IEICE TRANS. INF. & SYST., VOL.E107–D, NO.1 JANUARY 2024

LETTER

# Negative Learning to Prevent Undesirable Misclassification

**Kazuki EGASHIRA**[†a)], ***Member***, **Atsuyuki MIYAI**[†b)], **Qing YU**[†c)], ***Nonmembers***, **Go IRIE**[††d)], ***Member***, ***and*** **Kiyoharu AIZAWA**[†e)], ***Fellow***

**SUMMARY** We propose a novel classification problem setting where Undesirable Classes (UCs) are defined for each class. UC is the class you specifically want to avoid misclassifying. To address this setting, we propose a framework to reduce the probabilities for UCs while increasing the probability for a correct class.

***key words:*** *classification, negative learning*

## 1. Introduction

When evaluating the performance of Deep Neural Networks (DNNs), we usually focus on accuracy, or the number of correct answers obtained from DNNs. In doing so, misclassifications are treated equally, and it is often ignored which class the incorrect answers are. On the other hand, as some incidents in the past, such as the misclassification of humans as gorillas [1], have shown, DNNs sometimes produce undesirable results. The simplest way to prevent such misclassifications is to remove potentially problematic classes (in the case above, *gorilla*) from the target class set, as was done in the incident. However, this approach makes it impossible to recognize the objects that belong to the removed class even when the class itself is meaningful. What really matters is the association between the correct class and the misclassified class. Some previous studies have investigated the magnitude and trends of inappropriate misclassification [2], [3], but none have proposed solutions to mitigate the problem.

Therefore, we propose a novel problem setting where Undesirable Classes (UCs) are provided for each class. The concept of the setting is shown in Fig. 1. Typically, all the misclassifications are treated equally. In contrast, we assume that for each correct class, there are some misclassifications that are particularly inappropriate, which we define as UCs. Here, the best classification for each image is the correct class, followed by the normal misclassification classes, and the worst are the UCs.

To address this problem setting, we focus on Negative Learning (NL) [4]. NL is one of the training methods using a Complementary Label (CL), a label indicating that "the input image does not belong to this class." NL is proposed as a robust training method against label noise. In fact, NL can reduce the risk of providing the wrong information to a DNN because of the low risk of selecting the correct class as the CL. The idea in NL is to decrease the probability of the CL, as opposed to the typical loss to increase the probability of the true label.

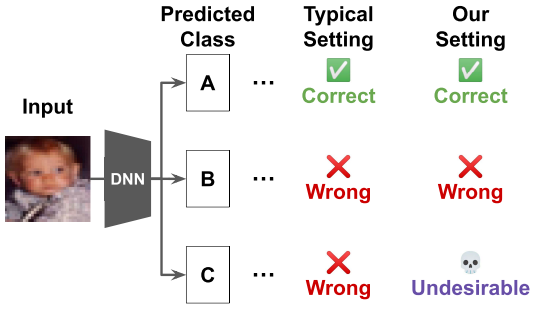


**Fig. 1** Concept of our problem setting.
We assume that there are some misclassifications that are particularly inappropriate (C in the figure), namely, Undesirable Classes (UCs).

In light of this concept, we propose to apply NL loss for our problem setting, specifying UCs as CLs. By combining NL loss with the typical cross entropy loss, we propose a framework to decrease the probability of the UCs as well as increase the probability of the correct class. We evaluate our method on various datasets and settings with different ways of selecting UCs and show that it successfully mitigates the UC error while maintaining comparable accuracy to a method that focuses only on accuracy.

Briefly, our contributions are as follows:

1. We propose a new classification problem setting where each class has Undesirable Classes (UCs), classes that we particularly want to avoid.
2. We propose a training framework to improve accuracy while avoiding the UC error as much as possible by using the modified version of NL loss [4].
3. We evaluate our method on various datasets and settings with different ways of selecting UCs, and experimentally demonstrated that it mitigates the UC error while maintaining comparable accuracy to a method that focuses only on accuracy.

Manuscript received August 19, 2023.
Manuscript publicized October 5, 2023.
[†]The authors are with The University of Tokyo, Tokyo, 113–8654 Japan.
[††]The author is with Tokyo University of Science, Tokyo, 125–8585 Japan.
a) E-mail: egashira@hal.t.u-tokyo.ac.jp
b) E-mail: miyai@hal.t.u-tokyo.ac.jp
c) E-mail: yu@hal.t.u-tokyo.ac.jp
d) E-mail: goirie@ieee.org
e) E-mail: aizawa@hal.t.u-tokyo.ac.jp
DOI: 10.1587/transinf.2023EDL8056

## 2. Related Work

### 2.1 Undesirable Class

Basically, existing methods to prevent inappropriate outputs involve excluding classes that could cause inappropriate misclassifications. An example of this is Google's response to the problem of mislabeling humans as *gorilla* [1]. By removing *gorilla* from the label set, their system became unable to recognize a gorilla, even though the label itself was meaningful. Another example is a study by Yang et al. [5] to filter out unsafe labels from ImageNet [6]. They measure whether the labels themselves are inherently offensive. In this study, *gorilla* is not considered unsafe because it is a legitimate label of the corresponding animal. However, they are clearly harmful when predicted for people from certain groups.

As highlighted, simply removing certain classes is not the best solution. We believe that inappropriateness should be considered in association with the correct class to which the object belongs. This is where the concept of UC comes in. There are some studies addressing the association issues [2], [3]. However, our problem setting with UCs differs from these studies in that (i) we propose solutions to mitigate the UC error while previous works mainly focus on investigating it, and (ii) UCs can be user-defined, whereas previous works pre-define them.

### 2.2 Complementary Label

Ishida et al. [7] proposed a classification problem setting where, instead of an ordinary class label, a Complementary Label (CL) is available. CL is a label that specifies a class to which an object does not belong. By making appropriate assumptions, they proved that models can be well trained from CLs without true labels.

Later, Kim et al. [4] proposed Negative Learning (NL) as a method for training DNNs with noisy data. They randomly select a class other than the one provided as the true (but noisy) label and treat it as a CL. Since the probability of selecting the correct class as the CL is low, NL reduces the risk of providing incorrect information, contributing to better performance. The NL loss proposed in this study is widely applied in later studies [8]–[10]. However, these studies all differ from ours in that they aim to improve accuracy whereas we aim to mitigate the UC error by using NL.

## 3. Method

### 3.1 Problem Setting

We assume that the true label for an image $x$ is $y^{\mathrm{gt}}$, which belongs to one of the $K$ classes denoted as $\{c_1, \ldots, c_K\}$. The output probabilities for each class are denoted as $p_k (k = 1, \ldots, K)$. We also denote the prediction of the model as $y^{\mathrm{pred}} = \arg\max_k p_k$. What is unique in our setting is that each class $c_i$ has a set of UCs $Z^{c_i} \subset \{c_1, \ldots c_K\} \setminus \{c_i\}$. By this definition, the UCs for $y^{\mathrm{gt}}$ are denoted by $Z^{y^{\mathrm{gt}}}$.

Note that $c_j \in Z^{c_i}$ does not necessarily mean $c_i \in Z^{c_j}$. This is convenient for practical use since misclassifying *human* as *gorilla* can cause ethical problems, while the opposite is usually not considered undesirable.

We say that the output is correct if $y^{\mathrm{pred}} = y^{\mathrm{gt}}$, and is a UC error if $y^{\mathrm{pred}} \in Z^{y^{\mathrm{gt}}}$. Our goal is to learn a model that yields the maximum number of correct outputs, and the minimum number of UC errors.

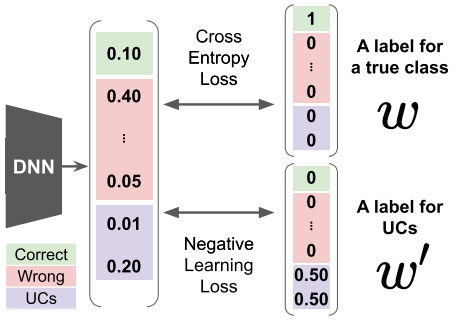


**Fig. 2** Proposed method.
In addition to the typical cross entropy loss to increase the probability of the correct class, we apply a modified version of NL Loss [4] to decrease the probability of UCs.

### 3.2 Proposed Approach

Figure 2 illustrates the overall concept of the proposed method. We propose to train a model with the weighted sum of two loss functions: one increasing the probability of the correct class, and the other decreasing the probabilities of the UCs.

Typically, DNNs for classification tasks are trained with the following Cross Entropy (CE) loss:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{k=1}^{K} w_k \log p_k = -\log p_{y^{\mathrm{gt}}}, \tag{1}$$

where $w_k$ denotes the k-th dimension of the label $\boldsymbol{w}$ representing the correct class, which is defined as follows:

$$w_k = \begin{cases} 1 & (k = y^{\mathrm{gt}}) \\ 0 & (\text{otherwise}). \end{cases} \tag{2}$$

Equation (1) is suitable to optimize the probability value corresponding to the true label as 1. Although the optimal solution for the probabilities corresponding to the UCs is already 0 in Eq. (1), it is often difficult to converge to the perfectly optimal solution. In such a case, DNNs are likely to converge to the next best solution, potentially increasing the probabilities of UCs. Therefore, we use NL [4] as a regularization term in order to prevent the model from becoming such and lead to other misclassifications that are relatively acceptable. The term is defined as follows:

$$\mathcal{L}_{\mathrm{NL}} = -\sum_{k=1}^{K} w'_k \log(1 - p_k), \tag{3}$$

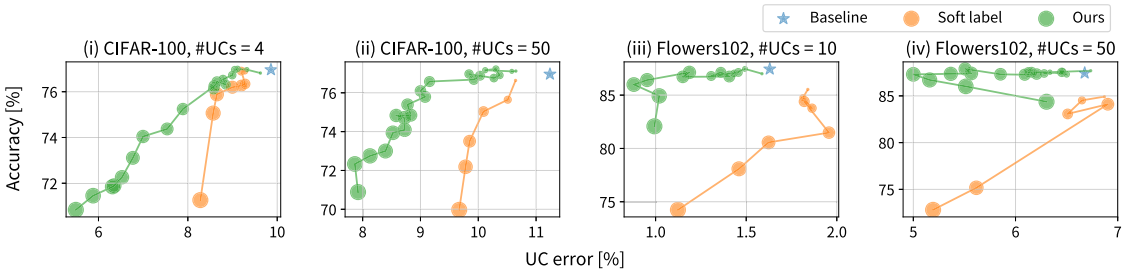


**Fig. 3** Overall results on test sets.
The darker the color, the larger the parameter $\lambda$ for ours and $\epsilon$ for the soft label, respectively. Each point is the average result of three experiments with the same hyperparameter. The upper left of each graph is the ideal state of high accuracy and low UC error. In all of the settings, our method can reduce the UC error with less accuracy loss than the soft label method.

where $w'_k$ denotes the k-th dimension of the label $\boldsymbol{w'}$ representing UCs for $y^{\mathrm{gt}}$, which is defined as follows:

$$w'_k = \begin{cases} \frac{1}{|Z^{y^{\mathrm{gt}}}|} & (k \in Z^{y^{\mathrm{gt}}}) \\ 0 & (\text{otherwise}), \end{cases} \tag{4}$$

where $|Z^{y^{\mathrm{gt}}}|$ denotes the number of the UCs for $y^{\mathrm{gt}}$.

Equation (3) allows optimizing the probability value $p_k$ to zero where $w'_k > 0$, which corresponds to the UCs. The form of the equation is the same as in [4]. However, we use a soft label $\boldsymbol{w'}$ representing UCs in order to reduce the probabilities of UCs. In contrast, [4] uses a hard label with a class that takes 1 being selected at random for every iteration, in order to reduce the probability of all wrong classes equally.

Taken together, the overall loss function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{NL}}, \tag{5}$$

where $\lambda \geq 0$ denotes a scaling parameter.

## 4. Experiment

### 4.1 Experimental Setup

**Datasets.** We experimented with two datasets, with two different ways of selecting UCs each, for a total of four settings. The first dataset is CIFAR-100 [11], which is a 100-class dataset with 50,000 images for training and 10,000 images for testing. In this dataset, the classes are grouped into 20 superclasses, each containing five fine classes. As UCs, we select (i) four classes that belong to the same superclass as the correct class, and (ii) 50 random classes. The other is Flowers102 [12], which is a 102-class fine-grained flower dataset with 1,020 images for training and 6,149 for testing. As UCs, we select (iii) 10 random classes, and (iv) 50 random classes.

Although the main purpose of this study is to prevent undesirable misclassification, it is ethically difficult to create a dataset that includes discriminatory expressions. Instead, we have created several settings to demonstrate whether it is possible to avoid classes that look similar by using CIFAR100 with the same superclass, i.e., (i), and fine-grained Flowers102, i.e., (iii), (iv).

**Models.** We implement ResNet-18 [13] pretrained with ImageNet-1K [6]. The network is trained using Stochastic Gradient Descent (SGD) [14] with a learning rate of 0.001 and a momentum of 0.9. At the end of each epoch, the validation loss is calculated, and the training is considered to have converged when there are no improvements for 10 consecutive epochs. The loss function is Eq. (5), and we experimented with $\lambda$ from 1 to 10 with an interval of 1, from 10 to 100 with an interval of 10, and from 100 to 1,000 with an interval of 100.

### 4.2 Comparison Method

We compare our method with two methods. Firstly, as a baseline, we compare our method with a model trained with the typical CE loss, which corresponds to $\lambda = 0$ in Eq. (5).

Secondly, we compare our method with a simple method using a soft label. We trained DNNs with CE loss, using the soft label $\boldsymbol{s}$ defined as follows:

$$s_k = \begin{cases} 1-\epsilon & (k = y^{\mathrm{gt}}) \\ 0 & (k \in Z^{y^{\mathrm{gt}}}). \\ \frac{\epsilon}{K-|Z^{y^{\mathrm{gt}}}|-1} & (\text{otherwise}) \end{cases} \tag{6}$$

$K - |Z^{y^{\mathrm{gt}}}| - 1$ corresponds to the number of classes over which $\epsilon$ is distributed, satisfying $\sum_{k=1}^{K} s_k = 1$.

When using CE loss, the optimal solution is when the output probability distribution is the same as the label. Therefore, by using $\boldsymbol{s}$, we expect the model to learn that UCs should have lower probabilities than other misclassified classes.

We experiment with various $\epsilon$ in a range that satisfies the following constraint that the label value of the correct class must be the largest:

$$1 - \epsilon > \frac{\epsilon}{K - |Z^{y^{\mathrm{gt}}}| - 1}. \tag{7}$$

Note that there are no existing methods that aim at exactly the same goal because our problem setting is new. We select this method for comparison because soft labels are widely used for various purposes to train the model with information beyond correct or wrong [15]–[17].

**Table 1** Comparison of the UC error at the same accuracy level. Since it is not possible to perfectly match the accuracy of each method, we report the average of the results including up to ±0.3 of the specified accuracy.

| Dataset | CIFAR-100 | | | | Flowers102 | |
|---|---|---|---|---|---|---|
| #UCs | 4 | | 50 | | 10 | 50 |
| Acc. | 76.30 | 71.30 | 75.00 | 72.10 | 85.00 | 84.30 |
| Soft label | 9.16 | 8.28 | 10.10 | 9.78 | 1.81 | 6.78 |
| Ours | **8.69** | **5.88** | **8.71** | **7.87** | **1.02** | **6.30** |

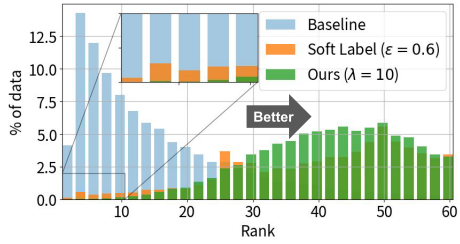


**Fig. 4** Ranks of the UCs (CIFAR-100, #UCs=4). We select $\lambda = 10$ for ours and $\epsilon = 0.6$ for the soft label because they exhibited similar performance in terms of top 1 accuracy and UC error. Our method successfully lowers the ranks of the UCs.

### 4.3 Result

Figure 3 demonstrates the overall result. Compared to the baseline where $\lambda = 0$, our method successfully mitigates the UC error as $\lambda$ increases. Although there is a tradeoff between UC error and accuracy, this tradeoff is superior to the soft label method.

In Table 1, we compare the representative points between each method at the same accuracy level. In all four settings we experimented with, the mitigation of the UC error is more significant in our method.

Figure 4 demonstrates the ranks of the UCs in the setting of CIFAR-100 with four UCs per class. For each of the 10k test data, we computed the mean rank of the probabilities of the four UCs. With the baseline method, many UCs appear in high ranks. The soft label method can mitigate this, but some UCs still remain. In contrast, our method successfully lowers the ranks.

## 5. Limitation

When there are many UCs, it becomes difficult to reduce the UC error. For instance, as indicated in Fig. 3, when 50 UCs are defined for CIFAR-100, the reduction of the UC error is less. Also, when 50 UCs are defined for Flowers102, the UC error starts to increase when NL is increased excessively. Although omitted from the Figure, these trends were consistently confirmed when there were even more UCs.

We do not consider this to be a major problem, since our motivation in this paper is to avoid particularly inappropriate classes, which are usually not very numerous. However, in order to make the method more general and trustworthy, further analysis is needed to explore robust methods for UC definition and to quantitatively analyze the relationship between UC definition and the effect of the method.

## 6. Conclusion

In this study, we presented a novel classification problem setting where Undesirable Classes (UCs) are provided for each class. To address this problem, we focus on Negative Learning (NL), a method to train a model with a Complementary Label (CL) that indicates a class to which a pattern does not belong. The concept of NL is to decrease the probability corresponding to the CL. In light of this, we propose a framework to decrease the probabilities for UCs by specifying UCs as CLs. Combined with the typical cross entropy loss, our method successfully mitigates the risk of the UC error with little loss of accuracy.

### References

[1] T. Simonite, "When It Comes to Gorillas, Google Photos Remains Blind." https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/, (Retrieved: 2023-02-05).
[2] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., "Learning transferable visual models from natural language supervision," ICML, 2021.
[3] P. Goyal, A.R. Soriano, C. Hazirbas, L. Sagun, and N. Usunier, "Fairness indicators for systematic assessments of visual feature extractors," ACM FAccT, pp.70–88, 2022.
[4] Y. Kim, J. Yim, J. Yun, and J. Kim, "Nlnl: Negative learning for noisy labels," ICCV, pp.101–110, 2019.
[5] K. Yang, K. Qinami, L. Fei-Fei, J. Deng, and O. Russakovsky, "Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy," ACM FaccT, pp.547–558, 2020.
[6] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," CVPR, pp.248–255, 2009.
[7] T. Ishida, G. Niu, W. Hu, and M. Sugiyama, "Learning from complementary labels," NeurIPS, 2017.
[8] H. Tokunaga, B.K. Iwana, Y. Teramoto, A. Yoshizawa, and R. Bise, "Negative pseudo labeling using class proportion for semantic segmentation in pathology," ECCV, vol.12360, pp.430–446, 2020.
[9] M.N. Rizve, K. Duarte, Y.S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," ICLR, 2021.
[10] Y. Kim, J. Yun, H. Shon, and J. Kim, "Joint negative and positive learning for noisy labels," CVPR, pp.9437–9446, 2021.
[11] A. Krizhevsky, G. Hinton, et al., "Learning multiple layers of features from tiny images," Technical Report., 2009.
[12] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," ICVGIP, pp.722–729, 2008.
[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CVPR, pp.770–778, 2016.
[14] H. Robbins and S. Monro, "A stochastic approximation method," The annals of mathematical statistics, vol.22, no.3, pp.400–407, 1951.
[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," CVPR, pp.2818–2826, 2016.
[16] G. Hinton, O. Vinyals, J. Dean, et al., "Distilling the knowledge in a neural network," NIPS Workshop, 2015.
[17] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, and X. Geng, "Deep label distribution learning with label ambiguity," IEEE Trans. Image Process., vol.26, no.6, pp.2825–2838, 2017.