



An influence study of task-irrelevant factors on the neural network accuracy

Yi Liu¹ · Changsheng Zhang^{1,3} · Haitong Zhao² · Chen Zhang¹ · Wenjing Wu²

Received: 1 December 2024 / Revised: 17 February 2025 / Accepted: 25 March 2025 / Published online: 12 May 2025
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2025

Abstract

In the field of data-driven deep learning, the dataset quality is one of the essential factors that determine the neural network (NN) accuracy. During the sampling process, the sample may contain many task-irrelevant factors. However, there is a lack of study on the influence of task-irrelevant factors in the dataset on the NN accuracy. This paper analyzes the influence of task-irrelevant factors through mathematical derivation, and finds that regularization cannot solve the problem of task-irrelevant factors affecting the NN inference performance. To reduce the influence of task-irrelevant factors, this paper proposes an idea, making the mathematical expectation of task-irrelevant features in the training set to be zero. Furthermore, this paper proposes a method that can reduce the influence of the background factor on the image classification model. To verify the effectiveness of this method, a series of controlled experiments are conducted on the image classification and object detection task. Experimental results show that this method not only effectively improves the accuracy of the classification model, but also improves the precision of the object detection model. The code related with this paper can be download from <https://github.com/CaedmonLY/BackgroundFactor/tree/master>.

Keywords Neural networks · Task-irrelevant factors · Regularization · Background factor · Image classification · Object detection

1 Introduction

The dataset quality has an important influence on the neural network (NN) inference performance in computer vision [1]. Various tuning techniques [2–5] and NN topologies [6–8] enable the NN to learn the potential knowledge in the dataset.

For the computer vision task, one factor involves multiple pixels in an image. The relevant factor involves pixels relevant to the classification object, e.g., the color of the classification object is relevant to the classification task, and is a relevant factor. Instead, the irrelevant factor is the pixel area irrelevant to the classification object, e.g., the background of the image is irrelevant to the classification task, and is an irrelevant factor.

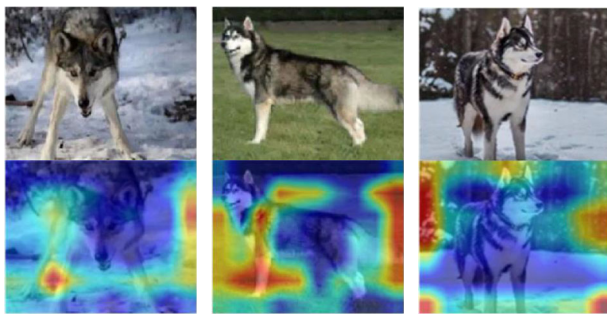
On the computer vision task, the NN classifies an image into different classes according to the pixel region in the image. In this process, irrelevant factors in the image may interfere with the judgment of the NN. In other words, features of the classification object learned by the NN contain irrelevant features. As shown in Figs. 1a and b, the NN can correctly distinguish the Arctic Wolf and Husky. However, if the Husky is placed in the snow, as shown in Fig. 1c, the NN determines that the Husky in the image is an Arctic Wolf [9], and visualization with GradCAM [10] shows that the NN's focus is on the image background. In the original dataset, images of Arctic Wolves are often sampled in the snow, while Huskies are in common living areas. Images of the Arctic Wolf and Husky in the dataset are quite different in the background. Therefore, the image background interferes with the classification result of the NN. Also in Fig. 2, the NN determines whether the object in the image is a horse based on whether the lower-left corner of the image contains a source tag. After removing the tag, the horse in the image is classified into other classes. Furthermore, when the tag is moved to an image containing a car, the car is classified as "horse" [11].

✉ Changsheng Zhang
zhangchangsheng@mail.neu.edu.cn

¹ Department of Software Engineering, Northeastern University, Shenyang 110167, China

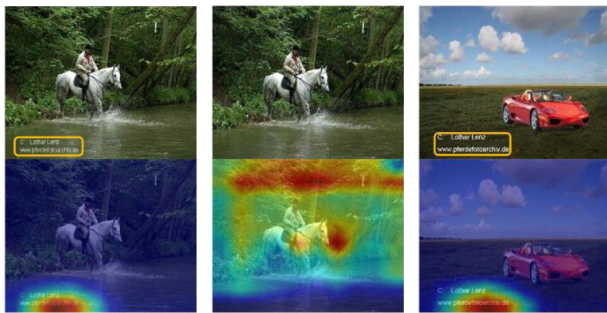
² Department of Computer Application Technology, Northeastern University, Shenyang 110167, China

³ College of Computer Science and Engineering, Ningxia Institute of Science and Technology, Shizuishan 753000, China



(a) The image of the Arctic Wolf in the background of snow is correctly classified. (b) The image of the Husky in the background of grass is correctly classified. (c) The image of the Husky in the background of snow is correctly classified.

Fig. 1 The classification result is disturbed by the background factor. The upper image is the test sample, while the lower image shows the focus of the NN on the test sample with GradCAM



(a) The horse with a source tag is classified as "horse". (b) The horse without a source tag is classified as "horse". (c) The car with a source tag is classified as "other classes".

Fig. 2 The classification result is disturbed by the source tag. The upper image is the test sample, while the lower image shows the focus of the NN on the test sample with GradCAM

Generally, this kind of misclassification caused by irrelevant factors, is considered to be due to insufficient feature extraction capabilities of the NN or insufficient training samples. To improve the feature extraction capability of the NN, increasing the NN depth is widely adopted. As a result, the NN has developed from AlexNet [12] with a depth of 8 layers to ResNet [13] with a depth of 152 layers. The depth increase improves the NN accuracy, and the problem of misclassification is alleviated to a certain extent. However, compared with the increase in model complexity and training cost, the improvement of the accuracy is very little. One solution for insufficient training samples is data augmentation [14–16]. Data augmentation realizes the expansion of the dataset and enriches training samples. However, it does not substantially improve the dataset quality. Whether increasing the NN depth or data augmentation, both of them overlook the problem of irrelevant factors.

The cases in Figs. 1 and 2 might be due to the sampling bias in the original dataset. In Fig. 1, the image background is

a task-irrelevant factor on the image classification task. When the background changes, the classification result of the NN also changes. In this case, this paper argues that irrelevant features are biased, that is, there is some consistency among irrelevant factors in the dataset. Irrelevant factors (e.g., background) restrict the inference performance and affect the accuracy of the NN. To reduce the influence of irrelevant factors, regularization methods can be used. Regularization avoids overfitting of the NN by restricting the size of the weight in the NN. However, regularization restricts not only the weight of irrelevant features, but also the weight of relevant features, resulting in a decrease in the sensitivity of the NN to relevant features. Therefore, this paper aims to find a method to restrict the weight of irrelevant features without affecting the weight of relevant features.

Through mathematical derivation, this paper analyzes the influence of irrelevant factors on the NN during training. Based on the analysis, an idea is proposed, that is, to make the mathematical expectation of irrelevant features in the dataset to be zero. According to this idea, this paper proposes a method that can reduce the influence of the background factor. Different from increasing the NN depth and traditional data augmentation, this method is to process the training set, and it substantially improves the dataset quality. Experimental results have shown that this method improves both accuracy and generalization ability of the NN.

Contributions of this paper are as follows:

- This paper analyzes the influence of task-irrelevant factors on the NN during model training through mathematical derivation.
- This paper proposes an idea to reduce the influence of task-irrelevant factors on the NN. Based on this idea, this paper proposes a method to reduce the influence of the background factor. The proposed method selectively processes the background feature, and improves the NN accuracy.
- The method proposed in this paper can also be used as a new data augmentation method. Different from traditional data augmentation, it substantially improves the dataset quality and generalization ability of the NN.

The rest of this paper is organized as follows: Sect. 2 reviews commonly used methods to improve the NN accuracy in the field of computer vision. Section 3 analyzes the influence of task-irrelevant factors on the NN inference performance. Section 4 introduces the proposed method. The experimental setup and evaluation results are presented in Sect. 5. Finally, Sect. 6 concludes this paper.

2 Related work

To improve the accuracy of the image classification model, most of the current study is still focused on the architecture of the NN. Going deeper to solve more complex tasks and to improve classification or recognition accuracy has become a trend. ResNet is a representative network and has shown good performance in image classification. At present, ResNet-152 is the state-of-the-art ResNet model, which has 152 layers. However, increasing the depth of the network increases the risk of overfitting and limits the generalization ability of the model, especially when there are few training samples.

Due to overfitting caused by insufficient samples and large scale of parameters, regularization methods have become a significant strategy in deep learning to improve the generalization ability of the model, such as L2 regularization and dropout [17]. L2 regularization is based on the assumption that models with smaller weights are simpler than those with larger weights. L2 regularization restricts weights by adding the sum of weights to the loss function. L2 regularization drives weights in the model smaller, which reduces the sensitivity of the model to input features. Dropout is proposed as a form of regularization for fully connected NN layers. Each layer's output is kept with probability p , otherwise being set to 0 with probability $(1 - p)$. Dropout simplifies the network structure by randomly removing neurons in the NN, which improves the generalization ability of the NN.

In computer vision, data augmentation is widely used to improve the generalization ability of models. Data augmentation can be divided into two parts: data augmentation based on a single sample and multiple samples. Data augmentation based on a single sample includes geometric transformation, color space transformation, flipping, cropping, etc [18]. Data augmentation based on multiple samples is to perform basic data transformation on multiple samples and then generate new samples through artificial synthesis. A typical method of data augmentation based on multiple samples is SMOTE [19], which uses interpolation to generate new samples. SamplePairing [20] randomly selects two samples from the training set, which can be of the same class or different classes. After data augmentation based on a single sample, the average value of two samples is used to generate a new sample. Mixup [21] is similar to SamplePairing. Instead of using average value, it uses a linear interpolation method to generate new samples based on the strategy of Vicinal Risk Minimization. From the perspective of the main idea, these three data augmentation methods all fit the real sample distribution by making the discrete sample set continuous, and obtain new samples from the continuous sample space.

Various regularization and data augmentation methods have improved the accuracy of the image classification model to a certain extent. However, these methods do not deal with task-irrelevant factors in the dataset. Phankokkrud [22] pro-

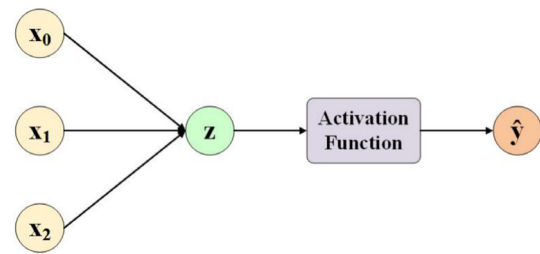


Fig. 3 Neural network forward propagation process

poses a convolutional neural network (CNN) model to deal with the problem of irrelevant factors in the image. By regulating and configuring parameters in the CNN, the accuracy of the model is improved. To alleviating the effect of task-irrelevant factors, Kim et al. [23] propose a novel method called Doubly Adversarial Suppression Network. This Network is trained to neglect irrelevant factors and to focus more on task-relevant factors. Furthermore, to evaluate the contribution of different features to the NN, Belloni et al. [24] analyze the change in classification scores following the occlusion of the target, shadow, clutter areas and their combinations. Han et al. [25] perform correlation analysis and redundancy analysis on image features through a fast filtering feature selection method called Fast Correlation-Based Filter (FCBF) [26].

In this paper, the dataset is preprocessed to reduce the influence of task-irrelevant factors on the NN accuracy. This paper provides a new idea to improve the generalization ability of the NN.

3 Analysis of the task-irrelevant factors' influence

When pixels/features of an image are input to the NN, a feature vector can be obtained, denoted as $x = (x_0, x_1, x_2, \dots)$, where each $x_i (i = 0, 1, 2, \dots)$ is a feature. As a result, one factor may be related to multiple x_i , while each x_i may also be related to multiple factors, including both task-relevant factors and task-irrelevant factors. For simplicity, this paper assumes that each x_i is only related to one factor, e.g., x_0 is an irrelevant feature, while x_1, x_2 is relevant features. When the irrelevant feature in the dataset is biased, after the model is trained, the weight of the irrelevant feature may converge to a certain non-zero value. As a result, the NN regards the irrelevant feature as the relevant feature, which affects the model accuracy.

Firstly, take the simplest NN as an example to analyze the influence of irrelevant features when irrelevant features are biased, as shown in Fig. 3. Let x_0, x_1 , and x_2 be features, where x_0 is an irrelevant feature, x_1 and x_2 are relevant features. And let w_0, w_1 , and w_2 be corresponding weights,

respectively. The forward propagation process of the NN can be defined as Eq. (1).

$$\hat{y} = G(z) = G(x_0 w_0 + x_1 w_1 + x_2 w_2 + b) \quad (1)$$

where $\hat{y}$ is the predicted value, $G(\cdot)$ is the activation function, b is a bias, and $z = x_0 w_0 + x_1 w_1 + x_2 w_2 + b$. This paper uses L2 loss as the loss function, that is, $loss = (\hat{y} - y)^2$, where y is the ground truth. To explore the influence of irrelevant features on the NN, in backpropagation, this paper only focuses on the weight update of the irrelevant feature x_0 , that is, Eq. (2).

$$w'_0 = w_0 - \alpha \frac{d_{loss}}{d_{w_0}} = w_0 - 2\alpha(\hat{y} - y) \frac{d_{G(z)}}{d_z} x_0 \quad (2)$$

where w'_0 is the updated w_0 , $\frac{d_{loss}}{d_{w_0}} = 2(\hat{y} - y) \frac{d_{G(z)}}{d_{w_0}}$, $\frac{d_{G(z)}}{d_z} \frac{d_z}{d_{w_0}} = \frac{d_{G(z)}}{d_z} x_0$, and α is the learning rate.

To simplify as much as possible, this paper uses the ReLU activation function, that is, $G(\cdot) = \max(0, \cdot)$. If $z \leq 0$, $\frac{d_{G(z)}}{d_z} = 0$ and w_0 will not be updated. Otherwise, $\frac{d_{G(z)}}{d_z} = 1$. Therefore, only the case of $z > 0$ needs to be considered. Then, using the same notation, the weight update of x_0 can be expressed as Eq. (3).

$$w'_0 = w_0 - 2\alpha(\hat{y} - y)x_0 \quad (3)$$

Equation (3) is the weight update process of a single sample. For multiple samples, this paper considers the mathematical expectation of $\frac{d_{loss}}{d_{w_0}}$, and the weight update after random sampling can be expressed as Eq. (4)

$$\begin{aligned} w'_0 &= w_0 - E[2\alpha(\hat{y} - y)x_0] \\ &= w_0 - 2\alpha(E[x_0^2]w_0 + (\beta - y)E[x_0]) \end{aligned} \quad (4)$$

where $E[\cdot]$ represents mathematical expectation and $\beta = x_1 w_1 + x_2 w_2 + b$.

Equation (4) shows that w'_0 depends on the current value of w_0 and mathematical expectation of x_0 . Since there is always a certain error between the predicted value and ground truth, $\beta - y$ cannot be zero. When the initial value of w_0 is positive, $E[x_0^2]w_0$ is positive, but $(\beta - y)E[x_0]$ is indeterminate. The update direction of w_0 is indeterminate. Similarly, when the initial value of w_0 is negative, it is also indeterminate. Therefore, w'_0 may not converge to zero, and the NN is affected by the irrelevant feature x_0 , especially when the irrelevant feature is biased.

Regularization is generally believed to be an effective way to improve the NN inference performance. To reduce the

influence of the irrelevant feature, L2 regularization is introduced into the NN, due to the fact that L2 regularization can make the weight tend to zero. The loss function $loss$ after introducing L2 regularization can be written as Eqs. (5), and (4) can be replaced by Eq. (6).

$$loss = (\hat{y} - y)^2 + \frac{1}{2}\lambda \sum_{i=0}^2 w_i^2 \quad (5)$$

$$w'_0 = (1 - \alpha\lambda)w_0 - 2\alpha(E[x_0^2]w_0 + (\beta - y)E[x_0]) \quad (6)$$

where λ denotes a penalty factor. Equation (6) shows that L2 regularization can limit the size of w'_0 to some extent, but cannot make w'_0 converge to zero. Therefore, L2 regularization still cannot solve the problem that the NN is affected by the irrelevant feature.

The above analyzes the influence of the irrelevant factor on the NN. However, in practice, traditional regression-based models rarely have this problem. In the traditional regression model, samples in the dataset have been carefully selected manually, and features in selected samples are considered relevant features. On the other hand, the preprocessing of each feature is completed independently, and each feature has undergone independent normalization, standardization, etc. These processes ensure that the mathematical expectation of irrelevant features in the dataset can be statistically zero. In addition, there are regularization methods such as penalty functions. Therefore, in traditional machine learning, irrelevant factors have little or no influence on the model accuracy.

However, for the NN, especially in the field of computer vision, all pixels in an image undergo the same transformation function. It is difficult to manually select features. When irrelevant features in the dataset are biased, the “attention” of the NN may be disturbed by these biased irrelevant features, resulting in the decline of the NN accuracy.

4 A method to reduce the task-irrelevant factors' influence

Based on the analysis in Sect. 3, we propose an idea to address the problem of task-irrelevant factors. According to this idea, we propose a method to reduce the influence of the background factor on the image classification task.

4.1 The idea

The analysis in Sect. 3 shows that when the irrelevant feature is biased, the NN is inevitably affected by task-irrelevant features. The existing approach is to use empirical algorithms to limit the solution space through the Lagrange multiplier

method. It is assumed that the weight of each feature in the solution space is close to zero. Generally, there are L1 regularization, L2 regularization, etc. Regularization restricts not only the weight of irrelevant features but also the weight of relevant features, which reduces the sensitivity of the NN to relevant features. Furthermore, the penalty factor of regularization is a hyperparameter that is set according to the experience of the researcher. Whether it is too large or too small, it will affect the convergence of the NN. Too large will degrade the NN into linear regression, while too small will make the regularization unsatisfactory. To solve the problem of irrelevant factors, some method is needed to make the weight of irrelevant features tend to zero without affecting the weight of relevant features.

To reduce the influence of irrelevant factors, the key is to reduce the absolute value of the irrelevant feature weight. Equation (4) shows that the update of w_0 depends on the mathematical expectation of x_0 and the error between the predicted value and ground truth. Since the error is inevitable, we focus on the mathematical expectation of x_0 . We preprocess the original training set to make the mathematical expectation of x_0 to be zero, that is, $E[x_0] = 0$. After preprocessing the training set, Eq. (4) can be replaced as follows:

$$w'_0 = w_0 - 2\alpha E[x_0^2]w_0 \quad (7)$$

It shows that w'_0 only depends on its current value. Then, w'_0 is analyzed from the initial state of model training. When the initial value of w_0 is positive and the calculated gradient value is positive, the w_0 update direction is negative, then the absolute value of w_0 decreases. When the initial value of w_0 is negative and the calculated gradient value is negative, the w_0 update direction is positive, then the absolute value of w'_0 decreases. Since the absolute value of w'_0 decreases, the update step of w'_0 during the next update is reduced. As a result, the overall trend of w_0 is constantly approaching zero. It turns out the idea proposed in this section can reduce the influence of task-irrelevant features. In particular, when the value of irrelevant features are made be zero, weights of irrelevant features are not updated during NN training. When the trained NN is used to infer the test sample with non-zero irrelevant features, the NN is still affected by irrelevant features. Therefore, the mathematical expectation of irrelevant features needs to be zero, rather than simply making the value of irrelevant features be zero.

4.2 The method

In an image, the background always exists. Generally, on the computer vision task, the entire image is input to the NN. When the image background is an irrelevant factor, the NN accuracy is affected by the background factor, as shown in

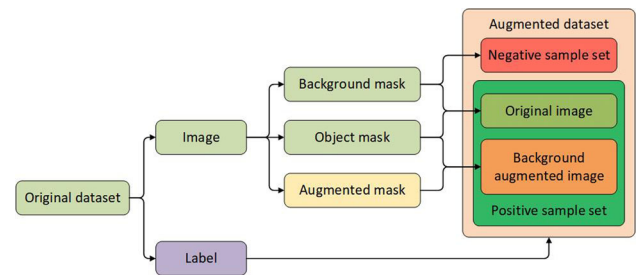


Fig. 4 Image dataset preprocessing method

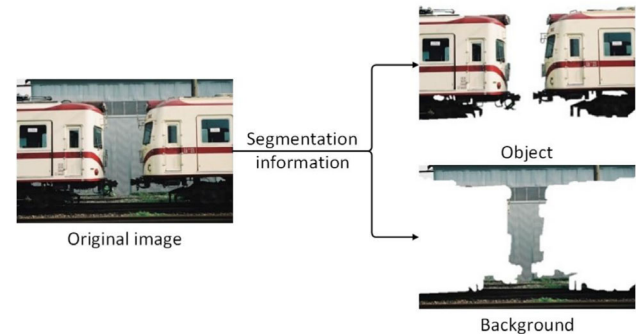


Fig. 5 Separate the object from the background

Fig. 1. The background factor is the most common irrelevant factor. Therefore, it is necessary to reduce the influence of the background factor on the NN. According to the idea proposed above, we propose a method that can reduce the influence of the background factor on the NN accuracy.

As mentioned earlier, since each feature may be related to multiple factors, it is difficult to select and process features related to the background factor. Based on the fact that deep features are calculated from shallow features, we deal with the shallowest features of the background factor, that is, pixels in the background area of the image. From the analysis in Sect. 4.1, it can be known that to reduce the influence of the background factor, the key is to make the mathematical expectation of the background feature to be zero. To perform a zero-expectation transformation on pixels in the background area, we propose a dataset preprocessing method as shown in Fig. 4. This method selectively deals with the background feature. Through this method, a positive sample set and a negative sample set are constructed for the NN training. This method can also be used as a new data augmentation method.

The method shown in Fig. 4 can be divided into three steps: (i) separating the object from the background, (ii) constructing augmented masks with zero-expectation, and (iii) constructing a background augmented dataset. Next, each step is described in detail.

4.2.1 Separate the object from the background

This step aims to separate an original image into an object mask and a background mask. To separate the object and background of an image, as shown in Fig. 5, this paper uses the segmentation information of the image. For example, in the image segmentation subset of PASCAL Visual Object Classes (VOC) dataset [27], each image has its segmentation information, which can be regarded as a table with the same size as the image. The value in the table is used to distinguish whether the pixel at the corresponding position in the image is an object pixel or a background pixel. If the value is 1, the pixel at the corresponding position of the image is an object pixel, otherwise, it is a background pixel. According to different values in the table, object pixels in the image can be separated from background pixels, and then the object can be separated from the background. Take an image from the original dataset and mark it as *img*, and mark the corresponding segmentation information as *mask*. The background mask and object mask of the image can be defined respectively as:

$$mask_{background} = \frac{(img \cdot (!mask))}{255} \times 2 - 1 \quad (8)$$

$$mask_{object} = \frac{(img \cdot mask)}{255} \times 2 - 1 \quad (9)$$

4.2.2 Construct augmented masks with zero-expectation

Firstly, a 3-dimensional matrix *A* is generated based on the standard normal distribution. Three dimensions of matrix *A* are *n*, *H*, and *W*, where *n* denotes the number of augmented masks, *H* denotes the height of the image, and *W* denotes the width of the image. According to the standard normal distribution, it can be known that $\Phi(2.75) = 0.997$. To ensure that the distribution of generated elements in the interval $[-1, 1]$ could be close to the normal distribution, we transform all elements in matrix *A*, that is, $A = A/2.75$. Each $A_i (i = 1, 2, \dots, n)$ can be regarded as an augmented mask, which is used to replace the background of a certain channel in an original image.

4.2.3 Construct a background augmented dataset

This step uses object masks and augmented masks of previous two steps to construct a background augmented dataset. Generally, the image input to the NN is an RGB 3-channel image. Therefore, an original image requires three augmented masks to generate a new sample. Firstly, three augmented masks are randomly selected from matrix *A* to form a matrix *B*, where the dimension of *B* is $[H, W, 3]$. Then, the pixel in the object mask is replaced with the pixel in $B_i (i = 1, 2, 3)$, where the

pixel is zero. A background augmented sample can be defined as:

$$sample_{augmented} = mask_{object} + (!mask) \cdot B \quad (10)$$

According to the method shown in Fig. 4, the augmented dataset used for model training contains three types of images: original image, background augmented image, and background image. Moreover, original images and background augmented images are used as positive samples, while background images are used as negative samples.

It can be seen that our method has the following advantages:

- Our method does not introduce extra hyperparameter. When using regularization, the penalty factor is often a hyperparameter, which requires the experience of the researcher. Whether it is too large or too small, it will affect the convergence of the training process. On the contrary, our method does not need to set the hyperparameter, which eliminates the possible influence of the hyperparameter on the NN.
- Our method reduces the influence of irrelevant factors. Regularization can make the weight of irrelevant features tend to zero to a certain extent. However, regularization acts on all features. Regularization restricts not only the weight of irrelevant features but also the weight of relevant features, resulting in the reduction of the NN sensitivity to relevant features. Unlike regularization, our method can reduce the influence of irrelevant features without affecting relevant features, so it does not affect the sensitivity of the NN to relevant features.
- The process of constructing a augmented dataset in our method can be regarded as a data augmentation method. Different from traditional data augmentation methods, our method focuses on improving the dataset quality.

5 Experimental analysis

In this section, we first conduct experiments with our method on the image classification task. Then, our method is extended to the field of object detection.

5.1 Experiment on the image classification task

For the image classification task, MNIST dataset [28] is used for the experiment. MNIST dataset contains handwritten digital images with white numbers on black background from 0 to 9. Pixels in the background area are displayed as black, while pixels in the digital area are displayed as white. Using different pixel values in different areas, it is easy to separate

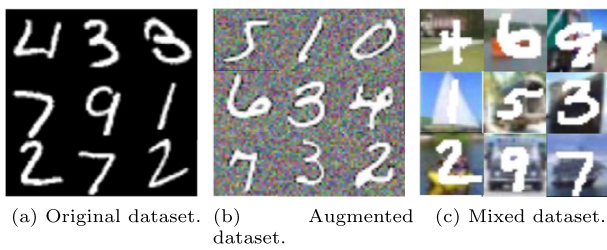


Fig. 6 Three types of datasets used for training or testing. **a** Original dataset: the original MNIST dataset, which is used for training and testing. **b** Augmented dataset: the augmented MNIST dataset, which is used for training and testing. **c** Mixed dataset: the mixed dataset generated by the original MNIST dataset and CIFAR-10 dataset, which is only used for testing

the digit from the background. Therefore, the data in MNIST dataset can act as the training set as well as the segmentation information. According to our method, an augmented MNIST dataset can be generated. To further study the difference in model generalization ability before and after using our method, MNIST dataset and CIFAR-10 dataset [29] are used to generate a new dataset-mixed dataset, which is used as a test set. The generation method is similar to our method, except that the augmented mask is replaced with the image in CIFAR-10 dataset. CIFAR-10 dataset does not participate in any form of model training. It is only used to generate a new test set. Fig. 6 shows three types of datasets used for training or testing.

To verify the universality of our method, several NN models [30–32] are selected for the experiment. The selection of models is representative, including mini NN models and deep NN models. Firstly, models are trained from scratch on the training set of the original dataset as a benchmark experiment. As a comparison, these models are trained from scratch on the training set of the augmented dataset. Then, these trained models are tested on the test set of the original dataset, augmented dataset, and mixed dataset, respectively. As evaluation metrics, the accuracy is considered in this experiment. Experimental results are analyzed in terms of three measures: (1) benchmark test, (2) cross-test, and (3) generalization ability test.

5.2 Experimental results on the image classification task

According to different training sets and test sets, each model has six types of test results. Table 1 shows each model's accuracy on the three types of test sets without using our method, while Table 2 shows each model's accuracy after using our method. Whether using our method, each model has high accuracy on the test set of the original dataset. However, on the test set of the augmented dataset and mixed dataset, the accuracy of the model trained without using our

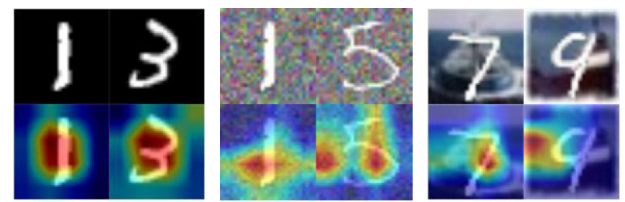


Fig. 7 Visualization examples of ResNet34 trained with the original dataset. The upper image is the test sample, while the lower image shows the focus of ResNet34 on the test sample with GradCAM

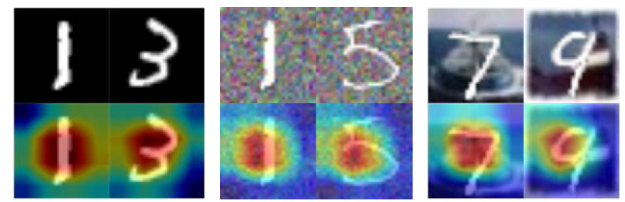


Fig. 8 Visualization examples of ResNet34 trained with the augmented dataset. The upper image is the test sample, while the lower image shows the focus of ResNet34 on the test sample with GradCAM

Fig. 8 Visualization examples of ResNet34 trained with the augmented dataset. The upper image is the test sample, while the lower image shows the focus of ResNet34 on the test sample with GradCAM

method has dropped sharply, while the model trained using our method still has a relatively high accuracy. Fig. 7 shows some visualization examples of ResNet34 trained with the original dataset on the original dataset, augmented dataset and mixed dataset, while Fig. 8 shows some visualization examples of ResNet34 trained with the augmented dataset on the original dataset, augmented dataset and mixed dataset. After trained on the original dataset, ResNet34 focuses on the digits on the original dataset, but focuses on parts of the digits and the background on the augmented dataset and mixed dataset. After trained on the augmented dataset, ResNet34 focuses on digits on different datasets. This shows that the ResNet34 trained on the augmented dataset is not affected by the background and can effectively recognize digits on different datasets. Next, experimental results are analyzed in more detail.

To verify the fitting ability of each model, the benchmark test result is presented in Fig. 9a. It can be seen that when the training set and test set are the datasets of the same type, each model's accuracy is close. The result shows that selected models have the ability to fit the distribution of the original dataset and augmented dataset.

To verify that the background factor affects the accuracy of the NN, the cross-test result is presented in Fig. 9b. It can be seen that the model trained on the original dataset has low accuracy on the augmented dataset, while the model

Table 1 Each model's accuracy on three types of test sets after training on the original dataset. The best results on each dataset are bolded

Model	Image classification accuracy (%)		
	Original dataset	Augmented dataset	Mixed dataset
Resnet18	99.08	39.59	41.99
Resnet34	99.13	48.91	55.69
Resnet50	98.49	34.90	45.30
Mobilenet_v2	98.32	23.17	29.18
Shufflenet_v2_x0_5	98.16	60.60	49.90
Shufflenet_v2_x1_0	98.51	63.19	53.89
Shufflenet_v2_x1_5	98.55	84.40	67.38
Shufflenet_v2_x2_0	98.61	48.59	49.50
Alexnet	97.02	45.11	25.51
Densenet-121	99.26	16.49	14.15
Densenet-161	99.17	41.56	38.59

Table 2 Each model's accuracy on three types of test sets after training on the augmented dataset. The best results on each dataset are bolded

Model	Image classification accuracy (%)		
	Original dataset	Augmented dataset	Mixed dataset
Resnet18	98.89	98.98	90.06
Resnet34	93.62	98.78	85.46
Resnet50	97.04	98.75	83.93
Mobilenet_v2	98.32	98.51	83.23
Shufflenet_v2_x0_5	97.73	97.87	80.31
Shufflenet_v2_x1_0	98.25	98.11	85.67
Shufflenet_v2_x1_5	98.57	98.33	82.23
Shufflenet_v2_x2_0	94.57	98.22	85.53
Alexnet	95.83	96.07	46.20
Densenet-121	82.98	99.26	68.17
Densenet-161	91.32	99.18	67.59

trained on the augmented dataset still maintains high accuracy on the original dataset. The experimental result shows that the background factor affects the NN accuracy on this classification task.

To verify the effectiveness of our method, the generalization ability test result is presented in Fig. 9c. It can be seen that the model trained on the augmented dataset still has high accuracy on the mixed dataset, while the model trained on the original dataset has low accuracy. The experimental result shows that our method can reduce the influence of the background factor on the NN and improves the generalization ability of the NN.

From the above three tests, it can be seen that the model trained on the original dataset, has high accuracy on the test set of the original dataset, but the accuracy decreases sharply when the background information changes greatly. The model trained on the augmented dataset remains high accuracy when the background information changes greatly. After using our method, the sensitivity of the image classification model to the background factor is reduced, and the generalization ability is improved.

5.3 Experiment on the object detection task

To further verify the effectiveness of our method, our method is extended to the field of object detection. Unlike the image classification task, object detection is a more complex computer vision task. In addition to classifying the object, it also needs to locate the object. Furthermore, an image often contains multiple objects to be detected. The object detection task is more difficult.

For the object detection task, the image segmentation subset in PASCAL VOC 2012 dataset is used in this experiment. According to our method, a new augmented dataset can be constructed, as shown in Fig. 10b.

For the selection of the model, YOLOv5 [33] is selected as a benchmark model for the experiment, and parameters of YOLOv5 are consistent with the official code. First, as a benchmark experiment, YOLOv5 is trained from scratch on the training set of the original VOC dataset. Then, as a contrast experiment, YOLOv5 is trained from scratch on the training set of the augmented dataset. For the object detec-

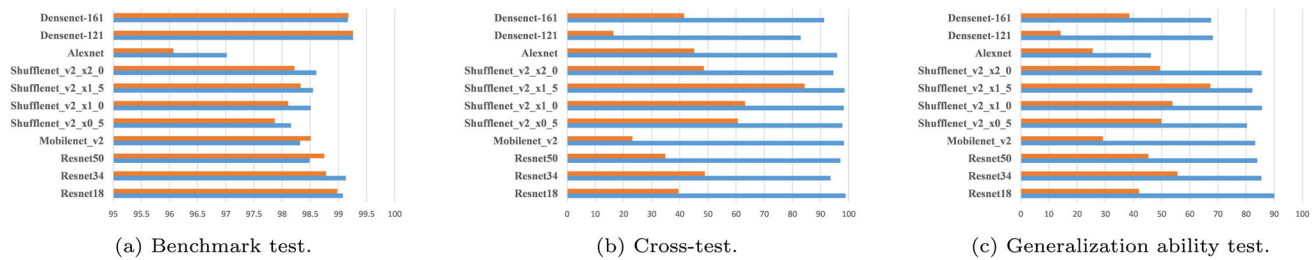


Fig. 9 Experimental results of benchmark test, cross-test, and generalization ability test. The orange represents the model trained on the original dataset, while the blue represents the model trained on the augmented dataset. For different comparative tests, the model is tested on different datasets. **a** Benchmark test: the orange and the blue are the

accuracies on the original dataset and augmented dataset, respectively. **b** Cross-test: the orange and the blue are the accuracies on the augmented dataset and original dataset, respectively. **c** Generalization ability test: the orange and the blue are the accuracies on the mixed dataset

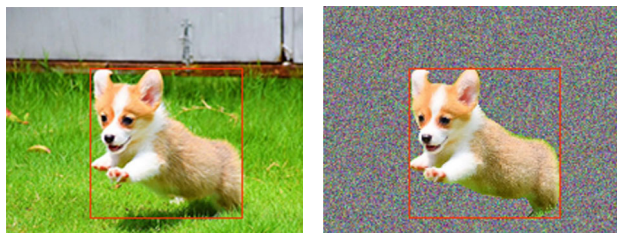


Fig. 10 Two types of datasets used for object detection experiments

Table 3 YOLOv5 inference performance. The best results of different evaluation metrics are bolded

Training set	mAP ¹	mAP@0.5 ²	Precision	Recall
Original dataset	50.6	72.7	0.8196	0.657
Augmented dataset	51.5	74.2	0.8013	0.668

¹ Mean Average Precision

² Mean Average Precision(Intersection over Union=0.5)

tion model, loss functions and mAP@0.5 are considered as evaluation metrics in the experiment.

5.4 Experimental results on the object detection task

During YOLOv5 training on two types of datasets respectively, changes of loss functions are shown in Figs. 11 and 12. Through the change of loss functions (localization loss, confidence loss, and classification loss), it can be seen that when YOLOv5 iterates about 120 rounds on either one, YOLOv5 converges. After YOLOv5 is trained on the augmented dataset, the lower limit of loss functions is lower, and YOLOv5 has a higher inference performance upper limit. The experimental result shows that our method can improve the inference performance of the object detection model.

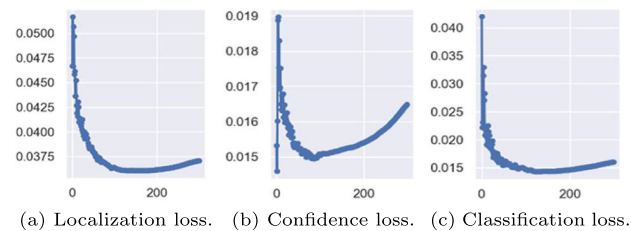


Fig. 11 The change of YOLOv5's loss functions over the iterations on the original dataset

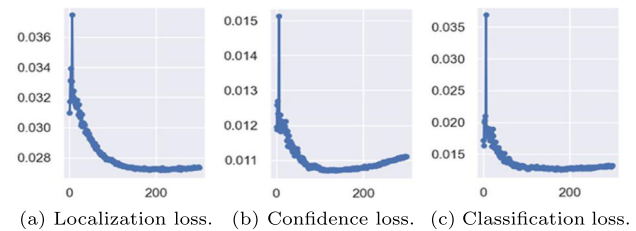


Fig. 12 The change of YOLOv5's loss functions over the iterations on the augmented dataset

Therefore, our method is still effective for the object detection task.

Before and after using our method, the difference in YOLOv5's precision on the test set of the original dataset is shown in Table 3. It can be seen that after using our method, mAP@0.5 of YOLOv5 is improved by 1.5%. Compared with the image classification model, the gain of precision is little in the object detection model. Through observation, it is found that for some detection objects (such as ships, bottles, etc.) with a wide disparity in length and width ratios, YOLOv5 after using our method is easy to recognize similar background areas as objects. This may be related to the fact that YOLOv5 does not separately detect the class "background". Figs. 13 and 14 show some visualization examples of YOLOv5 trained with the augmented dataset and the original dataset with gradCAM, respectively.

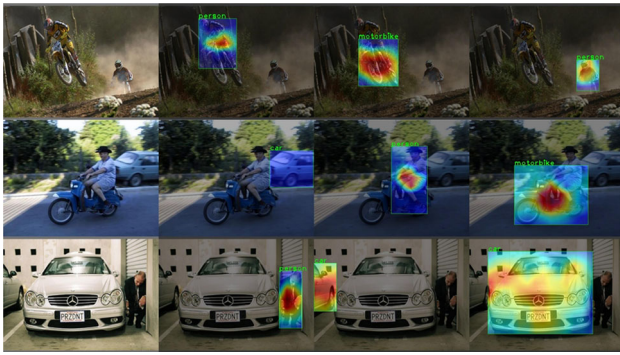


Fig. 13 Visualization examples with GradCAM of YOLOv5 trained on the original VOC dataset

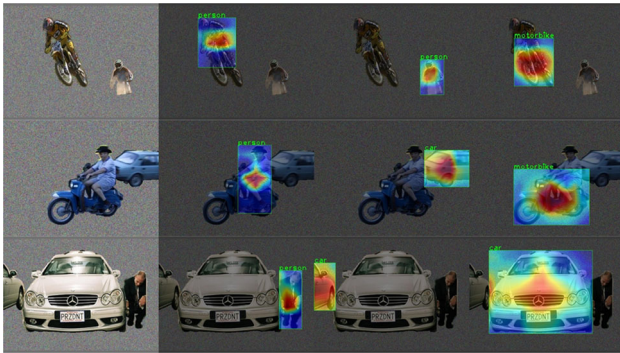


Fig. 14 Visualization examples with GradCAM of YOLOv5 trained on the augmented VOC dataset

5.5 Discussion on the applicability of the method

On image classification and object detection tasks, the effectiveness of our method is verified. In practice, our method relies on image segmentation information. For image tasks with the single background, background segmentation can be achieved based on pixel values. For example, the image background of the Mvtec 3D-AD dataset [34] is black. Based on different pixel values, the background and foreground of the images in the Mvtec 3D-AD dataset can be segmented. For the object detection task, our method can further improve the performance of YOLOv5. Pretrained YOLOv5 is used in various downstream tasks such as defect detection [35], state recognition [36], unmanned aerial vehicle [37], etc. Our method can improve the performance of pre-trained models on pretraining datasets, and make pre-trained models perform better in downstream tasks.

6 Conclusion

This study analyzes the influence of task-irrelevant factors on the NN through mathematical derivation. Through analysis, it is found that when irrelevant features in the dataset are biased,

irrelevant features will be regarded as relevant features by the NN, which affect the NN accuracy. Based on the analysis, this paper proposes an idea to address the problem of irrelevant factors. According to this idea, this paper proposes a method to reduce the influence of the background factor. This method selectively processes background features without affecting relevant features. Experimental results show that this method not only greatly improves the accuracy of the image classification model, but also effectively improves the precision of the object detection benchmark model. Moreover, this method can be used a data augmentation method, which substantially improves the dataset quality. In the future, reducing the background factor on the performance of image classification models without segmentation information will be studied.

Acknowledgements This work was supported by the Key Project of National Natural Science Foundation of China (U1908212), the Fundamental Research Funds for the Central Universities (N2017013, N2017014), and the Ningxia Natural Science Foundation (2024AAC 03349).

Author Contributions All authors contributed equally to this work.

Data availability The datasets are available in the MNIST repository, <http://yann.lecun.com/exdb/mnist/>, CIFAR-10 repository, <http://www.cs.toronto.edu/~protect/unhbox/voidb@x\penalty\@M\kriz/cifar.html>, and PASCAL VOC 2012 repository, <http://host.robots.ox.ac.uk:8080/pascal/VOC/>.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **106**, 249–259 (2018)
2. Şen, S.Y., Özkurt, N.: Convolutional neural network hyperparameter tuning with adam optimizer for ECG classification. In *2020 Innovations in Intelligent Systems and Applications Conference* (2020)
3. Wawrzyński, P., Zawistowski, P., Lepak, Ł.: Automatic hyperparameter tuning in on-line learning: classic Momentum and ADAM. In *2020 International Joint Conference on Neural Networks* (2020)
4. Nguyen, Q., Teku, N., Bose, T.: Epsilon greedy strategy for hyperparameters tuning of a neural network equalizer. In *2021 12th International Symposium on Image and Signal Processing and Analysis* (2021)
5. Vulpe-Grigoraşi, A., Grigore, O.: Convolutional neural network hyperparameters optimization for facial emotion recognition. In *2021 12th International Symposium on Advanced Topics in Electrical Engineering* (2021)
6. Zhang, A., Yao, F.: Analysis of effective e-commerce coordination big data processing strategies under infinite deep neural network topology. In *2021 Asia-Pacific Conference on Communications Technology and Computer Science* (2021)

7. Yang, X., Wang, J., Liu, C.: Reconstructing neural network topology from firing activity. In 2020 39th Chinese Control Conference (CCC) (2020)
8. Kucuk, M., Uysal, I.: Performance analysis of neural network topologies and hyperparameters for deep clustering. In 2020 International Joint Conference on Neural Networks (2020)
9. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should i trust you?” Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)
10. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision (2017)
11. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.R.: Unmasking Clever Hans predictors and assessing what machines really learn. *Nat. Commun.* **10**(1), 1096 (2019)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural Inform. Process. Syst.*, 25 (2012)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
14. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence (2020)
15. Waheed, A., Goyal, M., Gupta, D., Khanna, A., Al-Turjman, F., Pinheiro, P.R.: Covidgan: data augmentation using auxiliary classifier gan for improved Covid-19 detection. *IEEE Access* **8**, 91916–91923 (2020)
16. Li, B., Wu, F., Lim, S.N., Belongie, S., Weinberger, K.Q.: On feature normalization and data augmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
17. Hinton, G.E.: Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012)
18. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *J Big Data* **6**(1), 1–48 (2019)
19. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
20. Inoue, H.: Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929* (2018)
21. Zhang, H.: mixup: beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017)
22. Phankokkrud, M.: Convolutional neural network models for deep face recognition on limitation and interfering factors in image dataset. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) (2018)
23. Kim, T., Kim, Y.: Suppressing spoof-irrelevant factors for domain-agnostic face anti-spoofing. *IEEE Access* **9**, 86966–86974 (2021)
24. Belloni, C., Balleri, A., Aouf, N., Le Caillec, J.M., Merlet, T.: Explainability of deep SAR ATR through feature analysis. *IEEE Transact Aerosp Electron. Syst.* **57**(1), 659–673 (2020)
25. Han, P., Chen, Z., Wan, Y., Cheng, Z.: PolSAR image classification based on optimal feature and convolution neural network. In IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium (2020)
26. Song, Q., Ni, J., Wang, G.: A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transact. Knowl. Data Eng.* **25**(1), 1–14 (2011)
27. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **88**, 303–338 (2010)
28. Lecun, Y., Cortes, C.: The mnist database of handwritten digits, <http://yann.lecun.com/exdb/mnist> (2010)
29. rizhevsky, A., Hinton, G.: Learning Multiple Layers of Features from Tiny Images (2009)
30. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
31. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (2018)
32. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
33. Jocher, G., Nishimura, K., Mineeva, T.: YOLOv5, <https://github.com/ultralytics/yolov5> (2020)
34. Bergmann, P., Jin, X., Sattlegger, D., Steger, C.: The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. Proceedings of the 17th International Joint Conference on Computer Vision (2021)
35. Lailesh, A.K., Richi, J.A., Preethi, N.: A pre-trained yolo-v5 model and an image subtraction approach for printed circuit board defect detection. In 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE) (2023)
36. Liao, Y.Y., Ryu, K.: Status recognition using pre-trained YOLOv5 for sustainable human-robot collaboration (HRC) system in mold assembly. *Sustainability* **13**(21), 12044 (2021)
37. Liu, W., Quijano, K., Crawford, M.M.: YOLOv5-tassel: detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **15**, 8085–8094 (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.