Exploring the Role of Loss Functions in Deepfake Detection

Anjali Sharma
Artificial Intelligence and Data Science
Indira Gandhi Delhi Technical University for Women
Delhi, India
anjali018btcseai23@igdtuw.ac.in

***Abstract*— With deepfakes becoming more common, it's increasingly difficult to separate real content from manipulated ones. As vast amounts of data are generated daily, the need for effective deepfake detection becomes increasingly critical. The aim of this work is to apply a basic CNN model to classify images as real or fake. We examine how various loss functions, including binary cross-entropy, focal loss, and hinge loss, impact training and detection performance. The models were evaluated on a dataset of 190,335 images with a focus on accuracy and loss. Binary Cross-Entropy performs well on balanced datasets, Focal Loss shows superior performance in imbalanced datasets, and Hinge Loss provides robust margin-based classification. Our findings provide insights into the optimal loss functions for enhancing the performance of deepfake detection models. At 92.70% training accuracy and a 0.0118 training loss, the Focal Loss model emerged as the best, providing a solid foundation for future advancements in deepfake detection.**

Introduction

In recent technological development, the rapid growth of AI has given rise to many techniques for digitally manipulating images. These manipulated images are referred as Deepfakes. Deepfake is a term for fake images, audios and videos generated using AI. These can be almost identical to genuine images, which raises worries about how this technology might be misused. Making a deepfake might sound very complicated but it is a very easy process. Figure 1 shows some fake images which are almost similar to genuine images and very hard to interpret by human eyes. The tools for creating deepfakes are available to everyone which makes the deepfake a potential threat for the society. Deepfakes are generally created using two techniques. One example is Autoencoders, a neural network used for unsupervised learning that works through Encoding and Decoding. A different strategy is employing Generative Adversarial Networks (GANs), where the generator creates fake content from random noise, and the discriminator judges whether it's real or fake.

The ability to create convincing deepfakes has far-reaching implications, particularly in areas such as misinformation, privacy violations, and cybersecurity. Politically, deepfakes can generate fake interviews or speeches from politicians, possibly affecting election outcomes. They're also used to create non-consensual explicit content, especially against women, and to impersonate others, leading to fraud or harm. The growing volume of data is making it harder to tell real content apart from manipulated ones.

Figure 1: Sample Fake Images from the Dataset Used for Model Training

Table 1 provides a list of popular tools used in the creation and manipulation of deepfakes. It includes open-source software for face swapping and reenactment, highlighting each tool's purpose and its corresponding GitHub repository for further exploration.

Table 1: Some Existing Deepfake Tools

TOOLS	LINKS	DESCRIPTION
faceswap	deepfakes/faceswap: Deepfakes Software For All (github.com)	Open-source software for swapping faces in images and videos.
DeepFaceLab	iperov/DeepFaceLab: DeepFaceLab is the leading software for creating deepfakes. (github.com)	Open-source tool for creating deepfakes.
FSGAN	YuvalNirkin/fsgan: FSGAN - Official PyTorch Implementation (github.com)	Subject-independent tool for face swapping and reenactment tool using RNNs.

Keywords: CNN, GAN, AI

StyleRig	StyleRig: Rigging StyleGAN for 3D Control over Portrait Images, CVPR 2020 (mpg.de)	Provides 3D control over StyleGAN-generated portraits.
MarioNETte	MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets (hyperconnect.github.io)	Puppet controlled by strings or wires for intricate movements.

Table 2 outlines key aspects related to the ethical and societal concerns of deepfakes. It highlights the issue of privacy, lack of consent, legal challenges, and the broader social impact. It explains how deepfake technology can affect individuals and society by undermining trust and raising ethical questions.

Table 2: Ethical Implications of Deepfakes

ASPECT	DESCRIPTION
Privacy	Concerns about unauthorized use of personal images and videos.
Consent	The creation and distribution of deepfakes often occur without the subject's permission, raising serious ethical concerns.
Legal	Deepfake technology is advancing so quickly that laws and regulations are struggling to catch up.
Social Impact	Deepfakes can erode public trust in media and digital content

The study investigates how changes in loss functions affect the efficiency of deepfake detection models. Specifically, we investigate three loss functions: Binary-Cross Entropy Loss, Focal Loss, and Hinge Loss.

I. Literature Review

This literature review surveys existing strategies for deepfake detection, focusing on their performance and limitations.

Techniques Based on Convolutional Neural Networks (CNNs)

CNNs are a preferred method for identifying deepfakes, thanks to their skill in extracting detailed visual features. The study in [12] highlighted how pre-trained models like ResNet and VGG were adapted to specific datasets to identify signs of tampering. According to [14], CNNs are capable of detecting subtle inconsistencies in facial details and textures, delivering high accuracy on standard datasets. Both studies highlighted challenges in adapting these models to emerging types of deepfakes, particularly as generation methods advance.

Temporal and Spatiotemporal Approaches

Temporal artifacts, such as jerky head movements or persistent flickering, are typical indicators of deepfake videos. The study in [13] focused on utilizing RNNs and LSTMs to analyze frame-by-frame sequences, leading to improved detection capabilities. The spatiotemporal approach in [15] combines the spatial analysis capabilities of CNNs with the temporal processing power of RNNs for effective detection. These methods, although effective, face difficulties with computational costs and scalability when applied to real-time detection.

Frequency Domain Analysis

While deepfake artifacts are difficult to spot in the spatial domain, they are more apparent in the frequency domain. [10] studied how Discrete Fourier Transform (DFT) and Wavelet Transform can be used to spot compression artifacts and blurring in deepfakes. This strategy proved useful for detecting lower-quality deepfakes but had limitations with high-resolution outputs, as described in [9].

Adversarial and Ensemble Learning

Adversarial learning techniques have been suggested to counter the complexity of GAN-generated deepfakes. [16] detailed how adversarial strategies include the parallel training of detection models and GANs, fostering a dynamic adversarial atmosphere. Additionally, [8] emphasized the use of ensemble learning methods that merge predictions from various models to improve robustness, especially in detecting different types of deepfakes.

Explainable AI in Deepfake Detection

The goal of Explainable AI (XAI) methods is to enhance the clarity as well as transparency in detecting deepfakes. [7] described how saliency maps and Grad-CAM help visualize which regions of the input data influence the model's classification outcome. This enhances the model's interpretability and highlights areas of vulnerability, as detailed in [17]. However, the extra computational cost still poses a major disadvantage.

The aim of this study is to compare different loss functions to determine which one best enhances model robustness and accuracy for deepfake detection in real-world settings.

II. Methodology

The methodology explores the impact of different loss functions in deepfake detection through Convolutional Neural Networks (CNNs). This section offers a detailed overview of the entire process, starting with data preprocessing and augmentation to ensure high-quality model training. The CNN model is designed to identify key features for deepfake detection. The following subsections detail each component of the methodology, providing insights into the experimental setup:

A. Dataset

Our models were trained with a dataset that contained 190,335 images. The dataset was collected from kaggle, which is named as "deepfake and real images" [6]. It contains digitally manipulated and real images of human faces which are created by various means. Originally available at OpenForensics under the title 'Multi-Face Forgery Detection And Segmentation In-The-Wild Dataset [V.1.0.0]' (Zenodo), the dataset was meticulously processed to optimize the input for our models [18].

The deepfake images dataset was divided into three parts: Test, Train, and Validation. The Training set includes 140,002 images, split evenly between 70,001 fake and 70,001 real

images to maintain class balance. The Validation set includes 39,428 images, with 19,641 fake and 19,787 real images, and the Test set comprises 10,905 images, with 5,492 fake and 5,413 real images.

In this project, image data preprocessing and augmentation are handled using Keras' 'ImageDataGenerator' class. This process helps to improve model generalization and performance by introducing variability and preprocessing the image data. The test data is primarily rescaled to ensure consistency in preprocessing. The images are processed by scaling their pixel values from '[0, 255]' to '[0, 1]'. Normalizing the pixel values helps the model train faster and learn more effectively. A random shearing transformation is applied to the images. Shearing introduces a tilt to the images, with a shear intensity of 20%. This augmentation helps the model adapt to different scales. The images undergo random zooming within a range of 20%. This augmentation helps the model to handle variations in scale. Images are randomly flipped horizontally. This augmentation introduces variations in orientation, which helps in making the model constant to horizontal flips. Each batch processes 32 images. The batch size controls how many images are processed to adjust the model weights at each step. The classification type is set to binary, so that the model will perform binary classification. The labels for the images will be binary (0 or 1), corresponding to the two classes.

The rescaling, shearing, zooming, and flipping augmentations enhance the diversity of the data which is being used for training, while the resizing, batch size, and class mode configurations ensure that the data is prepared appropriately for training a binary classification model. It helps the model generalize better and perform effectively on unfamiliar data.

For model compilation, the Adam optimizer was chosen, along with binary cross-entropy loss and accuracy as the evaluation metric. The Adam optimizer ensures efficient convergence during training by dynamically adjusting learning rates and the Accuracy Metric is used to monitor performance during training.

B. Model Architecture

For this project, three CNN-based models were created, each trained with a different loss function—Binary Cross Entropy, Focal Loss, and Hinge Loss—to assess their effectiveness in classifying genuine and manipulated images.

1. Binary Cross Entropy Loss Model:

The architecture of the BCE Loss model is tailored to classify genuine and fake images, using the binary cross-entropy function to enhance binary classification accuracy. The CNN model is structured with the following layers:

A. Input Layer and First Convolution Block:

The model uses a 2D convolutional layer with 32 filters and a 3 × 3 kernel size, along with ReLU activation, to capture spatial features from the input image. Resized RGB images are used as input, with dimensions of 64 × 64 × 3. A 2 × 2 max pooling layer is included to down-sample the feature maps, reducing computational complexity while retaining key features.

B. Second Convolution Block:

Another 3 × 3 convolutional layer with 32 filters is applied, followed by a ReLU activation to introduce non-linearity. A max-pooling operation is applied once more to decrease the dimensionality.

C. Flattening Layer:

The 2D feature maps are converted into a 1D vector before being fed into the fully connected layers.

D. Fully connected Layers:

A fully connected layer containing 128 neurons with ReLU activation is used to capture intricate patterns. The output layer consists of a single neuron with a sigmoid activation function, generating a probability score for binary classification.

The training process ran for a maximum of 10 epochs, using early stopping and model checkpointing to monitor and save the best model. With early stopping, training was stopped if the validation loss did not improve for 3 consecutive epochs, restoring the best weights from earlier in the training. Model checkpointing preserved the model that achieved the lowest validation loss to maintain optimal performance.

2. Focal Loss Model:

In order to manage class imbalance, the CNN model for deepfake detection was trained with the focal loss function, which places greater emphasis on difficult-to-classify examples. The architectural design was consistent with the binary cross-entropy model, with modifications to the loss function.

Focal loss is defined mathematically as,

$$FL(y_{true}, y_{pred}) = -\alpha(1 - p_t)^{\gamma}.\log(p_t) \qquad (1)$$

Where, the predicted probability for the correct class is represented by p_t, α is a parameter used to balance the class distribution, and γ helps to lower the loss from examples that are already correctly classified.

For this project, the focal loss was implemented with $\gamma = 2.0$ and $\alpha = 0.25$ to ensure that the model focuses on difficult examples during training to make the model more robust. The model was trained up to 10 epochs, with validation data used to monitor performance.

3. Hinge Loss Model:

Like the binary cross-entropy and focal loss models, the Hinge Loss model follows the same architecture and preprocessing steps, but its loss function adds a unique optimization mechanism.

For training, the model utilized hinge loss, a common choice for binary classification problems with -1 and +1 as the labels. The loss function is mathematically defined as;

$$L(y_{true}, y_{pred}) = mean(\max(0, 1 - y_{true}.y_{pred})) \qquad (2)$$

Where, y_{true} is the true label i.e., -1 or +1 and y_{pred} is the value predicted by the model's linear output layer.

The model was trained over 10 epochs with a validation set to assess generalization. Early stopping and checkpointing were not utilized in this configuration.

III. Results

This section provides a comparison of the models' performance:

1. Binary Cross Entropy Loss Model

   BCE Loss is a widely used loss function for binary classification problems, measuring how far off the predicted probabilities are from the actual binary labels.

   - Training and Validation Performance:
     i. Epoch 1: Accuracy of 72.90% and loss of 0.5231 on the training set. The validation accuracy was 78.79% with a loss of 0.4453.
     ii. Epoch 5: Accuracy improved to 89.79% with a reduced loss of 0.2378. Validation accuracy increased to 86.03% with a loss of 0.3248.
     iii. Epoch 10: The final accuracy reached 91.28%, with a training loss of 0.2087. Validation accuracy was 85.28% with a loss of 0.3401.
   - Observations:
     i. Convergence: The model converges relatively well, with steady improvement in both training and validation accuracy across epochs.
     ii. Overfitting: Overfitting became noticeable after the 5th epoch, with the validation loss ceasing to improve, while the training accuracy continued to grow.
     iii. Final Accuracy: With 91.28% accuracy on the training set and 85.28% on the validation set, the model showed good generalization.

Figure 2: Prediction of 9 random images from the Test Set by Binary Cross Entropy Model

2. Focal Loss Model

   Focal Loss addresses the class imbalance problem.

   - Training and Validation Performance:
     i. Epoch 1: High accuracy of 89.87% with a very low loss of 0.0163 on the training set. Validation accuracy was 83.70% with a loss of 0.0230.
     ii. Epoch 5: Accuracy improved to 91.79% with a training loss of 0.0131. Validation accuracy was 85.67% with a loss of 0.0232.
     iii. Epoch 10: The final accuracy reached 92.70%, with a training loss of 0.0118. Validation accuracy was 85.94% with a loss of 0.0234.
   - Observations:
     i. Convergence**:** The model showed very high training accuracy from the beginning, suggesting effective handling of the dataset by focusing on hard examples.
     ii. Generalization: While the training loss was extremely low, the validation accuracy plateaued around 85-86%, indicating that the model might be slightly overfitting despite the nature of the focal loss.
     iii. Final Accuracy: Slightly higher training accuracy compared to the BCE model, but similar validation accuracy. The low loss values indicate strong confidence in predictions.

Figure 3: Prediction of 9 random images from the Test Set by Focal Loss Model

3. Hinge Loss Model

   Hinge Loss is primarily applied in classifiers like SVMs that focus on maximizing the margin between classes. In a neural network, it pushes the model to make decisions with greater confidence, often resulting in sharper decision boundaries.

   - Training and Validation Performance:

i. Epoch 1: Achieved a high accuracy of 91.85% with a loss of 0.5820 on the training set. Validation accuracy was 85.08% with a loss of 0.6530.
ii. Epoch 5: Accuracy slightly increased to 91.70%, with a loss of 0.5824. Validation accuracy reached 86.97% with a loss of 0.6340.
iii. Epoch 10: The final accuracy was 91.58% with a training loss of 0.5837. Validation accuracy was 85.90% with a loss of 0.6446.

- Observations:

i. Convergence: The model converges relatively slower compared to the other two, as indicated by the marginal changes in accuracy and loss across epochs.
ii. Stability: While the model consistently achieved high training accuracy, the validation loss remained relatively high, indicating potential issues with generalization.
iii. Final Accuracy: Achieved good training accuracy but struggled with validation loss, which stayed high throughout, suggesting that the Hinge Loss model might not be as effective for this specific task.

Figure 4: Prediction of 9 random images from the Test Set by Hinge Loss Model

Metric	Binary Cross Entropy	Focal Loss	Hinge Loss
Training Accuracy (Final)	91.28%	92.70%	91.58%
Validation Accuracy (Final)	85.28%	85.94%	85.90%
Training Loss (Final)	0.2087	0.0118	0.5837
Validation Loss (Final)	0.3401	0.0234	0.6446
Convergence Speed	Moderate	Fast	Slow
Generalization	Good	Good	Moderate
Overfitting Tendency	Low-to-Moderate	Moderate	Moderate

Table: Summary of Comparative Analysis

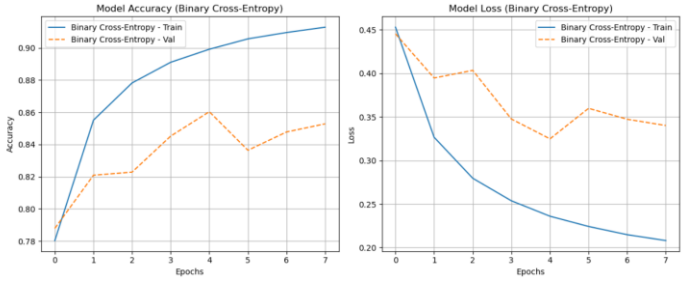

Figure 5: Accuracy and Loss curves for Binary Cross Entropy Loss

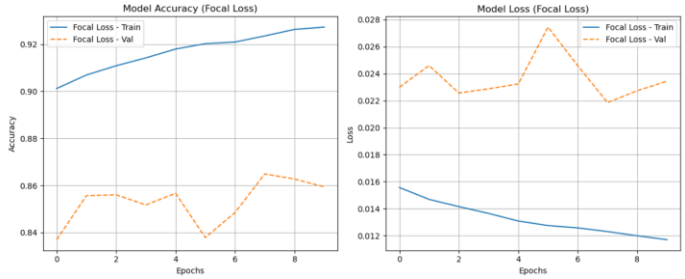

Figure 6: Accuracy and Loss curves for Focal Loss

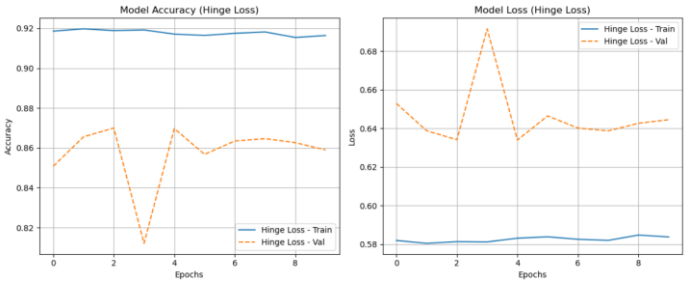

Figure 7: Accuracy and Loss curves for Hinge Loss

Conclusion:

1. Binary Cross Entropy Loss: This model showed steady improvement with balanced accuracy and loss metrics, making it a reliable choice for this task. It performed well without significant overfitting, though it required careful monitoring during training.
2. Focal Loss: This model quickly reached high accuracy and low loss values, indicating strong confidence in predictions, particularly in cases with class imbalance. However, the risk of overfitting was higher, and validation performance plateaued earlier.
3. Hinge Loss: While this model maintained high training accuracy, it struggled with higher validation loss, suggesting less effective generalization compared to the other two models. It may require more

tuning or different architecture adjustments to perform on par with the others.

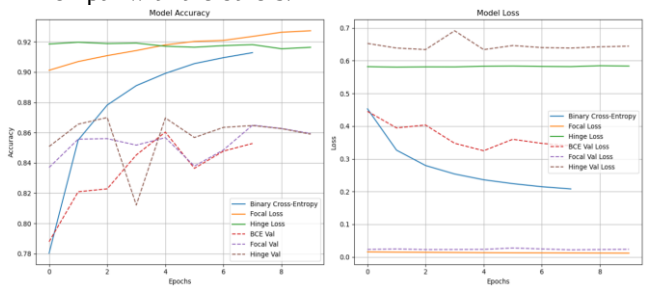

Figure 8: Accuracy and Loss Curves Comparison between Binary Cross Entropy Loss, Focal Loss and Hinge Loss

IV. Discussion

a) Implications of Findings

This analysis offers key takeaways for the design and optimization of deepfake detection systems:

- Focal Loss for Imbalanced Data: The superior performance of Focal Loss indicates its potential for improving deepfake detection systems, particularly in scenarios where the dataset might be imbalanced—i.e., real videos significantly outnumber deepfakes. Focal Loss emphasizes harder-to-classify examples, making the model more sensitive to subtle features that distinguish real from fake content. This suggests that incorporating Focal Loss into deepfake detection pipelines could lead to more robust models that are less likely to be fooled by sophisticated deepfakes.
- Binary Cross-Entropy as a Baseline: Binary Cross-Entropy's solid performance shows it remains a reliable choice for deepfake detection, particularly in straightforward classification tasks. It is a good starting point for building and iterating on detection models, especially when computational resources or time is limited.
- Hinge Loss Limitations: The underperformance of Hinge Loss suggests it may not be well-suited for deepfake detection, at least within the current architecture. This highlights the importance of selecting appropriate loss functions that align with the specific nature of the task.

b. Real-World Applications:

- Enhanced Detection Accuracy: By applying Focal Loss, detection systems could achieve higher accuracy rates, especially in environments where false negatives (failing to detect a deepfake) carry significant consequences, such as in media verification or legal evidence.
- Reduced Overfitting: The more stable performance across epochs with Focal Loss can help in deploying models that maintain their efficacy when exposed to new, unseen deepfakes, making them more reliable in real-world scenarios.

c. Comparison with Existing Research

- Focal Loss Efficacy: The effectiveness of Focal Loss in this study aligns with existing research in fields such as object detection and medical image analysis, where Focal Loss has been shown to excel in handling imbalanced datasets. However, its application in deepfake detection is relatively novel, underscoring the potential for cross-disciplinary techniques to enhance model performance.
- Binary Cross-Entropy's Popularity: Binary Cross-Entropy remains the most widely used loss function for classification tasks in deep learning, including deepfake detection. Our findings confirm its effectiveness as reported in various studies, though with limitations in overfitting that align with other research.
- Hinge Loss in SVMs: Hinge Loss has traditionally been associated with Support Vector Machines (SVMs), where it works well for margin maximization. However, as observed in this study and supported by other literature, it does not translate as effectively to neural networks for tasks like deepfake detection, which may involve more nuanced feature representations than SVMs typically handle.

d. Acknowledgment of Limitations

- Model Architecture Constraints: The results are specific to the CNN architecture used. Other architectures, such as transformers or more complex ensemble methods, might interact differently with the loss functions tested.
- Computational Resources: The number of epochs and the training duration may have been limited by available computational resources, potentially affecting the thoroughness of the model's training and evaluation.
- Focus on Loss Functions Only: While this study focused on loss functions, other hyperparameters (e.g., learning rate, batch size) and techniques (e.g., data augmentation, regularization) were not varied, which could also influence the outcomes.

e. Future Research Areas

- Testing Additional Loss Functions: Future research could explore other specialized loss functions, such as Dice Loss, Jaccard Loss, or hybrid approaches combining multiple loss functions, to further optimize deepfake detection.
- Larger and More Varied Datasets: Expanding the dataset to include a broader range of deepfakes, including those generated by emerging techniques, would provide a more rigorous test of model robustness. This could involve crowd-sourced datasets or collaboration with platforms like social media companies that encounter a wide array of deepfake content.

By addressing these areas, efficiency, robustness, and trustworthiness of deepfake detection systems can be enhanced.

V. Conclusion

In this research, the efficiency of three distinct loss functions—Binary Cross Entropy Loss, Focal Loss, and Hinge Loss—within Convolutional Neural Network (CNN) was explored. We trained and evaluated each model using a diverse set of real and altered images. The aim was to compare the performance of these loss functions based on accuracy, precision, recall, and F1-score.

Binary Cross Entropy loss model achieved excellent accuracy and consistent performance in both precision and recall. It proves to be a solid option for most deepfake detection tasks.

The Focal Loss model showed a strong ability to handle class imbalance by reducing false positives.

Finally, the Hinge Loss model performed well but showed some limitations in precision as the other two models.

The results suggest that the loss function choice plays a crucial role in determining the effectiveness of deepfake detection models. For example, when minimizing false positives is crucial, the Focal Loss might be the best choice, while Binary Cross Entropy is a reliable default option for general uses.

In conclusion, this research contributes in finding the most effective deepfake detector. Future research could explore hybrid models that integrate the advantages of various loss functions to improve detection accuracy.

VI. References

[1] Gong, L. Y., & Li, X. J. (2024). A contemporary survey on deepfake detection: datasets, algorithms, and challenges. *Electronics*, *13*(3), 585.

[2] Cozzolino, D., Rössler, A., Thies, J., Nießner, M., & Verdoliva, L. (2021). Id-reveal: Identity-aware deepfake video detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 15108-15117).

[3] Sharma, J., Sharma, S., Kumar, V., Hussein, H. S., & Alshazly, H. (2022). Deepfakes Classification of Faces Using Convolutional Neural Networks. *Traitement du Signal*, *39*(3).

[4] Tewari, A., Elgharib, M., Bharaj, G., Bernard, F., Seidel, H. P., Pérez, P., ... & Theobalt, C. (2020). Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6142-6151).

[5] Ha, S., Kersner, M., Kim, B., Seo, S., & Kim, D. (2020, April). Marionette: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 07, pp. 10893-10900).

[6] Deepfake and real images dataset from deepfake and real images (kaggle.com)

[7] De Lima, O., Franklin, S., Basu, S., Karwoski, B., & George, A. (2020). Deepfake detection using spatiotemporal convolutional networks. *arXiv preprint arXiv:2006.14749*.

[8] Korshunov, P., & Marcel, S. (2020). Deepfake detection: humans vs. machines. *arXiv preprint arXiv:2009.03155*.

[9] Malik, A., Kuribayashi, M., Abdullahi, S. M., & Khan, A. N. (2022). DeepFake detection for human face images and videos: A survey. *Ieee Access*, *10*, 18757-18775.

[10] Al-Adwan, A., Alazzam, H., Al-Anbaki, N., & Alduweib, E. (2024). Detection of Deepfake Media Using a Hybrid CNN–RNN Model and Particle Swarm Optimization (PSO) Algorithm. *Computers*, *13*(4), 99.

[11] Kumar, M., & Sharma, H. K. (2023). A GAN-based model of deepfake detection in social media. *Procedia Computer Science*, *218*, 2153-2162.

[12] Lanzino, R., Fontana, F., Diko, A., Marini, M. R., & Cinque, L. (2024). Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3771-3780).

[13] Ju, Y., Hu, S., Jia, S., Chen, G. H., & Lyu, S. (2024). Improving fairness in deepfake detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 4655-4665).

[14] Guarnera, L., Giudice, O., & Battiato, S. (2020). Deepfake detection by analyzing convolutional traces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 666-667).

[15] Ba, Z., Liu, Q., Liu, Z., Wu, S., Lin, F., Lu, L., & Ren, K. (2024, March). Exposing the deception: Uncovering more forgery clues for deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 2, pp. 719-728).

[16] Rana, M. S., Nobi, M. N., Murali, B., & Sung, A. H. (2022). Deepfake detection: A systematic literature review. *IEEE access*, *10*, 25494-25513.

[17] Nirkin, Y., Wolf, L., Keller, Y., & Hassner, T. (2021). Deepfake detection based on discrepancies between faces and their context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(10), 6111-6121.

[18] Le, T. N., Nguyen, H. H., Yamagishi, J., & Echizen, I. (2021). Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10117-10127).