Computer Science Tripos Part II Project Proposal

Investigating Robustness of GAN Fingerprint Detection Methods

Introduction and Description of the Work

The last few years have witnessed the steady evolution of image editing software into an easily accessible tool used for a wide variety of purposes which range from facilitating harmless pranks to fuelling entire defamation campaigns. This has been countered by an accompanying increase in the amount of research effort spent on developing algorithms to reliably detect the use of common image manipulation techniques.

A lot of these detection techniques leverage statistical inconsistencies introduced in the tampered region relative to the rest of the image. While this strategy works for identifying local manipulation as in the case of image splicing and copy-move forgeries, it fails in the face of content wholly generated by deep neural networks such as generative adversarial networks (GANs) [1] . As a result, we have to resort to widely different techniques to be able to distinguish between highly realistic GAN-generated images and images from the real world. A seminal paper by Marra *et al.* [3] claims that GANs leave fingerprints in the pixel domain on the content they generate which can not only help us label an image as GAN-generated but also let us identify the architecture of the GAN it came from.

This appears to be similar to the photo-response non-uniformity (PRNU) patterns that cameras have been found to leave on photographs and which have been successfully used for image attribution [2]. However, while the roots of the PRNU patterns can be traced to manufacturing imperfections, the GAN fingerprints appear to be the artifacts of the complex linear and non-linear GAN layers and dataset biases.

This project aims to be a study in the reliability and integrity of GAN fingerprints and will hopefully give us some insight into the potential of using them for forensic purposes. To this end, I will implement and train a set of GANs with different architectural components on a publicly available dataset and use the images generated by them to examine the robustness of the baseline fingerprint-detection algorithm outlined by Marra *et al.* [3]. I will first study how it holds up in the face of common adversarial and non-adversarial perturbations like JPEG compression, cropping, flipping etc. Besides that, I will also investigate its resilience to specially crafted 'white-box' attacks. If time and resources permit, I will also investigate the robustness of two other fingerprint detection and attribution algorithms [4] [5].

Starting Point

GANs

My present knowledge of neural networks is limited to the content on backpropagation that was covered in Part IB Artificial Intelligence. I have only very basic theoretical knowledge of GANs which comes from the papers and articles I read about them during the long vacation.

Python and PyTorch

I plan to implement the project in Python, a decision that mainly stems from the extensive support for scientific computing that Python provides with its mathematical libraries. The first time I used Python was during the Part 1A Scientific Computing Practical Course which involved some basic usage of NumPy and matplotlib. Besides that, I have only ever used Python for writing small toy programs and scripts for parsing and editing files. I have zero experience with PyTorch but I hope to quickly pick it up through the extensive tutorials available online during the first few weeks of the project.

Step-by-step breakdown of work to be done

1. **Preparation and Research** : I will need to thoroughly read up on and understand the research that has been conducted in the area of GAN fingerprinting. It, in itself, is not a very heavily-researched topic and so I also plan to adapt methods used in the related field of camera fingerprinting with PRNU patterns to my project.
2. **Training GANs and Generating Images** : I will be using a mixture of self-trained GANs and pre-trained GANs available publicly on the internet. Each GAN in this set will be used to generate two disjoint sets of images, one to extract the corresponding GAN's fingerprint from and the other for testing the performance of the algorithm by Marra *et al.* that I will be implementing.
3. **Extracting Fingerprints** : I will then proceed to approximate the fingerprints of the selected GANs from the images that they generate. This will involve picking a suitable denoising filter which also opens up the option of comparing the performance and robustness of the fingerprint detection algorithm depending on the choice of denoising filter.
4. **Implementing Fingerprint Detection and Attribution Algorithm** : I will then go on to implement the fingerprint detection algorithm outlined by Marra *et al.* and evaluate its performance at correctly classifying (by attributing) an image to the GAN that produced it.
5. **Investigating Robustness to Popular Image Manipulation Techniques** : I will then test the robustness of the algorithm to some simple image perturbations such as JPEG compression, blurring, flipping, changing contrast etc.
6. **Investigating Robustness to White-Box Attacks**: I plan to test the robustness of the algorithm to at least two white-box attacks which will be structured roughly as follows :
   (a) *Attack 1* : Additively introduce the fingerprint of GAN B into images generated by GAN A, where A and B are both members of the set of candidate GANs.
   (b) *Attack 2* : Finetune GAN A by training it with images generated by GAN B for a few epochs and observe how the algorithm behaves when asked to attribute images generated by the fine-tuned GAN A (let's call it GAN A'). It will be interesting to note whether the fingerprint of GAN B is able to seep into the distribution learned by GAN A'.
7. **Implementing Extensions**: If time and resources permit, I will investigate the robustness of the frequency-domain approach [4] as well as that of the fingerprint-learning classifier approach [5] to GAN fingerprinting.

8. **Writing the Dissertation**: The last part of the project will involve collating the results from my experiments and deciding on suitable visual representations for them. I will then put everything together into a well-structured dissertation.

Criteria for Success

1. I successfully implement and train a few GANs with different architectures on the public dataset of choice.
2. The algorithm proposed by Marra *et al.* is implemented correctly and results similar to those in the paper are obtained.
3. The robustness of the algorithm is investigated thoroughly against a wide range of image perturbations and attacks.

Possible evaluation metrics

If the Marra *et al.* algorithm is implemented correctly, we should expect to see significantly higher correlation between the residual of an image and the fingerprint of the GAN which generated it compared to the correlation of the residual with the fingerprint of any of the other candidate GANs.

To quantify its robustness, I will measure its accuracy and precision at classifying images (via attribution to the source GAN) before and after they are subjected to the aforementioned perturbations and attacks. If the attacks are implemented correctly, I expect to see a drop in these metrics in the latter case.

Possible extensions

I would like to investigate the robustness of two other approaches to GAN fingerprinting which have been outlined below.

1. Joslin and Hao [4] proposed extracting fingerprints in the frequency domain using FFT which differs from Marra *et al.*'s pixel domain based approach.
2. Yu *et al.* [5] built upon Marra *et al.*'s work to propose a classifier which learns GAN fingerprints and uses those to classify an image as real-world or GAN-generated.

If time and resources permit, I would like to implement these methods and measure their performance relative to the baseline Marra *et al.* algorithm as well as their robustness to the aforementioned perturbations and attacks.

Timetable and Milestones

I am dividing my work into two-week packets with a well-defined milestone at the end of each to enable me to keep track of my progress and make adjustments to my timetable as necessary.

Weeks 1 & 2 (22nd October 2020 - 4th November 2020)

1. Read up on related work in the field. In particular, I will make sure I thoroughly understand [3] as it outlines one of the algorithms central to the project.
2. Familiarize myself with neural network terminology, Google Colab and PyTorch by training a few toy networks.
3. Link the Colab notebooks I am going to be using to Github for backup and version control.

Milestone: *Summarize the algorithm in [3] as a LaTeX write-up. Be comfortable with any technology and tools I will be making use of in this project.*

Weeks 3 & 4 (5th November 2020 - 18th November 2020)

Will potentially be busy with unit of assessment practicals

1. Decide on a publicly available dataset that I'll use for training the GANs.
2. Look for existing implementations of GANs which have been trained on that dataset as well as train a few GANs myself. Choose from these to craft the set of candidate GANs

Milestone: *Have at least half of the complete set of candidate GANs ready.*

Weeks 5 & 6 (19th November 2020 - 2nd December 2020)

Will potentially be busy with unit of assessment practicals + assignment

1. Finish training all the GANs.

Milestone: *All GANs are trained.*

Weeks 7 & 8 (3rd December 2020 - 16th December 2020)

1. Decide on the denoising filters (for fingerprint extraction) I'm going to be evaluating performance of the fingerprint detection algorithm across.
2. Extract the fingerprint of each GAN from the images generated by it. Do this for all the denoising filters that I have chosen.

Milestone: *The fingerprint extraction part of the algorithm is implemented.*

Weeks 9 & 10 (17th December 2020 - 30th December 2020)

1. Implement the fingerprint detection algorithm and evaluate its accuracy at attributing images to their source GANs.
2. Look at other relevant statistics for measuring performance.

Milestone: *The fingerprint detection part of the algorithm is fully implemented.*

Weeks 11 & 12 (31st December 2020 - 13th January 2021)

1. Implement the white-box attacks.

2. Evaluate performance of the algorithm when subjected to image perturbations and attacks.

Milestone: *Attacks implemented and the effect of them on the Marra et al. algorithm evaluated.*

Weeks 13 & 14 (14$^{\text{th}}$ January 2021 - 27$^{\text{th}}$ January 2021)

Slack time to ensure core deliverables are ready

1. Start working on progress report and progress report presentation.

Milestone: *Success criteria for project met.*

Weeks 15 & 16 (28$^{\text{th}}$ January 2021 - 10$^{\text{th}}$ February 2021)

Progress report deadline : 5th February 2021

1. Finish and submit the Progress Report.
2. Start working on extension 1 i.e. the frequency domain approach to GAN fingerprinting.
   (a) Read the paper by Joslin and Hao [4].
   (b) Implement the fingerprint detection and attribution algorithm outlined in the above-mentioned paper.

Milestone *Summarize the frequency-domain fingerprint detection algorithm as a LaTeX write-up. Finish implementing it.*

Weeks 17 & 18 (11$^{\text{th}}$ February 2021 - 24$^{\text{th}}$ February 2021)

1. Investigate the robustness of the Joslin and Hao algorithm to image perturbations and attacks.
2. Start working on extension 2 i.e. a classifier which can learn the fingerprints.
   (a) Read the paper by Yu *et al.* [5].
   (b) Start implementing the classifier as proposed in the above-mentioned paper.
3. Start writing the draft Introduction and Preparation chapters.

Milestone: *Attacks implemented and the effect of them on the Joslin and Hao algorithm evaluated. Progress Report Presentation between 11th Feb 2021 - 16th Feb 2021.*

Weeks 19 & 20 (25$^{\text{th}}$ February 2021 - 10$^{\text{th}}$ March 2021)

Wrap up work on the extensions

1. Finish writing the draft Introduction and Preparation chapters.
2. Finish implementing the classifier.
3. Investigate the robustness of the classifier to image perturbations and attacks.

Milestone: *Draft Introduction and Preparation chapters ready and sent to supervisor and DoS for feedback. Classifier implemented. Attacks implemented and the effect of them on the classifier evaluated.*

Weeks 21 & 22 (11th March 2021 - 24th March 2021)

Note: *Revise for exams*

1. Incorporate feedback into the draft Introduction and Preparation chapters and finalize it.
2. Write the draft Implementation chapter.
3. Discuss evaluation criteria with supervisor and start working on the Evaluation chapter.

Milestone: *Draft Implementation chapter ready and sent to supervisor and DoS for feedback.*

Weeks 23 & 24 (25th March 2021 - 7th April 2021)

Note: *Revise for exams*

1. Incorporate feedback into the draft Implementation chapter and finalize it.
2. Finish writing the Evaluation chapter.

Milestone: *Draft Evaluation chapter ready and sent to supervisor and DoS for feedback.*

Weeks 25 & 26 (8th April 2021 - 21st April 2021)

Note: *Revise for exams*

1. Incorporate feedback into the draft Evaluation chapter and finalize it.
2. Write the Conclusion chapter.

Milestone: *Draft dissertation ready and sent to supervisor and DoS for feedback.*

Weeks 27 & 28 (29th April 2021 - 12th May 2021)

Dissertation Deadline : 14th May 2021

Note: *Revise for exams*

1. Incorporate final feedback into the dissertation.

Milestone: *Dissertation submitted!*

Resources Declaration

I will primarily be using my personal laptop – MacBook Air 2017 (1.8 GHz Intel Core i5, 8 GB 1600 MHz DDR3) – for all work related to the project. I will regularly push my code to Github for back up as well as version control. I will also take monthly backups of both my code as well as my dissertation, which will be written in Overleaf, on an external Seagate hard-drive. I will be able to continue doing my work on an MCS machine in case

my personal laptop fails in any way. I accept full responsibility for my machine and I have made contingency plans to protect myself against hardware and/or software failure.

The free version of Google Colab will hopefully be sufficient for computational purposes. If need be, I will request access to one of the two supercomputers (Peta4 and Wilkes2) provided by the UIS.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems 27 (NIPS 2014).*

[2] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukáš. Determining Image Origin and Integrity Using Sensor Noise. *IEEE Transactions on Information Security and Forensics, pp. 74-90, March 2008.*

[3] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do GANs leave artificial fingerprints? *2019 IEEE Conference on Multimedia Information Processing and Retrieval.*

[4] Matthew Joslin and Shuang Hao. Attributing and Detecting Fake Images Generated by Known GANs. *3rd Deep Learning and Security Workshop, co-located with the 41st IEEE Symposium on Security and Privacy, San Francisco, CA, May 2020.*

[5] Ning Yu, Larry Davis, Mario Fritz. Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints. *2019 International Conference on Computer Vision.*