobileNetV2 was a significant accomplishment, surpassing initial expectations.*
- *Efficient Model Performance: MobileNetV2's efficiency in terms of computational resources made it ideal for real-time applications.*
- *Comprehensive Comparative Analysis: The project offered a thorough comparison of different models, providing a deeper understanding of deep learning applications in deepfake detection.*



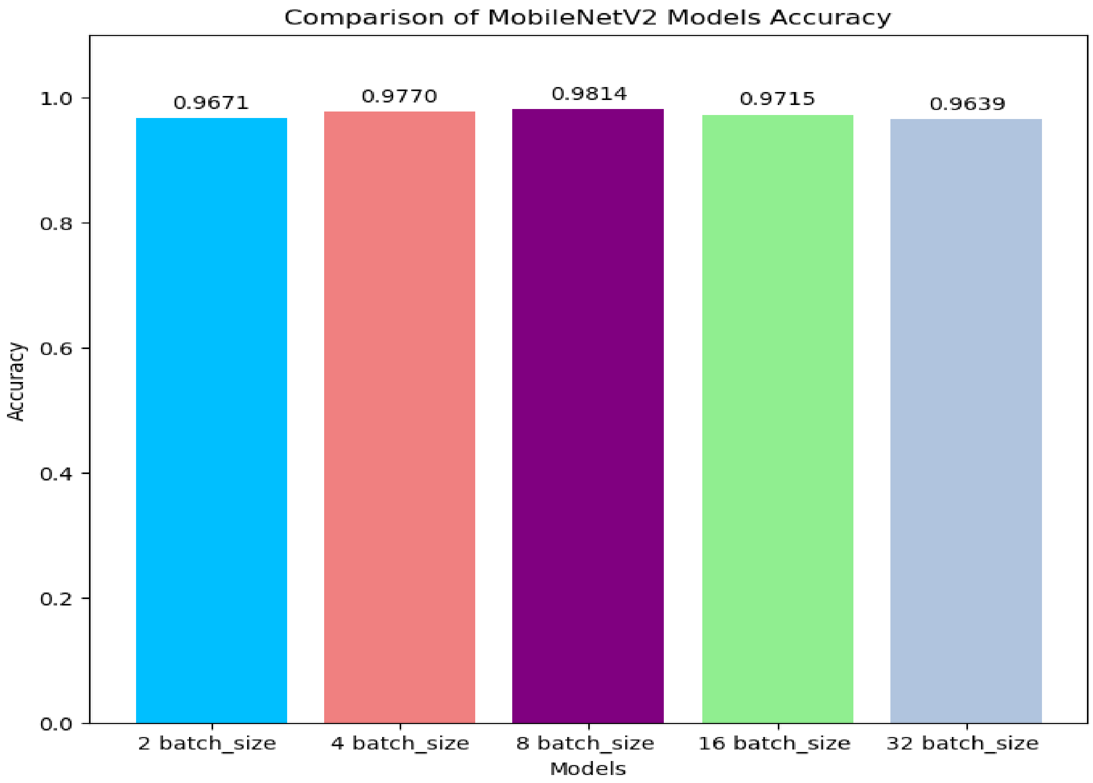


Figure 7: Comparison of MobileNetV2 accuracies

Challenges and Unexpected Efforts:

- *Data Preprocessing: The preprocessing phase, particularly image resizing and normalization, was more time-consuming than anticipated. Handling a large dataset required considerable effort in terms of data management and processing.*
- *Model Training and Tuning: Finding the right balance in hyperparameters for each model was challenging and required extensive experimentation, which was time-intensive.*
- *Handling Overfitting: Ensuring that the models did not overfit, especially with such a large dataset, required careful implementation of techniques like early stopping, which added complexity to the model training process.*

Potential Improvements:

- *Advanced Data Augmentation: Implementing more sophisticated data augmentation techniques could potentially improve model robustness.*
- *Exploring Other Architectures: Investigating other neural network architectures, such as transformer-based models, might yield better performance or insights.*
- *Optimized Hyperparameter Tuning: Utilizing automated hyperparameter optimization tools like Bayesian Optimization could streamline the tuning process.*

Impact on Learning Outcomes and Career Objectives:

This project significantly contributed to our understanding of deep learning and its applications in digital forensics. It honed our skills in data preprocessing, model selection, training, and evaluation all of which are crucial competencies in the field of AI and Machine Learning. The project's success aligns well with my career objectives, positioning me as a knowledgeable professional in AI applications for cybersecurity and digital media integrity.

Overall Evaluation of Project Success and Outcomes:

The project was a success, meeting its intended goals with high levels of efficiency and accuracy. It provided practical insights into the realm of deepfake detection, an area of growing importance in the digital age. The experience and knowledge gained through this project are invaluable and will undoubtedly contribute to my future endeavors in technology and digital forensics. The challenges encountered were instrumental in providing a comprehensive learning experience, emphasizing the importance of persistence and innovation in problem-solving. The project outcomes not only fulfill academic requirements but also pave the way for future research and career development in a rapidly evolving technological landscape.

4. Deployment Plan

Environment Setup and Requirements:

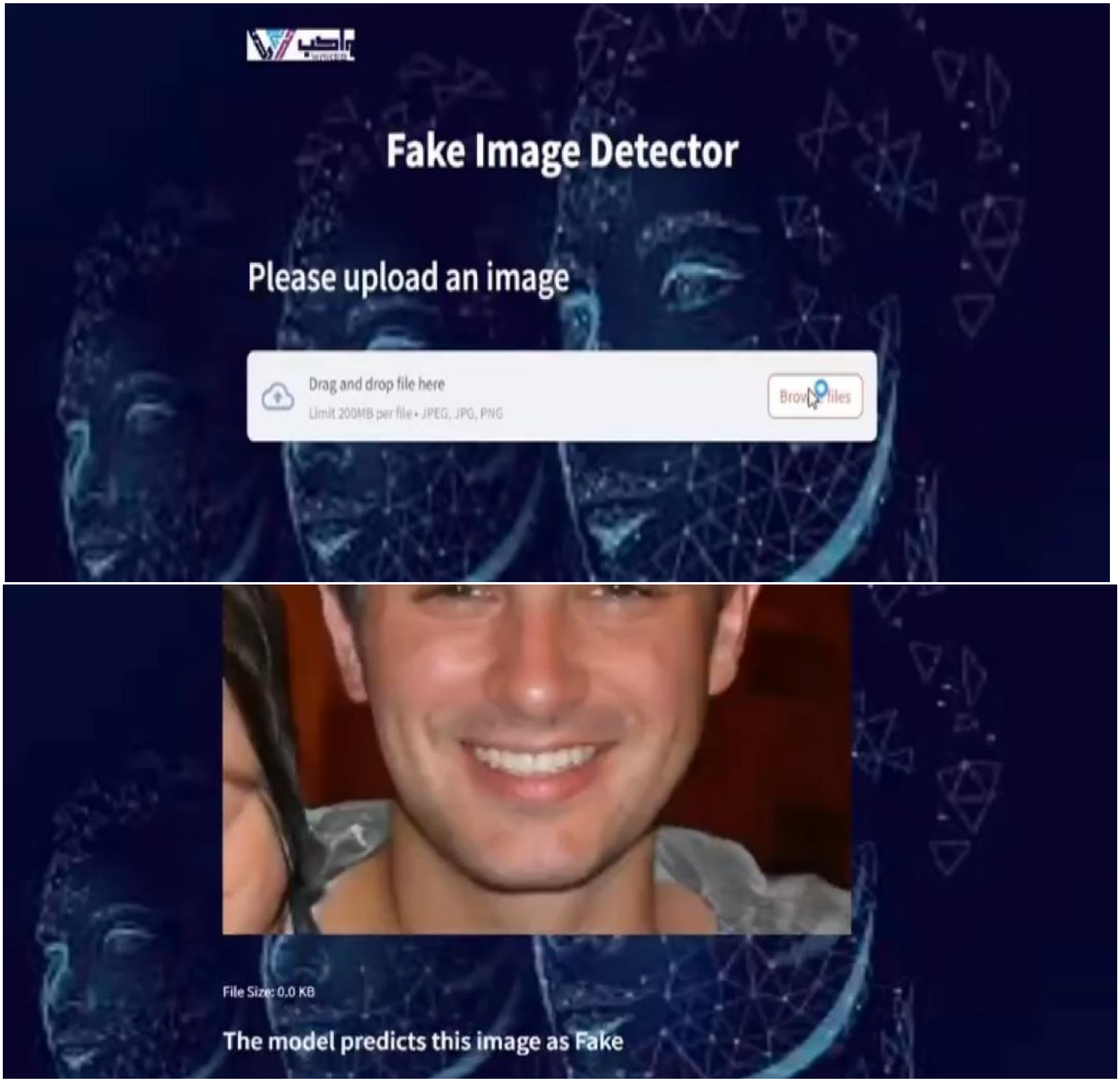
- *Server Infrastructure: A robust server setup is required to host the model and the web application. This includes adequate processing power and memory to handle concurrent requests.*
- *Software Dependencies: Ensure all necessary software, including the deep learning frameworks (like TensorFlow or PyTorch).*

Model Deployment:

- *Model Conversion: Convert the MobileNetV2 model into a format suitable for deployment, such as TensorFlow’s Saved Model.*
- *Load Testing: Conduct load testing to ensure the model can handle a realistic number of requests without significant latency or downtime.*

Web Application Deployment:

- *User Interface: Ensure the web demo (developed using StreamLit) is intuitive and user-friendly. It should allow users to easily upload images and view the results.*
- *Security Measures: Implement security protocols to protect user data and prevent unauthorized access to the system.*
- *Cross-Platform Compatibility: Test the web application across various devices and browsers for compatibility issues.*



Fake Image Detector

Please upload an image

Drag and drop file here

Limit 200MB per file • JPEG, JPG, PNG

File Size: 0.0 KB

The model predicts this image as Fake

Figure 8: Web application

5.Conclusions and Future Works

In this project, we conducted a comprehensive comparative study of deepfake detection techniques using advanced deep learning models. We focused on evaluating the performance of VGG16, ResNet50, and MobileNetV2 classifiers using a dataset of 140,000 real and fake faces. The project's crowning achievement was the identification of MobileNetV2 as the most effective model, achieving a remarkable test accuracy of 98.14% and the lowest test loss of 0.0800. This achievement is significant in the realm of digital forensics and contributes substantially to the ongoing efforts in combating deepfake technology.

The successful implementation of the deepfake detection system, particularly its integration into a user-friendly web interface, represents a significant step forward in providing accessible tools for identifying manipulated media. This project not only meets the immediate need for effective deepfake detection but also sets a benchmark for future research in this rapidly evolving field.

Next Steps and Future Research Directions:

- *Enhancing Model Accuracy: While MobileNetV2 has shown impressive results, there is always room for improvement. Future work could explore the integration of more sophisticated neural network architectures or ensemble methods to further enhance accuracy.*

- *Real-Time Detection: Developing a more efficient system capable of real-time deepfake detection is a crucial next step. This would involve optimizing the model for faster processing without compromising accuracy.*

- *Expanding Dataset Diversity: Incorporating a more diverse set of images, including those from different demographic groups and varied lighting conditions, could improve the model's robustness and reduce bias.*

- *Audio-Visual Deepfake Detection: Extending the project to include audio deepfake detection could provide a more comprehensive solution, as deepfakes often encompass both visual and auditory elements.*

- *Adapting to New Deepfake Techniques: As deepfake technology evolves, it's essential to continuously update the detection methods. Future research could focus on developing adaptive algorithms that can quickly learn and identify new deepfake techniques.*

- *Ethical and Legal Considerations: Exploring the ethical and legal implications of deepfake detection, particularly in terms of privacy and data security, is another important future direction.*

- *Deployment in Varied Domains: Customizing the solution for specific industries, such as media, law enforcement, or social media platforms, could make the tool more effective in different operational environments.*

- *User Feedback Integration: Incorporating user feedback into the system's development process can lead to improvements in usability and functionality.*

6.References

[1] H. S. Shad, et al., "Comparative Analysis of Deepfake Image Detection Method Using Convolutional Neural Network," ResearchGate, 2021. Available: https://www.researchgate.net/publication/357123371_Comparative_Analysis_of_Deepfake_Image_Detection_Method_Using_Convolutional_Neural_Network.

[2] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Rossler_FaceForensics_Learning_to_Detect_Manipulated_Facial_Images_ICCV_2019_paper.html

[3] O. Salpekar, "Deep Fake Image Detection," Stanford University CS230, Project Report, 2020.Available: http://cs230.stanford.edu/projects_spring_2020/reports/38857501.pdf.

[4] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, C. Canton Ferrer, "The Deepfake Detection Challenge (DFDC) Dataset," arXiv:2006.07397,June 2020. Available: https://arxiv.org/abs/2006.07397.

[5] Streamlit, "Streamlit Documentation," Streamlit, Available: https://docs.streamlit.io/

[6] Gaurav DuttakIIT. (n.d.). 140k Real and Fake Faces [Dataset]. Kaggle. Retrieved from https://www.kaggle.com/datasets/gauravduttakiit/140k-real-and-fake-faces