A Preliminary Approach for the Exploitation of Citizen Science Data for Fast and Robust Fuzzy k-Nearest Neighbour Classification

Manuel Jiménez[1], Mercedes Torres Torres[2], Robert John, *Senior Member, IEEE*[1,3] and Isaac Triguero, *Member, IEEE*[1]

[1]The Automated Scheduling, Optimisation and Planning Research Group (ASAP)

[2]Computer Vision Lab (CVL)

[3]Laboratory for Uncertainty in Data and Decision Making (LUCID)

School of Computer Science, University of Nottingham, UK

Email: {manuel.jimenezmorales, mercedes.torrestorres, robert.john, isaac.triguero}@nottingham.ac.uk

***Abstract*—Citizen science is becoming mainstream in a wide variety of real-world applications in astronomy or bioinformatics, in which, for example, classification tasks by experts are very time consuming. These projects engage amateur volunteers that are tasked to manually classify unannotated examples. As a result, we obtain a larger volume of labelled data that, however, contains a great level of uncertainty due to the wide range of expertise of the volunteers. Handling that inherent uncertainty is key to building robust and fast machine learning models that maximise the outcome of citizen science projects. In this work, we introduce a preliminary approach that first transforms the original results from a citizen science project to handle the uncertainty, and then uses this as input to a fuzzy k-nearest neighbour classifier. We leverage citizen science results in such a way that it naturally speeds up the learning and classification phases of the fuzzy classifier, and improves the classification performance. As a case study, we will focus on the Galaxy Zoo project that consisted of galaxy image classification. Our experimental results show that an appropriate use of citizen science data enables a faster and more robust classification using the fuzzy k-nearest neighbour classifier.**

I. Introduction

An increasingly number of research fields are faced nowadays with the automated generation of huge volumes of data. Taking the form of telescopes or geo-sensors, these modern facilities are providing professional researchers with tons of images hardly manageable on their own. In most cases, this data remains unlabelled and awaiting for experts to manually label it. Citizen science [1] refers to the engagement of amateur people in research, leveraging the capacity of the crowds for data processing at a scale not possible for experts alone. However, this approach still constitutes a temporary solution, since upcoming technologies announce the automated generation of data at rates not even possible to be processed with the aid of citizen science projects.

The use of Machine Learning (ML) [2] approaches that exploit citizen science data is expected to create more accurate models as they are provided with larger sets of annotated data. These techniques should learn from projects' outcomes in conjunction with available expert knowledge on the problem being tackled. Nonetheless, the use of ML in citizen science has generally been restricted to the replication of amateur classification skills [3], [4], which does not provide a valid solution in the long run as the uncertainty is not addressed.

Regardless its potential, citizen science still provokes scepticism across the scientific community [5]. Although it enables data analysis at a large scale, the practice is not widely accepted yet [6]. The employment of amateurs tends to degrade the quality of the data labelling, giving rise to a wide spread of uncertainties within citizen science results. To mitigate this, we developed a methodology to enhance the confidence on amateur classifications, taking expert labels as ground truth [7]. We provided a set of data transformations that takes advantage of the lack of consensus amongst amateur participants as well as the inner uncertainty measured in the course of the project. Using this information, we refined the original amateur votes pursuing more accurate results.

In this paper, we aim to make use of our previous approach to handle uncertainty in citizen science to build fast and robust automatic classifiers. In particular, this preliminary work is focused on the Fuzzy k-Nearest Neighbour (Fuzzy-kNN) [8], which has proven to be a very effective fuzzy-based classification algorithm [9]. This method is typically composed of two main phases: (1) a very time-consuming training phase in which class membership is computed for training instances based on their distances, and (2) a classification stage in which the membership degree is used to infer the class of unseen examples. The reduced scaling up capabilities of the Fuzzy-kNN algorithm have attracted the attention of researchers when dealing with big datasets [10] to especially speed up the first stage. In the context of citizen science, we propose to skip the first stage and directly apply the transformations provided in [7]. The main benefit is to accelerate the training phase and exploit the *natural* fuzzification of the class instances (in contradistinction to the uncertainty/class memberships computed by the Fuzzy-kNN algorithm). In addition, the refined citizen data provide a first-order evidence of the training instances' quality. Thus, we also investigate the effect of a very simple

978-1-5386-1728-1/19/$31.00 ©2019 Crown

data reduction approach [11] that consists of using a threshold to eliminate examples with low confidence from the training set. In comparison with applying any state-of-the-art instance selection techniques [12], the proposed thresholding approach does not require any extra computation.

In our experiments, we focus on the galaxy image classification problem, for which the Galaxy Zoo 1 (GZ1) project represents one of the most successful citizen science implementations to date [13]. To test the proposed method, we compare our results with the standard Fuzzy-kNN and the crisp version of the classification algorithm.

The rest of the paper is organised as follows. In Section II, we provide some preliminaries about citizen science and the Fuzzy-kNN algorithm. Section III introduces the proposed approach for the exploitation of citizen science data in conjunction with the Fuzzy-kNN classifier. In Section IV, we describe and discuss the experiments carried out. Finally, Section V concludes the paper and outlines future work directions.

II. Preliminaries

In this section, we cover the necessary information to understand the proposed classification approach. Subsection II-A introduces citizen science and briefly summarises previous work on that area. Subsection II-B succinctly explains the Fuzzy-kNN algorithm.

A. Handling uncertainty in citizen science data

As a form of crowd-sourcing, citizen science projects usually involve a great number of volunteers to help with diverse tasks that are typically tedious and very time consuming for experts to be conducted. The Zooniverse[1] platform for the development of citizen science projects currently hosts more than 80 different projects in areas such as space sciences, medicine, ecology, or humanities. Many of those projects have consisted of the classification of large collections of images. Amateurs are usually asked to classify the images displayed in the project website by choosing amongst a set of pre-defined categories. When a project is finalised, all the classifications performed by amateurs are recorded in the form of a count of votes for each class label in each image (e.g. image 1 has got 24 votes for class x, and 3 votes for class y).

ML techniques have been used in the context of citizen science. On-line approaches have recently been designed to improve amateurs' experience [14]. These approaches deal with the training of participants as they gain expertise in the task and the synergy between amateurs, experts and automated classifiers [15]. However, most efforts have been focused on off-line approaches that are focused on analysing the data resulting from citizen science projects. Various papers have been devoted to replicating amateur classification skills. This has been normally achieved by firstly applying a feature extraction algorithm from the images [3], [16], or using the image directly by means of a deep learning approach (that performs its own feature extraction) [4]. The current literature has shown that ML classifiers can achieve similar results to those obtained by amateurs, when these algorithms are trained using citizen science data [17]. However, those approaches do not tackle the uncertainty prevalent in the results, which considering the varied amateurs' skills may be very diverse. Thus, they are replicating the biases present in the data.

In [7], we proposed a sequence of fuzzy-based data transformations that take the original amateurs' count of votes and mitigate this uncertainty. Through various aggregation mechanisms, the proposed approach is able to deal with different types of uncertainty present in citizen science data (e.g. inherent uncertainty, due to the intrinsic lack of agreement between participants, measured uncertainty, which is derived from the clicks on the *Don't Know* alternative, etc.). As a result, we obtained an enriched version of the data that was evaluated by considering available expert classifications for the same data sample. However, to the best of our knowledge, this transformed data has not been used within a ML algorithm that could take advantage of this refined data for the automated classification of new examples.

B. Fuzzy k-Nearest Neighbour classification

In this paper, we use the Fuzzy-kNN algorithm [8] as a simple yet effective fuzzy-based classifier. Future experiments will involve other fuzzy-based classifiers. The Fuzzy-kNN algorithm was devised to improve upon the standard k-Nearest Neighbour algorithm (k-NN) [18]. The main weakness of k-NN resides in considering all neighbours as equally important in the classification, making the k-NN algorithm more vulnerable to noise at the class boundaries. This algorithm has shown to perform well in a wide variety of applications, and more important, it is capable of considering uncertainty when predicting the class label of a new example.

The Fuzzy-kNN usually performs a learning phase, in which the training set is modified to compute the class memberships of every single training instance. This stage consists of calculating the k nearest neighbours for each instance of the training set against the training set (using a leave-one-out scheme), and selecting those k instances holding the shortest distance. Class memberships for each training instance x is then calculated following Equation (1), where n_j is the number of examples of class j within the k_{memb} nearest neighbours computed.

$$u_j(x) = \begin{cases} 0.51 + (n_j/k_{memb}) \cdot 0.49 & if \quad j = i \\ (n_j/k_{memb}) \cdot 0.49 & if \quad j \neq i \end{cases} \tag{1}$$

The result of this first stage will be a modified training set in which the class label ω of each sample is replaced by a membership vector $(\omega_1, \omega_2, \ldots, \omega_l)$, where l is the number of classes. For each instance of the test set, the classification stage calculates its k nearest neighbours in the modified training data (with class memberships). Thus, it obtains the membership vector of each neighbour and aggregates this vector by applying the Equation (2), where the m parameter modulates

[1] http://www.zooniverse.org

how heavily the distances are weighted when the contributions of each neighbour are added up. Finally, the prediction will be based on the class with the higher membership.

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij}(1/||x - x_j||^{2/(m-1)})}{\sum_{j=1}^{k}(1/||x - x_j||^{2/(m-1)})} \quad (2)$$

It is well-known that the Fuzzy-kNN algorithm does not cope well with very large datasets, especially because the class membership computation involves a training vs. training comparison, which is very time consuming. Recent approaches based on big data technologies proposed parallel alternatives to speed up the fuzzification process [10]. Nevertheless, the use of citizen science data allows for skipping this fuzzification phase, considerably speeding up the total Fuzzy-kNN execution time, as we detail in the following section.

III. Methodology

In this section, we explain the main motivations for the approach proposed here. First, we introduce the data scores that enable an immediate fuzzification for the training examples (Subsection III-A). After this, we explain how our method also leverages the amateur labelling for a faster classification with the Fuzzy-kNN classifier (Subsection III-B).

A. Human-based fuzzification

In off-line approaches, each example shown to participants throughout the project holds a particular count of votes once the project has finished. These votes are then converted to scores, simply dividing the votes assigned to each category by the total number of votes received by the example. Let $\mathbf{N} = (n_1, n_2, ..., n_{|N|})$ be the vote vector, with $|N|$ the number of different options offered to participants, we get the score vector $\mathbf{X} = (x_1, x_2, ..., x_{|N|})$ by computing $x_i = \frac{n_i}{M}$, with $M = \sum n_i$ and $i \in \{1, 2, ..., |N|\}$. Hence, the amateur labelling does not result in final classifications as such. Conversely, there is available a set of scores for each example in the resulting data, as it is shown in Table I. This set of options for amateurs' clicks generally involve a set of main classes, which are actually the target of the classification problem. Nonetheless, amongst them, one *Don't Know* (DK) option is usually offered, assuring that every time the example is shown it gets a click from the person deciding at the moment. Unlike the classification target classes, this DK score provides a direct quantification of the example level of confidence: the more DK votes it gets, the higher is the expected noise introduced by the example to the final data.

Through the set of scores, each one accounts for the example's degree of membership to the $i-th$ option displayed in the web. Here, we propose to exploit these scores as a human-based fuzzification for the training of the Fuzzy-kNN classifier. However, the first *raw* scores directly obtained from amateur votes does not differentiate between the target classes, which tend to accumulate the major part of the votes, and the DK clicks and other minority classes, nor does the variability in the total number of votes received by the example, M. To

TABLE I
A SAMPLE OF CITIZEN SCIENCE DATA GENERATED IN A PROJECT INVOLVING IMAGE CLASSIFICATION

Image ID	***Votes***	***Class 1***	***Class 2***	$\cdots$	***Don't Know***
0152948451	58	0.310	0.414	$\cdots$	0.052
0152863349	14	0.643	0.214	$\cdots$	0.071
0152878152	33	0.000	1.000	$\cdots$	0.000
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
0152721030	19	0.316	0.263	$\cdots$	0.263

prevent this, we first refine the raw scores by the application of the data transformations that aggregate information about the uncertainty in the amateur votes, as it is thoroughly explained in [7]. Then, we directly use this refined scores obtained for the target classes as the fuzzy labels needed for the training phase of the Fuzzy-kNN algorithm. By this, we are skipping the learning phase that involves the computation of the fuzzy memberships for the training examples solely based on the distances rather than actually taking into account the uncertainty in amateur classifications (Equation 1). Although here we use the Fuzzy-kNN for carrying out the different experimental trials, these transformed scores could be used with any other fuzzy classifier.

B. Instance selection based on confidence

The score vector available for each of the examples in the data does not only tell us about the example degree of membership to each class of the classification problem. From a ML point of view, the target classes' scores also hold information about the quality of the instance for the training of the automated classifier. Uneven score vectors will come from examples for which there has been a higher consensus amongst the people who voted that example. Thus, they will represent valuable examples for the learning algorithm. On the contrary, examples for which the votes are equally spread through the classes convey a poor agreement, thus providing a low confidence.

As part of our proposal, we apply a set of filters over the training set. This results in a straightforward selection of instances [12] based on the examples' level of confidence. We investigate the application of a set of thresholds over the target classes scores: examples with scores that do not reach the threshold are removed from the training set. By this, we carry out a cleaning of the training data based on the samples confidence, which is given by the transformed scores that, in turn, incorporate information about the uncertainty within the data.

IV. Experiments and analysis

In this section, we investigate the behaviour of the proposed approach in one of the most well-known citizen science projects, the first edition of the Galaxy Zoo (GZ1) project [13]. Subsection IV-A defines the experimental set-up, describing the main features of the GZ1 data and the methodology followed through the experiments. Subsection IV-B presents and discusses the obtained results.

A. Experimental Set-up

The GZ1 project was concerned with classification of galaxy images in terms of their morphology. Later editions of this project were launched, providing more specific information about the morphological features of the galaxies, but they involved a considerably reduced number of images [19]. At the time the project was finished, GZ1 had engaged more than 100,000 volunteers that completed over 40 million classifications for a sample of nearly 900,000 galaxy images [20]. The main focus was on the distinction between elliptical and spiral morphologies, turning it into a true binary classification problem. Nonetheless, various aspects complicated the classification for the amateurs taking part, which explains the uncertainty encountered within the GZ1 data. These included the varied quality of the images, depending on the distance to the galaxy, its physical size and brightness, or the presence of artifacts that hinder the class identification [13].

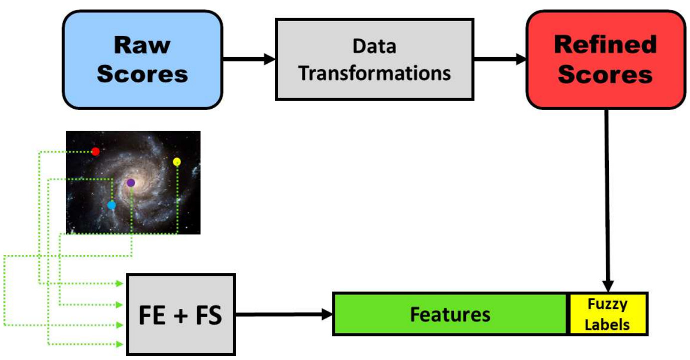

Fig. 1. Workflow of the proposed approach for the GZ1 classification problem.

The main GZ1 dataset[2] is composed of 667,944 galaxies for which there is available the galaxy ID in the Sloan Digital Sky Survey[3] database, total number of votes received, the set of original scores (referred from now on as *raw* scores) for all categories offered to participants in the web, and some more information (i.e. location in the sky, debiased scores; see [20] for more details). All images displayed held a common size of 423×423 pixels in order to provide a similar basis for all classifications, as described in [20]. However, the Fuzzy-kNN algorithm does not directly work on the images but on a feature set extracted from them. As the original images generally leave the galaxy in the centre of the image surrounded by a big amount of background pixels, we first crop the images to the half of their size (112×112 pixels), and then shrink them to a final size of 52×52 pixels. After some additional experiments that are outside of the scope of this contribution, we opted for a feature extraction (FE) based on auto-encoders (AEs) [21], employing a convolutional AE composed of three consecutive convolutional – pooling pairs of layers, from input to the encoding. This results in a feature set of 392 features extracted from the 52×52 images. Acknowledging that some of these features may be redundant or noisy, we decided to apply a randomised decision tree feature selection (FS) (50 trees) to choose the most relevant features. Figure 1 shows the general workflow of the proposed approach for the GZ1 classification problem.

Similarly to [7], we first make use of a subset of GZ1 composed of 41,424 galaxies (16,375 elliptical, 25,049 spiral) defined for the availability of expert classifications for these examples. In our experiments, we train the different k-NN variations on this sample, due to the high runtime required to handle the entire data. We refer to this sample as the GZ1 Validation (GZ1-V) subset. Nonetheless, the best performing approach is also tested on the whole GZ1 dataset. To test the adequacy of the data transformations, we make use of the transformation sequence that performed best in [7] (a combination of *Normalisation*, *DK votes shift*, and *Votes boost*, in this order). To assess how well the uncertainty is managed within an automatic classification process, we carry out all experimental trials using both raw and transformed scores, pursuing a first comparison amongst the unmodified data and the transformed data that incorporates the information about the uncertainty.

For the comparative study, we analyse the performance of the proposed approach using the raw and transformed scores as the fuzzy labels needed for completing the Fuzzy-kNN training, which we will refer from now on as Citizen Science Fuzzy-kNN (CSF-kNN). We compare the CSF-kNN approach against the Crisp k-NN (Crisp-kNN) algorithm and the original Fuzzy-kNN (i.e. using the algorithm's original fuzzification phase [8]). Therefore, for either the Fuzzy-kNN and the Crisp-kNN, we need to turn the scores into crisp labels. This is accomplished by a simply majority criterion: the class holding a greater score is assigned the positive label (1), whereas the other is labelled as negative (0). When a draw is encountered, which occurs for only 225 and 37 examples with raw and transformed scores, respectively, the labels are assigned at random.

We also analyse the impact of removing low confidence examples from the training set before running the algorithms detailed above. For this, we apply a set of thresholds (Th_{conf}) over both score types: raw and transformed. The mechanism is simple: examples with *both* elliptical and spiral scores strictly less than the selected threshold are removed.

The most well-known parameter for the Fuzzy-kNN is the number of neighbours (k) to be considered in order to take the final classification decision (Equation 2). We take k = 3 and m = 2.0 in our experiments. For the learning of the Fuzzy-kNN, we also take k_{memb} = 3 (Equation 1). The whole experimental setting specs are summarised in Table II. The final dataset is partitioned using a 5 fold cross-validation scheme to test all algorithms' performance with raw and transformed scores. As result metrics, we measure the accuracy and runtime required in each case.

[2]http://data.galaxyzoo.org
[3]http://www.sdss.org

TABLE II
PARAMETER SETTINGS OF THE ALGORITHMS INVOLVED

Algorithm	Parameters
Convolutional AE	3 × conv. – pooling layers ReLU activation function
Decision Tree FS	n_{trees} = 50
Crisp-kNN	k = 3 , m = 2.0
Fuzzy-kNN	$k = k_{memb}$ = 3, m = 2.0
CSF-kNN	k = 3, m = 2.0, Th_{conf} = {0.8, 0.9, 1.0}

All the experiments have been carried out in a single node with an Intel(R) Xeon(R) CPU E5-1650 v4 processor (12 cores) at 3.60GHz, and 64 GB of RAM. In terms of software, the Keras Python package was used for the training of the AE, and the Scikit-learn library[4] was used for the FS and the different experiments involving the training and testing of the different k-NN versions explained above.

B. Results and analysis

In this section, we present the results of all experimental trials performed to test the proposed approach. As detailed above, we first carry out a comparison between the algorithms using both raw and transformed scores for the two classes of the problem, elliptical and spiral. After this, we extend the testing of the CSF-kNN algorithm applying three thresholds that filter the training set. Finally, we complete the testing of the CSF-kNN on the GZ1 whole dataset after applying a stringent 1.0 threshold to both score types. This filter allows for selecting, between the whole dataset, the examples holding a perfect consensus amongst the amateurs who voted them.

The first experiment presents the comparative study amongst the Crisp-kNN, the Fuzzy-kNN, and the proposed CSF-kNN algorithms. The averaged accuracy and runtime are summarised in Table III.

TABLE III
ACCURACY AND RUNTIME VALUES FOR THE COMPARATIVE STUDY

Algorithm	Raw scores		Transformed scores	
	Accuracy	Runtime (s)	Accuracy	Runtime (s)
Fuzzy-kNN	0.8971	23,184	0.9166	23,192
Crisp-kNN	0.8958	5,607	0.9183	5,597
CSF-kNN	0.9004	3,617	0.9188	3,758

The second experiment introduces the instance selection on the training set before carrying out the training of the different algorithms. In each case, the selection is carried out applying the threshold either over raw or transformed scores, in such a way that if the example does not reach the threshold value in either of the elliptical or spiral class, it is rejected. Table IV presents the results when applying the thresholds Th_{conf} = {0.8, 0.9, 1.0}. At this respect, we reflect the proportion of the sample that is removed from the training set in Figure 2.

[4]https://scikit-learn.org

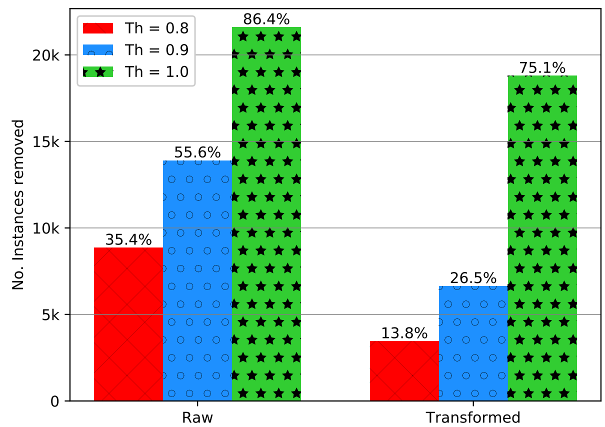

Fig. 2. Number of instances removed by the thresholds (Th). Percentages are shown on top of the bars.

TABLE IV
ACCURACY AND RUNTIME VALUES OF CSF-KNN WITH INSTANCE SELECTION

$\mathbf{Th_{conf}}$	Raw scores		Transformed scores	
	Accuracy	Runtime (s)	Accuracy	Runtime (s)
0.8	0.9032	2,354	0.9177	3,168
0.9	0.8997	1,497	0.9186	2,671
1.0	0.8820	476	0.9173	906

For a graphical comparison of the execution times, we represent the values corresponding to the first comparative study along with the CSF-kNN with Th_{conf} = 1.0 threshold in a bar chart. It is shown in Figure 3.

Finally, we present the values obtained when the CSF-kNN algorithm with 1.0 threshold instance selection is trained using the entire GZ1 dataset. The threshold is applied in order to speed up the execution up to a manageable runtime for this big dataset (∼893k examples). Table V presents the accuracy and execution times recorded for this last experiment.

TABLE V
RESULTS OF CSF-KNN WITH 1.0 THRESHOLD OVER THE WHOLE GZ1 DATASET

$\mathbf{Th_{conf}}$	Raw scores		Transformed scores	
	Accuracy	Runtime (s)	Accuracy	Runtime (s)
1.0	0.7616	23,254	0.8291	68,253

According to the results shown above, we can draw the following conclusions:

- In first place, the use of either raw or transformed scores is able to equalise the performance obtained by the

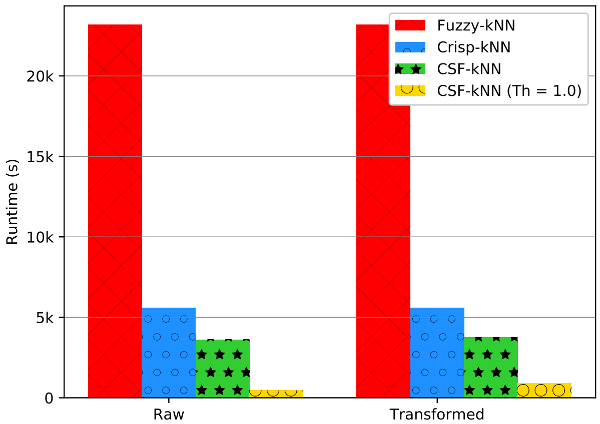

Fig. 3. Comparison of algorithms' runtime.

fuzzy and crisp implementations of the k-NN algorithm. However, the execution times are highly improved, as Figure 3 shows. The transformed scores also improve the accuracy in around two points with respect to the raw scores, confirming the utility of the data transformations presented in [7].

- The selection of instances across the training set by the application of thresholds provides a great reduction of the training times (Table IV), which is greater for the raw scores. This happens because the transformations tend to boost the raw scores for the major part of the GZ1-V examples. Nevertheless, the accuracy is not much affected.
- When the CSF-kNN algorithm is tested against the whole GZ1 dataset, we can observe that it generalises well. The transformed scores provide a better classification, making bigger the difference in accuracy with the raw scores.

V. Conclusions and Further Work

In this paper, we have proposed an innovative use of the data recorded in the course of a citizen science project to build robust and fast classifiers. The main novelty lies in the introduction of the classifications performed by amateur participants in both raw and refined forms as the fuzzy labels needed for the training of a Fuzzy k-Nearest Neighbours classifier. Throughout two sets of experiments on a real-world application, the proposed methodology has shown that a correct exploitation of citizen science results may lead to not only a good classification accuracy but also to a natural filter of noisy/imprecise training data. As future work, we consider the use of big data approaches to complete further experiments on larger datasets, and the use of other fuzzy-based classifiers that could take advantage of the methodology proposed here.

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. M. Jiménez holds a Ph.D. scholarship from the School of Computer Science of the University of Nottingham.

References

[1] J. Cohn, Citizen science: Can volunteers do real research?, BioScience 58 (3) (2008) 192–197.
[2] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
[3] M. Banerji, O. Lahav, C. Lintott, F. Abdalla, K. Schawinski, S. Bamford, et al., Galaxy zoo: Reproducing galaxy morphologies via machine learning, Monthly Notices of the Royal Astronomical Society 406 (1) (2010) 342–353.
[4] S. Dieleman, K. Willett, J. Dambre, Rotation-invariant convolutional neural networks for galaxy morphology prediction, Monthly Notices of the Royal Astronomical Society 450 (2) (2015) 1441–1459.
[5] H. Show, Rise of the citizen scientist, Nature 524 (7565) (2015) 265.
[6] M. Kosmala, A. Wiggins, A. Swanson, B. Simmons, Assessing data quality in citizen science, Frontiers in Ecology and the Environment 14 (10) (2016) 551–560.
[7] M. Jimenez, I. Triguero, R. John, Handling uncertainty in citizen science data: Towards an improved amateur-based large-scale classification, Information Sciences 479 (2019) 301–320.
[8] J. Keller, J. Givens, J.A., A fuzzy k-nearest neighbor algorithm, IEEE Transactions on Systems, Man and Cybernetics SMC-15 (4) (1985) 580–585.
[9] J. Derrac, S. Garcia, F. Herrera, Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects, Information Sciences 260 (2014) 98 – 119.
[10] J. Maillo, J. Luengo, S. Garcia, F. Herrera, I. Triguero, A preliminary study on hybrid spill-tree fuzzy k-nearest neighbors for big data classification, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018, pp. 1–8.
[11] I. Triguero, D. Garcia-Gil, J. Maillo, J. Luego, S. Garcia, F. Herrera, Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data, WIREs Data Mining and Knowledge Discovery 1289 (9) (2019) 1 – 24.
[12] S. Garcia, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: Taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 417–435.
[13] C. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, et al., Galaxy zoo: Morphologies derived from visual inspection of galaxies from the sloan digital sky survey, Monthly Notices of the Royal Astronomical Society 389 (3) (2008) 1179–1189.
[14] K. Crowston, C. Osterlund, T. K. Lee, Blending machine and human learning processes, in: Proceedings of the 50th Hawaii International Conference on System Sciences, HICSS'17, 2017, pp. 65–73.
[15] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, et al., Gravity spy: Integrating advanced ligo detector characterization, machine learning, and citizen science, Classical and Quantum Gravity 34 (6) (2017) 64003–64025.
[16] E. Kuminski, J. George, J. Wallin, L. Shamir, Combining human and machine learning for morphological analysis of galaxy images, Publications of the Astronomical Society of the Pacific 126 (944) (2014) 959–967.
[17] C. Beaumont, A. Goodman, S. Kendrew, J. Williams, R. Simpson, The milky way project: Leveraging citizen science and machine learning to detect interstellar bubbles, Astrophysical Journal Supplement Series 214 (1) (2014) 3.
[18] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.
[19] K. Willett, C. Lintott, S. Bamford, K. Masters, B. Simmons, K. Casteels, et al., Galaxy zoo 2: Detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey, Monthly Notices of the Royal Astronomical Society 435 (4) (2013) 2835–2860.
[20] C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, et al., Galaxy zoo 1: Data release of morphological classifications for nearly 900 000 galaxies, Monthly Notices of the Royal Astronomical Society 410 (1) (2011) 166–178.
[21] D. Charte, F. Charte, S. Garcia, M. del Jesus, F. Herrera, A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines, Information Fusion 44 (2018) 78–96.