Publications of the Astronomical Society of the Pacific, 131:108011 (17pp), 2019 October https://doi.org/10.1088/1538-3873/ab213d

Radio Galaxy Zoo: Unsupervised Clustering of Convolutionally Auto-encoded Radio-astronomical Images

Nicholas O. Ralph[1,2], Ray P. Norris[1,2], Gu Fang[1], Laurence A. F. Park[1], Timothy J. Galvin[3], Matthew J. Alger[4,5], Heinz Andernach[6], Chris Lintott[7], Lawrence Rudnick[8], Stanislav Shabala[9], and O. Ivy Wong[10]
[1] Western Sydney University, School of Computing, Engineering and Mathematics Locked Bag 1797, Penrith NSW 2751, Australia
[2] CSIRO Astronomy and Space Science, Australia Telescope National Facility, PO Box 76, Epping, NSW 1710, Australia
[3] CSIRO Astronomy and Space Science Kensington WA 6151, Australia
[4] Research School of A&A, The Australian National University, Canberra, ACT 2611, Australia
[5] Data61, CSIRO, Canberra, ACT 2601, Australia
[6] Departamento de Astronomía, DCNE, Universidad de Guanajuato, Apdo. Postal 144, CP 36000, Guanajuato, Gto., Mexico
[7] Department of Physics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford, OX1 3RH, UK
[8] Minnesota Institute for Astrophysics, University of Minnesota, Minneapolis, MN 55455, USA
[9] University of Tasmania, School of Natural Sciences, Private Bag 37, Hobart, Tasmania 7001, Australia
[10] International Centre for Radio Astronomy (ICRAR), M468, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia
Received 2018 July 13; accepted 2019 May 13; published 2019 September 11

Abstract

This paper demonstrates a novel and efficient unsupervised clustering method with the combination of a self-organizing map (SOM) and a convolutional autoencoder. The rapidly increasing volume of radio-astronomical data has increased demand for machine-learning methods as solutions to classification and outlier detection. Major astronomical discoveries are unplanned and found in the unexpected, making unsupervised machine learning highly desirable by operating without assumptions and labeled training data. Our approach shows SOM training time is drastically reduced and high-level features can be clustered by training on auto-encoded feature vectors instead of raw images. Our results demonstrate this method is capable of accurately separating outliers on a SOM with neighborhood similarity and K-means clustering of radio-astronomical features. We present this method as a powerful new approach to data exploration by providing a detailed understanding of the morphology and relationships of Radio Galaxy Zoo (RGZ) data set image features which can be applied to new radio survey data.

Key words: astronomical databases: miscellaneous – radio continuum: galaxies – methods: data analysis – surveys

Online material: color figures

1. Introduction

Large radio continuum surveys have played a key role in our understanding of the evolution of galaxies (Norris 2017a). Exceptionally large surveys such as Low-frequency Array (LOFAR), Two-meter Sky Survey (LOTSS; Shimwell et al. 2017) and the Evolutionary Map of the Universe (EMU; Norris et al. 2011) are expected to detect 30 million and 70 million radio sources respectively. The sheer scale and complexity of these data sets is pushing researchers towards automated techniques such as machine learning with neural networks (NNs).

NNs are networks of functions termed "neurons" that operate as function approximators. A typical implementation of NNs include the multi-layer perceptron (MLP), as a class of feed-forward NNs with multiple neuron layers. In these MLPs, neuron parameters are typically learned via backpropagation (Dreyfus 1973), where weights are updated using gradient descent of a given loss function as a difference measure between target and predicted output.

A NN trained to classify images with a specific orientation and scale will, however, encounter difficulties when classifying the same training image at an untrained angle or scale (Perantonis & Lisboa 1992). Affine transformations such as rotation, scaling, and translation, are a common cause of machine-learning prediction errors. A classical solution involves augmenting a training set with random rotations and scaling at the cost of training time. Alternatively, a network can be made invariant to scaling by adding convolutional and max-pooling layers. Rotational invariance is more easily solved with the addition of rotated training images.

NNs such as SkyNet (Graff et al. 2014) have accurately classified astronomical data using supervised learning of preclassified examples. Efforts to use these supervised neural networks have been supported with citizen science projects such as the Radio Galaxy Zoo (Banfield et al. 2015), which has created large labeled data sets of radio sources. This RGZ data set has been used to successfully train classifiers for source classification (Wu et al. 2019; Lukic et al. 2018) and radio source host galaxy cross-identification (Alger et al. 2018).

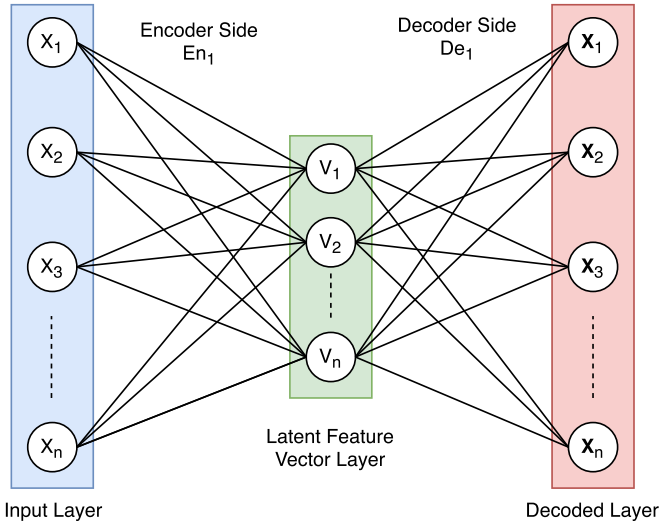

Figure 1. Network configuration of a simple fully connected NN variant autoencoder, featuring the encoder input layer, the downsampled latent feature vector layer, and the reconstructed decoder output layer.
(A color version of this figure is available in the online journal.)

However, this supervised training is not always suitable in outlier detection and separating source complexity as it requires a more complete knowledge of all potential classes of new unseen data. Given that most of the major discoveries in astronomy have been unplanned (Norris 2017b), this is a major shortcoming.

Unsupervised learning techniques bridge this gap by working with no assumptions about input data. An autoencoder is an example of unsupervised learning, designed for dimensionality reduction. Autoencoders work by extracting and compressing the features of input images into a feature vector (Sanger 1989). The ideal autoencoder is trained to perfectly compress and restore input data without loss. The layer configuration of a typical NN autoencoder variant (as shown in Figure 1) uses an MLP architecture with back-propagation learning to reduce input data to a compact feature representation on the encoding side before returning it to its original form on the decoding side. The layer configuration of the encoder and decoder are usually very similar. Autoencoder prediction loss is given as the difference between input data and the decoded output. This loss is naturally an indicator of the performance of the network but is also sensitive to differences between an input image and the training set. Since loss is calculated from the input data, a label set is not required and the network can be trained unsupervised. Autoencoders have seen success in many image processing applications with the addition of convolution, max pooling, and denoising architecture (Xie et al. 2012).

Abstract relationships and topology in large data sets can be interpreted by visualizing auto-encoded feature vectors with dimensionality reduction methods such as Principle Component Analysis (Hotelling 1933, PCA) onto a learning manifold. More complex approaches, such as a self-organizing map (SOM; Kohonen 1997) have been recognized as especially powerful unsupervised data exploration tools in astronomy (Polsterer et al. 2015; Tasdemir & Merényi 2009). By adapting to the shape of encoded latent vectors, these SOMs can display various topological relationships and morphology distributions. Moreover, these algorithms have been augmented to produce labeled classification and source separation with K-means clustering (Lloyd 1982).

This paper demonstrates a novel and efficient unsupervised clustering by combining a self-organizing map with a convolutional autoencoder as a variation of the convolutional neural network (CNN). Using our proposed method, we show that SOM training time is drastically reduced by training on the compressed autoencoder feature vectors of the RGZ image. This method is demonstrated as a powerful data exploration and visualization tool. This approach shows K-means clustering of trained SOM weights as a method of grouping radio-astronomical features in these RGZ images and effectively separating sources by their complexity. We demonstrate our method as an effective and efficient solution to understanding the morphology and relationships of RGZ images that can be applied to unexplored fields for discovery purposes. This approach is in contrast to typical CNN applications in astronomy such as Gravet et al. (2015), which instead create a system for morphological prediction and classification without this element of exploration. The use of abstracted image representations as auto-encoded feature vectors are a significant novel aspect of our method and offer great advantages in computational efficiency compared with this prior work and other CNN implementations such as the system in Dieleman et al. (2015), which also uses random rotation training augmentations for rotational invariance, but are trained on complete images alone.

2. Data

Radio Galaxy Zoo is a citizen science project for radio image classification by volunteers via web interface (Banfield et al. 2015). The majority of the radio image data in Radio Galaxy Zoo comes from the 1.4 GHz Faint Images of the Radio Sky at Twenty Centimetres (FIRST) survey catalog (Becker et al. 1995) version 14 March 2004. FIRST covers over 9000 square degrees of the northern sky down to a 1σ noise level of 150 μJy beam^{-1} at 5″ resolution. We use a total of 80,000 FIRST images from the RGZ Data Release 1 catalog (O. I. Wong 2019, in preparation).

Hand-labeled RGZ annotations of the data set contain the number of components for every resolved source in the image (Banfield et al. 2015). These annotations also include the number of brightness peaks above a set threshold within an

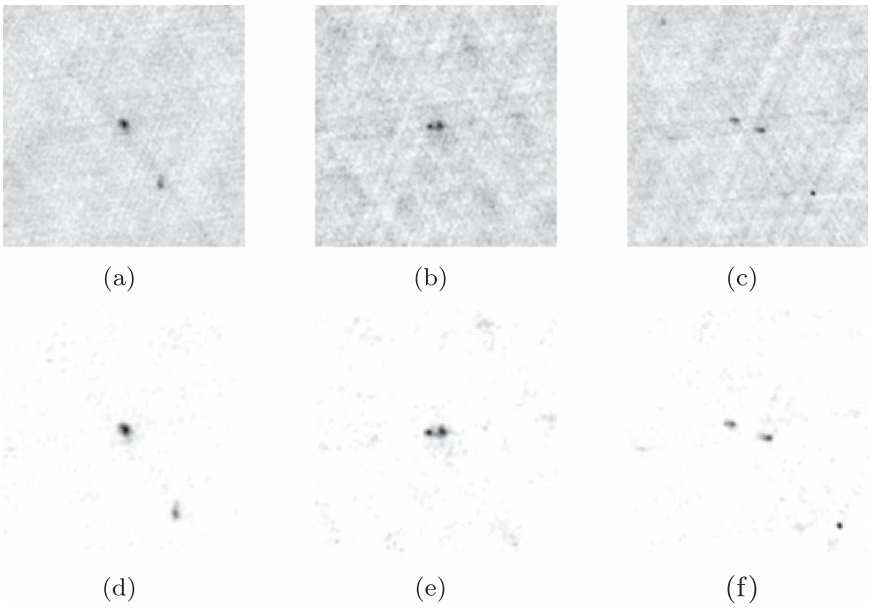

Figure 2. RGZ image preprocessing (a), (b), and (c) as unprocessed FIRST images. Images (d), (e), and (f) show the preprocessing output with noticeable improvement to noise, where a large portion of the background pixels are set to zero.

image. We have encoded these labels as components-peaks, e.g single component with a single peak is 11, two components with two peaks is 22. Table 1 shows that the largest fraction of the data set contains single-component, single-peak sources.

3. Image Preprocessing

Radio images are contaminated by remnants of the instrument's point-spread function (PSF). This contamination is often a major component of the feature space of the RGZ training set. These elements must be removed, as we found that without proper preprocessing, clustering resulted in two classes: "noisy" and "not noisy," distinguished only by intensity distribution. Early preprocessing methods used in this investigation were effective at removing noise but had a tendency to remove faint sources and produce artifacts. As a result, we adopted the preprocessing method of Galvin et al. (2019) with results shown in Figure 2. This approach corrects blank pixels in images at the edge of the FIRST image mosaic, sigma clips and normalizes pixel intensity using the following procedure:

1. Blank pixel regions found in images close to the edge of the FIRST mosaic are corrected. This correction replaces these masked values with a random sample of the mean and standard deviation of valid pixels around the outer-edge region of the image (assuming a normal distribution). These samples are extracted from the outer regions of the image with few astronomical features to properly sample the background noise.

Table 1
RGZ DR1 Classes by Population

RGZ Label	Population Division	Category
11	0.6110	Simple
12	0.1533	Complex
13	0.0153	Complex
14	0.0020	Anomalous
15	0.0003	Anomalous
16	0.0001	Anomalous
22	0.1438	Complex
23	0.0195	Complex
24	0.0028	Anomalous
33	0.0340	Complex
34	0.0053	Anomalous
35	0.0008	Anomalous
36	0.0003	Anomalous
44	0.0068	Anomalous
45	0.0014	Anomalous
46	0.0004	Anomalous
55	0.0020	Anomalous
56	0.0005	Anomalous
57	0.0002	Anomalous
67	0.0002	Anomalous

Note. All point sources (RGZ label 11) are categorized as simple; all sources having more than one component or peak as are categorized as complex; and any source with more than three components or peaks is categorized as anomalous.

2. Noise is removed and background flux is corrected with sigma clipping. This operation subtracts the mean background pixel value and clips all pixel intensities below 1σ to zero.

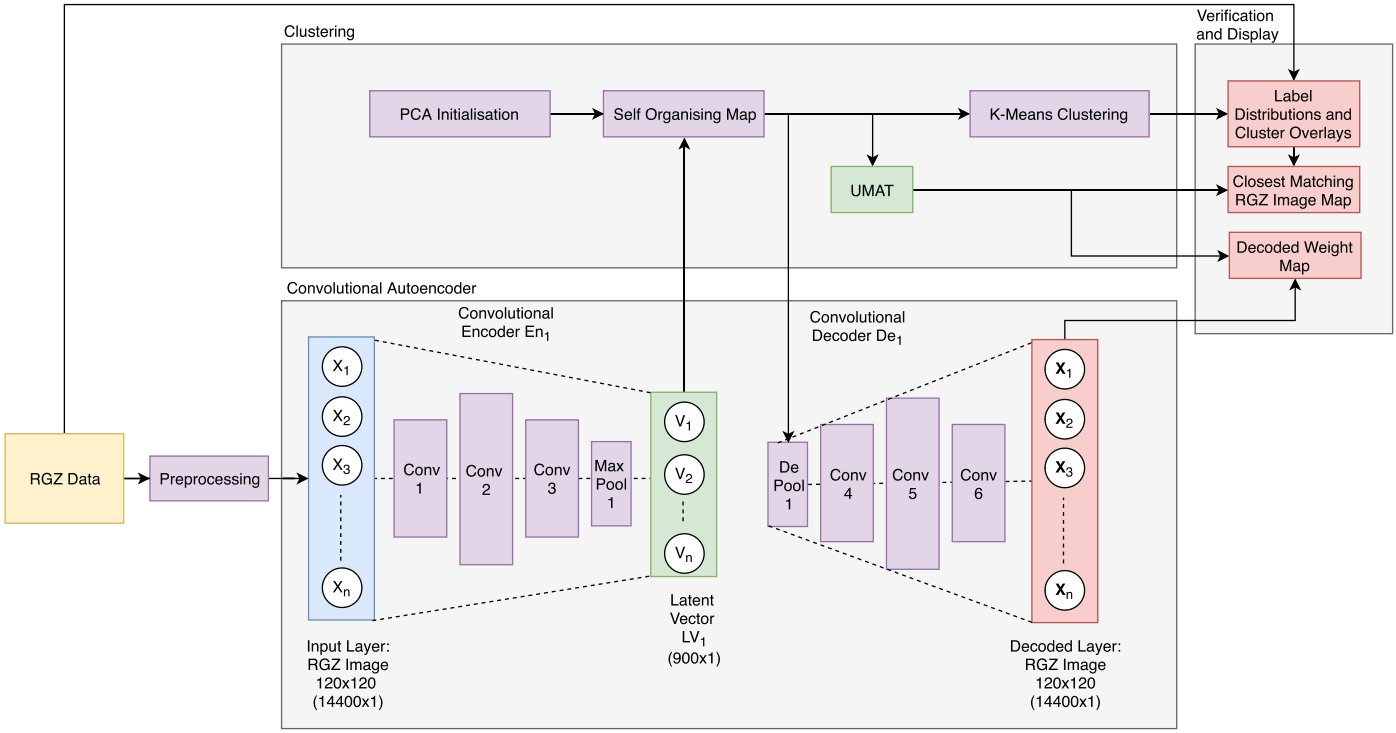

Figure 3. Overall pipeline configuration with the convolutional autoencoder architecture and self-organizing map used in this paper. This network compresses input images with 14400 elements to the latent feature vector with 900 elements for clustering. Encoder and decoder architecture is identical with three convolutional layers at 1, 32, and 1 layers deep, selected experimentally. The SOM weights are initialized using PCA and trained on the encoded latent vectors. Learned SOM weights are reconstructed using the decoder network of the autoencoder to display an approximation of the learned radio-astronomical features. K-means clustering is applied to SOM weights and labeled for verification to compare the map clusters against RGZ labels.
(A color version of this figure is available in the online journal.)

Table 2
Outline of Final Optimized Autoencoder Architecture and Layer Configuration, Where All Convolutional Layers Use LReLU Activation Functions

Network Section	Layer	Function	Input	Filter Size	Stride
Encoder	0	Input	$120 \times 120 \times 1$	⋯	⋯
	1	Convolution 1	$120 \times 120 \times 1$	$5 \times 5 \times 1$	$1 \times 2 \times 2 \times 1$
	2	Convolution 2	$60 \times 60 \times 1$	$5 \times 5 \times 32$	$1 \times 2 \times 2 \times 1$
	3	Convolution 3	$30 \times 30 \times 1$	$5 \times 5 \times 1$	$1 \times 2 \times 2 \times 1$
	4	Max-Pool 1	$30 \times 30 \times 1$	$3 \times 3 \times 1$	$1 \times 1 \times 1 \times 1$
Center	5	Latent Vector	$30 \times 30 \times 1$	⋯	⋯
Decoder	6	De-Pool 1	$30 \times 30 \times 1$	$3 \times 3 \times 1$	$1 \times 1 \times 1 \times 1$
	7	Convolution 4	$30 \times 30 \times 1$	$5 \times 5 \times 1$	$1 \times 2 \times 2 \times 1$
	8	Convolution 5	$60 \times 60 \times 1$	$5 \times 5 \times 32$	$1 \times 2 \times 2 \times 1$
	9	Convolution 6	$120 \times 120 \times 1$	$5 \times 5 \times 1$	$1 \times 2 \times 2 \times 1$
	10	Output	$120 \times 120 \times 1$	⋯	⋯

3. Intensity scaling is applied to normalize the global intensity of each image.
4. All images are additionally cropped for the purposes of this paper, from 132×132 pixels to the center 120×120 pixels to reduce the data set size while preserving salient features.

4. Method

In this section, we outline our method of reducing RGZ images with convolutional autoencoding to a compact feature vector for clustering and visualization using a SOM and K-means clustering. These methods were developed using

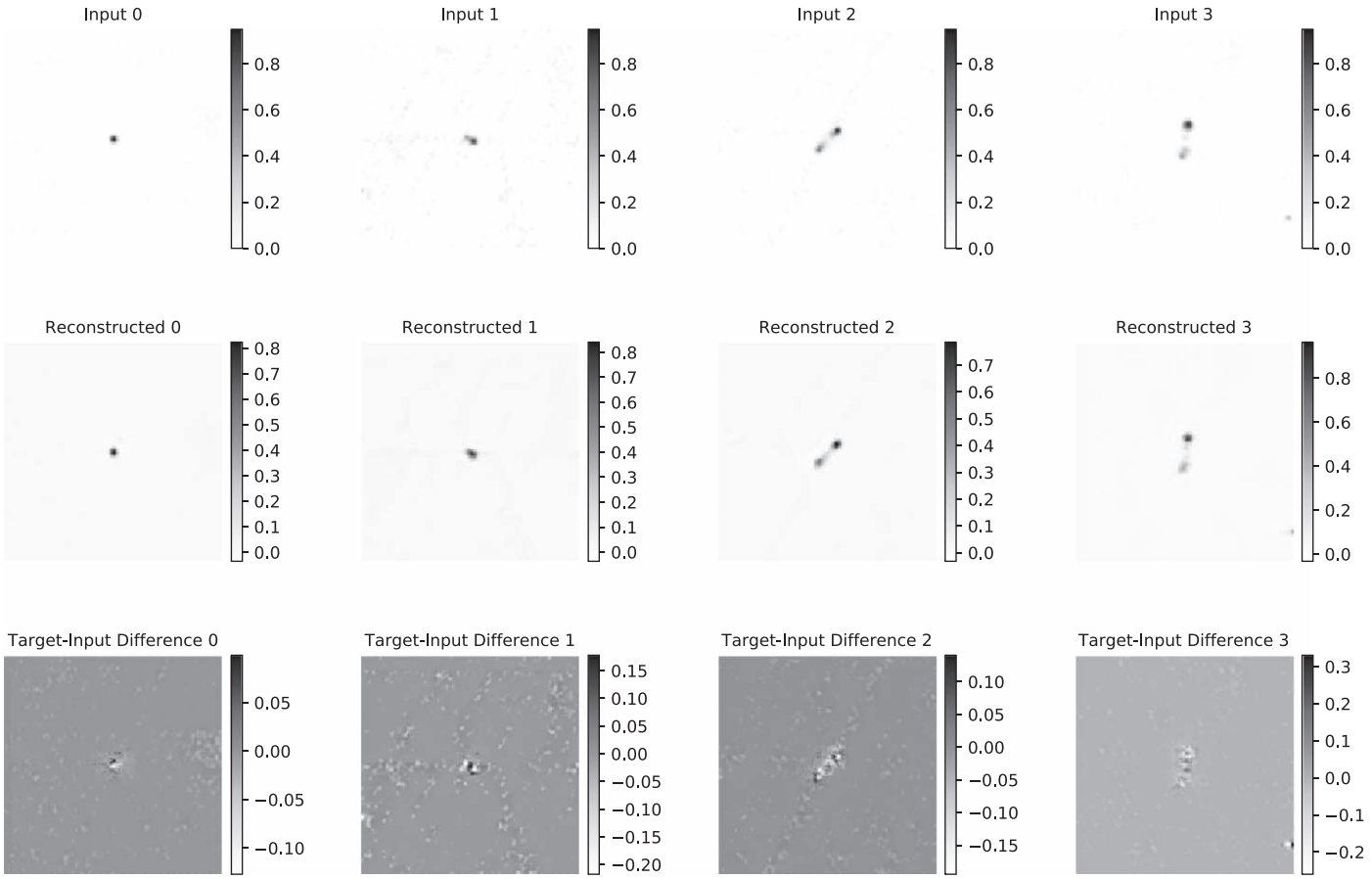

Figure 4. Convolutional autoencoder prediction of RGZ input images after three training epochs. Top row: Original preprocessed input with pixel intensity scale bars, Middle row: trained autoencoder prediction also with pixel intensity scale bars, Bottom row: Difference image between predicted and original image, with scale bars showing the difference in pixel intensity.

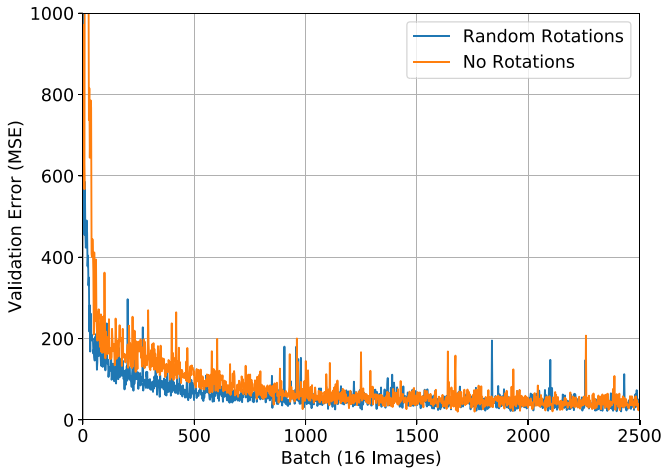

Figure 5. Autoencoder error per batch as mean squared difference between input target image and reconstructed image.
(A color version of this figure is available in the online journal.)

Python with a 24 core Intel(R) Xeon(R) Central Processing Unit (CPU) E5-2650 v4 at 2.20 GHz. We implemented our system using a number of Python machine-learning packages. The Google Tensorflow Machine Learning Library (Abadi et al. 2016) was used to create the autoencoder network, and Somoclu library (Wittek et al. 2013) was used to implement the SOM.

4.1. Affine Invariant Convolutional Autoencoders

In our method, we extract the latent relationships of RGZ image features using a convolutional autoencoder.

We use a convolutional autoencoder with three convolutional layers trained on a random sample of 10,000 images and validated on a separate set of 10,000 images. Table 2 outlines the implemented NN autoencoder with a MLP architecture. The architecture was chosen experimentally through a brief set of trials to determine the best performing configuration. All convolutional layers use the Leaky Rectified Linear Unit

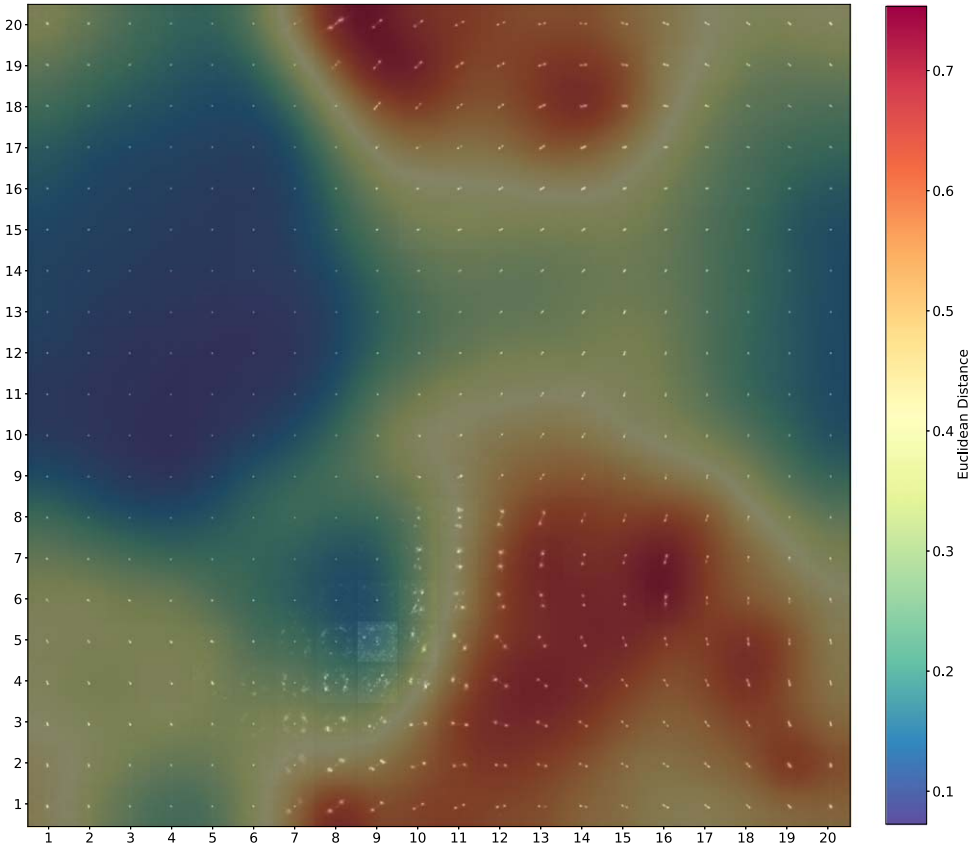

Figure 6. 20 × 20 toroidal SOM UMAT trained using latent vectors produced by an autoencoder without random rotation training augmentations. Each neuron is displaying false color decoded neuron weight images at 120 × 120 pixels.
(A color version of this figure is available in the online journal.)

activation function (LReLU; Maas et al. 2013) given its demonstrated success (LeCun et al. 2015). All activation functions in this network use an activation function slope of 0.2. Additionally, the Adaptive Moment (Adam; Kingma & Ba 2014) optimizer was chosen with a Tensorflow default learning rate of 0.01 due to its considerable use and success as a simple, computationally efficient and effective method in training large networks (Ruder 2016). We use a small batch size of 16 images during training. Small batch sizes in this range have been previously shown to allow autoencoder training to converge on solutions faster than larger batch sizes (Wang et al. 2017)

The encoder output layer here is a max-pooling operation, with the decoder input layer restoring the latent vector to its dimensions before max pooling with a linear interpolator. A latent vector with a 900 × 1 shape is the consequence of the number and dimensions of kernels used in the network. These dimensions can be modified by scaling the input image; however, it was found that training converged quickly with this 900 × 1 latent vector. The dimensions of this vector represent a significant reduction to the original image dimensions (120 × 120, 14400 × 1) while still containing sufficient free parameters to preserve information for decompression with minimum error.

Loss is calculated as the pixel mean square error (MSE) between the autoencoder prediction image and the input image, averaged across the batch. We investigate rotational invariance by also training on images randomly rotated during training. This rotational invariance ideally prevents clustering methods from recognizing rotation as a feature distinguished enough to separate it from its class. As the autoencoder is still being trained on rotation, these features will still be encoded into the latent vectors but with less weight.

4.2. Self-organizing Maps

Self-organizing maps (SOM) are data analysis methods used in unsupervised clustering and data exploration. SOMs create similarity maps or learning manifolds of input data where distinct groups of neurons reflect latent clusters in the data. A SOM models data sets by iteratively updating a grid of neuron weight vectors m_t. This is achieved by moving toward similar data points $x(t)$ on the SOM manifold by refining neurons

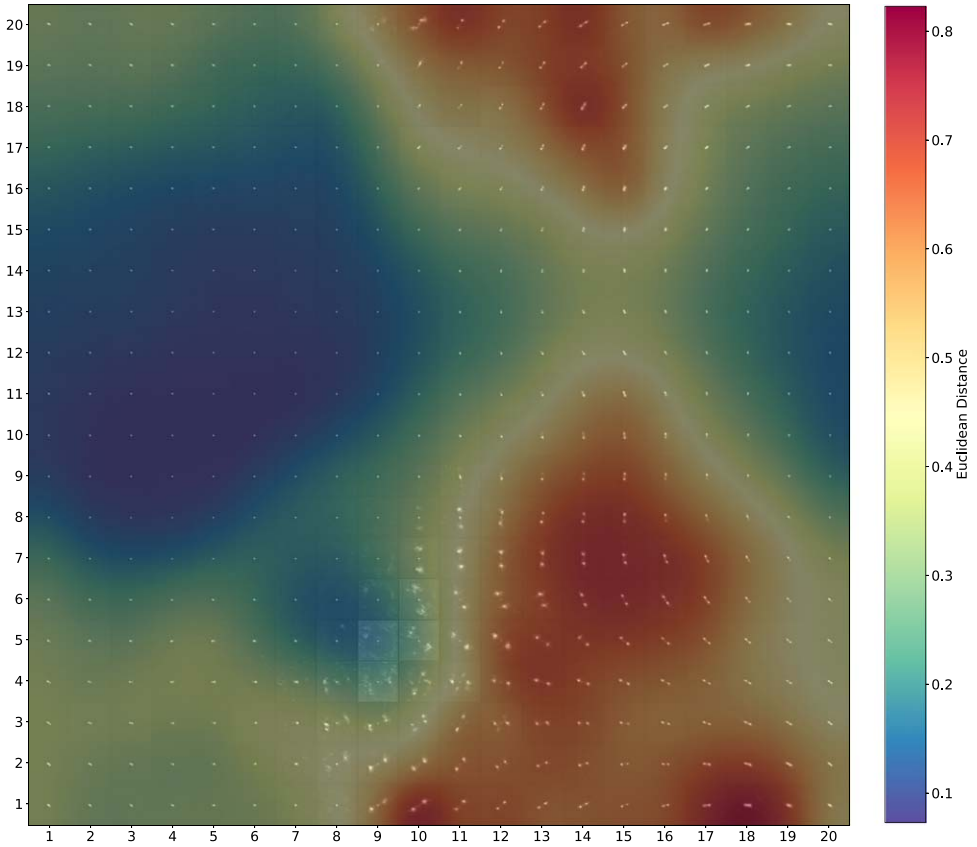

Figure 7. 20 × 20 toroidal SOM UMAT trained using latent vectors produced by an autoencoder with random rotation training augmentations. Each neuron is displaying false color decoded neuron weight images at 120 × 120 pixels.
(A color version of this figure is available in the online journal.)

weights with a neighborhood distance function h_{ci} of each neuron i, by a decaying learning rate α which is balanced to let all neurons stabilize in optimal time, as characterized in Equation (1). A well-trained SOM after m epochs will visualize the distribution of the input RGZ training data as various high-level topological relationships and morphology distributions:

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t)[x(t) - m_i(t)]. \quad (1)$$

We trained our SOM on a random sample of 30,000 autoencoder latent vectors and validated with a separate 30,000 latent vectors. Neither set includes encoded RGZ images used in autoencoder training or validation to ensure the system remains relatively generalizable. This training was conducted with a focus on efficiency and demonstrating reasonable clustering using the following procedure:

1. Initialize SOM grid neurons with a principle component analysis (PCA) learning manifold of the latent feature vector set. This approach allows the SOM to model an already delineated PCA space.
2. Select a random latent feature vector from the training set.
3. Locate best matching unit (BMU) neurons as the "closest" neuron to the selected data point. A common and reliable distance metric used to calculate this is Euclidean distance.
4. Move all neuron weights within the neighborhood toward the data point by updating neuron weights $m_i(t)$ as a function of neighborhood function $h_{ci}(t)$, decay function σ, and learning rate $\alpha(t)$, as shown in Equation (2). This neighborhood function can be represented with several shapes by their radius; namely,

   Linear:

$$h_{ci} = h_{c0}\sigma \quad (2)$$

   and Gaussian:

$$h_{ci} = e^{-\frac{D(n_c, n_i)}{2\sigma^2}}, \quad (3)$$

   where exponential decay is given by

$$\sigma_{\exp} = e^{-\frac{t}{\tau}} \quad (4)$$

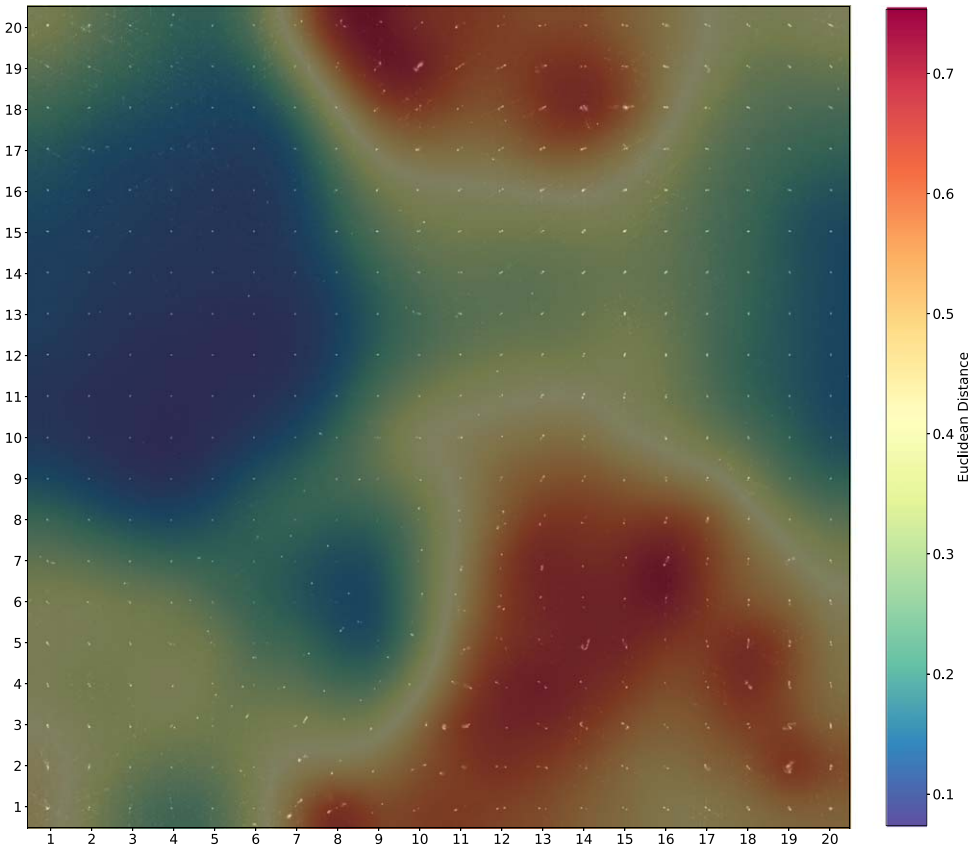

Figure 8. 20 × 20 toroidal SOM UMAT trained using latent vectors produced by an autoencoder without random rotation training augmentations. Each neuron is displaying the RGZ image with the closest matching latent vector transformation to the learned neuron weight.
(A color version of this figure is available in the online journal.)

and linear decay:

$$\sigma_{linear} = -\frac{t}{\tau}, \tag{5}$$

where τ is a decay constant, usually given as the number of training epochs. $D(x, y)$ is given as the distance function (Euclidean distance for the purposes of this paper) between the weight vector of the current excited neuron n_c and weight vector of the winning neuron n_i on the SOM grid position i.

5. Update learning rate and radius based on respective input decay rates. Similar to learning rate decay, this neighborhood decay function can be expressed with an exponentially or linearly decaying σ:

$$\alpha(t) = \alpha(0)\sigma \tag{6}$$

6. Iterate until a training epoch stop condition is met or learning and neighborhood rates have decayed to a limit or zero. In our approach, each iteration of the full data set is considered an epoch as the entire data set is taken into account with no mini-batch training.

The SOM output is a set of learned neuron weight vectors associated to locations on the SOM grid. We interpret these vectors and locations using a unified distance matrix (Ultsch 1993, UMAT). This UMAT is visualized as a heat map of the Euclidean distance between each neuron and its neighborhood. We display the learned weights of each neuron by reconstructing the weight vector as an image with the decoder side of the autoencoder. Given an appropriately trained SOM will contain weights mapped to the latent vector training set, they can ideally be reconstructed into an approximation of the radio-astronomical features encoded into the neuron weight. We display these weights and the RGZ image of the closest matching latent vector on each neuron over the UMAT. Additionally, we assess the ability of the SOM to separate complexity and anomalies by plotting the distribution of the UMAT distance value of each SOM neuron and color coding the closest matching RGZ label and source classification of each neuron.

4.3. K-means Clustering

We segment the SOM unified distance matrix (UMAT) in 4 and 8 clusters using the K-means algorithm (Lloyd 1982). This

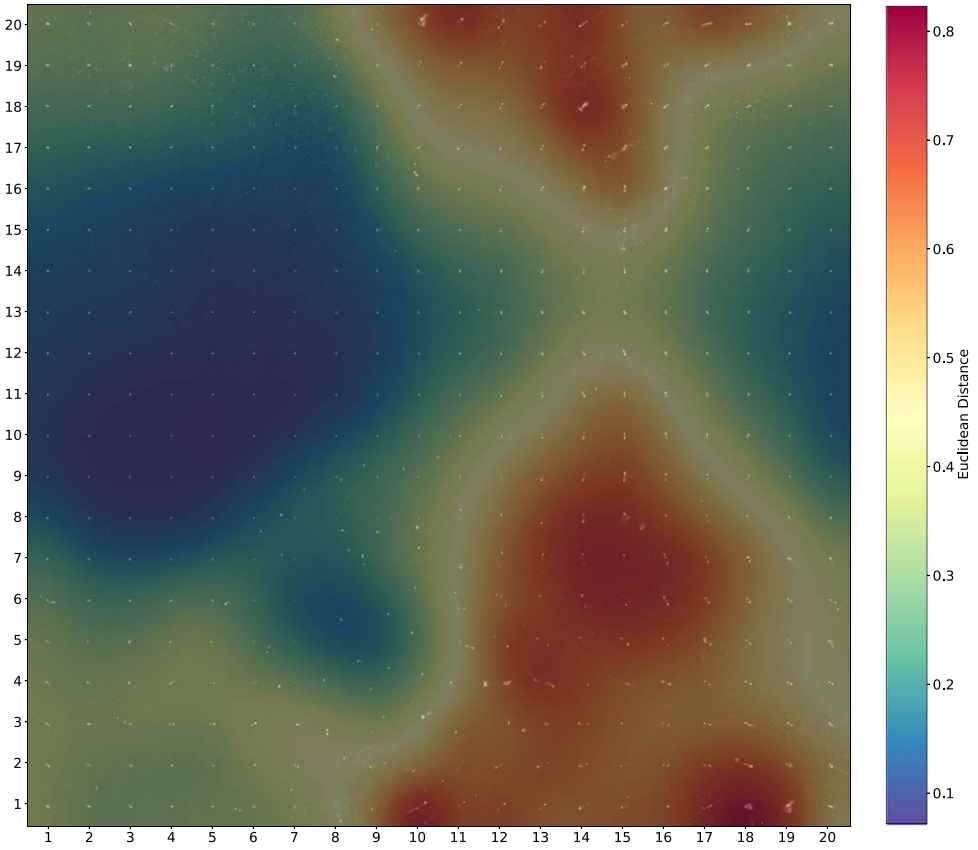

Figure 9. 20 × 20 toroidal SOM UMAT trained using latent vectors produced by an autoencoder with random rotation training augmentations. Each neuron is displaying the RGZ image with the closest matching latent vector transformation to the learned neuron weight.
(A color version of this figure is available in the online journal.)

algorithm groups objects by assigning inputs a cluster based on a metric such as Euclidean distance. This is an iterative process where the distance between each cluster pair is calculated as the average distance of its consistent objects. Input clusters are continually refined based on this distance until the changes in each cluster reach a stop condition. These clusters are discrete, where an object is assigned to only one cluster. We use these clusters as proxies for complex and simple feature vectors on the SOM. K-means clusters of K = 4 and K = 8 clusterings were chosen solely to demonstrate the general clustering ability of the system with the 20 × 20 sized SOM used in this paper. The K-means clustering is conducted on the learned weight vectors of each neuron and was implemented using the Scikit learn package (Pedregosa et al. 2011).

We display K-means clustering results by coloring each neuron on the UMAT grid to indicate its associated cluster Entropy, $\hat{E}$ is also used here as a metric to describe the distribution labels in the matching data samples (images) of each neuron's receptive field:

$$\hat{P}_i = \frac{n_i}{N}, \tag{7}$$

$$\hat{E} = -\sum_i \hat{P}_i \log_2 \hat{P}_i, \tag{8}$$

where n_i is the number of class occurrences i and N the total number of occurring classes. A low entropy indicates good consensus where most images matching a given neuron have the same label. Conversely, a high entropy indicates the matching images of a neuron have a wide range of different labels. We normalize this entropy for clarity to a range from 0 to 1.

The complete system outlined in this method section is shown in Figure 3.

5. Results

This section outlines the results and performance of our approach at each stage of the method.

5.1. Autoencoder Training and Image Reconstruction

The autoencoder trained on RGZ images demonstrates successful compression and decompression across the data set. This is demonstrated in Figure 4 where the reconstructed

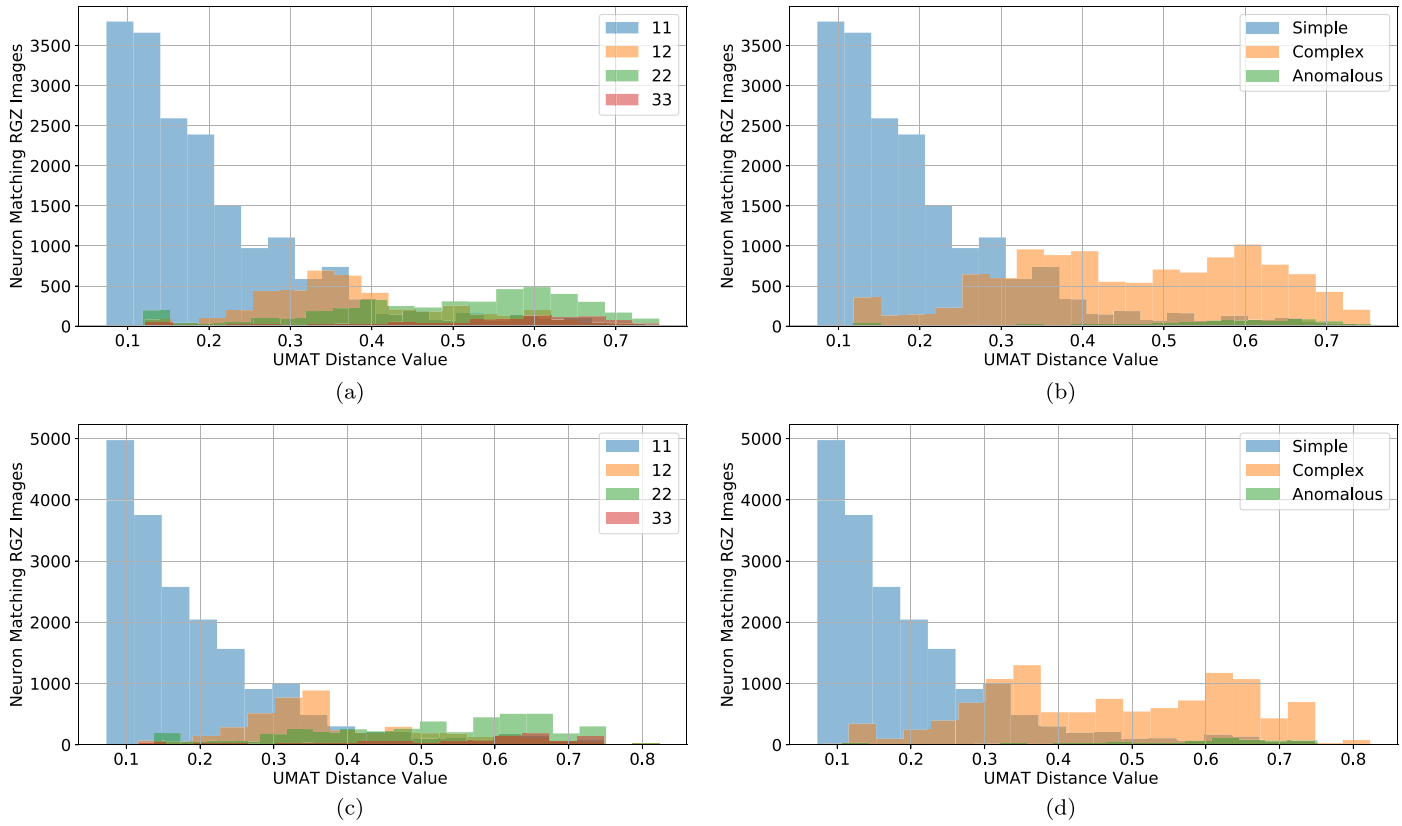

Figure 10. Distribution of neuron UMAT distance value from the 20 × 20 toroidal SOM UMAT trained with RGZ latent vectors produced by an autoencoder with random rotation training augmentations (a) and (b), and without random rotation training augmentations (c) and (d). Color coding in (a) and (d) indicates the RGZ label of the four most dominant labeled RGZ validation images matching each neuron. Color coding in (b) and (e) display the class of the labeled source as simple (RGZ 11), complex (not RGZ 11) and anomalous (RGZ label with more than three peaks or components).
(A color version of this figure is available in the online journal.)

Table 3
Cluster Population and Entropy Statistics for the 20 × 20 Toroidal SOM UMAT with K = 4 Clusters

K-means Cluster	Cluster Population Over Map	Minimum Entropy	Maximum Entropy	Mean Entropy
0	0.278	0.00	1.00	0.46
1	0.460	0.00	0.70	0.23
2	0.118	0.04	0.98	0.52
3	0.145	0.05	0.84	0.43

image strongly approximates the input image, the general morphology and most of the peak and component counts. From this figure, we determine the autoencoder is capable of recognizing and preserving enough key image features to successfully predict the original image from the compressed latent vector. The difference images in the figure show the autoencoder loses most fidelity around the edges of regions and reconstructs background noise with low error. Blurring in the reconstructed image has a square kernel shape and is expected due to the shape of the max pooling and convolutional layers. Autoencoder difference images also show the background noise of each image. Additional layers and training may allow the autoencoder to better generalize the data set image features to remove this blurring and background noise.

Figure 5 indicates that training with random rotation augmentations allows the autoencoder to converge on a solution faster and with slightly lower error than training without random rotation augmentations, despite training on the same training set. Faster convergence with the random rotations is apparent early in training even though the network has not yet been trained on more than one rotation per image. This is likely the result of increased variance in the images. This variance may be caused by the rotation of any remnants of the instrument PSF which are ordinarily oriented across all images. Similar effects have been observed with increased variance in autoencoder training using random noise injection which has been shown to improve autoencoder training (Vincent et al. 2010).

Table 4
Divisions of Clusters Based on the Label of the Closest Matching RGZ Image, Matching Clustered Label Divisions, and Entropy Statistics for Each of the K = 4 Clusters in the 20 × 20 Toroidal SOM UMAT

Cluster ID	RGZ Label	Division of Matching Label In Cluster	Division of All Image Labels In Cluster	Mean Entropy	Minimum Entropy	Maximum Entropy
0	22	0.622	0.742	0.61	0.12	1.00
	11	0.369	0.186	0.21	0.00	0.52
1	11	0.88	0.733	0.21	0.0	0.3
	12	0.12	0.256	0.38	0.2	0.7
2	12	0.511	0.279	0.50	0.15	0.75
	22	0.404	0.204	0.63	0.34	0.98
	11	0.085	0.018	0.21	0.04	0.48
3	12	0.672	0.453	0.52	0.18	0.84
	11	0.241	0.063	0.12	0.05	0.39
	22	0.086	0.054	0.64	0.51	0.78

Table 5
Cluster Population and Entropy Statistics for the 20 × 20 Toroidal SOM UMAT with K = 8 Clusters

K-means Cluster	Cluster Population Over Map	Minimum Entropy	Maximum Entropy	Mean Entropy
0	0.125	0.03	0.84	0.26
1	0.210	0.00	1.00	0.42
2	0.042	0.15	0.74	0.62
3	0.330	0.00	0.31	0.23
4	0.055	0.09	0.82	0.48
5	0.055	0.39	0.69	0.58
6	0.050	0.36	0.98	0.62
7	0.132	0.02	0.78	0.35

The average training time for this autoencoder using random rotation augmentations and with an unaugmented training set is 26.51 and 25.17 seconds per epoch of 10,000 images respectively. Figure 5 suggests that training has converged by the 3rd epoch after the 1500th batch for both training methods. At this epoch, the total training time is 79.53 seconds and 75.51 seconds for the random rotation and normal training conditions respectively. The slight increase in training time between these two training approaches is negligible and likely the consequence of performing the rotation operation on each training image. The time to encode 30,000 RGZ images for SOM training or validation is 252.75 seconds at an average of 0.0084 of a second per image. While the random rotation augmentation results in a total encoding time of 440.25 seconds at an average of 0.015 of a second per image.

5.2. Self-organizing Map of Latent Image Vectors

The self-organizing map (Kohonen 1997) was trained to produce a 20 × 20 neuron toroidal UMAT as displayed in Figures 6–9. This map was created in 25.528 seconds with an average of 2.127 seconds per epoch for 12 epochs. We trained the SOM using a linear learning rate and neighborhood decay function, with a Gaussian neighborhood function. This configuration was chosen as it was the set of training hyperparameters that provided the most accurate modeling of the RGZ images. An initial learning rate of 0.01 was chosen as a default in a manner similar to the selection of the autoencoder learning rate. The initial rate was decayed during training based on the linear decay function toward 0.001. This decay occurred over 12 training epochs, which was found to sufficiently model the training set. These many epochs were found to be sufficient to model the latent vector features and is similar to Geach (2012), which also use a nominal 10 epochs for training.

Both UMATs trained on latent vectors with and without random rotation augmentations are shown with an overlay of the decoded neuron weights and each neuron's closest matching RGZ images. In both training cases, these clearly show the morphology distribution of RGZ image features where the Euclidean distance between the learned weight of each neuron and their neighboring neurons displayed as a heat map. In these tests, the decoded weight map illustrates which relationships and morphologies have been modeled, while the map containing the closest matching images illustrates the real radio-astronomical features that match the neuron weights.

The morphological clusters in these maps are not highly discrete with neurons essentially representing a probability distribution of latent feature vectors. These clustered regions are subclustered by orientation, with similarly oriented objects clustered together with gradual transitions between classes. We expect to see this gradual transition between classes of images given these objects do not have entirely discrete classifications.

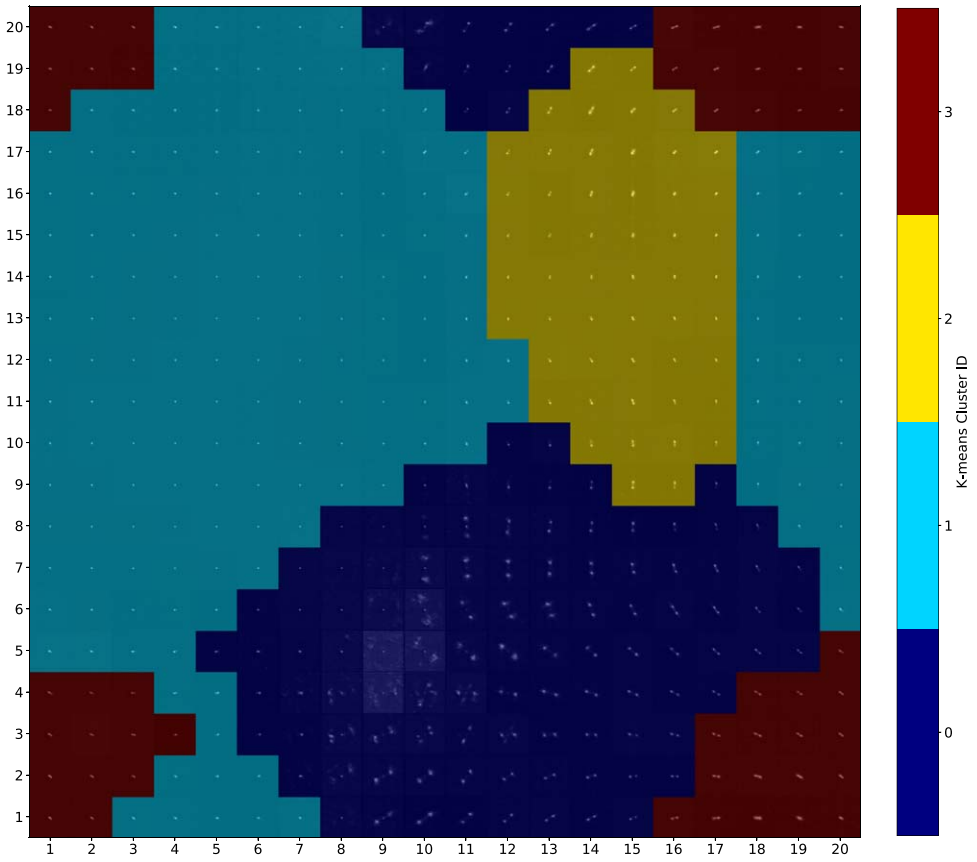

Figure 11. 20 × 20 toroidal SOM, displaying an overlay of decoded neuron weights with four color-coded K-means clusters. (A color version of this figure is available in the online journal.)

In both decoded neuron weight maps, we observe a number of neurons that appear as a rotated average of a central source radio image and extended emission. These morphologies are clear in Figure 6, in neurons of the region 5–3, 6–9, and Figure 7, in a similar region with neurons of the region 7–4, 8–10.

The low-distance regions of the UMATs contain prototypes and decoded neuron weights as compact single sources. Morphologies in this central compact region gradually progress in complexity to compact multi-point sources, sources with globular morphologies, and bent-tail sources toward higher-distance regions. Images placed in high-distance regions have a latent feature vector with a high UMAT distance value to surrounding neighbors, which highlight outliers within the RGZ image set. These results are also illustrated in Figure 10, which show the distribution of UMAT distance values of neurons across the map. This figure shows a clear separation of simple, complex, and anomalous classed sources, in addition to RGZ label 11, 12, 22, and 33, based on the UMAT distance value.

The differences between the results obtained when trained on the rotated augmentation latent vectors and with normal latent vectors are minor, with the random rotations producing slightly more defined peaks in the UMAT distance value distributions in Figure 10. However, it appears that the SOM trained on latent vectors produced by an autoencoder with random rotation contains less prominent average rotation weights. This improvement is likely due to the autoencoder encoding rotation information in the latent vector, which has allowed the SOM to separate these rotations and clean these average rotated weights. Although rotation is not a meaningful radio-astronomical feature, these changes produce an easier to interpret map for both the decoded weight map and the map displaying the closest matching RGZ neuron images. The map trained using latent vectors produced from random rotations in the autoencoder was used for all subsequent tests due to these improvements.

The increased rotational dependency observed in these tests raise questions regarding the true nature of rotational invariance in a SOM. For a neuron to be rotationally invariant, morphological features must be the only feature that is clustered by the system. For this to be the case, genuine rotational invariance would result in all neurons on the SOM being mapped with the same position angle, or for each neuron

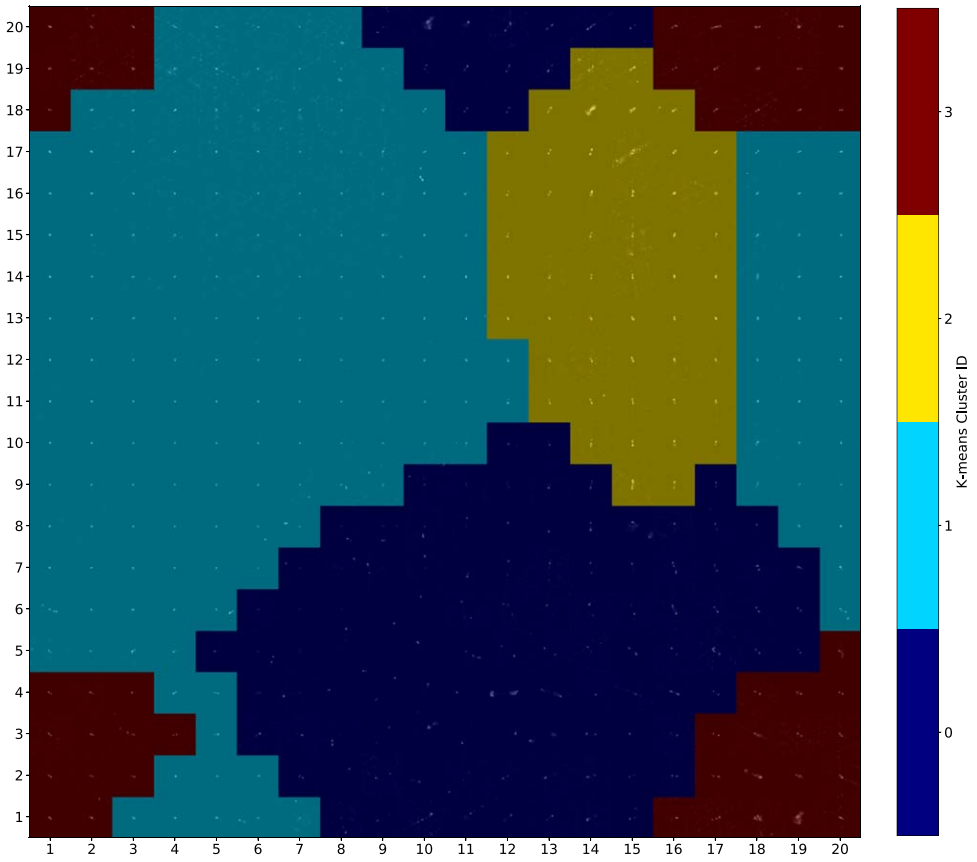

Figure 12. 20 × 20 toroidal SOM, displaying the closest matching RGZ images with four color-coded K-means clusters. (A color version of this figure is available in the online journal.)

to contain all possible position angles, as the observed rotated average morphology seen in decoded neuron images. These concepts suggest that greater rotational invariance may be found on maps with more of the observed averaged rotation neurons.

5.3. K-means Clustering and Verification with RGZ Labels

Test results for the map segmentation using K-means clustering are shown for each SOM using an image of the SOM grid with decoded neuron weights and map displaying the closest matching RGZ image, with a K-means color coding on each neuron for the assigned K-means cluster Identification Number (ID) number. All K-means cluster ID numbers are arbitrary as they are assigned in an unsupervised manner. All associated ID colors are assigned as discrete color intervals to visually differentiate individual clusters IDs. Two tables are included for each test. The Tables 3 and 6 describe the division of the map assigned to each cluster and associated entropy statistics. Tables 4 and 6 describe for every cluster, the division of neurons with the label of the closest matching RGZ image to each neuron, the total division of the RGZ images with that label in the cluster and the associated entropy statistics.

These results demonstrate that the K-means clustering is separating SOM neurons closely related to morphology and to a lesser extent, the rotation angle of the source features. Clusters also appear to segment morphologies by their relative complexity. There are definitive simple, complex, and intermediate groups divided by the clustering, with clear groups of relatively simple clusters which are comprised of mostly point sources (RGZ label 11), compact multi-peak single-component sources (RGZ label 12), complex sources with highly separated sources or sparse sources with distant companions. Regarding general clustering quality, all tests show reasonable connectedness with few neuron clusters inter-mixing. The total clustering time for this SOM is negligible at 0.160 and 0.176 seconds for $K = 4$ and $K = 8$ clusterings respectively

In the $K = 4$ clusters, the UMAT in Figures 11 and 12, appears to be segmented into four groups with varying complexity. As summarized in Table 6, in the most simple group, cluster ID 1, a 0.88 division of the cluster contains RGZ labeled point sources, which contains a division of 0.73 of all RGZ point sources in the data set. Similar clustering is seen in

Table 6
Divisions of Clusters Based on the Label of the Closest Matching RGZ Image, Matching Clustered Label Divisions and Entropy Statistics for Each of the K = 8 Clusters in the 20 × 20 Toroidal SOM UMAT

Cluster ID	RGZ Label	Division of Matching Label In Cluster	Division of All Image Labels In Cluster	Mean Entropy	Minimum Entropy	Maximum Entropy
0	12	0.545	0.140	0.59	0.39	0.69
	22	0.409	0.097	0.58	0.48	0.67
	11	0.045	0.005	0.39	0.39	0.39
1	22	0.548	0.495	0.61	0.12	1.00
	11	0.452	0.172	0.20	0.00	0.51
2	22	0.900	0.194	0.62	0.36	0.98
	11	0.050	0.005	0.48	0.48	0.48
	12	0.050	0.012	0.75	0.75	0.75
3	11	0.977	0.584	0.22	0.00	0.30
	12	0.023	0.035	0.29	0.28	0.31
4	22	0.824	0.151	0.67	0.55	0.74
	11	0.176	0.014	0.37	0.15	0.52
5	12	0.660	0.407	0.45	0.15	0.75
	11	0.302	0.072	0.11	0.02	0.17
	22	0.038	0.022	0.56	0.34	0.78
6	12	0.682	0.174	0.53	0.18	0.82
	22	0.182	0.043	0.61	0.49	0.78
	11	0.136	0.014	0.11	0.09	0.12
7	11	0.600	0.136	0.14	0.03	0.26
	12	0.400	0.233	0.43	0.25	0.84

the more complex cluster 0, where a vast majority of the cluster contains radio-doubles and a majority of the radio-doubles in the data set reside. The remaining clusters 2 and 3, appear to segment largely medium complexity sources such as RGZ labeled 12 and a mix of 11 and 22 labels.

We observe more meaningful clusters with the K = 8 clustering tests shown in Figures 13 and 14. These clusters segment the map into similar groups to the K = 4 clusters, but with clusters containing higher population divisions, as shown in Table 5, where many groups are dominated by divisions in some cases of between 0.600 and 0.977 of the same RGZ labeled source. This can be seen with complex clusters such as 2 and 4 containing mostly RGZ labeled 22 sources. Similarly, simple clusters such as cluster 3, are almost entirely comprised of RGZ label 11 point sources and contain most of the point sources from the data set. Similar to the K = 4 cluster tests, there are a number of intermediate clusters such as cluster 5 and 6, with medium complexity that contain a vast majority RGZ label 12 sources.

In the K = 4 and K = 8 clustering tests, neither tables are listing more complex labels such as 33, as these form a minute population of the data set, but are visible by their morphology in the learned UMAT, as contained in the identified highly complex clusters. Most notable are clusters 0 and 3 from the K = 4 cluster test, and clusters 1 and 2 in the K = 8 cluster tests, which contain highly complex and interesting sources showing a wide range of extended features. Although the K = 8 cluster test contains more meaningful clusters, it appears a balance must be reached with the number of clusters and neurons available in the map. Using too many clusters may cause groupings to split logical classes to satisfy the K cluster value, or too few clusters which may result in true divisions and relationships on the map being under-represented or not revealed.

It is evident through these observations and the population division tables, that the K-means algorithm is segmenting morphologies and associating relationships not entirely correlated to the RGZ labels despite the clustered maps showing reasonably clear and logically assigned clusters based on morphology and complexity. These results indicate that the relationships learned by the system may be more complex than peak and component counts and therefore under-represent them. This detection of labels different from those of the training set is wholly expected since the system is trained in an unsupervised manner.

Across all tests, there are several evident entropy and map population effects. Most notably, mean cluster entropy values

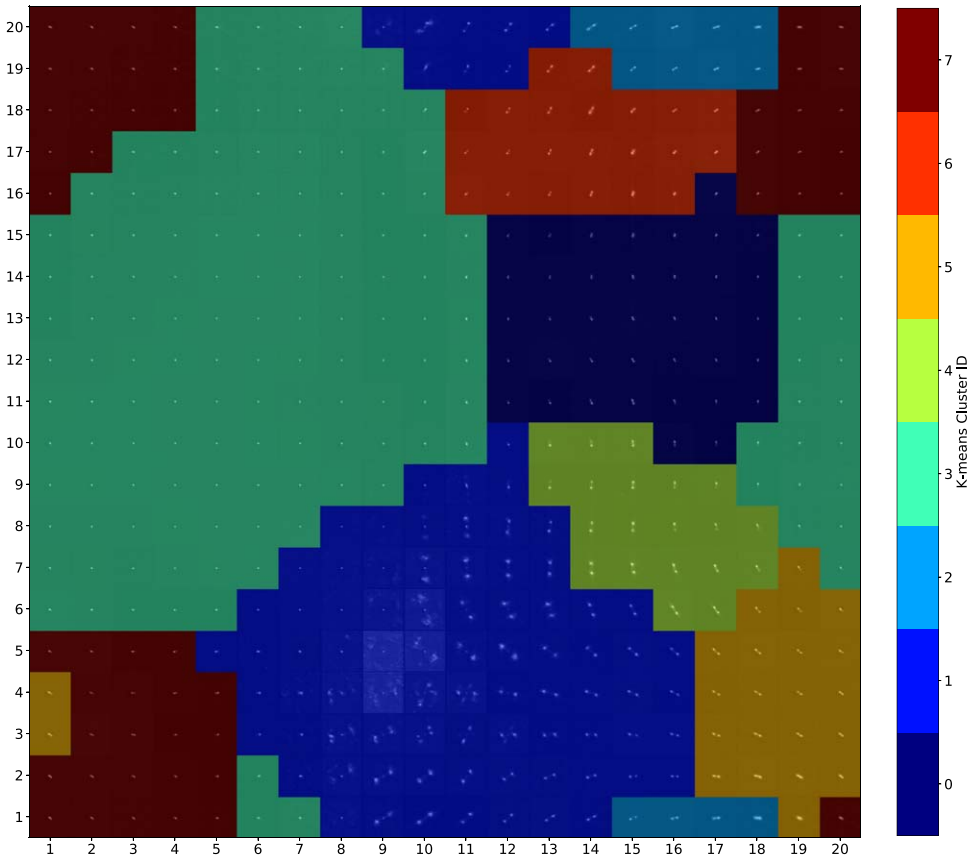

Figure 13. 20 × 20 toroidal SOM, displaying an overlay of decoded neuron weights with eight color-coded K-means clusters. (A color version of this figure is available in the online journal.)

increase with the presence of more complex and anomalous sources. As expected, point sources have the lowest mean entropy due to their simplicity and the greatest neuron population across the map due to the point source bias in the RGZ data set. Higher complexity sources, represent a smaller part of the training set and appear to have consistently smaller cluster populations, but significantly greater mean entropy due to their complexity. These population and entropy statistics effectively reveal not only the types and complexity of morphologies in the data set but also effectively describe the original label distributions and biases.

The difficulty when trying to use clusters as classifications on a continuous manifold can be seen by both the label crossovers found between many K-means classes and the point made by Kohonen (1997); SOMs are not explicitly designed for hard classification. The original principles of SOM learning will not produce highly distinct clusters, but will instead produce these results: a semantic map of outliers, regions, and morphologies rather than highly distinct groups. These qualities are largely seen with the blurring of features in neuron weights due to the relatively small 20 × 20 map size compressing the true feature space. It is possible that in exceptionally large SOM these relationships may have enough space to become sufficiently separated for discrete classification. Reduction of blurring effects in small crowded SOMs such as ours and the highlighting of outliers have also been successfully shown in Tasdemir & Merényi (2009), by instead implementing a new connectivity measure for the similarity of SOM prototypes that produces a more effective detection of manifold structures.

The total training time of this system is competitive with other methods from the literature such as Polsterer et al. (2016). Our approach produces similar SOM morphologies with square neurons and significantly reduced processing time as python code using a 24 core CPU requiring 14.55 minutes for the full autoencoder training of 10,000 images, encoding of 10,000 images, SOM training of 30,000 images, validation of an additional 30,000 images and final clustering, compared to 17 days with 200,000 images using python code on a 8 core CPU. The difference in data volume is the likely cause, where even with random rotation autoencoder training, our SOM training latent vectors contain only 900 elements per image, opposed to

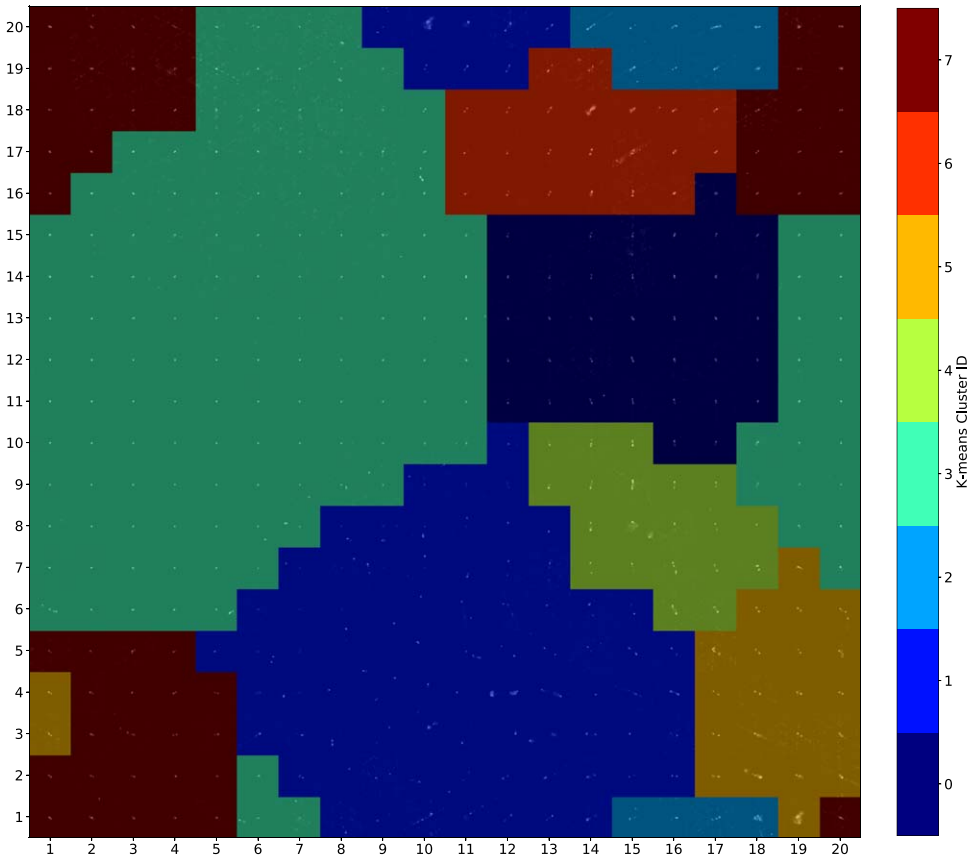

Figure 14. 20 × 20 toroidal SOM, displaying the closest matching RGZ images with eight color-coded K-means clusters. (A color version of this figure is available in the online journal.)

a total of 1,331,280 elements per image including rotations used in Polsterer et al. (2016).

6. Future Work

Our plans are to improve each of the three distinct components, the autoencoder, self-organizing map and clustering algorithms. By varying the latent vector sizes and structure of the autoencoder we will achieve a better balance between training time and accuracy. Additionally, we will be using a stacked architecture to train latent vectors for multi-channel data. The SOM will be further improved with heat map display of entropy in addition to gathering more performance metrics such as precision and reliability. We will also investigate other clustering algorithms trained in different learning spaces and projections. Additional variables such as map size can be eliminated and more dynamic relationships examined using a growing SOM (Rauber et al. 2002). We aim to further investigate affine invariance in SOM by training on images aligned to a common major axis and with all central components scaled to the same size. As previously discussed, the SOM output appears is continuous. Consequently, the challenge of more definitive and in-depth source classification can be viewed as a regression problem. We aim to continue working in this direction to create a machine-learning regression framework auxiliary to the SOM to regress the continuous SOM morphologies into discrete classes. Additionally, future investigations will also focus on determining the scalability of this method when applied to significantly increased data volume and more numerous classes.

7. Conclusion

We conclude that the coupling of self-organizing maps with convolutional autoencoders is an effective method of data exploration and unsupervised clustering of radio-astronomical images. Our approach directly addresses the growing survey processing time and provides a better means to explore large data sets automatically with a total processing time less than 15 minutes for 80,000 images. Our results demonstrate an accurate visualization of morphology distributions found within the RGZ data set. Our results show the capabilities of this method in locating outliers as high UMAT distance values and in K-means clustering with a distinct class of highly complex

sources with low data set population. By combining clustering with citizen science projects such as Radio Galaxy Zoo, greater efficiency can be achieved with volunteers inspecting only a small sample of objects from each cluster or being guided by likely morphologies in each cluster. The speed of this method holds implications for use on large future surveys, large-scale instruments such as the Square Kilometre Array (SKA; Johnston et al. 2007) and in other big data applications.

The authors acknowledge the Radio Galaxy Zoo Project builders and volunteers listed in full at http://rgzauthors.galaxyzoo.org for their contribution to RGZ data set and labels used in this paper. We also acknowledge the National Radio Astronomy Observatory (NRAO) and the Karl G. Jansky Very Large Array (VLA) as the source of this radio data. Partial support for L.R is provided by the U.S National Science Foundation grant AST17-14205 to the University of Minnesota. H.A benefited from grant DAIP #066/2018 of Universidad de Guanajuato.

References

Abadi, M., Barham, P., Chen, J., et al. 2016, OSDI, 16, 265
Alger, M., Banfield, J., Ong, C., et al. 2018, MNRAS, 478, 5547
Banfield, J. K., Wong, O., Willett, K. W., et al. 2015, MNRAS, 453, 2326
Becker, R. H., White, R. L., & Helfand, D. J. 1995, ApJ, 450, 559
Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441
Dreyfus, S. 1973, IEEE Trans. Autom. Control, 18, 383
Galvin, T. J., Huynh, M., Norris, R. P., et al. 2019, PASP, 131, 108009
Geach, J. E. 2012, MNRAS, 419, 2633
Graff, P., Feroz, F., Hobson, M. P., & Lasenby, A. 2014, MNRAS, 441, 1741
Gravet, R., Cabrera-Vives, G., Pérez-González, P. G., et al. 2015, ApJS, 221, 8
Hotelling, H. 1933, Journal of Educational Psychology, 24, 417
Johnston, S., Bailes, M., Bartel, N., et al. 2007, PASA, 24, 174
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Kohonen, T. 1997, IEEE Int. Conf. on Neural Networks, 1, PL1
LeCun, Y., Bengio, Y., & Hinton, G. 2015, Natur, 521, 436
Lloyd, S. 1982, IEEE Trans. Inf. Theory, 28, 129
Lukic, V., Brüggen, M., Banfield, J., et al. 2018, MNRAS, 476, 246
Maas, A. L., Hannun, A. Y., & Ng, A. Y. 2013, Proc. icml, 30, 3
Norris, R. P. 2017a, NatAs, 1, 671
Norris, R. P. 2017b, PASA, 34
Norris, R. P., Hopkins, A. M., Afonso, J., et al. 2011, PASA, 28, 215
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825
Perantonis, S. J., & Lisboa, P. J. 1992, IEEE Trans. Neural Networks, 3, 241
Polsterer, K., Gieseke, F. C., Igel, C., Doser, B., & Gianniotis, N. 2016
Polsterer, K. L., Gieseke, F., & Igel, C. 2015, in Conf. Proc. of Astronomical Data Analysis Software an Systems XXIV (ADASS XXIV) (San Francisco CA: ASP), 81
Rauber, A., Merkl, D., & Dittenbach, M. 2002, IEEE Trans. Neural Networks, 13, 1331
Ruder, S. 2016, arXiv:1609.04747
Sanger, T. D. 1989, Neural Netw., 2, 459
Shimwell, T. W., Röttgering, H. J. A., Best, P. N., et al. 2017, A&A, 598, A104
Tasdemir, K., & Merényi, E. 2009, IEEE Trans. Neural Networks, 20, 549
Ultsch, A. 1993, in Information and classification (Cham: Springer), 307
Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. 2010, Journal of Machine Learning Research, 11, 3371
Wang, H., Ren, K., & Song, J. 2017, III IEEE International Conference on Computer and Communications (ICCC), 2756
Wittek, P., Gao, S. C., Lim, I. S., & Zhao, L. 2013, Journal of Statistical Software, 78, 1, 2017
Wu, C., Wong, O., Rudnick, L., et al. 2019, MNRAS, 482, 1211
Xie, J., Xu, L., & Chen, E. 2012, in Proc. of the 25th International Conf. on Neural Information Processing Systems, Vol. 1 (Red Hook, NY: Curran Associates Inc.), 341