Strategies for Tackling the Class Imbalance Problem in Marine Image Classification

Daniel Langenkämper[1], Robin van Kevelaer[1], and Tim W Nattkemper[1]

Biodata Mining Group, Faculty of Technology, Bielefeld University, 33615 Bielefeld, Germany dlangenk@cebitec.uni-bielefeld.de
https://www.cebitec.uni-bielefeld.de/biodatamining/

Abstract. Research of deep learning algorithms, especially in the field of convolutional neural networks (CNN), has shown significant progress. The application of CNNs in image analysis and pattern recognition has earned a lot of attention in this regard and few applications to classify a small number of common taxa in marine image collections have been reported yet.
In this paper, we address the problem of class imbalance in marine image data, i.e. the common observation that 80%-90% of the data belong to a small subset of L' classes among the total number of L observed classes, with $L' << L$. A small number of methods to compensate for the class imbalance problem in the training step have been proposed for the common computer vision benchmark datasets. But marine image collections (showing for instance megafauna as considered in this study) pose a greater challenge as the observed imbalance is more extreme as habitats can feature a high biodiversity but a low species density.
In this paper, we investigate the potential of various over-/undersampling methods to compensate for the class imbalance problem in marine imaging. In addition, five different balancing rules are proposed and analyzed to examine the extent to which sampling should be used, i.e. how many samples should be created or removed to gain the most out of the sampling algorithms. We evaluate these methods with AlexNet trained for classifying benthic image data recorded at the Porcupine Abyssal Plain (PAP) and use a Support Vector Machine as baseline classifier. We can report that the best of our proposed strategies in combination with data augmentation applied to AlexNet results in an increase of thirteen basis points compared to AlexNet without sampling. Furthermore, examples are presented, which show that the combination of oversampling and augmentation leads to a better generalization than pure augmentation.

Keywords: class imbalance · CNN · marine imaging · deep learning · taxonomic classification

1 Introduction

1.1 Motivation

The classification of objects is of central importance for a multitude of areas, e.g. autonomous driving, biodiversity studies, public surveillance, etc. With the

emergence of deep neural networks, especially convolutional neural networks (CNNs), computer science has made a great leap forward in solving this problem. In recent years, the performance on benchmark datasets has even surpassed human performance for the first time[9].

However, in contrast to natural images, i.e. images showing everyday objects, or even customized benchmark data such as ImageNet [4], biological data have some unique characteristics in contrast to those mentioned above. The main differences are a) data quality issues, b) lack of training data and c) class imbalance. The quality issues are mainly caused by the heterogeneity in capture setups, i.e. different light sources, occlusion, cast shadows, different camera angles, different camera equipment, development of the images (white balancing, etc.) and others. The lack of training data is inherent in that the captured objects are not everyday objects. It is, therefore, more difficult to encounter these objects and harder to annotate these. For everyday objects, citizen science solutions are a quick way to acquire a lot of valid annotations. Otherwise, trained experts are needed to acquire a limited number of error-prone annotations. The class imbalance, i.e.the common observation that 80%-90% of the training data belong to a small subset of L' classes among the total number of L observed classes, with $L' << L$, is usually present to varying degrees in biological data, e.g. established by prey and predator relationships, where the prey is more abundant than the predators.

Marine images are also a special type of this biological data. Data is scarce due to the high investment in equipment and difficult setup of the imaging system needed to acquire underwater imagery. The annotation problem is exacerbated by the high diversity - low abundance phenomena observed in the deep sea. Trained experts' time is limited, and citizen science projects are difficult to establish, although public interest is generally high. This annotation problem further skews the class imbalance, since easy to spot/annotate objects will be annotated much more frequently.

1.2 Prior Work

Different methods for compensating class imbalances exist. These are over- and undersampling of data[3, 8, 18], class weights/class aware loss functions/cost sensitive learning[5, 11] and postprocessing the output class probabilities also known as thresholding, which could be regarded as a special case of cost-sensitive learning[12, 16]. While class weights are dependent on the algorithm used, e.g. applicable for the SVM, over-/undersampling are applied before the classification is run and are therefore independent of the algorithm used. Class aware loss functions were proposed for example for some CNN types. They are a powerful instrument but are algorithm dependent and not easy to tune.

Prior work has been published to investigate the influence of class imbalance on machine learning algorithms, e.g. [2], but no investigation concerning the case of marine imaging is known to the authors. For a review have a look at [7].

2 Dataset

The images used in this study were recorded using an autonomous underwater vehicle (AUV) at the Porcupine Abyssal Plain (PAP)[13], located southwest of the UK in international waters. The image set is composed of 12116 images $\mathcal{I}_{i=0...12115}$. 30149 circular annotations $\mathcal{A}_j = (x, y, r, i, l)$ (with x, y being the center of the circle with radius r on image $\mathcal{I}_i$) divided into 19 classes, i.e. morphtoypes/taxa l (see Figure 1) were done by experts. These were used to extract rectangular image patches $\mathcal{P}_{j=0...30148}$ containing the annotated object. As can be seen in Figure 1 the distribution of the classes l is skewed, and a class imbalance problem is present.

For the SVM, features are generated by flattening the RGB patches from$\mathcal{P}_j \in \mathbb{N}^{30149\times\text{width}\times\text{height}\times 3}$ to $\mathcal{P'}_j \in \mathbb{N}^{30149\times(\text{width}*\text{height}*3)}$. Then dimensionality reduction using a PCA on the patches $\mathcal{P'}_j$ is applied to get the dataset $\mathit{\Gamma}_{\text{SVM}} = \{PCA(\mathcal{P'}_j)\} \in \mathbb{R}^{30149\times 64}$.

For the CNN the image patches $\mathcal{P}_j$were resized to patches P_j'' of size $64 \times 64 \times 3$. These form the dataset $\mathit{\Gamma}_{\text{CNN}} = \{\mathcal{P}_|''\} \in \mathbb{N}^{30149\times 64\times 64\times 3}$.

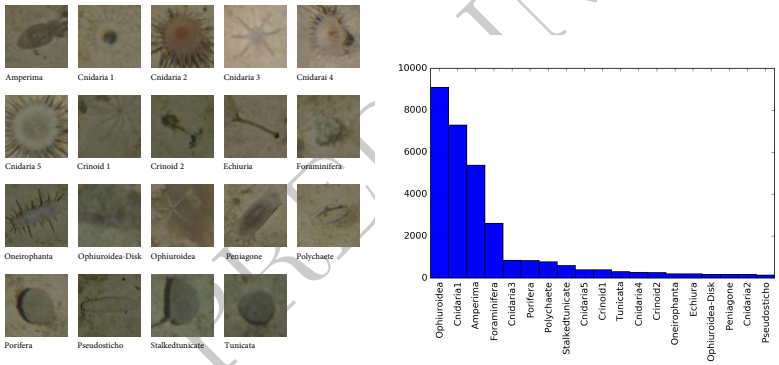

Fig. 1: Example image patches of all classes l and histogram of the classes

3 Methods

3.1 Over/Undersampling methods

Random Oversampling Random oversampling (ROS)[1] is a simple method designed to balance a dataset. With this method, $\mathcal{P}_j$ belonging to the classes that are to be oversampled are drawn with replacement and added to the data set until the desired class sizes are reached. This results in a larger data set that contains some of the $\mathcal{P}_j$ multiple times.

SMOTE Synthetic Minority Over-sampling Technique (SMOTE)[3] is an algorithm that generates synthetic samples from the existing ones of a dataset. It was originally formulated for two-class classification problems. If $s'(l)$ new samples are to be created for a class l, $s'(l)$ image patches $\{\mathcal{P}_j\}_{j=1,..,s'(l)} \subset \Gamma$ of this class are randomly selected. For each of these $\mathcal{P}_j$, the K nearest neighbors in Γ are estimated forming $P_{k=1,...,K}$, while K is a hyper-parameter and must be determined. One of these K nearest neighbors P_k is selected randomly. The new sample is determined to be:

$$\hat{P} = P_j + \lambda * (P_k - P_j) \tag{1}$$

with λ $[0, 1]$ being a random number.

ADASYN Adaptive Synthetic Sampling (ADASYN)[8] is an oversampling method that generates synthetic samples similar to the SMOTE algorithm and was originally formulated for a two-class classification problem. Unlike SMOTE, it does not select the sample pairs from which a new sample is generated only randomly, but according to the distribution of the data. The K nearest neighbors for every data point are computed. For each sample P_j of the minority class, the ratio $r_j = \frac{\delta_j}{k}$ with $\delta_j = |\{P_k : l(P_k) == P_{maj}\}|$ being the number of samples labeled with l_{maj} in the k neighborhood is determined.

All r_j are normalized $\hat{r}_j = r_j / \sum_j r_j$ so that the result is a probability distribution $\hat{r} = \sum_j \hat{r}_j = 1$. The number of synthetic samples s'_j that are generated for each P_j is computed as $s'_j = \hat{r}_j * s'(l)$. A new sample $\hat{P}$ is computed as follows:

$$\hat{P} = P_j + \lambda(P_k - P_j) \tag{2}$$

This algorithm results in more samples being created for a sample that has many neighbors from the majority class than for samples that have fewer such neighbors. If a sample has no neighbors from the majority class, no samples are created for it.

Data Augmentation The term *data augmentation* describes the application of different transformations to the training images. This can be e .g. the extraction of sections from images or flipping, rotations, or Gaussian blurring[10, 15, 14]. It can be used by temporarily creating randomly transformed copies of the training data during training and can therefore also be used additionally if the training set was previously oversampled. It has proven to be helpful to prevent overfitting and to improve classifier performance[10, 14].

Transformation Oversampling The image transformations used for data augmentation can also be applied as part of an oversampling method that can be employed to balance an imbalanced training dataset as proposed by Wang and Perez [15].

To balance the dataset the transformations are applied to image patches $\mathcal{P}_j$ from minority classes that are to be oversampled. Hereby, it is paid attention that if possible no transformations are multiply applied to the same $\mathcal{P}_j$, i.e. the same transformation is only applied multiple times to the same $\mathcal{P}_j$ if all the other transformations have already been used on it.

The transformations used here are a 90-degree rotation, Gaussian blur with $\sigma = 1$ and flipping using one of both image axes. In the following, this oversampling method is referred to as *Transformation Oversampling* (TROS).

Random Undersampling Undersampling means that image patches $\mathcal{P}_j$ are removed from the data. The method can be applied to the larger classes in an unbalanced dataset to reduce the imbalance. Random undersampling (RUS)[1] is a simple undersampling method that randomly removes $\mathcal{P}_j$ from all classes that are to be subsampled until the desired sample size $s(l)$ is reached.

3.2 Balancing rules

All the used sampling methods introduced above require a desired sample size $s(l)$ for each class l. The sample size is usually expressed as a percentage of the sample size $s(l)$ of the majority class l_{maj}, i.e. the class which is the most common.

The term *balancing rule* will be used to describe the rule that defines which sample size $s(l)$ a class should be sampled to in relation to $s(l_{\mathrm{maj}})$.

Three different rules for oversampling are introduced in this section $\{r_{100}, r_{50}, r_{50,75,100}\}$.

r_{100} is the most intuitive one setting the sample size $s(l)$ to the sample size of the majority class $s(l_{\mathrm{maj}})$.

$$r_{100} : s(l) = s(l_{\mathrm{maj}}) \tag{3}$$

When resampling imbalanced datasets, the synthetically generated samples are derived from only a small number of samples. Thus it may be the case that at some point generating more samples does not significantly increase the accuracy of the classifier trained on the dataset anymore. Additionally, there may be a loss of classification performance on the majority classes if all classes are sampled to the same size.

A solution for this may be oversampling rare classes to a size of $\frac{1}{2}s(l_{\mathrm{maj}})$ and keep the larger classes at their original size.

$$r_{50} : s(l) = \begin{cases} \frac{s(l_{\mathrm{maj}})}{2} & \text{if } s(l) < \frac{s(l_{\mathrm{maj}})}{2} \\ s(l) & \text{else} \end{cases} \tag{4}$$

This rule may increase the classification accuracy of the rare classes keeping that of the common classes reasonably high, thus preventing a high loss of average precision per class caused by misclassification of common classes.

Using the third rule $r_{50,75,100}$ the sample sizes $s(l)$ are divided up into three ranges.

$$r_{50,75,100} : s(l) = \begin{cases} \frac{1}{2}s(l_{\text{maj}}) & \text{if } s(l) \leq \frac{1}{4}s(l_{\text{maj}}) \\ \frac{3}{4}s(l_{\text{maj}}) & \text{if } \frac{1}{4}s(l_{\text{maj}}) < M_k \leq \frac{1}{2}s(l_{\text{maj}}) \\ s(l_{\text{maj}}) & \text{else} \end{cases} \tag{5}$$

In addition two rules $\{\hat{r}_{75}, \hat{r}_{50,100}\}$ combining oversampling with undersampling are evaluated. The first rule r_{75} completely balances the dataset, but decreases the variety of the largest classes by removing a certain share of their training samples randomly. Many of the synthetic minority class samples are generated from a small number of image patches $\mathcal{P}_j$. Because of this, the variance of these classes may be smaller than the variance of the majority classes even after oversampling. Applying this rule may reduce this difference.

$$\hat{r}_{75} : s(l) = \frac{3}{4}s(l_{\text{maj}}). \tag{6}$$

The other rule introduced here is adapted from a combined undersampling and oversampling approach introduced in [3]. The method mentioned there includes undersampling the majority class to half the size and oversampling the minority class to $s(l_{\text{maj}})$ in a two-class classification problem. This is extended to the multiclass classification problem at hand. The desired sample sizes $s(l)$ are computed as follows:

$$\hat{r}_{50,100} : s(l) = \begin{cases} \frac{s(l_{\text{maj}})}{2} & \text{if } s(l) \geq \frac{s(l_{\text{maj}})}{2} \\ s(l_{\text{maj}}) & \text{else} \end{cases} \tag{7}$$

3.3 Evaluation Metrics

The classification results are evaluated using the macro-averaged recall, precision[17] and the mean f1-score[6]. Macro-averaging means that the measure is first computed for each class separately, then the arithmetic mean of the per-class measures is computed to obtain a performance measure that is suitable for equally weighting all classes regardless of their sample sizes. If the average of the class-wise measures were weighted by class size, as usual, low scores for small classes would lower the average much less, while for common classes the loss would be much stronger. This is important to assess whether a classifier can classify rare classes as well as common classes.

The macro-averaged recall R_{macro} is defined as $R_{\text{macro}} = \frac{1}{L}\sum_l R(l)$ where $R(l)$ denotes the recall of class l.

The macro-averaged precision P_{macro} is defined as $P_{\text{macro}} = \frac{1}{L}\sum_l P(l)$ where $P(l)$ denotes the precision of class l.

To evaluate the overall classification performance, the macro-averaged f1-score $F_{1,\text{macro}}$, which is defined as $F_{1,\text{macro}} = \frac{1}{L}\sum_l F_1(l)$ with $F_1(l) = \frac{2R(l)P(l)}{R(l)+P(l)}$ where $F_1(l)$ is the class-wise f1-score, which is the harmonic mean of $P(l)$ and $R(l)$, with both values weighted equally.

$F_{1,\text{macro}}$		SMOTE	ADASYN	ROS	**TROS**
baseline	0.6868				
r_{50}		0.7571	0.7404	0.7416	0.7651
$\mathbf{r_{50,75,100}}$		0.7525	0.7432	**0.7445**	**0.7766**
r_{100}		0.7581	**0.7434**	0.7433	0.7621
$\hat{r}_{75}$		**0.7653**			0.7607
$\hat{r}_{50,100}$		0.7652			0.7578

(a)

R_{macro}		SMOTE	ADASYN	ROS	**TROS**
baseline	0.6585				
r_{50}		0.7225	**0.7082**	0.7266	0.7892
$r_{50,75,100}$		0.7332	0.7070	0.7249	0.7900
r_{100}		0.7250	0.7067	**0.7280**	0.7767
$\hat{r}_{75}$		0.7317			0.7888
$\mathbf{\hat{r}_{50,100}}$		**0.7400**			**0.7961**

(b)

P_{macro}		**SMOTE**	ADASYN	ROS	TROS
baseline	0.7345				
$\mathbf{r_{50}}$		**0.8159**	0.7915	0.7688	0.7495
$r_{50,75,100}$		0.7907	0.7985	**0.7739**	**0.7691**
r_{100}		0.8087	**0.8016**	0.7689	0.7563
$\hat{r}_{75}$		0.8153			0.7425
$\hat{r}_{50,100}$		0.8065			0.7334

(c)

	$F_{1,\text{macro}}$	R_{macro}	P_{macro}
baseline	0.6868	0.6586	0.7345
Only DA	0.7213	0.6989	0.7751
DA, SMOTE, r_{50}	0.8000	0.7903	0.8206
DA, TROS, $r_{50,75,100}$	0.7919	0.7847	0.8030
DA, SMOTE, $\hat{r}_{75}$	**0.8145**	0.8110	**0.8248**
DA, SMOTE, $\hat{r}_{50,100}$	0.8120	**0.8136**	0.8157

(d)

Table 1: CNN Results: Best results are shown in boldface.

4 Results

In table 1a) the results of the AlexNet classification using the different balancing rules compared to the classification results without any sampling (baseline) are shown. It is evident that sampling helps in increasing the classification performance significantly. The best results are achieved using TROS with the $r_{50,75,100}$ rule, which results in an increase of roughly 9 basis points for the $F_{1,\text{macro}}$ score. SMOTE oversampling combined with random undersmapling is almost as good comparing the $F_{1,\text{macro}}$ score (-1 basis point) but achieves a much higher macro precision than the aforementioned method (81.5% vs. 76.9%) at the cost of a much lower macro recall value (74% vs. 79,6%). ADASYN and ROS are underperforming therefore the undersampling experiments were not executed.

The results of combining the sampling methods with data-augmentation are shown in table 1 d). Here the runner-up from above – SMOTE combined with the $\hat{r}_{75}$ balancing rule is the best, which gains an additional 5 basis points using data-augmentation. Interestingly, data-augmentation without sampling only gains 3.5 basis points compared to the baseline.

Besides, as can be seen in Figure 2 according to the activations the AlexNet classifier tends to gain generalization performance, using oversampling in combination with data-augmentation compared to using pure data-augmentation. In this figure we can see that more, and also more unique filters are active and that the filters generated by the convolutional neural network are detecting more edges, or small details like the tentacles of the holothurian, rather than memorizing the whole holothurian.

Additionally, we investigated the influence of SMOTE and ADASYN oversampling on the SVM classifier. The SVM results are listed in table 3. It can be seen that oversampling is hurting the performance. This is unfortunately inherent with the way the SVM classifies and in that SMOTE and ADASYN are generating data. The SVM tries to find a separating hyperplane between data

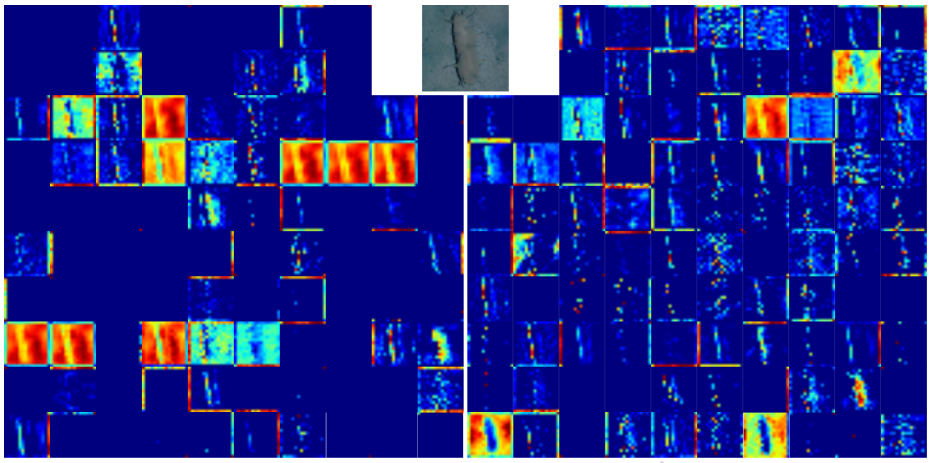

Fig. 2: Example plots of the activations of the first layer of AlexNet. The left image shows the activations for pure data augmentation and the right image of data augmentation combined with the best sampling approach (cmp. Table 1d).

classe	baseline F_1	DA, SMOTE, $\hat{r}_{75}$ F_1	class	baseline F_1	DA, SMOTE, $\hat{r}_{75}$ F_1
Amperima	0.9378	0.9662	Oneirophanta	0.812	0.9098
Cnidaria 1	0.9683	0.9784	Ophiuroidea	0.932	0.963
Cnidaria 2	0.731	0.7299	Ophiuroidea-Disk	0.3916	0.6063
Cnidaria 3	0.8043	0.9437	Peniagone	0.4663	0.6766
Cnidaria 4	0.78	0.8179	Polychaete	0.7276	0.8956
Cnidaria 5	0.8968	0.9013	Porifera	0.5811	0.7051
Crinoid 1	0.5423	0.8179	Pseudosticho	0.3807	0.7852
Crinoid 2	0.6063	0.7493	Stalkedtunicate	0.684	0.7913
Echiura	0.5178	0.7095	Tunicata	0.4244	0.5829
Foraminifera	0.864	0.9459			
			$F_{1,\text{macro}}$	0.6868	0.8145
			F_1	0.8841	0.935

Table 2: Single Class F_1 scores including total macro and weighted F_1 score

	SMOTE			ADASYN		
	R_{macro}	P_{macro}	$F_{1,\text{macro}}$	R_{macro}	P_{macro}	$F_{1,\text{macro}}$
baseline	**0.5571**	**0.6223**	**0.5796**	**0.5571**	**0.6223**	**0.5796**
r_{50}	0.5541	0.6095	0.5729	0.5525	0.6074	0.5711
$r_{50,75,100}$	0.5528	0.6123	0.5735	0.5494	0.6045	0.5684
r_{100}	0.5503	0.6036	0.5680	0.5475	0.5986	0.5643

Table 3: SVM results: Best results are shown in boldface.

points of different classes. SMOTE and ADASYN are introducing new data points in between data points of differently labeled data (cmp. equations 1 and 2). Therefore the data is placed near the separating hyperplane, thus increasing the number of support vectors (cmp. Figure 3) needed to establish a hyperplane still separating the differently labeled data, while still not gaining any better scores. This results in overfitting of the classifier.

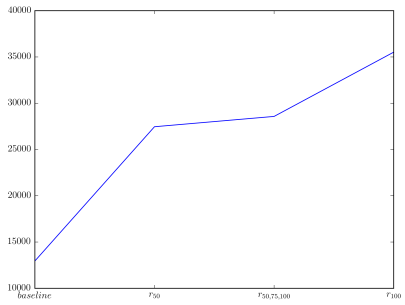

Fig. 3: Number of support vectors compared to the balancing rule applied.

5 Conclusion

To sum up the results of this thesis, it can be said that over-/undersampling is a method that is helpful to improve a classifier's result achieved on imbalanced marine image data. In contrast to other data domains combined over-/undersampling was only stronger than pure oversampling when combined with data augmentation. It was shown that over-/unddersampling is a well-suited method to improve the performance of a convolutional neural network, especially if it is combined with data augmentation. The balancing rules introduced and compared in this paper show a big improvement over the intuitive approach of oversampling every class to the maximum sampling size.

Which sampling algorithm and balancing rule to choose is a question of the desired result. Applying SMOTE alone, for example, yields a good precision while using TROS increases the recall more. If data augmentation is applied additionally to oversampling, the results are more balanced increasing the performance of rare classes. This leads to the best overall classification performance and increased generalization, which makes it recommendable to combine sampling with data augmentation.

Acknowledgment

Data were made available from National Oceanography Centre and made possible by funding from the Natural Environment Research Council (UK) through the 'Autonomous Ecological Surveying of the Abyss (AESA)' project (NE/H021787/1 to HA Ruhl and NE/H023569/1 to DM Bailey). We thank NVIDIA Corporation for donating the GPU used in this project. This project has received funding by Projektträger Jülich (grant no 03F0707C) under the framework of JPI Oceans.

References

1. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. **6**(1), 20–29
2. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. arXiv preprint arXiv:1710.05381 (2017)
3. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: Synthetic minority oversampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR 2009. pp. 248–255. IEEE (2009)
5. Elkan, C.: The foundations of cost-sensitive learning. In: IJCAI. vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
6. Ferri, C., Hernndez-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recogn. Lett. **30**(1), 27–38 (Jan 2009)
7. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications **73**, 220–239 (2017)
8. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: IJCNN 2008. pp. 1322–1328. IEEE (2008)
9. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV. pp. 1026–34 (2015)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
11. Kukar, M., Kononenko, I., et al.: Cost-sensitive learning with neural networks. In: ECAI. pp. 445–449 (1998)
12. Lawrence, S., Burns, I., Back, A., Tsoi, A.C., Giles, C.L.: Neural network classification and prior class probabilities. In: Neural networks: tricks of the trade, pp. 299–313. Springer (1998)
13. Morris, K.J., Bett, B.J., Durden, J.M., et al.: A new method for ecological surveying of the abyss using autonomous underwater vehicle photography. Limnol. Oceanogr.: Methods **12**, 795–809 (2014)
14. Pawara, P., Okafor, E., Schomaker, L., Wiering, M.: Data augmentation for plant classification. In: Acivs 2017. pp. 615–626. Springer (2017)
15. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. CoRR (2017), http://arxiv.org/abs/1712.04621
16. Richard, M.D., Lippmann, R.P.: Neural network classifiers estimate bayesian a posteriori probabilities. Neural computation **3**(4), 461–483 (1991)

17. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Process. Manage. **45**(4), 427–437 (Jul 2009)
18. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics (3), 408–421 (1972)