Help Me to Help You: Machine Augmented Citizen Science

DARRYL E. WRIGHT and LUCY FORTSON, University of Minnesota, USA
CHRIS LINTOTT, University of Oxford, UK
MICHAEL LARAIA, University of Minnesota, USA
MIKE WALMSLEY, University of Oxford, UK

The increasing size of datasets with which researchers in a variety of domains are confronted has led to a range of creative responses, including the deployment of modern machine learning techniques and the advent of large scale "citizen science projects." However, the ability of the latter to provide suitably large training sets for the former is stretched as the size of the problem (and competition for attention amongst projects) grows. We explore the application of unsupervised learning to leverage structure that exists in an initially unlabelled dataset. We simulate grouping similar points before presenting those groups to volunteers to label. Citizen science labelling of grouped data is more efficient, and the gathered labels can be used to improve efficiency further for labelling future data.

To demonstrate these ideas, we perform experiments using data from the Pan-STARRS Survey for Transients (PSST) with volunteer labels gathered by the Zooniverse project, Supernova Hunters and a simulated project using the MNIST handwritten digit dataset. Our results show that, in the best case, we might expect to reduce the required volunteer effort by 87.0% and 92.8% for the two datasets, respectively. These results illustrate a symbiotic relationship between machine learning and citizen scientists where each empowers the other with important implications for the design of citizen science projects in the future.

CCS Concepts: • **Human-centered computing** → **Collaborative interaction**; *Web-based interaction*; *Social tagging systems*;

Additional Key Words and Phrases: Deep learning, machine learning, clustering, citizen science, crowdsourcing

ACM Reference format:
Darryl E. Wright, Lucy Fortson, Chris Lintott, Michael Laraia, and Mike Walmsley. 2019. Help Me to Help You: Machine Augmented Citizen Science. *ACM Trans. Soc. Comput.* 2, 3, Article 11 (November 2019), 20 pages.
https://doi.org/10.1145/3362741

1 INTRODUCTION

The rate and volume at which data are collected across many scientific domains is accelerating. These datasets enable new science; however, our ability to fully extract meaningful information from these data is challenged. For example, the Large Synoptic Survey Telescope (LSST) project [15], now under construction, will provide 30 TB of imagery and millions of transient detections

Authors' addresses: D. E. Wright (corresponding author), L. Fortson, and M. Laraia, University of Minnesota, 116 Church St. SE, Minneapolis, MN 55455; email: darryl@zooniverse.org; C. Lintott and M. Walmsley, University of Oxford, Keble Road, Oxford, OX1 3RH, UK.

2469-7818/2019/11-ART11 $15.00
https://doi.org/10.1145/3362741

each night. To learn which alerts are worthy of follow-up resources requires a high-performance classification pipeline. However, previous classification solutions, in which a small group of experts manually processed the data they gathered, do not scale. Crowdsourcing projects which engage large numbers of human annotators is one recent solution that has been increasingly deployed to close this analysis gap. This is illustrated by the growth in online citizen science projects hosted on the Zooniverse[1] platform, from the initial Galaxy Zoo project [20] in 2007, to ~50 in January 2016 and over 150 projects today. By crowdsourcing annotations, a dataset can be processed faster than a research team could alone. However, even the crowd may not always be fast enough. A tension arises when new data is generated before all current data can be processed. LSST is expected to produce millions of alerts each night that require processing before the next night's data is produced to avoid a backlog. Although machine learning is a faster alternative to citizen science, it takes a long time for human classifiers (commonly citizen scientists themselves) to label enough data to adequately train a model [8, 9, 23]. Once trained, these models offer an alternative to citizen science, but it can take months if not years to gather the training data that made the models possible.

It is clear that efficiency gains must be made for crowdsourced annotations to remain a viable component of the research toolkit. In citizen science projects to date, volunteers have typically performed an exhaustive search through the data. Each *subject*[2] for classification is presented one at a time. To average out individual biases or mistakes, multiple independent volunteer annotations are gathered per subject. In most cases, a constant number of volunteers are required to annotate each subject and their contributions aggregated. But there is redundancy in this approach. We might expect that some subjects are easier to classify than others or that some citizen scientists are more accurate than others. Efficiency gains can thus be made either through optimizing the attention of high-performing members of the crowd, or through optimizing the way in which data are presented to the crowd. Work has already been done by several groups on tracking the response of individual labelers. For example, Simpson et al. [28] and Marshall et al. [22] found they could reduce the number of volunteer annotations needed to achieve the same performance compared to the simple aggregation mentioned above. Marshall et al. [22] were able to reduce the number of annotations to 34% of what they would otherwise have needed. Branson et al. [5] took a similar approach for applications on the Amazon Mechanical Turk[3] platform, but also incorporated computer vision achieving a reduction in annotation time by a factor of 4–11 for binary tasks. These techniques provide one avenue toward more efficient classification of large datasets. However, in this work, we explore potential gains in efficiency by optimizing how data are presented to volunteers.

To allow a new citizen science project to become more efficient and adopt machine learning sooner, we present a method that, through unsupervised learning, reduces volunteer effort to 30% of what would typically be required to annotate an example dataset. We then show how, by updating the model with supervised learning on an initial set of volunteer provided labels, we can refine the model to further reduce effort to ~18% when gathering the next batch of annotations. This same model can be used as a machine classifier for future data and we find that after one round of learning from volunteer labels the classifier can perform similarly to a single volunteer in terms of classification accuracy. The system therefore sets up a symbiotic relationship between volunteers and the model, whereby the initial model allows volunteers to classify more efficiently. The labeled data provided by volunteers "teach" the model how to perform better in the future,

[1] https://www.zooniverse.org/projects.
[2] Throughout, *subject* is used to refer to individual examples—usually images—drawn from a dataset.
[3] https://www.mturk.com/.

which in turn provides greater efficiency gains for volunteers. The model *helps* the volunteers to *help* the model. While the efficiency gains quoted here are specific to the particular application, in Appendix A we show that similar improvements can be expected for other datasets.

Our hypothesis is that an increase in efficiency can be gained by grouping subjects together, so that rather than asking volunteers to classify them one at a time, we could ask for classifications of the group. Say we choose groups to contain n subjects and our dataset contains c classes. Then in the worst case, every subject in a group belongs to a different class and requires $n - 1$ clicks to individually annotate each subject, since we can infer the class of the remaining unlabelled subject provided $c < n$. Under these conditions, the same amount of volunteer effort is required as the standard approach. On the other hand, if all the subjects in a group belong to the same class, with a suitable interface an entire group could be classified with a single click. So long as the groups are purer than the worst case, we will realize an efficiency gain over the standard approach, since volunteers need only click on members of the minority classes. In addition to c and n, the actual efficiency gains achieved will also depend on our ability to produce pure groupings. Assuming that volunteers are able to classify just as accurately in this new interface as with the standard approach, we can gather data more quickly without compromising quality. We test this hypothesis with an example citizen science project where $c = 2$ in experiments below while Appendix A also demonstrates efficiency gains for a problem where $c = 10$. In both cases, we will assume $n = 25$.

To achieve greater efficiency gains, we therefore want to increase the majority class' dominance within a grouping. But without knowing the labels for at least a portion of the dataset, we instead need to rely on some heuristic of similarity between subjects. This is the domain of clustering algorithms whose central goal is to group subjects into clusters such that those subjects assigned to a cluster are more similar than subjects assigned to different clusters. To achieve this, all clustering algorithms rely on some notion of distance within a feature space [13]. Those lying closer to each other in this space are assumed to be the most similar. For a citizen science project, we could cluster all unlabelled data before presenting groups from each cluster to citizen scientists for classification, aiming to leverage any structure in the unlabelled data that could be useful for distinguishing classes.

In practice, the feature space used to represent subjects to volunteers does not necessarily lend itself to meaningful clusters. For images, the feature space is defined by pixel values and clustering images from camera trap projects like Snapshot Serengeti[4] [29] for example, might group them based on the camera's location instead of the species within the image. Since the purpose of the project is species identification, this is problematic. We could try to mitigate this by hand-engineering features we believe are important for the classification of these objects and clustering in this new space. Either way, with no labels to begin with, we will be dependent on volunteer annotations to quantify the effectiveness of the feature space and we might discover that the features we designed need improving. We could try to iteratively refine the features we calculate, repeating the steps of feature design, clustering, and volunteer annotation. But a better solution would be to *learn* the feature representation. Better still, would be to learn the feature representation and clustering jointly. To this end, we explore an unsupervised method to learn a feature representation and clustering from unlabelled data and develop a feedback loop that allows us to incorporate human labels to refine both the feature representation and clustering. Labels provided by humans improve the cluster purity by helping the network identify salient features leading to greater efficiency gains in the next round of gathering labels, and the cycle continues. In the ideal case, the cycle would continue until the Bayes' error is achieved.

[4] https://www.zooniverse.org/projects/zooniverse/snapshot-serengeti.

The rest of this article continues as follows. In Section 2 we describe the network architecture. Section 3 presents our dataset and the experimental method in the context of citizen science. In Section 4, we report results and conclude in Section 5.

2 ARCHITECTURE

We desire an algorithm that is able to learn feature representations which suggests the use of deep learning [10, 24, 35]. Since the feature space is crucial to the success of clustering, it also seems desirable that during clustering, the network is free to update the feature representations it learns. There are a number of architectures that offer these properties (see Table 1 of Aljalbout et al. [1] for examples). We opted to base our approach on Deep Embedded Clustering (DEC) [34] because it is relatively well established in comparison to others and the network is composed of a Multi-Layered Perceptron, which makes fewer assumptions about the data than alternative architectures. We modify the DEC architecture to allow us to feedback volunteer labels to update the feature representations learned. We achieve this by adding new branches to the network which are turned off and on to enable switching between unsupervised and supervised learning. This approach is chosen for simplicity as our focus is on demonstrating the kinds of efficiency gains that could be achieved with similar approaches.

2.1 Deep Embedded Clustering

DEC learns to map the original representation of the data to a lower dimensional feature space in which the data are clustered. Training DEC begins with greedy layerwise pretraining of stacked (denoising) autoencoders (SAE) [30] that learn an initial feature representation for the input data. The second step involves embedding the data into this new feature space and initializing cluster centres with k-means [21]. The learned feature representation and cluster centroids are then optimised by minimizing the Kullback-Leibler (KL) divergence between a soft assignment of the data to cluster centroids and a desired target distribution for unsupervised learning. The soft assignments of individual examples to centroids takes the form of Student's t-distribution, which acts to measure the similarity between a data point embedded in the feature space and each cluster centroid. The target distribution is designed to emphasise confidently assigned data points, improve clustering purity, and mitigate against large clusters dominating the loss function (see Equation 3 of Xie et al. [34]). The target distribution is a function of the soft assignments, which are in turn dependent on the initial clustering and feature representation learned by the SAE. If the learned features are not discriminative for the desired classification task, for example, they capture the general colour of an image rather than the object it contains, then the optimization of DEC will act to reinforce this. DEC offers no guarantee that the identified clusters will be optimal for the classification task, as is the case with any unsupervised clustering algorithm.

2.2 Modifications

After pretraining the autoencoders, we modify the DEC model according to Figure 1. The top branch of the model aims to minimise the reconstruction loss and is composed of the deep autoencoder learned during pretraining. The intuition behind including the reconstruction loss is in preserving any important information learned by the model during the initial unsupervised phase [1]. The middle branch corresponds to the DEC architecture proposed by Xie et al. [34]. The bottom branch enables supervised updates to the network and is constructed by adding a clustering layer on top of the DEC encoder. This clustering layer shares cluster centroids with the middle branch and ensures that cluster centres learned by either branch are coherent. On top of this clustering layer, we add a softmax classification layer with one output unit for each class, real or bogus. This means that supervised updates to the model will also affect the learned cluster centres.

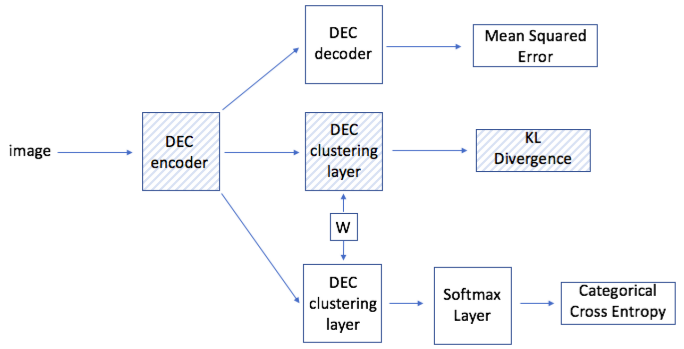

Fig. 1. Proposed network architecture with modifications to DEC. The hatched portions show the original DEC architecture.

The model is trained with back-propagation and the Adam optimiser [17] with a learning rate of 0.001 and all other parameters left as defaults. We use a weighted combination of the three loss functions and in Section 3 we will use these weights to break training into separate phases. For example, setting the weights of the mean squared error and categorical cross entropy losses to zero is equivalent to training DEC as described in the previous section. Table 1 details how the weights of each loss function are changed throughout training.

3 EXPERIMENTAL METHOD

Our experiments will compare how labels are gathered from volunteers comparing the standard interface to a simulation of an interface where groups are labelled. We also compare our proposed clustering method to k-means to highlight the advantages of joint training. Throughout, we will hold constant the data, architecture, and training method. The architecture of the DEC encoder is kept the same as in Xie et al. [34], where the dimensions are 500-500-2,000-10. Our experiments therefore simulate running multiple citizen science projects that differ in the way they gather volunteer labels and in how those labels are used to influence the way labels are gathered in the future.

3.1 Data

Our experiments use 20×20 pixel greyscale images extracted from Pan-STARRS1 (Chambers et al. [7]) difference images (see Section 2.1 of Wright et al. [33] for details). We take 9,219 images obtained between June 1, 2013 and June 20, 2014. These images contain either examples of image artefacts or detections of real astrophysical transients, with the goal of distinguishing between the two. This is the same dataset used to train the Convolutional Neural Network (CNN) described in Wright [32]. The data are also split along the same partitions as used to develop the CNN with 75% of the data for training and 25% for testing. In the context of a supernova survey, where data is constantly being collected every night, the 75% of the data designated for training will represent data we have in hand when we first launch a citizen science project. This training partition is further split into three sets: (1) a training set consisting of 3,458 subjects, (2) a development set consisting of 1,729 subjects, and (3) a validation set containing 1,729 subjects. We use volunteer provided labels collected for (1) to train the model, and volunteer labels collected for (2) to determine stopping criteria during training; this is to replicate as closely as possible the situation for a real citizen science project. (3) represents unlabelled data we have in hand but have not gathered volunteer labels for. We use this validation set for monitoring the model's performance throughout training, but the model only "sees" the validation set during the unsupervised steps

Table 1. Data Splits Used for Training, Validation, and Testing During Training Steps

Training step	Training data	Validation data	Test data	Mean squared error	KL divergence	Categorical cross entropy
Unsupervised	training	-	test	0.0	1.0	0.0
Clustering	development	-	-			
	validation	-	-			
Multitask	training	development	test	1.0	0.0	1.0
	-	validation	-			
Reclustering	training	validation	test	0.0	1.0	0.0
	development	-	-			

The last three columns show the weight assigned to each of the loss functions during the different training steps.

(initial unsupervised clustering and reclustering steps described below). The test set represents data obtained in the future that the trained model will be applied to. The test set is not used for training at any point and its sole purpose is to estimate the performance we could expect from the model on future data at different points in the training process. The test set is also used to measure benchmarks against which our approach is compared. We use expert provided labels for both the validation and test sets. Table 1 shows whether each data split is used for training, validation, or testing during each of the training steps. In Wright et al. [33], we demonstrate the utility of the volunteer labels for the training steps through a quantitative comparison between volunteer vote fraction labels and expert labels.

3.2 Training Process

3.2.1 Unsupervised Clustering. The first step is to perform an unsupervised clustering using DEC in order to make an initial attempt at grouping similar images together. To achieve this, we set the weights for the mean squared error and categorical cross entropy losses to zero and train the model unsupervised on the training, development, and validation sets. We choose to use 10 clusters, although volunteers are only asked to label the data as one of two classes (real or artefact). This represents the case where we may not know ahead of time how many classes exist in the data and it is best to err on the side of more clusters, leaving open the opportunity for the discovery of subtypes or new classes. There are several artefact types and the differences between them can be nuanced; given that artefacts make up the majority of the data, it makes sense to reduce the problem to a binary classification task for the volunteers rather than have them agonise over assigning artefacts to specific classes. The top left panel of Figure 2 shows the unsupervised clustering of the Supernova Hunters training partition.

3.2.2 Gathering Volunteer Feedback Simulation. To assess the quality of the unsupervised clustering and to provide labels to feedback to our model, we gather labels from volunteers. The above dataset was uploaded to the Supernova Hunters project between May 23, 2018 and May 24, 2018 gathering classifications in the standard way, that is, one subject at a time. During this period, we received a total of 254,175 individual classifications resulting in an average of 27.6 classifications per image with each image being classified by at least 10 volunteers.

We aggregate the individual classifications into a single label using vote fractions [31]. Vote fractions are calculated by taking the fraction of volunteers who classified each subject as real. This is then converted to a "hard" label by setting a threshold at 0.5, meaning if more than 50% of volunteers classified a subject as real, then it is labelled real, otherwise the subject is labelled as an

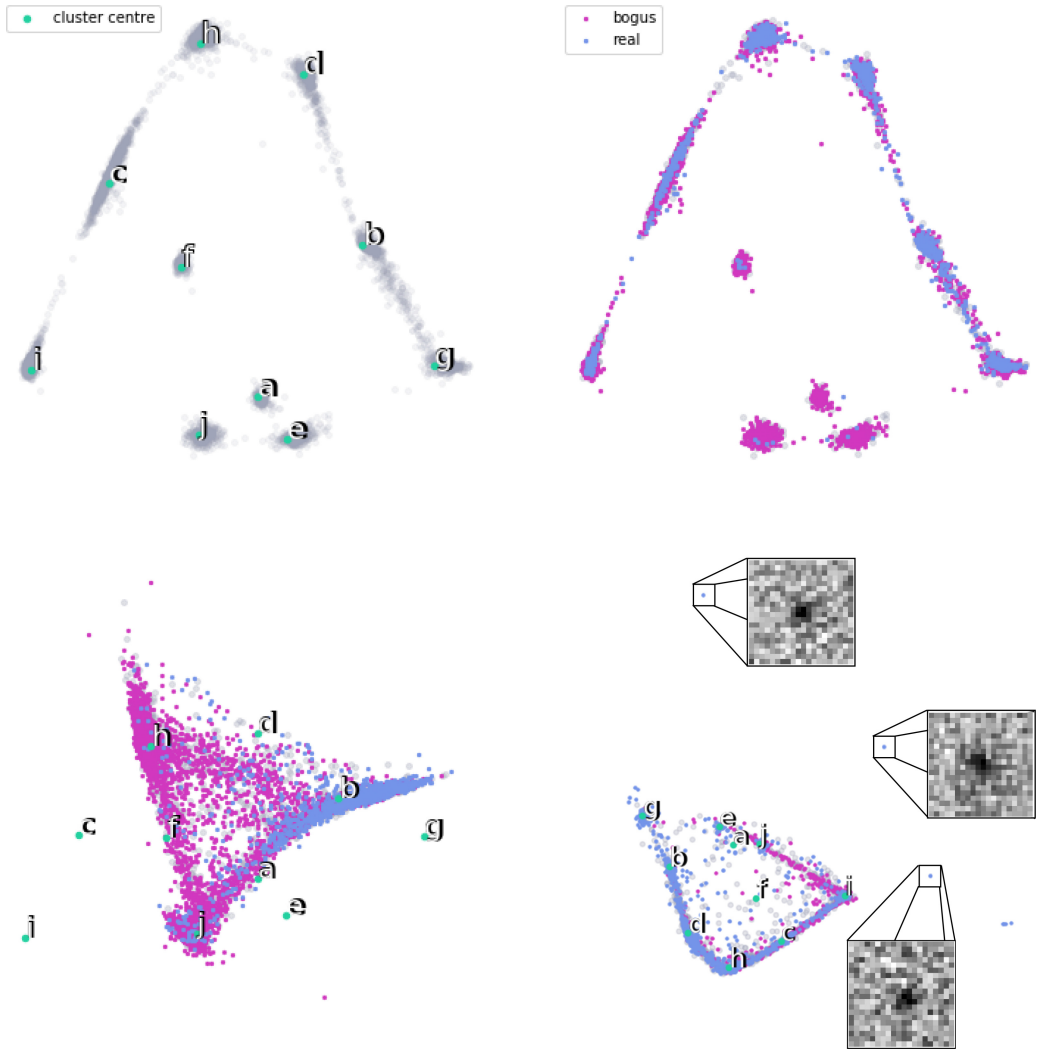

Fig. 2. (Top left) 2-D PCA projection of the initial embedding learned by DEC for the Supernova Hunters unlabelled dataset. The initial unlabelled training set is shown in grey with cluster centres in green. Each cluster is assigned a letter to identify it. (Top right) The same as before but with vote fraction labels derived from volunteer classifications overlaid for the training and development sets. (Bottom left) The embedded space after the multitask step. This space does not lend itself to clustering analysis. (Bottom right) The final reclustered embedding. This space provides good classification results and also offers the most efficiency gains when gathering volunteer labels. Highlighted are three outliers which turn out to be examples of low signal-to-noise detections of transients which are rare in the training data since they are close to the detection limit of the survey, providing one example of the type of analysis this space enables.

artefact. These are the labels we use for the training and development sets described in the previous section. We emphasize that, as this is a simulation, these labels are gathered in the standard way. In a "live" project, we would use a different interface, presenting groups of similar images together in a grid. We discuss such an interface in greater detail in Section 3.3 below.

3.2.3 Learning from Volunteer Feedback. At this stage, we have gathered volunteer labels for 75% of the data that we have available (consisting of the training and development sets). The next step is to feed these labels to the network in such a way that we improve the purity of the clusters and in turn the classification performance of the model. To achieve this, we perturb the embedded space such that members of each class will lie closer to each other. This implies that the feature representation learned by the model will have captured aspects of the subjects that are more meaningful for the target task of separating real and bogus detections. The goal of this stage is therefore to update the embedded space such that it provides a better initialisation for the next step of reclustering. At this point, the weight of the KL divergence loss is set to zero and the weights for the mean squared error and categorical cross entropy are set to one, allowing the network to perform supervised learning on the volunteer labels. We found that including the reconstruction loss (mean squared error) leads to more stable training. We call this phase of training the *multitask step* as we are optimising for two loss functions. Using early stopping with a patience of 5 epochs and monitoring the loss on the development set, the multitask step ends after 13 epochs with the best model checkpoint from epoch 8.

3.2.4 Reclustering. The lower left panel of Figure 2 shows the new embedded space we obtain at the end of the multitask step (Section 3.2.3). This new space is less amenable to clustering analysis and the aim of this step is therefore to cluster the data again, reapplying DEC by setting the KL divergence weight to one and the others to zero. We call this the *reclustering step*. Since the development set is no longer needed to determine the stopping condition for the multitask step, we append it to the training set.

3.3 User Interface Simulations

Finally, we run simulations to compare efficiency gains, counting how many clicks it would take volunteers to classify all the data in the test set, had the labels been gathered through different interfaces. One underlying assumption is that, no matter the interface, volunteers provide consistent labels and we leave a comparison of how different interfaces affect the quality of labels or changes in volunteer engagement to future work. We also assume that every click incurs a constant time cost and that the sum of clicks is a proxy for the actual time required to classify. It is also worth restating that in the case of Supernova Hunters, classification is a binary task and the analysis will differ for tasks with more classes.

In general, Zooniverse projects set a retirement limit that defines the number of volunteers who must classify each subject before it is considered "fully" classified and retired from the project. For example, Supernova Hunters operates with a retirement limit of 10. As this is a constant, we ignore it in our simulations.

Simulations of the standard interface are trivial and we will discuss them further in Section 4.2 where we also discuss the addition of other clicks required by the different interfaces, such as those needed to submit a final classification, which differ depending on the interface. The remainder of this section describes how we simulate the number of clicks that would be required to label the test set had the volunteers been classifying through an interface with groups of images. For example, a group might consist of 25 subjects drawn at random without replacement from the same cluster presented in a 5×5 grid. Volunteers would be directed to determine the *minority* class of those presented and to select all subjects belonging to that class by clicking on them. These are the clicks we count in the simulation.

The simulations run by randomly sampling from the embedded space at each stage and counting the number of subjects that belong to the minority class interpreting this count as the number of clicks. We loop through the data until all subjects in the test set have been labelled once. These

simulations are run at the unsupervised stage and after reclustering. We also simulate efficiency gains comparing DEC and k-means as the clustering algorithm.

4 RESULTS

4.1 Metrics

For experiments relating to efficiency, simulated click counts (see Section 3.3) are taken as an estimate of volunteer effort. Relative efficiency gains are compared in terms of the differences between these simulated counts and are the main objective of our work.

Since efficiency in terms of click counts is difficult to track during training, we monitor two additional metrics, namely, the F_1-score and homogeneity score. The F_1-score monitors classification accuracy while homogeneity tracks cluster purity. We expect efficiency gains to follow from improvements in these metrics, since for example, purer clusters imply grids presented to volunteers will be purer, which take fewer clicks to classify. To measure the F_1-score, we need to convert the clustering algorithm into a classifier. For this we rely on the volunteer labels which are used to calculate a cluster-to-label mapping. Each cluster is mapped to the majority label that volunteers provided for the random sample of subjects drawn from each cluster. Predictions are made by predicting the label mapped to a cluster for all subjects assigned to that cluster. We track both metrics as the F_1-score is dependent on the cluster-to-label mapping, while the homogeneity is independent of the mapping.

The advantage of the F_1-score rather than classification accuracy is that an increase in the measured F_1-score corresponds to a meaningful improvement in classification. For example, simply classifying all targets as real (i.e., the "all ones" benchmark) decreases the precision, which in turn decreases the F_1-score.

The homogeneity score tracks cluster purity while not requiring that all members of a class are assigned to a single cluster. The homogeneity score is maximised when each cluster only contains members of a single class [26]. Given a set of classes C and clusters K, the homogeneity score is calculated as

$$h = 1 - \frac{H(C \mid K)}{H(C)}. \tag{1}$$

Here $H(C \mid K)$ is the conditional entropy of the classes given the cluster assignments and $H(C)$ is the entropy of the classes. $H(C \mid K)$ is defined as

$$H(C \mid K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n} \cdot log\left(\frac{n_{c,k}}{n_k}\right), \tag{2}$$

and $H(C)$ is given by

$$H(C) = -\sum_{c=1}^{|C|}\frac{n_c}{n} \cdot log\left(\frac{n_c}{n}\right), \tag{3}$$

where c and k index classes and clusters, respectively, n is the total number of subjects, n_c is the number of subjects belonging to each class c and similarly n_k denotes the number of subjects assigned to cluster k, and finally, $n_{c,k}$ is the number of subjects assigned to cluster k from class c.

4.2 Labelling Efficiency

Here we present the results of the simulations described in Section 3.3 (User interface Experiment). First we consider the standard Zooniverse classification interface, where subjects are presented one at a time. In this case, volunteers make a single click to select which class they think the presented subject belongs to, plus an additional click to submit the final classification. The total number of

Table 2. User Interface Simulation Results Over Five Trials

Method	Classification clicks	Interface clicks	Total clicks	Efficiency gain
Standard	2,303	2,303	4,606	-
Worst case clustering	1,153	186	1,339	70.9%
Unsupervised pretraining + k-means	657	186	843	81.7%
DEC	671	186	857	81.4%
Reclustering step (k-means)	649	186	835	81.9%
Reclustering step (DEC)	412	186	598	87.0%

clicks to classify the Supernova Hunters test set can therefore be simply simulated as twice the total number of subjects in the set, that is, 4,606.

We test a number of approaches to grouping the images and presenting them in grids to volunteers. It is worth noting that in the worst case, every grid we present would have an equal number of real and bogus subjects and volunteers must then click on exactly half the subjects. Volunteers must also click to select the minority class plus an additional click to indicate when they are finished classifying, that is, two clicks for every grid in addition to the clicks associated with classification. The total of these additional clicks is two times the number of grids in the dataset. Throughout we will use 5×5 grids for our simulations; given the small image sizes (20×20 pixels), we expect that the individual images should appear large enough on a computer monitor for volunteers to be able to assess. The number of grids is given by the total number of subjects divided by the number of subjects in each grid and is therefore $2,303/25$ or 93 for a 5×5 grid. We add a constant of 186 clicks to the results of the click simulation described in Section 3.3. In this case, the upper bound on the number of clicks through an interface such as this would therefore be half the total number of subjects; $1,153$ plus the constant 186 clicks needed to drive the interface, or 1,339 clicks in total.

Next, we take the autoencoder trained as in the initialisation of DEC and run k-means on the encoded training set. We report the results of this simulation as "unsupervised pretraining + k-means" in Table 4. The feature space is then updated following the same protocol as the multitask step described above. We then cluster in this newly learned embedded space with k-means. The difference between this and the DEC approach is that the embedded space used to initialise the multitask step has not been updated by the unsupervised DEC clustering and the multitask step is instead applied directly to the embedded space learned by the autoencoder and recorded as "reclustering step (k-means)" in Table 4. We repeat this process but with DEC. In contrast to k-means, we observe a significant improvement in expected efficiency gains after updating the model with volunteer labels. We repeat each simulation for five trials and report the mean in Table 2 rounded to the nearest integer. These simulations show that our proposed approach ("reclustering step (DEC)") achieves an efficiency gain of ~28% over using DEC or k-means alone, and an 87% gain over the standard classification interface.

4.3 Training Process

In this section, we track the F_1-score and homogeneity score defined in Section 4.1 throughout the training process. We also calculate benchmarks to help set our method in the context of alternatives, such as clustering with k-means instead of DEC, or relying solely on experts for labels.

4.3.1 Benchmarks. Since we adopt the F_1-score we calculate the *all ones* benchmark, that is, the F_1-score that would be measured by predicting all subjects as real detections. Any classifier ought to at least outperform this metric and the actual benchmark depends on how skewed the data is. We

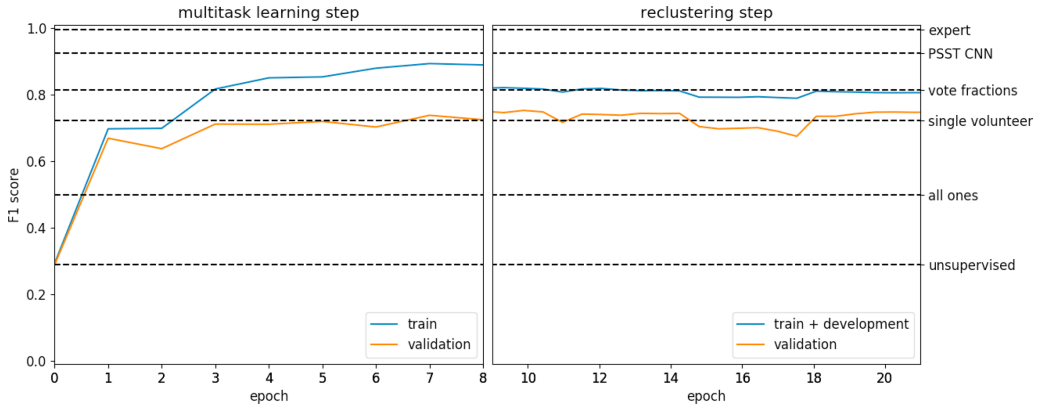

Fig. 3. Learning curves during the multitask and reclustering steps. The right panel shows F1-score during the multitask step on the training and validation steps. Using early stopping, the best performing model checkpoint on the development set is epoch 8. At this point, the model has likely over fit the training data outperforming the theoretical Bayes' error. We then continue to train the model with another unsupervised clustering step. The purpose of this step is to transform the feature space at the end of the multitask step (see bottom left panel of Figure 2) into a space that is more suited for drawing the next round of subjects for volunteers to label and for clustering analysis. For this step, we combine the training and development to increase the number of training examples. This accounts for the discontinuity in the blue line; the validation set (orange line) is held constant between both steps. The reclustering step converges after 13 epochs and we find that performance is at least as good as if a single volunteer was to label each image in the validation set.

also calculate several human-level performance benchmarks. First is the human expert benchmark which is estimated as an F_1-score of 0.996 based on previous analysis from Wright [32]. This is a rough estimate and likely to be an upper bound on the actual value. But it is helpful for framing our other results since we do not expect humans to perform any better. In addition, we calculate volunteer performance benchmarks. The vote fraction benchmark is the performance we get by aggregating the classifications from a crowd of volunteers (see Section 3.2.2), while the single volunteer benchmark is the performance we measure by taking the label submitted by the first volunteer to classify each image. This is similar to the classification performance we could expect from the simulations we performed in Section 3.3 since those simulations assumed a retirement limit of 1. Finally, we include a supervised learning benchmark based on the CNN we mentioned in Section 3.1 which was designed specifically for this task and trained on labels provided by experts. These benchmarks are shown as the top four rows in Table 4 and as horizontal black dashed lines in Figure 3.

4.3.2 Unsupervised Clustering. After greedy-layerwise pretraining of the stacked autoencoders, one could apply the DEC clustering step (as we propose) or use a more traditional clustering. In the latter case, the unlabelled training data would be encoded and then used as the feature representation for clustering. We compare both these approaches and find that clustering with k-means gives a slightly higher homogeneity than DEC. This is manifested in the differences in efficiency of the two methods reported in Section 4.2 and "unsupervised pretraining + k-means" and "DEC" in Table 2.

To measure the F_1-score the cluster-to-label mapping (Section 4.1) needs to be calculated. This could either be done with *gold standard data*, where we have high confidence in the correctness of the labels, for example from experts or simulations (if available), or we could wait to gather

Table 3. Comparison of Clusters at Various Stages in the Supernova Hunters Analysis

	Initial - gold mapping			Initial - volunteer mapping			Multitask step			Reclustering step		
Cluster	Assigned	Label	Purity	Assigned	Label	Purity	Assigned	Label	Purity	Assigned	Label	Purity
a	74	0	1.000	105	0	1.000	0	-	-	0	-	-
b	214	1	0.514	362	0	0.541	995	1	0.839	1,191	1	0.870
c	312	0	0.715	475	0	0.802	21	0	0.810	189	0	0.894
d	266	1	0.556	364	0	0.560	0	-	-	8	1	0.750
e	166	0	0.958	217	0	0.986	18	1	0.667	303	1	0.620
f	74	0	0.986	151	0	0.974	0	-	-	0	-	-
g	318	0	0.506	418	0→1	0.531	24	1	1.00	114	1	0.895
h	385	0	0.600	500	0	0.630	0	-	-	14	0	0.500
i	419	0	0.766	655	0	0.827	2,353	0	0.989	2,961	0	0.917
j	75	0	0.973	211	0	0.990	47	0	0.553	407	0	0.649

In the case where no cluster was assigned to one of the classes, the → symbol signifies when a cluster was chosen to represent the missing class with the label on the left changing to the label on the right. This cluster is chosen as it contains the most subjects belonging to the missing class.

Table 4. Supernova Hunters Results Showing the Test Set Performance on Our Proposed Approach, Along with Comparisons with Alternative Methods and Various Benchmarks (See Text for Details)

Method	F1-score	Homogeneity
All ones	0.499	-
Single volunteer	0.722	-
Volunteer vote fractions	0.813	-
Human expert	0.996	-
Unsupervised pretraining + K-means	0.614	0.193
DEC (initial - gold mapping)	0.414	0.134
DEC (initial - volunteer mapping)	0.290	0.134
Multitask step	0.702	0.319
Reclustering step (k-means)	0.441	0.194
Reclustering step (DEC)	0.713	0.347
Supervised (PSST CNN)	0.925	-

labels from volunteer**s**. Using gold standard data allows us to immediately measure the F_1-score without the need to wait for labels to be gathered, but the performance measure will be biased as the cluster-to-label mapping would be determined from the gold standard labels themselves. To illustrate this point, we use the test set to assign the cluster to label mapping as if it were a gold standard dataset, and measure an F_1-score of 0.414. We show an analysis of the mapping calculated with the test set as the *initial - gold mapping* column in Table 3. In comparison, using volunteer labels produces the mapping shown as *initial - volunteer mapping* in Table 3, which produces an F_1-score of 0.290 on the test set (shown as the *initial - volunteer mapping* in Table 4). The large measured performance difference between the two is due to the fact that in both cases the clusters assigned a label of 1 (meaning real) have a low purity and the cluster label (determined from the majority class) is therefore susceptible to small changes in the proportion of real and bogus detections assigned to those clusters. A change in the label assigned to a cluster results in the labels of all subjects belonging to that cluster having their predicted labels flipped; this in turn leads to

large fluctuations in the F_1-score. For the remainder of our analysis, we use the mapping derived from volunteer labels.

4.3.3 Multitask Step. We show the learning curve for the first eight epochs of the multitask learning step in the left panel of Figure 3 where the model is trained on the volunteer labels.

The fact that the training set performance is higher than the vote fraction benchmark indicates that the model has overfit the training data as it is unlikely our model, trained on volunteers labels, will outperform the volunteers themselves. A second observation is that the performance on the validation data is about equal to the single volunteer benchmark. At the end of the multitask learning step, the test set performance is 0.702; this is an estimate of the classification performance we expect on future data and a 0.412 improvement on the initial unsupervised classification performance.

4.3.4 Reclustering. The right panel of Figure 3 shows the learning curves for the training and validation sets during the reclustering step. The discontinuity in the blue line (training + development) is due to the addition of the development set to the training set and the F_1-score is therefore not directly related between the two steps. The validation set, on the other hand, is the same across both panels. We see that during the reclustering step the F_1-score changes very little, and we measure a small improvement over the performance achieved in the multitask step on the validation set. This also holds true for the held out test set, which improves to 0.713. After this first round of gathering volunteer labels, we have trained a classifier that we expect to perform almost as well as a single volunteer on future data.

5 SUMMARY AND DISCUSSION

We have presented a system for gathering labels from volunteers in citizen science projects, where we take advantage of unsupervised clustering to group subjects together in feature space such that we can expect to more efficiently gather labels from volunteers. We find that by gathering labels for a small subset of subjects and updating the learned feature space with these labelled examples, we can improve the clustering such that clusters become purer and we realise greater efficiency gains when gathering labels for the next subset of data. We performed experiments with data gathered from a live citizen science project, Supernova Hunters. We described the steps that must be taken to adapt the DEC model for our approach and found that we could reduce volunteer effort to label a new dataset to about 18% of the standard approach for gathering labels. In Appendix A, we apply our method to the MNIST dataset to demonstrate its ability to generalise beyond the Supernova Hunters data.

5.1 Limitations and Future Directions

There remain open questions with this approach. Our experiments so far have been simulations performed on data gathered through the standard classification interface, where each subject is classified individually. In contrast, the grid interface requires that many images be displayed at once. This implies a reduced window size per image that could result in a more difficult task for volunteers [14], potentially leading to reduced accuracy or engagement. Gains in efficiency then, need to be balanced against these other factors. The grid size can be adjusted to account for tasks that require volunteers to have access to greater image detail; a smaller grid size allows for larger images. For example, if this were a requirement for Supernova Hunters, in the extreme case, a grid size of three could still lead to efficiency gains under our analysis. This is of course assuming that our proxy for volunteer effort holds, namely, the number of clicks required by the interfaces to classify a dataset. This is a strong assumption in the regime of small grid sizes or highly confused clusters where the grid interface would demand many clicks which may take

longer given the increased cognitive load [25]. As the grid size increases and/or the clusters become purer, the assumption is somewhat weaker and any increase in the time per click required by the grid interface would likely be compensated for by the significantly fewer clicks required by the grid interface compared to the standard interface. In practice, the degree to which the time per click for each interface differs will likely depend on the details of a citizen science project. Our priority, therefore, is to test an interface that presents groups of images together and validate actual volunteer efficiency gains in terms of wall clock time and annotation accuracy when compared with the standard interface. We have taken initial steps in this direction with the Muon Hunter 2.0[5] citizen science project. Preliminary results suggest that volunteers spend one-tenth of the time per subject classifying with the grid interface as they do with the standard interface. Determining how classification accuracy is affected will require further work; however, at first glance it does not appear to be significantly affected by the grid interface.

Another concern is losing the opportunity for serendipitous discovery, one of the major routes through which citizen science projects have provided scientific impact in the past decade. Examples include the discovery of a class of highly star-forming dwarf galaxies [6], an unusual star whose rapid, irregular dimming might be due to the presence of an unusual dust cloud [3, 4], and a red gravitational lens found in a project whose training set consisted only of blue examples [11]. But how can we preserve this when humans do not review every subject? The obvious modification would be to employ more intelligent subject sampling rather than the random sampling we have used so far. Subjects are currently sampled without replacement from the entire dataset independent of the clustering. This has the effect of "targeting" those clusters with the most subjects assigned to them and the densest regions of the embedded space. But perhaps the model could benefit from more knowledge of cluster outliers, or active learning techniques [27] could help select subjects with the greatest expected benefit to the model. However, these methods miss the case where the model assigns an incorrect label with high confidence, so-called *unknown unknowns* in the literature [2, 18]. Combining some of these ideas to determine which groups of subjects volunteers should label could not only enable serendipitous discovery but benefit the model, helping it converge to a better clustering with less labelled data and providing even greater efficiency.

Another avenue for further exploration is different network architectures. Aljalbout et al. [1] provides a review of many deep clustering architectures that we could explore, and there have been modifications to the original DEC architecture that might offer improvements for images [12] or more interpretable [16] results. Currently, the *predicted* efficiency gains are significant, but the trained model does not achieve adequate classification performance for deployment by itself. Although not necessary, it seems preferable that the model we train could eventually be relied on to perform the task at hand. Considering most Zooniverse projects are image based, perhaps some of the modifications, especially those employing convolutional layers, could help us achieve greater classification performance.

APPENDIX

A MNIST EXAMPLE

In this Appendix, our aim is to show that our method can generalise to another dataset. We repeat the experiments above using the MNIST dataset [19] as an example. Xie et al. [34] achieved 84% classification accuracy on MNIST. The dataset contains 70,000 28×28 pixel greyscale images of handwritten digits. These images have accurate labels and have been preprocessed to remove effects like background noise and ensure that digits are centred in the images. We use this highly

[5] https://www.zooniverse.org/projects/dwright04/muon-hunters-2-dot-0/classify.

Table 5. MNIST User Interface Simulation Results Over Five Trials

Method	classification clicks	interface clicks	total clicks	efficiency gain
Standard	10,000	10,000	20,000	-
Worst case clustering	8,800	4,000	12,800	36.0%
DEC	1,152	767	1,919	90.4%
Reclustering step (DEC)	614	829	1,443	92.8%

Table 6. Comparison of Clusters After Applying DEC (Iinitial) and After Learning from Volunteer Labels (Updated)

	Initial			Updated		
Cluster	Assigned	Label	Purity	Assigned	Label	Purity
a	1,008	8	0.941	1,028	8	0.921
b	993	0	0.977	1,004	0	0.968
c	1,013	7	0.969	1,014	7	0.969
d	1,059	3	0.924	1,012	3	0.949
e	969	4	**0.514**	939	4	**0.947**
f	1,126	1	0.988	1,121	1	0.987
g	1,035	2	0.965	1,020	2	0.974
h	870	5	0.952	898	5	0.947
i	1,002	9	**0.506**	1,025	9	**0.890**
j	925	6	0.983	939	6	0.982

"sanitised" dataset to simulate a toy citizen science project. This will demonstrate our method in the absence of much of the noise inherent in "real-world" data.

We first divide the dataset into training, validation, and test sets. 50,000 images are used as the training set which, for our purposes, acts as the pool of images we have available and would like to gather labels for; it is assumed to be initially unlabelled. We divide the remaining 20,000 images equally among the validation and test sets and the labels for these are assumed to be available.

A.1 Labelling Efficiency

Similarly to Supernova Hunters (Section 3.3), the simulated total number of clicks to classify the MNIST test set through the standard interface is just twice the number of subjects in the test set, that is, 10,000. Again, we simulate groupings of 25 subjects. As such, each grouping will require at most 22 clicks to classify, since in the worst case there would be five classes with two subjects represented and five classes with three. Since we can infer the class of an unlabelled subject if all subjects belonging to the other $c - 1$ classes are labelled, rationally we would leave one of the classes with three subjects unlabelled. The increased number of classes in MNIST (ten) compared to Supernova Hunters (two) adds additional complexity to the calculation of other clicks required by the interface. In the worst case, nine clicks are required to classify each of the $c - 1$ classes represented in the grid plus one additional click to signal the grid has been completed. As with Supernova Hunters, the latter is equal to the total number of grids in the dataset, that is, $10{,}000/25 = 400$, while the former is calculated during simulation as it depends on the number of classes in the sample drawn for each grid. We report our results in Table 5, finding the worst case provides a 36% reduction in the effort required by the standard interface. Additionally, we find that

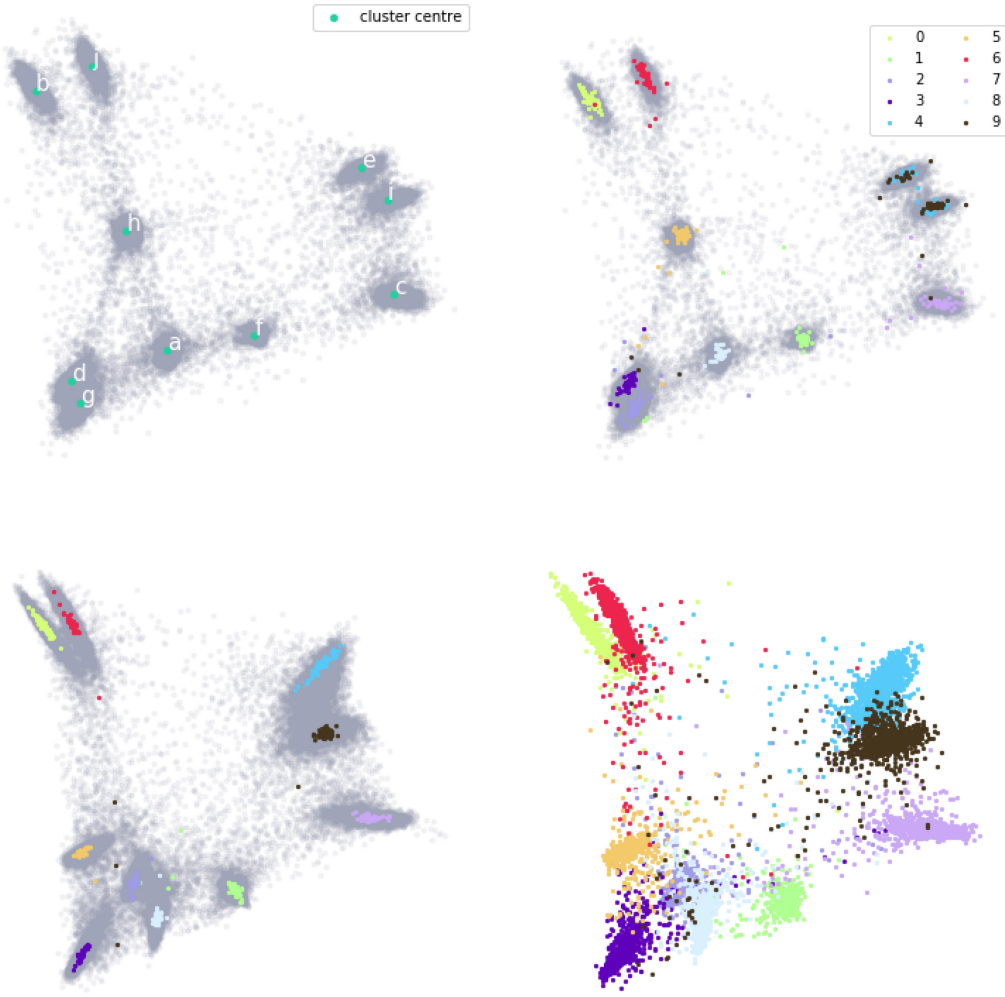

Fig. 4. (Upper left) 2-D PCA projection of the initial MNIST embedding learned by DEC. The initial unlabelled training set is shown in grey with cluster centres in green. Each cluster is assigned a letter to identify it in Table 6. (Upper right) Same as before but coloured points represent simulated volunteer classifications (as described in Section A.2.1) for 1% of the data (where each colour corresponds to a different class, i.e., a digit between 0 and 9) overlaid on the remaining unlabelled data. While some clusters appear pure, others are more confused, such as the two clusters on the right where the black and light-blue classes (digits 9 and 4, respectively) are poorly separated. (Lower left) The results of updating the learned feature space with the volunteer classifications. Clusters for digits 9 and 4 are now more clearly distinguished. (Lower right) The test set projected into the same feature space showing that the clustering generalises well.

DEC alone produces a 92.4% reduction in effort and the overall approach provides a 94.8% decrease in effort required by the standard approach.

A.2 Training Process

A.2.1 Unsupervised Clustering. The first training step is an unsupervised clustering of the data using DEC. This is equivalent to the MNIST experiment of Xie et al. [34]. The result of this step is visualised as the first panel of Figure 4. As in Xie et al. [34], we use the entire MNIST dataset

Table 7. MNIST Results

Method	Accuracy (%)	H
DEC	86.6	0.783
Reclustering step (DEC)	95.3	0.874

Test set performance from DEC and after learning from sampled labels.

for this step, mimicking a citizen science project where all the data to be labelled are available upfront. In a diversion from Xie et al. [34], we only measure performance on the 10,000 images we assigned to the test set. The test set acts as a fixed dataset not seen by volunteers, on which to compare performance between the steps we take.

It is worth reiterating at this point that, unless a labelled dataset is available before running the citizen science project, we could not measure the unsupervised clustering accuracy. Therefore, we simulate asking volunteers to label a subsample of the clustered data drawn at random from each cluster. This provides an estimate of the clustering accuracy and purity of each cluster. We sample 1% (500 images) of the training set and, assuming that volunteers would be perfect classifiers, use the labels provided in the MNIST dataset to represent the labels volunteers would return. The upper right panel of Figure 4 is similar to the upper left, but with the labelled subsample overlaid. From the figure, we see that the clusters corresponding to digits 4 and 9 are highly confused in this 2-D projection of the feature space. The purity of these clusters measured on the test set are shown as the fourth column in Table 6. Next, we calculate the cluster-to-label mapping as in Section 4.1 in order to measure classification accuracy. On the test set we achieve an unsupervised clustering accuracy of 86.6% (see Table 7) on the test set. Given the estimated high purity of many of the clusters, we might consider using this clustering to classify the remaining unlabelled data assigned to those clusters.

A.2.2 Multitask Step and Reclustering. Since some clusters appear to confuse classes, we aim to use the knowledge provided by volunteers to improve the learned feature space such that images from each class lie in distinct clusters. As with Supernova Hunters, we train the multitask step on the 500 labelled training images. The lower left panel of Figure 4 visualises the new feature space learned by training on this labelled data. The lower right panel shows the distribution of the test dataset images embedded in the same space, demonstrating that the embedded space generalises well for separating classes. In both panels the clusters for 4 and 9 are now more clearly distinguished. We achieve a clustering accuracy of 95.3% (see Table 7) on the test set. The homogeneity of the subjects assigned to clusters also improves from 0.783 after applying DEC to 0.874 after learning from volunteer labels.

ACKNOWLEDGMENTS

D.W., L.F., and M.L. gratefully acknowledge partial support through the US National Science Foundation grants IIS 1619177 and PHY 1806798. C.J.L. acknowledges support from STFC under grant ST/N003179/1. M.W. acknowledges funding from the Science and Technology Funding Council (STFC) Grant Code ST/R505006/1. The Pan-STARRS1 Surveys have been made possible through contributions of the Institute for Astronomy, the University of Hawaii, the Pan-STARRS Project Office, the Max-Planck Society and its participating institutes, the Max Planck Institute for Astronomy, Heidelberg and the Max Planck Institute for Extraterrestrial Physics, Garching, The Johns Hopkins University, Durham University, the University of Edinburgh, Queen's University Belfast, the Harvard-Smithsonian Center for Astrophysics, the Las Cumbres Observatory Global Telescope Network Incorporated, the National Central University of Taiwan, the Space Telescope Science

Institute, the National Aeronautics and Space Administration under Grant No. NNX08AR22G issued through the Planetary Science Division of the NASA Science Mission Directorate, the National Science Foundation under Grant No. AST-1238877, the University of Maryland, and Eotvos Lorand University (ELTE) and the Los Alamos National Laboratory. We also wish to acknowledge the dedicated effort of our citizen scientists who have made this work possible.

REFERENCES

[1] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers. 2018. Clustering with deep learning: Taxonomy and new methods. *ArXiv E-prints* (Jan. 2018). arxiv:1801.07648

[2] Gagan Bansal and Daniel S. Weld. 2018. A coverage-based utility model for identifying unknown unknowns. In *Proc. of AAAI*.

[3] T. Boyajian, S. Croft, J. Wright, A. Siemion, M. Muterspaugh, M. Siegel, B. Gary, S. Wright, J. Maire, A. Duenas, C. Hultgren, and J. Ramos. 2017. A drop in optical flux from Boyajian's star. *The Astronomer's Telegram* 10405 (May 2017).

[4] T. S. Boyajian, D. M. LaCourse, S. A. Rappaport, D. Fabrycky, D. A. Fischer, D. Gandolfi, G. M. Kennedy, H. Korhonen, M. C. Liu, A. Moor, K. Olah, K. Vida, M. C. Wyatt, W. M. J. Best, J. Brewer, F. Ciesla, B. Csak, H. J. Deeg, T. J. Dupuy, G. Handler, K. Heng, S. B. Howell, S. T. Ishikawa, J. Kovacs, T. Kozakis, L. Kriskovics, J. Lehtinen, C. Lintott, S. Lynn, D. Nespral, S. Nikbakhsh, K. Schawinski, J. R. Schmitt, A. M. Smith, Gy. Szabo, R. Szabo, J. Viuho, J. Wang, A. Weiksnar, M. Bosch, J. L. Connors, S. Goodman, G. Green, A. J. Hoekstra, T. Jebson, K. J. Jek, M. R. Omohundro, H. M. Schwengeler, and A. Szewczyk. 2016. Planet Hunters IX. KIC8462852—Where's the flux?*Monthly Notices of the Royal Astronomical Society* 457, 4 (2016), 3988–4004. DOI: http://dx.doi.org/10.1093/mnras/stw218 eprint=/oup/backfile/content_public/journal/mnras/457/4/10.1093_mnras_stw218/3/stw218.pdf.

[5] Steve Branson, Grant Van Horn, and Pietro Perona. 2017. Lean crowdsourcing: Combining humans and machines in an online system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7474–7483.

[6] C. Cardamone, K. Schawinski, M. Sarzi, S. P. Bamford, N. Bennert, C. M. Urry, C. Lintott, W. C. Keel, J. Parejko, R. C. Nichol, D. Thomas, D. Andreescu, P. Murray, M. J. Raddick, A. Slosar, A. Szalay, and J. Vandenberg. 2009. Galaxy zoo green peas: Discovery of a class of compact extremely star-forming galaxies. *Monthly Notices of the Royal Astronomical Society* 399 (Nov. 2009), 1191–1205. DOI: https://doi.org/10.1111/j.1365-2966.2009.15383.x arxiv:0907.4155

[7] K. C. Chambers, E. A. Magnier, N. Metcalfe, H. A. Flewelling, M. E. Huber, C. Z. Waters, L. Denneau, P. W. Draper, D. Farrow, D. P. Finkbeiner, C. Holmberg, J. Koppenhoefer, P. A. Price, A. Rest, R. P. Saglia, E. F. Schlafly, S. J. Smartt, W. Sweeney, R. J. Wainscoat, W. S. Burgett, S. Chastel, T. Grav, J. N. Heasley, K. W. Hodapp, R. Jedicke, N. Kaiser, R.-P. Kudritzki, G. A. Luppino, R. H. Lupton, D. G. Monet, J. S. Morgan, P. M. Onaka, B. Shiao, C. W. Stubbs, J. L. Tonry, R. White, E. Bañados, E. F. Bell, R. Bender, E. J. Bernard, M. Boegner, F. Boffi, M. T. Botticella, A. Calamida, S. Casertano, W.-P. Chen, X. Chen, S. Cole, N. Deacon, C. Frenk, A. Fitzsimmons, S. Gezari, V. Gibbs, C. Goessl, T. Goggia, R. Gourgue, B. Goldman, P. Grant, E. K. Grebel, N. C. Hambly, G. Hasinger, A. F. Heavens, T. M. Heckman, R. Henderson, T. Henning, M. Holman, U. Hopp, W.-H. Ip, S. Isani, M. Jackson, C. D. Keyes, A. M. Koekemoer, R. Kotak, D. Le, D. Liska, K. S. Long, J. R. Lucey, M. Liu, N. F. Martin, G. Masci, B. McLean, E. Mindel, P. Misra, E. Morganson, D. N. A. Murphy, A. Obaika, G. Narayan, M. A. Nieto-Santisteban, P. Norberg, J. A. Peacock, E. A. Pier, M. Postman, N. Primak, C. Rae, A. Rai, A. Riess, A. Riffeser, H. W. Rix, S. Röser, R. Russel, L. Rutz, E. Schilbach, A. S. B. Schultz, D. Scolnic, L. Strolger, A. Szalay, S. Seitz, E. Small, K. W. Smith, D. R. Soderblom, P. Taylor, R. Thomson, A. N. Taylor, A. R. Thakar, J. Thiel, D. Thilker, D. Unger, Y. Urata, J. Valenti, J. Wagner, T. Walder, F. Walter, S. P. Watters, S. Werner, W. M. Wood-Vasey, and R. Wyse. 2016. The pan-STARRS1 surveys. *Arxiv E-prints* (Dec. 2016). arxiv:astro-ph.IM/1612.05560

[8] Sander Dieleman, Kyle W Willett, and Joni Dambre. 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society* 450, 2 (2015), 1441–1459.

[9] H. Domínguez Sánchez, M. Huertas-Company, M. Bernardi, D. Tuccillo, and J. L. Fischer. 2018. Improving galaxy morphologies for SDSS with Deep Learning. *Monthly Notices of the Royal Astronomical Society* 476, 3 (2018), 3661–3676. DOI: https://doi.org/10.1093/mnras/sty338

[10] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2009. *Visualizing Higher-Layer Features of a Deep Network*. Technical Report 1341. *University of Montreal*.

[11] J. E. Geach, A. More, A. Verma, P. J. Marshall, N. Jackson, P.-E. Belles, R. Beswick, E. Baeten, M. Chavez, C. Cornen, B. E. Cox, T. Erben, N. J. Erickson, S. Garrington, P. A. Harrison, K. Harrington, D. H. Hughes, R. J. Ivison, C. Jordan, Y.-T. Lin, A. Leauthaud, C. Lintott, S. Lynn, A. Kapadia, J.-P. Kneib, C. Macmillan, M. Makler, G. Miller, A. Montaña, R. Mujica, T. Muxlow, G. Narayanan, D. O'Briain, T. O'Brien, M. Oguri, E. Paget, M. Parrish, N. P. Ross, E. Rozo, C. E. Rusu, E. S. Rykoff, D. Sanchez-Argüelles, R. Simpson, C. Snyder, F. P. Schloerb, M. Tecza, W.-H. Wang, L. Van Waerbeke, J. Wilcox, M. Viero, G. W. Wilson, M. S. Yun, and M. Zeballos. 2015. The Red

Radio Ring: A gravitationally lensed hyperluminous infrared radio galaxy at z = 2.553 discovered through the citizen science project SPACE WARPS. *Monthly Notices of the Royal Astronomical Society* 452 (Sept. 2015), 502–510. DOI: https://doi.org/10.1093/mnras/stv1243 arxiv:1503.05824

[12] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. 2017. Deep clustering with convolutional autoencoders. In *International Conference on Neural Information Processing*. Springer, 373–382.

[13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer New York, Inc., New York, NY.

[14] Michael E. Hodgson. 1998. What size window for image classification? A cognitive perspective. *PE & RS- Photogrammetric Engineering and Remote Sensing* 64, 8 (1998), 797–807.

[15] Ž. Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, D. Alonso, Y. AlSayyad, S. F. Anderson, J. Andrew, and et al.2008. LSST: From science drivers to reference design and anticipated data products. *ArXiv E-prints* (May 2008). arxiv:0805.2366

[16] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2016. Variational deep embedding: An unsupervised and generative approach to clustering. *Arxiv Preprint* (2016). arXiv:1611.05148

[17] D. P. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *ArXiv E-prints* (Dec. 2014). arxiv:1412.6980

[18] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. 2017. Identifying unknown unknowns in the open world: Representations and policies for guided exploration.. In *AAAI*, Vol. 1. 2.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.

[20] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. 2008. Galaxy Zoo: Morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society* 389 (Sept. 2008), 1179–1189. DOI: https://doi.org/10.1111/j.1365-2966.2008.13689.x.arxiv:0804.4483

[21] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. Oakland, CA, 281–297.

[22] P. J. Marshall, A. Verma, A. More, C. P. Davis, S. More, A. Kapadia, M. Parrish, C. Snyder, J. Wilcox, E. Baeten, C. Macmillan, C. Cornen, M. Baumer, E. Simpson, C. J. Lintott, D. Miller, E. Paget, R. Simpson, A. M. Smith, R. Küng, P. Saha, and T. E. Collett. 2016. SPACE WARPS—I. Crowdsourcing the discovery of gravitational lenses. *Monthly Notices of the Royal Astronomical Society* 455 (Jan. 2016), 1171–1190. DOI: https://doi.org/10.1093/mnras/stv2009 arxiv:astro-ph.IM/1504.06148

[23] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, C. Packer, and J. Clune. 2017. Automatically identifying wild animals in camera trap images with deep learning. *Arxiv Preprint* (2017). arXiv:1703.05830

[24] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature visualization. *Distill* (2017). Retrieved from DOI: https://doi.org/undefined, https://distill.pub/2017/feature-visualization.

[25] Sharon Oviatt. 2006. Human-centered design meets cognitive load theory: Designing interfaces that help people think. In *Proceedings of the 14th ACM International Conference on Multimedia*. ACM, 871–880.

[26] Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*.

[27] Burr Settles. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6, 1 (2012), 1–114.

[28] E. Simpson, S. Roberts, I. Psorakis, and A. Smith. 2012. Dynamic Bayesian combination of multiple imperfect classifiers. *ArXiv E-prints* (June 2012). arxiv:math.ST/1206.1831

[29] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer. 2015. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Scientific Data* 2, 150026 (2015). http://dx.doi.org/10.1038/sdata.2015.26

[30] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11 (Dec.2010), 3371–3408.

[31] K. W. Willett, C. J. Lintott, S. P. Bamford, K. L. Masters, B. D. Simmons, K. R. V. Casteels, E. M. Edmondson, L. F. Fortson, S. Kaviraj, W. C. Keel, T. Melvin, R. C. Nichol, M. J. Raddick, K. Schawinski, R. J. Simpson, R. A. Skibba, A. M. Smith, and D. Thomas. 2013. Galaxy Zoo 2: Detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society* 435 (Nov. 2013), 2835–2860. DOI: https://doi.org/10.1093/mnras/stt1458 arxiv:1308.3496

[32] Darryl Wright. 2015. *Machine Learning for Transient Surveys*. Ph.D. Dissertation. Department of Physics and Astronomy, Queen's University Belfast.

[33] D. E. Wright, C. J. Lintott, S. J. Smartt, K. W. Smith, L. Fortson, L. Trouille, C. R. Allen, M. Beck, M. C. Bouslog, A. Boyer, K. C. Chambers, H. Flewelling, W. Granger, E. A. Magnier, A. McMaster, G. R. M. Miller, J. E. O'Donnell, B. Simmons, H. Spiers, J. L. Tonry, M. Veldthuis, R. J. Wainscoat, C. Waters, M. Willman, Z. Wolfenbarger, and D. R. Young. 2017. A transient search using combined human and machine classifications. *Monthly Notices of the Royal Astronomical Society* 472, 2 (2017), 1315–1323. DOI: https://doi.org/10.1093/mnras/stx1812

[34] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*. 478–487.

[35] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*. Springer, 818–833.

Received March 2019; revised September 2019; accepted September 2019