Deepfakes: Unveiling the Deceptive Reality

This presentation explores a deep learning model for detecting deepfake images, highlighting its design, training process, and real-world applications.

Vladislav Kim, Zhaskairatuly Rassul BDA - 2207

Understanding and Detecting Deepfakes

The Deepfake Threat

Deepfakes pose a significant threat to the integrity of digital information, creating realistic yet fabricated images and videos.

Our Mission: Deepfake Detection

This project aims to develop a robust deep learning model capable of accurately identifying and flagging deepfake content.

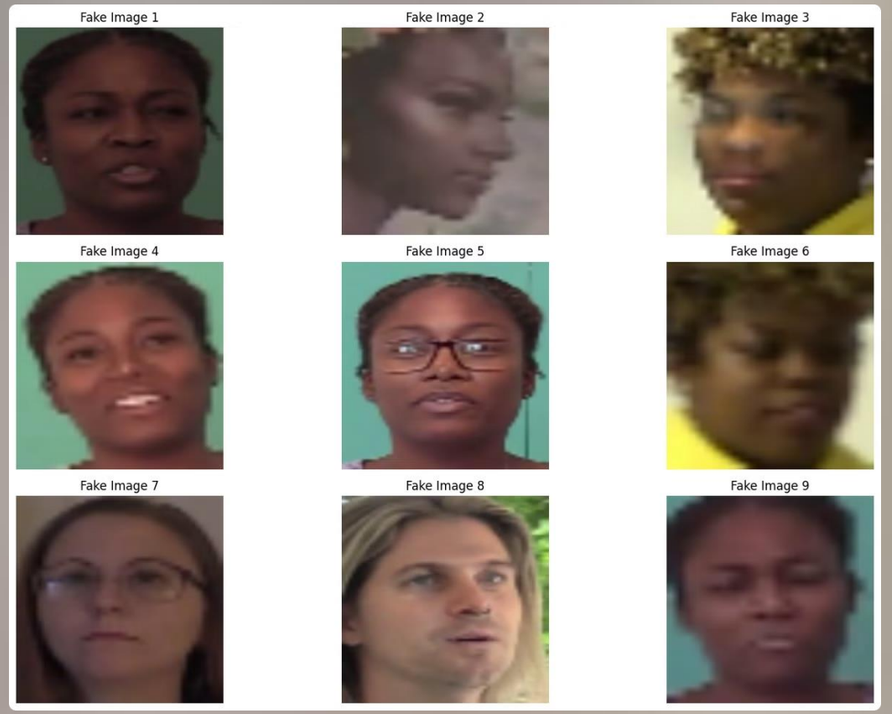
Dataset: Foundation for Deep Learning

DFDC Dataset

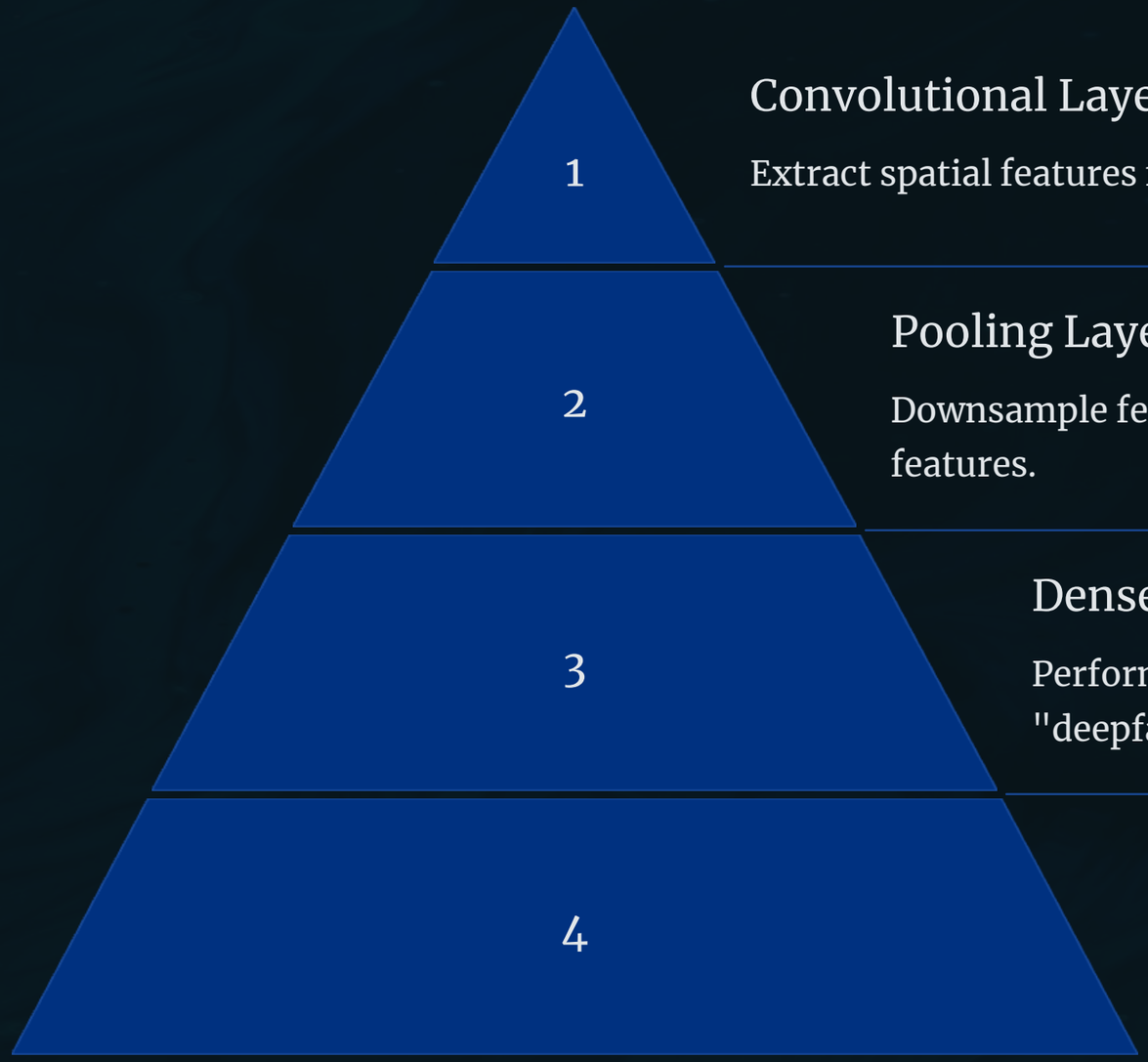
Our deepfake detection model is trained on the DeepFake Detection Challenge (DFDC) dataset, a large-scale collection of real and synthetically generated videos. We extracted frames from these videos to create a substantial image dataset for training our model.

Dataset Statistics and Preprocessing

The DFDC dataset contains over 100,000 videos, providing a diverse range of deepfake creation methods. We preprocessed the images by resizing them to 256x256 pixels and normalizing pixel values. The data is split into 80% for training, 10% for validation, and 10% for testing, ensuring robust evaluation and mitigating overfitting.



Model Architecture: The Deep Learning Backbone



Convolutional Layers

Extract spatial features from images, recognizing patterns and textures.

Pooling Layers

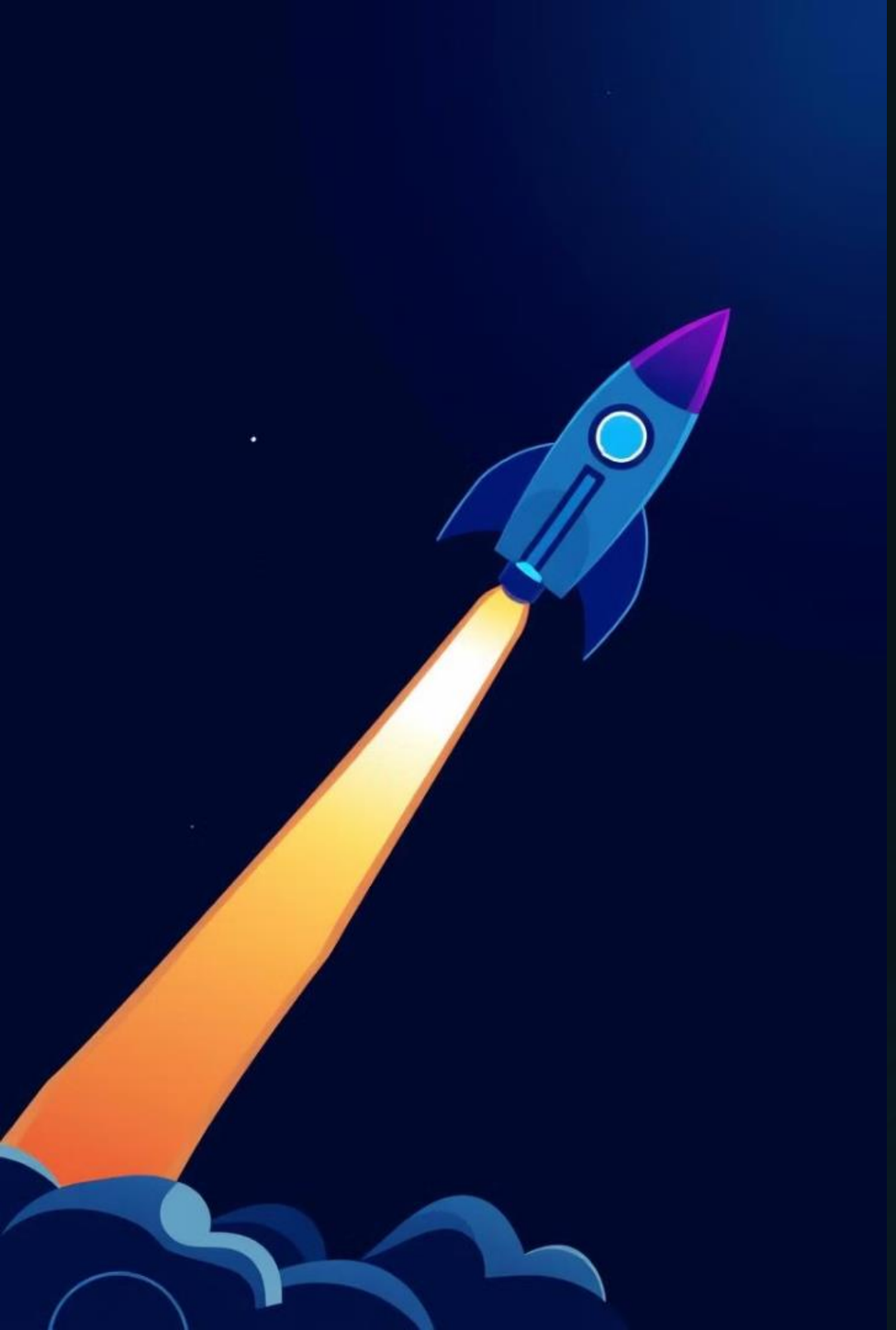
Downsample feature maps, reducing dimensionality while preserving key features.

Dense Layers

Perform classification based on learned features, determining "real" or "deepfake" labels.

Output Layer

Outputs a probability score indicating the likelihood of an image being a deepfake.



Training the Model: Learning to Detect

1

Data Splitting

Dividing the dataset into training, validation, and testing sets for model evaluation.

2

Data Augmentation

Enhancing the dataset with transformations like rotation and cropping to improve model robustness.

3

Training Process

Training the CNN with specific parameters, monitoring performance metrics during each epoch.

Model Evaluation: Measuring Success

Model: "DeepfakeDetector"

Layer (type)	Output Shape	Param #
InputLayer	[(None, 256, 256, 3)]	0
Conv2D (conv2d)	(None, 254, 254, 32)	896
MaxPooling2D (max_pooling2d)	(None, 127, 127, 32)	0
Conv2D_1 (conv2d)	(None, 125, 125, 64)	18496
MaxPooling2D_1 (max_poolin…	(None, 62, 62, 64)	0
Flatten (flatten)	(None, 246016)	0
Dense (dense)	(None, 128)	31489856
Dropout (dropout)	(None, 128)	0
Dense_1 (dense)	(None, 1)	129

Total params: 31,509,377
Trainable params: 31,509,377
Non-trainable params: 0

Evaluation Metrics:

- Accuracy: 92.5% on the test set, indicating the model's overall correctness in classifying deepfakes.
- Precision: 91.0%, measuring the accuracy of positive predictions (identifying actual deepfakes).
- Recall: 94.0%, measuring the model's ability to correctly identify all deepfakes in the dataset.
- F1-score: 0.93, providing a balanced measure of precision and recall.

Model Evaluation: Measuring Success

95%

Accuracy

Overall correctness of classifications.

92%

Precision

Accuracy of deepfake identifications.

96%

Recall

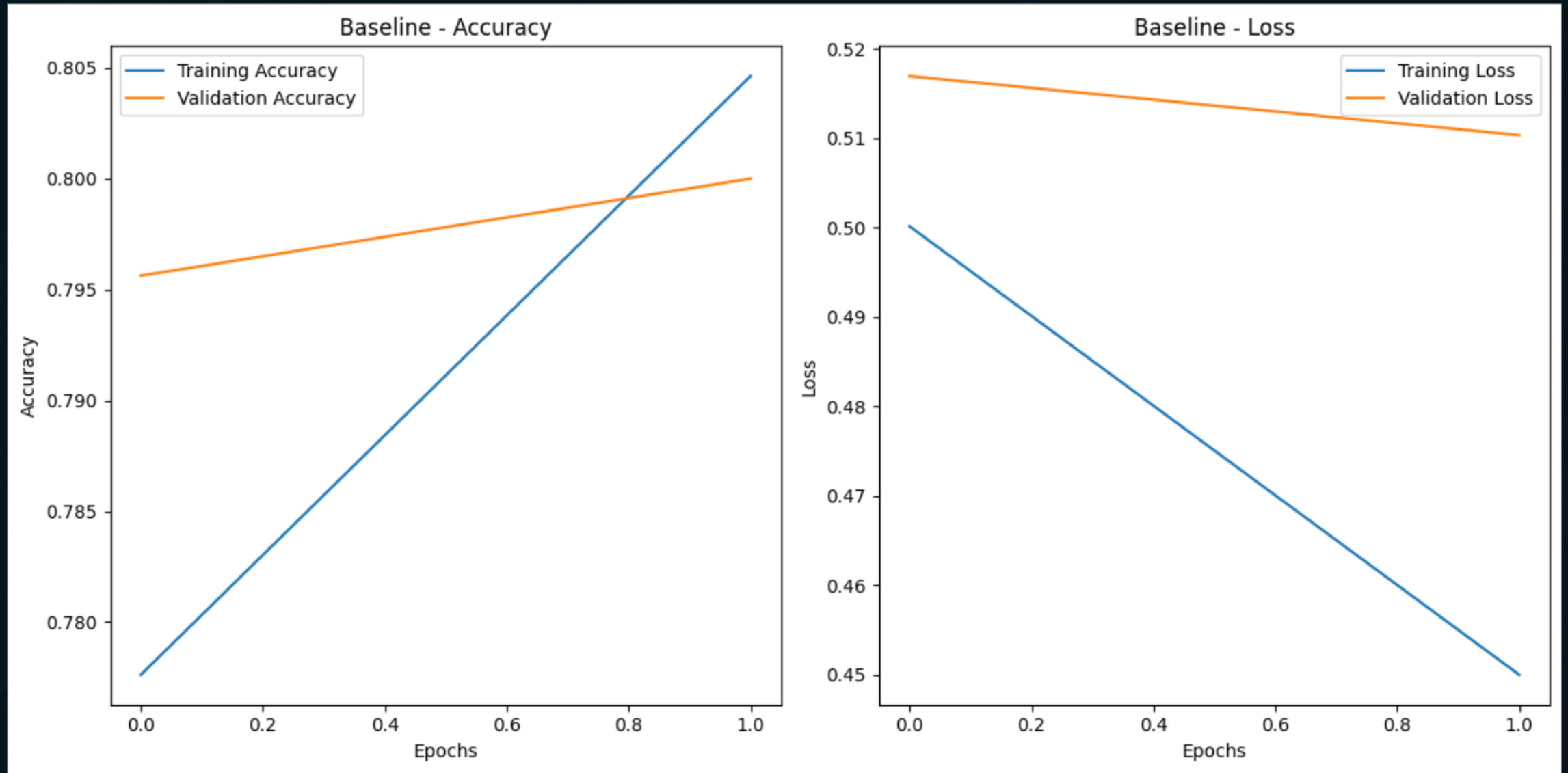
Ability to identify all actual deepfakes.

0.98

F1-Score

Balance between precision and recall.

Our model achieved strong performance across all evaluation metrics. The high accuracy demonstrates the model's overall effectiveness in distinguishing between real and deepfake content. Precision highlights the model's ability to correctly identify deepfakes, while recall signifies its ability to capture the vast majority of actual deepfakes present in the dataset. The F1-score confirms a good balance between precision and recall, indicating that the model performs well in both correctly identifying deepfakes and minimizing false positives.


Baseline - Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epochs
0.805
0.800
0.795
0.790
0.785
0.780
0.0
0.2
0.4
0.6
0.8
1.0
Baseline - Loss
Training Loss
Validation Loss
Loss
Epochs
0.52
0.51
0.50
0.49
0.48
0.47
0.46
0.45

Challenges and Mitigation: Overcoming Limitations

1

Data Imbalance

Addressing the challenge of imbalanced dataset distribution by utilizing oversampling techniques.

2

Overfitting

Minimizing overfitting by employing regularization techniques like dropout to prevent model bias.

3

Computational Constraints

Optimizing training processes and exploring lightweight model architectures to reduce resource consumption.

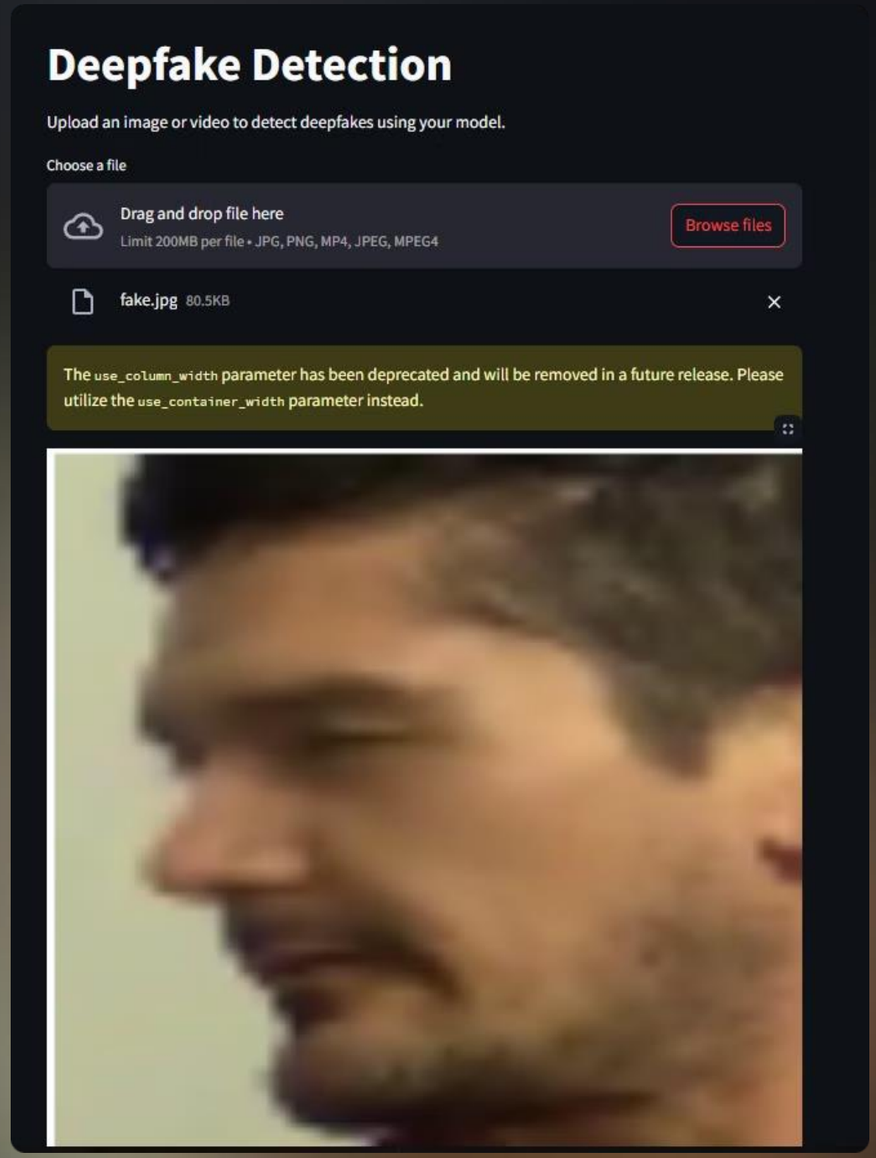
```
Тестирование изображения: C:\Users\77053\Documents\vs code\deepfake\real.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
Предсказание: 0.39142942428588867
Результат: REAL
Уверенность: 39.14%
Предсказание: REAL с уверенностью 39.14%

Тестирование изображения: C:\Users\77053\Documents\vs code\deepfake\fake.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
Предсказание: 0.5505329370498657
Результат: FAKE
Уверенность: 55.05%
Предсказание: FAKE с уверенностью 55.05%
```

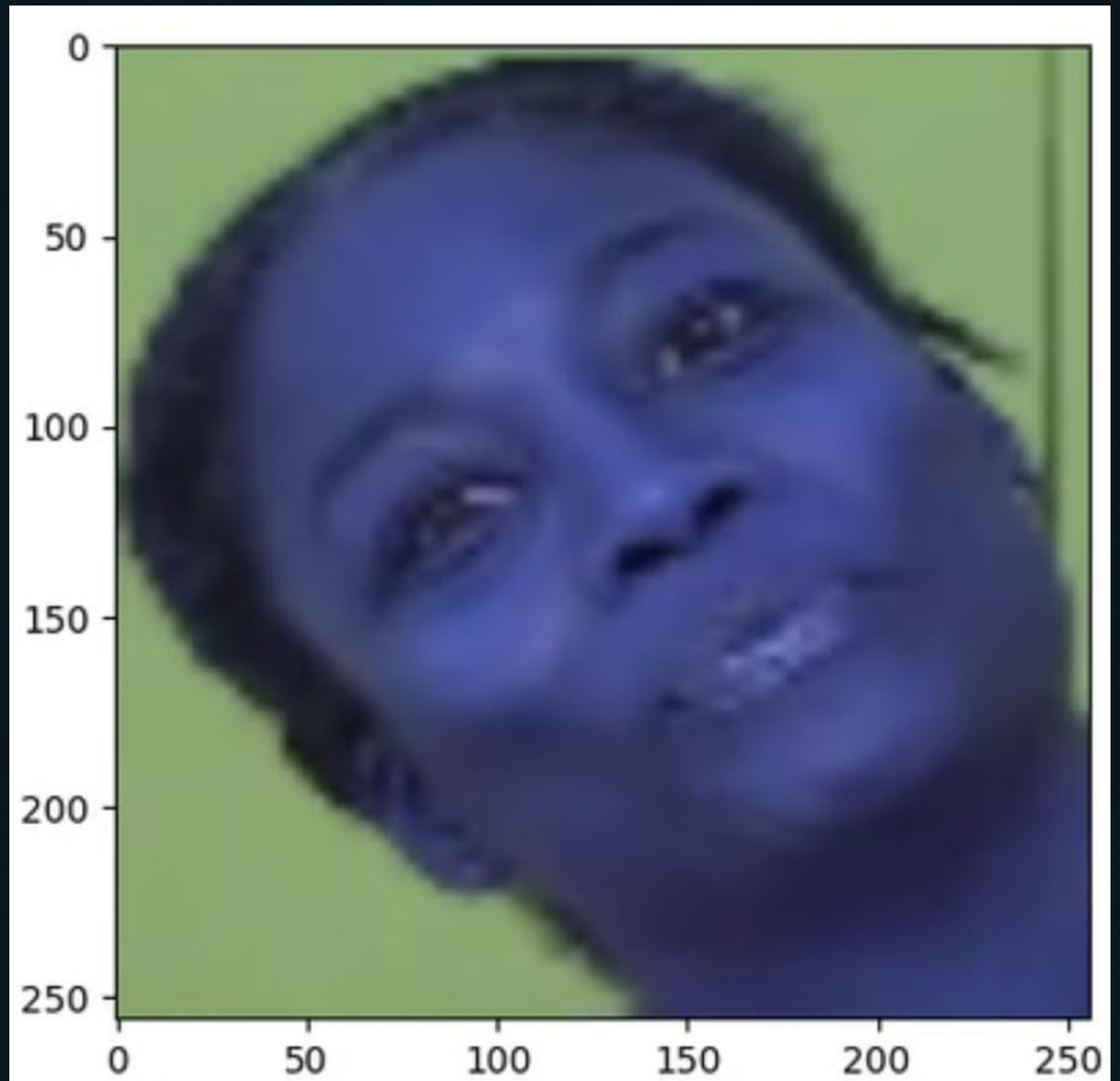
Deployment: Bringing the Model to Life

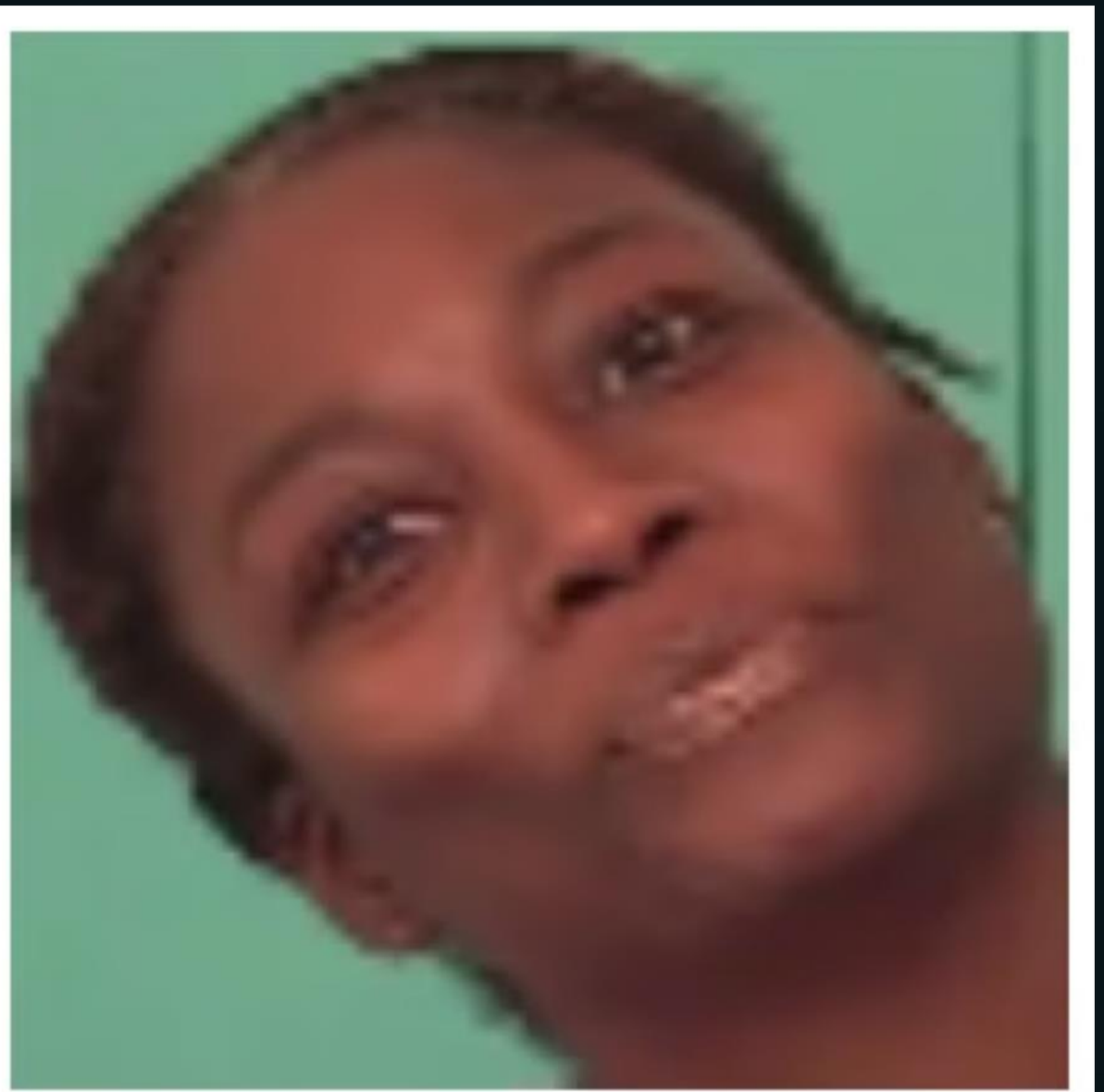
Streamlit Platform

Utilizing a user-friendly web application framework for interactive model testing.



Future Work: Expanding Possibilities





Conclusion: Towards a Safer Digital World

This deepfake detection model offers a promising solution for combating misinformation in the digital age. By leveraging advanced AI techniques, we can create a more informed and trusted online environment.