A Generalized Deep Neural Network Approach for Digital Watermarking Analysis

Weiping Ding, *Senior Member, IEEE*, Yurui Ming, Zehong Cao, *Member, IEEE*, and Chin-Teng Lin, *Fellow, IEEE*

***Abstract*—Technology advancement has facilitated digital content, such as images, being acquired in large volumes. However, requirement from the privacy or legislation perspective still demands the need for intellectual content protection. In this paper, we propose a deep neural network (DNN) based watermarking method to achieve this goal. Instead of training a neural network for protecting a specific image, we train the network on an image dataset and generalize the trained model to protect distinct test images in a bulk manner. Respective evaluations from both the subjective and objective aspects confirm the generality and practicality of our proposed method. To demonstrate the robustness of this general neural watermarking approach, commonly used attacks are applied to the watermarked images to examine the corresponding extracted watermarks, which still retain sufficient recognizable traits for some occasions. Testing on distinctive dataset shows the satisfying generalization of our proposed method, and practice such as loss function adjustment can cater to the capacity requirement of complicated watermark. We also discuss some traits of the trained model, which incur the vulnerability to JPEG compression attack. However, remedy seeking for this can potentially open a window to understand the underlying working principle of DNN in future work. Considering its performance and economy, it is concluded that subsequent studies that generalize our work on utilizing DNN for intellectual content protection might be a promising research trend.**

***Index Terms*—Deep Neural Network (DNN), Digital Content Protection, Digital Watermarking, Privacy.**

I. Introduction

RECENT technology advancement has undoubtedly accelerated the speed and volume of digital content acquisition [1]. An obvious example is the ubiquitous cameras in the variety of circumstances, such traffic monitoring [2], assembly line inspection [3], environmental hazard detection [4]. These cameras capture images in massive volumes. However, the reduction cost of the digital content acquisition in no way compromises the importance of content protection. For instance, in the pursuit of traffic violations, the police should present scene images that are intact or authentic to support these cases [5]. Therefore, the research on relevant methods is of persistent interest.

Among the technologies that facilitate embedding information into digital content for authentication or protection purpose, digital watermarking is a widely and actively used method [6]–[9]. In this paper, we focus on the case of images with invisible digital watermarks. Currently, watermarking techniques are mainly divided into two categories. The approaches in the first category are implemented in the spatial domain, such as manipulating the least significant bits [10] and patch-based methods [11]. The advantage is that these methods are simple to implement; however, they are not resistant to operations applied to the watermarked image, such as filtering, transform and re-quantization. The second category of methods work in the frequency domain or transformed domain. For example, embedding the digital secrecy into the intermediate frequency components of the image after transformed from the spatial domain via discrete cosine transform (DCT) or discrete wavelets transform (DWT) [12]–[15]. Although these methods are robust to manipulations of the watermarked image, they are complicated in implementation.

There are several criteria that must be satisfied for a method to be considered for digital watermarking. These criteria are mainly from perceptive and robust perspectives, in addition to other aspects such as non-removal and unambiguity characteristics. The aspect of perception requires that the embedded information should not be perceived in an obvious and subjective way, even under intended manipulations. This is critical especially for invisible watermarks. The robustness criterion demands that the watermarked image should be resistant to common filtering operations such as blurring and enhancing to retain the secret information [16], regardless of whether these operations occur in the spatial domain or frequency domain. These criteria are also considered when we evaluate our proposed method later in this paper.

Essentially, digital watermarking requires an invertible and complex method to embed information into the target image and is thus generally non-linear, regardless of the domains. Hence,

Manuscript received September 25, 2020; revised December 27, 2020; accepted January 19, 2021. This work was supported in part by the Australian Research Council (ARC) under discovery under Grants DP180100670 and DP180100656, in part by the NSW Defense Innovation Network and NSW State Government of Australia under Grant DINPP2019 S1-03/09, in part by the Office of Naval Research Global, US under Cooperative Agreement ONRG-NICOP-N62909-19-1-2058, in part by the National Natural Science Foundation of China under Grants 61300167 and 61976120, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191445, and in part by Qing Lan Project of Jiangsu Province. *(Corresponding author: Weiping Ding.)*

Weiping Ding is with the School of Computer Science and Technology, Nantong University, Nantong 226019, China (e-mail: dwp9988@163.com).

Yurui Ming and Chin-Teng Lin are with the School of Computer Science, Centre for Artificial Intelligence, University of Technology Sydney, NSW 2007, Australia (e-mail: yrming@gmail.com; chin-teng.lin@uts.edu.au).

Zehong Cao is with the Discipline of ICT, School of Technology, Environment and Design, University of Tasmania, Hobart, TAS 7001, Australia (e-mail: zhcaonctu@gmail.com).

Digital Object Identifier 10.1109/TETCI.2021.3055520

neural network, especially the deep neural network (DNN) that exhibits high non-linearity, can be a candidate approach. There already exists utilizations of neural networks for digital watermarking; however, these approaches mainly address how to tailor a neural network for protecting one particular digital image (or cover image) [17]–[22], or as an auxiliary to assist the watermarking process [23], or to address special aspect of watermarking such as high robustness [24]. Although with respective outstanding achievements, limits are still accompanied with the above approaches. One eminent curtailment is that, considering the overhead for training neural network especially DNN, it is hardly to afford the cost for training individual neural network for protecting each image. Therefore, case-by-case watermarking presents great challenge and is impractical for real applications especially in a volume way.

However, DNN in various successful applications implicates its potential for digital watermarking [25]. The complexity of the deep network structure provides the possibility of blending the secret information and target images in a more intangible but appropriate approach; and the general monotonic activation function indicates an invertible process to retrieve the hidden information from a watermarked image [26].

In this paper, we design a DNN architecture for digital watermarking in a general way. To the best of our knowledge, we are the first to utilize DNN for watermarking in this economic manner, and by experiment we are confident of further generalization. We also investigate the characteristic of learned weights to interpret one case in which this method is ineffective. These are our major contributions.

The rest of the paper is structured as follows. In Section II, we describe the motivation and design of the DNN model. The overall treatment is to train the model on an image dataset, and then after the network learns how to embed the watermark into original images and retrieve the watermark from the watermarked images, the model is tested against distinct images to verify the generalization capability. In Section III, by instantiating the network according to specified configurations, the proposed method is evaluated on a public image dataset and intriguing results are illustrated. In Section IV, the method is systematically assessed by referring to the criteria of watermarking to verify the competence of our approach.

II. Motivation and Network Architecture

A. Motivation

Recent years witness the great achievements of DNN in various applications. It also revolutionizes some conventional fields with surprising accomplishments, such as generative model [27], reinforcement learning [28], etc. The computational capability of the DNN, either as a mapping from the uniform random distribution to a specific random distribution, or as a powerful function estimator, have demonstrated the superior competence over other methods in a variety of tasks. In these applications, no explicit rules are established for DNN to guide its behavior, and DNN learns from samples or experiences and generalizes to new situation. Therefore, considering the capabilities of DNN, it is appealing to consider its potentiality for digital watermarking.

A retrospection of the general application of DNN reveals that the paradigm is rather stereotypical, i.e., training the designed network with a training set and test it on a separate test set. Hence, a straightforward migration of DNN to digital watermarking is similar, i.e., training the designed network on substantial images to enable the network to learn the way for embedding and retrieving of the watermark image; then test the learned capability of network on other distinct images. Extra factors are also needed to be considered, such as training the DNN in an end-to-end way.

The above concept of utilizing DNN for watermarking might be potentially compliant with the theoretic analysis. If G is implemented by DNN, i.e., parameterized by weights θ, and we can tactically train G to have optimal θ that is common to all images, thus G can be parameterized by θ in a latent but general manner:

$$W = G_{\theta}^{-1}\, G_{\theta}\,(W) \tag{1}$$

where indicates this reparameterization frees G from being dependent on a specific image but based on features common to an image collection. It releases the powerfulness of G_{θ} so that it can be utilized to watermarking other images in general.

Notably, the deduction above also means that either in training or testing, a series of images are processed by the proposed model. Therefore, we avoid the terminology cover image, which means a particular image with which various watermarking methods in previous literatures are working on. We adopt the image instead of cover image to reflect this generality.

B. Network Architecture

The above description leads to the establishment of the overall designed neural network architecture in Fig. 1. It consists of several modules or subnetworks that work together to meet the final goal. We first utilize transpose convolution [29] to convert the original image and the watermark into a higher dimensional space to blend them together. The module for upscaling the dimension is named up-sampler, and the blending operation is performed by the module named blender, both are neural networks. After embedding the watermark information, a component named down-sampler is used to make the blended image have the same dimensions as the original image, as well as restore some features after blending. To assess that a digital artwork is protected by watermarking, an extracting sub-network called extractor is designed to retrieve the embedded information.

The reason for designing the up-sampler module is as follows. As mentioned above, watermarking in transformed domain is effective; however, it is not intuitive to design a transformed domain with distinctive properties for neural networks. Instead, we postulate that a higher dimensional space (or latent space) resulted from network operations might resemble some similarities, for example, it has a higher freedom to blend the pixels from the original image and the watermark. This freedom might be beneficial for the network since it can choose the most

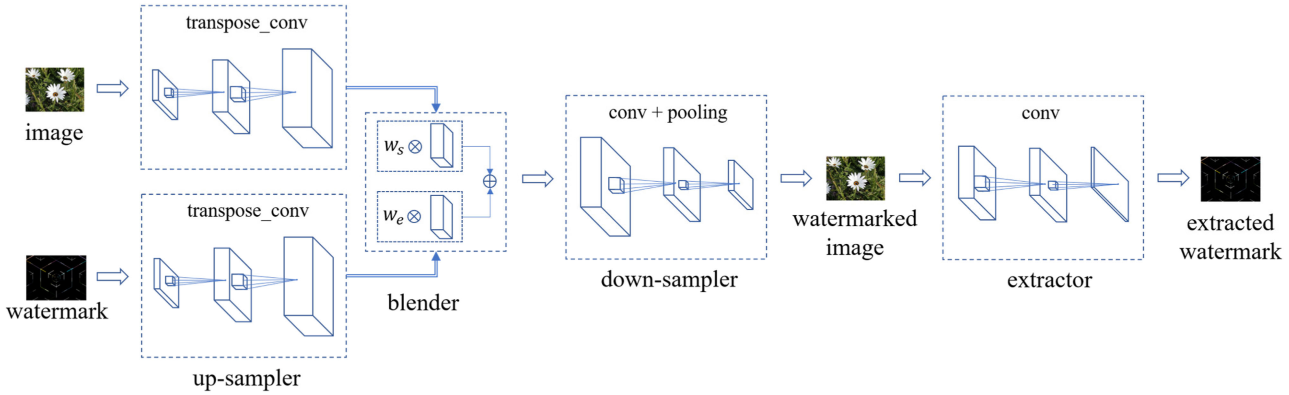

Fig. 1. Network architecture for digital watermarking.

appropriate way by learning to increase the quality of embedding and resist manipulations to the watermarked image.

Furthermore, although blending of the up-sampled image and watermark can be as trivial as an element-wise addition, to seek an optimal way in doing it, a variant of attention mechnism is employed here to let the model learn the best way of composing the original image and watermark. To do this, trainable variables w_s and w_e with the same dimension as up-sampled image are created to weight the up-sampled image and watermark before blending. It is expected that during the training process, the places which should be blended different from other parts are fine-tuned by error-propagation. This treatment, i.e., appropriate weighting of the image pixels and watermark pixels when blending, resembles the prevailing attention mechnisim in DL practice [30], [31], can potentially improve the watermarking quality.

The functionality and performance of the model are enforced and measured by simultaneously comparing the original image with watermarked image and the original watermark with the extracted watermark according to (2), which is in the form of mean square error as follows:

$$L\ (\theta) = \sum_{i,j}\left(\bar{I}_{i,j}\,(\theta) - I_{i,j}\right)^2 + \sum_{i,j}\left(\bar{W}_{i,j}\,(\theta) - W_{i,j}\right)^2 \quad (2)$$

where I, W, $\bar{I}$ and $\bar{W}$ denote the original image, original watermark, watermarked image and extracted watermark respectively, and θ represents the overall parameter set.

Notably, the size of the watermark used in our work is the same size as the image, and we know in other literatures the watermark might be smaller than the cover image in magnitude. We choose this due to the following reason. Because of the black-box property and limited achievements in explainability of DNN computations, it is difficult in posing a mathematical proof, but we still aim to demonstrate that the operations of neural networks might bear the feasibility of performing watermarking intrinsically. Actually, without non-linear activation, computation of a specific neural network is equivalent to linear transformation, for instance,

$$y\ = M_{l_n}\ (\cdots(M_{l_2}\,(M_{l_1}x))) \quad (3)$$

Here M_{l_i} represents the weights of i-th layer, and this is indeed matrices composition, which can be simplified as $y = Mx$. Furthermore, according to our choice, $\dim\,(I) = \dim(W)$, denote this original space shared by images and watermark by $\mathcal{S}$, we have $I \in \mathcal{S}$ and $W \in \mathcal{S}$. Assume the transform realized by neural network brings $\mathcal{S}$ into $\bar{\mathcal{S}}$. Let $\bar{\mathcal{S}}_I$ and $\bar{\mathcal{S}}_W$ denote the subspace of $\bar{\mathcal{S}}$ where transformed I and W potentially resides after transform, i.e., $MI \in \bar{\mathcal{S}}_I$, $MW \in \bar{\mathcal{S}}_W$; if to some extent, M can be tuned to have $\bar{\mathcal{S}}_I$ and $\bar{\mathcal{S}}_W$ perpendicular, i.e., $\bar{\mathcal{S}} = \bar{\mathcal{S}}_I\ \oplus \bar{\mathcal{S}}_W$, then it can be asserted that neural networks can potentially perform watermarking, at least it is conceptionally plausible, because $\bar{\mathcal{S}}_I$ and $\bar{\mathcal{S}}_W$ impose the lest interference to each other. However, a rigorous proof is still a future work.

III. Experiments

For convenience and the avoidance of copyright infringement, we utilize a dataset from Kaggle [32]. The dataset is a collection of images for flower recognition. The preference of this dataset also lies in other considerations. For example, the images in the dataset are about 320 × 240 pixels, a reasonable resolution for carrying out the experiments. In addition, the images are categorized into five categories; we can use the first four categories of images for training, and the final category for testing. The distinctiveness of training images and test images is a stronger evidence to show the practicality of the proposed network architectures on success.

These images and the watermark are shown in Fig. 2. Fig. 2(a) is a snapshot of the training images. The canonical dimensions of images processed with the network are set to 320 × 240, so the images are selected and manipulated to match the dimension constraint. After rectifying the images, there are 1703 images for training and 427 images for testing. A watermark image collected from the Internet is shown in Fig. 2(b), by courtesy of the original provider. We are with no intention to infringe copyright besides the sole research purpose in this paper. The watermark is chosen to be the same size as the training images to simplify network operations, i.e., the same up-sampler structure can be applied indifferently to image and watermark. Another reason for choosing a large size watermark is that watermarking is performed in the spatial domain in the way designed above.

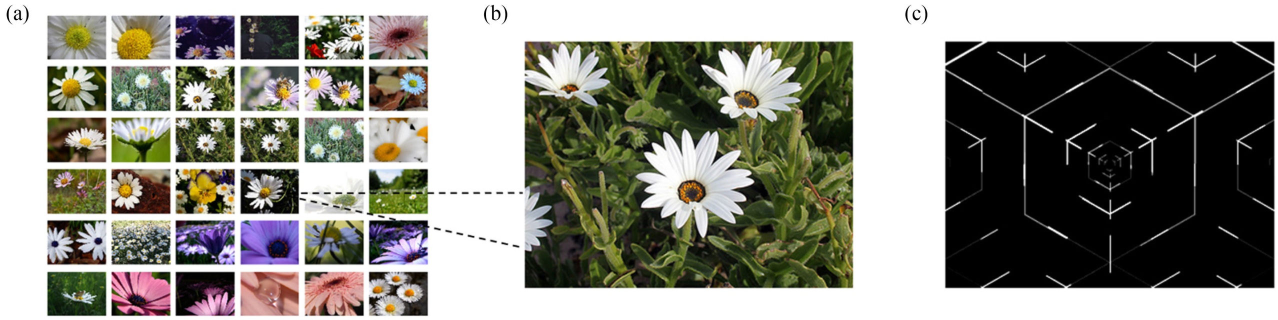

Fig. 2. Images for training and the corresponding image of watermark (note the image of watermark is rescaled with the maximal pixel value from 63 to 255 for better illustration). (a) A snapshot of the training images, (b) A watermark image collected from the Internet, and (c) Watermark.

TABLE I
NETWORK CONFIGURATION*

Name	Operation	#Features	Filter Size	Stride	Comment
Up-sampler	Conv2DTranspose	16	5x5	2	No bias
	Conv2DTranspose	16	5x5	2	No bias
	Conv2DTranspose	16	5x5	2	No bias
Down-sampler	Conv2D	12	5x5	1	No bias
	AvgPool2D	-	2x2	1	
	Conv2D	6	5x5	1	No bias
	AvgPool2D	-	2x2	1	
	Conv2D	3	5x5	1	No bias
	AvgPool2D	-	2x2	1	
Extractor	Conv2D	12	5x5	1	No bias
	Conv2D	6	5x5	1	No bias
	Conv2D	3	5x5	1	No bias

*Blender is not included in the table due to the descriptive difficulty aligning with other layers. The operation in blender is an element-wise addion of up-sampled image and watermark. The extra blending weights is to employ attention alike mechanism, emphasizing the postions that matter for efficient blending, as explained in the text.

There lacks conclusion about the diffusion of watermark information to the original image upon blending. However, the robustness of watermarking also requires that watermark information can scatter into as large spatial domain as that covering the actual image size, and this urges us to choose a large size watermark. To relieve the impact of large size watermark upon blending, a thin watermark, i.e., watermark with simple texture and regular pattern is chosen to offset the size factor. To consider a smaller size but more complicated texture watermark with sufficient watermarking quality is still under research following this thread.

The instantiation of the architecture uses the de facto modules provided by the library TensorFlow [33] and no customized operations for ease of replication to benefit subsequent research. The configuration of the network architecture is shown in Table I.

We mention some subtle parameters worthy of consideration in Table I. For the filter size of up-sampler, it should be at least with 5x5 to ensure patch coverage, because transpose convolution from the previous convolutional layer to the current layer, is equivalent to convolution with stride equal to 2 from the current layer to the previous layer, reversely. For the down-sampler, the number of feature maps of current layers drops by half compared with previous layer, and the number of final layer is restricted to 3 channels to output a color watermarked image. The choices for the numbers of feature maps and fiter size of other layers are mostly empirical. The bias can be used to compare with the illuminance level of an image, but here it is not necessary to learn biases here due to the diversity of images.

With the above configuration, we train the neural network for 10,000 iterations with a batch size of 8 and a learning rate of 0.001. To stabilize the training process, the learning rate is reduced each 400 iterations by a factor of 0.92. To avoid overfitting, we also use images from test image set to monitor the training process. The statistical distributions of the training image set and test image set are different from each other, so it is more objective to assess the training process. The feasibility of our proposed method can be preliminarily asserted by the convergent validation process in Fig. 3.

IV. EVALUATIONS

A. Watermarked Images

To systematically assess the quality of watermarked images, we evaluate them from two perspectives. The first is from the subjective perspective. We select 6 images from the test set at random and present them to 6 university students, who are with normal or correct to normal vision. For each selected image, paired with the corresponding watermarked image, they are shown to the subjects for subjective discerning between them. Then, the watermark is shown as a clue to let the subject repeat the process again. Finally, the subjects rank the difficulty level to distinguish original image and watermarked image among image pairs. The ranked difficulty levels are used to assert the perceptive quality of watermarked image. The reason for not presenting watermark as a clue at the first stage is to make the contrast so subjects have a better sense at the difficulty level.

For images with simple and monotonous textures, it is reported tiny perceivable difference between the original image and watermarked image. For images with moderately complex textures or scenarios, the perceivable difference is neglectable and only noticeable upon presence of the image of watermark as a cue. For images with highly complex textures or scenarios, there is no subjective difference even with presence of the image of watermark as a cue. We show some example cases in Figs. 4 and 5 respectively. As the first attempt in performing watermarking in a general way, the subjective results illustrate

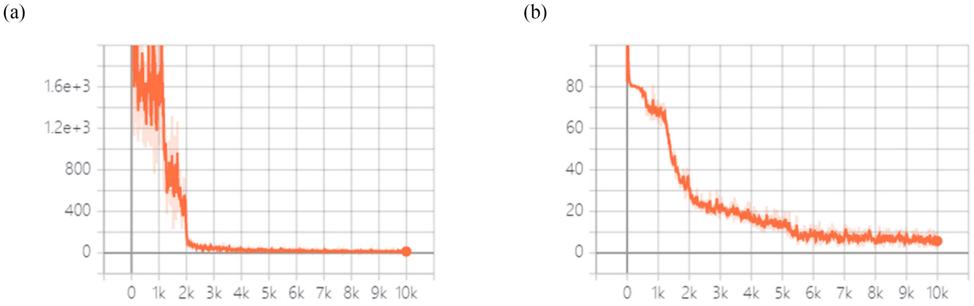

Fig. 3. Validation loss during the training process. (a) image loss, and (b) watermark loss. x-axis labels the training iterations, and y-axis indicates the mean squared error between the original image and watermarked image.

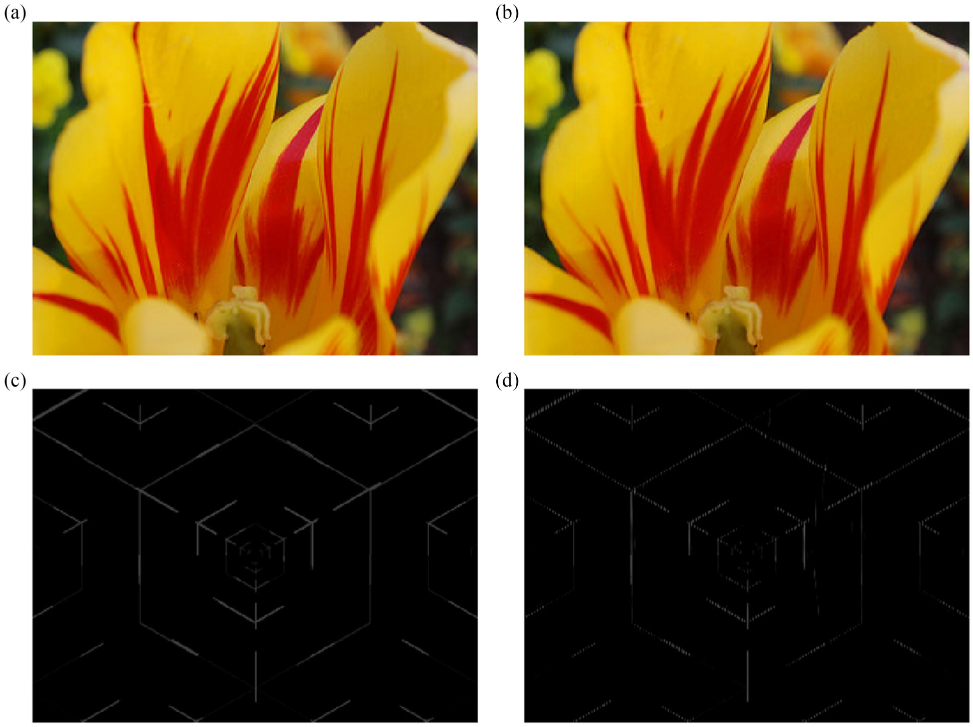

Fig. 4. Case for an image with simple and monotonous textures. The perceptive differences can be spotted only from special observation angles, but they are overall still quite tiny. (a) Original image, (b) Watermarked image, (c) Original watermark, and (d) Extracted watermark.

the potential of considering more tricks in the neural network to refine the perception.

To objectively assess our method, we adopt the peak signal-to-noise ratio (PSNR) defined in [34] for evaluation:

$$PSNR = -10 \cdot \log_{10} \frac{\frac{1}{3*M*N}\sum_{k=1}^{3}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(\bar{I}\left(m,n,k\right)-I\left(m,n,k\right)\right)^2}{255^2} \tag{4}$$

TABLE II
QUALITY METRICS OF WATERMARKED IMAGES

IMG ID	123	99	174	333	396	294	Average	Baseline
PSNR*	41.8	39.1	38.1	36.2	32.1	44.4	38.6	38
SSIM‡	99.7	99.5	99.7	99.4	99.4	99.7	99.5	99

*units: dB, ‡percentage.

It is reported in [34] that PSNRs larger than 38 dB are associated with high-quality watermarked images. In [28], the PSNR threshold is recommended as 30 dB; and [22] indicates that a minimum PSNR of 35 dB can underpin satisfying watermarked image in various cases. The PSNRs of the 6 watermarked images chosen at random are given in Table II, with an average of 38.6. However, PSNR might not in fully compliance with the subjective perception. Fig. 6 shows the case of watermarked image with the lowest PSNR with 32.1, and the intrinsic texture of the image still prohibits easy discrimination. Overall, the objective assessment confirms the promising adoption of this general method.

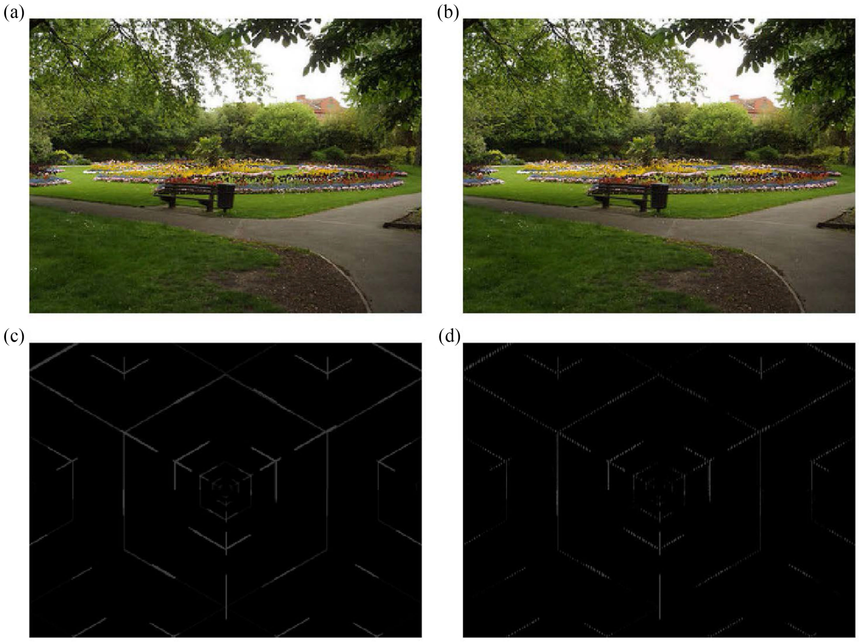

Fig. 5. Case for an image with highly complicated texture and scenario. The perceptive difference is invisible to observers even with the presence of watermark image as a reminder. (a) Original image, (b) Watermarked image, (c) Original watermark, and (d) Extracted watermark.

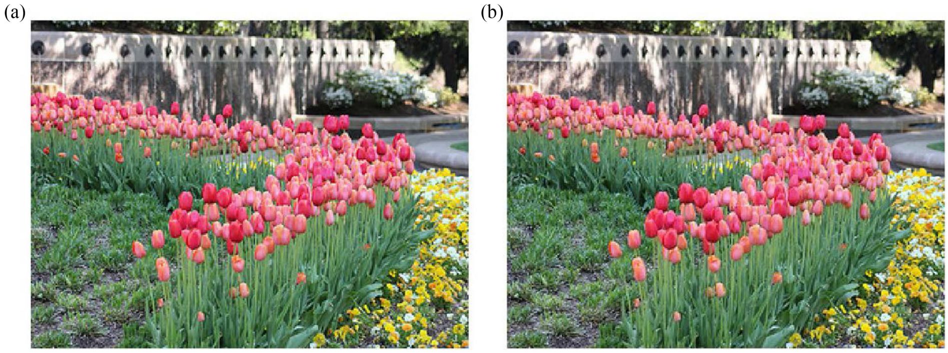

Fig. 6. Image with intrinsic complex context reporting low PSNR of 32.1 still prohibits easy discrimination between the original image and watermarked image. (a) Original image; and (b) Watermarked image.

Further more, we also employ the structural similarity index (SSIM) defined in [35] to asseess the watermarked image. PSNR with high value might not be fully compliant with the perceptive chracteristic of human visual system (HVS), of which SSIM takes special consideration instead. Due to the various images and watermarks used in different literatures, there lacks a common benchmark. Both [7] and [9] are the most up-to-date surveys of digital watermarking, from which we empirically assume 90% as benchmark; henceforce, the SSIM values shown in Table II indicate postive prospective of the proposed method.

B. Extracted Watermark

To assess the robustness of the proposed watermarking mechanism, several modifications or attacks are applied to the watermarked images to examine the extracted watermark. Similar to the evaluation of watermarked images, the assessment of the extracted watermark is categorized into subjective and objective way respectively. The attacks considered here include clipping, rotation, low-pass filtering, high-pass filtering, median filtering, noise degradation, and JPEG compression [5].

For clipping, the half of the watermarked image is chopped from the watermarked image and replaced by zero values to maintain the original size. This is to align with the neural network's input size requirement. For rotation, the watermarked image is rotated by 45 degrees in a counterclockwise manner. For filtering, we apply a Gaussian filter for the low-pass filtering and a Laplacian filter for the high-pass filtering in the spatial domain in Fig. 7; and the size of the median filter is 3 × 3, the same dimension as other filters. For the noise degradation, the random Gaussian noise of 25 dBw (decibel watt) is merged into the watermarked image to inspect the effect on the extracted watermark. For JPEG compression, we save the primitive RGB

(a)

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

(b)

− 0.125	− 0.25	− 0.125
− 0.25	2.5	− 0.25
− 0.125	− 0.25	− 0.125

Fig. 7. Filtering results. (a) Gaussian filter, and (b) Laplacian filter.

TABLE III

NC OF THE EXTRACTED WATERMARK FOR A GIVEN IMAGE

Attacks	Clipping	Rotation	Noise Degrading	High-pass Filtering	Low-pass Filtering	Median Filtering	JPEG
NC^*	55.42	14.96	75.99	64.89	4.22	10.29	7.07

*units: percentage.

values into a JPEG image with quality 95 and reload it for watermark extraction.

For the subjective evaluation, we find that clipping, noise degrading and high-pass filtering can retain the watermark with sufficient quality from the perceptive perspective, while other attacks incur obvious perceptive distinction. We only illustrate the case of high-pass filtering in Fig. 8, and the case of low-pass filtering in Fig. 9. We defer the vulnerability analysis of rotation, the blurring and JPEG compression attacks in the discussion section.

To objectively assess the extracted watermark after various modifications (or attack) to the watermarked image, we adopt the measurement in [36], i.e., the normalized correlation (NC) described by :

$$NC = \frac{W \cdot \bar{W}}{\sqrt{\|W\|^2}\sqrt{\|\bar{W}\|^2}} \tag{5}$$

$$\|W\|^2 = \sum_{k=1}^{3}\sum_{m=1}^{M}\sum_{n=1}^{N} w(m,n)^2 \tag{6}$$

$$W \cdot \bar{W} = \sum_{k=1}^{3}\sum_{m=1}^{M}\sum_{n=1}^{N} w\left(m,n\right) * \bar{w}\left(m,n\right) \tag{7}$$

where W denotes the original watermark, and $\bar{W}$ denotes the watermark extracted from the modified watermarked image.

Notably, this measurement implicitly assumes that the NC between the extracted watermark from the intact watermarked image and the original watermark are identical; however, this is usually not the case. Regardless of whether the watermarked image undergoes some attack or not, the watermark tends to exhibit some degradation after extraction. Denote the NC between the original watermark and watermark extracted from the intact watermarked image as NC_0, we can rewrite (5) as (8) as follows:

$$NC = \frac{W \cdot \bar{W}}{\sqrt{\|W\|^2}\sqrt{\|\bar{W}\|^2}} \cdot \frac{1}{NC_0} \tag{8}$$

We calculate the NC based on (7) where NC_0 equals 70.34%; and the results are shown in Table III. There is no explicit specification of NC in literatures to benchmark. According to our estimation, a threshold of 50% can be accepted. Table III indicates that the commonly used attacks, such as clipping and sharpening (or enhancement), exhibit satisfactory NC, while other attacks are not. These preliminary results provide more promising achievements following by subsequent research, and improved robustness of proposed method under specific attacks requires more study of the proposed network model as well as more reference to other literatures. For example, [21], [22] and [24] adopt different network architectures, respectively and specifically demonstrate resistance to rotation and compression attacks with satisfying results. We plan to further improve our proposed method in this regard as future work.

TABLE IV

QUALITY METRICS OF WATERMARKED IMAGES

IMG ID	123	99	174	333	396	294	Average	Baseline
PSNR*	42.9	45.4	45.3	38.0	35.1	39.5	41.0	38
SSIM‡	98.8	99.1	99.1	98.7	98.7	98.9	98.9	99

*units: dB, ‡percentage.

C. Generalization

As mentioned above, the uniqueness of our proposed method is its generalizing ability. Conventional watermarking algorithms utilizing neural networks tend to focus on one or several images and train the network to best fit the characteristics of these limited number of images. This renders higher performance on the cost of computational overhead of repeated training for each image case. Our proposed method emphasizes the one-time training and generalizes the trained model for watermarking on new images. This method might not lead to the promising result for a given image, however, the reduced computational expense can entitle as a more practical usage.

The dataset used above mainly concerns of flowers, which might just cover part of commonly encountered image scenes. To understand the generalizing ability of the trained network, we consider another dataset VOC2012, which is available from [37] and containing realistic scenes of four categories. We demonstrate that our model can directly generalize to this dataset. To do this, we randomly choose 6 images to watermark and calculate the corresponding PNSR and SSIM. Fig. 10 illustrates the chosen images and one watermarking instance, and Table IV shows the statistics. Contrasted with Table II, Table IV shows a comparable PSNR and slightly dropped SSIM; overall, they still indicate a satisfactory watermarking by a straightforward generalization.

D. Capacity

An interesting question could be asked is whether the approach proposed by us is general enough to process any image dataset and watermark. Although it is hard to specify some quantity, we can consider from various aspects such as resolution, contrast, texture, artifacts, distortion, etc., to indicate the watermark the proposed method can work with. In our work, we choose a watermark image with simple texture and regular pattern to meet this criterion.

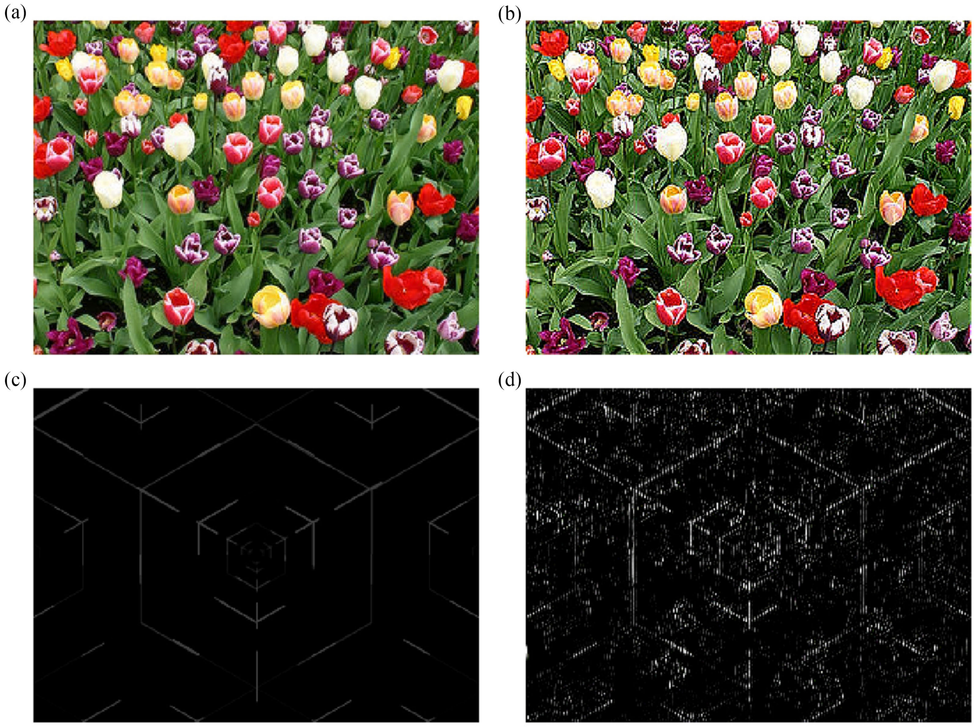

Fig. 8. Case for high-pass filtering. Although there are changes to the extracted watermark, the overall contours are still relatively perceivable. (a) Watermarked image, (b) High-pass filtered image, (c) Original watermark, and (d) Extracted watermark.

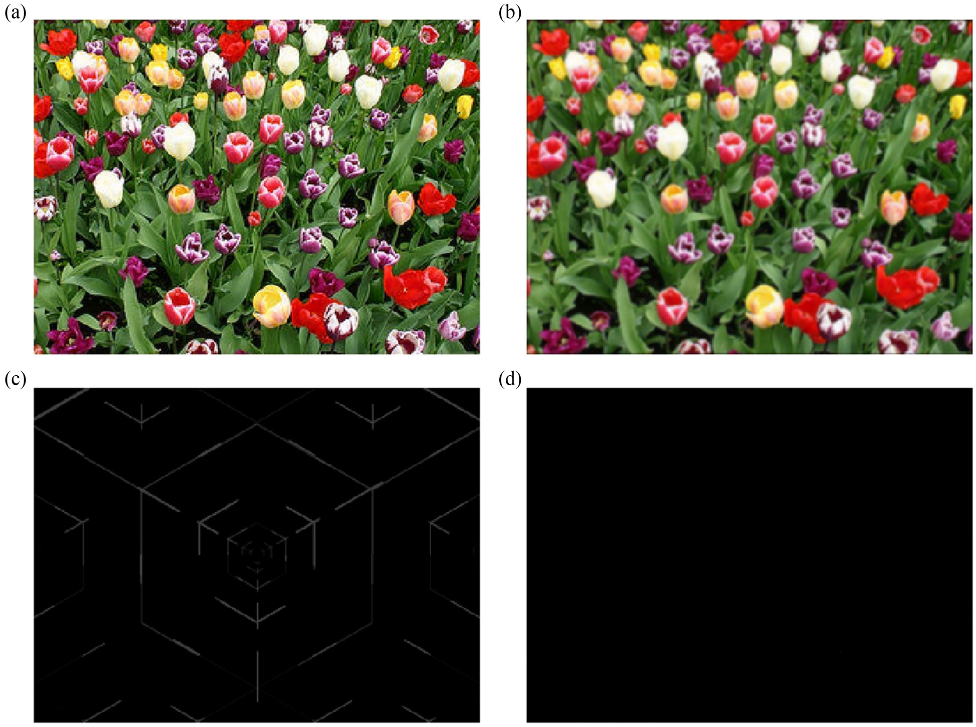

Fig. 9. Case for low-pass filtering. The network fails in extracting the watermark. (a) Watermarked image, (b) Low-pass filtered image, (c) Original watermark, and (d) Extracted watermark.

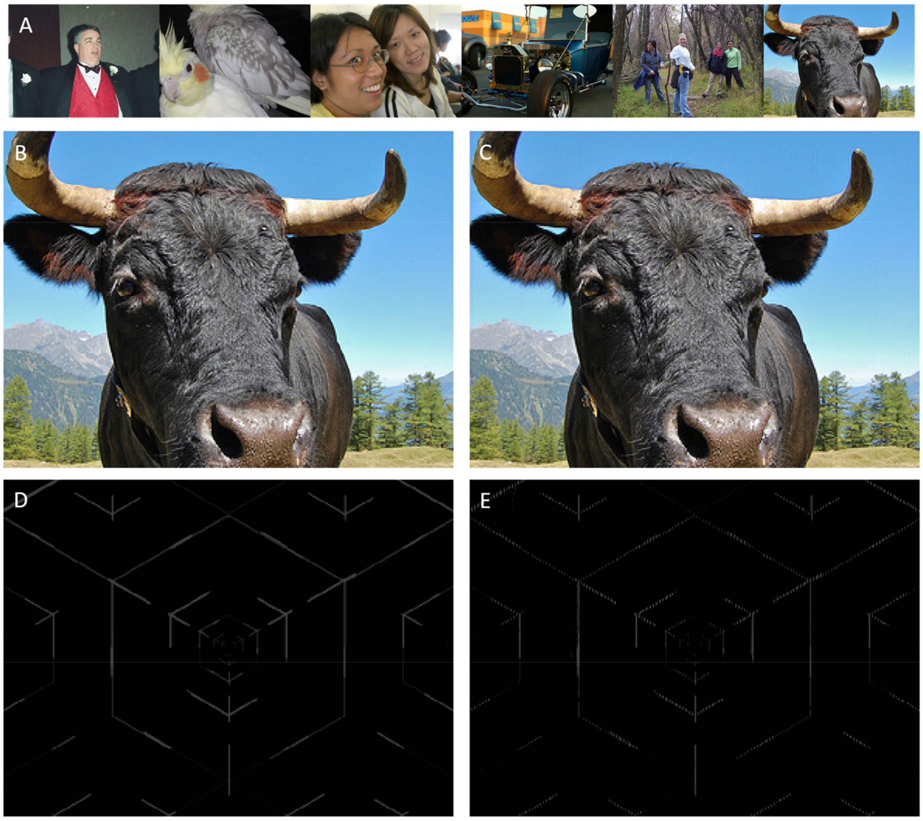

Fig. 10. The chosen images and one watermarking instance. (a) 6 randomly chosen images from VOC2012 test dataset, (b) Original image, (c) Watermarked image, (d) Original watermark, (e) Extracted watermark.

Meantime, if some characteristics of images in a dataset can be maintained through the dataset, such as with high contrast, complex texture, our proposed method is more likely to leading to high performance. Usually, natural scene image can meet this requirement, and our model is to some extent independent of the image dataset. And to evince our assertion, we use the above VOC2012 test dataset to demonstrate: (1) Increased complexity of watermark can deteriorate our proposed method performance; (2) Lessen the quality of retrieved watermark allows the usage of complex watermark.

Fig. 11 shows an instance from dataset and subsequent operations on it. Fig. 11(b) and (d) illustrate the watermark of increased complexity and the corresponding result. It is obvious that our model can fail in processing complicated watermark image. Fig. 11(d) and (e) demonstrates that, by sacrificing the quality of retrieved watermark via assigning a small value to β in (9), which a variant of (2), better imperceptive effect of watermarked image can be achieved:

$$L\ (\theta) = \sum_{i,j} \left(\bar{I}_{i,j}\left(\theta\right) - I_{i,j}\right)^2 + \beta \cdot \sum_{i,j} \left(\bar{W}_{i,j}\left(\theta\right) - W_{i,j}\right)^2 \tag{9}$$

By regulating the weighting of different losses via (9), the network behavior can be adjusted to some extent.

It is obvious that the rationality for (9) is that watermarking is usually for copyright protection, but not for encryption. It means that most of time, as long as the retrieved information is discernable to prove the ownership of digital content or identify the authenticity, it can be acceptable. However, to take more tricks from neural network practice for watermarking is an ongoing research, and Fig. 11 shows that our proposed approach can be considered for more complicated watermark by customizing the loss. Moreover, if watermark with more vivid pattern and texture is required, this can be fulfilled by designing a bilinear mapping, which projects the pixel values of the actual small size watermark into the large surrogate watermark image of regular patterns, and Fig. 12 illustrates this technique.

E. Complexity

The estimation of the complexity of the proposed method can help to assess the hardware feasibility when deploying the algorithm, and benchmark subsequent research to reduce the complexity. In this subsection, we explore the computational complexity of watermarking via our method.

For neural networks, the commonly used metric for complexity is floating point operations (FLOPs) [38]. For the network architecture in this paper, two operations are mainly involved: convolution and pooling. We derive formulas as variants in [38] to calculate the FLOPs.

For an image $I \in \mathcal{R}^{H\times W\times C}$, and a kernel $K \in \mathcal{R}^{k\times k}$, assume convolutions and pooling are implemented as series of multiply-accumulate operations (MACs). Because each MAC

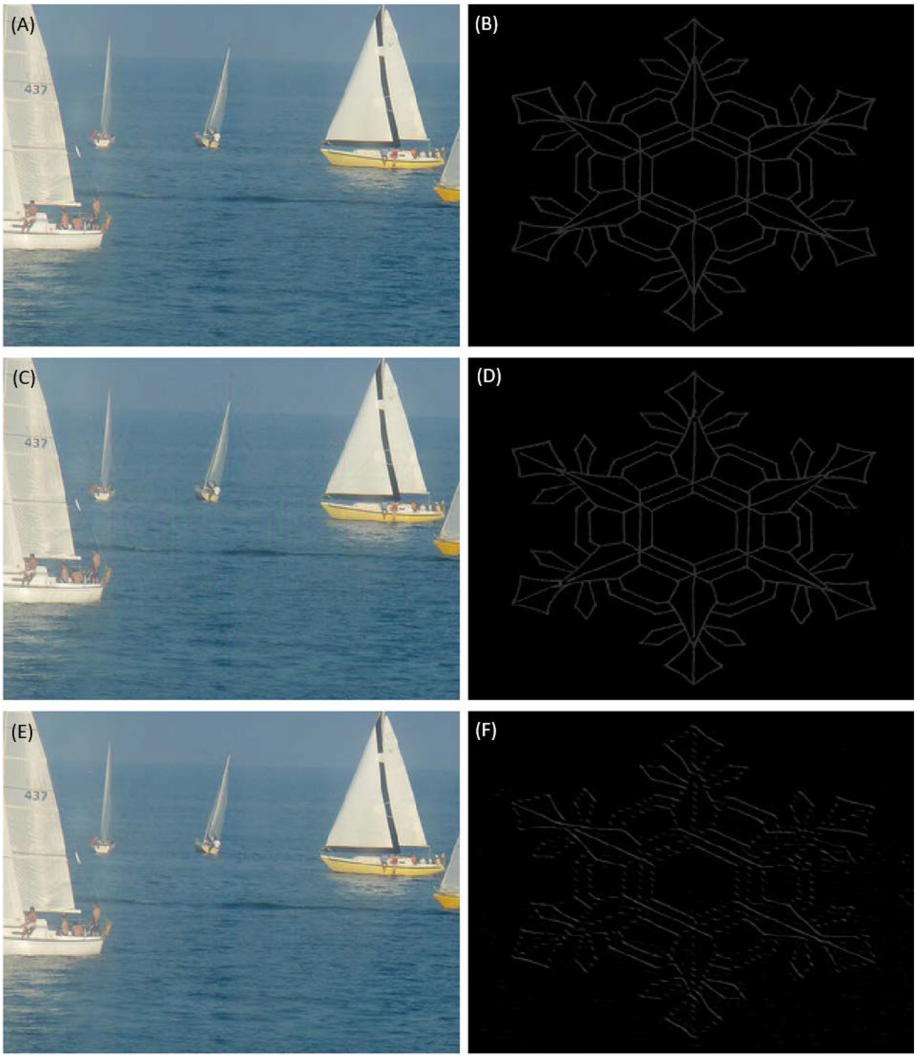

Fig. 11. An instance from dataset and subsequent operations. (a) Original image, (b) Original watermark, (c) Watermarked image according to loss defined by (4), (c) Extracted watermark according to loss defined by (2), (d) Watermarked image according to loss defined by (9), and (e) Extracted watermark according to loss defined by (9).

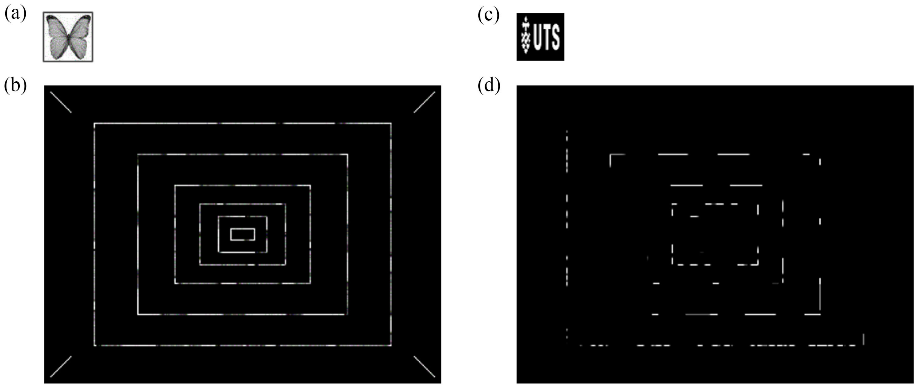

Fig. 12. Bilinear mapping technique. (a) Original butterfly watermark, (b) Watermark embedded into a specially designed regular pattern of a large surrogate image, (c) Original UTS logo watermark, and (d) Watermark embedded into a specially designed regular pattern of a large surrogate image.

TABLE V
FLOPs of The Network Model

Name	Operation	Height	Width	#Feature-maps	Filter Size	Stride	FLOPs	Comments
Input		240	320	3				
Up-sampler*	Conv2DTranspose	480	640	16	5x5	2	$O(184.3m)^{\dagger}$	No bias
	Conv2DTranspose	960	1280	16	5x5	2	$O(3.9b)^{\dagger}$	No bias
	Conv2DTranspose	1920	2560	16	5x5	2	$O(15.7b)$	No bias
Blender	Mul/Add	1920	2560	1			$O(176.9m)$	
Down-sampler	Conv2D	1920	2560	12	5x5	1	$O(47.1b)$	No bias
	AvgPool2D	960	1280	12	2x2	1	$O(58.9m)$	
	Conv2D	960	1280	6	5x5	1	$O(4.4b)$	No bias
	AvgPool2D	480	640	6	2x2	1	$O(7.3m)$	
	Conv2D	480	640	3	5x5	1	$O(276.4m)$	No bias
	AvgPool2D	240	320	3	2x2	1	$O(921.6m)$	
Total							$O(91.8b)$	

*There are two up-sampler modules, one for image, the other for watermark; †b: billion, m: million.

equals two FLOPs, hence FLOPs involved in one convolution is $2C_{in}K^2$. Assume FLOPs for activation functions is the same order as the dimensions of output feature maps, so the total number of FLOPs for convolutional layer is (10). Note it is assumed no bias.

$$\text{FLOPs} = 2H_{in}W_{in}C_{in}K^2C_{out} + O\left(H_{out}W_{out}C_{out}\right) \tag{10}$$

Because usually for convolution, input and output images are only different from number of channels, and $2C_{in}K^2 \gg 1$, (10) can be rewritten as

$$\text{FLOPs} = O\left(2K^2H_{in}W_{in}C_{in}C_{out}\right) \tag{11}$$

For transpose convolution, an equivalent viewpoint is to treat it as convolution from output to input with stride S, FLOPs is calculated by

$$\text{FLOPs} = \text{O}\left(2K^2H_{out}W_{out}C_{out}C_{in}/S^2\right) \tag{12}$$

For pooling, let S denote the stride and notice the input and output share the same number of channels, we can calculate FLOPs by (13). Notably, the factor 2 is omitted because MACs for pooling is approximately the half of a convolution. In addition, pooling is applied feature-map-wise, therefore, only either input channel or output channel is taken into account.

$$\text{FLOPs} = H_{in}W_{in}C_{in}K^2/S^2 \tag{13}$$

Based on these formulas, Table V shows the rough estimation of FLOPs of the network. It can be noticed that the transpose convolution and convolution sandwiched blender module dominate the total number of operations, which indicates the direction of work in future research in reducing the complexity. We also compare the execution time with such methods as Samee's Method [39], Yu's Method [40], as in Table VI. Our method runs a comparable execution time n a Quadro RTX 6000 Graphics Card. The execution time comparison among different methods is shown in Table VI. We also compare our proposed method with Guo's Method [41], Li's Method [42] and Wang's Method [43]. Due to the page limitation, we here do not provide more results. It is obvious that we achieve a clear path to reduce the complexity and have a confidence to shorten the execution time dramatically by restricting the dimensions of space where blending performs.

TABLE VI
Execution Time Comparison Among Different Methods

Methods	#Operations	Execution time (seconds)
Samee's Method [39]	10298433187	16.2
Yu's Method [40]	2043657327	4.4
Our proposed Method	$O(91.8b)$	4.4

V. Discussion and Analysis

In this paper, we introduce DNN to perform digital watermarking in a general way, and there exists several aspects that need further study. For example, how to architect more elegant DNN to perform the watermarking task, meanwhile improve the robustness for modifications against watermarked image. In this discussion, we further analyze these concerns to inspire subsequent research following this thread.

A. Low-Pass Filtering Attack

We first discuss the low-pass filtering issue raised in the last section, i.e., the potential reason that low-pass filtering induces dramatic impact on the watermarked image.

To understand the low-pass filtering effect, it is necessary to examine the characteristic of the sub-network (extractor) to extract the watermark from the watermarked image. The extractor is a deep convolutional network; hence, we only need to investigate the traits of the weights. For convenience, we focus the weights (or kernels) of the last convolutional layer.

Notably, different research domains adopt different terminologies for the same concept, such as weights in neural networks, masks in image processing, kernels in mathematics, and filters in signal processing, etc., they are indeed all referring to the same thing. However, treatment in a specific domain might facilitate the investigation of weights. For example, if we treat image processing as a special case of signal processing. One important task for signal processing is filter design, which usually takes place in frequency domain. Drawing inspiration from this, to invesitage the trait of weights, we can instead study the corresponding frequency response, which is more intuitive.

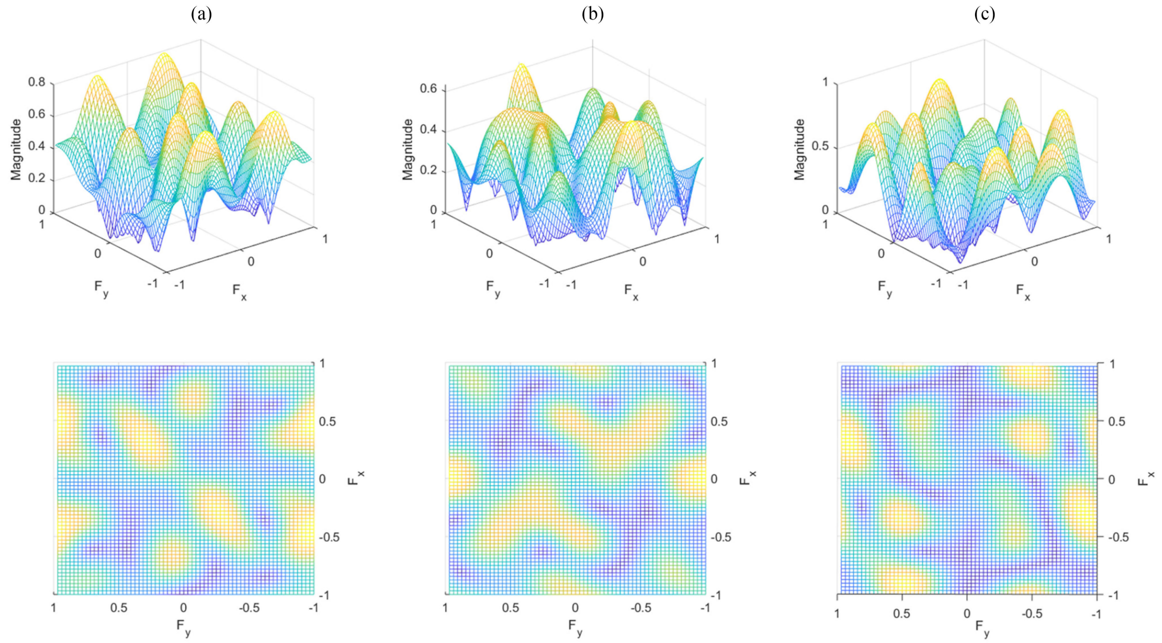

Fig. 13. Frequency responses of the clustered kernel centroids. The bottom figure in each column is the rotated version of the top figure.

However, even for the final layer, there are 18 kernels in total. It is time consuming to investigate all these kernels individually. There are two ways to simplify this task. The first is only to investigate some of them, for example, three of all the 18 filters. But this way renders the potential of missing some interesting filters. The second way is to cluster these filters into several categories and investigate the representatives (centroids) of these categories. By this way, we might not be able to faithfully analyze the original filters, but it can grossly cover all filters. Therefore, by flattening each kernel into a vector, we cluster these kernels via K-means method with metric of Euclidean distance [44], to partition them into 3 categories and study their centroids. Fig. 13 shows the frequency responses of the centroids, after reshaping the centroids into same dimension as the original kernels.

From these responses, especially shown in the bottom figures, it is manifest that the kernels are potentially centrosymmetric. They might not be strictly high- or band-pass filters, however, they are obviously not low-pass filters. Along a specific direction, most of them can be regarded as high-pass filters. This might suggest that the extractor network is by “sharpening” the watermarked image to extract the buried watermark information. However, if the watermarked image undergoes a low-pass filtering attack (the high frequency components get severely modified), this might counter the effectiveness of extraction.

Fig. 14 shows the failed case of JEPG compression. JPEG standard is known to retain the low spatial frequency components and modify high spatial frequency components. Fig. 14(f) illustrates the per pixel difference between the original image and image restored from saved JPEG image file. Note to better contrast the dissimilarity before and after JPEG algorithm modulation, the pixels are rescaled into $[0,\ 1]$ and applied a histogram equalization. The result shows that the network seems to be trained to blend the watermark into high components of images, which results in the vulnerability to high frequency component modulations. We will further study more elegant network architectures to embed watermark operations to improve the robustness to such attacks.

B. Blending Operation

There might exist other ways to embed watermark information into images by considering the characteristic of available neural network operations, such as convolution, pooling, etc. For simplicity, in this work we only utilize a direct blending operation to embed watermark into image. However, qualified watermarking requires a tactical fusing of image and watermark, to make the blending more suitable to the requirement of watermarking, some tricks are employed here. In the following, we discuss more findings about the blender module to seek deeper understandings of the blending operation.

Here we only discuss the attention mechanism which allows the network to automatically learn a more suitable way to fuse them together. The attention mechanism is realized by allocating two variable matrices or masks, w_s and w_e, with the same size as images (or watermark). They are initialized constantly and tuned during the training process respectively. Fig. 15(a) illustrates the perspective that is interpreted in a neural network manner. For a given pixel, the dashed box bounds an individual simple neural network without hidden layer. The input to the network is an individual pixel value $p_{i,j,k}$, and the output is the weighted value $o_{i,j,k}$, for all channels k across the same position (i,j). Regardless of the triviality ($o_{i,j,k} = w_{i,j}\ p_{i,j,k}$), $w_{i,j}$can be trained by error back-propagation. Fig. 15(b) display the pattern of learned w_e. It is interesting to see that these weights get tuned and their final values also favour the watermark

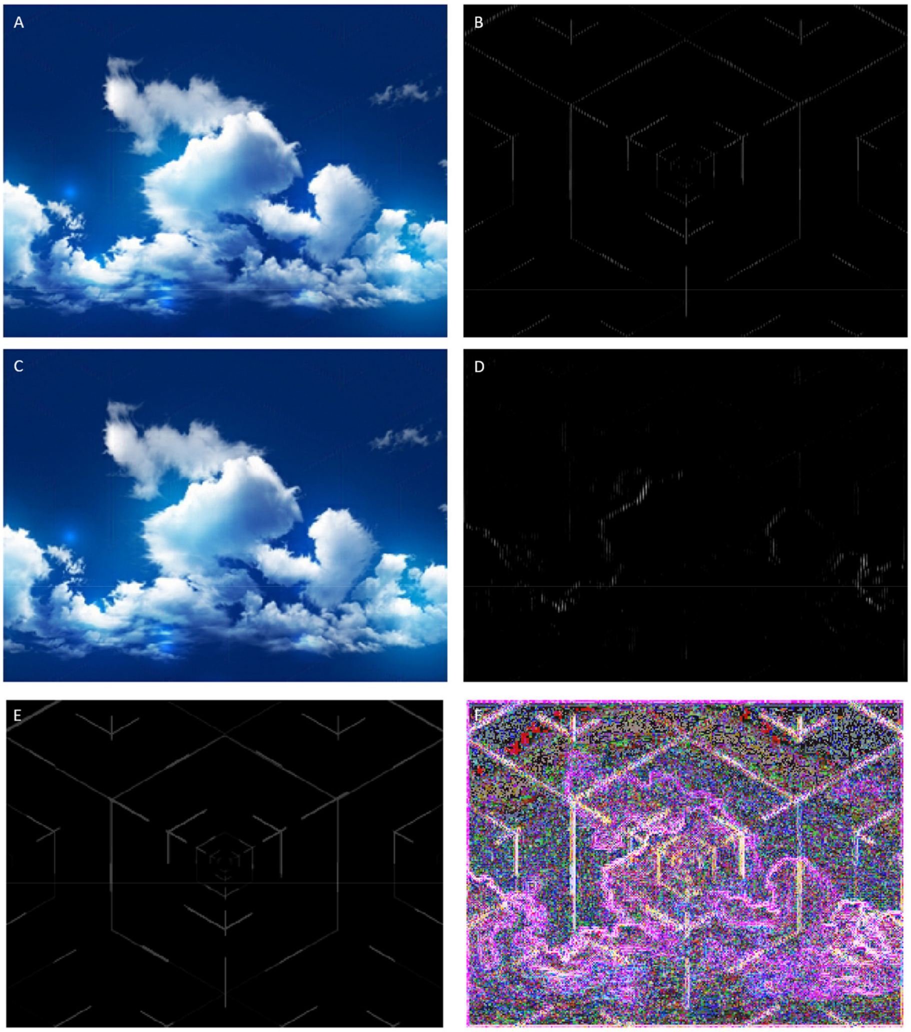

Fig. 14. The failed case of JEPG compression. (a) Intact watermarked image, (b) Watermark extracted from intact watermarked image, (c) Watermarked image reloaded from saved JPEG format image, (d) Watermark extracted from reloaded watermarked image, (e) Original watermark, and (f) Pixel value difference image between intact watermarked image and reloaded watermarked image (undergoes a rescale and histogram equalization).

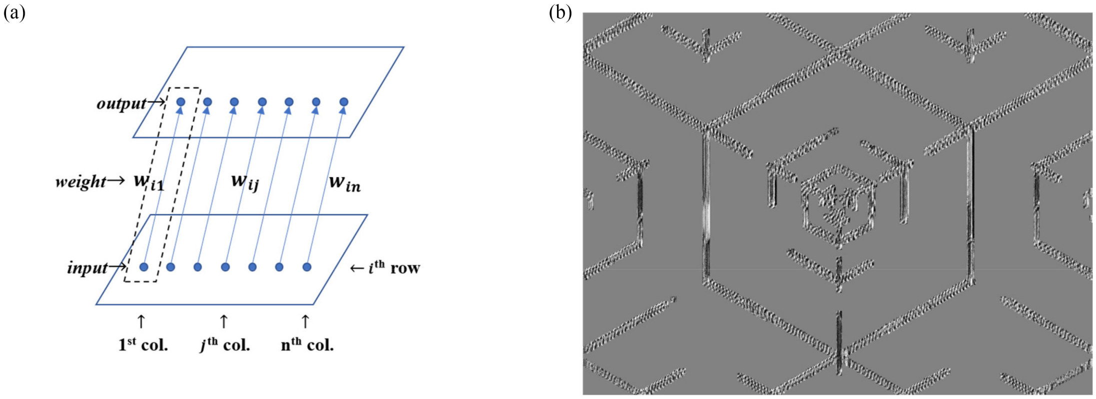

Fig. 15. Attention mechanism processing framework. (a) Attention mechanism realized by auto-tuned weighting, and (b) Learned pattern of w_e (The weights themselves undergo a rescale to $[0,\ 1]$ and histogram equalization for better displaying).

pattern. It should also be noted that, the learned pattern tends to be position dependent, which might incur the vulnerability to certain attack such as rotation. There is obviously pixel position displacement between the original watermarked image and the modified watermarked image. This is another our subsequent research of failure in extracting the embedded watermark.

VI. Conclusion

In this paper by considering the paradigm of utilizing DNN and its accomplishments, we proposed a generalized DNN approach for digital watermarking analysis. By constructing a DNN to suit the problem and training it on a set of images, the experimental results on test images revealed the potential of the proposed method. The subjective and objective assessments both demonstrated the practicality and economy of this proposed approach. We addressed aspects such as generalization, capacity, and complexity of the method, and pointed out the future research directions to mitigate the current limitations. Finally, we discussed traits of neural networks for specific applications. To the best of our knowledge, we are the first to conduct utilizing DNN in a general way for digital watermarking, and the preliminary achievements can provide certain guidance for further research in this thread.

In the future work, we will further study more elegant network architectures and other ways to embed watermark operations to improve the robustness to the attacks. We also will extract the embedded watermark in the original watermarked image and the modified watermarked image, especially in large-scale electronic medical record images.

References

[1] H. Ma, S. Yu, M. Gabbouj, and P. Mueller, “Guest Editorial special issue on multimedia big data in Internet of Things,” *IEEE Internet Things J.,* vol. 5, no. 5, pp. 3405–3407, Oct. 2018.

[2] L. Xiao and Z. Wang, “Internet of Things: A new application for intelligent traffic monitoring system,” *J. Netw.,* vol. 6, no. 6, pp. 87–894, Jun. 2011.

[3] C. Chalifoux, “Smart cameras in a manufacturing environment: Today and in the future,” *Quality,* vol. 57, no. 1, pp. 6–7, Jan. 2018.

[4] X. Wang, G. Luo, and L. Tian, “Application of hyperspectral image anomaly detection algorithm for Internet of Things,” *Multimedia Tools Appl.,* vol. 78, no. 5, pp. 5155–5167, 2019.

[5] A. Nortjé, “The admissibility of electronic documents in court proceedings.” *SchoemanLaw Inc.*, Accessed: Feb. 1, 2020. [Online]. Available: https://www.polity.org.za/article/ the-admissibility-of-electronic-documents-in-court-proceedings-2016-06-21

[6] S. Juergen, *Digital Watermarking for Digital Media*. Hershey, PA: Information Science Pub., 2005.

[7] N. Agarwal, A. Singh, and P. Singh, “Survey of robust and imperceptible watermarking,” *Multimedia Tools Appl.,* vol. 78, no. 7, pp. 8603–8633, 2019.

[8] M. A. Nematollahi, C. Vorakulpipat, and H. G. Rosales, *Digital Watermarking: Techniques and Trends (Springer topics in Signal Processing, Volume 11.)*, Singapore: Springer, 2017.

[9] A. Anand and A. K. Singh, “Watermarking techniques for medical data authentication: A survey,” *Multimedia Tools Appl.,* Apr. 2020, doi: 10.1007/s11042-020-08801-0.

[10] J. Abraham, “Gray scale image watermarking using LSB modification,” *Int. J. Adv. Res. Comput. Sci.,* vol. 2, no. 5, pp. 441–443, Sep./Oct. 2011.

[11] X.-C. Yuan, C.-M. Pun, and C. L. Chen, “Geometric invariant watermarking by local Zernike moments of binary image patches,” *Signal Process.,* vol. 93, no. 7, pp. 2087–2095, Jul. 2013.

[12] A. Pal and S. Roy, “A robust and blind image watermarking scheme in DCT domain,” *Int. J. Inf. Comput. Secur.,* vol. 10, no. 4, pp. 321–340, Jan. 2018.

[13] J. Jeswani and T. Sarode, “An improved blind color image watermarking using DCT in RGB color space,” *Int. J. Comput. Appl.,* vol. 92, no. 14, pp. 50–56, Mar. 2014.

[14] S. Rani and S. Kumari, “Watermarking using DWT and PCA,” *Int. J. Adv. Res. Comput. Sci.,* vol. 6, no. 6, pp. 117–120, Jul. /Aug. 2015.

[15] D. Savakar and S. Pujar, “Digital image watermarking using DWT and FWHT,” *Int. J. Image, Graph. Signal Process.,* vol. 11, no. 6, pp. 50–67, Jun. 2018.

[16] P. Singh and R. Chadha, “A survey of digital watermarking techniques, applications and attacks,” *Int. J. Eng. Innov. Technol.,* vol. 2, no. 9, pp. 165–175, 2013.

[17] A. Shareef and R. Fadel, “An approach of an image watermarking scheme using neural network,” *Int. J. Comput. Appl.,* vol. 92, no. 1, pp. 44–48. Apr. 2014.

[18] M. Islam and A. Roy, “Neural network based robust image watermarking technique in LWT domain,” *J. Intell. Fuzzy Syst.,* vol. 34, no. 3, pp. 1691–1700, Mar. 2018.

[19] S.-M. Mun *et al.*, “Finding robust domain from attacks: A learning framework for blind watermarking,” *Neurocomputing*, vol. 337, pp. 191–202, Apr. 2019.

[20] P. Wei, W. Zhang, H. Yang, and D. Yang, “A novel blind digital watermark algorithm based on neural network and chaotic map,” in *Proc. Int. Conf. Neural Inf. Process.*, Hong Kong, China, 3-6 Oct. 2006, pp. 243–250.

[21] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, “HiDDeN: Hiding data with deep networks,” in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, 8-14 Sep. 2018, pp. 682–697.

[22] I. Hamamoto and M. Kawamura, “Neural watermarking method including an attack simulator against rotation and compression attacks,” *IEICE Trans. Inf. Syst.,* vol. E103.D, no. 1, pp. 33–41, Jan. 2020.

[23] A. K. Singh, B. Kumar, S. K. Singh, S. Ghrera, and A. Mohan, “Multiple watermarking technique for securing online social network contents using back propagation neural network,” *Future Gener. Comput. Syst.,* vol. 86, pp. 926–939, Sep. 2018.

[24] S. M. Mun, S. H. Nam, H. Jang, D. Kim, and H. K. Lee, “Finding robust domain from attacks: A learning framework for blind watermarking,” *Neurocomputing,* vol. 337, pp. 191–202, Apr. 2019.

[25] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature,* vol. 521, no. 7553, pp. 436–444, May 2015.

[26] I. Goodfellow, *Deep Learning (Adaptive computation and Machine Earning Series.)*, Cambridge, MA: MIT Press, 2016.

[27] I. J. Goodfellow *et al.*, “Generative adversarial networks,” Jun. 2014, *arXiv:1406.2661v1*.

[28] V. Mnih *et al.*, “Playing atari with deep reinforcement learning,” Dec. 2013, *arXiv: 1312.5602v1*.

[29] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, 7-13 Dec. 2015, pp. 1520–1528.

[30] A. Vaswani *et al.*, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[31] S. Chaudhari, G. Polatkan, R. Ramanath, and V. Mithal, “An attentive survey of attention models,” Apr. 2019, *arXiv:1904.02874*.

[32] A. Mamaev, “Flowers recognition.” Accessed: Jan. 1, 2020. [Online]. Available: https://www.kaggle.com/ alxmamaev/ flowers-recognition

[33] M. Abadi *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous distributed systems,” Mar. 2016, *arXiv:1603. 04467v2*.

[34] C.-R. Piao, S. Cho, and S.-S. Han, “Color image watermarking algorithm using BPN neural networks,” in *Proc. 13th Int. Conf. Neural Inf. Process.*, Hong Kong, China, Oct. 3-6, 2006, pp. 234–242.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[36] G. Kalra, R. Talwar, and H. Sadawarti, “Adaptive digital image watermarking for color images in frequency domain,” *Multimedia Tools Appl.*, vol. 74, no. 17, pp. 1–21, Mar. 2014.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman,“The pascal visual object classes (voc) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[38] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, “Pruning convolutional neural networks for resource efficient inference,” in *Proc. 5th Int. Conf. Learn. Representations.*, Toulon, France, pp. 24–26 Apr. 2017.

[39] M. K. Samee and J. Gotze, “Increased robustness and security of digital watermarking using DS-CDMA,” in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Giza, Egypt, 15-18 Dec. 2007, pp. 185–189.

[40] J. J. Yu, F. Wang, and L. T. Zhao, “A low power and complexity watermarking algorithm in DS-CDMA communication,” in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, Chengdu, China, 9-11 Jul. 2010, pp. 547–551.

[41] J. Guo and M. Potkonjak, "Watermarking deep neural networks for embedded systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, San Diego, CA, 5-8 Nov. 2018, pp. 1–8.

[42] D. Li *et al.*, "A novel CNN based security guaranteed image watermarking generation scenario for smart city applications," *Inf. Sci.*, vol. 479, pp. 432–447, Apr. 2019.

[43] T. Wang and F. Kerschbaum, "Attacks on digital watermarks for deep neural networks," in *Proc. ICASSP IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, United Kingdom, 12-17, May 2019, pp. 2622–2626.

[44] J. Wu, *Advances in K-means Clustering A Data Mining Thinking*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

Weiping Ding (Senior Member, IEEE) received the Ph.D. degree in computation application from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. He was a Visiting Scholar with the University of Lethbridge, Lethbridge, AB, Canada, in 2011. From 2014 to 2015, he was a Postdoctoral Researcher with the Brain Research Center, National Chiao Tung University, Hsinchu, Taiwan. In 2016, He was a Visiting Scholar with the National University of Singapore, Singapore. From 2017 to 2018, he was a Visiting Professor with the University of Technology Sydney, Ultimo, NSW, Australia. He has authored or coauthored more than 100 research peer-reviewed journal and conference papers, including IEEE T-FS, T-NNLS, T-CYB, T-SMCS, T-BME, T-II, T-ETCI, and T-ITS. His main research interests include data mining, granular computing, evolutionary computing, machine learning, and big data analytics. He is currently the Chair of IEEE CIS Task Force on Granular Data Mining for Big Data. He is a Member of Senior IEEE, IEEE-CIS, ACM, CCAI and Senior CCF. He is a Member of Technical Committee on Soft Computing of IEEE SMCS, on Granular Computing of IEEE SMCS, and on Data Mining and Big Data Analytics of IEEE CIS. He is currently with the Editorial Advisory Board of *Knowledge-Based Systems* and Editorial Board of *Information Fusion*, *Neurocomputing* and *Applied Soft Computing*. He is an Associate Editor for IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Information Sciences*, *Swarm and Evolutionary Computation*, IEEE ACCESS and *Journal of Intelligent & Fuzzy Systems*, and Co-Editor-in-Chief of *Journal of Artificial Intelligence and System*. He is the Leading Guest Editor of Special Issues in several prestigious journals, including IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Information Fusion*, *Information Sciences*, and *Applied Soft Computing*. He has delivered more than 20 keynote speeches at international conferences and has Co-Chaired several international conferences and workshops in the area of data mining, fuzzy decision making, and knowledge engineering.

Yurui Ming received the Ph.D. degree in artificial intelligence from the University of Technology Sydney, Ultimo NSW, Australia, in 2020. He was a Software Engineer with Telecommunication companies, design and implementation of protocol stacks for computer networks. His current research interest focuses on applying variant neural networks especially deep ones to analyze electroencephalogram (EEG) data.

Zehong Cao (Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Northeastern University, Shenyang, China and The Chinese University of Hong Kong, Hong Kong, and the Ph.D. degree in information technology from the University of Technology Sydney (UTS), Ultimo NSW, Australia. He is a Lecturer (a.k.a. Assistant Professor) with the Discipline of Information and Communication Technology, School of Technology, Environments and Design, College of Sciences and Engineering, University of Tasmania, Hobart, TAS, Australia, and an Adjust Fellow with the School of Computer Science, Faculty of Engineering and Information Technology, UTS. His research interests include the brain computer interface, computational intelligence, and machine learning. He is currently focusing on the capacity of Human-In-The-Loop machine learning and applications. He is an Associate Editor for *Nature Scientific Data* (2019), Journal of *Journal of Intelligent and Fuzzy Systems* (2019) and IEEE ACCESS (2018–2019), and the Guest Editor of IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE (2019), *Swarm and Evolutionary Computation* (2019), and *Neurocomputing* (2018). He has authored or coauthored more than 40 papers in well known conferences, such as AAMAS, IJCNN, IEEEFUZZY, and top tier journals, such as IEEE TFS, TNNLS, TCYB, TSMC-S, TBME, TCDS, TITS, TII, TIA, IoT, IEEE or ACM TCBB, ACM TOMM, TOIT, Elsevier INS, NC, IJNS, NeuroImage and Nature: Scientific Data, of which two are ESI highly cited papers (2019–2020). He was awarded for the UTS Centre for Artificial Intelligence Best Paper Award, the UTS Faculty of Engineering and I.T. Publication Award, and the UTS President Scholarship.

Chin-Teng Lin (Fellow, IEEE) received the B.S. degree from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1986, the master's, and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989 and 1992, respectively. He is currently a Distinguished Professor of Faculty of Engineering and Information Technology, and the Co-Director of Center for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW, Australia. He is also invited as Honorary Chair Professor of Electrical and Computer Engineering, NCTU, and Honorary Professorship of University of Nottingham, Nottingham, U.K. He is the Co-Author of *Neural Fuzzy Systems* (Prentice-Hall), and the Author of *Neural Fuzzy Control Systems with Structure and Parameter Learning* (World Scientific). He has authored or coauthored more than 300 journal papers (Total Citation: 19,232, H-index: 64, i10-index: 243) in the areas of neural networks, fuzzy systems, brain computer interface, multimedia information processing, and cognitive neuro-engineering, including more than 120 IEEE journal papers. In 2005, he was elevated to be an IEEE Fellow for his contributions to biologically inspired information systems, and in 2012, was elevated International Fuzzy Systems Association Fellow. He was the recipient of the IEEE Fuzzy Systems Pioneer Awards in 2017. From 2011 to 2016, he was the Editor-in-Chief of IEEE TRANSACTIONS ON FUZZY SYSTEMS. He was also on the Board of Governors at IEEE Circuits and Systems (CAS) Society from 2005 to 2008, IEEE Systems, Man, Cybernetics (SMC) Society from 2003 to 2005, IEEE Computational Intelligence Society from 2008 to 2010, and the Chair of IEEE Taipei Section from 2009 to 2010. He was the Distinguished Lecturer of IEEE CAS Society from 2003 to 2005 and CIS Society from 2015 to2017. In 2018, he was the Chair of IEEE CIS Distinguished Lecturer Program Committee. From 2006 to 2008, he was the Deputy Editor-in-Chief of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-II. He was the Program Chair of IEEE International Conference on Systems, Man, and Cybernetics in 2005 and General Chair of 2011 IEEE International Conference on Fuzzy Systems.