# Image Feature Learning with a Genetic Programming Autoencoder

Stefano Ruberto
Institute for Biomedical Informatics
Philadelphia, USA
stefano.ruberto@pennmedicine.upenn.edu

Valerio Terragni
Faculty of Informatics
Lugano, Switzerland
valerio.terragni@usi.ch

Jason H. Moore
Institute for Biomedical Informatics
Philadelphia, USA
jhmoore@upenn.edu

## ABSTRACT

Learning features from raw data is an important topic in machine learning. This paper presents a novel GP approach to learn high-level features from 2D images. It is a generative approach that resembles the concept of an autoencoder. Our approach executes multiple GP runs, each run generates a (partial) model that focuses on a particular high-level feature of the training images. Then, it combines the models generated by each run into a parametric function that reconstructs the observed images. We evaluated our approach on the popular MNIST dataset of 2D images representing handwritten digits. Our evaluation results show that our parametric approach can precisely reconstruct the MNIST hand-written digits.

## CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**; **Image representations**; Unsupervised learning;

## KEYWORDS

Genetic Programming, Feature learning, MNIST, Autoencoder

**ACM Reference Format:**
Stefano Ruberto, Valerio Terragni, and Jason H. Moore. 2020. Image Feature Learning with a Genetic Programming Autoencoder. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3389981

## 1 INTRODUCTION

Feature learning is an important topic in machine learning, as it powers many classification and knowledge discovery techniques. Such techniques need numeric representations of raw data (features) that are mathematically and computationally convenient to process. Feature learning becomes a key task when dealing with raw high-dimensional data (e.g., 2D images, videos and sound), which do not have well-defined features.

Recently, we have seen the first GP approaches that learn 2D images features based on the autoencoder method [3–5]. These approaches emulate the behavior of a neural-network *autoencoder*, with forests of GP trees that reconstruct (encode and decode) the pixels of a given image [5]. Thus, following the classical Neural Network (NN) autoencoder architecture with *encoder*→*code*→*decoder*.

*Structured Layered GP* (SLGP in short) [5] by Rodriguez-Coayahuitl et al. splits each image in input into smaller "image portions" to be considered individually. SLGP evolves two sets of populations, one set encodes the image portions outputting the *code* (i.e, latent space), the other set decodes the *code* into the reconstructed image. SLGP generates as many encoding GP trees as the size of *code*, which has to be chosen in advance.

Concurrently with SLGP, McDermott proposed a similar autoencoder GP [4] that relies on two *multi-value linear GP*, one for the *encoder* and one for the *decoder*.

GPMaL [3] by Lensen et al. is a GP *manifold learning* technique, which relates to both SLGP and McDermott's approaches. Manifold learning aims to reduce the dimensions of raw data. This is similar, in principle, to the *encoder* component of most autoencoders, which transforms the input into a lower dimensional code (latent space). GPMaL resembles the *encoder* of SLGP, as it also uses as many GP trees as the number of dimensions of the latent space.

## 2 GP AUTO-ENCODER

This paper presents a Genetic Programming Autoencoder for feature learning on 2D images, but we believe that our approach is general enough to be applied to other types of high-dimensional data, such as videos or sounds.

There are three fundamental differences between our GP autoencoder and the previous ones. First, previous attempts emulate the classical NN autoencoder architecture with three distinct components: *encoder*→*code*→*decoder*. Conversely, we follow the general idea of the classical NN autoencoder, without emulating its internal architecture. As such, we avoid the issue of aligning the encoder and decoder components. Second, previous attempts generate as many decoding GP trees as the number of pixels in the image and as many encoding GP trees as the size of *code*, which has to be chosen in advance. Differently, our autoencoder dynamically adapts the number of high-level features during the evolution. Third, our approach relies on the pixel coordinates (structural information) to learn high-level features, while previous attempts overlook this important information.

The key challenge of using structural information for feature learning is the variability of structural information among images that represent the same concept. For example, when considering the problem of classifying handwritten digits, "1" or "l" are two popular styles for writing the number one. These styles have different albeit similar structural information. A single non-parametric function


*GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico*

ACM ISBN 978-1-4503-7127-8/20/07.
https://doi.org/10.1145/3377929.3389981

cannot output different pixel values for the same coordinate in input, and thus cannot encode both styles.

Our autoencoder accounts for such variability *by parameterizing the GP model $f_{gp}$ so that changing the parameter values reproduces the observed variability*. That is, it allows $f_{gp}$ to output different pixel values for the same coordinate. We define such parameters as the coefficients of a linear combination of multiple GP models.

More specifically, we evolve a series of GP models (called partial models) that learn high-level features by relying on both the values of the pixels and their 2D coordinates. Our autoencoder takes as input a training dataset of images and outputs a model represented as a function $f_{gp}$ that given a 2D coordinate $(c_1, c_2)$ returns the value of a pixel $p$, i.e., $f_{gp}(c_1, c_2) = p$.

Under the hood, we follow the SGP-DT approach [6] that executes multiple GP runs (called external iterations). Each external iteration evolves the partial models driven by a dynamic "*target*" that changes at each iteration. Each target focuses on a particular high-level feature of the training images, and it is defined as the *residual errors* between the previous and current iterations. As such, the next iteration will focus on the characteristics of the images that the previous iteration does not approximate well. Each external iteration will output a partial model that focuses on a specific high-level feature of the images. At the end, the autoencoder creates $f_{gp}$ with a linear combination of the partial models, using *linear scaling* [2] to compute the coefficients of such a combination. Differently from SGP-DT, these coefficients are the parameters of the model and their values will be recomputed for any new image in input. Notably, the residual errors and the corresponding coefficients must be computed following the same order used during the training.

## 3 EXPERIMENTS

**Evaluation Setup.** We performed a preliminary evaluation of our approach on the popular MNIST dataset of 2D images representing handwritten digits [1]. MNIST comprises a training set of 60,000 examples, and a test set of 10,000 examples. Each example is a grayscale numeral bitmap image of 28×28 pixels representing a handwritten digit from 0 to 9. MNIST is widely-used as a standard benchmark in the ML community [1].

We considered a subset of the MNIST training set of 60,000 images to train the $f_{gp}$ models, by sampling 100 images for each of the ten digits. We trained a different $f_{gp}$ model for each of the ten digits by giving in input to our autoencoder the sampled MNIST images corresponding to each digit. Due to the stochasticity nature of GP, we relied on an ensemble methodology that combines the results of multiple models for reconstructing the images. In our experiments, we used ensembles of 50 models for one digit. As such, we obtained 500 models overall (10 digits × 50 ensembles). When reconstructing images, for each of the 28×28 coordinates, we averaged the values of the pixels returned by the 50 ensemble models.

**Evaluation Results.** The matrix in Figure 1 shows five images from the MNIST test set and their reconstructions at various numbers of external iterations (i.e., partial models). Column $N_{\text{ext}}$ shows the images that our approach reconstructs using the linear combination of the first $N_{\text{ext}}$ partial models. With a low value of $N_{\text{ext}}$, the reconstructed images focus on the macro characteristics of the images. For instance, the images of Column $N_{\text{ext}} = 2$ show clouds of dust that resemble the shape of the digits. When $N_{\text{ext}}$ increases, the finer details gradually appear because the autoencoder focuses on the residual errors of previous iterations. The first models focus on the macro characteristics of the image while the successive models take care of the finer details. As Figure 1 exemplifies, the process looks like a progressive cleansing of the images. This is because the residual errors decrease at the increase of external iterations.

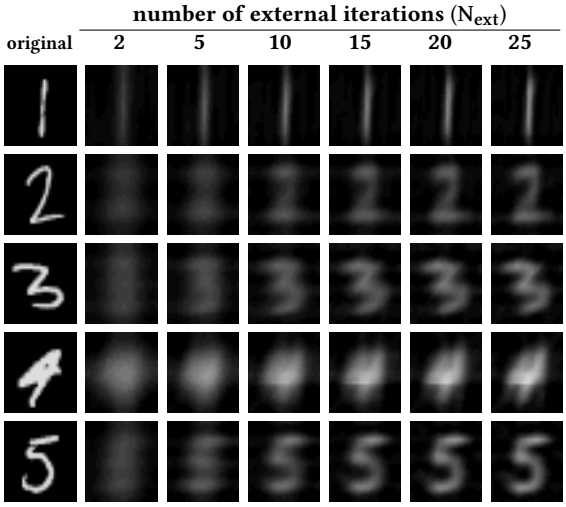


**Figure 1: Examples of test images reconstructed by our autoencoder at the increasing of the external iterations.**

## 4 CONCLUSION

The parametric nature of our GP autoencoder shows promising results. It successfully learns high-level features of digit images of multiple hand-written styles. These are important results, considering that our autoencoder is one of the first GP attempts to learn high-level features from 2D images.

## REFERENCES

[1] Alejandro Baldominos, Yago Saez, and Pedro Isasi. 2019. A Survey of Handwritten Character Recognition with MNIST and EMNIST. *Applied Sciences* 9, 15 (2019), 31–69. https://doi.org/10.3390/app9153169

[2] Maarten Keijzer. 2003. Improving Symbolic Regression with Interval Arithmetic and Linear Scaling. In *Proceedings of the European Conference on Genetic Programming (EuroGP '03)*. 70–82. https://doi.org/10.1007/3-540-36599-0_7

[3] Andrew Lensen, Mengjie Zhang, and Bing Xue. 2020. Multi-Objective Genetic Programming for Manifold Learning: Balancing Quality and Dimensionality. *Genetic Programming and Evolvable Machines (to appear)* (2020). https://arxiv.org/abs/2001.01331

[4] James McDermott. 2019. Why Is Auto-Encoding Difficult for Genetic Programming?. In *Proceedings of the European Conference on Genetic Programming (EuroGP '19)*. 131–145. https://doi.org/10.1007/978-3-030-16670-0_9

[5] Lino Rodriguez-Coayahuitl, Alicia Morales-Reyes, and Hugo Jair Escalante. 2019. Evolving Autoencoding Structures Through Genetic Programming. *Genetic Programming and Evolvable Machines* 20, 3 (2019), 413–440.

[6] Stefano Ruberto, Valerio Terragni, and Jason H. Moore. 2020. SGP-DT: Semantic Genetic Programming Based on Dynamic Targets. In *Proceedings of the European Conference on Genetic Programming (EuroGP '20)*. (to appear) http://arxiv.org/abs/2001.11535.