Monday 20th, May 2024
Master 1 Computer Science

Project Report on "Chirurgie plastique" Face forgery detection methods

Members :
Gaëtan COULOMBIER
Aref ELAGGOUN
Nathan FRANCLET
Alex SUMAQIE

Tutors :
Christophe CHARRIER
Emmanuel GIGUET

Jury :
Christophe CHARRIER
Emmanuel GIGUET
Adeline Roux-Langlois
Matthieu Dien

UNIVERSITÉ
CAEN
NORMANDIE

Contents

1 Abstract

1.1 Brief Overview

This project focuses on developing and evaluating methods to detect deepfakes and differentiate them from genuinely generated faces. Although both deepfakes and generated faces are produced by generative models, their uses and ethical implications differ significantly. This project aims to create or make use of already existing detection mechanisms that can accurately identify deepfakes.

1.2 Objectives

This briefly shows our procedure and objectives during the course of the project.

- Research the best Deepfake detection methods.
- Evaluate and compare detection methods to choose the best.
- evaluate models in detail (accuracy,recall,confusion matrix etc...) to identify the best.

1.3 Key findings

- Effectiveness:
  - GANs : Demonstrated high accuracy in distinguishing deepfakes.
  - RNNs : Showed strength in detecting temporal inconsistencies in deepfake videos.
  - CNNs : Performed well in identifying subtle artifacts in static images.
- Performance Metrics:
  - Achieved good test results while evaluating models across different datasets.

2 Introduction

In recent years, the rapid development of Artificial Intelligence (AI) and Machine Learning has facilitate the creation of deepfakes. These are highly realistic fake videos or images where someone's face or voice is manipulated to say or do things they never actually did. Although this technology can be fun and impressive, it also raises serious concerns about privacy, security, and trust. As deepfakes become more common and harder to detect, finding reliable solutions to identify them is more important than ever.

Deepfakes are usually made using Generative Adversarial Networks (GANs), which are AI models consisting of two parts: a generator and a discriminator. The generator creates fake media, and the discriminator tries to tell the difference between real and fake content. This back-and-forth process makes the fake media look very realistic due to the constant training, making it difficult for people and basic detection tools to spot the fakes.

Our project aims to develop and evaluate detection methods to accurately identify deepfakes. We will compare various techniques like GANs, RNNs, CNNs and improve existing programs to reliably identify deepfakes with high accuracy. By delving into noise patterns, GAN methods, and advanced classifiers, our goal is to contribute to the evolving field of image forensics, providing a nuanced understanding of the authenticity of images. To ensure that our methods work well, we will test them on different datasets the effective of our detection system across various types of deepfakes.

2.1 Objectives

To dive into our objectives with more details, we split them into two parts : the goals for the first half of our project and what we achieved by the end.

2.2 First half

- Extensive research on existing detection methods for deepfakes.
- Research on existing GitHub projects that cover deepfake detection methods.
- Comparing the results of various deepfake detection methods between different GitHub projects to choose the best.

2.3 Second half

- Implementing detection methods from GitHub.
- Testing different projects while modifying them to meet our needs.
- Achieving the highest detection accuracy across different datasets.

2.4 Key differences

The first half of the project focused on extensive research. We examined countless of papers on deepfake detection ,including those on GitHub, universities papers, news articles, and YouTube videos. Our main goal was to gain a deep and profound understanding of deepfakes and the methods available to best benefit our project.

The second half focused on implementation and modifying projects to meet our needs. We selected two projects, which we will discuss later, modified them to our needs, and evaluated our model's deepfake detection capabilities.

3 Deepfake detection methods

3.1 GANs, RNNs, CNNs

GANs, RNNs, CNNs... What are they exactly ? Here, we will provide a brief description of the terms we came across the most and used in our project.

- GANs (Generative Adversarial Networks) :
  - A type of AI model consisting of two neural networks (generator and discriminator) that compete against each other to create realistic synthetic data.
- RNNs (Recurrent Neural Networks) :
  - A type of neural network designed for sequential data, where connections between nodes form directed cycles to capture temporal dependencies.
- CNNs (Convolutional Neural Networks) :
  - A class of deep neural networks primarily used for processing structured grid data like images, employing convolutional layers to extract features.

3.2 Evaluating methods

- Accuracy Score :
  - The ratio of correctly predicted instances to the total instances.
- Recall score :
  - The ratio of correctly predicted positive observations to all actual positives.
- Confusion Matrix :
  - A table used to describe the performance of a classification model by displaying true positives, false positives, true negatives, and false negatives.
- ROC Curve (Receiver Operating Characteristic Curve) :
  - A graphical representation of a classification model's performance, plotting the true positive rate against the false positive rate at various threshold settings.
- AUC (Area Under the ROC Curve) :
  - A single scalar value summarizing the overall ability of the model to discriminate between positive and negative classes, with 1 being perfect and 0.5 being random.
- Precision Score :
  - The ratio of correctly predicted positive observations to the total predicted positives.
- F1 score :
  - The harmonic mean of precision and recall, providing a balance between the two metrics, especially useful for imbalanced datasets.

3.3 GANs in deepfake

Why are GANs so important in deepfake ? Our choice is always relies on GANs as they have the best models to evaluate deepfakes.However, we also said that GANs help creative deepfakes that eventually become hard for us to identify,so how are they helpful in detecting deepfakes ? Here is a brief explanation of how we can use them to our advantage.

- Training the Discriminator :
  - A discriminator, originally designed to differentiate between real and fake images, can be repurposed to focus solely on identifying deepfakes. It learns to recognize subtle artifacts and inconsistencies that are often present in generated media but not in real images or videos.
- Detecting Artifacts :
  - GAN-generated deepfakes often contain minor imperfections or artifacts that are not typically found in real media. These artifacts can include inconsistencies in texture, lighting, and facial expressions. The discriminator is trained to detect these artifacts, thereby identifying the deepfakes.
- Adversarial Robustness :
  - By continually refining the discriminator through adversarial training, it becomes increasingly proficient at spotting even the most subtle deepfake manipulations. This makes it a valuable tool in the arsenal against deepfake proliferation.

4 Detection method analysis

Detection methods, varying in efficiency, appeared in recent years. To provide an overview of the latest technologies and procedures adopted, we have listed the methods that could be used in our future experimental phase in the table below :

Detection type	**Method**	**Functioning principles**	**Performance**
GANs or DeepFakes	CNN [29]	Uses Convolutional Neural Networks for image analysis	Acc : 90%
DeepFakes	RNN [22]	Sequential data analysis	Acc : 74.08% - 97.51%
Face Swapping	MultiAtt [21]	Uses multiple attention mechanisms to focus on different parts of the face	Acc : 97.62% - 98.53%
Face Swapping	RECCE [14]	Uses a combination of different techniques to detect face swapping	Acc : 74.08% - 97.51%
GANs	SVM classifier[28]	Uses Support Vector Machines for the classification	Acc : 96.23%
GANs	Learned noise pattern [27]	Analyzes noise patterns in images done by AI	Acc : 89.2% - 99.7%
GANs	GAN Fingerprint [17]	Find the noise fingerprint of a GAN by analysing a set of images	Acc : 99.5%
GANs	CNN + LSTM (Long Short-Term Memory) [16]	Uses CNN and LSTM for sequential data analysis	Acc : 84.75% - 91.48%

Table 1: List of methods that are used in the detection of AI generated or modified images.

5 Datasets and limitations

5.1 Datasets

While doing extensive research on deepfakes, we also learned about the datasets that accompany detection methods. Datasets in deepfake detection are collections of real and manipulated media (images or videos) used to train and evaluate detection models. Popular datasets like FaceForensics++, Celeb-DF, and the DeepFake Detection Challenge Dataset contain diverse examples of deepfakes and genuine media, providing a comprehensive resource for developing robust detection algorithms. These datasets typically include annotated data with labels indicating whether a media instance is real or fake, facilitating supervised learning and benchmarking of deepfake detection techniques.

Dataset	**Content**
ImageNet	Images that are used for image classification and object detection
Google DFD	3068 fake videos
CelebA	200k celebrity images with alterations with 40 attributes to each image
Celeb-DF	590 original videos collected from YouTube and 5639 corresponding DeepFake videos
AFW	205 images with 468 faces
FaceForensics++	1000 original video sequences that have been manipulated in 4 different ways
Face2Face	1.8 million manipulated face images

Table 2: List of datasets that are used in the detection of AI generated or modified images.

5.2 Limitations

The methods shown in **Table 1** have limitations that depend on various parameters. Certain limitations often recur across different methods. Here is a non-exhaustive list that allows us to evaluate the performance of a method :

- Degraded performance for blurry images.
- Inappropriate for people with physical illness.
- Degraded detection performance of face images with closed eyes.
- Unable to generalize well to unseen deepfakes.
- Inappropriate for images with high compression levels.
- Computationally complex because of large feature vector space.

6 Implementation

After a good amount of research and comparing the results between already made projects and GAN models, we decided to implement two existing GitHub projects.

6.1 Project 1

The first project is The GAN Image Detection. It focuses on detecting GAN-generated images using a model called ResNet50 NoDown. The project evaluates the state-of-the-art in GAN image detection, specifically assessing images produced by Progressive Growing GAN and StyleGAN2 architectures.

6.1.1 What is ResNet50 NoDown ?

ResNet50 NoDown is a variant of the ResNet50 neural network architecture used for image recognition. To better understand it :

- ResNet50 : A deep learning model with 50 layers, designed to recognise images by learning various features (like edges, textures, etc.) from a large dataset.
- Downsampling Layers : In typical ResNet50, Downsampling layers (like pooling layers and convolutional layers with strides greater than 1) reduce the spatial resolution of the image while increasing the depth (number of feature maps).

What is NoDown?

- **NoDown** : In the ResNet50 NoDown model, some or all downsampling layers are modified or removed.
- **Purpose** : By keeping the image resolution higher through the network, this model retains more detailed information about the image.
- **Benefit** : For tasks like deepfake detection, where subtle artifacts and small differences

are crucial, maintaining higher resolution helps the model detect these fine details.

6.2 Project 2

The second project is The-GAN-Image-Detection focuses on identifying GAN-generated images using a deep learning model called DCT (Discrete Cosine Transform) + CNN. This model leverages DCT to extract features that enhance CNN's ability to detect GAN artifacts.

6.2.1 What is Discrete Cosine Transform ?

The Discrete Cosine Transform (DCT) is a mathematical technique used to convert spatial data (such as an image) into frequency data

- Frequency Representation : DCT transforms an image into its frequency components, separating the image into parts of differing importance with respect to the image's visual quality.
- Compression : It is widely used in image compression formats, such as JPEG, because it can compact most of the image's visually significant information into a few coefficients.
- Artefact Detection : By analysing the frequency domain, DCT helps in identifying subtle artifacts and inconsistencies that are common in GAN-generated images but not in real ones.

6.2.2 How DCT Works in the GAN-Image-Detection Project ?

1. **Image Transformation** : Each image is first transformed using DCT, converting it from the spatial domain (pixels) to the frequency domain.
2. **Feature Extraction** : This transformation highlights frequency anomalies, which are typical in synthetic images but less common in real ones.

3. **CNN Analysis** : The transformed data (DCT) are then fed into a Convolutional Neural Network (CNN) that is trained to detect these anomalies and distinguish between real and GAN-generated images.

6.3 Datasets and GANs

GANs used in the test dataset (from CNNDetection) of the projects :

- **BigGAN** : Known for generating high-resolution, high-quality images, primarily used for image synthesis tasks.
- **CRN** : Contextual Residual Network, focuses on generating images with detailed contextual information.
- **CycleGAN** : Specializes in image-to-image translation without requiring paired datasets (e.g., converting horses to zebras).
- **Deepfake** : Used to create realistic fake videos by swapping faces in existing footage.
- **GauGAN** : Enables converting sketches into photo realistic images, particularly landscapes.
- **IMLE** : Implicit Maximum Likelihood Estimation, focuses on improving image generation quality.
- **ProGAN** : Progressive GAN, generates images by progressively increasing the resolution during training.
- **SAN** : Self-Attention GAN, enhances image quality by focusing on relevant image parts.
- **SeeingDark** : Likely focuses on enhancing images taken in low-light conditions.

- **StyleGAN** : Known for its ability to generate highly realistic images with control over style variations.

- **StyleGAN2** : An improved version of StyleGAN, offering better image quality.

- **StarGAN** : Capable of multi-domain image-to-image translation.

Each of them has a balanced dataset of images. That means that if we have 1000 images in one dataset for example, then we have 500 real images and 500 fake images in the dataset, creating a perfectly balanced dataset. Having a balanced dataset is important :

1. **Avoiding Bias :** Ensures that the model does not become biased towards the more prevalent class. If one class were more frequent, the model might simply learn to predict that class more often.

2. **Accurate Recall :** Recall measures the ability of the model to correctly identify all positive samples. A balanced dataset ensures that the recall metric reflects the model's true performance across both classes.

3. **Reliable F1 Score :** The F1 score is the harmonic mean of precision and recall. A balanced dataset ensures that both precision (accuracy of positive predictions) and recall are accurately measured, providing a reliable F1 score.

4. **Effective Training :** Balanced datasets help in evenly distributing the learning process, preventing the model from over-fitting to the dominant class.

6.4 Challenges and solutions

So the projects in general were made well, the authors did a wonderful job at creating something that works well, however there were problems that accompanied these projects as nothing is perfect. Here are some of the problems that we have faced :

- Resource limitations (RAM, GPU) :
  - Challenge : Insufficient resources such as RAM and GPU can hinder the processing and training of large datasets.
  - Solution : Using parallel processing and reducing the size of the data set to efficiently manage resource constraints.
- Poorly commented code :
  - Challenge : Lack of comments or inadequate comments in the code can make projects difficult to understand and maintain.
  - Solution : Going through the code and methods provided to be able to understand them without the need of comments from the author.
- Reduced/low accuracy :
  - Challenge :The detection had reduced accuracy when dealing with images that were blurry
  - Solution : Using high-resolution models like ResNet50 NoDown to maintain higher image resolution through the network

7 Results and Discussion

7.1 Results

In this section, we will present and analyse the results obtained from our deepfake detection experiments. We will compare the performance of different models and their databases like presented previously. Key metrics like accuracy, recall, precision, F1-score, and AUC-ROC will be used to evaluate the effectiveness of each method. In addition, we will discuss the implications of these results, highlighting the strengths and limitations of each approach, and suggesting potential areas for future improvement.

Table 3: Comparison of Deepfake Detection Methods (Part 1)

Method	Metric	Project 2	Project 1
5	Accuracy	0.794	0.968
	Recall	0.648	0.938
BigGAN	Precision	0.915	0.998
	F1-score	0.759	0.967
	AUC-ROC	0.79	0.97
10	Accuracy	0.786	0.984
	Recall	0.874	0.970
CRN	Precision	0.743	0.998
	F1-score	0.803	0.984
	AUC-ROC	0.79	0.98
15	Accuracy	0.884	0.915
	Recall	0.838	0.832
CycleGAN	Precision	0.923	0.998
	F1-score	0.878	0.907
	AUC-ROC	0.88	0.91
20	Accuracy	0.623	0.502
	Recall	0.932	0.004
Deepfake	Precision	0.576	1.000
	F1-score	0.712	0.008
	AUC-ROC	0.62	0.50
25	Accuracy	0.635	0.904
	Recall	0.360	0.808
GauGAN	Precision	0.800	1.000
	F1-score	0.497	0.894
	AUC-ROC	0.64	0.90
30	Accuracy	0.790	0.993
	Recall	0.864	0.988
IMLE	Precision	0.753	0.998
	F1-score	0.804	0.993
	AUC-ROC	0.79	0.99
35	Accuracy	0.864	0.999
	Recall	0.748	0.998
ProGAN	Precision	0.974	1.000
	F1-score	0.846	0.999
	AUC-ROC	0.86	1.00

Table 4: Comparison of Deepfake Detection Methods (Part 2)

Method	Metric	Project 2	Project 1
40	Accuracy	0.666	1.000
	Recall	0.998	1.000
StarGAN	Precision	0.600	1.000
	F1-score	0.749	1.000
	AUC-ROC	0.67	1.00
45	Accuracy	0.532	0.507
	Recall	0.082	0.018
SAN	Precision	0.818	0.800
	F1-score	0.149	0.036
	AUC-ROC	0.53	0.51
50	Accuracy	0.917	0.992
	Recall	1.000	1.000
SeeInDark	Precision	0.857	0.984
	F1-score	0.923	0.992
	AUC-ROC	0.92	0.99
55	Accuracy	0.815	1.000
	Recall	0.672	1.000
StyleGAN	Precision	0.941	1.000
	F1-score	0.784	1.000
	AUC-ROC	0.82	1.00
60	Accuracy	0.784	1.000
	Recall	0.594	1.000
StyleGAN2	Precision	0.958	1.000
	F1-score	0.733	1.000
	AUC-ROC	0.78	1.00
65	Accuracy	0.991	1.000
	Recall	0.990	1.000
WhichFaceIsReal	Precision	0.992	1.000
	F1-score	0.991	1.000
	AUC-ROC	0.99	1.00

7.2 Result discussion

In this section, we will present and analyze the results obtained from our deepfake detection experiments. We compare the performance of different models and their respective datasets, focusing on key metrics such as accuracy, recall, precision, F1-score, and AUC-ROC.

7.2.1 Overall Performance Comparison

Project 1 generally outperforms Project 2 across most models, demonstrating higher accuracy and more consistent results. Models such as BigGAN, CRN and CycleGAN in Project 1 show significantly higher precision and recall compared to Project 2, indicating better detection capabilities and fewer false positives and false negatives.

7.2.2 Key Metrics Analysis

- **Accuracy** : Project 1 consistently achieves higher accuracy across models like BigGAN (0.968) and CRN (0.984) compared to Project 2, indicating more reliable overall performance.

- **Recall and Precision** : Project 1 excels in recall and precision for most models, highlighting its effectiveness in correctly identifying both real and fake images. For instance, the recall for CRN in Project 1 is 0.970, while in Project 2, it's 0.874.

- **F1-Score** : Reflecting a balance between precision and recall, the F1-scores in Project 1 are superior, showing robust performance across various scenarios.

- **AUC-ROC** : High AUC-ROC values in Project 1, such as 0.97 for BigGAN, indicate excellent model discrimination capabilities between real and fake images.

7.2.3 Strengths and weaknesses

Now after talking about some of the Key metrics, let's dive into the general aspect of the strengths and weaknesses of each of the projects that we have done.

Strengths of Project 1:

- Higher Precision and Recall : Indicates fewer false positives and negatives.
- Robust Performance : Consistent results across different models and metrics.
- High AUC-ROC Values : Demonstrates strong model performance in distinguishing between classes.

Weaknesses of Project 1:

- Resource Intensive: Higher accuracy might come at the cost of greater computational resources.

Strengths of Project 2:

- Good Precision in Certain Models: For example, high precision in models like CycleGAN.
- Adaptability: Shows potential for improvement with further optimization.

Weaknesses of Project 2:

- Lower Overall Accuracy: Less reliable performance across various models.
- Variable Recall: Indicates potential issues in consistently identifying deepfakes.

7.3 Limitations and Improvements

Project 1:

- Limitations: Primarily resource-intensive, may require high computational power.
- Improvements: Explore more efficient model architectures to reduce computational load and further parallelize the code.

Project 2:

- Limitations: Lower performance metrics Compared to project 1,indicating a need of improvement.
- Improvements: Enhance model training processes and explore advanced techniques like data augmentation for example to boost accuracy and consistency.

8 Conclusion and Future Work

8.1 Conclusion

In conclusion, this project has been a highly engaging and intellectually stimulating adventure. Understanding the intricacies of deepfake detection and each model's unique methodologies required significant patience and dedication. Through extensive research and comparison of two distinct projects, we identified the strengths and weaknesses of each. Project 1, with its higher accuracy and precision, proved more effective in detecting deepfakes compared to Project 2, which exhibited lower performance metrics despite a balanced dataset. This comparison underscored the importance of both robust model design and resource optimization in achieving reliable deepfake detection.

8.2 A quick summary

- Project 1 : Achieved high accuracy and robust performance across multiple models, proving effective in deepfake detection.

- Project 2 : Demonstrated potential with some models but requires further optimization for consistent results.

8.3 A quick reflection

- Project 1 : Sets a high benchmark for deepfake detection, contributing valuable insights into effective detection methods.

- Project 2 : Highlights the need for continual improvement and optimization in deepfake detection technologies.

8.4 Potential future work that we could be done on the project

- Enhanced Models : Explore advanced architectures and techniques to improve detection accuracy and efficiency.
- Ethical Considerations : Investigate the ethical implications of deepfake detection and ensure responsible usage.
- Cross-Dataset Validation : Extend testing to more diverse datasets to ensure generalizability of the models.

8.5 Work distribution

There wasn't a strict "work distribution" in our group. Research was conducted collaboratively while we were on call together in general. Everyone contributed by searching, reading, and posting their findings in our collective channel. Similarly, we worked on the projects together, often favoring the person with the highest computational power to run the processes on their PC. The rest of the group provided guidance and helped with setting up and troubleshooting, ensuring that the entire team was engaged and contributing throughout the project.

9 Appendix

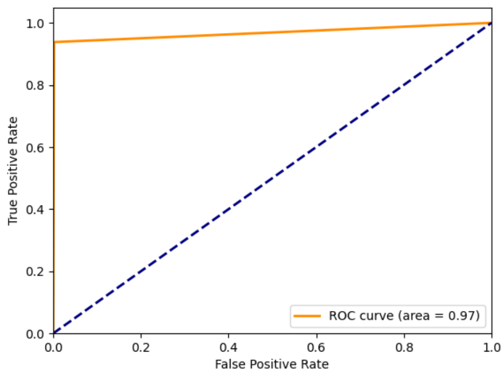

Figure 1: BigGAN Curve Project 1

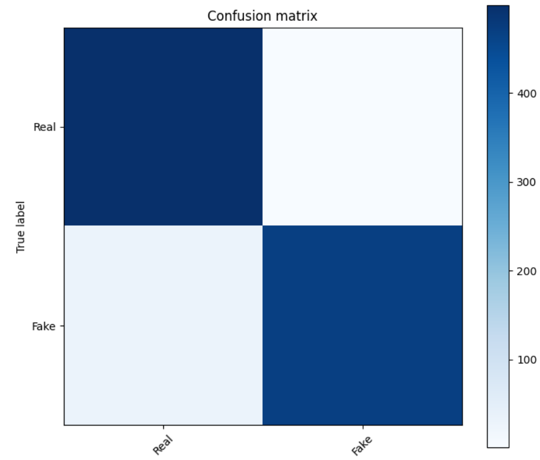

Figure 2: BigGAN Matrix (Project 1)

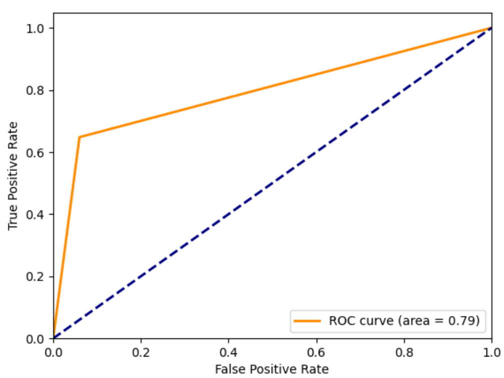

Figure 3: BigGAN Curve (Project 2)

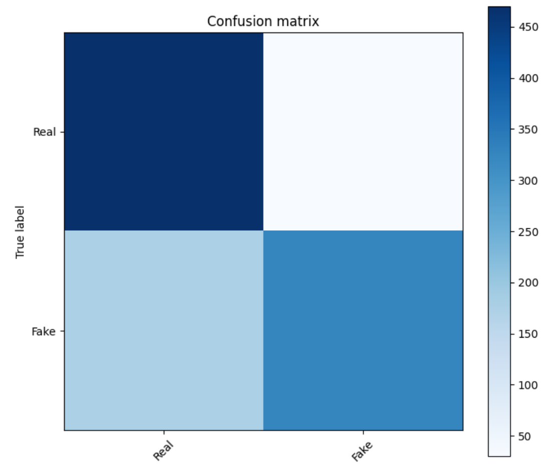

Figure 4: BigGAN Matrix (Project 2)

Figure 5: BigGAN Results

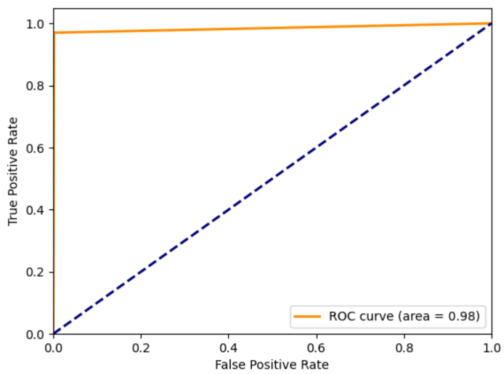

Figure 6: CRN Curve (Project 1)

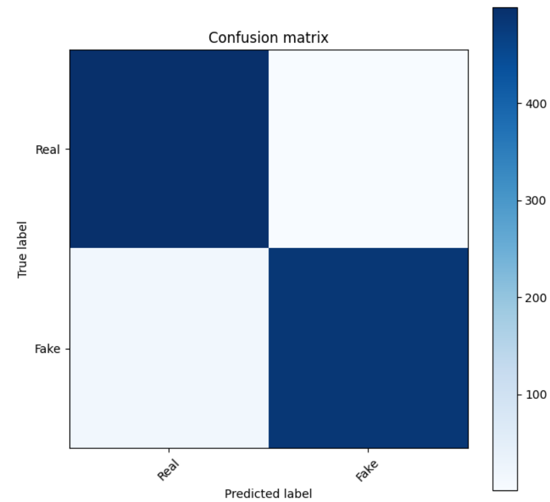

Figure 7: CRN Matrix (Project 1)

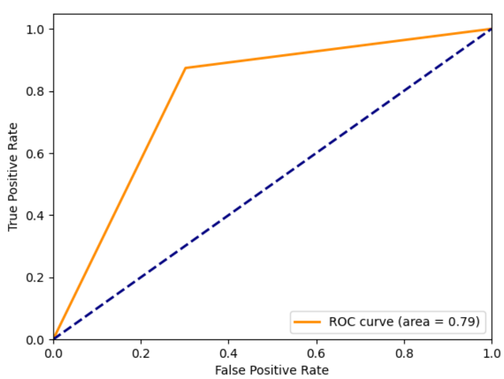

Figure 8: CRN Curve Project 2

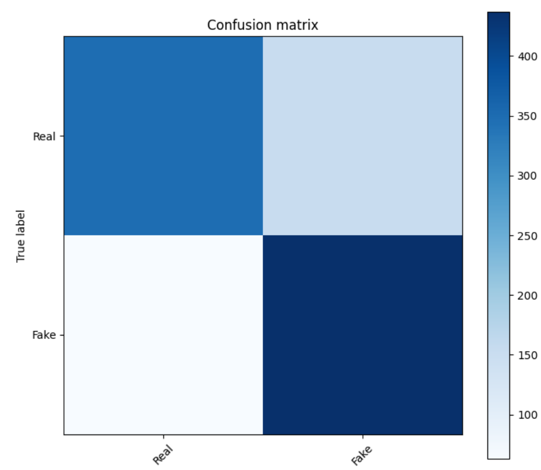

Figure 9: CRN Matrix Project 2

Figure 10: CRN Results

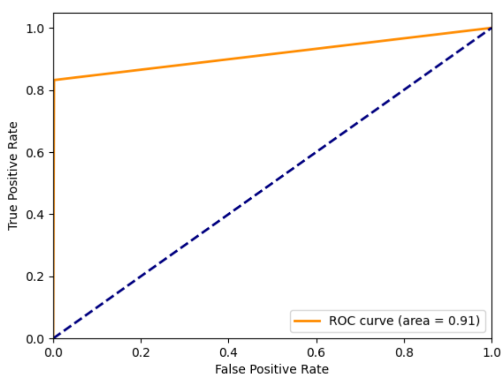

Figure 11: CycleGAN Curve (Project 1)

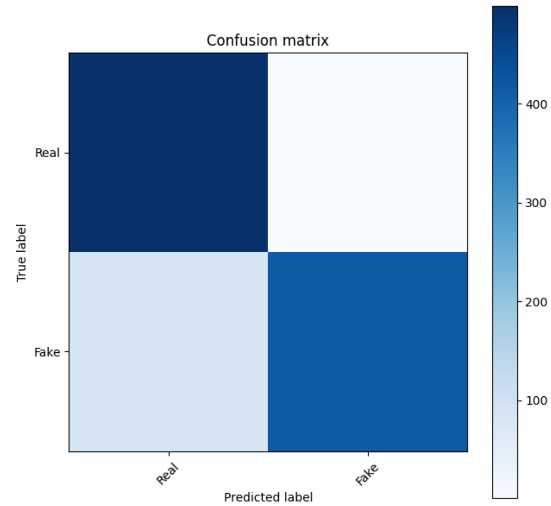

Figure 12: CycleGAN Matrix (Project 1)

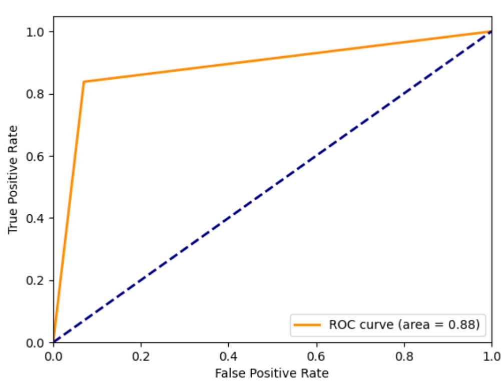

Figure 13: CycleGAN Curve (Project 2)

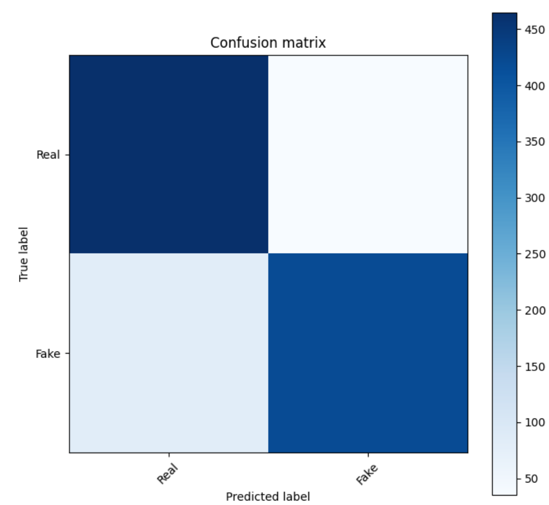

Figure 14: CycleGAN Matrix (Project 2)

Figure 15: CycleGAN Results

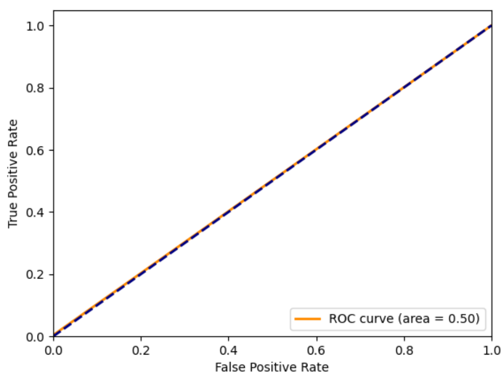

Figure 16: Deepfake Curve (Project 1)

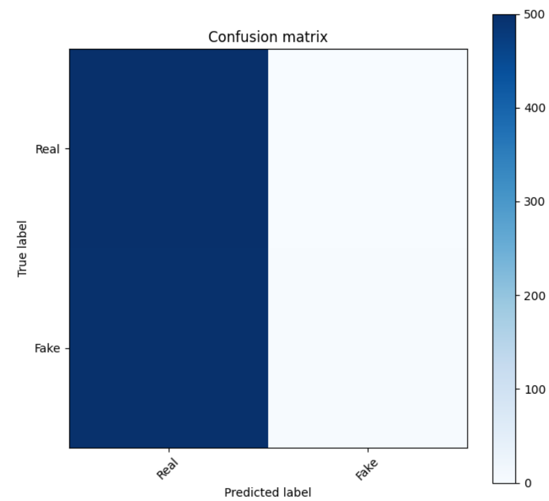

Figure 17: Deepfake Matrix (Project 1)

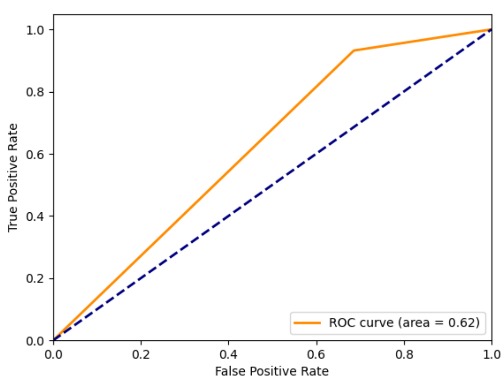

Figure 18: Deepfake Curve (Project 2)

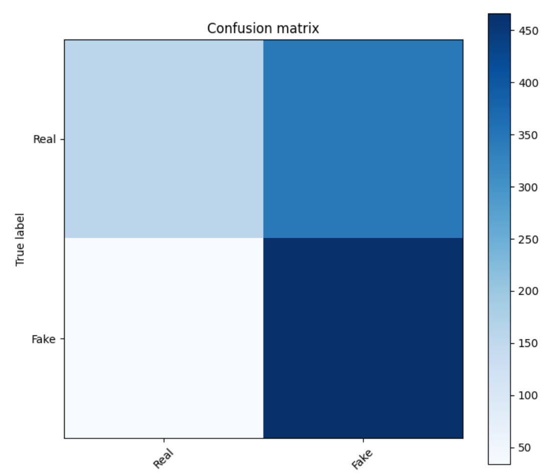

Figure 19: Deepfake Matrix (Project 2)

Figure 20: Deepfake Results

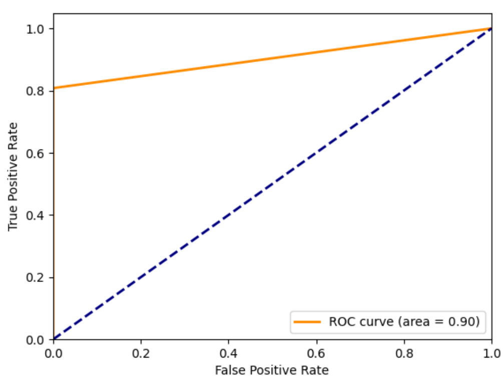

Figure 21: GauGAN Curve (Project 1)

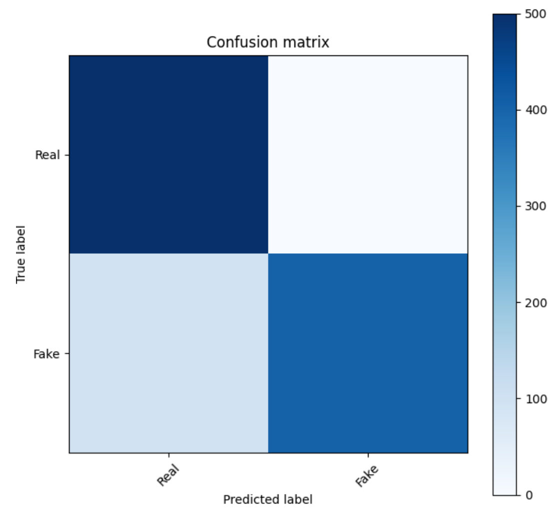

Figure 22: GauGAN Matrix (Project 1)

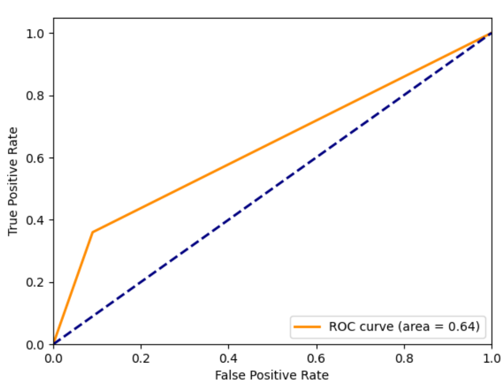

Figure 23: GauGAN Curve (Project 2)

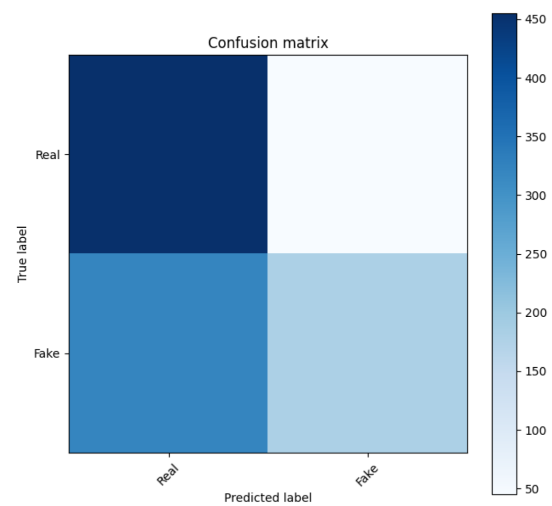

Figure 24: GauGAN Matrix (Project 2)

Figure 25: GauGAN Results

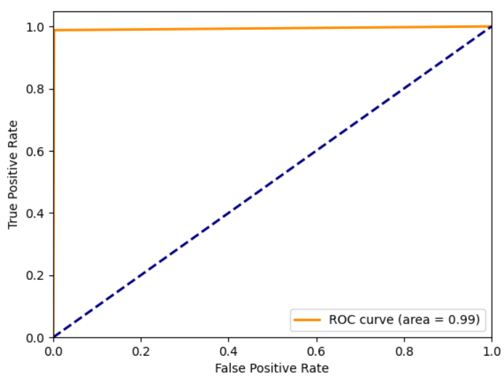

Figure 26: IMLE Curve (Project 1)

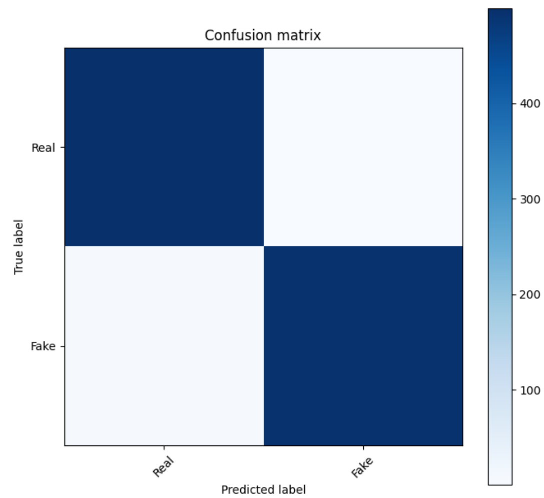

Figure 27: IMLE Matrix (Project 1)

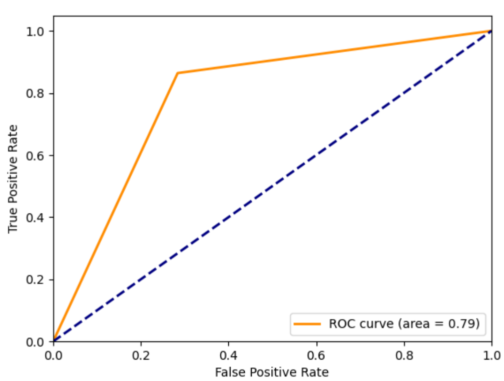

Figure 28: IMLE Curve (Project 2)

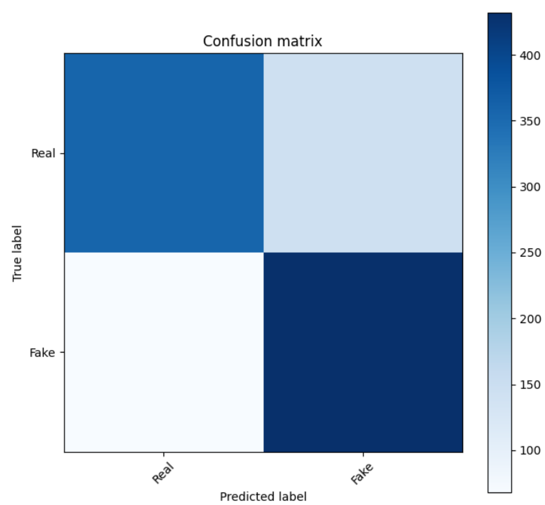

Figure 29: IMLE Matrix (Project 2)

Figure 30: IMLE Results

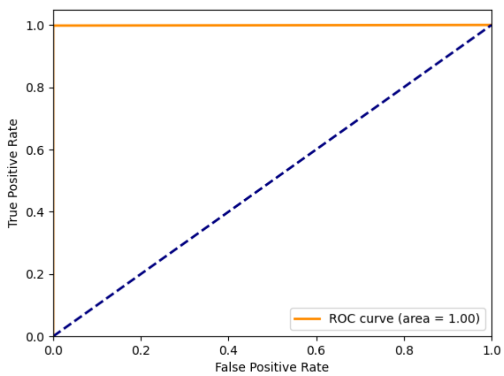

Figure 31: ProGAN Curve (Project 1)

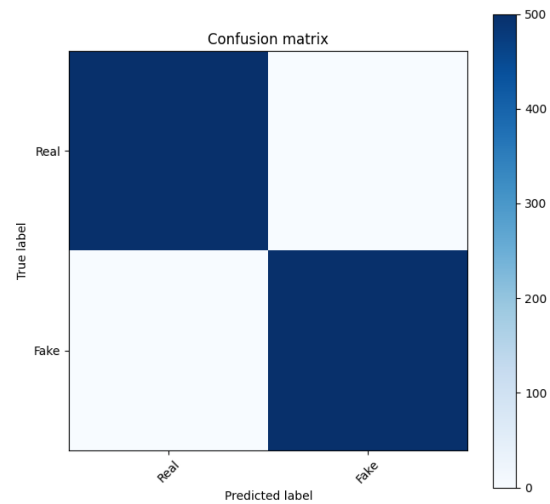

Figure 32: ProGAN Matrix (Project 1)

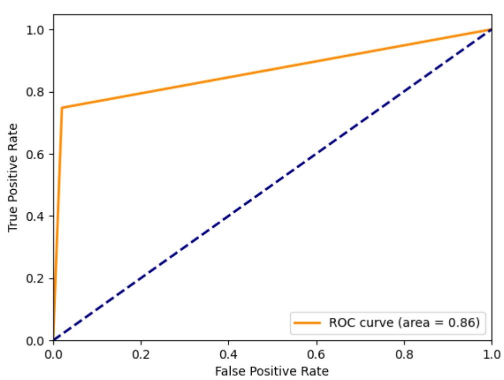

Figure 33: ProGAN Curve (Project 2)

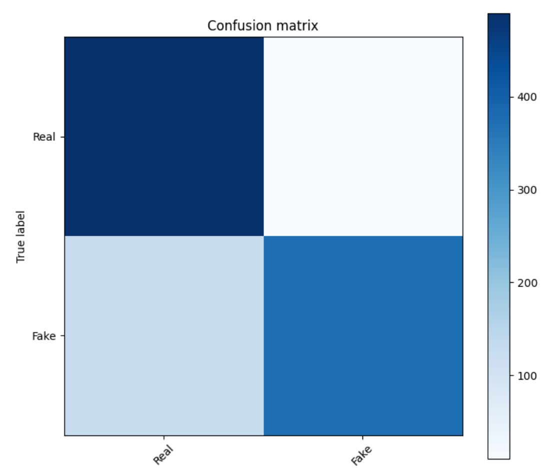

Figure 34: ProGAN Matrix (Project 2)

Figure 35: ProGAN Results

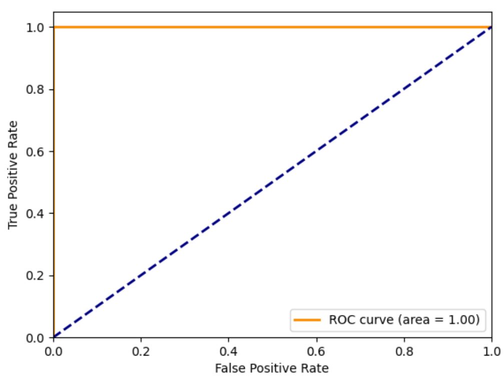

Figure 36: StarGAN Curve (Project 1)

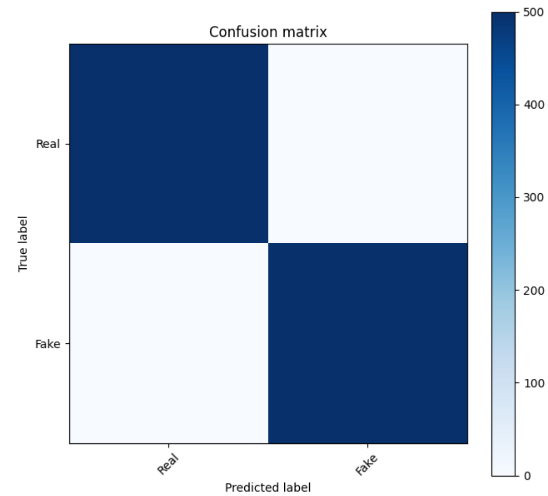

Figure 37: StarGAN Matrix (Project 1)

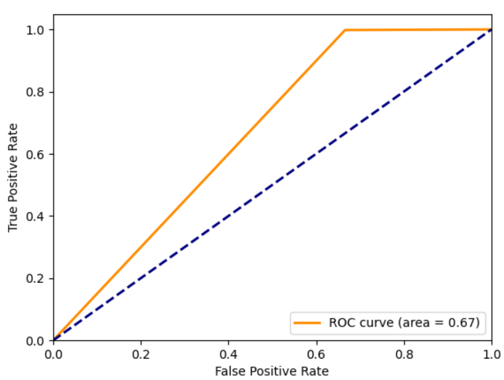

Figure 38: StarGAN Curve (Project 2)

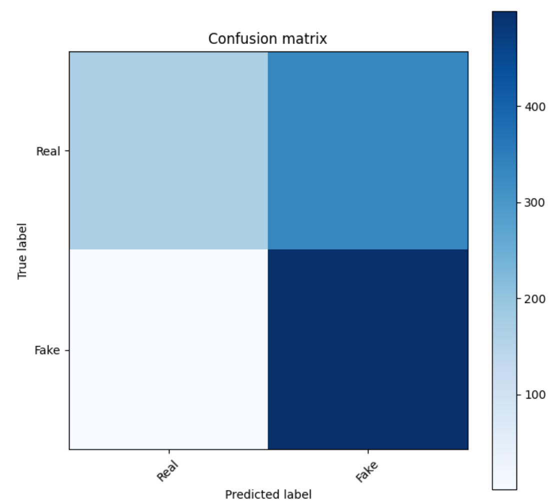

Figure 39: StarGAN Matrix (Project 2)

Figure 40: StarGAN Results

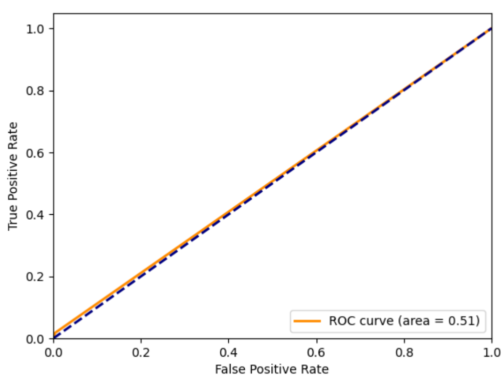

Figure 41: SAN Curve (Project 1)

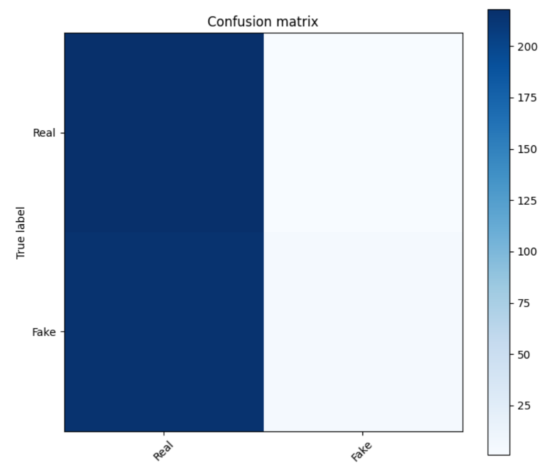

Figure 42: SAN Matrix (Project 1)

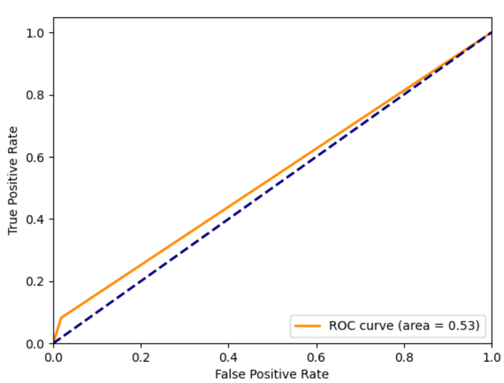

Figure 43: SAN Curve (Project 2)

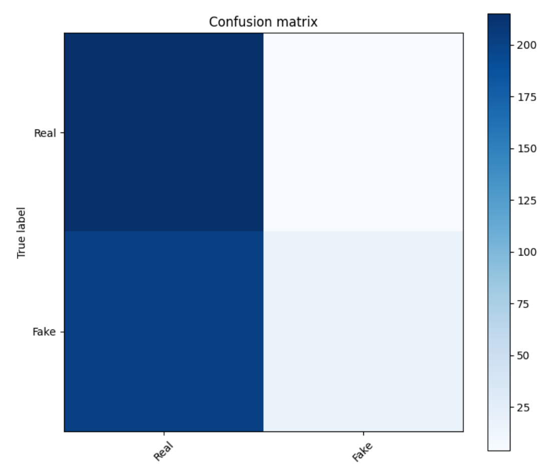

Figure 44: san Matrix (Project 2)

Figure 45: san Results

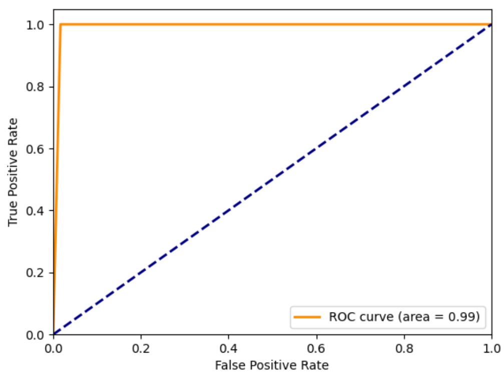

Figure 46: SeeInDark Curve (Project 1)

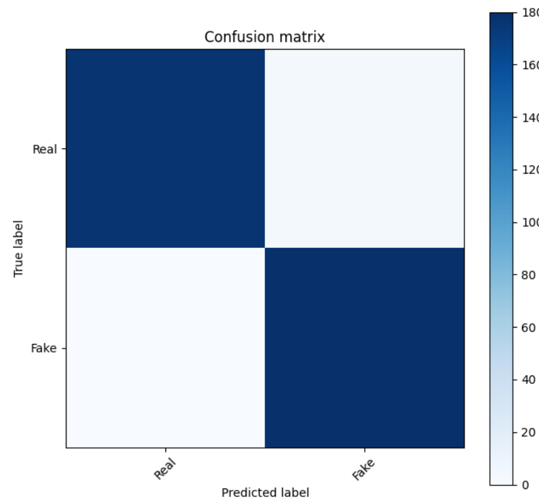

Figure 47: SeeInDark Matrix (Project 1)

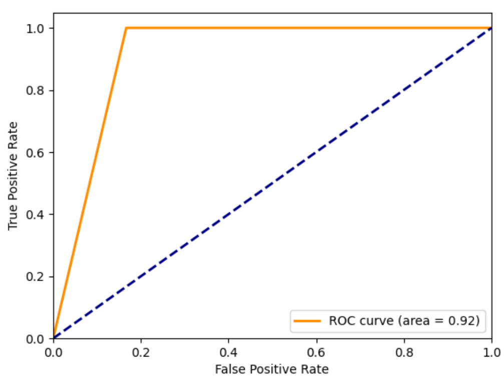

Figure 48: SeeInDark Curve (Project 2)

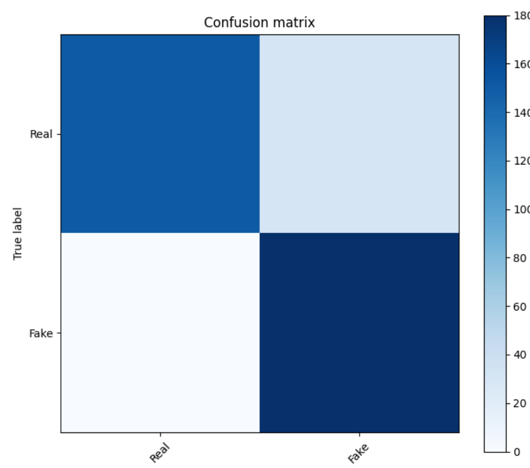

Figure 49: SeeInDark Matrix (Project 2)

Figure 50: SeeInDark Results

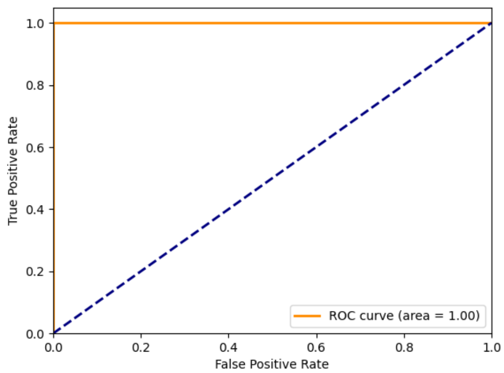

Figure 51: StyleGAN Curve (Project 1)

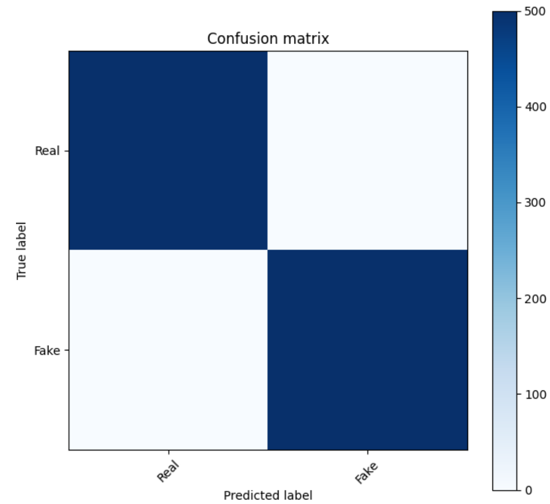

Figure 52: StyleGAN Matrix (Project 1)

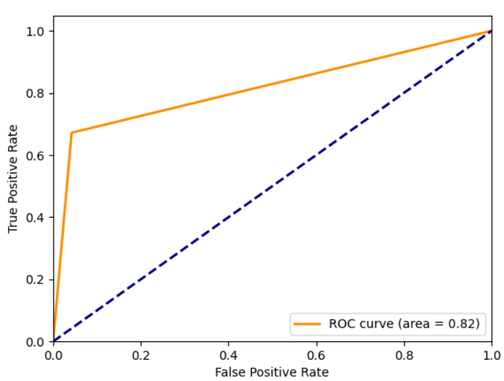

Figure 53: StyleGAN Curve (Project 2)

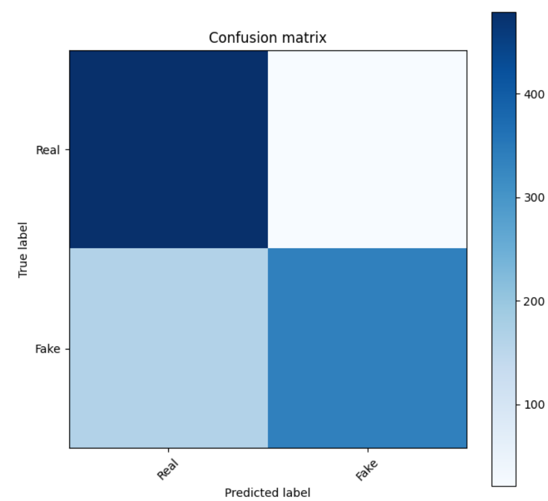

Figure 54: StyleGAN Matrix (Project 2)

Figure 55: StyleGAN Results

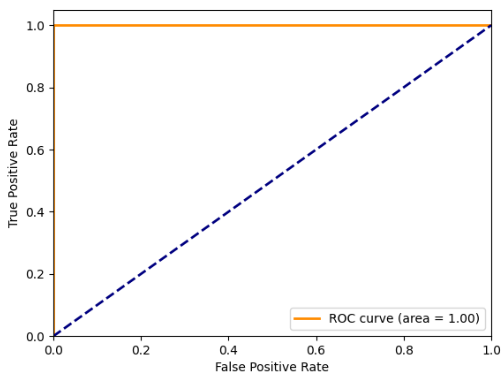

Figure 56: StyleGAN2 Curve (Project 1)

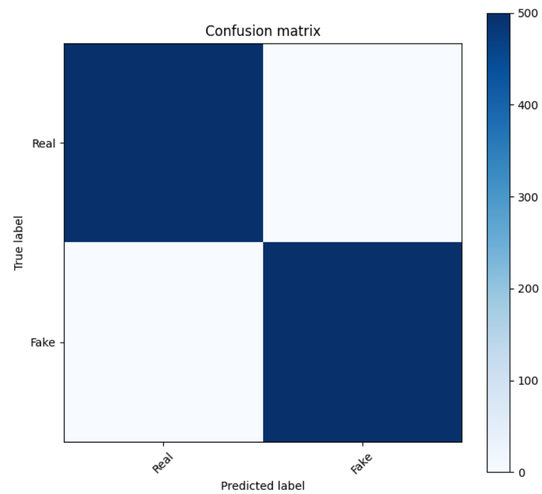

Figure 57: StyleGAN2 Matrix (Project 1)

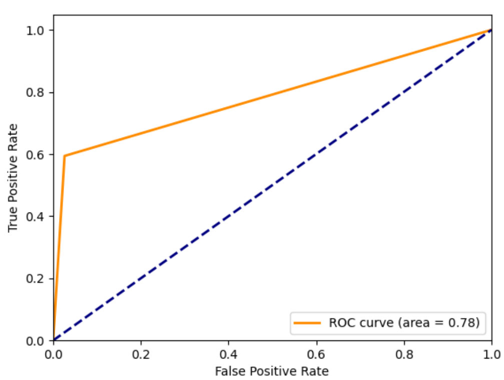

Figure 58: StyleGAN2 Curve (Project 2)

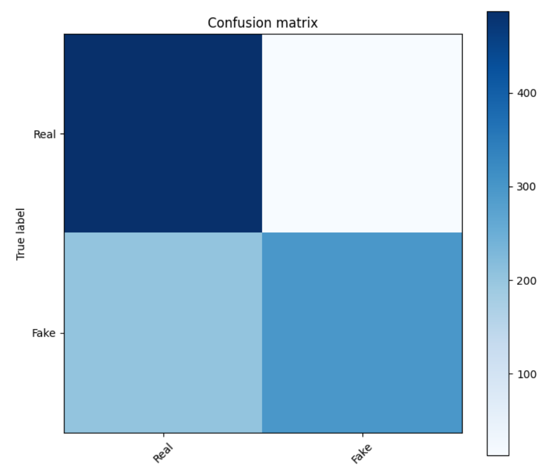

Figure 59: StyleGAN2 Matrix (Project 2)

Figure 60: StyleGAN2 Results

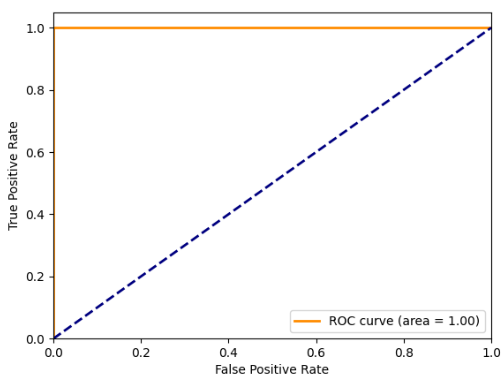

Figure 61: WhichFaceIsReal Curve (Project 1)

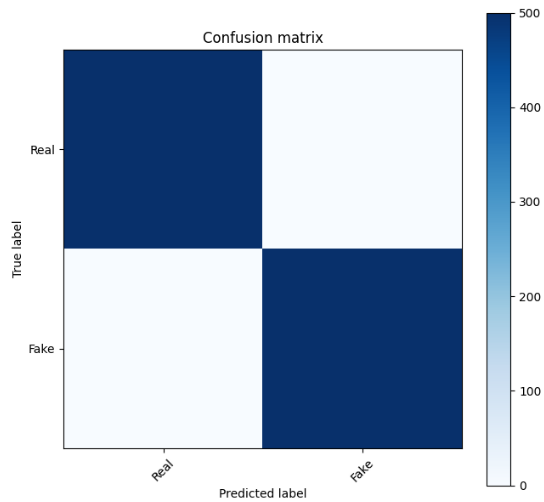

Figure 62: WhichFaceIsReal Matrix (Project 1)

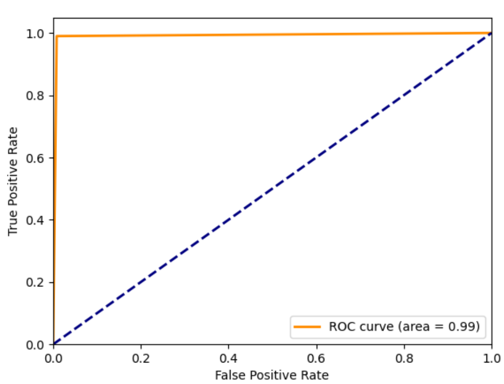

Figure 63: WhichFaceIsReal Curve (Project 2)

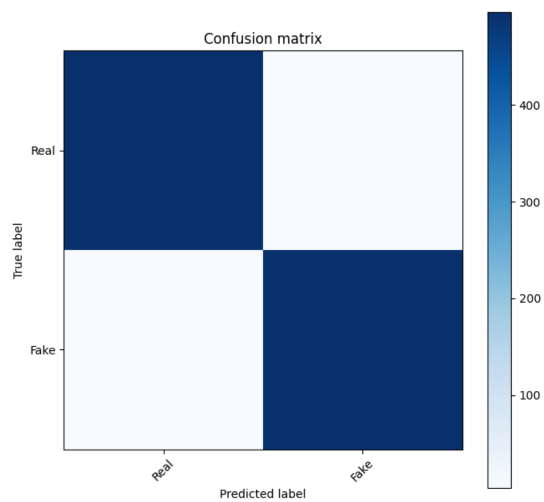

Figure 64: WhichFaceIsReal Matrix (Project 2)

Figure 65: WhichFaceIsReal Results

References

[1] Ashraf Abdel-Karim Abu-Ein et al. *Analysis of the current state of deepfake techniques-creation and detection methods*. https://www.researchgate.net/publication/364243272_Analysis_of_the_current_state_of_deepfake_techniques-creation_and_detection_methods. 2022.

[2] Sarfraj Ahmed and Mohd Akbar Shaun. *Fast and Effective Deepfake Detection Method Using Frame Comparison Analysis*. https://assets.researchsquare.com/files/rs-3033313/v1_covered_c7bdf178-bdea-492e-958e-c49b0f76bc47.pdf?c=1686660286. 2023.

[3] Nuha Aldausari et al. *Video Generative Adversarial Networks: A Review*. https://dl.acm.org/doi/fullHtml/10.1145/3487891. 2022.

[4] José P. Amorim et al. "Evaluating the faithfulness of saliency maps in explaining deep learning models using realistic perturbations". In: *Information Processing Management* 60.2 (2023), p. 103225. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2022.103225. URL: https://www.sciencedirect.com/science/article/pii/S0306457322003260.

[5] François Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions*. https://arxiv.org/pdf/1610.02357.pdf. 2017.

[6] Diego Gragnaniello et al. *Are GAN generated images easy to detect? A critical analysis of the state-of-the-art*. 2021. arXiv: 2104.02617 [cs.CV].

[7] Neeraj Guhagarkar et al. *DEEPFAKE DETECTION TECHNIQUES: A REVIEW*. https://www.viva-technology.org/New/IJRI/2021/2.pdf. 2021.

[8] Staffy Kingra, Naveen Aggarwal, and Nirmal Kaur. *Exploiting source camera noise discrepancies using Siamese Network for Deepfake Detection*. https://www.sciencedirect.com/science/article/abs/pii/S002002552300926X. 2023.

[9] Shweta Lamba, Anupam Baliyan, and Vinay Kukreja. *Generative Adversarial Networks based Data Augmentation for Paddy Disease Detection using Support Vector Machine*. https://ieeexplore.ieee.org/document/9964506. 2022.

[10] Yuhang Li et al. *Xception: Deep Learning with Depthwise Separable Convolutions*. https://arxiv.org/pdf/1912.02057.pdf. 2019.

[11] Bo Liu et al. *Detecting Generated Images by Real Images*. https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136740089.pdf. 2022.

[12] Sara Mandelli et al. "Detecting Gan-Generated Images by Orthogonal Training of Multiple CNNs". In: *2022 IEEE International Conference on Image Processing (ICIP)*. 2022, pp. 3091–3095. DOI: 10.1109/ICIP46576.2022.9897310.

[13] Sara Mandelli et al. "Training CNNs in Presence of JPEG Compression: Multimedia Forensics vs Computer Vision". In: *IEEE International Workshop on Information Forensics and Security (WIFS)*. 2020. DOI: 10.1109/WIFS49906.2020.9360903.

[14] Momina Masood et al. *Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward*. https://link.springer.com/article/10.1007/s10489-022-03766-z. 2022.

[15] Paolo Masulli et al. *Data-driven analysis of gaze patterns in face perception: Methodological and clinical contributions*. https://www.sciencedirect.com/science/article/pii/S0010945221003695. 2022.

[16] Richi Nayak Md Abul Bashar. *Time Series Anomaly Detection with Adjusted-LSTM GAN*. 2009. URL: https://arxiv.org/pdf/2308.06663.

[17] João C. Neves et al. *GAN Fingerprints in Face Image Synthesis*. https://link.springer.com/chapter/10.1007/978-981-16-7621-5_8. 2022.

[18] Jeremy Nixon, Mike Dusenberry, and Ghassen Jerfef. *Measuring Calibration in Deep Learning*. 2020.

[19] Christopher Olah. *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. 2015.

[20] Andreas Rössler et al. *FaceForensics++: Learning to Detect Manipulated Facial Images*. https://arxiv.org/pdf/1901.08971.pdf. 2019.

[21] YuYang Sun et al. *Face Forgery Detection Based on Facial Region Displacement Trajectory Series*. https://arxiv.org/pdf/2212.03678.pdf. 2022.

[22] Abdul Jamsheed V. and Janet B. *Deep Fake Video Detection Using Recurrent Neural Networks*. https://www.isroset.org/pub_paper/IJSRCSE/4-ISROSET-IJSRCSE-05540.pdf. 2021.

[23] Sofia Visa et al. "Confusion Matrix-based Feature Selection." In: vol. 710. Jan. 2011, pp. 120–127.

[24] Grega Vrbančič and Vili Podgorelec. *Transfer Learning With Adaptive Fine-Tuning*. 2020. DOI: 10.1109/ACCESS.2020.3034343.

[25] Xin Wang et al. *GAN-generated Faces Detection: A Survey and New Perspectives*. https://arxiv.org/pdf/2202.07145.pdf. 2023.

[26] Vera Wesselkamp et al. *Misleading Deep-Fake Detection with GAN Fingerprints*. https://arxiv.org/pdf/2205.12543.pdf. 2022.

[27] Hanshu Yan et al. *Unsupervised Image Noise Modeling with Self-Consistent GAN*. 2020. URL: https://arxiv.org/abs/1906.05762.

[28] Kun Liu; Aimei Li; Xi Wen; Haiyong Chen; Peng Yang. *Steel Surface Defect Detection Using GAN and One-Class Classifier*. 2022. URL: https://ieeexplore.ieee.org/abstract/document/8895110.

[29] Ning Zhou, Renyu Liang, and Wenqian Shi. *A Lightweight Convolutional Neural Network for Real-Time Facial Expression Detection*. https://ieeexplore.ieee.org/document/9305261. 2021.