Forensic Analysis of GAN Training and Generation: Output Artifacts Assessment of Circles and Lines

Stefan Seidlitz, Jana Dittmann
Otto-von-Guericke University, Magdeburg, Germany
stefan.seidlitz@ovgu.de

The Eighteenth International Conference on Emerging Security Information, Systems and Technologies
SECURWARE 2024

About the presenter

Stefan Seidlitz
stefan.seidlitz@ovgu.de
Otto-von-Guericke University, Magdeburg, Germany

- Research assistant at Advanced Multimedia and Security Lab (AMSL) at Otto-von-Guericke University Magdeburg (OvGU)
- 2019: received his masters degree in Computer Science at OvGU
- worked in past research projects: “GENSYNTH” and “FAKE-ID”
- currently working on the research project “CySec-II”
- research field: media forensic

Content

- Introduction and Motivation
- Overview over our Concept Pipeline
- Implementation and Evaluation
- Discussion
- Summary, Conclusions and Future Work

Introduction and Motivation

- motivated by the DeepFake case to identify characteristic traces
- forensic analysis by using simplified and well-defined shapes on the example of geometric shapes of circles and lines
- train models with simplified and well-defined shape images to measure and study the output from the generation.
- well-defined training guides the comparison and allows measuring artifacts in the output
- goals:
  - visual human-based assessment
  - first, straight forward automated analysis on the example of 4 circle data sets

Overview over our Concept Pipeline

- Approach is divided into three phases:
  - Training Mode, including the generation of ImageSet with different geometric shapes.
  - Generation Mode using the implementation of StyleGAN3 [2]
  - Test and Comparison Concept

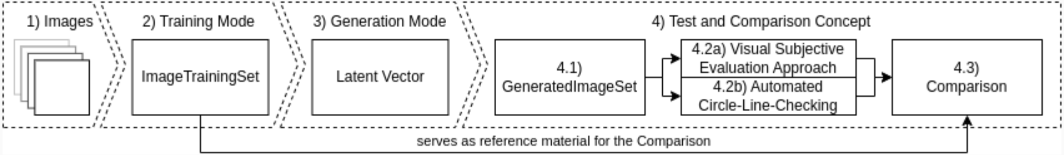


Figure 1: Conceptual Pipeline for our approach, with StyleGAN3 implementation from [2]

Implementation: Training Mode, Introduction of our Data Sets

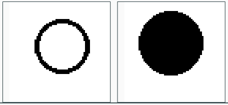
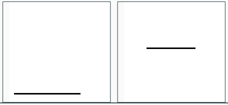
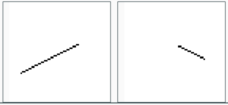
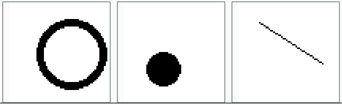
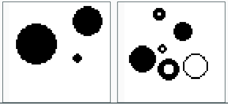
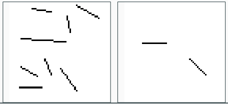
- Creation of six different image data sets
- specific features, which are used for images in all image data sets:
  - same image count: $50,000$
  - same image size: 64×64 pixels
  - image color: homogeneous black for shapes / white for background
  - no intersections between objects and image boundaries
- differentiation between the data sets:
  - single (only one geometric object) and multi (between one and ten geometric objects) data sets
  - two circle data sets (single & multi) with full circles and circled rings
  - three line data sets (single horizontal, single and multi line with random direction)
  - one mixed single data set with 25k lines and 25k circles

Implementation: Training Mode, Introduction of our Data Sets

Table 1: Overview of all created testing data sets for step 1 (compare figure 1)

ID	Type	Content	Image Count	Image Size	Image Color	Number and Type of Shapes per Image
1	single circle	black circles and circled black rings	50.000	64×64	black / white	only one with random size and position, no connection with border
2	multiple circle	black circles and circled black rings	50.000	64×64	black / white	between one and ten with random size and position, no connection between other circles and with border
3	single horizontal line	black horizontal line	50.000	64×64	black / white	only one with random size and position, no connection with border
4	single line	black line	50.000	64×64	black / white	only one with random size, line direction and position, no connection with border
5	multiple lines	black lines	50.000	64×64	black / white	between one and ten with random size, line direction and position, no connection with other lines and with border
6	single circle or line	black circles rings, and lines	circles and rings: 25.000; lines: 25.000	64×64	black / white	only one with random size, direction and position, no connection with border; sub sets are randomized reused from data set 1 and 4

Table 2: Example images for all data sets from table 1, see step 1 of figure 1

single circles and circled rings; ID 1	single horizontal line; ID 3	single line with random direction; ID 4
		
single circle and single line with random direction; ID 6	multi circle; ID 2	multi line; ID 5
		

all Data sets are available at [4]

Implementation: Training Mode, Preparation of the StyleGAN3 Training

- using the default configuration of StyleGAN3 from Karras et al. [2, 3]
- every data set was trained for two final models:
  - same training configuration for both models
  - only the snap-parameter was changed to get more frequently model snapshots
  - Reason for this procedure:
    1. Reproducibility: Are the observations in both models the same?
    2. Get a finer view to the training process of the current model.

all final models are available at [4]

Table 3: Used training configuration for stylegan3

Required		
outdir	*path to the output directory*	
cfg	stylegan3-t	
data	*path to the training data set*	
GPUs	1	(proposed values from the Readme file for the training parameter)
batch	32	
gamma	8.2	
Optional features		
mirror	1	
all other parameters are not set, either the default configurations were used or the specific parameter was not used here		
Misc hyperparameters		
those parameters are not set, either the default configurations were used or the specific parameter was not used here		
Misc settings		
kimg	25000 (default value)	
snap	50 (default value) or 10	
all other parameters are not set, either the default configurations were used or the specific parameter was not used here		

Implementation: Training Mode, Training Intentions

Table 4: Training intentions, every model is trained with the configuration of table 8, only the snapshot sequence was configured witch is given in the column "snapshots" (see step 2 of figure 1)

ID	used data set	snapshots	training intention / expected training behavior
1	multi: circle & rings; data set id: 2	every 200 kimg	Size, shape and color of circles and ring should be similar to the training data set. The generator should create between 1 and 10 objects (circles and/or rings) to emulate images from the data set. No object should have connections with the border of the image or with other objects.
2		every 40 kimg	
3	single: circle & rings; data set id: 1	every 200 kimg	Size, shape and color of circles and ring should be similar to the training data set. The generator should create only 1 object (circle or ring) to emulate images from the data set. No object should have connections with the border of the image.
4		every 40 kimg	
5	single: horizontal lines; data set id: 3	every 200 kimg	Size, shape, alignment and color of lines should be similar to the training data set. The generator should create only 1 horizontal line to emulate images from the data set. No line should have connections with the border of the image.
6		every 40 kimg	
7	single: lines with a random direction; data set id: 4	every 40 kimg	Size, shape, alignment and color of lines should be similar to the training data set. The generator should create only 1 line with an indifferent alignment to emulate images from the data set. No line should have connections with the border of the image.
8		every 200 kimg	
9	multi: lines with a random direction; data set id: 5	every 200 kimg	Size, shape, alignment and color of lines should be similar to the training data set. The generator should create between 1 and 10 lines with an indifferent alignment to emulate images from the data set. No line should have connections with the border of the image or intersections/connections with other lines.
10		every 40 kimg	
11	single: circles, rings & lines; data set id: 6	every 40 kimg	The behavior of this training test should be similar to the training IDs 3 and 4 in combination to 7 and 8. The influence from specific features of one training set to the other training set is unexpected before the training process starts.
12		every 200 kimg	

Evaluation: subjective inspection of generated geometric shapes

- samples are randomly analyzed over different training iterations over every training approach
- different errors are differentiate between general and specific errors:
  - general errors: present in all images over all training approaches with each data set
  - specific errors: occurs only in specific scenarios

Evaluation: subjective inspection of generated geometric shapes

- general errors:
  - areas of the same visible color (e.g., background as well as geometric objects) are not homogeneous
  - the generator of StyleGAN3 was not able to count the geometric shapes

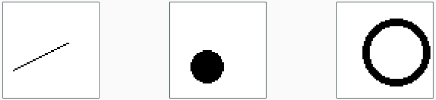


Figure 2: Scaled color scheme on real images using the image processing tool Gimp [5] (which have no effect here)

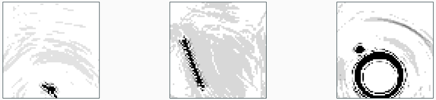


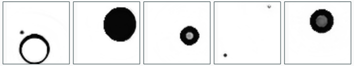
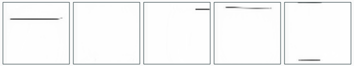
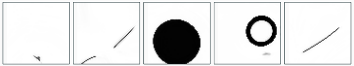
Figure 3: Scaled color scheme on fake images using the image processing tool Gimp [5]

Evaluation: subjective inspection of generated geometric shapes

- observation on circles and rings:
  - no homogeneous geometric area
  - no given symmetry
  - no circle shape
  - objects also in area of border possible
- observation on lines:
  - only horizontal lines appear to be straight with a homogeneous color (but not always)
  - non-horizontal lines are usually not straight lines and have one or two turning points on the line

additional generation scripts are available at [4]

Table 5: Example images for all data sets which are shown in table 1

single circle (circles and circled rings); ID 1	
single horizontal line; ID 3	
single line with random direction; ID 4	
single circle and single line with random direction; ID 6	
multi circle; ID 2	
multi line; ID 5	

Evaluation: subjective inspection of generated geometric shapes

Table 6: Visual observed errors of all trainings (see 4.2a of fig. 1), objective evaluation performed with ID 1-4 and confirm errors

ID	used data set	human observations
1 2	circle and rings (multi); data set id: 2	no homogeneous geometric area, no given symmetry, no circle shape, objects also in area of border possible
3 4	circle and rings (single); data set id: 1	no homogeneous geometric area, no given symmetry, no circle shape, objects also in area of border possible, sometimes more than one object
5 6	lines (horizontal); data set id: 3	lines have mostly the same horizontal direction, pixel values of lines are mostly homogeneous, only at the line border are different pixel values possible
7 8	lines (single, random direction); data set id: 4	non straight lines, partially one to two turning points on the line, smoother transitions due to gray value change on line segments
9 10	lines (multi, random direction); data set id: 5	same visible observation like ID 7 or ID 8; line alignments are similar to the line alignments of the initial data set (compare table 1 ID 5)
11 12	circles, rings and lines (single); data set id: 6	shapes have the same errors which are described for the training runs of ID 3, ID 4, ID 7 and ID 8; a feature transfer (or error transfer) from line to circle and vice versa are not visible

Evaluation: objective evaluation with automated circle detection

- currently in an early state
- real or a fake image decision is made by the examiner (not by detector)
- position of real circles is mostly detectable, size not always
- for fake circles mostly many circles are detected on a given circle

Table 7: Comparison of automated circle detection for real and fake images, red lines highlights the detected circles (see step 4.2b of figure 1)

automated detection on real circles and circled rings);	
automated detection on fake circles and circled rings);	

Discussion

- in real scenarios geometrical shapes are very rare, but also possible
- iris or pupils consists of circled shapes
- lines are less typical in a face

Figure 4: Circle detection on an eye of a real person using the London Face Set [1]

Figure 5: Circle detection on an eye of a fake person generated by StyleGAN2 using the web page `https://thispersondoesnotexist.com/`

Discussion

- introducing of new image data sets with geometric shapes:
    - larger image size with larger geometric shapes
    - other configurations for geometric shapes
- training of other architectures for generative AI
- Use for other applications than DeepFakes

Summary, Conclusions and Future Work

- following the challenge to identify characteristic artifacts on geometric shapes which are a result of its generation process.

- elaborating of an Automated Circle-Checking and enhancing to an Automated Circle-Line-Checking approach
- establishing our approach to other generative AI technologies such as AutoEncoders
- introducing of an improved user based comparison of potential fake and ideal circles
- quantitative evaluation method with the definition of scores based on the overlapping areas of original and reproduces circles.

Thank you for your attention!

Acknowledgments

The work in DeepFakes in this paper is funded in part by the German Federal Ministry of Education and Research (BMBF) under grant number FKZ: 13N15736 (project “Fake-ID”). The research for transparency with Open Source and Open Data conducted within this paper was partly funded by the European Union Project “CyberSecurity-Verbund LSA II” (Grant No.: ZS/2023/182058) - “CyberSecurity-Verbund LSA II – Prävention, Detektion und Reaktion mit Open Source-Perspektiven”.

References

[1] Lisa DeBruine and Benedict Jones. Face Research Lab London Set. 5 2017.

[2] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *CoRR*, abs/2106.12423, 2021.

[3] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. GitHub repository of StyleGAN3. `https://github.com/NVlabs/stylegan3`.

[4] Stefan Seidlitz and Jana Dittmann. Specific Scripts, Models and Data of this Paper. `https://cloud.ovgu.de/s/sBRWzxSLYikp64x`.

[5] The GIMP Development Team. Gimp. `https://www.gimp.org`.