Truth in Motion: Depth and Flow Enhanced DeepFake Detection

Neli Čatar, Gellert Toth, Aimee Lin

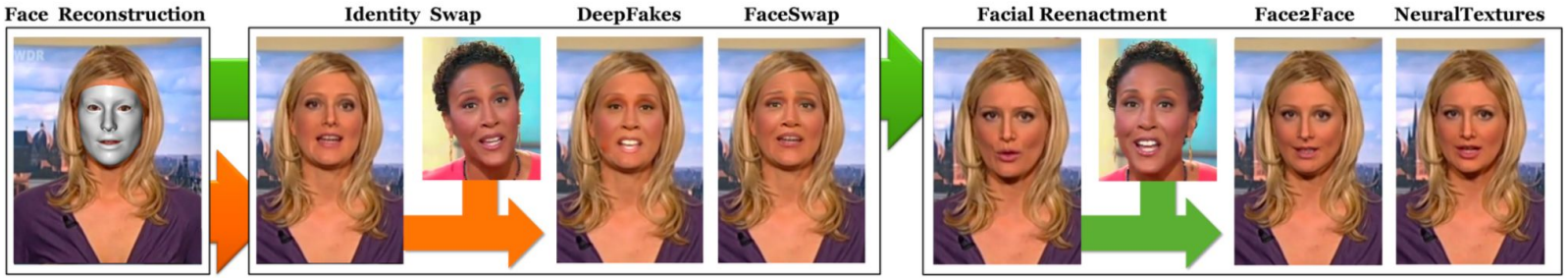
Computer Vision A.A. 2024-2025

Outline

1. Problem Statement
2. State of the Art
3. Dataset
4. Proposed Method
5. Experimental Setup
6. Model Evaluation
7. Conclusions
8. References

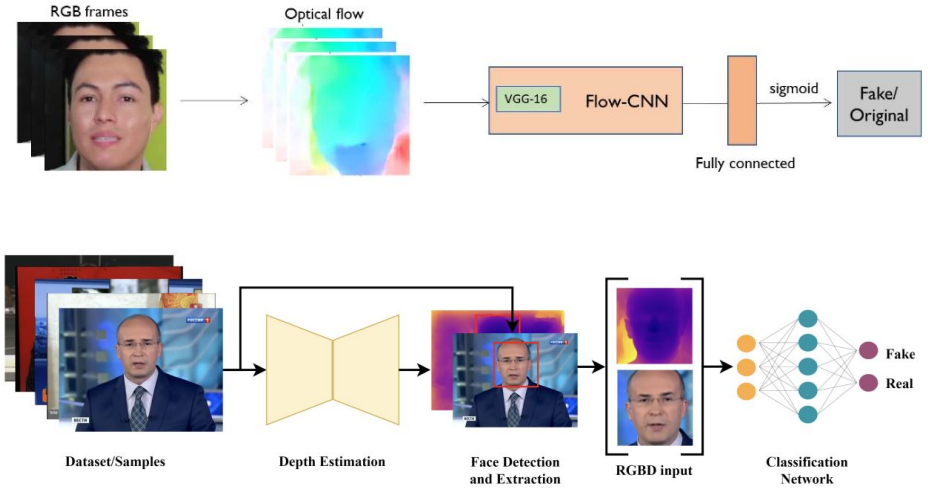
1. Problem Statement

- AI-driven methods have made the production of fake and manipulated videos simpler than ever [1]
- DeepFakes can be used to harm reputations and alter public opinions
- Accurately separating authentic videos from fake ones is crucial to preserve data integrity
- Also important to correct balance between detection efficacy and computational efficiency



2. State of the Art

- Steganalysis features with SVMs [1]
- Learned features with CNNs [1]
  - Flows [2, 3]
  - Depths [4]
- Transformers vs CNNs



3. Dataset

- FaceForensics++ [1]
  - 1000 original videos from Youtube manipulated with DeepFakes, Face2Face, FaceSwap, NeuralTextures, and FaceShifter
  - Also includes the original and manipulated videos from the DeepFake Detection Dataset hosted by Google and Jigsaw

4. Proposed Method

- Pre-processing
- Feature extraction generation
  - Flow using PWC-Net [5,6]
  - Depth using Depth-v2 [7]
- Training
  - Dino v2 [8]
- Compression
  - Quantization
  - Distillation

5. Experimental Setup

- Python scripts for pre-processing and feature extraction generation
- Jupyter Notebooks for training and compression locally and on Google Colab
- Combined all code into one notebook

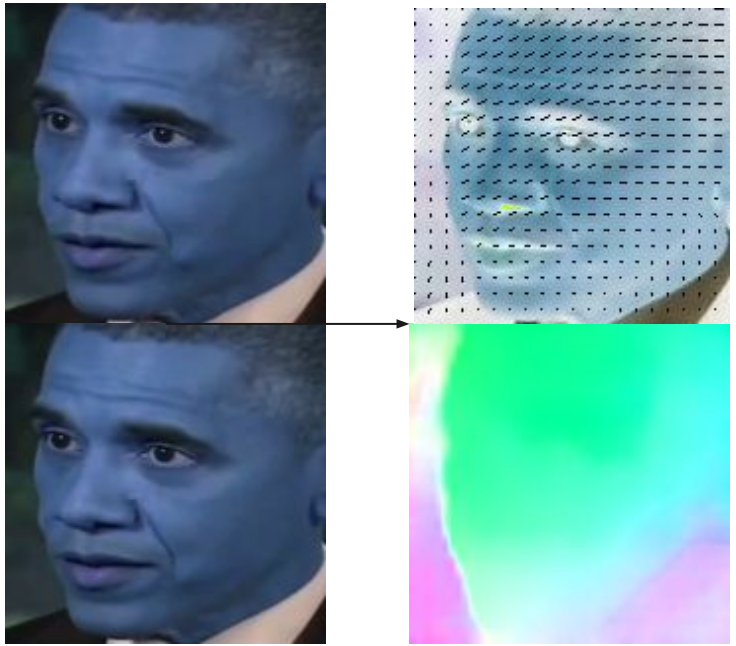
5.1 Pre-processing

- Downloaded dataset using script from authors
- Used OpenCV [9] to process videos and MTCNN [10] to extract faces from video frames
- Saved frames as 160 x 160 BGR images

5.2 Feature Extraction Generation

Flows

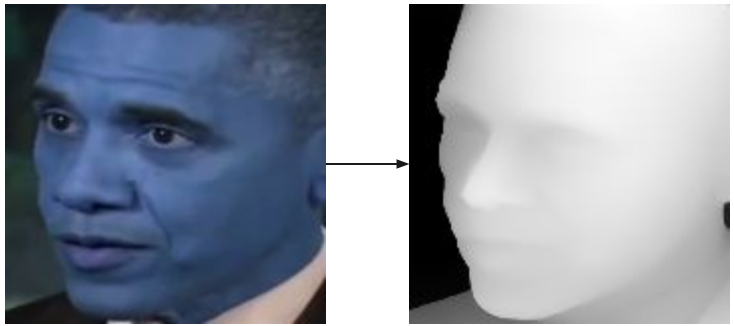
- Used PWC-Net [5,6] to extract flows from consecutive video frames
  - Output two channels (horizontal and vertical flows)
- Converted flows to RGB images where colour represents direction and saturation represents magnitude



5.2 Feature Extraction Generation

Depths

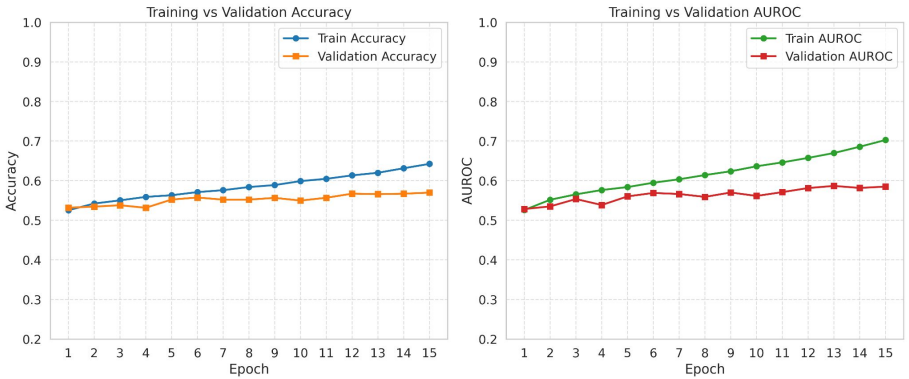
- Used Depth Anything V2 [7] to estimate depths from video frames
  - Output one channel (depths)
- Converted depths to grayscale images
  - Used grayscale to prevent introducing artefacts with colour maps



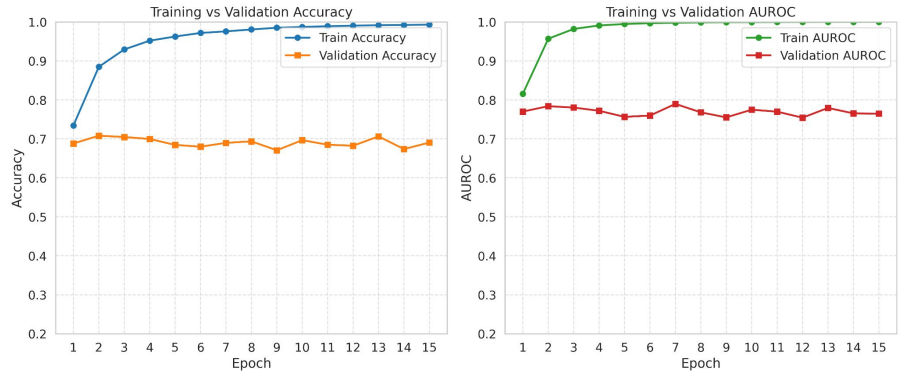
5.3 Training

- Dino v2 [8] takes 224 x 224 RGB images as input
  - Images resized to correct dimensions
  - Depths also replicated three times to generate pseudo-RGB images
- Last encoder layers and classification head parameters are trainable everything else is frozen

Dino-v2 base fine tuned on flow data



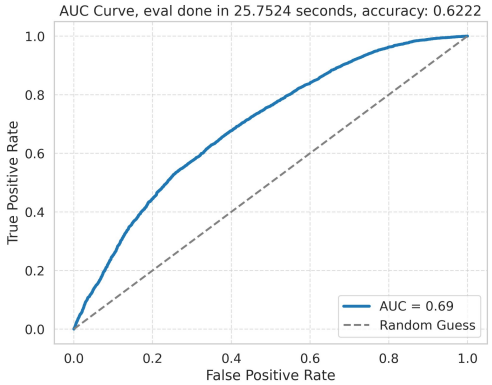
Dino-v2 base fine tuned on depth data

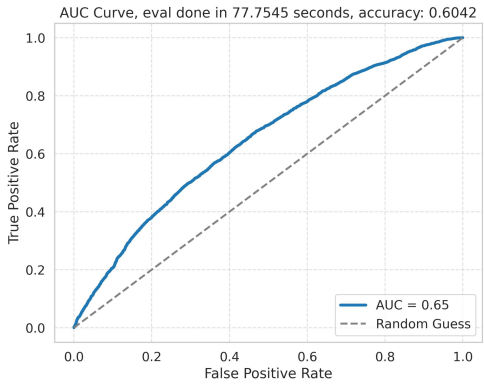


5.4 Compression

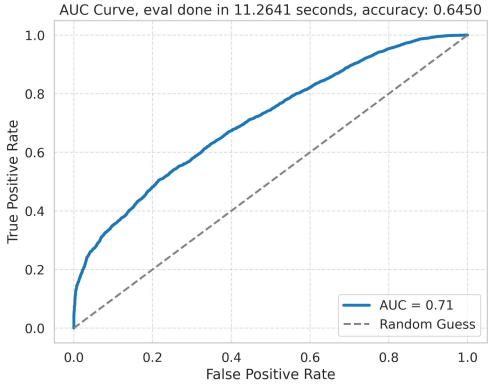
- Quantization:
  - Using float 16 instead of float 32
  - Done with autocast
- Distillation
  - KL divergence between final encoder representation of teacher and learner model
  - Because of different embedding sizes first project cls_token

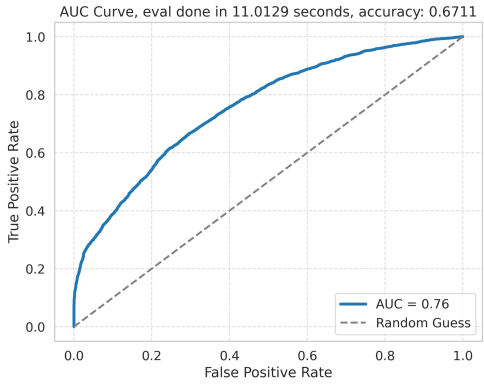
Dino-v2 base fp16 vs fp32





Dino-v2 small vs small distilled





6. Model Evaluation

	Accuracy	AUROC	Avg. Time / Epoch
Flow based fp16 model - Dino v2 base	0.5347	0.54	26.6
Depth based fp32 model - Dino v2 base	0.6042	0.65	77.75
Depth based fp16 model - Dino v2 base	0.6222	0.69	25.75
Depth based fp16 model - Dino v2 small	0.6450	0.71	11.26
Depth based distilled model - Dino v2 small	0.6711	0.76	11.01
Amerini et al. Flow & VGG16 [2] [a]	0.8161	-	-
Nassif et al. Flow & VGG16 [3] [b]	0.7584	0.8215	-
Maiono et al. Depth & XceptioNet [4]	0.9193	-	-

[a] Only Face2Face [b] Only DeepFake and Face2Face

7. Conclusions

- Final considerations
  - Overfitting
  - Depth worked considerably better than flows
- Future Work
  - Pre-process images to correct dimensions for transformers
  - Train on larger portion of the dataset
  - Potentially combining both depth and flow methods
    - Concat cls_token from the two transformer models

8. References

[1] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “FaceForensics++: Learning to detect manipulated facial images,” arXiv preprint arXiv:1901.08971, 2019.

[2] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, “Deepfake video detection through optical flow based CNN,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), Seoul, Korea (South), 2019, pp. 1205–1207, doi: 10.1109/ICCVW.2019.00152.

[3] A. B. Nassif, Q. Nasir, M. A. Talib, and O. M. Gouda, “Improved optical flow estimation method for deepfake videos,” Sensors, vol. 22, no. 7, p. 2500, Mar. 2022, doi: 10.3390/s22072500.

[4] L. Maiano, L. Papa, K. Vocaj, and I. Amerini, “DepthFake: A depth-based strategy for detecting Deepfake videos,” in Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges, Lecture Notes in Computer Science, vol. 13774, Springer, 2023, pp. 17–31, doi: 10.1007/978-3-031-37745-7_2.

[5] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 8934–8943, doi: 10.1109/CVPR.2018.00931.

8. References

[6] S. Niklaus, A Reimplementation of PWC-Net Using PyTorch, 2018. [Online]. Available: https://github.com/sniklaus/pytorch-pwc

[7] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, “Depth Anything V2,” in Advances in Neural Information Processing Systems, vol. 37, NeurIPS 2024, doi: 10.48550/arXiv.2406.09414.

[8] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, et al., "DINOv2: Learning Robust Visual Features without Supervision," arXiv preprint arXiv:2304.07193, 2023.

[9] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2008.

[10] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.

Thanks! Questions?