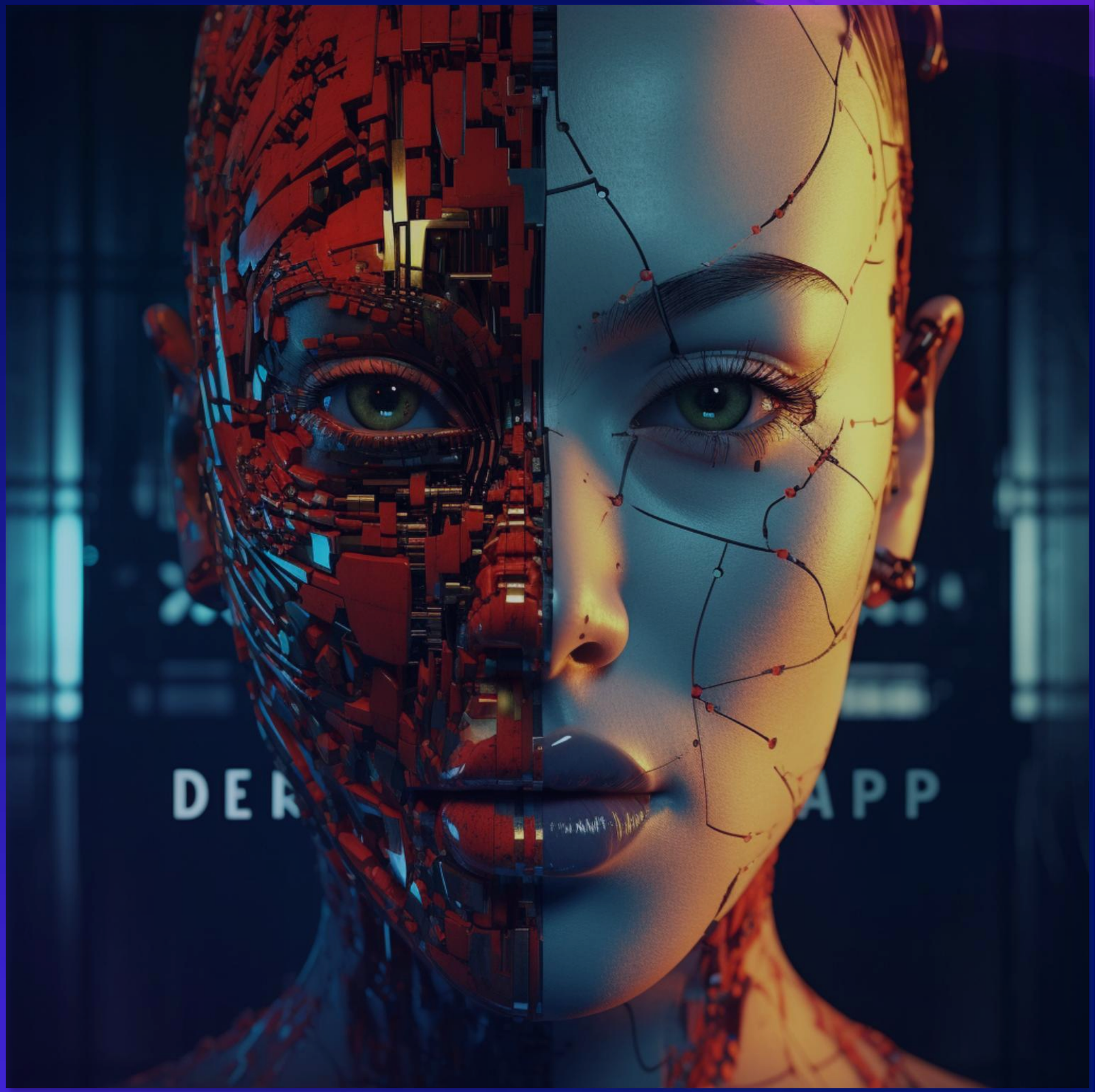
ORGANIZED BY

School of Computer Engineering

DEEPFAKE DETECTION USING Artificial Intelligence



Guided by:
Prof. Sujoy Datta
Assistant Professor, School of Computer Engineering
KIIT (Deemed to be University), Bhubaneswar

Presented by:
Sanskar Shukla(21051247)
Riya Singh(21052783)
Priyanshu Gupta(2105393)
Kunal Kishore(21051060)
Siddhant Kumar(21052970)

CONTENTS

01 Abstract

02 Introduction

03 Objective

04 Methodology

05 Working Flow

06 Libraries Used

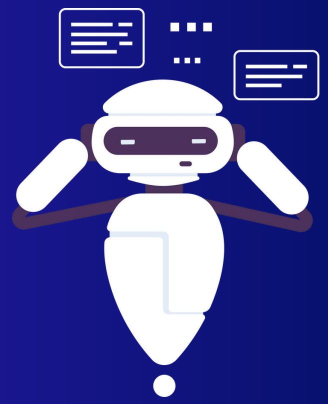
07 Layers of Keras Used

08 Result Analysis

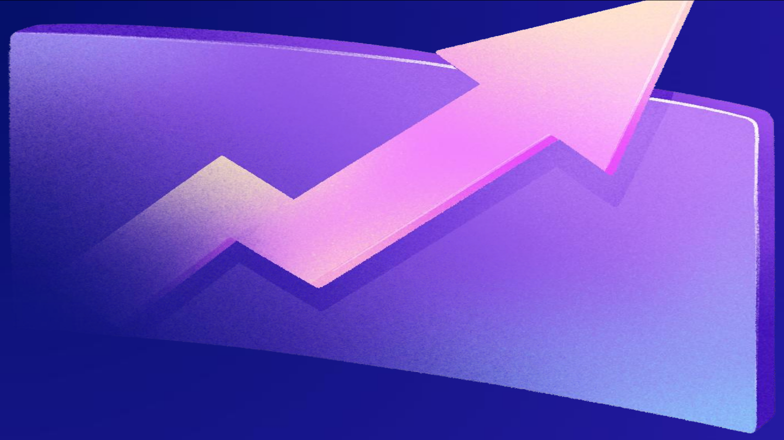
09 Conclusion

10 References

Abstract



With the rise of deepfake technology, distinguishing authentic multimedia content from manipulated media has become increasingly challenging. This paper presents a concise summary of recent advancements in utilizing machine learning for deepfake detection. Various methodologies, including CNNs, RNNs, and GANs, are explored, alongside discussions on feature selection, dataset choices, and evaluation metrics. The paper also highlights ongoing challenges and future directions in the field, emphasizing the urgency for robust solutions against the spread of synthetic media.



INTRODUCTION

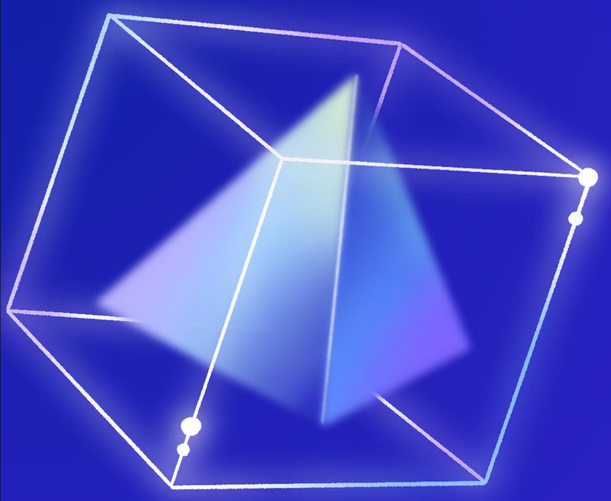
- **Rise of Deepfake Technology:** Introduction to the growing prevalence of deepfake technology in generating manipulated multimedia content.

- **Concerns and Challenges:** Discussion on the implications and challenges posed by deepfakes, including their potential to spread misinformation and deceive viewers.

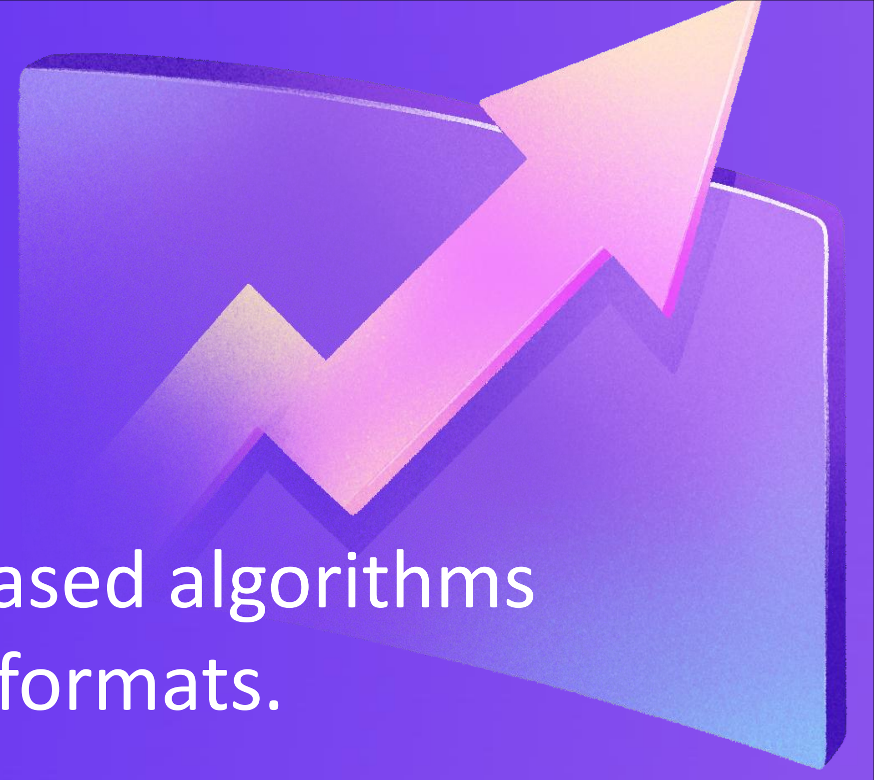
- **Importance of Detection:** Emphasizing the critical need for effective deepfake detection methods to maintain the authenticity and trustworthiness of multimedia c**ontent.**

- **Leveraging Machine Learning:** Introduction to the utilization of machine learning techniques for deepfake detection, highlighting its potential in addressing this pressing issue.

- **Overview of Presentation:** Brief outline of the presentation's focus on exploring various machine learning approaches, including CNNs, RNNs, and GANs, for detecting deepfakes.



OBJECTIVE



- **Develop Robust Detection Methods:** To create and refine machine learning-based algorithms capable of accurately identifying deepfake content across various multimedia formats.

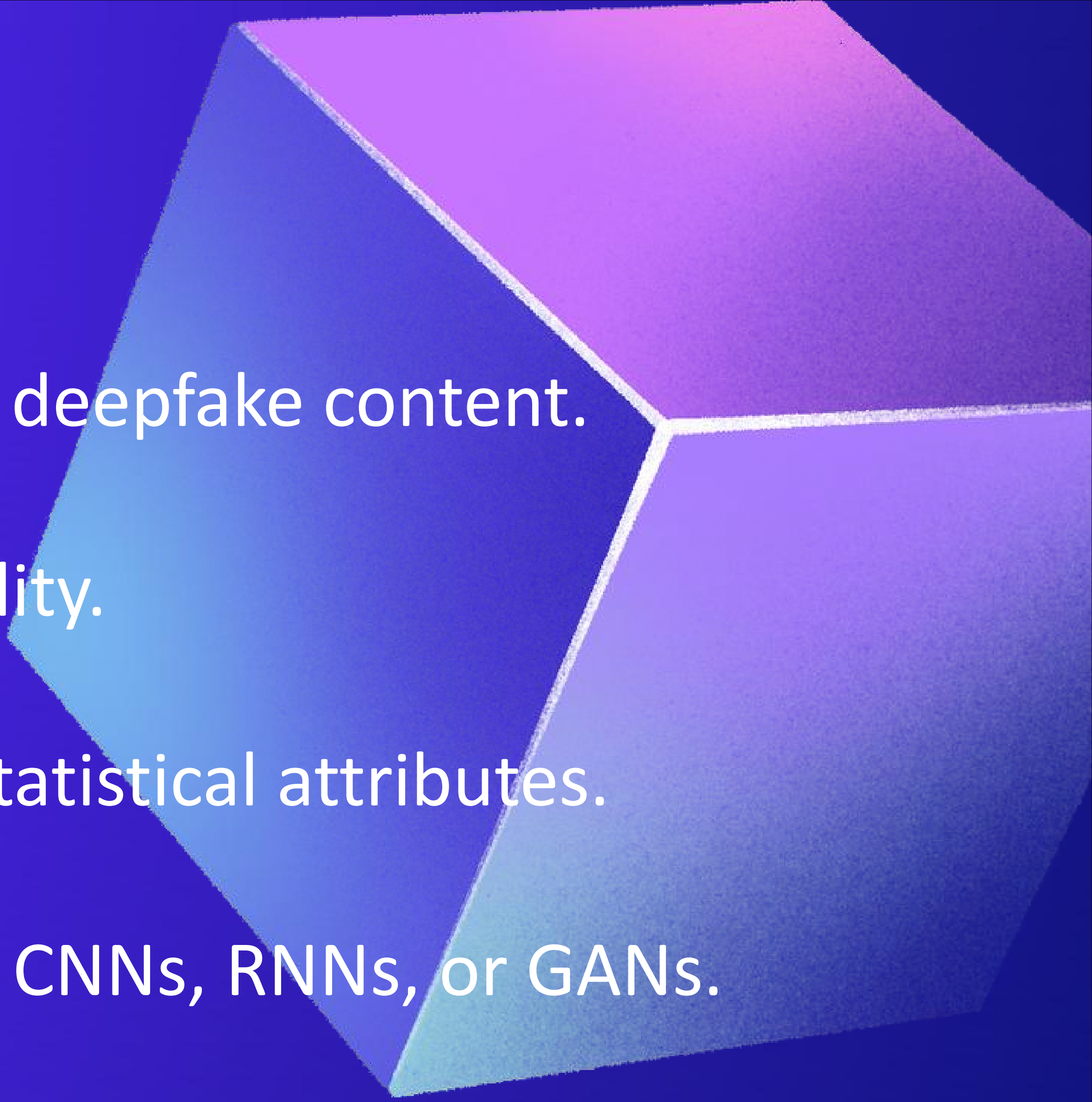
- **Enhance Detection Accuracy:** Improve the precision and reliability of deepfake detection techniques to effectively differentiate between authentic and manipulated media.

- **Address Emerging Threats:** Stay abreast of evolving deepfake technologies and anticipate future challenges by continuously updating detection methodologies and strategies.

- **Foster Trust and Integrity:** Promote transparency and accountability in multimedia content by establishing trustworthy mechanisms for detecting and flagging deepfakes.

- **Empower Stakeholders:** Equip users, content creators, and platforms with tools and resources to identify and mitigate the impact of deepfake dissemination on digital ecosystems.

METHODOLOGY



- **Data Collection:** Gather diverse datasets containing both authentic and deepfake content.
- **Preprocessing:** Standardize and preprocess data for uniformity and quality.
- **Feature Extraction:** Extract relevant features like facial landmarks and statistical attributes.
- **Model Selection:** Choose appropriate machine learning models such as CNNs, RNNs, or GANs.
- **Training:** Train models using preprocessed data, employing techniques like transfer learning.
- **Evaluation:** Assess performance using metrics like accuracy, precision, recall, and F1-score.
- **Refinement:** Fine-tune model parameters and adjust feature selection criteria based on evaluation results.
- **Validation:** Validate methodology through real-world testing and comparison with existing approaches.

WORKING FLOW

Understanding the Deepfake Workflow

1. Data Collection:

- The heart of any deepfake lies in large datasets of images or videos featuring the target person whose likeness is to be manipulated.
- The more data (different angles, lighting conditions, expressions), the more realistic the ultimate deepfake.

2. Autoencoders:

- Encoder: This neural network compresses the input image/video into a compact representation called an 'encoding' or 'latent representation'. It focuses on extracting the essential features of the target person's face.
- Decoder: This neural network takes the encoding and attempts to reconstruct the original image/video. It learns how to generate a face that closely resembles the target face.

Adversarial Training:

- Two Decoders: Deepfake systems use two decoders - one trained on the target person ('Decoder A') and one trained on a different person ('Decoder B').
- Generator: This component is shared by both decoders. Its role is to take an encoding and pass it to the appropriate decoder.
- Discriminator: This neural network aims to distinguish between real images/videos of the target and images/videos generated by the decoders.
- Competition: The generator and the discriminator are locked in an adversarial game. The generator tries to fool the discriminator, while the discriminator improves its ability to spot fakes. This forces the generator to produce increasingly realistic deepfakes.

Face Swapping:

- Encoding Extraction: An image or video of the person to be inserted into the deepfake has its face encoded.
- Feeding the Encoding: This encoding is fed into the trained generator, along with the target decoder (Decoder A).
- Output: The generator produces a new image or video frame where the original face has been replaced with the new person's face, realistically blending with the target's features and expressions.

Libraries Used

These libraries cover the core tasks of data handling, feature extraction, model training, and deployment in deepfake detection using machine learning.

1. Data Handling and Preprocessing:

- NumPy
- OpenCV

2. Feature Extraction:

- Dlib
- TensorFlow or PyTorch (for deep learning-based features

3. Model Selection and Training:

- TensorFlow or PyTorch
- Scikit-learn

4. Deployment:

- TensorFlow Serving or TensorFlow Lite
- PyTorch Mobile

LAYERS OF KERAS USED

In deepfake detection using machine learning with Keras, typically layers from the following categories are commonly used:

1. Input Layers:

- Input

2. Convolutional Layers:

- Conv2D
- MaxPooling2D

3.Normalization Layers:

- BatchNormalization

4.Activation Layers:

- ReLU, LeakyReLU, ELU, etc.

5. Flattening and Reshaping Layers:

- Flatten
- Reshape

6. Dense Layers:

- Dense
- Dropout

7. Output Layers:

- Dense (with appropriate activation)

RESULT ANALYSIS

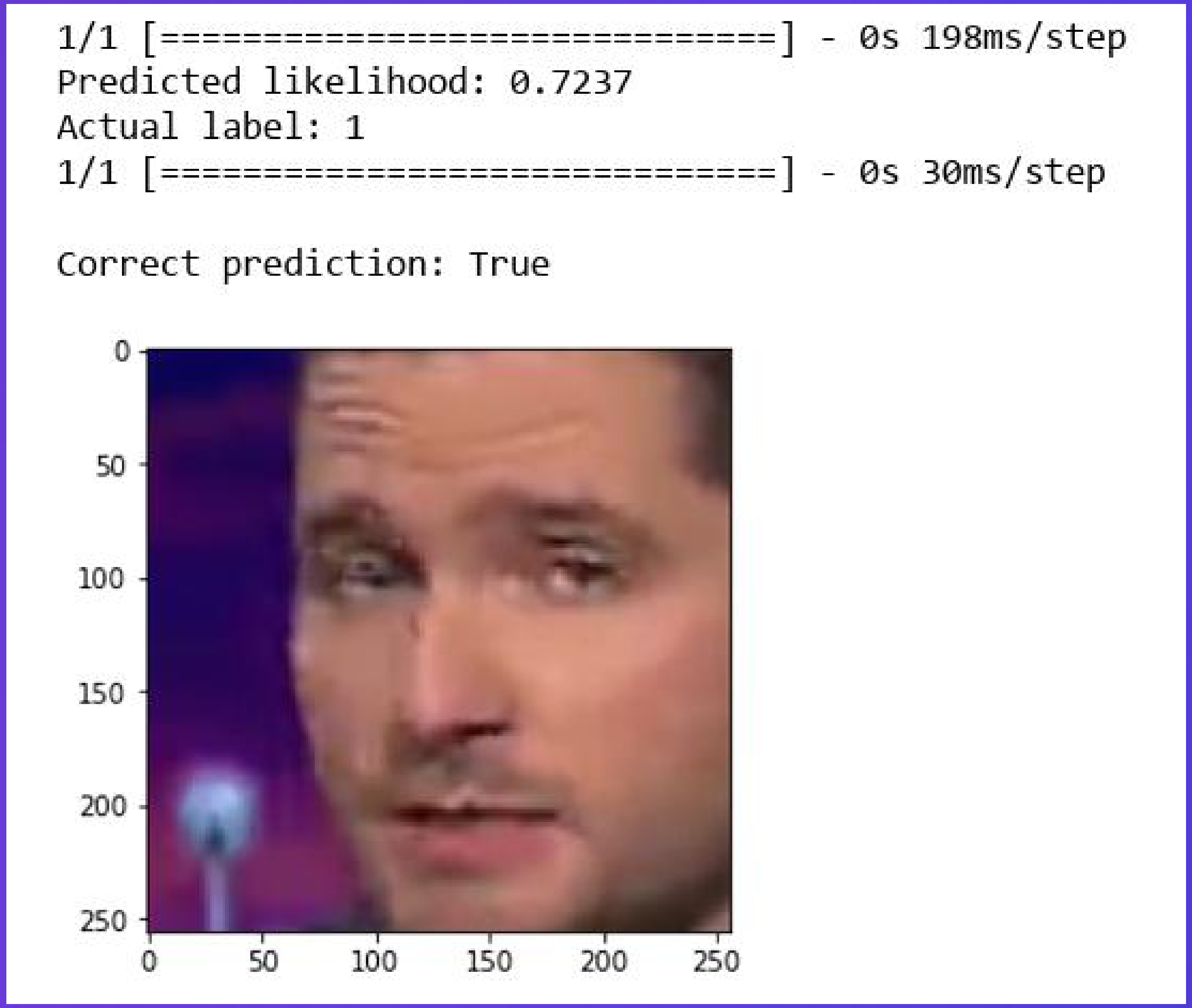
The result analysis phase involves interpreting the performance metrics obtained during testing and drawing conclusions about the effectiveness of the deepfake detection models. Key aspects of result analysis include:

Performance Comparison: Compare the performance of different detection models using evaluation metrics to identify the most effective approach for deepfake detection.

Generalization: Assess the generalization capabilities of the trained models across different datasets and manipulation techniques to determine their applicability in real-world scenarios.

Future Directions: Based on the result analysis, propose recommendations for future research and development efforts to enhance the effectiveness and robustness of deepfake detection methods.

01



```
1/1 [==============================] - 0s 198ms/step
Predicted likelihood: 0.7237
Actual label: 1
1/1 [==============================] - 0s 30ms/step

Correct prediction: True
```

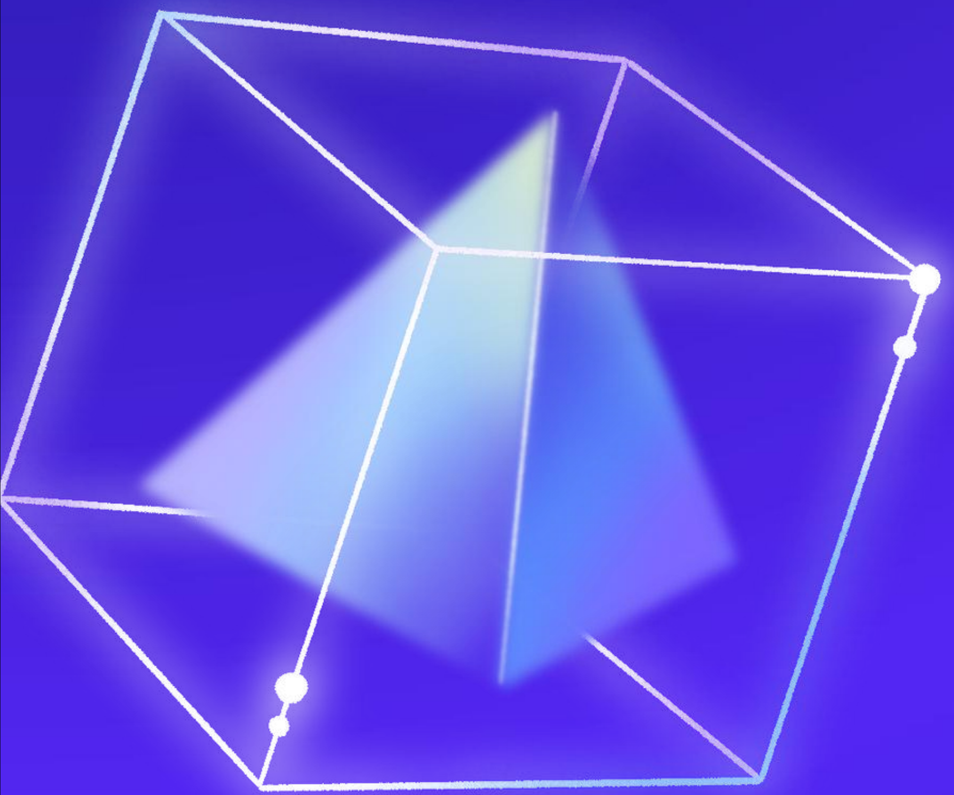
CONCLUSION

- Deepfake detection is a critical area of research and application due to the increasing sophistication of AI-generated content.
- Machine learning, particularly using frameworks like Keras, offers powerful tools for identifying and mitigating the risks posed by deepfakes.
- By leveraging techniques such as convolutional neural networks (CNNs) and advanced layers in Keras, we can build robust models capable of detecting manipulated media.
- However, the arms race between deepfake creators and detection methods continues, highlighting the need for ongoing research and collaboration in this field.
- As we move forward, it's crucial to stay vigilant, innovate, and adapt our approaches to effectively combat the challenges posed by deepfake technology.

REFERENCES

- Y. Zhang and V. M. Patel, "A Review of Deep Learning-Based Methods for Deepfake Detection," IEEE Access.
- R. Sabatini, "Detecting Deepfakes Using Keras and TensorFlow," Towards Data Science, 2021.
- K. Hao, "Deepfake Detection: Current Challenges and Next Steps," Stanford HAI Blog, 2020.
- M. Ye et al., "DeepFake Detection Based on Attention Mechanism and LSTM," IEEE Access.
- H. Farid, "Deepfake Detection: Current Techniques and Future Directions," IEEE Signal Processing Magazine.

END OF PRESENTATION



THANK YOU!