

Article

Boosting Minority Class Prediction on Imbalanced Point Cloud Data

Hsien-I Lin *  and Mihn Cong Nguyen

Graduate Institute of Automation Technology, National Taipei University of Technology, No. 1, Sec. 3, Zhongxiao E. Rd., Taipei 10608, Taiwan; minhnc.edu.tw@gmail.com

* Correspondence: sofin@ntut.edu.tw; Tel.: +886-2-27713343

Received: 23 December 2019; Accepted: 26 January 2020; Published: 2 February 2020



Abstract: Data imbalance during the training of deep networks can cause the network to skip directly to learning minority classes. This paper presents a novel framework by which to train segmentation networks using imbalanced point cloud data. PointNet, an early deep network used for the segmentation of point cloud data, proved effective in the point-wise classification of balanced data; however, performance degraded when imbalanced data was used. The proposed approach involves removing between-class data point imbalances and guiding the network to pay more attention to majority classes. Data imbalance is alleviated using a hybrid-sampling method involving oversampling, as well as undersampling, respectively, to decrease the amount of data in majority classes and increase the amount of data in minority classes. A balanced focus loss function is also used to emphasize the minority classes through the automated assignment of costs to the various classes based on their density in the point cloud. Experiments demonstrate the effectiveness of the proposed training framework when provided a point cloud dataset pertaining to six objects. The mean intersection over union (mIoU) test accuracy results obtained using PointNet training were as follows: XYZRGB data (91%) and XYZ data (86%). The mIoU test accuracy results obtained using the proposed scheme were as follows: XYZRGB data (98%) and XYZ data (93%).

Keywords: deep learning; imbalanced data; hybrid-sampling; balanced focus loss function

1. Introduction and Motivation

The data imbalance commonly encountered in deep network training can have a profound effect on the training process and detection capability of the network [1]. Data imbalance refers to situations where the classes are not represented equally [2]. Specifically, the imbalance can be found in the number of data points pertaining to separate objects in a given point cloud. Furthermore, the number of data points pertaining to the objects differs significantly from the number of data points pertaining to the background. Under these conditions, the network often learns to detect only the background and large surface objects; i.e., it tends to skip smaller objects.

Growing interest in deep learning has brought the problem of data imbalance to the foreground, particularly in the field of data mining [3], medical diagnosis [4], the detection of fraudulent calls [3], risk management [5–7], text classification [8], fault diagnosis [9,10], anomaly detection [11,12], and face recognition [13]. Conventional machine learning models, i.e., non-deep learning, have been extensively applied in the study of class imbalance; however, there has been relatively little work using deep learning models, despite recent advances in this field [3,14]. For the imbalanced data problem, there were three main methods: data-based, algorithm-based, and ensemble methods.

1.1. Data-Based Methods

Data-based methods use sampling methods to rebalance the distribution of classes during pre-processing. This involves either oversampling instances of the minority class or undersampling instances of the majority class. Oversampling involves the random duplication of instances from minority classes [15–17]. Undersampling involves the random removal of instances from majority classes. Schemes that use oversampling in conjunction with undersampling are referred to as hybrid-sampling. These techniques are meant to produce a balanced dataset in which classifiers would tend not to be biased toward one class or another. However, in practical situations, this is not always the case. Oversampling minority classes can lead to overfitting through the duplication of instances drawn from an already small pool. Undersampling majority classes often leads to the exclusion of important instances required to differentiate between two classes. This has led researchers to develop more complex methods referred to as synthetic minority oversampling techniques (SMOTE) [18,19]. This approach can reduce the risk of data loss and overfitting; however, it is still prone to over-generalization or variance [20,21].

1.2. Algorithm-Based Methods

Algorithm-based methods emphasize minority classes. One popular strategy is cost-sensitive learning [22–25], in which a cost of variable value is assigned to different classes. In regular learning, the equal treatment of all misclassifications can lead to the problem of imbalanced classification, due to the lack of an additional reward for identifying a minority class over a majority class. Cost function-based methods overcome this issue using a function $C(p, t)$ that specifies the cost of misclassifying an instance of class t as class p . This makes it possible to penalize misclassifications of a minority class more heavily than those of the majority class with the aim of increasing the true positive rate. One common scheme involves assigning a cost equal to the inverse of the dataset proportion attributable to a given class. This leads to an increased degree of penalization with a decrease in class size.

Another strategy is the threshold-moving technique in which the decision threshold is shifted in a manner that reduces bias towards the negative class [15–17,26]. It applies to classifiers that, given an input tuple, return a continuous output value. Rather than manipulating the training tuples, this method returns a classification decision based on output values. In the simplest form, in tuples for which $f(X) \geq t$, and t is considered positive, all other tuples are considered negative.

1.3. Ensemble Methods

Ensemble methods involve the combination of data- and algorithm-based methods to overcome the problem of class imbalance [27–30]. One strategy involves data sampling to reduce class noise and imbalance, followed by cost-sensitive learning or thresholding, to enable a further reduction in bias towards the majority group. Several techniques presented in Reference [22] combine ensemble methods with sampling and cost-sensitive learning. Liu et al. [23] proposed two algorithms, EasyEnsemble and BalanceCascade, which learn multiple classifiers by combining subsets of the majority group with those of the minority group, to create pseudo-balanced training sets for each individual classifier. SMOTEBoost [24] introduced synthetic instances using SMOTE data preprocessing algorithms. The weights of the new instances in a dataset are proportional to the number of instances. RUSBoost [25] performs similarly to SMOTEBoost, but it removes instances from majority classes by random undersampling datasets in each iteration. Thus, it does not need to assign new weights to new instances. DataBoost-IM [31] and JOUS-Boost [32] are both examples of combining sampling with ensembles. DataBoost-IM combines AdaBoost.M1 algorithm with a data generation strategy. It identifies hard examples and then carries out a re-balanced process.

Sun et al. [33] introduced three cost-sensitive boosting methods (AdaC1, AdaC2, and AdaC3), which introduce cost functions to update the weights of the AdaBoost algorithm to increase the impact

of the minority group. Sun showed that the ensembles boosted in a cost-sensitive manner outperformed conventional boosting methods in most cases. The drawback of this method is the complexity of the cost function, which leads to a low computation speed in implementations. Bagging-based ensemble methods are developed to deal with the imbalanced problem because of its simplicity and good generalization. The concept is to obtain a useful classifier in each iteration by memorizing the importance of the diversity. In addition, some prominent methods are proposed, such as OverBagging [34], UnderBagging [35], and IIVotes [36]. Galar et al. [37] reviewed the performance and complexity of some ensemble methods for the imbalanced problem. The RUSBoost and UnderBagging methods seem to be more robust than others. However, considering the computational complexity, RUSBoost is the most appropriated ensemble method. In addition, bagging techniques are commonly used because they are not only easy to develop, but also powerful when dealing with imbalanced classes. In the above-mentioned methods, models trained using an imbalanced dataset presented a pronounced inclination toward classes as the accuracy was refined using conventional learning algorithms. Thus, some researchers developed inductive classifiers to decrease the number of training-based faults by overlooking classes with a limited number of instances [38].

Inspired by ensemble methods, we developed a two-stage scheme to deal with the data imbalanced problem in the segmentation task using the PointNet network [39]. We used the PointNet deep network as the learning model because it recently became a popular method by dealing with point cloud data directly, which decreased a lot of time-consuming and efforts of data pre-processing. The novelty of our approach is to propose a framework to integrate the novel hybrid-sampling method that eases the imbalanced problem in the training of the deep segmentation network using point cloud and a novel loss function automatically assigning varying costs according to instance probabilities of classes in the batch sample. Within this framework, smaller objects with fewer data points would be assigned a higher cost-factor, whereas larger objects with a greater number of data points would be assigned a lower cost-factor. The easily implemented hybrid-sampling scheme can improve performance without enlarging the network or using additional weight parameters. Furthermore, our use of a simple training process is more efficient than baseline training schemes.

The proposed scheme simultaneously applies oversampling to minority classes and undersampling to majority classes in order to correct for imbalances in the number of instances associated with the various classes. Training is conducted in three rounds. The first round involves training the network using sampled data with a large ratio of oversampling. The second round uses sampled data with a small ratio of oversampling. The third and final round uses normal data without oversampling or undersampling. The proposed cost-sensitive algorithm also includes a novel loss function (referred to as a balanced focus loss function), which automates the assignment of costs in accordance with the probability that a given class will occur in the batch sample.

We employed the PointNet deep network as our learning model for its ability to deal with point cloud data directly, thereby enhancing computational efficiency by eliminating the need for pre-processing. The mean intersection over union (mIoU) test accuracy results obtained using PointNet training were as follows: XYZRGB data (91%) and XYZ data (86%). The mIoU test accuracy results obtained using the proposed scheme were as follows: XYZRGB data (98%) and XYZ data (93%). Our prediction results are clearly more accurate and stainless than the baseline results.

The goal of this paper is to improve the segmentation deep network using imbalanced point cloud data. Our main contributions are as follows.

- A novel two-stage scheme is proposed, which combines the hybrid-sampling method and the balanced focus loss function to improve object segmentation using imbalanced point cloud data.
- A novel two-stage scheme outperforms either sampling or loss function technique for the imbalanced problem.
- The mIoU test accuracy results obtained using the proposed method outperforms the baseline method (PointNet) by 6%.

The remaining sections are organized as follows. Section 2 describes in detail the proposed framework using hybrid-sampling and balanced focus loss function. Section 3 outlines experiment results. Concluding remarks and future research are presented in Section 4.

2. Framework of Training Network for an Imbalanced Point Cloud

As long as there is no significant difference in the amount of data for each class, deep networks for point clouds (e.g., PointNet) are very effective. However, the number of points in the various classes is not necessarily equal, and the disparity between the background and specific small objects can be enormous. This makes it impossible for the network to learn effectively. Furthermore, the fact that minority-class point clouds are disregarded makes it impossible to segment the point clouds of the small objects out of the scene.

We sought to overcome this matter using two methods: (1) hybrid-sampling to reduce imbalances in the data, and (2) a balanced focus loss function to direct the network. In the following section, we present the proposed data sampling method and our underlying motivation. We then describe the proposed loss function and its weaknesses. Finally, we outline the proposed two-stage scheme combining the two methods.

2.1. Hybrid-Sampling

The proposed hybrid-sampling method undersamples points associated with the background and oversamples points associated with objects. The concepts of oversampling, undersampling, and hybrid-sampling are illustrated in Figure 1.

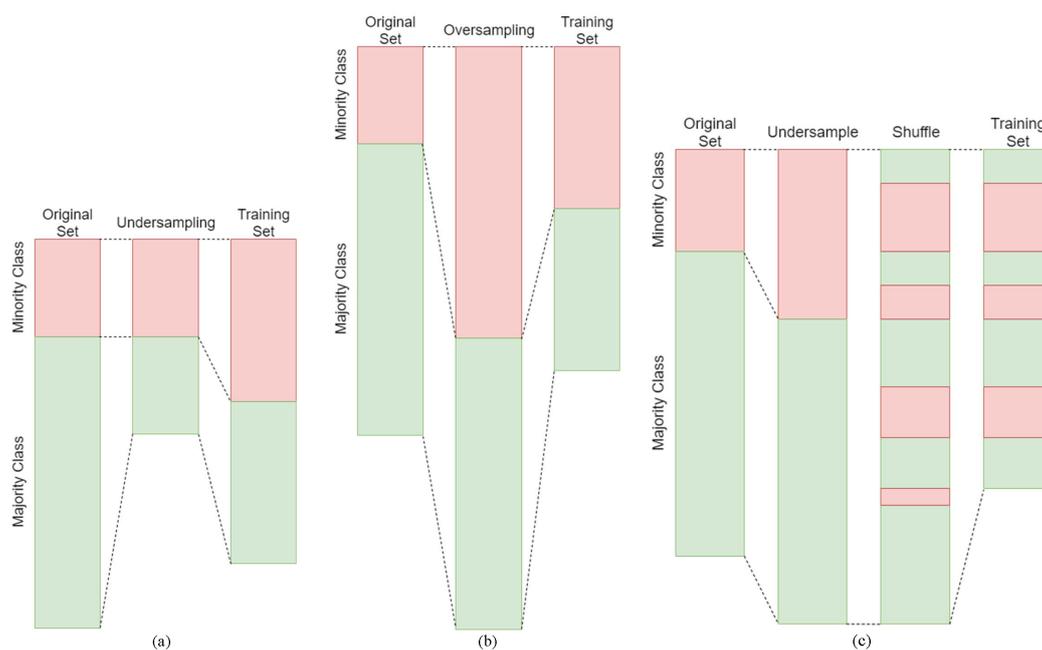


Figure 1. Sampling methods: (a) Undersampling: The majority class (green) is downsampled to make it equal in size to the minority class (red), and the resulting set is downsampled to fit the network; (b) oversampling: The minority class (red) is upsampled to make it equal in size to the majority class (green), and the resulting set is downsampled to fit the network; (c) hybrid-sampling: The minority class (red) is upsampled slightly, whereupon the resulting dataset undergoes random mixing of the majority and minority classes, followed by truncation to ensure that the result is compatible with the size of the network.

2.1.1. Undersampling

We first differentiate the minority and majority classes from an unbalanced set of data. Some of the instances of the majority classes are removed to ensure that the final instance number is equal to the instance number of minority classes. Note that the instance number of minority classes remains unchanged. Despite the fact that this results in a balanced set of data, the size of the dataset is not necessarily compatible with the network input. It is, therefore, necessary to implement an additional step involving the duplication or removal of data (evenly balanced between minority and majority classes) to fit the network.

2.1.2. Oversampling

As with the undersampling method, we first differentiate the minority and majority classes. However, we duplicate the instances of the minority classes (instead of undersampling the instances of the majority classes to ensure that there is an equal amount of data associated with both classes. Despite the fact that this results in a balanced set of data, the size of the dataset may be too large for the network input. Remedying this situation requires down sampling.

2.1.3. Conventional Hybrid-Sampling

The duplication of minority classes follows guidelines derived for a given situation by an expert in the field. The resulting mixed dataset is down sampled to make it compatible with the size of the network input. The data then undergoes shuffling and random downsampling to ensure that the training data is highly variable, thereby moderating the effects of data loss due to undersampling and overfitting due to upsampling.

2.1.4. Proposed Hybrid-Sampling

The experiments using Naive Bayes and C5.0 in Reference [40] demonstrated that oversampling and undersampling had equivalent effects on performance. In Reference [41], it was reported that oversampling was more robust than undersampling. In contrast, experiments conducted using C4.5 decision trees by Drummond [42] and other research [43,44] indicated that undersampling had a more pronounced effect than oversampling. In general, undersampling is preferred instead of oversampling when dealing with large amounts of data [45].

Shelke et al. [46] demonstrated that combining oversampling of the minority class with undersampling of the majority class could improve classifier performance (in ROC) to a greater degree than could be achieved by only undersampling the majority class. In intensive experiments, Seiffert et al. [47] demonstrated that hybrid-sampling can often outperform single sampling methods. They reported that oversampling, followed by undersampling was slightly more effective than the inverse. Table 1 presents a comparison of the methods described in Reference [46,47].

Table 1. Rank of sampling methods according to performance. Random Over-Sampling (ROS) is oversampling method, Random Under-Sampling (RUS) is under sampling method, RUS-ROS is method of undersampling then oversampling, and ROS-RUS is method of oversampling and then undersampling.

Method	Rank of Performance
ROS	IV
RUS	III
RUS-ROS	II
ROS-RUS	I

The proposed hybrid-sampling method involved oversampling, followed by undersampling. The proposed scheme differs from previous hybrid-sampling methods by the way that majority data

is mixed with minority data and shuffled to create a new representative dataset. After upsampling, the minority data undergoes shuffling with the majority data, all of which is then downsampled to make it compatible with the network input. Unlike previous methods, the ratio of minority to majority data after balancing is not a fixed value, but rather oscillates slightly in accordance with the shuffling process to make it more varied.

The pseudocode of the proposed algorithm is presented as Algorithm 1. The point cloud of the background is isolated from the lists of point sets and labels. The point cloud of the background is assigned a label value of 1, whereas the point clouds of objects are assigned labels values of >1. The indices of objects and that of the background are derived separately. The indices of objects are then duplicated according to a sampling ratio input by an expert in the field, where the sampling ratio is determined by the degree of data imbalance of datasets; to be less dependent on the ratio, we propose the balanced focus loss function in the next subsection. The new set of object indices is concatenated to the indices of the background and then shuffled. Finally, the shuffled set is separated from the last part, and the first part is kept as the selected part. From this selected index, we obtain a set of points with a set of corresponding labels as training data for the network.

Algorithm 1 Algorithm for hybrid-sampling algorithm to sample point clouds.

Input: A list of points *Points*, a list of labels *Labels* that are imbalanced, and an integer value of sampling factor *n_sampling*, and an integer value of number of points *n_points*.

Output: New list of points *Points* and new list of labels *Labels* that are already sampled.

Initialization :

- 1: *ObjIndices* ← *argwhere(labels > 1)*
- 2: *BgrIndices* ← *argwhere(labels = 1)*
- 3: *Choices* ← []

LOOP Process

- 4: **for** *i* ← 2 → *n_sampling* **do**
 - 5: *Stack(Choices, ObjIndices)*
 - 6: **end for**
 - 7: *Stack(Choices, BgrIndices)*
 - 8: *Shuffle(Choices)*
 - 9: *Choices* ← *Choices[: n_points]*
 - 10: *Points* ← *Points[: n_points]*
 - 11: *Labels* ← *Labels[: n_points]*
 - 12: **return** *Points, Labels*
-

2.2. Balanced Focus Loss Function

In this study, we considered three loss functions: the normal categorical cross-entropy loss function, a weighted loss function, and a focus loss function. The normal categorical cross-entropy loss function treats all classes and all misclassifications equally, and there is no additional reward for identifying the minority class over the majority class. This can lead to the misclassification of the point clouds of minority classes. The categorical cross entropy loss function is defined as follows:

$$CE(\theta) = - \sum_{i=1}^n y_i \log(\hat{y}_i(x_i, \theta)), \quad (1)$$

where $\hat{y}(x, \theta)$ is the posterior probability obtained by applying the softmax function to the network output layer, and θ denotes the trainable parameters of the network; x is the input sample, and y is the ground truth.

One common strategy that helps to increase the true positive rate is to penalize misclassifications of the minority class more heavily than those of the majority class. Thus, a weighted loss function is used to assign each class a ratio that is smaller for the background than for small objects. Unfortunately, the highly sensitive process of assigning an appropriate ratio set can be time-consuming. Any given set of ratios with a fixed value cannot be effective for all cases. Furthermore, the ratio of class instances varies according to the situation. The weighted loss function is formulated as follows:

$$WCE(\theta) = - \sum_{i=1}^n \alpha_i y_i \log(\hat{y}_i(x_i, \theta)), \quad (2)$$

where α is the penalization weight for false negative errors, which is usually selected manually and based on the experience of experts.

The balanced focus loss function presented in this study to integrate categorical cross-entropy and weighted loss functions is defined as follows:

$$E_{BL}(\theta) = - \sum_{i=1}^n (1 - \hat{y}(x, \theta))^\gamma \alpha_i y_i \log(\hat{y}_i(x_i, \theta)), \quad (3)$$

where $(1 - \hat{y}(x, \theta))^\gamma$ is the penalization weight for the class, $\gamma > 0$ is a tunable focusing parameter, and α is the factor emphasizing the minority classes according to the ratio of each the classes in that particular, formulated as follows:

$$\alpha(c) = \frac{1}{dist_prob(c)}, \quad (4)$$

where $dist_prob(c)$ is the distribution probability of class c in training batch, calculated as

$$dist_prob(c) = \frac{1}{length(y)} \sum (y == c). \quad (5)$$

2.3. Two-Stage Framework

Either sampling or loss function technique has not been shown not to have a significant effect on training performance. Furthermore, any misuse of those methods can lead to instability in the training process. Thus, we developed a two-stage framework combining hybrid-sampling with a balanced focus loss function to leverage the benefits of both methods. The network is trained through a number of epochs using balanced focus loss with a descending sampling ratio, as shown in Figure 2.

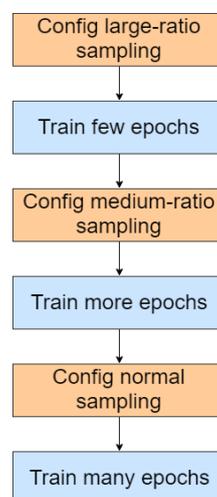


Figure 2. Descending sampling ratio.

At the beginning of training, a large sampling ratio is first used to lead the network to learn more about the minority classes. Note that the sampling ratio should not be so high that it causes excessive fluctuations in the training metric curve. The influence of the balanced focus loss function is not particularly pronounced; therefore, the sampling ratio is decreased gradually to shift the attention to the background. Eventually, when the sampling process halts, the network is trained using only the original dataset with the balanced focus loss function.

3. Experimental Validation

Experiments were conducted to compare the efficacy of the proposed two-stage framework with that of baseline methods when applied to imbalanced data. Figure 3 illustrates the system setup. The system table included two pipes and three wrenches, above which, at a distance of 67 cm, was an RGB-D camera (Orbbec Astra S) with resolution of 480×640 pixels. The experiments were validated using Python on the Tensorflow-Keras framework. The network was trained on a NVIDIA Geforce 1050-4Gb graphical card. In the following example, the point numbers of the background, pipe, wrench, yellow cube, and cyan cube were 25655 (82.02%), 1653 (5.28%), 1859 (5.94%), 518 (1.66%), and 1596 (5.10%), respectively.

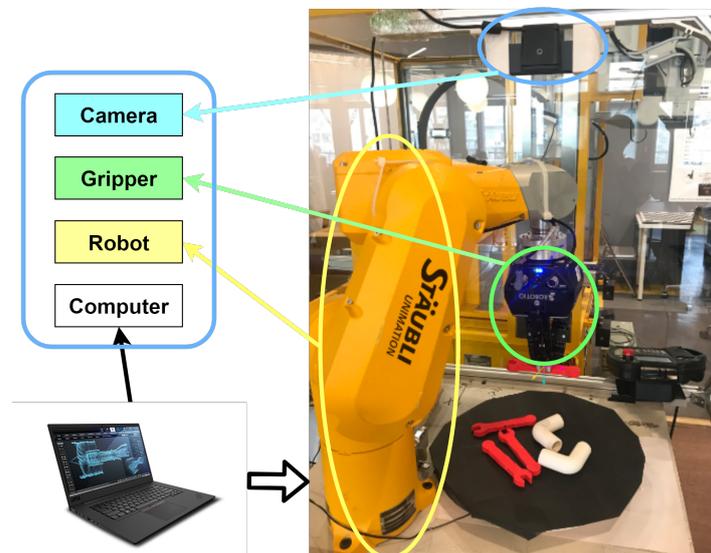


Figure 3. Hardware system.

3.1. Evaluation Metric

When dealing with imbalanced data, it is essential to find a metric that can be used to generate a generalized model for the segmentation of point cloud data. Our metric in this study was the median of intersection over unit (mIoU), which is calculated as follows:

$$mIoU = \frac{1}{C} \sum_{c=1}^C IoU(c) \quad (6)$$

and

$$IoU(c) = \frac{TP}{TP + FN + FP'} \quad (7)$$

where $IoU(c)$ is the intersection over a unit of class c , TP is a true positive, FN is a false negative, FP' is a false positive, and C is the total number of classes.

3.2. Baseline Methods

The deep network used in this study was the full version of PointNet [39]. Four training methods were included in the performance comparison.

1. PointNet without hybrid-sampling and a balanced focus loss function.
2. PointNet with a balanced focus loss function.
3. PointNet with hybrid-sampling.
4. PointNet with both hybrid-sampling and a balanced focus loss function.

The PointNet architecture included a feature-extraction module, a feature aggregation module, and a point-wise classification module. Note that each feature-extraction module had a feature transformation block and a multi-layer perceptron block. Each feature transformation block included six hidden layers. The first feature transformation block included the following number of nodes: [64, 128, 1024, 512, 256, n], where n indicates the number of input channels of the point cloud. The second feature transformation block included the following number of nodes: [128, 256, 1024, 512, 256, 128]. The first multi-layer perceptron included the following number of hidden nodes: [64, 128, 128]. The second multi-layer perceptron included the following number of hidden nodes: [512, 2048]. The point-wise classification module included the following number of hidden nodes: [512, 256, 128, c], where c indicates the number of categories.

3.3. Comparison Results

This comparison was conducted using two types of point cloud data: XYZRGB and XYZ.

3.3.1. XYZRGB Point Cloud Data

There were 16,384 XYZRGB points in the cloud data, such that the dimensions of the input data were [16384; 6]. The number of points in each image varied between 30,000 and 40,000. The number of points of objects in each scene varied between 400 and 4000. The alpha value for the balanced focus loss function was calculated automatically in each training batch.

Figure 4 illustrates the mIoU curves obtained using the four methods with 100 epochs. The blue validation curve (baseline training method) reached a maximum value of 91.74% at epoch 37. Note that it reached saturation quickly and fluctuated around 80%. The green curve obtained using the balanced focus loss function reached 96.35% at epoch 62, i.e., 5% higher than the baseline method with less fluctuation. The red curve obtained using hybrid-sampling fluctuated considerably; however, it reached 97.55% accuracy at epoch 60, i.e., 1.2% higher than the balanced focus loss method, and 5.81% higher than the baseline method. Finally, the pink curve obtained using the two-stage framework combining hybrid-sampling with balanced focus loss reached 98.26% accuracy at epoch 97, i.e., 1.91% higher than balanced focus loss, 0.71% higher than hybrid-sampling, and 6.52% higher than the baseline training method.

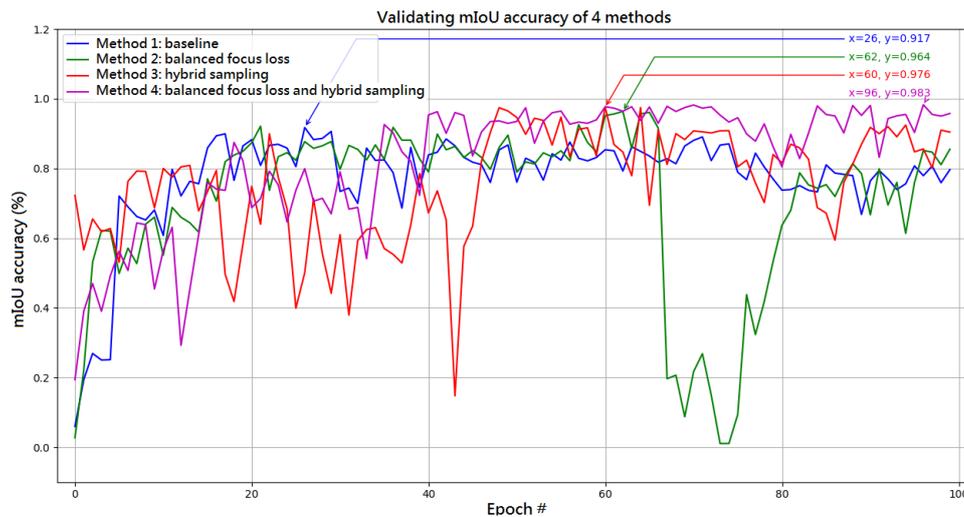


Figure 4. Comparison of mean intersection over union (mIoU) curves obtained using the four methods to process XYZRGB point cloud data.

Figure 5a–c present the ground truth of a point cloud, the prediction results using the baseline PointNet method, and the prediction results using the proposed two-stage framework when using XYZRGB data. The pink points indicate a single small cuboid. Clearly, the prediction result using the baseline method was incorrect, as the small cuboid was recognized as a pipe, as indicated by the red points. The proposed framework correctly recognized the small cuboid, as indicated by the pink points. This is a clear indication that the proposed two-stage framework outperformed the baseline PointNet training method when using XYZRGB data.

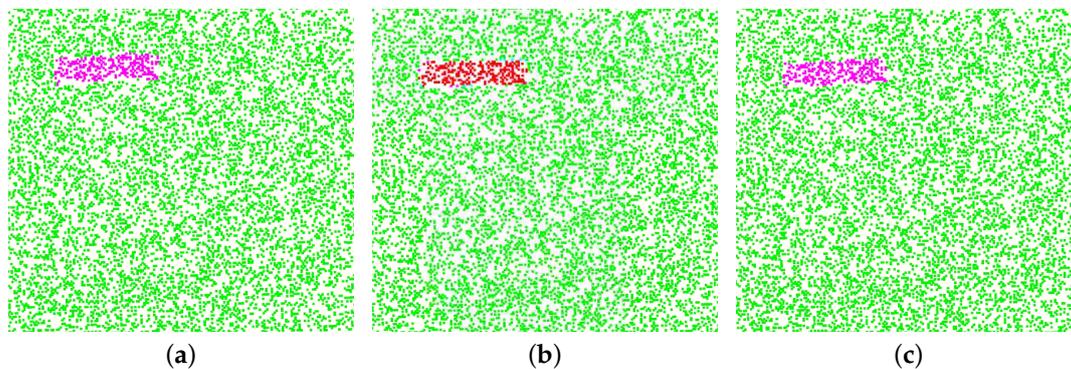


Figure 5. Comparison of ground truth and prediction results obtained when XYZRGB point cloud data was applied to the baseline or proposed training methods. The prediction results from the baseline training method were clearly erroneous: (a) ground truth; (b) point cloud predicted using conventional training method, showing erroneous prediction where the small cube is predicted as the pipe; (c) point cloud predicted using the proposed hybrid training method.

A further experiment was conducted to evaluate the network in the case of multiple objects. Figure 6a–c compare the ground truth, the baseline training method, and the two-stage framework. In this experiment, the results of the baseline method and the proposed framework were both reasonably good, i.e., most of the point clouds of the objects were detected correctly. Nonetheless, the prediction results obtained using the baseline method in Figure 6b revealed a number of pink points on the point cloud predicted for the wrench. The prediction results obtained using the proposed framework in Figure 6c do not include any points due to noise at the border of the predicted point cloud.

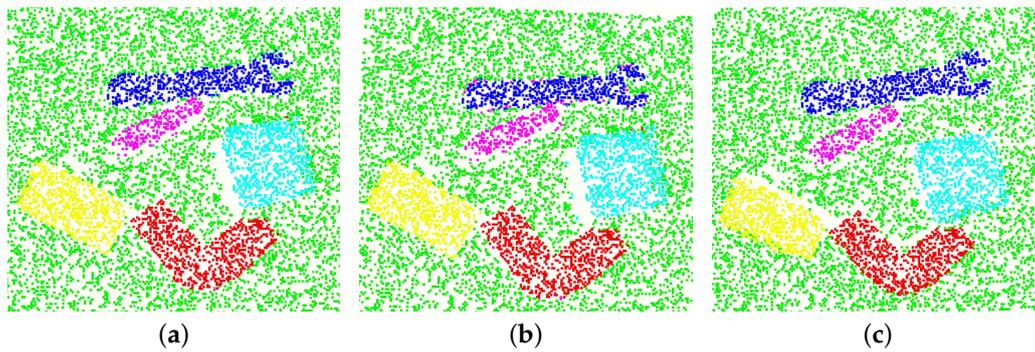


Figure 6. Comparison of ground truth and prediction results obtained when XYZRGB point cloud data was applied to the baseline or proposed training methods. The prediction results obtained using the baseline training method were affected more by noise than were the results obtained using the proposed method: (a) ground truth; (b) point cloud predicted using conventional training method, showing erroneous prediction on the wrench; (c) point cloud predicted using the proposed hybrid training method.

3.3.2. XYZ Point Cloud Data

Experiments were conducted to assess the efficacy of the four methods when applied to XYZ point cloud data. The number of input points was 4096; therefore, the dimensions of the input data were [4096; 3]. Figure 7 presents the mIoU curves of the four training methods. The curves obtained from XYZ data presented more pronounced fluctuations than did the training curves obtained using XYZRGB data. This can be attributed to the fact that the XYZ dataset did not include color elements, and there were only 25% as many data point. As shown in Figure 7, the blue curve obtained using the baseline training method achieved accuracy of 86.23% (at epoch 78), which is 5% below the accuracy obtained using XYZRGB data. The green curve obtained using the balanced focus loss function achieved accuracy of 88.20% accuracy (at epoch 47), which is 2% better than baseline method. The red curve obtained using the hybrid-sampling method achieved accuracy of 90.08%, which is 4% higher than the baseline method and slightly better than the balanced focus loss function. The pink curve obtained using the hybrid-sampling method with the balanced focus loss function achieved accuracy of 92.62%, which is 6.39% higher than the baseline method, 2% higher than hybrid-sampling, and 4% higher than the balanced focus loss function.

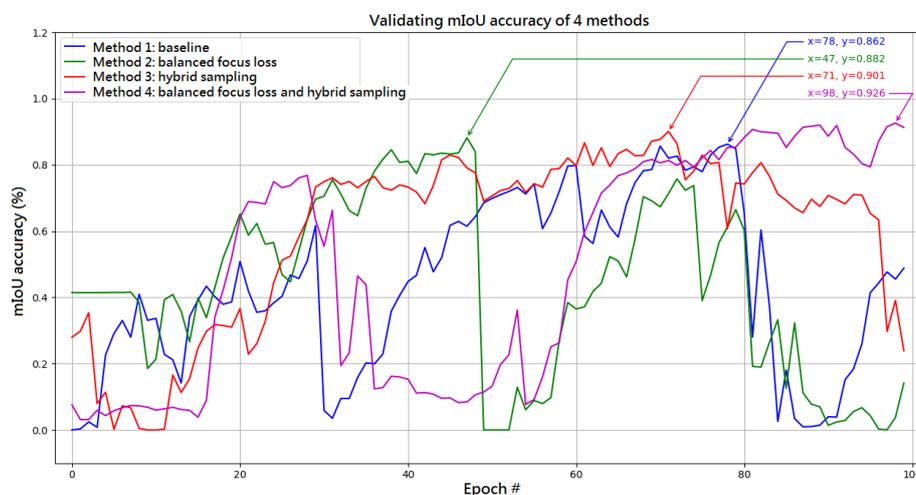


Figure 7. Comparison of mIoU curves obtained using the four methods to process XYZ point cloud data.

Figure 8a–c compare the ground truth of the point cloud with the prediction results obtained using the baseline PointNet method, and the prediction results obtained using the proposed two-stage framework when applied to XYZ point cloud data. As shown in Figure 8b, the pipe (red points) was erroneously segmented as blue points by the baseline method. As shown in Figure 8c, the proposed framework succeeded in segmenting the pipe as red points.

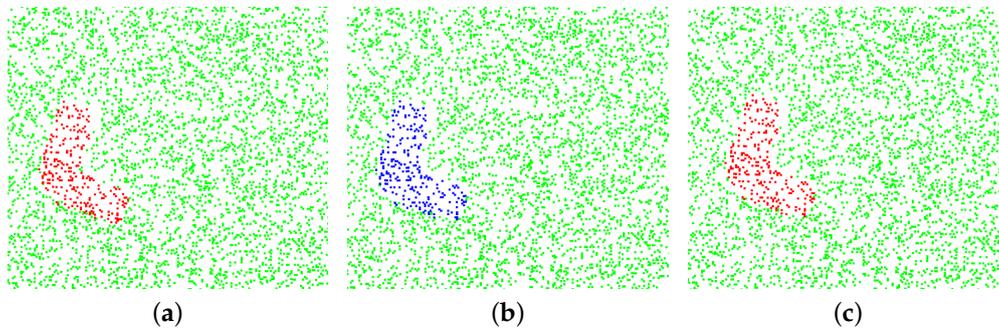


Figure 8. Comparison of ground truth and prediction results obtained when XYZ point cloud data was applied to the baseline or proposed training methods. The prediction results from the baseline training method were clearly erroneous: (a) ground truth; (b) point cloud predicted using conventional training method, showing erroneous prediction where the pipe merges into the wrench; (c) point cloud predicted using the proposed hybrid training method.

Figure 9a–c present another comparison of the ground truth with the baseline training method and the proposed two-stage framework when applied to XYZ point cloud data in the case of multiple objects. As shown in Figure 9b, the baseline model was unable to predict the wrench completely. As shown in Figure 9c, the proposed two-stage framework succeeded in predicting the point clouds of both the pipe and the wrench. Again, the two-stage framework outperformed the baseline training method when applied to XYZ point cloud data.

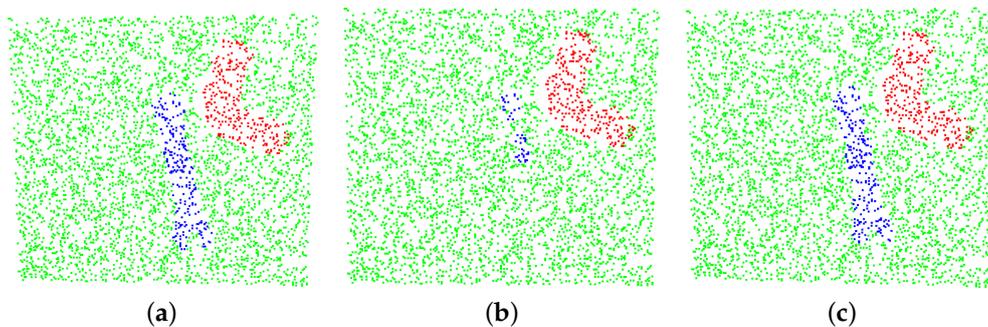


Figure 9. Comparison of ground truth and prediction results obtained when XYZ point cloud data was applied to the baseline or proposed training methods. The results obtained using the baseline training method were clearly less accurate than those obtained using the proposed method: (a) ground truth point cloud; (b) point cloud predicted using conventional training method, showing that most of the points associated with the wrench were not detected; (c) point cloud predicted using the proposed hybrid training method.

3.4. Other Examples

To validate the proposed method, we tested more examples for semantic segmentation. Figure 10 shows the objects tested in the experiment. Figure 11a shows that many of point data of the wrench were misclassified into another class. However, Figure 11b shows that the proposed method segmented most of the point clouds correctly. The reason is that the imbalanced problem makes the baseline

method insufficiently trained, causing inaccurate predictions. This impact makes the segmentation quality low and fail robotic grasps. Figure 11c,d show that the point data of the eraser were also misclassified into another class by the baseline method but not by the proposed method. Figure 11e,f show that many points of the eraser were misclassified into the wrench.

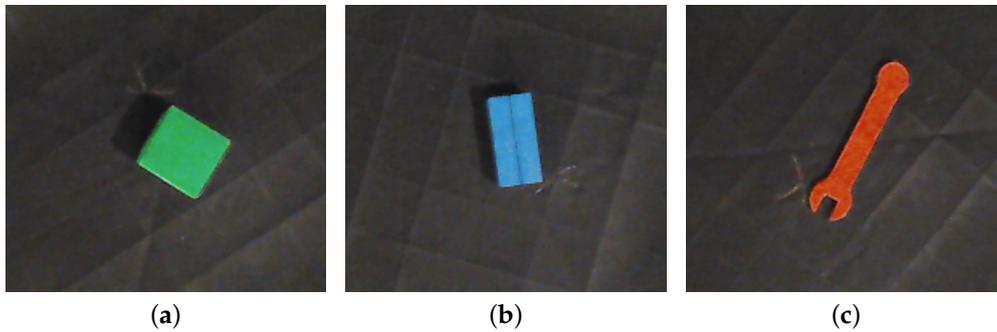


Figure 10. Tested objects: (a) cube; (b) eraser; (c) wrench.

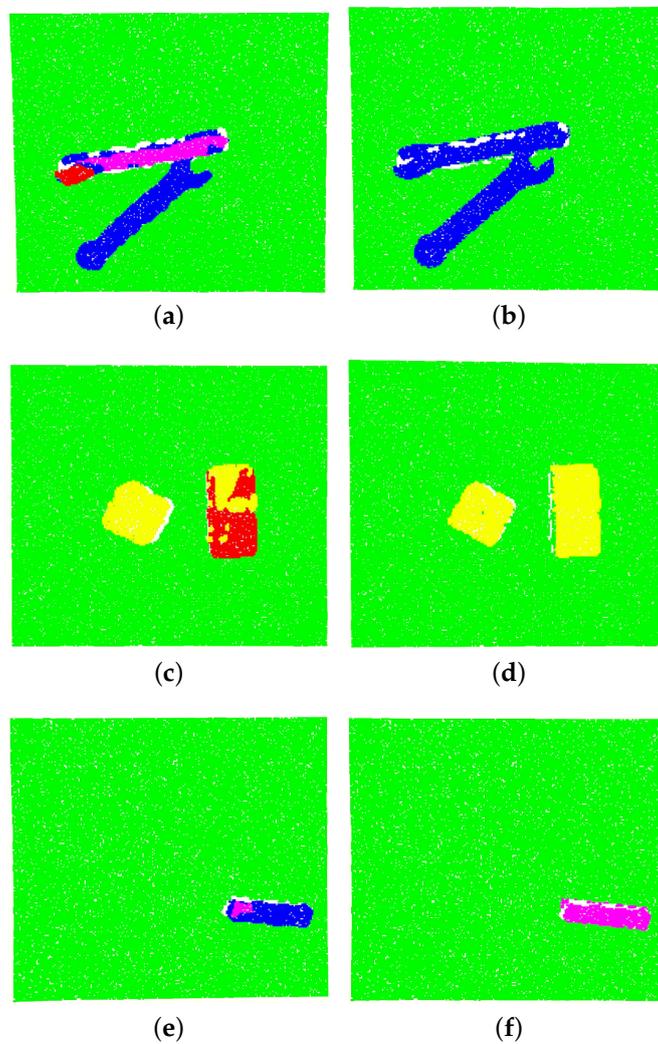


Figure 11. Comparison of the baseline deep network and the proposed method: (a,c,e) are the results of the baseline method; (b,d,f) are the results of the proposed method.

4. Conclusions and Future Research

This paper presents a two-stage framework aimed at resolving the problem of using imbalanced point data for deep network segmentation. The use of hybrid-sampling while balancing the focus of the loss function was shown to boost the effectiveness of object segmentation. Intensive experiments were conducted using five objects under a variety of environmental conditions. The separate use of hybrid-sampling or the balanced focus loss function improved segmentation performance by approximately 2.5%. The two-stage framework improved performance by 6~7%. When applied to validation sets, the mIoU values were as follows: XYZRGB point cloud data (98%) and XYZ point cloud data (93%). In addition, the computational effort of the proposed method was also low. Table 2 shows the average prediction time of the proposed method using the number of test data points from 100 trials. Since the number of points in each image varied between 30,000 and 40,000, the average prediction time based on the first-order linear regression on Table 2 varied between 78.39 and 103.54 ms, which was sufficiently fast in a real situation.

Table 2. Average prediction time.

Size (Points)	Prediction Time (ms)
1024	4.82
2048	7.34
4096	12.8
8192	25.7
16384	44.4
32768	84.8

Even though we demonstrated that the proposed method can reduce the number of false positives associated with object segmentation, there are still a number of issues that may influence the performance.

- Even when there is relative balance between the number of background instances and object instances, there can still be a significant difference between the number of instances associated with large and small objects. This can make it difficult for the network to detect small objects.
- The overfitting problem exists. When there is a high ratio of oversampling and a large number of epochs, the network tends to remember trained object point clouds, which undermines the generalization of the model and hinders segmentation. Thus, experiments must be conducted on that specific dataset to enable the selection of a suitable ratio for sampling.
- Oversampling magnifies noise. Training datasets inevitably contain noise, with the result being that a small proportion of the data points are labeled erroneously. To avoid magnifying the noise, the oversampling ratio should not be set too high. Setting a lower ratio can help to ensure that the network does not learn an excessive number of wrong instances, which would otherwise compromise detection performance.

In future research, the validation of the proposed two-stage scheme will be conducted on other datasets and segmentation deep networks, such as PointNet++, DGCNN (Dynamic Graph CNN), and GCN (Graph Attention Convolution), using point cloud data. In addition, the relationship between the success of robotic grasps and the mIoU accuracy of object segmentation will be studied.

Author Contributions: Conceptualization, H.-I.L.; methodology, H.-I.L. and M.C.N.; software, M.C.N.; validation, M.C.N.; formal analysis, H.-I.L.; investigation, H.-I.L.; resources, H.-I.L.; data curation, M.C.N.; writing—original draft preparation, H.-I.L.; writing—review and editing, H.-I.L.; visualization, H.-I.L.; supervision, H.-I.L.; project administration, H.-I.L.; funding acquisition, H.-I.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Taipei University of Technology grant number NTUT-BIT-105-1.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Somasundaram, A.; Reddy, U.S. Data imbalance: Effects and solutions for classification of large and highly imbalanced data. In Proceedings of the 1st International Conference on Research in Engineering, Computers and Technology (ICRECT 2016), Tiruchirappalli, India, 8–10 September 2016; pp. 28–34.
2. Jayasree, S.; Gavya, A.A. Addressing imbalance problem in the class—A survey. *Int. J. Appl. Innov. Eng. Manag.* **2014**, 239–243.
3. Maheshwari, S.; Jain, R.; Jadon, R. A Review on Class Imbalance Problem: Analysis and Potential Solutions. *Int. J. Comput. Sci. Issues (IJCSI)* **2017**, 14, 43–51.
4. Song, Y.; Morency, L.P.; Davis, R. Distribution-sensitive learning for imbalanced datasets. In Proceedings of the 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Shanghai, China, 22–26 April 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–6.
5. Zieba, M.; Świątek, J. Ensemble SVM for imbalanced data and missing values in postoperative risk management. In Proceedings of the 2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013), Lisbon, Portugal, 9–12 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 95–99.
6. Birla, S.; Kohli, K.; Dutta, A. Machine learning on imbalanced data in credit risk. In Proceedings of the 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 13–15 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
7. Khalilia, M.; Chakraborty, S.; Popescu, M. Predicting disease risks from highly imbalanced data using random forest. *BMC Med. Inf. Decis. Mak.* **2011**, 11, 51. [[CrossRef](#)] [[PubMed](#)]
8. Li, Y.; Sun, G.; Zhu, Y. Data imbalance problem in text classification. In Proceedings of the 2010 Third International Symposium on Information Processing, Qingdao, China, 15–17 October 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 301–305.
9. Zhang, Y.; Li, X.; Gao, L.; Wang, L.; Wen, L. Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. *J. Manuf. Syst.* **2018**, 48, 34–50. [[CrossRef](#)]
10. Zhang, X.; Jiang, D.; Han, T.; Wang, N.; Yang, W.; Yang, Y. Rotating machinery fault diagnosis for imbalanced data based on fast clustering algorithm and support vector machine. *J. Sens.* **2017**, 2017, doi:10.1155/2017/8092691. [[CrossRef](#)]
11. Zhu, Z.B.; Song, Z.H. Fault diagnosis based on imbalance modified kernel Fisher discriminant analysis. *Chem. Eng. Res. Des.* **2010**, 88, 936–951. [[CrossRef](#)]
12. Khreich, W.; Granger, E.; Miri, A.; Sabourin, R. Iterative Boolean combination of classifiers in the ROC space: An application to anomaly detection with HMMs. *Pattern Recognit.* **2010**, 43, 2732–2752. [[CrossRef](#)]
13. Tavallaee, M.; Stakhanova, N.; Ghorbani, A.A. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Tran. Syst. Man Cybern. Part C (Appl. Rev.)* **2010**, 40, 516–524. [[CrossRef](#)]
14. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, 6, 27. [[CrossRef](#)]
15. Hlosta, M.; Stríz, R.; Kupčík, J.; Zendulka, J.; Hruska, T. Constrained classification of large imbalanced data by logistic regression and genetic algorithm. *Int. J. Mach. Learn. Comput.* **2013**, 3, 214. [[CrossRef](#)]
16. Zou, Q.; Xie, S.; Lin, Z.; Wu, M.; Ju, Y. Finding the best classification threshold in imbalanced classification. *Big Data Res.* **2016**, 5, 2–8. [[CrossRef](#)]
17. Krawczyk, B.; Woźniak, M. Cost-sensitive neural network with roc-based moving threshold for imbalanced classification. In Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, Wrocław, Poland, 14–16 October 2015; Springer: Cham, Switzerland, 2015; pp. 45–52.
18. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, 16, 321–357. [[CrossRef](#)]
19. Mathew, J.; Luo, M.; Pang, C.K.; Chan, H.L. Kernel-based SMOTE for SVM classification of imbalanced datasets. In Proceedings of the IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society, Yokohama, Japan, 9–12 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 001127–001132.
20. Wang, S.; Liu, W.; Wu, J.; Cao, L.; Meng, Q.; Kennedy, P.J. Training deep neural networks on imbalanced data sets. In Proceedings of the 2016 International Joint conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 4368–4374.
21. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, 21, 1263–1284.

22. Nashnush, E.; Vadera, S. Cost-sensitive Bayesian network learning using sampling. In *Recent Advances on Soft Computing and Data Mining*; Springer: Cham, Switzerland, 2014; pp. 467–476.
23. Liu, X.Y.; Wu, J.; Zhou, Z.H. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2008**, *39*, 539–550.
24. Chawla, N.V.; Lazarevic, A.; Hall, L.O.; Bowyer, K.W. SMOTEBoost: Improving prediction of the minority class in boosting. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Cavtat-Dubrovnik, Croatia, 22–26 September 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 107–119.
25. Seiffert, C.; Khoshgoftaar, T.M.; Van Hulse, J.; Napolitano, A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2009**, *40*, 185–197. [[CrossRef](#)]
26. Collell, G.; Prelec, D.; Patil, K.R. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing* **2018**, *275*, 330–340. [[CrossRef](#)] [[PubMed](#)]
27. Wozniak, M. *Hybrid Classifiers: Methods of Data, Knowledge, and Classifier Combination*; Springer: Cham, Switzerland, 2013; Volume 519.
28. Woźniak, M.; Graña, M.; Corchado, E. A survey of multiple classifier systems as hybrid systems. *Inf. Fusion* **2014**, *16*, 3–17. [[CrossRef](#)]
29. Wang, S.; Li, Z.; Chao, W.; Cao, Q. Applying adaptive over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, QLD, Australia, 10–15 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–8.
30. Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [[CrossRef](#)]
31. Guo, H.; Viktor, H.L. Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *ACM Sigkdd Explor. Newslett.* **2004**, *6*, 30–39. [[CrossRef](#)]
32. Mease, D.; Wyner, A.J.; Buja, A. Boosted classification trees and class probability/quantile estimation. *J. Mach. Learn. Res.* **2007**, *8*, 409–439.
33. Sun, Y.; Kamel, M.S.; Wong, A.K.; Wang, Y. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit.* **2007**, *40*, 3358–3378. [[CrossRef](#)]
34. Wang, S.; Yao, X. Diversity analysis on imbalanced data sets by using ensemble models. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining, Nashville, TN, USA, 30 March–2 April 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 324–331.
35. Barandela, R.; Valdivinos, R.M.; Sánchez, J.S. New applications of ensembles of classifiers. *Pattern Anal. Appl.* **2003**, *6*, 245–256. [[CrossRef](#)]
36. Błaszczyński, J.; Deckert, M.; Stefanowski, J.; Wilk, S. Integrating selective pre-processing of imbalanced data with ivotes ensemble. In Proceedings of the International Conference on Rough Sets and Current Trends in Computing, Warsaw, Poland, 28–30 June 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 148–157.
37. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2011**, *42*, 463–484. [[CrossRef](#)]
38. Liu, Y.H.; Chen, Y.T. Total margin based adaptive fuzzy support vector machines for multiview face recognition. In Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI, USA, 12 October 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 2, pp. 1704–1711.
39. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
40. Maloof, M.A. Learning when data sets are imbalanced and when costs are unequal and unknown. In Proceedings of the ICML-2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC, USA, 21–24 August 2003; Volume 2, pp. 1–8.
41. Kaur, P.; Gosain, A. Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. In *ICT Based Innovations*; Springer: Singapore, 2018; pp. 23–30.
42. Drummond, C.; Holte, R.C.; others. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In Proceedings of the ICML-2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC, USA, 21–24 August 2003; Volume 11, pp. 1–8.

43. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **2018**, *106*, 249–259. [[CrossRef](#)] [[PubMed](#)]
44. Dittman, D.J.; Khoshgoftaar, T.M.; Wald, R.; Napolitano, A. Comparison of data sampling approaches for imbalanced bioinformatics data. In Proceedings of the The Twenty-Seventh International FLAIRS Conference, Pensacola Beach, FL, USA, 21–23 May 2014.
45. Japkowicz, N. Learning from imbalanced data sets: A comparison of various strategies. In Proceedings of the AAAI Workshop on Learning from Imbalanced Data Sets, Austin, TX, USA, 31 July 2000; Volume 68, pp. 10–15.
46. Shelke, M.S.; Deshmukh, P.R.; Shandilya, V.K. A review on imbalanced data handling using undersampling and oversampling technique. *IJRTER* **2017**, *3*, 444–449.
47. Khoshgoftaar, T.; Seiffert, C.; Van Hulse, J. Hybrid Sampling for Imbalanced Data. In Proceedings of the 2008 IEEE International Conference on Information Reuse and Integration, Las Vegas, NV, USA, 13–15 July 2008; Volume 8, pp. 202–207.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).