End-to-End Automatic Image Annotation Based on Deep CNN and Multi-Label Data Augmentation

Xiao Ke, *Member, IEEE*, Jiawei Zou, and Yuzhen Niu, *Member, IEEE*

***Abstract*—Automatic image annotation is a key step in image retrieval and image understanding. In this paper, we present an end-to-end automatic image annotation method based on a deep convolutional neural network (CNN) and multi-label data augmentation. Different from traditional annotation models that usually perform feature extraction and annotation as two independent tasks, we propose an end-to-end automatic image annotation model based on deep CNN (E2E-DCNN). E2E-DCNN transforms the image annotation problem into a multi-label learning problem. It uses a deep CNN structure to carry out the adaptive feature learning before constructing the end-to-end annotation structure using multiple cross-entropy loss functions for training. It is difficult to train a deep CNN model using small-scale datasets or scale up multi-label datasets using traditional data augmentation methods; hence, we propose a multi-label data augmentation method based on Wasserstein generative adversarial networks (ML-WGAN). The ML-WGAN generator can approximate the data distribution of a single multi-label image. The images generated by ML-WGAN can assist in the reduction of the over-fitting problem of training a deep CNN model and enhance the generalization ability of the trained CNN model. We optimize the network structure by using deformable convolution and spatial pyramid pooling. We experiment the proposed E2E-DCNN model with data augmentation by the proposed ML-WGAN on several public datasets. The experimental results demonstrate that the proposed model outperforms the state-of-the-art automatic image annotation models.**

***Index Terms*—Image annotation, convolutional neural network, deep learning, generative adversarial networks, data augmentation.**

Manuscript received April 18, 2018; revised August 30, 2018, November 6, 2018, and January 9, 2019; accepted January 10, 2019. Date of publication January 25, 2019; date of current version July 19, 2019. This work was supported in part by the National Natural Science Foundation of China under Grants 61502105 and 61672158, in part by the Technology Guidance Project of Fujian Province under Grant 2017H0015, in part by the Natural Science Foundation of Fujian Province under Grant 2018J1798, in part by the University Production Project of Fujian Province under Grant 2017H6008, and in part by the Fujian Collaborative Innovation Center for Big Data Application in Governments. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jingdong Wang. *(Corresponding author: Yuzhen Niu.)*

X. Ke and Y. Niu are with the Fujian Key Laboratory of Network Computing and Intelligent Information Processing, College of Mathematics and Computer Science and the Key Laboratory of Spatial Data Mining and Information Sharing, Ministry of Education, Fuzhou University, Fuzhou 350116, China (e-mail: kex@fzu.edu.cn; yuzhenniu@gmail.com).

J. Zou is with the Fujian Key Laboratory of Network Computing and Intelligent Information Processing, College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China (e-mail: zjw_gary@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2019.2895511

I. Introduction

AN AUTOMATIC image annotation system automatically generates a number of keywords for an image to represent its visual content. The automatic image annotation method aims to learn the relationship between the underlying visual features and high-level image semantics from the labeled images in the training set. The method can predict the keywords for an unlabeled image.

According to the mathematics adopted by each annotation method, annotation methods can be classified into three categories, which are based on probability and statistics [1]–[4], graphs [5]–[12], and machine learning [13]–[26].

The most popular early image annotation methods are the probability and statistics-based methods. These methods first define the probability distribution between the image features and keywords before constructing a training set to solve the probability distribution parameters, and then the matching probability of each keyword is estimated according to the image features. These annotation methods can be extended and applied to large-scale datasets. However, the assumptions of the distributions between images and keywords do not accurately represent the distribution in the real image environment. Furthermore, the difference between the assumed and real distributions may limit the performances of these annotation methods in real applications.

Graph-based annotation methods have become popular among scholars in recent years. These methods use a graph structure to represent the similarities among images, but the construction process of the graph is complex. Once the graph is constructed, it is time consuming to traverse the graph and add new nodes. The complex construction process and time-consuming node insertion process make it difficult for these methods to deal with large visual data in real applications.

In recent years, machine-learning-based annotation methods have achieved some progresses in image annotation. These methods can be used to quickly annotate images if the models are well trained. However, current machine learning algorithms, either classification or regression, are mostly shallow structure algorithms. Furthermore, annotation methods based on deep learning for large amounts of images have difficulties relating to over-fitting and weak generalization abilities.

In this paper, we propose an end-to-end automatic image annotation model based on the deep convolutional neural network (E2E-DCNN). The E2E-DCNN transforms the image annotation problem into a multi-label learning problem. Traditional shallow machine-learning-based annotation models

usually treat the feature extraction and annotation as two independent tasks optimized by different technologies. These models are usually complicated in terms of structures and procedures. Different from traditional models, the proposed E2E-DCNN model uses a deep convolutional neural network (DCNN) structure to carry out adaptive feature learning, and then constructs the end-to-end annotation structure by using the multiple cross-entropy loss function. In combination with the data augmentation method proposed in this paper, the E2E-DCNN can simultaneously accomplish the automatic multi-label image annotation task by using only a single DCNN structure, and significantly improve the annotation performance.

For a learning-based method using a DCNN structure, it is necessary to use large-scale datasets for training to achieve good performance. Therefore, in this paper, we propose a multi-label data augmentation method based on Wasserstein generative adversarial networks (ML-WGAN) to scale up the number of good multi-label images in the existing small-scale image annotation datasets. Through training, the noise distribution gradually approximates the data distribution of a single multi-label image. Subsequently, the images generated by the generator of the ML-WGAN are used to complement the original multi-label images in the dataset. The proposed method can significantly scale up the size of a multi-label dataset, mitigate the risk of over-fitting when training a deep neural network, and enhance the generalization ability of an image annotation model from the data perspective.

We experiment the proposed E2E-DCNN model with the proposed ML-WGAN-based data augmentation method on three public datasets. The experimental results show that the model can improve the annotation robustness to low-frequency keywords, and the performance outperforms the state-of-the-art automatic image annotation models.

Our main contributions are summarized as follows.

1) We propose an end-to-end automatic image annotation model based on DCNN and multi-label data augmentation, which can significantly improve the performance of the multi-label image annotation. We optimize the network structure by using deformable convolution and spatial pyramid pooling to improve the scale-invariance of images, utilize the contextual information from the image, and enhance the transformation capability of the proposed model.
2) We propose a multi-label data augmentation method based on Wasserstein GAN (WGAN) to mitigate the risk of overfitting. It generates a fake image from the data distribution of a single multi-label image and achieves the consistency between a fake image and its keywords.

The rest of this paper is organized as follows. In Section II, we discuss the related works. The details of the presented ML-WGAN and E2E-DCNN are described in Section III. The experimental results are given in Section IV, and the paper is concluded in Section V.

II. Related Work

Automatic image annotation is similar to image classification in some ways, but the two essentially belong to different research fields. The methods and public datasets that automatic image annotation and image classification use are all different. The task of image classification means to output the category of the image according to its visual content. Mostly, a fixed or a few image major category labels are assigned to an image in the image classification task. The task of automatic image annotation generates multiple labeling keywords according to the visual content of each image. Therefore, there is no difference between the foreground and the background of the image for the automatic image annotation task. We need to label all the semantic contents involved in the image because every region may contain important semantic information. According to different situations, automatic image annotation needs to process hundreds, thousands or even more objects at a time, and learn all the annotated words at the same time. For example, Fig. 1 may give a few category labels, such as 'building', from the perspective of image classification. Meanwhile, from the perspective of automatic image annotation, the labels given by the dataset are as follows: 'building,' 'school,' 'trees,' 'bike,' 'shadow,' 'house,' and 'farm.' Obviously, some labels are not suitable as a category for the image classification task.

Fig. 1. Example image for image classification and automatic multi-label image annotation.

One challenge in image annotation is that the number of images per label is extremely unbalanced compared to the field of object recognition and image classification. Label distribution in the automatic image annotation task belongs to the long tail distribution. Therefore, it is very difficult to learn the labels that appear less frequently. On the contrary, the number of images per label in the image classification task is more balanced. Moreover, another challenge is that there are many obscure semantic labels in the image annotation than in the multi-label image classification, for example, 'handle,' 'old,' 'group,' 'view,' 'horizon,' and 'uniform.' On the contrary, the categories of image classification are more meaningful. Many labels are not suitable as a category for the image classification task.

In this section, we first describe the three categories of annotation methods, which are annotation methods based on probability and statistics, graphs, and machine learning. Because our data augmentation method is proposed based on WGAN, we then describe WGAN.

A. Annotation Methods Based on Probability and Statistics

The cross-media relevance model (CMRM) [1] first constructs a joint distribution of visual features and semantic

keywords, and then annotates an image by deducing a probability formula. As a follow-up improvement model for the CMRM, the continuous relevance model (CRM) [2], [3] uses a kernel-based estimation method in the probability distribution estimation of the underlying visual features, and models the semantic keywords by using the multiple Bernoulli distribution. Different from the discrete model, the CRM achieves a continuous estimation of the probability of generating the bottom visual features. Barnard *et al.* [4] first used the Gaussian mixture model to model the distribution of multiple keyword categories in the training set, and then used the expectation maximization algorithm to estimate the model parameters.

As the assumptions of the distributions between images and keywords cannot accurately represent the distribution in the real image environment, the performance of these annotation methods in real applications is limited.

B. Annotation Methods Based on Graph

Graph-based annotation methods are usually semi-supervised learning methods. In these methods, the similarities among images are represented by graph structures. If more training images are added, it is often necessary to retrain the model for a supervised learning method. However, a graph-based annotation method does not need to retrain the model. Wang *et al.* [5] proposed a graph-based semi-supervised learning framework that uses the priori information of the training images. However, the traditional graph-based semi-supervised learning methods only use the feature map to calculate the similarities among images, ignoring the label graph. Wang *et al.* [6] proposed a bi-directional graph algorithm. The bi-directional graph combines the characteristics of the feature and label graphs. Xu *et al.* [7] addressed the multi-label image classification problem by jointly considering the complementary nature of multimodal features and correlated the nature of labels. They formulated the problem as a semi-supervised label diffusion process on a unified bi-relational graph. Amiri *et al.* [8] constructed a special sub-graph based on the visual morphology of each feature, and then connected these special sub-graphs in a way to form a supergraph structure to perform the image annotation. Gao *et al.* [9], [10] proposed an optimal graph structure according to the label based on a part of image and various visual feature information. Ding *et al.* [11] proposed a novel context-aware multi-instance multi-label model to integrate the instance context and label context into a general framework. Lei *et al.* [12] proposed an automatic image annotation method by using the analysis of social diffusion records based on a common-interest model. Their approach relates the annotations to the diffusion graphs of images by generating different subgraphs and extracting features from them.

The construction process of a graph is complex. Once the graph is constructed, it is time-consuming to traverse the graph and add new nodes. It is difficult for graph-based methods to deal with big visual data in real applications.

C. Annotation Methods Based on Machine Learning

Annotation methods based on machine learning regard semantic keywords as category labels and classify them through the training of traditional supervised learning models. Makadia *et al.* [13] used multi-scale image features to search the image database before using the greedy label propagation mechanism to select the best category label from the images with similar characteristics. Guillaumin *et al.* [14] selected many images of the highest similarity in the training set, and then assigned different weights to the category labels of images according to the degrees of similarity computed by metric learning.Yang *et al.* [15] proposed a non-symmetric two-terminal support vector machine structure to explore image annotation from the perspective of multi-instance learning. Fu *et al.* [16] used the idea of the random forest algorithm to generate a stochastic tree for the labels of training images. They used the ranking learning algorithm to annotate the semantic neighborhood based on semantic similarity. Servajean *et al.* [17] proposed a set of data-driven algorithms by using active learning to train image annotators on how to disambiguate among automatically generated candidate labels. Li *et al.* [18] proposed a query-specific model for image annotation with missing labels via ranking-preserving low-rank factorization.

In recent years, deep learning methods have provided successes in pattern recognition and computer vision fields, including image classification, image recognition [27], and scene recognition. However, there are only a few image annotation methods based on deep learning. Work [19] utilizes deep learning in the preprocessing of images, which is not a direct application of deep learning in image annotation. Work [20] could only annotate an image with only one object, which is essentially a single-label classification problem. This cannot be used to carry out complex multi-label image annotation. Wu *et al.* proposed a diverse image annotation (DIA) model [21] and a diverse and distinct image annotation model based on GAN (D^2-GAN) [22]. Relevant and distinct tag subsets are interrelated to the image contents and semantically distinct to each other. Ke *et al.* [23] proposed a framework by using attribute discrimination annotation (ADA). Adapted stacked autoencoder (SAE) and local semantic propagation (LDE-SP) algorithms are used to annotate the high- and low- frequency images, respectively. To the best of our knowledge, multi-label image annotation based on deep CNN has not been studied extensively in the literature. Gao *et al.* [24] proposed an attention-based LSTM model with semantic consistency for video captioning. Song *et al.* [25] proposed an unsupervised deep hashing method which achieve the good performance on unsupervised video retrieval. Wang *et al.* [26] proposed a two-stream 3-D convNet fusion method which can recognize human actions in videos with arbitrary size and length.

In general, for the machine learning and deep learning methods mentioned above, some machine learning methods only have shallow structures and weak generalization abilities. They cannot achieve breakthroughs in the labeling effect. Deep learning methods, such as work [20], can deal with only single-label annotation. Some deep learning methods, such as LDE-SP and ADA [23], in recent years divide feature extraction and classification into two independent tasks, doing so makes the corresponding annotation model structure cumbersome.

Multi-label image annotation has practical applications in the fields of image retrieval and understanding. Therefore, this

paper proposes an end-to-end deep CNN model with multi-label data augmentation to solve the problem of complex multi-label/multi-object image annotation.

D. Generative Adversarial Networks (GAN) and Wasserstein GAN (WGAN)

GAN is a generative model proposed by Goodfellow *et al.* [28] in 2014 to estimate complex objective functions by using supervised learning. The goal of the generator, G, of GAN is to generate fake samples that are close to the true samples to deceive discriminator, D, of GAN. Meanwhile, the goal of D is to distinguish the fake samples generated by G from the real samples. Because samples generated by GAN can be deceptive, GAN is a good data augmentation method. The confrontation process of GAN can be expressed as follows:

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_G(z)}[\log(1-D(G(z)))], \quad (1)$$

where x represents the real data sample, $P_{data}(x)$ represents the true data distribution, z represents the noise input to G, $P_G(z)$ represents the data probability distribution generated by G, $D(x)$ represents the probability that D determines the input sample is real data, and $G(z)$ represents the fake sample data generated by G. The stronger the ability of D, the larger $D(x)$, the smaller $D(G(z))$, and the larger $V(D,G)$. Therefore, the goal of D is to maximize $V(D,G)$.

In the field of deep learning, some works combine GAN with CNN to train the models for visual tasks. Radford *et al.* [29] proposed a GAN based on a deep convolutional structure (DCGAN).

In practice, we sample m samples, $\{x^1, x^2, \ldots x^m\}$, from the real sample distribution $P_{data}(x)$, and sample m samples, $\{z^1, z^2, \ldots z^m\}$, from the sample distribution $P_G(z)$ generated by the G network. For D,

$$\tilde{V}_D = \frac{1}{m}\sum_{i=1}^{m}\log D(x^i) + \frac{1}{m}\sum_{i=1}^{m}\log(1-D(z^i)). \quad (2)$$

For G, $\tilde{V}_G$ can be transformed to

$$\tilde{V}_G = \frac{1}{m}\sum_{i=1}^{m}\log(1-D(z^i)). \quad (3)$$

In summary, by competing with D, G can generate images that are very close to the real images that meet the requirement of data augmentation. We can use the trained GAN network to expand the scale of a dataset significantly.

However, the DCGAN model has the following two issues:

1) The training of DCGAN is unstable, and the training processes of G and D need to be balanced. If D is trained too well, G is unable to update the gradient, resulting in a vanishing gradient and the generation of additional similar data. Conversely, if D is trained poorly, it may not be able to instruct G to update the gradient in the correct direction, so that G generates data with a low similarity or even invalid data.
2) In the training process of DCGAN, there is no index similar to the accuracy rate to indicate the progress of the training. Hence, occasionally when G is able to generate images that are similar to real images as for the human eye, the loss of D remains 0, indicating that D can easily discern images that are not real. This creates difficulties in judging when G and D reach equilibrium and controlling the training process by adjusting the learning rate.

In response to the above issues, Arjovsky *et al.* [30] proposed a new GAN model with the Wasserstein distance as a measurement. The average distance required for a possible joint distribution, γ, is computed as follows:

$$B(\gamma) = E_{(x_p,x_q)\sim\gamma}[\| x_p - x_q \|], \quad (4)$$

where x_p and x_q are sampled from the joint distribution γ, and x_p and x_q are a true data sample and a generated data sample, respectively.

The smallest average distance is the Wasserstein distance:

$$W(P,Q) = \min_{\gamma\in\Pi} B(\gamma), \quad (5)$$

where Π is the set of all possible joint distributions.

For the theoretical representation of the original GAN in Equation (1), the similarity metric of WGAN can be expressed as:

$$W(P_{data}, P_G) = \max_{D\in 1-Lipschitz}\{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[D(x)]\}, \quad (6)$$

where $1-Lipschitz$ represents the condition $\| D(x_1) - D(x_2) \| \leq K \| x_1 - x_2 \|$ when $K = 1$. It limits the influence speed that the input change has on the output change.

The losses of G and D in the WGAN become

$$\tilde{V}_D = \frac{1}{m}\sum_{i=1}^{m} D(x^i) - \frac{1}{m}\sum_{i=1}^{m}(D(z^i)), \quad (7)$$

and

$$\tilde{V}_G = -\frac{1}{m}\sum_{i=1}^{m}(D(z^i)), \quad (8)$$

respectively.

In contrast to Equations (2) and (3), the WGAN does not contain logarithm losses and changes in the sign of G. In addition, WGAN also adopts strategies to stable the training process, such as the removal of the sigmoid function of the final layer of the discriminator and does not use the momentum algorithm as in the Adam algorithm.

As shown in Fig. 2, as opposed to the Jensen-Shannon (JS) divergence, the Wasserstein distance reflects the distance between two distributions with no or negligible overlap. Therefore, the Wasserstein distance can also indicate the training progress and finally solve the difficulties in training and instability problems of the DCGAN.

In this paper, we chose the GAN as a tool for data augmentation while training the deep annotation model because we need to generate new samples steadily and gradually. As the WGAN

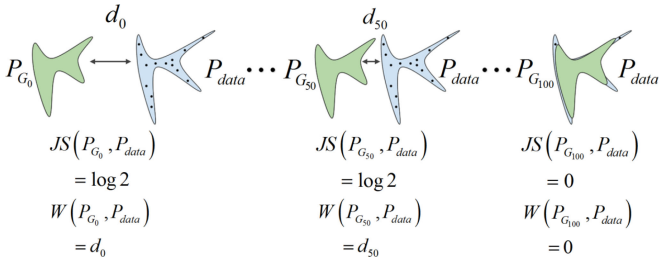


Fig. 2. Comparison between Wasserstein distance and JS divergence.

can solve the problem of the DCGAN, the data augmentation method proposed in this paper is based on the WGAN.

III. End-to-End Automatic Image Annotation Model

In this section, we first propose a multi-label data augmentation method based on WGAN (ML-WGAN) for multi-label image annotation. We then propose an end-to-end automatic image annotation model based on deep CNN (E2E-DCNN).

A. Multi-Label Data Augmentation Based on WGAN

In general, a complex model has a large Vapnik-Chervonenkis dimension; therefore, many training samples are required to minimize the penalty value. If the size of the training set is too small, the generalization ability of the model will deteriorate, resulting in over-fitting.

Data augmentation for small datasets is essential as a means of expanding the amount of data to train a complex deep learning model. In this paper, small-scale datasets and the deep CNN annotation model are used. The data augmentation method is required to solve the over-fitting problem and improve the annotation performance of our proposed method.

Traditional data augmentation methods can boost the performance of conventional visual tasks, such as image classification and recognition. For a single-label image with a unique foreground object, the traditional data augmentation methods can also expand the number of images to several times or even tens of times of that of the original images. Therefore, they can effectively enhance the generalization ability of the trained image annotation model for single-label images.

However, traditional data augmentation methods have some limitations for multi-label image classification tasks. The image annotation task generally involves multi-label images. For a multi-label image, all content in the image may contain keyword information; therefore, the concepts of background and foreground are not important for the annotation task. Furthermore, for the traditional data augmentation methods, the transformed images may not include image regions corresponding to all real objects (keywords), but they have the same keywords as the original images. Wrong keywords are then used to train the annotation model, leading to an unreliable model. As real objects of multi-label images usually cover almost the entire image, the inconsistency between the image content and keywords has a greater impact on the multi-label annotation than for the single-label annotation.

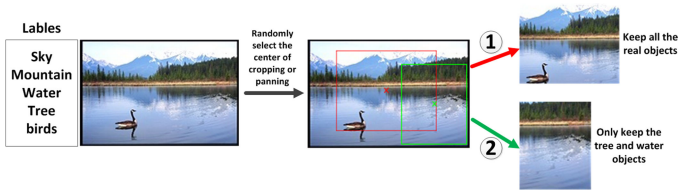


Fig. 3. Traditional geometric transformations may lead to inconsistency between image content and keywords.

As shown in Fig. 3, the original image has five keywords: 'sky,' 'mountain,' 'water,' 'tree,' and 'birds.' There are two possible cases for the transformed images. The first is that the transformed images are changed and retain the regions for each keyword. Such a transformation is effective and can achieve the purpose of data augmentation, as shown by the red box in Fig. 3. The other case is that the transformed images are changed and fail to retain regions for at least one keyword. As shown in the green box in Fig. 3, only the regions corresponding to the tree and water are retained. Therefore, the other three keywords assigned to the transformed image, including 'sky,' 'mountain,' and 'birds,' are wrong keywords that can mislead the training of the annotation model.

As shown in Fig. 3, the traditional data augmentation methods have limitations for the multi-label image augmentation. Therefore, other methods that overcome the defects of traditional methods are required to expand the annotation datasets.

Section II-D includes a description of using the WGAN to generate high quality new samples. Because a deep image annotation model is proposed in this paper, we require a significant expansion of the small-scale public datasets. WGAN often generates new images with similar characteristics to single-label images. Multi-label image annotation has practical applications in the fields of image retrieval and understanding; hence, the images in image annotation datasets are mostly multi-label images. Every two images in a common dataset usually have different numbers and types of labels. This is problematic for images generated by WGAN, because it is hard to determine image labels. Therefore, WGAN cannot be directly used for the multi-label data augmentation.

To facilitate the expansion of multi-label training samples, this paper proposes a multi-label data augmentation method based on WGAN, termed ML-WGAN. The process of ML-WGAN proposed in this paper is shown in Fig. 4. As can be seen from Equations (1), (6), and Fig. 3, the Wasserstein distance, indicating the training progress of WGAN, reflects the distance between the real data distribution $P_{data}(x)$ and data distribution $P_G(x)$ generated by generator G. As the training iteration progresses, the Wasserstein distance of $P_{data}(x)$ and $P_G(x)$ becomes increasingly similar, until $P_{data}(x)$ completely coincides with $P_G(x)$. In this paper, each time we selected a multi-label image in the common dataset, $P_{data}(x)$, the noise z input by G was only iteratively approximated by the distribution of that image. We set G to generate an image after every T rounds of iteration, and as the iteration progressed, the noise distribution approached the true distribution and the image generated by G became increasing similar to the original image. The

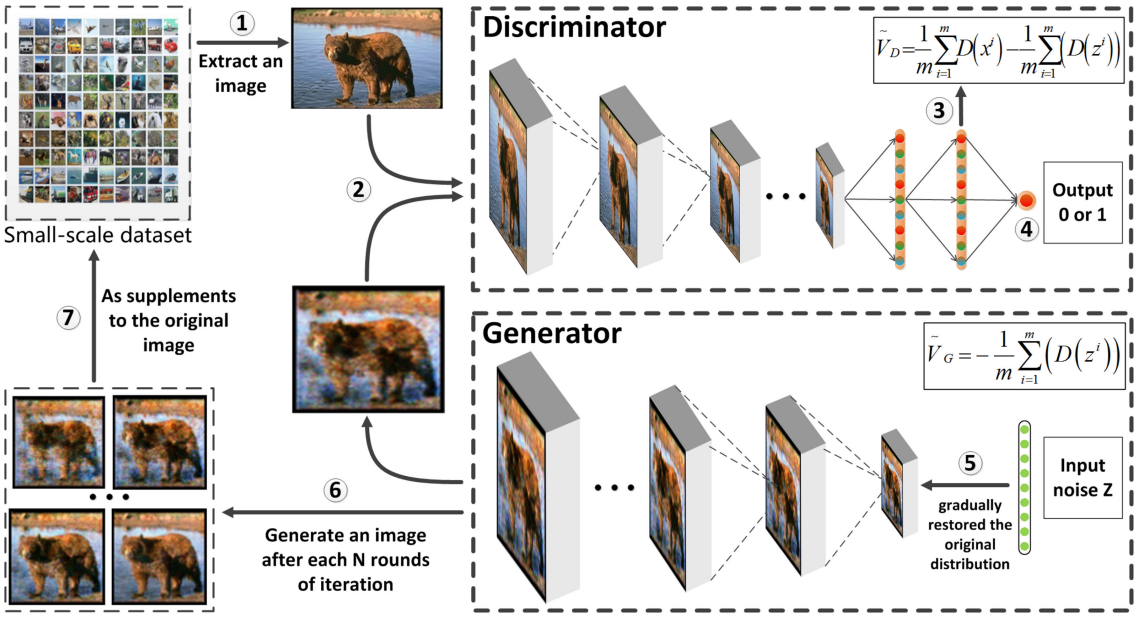


Fig. 4. Process of ML-WGAN: 1) A single image is selected as the original distribution each time. 2) The original image and images generated by G are input to D. 3) The Wasserstein distance is calculated to indicate the training progress. 4) D determines the authenticity of images. 5) G gradually restores the original distribution during the iteration. 6) After T rounds, the generator outputs an image. 7) Combining with the constrained traditional data augmentation methods, such as flip, noise, limited rotation and translation, ML-WGAN allows a significant scale up of the sizes of small datasets.

images were then output in the iterative process as extended data samples.

As the images only use one original image as the real data distribution $P_{data}(x)$, they all have the same number and type of labels, and have their own local differences while the overall distributions are similar. These images are used as complements to the original images. Therefore, ML-WGAN can augment data with multi-labels. Combining various constrained traditional data augmentation methods, ML-WGAN allows the scaling up of the sizes of small-scale datasets to larger orders of magnitude.

B. End-to-End Annotation Model Based on Deep CNN

Problem definition: Let $X = \{x_1, x_2, \ldots, x_N\}$ denotes a set of N training images in an image dataset $x_i \in \mathbb{R}^d$, and $Y = \{y_1, y_2, \ldots, y_M\}$ represents the dictionary of M possible annotation keywords. The automatic image annotation task can be represented as N pairs of mapping between each image and a subset of keywords, expressed as: $P = \{(x_1, Y_1), (x_2, Y_2), \ldots, (x_N, Y_N)\}$, $Y_i \subseteq Y$. For convenience, we denote Y_i as an M-dimensional vector $Y_i \in \{0, 1\}^M$. $Y_i^j = 1$ indicates that the ith image, x_i, is annotated with the keyword y_j, and $Y_i^j = 0$ indicates that the ith image, x_i, is not annotated with the keyword y_j.

In this paper, we transform the task of image annotation into a multi-label learning problem. According to the definition of image annotation, all keywords of each image in the dataset are transformed into a vector of M dimensions with 0 or 1. M is the total number of keyword categories in the dataset. For the jth element of vector Y_i that represents image x_i, if x_i is annotated with keyword y_j, $Y_i^j = 1$; otherwise, $Y_i^j = 0$.

Based on the above label processing of the dataset, we require an appropriate multi-label loss function to calculate the error between the predicted and true labels of each image correctly. The softmax function can usually only deal with single-label classification problems and cannot be used directly for multi-label learning. In this study, we used multiple cross-entropy loss functions as follows:

$$L = -\frac{1}{m}\sum_{n=1}^{m}\left[Y_n \log \hat{Y}_n + (1 - Y_n)\log\left(1 - \hat{Y}_n\right)\right], \quad (9)$$

where m is the number of samples, $\hat{Y}_n$ is the predicted value for the nth sample, and Y_n is the true value. $\hat{Y}_n = \sigma(s_n)$ where $\sigma(.)$ is the sigmoid function and s_n is the score vector computed by the last fully connected layer of the DCNN. L approaches 0 when $\hat{Y}_n$ is close to Y_n. Equation (9) is non-negative.

When the network performs back propagation, the gradient of parameter ω_j and bias b are only affected by error $\hat{Y}_n - Y_n$. When the error is large, the gradient is updated quickly. When the error is small, the gradient is updated slowly. This is a good property for SGD that can speed up the training process.

We used the sigmoid function to convert the score of each label to the probability in the last layer, and the multiple cross-entropy loss function to estimate the error of each dimension. Once the annotation model has been trained, unknown images can be quickly annotated. Correspondingly, the sigmoid function was used in the test phase as a binary classifier. Then k labels with the highest k probabilities of the unknown image were selected, and the keywords corresponding to these labels were annotated to the unknown image to complete the automatic image annotation process.

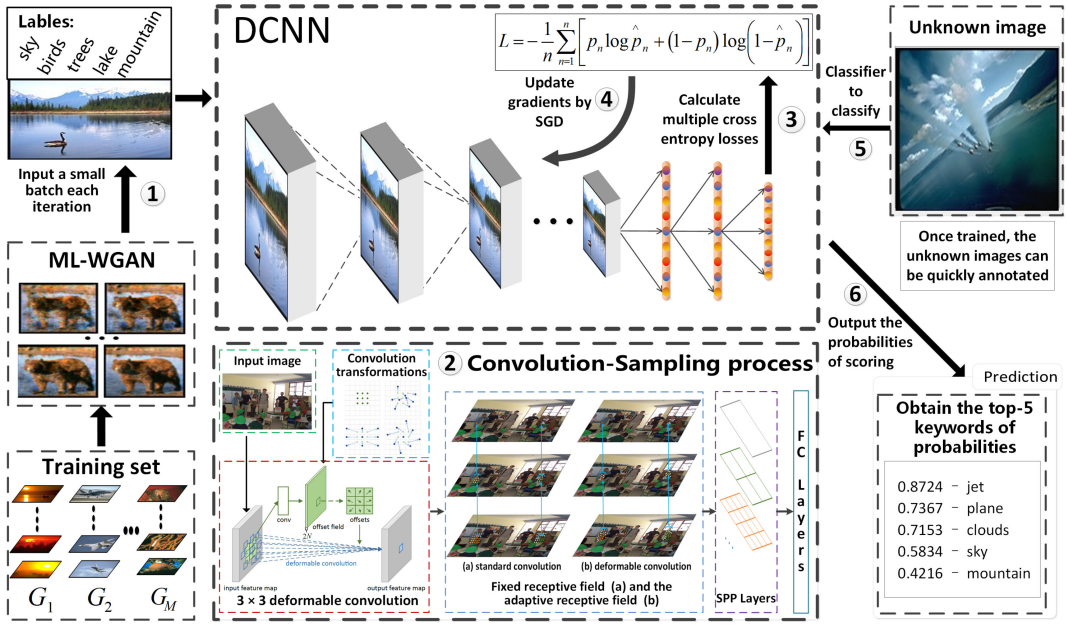


Fig. 5. Illustration of the proposed end-to-end deep convolutional neural network annotation model.

Different from traditional machine-learning-based image annotation methods, our proposed E2E-DCNN method does not need to extract the features during the training process manually. It adaptively learns the features and directly connects the classifiers in the last layer, unifying the feature learning and annotation. The entire process uses only one DCNN structure to accomplish the task of image annotation, providing a true end-to-end automatic image annotation.

Fig. 5 illustrates the proposed E2E-DCNN model. The proposed automatic image annotation based on E2E-DCNN is described in Algorithm 1. As shown in Algorithm 1, the proposed E2E-DCNN model has two phases, namely training and annotating. The training phase has two steps. In step 1, the training images pass through forward and backward computing processes. First, the training images pass through the convolutional and pooling layers to obtain the features maps. Second, the feature maps pass through the fully connected layers, obtaining a score vector for each image, which is then converted to a probability vector by using the sigmoid function. Third, the loss function computes the loss using the probability vector. Finally, the loss is back propagated to update the weights. In step 2, the first step is repeated until the loss is convergent. In this paper, we experimented with VGG-16 [31], ResNet-50 [32], and ResNet-101 [32] as the structure of our DCNN.

Automatic image annotation has some peculiarities and difficulties, such as unbalanced label distribution and obscure labels. In this case, mining the contextual information from the image is critical, especially for difficult labels. However, traditional automatic image annotation models do not take images context information into consideration. In view of these problems, we optimize the network structure by adding spatial pyramid pooling (SPP) method. There are two advantages of using the SPP: one is to improve the scale-invariance of images and the other is the image features rich in object context information can be

Algorithm 1: Automatic Image Annotation Based on E2E-DCNN

INPUT: Unknown image x_0, Training dataset X, an M-dimensional vector $Y_i \in \{0,1\}^M$ for each training image x_i, L is the number of DCNN layers;

OUTPUT: $\hat{Y}_0$ for x_0;

1. Training phase

1) For convolutional layer, input a small batch of training set X to i-th convolutional layer, doing i-th convolution operation:
$X^{(i+1)} = f(K^{i+1} \otimes X^{(i)} + b^{(i+1)}), i \in [1, L]$;

2) For pooling layer, do i-th subsampling operation:
$X^{(i+1)} = pool(X^{(i)})$;

3) For fully connected layer, the feature map of training set X is flattened into an M-dimensional vector:
$S = f(W_i X^{(i-1)} + b^{(i)})$;

4) Score is converted to probability by sigmoid function:
$\sigma(S) = \frac{1}{1+e^S}$;

5) Estimate loss and update weights by SGD:
$L = -\frac{1}{m}\sum_{n=1}^{m}[Y_n \log \hat{Y}_n + (1 - Y_n)\log(1 - \hat{Y}_n)]$;

6) The model minimizes the loss after T iterations:
$L^* = \arg\min L$

2. Annotating phase

1) For convolutional layer:
$x_0^{(i+1)} = f(K^{i+1} \otimes x_0^{(i)} + b^{(i+1)})$;

2) For pooling layer: $x_0^{(i+1)} = pool(x_0^{(i)})$;

3) For fully connected layer: $S_0 = f(W_i x_0^{(i-1)} + b^{(i)})$;

4) Compute scores: $\sigma(S_0) = \frac{1}{1+e^{S_0}}$;

5) Take the first k labels with the highest probability as the keywords of the unknown image x_0:
$\hat{w} = \arg\max_k[\sigma(S_0)]$, convert $\hat{w}$ to $\hat{Y}_0$.

beneficial to correspondence estimation, which is useful for the automatic image annotation task.

In a typical CNN structure, a fully connected layer is usually connected behind the convolution layer. The feature number of the fully connected layer is fixed; therefore, so the network input will be of a fixed size. In reality, the size of input image is always unable to meet the size required for the input and the usual way is to crop and warp. This will change the aspect ratio and the size of the input image, which means that the original image will be distorted. The SPP layer uses multiple pooling windows and different scales of the same image as the input to get the same length of pooling features, which can solve the above problem more successfully [33]. Furthermore, the automatic image annotation task needs to annotate multiple objects in an image; hence, it is difficult to determine the context relationship solely from pixel intensities (gray or RGB values). Therefore, the image features rich in object context information can be beneficial to correspondence estimation, especially for the ill-posed regions. Accordingly, the SPP module learns about the relationships between objects (such as houses) and their subregions (doors, windows, roofs, etc.) to incorporate hierarchical context information. In the current work, similar to [34], we design four scales adaptive fixed-size average pooling blocks, and 1×1 convolution is used to reduce the feature dimension. The low-dimensional feature maps are upsampled to the same size of the original feature map by using bilinear interpolation. The different levels of feature maps are concatenated as the final SPP feature maps. Connecting the SPP layer to the last convolution layer, the SPP layer produces a fixed size of the output, which is sent to the fully connected layer. That is to say, we import a new layer between the convolution layer and the fully connected layer, which can accept different sizes of the input but producing the same size of the output. In this way, we can avoid the need to have the same size of the network input so that an arbitrary input scale is accepted.

CNNs are inherently limited to model geometric transformations owing to the fixed geometric structures in their convolution kernels. However, a great challenge in visual recognition is how to model geometric transformations in the object scale, pose, viewpoint, and part deformation. In general, building augmented training datasets with geometric variations (or using transformation-invariant features) is a common way to handle it. These two methods are based on the assumption that geometric transformations are fixed and already known. Therefore, they perform worse in generalization to unknown or complex geometric transformations. Despite that CNNs have achieved significant success recently, the fact that they just sample feature maps in fixed locations makes them insufficient to model large and unknown geometric transformations. For example, receptive field sizes in the same CNN layer are equal. This is undesirable for the automatic image annotation task, because objects in different locations could have unequal sizes. If the receptive field size can be adaptive, it would highly promote robustness of CNNs especially in the automatic image annotation task.

Considering that most of the objects in the automatic image annotation task are non-rigid and the object scales vary, we optimize the network structure by adding the deformable convolution modules [35] to enhance the transformation capability of CNNs. Moreover, 2D offsets are added to the regular grid sampling locations in the standard convolution. Common geometric transformations can be achieved by sampling in this flexible way, and the offsets are designed to be learnable.

For example, assuming a standard convolution with the 3×3 kernel, the grid of the convolution kernel can be defined as R:

$$R = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\}. \tag{10}$$

For each point p_0 on the output feature map y,

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n). \tag{11}$$

In deformable convolution, the regular grid R is augmented with offsets. The $y(p_0)$ of the deformable convolution module can be modified to:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n). \tag{12}$$

As the offset is typically fractional, bilinear interpolation is used to determine the value of the sample point after offsetting.

Deformable convolution is manipulated by using the input feature map combined with the offset field of the same size. All these parameters can be learned from back propagation. The deformable convolution network can be obtained by integrating the deformable module into the CNNs. Deformable convolution can change locations of sampling points by learning the offsets of the corresponding sampling points. Therefore, the sample locations could be adaptive to different image contents. The capacity for locating objects is strengthened through the deformation module, especially for non-rigid objects. In contrast to standard convolution, receptive field sizes in the deformable convolution module can be adaptively changed to fit different image contents. The generalization capability of the model can be further improved.

In the annotating phase, an unknown image is input into the trained E2E-DCNN model and the forward computing process computes a probability vector using the sigmoid function for the image. Then k labels with the highest probability values are selected to annotate the unknown image.

The data augmentation method based on ML-WGAN proposed in Section III-A expands the data scale for the deep image annotation model and solves the over-fitting problem of a deep network. In addition, we adopted the following strategies in our optimization of the DCNN structure to improve the generalization ability.

1) The dropout [36] strategy, termed random inactivation, is used to alleviate the problem of over-fitting. As the name implies, neurons are closed in a random way. Each execution of dropout is equivalent to constructing a new network structure. The results of all the new structures are finally merged and comprehensive opinions are adopted to improve the generalization ability of the model.

2) Regularization also inhibits over-fitting. We use the L_2 regularization as follows:

$$\|\theta\|_2 = (w_1)^2 + \cdots + (w_j)^2, \tag{13}$$

where w_1, w_j are the parameters to be solved in the E2E-DCNN. We add an L_2 regular term to the model loss function and update Equation (9) as follows:

$$\tilde{L}(\theta) = L(\theta) + \frac{1}{2}\lambda\|\theta\|_2. \tag{14}$$

3) The training of a neural network is affected by the learning rate η when the gradient is updated. Usually η and its decay rate are set as hyperparameters. For different networks, data sizes and iterations, the optimal η and its decay rate differ. This requires a substantial practical experience to set them effectively. Therefore, this paper uses the RMSProp [37] algorithm so that the model dynamically and adaptively adjusts the value of the learning rate η during the training phase according to the update of the gradient.
RMSProp allows each layer to integrate the gradient learning situations of previous layers and dynamically adjusts the learning rate to assist in the avoidance of long training periods caused by learning too slowly or falling into local optima caused by learning too fast.
4) When training a DCNN with hundreds of layers, it is necessary to control the transmission of information carefully; otherwise, inappropriate parameter values of initialization may result in a decreased training effect. We introduce batch normalization [38] (BN) to reduce the dependence on the parameter values of initialization, ensure the network is smooth during the gradient transmission, and prevent the vanishing gradient.

IV. Experiments

In this section, we provide details of experiments with the proposed ML-WGAN and E2E-DCNN on three public annotation datasets. In section IV-A, we describe the experimental settings that include datasets, environment, parameters, and evaluation metrics. In section IV-B, we show the experimental results and analyses of ML-WGAN and E2E-DCNN. We also compare the E2E-DCNN with classic models in the field of image labeling and various improved algorithmic models proposed in recent years. Visual examples of data augmentation and image annotation are also provided.

A. Experimental Settings

1) Datasets: We chose three common public datasets from the field of image annotation: Corel5k [39], Espgame [14], and Iaprtc12 [40]. Corel5k is the most commonly used dataset of image annotation. It contains 5000 images of nature scenery and daily life, such as animals, landscapes, and humanities. The Espgame dataset contains 20,770 images taken from ESP collaborative image labeling tasks. The images cover a wide variety of subjects, such as personal photos, logos, and game screens. The Iaprtc12 dataset contains 19,627 images of humanities, social life, flora and fauna, architecture, natural landscapes, and modern urban life. Most image labels correspond to the textual representations within the images. Specific information of the three datasets is provided in Table I.

TABLE I
Information of Three Public Annotation Datasets

Name	Size	Number of labels	Size of training set	Size of testing set	Avg number of labels per image	Avg number of occurrences per label
Corel5k	5,000	260	4,500	500	3.4	58.6
Espgame	20,770	268	18,689	2,081	4.7	326.7
Iaprtc12	19,627	291	17,665	1,962	5.7	347.7

2) Environment and Parameters: The experimental environment for the experiments was as follows: the Linux Ubuntu 14.04 operating system, Tesla K80, graphics card, 128 GB memory, and Python programming language. The E2E-DCNN model was implemented using Caffe, a commonly used framework for deep learning. The ML-WGAN based data augmentation method was implemented using Pytorch and Keras libraries.

In the experiments, we used the 16-layers VGGNet, 50-layers and 101-layers ResNet, respectively, as the DCNN structures for the proposed E2E-DCNN annotation model. During the DCNN training, the maximum number of iterations was set to 100 K, initial learning rate was set to 0.005, and weight decay rate was set to 0.0001. We used the Xavier initialization to initialize the weights of the DCNN. The dropout ratio was set to 0.5 and the batch size was set to 64.

3) Evaluation Metrics: We strictly followed the guidelines for testing in the field of image annotation [13]. During the test phase, the test images were uniformly labeled with 5 keywords. For each class of labels, for example y_j, belonging to M classes, we calculated the precision P^j and recall R^j of the testing set, $j \in \{1, \ldots, M\}$. P^j and R^j are calculated as follows:

$$P^j = \frac{Precision(y_j)}{Prediction(y_j)}, \tag{15}$$

$$R^j = \frac{Precision(y_j)}{Ground(y_j)}, \tag{16}$$

where $Precision(y_j)$ represents the counts of label y_j that were correctly predicted, $Prediction(y_j)$ represents the counts of label y_j that were predicted, and $Ground(y_j)$ represents the counts of label y_j that manually annotated in the dataset. Then we averaged the precision and recall values of all M classes to define the average precision P and average recall R:

$$P = \frac{1}{M}\sum_{j=1}^{M} P^j, \tag{17}$$

$$R = \frac{1}{M}\sum_{j=1}^{M} R^j. \tag{18}$$

As both P and R are important evaluation metrics, a model has a good performance only when both P and R are high. Therefore, the calculation of the average harmonic value, F_1, is

required to reflect the comprehensive performance of the model. Furthermore, the number, N^+, of labels that have been correctly predicted at least once must simultaneously be counted. N^+ is the number of labels with a non-zero recall value, representing the coverage performance of the model on all the labels of the dataset. The calculation formulas of F_1 and N^+ are as follows:

$$F_1 = \frac{2 \times P \times R}{P + R}, \tag{19}$$

$$N^+ = \sum_{j=1}^{M} Sgn(R^j), \tag{20}$$

where $Sgn(x)$ is expressed as a symbolic function. For $x > 0$, $Sgn(x) = 1$; and for $x = 0$, $Sgn(x) = 0$.

B. Experimental Results

In section IV-B1, we discuss traditional data augmentation methods and their limitations. In section IV-B2 and IV-B3, we demonstrate the effectiveness and visual examples of ML-WGAN, respectively. In section IV-B4 and IV-B5, we demonstrate the effectiveness and visual examples of E2E-DCNN, respectively.

1) Traditional Data Augmentation Methods and Their Limitations: For visual tasks, such as image classification and image recognition, traditional data augmentation methods can be used to scale up datasets. Traditional data augmentation methods mainly use geometric and color transformations as follows:

1) Rotation: changes the orientation of the image content by rotating the image by a certain angle.
2) Scaling: reduces or enlarges the image by a certain percentage.
3) Translation: moves the image horizontally or vertically.
4) Flip: mirrors the image in the horizontal or vertical direction.
5) Contrast: changes the saturation and luminance channels of an image in the HSV color space using an exponential function, while keeping the hue channel unchanged.
6) Color jitter: performs the principal component analysis (PCA) operation on the color space to obtain three main direction vectors and eigenvalues, and then performs PCA jittering using the main direction vectors and eigenvalues to each pixel of the image.
7) Noise: adds noise to each channel in the color space. Commonly used noises are Gaussian and salt and pepper noises.

The geometric transformations to an image, including rotation, scaling, translation, and flip, do not change the value of each image pixel, but change the position of each pixel. These methods may result in the lose of parts of the image content. The contrast, color jitter, and noise transformations to an image do not change the position of each pixel, but they change the value of each pixel. These methods may degrade the quality of images [41], [42] and generate unnatural images. A combination of transformations is usually used to obtain more images and enlarge the dataset. In this way, the network model can learn more characteristics of the image invariance and increase the generalization ability of the model. Fig. 6 shows the original color images (first column) and images obtained by traditional data augmentation methods (the other five columns).

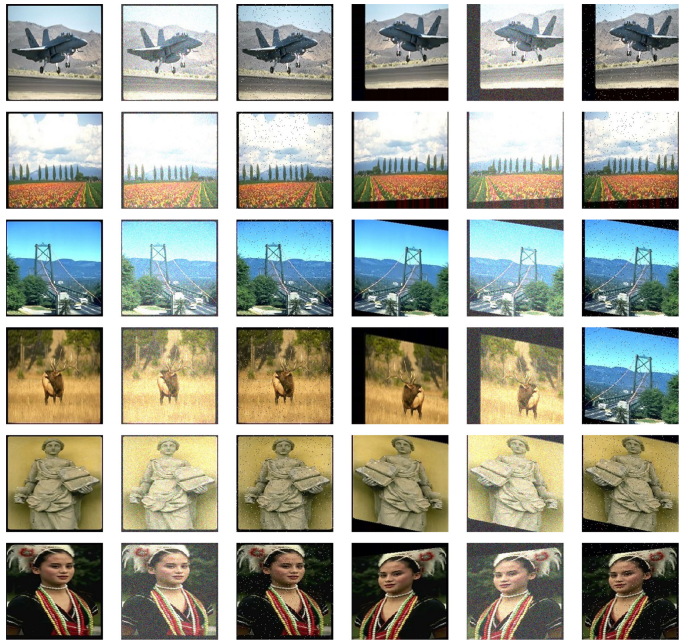


Fig. 6. Examples of traditional data augmentation methods. The images in the first column are the original images. The images in other columns are obtained by using traditional data augmentation methods.

TABLE II
EFFECTIVENESS OF ML-WGAN FOR DEEP MODEL TRAINING. THE BEST VALUES FOR EACH DCNN MODEL ARE FORMATTED IN BOLDFACE

Data Augumentation	CNN model	Corel5k			Espgame			Iaprtc12		
		Orig. size	New size	F1	Orig. size	New size	F1	Orig. size	New size	F1
None	VGG-16	4,500	4,500	0.26	18,689	18,689	0.24	17,665	17,665	0.28
	ResNet-50			0.21			0.23			0.27
	ResNet-101			0.16			0.20			0.21
Traditional	VGG-16	4,500	27,000	0.31	18,689	112,134	0.29	17,665	105,990	0.32
	ResNet-50			0.34			0.33			**0.40**
	ResNet-101			0.29			0.31			0.37
ML-WGAN	VGG-16	4,500	216,000	**0.36**	18,689	560,670	**0.31**	17,665	529,950	0.35
	ResNet-50			**0.41**			**0.37**			**0.40**
	ResNet-101			**0.44**			**0.38**			**0.42**

2) Effectiveness of ML-WGAN: When using traditional data augmentation methods, to avoid the loss of keyword information caused by geometric transformation, we constrained the range of translation and rotation angles to restrict the loss of image content to be within 10% of the whole image. The constraint increases the probability that the augmented images will be extremely similar, or even be the same, which cannot improve the performance of the DCNN model. Therefore, the constraint will limit the size of data expansion to a certain extent. As shown in Table II, the traditional data augmentation methods expand each dataset by 6 times.

We further expanded the datasets using the proposed ML-WGAN method. As shown in Table II, the final sizes of the Corel5k, Espgame, and Iaprtc12 were 48, 30, and 30 times of the original sizes, respectively. Table II shows the performance

of the proposed E2E-DCNN model using the structure of VGG-16, ResNet-50, and ResNet-101 with the original datasets, data augmented datasets by traditional methods, and data augmented datasets by ML-WGAN.

As shown in Table II, under the original dataset scale, all the DCNN models have a poor annotation performance on the three datasets. As the complexity of the DCNN model increases, from VGG-16 to ResNet-101, the annotation performance decreases. This indicates that the over-fitting problem becomes more serious for the more complex DCNN model because of insufficient training data.

With the traditional data augmentation, the training sets were scaled up to 6 times of the original sizes and the annotation performance was improved. However, the ResNet-101 was still not as effective as the ResNet-50 for the three datasets. This indicates that the current amount of data after expansion does not sufficiently enhance the generalization ability of the deeper annotation model, and there is still a certain degree of over-fitting.

After using the data augmentation method based on ML-WGAN proposed in this paper, which was also combined with the traditional data augmentation, the scale of each dataset was considerably expanded. The annotation performance also increased, and the most complex model, ResNet-101, achieved the best performance. For example, for ResNet-101, the F_1 values reached 0.44, 0.38, and 0.42 on Corel5k, Espgame, and Iaprtc12 datasets, respectively. Compared to the F_1 values of 0.16, 0.20, and 0.21 when using the original datasets, the increased ratios obtained using the proposed data augmentation method based on ML-WGAN were 175%, 90%, and 100% on the Corel5k, Espgame, and Iaprtc12 datasets, respectively.

In summary, the data augmentation method based on ML-WGAN proposed in this paper was successful in expanding the training data to a large scale, assisting the complex annotation model to solve the over-fitting problem, and enabling the deep annotation model to achieve a better performance.

3) Visual Examples of ML-WGAN: Fig. 7 shows examples of data augmentation results of the proposed ML-WGAN. The first column shows the original image, and the subsequent columns show the images generated by ML-WGAN after different numbers of iterations. As can be seen from the images in Fig. 7, as the number of iterations increase, the images generated by ML-WGAN are more similar to the original image. Finally, the original images are almost restored after 5,000 iterations. The images generated by ML-WGAN after 1,000 iterations, combined with the images generated by various constrained traditional data augmentation methods, were used to augment the original datasets to achieve data augmentation on a large sale.

4) Effectiveness of E2E-DCNN: The effectiveness of the ML-WGAN was verified in section IV-B2. This section shows the annotation performance of the proposed E2E-DCNN based on different DCNN structures on three public datasets after data augmentation by ML-WGAN. The experimental results are reported in Table III. As shown in Table III, ResNet-101 achieved the best performance values, except for the p-value

(a) Input (b) 100 (c) 500 (d) 1K (e) 2K (f) 3K (g) 5K

Fig. 7. Visual examples of ML-WGAN after different numbers of iterations.

TABLE III
PERFORMANCE COMPARISON OF E2E-DCNN BASED ON DIFFERENT DCNN STRUCTURES. THE BEST PERFORMANCE VALUES ARE FORMATTED IN BOLDFACE

Model	Corel5k				Espgame				Iaprtc12			
	P	R	F1	N+	P	R	F1	N+	P	R	F1	N+
VGG-16	0.30	0.45	0.36	167	0.38	0.27	0.31	236	0.39	0.33	0.35	253
ResNet-50	0.36	0.49	0.41	179	**0.44**	0.33	0.37	252	**0.45**	0.37	0.40	268
ResNet-101	**0.39**	**0.51**	**0.44**	**182**	**0.44**	**0.35**	**0.38**	**254**	0.44	**0.39**	**0.42**	**271**

on the Iaprtc12 dataset, which was smaller by 0.01 of the best p-value of 0.45.

ResNet, with its unique residual block structure, provides a depth for the network of up to hundreds of layers, while overcoming the difficulty of information transmission for training a deep neural network. We also used BN and other strategies introduced in section III-B to ensure the transmission of the gradient information fluent. As the complexity of the E2E-DCNN model increased, from VGG-16 to ResNet-101, the annotation performance increased. This demonstrates the advantages of DCNN for the task of multi-label image annotation.

All three public datasets used in the experiments are non-equilibrium datasets and such datasets often lead to the problem that the low-frequency labels cannot be correctly predicted, thereby affecting the recall R. Therefore, we verified that the E2E-DCNN is insensitive to unbalanced labels. In other words, E2E-DCNN should have a good annotation performance on both high and low-frequency labels. In this paper, we used ResNet-101, which had the best performance, as an example and selected 20 representative labels from each dataset, with ten high -and ten low-frequency labels, as shown in Figs. 8–10. The green histogram corresponds to the left vertical axis and indicates the label frequency. The red (circular) and blue

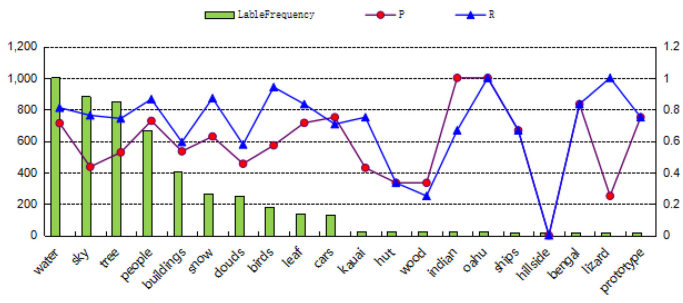


Fig. 8. Precision and recall of high-/low-frequency labels predicted by E2E-DCNN on Corel5k.

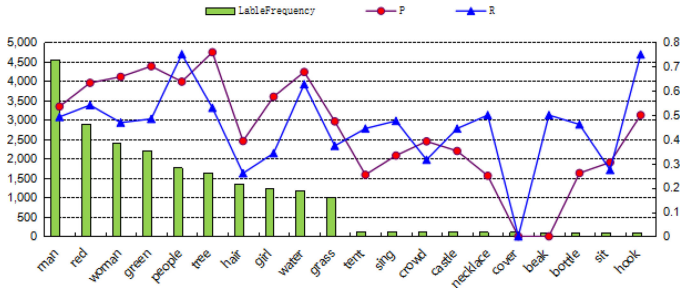


Fig. 9. Precision and recall of high-/low-frequency labels predicted by E2E-DCNN on Espgame.

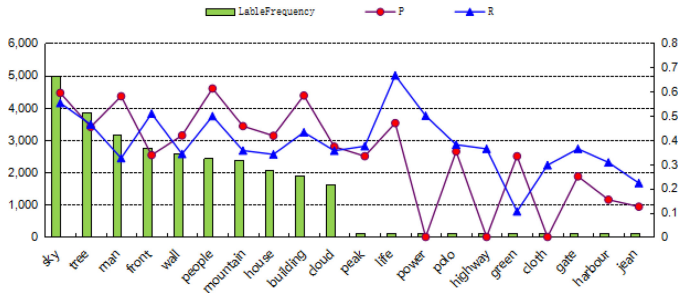


Fig. 10. Precision and recall of high-/low-frequency labels predicted by E2E-DCNN on Iaprtc12.

(triangle) line charts correspond to the right vertical axis, representing the precision P and recall R of the model, respectively. As shown in Figs. 8–10, E2E-DCNN provided good P and R for both high-and low-frequency keywords, especially the recall R of low-frequency labels, demonstrating that the E2E-DCNN model presented in this paper is not sensitive to unbalanced data. Therefore, E2E-DCNN is suitable for annotating images under unbalanced conditions.

We also compared the proposed E2E-DCNN model with various early classical models and some advanced models proposed in recent years in the field of image annotation. Here, we optimized the network structure by using deformable convolution and SPP. As shown in Table IV, the classical annotation model had low performance values, such as MBRM [3], GS [43], and JEC [13]. The advanced models proposed in recent years, such as LM3L [44], χ^2 Kernel [45], ANNOR-G [46], FFSS [47], MLRank [48], DIA [21], and D^2-GAN [22], achieved limited improvements in performance, especially for the performance metrics R and N^+. However, the E2E-DCNN model proposed

TABLE IV
COMPARISON BETWEEN E2E-DCNN AND SOME OTHER IMAGE ANNOTATION MODELS. THE BEST PERFORMANCE VALUES ARE FORMATTED IN BOLDFACE

Model	Corel5k				Espgame				Iaprtc12			
	P	R	F_1	N^+	P	R	F_1	N^+	P	R	F_1	N^+
MBRM [3]	0.24	0.25	0.24	122	0.18	0.19	0.18	209	0.24	0.23	0.23	223
GS [40]	0.30	0.33	0.31	146	-	-	-	-	0.32	0.29	0.30	252
JEC [13]	0.27	0.32	0.29	139	0.24	0.19	0.21	222	0.29	0.19	0.23	211
LM3L [41]	0.33	0.37	0.35	146	0.40	0.26	0.32	239	0.44	0.28	0.34	242
RNN [46]	0.31	0.34	0.32	149	-	-	-	-	0.33	0.31	0.32	255
χ^2Kernel [42]	0.31	0.39	0.35	153	0.38	0.21	0.27	214	0.42	0.24	0.31	239
ANNOR-G [43]	0.22	0.29	0.25	129	0.36	0.29	0.32	231	0.38	0.31	0.34	242
FFSS [44]	0.27	0.33	0.30	141	0.21	0.23	0.22	221	0.29	0.29	0.29	251
MLRank [45]	0.32	0.37	0.34	151	-	-	-	-	0.38	0.32	0.35	259
TagProp [14]	0.33	0.42	0.37	160	0.39	0.27	0.32	238	0.45	0.34	0.39	260
SDMKL [47]	0.25	0.38	0.30	158	-	-	-	-	-	-	-	-
NL-ADA [23]	0.32	0.40	0.36	173	0.36	0.21	0.27	251	0.42	0.30	0.35	280
DIA [21]	-	-	-	-	0.35	0.41	0.37	-	0.33	0.41	0.37	-
D^2-GAN [22]	-	-	-	-	0.35	**0.42**	0.38	-	0.33	**0.45**	0.40	-
E2E-DCNN	**0.41**	**0.55**	**0.47**	**192**	**0.48**	0.39	**0.43**	**261**	**0.48**	0.43	**0.45**	**285**

in this paper achieved significant improvements on many evaluation metrics in three datasets, especially in the precision P. For example, on Corel5k compared with NL-ADA [23], P increased from 32% to 41%, with a total 9% increase; R increased from 40% to 55%, with a total increase of 15%; F_1 increased from 36% to 47%, with a total increase of 11%; and N^+ increased from 173 to 192, with a total increase of 19. On EspGame, P increased from 36% to 48%, with a total increase of 12%. And on Iaprtc12, P increased from 42% to 48%, with a total increase of 6%.

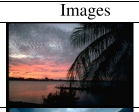
D^2-GAN [22] simulates the diversity and distinctiveness of the tags generated by human annotators. The image feature is incorporated into a determinantal point process model that also encodes the weighted semantic paths, from which a sequence of different tags is generated by sampling. The model achieved the best performance values on recall R, which reached 42% and 45% on Espgame and Iaprtc12 datasets, respectively. However, P and F_1 of D^2-GAN are 13% and 5% lower than our E2E-DCNN on the Espgame dataset, respectively, and 15% and 5% lower than our E2E-DCNN on the Iaprtc12 dataset, respectively. Although R of D^2-GAN is high, its P is low, resulting in the comprehensive F_1 being affected and to be low.

In summary, among all of the 12 best performance values, the E2E-DCNN achieved the 10 best values, and D^2-GAN [22] achieved the best values of R on EspGame and Iaprtc12. Therefore, the proposed automatic image annotation model based on E2E-DCNN outperforms the state-of-the-art annotation methods.

5) Visual Examples of E2E-DCNN: Table V shows some annotation instances of the E2E-DCNN proposed in this paper on three public annotation datasets. In the column showing the E2E-DCNN annotation, the labels highlighted in boldface indicate that the automatic annotation results of the E2E-DCNN model are consistent with the manual annotation; whereas the italicized labels indicate they are not annotated manually but can still reflect the content of the image.

For the listed instances, we did not select the exact predicted images from the results of model annotation. Instead, we se-

TABLE V
ANNOTATION INSTANCES OF THE E2E-DCNN MODEL

Datasets	Images	Manual annotation	E2E-DCNN annotation
Corel5k		sun, clouds, tree, sea	*sky*, *sunset*, **clouds**, **tree**, *water*
		sky, jet, plane, smoke	*clouds*, **jet**, **plane**, **sky**, *mountain*
EspGame		boat, bay, sea, bridge, wind, sail, mountain	**boat**, **sea**, **bridge**, *sky*, **mountain**
		building, school, trees, bike, shadow, house, farm	**buildings**, **trees**, *sky*, **school**, *tower*
Iaprtc12		balcony, car, church, front, house, people, side, street	**balcony**, *building*, column, **house**, **people**
		boy, grey, hat, kid, shirt, zip, vest	**boy**, *fog*, **hat**, *mountain*, **grey**

lected some images that could better reflect the advantages of the proposed model. It can be seen from Table V that although the labels of the image annotated by our model are different from the keywords from manual annotation, the keywords annotated by our model can reflect and describe the image content more accurately and appropriately. For example, for the first image in Table V, from the human visual point of view, 'sunset' annotated by our model is more appropriate than 'sun' from the manual annotation; the waterfront in the image may not be 'sea,' and 'water' is more appropriate; the manual annotation also omits the term 'sky' that can be directly derived from the image. For the second example in Table V, the manual annotation clearly missed two labels, 'mountain' and 'clouds,' which occupy a large area in the image.

The above results also illustrate some of the problems that exist in traditional manual annotation. Manual annotation may miss some annotations for objects in the image, and potential subjective differences exist when different people annotate the same image. Therefore, the proposed E2E-DCNN model can serve as a useful complement to the manual annotation of original images.

V. CONCLUSION

In this paper, we presented an end-to-end automatic image annotation method based on deep CNN and multi-label data augmentation. Traditional annotation models usually consider feature extraction and annotation as two independent tasks. We proposed an end-to-end automatic image annotation model based on deep CNN (E2E-DCNN) to formulate these two tasks as a multi-label learning problem. Because it is difficult to train a deep CNN model using small-scale datasets and scale up multi-label datasets using traditional data augmentation methods, we proposed a multi-label data augmentation method based on the Wasserstein GAN (ML-WGAN). The images generated by ML-WGAN can assist in reducing the over-fitting problem when training the E2E-DCNN model and enhance the generalization ability of the trained E2E-DCNN model. The experimental results of the proposed E2E-DCNN model with data augmentation by ML-WGAN on three public datasets demonstrated that the proposed E2E-DCNN model outperforms the state-of-the-art image annotation methods.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for the helpful comments and suggestions.

REFERENCES

[1] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 119–126.
[2] M. Wang, X. D. Zhou, J. Q. Zhang, H. T. Xu, and B. L. Shi, "Image auto-annotation via an extended generative language model," *J. Softw.*, vol. 19, no. 9, pp. 2449–2460, 2008.
[3] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple Bernoulli relevance models for image and video annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. II-1002–II-1009.
[4] K. Barnard and D. Forsyth, "Learning the semantics of words and pictures," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001, pp. 408–415.
[5] H. Wang, H. Huang, and C. Ding, "Image annotation using multi-label correlated green's function," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2010, pp. 2029–2034.
[6] H. Wang, H. Huang, and C. H. Q. Ding, "Image annotation using bi-relational graph of images and semantic labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 793–800.
[7] J. Xu, V. Jagadeesh, and B. S. Manjunath, "Multi-label learning with fused multimodal bi-relational graph," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 403–412, Feb. 2014.
[8] S. H. Amiri and M. Jamzad, "Efficient multi-modal fusion on supergraph for scalable image annotation," *Pattern Recognit.*, vol. 48, no. 7, pp. 2241–2253, 2015.
[9] L. Gao *et al.*, "Optimal graph learning with partial tags and multiple features for image and video annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4371–4379.
[10] J. Song *et al.*, "Optimized graph learning using partial tags and multiple features for image and video annotation," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 4999–5011, Nov. 2016.
[11] X. Ding *et al.*, "Multi-instance multi-label learning combining hierarchical context and its application to image annotation," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 1616–1627, Aug. 2016.
[12] C. Lei, D. Liu, and W. Li, "Social diffusion analysis with common-interest model for image annotation," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 687–701, Apr. 2016.
[13] A. Makadia, V. Pavlovic, and S. Kumar, "A new baseline for image annotation," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 316–329.
[14] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 309–316.
[15] C. Yang, M. Dong, and J. Hua, "Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2057–2063.
[16] H. Fu, Q. Zhang, and G. Qiu, "Random forest for image annotation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 86–99.
[17] M. Servajean, A. Joly, D. Shasha, J. Champ, and E. Pacitti, "Crowdsourcing thousands of specialized labels: A Bayesian active training approach," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1376–1391, Jun. 2017.
[18] X. Li, B. Shen, B.-D. Liu, and Y.-J. Zhang, "Ranking-preserving low-rank factorization for image annotation with missing labels," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1169–1178, May 2018.
[19] R. Kiros and C. Szepesvári, "Deep representations and codes for image auto-annotation," *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 917–925.

[20] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3460–3469.
[21] B. Wu, F. Jia, W. Liu, and B. Ghanem, "Diverse image annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6194–6202.
[22] B. Wu *et al.*, "Tagging like humans: diverse and distinct image annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7967–7975.
[23] X. Ke, M. Zhou, Y. Niu, and W. Guo, "Data equilibrium based automatic image annotation by fusing deep model and semantic propagation," *Pattern Recognit.*, vol. 71, pp. 60–77, 2017.
[24] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based LSTM and semantic consistency," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2045–2055, Sep. 2017.
[25] J. Song *et al.*, "Self-supervised video hashing with hierarchical binary auto-encoder," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3210–3221, Mar. 2018.
[26] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, "Two-stream 3-D convnet fusion for action recognition in videos with arbitrary size and length," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 634–644, Mar. 2018.
[27] X. Ke, L. Shi, W. Guo, and D. Chen, "Multi-dimensional traffic congestion detection based on fusion of visual features and convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2018.2864612.
[28] I. J. Goodfellow *et al.*, "Generative adversarial nets," *Proc. Adv. Neural Inf. Process. Syst.* , 2014, vol. 27, pp. 2672–2680.
[29] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.
[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv preprint arXiv:1701.07875, 2017.
[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations*, 2015, pp. 1–14.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[34] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5410–5418.
[35] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.
[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[37] T. Kurbiel and S. Khaleghian, "Training of deep neural networks based on distance measures using RMSProp," arXiv preprint arXiv:1708.01911, 2017.
[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[39] M. Chen, A. X. Zheng, and K. Q. Weinberger, "Fast image tagging," in *Proc. Int. Conf. Int. Conf. Mach. Learning*, 2013, pp. 1274–1282.
[40] Z. Feng, R. Jin, and A. Jain, "Large-scale image annotation by efficient and robust kernel metric learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2014, pp. 1609–1616.
[41] Y. Niu, Y. Zhong, W. Guo, Y. Shi, and P. Chen, "2D and 3D image quality assessment: A survey of metrics and challenges," *IEEE Access*, vol. 7, pp. 782–801, 2019.
[42] Y. Niu, Y. Yang, W. Guo, and L. Lin, "Region-aware image denoising by exploring parameter preference," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2433–2438, Sep. 2018.
[43] S. Zhang *et al.*, "Automatic image annotation using group sparsity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3312–3319.
[44] B. Hariharan, L. Zelnik-Manor, S. V. N. Vishwanathan, and M. Varma, "Large scale max-margin multi-label classification with priors," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 423–430.
[45] Y. Wang, H. Dawood, Q. Yin, and P. Guo, "A comparative study of different feature mapping methods for image annotation," in *Proc. Int. Conf. Adv. Comput. Intell.*, 2015, pp. 340–344.
[46] E. Kuric and M. Bielikova, "ANNOR: Efficient image annotation based on combining local and global features," *Comput. Graph.*, vol. 47, pp. 1–15, 2015.
[47] X. Zhang and C. Liu, "Image annotation based on feature fusion and semantic similarity," *Neurocomputing*, vol. 149, pp. 1658–1671, 2015.
[48] Z. Li, J. Liu, C. Xu, and H. Lu, "Mlrank: Multi-correlation learning to rank for image annotation," *Pattern Recognit.*, vol. 46, no. 10, pp. 2700–2710, 2013.
[49] C. Cui, J. Ma, T. Lian, X. Wang, and Z. Ren, "Ranking-oriented nearest-neighbor based method for automatic image annotation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 957–960.
[50] M. Jiu and H. Sahbi, "Nonlinear deep kernel learning for image annotation," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1820–1832, Apr. 2017.

Xiao Ke received the Ph.D. degree in artificial intelligence from Xiamen University, Xiamen, China, in 2011. He is currently an Associate Professor with Fuzhou University, Fuzhou, China. His research interests are related to multimedia, computer vision, pattern recognition, machine learning, and their relations with innovative technologies.

Jiawei Zou is currently working toward the M.S. degree with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His research interests are related to multimedia, pattern recognition, computer vision, deep learning, and their relations with innovative technologies.

Yuzhen Niu received the Ph.D. degree in computer science from Shandong University, Jinan, China, in 2010. She is currently a Professor with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. Her research interests include multimedia, computer vision, and artificial intelligence.