BENNETT UNIVERSITY
TIMES OF INDIA GROUP

imageINTELLECT

Improving Deep Fake Detection Using Deep Learning

Manvi Tandon
E21CSEU0415

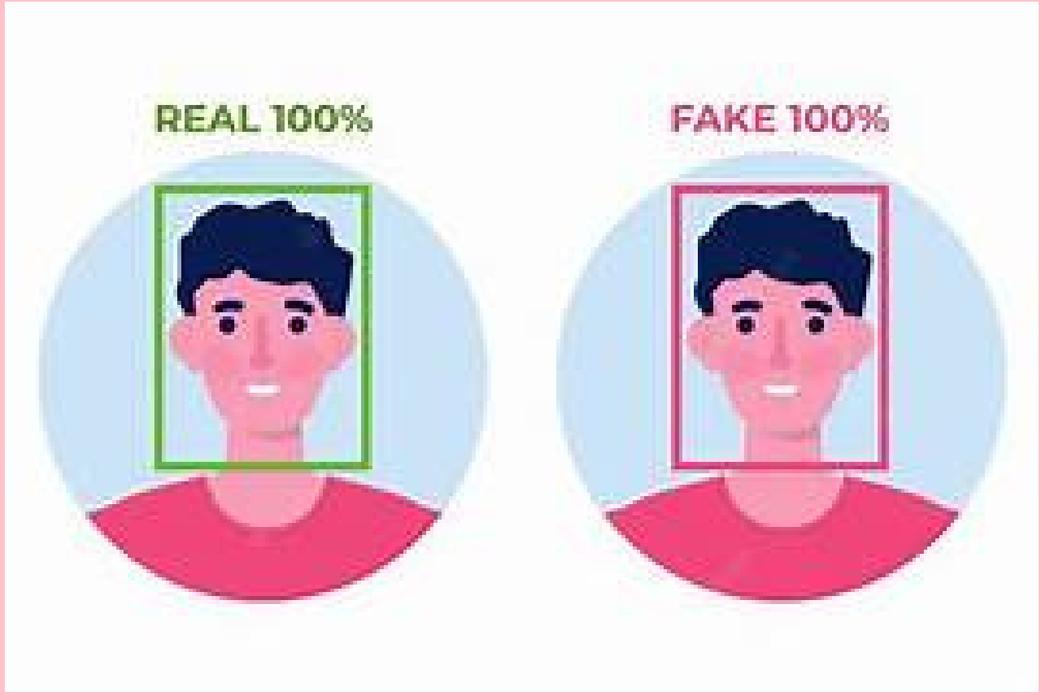
N Bala Yashaswini
E21CSEU0432

Naman Agarwal
E21CSEU0952

What is DeepFake Technology?

Deepfake technology has advanced significantly in recent years, posing serious threats to various industries. Our team aims to enhance deepfake detection using deep learning techniques to mitigate the risks associated with this technology.

What is DeepFake?

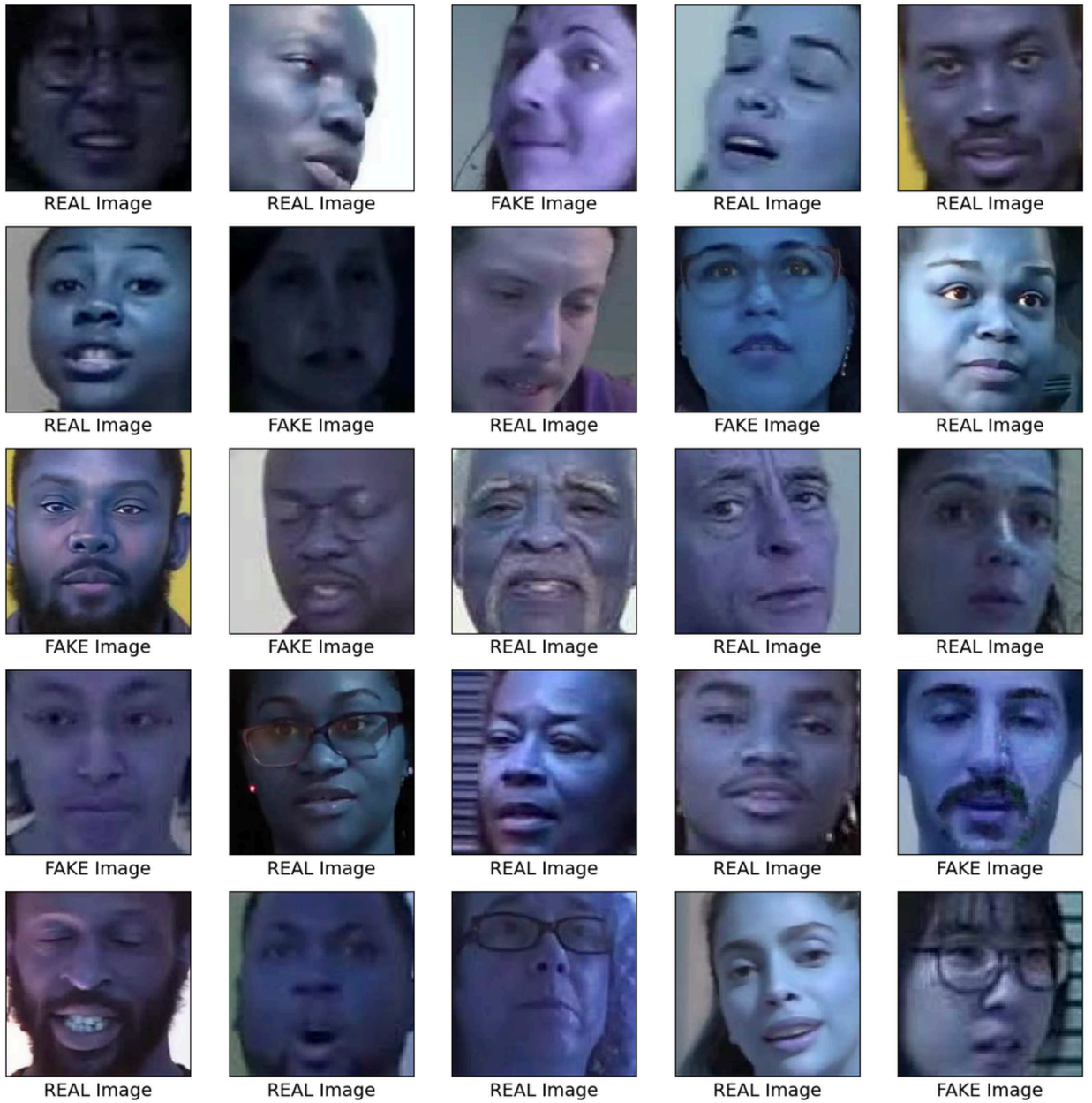


Deepfake is a term derived from "deep learning" and "fake." It refers to the use of advanced artificial intelligence, particularly deep learning algorithms, to manipulate or generate synthetic media, such as images, videos, or audio recordings, that appear convincingly real but are actually fabricated or altered. This technology can be used to superimpose the likeness of one person onto another in videos, create realistic but false scenes, or generate entirely fake content, often with the intention to deceive or spread misinformation.

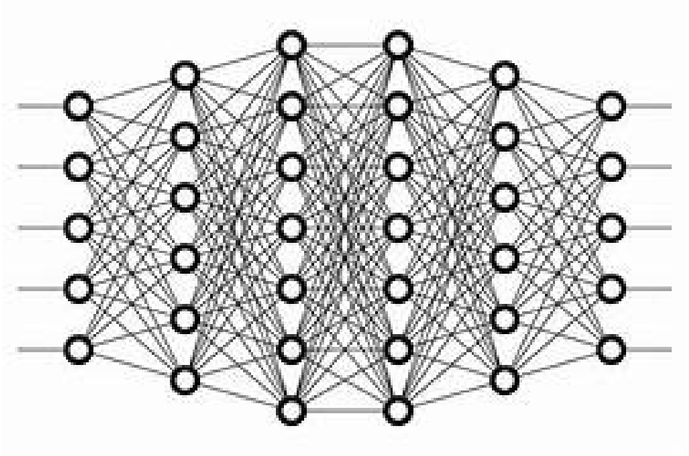
DATASET

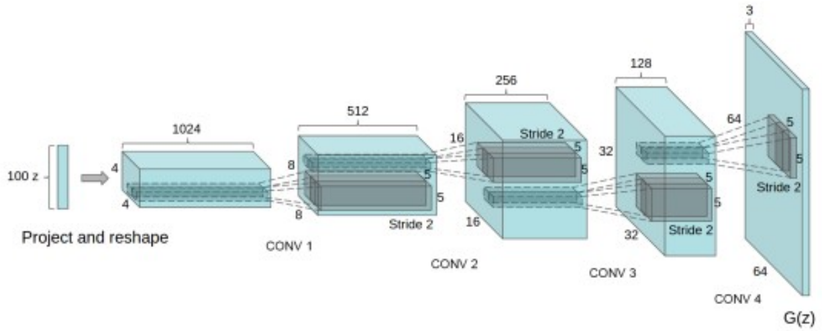
Two datasets were used:

- One contained the **faces_224** data with pictures. The faces in this dataset were taken from the Kaggle deepfake-detection challenge. Every image had a 224 x 224 resolution.
- The other is the **metadata.csv** .



Solution Approach





Custom CNN Architecture Development:

Implemented a Convolutional Neural Network (CNN) architecture tailored for deepfake detection. Preprocessed and normalized the dataset to enhance model training efficacy.

Integration of GAN with Custom CNN:

Incorporated Generative Adversarial Networks (GANs) into the custom CNN framework to augment detection accuracy

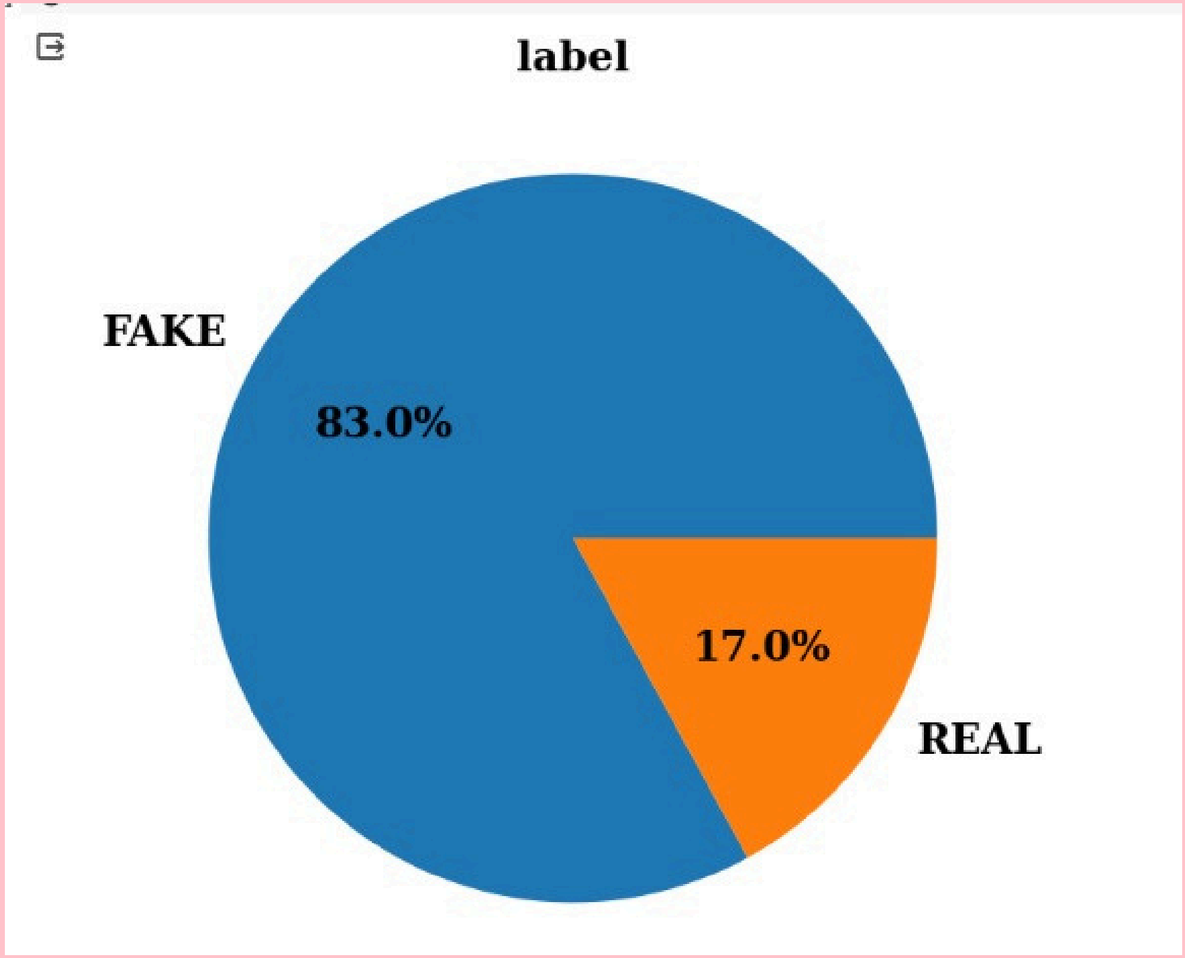
Comparative Analysis of GAN Types:

Conducted a comparative evaluation between two prominent GAN architectures, Bicyclic GAN and Deep GAN.

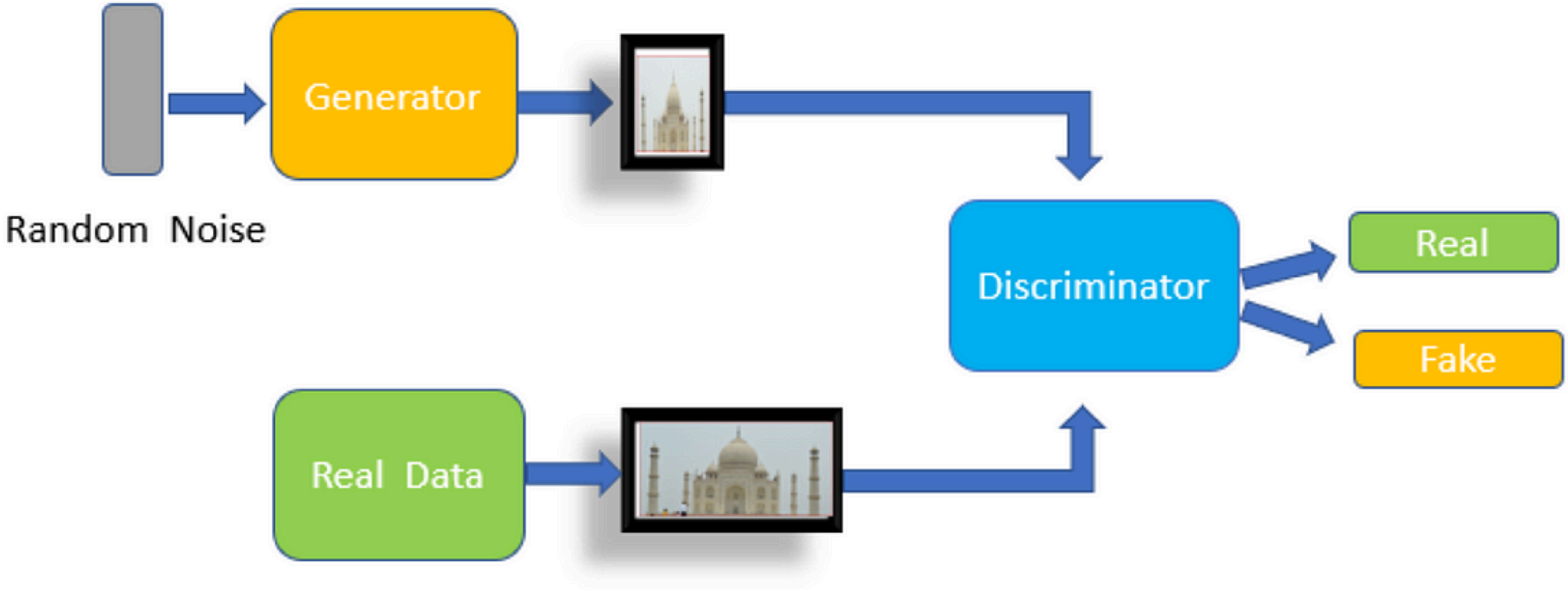
Solution Approach

By leveraging convolutional neural networks (CNNs) and GANs (Bicyclic & Deep), we aim to create a tool capable of accurately identifying synthetic media content across various platforms and scenarios.

Through extensive dataset curation, model training, and integration into a user-friendly interface, we aim to combat the spread of disinformation and preserve trust in digital media integrity.



Solution Approach



Individual Contribution

N Bala Yashaswini

- Literature Survey (Research phase)
- Worked on GAN model and Types of GANs

Manvi Tandon

- Literature Survey (Research phase)
- Data Collection and Pre Processing

Naman Agarwal

- Literature Survey (Research phase)
- Worked on Custom CNN architecture

/09

Conclusion

```
Generator(
  (main): Sequential(
    (0): ConvTranspose2d(100, 512, kernel_size=(4, 4), stride=(1, 1), bias=False)
    (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): ConvTranspose2d(512, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (4): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): ConvTranspose2d(256, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (7): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (8): ReLU(inplace=True)
    (9): ConvTranspose2d(128, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (10): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (11): ReLU(inplace=True)
    (12): ConvTranspose2d(64, 3, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (13): Tanh()
  )
)
```

```
Discriminator(
  (main): Sequential(
    (0): Conv2d(3, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (1): LeakyReLU(negative_slope=0.2, inplace=True)
    (2): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (3): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): LeakyReLU(negative_slope=0.2, inplace=True)
    (5): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (6): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): LeakyReLU(negative_slope=0.2, inplace=True)
    (8): Conv2d(256, 512, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1), bias=False)
    (9): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (10): LeakyReLU(negative_slope=0.2, inplace=True)
    (11): Conv2d(512, 1, kernel_size=(4, 4), stride=(1, 1), bias=False)
    (12): Sigmoid()
  )
)
```

References

- https://scholar.google.com/
- https://www.kaggle.com/
- https://www.analyticsvidhya.com/

Thank You