



Fooling Facial Recognition Models

Using Deep Convolutional Generative Adversarial Networks



Abstract

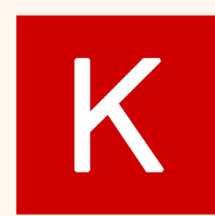
Facial Recognition software is becoming more and more prevalent in our daily lives from our phones to law enforcement. To help prevent the misuse of this kind of software we should further our understanding of these algorithms so they don't remain as mysterious black boxes. This project's goal was to test the accuracy of open-source pre-trained Facial Recognition models using Deep Convolutional Generative Adversarial Networks or DCGANs. The DCGAN was used to generate images from random noise which would then be scored based on what confidence score it can elicit from pre-existing facial recognition models for a specific person's Identity (such as The Rock). This project focuses on generating images that are identified as one of four recognized Identities by the recognition model. The generated test images that had a high confidence score of being a recognized image were subsequently looked at by humans to see if they looked like their classified identities. This will give us insight on how an end user can create fake generated images and still elicit high confidence scores from facial recognition software.

Conclusion

The results of this project show that it is possible to train a Generator for each Identity in the VGGFace2 and achieve a high confidence score. This method is still contingent on having access to the full model. This indicates that open source models such as these can be fooled by potentially malicious actors while other closed-source models may remain safe, but more research would be advised. Future research would best be targeted at replication using smaller generators, testing other learning rates and loss functions, testing this method for Facial Detectors since they are often paired with Facial Recognition⁴, and using this method to improve model accuracy through training with generated false images.

References

- [1] Deep convolutional generative Adversarial Network : TensorFlow Core. TensorFlow. (n.d.). Retrieved May 5, 2022, from <https://www.tensorflow.org/tutorials/generative/dcgan>
 - [2] Refikcanmali(2020) keras_vggface[Oxford VGGFace Implementation using Keras Functional Framework v2+]. <https://github.com/remalli/keras-vggface#projects--blog-posts>
 - [3] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018, May 13). VGGFACE2: A dataset for recognising faces across pose and age. arXiv.org. Retrieved May 5, 2022, from <https://arxiv.org/abs/1710.08092>
 - [4] Brownlee, J. (2020, August 23). How to perform face recognition with VGGFACE2 in Keras. Machine Learning Mastery. Retrieved May 5, 2022, from <https://machinelearningmastery.com/how-to-perform-face-recognition-with-vggface2-convolutional-neural-network-in-keras/>
- Experiment Code Github:
<https://github.com/brtrahms/Fooling-Facial-Recognition-Models-Using-Deep-Convolutional-Generative-Adversarial-Networks>

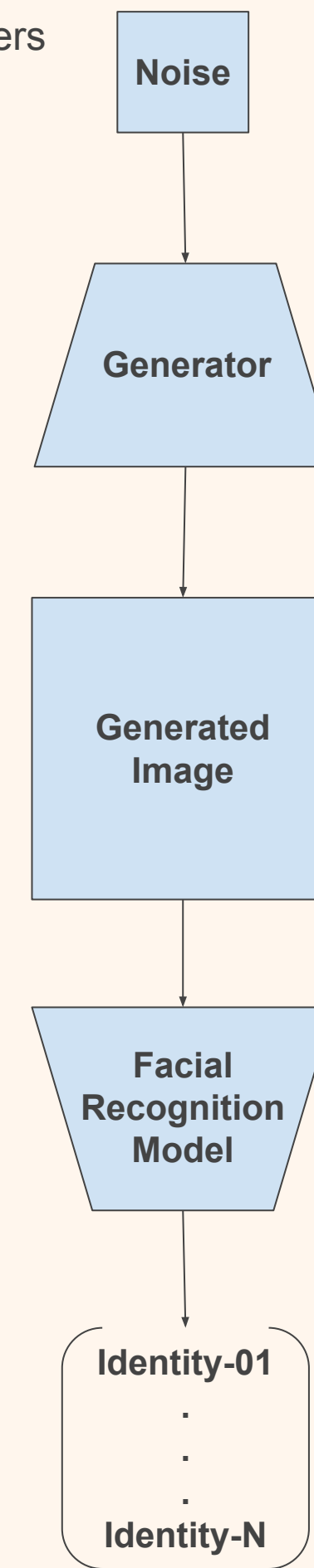


Brandon Trahms
Spring 2022




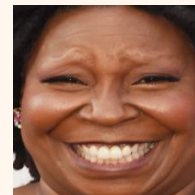




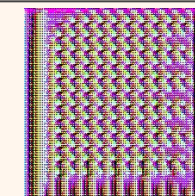
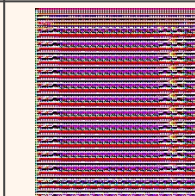
Network Design

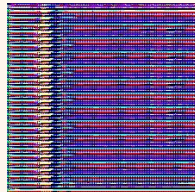
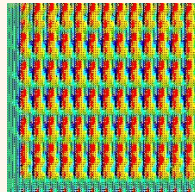
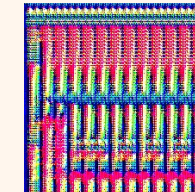
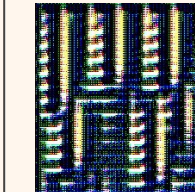
- **Noise** is normally distributed random numbers
- **Generator:** Deep Convolutional Generator¹
 - Input Layer:
 - 100 values
 - Hidden Layers:
 - Fully Connected
 - Transposed Convolution A.K.A Deconvolution
 - Activation Function: LeakyReLU
 - Output Layer:
 - 224 x 224 pixels
 - Activation Function: Sigmoid
- **Facial Recognition Model**
 - VGGFace2 Models³ implemented in Keras²
 - Input preprocessing
 - Convolutional Architectures
 - ResNet50
 - SENet50
 - Outputs confidences on 8631 Identities
- **Training the Generator**
 - Setup in Jupyter using Python
 - Each Generator focuses on 1 Identity
 - Batch size of 5
 - Loss based on how high a confidence the Recognition Model outputs
 - Loss Calculated using Cross Entropy
 - Exponentially Decaying Learning Rate
- **GPU:** NVIDIA GeForce GTX 1660 Ti with Max-Q Design



Generated Results

Control	Channing Tatum	Dwayne Johnson	Simon Cowell	Whoopi Goldberg
				
	Control Image	ResNet50		SENet50
	Channing Tatum	99.325%		99.984%
	Dwayne Johnson	99.926%		99.957%
	Simon Cowell	98.255%		99.227%
Whoopi Goldberg	98.002%		99.568%	

ResNet50	Target Identity	Channing Tatum	Dwayne Johnson	Simon Cowell	Whoopi Goldberg
	Generated Image				
	Confidence	99.81%	99.967%	99.997%	99.394%

SENet50	Target Identity	Channing Tatum	Dwayne Johnson	Simon Cowell	Whoopi Goldberg
	Generated Image				
	Confidence	99.256%	99.999%	99.97%	99.999%

Generator Batch Loss

