Implementing Deepfake Detection with FACTOR

CSCI1470 Final Project, Spring 2025

By Eva Azazoglu, Zoey Katzive, Subash Matu, and Huda Saeed

I. Introduction

Our group decided to implement an existing paper, "Detecting Deepfakes Without Seeing Any," that aims to create a model to detect deepfakes, in particular deepfakes that the model has not encountered before, which is increasingly critical as the breadth of deepfake generation techniques continues to increase (Reiss et. al, 2023). Such deepfakes are called 'zero-day' attacks, which traditional deepfake classifiers perform poorly on. This is because traditional classifiers are supervised models and thus only perform well on the deepfakes they have been trained to detect. This paper proposes a deepfake detection method that leverages the fact that deepfakes are often accompanied with implicit or explicit claims promoting the attacker's false message (i.e. a caption, a face in an image is attributed to a particular person, the visuals and audio of a video are aligned, etc.) that are not synthesized perfectly in the deepfake; by computing the similarity between the claim and the deepfake the model determines if the visual is a deepfake if they are too different. This is consequently a classification problem, as it aims to determine if the input is a deepfake or not. We chose this paper because of its importance, as deepfake attacks can be extremely harmful due to their potential to spread misinformation, which can have ramifications spanning from the destruction of a person's reputation to inciting global conflict.

II. Methodology

Our paper is "training-free" and utilizes open-source models to extract features.

A. Face swapping

For face-swap detection, FACTOR works by calculating a similarity between a face-swapped image and a true image of the deepfake identity. The model inputs are an image x, the image's supposed identity f, and it additionally depends on an open-source model FaceX-Zoo for computing facial features denoted in the paper as ϕ_{id}. The similarity score is then calculated as the cosine similarity over ϕ_{id} features. We used the FaceForensics++ dataset (Rößler et al., 2019),

which contains 1,000 videos; due to computational constraints on running the dataset locally, we used a sample of 50 videos from the dataset (25 real and 25 fake).

To evaluate our model, we used average precision (AP) and Receiver Operating Characteristic Area Under the Curve (ROC-AUC) as criteria. Our ablation study consisted of two different ablations that we consider in conjunction; we chose to do this for reasons that we outline in the Challenges section. In our first ablation study, we varied the number of unpaired reference identities (range between 1 and 25) that are used to compute the distance between real and fake test samples, and then calculated ROC-AUC and AP scores in order to evaluate how the reference set size impacts the model's performance. In our second ablation study, we compared each fake video to the real video of the corresponding identity, in order to establish a baseline for true distance, and then we tested the model's performance using unpaired reference sets of unrelated real identities (we varied these sets between 5, 10, 15, and 20 identities). We then measured the minimum L2 distance between each fake set and the unrelated reference set, and instead of ROC-AUC and AP scores, we visualized how distance distributions shifted compared to the true distance baseline. We designed these two ablation studies so that we could consider them in conjunction in order to evaluate both the sensitivity of FACTOR to the size of the reference pool, and its ability to distinguish deepfakes from real videos under increasingly unconstrained conditions. Together, these two ablation studies serve to assess FACTOR's robustness in face of sparse reference data and highly generalized data.

B. Audio-visual

For audio-visual deepfake detection, we used the PolyGlotFake dataset, which has real videos and deepfake videos (Hou et. al, 2024). With a sample of 25 real videos and 25 fake videos, we preprocessed the videos to extract the audio and regions of interest (the latter of which was generated by an off-the-shelf landmark detector following Av-HUBERT's implementation), extracted features from the audio and visual components using the Av-HUBERT Large model as the feature encoder, and then evaluated the similarity between the two via cosine similarity. However, this score is for every video frame, so to compute a score for the overall video, we choose the λ% percentile truth score. We then used average precision (AP) and Receiver Operating Characteristic Area Under the Curve (ROC-AUC) as our evaluation criteria. As an ablation study, we also varied λ to see its effect on our results.

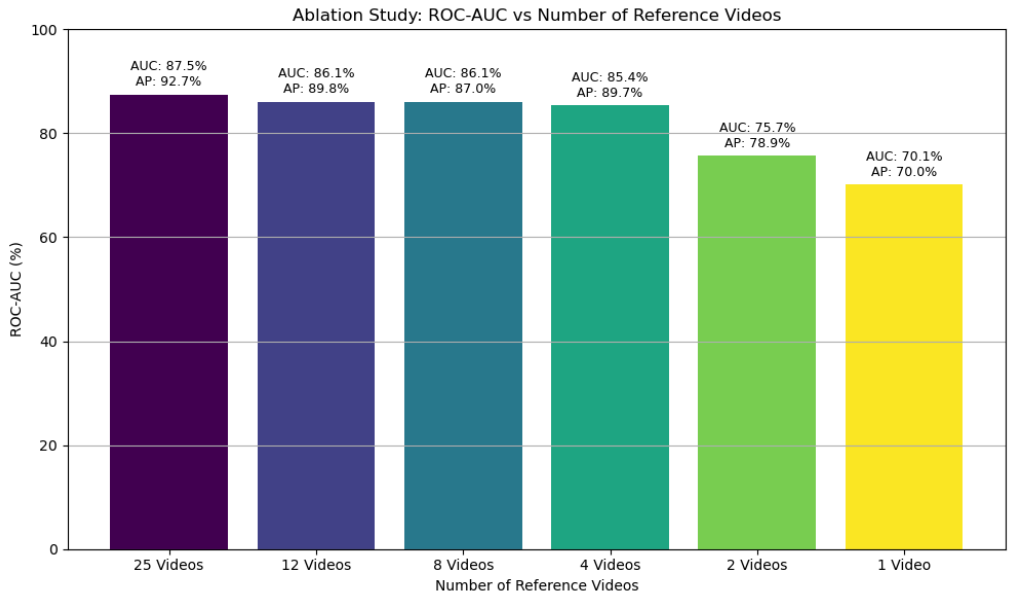
Finally, we trained a small, three-layer MLP on top of the embeddings produced by our audio and visual encoders and the cosine similarity between them. We initially tried to implement this in Tensorflow, but many hours spent debugging proved that Torch was a more promising approach. We created two linear layers and one layer with ReLU activation, and used the softmax function to calculate the probabilities that each inputted file was a deepfake. We trained

the model for ten epochs and evaluated the results of the MLP with cross-entropy loss and the classification accuracy.

III. Results

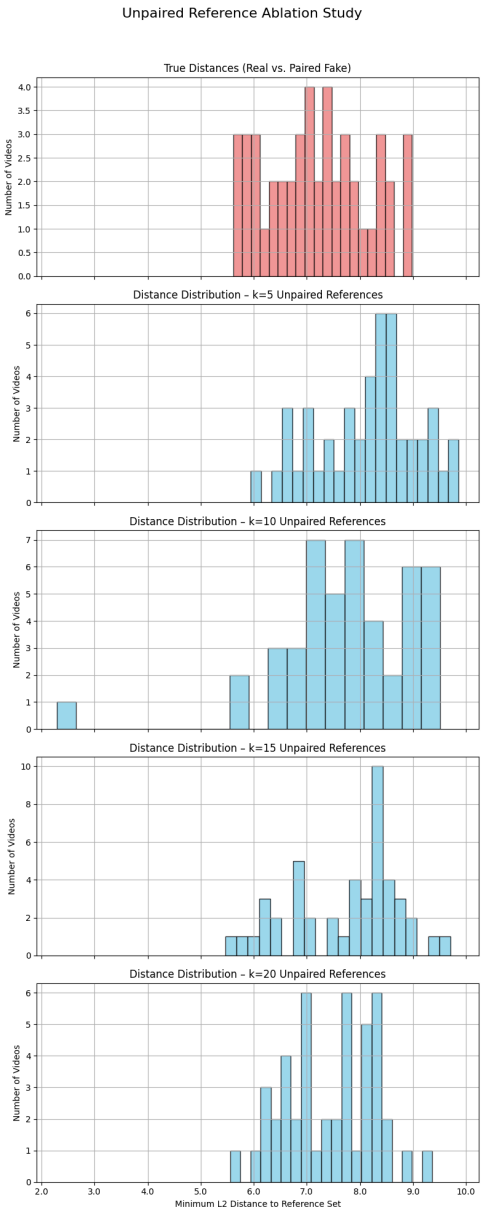
A. Face swapping

Upon evaluating FACTOR's face swapping performance on our dataset of 50 videos, we calculated a ROC-AUC score of 87.50 and an AP score of 92.68 as a baseline, with 25 videos in the training (reference) set and 25 videos in the test set. Given that we worked with a relatively small sample size, this suggests that FACTOR is able to make reasonably good classifications even with a limited sample size. For our first ablation study, we can observe the results in the chart below:



We can see that ROC-AUC and AP scores are higher with a larger reference set size. For example, at reference size 25, we had a ROC-AUC score of 87.50 and an AP score of 92.68 (as in the baseline), while at reference size 1, we had a ROC-AUC score of 70.14 and an AP score of 70.01. This reveals that FACTOR performs better with larger reference sets and its classification confidence is higher when it has more references. However, as we can see in these results, the model was still able to perform surprisingly well even with just one reference video, which could point to FACTOR's robustness in classification tasks.

For our second ablation study, we can observe the results below:



We can see that the baseline true distances are clustered tightly between 5.5 and 9.0, while the unpaired references (k=5-20) tend to skew left and exhibit higher averages as the number of references (k) increases. We can infer from this that FACTOR is successful at distinguishing fake videos from unrelated real identities; additionally, we can see that an increase in the size of the unrelated reference set leads to larger minimum distances, which means that it becomes easier to separate fake videos from actual identity matches as the reference set becomes more diverse.

Considering the two ablation studies in conjunction, we can infer that FACTOR is robust at making predictions across various constraints, as it is able to perform well with smaller reference

sets (though it benefits from larger reference pools) (Ablation 1), and it is also able to perform strongly even with unpaired and unrelated reference pools (Ablation 2).

B. Audio-visual

When we evaluate FACTOR's audio-visual performance using 50 videos uniformly subsampled to thirty-two frames, we calculate a mean ROC-AUC of 78.50 and a mean AP of 84.43. This demonstrates that even with a small sample size, FACTOR classifies audio-visual deepfakes fairly well. When we ablate our hyperparameter λ , which represents the percentile to take over the per‑frame cosine similarities, we see that ROC-AUC and AP increase up until an optimal point and then sharply decrease. This optimal point is estimated at around the 20th percentile.

The results of the small MLP we implemented are promising as well and lead us to believe that with more depth and training time, model performance would increase significantly. The softmax fake probabilities assigned to each fake video were greater than 0.5 and and for each real video they were less than 0.5, indicating that the model classifies each file correctly. The cross-entropy loss was relatively large after ten epochs, finishing at around 0.6. This is likely because while the model does classify every input correctly, the softmax probabilities are quite close to the decision boundary, meaning that the cross-entropy loss function will penalize the model's lack of "confidence" in its correct prediction. Since we only input 25 real and 25 fake videos, it makes sense that the MLP would learn a relatively weak separation and train with a moderate loss result. With more computational power and time, it would be interesting to continue to build on this model and get a clearer picture of how neural networks may help extend the positive results of the FACTOR implementation.

IV. Challenges

A. Face swapping

In the face-swapping section, we faced significant challenges with preprocessing; in particular, fitting the FaceForensics++ dataset to run with the FACTOR code was difficult due to mismatches in expected data types and storage limitations, as well as syntax errors and misconfigurations within the FACTOR preprocessing code. It therefore took significant manual debugging (particularly resolving unexpected tensor shapes) to align our dataset with FACTOR's expected preprocessing pipeline. While this was frustrating, it provided valuable exposure to troubleshooting externally written code and teamwork to brainstorm potential approaches going

forward. While we considered looking for an alternative dataset at the last minute, we valued FaceForensics++ for its comprehensive and accessible contents and structure. Therefore, we ultimately decided to stick with debugging the FaceForensics++ preprocessing errors, and ran the FACTOR face-swapping program on a smaller subset of the dataset to ensure we stayed on track with our timeline.

Additionally, since we were working on MacBooks without access to CUDA-compatible GPUs, we had trouble with feature extraction and some parts of preprocessing that relied on CUDA. To resolve this challenge, we took these problems into account when we rewrote our evaluation pipeline in TensorFlow, and ended up only using a subset FaceForensics++ consisting of 50 videos, but these computational limitations did constitute a significant challenge in our initial model setup.

In our ablation study, we also faced a challenge with the compatibility of the format between the dataset and the model. Initially, we stuck closely to the research paper's ablation process in the hopes of replicating their study; however, we soon realized that our dataset had somewhat limited applicability for that exact ablation study because it doesn't contain real-fake video pairs for each given identity in the same way as the paper's dataset did. As a result, we found that we could achieve a more comprehensive analysis by performing two different ablation studies and reflecting on their results in conjunction (our 2 ablation studies are outlined in the Methodology section). In our second study, we deviate slightly from the paper's original ablation study, as we mitigate the lack of real-fake video pairings by including only fake identities in the test set and running it on the reference. However, this brought up another challenge: since our test set now only contained fake videos, we could no longer calculate ROC-AUC scores. This roadblock pushed us to think outside the box of what we had thought would be a valid evaluation, and we instead ended up calculating distances and evaluating the correspondingly produced graphs in conjunction with the ROC-AUC scores for the original and the first ablation.

In general, the challenges we experienced with our ablation study design presented a valuable learning opportunity to us, particularly because it raised important questions about the process of choosing datasets and ensuring compatibility between dataset and model. Before we encountered these challenges, we had assumed that a dataset simply containing deepfakes in the correct format (MP4 videos in our case) would be sufficient for compatibility with a model; conversely, we had also assumed that a classifier model would be able to work easily with any given dataset that contained real and fake videos in our case. While this ablation process was a significant technical and conceptual challenge for us, it has allowed us to explore the interplay between data organization and model performance in a practical setting.

B. Audio-visual

The steps described in the Methodology section are encapsulated in 3 scripts: preprocess.py, inference.py, and eval.py. Each script had to be adjusted to work on our computers given that we have MacBooks without a CUDA-compatible GPU which was embedded in the paper's code and a new dataset (this repository was not dataset-agnostic), which required a great deal of work to understand the structure and the code of this complex repository which was nested within Av-HUBERT's repository. It took extensive time to debug our repository to run, especially given the fact that we are not extensively familiar with CLI commands, which were crucial in setting up this infrastructure. We adjusted the code for preprocessing and evaluation and rewrote the inference script in Tensorflow.

After this software hurdle, the code itself was also a challenge to first conceptually wrap our minds around, as we had never worked with MP4 files before. Our replacement of PyTorch with TensorFlow was also challenging because the entire pretrained encoder, Av-HUBERT, was built from the Fairseq toolkit which is inherently built upon the PyTorch library. To simply rewrite our inference file in Tensorflow, we would need to go back and manually edit the encoders themselves, which we determined was beyond the scope of this project. Instead, we were able to export the entire model to ONNX, which is an open-source ecosystem that helps facilitate the exchange of models between frameworks. This allowed us to change the Torch calls in our inference file to Tensorflow and Numpy compatible calls, but also took up extensive coding and debugging time.

Finally, training the MLP required additional coding and debugging. We initially tried to run it in Tensorflow but after hours debugging decided to approach it using PyTorch, which ran successfully somewhat faster. There were several decision points here, including the architecture of the network itself as well as how many epochs to train for. This implementation took a significant amount of time but eventually came together successfully.

V. Reflections

A. Face swapping

Ultimately, the results of this project are promising. We hoped that our implementation would allow FACTOR to run effectively enough to obtain similar results to the paper, and despite the limited size of our dataset, we were pleased with our results in both ablation studies. We successfully met our base goal as we implemented the FACTOR pipeline and ran evaluations on

both datasets (FaceForensics++ in the case of face swapping) and used ROC-AUC and AP to quantify performance (though our second ablation study did not use these metrics, so this point is slightly limited in that sense). Additionally, we also achieved our target goal, as we ran evaluations on FaceForensics++ in the face swapping section and also conducted not just one but two ablation studies in conjunction.

The main way in which our approach changed over time was in the ablation study, where we pivoted from one to two studies in order to accommodate for the limitations of our dataset due to compatibility issues between the dataset and the model. While we originally did not plan for this, we were able to adapt to these changing circumstances and obtain valuable results. However, this adaptation of our project did involve significant unexpected time commitments; therefore, if we could do our project over again, we would make sure to familiarize ourselves early on with FACTOR's specific reference requirements and that our dataset has matched real-fake identity pairs, in order to ensure compatibility between our model and dataset even at a level that is below the surface of initial observation. If we had more time to work on this project, we would like to run our second ablation with a larger identity set to observe how this affects the model's performance, and we would also like to standardize the evaluation metrics to add some more nuance beyond distance and ROC-AUC.

A key takeaway from the face swapping section is that FACTOR generalizes surprisingly well, even under conditions that involve unpaired or sparse data, and also that the size and diversity of the reference set significantly impacts the quality of the model's detection of deepfakes. Additionally, we learned that it is crucial to construct ablations with methodological rigor in order to obtain valuable and meaningful results.

B. Audio-visual

Here as well, we found our results to be encouraging. We hoped to achieve a relatively similar result to the paper in spite of our limited data, and we were able to achieve fairly high ROC-AUC and AP scores which we consider a success. Our model worked more or less as expected, and the addition of the MLP initially seemed ambitious but came together quite well. In terms of the goals we set at the beginning of this project, we met our base goal (implement the FACTOR pipeline and run AP and ROC-AUC evaluations on PolyGlotFake) and our target goal (run an ablation study on λ). We did not have time to meet the stretch goal (extend evaluation with different encoders, extend additional deepfake modalities, and build tools to visualize truth scores) but we were able to go beyond the paper with our MLP.

One of the key pivots we made was the decision to add the MLP at all. We had already spent a significant amount of time debugging our reimplementation of the code, but thought it would be interesting to extend the results of the paper and create our own model from scratch. This ended up informing us a lot about the promise of this paper and opportunities to build on the research.

While we would not necessarily change anything if we did our project again, we would certainly be able to speed up our process knowing what we know now and we would have time to create a deeper neural network than our current three-layer MLP. We would be very interested to see how a deeper network could learn a stronger separation between real and fake videos.

Our biggest takeaway from this project is that there appear to be some inherent characteristics of deepfakes that can be harnessed to counter these dangerous attacks. This makes us hopeful for the future of combating deepfakes even as artificial intelligence continues to advance. From a deep learning perspective, another big takeaway is the power of using off-the-shelf models for a variety of problems, which allows for high efficiency and performance.

Works Cited

Hou, Y., Fu, H., Chen, C., Li, Z., Zhang, H., & Zhao, J. (2024, December). Polyglotfake: A novel multilingual and multimodal deepfake dataset. In *International Conference on Pattern Recognition* (pp. 180-193). Cham: Springer Nature Switzerland.

Reiss, T., Cavia, B., & Hoshen, Y. (2023). Detecting deepfakes without seeing any. arXiv. https://arxiv.org/abs/2311.01458

Rößler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).