Application of Animation Character Intelligent Analysis Algorithm Based on Deep Learning

Lei Duan
Sanmenxia Polytechnic, Sanmenxia 472000, China

Correspondence should be addressed to Lei Duan; duanlei0305@outlook.com

Received 8 July 2022; Accepted 20 August 2022; Published 12 September 2022

Copyright © 2022 Lei Duan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The exit situation of characters is unstable, and the image is easy to change with the scene. Therefore, in this paper, an intelligent analysis algorithm of animation characters based on SSD target detection is proposed, which is trained by data enhancement strategy, where the weight factor of positive and negative samples are adjusted by an optimization method. Finally, it analyzes the overall process of intelligent analysis ability service application of animation characters. The test results verify the effectiveness of the improved algorithm that the proposed method can optimize the training process of the model, which helps users to more conveniently understand all kinds of animation characters, effectively enhances the efficiency of information dissemination, and ultimately plays a role in promoting the development of animation industry.

1. Introduction

With the vigorous development of animation industry, a large number of excellent works have emerged. These animation works are more and more widely concerned and welcomed, among which the role of animation is deeply loved by people of different ages. The market value of the animation industry has increased significantly, and its competition is becoming increasingly fierce. In order to occupy a place in the market, animation-related manufacturers constantly introduce new animation works and applications in their own products, in order to retain and attract users. It will be a very novel and interesting application to integrate the technology of artificial intelligence into products [1, 2]. As an important part of AI and a new growth point, computer vision is bound to present an explosive development posture, and is bound to be widely used in all walks of life.

As an important direction of computer vision, image target detection, with the support of artificial intelligence technology, will have more application development space. The analysis of the animation characters in the image in the form of target detection can help users easily understand unfamiliar animation characters and arouse users’ interest in animation characters and related works, which make animation products more entertaining and attractive [3]. The target detection task includes the prediction of target location coordinates and the classification of target categories [4]. In the former, the main problem is the location where it is mainly to complete the category prediction for all possible targets, and assign the target to the most credible category among the preset categories. Common target detection tasks can be divided into general detection tasks and special detection tasks. The general detection task does not specifically define the category of detection targets, which is often necessary to detect a large number of objects in daily life. While special detection tasks often focus on specific scenes, such as pedestrian detection in security monitoring, face detection in face authentication scene, and vehicle detection in intelligent traffic scene. Girshick et al. proposed the R–CNN model [5]. Compared with traditional methods, R–CNN uses CNN to extract image features of candidate regions, which improves the detection accuracy effectively. However, because the feature extraction of selected candidate region and convolution network is carried out separately, R–CNN needs to repeatedly calculate convolution operations, which directly leads to the slow running speed of the algorithm. Although Fast–R–CNN can complete
feature extraction, classification and candidate frame regression in target detection, it still uses the traditional region search method, which greatly affects the efficiency. To solve this problem, Ren [6] added the generation of candidate regions to convolution networks, and a complete end-to-end framework from candidate region production to target detection is realized.

In recent years, some researchers have proposed a target detection algorithm that is not based on anchor free. The main idea is to transform the detection of frame into the detection of key points, locate the target by detecting the position of the target frame, and determine which points belong to the same target by embedding features [7–9]. Examples of such algorithms are Cornernet and Centernet. This algorithm can really realize the end-to-end training. However, it is more difficult to detect the points. In order to achieve good detection accuracy, the input image needs to be processed twice by Hourglass Network; in addition, due to the large number of up sampling operations, the network parameters and computation are large, which requires high memory and computing power of training equipment, and the process of training and reasoning is slow.

Wang et al. [10] introduced the current video detection algorithms from three technical challenges (improvement and optimization, maintaining spatiotemporal sequence consistency, and model lightweight). There are four types: motion-based information, combination of detection and tracking, lightweight video detection, and the use of crossover models (such as combining transformer and video detection in the field of natural language processing). Zhou [11] studied the detection method based on time sequence characteristics, combined with feature fusion and dual model to detect video frame by frame, corrected the detection result of the current frame through the feedback of the previous frame so as to improve the continuity between frames and improve the detection accuracy and video continuity.

Based on the above analysis, defining each animation character as a category can realize the detection of animation characters in the animation scene. Therefore, through partial selection of SSD detection module, this paper proposes an intelligent analysis algorithm of animation characters based on target detection, in order to reduce the detection accuracy. At the same time, it analyzes the application scenarios of the algorithm, so as to help users understand all kinds of animation characters more conveniently, effectively enhance the efficiency of information dissemination, and finally play a role in promoting the development of the animation industry.

2. Intelligent Analysis Algorithm of Animation Characters Based on Deep Learning

2.1. SSD Target Detection Algorithm

2.1.1. Network Architecture. Network structure of SSD is composed of two parts: front-end network and auxiliary network structure. The front-end network of the basic SSD is the classical convolutional neural network structure VGG16. Then, there are Conv6 and Conv7, which are transformed from the original FC6 full connection layer network.

On this basis, the SSD network continued to add Conv8_2, Conv9_2, Conv10_2, Conv11_2, and the previous Conv6 and Conv7 to form the auxiliary network structure. As shown in Figure 1, the width and height of the cube represent the size of the feature graph, and the thickness represents the number of channels.

2.1.2. Selection of Default Box. In the design of network structure based on end-to-end deep learning target detection, the generation of default box is very important for the tasks that can be targeted by the network and the performance of detection. As shown in Figure 2, it is assumed that there are $8 \times 8$ and $4 \times 4$ different feature graphs. The feature graph grid refers to each small grid in the feature graph that there are 64 and 16 grids in the feature mAP of $8 \times 8$ and $4 \times 4$, respectively. The default box means that each grid of the feature corresponds to preset boxes of fixed size, as shown in Figure 2 with four dotted lines corresponding to a small grid.

Assuming that each feature grid pair should have $k$ default boxes, then $n_{\text{label}}$ category confidence and 4 positional offsets relative to the default box need to be predicted for each default box. In addition, assuming that the size of a feature graph is $m \times n$, and the convolution operation using some small $3 \times 3$ convolution check feature graph will output $(n_{\text{label}} + 4) \times k \times m \times n$ predicted values.

Feature mAPs with different depth of network layer have different feelings of corresponding original image size. Suppose we use feature mAPs of $m$ network layers with different depths for prediction, and the default box size ratio formula of feature mAPs of different layers is as follows:

$$s_k = s_{\text{min}} + \frac{s_{\text{max}} - s_{\text{min}}}{m - 1} (k - 1).$$

Each default box has a different shape, and in the base SSD algorithm, the width of each default box is calculated as follows:

$$w_k = s_k \sqrt{a_r}.$$ (2)

The height of each default box is calculated as follows:

$$h_k = \frac{s_k}{\sqrt{a_r}}.$$ (3)

Though animation characters in the animation story at different times is uncertain, the animation character’s head is the highest frequency, which contains the hairstyle and the entire facial features of animation character. Therefore, this paper chooses to label the animation character’s head to represent the animation character. However, the shape of the head is relatively single. In this paper, a new default box shape is set based on a large number of labeled head shapes. The new ratio of width to height is $\{1/2,1,3/2\}$.

2.1.3. Algorithm Flow. During the training of SSD frame, the real label box and default box are matched in the following way:
First of all, find the default box with the largest Jaccard overlap with each real annotation box, so as to ensure that each real annotation box corresponds to a unique default box.

(2) Then try to pair the remaining default box that has not been paired with any real annotation box. As long as the Jaccard overlap between the two is greater than the threshold, it is considered as a match. In this way, the default box paired to the real box is the positive sample, and the default box that is not paired to the real box is the negative sample.

The training of SSD algorithm is based on reference [10]. The overall objective loss function is as follows:

$$L(x, c, l, g) = \frac{1}{N} \left( L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right),$$  \hspace{1cm} (4)

where $c$ represents category confidence, $l$ represents target prediction box, $g$ represents target real label box, and the default $\alpha = 1$.

The calculation of Loc is as follows:

$$L_{loc}(x, l, g) = \sum_{i \in \text{Pos}}^{N} \sum_{m \in \{c, x, y, w, h\}} x_{ij}^k \text{smooth}_L(l_i^m - g_j^m).$$  \hspace{1cm} (5)

Among them, $(cx, cy)$ represents frame coordinates of the center of the $w$ and $h$, respectively frame’s width and height, $(g_{ij}^x, g_{ij}^y, g_{ij}^w, g_{ij}^h)$ represents the bubbles, $(d_{ij}^x, d_{ij}^y, d_{ij}^w, d_{ij}^h)$ represents the default frame, and $(l_{ij}^x, l_{ij}^y, l_{ij}^w, l_{ij}^h)$ represents the offset of the predicted target box.

Conf is a typical Softmax loss function, shown as follows:

$$L_{conf}(x, c) = - \sum_{i \in \text{Pos}}^{N} x_{ij}^p \log(c_i^p) - \sum_{i \in \text{Neg}} \log(c_i^0)$$

where $c_i^p = \frac{\exp(c_i^p)}{\sum_{p} \exp(c_i^p)}$. \hspace{1cm} (6)

2.2. Algorithm Optimization. For SSD algorithm, a serious problem is the imbalance of labeling classes. The quantity of negative cases (foundation classes) is a lot bigger than the quantity of positive examples. Each training picture can produce up to 8732 candidate frames of various scales in the feedforward process of the network. Because the number of labeled boxes in a training mAP is usually only a few or even one, even if the IOU threshold of 0.5 is used for matching, the number of candidate boxes that can match the super-box is still only a small part of all the candidate boxes.

In the training of animation character recognition deep learning model, the number of background samples is far more than the positive samples including animation characters [12]. Therefore, the model tends to learn “what is the background” rather than “what is the target,” which reduces the accuracy of target detection in application. Since the main application scenario of this paper is the main animation characters in the picture, it is not necessary to identify the small animation characters in the image, and the characters with resolution below $150 \times 150$ are also excluded from the annotation data. We can selectively retain some
contents and items in the original network, and need a
detection module with high matching degree, thus the ef-
ficiency of model training and reasoning can be improved
without reducing the accuracy.

Therefore, a weight parameter \( w \) is added to the con-
fidence loss function of SSD. New confidence loss function is
as follows:

\[
L_{\text{conf}} (x, c) = -w^* \sum_{i \in \text{Pos}} x_i^* \log \left( \frac{e^{z_i^*}}{e^{z_i^*} + e^{z_i^{\text{neg}}}} \right) - \sum_{i \in \text{Neg}} \log \left( \frac{e^{z_i^{\text{neg}}}}{e^{z_i^*} + e^{z_i^{\text{neg}}}} \right).
\]

3. Model Test

3.1. Data Sources. First of all, a large number of cartoon
character images are needed, and the characters in the
pictures are marked. 36 animation characters in 11 classic
animation works need to be identified with 500 pictures. In
addition, no less than 30% of the training set sample size
should be prepared as the test set. The format of animation
character pictures is JPG, and the resolution should not be
less than 640 * 480, and the size of characters in the picture
should not be less than 150 * 150 pixels. The process of
obtaining training data source is shown in Figure 3:

As far as the whole animation works are concerned, the
number of image data of animation characters sorted out in
this paper is relatively small, and the training of actual
animation character detection model is far from enough.
Therefore, this paper enhances the original animation image
data. To some extent, data enhancement can prevent over
fitting of training, and it assumes a significant part in the last
acknowledgment capacity and speculation capacity of the
prepared model. Its methods are shown in Figure 4.

3.2. Data Label. For the prepared source data images con-
taining the animation characters, the annotation terminal
downloads the source data images from the annotation
subsystem through FTP protocol, and then labels them. The
specific labeling method is to mark the position of the
cartoon characters in the picture in a rectangular box, that is,
frame out the position of the characters, and label the
horizontal and vertical coordinates of the upper left corner and
the lower right corner of the picture. The requirements are as follows:

(1) Use a rectangular frame to label the target cartoon
characters in the pictures;
(2) The head and body of the character are framed re-
spectively, and the rectangular box is close to the
edge of the character;
(3) If there are too few body parts of animation char-
acters to be framed, the body cannot be selected by frame selection;
(4) All the front, side, and back of characters should be
selected by frame;
(5) When the animation character size is less than
150 * 150 pixels, it is not labeled;
(6) When the annotator cannot judge and the prior data
are missing, the animation character is not labeled;
(7) Try to avoid adding subtitles and other irrelevant
information to animation characters;
(8) When the rectangle box is on the edge of the image,
the rectangle box should be close to the edge.

For large-scale training data generated by data en-
hancement, the size and position of animation characters are
known information in the process of enhancement, which
can directly generate annotation information without
manual annotation, which can save a lot of labor cost and
avoid the error caused by manual annotation.

3.3. Test Code. All experiments in this paper are imple-
mented on Caffe open source framework [13]. The improved
SSD with a positive loss weighting factor of 3 was used, and
the training effect was compared with the original SSD. To
implement the positive loss weighting method, we need to
add the code of calculating the loss value of Caffe-SSD source
code.

3.4. Evaluation Index. The training set includes about 15000
images in the original animation image data and the en-
hanced images of these data. The verification set is randomly
extracted from the original animation images, with a total of
5000 pieces. The training iteration number is set to 30000
times, and each iteration is verified once every 1000 times,
the results of verification use the mean average precision
(mAP) as the evaluation standard. For the target detection
task, each detected target can calculate its precision and
recall. After many experiments, we can get a P-R curve for
each target category. The value of AP is the area under the
curve, and the value of mAP is the average value of AP value
of the target category. The value range of mAP is 0 to 1.

4. Results and Discussion

4.1. Data Enhancement. Figure 5 is a graph of animation
image data enhancement mAP. Among them, the red curve
represents the change of mAP value of verification set with
the increase of iteration times when the original data are
used as training set. The green curve shows the change of the
mAP value of the verification set with the increase of iter-
atation times when the enhanced data is used as the training
set for training.

It can be clearly seen from Figure 5 that although before
about 12500 iterations, the mAP value of the red curve is
slightly higher than that of the green curve, but after that, the
growth of the mAP value of the red curve tends to be flat,
while the mAP value of the green curve continues to
maintain the growth trend, and soon exceeds the mAP value
of the red curve. Finally, the mAP value of the model trained
by the original data is stable at about 0.65, while trained with
the enhanced data is stable at about 0.72, which is increased by
10.7%. It can be seen that data enhancement can enhance the
performance.
Figure 3: Process of data generation.

Figure 4: Data enhancement method.

Figure 5: mAP of animation image data enhancement.

Figure 6: mAP of original SSD and improved SSD.
4.2. Model Optimization. As shown in Figure 6, the ordinate is the mAP result of verification of the verification set in the training process. The red curve and the yellow curve represent the change of the mAP value verified by the verification set when \( w \) is set to 0 and 3 for training.

As can be clearly seen, for the improved SSD, mAP showed the fastest growth trend and stabilized at a high value of 0.77, while the original SSD stabilized at 0.69, which decreased by 10.4% in comparison. It can be seen that properly increasing the loss weight of positive samples can improve the performance of the animation character detection model.

Figure 7 shows the comparison of loss function between original SSD and improved SSDs.

It can be clearly seen from the figure that the variation trend of the corresponding convergence curve varies with different values of \( W \), and as the value of \( W \) increases, the final convergence loss function value increases correspondingly. This indicates that the loss weight of positive samples does affect the training of the model.

5. Application Scenarios

After the training of animation character intelligent recognition model is completed, it needs to be deployed as a back-end service for the server to call. This service needs to be deployed in a server that supports fast forward derivation of the deep learning model, as shown in Figure 8.
5.1. Mobile Terminal. When watching animation, mobile client users will see unfamiliar animation characters and want to know more about the animation characters. They can take animation pictures with the camera in the client system or read local animation pictures directly. They can upload pictures by using the image search function in global search.

5.2. Client. The product side server will store the image data in the dimensional download image server, the dimensional download image server returns the image URL address to the product side server; while the client sends a request to the intelligent analysis capability service of animation characters through the server on the product side. The latter obtains the image source file from the dimensional download image server by analyzing the URL address of the picture, analyzes the animation characters in the picture, and sends the processing result to the server on the product side.

5.3. Product Terminal. On the product side, the server selects the animation character with the highest correlation according to the processing results, uses the name of the animation character as the keyword to search the works related to the animation character through the search engine, and recommends the relevant information to the client user.

6. Conclusion

This paper proposes an intelligent analysis algorithm of animation characters based on SSD target detection. Through online crawler to collect a database containing 36 kinds of animation characters, large-scale and high-precision training data are produced in the way of data enhancement. The SSD is improved by modifying the classification weight in the loss function. The results show that, without increasing the model parameters and test time, the accuracy of animation character recognition is improved from 69% to 77%, which provides a strong algorithm support for the actual application of animation character recognition. By deploying the application service of intelligent analysis ability of animation characters, users can easily understand all kinds of cartoon characters and effectively enhance the efficiency of information dissemination, which plays a role in promoting the development of the animation industry.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Cao, “China’s animation IP market ecology and its formation in we media era,” Popular Literature and Art, vol. 12, no. 21, pp. 129-130, 2020.
[2] G. Bai and F. Liu, “Application and development of inertial motion capture technology in film and television animation,” Electroacoustic Technology, vol. 43, no. 12, pp. 41-42, 2019.
[3] J. Zhang, Research on Animation Character Recognition Based on Neural Network University of Electronic Science and Technology, 2019.
[4] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: ideas, influences, and trends of the new age,” ACM Computing Surveys (CSUR), vol. 40, no. 2, pp. 1–60, 2008.
[5] R. Girshick, J. Donahue, and T. Darrell, “Rich feature hierarchies for accurate object detection and semantic segmentation,” 2013, https://arxiv.org/abs/1311.2524.
[6] S. Ren, K. He, and R. Girshick, “Faster R-CNN: towards Real-Time object detection with region proposal networks,” IEEE Trans. On Pattern Analysis & Machine Intelligence, vol. 39, 2017.
[7] X. Wu, X. Song, S. Gao, and C. Chen, “A survey of target detection algorithms based on deep learning,” Sensors and Microsystems, vol. 40, no. 02, pp. 4–7, 2021.
[8] H. Law and J. Deng, “CornerNet: detecting objects as paired keypoints,” Proceedings of the European Conference on Computer Vision, vol. 46, pp. 734–750, 2018.
[9] K. Duan, S. Bai, and L. Xie, “CenterNet: keypoint triplets for object detection,” Proceedings of the IEEE/CVF International Conference on Computer Vision, vol. 33, pp. 6569–6578, 2019.
[10] D. Wang, C. Bai, and K. Wu, “Survey of video object detection based on deep learning,” Computer Science and Exploration, vol. 15, no. 9, pp. 1563–1577, 2021.
[11] N. Zhou, “Research on video object detection based on timing characteristics,” Journal of Chinese Academy of Electronic Sciences, vol. 16, no. 2, pp. 157–164, 2021.
[12] C. Szegedy, S. Reed, and D. Erhan, “Scalable, high-quality object detection,” pp. 1412–1441, 2014, https://arxiv.org/abs/1412.1441.
[13] S. Xu, “Application of deep learning framework Caffe in image classification,” Computer and Information, vol. 23, 2019.