Generative Embedded Mapping Systems for Design

Tom White, Phoebe Zeller, and Hannah Dockerty
School of Design
Victoria University of Wellington
Wellington, New Zealand
{tom.white@vuw.ac.nz, zellerphoe@myvuw.ac.nz, dockerhann@myvuw.ac.nz}

Abstract

We introduce a computational design workflow based on training examples: Generative Embedded Mapping Systems (GEMS). GEMS enable a designer to leverage embedded spaces to provide parametric systems with automatic mappings to future input data. Instead of specifying a rule-based system for mapping data based on predefined attributes, designers provide parameter values across a set of training examples. These training values serve as anchor points bridging a parameter and embedding space. Settings for new data can then be extrapolated from known pairs. A Neural Caricature application is examined as an example implementation of this workflow. The Neural Caricature architecture is more modular, upgradable, and can capture more nuance than systems using more traditional representation based mappings.

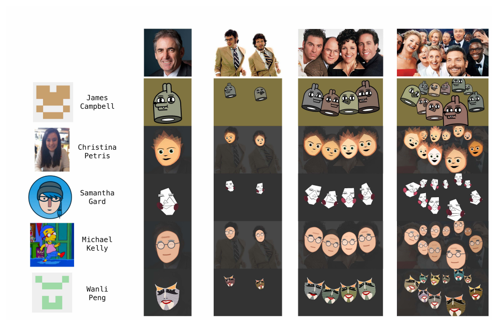

Figure 1: Neural Caricature examples: Matrix of solutions from five different students applied to four images. Each row represents a unique caricature system applied to an input photo containing one, two, four, or ten faces. Some extrinsic settings are provided by facial landmarks, such as position and scale of the face. Other variations are driven from the mapping to the embedded space. In practice, these mappings have been shown to capture qualities such as gender, skin tone, and facial expression. The designer provides settings for sample inputs and need not consider any of these named attributes explicitly.

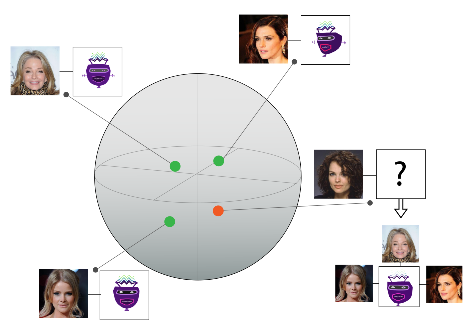

Figure 2: Neural Caricature workflow: Designer creates a parametric system for representing a face (in this case, as a robot). Parameters to the system are provided which match a set of training images that can be paired with vectors in an embedded space (green dots). When the system later encounters new input images, parameters can be inferred by similarly mapping the new image to a vector in the embedded space (red dot) and then interpolating the parameter values from nearest neighbors. The types and quality of inferences available will depend on the nature of the embedded space and the training data provided.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

1 Introduction

Modern graphic design includes the use of algorithms to navigate design spaces, a design practice called computational or parametric design (Madsen 2017). We have developed a workflow for applying settings to a parametric design that balances rule based development of the parametric space with example based exploration of mappings to real world data. We introduce a Neural Caricature system as our first instantiation of this workflow and propose how the system could be generalized to other applications.

2 Neural Caricature

Neural Caricature is a system of replacing faces in images with graphics based on parametric designs. In addition to facial landmarks which provide properties of the face such as position, scale, and orientation, our system utilizes a mapping to a neural embedded space to capture the intrinsic facial qualities such as skin tone, gendered appearance, and emotive expression. The combined effect is demonstrated in Figure 1. More information about our Neural Caricature system is available online.[1]

2.1 Workflow

In the previous era of Artificial Intelligence, it was common to start a problem by addressing the knowledge representation necessary to codify the domain. Similarly, mappings in computational design are often driven by a set of explicit attributes chosen by the designer to serve as adjustable parameters for applying variation. For parametric faces, characteristics might be hair color, nose length, or eye shape.

Neural Caricature breaks this representation based workflow by asking the designer to instead attend to two interlocking tasks: (1) designing a parametric system with a distribution of possibilities (2) specifying a set of specific values in this system which best map to known sample inputs (Figure 2). The designer provides a mapping from a set of training face images to specific matching values. Because these face images can be mapped to neural face embeddings, they provide points of reference for images outside the training set; an explicit ontology of attributes is not required.

2.2 Modularity

In the Neural Caricature system, a designer's work is complete when the generative system has been codified and the parameter values have been provided for a set of training examples. However the architecture is modular, and the initial mapping can improve subsequent to this work without adjusting the generative system or revisiting training values.

One method of improvement is swapping out the entire embedded space. We have tested a number of popular face embeddings including dlib (King 2009), vgg-face (Parkhi 2015), and FaceNet (Schroff 2015). These are orthogonal to the rest of the system, and so can be swapped out in order to capture different sets of qualities or even "upgraded" at a future time as new facial embedding systems are developed - without any modification to the original system. Similarly, the method of performing the mapping across embedded space and parameter space can be changed without altering the designer's original code. In our system, we use interpolation to perform a weighted average across a small set of nearest neighbors. Another option would be to use nearest neighbor alone which would only allow value combinations explicitly considered together at design time.

3 GEMS Architecture

Various structured embedded spaces in machine learning could also serve as flexible knowledge representations for computational design. The GEMS architecture provides a machine learning approach to computational design that offers an example based workflow and software engineering modularity. We are also investigating adapting the GEMS architecture to other domains such as word meaning (Mikolov 2013), product search (Van Gysel 2016), and 3D object shapes (Wu 2016).

[1] `https://vusd.github.io/gems/`

References

King, D. (2009) Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, volume 10, pp. 1755-1758.

Madsen, R. (2017) Programming Design Systems. https://programmingdesignsystems.com/

Parkhi, O.; Vedaldi, A.; and Zisserman A. (2015) Deep Face Recognition *British Machine Vision Conference*

Schroff, F.; Kalenichenko D; and Philbin J. (2015) FaceNet: A Unified Embedding for Face Recognition and Clustering. *ArXiv e-print* https://arxiv.org/abs/1503.03832

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean J. (2013) Distributed representations of words and phrases and their compositionality. *Neural Information Processing Systems*

Van Gysel, C.; de Rijke, M.; and Kanoulas, E. (2016) Learning Latent Vector Spaces for Product Search. *ArXiv e-print* https://arxiv.org/abs/1608.07253

Wu, J.; Zhang, C.; Xue, T., Freeman, W.; and Tenenbaum, J. (2016) Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. *Neural Information Processing Systems*

Acknowledgements

Thanks to the Victoria University of Wellington School of Design undergraduate students for allowing their course work to be featured in this paper: James Campbell, Samantha Gard, Michael Kelly, Wanli Peng and Christina Petris (figure 1); Shane Watters (figure 2).