Detecting Deepfakes in Video Conferencing Scenarios: A Study on Domain-Specific Challenges and Model Adaptations

Brendan Murphy
CS229 Autumn 2024
bigsur@stanford.edu

1. Introduction

The rise of video conferencing platforms has created new vulnerabilities to deepfake attacks, particularly through advances in real-time face-swapping technologies like DeepLiveCam (DLC).[1] Traditional deepfake detection models, while effective in controlled environments, often struggle with the unique challenges presented by video conferencing platforms, including artifacts introduced by compression, variations in lighting, and inconsistent video quality.

The input to my algorithm is a sequence of video frames from video conferencing platforms, processed in 32-frame clips at 224x224 pixel resolution. Using the AltFreezing model, which combines a 3D ResNet-50 backbone with a transformer head, my system outputs a binary classification indicating whether the video contains deepfake manipulation.[2] The model specifically targets artifacts and inconsistencies introduced by the DLC face-swapping method, which represents a significant threat in video conferencing scenarios.

I address the challenge of detecting deepfake manipulations in video conferencing by conducting four distinct training experiments: (1) training with gradual unfreezing on the FaceForensics++ (FF++) DLC dataset, (2) training with gradual unfreezing on the Zoom DLC dataset, (3) training with alternate freezing on the FF++ DLC dataset, and (4) training with alternate freezing on the Zoom DLC dataset.[3] I created two datasets: first by processing the FF++ dataset using the DLC face-swapping method to introduce realistic impersonation artifacts, and second by curating and manipulating Zoom videos to create a domain-specific dataset that incorporates challenges such as compression artifacts, variable lighting, and resolution inconsistencies. Each of these four training configurations was subsequently evaluated on a hold-out Zoom dataset. This approach allowed for a comparative analysis of the training strategies and datasets, demonstrating the trade-offs between optimizing for high-quality, general-purpose samples (FF++) and low-quality, domain-specific samples (Zoom).

The evaluation is performed on a hold out set of the Zoom videos with DLC manipulations, representing the actual target domain. I also employ the Celeb-DF dataset as a benchmark to detect whether the model maintains general deepfake detection capabilities on high resolution video after the fine-tuning process.[4]

This work advances the field of deepfake detection through several key contributions. First, I explore two fine-tuning methods that adapt deepfake detection systems to video conferencing environments, bridging the gap between controlled testing conditions and real-world applications. Second, I construct two custom datasets: the FF++ DLC Dataset, focusing on cross-subject impersonation artifacts, and the Zoom DLC Dataset, targeting domain-specific challenges inherent to video conferencing. These datasets provide valuable resources for training and evaluating detection systems in realistic scenarios. Third, I refine the evaluation framework by incorporating both benchmark data (e.g., Celeb-DF) and real-world video conferencing data, enabling a comprehensive assessment of the model's generalization capabilities and domain-specific robustness.

The success of this approach depends on effectively bridging multiple domain gaps: from high-quality deepfake detection to degraded video conferencing conditions, and from traditional deepfake methods to DLC-specific artifacts. By leveraging multiple datasets and fine-tuning methods, this work demonstrates a pathway toward robust and practical deepfake detection in real-world video conferencing scenarios.

2. Related Work

Deepfake detection research spans spatial, temporal, and domain adaptation approaches. Each category contributes unique strengths but also has limitations in addressing domain-specific challenges like video conferencing.

Spatial methods focus on detecting inconsistencies within individual frames. Rossler et al. introduced XceptionNet, a convolutional neural network adapted from image classification, which identifies pixel-level artifacts and has demonstrated strong performance on benchmark datasets such as FF++ [3]. While XceptionNet serves as a robust baseline, its reliance on frame-level features limits its ability to capture temporal dependencies, reducing its effectiveness in video scenarios.

Temporal methods expand upon spatial approaches by modeling inconsistencies across consecutive frames. The Spatiotemporal Inconsistency Learning (STIL) method captures temporal artifacts along orthogonal directions, leveraging inter-frame relationships for robust detection of manipulated videos [5]. STIL demonstrates strong generalization by focusing on dynamic inconsistencies; however, its computational complexity can hinder deployment in real-time or domain-specific applications.

Domain adaptation addresses the challenge of transferring models trained on controlled datasets to real-world settings. Cozzolino et al. proposed forensic transfer learning, using domain-specific augmentations to adapt detection models to new environments [6]. This approach highlights the importance of dataset tailoring for improved generalization. Despite its effectiveness, limited attention has been given to domain-specific scenarios like video conferencing, where compression artifacts and low frame rates pose additional challenges.

Building on these foundations, Wang et al. introduced AltFreezing, a training strategy that alternately freezes spatial and temporal network weights during training [2]. This approach forces the model to learn both types of artifacts rather than relying solely on easier-to-detect spatial features, enabling better generalization to various types of deepfake manipulations.

My approach integrates spatial and temporal features using a modified AltFreezing architecture, applying it to datasets tailored for video conferencing scenarios. Unlike XceptionNet, which focuses more on spatial features, my method combines temporal and spatial to target dynamic inconsistencies in manipulated videos. Additionally, inspired by Cozzolino et al., I use fine-tuning strategies to adapt models to compression and lighting challenges unique to Zoom-like conditions. This comprehensive approach addresses gaps in prior work, emphasizing the trade-offs between domain-specific performance and generalizability.

3. Dataset and Features

This work utilizes three distinct datasets, carefully selected and processed to ensure robust training and evaluation of deepfake detection tailored to video conferencing environments. Each dataset serves a specific role in the training or evaluation pipeline, as described below.

3.1. FF++ DLC

The FF++ DLC dataset was constructed by modifying the FF++ dataset using the DLC face-swapping method. This method swaps faces between individuals, introducing realistic impersonation artifacts while preserving consistent lighting, pose, and framing conditions.

The dataset comprises 120 videos, split into 100 training videos (50 real, 50 swapped) and 20 validation videos (10 real, 10 swapped). These were further segmented into approximately 4,500 clips, with each clip containing 32 frames at a resolution of 224x224 pixels. Preprocessing steps included:

- **Face Detection and Alignment:** Frames were processed using FasterCropAlignXRay to detect facial landmarks, crop faces with a scale factor of 0.5, and ensure consistent alignment across frames.
- **Normalization:** Pixel values were normalized using ImageNet statistics (mean: [0.485, 0.456, 0.406], std: [0.229, 0.224, 0.225]).[7]
- **Feature Extraction:** Inputs were structured as 3D tensors capturing spatiotemporal features from aligned clips. These features emphasize temporal inconsistencies across frames caused by face-swapping manipulations.

This dataset highlights identity mismatches and subtle inconsistencies introduced during face swapping, making it valuable for general-purpose deepfake detection.

3.2. Zoom DLC

The Zoom DLC dataset was curated to address the specific challenges posed by video conferencing environments, including natural variations in lighting, resolution, and compression artifacts. It represents real-world scenarios where deepfake manipulations may occur during live video conferencing.

The dataset contains 114 videos, split into 48 training videos (24 real, 24 fake) and 66 evaluation videos (36 real, 30 fake). These were further segmented into approximately 4,300 clips, ensuring sufficient data for training and evaluation. Videos were manipulated using the DLC method to simulate impersonation attempts under realistic Zoom conditions.

Preprocessing was identical to that of the FF++ dataset, including face detection, alignment, and normalization using ImageNet statistics, ensuring consistent input formatting. This dataset specifically targets domain-specific artifacts such as compression-induced distortions (e.g., noise and pixelation introduced by Zoom's compression algorithms), resolution inconsistencies (e.g., variations in video quality due to bandwidth limitations), and dynamic lighting variations across frames. By capturing these real-world challenges, the Zoom dataset enhances the model's ability to generalize to manipulations encountered in video conferencing applications.

3.3. Celeb-DF

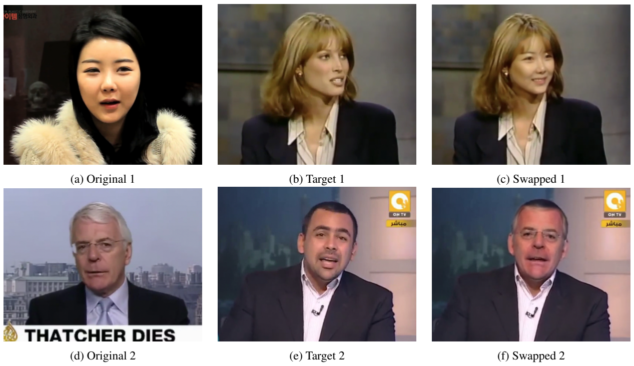
The Celeb-DF dataset serves as an external benchmark to evaluate the model's generalization to unseen deepfake content and to assess whether it retains its original detection capabilities after fine-tuning. This widely used dataset consists of 590 real videos and 5,639 high-quality synthetic videos. For this study, a random subset of 100 videos (50 real, 50 fake) was used for evaluation. These videos were segmented into approximately 3,200 clips.

Unlike the FF++ and Zoom datasets, Celeb-DF videos were not modified or augmented to simulate video conferencing conditions. This ensures that evaluation focuses solely on the model's ability to detect high-quality synthetic manipulations in out-of-distribution data.

3.4. Dataset Overview and Features

Dataset	# Videos	# Clips	Target Artifacts
FF++ DLC	120	∼4,500	Identity mismatches
Zoom DLC	114	∼4,300	Compression, lighting, resolution
Celeb-DF	100	∼3,200	High-quality manipulations

Table 1: Summary of datasets, including the number of videos and corresponding 32-frame clips. This dual representation offers a comprehensive view of the data quantity used for training and evaluation..



(a) Original 1 (b) Target 1 (c) Swapped 1

(d) Original 2 (e) Target 2 (f) Swapped 2

Figure 1: Results of the DLC face-swapping method on the FF+ dataset. Each row illustrating the original face, the target video frame, and the resulting manipulation.

4. Methods

The project evaluates two training strategies: gradual unfreezing of layers and the alternate freezing methodology, applying each approach to the pre-trained AltFreezing model to compare their effectiveness in domain-specific deepfake detection.[2] These strategies allow the model to progressively adapt to domain-specific tasks while balancing the optimization of spatial and temporal features. Given an input video sequence $X = \{x_1, ..., x_T\}$ where each $x_t \in \mathbb{R}^{H\times W\times 3}$ represents a frame, my goal is to learn a function $f_\theta : X \rightarrow \{0, 1\}$ that classifies the sequence as real or manipulated.

4.1. Model Architecture and Fine-tuning

The network architecture consists of a pretrained 3D ResNet-50 backbone $\phi(X)$ that extracts spatio-temporal features and a trainable classification head $h()$ that produces the final prediction. The model output is computed as:

$$f_\theta(X) = \sigma(h(\phi(X))), \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid activation function. Fine-tuning is implemented using two distinct strategies:

- **Gradual Unfreezing:** Layers of the backbone are unfrozen incrementally during training, allowing the model to adapt to the new domain while maintaining the stability of pre-trained features. Let $\mathcal{U}_k$ denote the set of trainable layers at epoch k, then:

  $$\theta = \{\theta_h\} \cup \{\theta_{\phi_l} \mid l \in \mathcal{U}_k\}, \tag{2}$$

  where θ_{ϕ_l} represents the parameters of the l-th layer of the backbone.

- **Alt-Freezing:** This strategy alternates between updating spatial and temporal parameters across iterations. Spatial parameters focus on per-frame artifacts, while temporal parameters capture inconsistencies across video frames. The updates are defined as:

  $$\begin{aligned} &\text{Update spatial: } \theta_s = \theta_s - \eta\nabla_s\mathcal{L}(\theta_s), \\ &\text{Update temporal: } \theta_t = \theta_t - \eta\nabla_t\mathcal{L}(\theta_t), \end{aligned} \tag{3}$$

  where $\mathcal{L}$ is the loss function, and η is the learning rate.

4.2. Implementation Framework

The training process employs the Adam optimizer with carefully tuned hyperparameters optimized for stable fine-tuning. I use a learning rate of 5×10^{-6} with weight decay set to 0.005, processing data in small batches of 8 to maintain training stability. Early stopping is implemented with a patience of 3 epochs to prevent overfitting while ensuring sufficient model convergence.[8]

The loss function used is BCEWithLogitsLoss, combined with a ReduceLROnPlateau scheduler that reduces the learning rate by a factor of 0.5 when validation performance plateaus.[9] This combination ensures robust model selection and training stability across epochs. Checkpoints are saved every two epochs, allowing for recovery and evaluation at multiple stages of training. Comprehensive model evaluation is performed after each epoch, reporting metrics such as accuracy, precision, recall, F1 score, and AUC.

Input frames undergo ImageNet-style normalization using mean $\mu = [0.485, 0.456, 0.406]$ and standard deviation $\sigma = [0.229, 0.224, 0.225]$. The training pipeline processes video sequences in batches, with each sequence containing 32 frames at 224×224 pixel resolution. The implementation leverages PyTorch's DataLoader system for efficient batch processing, with separate data loaders for training and validation sets.[9]

Visualization tools included Grad-CAM for heatmaps[10], Matplotlib for plotting curves, and scikit-learn for implementing the logistic regression baseline.[11, 12]

4.3. Training Strategy

The training process involved four distinct experiments by combining two fine-tuning strategies—gradual unfreezing and alternate freezing with two datasets; the FF++ DLC dataset and the Zoom DLC dataset. Specifically, the model was trained using: Gradual Unfreezing on FF++ DLC, Gradual Unfreezing on Zoom DLC, Alternate Freezing on FF++ DLC, Alternate Freezing on Zoom DLC. Each configuration was independently trained and subsequently evaluated on a hold-out Zoom dataset and the Celeb-DF dataset to assess performance under video conferencing conditions and generalizability to out-of-distribution samples. This approach facilitated a comparative analysis of fine-tuning strategies and datasets, highlighting the balance between generalization and domain-specific optimization.

5. Experiments

5.1. Experiment Setup

The experiments were designed to compare the performance of different models, datasets, and fine-tuning strategies for detecting deepfake manipulations in video conferencing environments. Implementation was performed on an NVIDIA A100 80GB PCIe GPU with CUDA 11.5 support, chosen for its robust deep learning capabilities and efficient GPU utilization.[13]

Training employed a batch size of 8 clips (each consisting of 32 frames at 224×224 resolution) and early stopping with a patience of 3 epochs. During training, a standard classification threshold of 0.5 was used for predictions and validation evaluations. The Adam optimizer was used with a learning rate of 5×10^{-6} and a weight decay of 0.005. For final evaluation on the hold-out Zoom dataset and out-of-distribution Celeb-DF dataset, a lower threshold of 0.1 was chosen to prioritize high recall in detecting manipulated videos, balancing between the original model authors' recommended 0.04 threshold and my empirical testing.

Each training epoch involved saving model checkpoints and evaluating performance on a hold-out validation set.

5.2. Evaluation Metrics

Metrics like AUC and AUPRC, which are threshold-independent, were prioritized for performance comparisons because they provide a more reliable measure of model performance, especially under varying evaluation conditions. AUC and AUPRC highlight differences in generalizability and domain adaptation, as observed in the comparison of models fine-tuned for Zoom-specific data versus general-purpose data. Threshold dependent metrics such as Accuracy and F1 score were also reported but are less reliable given the constant classification threshold of 0.1 used in all evaluations.

5.3. Comparison of Fine-Tuning Strategies

The experiments compared two fine-tuning strategies: **gradual unfreezing** and **alternate freezing**, applied to two datasets: the FF++ DLC dataset and the Zoom DLC dataset. Each combination was evaluated on the hold-out Zoom dataset to assess performance under video conferencing conditions, and on the Celeb-DF dataset to evaluate generalizability to out-of-distribution (OOD) high-quality samples.

5.4. Results and Analysis

Table 2 summarizes the results across all experiments. Key observations are highlighted below:

1. **Logistic Regression Baseline:** Logistic regression performed poorly, achieving an AUC of 0.04 on the Zoom dataset. This highlights the limitations of simple models in capturing complex spatiotemporal artifacts.

2. **Base Model Performance:** The pre-trained AltFreezing model without fine-tuning achieved an AUC of 0.81 on the Zoom dataset, indicating strong baseline performance but limited adaptability to domain-specific artifacts.

3. **Fine-Tuning with Gradual Unfreezing:** Gradual unfreezing on the FF++ DLC dataset achieved an AUC of 0.83 and an F1 score of 0.76, outperforming alternate freezing in terms of generalization to Zoom data.

4. **Fine-Tuning with Alternate Freezing:** Alternate freezing achieved the highest AUC (0.85) on the Zoom dataset when fine-tuned on Zoom data, demonstrating its efficacy in leveraging domain-specific information. However, it struggled to generalize to OOD data, achieving an AUC of 0.52 on Celeb-DF.

5. **Challenges in Subject Position, Lighting, and Framing:** The model struggled with videos where subjects were far from the camera or poorly lit, leading to reduced facial detail and ambiguity in feature detection. Portrait-mode videos with black borders further

Model/Method	Train Data	Eval Data	AUC	AUPRC	F1	Accuracy
logistic regression	Zoom	Zoom	0.04	-	0.18	0.10
base model	N/A	Zoom	0.81	0.80	0.67	0.68
base model	N/A	Celeb DF	0.93	0.92	0.72	0.62
alternate freezing	FF++(dlc)	Zoom	0.83	0.79	0.70	0.62
alternate freezing	Zoom	Zoom	**0.85**	0.76	0.70	0.62
gradual unfreezing	FF++(dlc)	Zoom	0.83	0.79	**0.76**	**0.72**
gradual unfreezing	Zoom	Zoom	0.82	**0.81**	0.65	0.68
alternate freezing	Zoom	Celeb DF	0.52	0.48	0.62	0.50

Table 2: Performance comparison of models across training and evaluation configurations on Zoom and Celeb-DF datasets. Metrics include AUC, AUPRC, F1 Score, and Accuracy, with bold values indicating the best performance of the optimized models, highlighting trade-offs between domain-specific optimization and generalization.

disrupted face detection and introduced spatial inconsistencies, likely due to training on consistently framed data.

6. **False Positives from Monitor Glare:** Close proximity to the computer screen caused glare, creating artificial lighting effects misinterpreted as manipulation artifacts.

The results underscore the trade-off between optimizing for domain-specific performance (Zoom) and maintaining generalization across OOD datasets (Celeb-DF). Gradual unfreezing demonstrated better generalizability, while alternate freezing excelled in domain-specific scenarios.

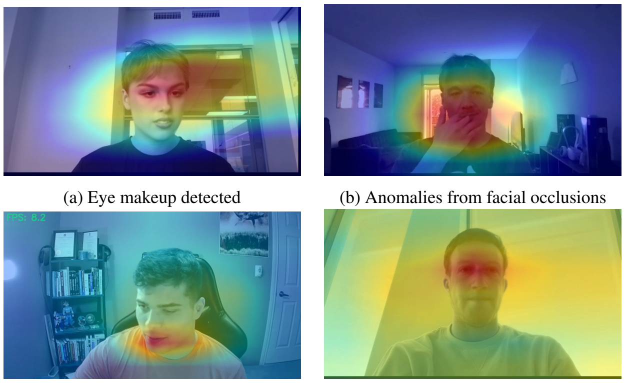
5.5. Discussion and Insights

The experiments highlight distinct strengths of the two fine-tuning strategies. Gradual unfreezing retained generalization capabilities, performing well on Celeb-DF, while alternate freezing leveraged domain-specific features, achieving the highest AUC on Zoom data. However, challenges such as poor lighting, distant subjects, and unusual framing styles revealed limitations in handling real-world variability, emphasizing the need for a larger, more diverse and representative training dataset.

False positives caused by monitor glare suggest that augmenting training data with synthetic glare effects or improving preprocessing could enhance robustness. Balancing domain adaptation and generalization remains a key challenge, pointing to the potential of hybrid models and larger datasets for improved performance across varied scenarios.

6. Conclusion and Future Work

The results underscore the importance of dataset relevance over size. While the larger FF++ dataset offered broader training coverage, its limited similarity to Zoom videos reduced its impact. In contrast, the smaller Zoom dataset outperformed due to its closer alignment with the target domain. Future work should focus on larger, domain-specific datasets to improve performance further.



(a) Eye makeup detected

(b) Anomalies from facial occlusions

(c) Mouth artifacts detected

(d) Artifacts near eyes detected

Figure 2: Examples of artifact detection on Zoom videos. Artifacts include eye makeup effects, hand occlusions, and synthetic patterns around mouth and eyes. These visualizations show the model's sensitivity to subtle manipulations.

7. Code

My code and instructions on running the experiments are available on my GitHub
https://github.com/csbrendan/CS229

8. Appendix

This appendix contains supplementary information to support the main content of the paper. Additional details, results, and explanations are provided below.

9. Contributions

This work was conducted as part of my contributions to Validia. My colleagues at Validia provided several of the real-world Zoom video examples, which were instrumental in creating the Zoom DLC dataset.

10. Model Development

10.1. Augmentation Training

In an initial effort to adapt the FF++ dataset for video conferencing conditions, I applied domain-specific augmentations to simulate artifacts commonly observed in such environments. These included JPEG compression with a quality factor of 80 to replicate lossy encoding artifacts, Gaussian blur using a 5x5 kernel to mimic lower resolution and out-of-focus effects, and brightness adjustments of ±10% to reflect dynamic lighting conditions.

While this approach aimed to enhance the model's robustness to video conferencing degradations, it did not fully capture the complex patterns present in real Zoom recordings. As a result, the augmentation strategy was ultimately replaced by training on custom datasets created using the DLC method, which better represent the unique characteristics of the target domain.

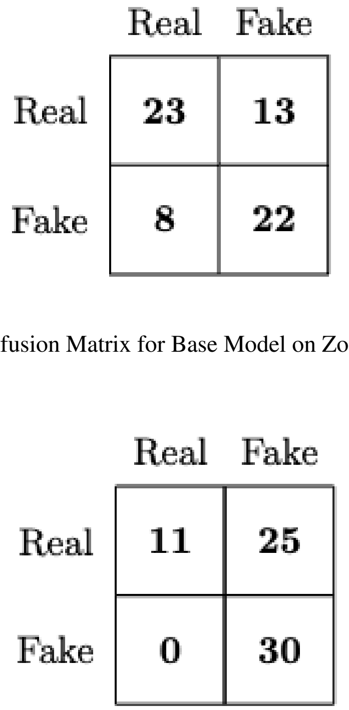
10.2. Self-Swap Dataset

The Self-Swap dataset was created to generate synthetic videos where the subject's own face is swapped onto their original frames. This approach introduces subtle temporal inconsistencies. The dataset consists of 100 training videos, equally split between original and self-swapped samples, and 20 validation videos, maintaining the same ratio. Original videos were sourced from the FF++ dataset to ensure consistent quality.

Despite its intended purpose of challenging the model to detect manipulation subtleties within realistic scenarios, this dataset did not appear to significantly improve the model's performance. The likely reason is that the pretraining on FF++ had already saturated the model's capacity to detect these frames, given the minimal manipulation required to swap one's own face onto themselves. This suggests that further fine-tuning with such minimally altered data may not provide additional value in cases where the model has already been exposed to similar patterns during pretraining.

11. Performance Metrics

11.1. Confusion Matrix Analysis



	Real	Fake
Real	**23**	**13**
Fake	**8**	**22**

(a) Confusion Matrix for Base Model on Zoom Dataset.

	Real	Fake
Real	**11**	**25**
Fake	**0**	**30**

(b) Confusion Matrix for Alternate Freezing on Zoom Dataset.

Figure 3: Confusion matrices comparing the performance of the Base Model and Alternate Freezing strategy on the Zoom dataset. The Base Model provides a more balanced classification, while the Alternate Freezing strategy excels at detecting fake videos with zero false negatives but incurs a higher false positive rate.

11.2. ROC and AUPRC Analysis

The ROC and AUPRC curves compare the performance of the Base Model and the Alternate Freezing model on the Zoom dataset. The ROC curve demonstrates the trade-off between the true positive rate and the false positive rate across different decision thresholds. Meanwhile, the AUPRC curve highlights the relationship between precision and recall, providing insights into the model's performance on imbalanced datasets.

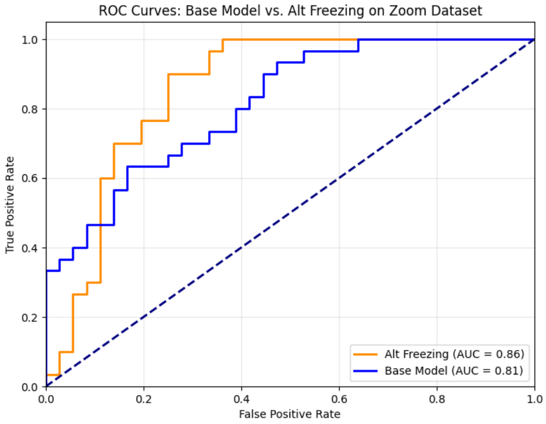


Figure 4: ROC Curve comparing the Base Model and Alternate Freezing on the Zoom dataset. The Alternate Freezing model achieves a higher AUC (0.86) compared to the Base Model (0.81), indicating improved true positive rates with fewer false positives.

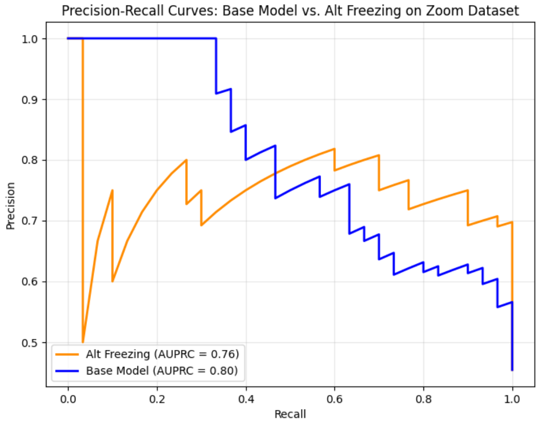
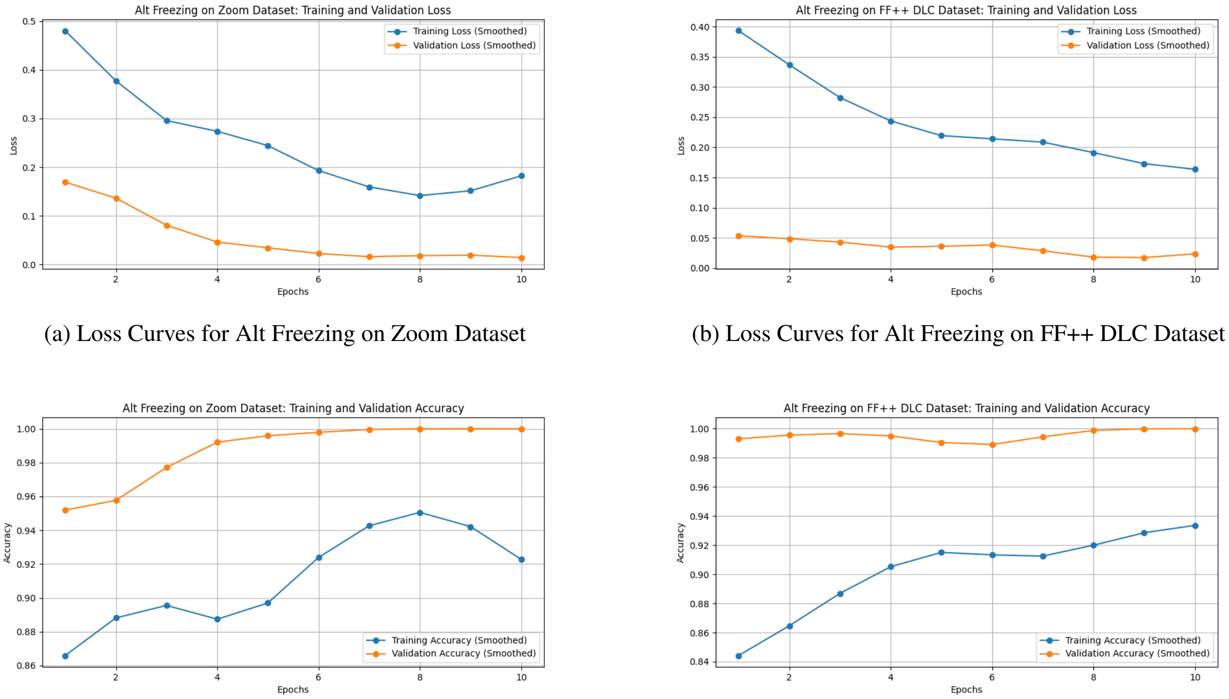


Figure 5: AUPRC Curve comparing the Base Model and Alternate Freezing on the Zoom dataset. While the Base Model achieves a slightly higher AUPRC (0.80) than Alternate Freezing (0.76), the gap suggests areas for further optimization in recall and precision trade-offs.

These ROC and AUPRC values are rounded and may differ slightly from those in Table 2.

12. Training and Validation Analysis



(a) Loss Curves for Alt Freezing on Zoom Dataset

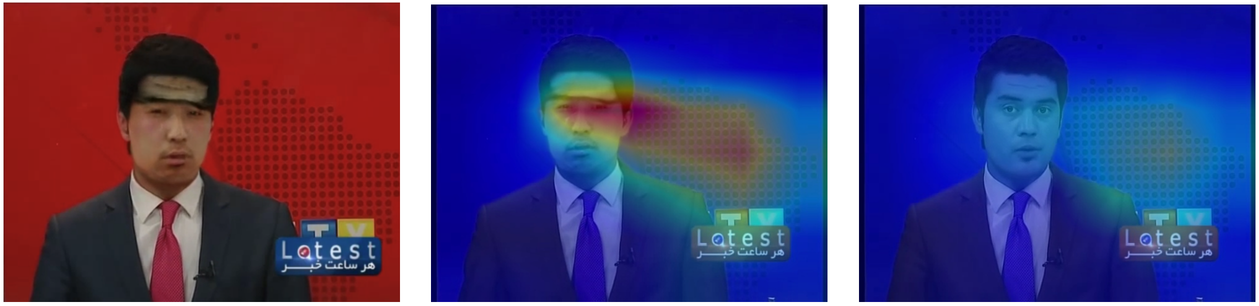
(b) Loss Curves for Alt Freezing on FF++ DLC Dataset

(c) Accuracy Curves for Alt Freezing on Zoom Dataset

(d) Accuracy Curves for Alt Freezing on FF++ DLC Dataset

Figure 6: The validation accuracy and loss curves highlight key differences between the FF++ DLC and Zoom datasets. For the FF++ DLC dataset, high initial validation accuracy and minimal loss improvement suggest the model's pretraining on similar data limited further learning. In contrast, the Zoom dataset shows noticeable gains in validation accuracy and loss reduction, demonstrating effective domain-specific fine-tuning for novel video conferencing artifacts. These results underscore the importance of dataset similarity and diversity in training performance. Note that training accuracies appear higher than final evaluation metrics due to the use of sliding window predictions on 32-frame clips during training versus video-level mean predictions during real-world evaluation.

13. Additional Examples



(a) Severe synthetic anomaly

(b) Heatmap highlighting manipulation artifacts

(c) Heatmap on unmanipulated original for comparison

Figure 7: Example of a severe synthetic anomaly and corresponding heatmaps. (a) Severe manipulation artifact, (b) Heatmap effectively detects anomalies, (c) Heatmap of an unaltered original video frame for baseline comparison.

References

[1] hacksider. Deep live cam: Real-time face swap and one-click video deepfake with only a single image. `https://github.com/hacksider/Deep-Live-Cam`, 2023. Accessed: 2024-12-06.

[2] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, and Houqiang Li. Altfreezing for more general video face forgery detection, 2023.

[3] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces, 2018.

[4] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics, 2020.

[5] Zhihao Gu, Yang Chen, Taiping Yao, Shouhong Ding, Jilin Li, Feiyue Huang, and Lizhuang Ma. Spatiotemporal inconsistency learning for deepfake video detection, 2021.

[6] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection, 2019.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. Presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009. DOI: `10.1109/CVPR.2009.5206848`.

[8] Lutz Prechelt. Early stopping - but when? In Genevieve B. Orr and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pages 55–69. Springer, 1998.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library, 2019.

[10] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruvan Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[11] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 12, 2024.