# StyleArm: a style-transferring robot arm.

Vernon Stanley Albayeros Duarte

**Abstract**

As the paradigm of human-computer interaction shifts to increasingly intelligent systems that require no direct user inputs to provide services, the way we interact with our machines is still primarily ”active interactions”, as the computer requires some sort of direct user input to provide it's content.

In this master's thesis, we create a physical proof of concept that implements computer vision algorithms to aid in interaction, in the form of a self-built robotic arm with a camera that is able to interface with a Raspberry Pi, a small computer. The proof of concept utilizes a server-client connection with a higher powered machine capable of relaying images treated by a fast Style Transfer neural network in near real-time, switching the context of the style transfer depending on the facial expression of the user. The goal of this robot is to provide the user with a way to interact with computer vision processes they might find useful, like Style Transfer or image stitching for story-telling and publication on social media accounts. Initially, this project meant to optimize a style transfer GAN for use on a Raspberry Pi and provide a completely autonomous project, but we were unable to produce results that could be used in real time.

**Index Terms**

Computer Vision, GAN, Generative Adversarial Network, Style Transfer, Deep Learning, Neural Networks, Face Detection, Raspberry Pi, Robotics

## I. Introduction

HUMAN-COMPUTER interaction has remained somewhat stagnant in terms of how we interact with our computers. Our ”workflow” to interact with them has not had any major changes since modern operative systems were invented. This workflow would be to use some sort of input device, like a mouse or a touchscreen, to demand the computer to provide some sort of content. There have been alternative input methods proposed, but the overall paradigm has never changed. These alternative input methods, in our opinion, never reached full market potential because of one major issue: they require more work from the user, or an alteration of their traditional workflow.

Microsoft's Hololens [1], one of the more recent proposals, is interesting because even though it requires a complete change in the user's interaction pattern with their device, it presents the user with intuitive interaction patterns in the form of ”augmented reality” overlays on their field of view. The way this approach integrates use cases into a new ”input space inspired this master's thesis.

In this Master's Thesis, we aim to produce a physical prototype slash a proof-of-concept, that integrates a few computer vision use cases for simple use by an average user. This project builds and improves upon my own Bachelor's final Thesis project [2].

The aim is to provide a simple gateway for any user to be able to take and save a picture after it has been treated by a Style Transfer neural network for story-telling and publication on social media accounts. An extra layer of interaction is provided by an emotion recognition system to provide relevant styles to the photographs, and the results can be viewed in an attached display. The robot is able to function as a picture frame and takes it's own pictures for the user whenever it's not in use (and has been allowed to).

## II. Introduction

Neural style transfer (NST) methods manipulate images or videos to adapt to the visual characteristics of another image. NSTs is most commonly implemented through deep neural networks. The first publication using neural networks to perform NST was the work of Leon Gatys et al. [3] on 2015. This paper used a VGG19 architecture pre-trained on the ImageNet dataset.

Just a year earlier in 2014, Ian Goodfellow's groundbreaking Generative Adversarial Networks paper [4], proposed a new framework for estimating generative models using two, simpler models. Goodfellow's team was looking for an alternative to the, at the time, state of the art undirected graphical models with latent variables such as restricted Boltzmann Machines [5], [6]. The main problem with this approach at the time, was that it was intractable for all but the most simple problems. The alternatives that did not involve defining a probability distribution explicitly, meaning that they could be trained by back-propagation, were in the form of generative stochastic networks [7]. This approach, however, still required Markov chains for sampling. Goodfellow's proposed Generative Adversarial Networks (GANs from now on), does away with the need of Markov

Author: Vernon Stanley Albayeros Duarte, stanley.albayeros@gmail.com
Advisor: Fernando Vilariño, Computer Vision Centre, Universitat Autònoma de Barcelona
Thesis dissertation submitted: September 2021

chains for sampling, and because GANs do not require feedback loops during generation, they can leverage piecewise linear units, and improve the backprop performance.

This ”family” of networks has, over time, proven useful for NST. Style transfer is a field that manipulates images or videos in order to apply the ”style” or appearance of another image or video. Using GANs for style transfer was first explored by Tero Karras et. al in their paper: A Style-Based Generator Architecture for Generative Adversarial Networks [8]. An example of style transfer can be seen in Figure 1.

While GANs have been proven to be on the bleeding edge regarding performance with NST, use cases with reduced computational capabilities might not take full advantage of them.

Fig. 1: Style transfer examples

## III. Generative Adversarial Networks

Goodfellow’s approach was to use two models to solve the problem. The ”generator” network generates candidates in the form of randomly sampled noise from true data distribution, while the ”discriminator” network evaluates these samples. These two networks play a min-max game where the generator tries to generate ”fake” samples that it thinks the discriminator will think are real, and ”real” samples that it thinks the discriminator will think are fake, while the discriminator tries to ”beat” the generator network by correctly classifying the incoming samples. Over the years after Goodfellow’s original GAN paper, the performance of these networks has improved quite a bit. Alec Radford’s DCGANs [9] proposed the use of convolutional networks in place of fully connected networks, which improved the results and performance of the models. Further improvements for implementing GANs on large scale images such as Andrew Brock’s 2018 [10] paper on the subject, further proved that GANs could be used for larger resolution images.

## IV. StyleGAN

Tero Karras from NVIDIA proposed in 2018 an alternative GAN architecture to implement a style-transfer network. [11]

While the latent space code is provided to the generator network through it’s input layer, Karras chooses to omit the input layer altogether and start from a pre-learned constant.

NVIDIA’s team provides their novel generator with a direct means to generate stochastic detail by using explicit noise inputs, as single-channel images consisting of uncorrelated Gaussian noise. These noise images are broadcasted to all feature maps using the learned per feature scaling factors and then added again to the output of their respective convolutions.

Karras et al. had amazing results using this approach, as seen in Figure 2.

Fig. 2: StyleGAN results

In 2019, NVIDIA's team set out to improve StyleGAN by targeting several of it's characteristic artifacts [11]. In this paper, the team sets out to redesign the generator normalization and regularize the generator to encourage good conditioning in the mapping from latent codes to images.

To remove normalization artifacts, NVIDIA's team identifies an AdaIN operation that normalizes mean and variance of every feature map sepearately, destroying any information found in the magnitudes of the features relative to each other. They instead remove the normalization step from the generator, causing the droplet artifacts to dissapear.

On the other hand, the generator's architecture is revisited completely. Where the original StyleGAN applies bias and noise within the style block, more predictable results are obtained by their modification, which is moving these operations outside of the style block, operating on normalized data. They also remove the application of bias, noise and normalization to the constant input without any observable drawback in performance or quality.

### A. *StyleGAN2*

Later in 2020, Karras' team releases the "StyleGAN2" paper [12]. This time around, Karras and his team set out to improve their StyleGAN to work with limited amounts of data. They propose the implementation of an adaptive discriminator augmentation mechanism that stabilizes training when the dataset used is small. They demonstrate that results are not greatly affected even without modifying the loss functions or the architecture of the network.

We tried implementing a variant of the StyleGAN2 for use in our project initially, but our results proved to be too slow, and too poor compared to Karras' team's release.

## V. Method

In this section, we will describe the used dataset for emotion recognition and style transfer, and the methods used for face recognition. We also propose the methods we think are needed in order to achieve a minimum viable product for our robot.

To achieve a MVP (Minimum Viable Product), and be worthy of a MSc project, we set our goals as the following:

- Remove the need to use Google's Cloud API for all computer vision tasks, implementing local solutions.
- Implement new routines involving engaging computer vision algorithms like Style Transfer.
- Make the robot completely autonomous: no need for an outlet or a server to offload the heavy algorithm computations.
- Achieve a real-time representation of the robot's image processing.
- The user should easily be able to obtain the style transferred files.
- The visual representation on the screen should be at least visually pleasing enough that users will want to keep it on as a background element.

### A. Face detection and emotion recognition

Previously, our pipeline for the robot required sending images to Google's Cloud Services (GCS) for emotion recognition. As we have removed GCS, this is no longer the case. To replace the previous face detection + emotion detection parts of the pipeline, we need to implement local methods face detection and emotion detection based off state of the art methods.

Face detection serves two purposes for our robot: the first is allowing the arm to follow a subject should it approach the limits of the camera's viewing angle. It also serves to improve emotion detection, as described in Gunwan et al. [13], a neural network trained on cropped faces performs optimally on similiarly cropped images. We propose using a Haar-Cascade classifier to detect faces.

Emotion detection will be used to change the context of the style transfer network, loosely adapting the target style transfer to a user's visible emotional status

For our face detection, we have to consider that our robotic arm does not move smoothly, and covers a big arc of movement on every sweep. To counter this, we implement optical flow detection to stabilize frames with a high likelyhood of having a face in them, and only run our face and emotion detection algorithm on those frames, reducing the performance impact of these algorithms. On our previous implementation, we could detect faces at a rate of 1 face detection per 3 seconds given the overhead the program needed to make all the necessary preparations for the GCS-based emotion detection.

TODO

As we are already going optical flow calculations, we can use a "bundle" of similiarly-flowing pixels to detect wether a person has left the frame, and use face detections more sparingly as it would not be necessary.

The arm aims to keep the people in the frame within the middle third of the frame as long as it's movement allows it. If there is more than one person in frame, we keep track of the position of the face belonging to the last person that entered the frame, and use that detection for the centering algorithm. The algorithm is able to keep track of multiple faces in the frame, but we found that movement would be extremely jerky when we tried to keep the camera centered in the middle of the faces, and the field of view is wide enough for this to be unnecessary.

TODO

At this point in the timeline, each frame takes [TIME] seconds to process and be prepared for the next part of the pipeline. This already represents a TIMESX speedup compared to the processing time required per frame on the previous implementation, and it is done completely locally.

TODO

This train of thought proved to be flawed, as we minimally improved the speed of inference of the models, while taking a massive hit on image quality for style transfer.

TODO

## VI. Experiments

- Fast neural Style: pytorch vgg19 experiments.
- Emotion detection: FER
-

## VII. Results

- Face detection performance (time per frame)
- Emotion detection performance (time per frame)
- style change on the fly performance (time per frame)
- generate high resolution style transfer + save+send image performance (seconds)

### A. Preprocessing: face and emotion detection

We measure the performance of our face and emotion detection algorithms by how much time it takes for them to produce a result, as each passing frame affects the quality of the final product.

Towards real-time: detecting face-¿crop-¿emotion detector if face is persistent

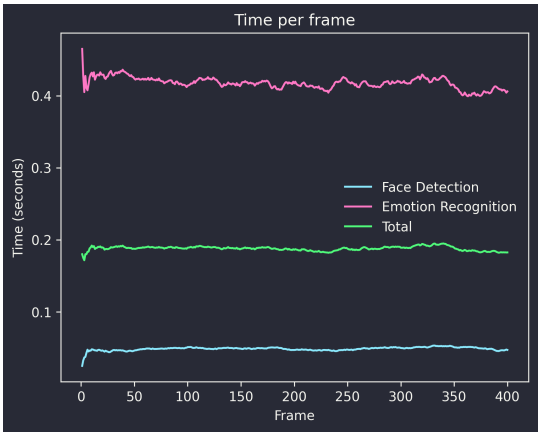
-¿ switch style context if emotion is persistent. Explanation about the performance evaluation procedure and results analysis.

### B. Style transfer

We started by taking Karras' team StyleGAN2 implementation and trying to tweak it to run on the Raspberry Pi 4s hardware.

The purpose of using Karras' model was to try to minimize the requirements for inference of a trained model following their architecture. Our logic was that minimizing the footprint of the model, inference would be fast enough to complete in real time for our robot. To achieve this, we reduced the dimensions of several layers of the model to output images at a 128x128 pixel resoluion, with the intent of upscaling the images ran taken through the inference in real-time to achieve a good framerate, and switch to a larger model when saving an image for the user.


(a) Emotion variation by frame of a sample.




(b) Time needed to process a frame.

Fig. 3: Overall caption

## VIII. Conclusions

Summary about the degree of achievement according to the given problem and the adopted hypothesis; and outline about open research lines...

## Appendix A
## Appendix title

Appendix one text goes here.

## Acknowledgment

The authors would like to thank...

## References

[1] Microsoft, 2021. [Online]. Available: https://www.microsoft.com/en-us/hololens/
[2] S. Albayeros Duarte, "Face-following robot arm with emotion detection," 2019. [Online]. Available: https://ddd.uab.cat/pub/tfg/2019/tfg$_1$51968/$Final.pdf$
[3] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," vol. abs/1508.06576, 2015. [Online]. Available: http://arxiv.org/abs/1508.06576
[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.
[5] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, p. 1527–1554, Jul. 2006. [Online]. Available: https://doi.org/10.1162/neco.2006.18.7.1527
[6] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, 1986.
[7] Y. Bengio, E. Thibodeau-Laufer, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," *CoRR*, vol. abs/1306.1091, 2013. [Online]. Available: http://arxiv.org/abs/1306.1091
[8] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *CoRR*, vol. abs/1812.04948, 2018. [Online]. Available: http://arxiv.org/abs/1812.04948
[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015. [Online]. Available: https://arxiv.org/abs/1511.06434
[10] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," *CoRR*, vol. abs/1809.11096, 2018. [Online]. Available: http://arxiv.org/abs/1809.11096
[11] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *CoRR*, vol. abs/1912.04958, 2019. [Online]. Available: http://arxiv.org/abs/1912.04958
[12] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *CoRR*, vol. abs/2006.06676, 2020. [Online]. Available: https://arxiv.org/abs/2006.06676
[13] T. Gunawan, A. Ashraf, B. Riza, E. Haryanto, R. Rosnelly, M. Kartiwi, and Z. Janin, "Development of video-based emotion recognition using deep learning with google colab," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 18, pp. 2463–2471, 10 2020.