Realistic Image Synthesis and Classification

Differentiating Computer Graphics vs. Photos using CNNs

Wendy Li (liwendy)
Chris Fontas (cfontas)
Emanuel Mendiola (emanuelm)
CS 230

Abstract

As techniques for creating photo realistic imagery evolve, it becomes more and more difficult to differentiate between what is a real photo and what is computer generated. In our project we utilize custom high pass filters and a CNN framework to take CG and natural image inputs and classify them as CG or real. Building on from past projects for photo vs. computer graphic differentiation, we were able to achieve 99.7% accuracy on 100x100 resolution image blocks, better than the current state of the art published results.

Predicting

The aim of our project was to produce a better classifier than past models, such as those proposed by Rahmouni et al[1], and Yao et al[2]. Our model takes in a PNG and performs a binary classification, predicting whether the input image is a CG image or a real photo. Image inputs are broken up into small 100x100 tiles that used a custom pooling layer to extract simple statistical features from each tile which are then used by a CNN framework to differentiate CG from photographic images.

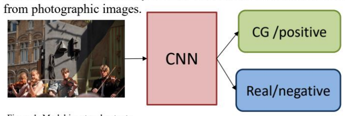


Figure 1: Model input and outputs

Data

Our data set is collected from two sources. Real photos are taken from the RAISE database, a collection of high quality RAW images. CG photos are taken from Level Design Reference Database, and only high quality realistic CG images are picked. All images are between (1000-3000) x (1600-5000) pixels in dimension.

	Number of images	Number of generated 100x100 blocks
Training set	2500	40,000
Validation set	546	n/a
Test set	723	n/a

Table 1: Data set information

Figure 2: Sample dataset images (real photo on left, CG image on right)

References

[1] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In Information Forensics and Security (WIFS), 2017 IEEE Workshop on, pages 1–6. IEEE, 2017.
[2] Ye Yao, Weitong Hu, Wei Zhang, Ting Wu, and Yun-Qing Shi. Distinguishing computer-generated graphics from natural images based on sensor pattern noise and deep learning. Sensors, 18(4):1296, 2018.
[3] Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; Webb, R.; Learning from Simulated and Unsupervised Images through Adversarial Training

Model

-1	+2	-2	+2	-1
+2	-6	+8	-6	+2
-2	+8	-12	+8	-2
+2	-6	+8	-6	+2
-1	+2	-2	+2	-1

(a) SQUARE 5 × 5

0	0	0	0	0
0	0	-1	0	0
0	-1	+4	-1	0
0	0	-1	0	0
0	0	0	0	0

(b) EDGE 3 × 3

0	0	0	0	0
0	-1	+2	-1	0
0	+2	-4	+2	0
0	-1	+2	-1	0
0	0	0	0	0

(c) SQUARE 3 × 3

Figure 3: High-pass Filters in layer 3

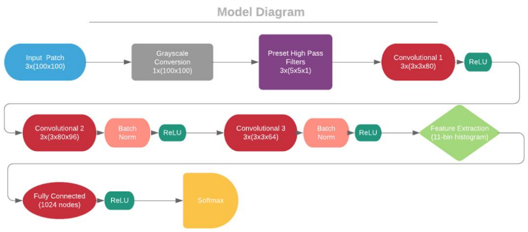


Figure 4: Model Architecture

Our model takes as input 100x100 pixel RGB patches from a larger image that are then converted to grayscale and convolved with 3 hand-crafted high pass filters (a,b,c on left) which suppress low frequency data and bring out high frequency sensor pattern noise.

The image is then passed to 3 convolutional layers that all use a ReLU activation function and the last 2 convolutional layers use Batch Normalization. Finally an 11-bin histogram extracts features for a fully connected layer with 1024 nodes with a ReLU activation that is then input to a softmax layer. Gradient descent is calculated using Adam Optimization.

Results

We compared our results to that of the model implemented by Rahmouni et al and Yao et al. For a fair comparison, we used the same training, validation and test set as both previous models.

Model	Filters	Patch Size	Initial Learning Rate	Learning Update	Dropout	Recall	Precision	Accuracy	F1 Score
Rahmouni	[32, 64]	100	0.001	Constant	0.65	0.98	0.808	0.875	0.8875
Yao	[8,16,32,64,128]	650	0.001	Inverse Decay; Gamma = 0.0001	1.0	1.0	1.0	1.00	1.0
Our Model	[80, 96, 64]	100	0.001	0.0001 after 15000 iterations	0.79	**0.99722**	**0.99722**	**0.99722**	**0.99722**

Table 2: Model Accuracy, Precision, Recall and F1 score comparisons

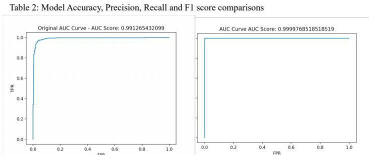


Figure 5: AUC curves (Rahmouni on left, Our model on right)

	Actual CG	Actual Real
Predicted CG	361	1
Predicted Real	1	360

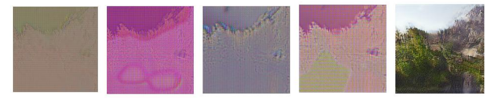
Figure 6 : Confusion matrix generated by our model

Cycle GAN

Source CG Image

Default Cycle GAN discriminator

We were curious to see if our classifier could be an effective discriminator in a GAN. Starting from the full model, we stripped away layers until the GAN converged. The final results were unfortunately, not as convincing as the default discriminator.



Full model — Removed grayscale — Removed high pass filters — Removed simple stats feature extraction and fully connected MLP — Successful convergence using a 3 layer CNN with batch normalization.

Figure 7. Step by step development of cycle GAN discriminator based on our CG vs photo discriminator

Discussion

- Our model has better results in all aspects compared to Rahmouni's et al's model. Additionally, our near 100% F1 score was achieved using only 100x100 image patches, whereas Yao et al required 650x650 patches and a much deeper architecture to achieve the same result, making our approach more computationally efficient.
- Our model is robust and even outperforms human classification accuracy.
- To test for overfitting we gathered a new test set of 245 high-resolution, hyper-realistic CG images and achieved a 96% recall. This was much higher than human performance as many mistook the hyper-realistic CG images as real photographs.

	Actual CG	Actual Real
Predicted CG	236	1
Predicted Real	9	250

Figure 8: Confusion matrix for new test set

Figure 9: Hyper-realistic CG image from new test set

Future Work

1. Generalize to edge case photos (blurry, noisy, night images)
2. Apply classification to videos or images where CGI and real images are combined into a single frame
3. Update our GAN to work on higher resolution images
4. Use our model as the basis for commercial CG quality benchmark testing software