Deepfake Image Detection

Bucă Mihnea-Vicențiu Căpățînă Răzvan-Nicolae Luculescu Teodor

University of Bucharest, Faculty of Mathematics and Informatics

Objectives

There are two main parts to this project:

1. Analysis of cross-generator deepfake detection using three different approaches
2. Model attribution

Dataset

The dataset contains real images from the *CelebAHQ* and locally manipulated images produced by four generators: *LDM*, *Pluralistic*, *LAMA*, *Repaint*. You can read more about how this dataset was produced in **Section 3.3** of the following paper: [1]

Cross-Generator Deepfake Detection

First Method

This method trains an *xception41* [2] model from scratch (from *timm* [3]) on multiple datasets for one epoch each, using **Adam** optimizer and **cross-entropy loss**.

Tested On \Trained On	lama	ldm	repaint	pluralistic
lama	0.641	0.493	0.536	0.517
ldm	0.569	0.496	0.511	0.536
repaint	0.572	0.487	0.517	0.531
pluralistic	0.667	0.491	0.580	0.700

Table 1. Performance of model *xception41* (not pretrained); trained and tested on the provided datasets.

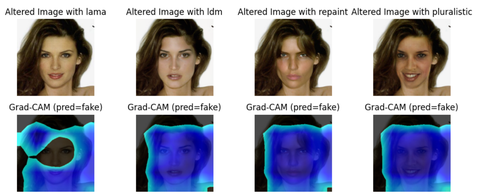

Figure 1. Example of **GRAD-CAM** on the testsets, model is trained on pluralistic.

Method Performance

- **Cross-domain** generalization is limited;
- *Pluralistic* performs best overall;
- *LDM* and *Repaint* are harder to detect;
- **Grad-CAM** activations often focus on background or peripheral areas;

The Second Method

The second method uses the same *xception41* architecture but initializes the model with pretrained weights (*ImageNet*, via the *timm* library), obtained in a self-supervised manner.

Tested On \Trained On	lama	ldm	repaint	pluralistic
lama	0.998	0.310	0.445	0.626
ldm	0.410	0.993	0.906	0.460
repaint	0.486	0.636	0.800	0.531
pluralistic	0.613	0.421	0.586	0.790

Table 2. Accuracy of pretrained XceptionNet (xception41).

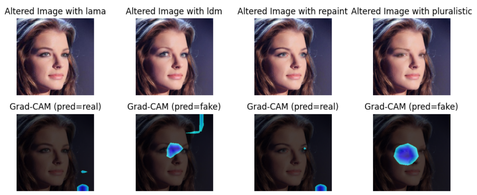

Figure 2. Example of **GRAD-CAM** on the testsets, model is trained on pluralistic.

Method Performance

- Strong memorization of known artifacts.
- Still struggles with unfamiliar artifact distributions.
- *LDM* and *Repaint* is the best cross-domain generalization.
- Successful detections focus on localized, high-contrast regions, but in many failed cases, activations are diffuse or absent—implying the model overlooks subtle inpainting traces.

Third Method

The third approach uses large self-supervised models (*CLIP* and *SAM*) as frozen feature extractors, training only a linear classifier on top. Unlike [4], who targeted fully-generated images, we focus on detecting inpainted regions via linear probing.

Tested On \Trained On	lama	ldm	repaint	pluralistic
lama	1.000	0.535	0.522	0.675
ldm	0.600	1.000	0.901	0.977
repaint	0.511	0.620	0.723	0.615
pluralistic	0.765	0.951	0.880	0.992

Table 3. Cross-domain accuracy matrix using *CLIP* features and a linear classifier.

The Third Method

Tested On \Trained On	lama	ldm	repaint	pluralistic
lama	0.870	0.420	0.514	0.588
ldm	0.349	0.741	0.552	0.561
repaint	0.536	0.543	0.547	0.575
pluralistic	0.647	0.527	0.543	0.627

Table 4. Cross-domain accuracy matrix using *SAM* features and a linear classifier.

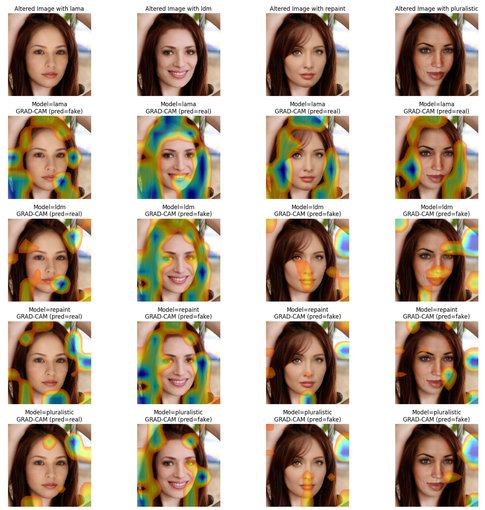

Figure 3. Example of **GRAD-CAM** on the testsets, of all the models trained with *CLIP*.

Method Performance

- *CLIP* outperforms *SAM* in cross-domain generalization.
- *CLIP* is more robust to domain shifts.
- Each detector is most confident catching fakes created by its own algorithm.
- Diffusion-based models can flag each other's edits and manage repaint reasonably well.
- The *ldm* variant generalizes best across different inpainting styles.
- **Grad-CAM** zeroes in on manipulated regions when the detector is right and spreads out when it struggles.

Model attribution.

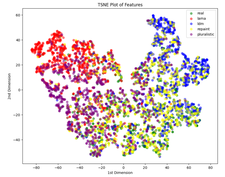

Figure 4. t-SNE plot of the xception41 (not pretrained) model tested on the datasets. **overall:** 45.04%; real: 54.33%; lama: 73.44%, ldm: 83.11%; repaint: 1.22%; pluralistic: 13.11%

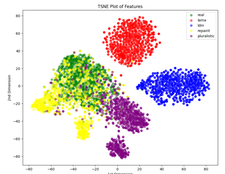

Figure 5. t-SNE plot of the xception41 (pretrained) model tested on the datasets. **overall:** 80.71%; real: 85.22%; lama: 98.44%, ldm: 100%; repaint: 66.88%; pluralistic: 53.00%

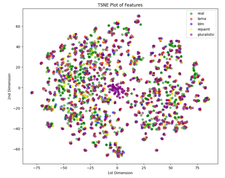

Figure 6. t-SNE plot of the CLIP model tested on the datasets. **overall:** 80.24%; real: 68.55%; lama: 99.22%, ldm: 88.55%; repaint: 51.77%; pluralistic: 93.11%

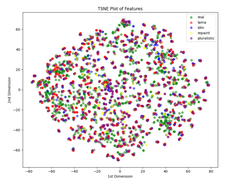

Figure 7. t-SNE plot of the SAM model tested on the datasets. **overall:** 46.20%; real: 17.77%; lama: 92.66%, ldm: 78.55%; repaint: 21.44%; pluralistic: 20.55%

Conclusions

- Scratch-trained model works well on some generators, poorly on others;
- Pretrained model improves detection, especially for LAMA and LDM;
- Repaint and Pluralistic are still harder to detect;
- Pretraining helps with real/fake separation, but domain gaps remain;

References

[1] Dragos Tantaru, Elisabeta Oneata, and Dan Oneata. *Weakly-supervised deepfake localization in diffusion-generated images*. 2023. eprint: `arXiv:2311.04584`.

[2] *timm/xception41.tf_in1k · Hugging Face — huggingface.co*. `https://huggingface.co/timm/xception41.tf_in1k`. [Accessed 03-05-2025].

[3] *timm — huggingface.co*. `https://huggingface.co/docs/timm/index`. [Accessed 03-05-2025].

[4] *GitHub - WisconsinAIVision/UniversalFakeDetect — github.com*. `https://github.com/WisconsinAIVision/UniversalFakeDetect`. [Accessed 03-05-2025].

[5] *Google Colab — Task 1 - Method 1*. `https://colab.research.google.com/drive/1kQSpdjYdTH8lV_ORXNrqZSlcrWmwg5n?usp=sharing`. [Accessed 03-05-2025].

[6] *Google Colab — Task 1 - Method 2*. `https://colab.research.google.com/drive/16zv3wHFGM0JxiIGN17RW7ZBafFPpWDqC?usp=sharing`. [Accessed 03-05-2025].

[7] *Google Colab — Task 1 - Method 3*. `https://colab.research.google.com/drive/1e7vWzGLktPXleivZwDnxtsDJqXHacjHt?usp=sharing`. [Accessed 03-05-2025].

[8] *Google Colab — Task 2 - Method 1*. `https://colab.research.google.com/drive/1Filsccd-qf50-eJRGZnI5tzaOJrEMtw_?usp=sharing`. [Accessed 03-05-2025].

[9] *Google Colab — Task 2 - Method 2*. `https://colab.research.google.com/drive/1gsMK-isI2DovbqFziZpYSfQ7HUt6DHM3?usp=sharing`. [Accessed 03-05-2025].

[10] *Google Colab — Task 2 - Method 3*. `https://colab.research.google.com/drive/1YlelBIgU5wMBoBCVxOdNYrTxCc0ohrT-?usp=sharing`. [Accessed 03-05-2025].

[11] *GitHub - Project — github.com*. `https://github.com/Razvan48/Proiect-Deep-Learning-DL`. [Accessed 03-05-2025].