



Image entropy equalization: A novel preprocessing technique for image recognition tasks



Toshitaka Hayashi ^a, Dalibor Cimr ^a, Hamido Fujita ^{b,c,d,*}, Richard Cimler ^a

^a Faculty of Science, University of Hradec Kralove, Hradec Kralove, Czech Republic

^b Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi, 54100 Kuala Lumpur, Malaysia

^c DaSCI Andalusian Institute of Data Science and Computational Intelligence, University of Granada, Granada, Spain

^d Regional Research Center, Iwate Prefectural University, Takizawa, Japan

ARTICLE INFO

Keywords:

Image entropy
Auto encoder
Preprocessing
One-class classification

ABSTRACT

Image entropy is the metric used to represent a complexity of an image. This study considers the hypothesis that image entropy differences affect machine learning algorithms' performance. This paper proposes a novel preprocessing technique, image entropy equalization, to delete the image entropy differences. The goal is to transform all images into the same entropy. Such a process is implemented by editing all images into the same histogram. Image entropy equalization is evaluated by comparing the original and equalized images in various machine learning tasks. The main advantage of image entropy equalization is to improve the AUC score for one-class autoencoder (OCAE). This result gives a new hypothesis that using image entropy equalization could improve various studies using autoencoder (AE). In addition, the proposed method shows fair results for classification and regression tasks. On the other hand, the main challenges are that the equalization process depends on a reference histogram and is affected by diverse backgrounds.

1. Introduction

In computer science, images are a common data type for machine learning. Image recognition is one of the widespread challenges, and deep learning techniques [1] have increased attention to tackling such data.

Besides, image entropy is the metric used to represent image complexity. Such a metric was initially developed by Shannon [2] in the context of information theory and was imported to image data by several researchers [3,4].

The core hypothesis is that image entropy differences give bias to machine learning algorithms. The idea was born from autoencoder (AE), which is a neural network to reconstruct input [5,6]; the initial hypothesis was that high entropy images are more challenging to reconstruct than low entropy ones [7], and the bias is considered as a problem where the system uses model errors (ex., one-class classification (OCC)).

Image entropy equalization is proposed to tackle the bias due to image entropy differences. The goal is to make all images into the same entropy, and for this purpose, all images are edited into the same histogram. Initial ideas were reported in the conference paper [7].

* Corresponding author at: Kotorizawa, 2-27-5, Morioka, Iwate 020-0104, Japan.

E-mail addresses: toshitaka.hayashi@uhk.cz (T. Hayashi), dalibor.cimr@uhk.cz (D. Cimr), fujitahamido@utm.edu, hfujita799@acm.org (H. Fujita), richard.cimler@uhk.cz (R. Cimler).

This paper is an extended version with the motivation to address the following two questions, specifically whether image entropy equalization is effective for other computer vision tasks, and what is the best entropy value (histogram) to equalize. The main changes from [7] are: 1) applying image entropy equalization to more tasks, such as classification and image transformation, while [7] considered only the bias in one-class autoencoder (OCAE),, and 2) The number of histograms for equalization is increased to 19 from 1.

The contributions in this paper are listed as follows:

- The main originality is the image entropy equalization process. The proposed method is implemented by editing all images into the same histogram because the images with the same histogram have the same entropy. To our knowledge, this paper is the first journal article to describe equalizing image entropy.
- Evaluation is made by comparing the results with/without image entropy equalization on classification, one-class classification, reconstruction, and image transformation tasks. Twenty histograms are used for this experiment.
- The main advantage of image entropy equalization is reducing the bias for AE and OCAE. This aspect might improve various studies using AE.
- Other negative results, and discussions for future directions, are provided.

The organization of the paper is summarized as follows. Section 2 describes related work for image entropy and preprocessing methods. Section 3 presents the image entropy equalization process. Sections 4 and 5 provide experiment results and discussions, respectively. Finally, section 6 gives the conclusion.

2. Related work

2.1. Image entropy

Image entropy is computed by (1):

$$-\sum_v p_v \log_2(p_v) \quad (1)$$

where v is the gray level, and p_v is the probability associated with v . This equation is known as Shannon entropy [1] and was applied to image analysis in the 70 s and 80 s [3,4]. Such a value is computed from an image histogram. For 8-bit data, image entropy acquires values from 0 to 8; the minimum is 0, where all pixels have the same values, while the maximum is 8, where all pixel values are equally distributed.

Several studies aim to change the image entropy values. These studies are classified into two purposes: increasing and decreasing entropy. The common process is to change the histogram distribution.

Histogram equalization is the method to increase image entropy by distributing pixel values equally. Hall [8] proposed initial histogram equalization to improve “image visibility” in the whole image. Ketcham [9] proposed local histogram equalization to enhance visibility in the local area. Verdenet et al. [10] connected the concepts of histogram equalization and image entropy by reporting that histogram equalization maximizes image entropy.

On the other hand, Guo et al. [11] proposed image entropy minimization; their method minimizes image entropy by reconstruction. Recently, Chen et al. [12] proposed histogram mergence, which merges pixel values to minimize entropy; the idea is the opposite of histogram equalization.

Besides, histogram specification is the technique to transform the images into the reference histogram [13]. Such an approach can address both entropy maximization and minimization.

Overall, existing image editing techniques consider the process at a single image level. On the other hand, the proposed image entropy equalization is the dataset-level method related to preprocessing algorithms described in the next section.

2.2. Preprocessing techniques

The preprocessing technique aims to transform the dataset into suitable inputs for machine learning. For this purpose, various methods are proposed [14].

Image normalization is the technique to scale the pixel values [15]. Generally, pixel values are normalized into the range of (0, 1), (-1, 1), or (0, 255). For this purpose, min–max normalization is the most popular approach. However, such a process is degraded where the minimum or the maximum values are an outlier. In contrast, 1%-99% normalization is robust to outlier pixels. In addition, $\pm 3\sigma$ normalization considers the case where original data has Gaussian distribution.

Image segmentation is the technique to divide image into multiple regions. For this purpose, Hinojosa et al. [40] applied Stochastic Fractal Search to minimize three entropies to decide the best threshold to split the pixels. It is important to note that entropies in [40] correspond to uncertainty of classification that are different concepts from image entropy to represent data complexity.

Correspondingly, foreground/background separation is a significant challenge because the diversity of background affects computer vision tasks. For example, Liu et al. [16] separated foreground and background by color, while Shao et al. [17] applied item detection to extract the foreground from X-ray images.

In addition, noise and blur are trade-off practices, where solving one issue causes another problem. Noise reduction is performed by

filters, such as average, Gaussian, median, and transformation (like wavelet transform) and its reconstruction [14], while super-resolution [18] aims to improve blur.

Image augmentation is the technique to increase training images [19] and can address the problems due to limited training data like data imbalance [20] and few samples [21]. Various augmentation algorithms are proposed, such as geometric transformation [22], mixing images [23], random erasing [24], using generative adversarial net (GAN) [25], and other preprocessing techniques mentioned above.

Besides, Ma et al. [26] proposed the cross-entropy normalization on the similarity score ranking curve for image retrieval. They used the reference curve for normalization. Such an idea is similar to image editing. However, cross-entropy is a similarity metric, while image entropy is the complexity of the image.

The following section describes image entropy equalization. With our knowledge, a preprocessing method that aims to equalize image entropy did not exist (except for our conference paper [7] and this submission).

3. Image entropy equalization

Image entropy equalization aims to make all images into the same entropy. The originality is changing image entropy at the dataset-level while existing image entropy editing/processing methods address the issues at a single image-level.

Image entropy equalization is implemented by editing all images into the same histogram. The input is an image dataset, and the output is equalized images. In addition, a histogram is required as a parameter for equalization.

Let an input image as X , which is an $M \times N$ matrix as shown in (2):

$$X = \begin{bmatrix} X_{11} & \cdots & X_{1N} \\ \vdots & \ddots & \vdots \\ X_{M1} & \cdots & X_{MN} \end{bmatrix} \quad (2)$$

In addition, an output image is defined as X' with the same size as X .

The goal of image editing is to replace pixels in X to pixel set P , which is a sequence created from an arbitrary histogram (generated by optional ways, such as random, average of training images, or the extraction from a specific image as the same size as X) as shown in (3):

$$P = [P_1, P_2, \dots, P_i, \dots, P_{M \times N}], \quad (3)$$

where i is an index; moreover, values in P are sorted.

Then, images are edited by applying Algorithm, which has three steps: 1) sorting pixels in X , 2) indexing positions for sorted pixels, and 3) assigning pixel set P to sorted positions using the index.

Algorithm. Image editing process

```

Input: image X. Pixel set P
Output: edited image X'
//1. Sort pixels in X.
Sorted_pixel = sort (pixels in X)
//2. Index the positions for sorted pixels.
Sorted_position = positions for sorted_pixel
//3. Assign the pixel set to the sorted position.
X' = np.zeros([M, N])
For i in range(len(sorted_position)):
    X'[sorted_position[i]] = P[i]

```

If the image has multiple channels (ex., RGB), the process can be applied channel by channel. Another option is to transform the RGB image to grayscale. This study used the first option because more channels give more information.

4. Experiment

The experiment is performed by comparing the original and entropy equalized images for image recognition tasks. Section 4.1 describe the dataset. Section 4.2 explains image recognition tasks and evaluation metrics. Finally, section 4.3 gives the experiment

Table 1

Dataset descriptions.

Dataset	Class	Training	Testing	Size
MNIST	10	60,000	10,000	$28 \times 28 \times 1$
FMNIST	10	60,000	10,000	$28 \times 28 \times 1$
CIFAR10	10	50,000	10,000	$32 \times 32 \times 3$

results.

4.1. Dataset

This study uses three datasets, MNIST [27], Fashion MNIST (FMNIST) [28], and CIFAR10 [29]. These datasets are common public datasets for computer vision tasks. Table 1 provides the dataset descriptions, number of classes, data balance for training and testing data, and image size. All datasets are balanced and have 10 classes with 10,000 testing data. MNIST and FMNIST are grayscale images, while CIFAR10 contains RGB channels.

In addition, Table 2 reports the average and standard deviations of image entropies for each class. The reported entropy on CIFAR10 is the summation of entropy in the RGB channels.

Training data are split into the train (data for updating the weight) and valid (data for evaluating the model in the training phase) in 7:3 ratios. In addition, pixel values are normalized in the 0–255.

4.2. Machine learning task and evaluation metrics

This study aims to evaluate image entropy equalization using various machine learning tasks. For this purpose, four tasks are considered: supervised classification, one-class classification (OCC), reconstruction, and image transformation.

Supervised classification is well-developed research area:

$$f : X \rightarrow Y, \text{ where } Y = \{C1, C2, \dots, Cn\}, \quad (4)$$

where C is class labels, and n represents the number of classes. Evaluation is done by accuracy, computed as the number of correctly classified images divided by the number of all images.

One-class classification is a task with only one class as training data. The objective of OCC is to classify data into seen or unseen classes as (5):

$$OCC : X \rightarrow Y, \text{ where } Y = \{\text{Seen}, \text{Unseen}\} \quad (5)$$

OCC is processed as (6):

$$OCC(X) = \begin{cases} \text{Seen} & (\text{score}(X) \geq \lambda) \\ \text{Unseen} & (\text{score}(X) < \lambda), \end{cases} \quad (6)$$

where λ is a threshold value, and the score is an arbitrary function (the main challenge to developing OCC algorithms) related to the seen class (If OCC algorithms use arbitrary distance metrics from seen class, the score will be its negative values.). In addition, OCC is evaluated by Area Under the ROC curve (AUC) because deciding the threshold value is challenging for OCC; using the AUC value can ignore this aspect because it is computed from all possible thresholds.

This study applies OCAE (One-class autoencoder) [30] and OCITN (One-class image transformation network) [31] for evaluation. These methods use model errors to discriminate between seen and unseen classes.

OCAE uses an Autoencoder (AE) [30], which is a neural network combining two sub-networks, an encoder and a decoder. The encoder aims to compress input, as shown in (7), while the decoder seeks to reconstruct the input from the compression, as shown in (8). AE reconstructs the input by connecting these two networks, as shown in (9):

$$\text{Encoder} : X \rightarrow F, \quad (7)$$

$$\text{Decoder} : F \rightarrow X, \quad (8)$$

$$AE : X \rightarrow \text{Decoder}(\text{Encoder}(X)) = X \rightarrow F \rightarrow X, \quad (9)$$

Table 2
Image entropy for datasets.

MNIST		FMNIST		CIFAR10	
Label	Entropy	Label	Entropy	Label	Entropy
0	1.91 ± 0.27	0	4.60 ± 0.58	PLANE	19.29 ± 2.75
1	0.94 ± 0.19	1	3.19 ± 0.49	CAR	21.54 ± 1.35
2	1.76 ± 0.29	2	5.02 ± 0.50	BIRD	19.90 ± 2.10
3	1.74 ± 0.29	3	3.68 ± 0.70	CAT	21.17 ± 1.35
4	1.55 ± 0.24	4	4.90 ± 0.53	DEER	20.46 ± 1.42
5	1.67 ± 0.29	5	2.95 ± 0.83	DOG	21.33 ± 1.18
6	1.71 ± 0.29	6	4.87 ± 0.62	FROG	20.70 ± 1.30
7	1.42 ± 0.24	7	3.15 ± 0.45	HORSE	21.29 ± 1.15
8	1.88 ± 0.29	8	4.78 ± 0.76	SHIP	20.96 ± 1.41
9	1.59 ± 0.24	9	4.19 ± 0.52	TRUCK	21.72 ± 1.10
Average	1.61	Average	4.13	Average	20.84

where F is the compression of an image. In addition, AE is evaluated by reconstruction error RE, which is a distance metric between input and reconstructed input, as shown in (10):

$$RE(X) = |X - AE(X)|. \quad (10)$$

OCAE uses the hypothesis that RE for unseen class is large relative to seen class, as shown in equation (11):

$$OCAE(X) = \begin{cases} \text{Seen } (RE(X) \leq \lambda) \\ \text{Unseen } (RE(X) > \lambda) \end{cases}, \quad (11)$$

where AE is trained from images belonging to seen class.

On the other hand, OCITN [31] uses an image transformation network (ITN), which aims to transform all images into one image, as shown in formula (12):

$$ITN : X \rightarrow I, \quad (12)$$

I is an image called goal image, an ideal model output. ITN is evaluated using construction error CE, which is the distance metric between I and model output, as shown in equation (13):

$$CE(X) = |I - ITN(X)|. \quad (13)$$

OCITN uses the hypothesis that CE for unseen class is large relative to seen class, as shown in formula (14):

$$OCITN(X) = \begin{cases} \text{Seen } (CE(X) \leq \lambda) \\ \text{Unseen } (CE(X) > \lambda) \end{cases} \text{ where } (ITN : X_{\text{seen}} \rightarrow I). \quad (14)$$

In addition, experiments are done for reconstruction and image transformation processes. Although these tasks are subtasks of OCC and not realistic problems, considering the relation between image entropy and model errors could give the idea to improve image processing.

Table 3 summarizes experimented machine learning tasks in terms of the number of trained classes and evaluation metrics. OCC uses only one trained class, while other tasks use all classes for training model.

All experiments are done five times, and the average and standard deviations are reported. The machine used in the experiment has the following characteristics: NVIDIA GeForce RTX 3080, Intel(R) Core (TM) i7-10700 K CPU @ 3.80 GHz, RAM 64 GB.

4.3. Experiment result

Table 4 shows compared histograms in the experiment. For this purpose, 20 histograms are prepared, where ID 1 is the original, and other histograms are parameters for image entropy equalization.

Defined histograms (IDs 2–8) are created to distribute all pixel values equally as possible (Since images in MNIST and FMNIST have 784 pixels, these pixels are not dividable by 256, 128, 64, and 32.). Images with entropy = 8 use pixel values 0–255. Histograms with entropy = 7 and entropy = 6 use pixel values between 0–127 and 0–63, respectively; other histograms are prepared similarly. Finally, histograms are rescaled to 0–255. Alternatively, IDs 10–20 are computed as the average histograms in training data. These histograms are different for each dataset.

Figs. 1–3 provide the image entropy equalization results for MNIST, FMNIST, and CIFAR10, respectively. The first row shows the original images (taken from the first indexes) for each class, while other rows show the edited images correspond to reference histograms.

4.3.1. Classification results

First, image entropy equalization is evaluated in supervised classification task. The Wilcoxon signed-rank test is conducted to analyze the classification accuracy. Null hypothesis is that entropy equalized images do not outperform the original images in terms of accuracy. P-value under 0.05 is considered indicative of a significant improvement, while P-value above 0.95 is regarded as significant degradation.

The experiment applies deep learning models included in the Keras package [32], concretely, VGG16 [33], DenseNet201 [34],

Table 3
Summary of experiment.

Task	Number of trained classes	Evaluation Metric
Supervised Classification	10	Accuracy
One-class classification (OCAE and OCITN)	1	Area Under ROC curve (AUC)
Reconstruction (AE)	10	Reconstruction Error (Mean Absolute Error)
Image transformation (ITN)	10	Construction Error (Mean Absolute Error)

Table 4
Compared histograms.

ID	Histogram	ID	Histogram
1	Original	11	Average in class 1
2	Defined, entropy≈8	12	Average in class 2
3	Defined, entropy≈7	13	Average in class 3
4	Defined, entropy≈6	14	Average in class 4
5	Defined, entropy≈5	15	Average in class 5
6	Defined, entropy = 4	16	Average in class 6
7	Defined, entropy = 3	17	Average in class 7
8	Defined, entropy = 2	18	Average in class 8
9	Defined, entropy = 1	19	Average in class 9
10	Average in class 0	20	Average for all training data

ResNet50 [35], Xception [36], and EfficientNetB5 [37]. It is important to note that VGG16 and DenseNet201 give the error messages to the original inputs for MNIST and FMNIST; inputs must be resized to $32 \times 32 \times 3$. Resize is made by copying the grayscale to all RGB channels and applying zero-padding around the original images. Other models use the original sizes as inputs.

Table 5 provides the accuracy for MNIST; image entropy equalization improves accuracy on VGG16, DenseNet201, ResNet50, and EfficientNetB5. In contrast, the accuracy of Xception is not improved. According to statistic test, image entropy equalization with histogram from class 3 supplies meaningful accuracy improvement, while histogram with entropy equal to 8 underperforms the original images.

Table 6 shows the accuracy of the FMNIST dataset. Accuracy on VGG16, ResNet50 and EfficientNetB5 are improved for 0.6%, 1.0%, and 7.7%, respectively. Alternately, accuracy on DenseNet201 and Xception are degraded by 0.2% and 0.3%. According to statistic test, there were no significant improvement or degradation.

Table 7 provides the experiment result for the CIFAR10 dataset. Image entropy equalization improves VGG and ResNet50 by 9.9% and 11.9%, respectively. Otherwise, original data gives the best accuracy. Statistic result does not support significant improvement and the accuracies are rather degrading in several histograms.

Image entropy equalization with VGG16 gives the best accuracy for MNIST and FMNIST, as shown in **Tables 5 and 6**. However, the original data provides the best results for CIFAR10, as given in **Table 7**.

4.3.2. One-class classification results

An experiment for OCC was done by applying OCAE and OCITN (see [section 4.2](#)). The Wilcoxon signed-rank statistical test is executed in the same manner as a classification task, with the exception that the evaluation metric used is the AUC score. Null hypothesis assumes that entropy equalized images do not outperform the original images in terms of AUC.

4.3.2.1. OCAE.

Table 8 provides the AE structure for MNIST and FMNIST.

Table 9 shows the AE structure for CIFAR10.

Table 10 provides the AUC for OCAE on MNIST. Image entropy equalization improves the AUC score for nine seen classes. The best histogram is the average of class 8, which outperforms the original images by 7.84 points, which is a significant improvement. On the other hand, the worst histogram is the average of class 1, degrading the average AUC by 17.97 points. The digit “1” has the smallest white area in all handwritten digit data; editing other digits to such a histogram is challenging.

Table 11 shows the AUC for OCAE on FMNIST. Equalized images show the better AUC on eight seen classes. The best histogram is the average of class 4, which outperforms the original data by 2.5% AUC. According to the statistic test, the improvement is significant. On the other hand, the histograms of classes 1, 5, and 7 provide low AUC scores.

Table 12 provides the AUC score for OCAE on CIFAR10. Equalized images improve AUC scores for six seen classes. However, statistical difference is not given in the average AUC scores. The results for CIFAR10 are either enhanced or degraded significantly. These results suggest that some seen classes get advantages from image entropy differences, whereas degraded seen classes have relatively low entropy compared to improved ones (see **Table 2**).

Tables 10 and 11 demonstrate that image entropy equalization gives significant AUC improvement on MNIST and FMNIST. Therefore, AE seems to be biased due to the image entropy differences. However, the average AUC on CIFAR10 is degraded, as shown in **Table 12**. The reason is considered background; CIFAR10 has various backgrounds, while MNIST and FMNIST have black ones.

4.3.2.2. OCITn.

Table 13 provides ITN architecture, which consists of only convolutional layers; zero paddings are applied to keep the image size. M, N, and C correspond to the shape of input images. In addition, Lenna, a popular image for computer science, is used as a goal image.

Table 14 shows AUC for OCITN on MNIST. The histograms from the classes’ average provide better results than the defined histograms. Equalized images improved the AUC scores for six seen classes. However, the average AUC score is degraded from the original. According to statistic test, image entropy equalization degrades AUC scores in several histogram.

Table 15 provides AUC for OCITN on FMNIST. Equalized images improve four seen classes. However, the original image is the best in terms of average AUC. In addition, the histograms from the classes’ average outperform the defined histograms.

Table 16 shows AUC scores for OCITN on CIFAR10. Image entropy equalization improves AUC for four seen classes and degrades

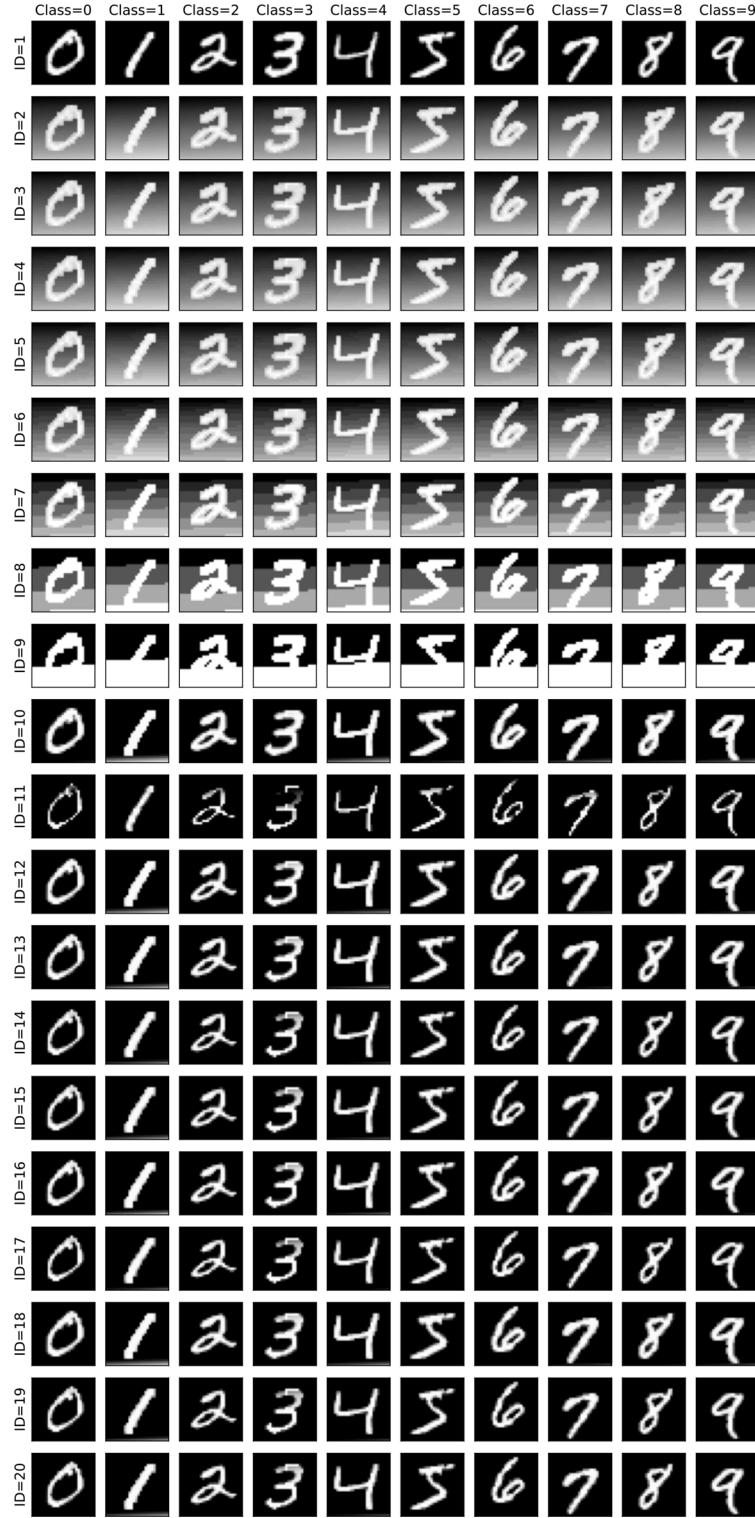


Fig. 1. Output of image entropy equalization (MNIST).

the average AUC score. Whereas improved seen classes have relatively medium image entropy (see Table 2), however, the AUC scores decreased for high (CAR and TRUCK) and low entropy seen classes (PLANE, BIRD, DEER, and FROG).

Tables 14 and 16 show that image entropy equalization improved AUC scores on several seen classes. However, the average AUC



Fig. 2. Output of image entropy equalization (FMNIST).

scores are degraded in all datasets.

Overall, image entropy equalization does not improve OCITN. Such a method is not affected by bias due to image entropy differences.

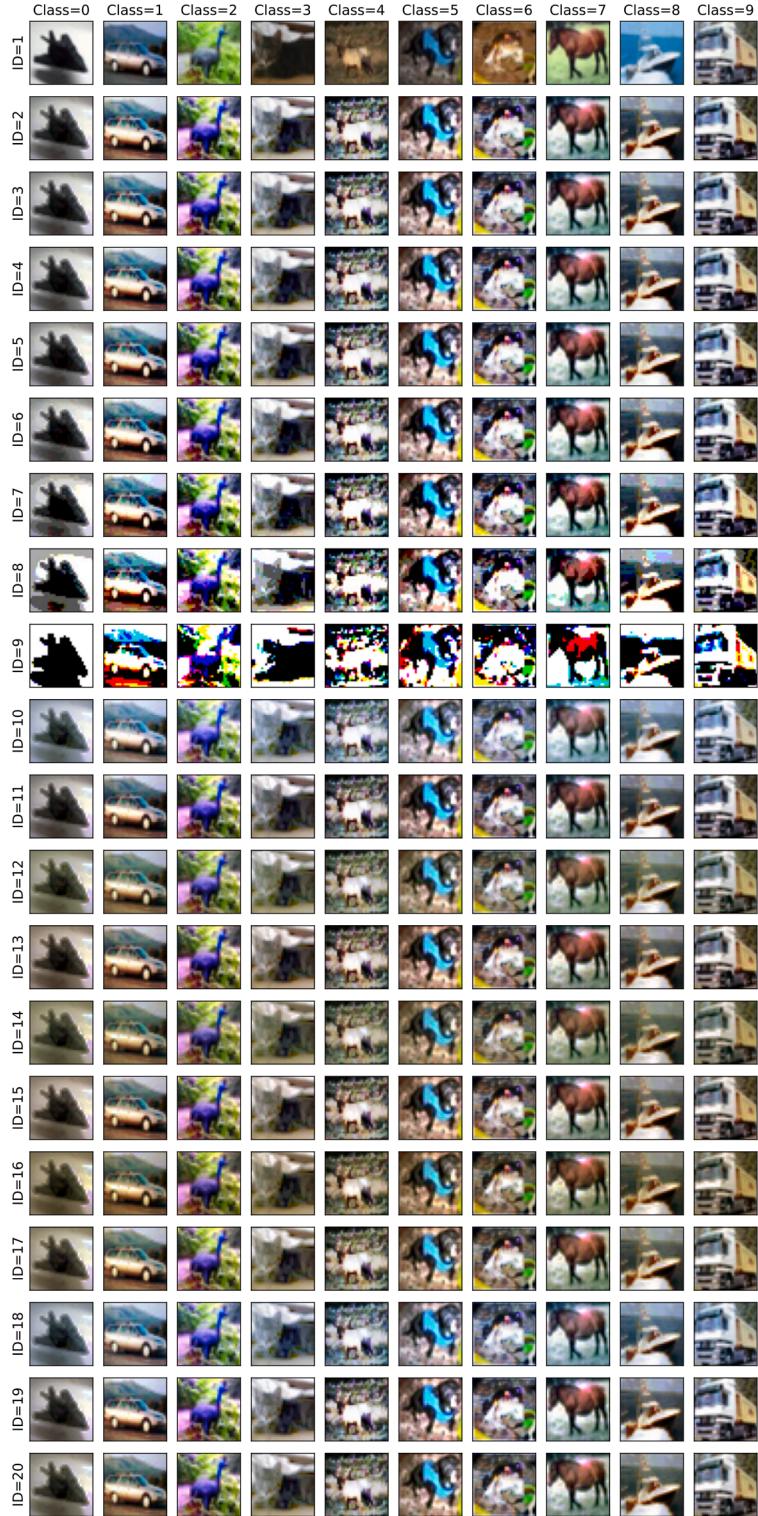


Fig. 3. Output of image entropy equalization (CIFAR10).

4.3.3. Reconstruction and image transformation

This subsection analyzes the relations between image entropy and model errors for AE and ITN. The structures of these networks are the same as in [section 4.3.2](#).

Table 5

Classification accuracy for MNIST.

Histogram	VGG16	DenseNet201	ResNet50	Xception	EfficientNetB5	P-value
Original	97.8 ± 2.1	88.0 ± 13.4	97.0 ± 1.5	98.6 ± 0.6	98.2 ± 0.5	None
H≈8	80.8 ± 35.3	79.5 ± 11.9	91.0 ± 11.7	95.6 ± 2.9	98.0 ± 0.3	1.0000
H≈7	98.6 ± 0.2	90.2 ± 5.9	96.0 ± 1.3	96.3 ± 2.5	98.0 ± 0.3	0.6875
H≈6	98.4 ± 0.2	92.1 ± 4.9	94.0 ± 2.8	97.4 ± 0.8	98.2 ± 0.4	0.5000
H≈5	98.3 ± 0.4	77.8 ± 26.1	79.4 ± 34.7	98.0 ± 0.4	98.0 ± 0.4	0.9375
H = 4	98.2 ± 0.3	86.4 ± 8.2	79.9 ± 21.3	95.8 ± 3.8	98.3 ± 0.3	0.9062
H = 3	98.3 ± 0.5	93.3 ± 3.0	87.0 ± 15.9	97.0 ± 1.0	97.0 ± 2.4	0.7812
H = 2	98.1 ± 0.2	92.7 ± 5.2	88.3 ± 13.5	97.4 ± 1.3	97.8 ± 0.4	0.7812
H = 1	96.3 ± 0.5	93.5 ± 1.7	95.2 ± 0.7	95.5 ± 0.7	95.4 ± 0.9	0.7812
Class = 0	98.6 ± 0.3	96.4 ± 2.4	91.0 ± 13.8	98.3 ± 0.8	98.5 ± 0.3	0.4062
Class = 1	98.0 ± 0.3	97.0 ± 0.6	95.2 ± 1.7	96.8 ± 1.3	97.5 ± 0.4	0.6875
Class = 2	98.4 ± 0.2	96.1 ± 2.7	95.0 ± 5.2	98.6 ± 0.4	98.6 ± 0.4	0.2326
Class = 3	98.6 ± 0.3	97.8 ± 1.0	97.2 ± 2.0	98.6 ± 0.2	98.6 ± 0.3	0.0339
Class = 4	98.2 ± 0.3	97.3 ± 2.5	85.5 ± 21.5	98.6 ± 0.3	98.4 ± 0.5	0.3575
Class = 5	98.8 ± 0.2	94.9 ± 4.6	92.1 ± 11.8	97.3 ± 2.0	98.3 ± 0.4	0.5000
Class = 6	98.6 ± 0.2	97.1 ± 2.5	97.5 ± 1.0	98.2 ± 0.6	98.3 ± 0.5	0.0938
Class = 7	98.5 ± 0.3	94.0 ± 7.1	96.8 ± 0.7	97.1 ± 1.0	98.5 ± 0.4	0.3125
Class = 8	98.6 ± 0.3	96.3 ± 2.2	97.0 ± 2.2	98.3 ± 0.5	98.6 ± 0.1	0.0721
Class = 9	98.5 ± 0.3	96.9 ± 0.7	95.4 ± 1.7	98.1 ± 1.2	98.4 ± 0.3	0.4062
Class = all	97.6 ± 1.7	97.8 ± 1.0	92.1 ± 11.3	96.6 ± 3.4	98.7 ± 0.3	0.5938

Table 6

Classification accuracy for FMNIST.

Histogram	VGG16	DenseNet201	ResNet50	Xception	EfficientNetB5	P-value
Original	90.1 ± 0.8	89.2 ± 0.6	85.6 ± 2.4	88.8 ± 0.8	79.1 ± 14.6	0.8438
H≈8	90.3 ± 0.7	80.8 ± 4.9	69.9 ± 18.1	86.8 ± 2.2	86.3 ± 0.5	0.7812
H≈7	90.3 ± 0.5	83.5 ± 5.1	73.1 ± 22.5	85.2 ± 1.9	86.4 ± 0.8	0.9062
H≈6	89.7 ± 1.1	82.0 ± 5.0	73.6 ± 17.2	85.0 ± 2.3	85.5 ± 1.2	0.6875
H≈5	90.3 ± 0.5	83.1 ± 4.9	78.3 ± 14.9	84.2 ± 5.0	86.6 ± 1.3	0.9375
H = 4	90.7 ± 0.3	77.7 ± 3.0	85.5 ± 1.8	87.7 ± 1.6	72.2 ± 29.2	0.9375
H = 3	89.6 ± 0.8	82.4 ± 4.7	66.9 ± 20.1	77.2 ± 17.8	85.3 ± 2.2	0.7812
H = 2	89.8 ± 0.2	84.4 ± 2.7	80.4 ± 8.0	81.6 ± 3.2	87.1 ± 0.9	0.8438
H = 1	87.6 ± 0.5	86.8 ± 0.7	70.1 ± 13.9	84.4 ± 1.3	84.3 ± 1.2	0.5938
Class = 0	90.5 ± 0.3	89.0 ± 0.4	84.8 ± 2.9	88.2 ± 1.4	86.0 ± 1.3	0.7812
Class = 1	89.0 ± 0.5	86.8 ± 1.2	81.2 ± 10.3	85.6 ± 2.3	83.8 ± 2.0	0.9062
Class = 2	90.6 ± 0.4	88.0 ± 1.8	81.9 ± 5.4	88.5 ± 0.9	78.9 ± 15.6	0.8438
Class = 3	88.9 ± 1.1	86.5 ± 2.5	71.7 ± 28.7	86.5 ± 0.6	86.8 ± 1.8	0.7812
Class = 4	58.1 ± 39.2	83.7 ± 4.0	86.6 ± 1.5	87.8 ± 1.5	86.6 ± 0.7	0.7812
Class = 5	89.0 ± 0.5	87.1 ± 0.9	85.3 ± 1.6	85.9 ± 1.4	83.6 ± 1.1	0.6875
Class = 6	90.5 ± 0.5	88.7 ± 0.6	81.0 ± 6.7	87.9 ± 0.8	85.9 ± 2.4	0.7812
Class = 7	89.6 ± 0.4	87.2 ± 1.3	84.7 ± 1.3	85.4 ± 1.6	85.6 ± 0.6	0.7812
Class = 8	90.4 ± 0.5	87.5 ± 1.7	77.3 ± 18.6	87.3 ± 2.3	85.5 ± 1.0	0.7812
Class = 9	89.9 ± 0.5	86.9 ± 1.5	80.8 ± 6.6	86.0 ± 2.6	86.0 ± 1.4	0.5938
Class = all	89.4 ± 0.9	87.0 ± 1.4	87.5 ± 1.2	86.1 ± 2.5	86.4 ± 0.7	0.8438

Table 17 provides the reconstruction errors for original data and equalized histograms. According to the result, the original images supply the lowest reconstruction error for FMNIST and CIFAR10. In addition, equalization to low entropy increases the reconstruction error. This result is the opposite of the initial hypothesis [7] that low-entropy images are more accessible to reconstruct than high-entropy ones. For this reason, low entropy images are low-quality due to limited pixel values. (Histogram with H = 1 has only two values, 0 and 255. H = 2 has only four values, 0, 85, 170, and 255.)

It is important to note that this comparison is not fair because compared models and ideal outputs are different for all settings; fair comparison requires the same model (from the same training data) or the same ideal outputs (like class labels or a goal image). Since the objective of AE is to reconstruct input, such a comparison needs to apply different preprocessing methods to training and testing data. This kind of processing is not general in machine learning, but it should be considered to analyze the relation between image entropy and reconstruction error in future work.

Besides, Table 18 shows the comparison of construction errors for ITN. Experiment results show that higher entropy histograms provide smaller construction errors. Image entropy may be an ingredient for image transformation; low entropy images increase errors due to lacking an element.

In summary, low-entropy histograms increased the errors for reconstruction and image transformation. These tasks could require an image entropy as an ingredient to create an output.

Table 7

Classification accuracy for CIFAR10.

Data	VGG16	DenseNet201	ResNet50	Xception	EfficientNetB5	P-value
Original	59.5 ± 24.8	76.4 ± 1.9	58.2 ± 16.1	78.5 ± 1.2	81.8 ± 0.5	None
H = 24	57.5 ± 23.7	72.9 ± 2.2	68.8 ± 2.0	73.5 ± 2.5	78.6 ± 0.4	0.7812
H = 21	68.8 ± 0.8	70.0 ± 4.8	65.9 ± 5.7	74.6 ± 1.5	78.1 ± 0.4	0.4062
H = 18	68.9 ± 1.0	72.2 ± 1.8	70.1 ± 0.7	75.3 ± 1.7	78.3 ± 0.3	0.4062
H = 15	43.8 ± 27.7	70.1 ± 6.5	67.3 ± 5.2	74.4 ± 1.0	78.5 ± 0.3	0.8438
H = 12	45.0 ± 28.6	54.9 ± 14.0	62.2 ± 12.0	74.3 ± 0.5	77.7 ± 0.5	0.9688
H = 9	55.6 ± 22.9	71.6 ± 2.0	67.0 ± 2.8	74.3 ± 1.0	77.0 ± 0.5	0.7812
H = 6	63.3 ± 1.6	68.2 ± 2.6	37.1 ± 20.2	70.5 ± 1.0	73.6 ± 0.6	0.9688
H = 3	55.3 ± 0.9	58.1 ± 6.9	51.6 ± 8.9	60.3 ± 1.0	63.7 ± 0.1	1.0000
Class = 0	67.3 ± 0.6	68.3 ± 5.8	61.0 ± 9.2	75.3 ± 0.8	78.2 ± 0.4	0.7812
Class = 1	21.7 ± 23.5	54.4 ± 23.5	68.6 ± 3.8	75.7 ± 0.9	78.8 ± 0.4	0.9062
Class = 2	68.8 ± 1.2	70.1 ± 5.9	63.0 ± 4.6	73.2 ± 2.6	78.5 ± 0.7	0.5938
Class = 3	69.3 ± 1.2	72.8 ± 4.7	58.0 ± 11.2	75.6 ± 1.1	78.1 ± 0.3	0.7812
Class = 4	57.0 ± 23.5	51.9 ± 24.5	61.4 ± 11.7	74.7 ± 2.0	78.2 ± 0.8	0.9375
Class = 5	45.5 ± 29.0	59.2 ± 13.9	48.2 ± 18.1	75.2 ± 1.0	78.7 ± 0.5	1.0000
Class = 6	67.8 ± 4.4	72.5 ± 4.8	55.9 ± 9.8	74.3 ± 1.0	78.9 ± 0.5	0.7812
Class = 7	55.8 ± 22.9	67.8 ± 8.4	67.8 ± 5.2	74.1 ± 1.2	78.2 ± 0.5	0.7812
Class = 8	33.3 ± 28.6	69.4 ± 7.3	64.4 ± 4.8	75.5 ± 0.3	78.5 ± 0.6	0.9062
Class = 9	55.6 ± 22.8	63.1 ± 19.7	65.3 ± 6.1	74.5 ± 1.1	78.4 ± 0.6	0.8438
Class = all	69.4 ± 1.1	69.6 ± 8.0	57.1 ± 16.3	75.1 ± 0.7	78.2 ± 0.3	0.7812

Table 8

AE architecture for MNIST and FFMNIST.

Encoder	Decoder
Conv2D (8, (5,5), "Relu")	Reshape ((4,4,2))
Maxpooling2D(size = 2)	UpSampling2D(size = 2)
Conv2D(4, (5,5),"Relu")	Conv2DTranspose(4, (5,5),"Relu")
Maxpooling2D(size = 2)	UpSampling2D(size = 2)
Flatten ()	Conv2DTranspose(8, (5,5), "Relu")
Dense (32, "Relu")	Conv2DTranspose(1, (1,1)), "Relu")

Table 9

AE architecture for CIFAR10.

Encoder	Decoder
Conv2D(32,(5,5),"Relu")	Reshape((4,4,8)))
Conv2D(64, (5,5),"Relu")	UpSampling2D(size = 2))
MaxPooling2D(size = 2))	Conv2DTranspose(128, (5,5),"Relu")
Conv2D(128, (5,5),"Relu")	UpSampling2D(size = 2))
MaxPooling2D(size = 2)	Conv2DTranspose(64, (5,5),"Relu")
Flatten()	Conv2DTranspose(32, (5,5),"Relu")
Dense(128, "Relu")	Conv2DTranspose(3, (1,1)), "Relu")

5. Discussions

5.1. Advantages and limitations

Image entropy equalization improved OCAE by eliminating differences in image entropy. It is possible that AE is influenced by bias arising from image entropy differences, and this bias was reduced through entropy equalization. Therefore, image entropy equalization has the potential to enhance various studies using AE. It is important to note that the applications of AE can be categorized as follows: 1) Using reconstruction error for OCC, anomaly detection, or other related methods, and 2) Using compression for representation learning. The experimental results demonstrate that image entropy equalization improves the first category of applications, while its effectiveness for the second category is still unknown.

On the other hand, image entropy equalization has several limitations. The first question is whether image entropy differences are always problematic. Entropy differences could be essential features for classification or other tasks, and removing such features might degrade accuracy. Further studies are needed to clarify the situations in which entropy equalization is successful.

Additionally, the experiment results for the CIFAR10 dataset did not show improvement with image entropy equalization. One potential reason is that the color information is lost by editing RGB images in channel by channel.

One possible solution to minimize color information loss is to reduce the difference between input and output images. To achieve this, a promising approach is to use multiple histograms with the same entropy and edit all images to match the most similar histogram.

Table 10
AUC score for OCAE on MNIST.

Histogram	Seen class										Average	P-value
	0	1	2	3	4	5	6	7	8	9		
Original	98.7	99.9	75.7	87.0	83.4	87.6	94.9	85.8	80.7	92.9	88.66	None
H≈8	97.5	99.4	82.6	87.1	89.5	84.7	91.6	93.5	83.5	92.6	90.20	0.3125
H≈7	97.6	99.6	82.6	87.5	90.5	84.9	91.2	94.7	85.1	92.5	90.62	0.1875
H≈6	98.6	99.5	82.3	86.9	89.3	84.9	93.3	93.9	82.9	92.6	90.42	0.3125
H≈5	98.5	99.7	82.6	86.6	90.2	84.7	93.2	94.2	82.6	93.2	90.55	0.2158
H = 4	98.6	99.6	80.5	88.3	88.9	87.3	93.3	94.7	84.8	93.8	90.98	0.0654
H = 3	98.1	99.5	86.4	87.8	89.5	86.6	84.3	94.3	85.5	92.1	90.41	0.2783
H = 2	88.8	99.1	86.8	88.5	87.4	75.2	89.8	93.5	86.9	91.5	88.75	0.4609
H = 1	93.9	99.8	78.9	84.3	86.3	87.5	88.2	92.9	78.7	92.4	88.29	0.6875
class = 0	91.4	99.8	95.1	95.1	95.6	92.8	98.4	96.5	82.3	96.1	94.31	0.0186
class = 1	61.8	77.4	55.7	66.5	86.2	63.9	68.7	86.5	55.2	85.0	70.69	0.9971
class = 2	99.3	99.9	95.8	95.6	96.4	93.5	98.6	96.9	92.1	96.4	96.45	0.0038
class = 3	99.3	99.8	95.0	86.6	95.8	94.9	98.4	96.9	91.7	97.0	95.54	0.0049
class = 4	89.8	99.9	85.3	94.6	96.7	85.8	97.5	97.6	88.3	96.4	93.19	0.0330
class = 5	79.4	99.8	68.3	86.4	96.1	93.3	98.2	97.0	82.8	96.6	89.79	0.2461
class = 6	90.2	99.8	93.9	77.4	69.0	93.8	98.3	97.5	83.1	97.2	90.02	0.3477
class = 7	79.0	99.8	59.6	85.5	87.4	67.9	98.1	97.4	88.4	87.4	85.05	0.7842
class = 8	99.4	99.9	94.6	95.6	96.1	94.1	98.1	97.3	93.1	96.8	96.50	0.0038
class = 9	81.5	99.9	75.9	86.3	96.4	67.9	98.1	97.3	89.1	96.5	88.89	0.3392
class = all	90.5	99.9	93.8	77.8	96.7	84.4	88.9	87.6	82.8	97.0	89.94	0.4295

Table 11
AUC scores for OCAE on FMNIST.

Histogram	Seen class										Average	P-value
	0	1	2	3	4	5	6	7	8	9		
Original	88.6	97.5	83.1	84.6	85.1	88.9	76.3	98.0	79.1	93.2	87.44	None
H≈8	87.0	98.0	87.9	91.7	88.1	85.3	74.7	98.8	77.5	95.9	88.49	0.2461
H≈7	87.3	98.0	87.5	91.2	87.5	85.8	74.6	98.8	78.9	96.2	88.58	0.1875
H≈6	88.1	98.1	88.2	91.2	86.9	85.7	74.3	98.8	79.9	96.2	88.74	0.1162
H≈5	86.1	97.8	87.9	91.6	87.2	85.7	75.1	98.8	76.8	95.9	88.29	0.3125
H = 4	88.4	98.1	87.6	91.4	87.5	85.6	74.4	98.8	78.8	95.8	88.64	0.1377
H = 3	87.3	98.0	87.8	91.1	88.2	85.3	75.7	98.9	80.7	96.3	88.93	0.0967
H = 2	88.6	97.9	88.1	90.2	89.2	86.4	75.5	98.7	84.0	96.1	89.47	0.0332
H = 1	90.7	97.8	80.1	89.1	80.3	83.8	69.6	98.6	82.9	98.4	87.13	0.5771
class = 0	90.1	98.0	82.6	92.6	84.4	85.0	70.6	98.9	86.7	97.9	88.68	0.2158
class = 1	72.8	98.2	67.0	90.7	58.1	83.4	51.3	98.8	71.4	91.1	78.28	0.9814
class = 2	91.1	98.0	84.7	91.5	88.4	85.2	74.7	98.9	87.5	97.0	89.70	0.0420
class = 3	79.1	98.1	68.9	92.7	65.5	83.8	55.5	98.6	75.4	95.4	81.30	0.9473
class = 4	92.2	98.0	84.0	91.8	87.7	86.6	74.0	98.8	88.7	97.6	89.94	0.0322
class = 5	74.5	98.0	84.4	87.5	69.7	86.9	59.0	98.6	69.6	85.7	81.39	0.9580
class = 6	90.5	98.0	84.2	91.8	86.2	85.7	73.7	98.9	90.0	97.1	89.61	0.0801
class = 7	72.7	98.0	71.3	88.9	66.0	86.2	52.0	98.8	70.5	87.9	79.23	0.9863
class = 8	90.2	98.1	82.1	91.8	86.1	85.3	71.7	98.9	87.1	98.0	88.93	0.1611
class = 9	82.5	98.1	75.0	93.0	74.1	83.0	60.1	98.6	79.8	97.7	84.19	0.8389
class = all	84.3	98.3	75.1	84.4	74.5	83.9	62.6	99.0	80.6	97.6	84.03	0.9033

The main challenge lies in preparing identical entropy histograms and developing a fast computing process to identify the most similar histogram for each image. Another option is to consider all color channels together. One potential idea is to employ image entropy equalization using a three-dimensional histogram, where each dimension corresponds to the RGB channels. However, this idea is still at a conceptual stage, and further discussion is required for future work.

Another issue is that image entropy equalization is influenced by diverse backgrounds since the equalization process uses only color information and does not take into account the foreground and background. This aspect poses a challenge specifically for the CIFAR10 dataset, which consists of images with more varied backgrounds, whereas datasets like MNIST or FMNIST contain images with a consistent black background.

Additionally, the reference histogram is an important parameter for image entropy equalization. While the optimal histogram is not known, the worst histogram has zero entropy, where all pixels have the same value (which is evident without conducting experiments). In this case, classification becomes impossible as all images are edited to have a single color.

It is important to note that this study has limitations in that the experiment results do not include the state-of-the-art (SOTA). One possible solution is to combine image entropy equalization with SOTA. However, achieving this idea is not easy because SOTA requires extensive resources, while our experiment environment has only a single GPU and 64 GB of memory space. The experiments with large-

Table 12

AUC scores for OCAE on CIFAR10.

Histogram	Seen class										Average	P-value
	PLANE	CAR	BIRD	CAT	DEER	DOG	FROG	HORSE	SHIP	TRUCK		
Original	70.0	43.0	66.6	55.5	71.2	55.6	61.5	48.9	72.9	42.4	58.76	None
H = 24	63.6	54.3	44.7	58.2	41.4	61.4	37.0	52.7	77.0	58.8	54.91	0.6875
H = 21	63.8	54.6	44.9	58.1	41.1	61.2	36.8	52.5	76.4	58.3	54.77	0.6875
H = 18	63.9	54.1	44.9	58.6	41.5	61.2	37.1	52.9	76.5	59.0	54.97	0.6875
H = 15	63.6	54.2	44.8	58.3	41.2	61.5	37.0	52.6	76.6	58.4	54.82	0.6875
H = 12	63.5	54.2	44.9	58.5	40.9	60.8	36.7	52.8	76.9	58.6	54.78	0.6875
H = 9	63.7	53.9	44.8	58.2	41.2	61.7	36.9	53.1	77.0	58.1	54.86	0.6875
H = 6	63.5	53.4	44.4	58.1	41.0	61.4	35.6	52.6	76.8	59.3	54.61	0.6875
H = 3	62.4	51.2	45.3	55.3	42.2	58.6	39.5	50.2	77.6	57.8	54.01	0.7217
class = 0	64.2	54.4	45.1	58.1	41.4	61.4	36.6	52.7	76.6	57.9	54.84	0.6875
class = 1	63.0	54.7	44.0	58.2	41.1	61.9	36.8	53.6	75.2	60.0	54.85	0.6875
class = 2	63.6	55.5	44.2	58.4	40.8	62.1	36.5	53.7	75.8	60.0	55.06	0.6523
class = 3	63.2	54.7	44.4	58.3	41.2	61.8	36.9	53.2	75.8	59.8	54.93	0.6875
class = 4	62.6	55.2	44.5	58.6	41.5	62.2	37.8	53.5	75.8	59.6	55.13	0.6875
class = 5	63.8	54.8	44.4	58.6	41.1	61.4	36.9	53.3	76.1	59.7	55.01	0.6875
class = 6	63.1	55.6	44.5	58.7	41.2	61.5	37.0	53.7	75.5	60.3	55.11	0.6875
class = 7	63.1	55.6	44.5	57.9	40.1	61.5	36.8	53.4	75.9	60.3	54.91	0.6875
class = 8	63.8	55.1	44.7	58.6	41.7	61.7	36.8	52.9	76.4	58.0	54.97	0.6875
class = 9	63.0	54.5	44.1	58.2	40.7	61.1	36.8	53.0	75.8	60.2	54.74	0.6875
class = all	62.9	55.1	44.4	58.4	40.9	61.7	36.9	53.3	75.6	59.9	54.91	0.6875

Table 13

ITN architecture.

Layer	Output Shape	Conv size	Activation	Padding
1. Conv2D	(None, M, N, 32)	(3, 3)	relu	zero-padding
2. Conv2D	(None, M, N, 32)	(3, 3)	relu	zero-padding
3. Conv2D	(None, M, N, 32)	(3, 3)	relu	zero-padding
4. Conv2D	(None, M, N, 32)	(3, 3)	relu	zero-padding
5. Conv2D	(None, M, N, 32)	(3, 3)	relu	zero-padding
6. Conv2D	(None, M, N, 32)	(3, 3)	relu	zero-padding
7. Conv2D	(None, M, N, 32)	(3, 3)	relu	zero-padding
8. Conv2D	(None, M, N, C)	(3, 3)	relu	zero-padding

Table 14

AUC scores for OCITN on MNIST.

Histogram	Seen class										Average	P-value
	0	1	2	3	4	5	6	7	8	9		
Original	99.5	99.7	97.1	94.6	95.4	98.3	99.4	94.7	96.7	97.5	97.29	None
H≈8	99.4	99.8	90.4	90.3	91.9	91.0	99.0	91.8	91.5	96.3	94.14	0.9990
H≈7	99.2	99.7	90.8	91.1	93.5	90.4	98.6	93.1	92.9	96.4	94.57	0.9962
H≈6	99.2	99.7	87.9	89.9	92.8	90.7	98.7	93.2	92.6	96.0	94.07	0.9962
H≈5	99.0	99.7	87.9	89.2	92.0	89.8	98.0	92.7	91.9	96.0	93.62	0.9962
H = 4	99.1	99.7	86.1	89.8	92.1	87.3	97.8	92.8	92.5	95.4	93.26	0.9962
H = 3	98.7	99.6	80.5	87.3	92.5	88.3	97.7	92.4	92.0	95.6	92.46	1.0000
H = 2	98.7	99.7	88.4	91.0	92.0	88.8	97.8	92.3	92.6	94.9	93.62	0.9962
H = 1	99.3	99.6	84.6	89.5	91.5	90.0	98.1	93.7	91.7	94.8	93.28	1.0000
class = 0	99.5	99.8	96.0	92.0	94.3	97.2	99.5	94.1	94.3	96.6	96.33	0.9897
class = 1	99.4	99.7	96.1	93.6	95.8	96.8	99.1	94.4	92.6	96.7	96.42	0.9859
class = 2	99.5	99.8	96.2	94.0	95.3	97.3	99.5	94.3	94.7	96.9	96.75	0.9897
class = 3	99.5	99.8	96.6	94.0	95.3	97.2	99.5	94.1	95.4	96.8	96.82	0.9896
class = 4	99.5	99.8	96.8	93.5	95.6	97.3	99.5	95.0	94.5	97.2	96.87	0.9144
class = 5	99.5	99.8	96.7	94.0	95.6	97.3	99.4	94.5	94.9	97.0	96.87	0.9822
class = 6	99.4	99.8	96.9	93.8	95.5	97.9	99.5	94.4	94.7	96.7	96.86	0.9863
class = 7	99.4	99.8	97.3	94.4	95.9	97.6	99.4	94.5	94.3	96.7	96.93	0.8936
class = 8	99.5	99.8	96.5	93.2	94.6	96.9	99.5	94.4	94.6	96.8	96.58	0.9897
class = 9	99.6	99.8	97.0	94.3	95.8	97.0	99.4	94.4	94.9	97.0	96.92	0.9315
class = all	99.5	99.8	96.4	93.6	95.5	97.1	99.5	94.3	95.2	97.2	96.81	0.9751

Table 15

AUC scores for OCITN on FMNIST.

Histogram	Seen class										Average	P-value
	0	1	2	3	4	5	6	7	8	9		
Original	95.6	99.1	93.8	96.4	93.6	95.5	85.7	98.9	97.9	98.9	95.54	None
H≈8	93.4	98.6	91.3	94.7	92.7	95.4	81.8	98.5	94.3	98.8	93.95	1.0000
H≈7	93.5	98.8	92.0	94.8	92.1	95.1	82.2	98.5	94.4	98.8	94.02	1.0000
H≈6	93.5	98.8	91.8	94.9	92.1	95.8	82.1	98.5	94.0	98.7	94.02	0.9980
H≈5	93.3	98.8	91.8	94.9	92.1	93.8	82.2	98.5	93.7	98.6	93.77	1.0000
H = 4	93.7	98.9	92.1	95.1	92.2	94.0	82.3	98.3	93.8	98.3	93.87	1.0000
H = 3	93.8	98.6	91.6	94.7	92.3	92.9	82.2	98.5	93.6	98.2	93.64	1.0000
H = 2	94.5	98.5	92.2	94.0	92.4	94.3	83.5	98.6	95.6	98.8	94.24	1.0000
H = 1	95.5	98.5	92.1	95.9	93.5	95.7	83.5	98.8	97.3	99.3	95.01	0.9756
class = 0	95.2	98.8	93.0	95.8	93.9	96.3	84.3	98.7	97.8	98.8	95.26	0.9346
class = 1	94.7	99.0	92.4	96.7	93.3	95.8	83.8	98.7	95.7	98.3	94.84	0.9814
class = 2	94.8	98.8	92.9	95.5	93.7	96.3	84.3	98.6	97.3	98.8	95.10	0.9814
class = 3	94.8	99.0	92.8	96.5	93.7	95.9	83.9	98.6	96.4	98.5	95.01	0.9678
class = 4	95.2	98.7	92.9	95.7	93.7	96.1	84.1	98.7	97.6	98.9	95.16	0.9670
class = 5	94.2	99.0	92.1	96.1	93.4	96.6	83.7	98.8	95.6	98.4	94.79	0.9902
class = 6	95.1	98.8	92.3	95.7	93.8	96.5	84.1	98.8	97.4	98.8	95.13	0.958
class = 7	94.6	99.0	92.2	96.4	93.4	95.8	83.8	98.8	95.7	98.4	94.81	0.9858
class = 8	95.4	98.8	92.4	95.7	94.1	96.4	84.7	98.7	97.5	98.8	95.25	0.9199
class = 9	95.0	98.9	93.0	96.4	94.0	95.7	84.4	98.7	96.9	98.8	95.18	0.9515
class = all	94.9	98.9	92.6	96.2	94.0	95.5	84.4	98.7	97.5	98.7	95.14	0.9785

Table 16

AUC scores for OCITN on CIFAR10.

Histogram	Seen class										Average	P-value
	PLANE	CAR	BIRD	CAT	DEER	DOG	FROG	HORSE	SHIP	TRUCK		
Original	78.2	87.2	67.4	58.2	70.4	71.0	76.7	79.2	82.6	84.2	75.50	None
H = 24	74.6	86.6	63.2	60.2	68.1	71.1	76.4	80.7	82.8	83.6	74.73	0.8838
H = 21	75.2	86.3	63.3	60.0	68.2	70.8	75.3	80.0	83.4	83.6	74.61	0.9199
H = 18	74.9	86.8	63.8	59.9	67.5	70.6	76.3	79.6	83.2	83.5	74.61	0.9346
H = 15	75.0	86.1	62.9	61.8	67.6	71.3	76.4	79.5	82.9	83.7	74.72	0.8623
H = 12	74.0	86.2	62.8	60.4	67.6	70.7	76.5	78.8	83.0	83.4	74.34	0.9678
H = 9	74.6	86.1	62.2	59.9	67.4	68.3	75.7	79.6	81.7	83.0	73.85	0.9863
H = 6	73.1	83.6	62.0	60.7	66.2	69.4	75.9	79.0	80.5	82.7	73.31	0.9902
H = 3	70.7	79.6	57.3	61.8	63.0	65.1	74.7	74.7	80.4	79.1	70.64	0.9971
class = 0	74.6	87.0	63.8	60.4	67.6	71.9	76.9	79.1	83.4	83.8	74.85	0.7842
class = 1	74.4	86.7	62.5	60.8	67.8	72.3	75.9	79.3	82.6	83.6	74.59	0.8932
class = 2	74.3	86.0	63.6	61.9	67.6	70.4	75.6	79.2	82.5	83.5	74.46	0.9668
class = 3	74.1	86.6	63.5	59.7	68.0	69.9	75.5	79.6	82.6	84.2	74.37	0.9536
class = 4	73.8	86.4	64.1	60.0	67.6	71.5	75.7	80.0	82.8	83.4	74.53	0.9199
class = 5	74.5	86.7	64.0	61.8	66.4	71.0	76.3	79.8	82.3	83.2	74.60	0.9135
class = 6	74.2	86.4	62.8	59.3	67.5	73.0	76.0	79.6	81.9	83.7	74.44	0.9199
class = 7	73.2	86.4	63.9	59.7	68.0	69.8	75.5	79.3	82.3	83.7	74.18	0.9814
class = 8	74.0	86.9	63.4	61.3	67.9	70.4	76.1	80.3	83.2	83.3	74.68	0.8623
class = 9	74.4	87.0	62.9	61.0	68.7	70.0	75.7	79.8	82.9	83.7	74.61	0.9199
class = all	73.9	87.1	63.3	60.6	67.0	70.0	74.9	79.3	82.8	83.6	74.25	0.9580

scale datasets are not included for the same reason. Moreover, investigated DL parameters are limited, and different parameters might change the results. Further studies are needed to address these challenges.

In summary, image entropy equalization is a promising preprocessing method for AE. Although it does not show many advantages over other tasks, image entropy equalization could contribute to broad areas because AE is applied in various fields. Moreover, addressing current limitations could enhance the effectiveness of image entropy equalization.

5.2. Processing time

Image entropy equalization takes extra processing time. Table 19 shows the processing speed for image editing (algorithm), which is computed in the following characteristics: CPU Intel(R) Core (TM) i9-9900 K CPU @ 3.60 GHz, RAM 64 GB. It is important to note that creating histograms takes another time (especially, computing the average of histograms takes time), but the editing speed in the testing stage is unaffected.

This speed is acceptable for real-time processing. However, the concern is scalability for processing larger size of images. In such a case, applying arbitrary image equalization models should be faster than image editing. On the other hand, using the model to achieve

Table 17
Comparison of reconstruction error for AE.

Histogram	MNIST	FMNIST	CIFAR10
Original	17.4 ± 8.2	21.2 ± 1.3	21.4 ± 0.9
$H \approx 8$ or $H = 24$	15.9 ± 1.0	26.1 ± 0.9	31.4 ± 0.8
$H \approx 7$ or $H = 21$	16.0 ± 0.6	26.4 ± 1.3	31.1 ± 1.2
$H \approx 6$ or $H = 18$	15.6 ± 1.6	25.2 ± 0.6	32.4 ± 0.9
$H \approx 5$ or $H = 15$	16.1 ± 1.4	26.5 ± 0.4	34.9 ± 1.5
$H = 4$ or $H = 12$	16.4 ± 1.0	27.1 ± 1.1	34.0 ± 1.3
$H = 3$ or $H = 9$	19.9 ± 1.0	29.7 ± 1.0	36.4 ± 2.2
$H = 2$ or $H = 6$	22.3 ± 1.7	33.7 ± 1.6	41.1 ± 1.9
$H = 1$ or $H = 3$	21.3 ± 2.8	34.4 ± 0.8	50.3 ± 12.6
class = 0	16.8 ± 1.4	31.2 ± 1.1	28.4 ± 1.6
class = 1	14.9 ± 3.7	31.2 ± 0.5	29.2 ± 1.1
class = 2	14.7 ± 1.5	30.0 ± 0.5	26.8 ± 1.1
class = 3	15.4 ± 1.3	38.5 ± 13.8	28.0 ± 1.4
class = 4	13.9 ± 1.5	29.7 ± 1.4	23.2 ± 0.7
class = 5	14.9 ± 0.6	23.9 ± 0.3	28.0 ± 0.4
class = 6	18.7 ± 8.2	29.5 ± 0.9	25.3 ± 1.3
class = 7	13.0 ± 1.5	30.4 ± 6.2	27.3 ± 1.7
class = 8	18.9 ± 9.8	30.6 ± 1.5	28.5 ± 1.4
class = 9	12.3 ± 1.1	31.3 ± 1.2	30.9 ± 1.3
class = all	18.1 ± 7.2	31.7 ± 1.8	28.1 ± 1.7

Table 18
Comparison of construction error for ITN.

Histogram	MNIST	FMNIST	CIFAR10
Original	11.4 ± 0.2	10.7 ± 0.2	19.2 ± 0.1
$H \approx 8$ or $H = 24$	5.3 ± 0.2	6.8 ± 0.4	19.2 ± 0.1
$H \approx 7$ or $H = 21$	5.1 ± 0.4	6.7 ± 0.3	19.2 ± 0.1
$H \approx 6$ or $H = 18$	5.6 ± 0.2	6.6 ± 0.1	19.3 ± 0.1
$H \approx 5$ or $H = 15$	5.5 ± 0.3	6.8 ± 0.1	19.4 ± 0.2
$H = 4$ or $H = 12$	5.3 ± 0.3	6.7 ± 0.2	19.3 ± 0.1
$H = 3$ or $H = 9$	5.8 ± 0.2	7.3 ± 0.2	19.4 ± 0.2
$H = 2$ or $H = 6$	7.8 ± 0.2	9.0 ± 0.2	19.6 ± 0.1
$H = 1$ or $H = 3$	10.4 ± 0.2	11.7 ± 0.2	20.4 ± 0.1
class = 0	10.2 ± 0.3	10.0 ± 0.4	19.4 ± 0.2
class = 1	12.0 ± 0.4	11.5 ± 0.2	19.2 ± 0.1
class = 2	10.6 ± 0.3	9.4 ± 0.3	19.2 ± 0.1
class = 3	10.5 ± 0.1	11.2 ± 0.3	19.1 ± 0.1
class = 4	10.7 ± 0.2	10.1 ± 0.5	19.2 ± 0.1
class = 5	10.8 ± 0.3	11.7 ± 0.5	19.2 ± 0.1
class = 6	10.6 ± 0.1	9.6 ± 0.3	19.3 ± 0.1
class = 7	11.0 ± 0.2	11.6 ± 0.1	19.2 ± 0.1
class = 8	10.4 ± 0.2	10.0 ± 0.2	19.3 ± 0.1
class = 9	10.9 ± 0.1	10.8 ± 0.3	19.2 ± 0.1
class = all	10.6 ± 0.1	10.6 ± 0.2	19.3 ± 0.1

Table 19
Image editing speed.

Image shape	Editing speed (images/s.).
$28 \times 28 \times 1$	2336.4
$32 \times 32 \times 3$	487.8

perfect entropy equalization is challenging. Future algorithms and hardware developments could address this challenge.

5.3. Histogram

Histogram is a parameter for image entropy equalization; this study compared 19 histograms. Experiment results (see Section 4) show that histograms with average training data outperform equally distributed ones. The reason might be the distance from the original images; Table 20 shows the mean absolute errors between the original images and edited images. Generally, histograms created by average training data provide smaller editing errors. In the low entropy histograms (MNIST and FMNIST with $H = 1$, CIFAR10 with $H = 3$), image entropy equalization creates significantly different outputs from inputs. These outputs might destroy

essential features for classification and degrade accuracy.

Still, the number of experimented histograms is limited; Further experiments are necessary to find better histograms. Additionally, editing into a single histogram has a limitation for minimizing the distance from the original image. One possible solution is to use multiple histograms for equalization. Shuffling an arbitrary histogram can create different histograms with the same entropy. Moreover, editing images into the nearest histogram can reduce the editing distance.

5.4. Future directions

Future research directions are categorized into three groups: 1) Improving image entropy equalization, 2) extending entropy equalization to other tasks and data types, and 3) applying image entropy equalization to real application or other usages.

The first group is focused on improving image entropy equalization. In the future, new methods will be proposed as image entropy equalization algorithms and will be compared to the approach presented in this paper. The comparison metrics should include processing speed, the distance between the original and output images, the variance of image entropy (in cases where image entropy equalization is achieved through machine learning models), and accuracies/losses (or related metrics) for image recognition tasks.

For instance, editing images into multiple histograms with the same entropy could reduce the differences between the inputs and outputs of equalization process. Considering other image entropy equalization methods that do not involve image editing, could improve the processing speed, albeit with a trade-off in entropy variance. Additionally, combining the method with foreground/background separation [17] could address the issue of diverse backgrounds in images.

In addition, there are several possible extensions that were not covered in this study. On extension involves considering other image entropy metrics [39]. Additionally, conducting more experiments for further image processing tasks, such as representation learning [5], GAN [25], and object tracking [38] would provide further insights. Another extension pertains to applying entropy equalization to other data types. However, this poses challenges in terms of computing entropy and editing other data types, as their formats differ from that of images.

Furthermore, the current image entropy equalization method could be applicable to problems where the background does not exhibit diversity, or cases where there is no foreground and background. Specifically, potential applications could include handwriting analysis, thermal image analysis, and the analysis of short-time Fourier transform for time series data.

Finally, image entropy equalization can be applied to various purposes. For instance, equalized images can be used for data augmentation [19]. Classification or transformation between the original and equalized images holds promise as pretext tasks for OCC. It would be interesting to explore the effects of applying different preprocessing methods to training and testing data (ex. training with high entropy and testing with low entropy), to determine if it improves model accuracy or reduces errors. Using image entropy equalization could attack pretrained image recognition models.

6. Conclusion and future work

This paper introduces image entropy equalization as a preprocessing method for image recognition tasks, aiming to address bias caused by differences in image entropy. The proposed method is evaluated by comparing original images with entropy equalized images across four machine learning tasks: supervised classification, one-class classification, autoencoder, and image transformation. The results show that image entropy equalization led to improved AUC scores for OCAC on the MNIST and FMNIST datasets, suggesting that image entropy differences affect AE. However, image entropy equalization did not improve classification accuracy. These findings

Table 20
Mean absolute error from the original images.

Histogram	MNIST	FMNIST	CIFAR10
Original	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
$H \approx 8$ or $H = 24$	93.4 ± 9.3	60.2 ± 22.2	38.1 ± 15.9
$H \approx 7$ or $H = 21$	93.5 ± 9.4	60.3 ± 22.3	38.3 ± 15.9
$H \approx 6$ or $H = 18$	93.7 ± 9.4	60.5 ± 22.4	38.6 ± 15.9
$H \approx 5$ or $H = 15$	94.1 ± 9.6	60.9 ± 22.5	39.2 ± 15.8
$H = 4$ or $H = 12$	95.0 ± 9.9	61.5 ± 23.2	40.5 ± 15.7
$H = 3$ or $H = 9$	94.5 ± 10.5	61.1 ± 23.5	43.7 ± 15.3
$H = 2$ or $H = 6$	94.2 ± 11.0	61.1 ± 23.5	52.6 ± 14.4
$H = 1$ or $H = 3$	94.2 ± 11.0	63.1 ± 24.1	86.2 ± 14.3
class = 0	13.3 ± 8.1	32.8 ± 14.6	39.5 ± 17.7
class = 1	14.7 ± 10.0	34.1 ± 20.1	35.4 ± 16.8
class = 2	10.0 ± 6.7	37.5 ± 17.5	32.7 ± 16.7
class = 3	9.4 ± 6.5	32.3 ± 17.0	34.3 ± 16.9
class = 4	9.0 ± 6.9	37.9 ± 19.3	32.9 ± 17.9
class = 5	8.9 ± 6.6	42.5 ± 27.8	33.6 ± 16.8
class = 6	9.2 ± 6.5	33.7 ± 14.2	34.6 ± 18.9
class = 7	9.3 ± 7.3	38.3 ± 25.2	33.5 ± 16.7
class = 8	10.1 ± 6.8	34.7 ± 16.7	36.4 ± 16.4
class = 9	9.0 ± 6.8	32.4 ± 15.3	35.6 ± 16.2
class = all	9.0 ± 6.5	31.7 ± 15.1	33.6 ± 16.5

imply that image entropy differences are not problems in classification tasks and may even enhance accuracy. Although the current advantage of image entropy equalization is limited to improving the methods using AE, the contribution could be broad areas because AE is applied in various fields. Moreover, future directions, such as 1) improving image entropy equalization, 2) extending entropy equalization to other datatypes, and 3) application and other usages, could enhance the image recognition and machine learning tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used described in the paper

Acknowledgments

This paper is an extended version of the conference paper published in SOMET2022 [7]. The authors are grateful to Excelence project PrF UHK 2215/2023-2024 for the financial support. This study is supported by JSPS/Japan KAKENHI (Grants-in-Aid for Scientific Research) #JP-23K11235.

References

- [1] N.K. Singh, K. Raza, Progress in deep learning-based dental and maxillofacial image analysis: a systematic review, *Expert Syst. Appl.* 199 (2022), 116968.
- [2] C.E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.* 27 (3) (1948) 379–423, <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>.
- [3] S. Uchida, D.Y. Tsai, Evaluation of radiographic images by entropy: application to development process, *Jpn. J. Appl. Phys.* 17 (1978) 2029–2034, <https://doi.org/10.1143/JJAP.17.2029>.
- [4] C.h. Thum, Measurement of the entropy of an image with application to image focusing, *Optica Acta: Int. J. Optics* 31 (2) (1984) 203–211, <https://doi.org/10.1080/713821475>.
- [5] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, *Parallel Distributed Processing Vol 1* (1986).
- [6] Pierre Baldi. 2011. Autoencoders, unsupervised learning and deep architectures. In Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop - Volume 27 (UTLW'11). JMLR.org, 37–50.
- [7] Toshitaka Hayashi, Dalibor Cimler, Richard Cimler, Image Entropy Equalization for Autoencoder-Based One-Class Classification, *Frontiers in Artificial Intelligence and Applications*, Volume 355: New Trends in Intelligent Software Methodologies, Tools and Techniques, Pages 310 - 321, 2022, DOI10.3233/FAIA220261.
- [8] E.H. Hall, Almost uniform distribution for computer image enhancement, *IEEE Trans. Comput.* 23 (2) (1974) 207–208.
- [9] David J. Ketcham Real-Time Image Enhancement Techniques, Proc. SPIE 0074, Image Processing, (9 July 1976); <https://doi.org/10.1117/12.954708>.
- [10] J. Verdenet, J.C. Cardot, M. Baud, H. Chervet, J. Duvernoy, R. Bidet, Scintigraphic image contrast-enhancement techniques: Global and local area histogram equalization, *Eur. J. Nucl. Med.* 6 (6) (1981), <https://doi.org/10.1007/BF00251349>.
- [11] L. Guo, M. Garland, The use of entropy minimization for the solution of blind source separation problems in image analysis, *Pattern Recogn.* 39 (6) (2006) 1066–1073, <https://doi.org/10.1016/j.patcog.2005.09.006>.
- [12] C. Chen, Y.-L. Li, L. Huang, An entropy minimization histogram merging scheme and its application in image compression, *Signal Process. Image Commun.* 99 (2021), 116422, <https://doi.org/10.1016/j.image.2021.116422>.
- [13] R. Roy, S. Ghosh, A. Ghosh, Clinical ultrasound image standardization using histogram specification, *Comput. Biol. Med.* 120 (2020), 103746, <https://doi.org/10.1016/j.combiomed.2020.103746>.
- [14] M.M. Goetz, M.C. Torres-Madroñero, S. Röthlisberger, E. Delgado-Trejos, Preprocessing of 2-Dimensional gel electrophoresis images applied to proteomic analysis: a review, *Genom. Proteomics Bioinformat.* 16 (1) (2018) 63–72.
- [15] Michał Strzelecki, Rafal Obuchowicz, Does image normalization and intensity resolution impact texture classification?, *Computerized Medical Imaging and Graphics*, Volume 81, 2020, 101716, <https://doi.org/10.1016/j.compmedimag.2020.101716>.
- [16] J. Liu, X. Leng, Y. Liu, “Deep Convolutional Neural Network Based Object Detector for X-Ray Baggage Security Imagery,” 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 1757–1761, doi: 10.1109/ICTAI.2019.00262.
- [17] F. Shao, J. Liu, W.u. Peng, Z. Yang, W.u. ZhaoYang, Exploiting foreground and background separation for prohibited item detection in overlapping X-Ray images, *Pattern Recogn.* 122 (2022), 108261, <https://doi.org/10.1016/j.patcog.2021.108261>.
- [18] K. Honda, M. Kurematsu, H. Fujita, A. Selamat, Multi-task learning for scene text image super-resolution with multiple transformers, *Electronics* 11 (2022) 3813, <https://doi.org/10.3390/electronics11223813>.
- [19] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, *J Big Data* 6 (2019) 60, <https://doi.org/10.1186/s40537-019-0197-0>.
- [20] S. Calderon-Ramirez, S. Yang, A. Moemeni, D. Elizondo, S. Colreavy-Donnelly, L.F. Chavarría-Estrada, M.A. Molina-Cabello, Correcting data imbalance for semi-supervised COVID-19 detection using X-ray chest images, *Appl. Soft Comput.* 111 (2021) 107692.
- [21] B. Wang, L. Li, M. Verma, Y. Nakashima, R. Kawasaki, H. Nagahara, Match them up: visually explainable few-shot image classification, *Appl. Intell.* 53 (9) (2023) 10956–10977.
- [22] E.D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, Q.V. Le, AutoAugment: Learning Augmentation Strategies From Data, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019, 2019, pp. 113–123, <https://doi.org/10.1109/CVPR.2019.00020>.
- [23] H. Inoue, Data Augmentation by Pairing Samples for Images Classification, 2018. ArXiv, abs/1801.02929.
- [24] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random Erasing Data Augmentation. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07), 2020, 13001–13008. <https://doi.org/10.1609/aaai.v34i07.7000>.
- [25] V. Sandfort, K. Yan, P.J. Pickhardt, et al., Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks, *Sci. Rep.* 9 (2019) 16884, <https://doi.org/10.1038/s41598-019-52737-x>.
- [26] W. Ma, T. Zhou, J. Qin, X. Xiang, Y. Tan, Z. Cai, Adaptive multi-feature fusion via cross-entropy normalization for effective image retrieval, *Inf. Process. Manag.* 60 (1) (2023), 103119.
- [27] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* 86 (11) (1998) 2278–2324.
- [28] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. ArXiv, abs/1708.07747.
- [29] A. Krizhevsky, G.E. Hinton, Learning multiple layers of features from tiny images., 2009.

- [30] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier Detection Using Replicator Neural Networks. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds) Data Warehousing and Knowledge Discovery. DaWaK 2002. Lecture Notes in Computer Science, vol. 2454. Springer, Berlin, Heidelberg, 2002. https://doi.org/10.1007/3-540-46145-0_17.
- [31] T. Hayashi, H. Fujita, A. Hernandez-Matamoros, Less complexity one-class classification approach using construction error of convolutional image transformation network, Inf. Sci. 560 (2021) 217–234, <https://doi.org/10.1016/j.ins.2021.01.069>.
- [32] F. Chollet. Keras. <https://github.com/fchollet/keras>, 2015.
- [33] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. The 3rd International Conference on Learning Representations (ICLR2015), 2015. <https://arxiv.org/abs/1409.1556>.
- [34] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 770-778.
- [35] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, 2017, pp. 2261–2269, 10.1109/CVPR.2017.243.
- [36] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, 2017, pp. 1800–1807.
- [37] Mingxing Tan, Quoc Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, PMLR 97:6105-6114, 2019.
- [38] Y. Dai, Z. Hu, S. Zhang, L. Liu, A survey of detection-based video multi-object tracking, Displays 75 (2022), 102317.
- [39] Y. Li, H. Fujita, J. Li, C. Liu, Z. Zhang, Tensor approximate entropy: An entropy measure for sleep scoring, Knowl.-Based Syst. 245 (7) (June 2022), 108503, <https://doi.org/10.1016/j.knosys.2022.108503>.
- [40] S. Hinojosa, K.G. Dhal, M.A. Elaziz, D. Oliva, E. Cuevas, Entropy-based imagery segmentation for breast histology using the Stochastic Fractal Search, Neurocomputing 321 (2018) 201–215.