Project Report

Deepfake Image Detection Challenge

Chapter Lead
Imane El Maakoul

Task Leads
Parnika Damle
Rishabh Sabharwal
Akash Kundu
Vishu Kalier
Harsha Tejaswi
Vinod Cherian

Table of Contents

Introduction

Over the past few years, deepfakes have emerged as one of the most significant threats to the integrity of digital media. Deepfake technology uses advanced machine learning algorithms to create convincing fake images, videos, or audio recordings. These deepfakes can be used for various purposes, including spreading misinformation, creating propaganda, or even impersonating individuals. With the proliferation of social media and the widespread availability of sophisticated tools and software, it has become increasingly easier to create deepfakes that are convincing and difficult to detect.

The potential harm that deepfakes can cause is enormous. They can be used to manipulate public opinion, discredit individuals or organizations, or even cause real harm by disseminating false information. Deepfakes can also be used for financial gain, such as in the case of impersonating bank officials or celebrities to extract sensitive information or money. As such, there is a growing need for reliable and effective deepfake detection methods that can help identify and mitigate the potential harm caused by deepfakes.

The development of an optimized model learning to detect deepfakes can help identify and prevent the spread of malicious content, safeguard the integrity of digital media, and protect individuals and organizations from potential harm. With the increasing sophistication of deepfake technology, it is critical that we invest in research and development of detection methods that can keep pace with the latest advancements in deepfake technology. A project focused on deepfake detection can be an essential step towards achieving this goal and can significantly contribute to ensuring the safety and security of our digital world.

Project overview

The project aims to develop a machine learning model that can detect deepfake facial images with high accuracy. Deepfakes are manipulated videos or images generated using artificial intelligence, which can be used to spread false information, defame individuals, or create hoaxes. The proposed solution uses a deep neural network to analyze various facial features to identify any inconsistencies that may indicate the presence of a forgery.

The project involves collecting and preparing a large dataset of real and fake facial images generated from different GANs to train the deep learning model, then defining

and applying augmentation techniques on the images. The model will be trained using state-of-the-art techniques such as convolutional neural networks and transfer learning to improve its accuracy and robustness.

Once the model is trained, it is planned to be deployed to the cloud as a web service and provide an open-source API to analyze input images and flag any deepfakes.

The project's main objective is to develop an ML model able to detect deepfake facial images with a 95% accuracy rate.

The challenges facing this project are:

- Model generalization on unseen data.
- High computational requirements to train the model.
- Ethical consideration and potential misuse of the model.

Project scope

The goal of this project is to develop a machine learning model capable of detecting deepfake images with high accuracy. Deepfake images are manipulated images generated using artificial intelligence, which can be used to spread false information, defame individuals, or create hoaxes. The proposed solution will use a deep neural network to analyze various facial features such as eye movements, skin texture, and facial expressions to identify any inconsistencies that may indicate the presence of a deepfake.

Dataset Preparation

The project will begin with collecting and preparing a large dataset of real and fake images. The dataset will include a variety of real images and deepfake images generated using various techniques such as GANs (Generative Adversarial Networks). The dataset will be labeled to indicate which images are real and which are deepfake images.

Model Development

Once the dataset is prepared, the next step is to develop a deep learning model capable of detecting deepfake images. The model will be developed using state-of-the-art techniques such as convolutional neural networks (CNNs) and transfer learning to improve its accuracy and robustness.

The model will be trained on the dataset using a supervised learning approach, where the model will learn to differentiate between real and fake images by analyzing various facial features. The trained model will be evaluated using a separate test dataset to measure its accuracy and performance.

Model Deployment

Once the model is trained and evaluated, it will be deployed into a cloud hosted server and an API will be provided to analyze images in real-time and flag any deepfakes.

Key stakeholders

Omdena Local Chapter Munich
Omdena is a global platform for collaborative AI innovation. It connects data science enthusiasts with organizations to solve real-world problems using AI and machine learning. Omdena Munich is a local chapter with a mission of collaborating with local and global AI enthusiasts to solve local and global challenges.

General Public
The wider community that may be affected by the AI solutions developed by Omdena's projects, such as through their impact on social, environmental, and economic issues.

AI Experts
Diverse AI experts from around the world who work collaboratively on projects, bringing their unique skills and experiences to the table.

Project team

Name	Task
Qutaiba Ahmed Ansari	Model training
Chirag Gautam	Model training
Vishu Kalier	Model training task lead
Mussie Berhane	Model training Data collection
Rishabh Sabharwal	Data preprocessing task lead Model training
Saurav Suman	Data preprocessing
Akash Kundu	Data collection Data preprocessing Model training Model testing task lead

	Model deployment Workshop on model training
Abdelrahman Youssry	Data collection Data preprocessing Model training Model testing Model deployment
Vinod Cherian	Data collection Model deployment task lead
Reem Abdel-Salam	Model training Model testing Workshop on transfer learning
Parnika Damle	Data collection task lead Model training Model testing
Sachin Srivastav	Data collection
Abdul Rahman	Data pre-processing Model training
Imane El Maakoul	Data pre-processing Model training
Harsha Tejaswi	Model deployment task lead Workshop on model deployment

Project timeline

The project was kicked off on the 27.02.2023 and closed on the 17.04.2023. The project duration was 7 weeks. In the figure below, the project timeline is presented with the duration of the 5 major tasks.

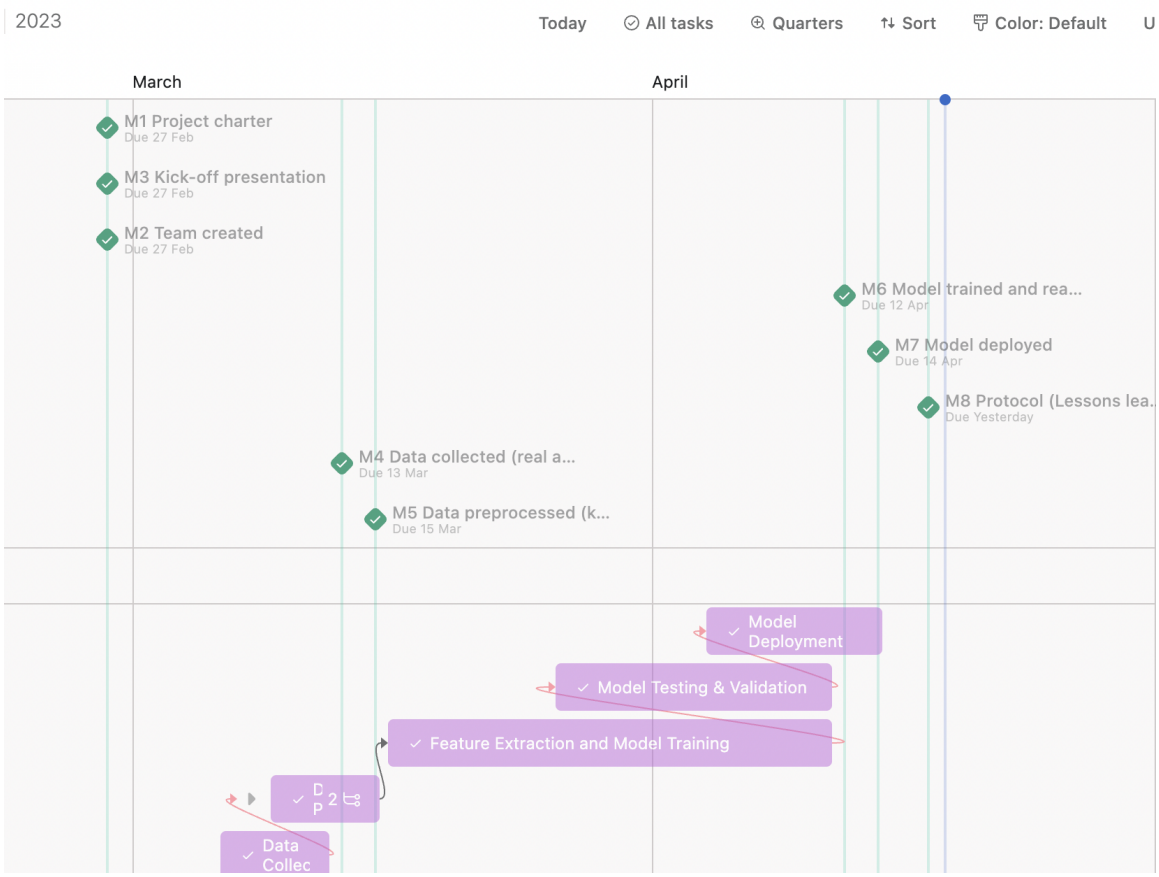

Figure 1: Project timeline

During the project execution, the project planning was reviewed in a weekly sync meeting, during which the task status and potential risks were communicated, and the planning updated accordingly.

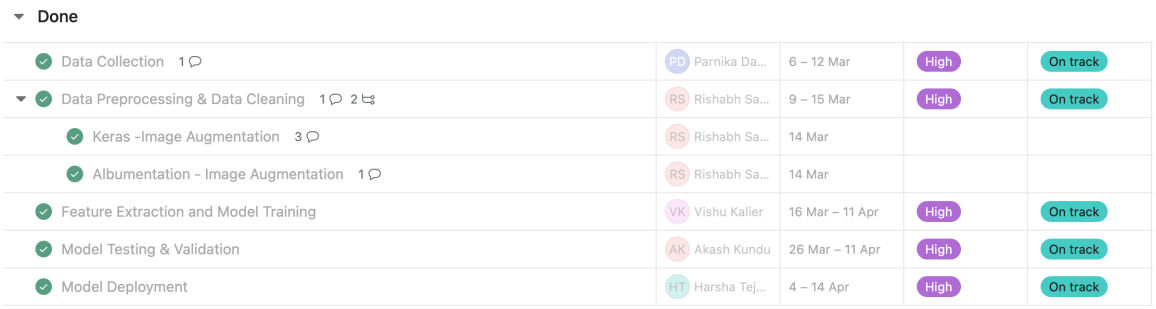

Figure 2: Project task list dashboard along with task leader, priority, status and due date

All the milestones set ahead were reached, although we have faced technical issues with regards to model deployment which delayed a bit the task completion but we could finish on time.

Task name	Assignee	Due date	Priority	Status
Milestones				
M1 Project charter	Imane	27 Feb	High	On track
M2 Team created	Imane	27 Feb	High	On track
M3 Kick-off presentation	Imane	27 Feb	High	On track
M4 Data collected (real and deepfake from different GANs) and uploa	Parnika Da...	13 Mar	High	On track
M5 Data preprocessed (keras and albumentations) and uploaded	Rishabh Sa...	15 Mar	High	On track
M6 Model trained and reached the OKR	Akash Kundu	12 Apr	High	On track
M7 Model deployed	Harsha Tej...	14 Apr	High	On track
M8 Protocol (Lessons learned, documentation, closing)	Imane	Tomorrow	High	On track

Figure 3: Project milestones

Dataset collection

The goal of this project is to develop a machine learning model capable of detecting deepfake vs real images with high accuracy. The team should be able to accurately detect whether an input image is real or deepfake.

Most of the publicly available datasets for deepfake detection are videos (image sequences). However, the project initially aims at detecting deepFakes in face images, and later, it can be extended for videos. Several works on deepfake detection are present in the literature, but unfortunately most of them lack robustness and generalization; they do not work in real-case scenarios. Therefore, our team decided to use the dataset developed by 'Deepfake Detection and Reconstruction Challenge' held by 21st International Conference on Image Analysis and Processing (ICIAP). This dataset is more challenging than usual detection approaches, having deepfake images generated by different state-of-the-art architectures.

Dataset description

Two datasets of real face images were used for the employed experimental phase: CelebA and FFHQ. Different deepfake images were generated considering StarGAN, GDWCT, AttGAN, StyleGAN and StyleGAN2 architectures. In particular, CelebA images were manipulated using pre-trained models on StarGAN, GDWCT and AttGAN. Images of StyleGAN and StyleGAN2 were created starting from FFHQ dataset.

The training set was composed of images organized into several ZIP files. The test set was composed of several real and deepfake images similar to those of the training set,

and in addition, images obtained by applying some processing (rotation, mirroring, gaussian-filtering, scaling, cropping and re-compression) were introduced.

A detailed description of the obtained images is given below.

Dataset	# Images	Resolution	Description
CelebA	5000	178 × 218	a large-scale face attributes dataset with > 200K celebrity images
FFHQ	5000	1024 × 1024	human faces with variations in terms of age, ethnicity and image background
StarGAN	1000	256 × 256	CelebA images were manipulated
GDWCT	1000	216 × 216	CelebA images were manipulated
AttGAN	1000	256 × 256	CelebA images were manipulated
StyleGAN	1000	1024 × 1024	Images generated using FFHQ as the input dataset
StyleGAN2	1000	1024 × 1024	Images generated using FFHQ as the input dataset

Table 1: Dataset description

Data preprocessing

The dataset had images of different sizes ranging from (216, 216) to (1024, 1024) so we decided to resize all the images to (160,160) to maintain a uniformity in the images and also, many online platforms downsize the images to reduce memory consumption, hence our model will be compatible with images that are fed straight from these online platforms.

Moreover, it was observed that all the images were upright thus to prevent our model from overfitting on only upright images, we applied random rotation, vertical and horizontal flip. Since most of the deepfakes are generated by imposing or altering two or more real images and combining them, they often leave some traces of face blending artifacts and mostly these traces are seen near eyes, nose or mouth. Hence, to avoid our model from overfitting only onto these traces, we applied blackout of facial features that randomly blacked out either eyes, nose or mouth.

Further, we added gaussian noise, blurring and used JPEG compression to reduce the quality of images so that our model can generalize well on real world images that are often blurry and compressed. Hence, data preprocessing was a vital step in this project that not only helped our model to learn from a range of different images but also reduced the computation cost by a huge extent and helped us save a lot of time and resources.

The pre-processing steps taken were:

1. Resizing images to (160, 160)
2. Random blackout of facial features such as eyes, nose or mouth
3. Random rotation, horizontal or vertical flip
4. JPEG compression
5. Gaussian blurring, noise addition
6. Changing the brightness, saturation, hue and contrast
7. Shear shift, zooming, height and width shift

Proper care was taken to not augment images to an extent such that it degrades the quality of our dataset.

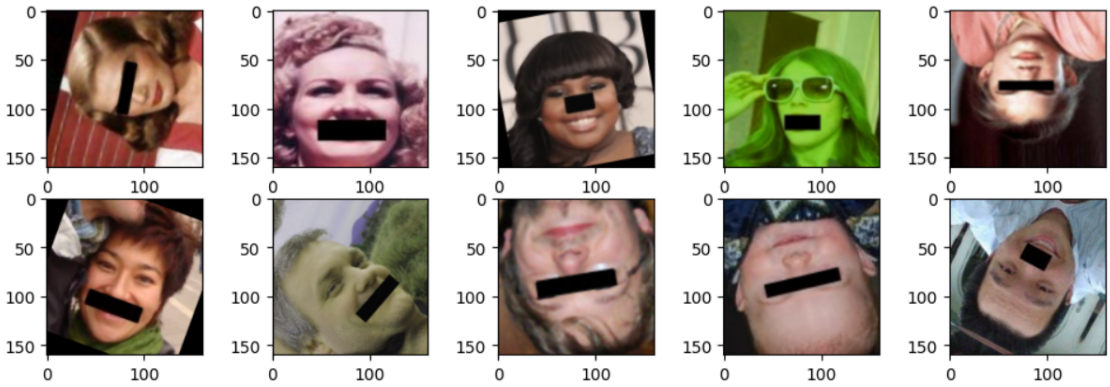

Figure 4: Sample of augmented real images

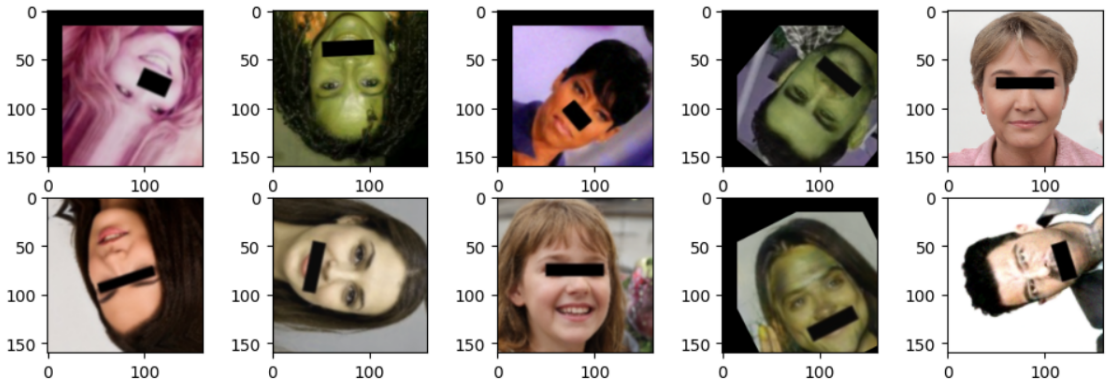

Figure 5: Sample of augmented deepfake images

The original dataset had 10K real images and 5K deepfake images. To maintain the balance between the classes, we decided to augment 30K real images and 30K deepfake images. Dataset was split according to the following ratio - 80% for train, 10% for validation and 10% for the test set. Due care was given to balance each of these datasets, hence out of 24K train images, we had 12K real and 12K deepfake images. Similar approach was used for both validation and test set.

We decided to work in two teams, one team would augment the images with albumentations library and the other team would work with keras library. Within each team, the work was further divided such that each person worked on utmost 2 folders and everyone used their own augmentation pipeline thereby increasing the variation in the augmented images. Each team generated a total of 60K images, thus at the end of pre-processing, we had a total of 120K images, divided as two separate sets with an aim to train our model on both the sets to improve its generalizability.

Feature extraction

The feature extraction was performed on the pre-processed Images to extract the relevant features for training of the model. We applied many processes and extraction kernels and filters like blur, gray-scale, contrast..etc.

We first read the documentations and research papers at hand to fully understand the kernels and the process of feature extraction. We then held weekly meetings to collaboratively and collectively move the task towards its completion. We distributed the work for everyone in the team so that uniformity is formed. Everyone in the team performed their own methodologies to extract the kernels and filters of the Images provided in the training dataset.
Then, the features were passed to the Model Training task to train the neural network.

Model training

During this task, we used many models and tested them on the preprocessed datasets. We used many advanced techniques of Model Training like Transfer Learning, Deep Neural Networks, Object Detection models, Image Segmentation models, etc.

The task was divided among the team members to train the model using a small amount of the training data. The models were thoroughly researched and then provided to the team and each member took one model as per their desire. We then held weekly meetings to discuss the model training advancements and if the model

was giving good accuracy for the small fraction of the dataset then, the model was asked to be trained on the entire dataset.

The code, implementation and accuracy of every contributor is available on github of the project. The best model was chosen for testing and cross-validation among the many created models by the team. Around 17 models were chosen by the team, out of which 11 models were fully created and Implemented on the entire dataset. After the implementation of the chosen models, the best performing models were chosen to be passed on to the Model Testing Team.

Fully Created Models

Name of Model	Trained by	Accuracy	Model	Code
AlexNet	Mussie Berhane	85%	Model	Notebook
EfficientNetB5	Abdelrahman Youssry	90%	Model	Notebook
MogaNetXTiny	Akash Kundu	85.8%	Model	Notebook
DenseCNN	Rishabh Sabharwal	91.78%	Model	Notebook
MobileNetV3	Reem Abdel-Salam	89%	Model	Notebook
VOLO-d2	Parnika Damle	96.52%	Model	Notebook
ConvNextTiny	Reem Abdel-Salam	97%	Model	Notebook
EfficientNetB7	Abdelrahman Youssry	92.5%	Model	Notebook
ResNet50	Qutaiba Ansari	93%	Model	Notebook
VGG16	Abdelrahman Youssry	93.5%	Model	Notebook
VGG19	Abdelrahman Rabah	80%	Model	Notebook
DenseNet-264	Abdul Rahman	50%	Model	Notebook

Table 2: Classification report of the trained models along with the contributor's name and code

Model validation and testing

After completion of the model training, we conducted validation and testing of the best-performing models to confirm the results and determine the best model for deployment. To evaluate the performance of our deepfake image detection model.

Model Evaluation Matrix

Model Type	Accuracy	Precision	Recall	F1 Score	Tested by	Code
Dense CNN	63%					
Deepfake		0.74	0.73	0.74	**Rishabh Sabharwal**	Notebook
Real		0.35	0.36	0.35		
ResNet50	65%				**Qutaiba Ansari**	Notebook
Deepfake		0.71	0.86	0.78		
Real		0.27	0.14	0.18		
EfficientNetB7	73%				**Abdelrahman Youssry**	Notebook
Deepfake		0.83	0.77	0.8		
Real		0.52	0.6	0.56		
EfficientNetB5	76%				**Abdelrahman Youssry**	Notebook
Deepfake		0.81	0.86	0.84		
Real		0.59	0.5	0.54		
VGG16	77%				**Abdelrahman Youssry**	Notebook

Deepfake		0.81	0.9	0.85		
Real		0.65	0.46	0.54		
ConvNextTiny	81%				**Reem Abdel-Salam**	Notebook
Deepfake		0.80	0.99	0.88		
Real		0.92	0.39	0.54		
VOLO-D2	81%				**Parnika Damle**	Notebook
Deepfake		0.81	0.95	0.88		
Real		0.79	0.45	0.57		
MobileNetV3	81%				**Reem Abdel-Salam**	Notebook
Deepfake		0.92	0.81	0.86		
Real		0.63	0.82	0.71		

Table 3: Classification report of the tested models along with the contributor's name and code

Final Model Evaluation Matrix Explanation

- Based on the model results in the table above, **MobileNetV3** was chosen to be deployed.
- Although, **Volo-D2**, **ConvNextTiny**, and **MobileNetV3** had almost similar accuracy (81%), the recall for both the deepfakes (0.81) and real images (0.82) is better compared to the other models which had great recall for the deepfake images but performed poorly in case of real images.
- Although, **VOLO-D2** showed higher precision for real images, **MobileNetV3** had significantly higher recall. As a result, **MobileNetV3** was ultimately chosen over **VOLO-D2**.
- The F1 Score for the real images (0.71) is better than all of the models present in the table.

Confusion Matrix of MobileNetV3

In the given test dataset, there were 5000 deepfake images and 2000 real images. The model predicted **4048** deepfakes and **1632** real images correctly. i.e **5680** images out of **7000** images were correctly predicted by our final model.

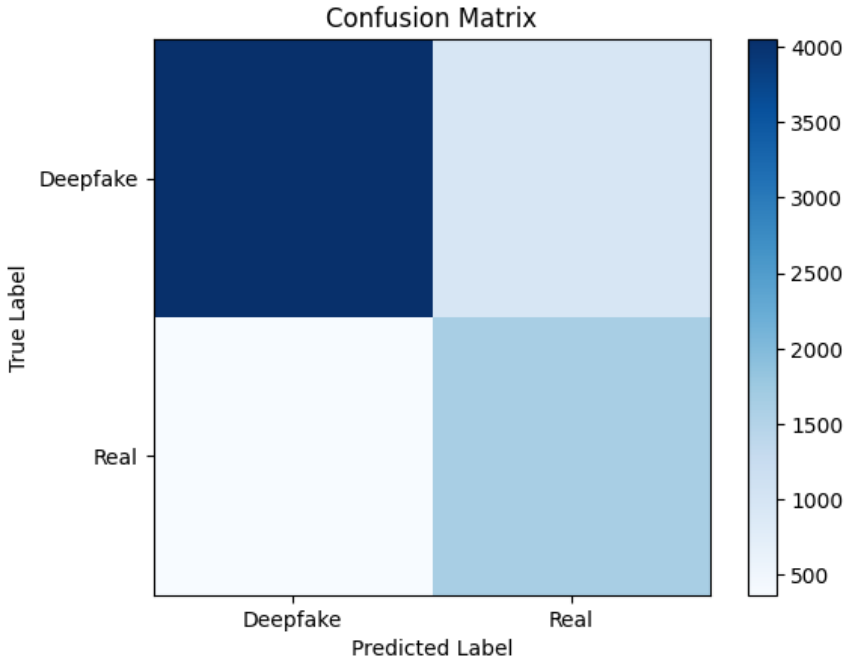

Figure 6: Confusion matrix of the chosen **MobileNetV3** model

Comparative results

The classification matrix of our model was finally compared with the top 5 performing teams from the competition.

a) VisionLabs	**Precision**	**Recall**	**F1 Score**
Real	0.89	0.88	0.89
Deepfake	0.95	0.96	0.96
Accuracy			0.94

b) DC-GAN (Amped team)			
Real	0.86	0.78	0.82
Deepfake	0.92	0.95	0.93
Accuracy			0.90
c) Team Nirma			
Real	0.55	0.80	0.65
Deepfake	0.90	0.74	0.81
Accuracy			0.75
d) AIMH Lab			
Real	0.52	0.49	0.51
Deepfake	0.80	0.82	0.81
Accuracy			0.73
e) PRA Lab-Div. Biometria			
Real	0.43	0.76	0.55
Deepfake	0.86	0.59	0.70

Accuracy			0.64
f) Proposed model			
Real	0.63	0.82	0.71
Deepfake	0.92	0.81	0.86
Accuracy			0.81

Table 4: Classification report of the best performing models compared with the proposed model

Based on the results, it is apparent that our model achieved significantly better performance compared to "Team Nirma" which ranked third in the competition. With the exception of the top two teams, Visionlabs and Amped Team, our proposed model outperforms all the other teams present in the competition.

Scopes of Improvement

The wrong predictions of the model were carefully observed. From the training dataset, we found a few deepfake images that were classified as real.

Figure 7: Deepfake images misclassified as real

All of the above images were generated by StyleGAN2, a very advanced GAN and these images do look very real.

Our model did predict images from these advanced GANs correctly as well.

Figure 8: Deepfake images correctly classified as deepfakes

It was observed that female deepfake faces were classified more accurately compared to male deepfakes. It might be the case that there are some features in female deepfakes that the model is catching well that is absent in case of male faces.

There is another possibility that there were more female faces in the training set compared to males which helped in extracting those features but we cannot be sure if the data was balanced based on gender since we did not perform gender classification.

There have also been instances when the real images were classified as fake images.

Figure 9: Real images misclassified as deepfakes

It is visible that the real images which are classified as fakes are of very low quality and blurry in nature. Resizing these images to a smaller size reduced the quality further that resulted in the images being misclassified.

Model deployment

For the model deployment task, we decided to use Streamlit as the front-end as it's easier to develop and flexible to deploy either on a Streamlit cloud platform or any other cloud provider like AWS, GCP, etc.

Nowadays, Data Scientists mainly use Streamlit to make interactive GUI for their Data Science Applications. Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps

To build a highly interactive application using Streamlit you don't need any frontend technology stack experience, instead only Python knowledge and some basic HTML, CSS knowledge is needed.
Works with TensorFlow, Keras, PyTorch, Pandas, Numpy, Matplotlib, Seaborn, Altair, Plotly, Bokeh, Vega-Lite, and more. Streamlit watches for changes on each save and updates the app live while you're coding. Code runs from top to bottom, always from a clean state, and with no need for callbacks.

In this project, the first page named "Project" has all the project-related information like project introduction, project goals, real-world problems that are trying to resolve, Chapter lead and Active members for each task activity, etc.

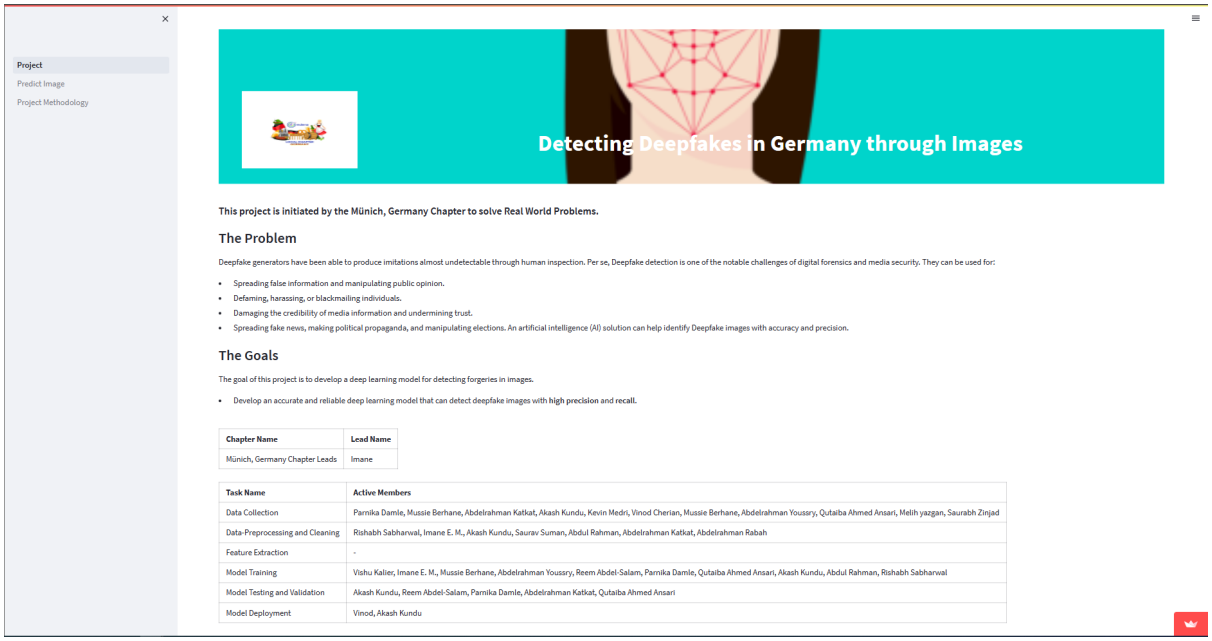

Figure 10: Project Overview on Streamlit

The second page is named "Predict Image", where we test the model chosen for its highest accuracy compared to other evaluated models and that is MobileNetV3.

In this page, we upload an image as input to the model to predict whether the input image is deepfake or real. The input image are not stored anywhere but is present in-memory and gets destroyed once the user inputs another image as input or leave the app page.

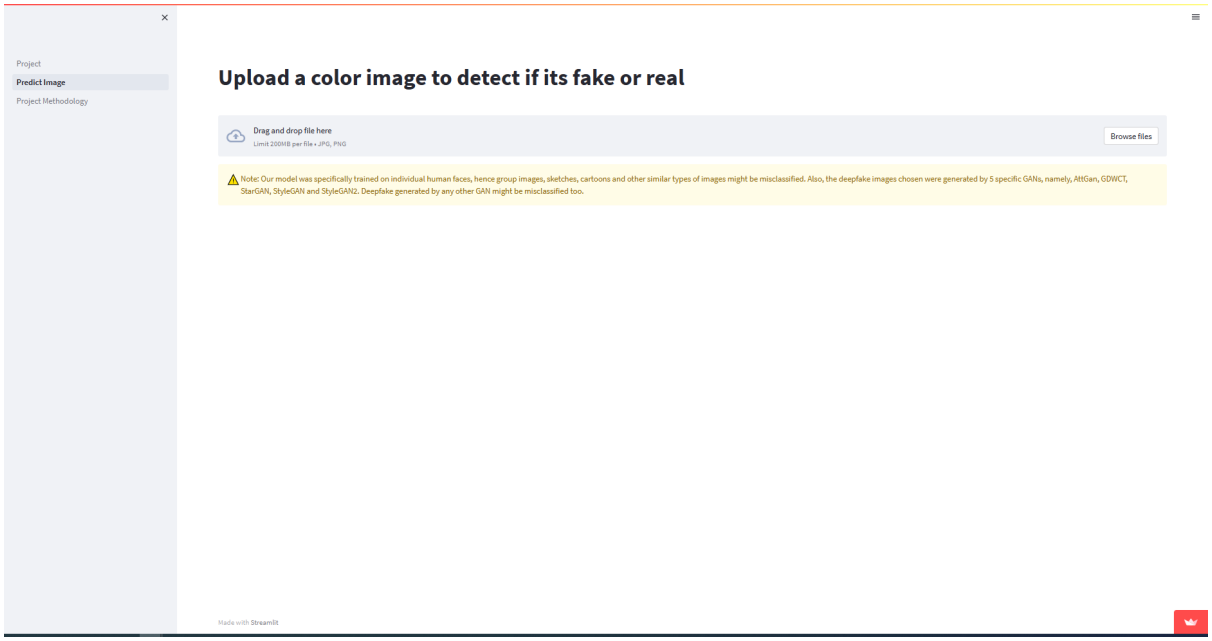

Figure 11: Image Classification Frontend

The model correctly detected the input image as Fake with a confidence level in percentage.

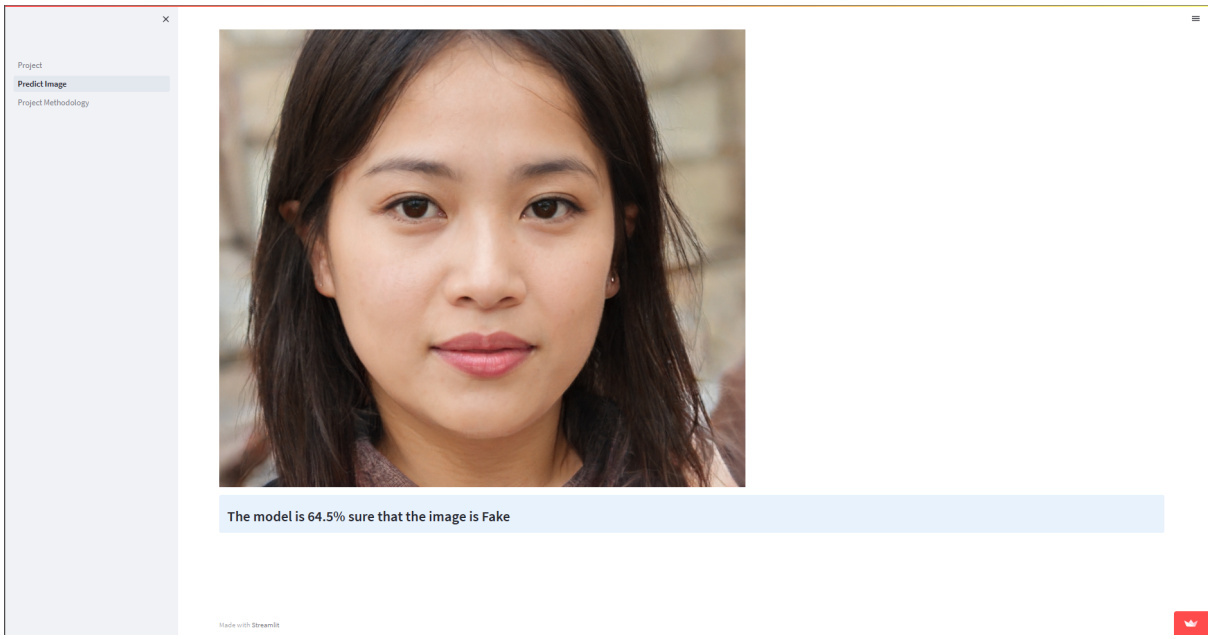

Figure 12: Model response for a deepfake image input

The model correctly detected the input image as Real with a confidence level in percentage.

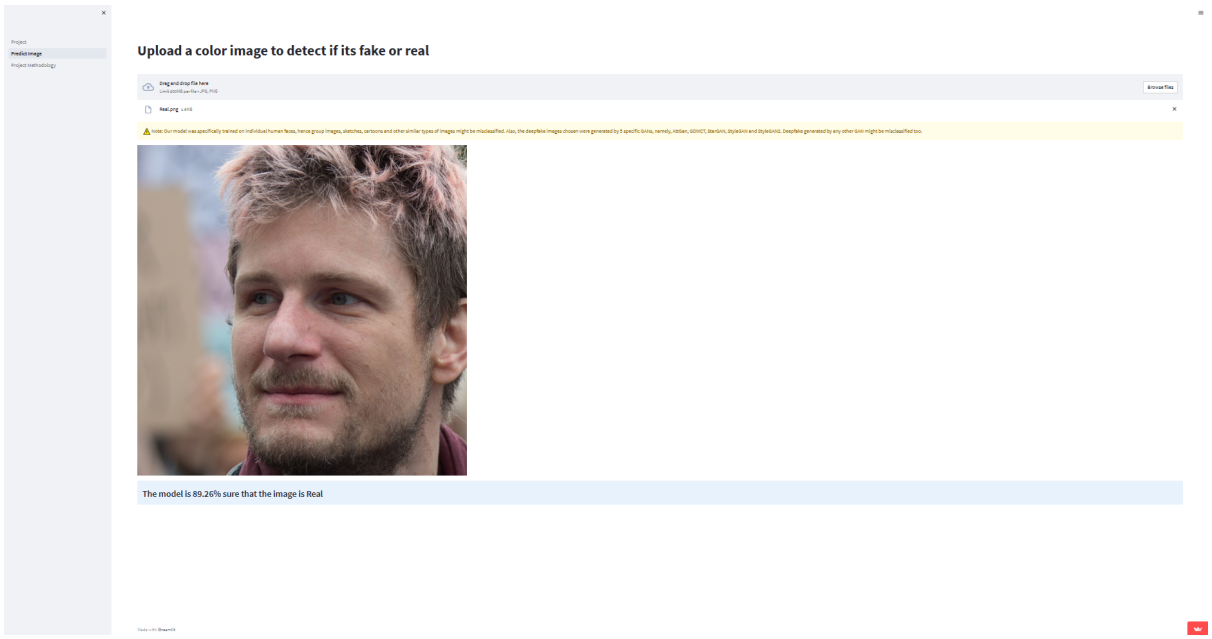

Figure 13: Model response for a real image input

The third page named “Project Methodology” has all the project information about the methodology used in the project like data collection, dataset description of the dataset used in the project, data preprocessing methods used in the project, feature extraction, model training, list of all the evaluated models with their evaluation

matrix, and finally the confusion matrix of the MobileNetV3 as this model was chosen for its highest accuracy compared to other evaluated models.

Future work

It is worth noting that the performance of the models can be heavily influenced by the choice of the augmented dataset and specific augmentations used. For example, the **MobileNetV3** model achieved better results when trained on the albumentations augmented dataset, while the **VOLO-D2** model performed better with the keras augmented dataset.

These observations highlight the importance of careful consideration when selecting and applying augmentations during training. In addition to augmentations, increasing the number of epochs, increasing the size of the dataset (by combining both of the augmented datasets), and hyperparameter tuning of the model could have improved the accuracy of the model. Experimenting with these factors in future can help to identify the optimal combination for achieving a better performance.

References

1. Liu, Ziwei, et al. "Deep learning face attributes in the wild." Proceedings of the IEEE international conference on computer vision. 2015.
2. https://github.com/NVlabs/ffhq-dataset
3. Choi, Yunjey, et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
4. Cho, Wonwoong, et al. "Image-to-image translation via group-wise deep whitening-and-coloring transformation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.
5. He, Zhenliang, et al. "Attgan: Facial attribute editing by only changing what you want." IEEE transactions on image processing 28.11 (2019): 5464-5478.
6. Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
7. Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
8. Choi, Yunjey, et al. "Stargan v2: Diverse image synthesis for multiple domains." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
9. Guarnera, Luca, et al. "The Face Deepfake Detection Challenge." Journal of Imaging 8.10 (2022): 263.
10. Guarnera, Luca, Oliver Giudice, and Sebastiano Battiato. "Fighting deepfake by exposing the convolutional traces on images." IEEE Access 8 (2020): 165085-165098.
11. Giudice, Oliver, Luca Guarnera, and Sebastiano Battiato. "Fighting deepfakes by detecting gan dct anomalies." Journal of Imaging 7.8 (2021): 128.
12. Patel, Yogesh, et al. "An Improved Dense CNN Architecture for Deepfake Image Detection." IEEE Access 11 (2023): 22081-22095.
13. Dave, Rushit. "Deepfake Detection Analyzing Hybrid Dataset Utilizing CNN and SVM." (2023).
14. Shad, Hasin Shahed, et al. "Comparative analysis of deepfake image detection method using convolutional neural network." Computational Intelligence and Neuroscience 2021 (2021).