

Deep fake detection in face images and video by tuning a pretrained CLIP model

Maomao Ling

Warsaw University of Technology
Faculty of Electronics and Information Technology
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland
Email: maomao.ling.stud@pw.edu.pl

Włodzimierz Kasprzak

0000-0002-4840-8860
Warsaw University of Technology
Institute of Control and Computation Eng.
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland
Email: wlodzimierz.kasprzak@pw.edu.pl

Abstract—This study develops a video deepfake detection system that addresses the critical challenge of cross-dataset generalization in real-world scenarios. It adopts CLIP (Contrastive Language-Image Pre-training) as the foundation, leveraging its strong vision-language representations for detecting subtle facial manipulations. A parameter-efficient fine-tuning approach is proposed that achieves superior cross-dataset performance while updating only a minimal fraction of model parameters. The CLIP ViT-B model, utilizing the LayerNorm tuning, achieves an average video-level AUC of 90.9% across seven benchmark datasets, while training only 0.0344% of the total parameters. This approach is also extended to CLIP-STAN, a temporal video detector, achieving competitive results with 9.46% trainable parameters. The work integrates CLIP with LayerNorm tuning into the Deepfake Bench framework, enabling comprehensive cross-dataset generalization analysis. Notably, the implemented solution - base CLIP ViT-B with LayerNorm tuning - matches more complex CLIP variants in generalization performance, underscoring the significance of high-quality pre-training over architectural complexity.

I. INTRODUCTION

DEEPFAKE means an AI-generated content technique that manipulates the truth and misleads the public [1]. It is attracting widespread attention from academia and industry, especially with the rapid development of foundation models such as Stable Diffusion [2], DALL-E [3], and StyleCLIP [4], which have made synthetic data more diverse and realistic in appearance.

In this work, our interest is in face image/video deepfake detection. Such deepfakes can be categorized based on the type and extent of facial modification. The most common form of such deepfakes is *face swapping* involving the complete replacement of one individual's face with another's while preserving facial expressions and head movements. This technique is prevalent in datasets like DFDC [5] and Celeb-DF-v2 [6]. *Face Reenactment* means transferring facial expressions, head poses, or eye movements from a driving video to a target identity (e.g., [7]). The more recent techniques of *lip-sync* synchronize lip movements in video with unrelated audio content, creating the illusion that a person spoke words they never actually said [8], [9]. Modifying specific facial attributes such as age, gender, expression, or hairstyle while maintaining overall identity are called *attribute editing* techniques. These

manipulations are often more subtle and harder to detect. Finally, *synthetic faces* means creating entirely artificial facial identities that do not correspond to any real person. While not technically "fake" versions of real people, they pose challenges to identity verification systems.

The aim of this work is to design and evaluate an face image deepfake detection model. But understanding these manipulation categories and deepfake generation techniques provides the foundation for developing effective detection strategies.

Generative adversarial nets (GANs) [10] form the backbone of most deepfake generation systems (in particular - Progressive GANs, StyleGAN). The adversarial training framework, consisting of a generator and discriminator network, enables the creation of highly realistic synthetic faces. The GAN training process can be formulated as a minimax game between the generator and discriminator. Recent advances in diffusion models [11] have shown remarkable capabilities in generating high-fidelity images. These models work by learning to reverse a noise corruption process, enabling more stable training and potentially higher quality outputs compared to GANs. Diffusion models operate through a forward process that gradually adds noise to data and a reverse process that learns to denoise. Autoencoder-based approaches [12] learn compressed representations of facial features and can reconstruct or swap faces by manipulating the latent space. In particular, Variational Autoencoders (VAEs) and their variants provide a probabilistic framework for face generation and manipulation, often offering more controllable generation processes.

The recent trend in deepfake generation and detection are cross-modal systems, combining image-and-text [13] or image-and-LLM [14] or audio-and-video [15]. CLIP [16], as the first large-scale visual-language pretraining model, provides basic support for universal visual-text feature extraction network to complete various cross-modal tasks, like deepfake generation. It is the baseline for recent image-and-text or image-and-LLM face deepfake generation techniques. But DeepFake generation and detection is a continuous adversarial game. To achieve reliable DeepFake detection, it is crucial to adapt and refine defense strategies from diverse perspectives. This observation has motivated us to pursue a CLIP-based

deepfake detection approach.

The main contributions of our work are:

- Complete and comprehensive CLIP-based deepfake detection system spanning the entire pipeline from data preprocessing through model training to cross-dataset evaluation, demonstrating practical applicability and engineering completeness.
- Extensive experiments show that the proposed solution is consistently outperforming non-CLIP state-of-the-art methods and matching or surpassing more complex CLIP-based designs.
- Standardized multi-dataset evaluation conducted on seven major deepfake datasets providing consistent and comparable benchmarks for cross-dataset generalization.
- Design insights for cross-dataset detection: The findings highlight that in deep fake cross-domain detection, high-quality pre-training can outweigh architectural complexity.

Section 2 presents the solution design, detailing the network architectures for the CLIP image detector and CLIP-STAN video detector, along with the proposed adaptation strategies. In section 3 details of data preparation, model training, testing procedures, and evaluation metrics are provided. In section 4 comprehensive experimental results are provided, demonstrating superior cross-dataset performance. In section 5 key findings are highlighted.

II. SOLUTION

A deepfake face detection approach was developed based on CLIP (Contrastive Language-Image Pre-training) [17] and integrated into DeepfakeBench [18].

A. Preprocessing

The DeepfakeBench platform [18] provides a unified framework that standardizes preprocessing pipelines across different datasets, ensuring consistent input quality and enabling fair cross-dataset evaluation. Our preprocessing strategy strictly follows DeepfakeBench's established protocols, incorporating face detection, alignment, and standardized sampling strategies that have been validated across multiple benchmark datasets.

B. The CLIP model

The CLIP model structure is presented in Fig. 1. CLIP employs a dual-encoder architecture consisting of a vision encoder and a text encoder, trained through contrastive learning on 400 million image-text pairs. The vision encoder processes visual inputs while the text encoder handles textual descriptions, with both encoders mapping to a shared 768-dimensional embedding space. This shared representation space enables the model to understand the semantic relationships between visual content and textual descriptions, which is particularly valuable for detecting manipulations that may violate natural semantic consistency. A CLIP image detector performs both image-level and video-level detection through frame-level analysis.

The CLIP image detector utilizes the pre-trained CLIP ViT-B/16 backbone, which has been trained on large-scale

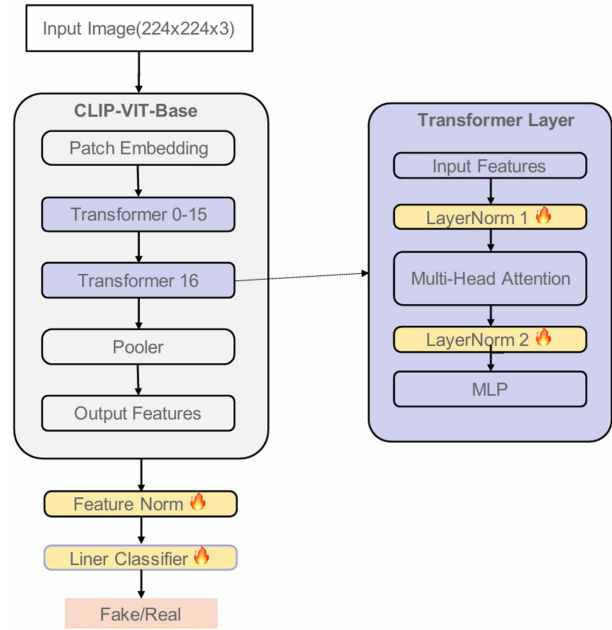


Fig. 1. CLIP Image Detector Architecture: The CLIP ViT-B/16 backbone processes input images through patch embedding and transformer layers. The LayerNorm layers and the classification head are trainable (highlighted in light yellow) during fine-tuning, enabling parameter-efficient adaptation.

image-text pairs, providing rich semantic priors beneficial for detecting subtle facial manipulations.

The Patch Embedding step divides input images of size 224×224 pixels into 16×16 pixel patches, resulting in 196 patches per image (a 14×14 grid). Each patch is linearly projected to a 768-dimensional embedding vector using a learnable projection matrix, adding positional embeddings to preserve spatial information.

The transformer's Encoder consists of 12 transformer blocks. All LayerNorm layers within the backbone blocks will be unfrozen and fine-tuned to adapt the feature normalization specifically to the deepfake domain. After the final transformer layer, the patch embeddings are averaged across the spatial dimension to produce a single 768-dimensional feature vector representing the entire image.

The classifier is implemented as a single fully connected layer that maps the 768-dimensional CLIP embeddings to binary logits, preceded by feature normalization to stabilize training. This design choice maintains simplicity while leveraging the rich semantic representations learned by the CLIP backbone. The binary output indicates whether the input image is authentic (0) or manipulated (1).

C. Parameter-Efficient Fine-Tuning

As models grow larger, parameter-efficient fine-tuning has become crucial for practical deployment and generalization. Several techniques have been proposed, like "Adapter Layers", "Low-Rank Adaptation (LoRA)", "Prefix Tuning", "BitFit" or "LayerNorm Tuning" [17]. LoRA is eventually the most popular one - it decomposes weight updates into low-rank matrices,

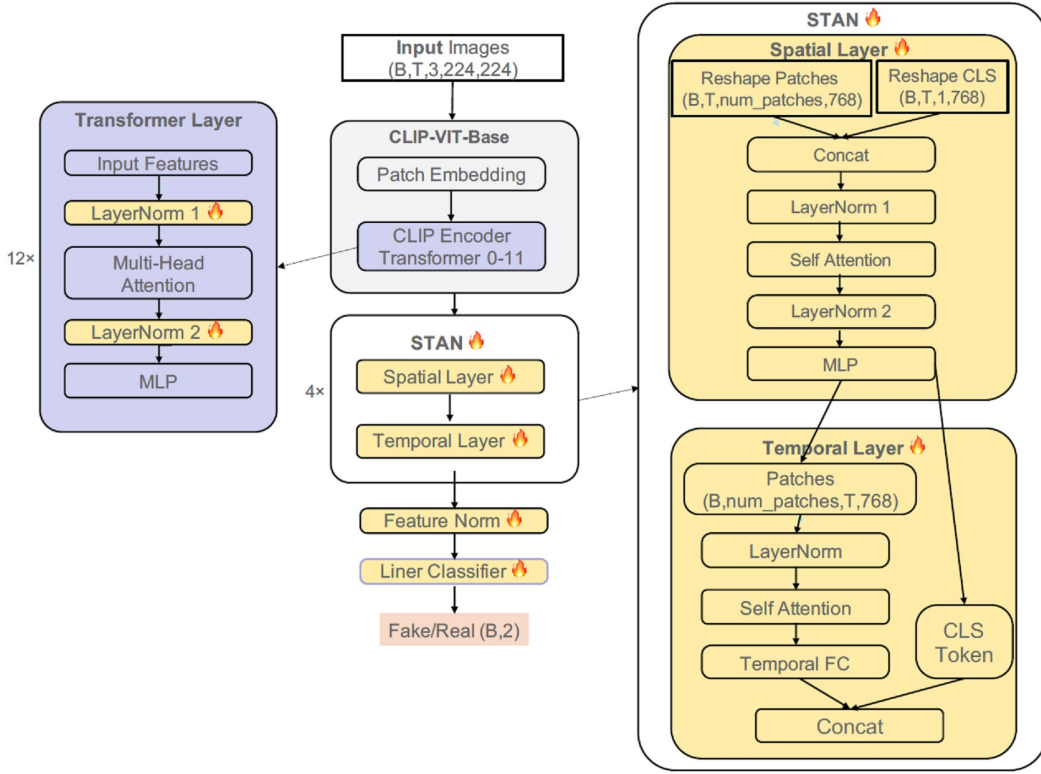


Fig. 2. CLIP-STAN architecture. It extends CLIP with Spatial-Temporal Attention Networks (STAN). Trainable components (highlighted in light yellow) include LayerNorm layers in CLIP, Spatial Attention modules in STAN, entire Temporal Attention blocks, and the classification head.

dramatically reducing trainable parameters while maintaining performance. In our work the LayerNormTuning was chosen. It specifically updates normalization layers while freezing other parameters. Prior work [17] showed that normalization layers are highly effective for adapting vision transformers to new domains with minimal computational overhead, while also enabling the model to better adjust to dataset-specific feature distributions.

D. Temporal Extension

An extension of the vision-language model CLIP to temporal domains is CLIP-STAN [21]. Its architecture is shown in Figure 2. We adopted this architecture to the deepfake detection task - to the author's knowledge, this is the first application of CLIP-STAN to such task. Thus, an efficient training strategy was implemented: the LayerNorm layers in CLIP, the Spatial Attention modules in STAN, the entire Temporal Attention blocks in STAN, and the final classifier were trained. A comprehensive study comparing different numbers of STAN layers to optimize the architecture for deepfake detection was also conducted.

III. IMPLEMENTATION

Figure 3 illustrates the overall implementation pipeline of the deepfake detection system that includes data preparation, model training, testing procedures, and evaluation metrics.

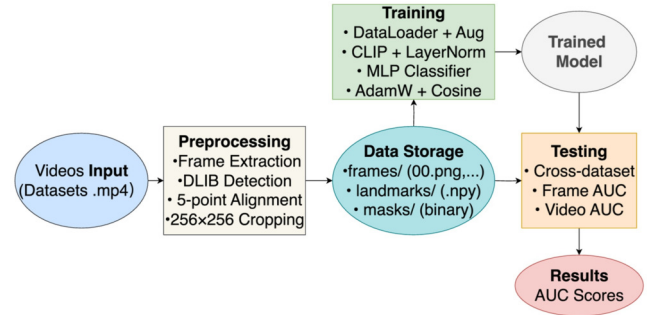


Fig. 3. The system's implementation pipeline

The implementation has utilized the following software stack:

- Deep Learning Framework: PyTorch 2.6.0 [19] with CUDA 12.4 [20] support
- Computer Vision: torchvision 0.15.1 [21], opencv-python 4.11.0 [22]
- DeepfakeBench Framework: DeepfakeBench [18] integrated for standardized deep fake detection evaluation
- Transformer Models: transformers 4.50.0 [23] for CLIP model implementation
- Additional Libraries: dlib 19.24.9 [24] for face detection and alignment, tensorboard 2.19.0 [25] for training

monitoring and visualization, scikit-learn 1.6.1 [26] for evaluation metric computation.

A. Preprocessing

The preprocessing pipeline implements DLIB-based face detection and alignment:

- Detection: DLIB's HOG-based face detector identifies face regions;
- Landmark Localization: 5 key facial landmarks (eyes, nose, mouth) for alignment using shape predictor model;
- Alignment: Affine transformation aligns faces to canonical pose using 5 key points (eyes, nose, mouth);
- Standardization: All faces are resized to 256×256 pixels with 30% margin padding.

The implementation follows DeepfakeBench's standardized frame sampling approach:

- Default Mode: Fixed number of frames (32 frames per video);
- Alternative Mode: Fixed stride sampling (every 10th frame);
- Sampling Logic: Linear interpolation for even distribution across video duration;
- Frame Selection: Automatically selects optimal frames based on video length.

The evaluation metric is Area Under the ROC Curve (AUC), which measures the model's ability to distinguish between real and fake videos. The ROC curve is: $TPR = f(FPR^{-1})$ where TPR - True Positive Rate, FPR - False Positive Rate. The evaluation focuses on both frame-level and video-level AUC computation. Video-level AUC is calculated by first averaging frame-level predictions for each video, then computing AUC based on these video-level aggregated scores. Mean aggregation was adopted because it is simple, robust to noisy frames, and showed lower variance than max or median pooling in preliminary experiments conducted for this study. AUC was chosen as the evaluation metric because it is threshold-free and robust under class imbalance, ensuring fair comparison across datasets with varying positive rates.

B. Pretrained model tuning

The implementation focuses on parameter-efficient adaptation of CLIP models: CLIP ViT-B: pre-trained CLIP ViT-B/16 backbone with LayerNorm tuning (primary approach), CLIP ViT-L: pre-trained CLIP ViT-L/14 backbone with LayerNorm tuning (best performance), and CLIP-STAN: CLIP ViT-B extended with spatial-temporal attention networks (limited benefit).

The core of LayerNorm tuning is parameter-efficient fine-tuning through LayerNorm adaptation. The following implementation strategy was employed:

- Frozen components: CLIP backbone weights, embedding layers, attention weights
- Trainable components: all LayerNorm layers, classification head
- Parameter efficiency: only 0.0344% of total parameters updated

TABLE I
DEEPPAKE VIDEO DATASETS: SUMMARY OF ANNOTATIONS AND MANIPULATION TECHNIQUES

Dataset	Real Videos	Fake videos	Annotations
FaceForensics++	1,000	3,000	Method labels
DFDC	19,154	86,568	Real/Fake labels
DFDCP	1,131	4,119	Video-level labels
Celeb-DF-v2	6,277	5,639	Real/Fake labels
UADFV	49	49	Real/Fake labels
FaceShifter	1,000+	1,000+	Real/Fake labels
DeepFakeDetection	363	3,068	Real/Fake labels

Dataset	Manipulation Techniques
FaceForensics++	DeepFakes, Face2Face, FaceSwap, NeuralTextures
DFDC	Multiple models: GANs, Autoencoders, FaceSwap
DFDCP	Face Swapping (unknown methods)
Celeb-DF-v2	Autoencoder-based DeepFakes
UADFV	DeepFakes
FaceShifter	GAN-based face swapping
DeepFakeDetection	Deepfake face swapping with minimal artifacts

C. Classification head

The classification head supports both single-layer and multi-layer configurations. The hidden dimension of 512 in the two-layer variant follows standard lightweight transformer head design, providing a balance between capacity and efficiency. Comparing one-layer and two-layer variants also allows assessment of whether added non-linearity improves generalization. This choice was made to evaluate whether modest added non-linearity improves cross-dataset generalization while keeping the head lightweight for efficient deployment.

IV. RESULTS

A. Datasets

For evaluation we utilize following datasets (TABLE I):

- Training dataset: FaceForensics++ (FF++) [27]
- Testing datasets: DFDC [5], DFDCP [28], Celeb-DF-v2 [6], UADFV [29], FaceShifter [30], DFD [31]

B. Experiments

There have been conducted two series of experiments:

- 1) Individual Dataset Results: AUC scores computed independently for each test dataset. Frame vs. Video Level: Both frame-level and video-level AUC were computed
- 2) Cross-Dataset Generalization: Performance evaluated across diverse manipulation types and data domains.

The CLIP ViT-B model demonstrates superior performance across multiple datasets, and importantly, it performs competitively when compared to the recent UDD approach [32]. UDD is also built upon a CLIP backbone but integrates a more complex architecture with additional detection-specific modules and training objectives to enhance local artifact modeling. Despite this increased architectural complexity, the proposed approach - which relies on parameter-efficient LayerNorm tuning and a lightweight classification head - achieves competitive

TABLE II

FRAME-LEVEL AUC(%) OF MODELS TRAINED ON FF++ (C23). BOLD AND UNDERLINE INDICATE THE BEST AND THE SECOND-BEST RESULTS. BASELINE RESULTS ARE REPORTED FROM THE UDD PAPER [32].

Method	CDF	DFDCP	DFDC	DFD
Xception	73.7	73.7	70.8	81.6
Efficient-b4	74.9	72.8	69.6	81.5
Face X-ray	67.9	69.4	63.3	76.7
RECCE	73.2	74.2	71.3	81.2
LSDA	83.0	81.5	73.6	88.0
ViT-B CLIP	81.7	80.2	73.5	86.6
UDD	86.9	85.6	<u>75.8</u>	<u>91.0</u>
Proposed CLIP ViT-B	81.8	<u>82.0</u>	80.4	91.8

TABLE III

VIDEO-LEVEL AUC(%) OF MODELS TRAINED ON FF++(C23) COMPARED TO PREVIOUS SOTA METHODS. BOLD AND UNDERLINE INDICATE THE BEST AND THE SECOND-BEST RESULTS. BASELINE RESULTS ARE REPORTED FROM THE UDD PAPER [32].

Method	CDF	DFDCP	DFDC	DFD
Xception	81.6	74.2	73.2	89.6
Efficient-b4	80.8	68.0	72.4	86.2
LipForensics	82.4	-	73.5	-
FTCN	86.9	74.0	71.0	94.4
RECCE	82.3	73.4	69.6	89.1
SBI	90.6	<u>87.7</u>	75.2	88.2
PCL+I2G	90.0	74.4	67.5	-
TALL++	<u>92.0</u>	-	78.5	-
ViT-B CLIP	88.4	82.5	76.1	90.0
UDD	93.1	88.1	<u>81.2</u>	<u>95.5</u>
Proposed CLIP ViT-B	89.0	86.6	83.6	96.2

results to UDD on several benchmarks. For instance, in frame-level detection, the proposed model reaches 91.8% AUC on DFD (best result) and 80.4% AUC on DFDC (best result), while UDD reports 91.0% and 75.8%, respectively (TABLE II).

In video-level detection (TABLE III), the model achieves 96.2% AUC on DFD and 83.6% AUC on DFDC, compared to UDD's 95.5% and 81.2%. This comparison underscores that a carefully chosen pretrained backbone combined with minimal yet effective adaptation can rival more elaborate designs, emphasizing the importance of efficient tuning strategies for cross-dataset generalization.

The ablation study in TABLE IV reveals several important trends that clarify the contributions of different design choices. First, regarding the MLP head depth, the CLIP ViT-B model with LayerNorm tuning and a single-layer MLP achieves the best balance between performance and parameter efficiency, reaching an average video-level AUC of 90.9% across seven datasets while updating only 0.0344% of the model parameters. Increasing the MLP depth to two layers does not consistently improve performance, while it boosts results on some datasets (e.g., DFDCP), the average AUC remains similar (91.0%), suggesting diminishing returns and

TABLE IV

VIDEO-LEVEL AUC COMPARISON ACROSS 7 DEEFAKE DATASETS. ALL MODELS ARE TRAINED ON FACEFORENSICS++ (FF++). THE PROPOSED CLIP ViT-L (LAYERNORM TUNING ONLY) OUTPERFORMS OTHER BASELINES ON MOST DATASETS.

Model	ViT-B (LN +1-L MLP)	ViT-B (LN +2-L MLP)	ViT-L (LN +1-L MLP)	STAN (LN +4 × T A)
Test dataset				
DFDC	83.6	82.2	88.6	85.0
DFDCP	86.6	94.9	89.2	85.0
DFD	96.2	95.9	98.2	98.1
FS	74.4	84.4	87.7	79.0
FF++	99.6	97.3	99.8	96.7
UADFV	98.4	99.5	99.4	99.0
CDF-v2	89.0	81.7	95.0	83.8

potential overfitting with additional parameters. Finally, the CLIP ViT-L backbone with LayerNorm tuning delivers the strongest overall results, achieving top performance on 5 out of 7 datasets, although at higher computational cost. CLIP-STAN (LN+4 × T A) attains its highest score (98.1 %) for DFD, but it underperforms on DFDC and other challenging cross-domain datasets.

V. CONCLUSIONS

This work has developed a complete deepfake video detection system based on efficient and lightweight adaptation of the CLIP model for generalizable deepfake video face detection. The following key findings were obtained:

- By adopting a LayerNorm-tuning strategy, such as ViT-B (CLIP, 84.3%) and EfficientNet-B4 (76.85%), while updating only 0.0344% of the model parameters. This demonstrates the remarkable achievement of parameter-efficient adaptation on pretrained vision-language models was demonstrated. The CLIP ViT-B model achieves an average video-level AUC of 90.9% across seven datasets, outperforming previous methods.
- While CLIP-STAN achieved competitive results on DFD, its performance was not consistently superior to CLIP-only models in cross-dataset generalization.
- Compared to full fine-tuning (which failed to converge), the results show that minimal adaptation with careful parameter selection is not only sufficient but preferable for transfer learning in the context of deepfake detection.
- The integration of CLIP into DeepfakeBench provides a standardized platform for benchmarking vision-language models in this domain. This work thus contributes both empirical findings and reproducible baselines that can guide future research.

REFERENCES

- [1] Shichuang Xie, Tong Qiao, Sheng Li, Xinpeng Zhanga, Jiantao Zhou, Guorui Feng, "DeepFake detection in the AIGC era: A survey, benchmarks, and future perspectives". *Information Fusion*, 127 (2026) 103740, <https://doi.org/10.1016/j.inffus.2025.103740>
- [2] -: "Stable Diffusion". <https://stability.ai/>
- [3] -: "DALL-E". <https://openai.com/index/dall-e/>

- [4] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, D. Lischinski, "Style-clip: textdriven manipulation of stylegan imagery", in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2065–2074 <https://doi.org/10.1109/ICCV48922.2021.00209>
- [5] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, C. Canton Ferrer, "The deepfake detection challenge (dfdc) dataset", arXiv: 2006.07397, 2020. <https://doi.org/10.48550/arXiv.2006.07397>
- [6] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for deepfake forensics", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3207–3216 <https://doi.org/10.48550/arXiv.1909.12962>
- [7] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2387–2395 <https://doi.org/10.1109/CVPR.2016.262>
- [8] W. Zhong, C. Fang, Y. Cai, P. Wei, G. Zhao, L. Lin, G. Li, "Identity-preserving talking face generation with landmark and appearance priors", in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9729–9738. <https://doi.org/10.48550/arXiv.2305.08293>
- [9] C. Zhang, C. Wang, Y. Zhao, S. Cheng, L. Luo, X. Guo, "Dr2: disentangled recurrent representation learning for data-efficient speech video synthesis", in: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 6192–6202. <https://doi.org/10.1109/WACV57701.2024.00609>
- [10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative adversarial nets", in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, 2014, pp. 2672–2680. https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf
- [11] J. Ho, A. Jain, and P. Abbeel, "Denosing diffusion probabilistic models", in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6840–6851. <https://doi.org/10.48550/arXiv.2006.11239>
- [12] D.P. Kingma and M. Welling, "Auto-encoding variational bayes", in *International Conference on Learning Representations (ICLR)*, 2014. <https://doi.org/10.48550/arXiv.1312.6114>
- [13] R. Liu, B. Ma, W. Zhang, Z. Hu, C. Fan, T. Lv, Y. Ding, X. Cheng, "Towards a simultaneous and granular identity-expression control in personalized face generation", in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2114–2123. <https://doi.org/10.1109/CVPR52733.2024.00206>
- [14] S. Xu, G. Chen, Y.-X. Guo, J. Yang, C. Li, Z. Zang, Y. Zhang, X. Tong, B. Guo, *Vasa-1: lifelike audio-driven talking faces generated in real time*, arXiv: 2404.10667 (2024) <https://doi.org/10.48550/arXiv.2404.10667>
- [15] J. Sun, Q. Deng, Q. Li, M. Sun, Y. Liu, Z. Sun, "Anyface++: a unified framework for free-style text-to-face synthesis and manipulation", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 11, pp. 9438–9453, Nov. 2025, <https://doi.org/10.1109/TPAMI.2023.3345866>
- [16] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., "Learning transferable visual models from natural language supervision", in: *International Conference on Machine Learning*, PMLR, vol. 139, 2021, pp. 8748–8763. <https://proceedings.mlr.press/v139/radford21a.html>
- [17] A. Yermakov, J. Cech, and J. Matas, "Unlocking the hidden potential of clip in generalizable deepfake detection", arXiv: 2503.19683, 2025 <https://doi.org/10.48550/arXiv.2503.19683>
- [18] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu, "Deepfakebench: A comprehensive benchmark of deepfake detection", in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 4534–4565. <https://doi.org/10.48550/arXiv.2307.01426>
- [19] A. Paszke, S. Gross, F. Massa, et al., "Pytorch: An imperative style, high-performance deep learning library", in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035. <https://doi.org/10.48550/arXiv.1912.01703>
- [20] NVIDIA Corporation, "Cuda toolkit documentation", <https://docs.nvidia.com/cuda/>
- [21] TorchVision maintainers and contributors, "Torchvision: Pytorch's computer vision library", <https://github.com/pytorch/vision>
- [22] – "OpenCV", <https://opencv.org/releases/>
- [23] – "Transformer 4.5.0", <https://huggingface.co/transformers/v4.5.0/index.html>
- [24] D. E. King, "Dlib-ml: A machine learning toolkit", *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009. <https://www.dlib.net/python/>
- [25] TensorFlow Team, "Tensorboard - Tensorflow's visualization toolkit". <https://www.tensorflow.org/tensorboard>
- [26] – "Scikit-learn". <https://scikit-learn.org/stable/index.html>
- [27] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1–11. <https://doi.org/10.1109/ICCV.2019.00009>
- [28] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (dfdc) preview dataset", arXiv: 1910.08854, 2019. <https://arxiv.org/pdf/1910.08854>
- [29] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts", in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 2018, pp. 46–52. <https://doi.org/10.48550/arXiv.1811.00656>
- [30] L. Li, J. Bao, H. Zhang, et al., "Faceshifter: Towards high fidelity and occlusion aware face swapping", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7390–7398. <https://doi.org/10.48550/arXiv.1912.13457>
- [31] Nick Dufour and Andrew Gully, "Contributing data to deepfake detection research", [Online] 2019. <https://research.google/blog/contributing-data-to-deepfake-detection-research/>
- [32] X. Fu, Z. Yan, T. Yao, S. Chen, and X. Li, "Exploring unbiased deepfake detection via token-level shuffling and mixing", arXiv: 2501.04376, 2025. <https://doi.org/10.48550/arXiv.2501.04376>