

Automatic Image Annotation Using Improved Wasserstein Generative Adversarial Networks

Jian Liu, Weisheng Wu

Abstract—In an image annotation model based on deep learning, the number of neurons in its output layer is proportional to the vocabulary of the annotation, i.e., the model structure changes with a change in the vocabulary, thereby reducing the accuracy of image annotation. To solve this problem, in this study a new annotation model combining the improved Wasserstein generative adversarial network (GAN) and word2vec was proposed. First, the tagged vocabulary was mapped to a fixed multidimensional word vector by word2vec. Second, a neural network model (GAN-IW) was constructed by using the generated confrontation network. It was observed that the number of neurons in the output layer was equal to the dimension of the multidimensional word vector and no longer relevant to the vocabulary. Finally, the model was tested for the Corel 5K and IAPRTC-12 image annotation datasets. Compared to the convolutional neural network regression method, the model accuracy, the recall rate, and the F1 value increased by 16%, 6%, and 9%, respectively, when the model was tested on the Corel 5K dataset. Compared to the two-pass K-nearest neighbor models, our model accuracy, recall rate, and F1 value were increased by 8%, 6%, and 4%, respectively, when the model was tested on the IAPRTC-12 dataset. The experimental results showed that the GAN-IW model can solve the problem of change in the number of output neurons with a change in the vocabulary and the number of labels annotated with each image is adaptive, making the results of model annotation more in line with the actual image annotation.

Index Terms—Automatic image annotation, Generative adversarial networks, Improved Wasserstein, Machine learning.

I. Introduction

WITH the rapid increase in the volume of image data, manually marking of images has become a cumbersome and undesirable task. For effective management and retrieval of images, we should move to automatic marking of the image content, thus enabling more efficient use of the abundant image information. At present, the main annotation method involves constructing an image annotation model through machine learning. By learning the potential connection between the image and its corresponding annotation, a keyword describing the content is added to the unknown image to realize its annotation.

The image annotation model based on machine learning is roughly divided into three categories: the generation model, the nearest neighbor model, and the discriminant model. The generation model first extracts the

image features and then calculates the joint probability of the image features and the image tags. Finally, the model calculates the probability of each tag according to the characteristics of the test images and determines the tags corresponding to the images. Representative methods are: Cross Media Relevance Model (CMRM) [1], Multiple Bernoulli Relevance Model (MBRM) and Sparse Kernel Learning Continuous Relevance Model (SKL-CRM)[2] .

The nearest neighbor model first finds a plurality of images similar to the predicted image based on some distance based on the image features and then determines the annotation of the predicted image based on the annotation of similar images. Representative methods are: 2PKNN (Two-Pass K-Nearest Neighbor) model [4], JEC (Joint Equal Contribution) model[3], and TagProp_ML (Tag Propagation Metric Learning) model [5]. The discriminant model considers the image label as a classification of the image; therefore, image annotation can be regarded as a multi-classification of the image and the label of the image is determined by the classification result of the image. The representative methods are: PAMIR (Passive-Aggressive Model for Image Retrieval) model [7], CBSA (Content-Based Soft Annotation) model[6], and ASVM-MIL (Asymmetrical Support Vector Machine-Based MIL Algorithm) model [8]. In recent years, deep learning has achieved good results from image classification. Therefore, the method of deep learning has gradually been applied to image annotation tasks. For example, in 2018, Li et al. [9] proposed combining the depth convolutional neural network (CNN) and the CNN-ensemble of classifier chains model of the integrated classifier chain. SSN-GAN was proposed by Gang et al. [24] to obtain colorful anime images with higher visual quality.

Compared with the traditional annotation methods, these models have achieved good results and significantly improved performance in image annotation. However, these deep learning annotation models have a common drawback, i.e., the number of model output layer neurons (or classifiers) is proportional to the vocabulary of annotation. This leads to two problems: 1) With the increase in the vocabulary of the dataset label, there is a proportional increase in the number of neurons in the output layer. When the vocabulary of the dataset is small, it has almost no effect on the model; however, in case of a dataset with a larger vocabulary, the number of neurons in the output layer of the model becomes very large, for example, in the Open Images dataset, the number of neurons exceeds 20,000. Thus, a large number of neurons in the output layer makes the designing of a reasonable neural network structure difficult and leads to a sudden increase in the number of model parameters,

Manuscript received July 2, 2019, This work was supported in part by the China Jiangxi Provincial Department of Education Science and Technology (GJJ151264).

Jian Liu is a lecturer in the College of Information and Computer Engineering, Pingxiang University, Pingxiang, 337055, China (e-mail: liujianpxu@163.com).

Weisheng Wu is an assistant professor in the Network and Educational Technology Center, Pingxiang University, Pingxiang, 337055, China.

increasing the difficulty of model training and the size of the model weight file, which is not conducive to the actual model application; 2) Even when only a certain vocabulary is added or deleted, the vocabulary of the label changes as the number of output neurons of the model is proportional to the vocabulary, thereby requiring the model network structure to be modified. However, adding vocabulary of practical applications is almost inevitable, which will modify the model structure frequently, thus resulting in poor model stability.

To eliminate this problem, this study combines the generative adversarial networks (GANs) [10] with word2vec that is modeled on natural language processing to construct GAN-IW—a new image annotation model. The main steps for constructing the GAN-IW model are as follows: First, word2vec is used to convert the label into a multidimensional space vector of a fixed dimension, which is freely selected. The number of neurons in the model output layer will only be related to the dimension of the multidimensional vector. Relate to the tag vocabulary. In addition, when the vocabulary changes a little, instead of modifying the model structure, only the word vector conversion table of word2vec needs to be modified. Subsequently, the annotation model no longer outputs images corresponding to all annotations at once but uses the GAN network to output a multidimensional spatial vector corresponding to one candidate annotation at a time. Through the perturbation of random noise in the GAN network, the GAN network can output a multidimensional spatial vector corresponding to the image and different candidate annotations, each time. Lastly, the final labels of the image are filtered according to the multiple outputs of the model.

II. Wasserstein GAN

The core idea of the generating confrontation network is derived from the Nash Equilibrium of Game Theory. The model is shown in figure1. It consists mainly of a generator (G) and a discriminator (D). The generator is generated by random noise generation. False data is distributed over the dataset; therefore, the discriminator needs to distinguish whether the data input from the generator is derived from the generator or the dataset.

For any random vector $z \sim \mathbb{P}_z$, WANG et al.[11] designed to learn generator network $G(z)$, minimized the Wasserstein distance between the resulting distributed \mathbb{P}_G .

Hence, for the generated sample $G(z)$ and the actual distribution \mathbb{P}_r under the observation data point x , the minimized Wasserstein distance is $\min_G W(\mathbb{P}_r, \mathbb{P}_G)$. Wasserstein distance $W(\mathbb{P}_r, \mathbb{P}_G)$ proved to be a more reasonable cost function than other popular distribution differences and distances. Jensen-Shannon differences implied in GAN[10], for learning the distribution of low-dimensional manifold support. Due to the impact of the Wasserstein distance on the Kantorovich-Rubinstein duality [12], the Wasserstein GAN (WGAN) value function is written as

$$\min_G \max_{D \in \Phi} E_{x \sim \mathbb{P}_r} [D(G(z))] - E_{z \sim \mathbb{P}_z} [D(z)] \quad (1)$$

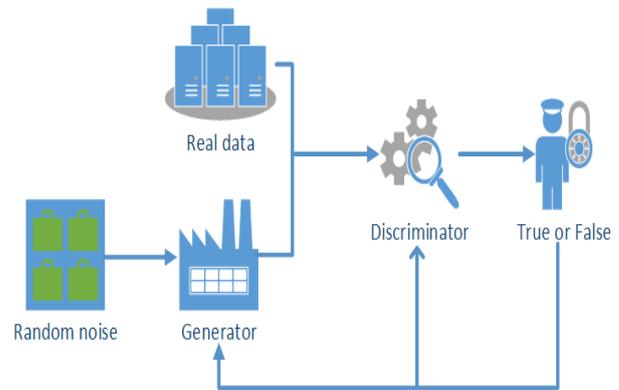


Fig. 1. GAN model

where Φ is a collection of 1-Lipschitz functions. Similar to GAN, D is a "discriminator". It tries to distinguish actual data from the data created by the generator. It could use any network architecture appropriate to the type of data it's classifying. Arjovsky et al.[15] specifies this function family Φ through the neural network and uses the weight limit to enforce the Lipschitz continuity.

GAN training requires alternating optimization of generators and discriminators. When optimizing generators, the objective function is minimized; thus, the generated data $G(z)$ is closer to the data set. The output of the discriminator $D(G(z))$ is closer to 1, i.e., the discriminator cannot distinguish the generated data $G(z)$ from the real data x . When optimizing the discriminators, the objective function is maximized; thus, $D(G(z))$ is close to 0, while $D(x)$ is close to 1. Therefore, the discriminator judges, as accurately as possible, whether the input data is from the actual data x of the data set or from the data $G(z)$ generated by the generator. Optimizing the generator and discriminator multiple times improves their performance. Finally, the performance of the generator and the discriminator reaches the Nash equilibrium; thus, the data distribution generated by the generator approximates the distribution of the original data set.

Gulrajani and others have improved WGAN's training. They illustrate the dangers of weight cut through more specific examples and propose another way to impose Lipschitz continuity. Lipschitz continuity is the central condition of the Picard–Lindelöf theorem which guarantees the existence and uniqueness of the solution to an initial value problem. A special type of Lipschitz continuity, called contraction, is used in the Banach fixed-point theorem. Gulrajani discovered the differentiable discriminator is 1-Lipschitz $D()$, When and only when the norm of its gradient is up to 1 anywhere. Therefore, gradient penalty is introduced.

$$GP | \hat{x} := E_{\hat{x}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (2)$$

However, unlike the weight limit, the term is never used to punish any place by a limited number of training iterations. Therefore, the gradient penalty, GP works just on the sampling point \hat{x} . In general, the number of sampled points is limited. Thus, no significant part of the support domain is examined at all. especially, think about the observed data points and its basic manifolds that support the actual distribution of \mathbb{P}_r . In the first

place of the training phase, the resulting samples $G(z)$ and \hat{x} can be kept away from the manifold. Until the build model \mathbb{P}_G becomes close enough to the real \mathbb{P}_r , if it can, it does not enforce Lipschitz continuity on the manifold.

Therefore, we recommended that WGAN training could be improved by setting additional Lipschitz continuity conditions on the manifold of the actual data $x \in \mathbb{P}_r$.

Previous approaches have focused on one particular data point at a time. Our approach is different, based on the most basic definition of 1-Lipschitz continuity, which is defined as follows: for a function $f(x)$, if x_1, x_2 in any of its defined domains, there exists $L > 0$ such that $|f(x_1) - f(x_2)| \leq L|x_1 - x_2|$. There exists a real number L such that for each pair of points on the function $f(x)$, the absolute value of the slope of the line connecting them is not greater than this real number L . This allows us to set the rule on a pair of data points plotted near the stream shape. It is worth mentioning that we perturb each actual data point x twice. To limit the difference between the discriminator's response to the perturbed data points x' and x'' , we use the Lipschitz constant.

A new consistent regular CT was used to force the Lipschitz continuity of the data manifold and its neighbouring region to essentially supplement and improve the gradient penalty GP for improving WGAN training. Therefore, the novel loss function to update the weight of a discriminator is defined as follows:

$$L = E_{z \sim \mathbb{P}_z} [D(G(z))] - E_{x \sim \mathbb{P}_r} [D(x)] + \lambda_1 GP|_{\hat{x}} + \lambda_2 CT|_{x', x''} \quad (3)$$

We borrowed $\lambda = 10$ from Gulrajani et al.[16], and irrespective on which dataset, all of our experiments used $\lambda = 2$. Consistency regular CT is used in the following form.

$$CT|_{x', x''} = E_{x \sim \mathbb{P}_r} \left[\frac{\max(0, d(D(x'), D(x'')) + 0.1 \cdot d(D_-(x'), D_-(x'')) - M')}{2} \right] \quad (4)$$

The value of M' between 0.2 and 0 will produce roughly the same result in our experiment. After losing the hidden layer of the discriminator, $D(x')$ is the calculation result of the discriminator for a given x . First, we pass the "virtual" data point x' through a clean discriminator. Then through the penultimate layer $D_-(\cdot)$ of the discriminator, that is, $d(D_-(x'), D_-(x''))$, the performance is slightly improved.

Set d represents the ℓ_2 metric on the input space used in our study. If there is a real number constant $M \geq 0$, the discriminator $D : x \rightarrow y$ is Lipschitz continuous, for all $x_1, x_2 \in x$,

$$d(D(x_1), D(x_2)) \leq M \cdot d(x_1, x_2) \quad (5)$$

A. word2vec

The text data should be converted to numerals since a neural network cannot directly process the text data. A traditional method is to convert the text data into a one-hot word vector, i.e., the word vector dimension is

equal to the vocabulary, and all words correspond to a certain dimension of the vector. If the word exists, the corresponding dimension takes a value of 1; otherwise, it takes a value of 0. For example, in a 5-dimensional word vector, a cat may be represented as $[0 \ 0 \ 0 \ 1 \ 0 \ 0]$, and a dog as $[0 \ 1 \ 0 \ 0 \ 0 \ 0]$. The one-hot representation method is a high-dimensional sparse method. The word vector dimension is directly proportional to the vocabulary. This method is computationally inefficient, and each dimension is orthogonal to the other and cannot reflect the semantic relationship between words.

In 2013, Google developed a new word vector generation tool word2vec, which can map words with multidimensional space vectors. For example, a cat may be expressed as $[0.1, 0.25, 0.3, 0.01, 0.9, 0.6]$. At present, word2vec is widely utilized in Natural Language Processing (NLP) missions. The core of word2vec are vocabularies with same or similar context, which may have similar semantics. According to the lexical context, each vocabulary in the text corpus is mapped to a corresponding unified N-dimensional vector in the vocabulary space. The vocabularies with similar semantics are similar in position in the space. For example, the spatial distance between the corresponding word vectors of cat and kitten is smaller than the spatial distance between cat and iPhone, thus reflecting the relationship between vocabularies and avoiding the disadvantage of the one-hot word vector.

III. Model design of our method

A. Model structure

The model structure used in our study is shown in figure2. The overall framework of the model adopted the conditional generative adversarial network (CGAN) architecture, and the input image size was unified with (299, 299, 3). The N-dimensional feature vector corresponding to the image was used as a condition, and the corresponding M-dimensional word vector was labeled as real data. Based on the condition and 100-dimensional random noise, the generator outputs the M-dimensional vector as the generated data. We used the Inception-ResNetV2[18] model for convolutional neural network (CNN) feature extraction and pre-training on the ImageNet dataset. After removing the final classifier layer, the migration learning method was applied to the model. The word2vec function was implemented using the Word2Vec module of the Gensim library, and the generated word vector dimensions were unified to 500 dimensions. Both the generator and the discriminator used a fully connected layer. The eigenvectors and the random noise were respectively mapped to different dimensions and then spliced. After repeated operations, the map was mapped to the output fully connected layer and the number of output connected-layer neurons was equal to the word vector dimension. Our method used the Improved WGAN model to train the GAN, and the discriminator output layer removed the sigmoid activation layer.

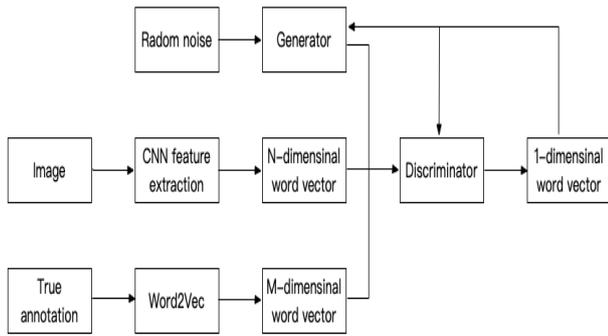


Fig. 2. our Model

B. Loss function

The uneven distribution of the vocabulary is a common problem of image annotation. Some annotations such as cafe and butterfly appear only twice in the Corel5K dataset, while water, sky, and tree appear more than 800 times. The frequency of appearance of different words with annotations is different. Hence, if they are not processed, the model easily ignores the influence of the low frequency label, resulting in a decrease in the accuracy of the low frequency vocabulary labeling Zhulinli2017. Therefore, for the unbalanced label distribution problem, the model optimized the loss function, multiplying the loss of different labels by a balance coefficient; thus, the label with low-word frequency had greater weight loss. The L2 regularization reduced the model over-commitment. Therefore, the revised loss is:

$$L_D = -E_{x,y \sim \mathbb{P}_{x,y}} [\log D(y|x)] - E_{z \sim \mathbb{P}_z} [\log D(K+1|G(z))] - E_{x \sim \mathbb{P}_r} [\log(1 - D(K+1|x))] + \lambda CT|_{x',x''} \quad (6)$$

where K is the count of categories of interest and the $K+1$ neuron is retained for using the Wasserstein distance between WGAN to compare the resulting sample with the actual data. The first three terms are the same as Salimans[17]. The final consistency is calculated after exiting the application of the discriminator. The fourth term is Consistency regular CT. According to Equation 4, the consistency regular term can generate a temporal self-integration scheme, which further aids semi-supervised learning. For more information on this topic, see Laine [13].

$$L_G = \|E_{z \sim \mathbb{P}_z}(D_-(G(z))) - E_{x \sim \mathbb{P}_r}(D_-(x))\|_2^2 \quad (7)$$

C. Label sort

Since the model outputs a candidate annotation word vector corresponding to one image at a time, the annotation ordering method of our study adopted the image-appearance order. Firstly, the image was N predicted by the trained model to obtain N word vectors. Secondly, for each word vector, the probability of corresponding M candidate words and their tag words was obtained by the Word2Vec model. Finally, the probability corresponding to the annotated word was used as the number of

occurrences of the annotated word, and the number of occurrences of all the candidate annotated words was counted. Then, the candidate annotation whose occurrence number was greater than the threshold was filtered by the threshold as the final annotation of the image.

IV. Experiment

A. Dataset

The datasets for this experiment are common datasets for image annotation: Corel 5K and IAPRTC-12 datasets. The Corel 5K dataset—the standard dataset for image experiments—is a collection of 5,000 images by Corel, which is commonly used in scientific image experiments such as image classification and retrieval. The IAPRTC-12 dataset was originally used for cross-language retrieval tasks. Each image has its descriptions in English, German, and Spanish. Researchers used natural language processing techniques to extract the common nouns in the graphic description as image tags. The IAPRTC-12 dataset is also used as a common dataset for image annotation tasks. The detailed statistics for the Corel 5K and IAPRTC-12 datasets are as shown in Table I.

 TABLE I
Dataset information

	Corel 5k	IAPRTC-12
Number of pictures	5000	19627
Number of tags	260	291
Test/training	500/4500	1962/17665
Average number of tags	3.4	5.7

B. Evaluation Method

The evaluation method used in the experiment calculates the accuracy (P) and the recall rate (R) and the F1 values for each label in the dataset. Suppose a related image of the test set is N and the relevant image predicted by the model is N_1 . The predicted number of related images is N_2 , then the accuracy is $P = N_2/N_1$, and the recall rate $R = N_2/N$ and $F1 = 2 \times P \times R / (P + R)$.

C. Labeling Results and Analysis

1) Influence of Different Thresholds on Image Labeling: Different labeling thresholds have a great impact on the final labeling performance of this model. In order to explore the relationship between different thresholds and labeling performance, this paper tests the labeling performance of models with different thresholds. Figures 3 and 4 show the relationship between the accuracy, recall rate, F1 value, and threshold of the model labeling. During the test, the number of model predictions is 128. Each candidate is selected to be the closest to the output vector. All candidate labels are counted. The number of occurrences greater than the threshold value is selected as the final label of the image.

It can be seen from figures 3 and 4 that the accuracy P of the label decreases first and then decreases with

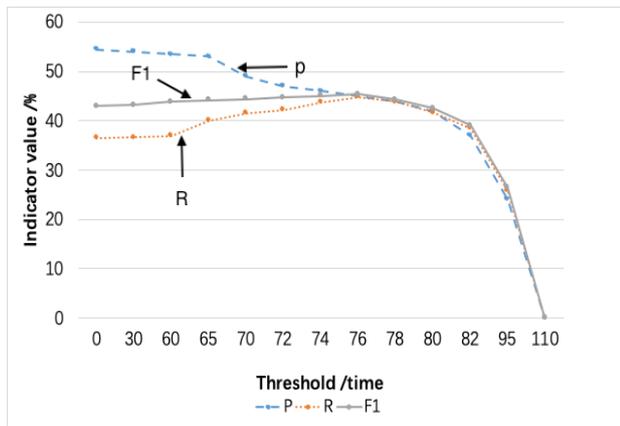


Fig. 3. Impact of Thresholds on Corel 5K Datasets

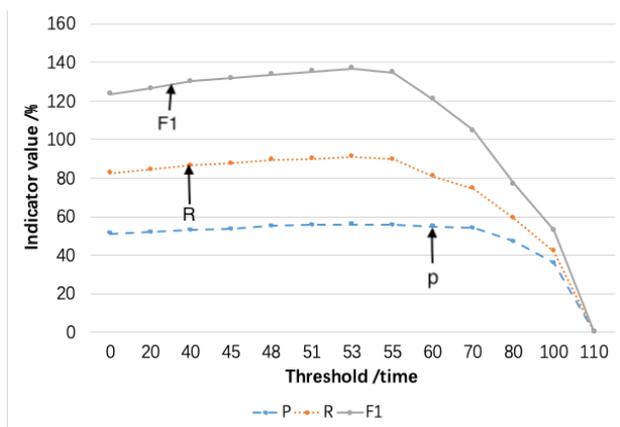


Fig. 4. Impact of Thresholds on IAPRTC-12 Datasets

the threshold and that the recall rate R decreases as the threshold increases. The value of $F1$ decreases substantially from the threshold and then decreases. The reason for this phenomenon being the model can learn the mapping relationship between image features and label vectors. By training the model, it has a certain ability to label. For the prediction results of most tags, the correct prediction corresponds. The number of occurrences is generally higher. When the threshold is particularly small, the general occurrence of the label corresponding to the threshold is greater than the threshold, the predicted result of the label is not substantially filtered by the threshold, and the labeling accuracy P and the recall rate R are unchanged. When the threshold is increased to a certain value, the partial error prediction is gradually filtered. The correct prediction is basically unaffected because of the large number of occurrences, the accuracy rate P rises, and the recall rate R is basically unchanged. The threshold continues to increase, and the correct prediction begins to be filtered; however, as the correct prediction situation is mostly concentrated on the higher occurrences, the increase in the threshold value has a greater impact on the correct prediction, which is filtered faster than the error. The predicted part eventually reduces the labeling accuracy P and the recall rate R until the correct prediction is completely filtered by the threshold and P and R are both zero.

The change in the $F1$ value is determined by the

TABLE II
Comparison of results from multiple methods

Dataset	IAPRTC-12			Corel 5K		
	P	R	F	P	R	F
2PKNN-ML	53%	32	40	41	46	43
SKL-CRM	51	32	39	39	46	42
KSVM-VT	47	29	36	32	42	44
NN-CNN	54	32	41	42	45	44
CNN-R	49	31	38	32	41	37
ADA	42	30	35	32	40	36
RF-opt	43	31	37	30	40	35
2PKNN	48	32	39	38	39	40
Our	56	38	43	46	47	46

change in the accuracy P and the recall rate R . The performance of the model varies from the threshold. In order to compare with other model labeling performance and the actual labeling effect of the model, it is necessary to determine the optimal threshold of the model. Since the $F1$ value can take both P and R , the $F1$ value is used as the reference for selecting the optimal threshold of the model, and the threshold with the largest $F1$ value is selected as the optimal threshold of the model. Since the optimal thresholds for different dataset models are different due to the differences between different datasets, for the Corel 5K and IAPRTC-12 datasets, the model selects 80 and 60, respectively, as the optimal thresholds for the model in the case of model predictions of 130.

2) Performance comparison of different annotation methods: This study compares the GAN-IW model with other classical annotations to verify the validity of the proposed model. The methods involved here include the traditional model methods, RF-opt (Random Forest-Optimize) [19], 2PKNN [4], 2PKNN-ML (2PKNN-Metric Learning) [4], SKL-CRM [2], KSVM-VT [20] and methods using deep convolutional neural networks NN-CNN (NearestNeighbor-CNN) [21], CNN-R (CNNRegression) [22], ADA (Attribute Discrimination Annotation) [23]. Table 2 shows the comparison of the performance of the GAN-IW model with other models on the Corel 5K and IAPRTC-12 datasets.

It can be seen from table II that the performance of the GAN-IW model is much better than that of the traditional method on the Corel 5K dataset. The recall rate of the proposed model is higher than that of the RF-opt method by 7%. The accuracy and the $F1$ value are 16% and 11% higher than that of the RF-opt method. In the method using the convolution model, the recall rate is 2% higher than the NN-CNN method. Both accuracy and $F1$ values are higher. On the IAPRTC-12 dataset, the model also performed well, with higher accuracy, recall rate and $F1$ values. The performance metric data onto the integrated GAN-IW model on the Corel 5K and IAPRTC-12 datasets can be concluded. Compared with other methods, the GAN-IW model has better performance than CNN methods. At the same time, both the model accuracy and the $F1$ value have been greatly improved and the best results have been achieved. The comprehensive performance of the model is significantly improved compared with other models.

Images	Original annotation	Our annotation
	water bridge lighthouse railroad	water bridge lighthouse railroad
	people waves oahu	people waves oahu
	meadow horses mare	meadow horses mare
	city buildings light street	city buildings light street
	sky sand elephant desert	sky sand elephant desert
	people buildings crafts	people buildings stone

Fig. 5. Actual annotation results of our method

3) Actual annotation results of our method: The actual results of the model automatic labeling are shown in Figure 5. The unified prediction number of the model is one batch size, i.e., 128 times. The threshold selected when testing the Corel 5K dataset is 80. The label with the number of occurrences greater than the threshold is selected as the final label of the graph.

The following conclusion can be drawn from figures 5 and 6: First, unlike most annotation models, the number of annotations per image is different or not fixed. In a real image annotation scenario, the number of image annotations will usually be various for different pictures. When trained, the GAN-IW model can learn the mapping relationship between the image features and the label vector. Each time a new image is predicted, the model outputs a certain feature on the basis of the visual features of the predicted picture. The matching label vector. For a semantically simple image, the image visual feature only contains the features corresponding to a certain tag. Therefore, the vector output of the model is close to the tag so that the corresponding number of occurrences of the tag is higher, and the number of occurrences of other tags is less than The threshold are

filtered out and the model finally has a small number of labels. The visual features of complex images are also usually complex, and these visual features may contain multiple labels. To extract these feature labels, we use a random noise interference method so that each label in multiple labels has a high probability of becoming a model. All the output labels are tested several times, and we find that most of the labels appear pretty often. We set a threshold value based on empirical values and add the labels with occurrences greater than the threshold value to the model, making the model have a large number of labels. For example, the flag in the last image of figure 6 is not in the original annotation but appears in the dataset with hotel and tree. Therefore, the flag is added as one of the final outputs and is still associated with the image content.

Second, some annotations do not conform to the original annotations but may be consistent with or related to the semantics of the test images. Some annotations such as people, buildings, and stone have a higher frequency of co-occurrence in the dataset. Therefore, when these annotations are vectorized using Word2vector, the distance between their corresponding multidimensional vectors is very close. As a result, the closest label words corresponding to the output vector often appear together and the probability of correspondence between the annotation words is very small. Hence, some of the annotations are not the original annotations. However, the number of occurrences of the final statistics is still large and is determined as one of the image annotations. These annotations often appear together in the dataset, thereby proving the deep connection between them. Therefore, in the new test image, the labels often appearing together with the original annotation still have a high probability of being related to the test image. For example, the stone in the last image of figure 5 is not in the original annotation but appears in the dataset with people and building. Therefore, the stone is used as one of the final outputs and is still associated with the image content.

V. Conclusion

In summary, this study aimed at solving the issue of change in the model structure with a change in the vocabulary, which reduces the accuracy of image annotation. In this study, a new annotation model, i.e., GAN-IW, based on the generated confrontation network and word vector model, on the Corel 5K and IAPRTC-12 datasets was proposed. The experimental results showed that the accuracy rate P, recall rate R, and F1 of the GAN-IW model significantly improved compared with other models, thereby proving that the model can be better applied to the image labeling task and the labeling result was more in line with the actual labeling situation. However, the following aspects are worthy of further improvement and research in this model: 1) The training result of the word vector lacks a good evaluation criterion. 2) The network model of the generator and the discriminator needs further optimization. 3) Better features need to be selected. Extract model and label to balance factors.

Images	Original annotation	Our annotation
	mountain jet plane	mountain jet plane
	mountain people road	mountain people road
	people town street horses	people town street horses
	people buildings street sidewalk	people buildings street sidewalk
	tree hotel maui	tree hotel maui flag

Fig. 6. Actual annotation results of our method

References

[1] Jiwoon Jeon, Victor Lavrenko and Raghavan Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," in Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval 2003, pp119-126, 2003.

[2] Sean Moran and Victor Lavrenko, "A sparse kernel relevance model for automatic image annotation," Journal of Multimedia Information Retrieval, vol. 3, no. 4, pp209-229, 2014.

[3] Ameesh Makadia, Vladimir Pavlovic and Sanjiv Kumar, "Baselines for image annotation," International Journal of Computer Vision, vol. 90, no. 1, pp88-105, 2010

[4] Yashaswi Verma and C. V. Jawahar, "Image annotation using metric learning in semantic neighborhoods," in Proceedings of the European Conference on Computer Vision 2012, pp836-849, 2012.

[5] Matthieu Guillaumin, Thomas Mensink and Jakob Verbeek and Cordelia Schmid, "TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation," in Proceedings of the IEEE International Conference on Computer Vision 2009, pp309-316, 2009.

[6] Edward Chang, Kingshy Goh, Gerard Sychay and Gang Wu, "CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 1, pp26-38, 2003.

[7] David Grangier, Samy Bengio, "A Discriminative Kernel-Based Approach to Rank Images from Text Queries," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 8, pp1371-1384, 2008.

[8] Changbo Yang, Ming Dong and Jing Hua, "Region-based Image Annotation using Asymmetrical Support Vector Machine-based Multiple-Instance Learning," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2006, vol. 2, pp2057-2063, 2006.

[9] Z. Li, Y. Zheng, C. Zhang and Z. Shi, "Combining Deep Feature and Multi-label Classification for Semantic Image Annotation,"

Journal of Computer-Aided Design and Computer Graphics, vol. 30, no. 2, pp318-326, 2018.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative adversarial nets," in Proceedings of the Conference on Advances in Neural Information Processing Systems 2014, pp2672-2680, 2014.

[11] Kun-Feng Wang, Chao Gou, Yan-Jie Duan, Yi-Lun Lin, Xin-Hu Zheng and FY Wang, "Generative adversarial networks: the state of the art and beyond," ACTA Automatica Sinica, vol. 43, no. 3, pp321-332, 2017.

[12] Cédric Villani, "Optimal transport: old and new," Springer Science and Business Media, vol. 338, 2008.

[13] Samuli Laine and Timo Aila, "Temporal ensembling for semi-supervised learning," arXiv preprint arXiv: 1610. 02242, 2016.

[14] Mehdi Mirza and Simon Osindero, "Conditional Generative Adversarial Nets," arXiv preprint arXiv: 1411. 1784, 2014.

[15] Martin Arjovsky, Soumith Chintala and Bottou, Léon, "Wasserstein GAN," arXiv preprint arXiv: 1701. 07875, 2017.

[16] Ishaan Gulrajani, Faruk Ahmed and Martin Arjovsky, Vincent Dumoulin and Aaron C. Courville, "Improved Training of Wasserstein GANs," in Proceedings of the Advances in Neural Information Processing Systems 2017, vol. 30, pp5767-5777, 2017.

[17] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, "Improved techniques for training gans", In Advances in Neural Information Processing Systems, vol. 29, pp2234-2242, 2016.

[18] Christian Szegedy, Sergey Ioffe and Vincent Vanhoucke and Alex Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," arXiv preprint arXiv: 1602.07261, 2017.

[19] Hao Fu, Qian Zhang and Guoping Qiu, "Random forest for image annotation," in Proceedings of the European conference on computer vision 2012, pp86-99, 2012.

[20] Yashaswi Verma and CV Jawahar, "Exploring SVM for image annotation in presence of confusing labels," in Proceedings of the British machine vision conference 2013, 2013.

[21] Mahya Mohammadi Kashani and S. Hamid Amiri, "Leveraging deep learning representation for search-based image annotation," in Proceedings of the Artificial Intelligence and Signal Processing Conference 2017, pp156-161, 2017.

[22] Venkatesh N. Murthy, Subhransu Maji and R. Manmatha, "Automatic image annotation using deep learning representations," in Proceedings of the ACM on International Conference on Multimedia Retrieval 2015, pp603-606, 2015.

[23] MingKe Zhou, Xiao KE and MingZhi Du, "Enhanced deep automatic image annotation based on data equalization," Journal of Software, vol. 28, no.7, pp1862-1880, 2017.

[24] Gang Liu, Xin Chen and Yanzhong Hu, "Anime Sketch Coloring with Swish-gated Residual U-net and Spectrally Normalized GAN," Engineering Letters, vol. 27, no. 3, pp396-402, 2019.

[25] Linli Zhu, Yu Pan, Muhammad Kamran Jamil and Wei Gao, "Boosting Based Ontology Sparse Vector Computation Approach," Engineering Letters, vol. 25, no. 4, pp406-415, 2017.