Text to Animation

YICHENG LI
a1698732

Abstract

This paper provides a simple method to turn text to animation, the basic idea is to analyze the text to obtain the desired information, then according to the information to find background video and character video, combining these 2 videos to generate an animation based on the description. Overall, this method can successfully turn text to animation, however, there is a significant limitation, the text is limited by the video dataset, so, it could be considered as future work, in future, I will focus on using neural network to generate video dataset.

1. Introduction

Computer vision is a technology to help the computer to see the real world, it is widely used in daily life, such as, face recognition and object detection. With the development of the technology, it can do more and more incredible things, especially, make animation. In this paper, I will provide a simple method to transfer text to animation by using computer vision techniques, it includes 4 basic steps, text analysis, character extraction, background subtraction, and coordinates optimization. The section 2 will introduce the related work, such as the algorithm that used in this project. The section 3 will explain the 4 steps to make animation in detail.

2. Related Work

2.1. YOLO

YOLO is the abbreviation of You Only Look Once, it is an object detection algorithm developed by Joseph et al. (2016) [5], in comparison with traditional object detection algorithm such as R-CNN [3], YOLO is faster, because R-CNN method will generate many potential bounding boxes at first, then run classification algorithm to evaluate whether there is an object in the potential bounding boxes, so run R-CNN model is expensive and slow. However, in terms of the YOLO, it reframes the object detection problem to regression problem, as figure 1 showed the basic idea of the YOLO, the input image is divided into SxS grid, then, if the object centre fall into a grid cell, this cell will start to detect the object, in detail, it will predicate B bounding boxes and calculate the confidence score of every box, the confidence score represents the probability of finding object in a box , finally, it will output the box with highest confidence.

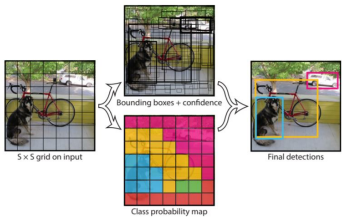

Figure 1. YOLO

Figure 2 showed the comparison between YOLO and other object detection algorithm, it is clear that the mAP value of YOLO is similar to other algorithms performance, but the FPS of YOLO is higher, so, YOLO is the best choice when you want to do a real-time detection, because the FPS of R-CNN or faster R-CNN is less than 20, it is hard to do real-time detection or video detection. Therefore, I consider using YOLO to extract object and make animation.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Figure 2. comparison

2.2. CANNY

CANNY is an edge detection algorithm that developed by Canny (1987) [2], it includes 5 steps, at first, the Gaussian filter is used to remove the noise as image noise will affect edge detection result. Then, CANNY algorithm uses different filters to detect horizontal, vertical and diagonal

edge, because the direction of an edge is different. After obtaining the direction gradient, CANNY will remove all pixels which not constitute the edge, this process also called 'thin edge'. Finally, all edges will be assessed to decide whether the edge is a true edge or a fake edges caused by color variation or noise, then all fake edges will be removed, Canny (1987) [2] uses the double Threshold to assess edges, firstly, setting a maximum threshold value and a minimum threshold value, all edges that intensitity gradient greater than maximum value will be identified as true edges, if the intensitity gradient less than minimum value, the edge will be identified as fake edge and it will be removed, when the gradient is greater than mimimum value and less than maximum value, if this edge is connected to a true edge, it will be preserved, otherwise it will be removed. Therefore, the 2 threshold value will strongly affect detection result.

In addition, in comparison with other edge detection algorithm, Raman and Himanshu (2009) [4] state that CANNY algorithm performs better than other algorithms, such as Sobel, Prewitt and LoG, the main weakness is that CANNY performance strongly depends on these adjustable parameters, for example, threshold value. Thus, I consider use CANNY to do edge detection, and I will carefully adjust threshold value.

2.3. NLTK

Text analysis is another important factor to turn text to animation, the NLTK is used to analyze text. The NLTK is a natural language toolkit that developed by Bird and Loper (2004) [1], moreover, NLTK has been adopted by many universities, it approved the NLTK is a reliable text analysis tool, therefore, I consider using NLTK to analysis text and extract the information that can make animation.

3. Text to Animation

In order to turn text to animation, I consider 4 steps, at first, analyzing text to obtain character, background, action and direction information. Then, extracting character animation and background from video dataset by using YOLO. In addition, using CANNY to remove unwanted background information. Finally, optimizing coordinate and combining these 2 animations.

3.1. Text Analysis

In terms of text analysis, I expect 5 different information, character, action, direction, object and background. For example:

'Demonhunter is standing on the snowland on the right of the sunwell.'

Demonhunter is a character, stand is action, snowland is background, right is direction, sunwell is an object. In order to extract this information, at first, every word in the sentence will be classified by using NLTK according to the tag, the tag of Demonhunter is NNP (proper nouns), the tag of standing is VB (verb), the tag of snowland, right and sunwell is NN (noun). Then, I only need to distinguish these 3 types word. I create 3 lists, a background list that include all background video information, such as snowland, badland and marbleland, so if the NN word can be found in background list, it will be identified as background, I perform the same process in terms of direction list and object list. In addition, as the variation of the verb form, I will perform a revert process, standing will be convert to stand. As a result, I can obtain all information that I want, character, action, background, direction and object.

3.2. Animation Extraction

In order to generate the desired animation, I have built a video dataset which includes all character animation and background animation, all I have to do is combining these 2 animations according to the text analysis. For example, currently, I have a demonhunter stand video and a badland video, as figure 3 showed, if the text is the demonhunter is standing on badland on the right side of the sunwell, at first, I need to detect demonhunter and extract demonhunter animation, then detect the sunwell location in the badland video, finally, combining these 2 video, put the demonhunter to the right side of the sunwell.

Background

Demonhunter

Figure 3. Character and Background

So, I use YOLO to achieve object detection, training YOLO is the first step, I manually collect a list of images of demonhunter and sunwell as training data, then use label tool to mark demonhunter and sunwell position in the images and generate an XML file which describes the

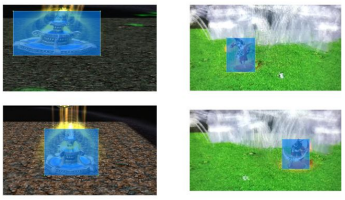

Figure 4. Training Data

object location, as figure 4 showed.

Then the images and XML files will be input into the network, after 50 epochs training, this network can achieve over 90% accuracy detection, as figure 5 illustrated, I have done a test, this model can accurately identify the demonhunter and the sunwell location, and it will return the coordinates of the object (xmin,ymin,xmax,ymax) and the label of the object(demonhunter, sunwell).

Figure 5. Detection Test

Figure 6. Video Combination

After detection process. the next step is to extract demonhunter animation from the video and put it to the background video, this process is pretty simple, the video is a list of images, so I only need run the detection algorithm on every image and extract the demonhunter images according to the coordinates, then, for each frame of background video, I will identify the location of the sunwell and replace the right area of the sunwell to the demonhunter image. As a result, the final video is demonhunter stands on the right side of the sunwell as figure 6 showed.

3.3. Background Subtraction

The quality of the combination video is poor, when putting demonhunter to the badland animation, the snow land background is preserved and it is not what I expect, therefore, I consider use CANNY to detect the object edge and subtract background, in addition, as mentioned before, CANNY edge detection result will strongly affected by the adjustable parameters, especially the maximum threshold value and mimimum threshold value, I have tested several different configurations. As figure 7 showed the test result, when setting a small maximum value and a minimum value, only a little background information is subtracted, when increasing the maximum value and the minimum value, more and more background information is subtracted from the image, however, if keeping increase these values, some character feature disappeared, so, according to the test result, I set the maximum threshold value to 400 and the minimum value to 380, most of the background feature is subtracted and most character feature is preserved.

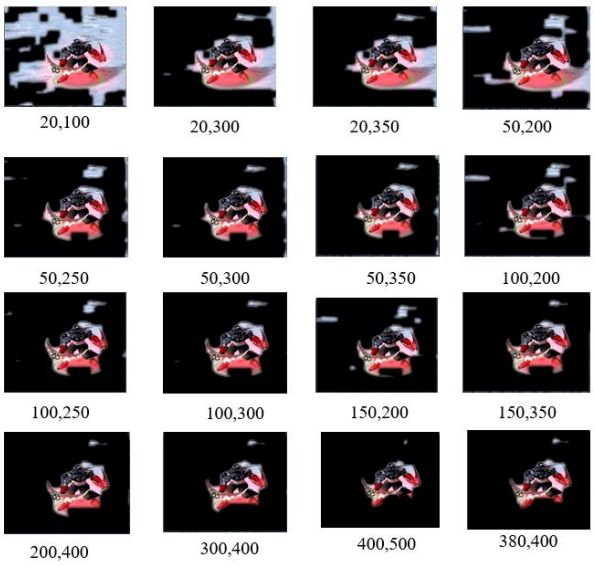

Figure 7. Background Subtraction Test

Figure 8 showed the screenshot of final output video after adding background subtraction function, it is clear that the quality of this screenshot image is better, most of the snowland background feature disappeared.

3.4. Coordinate Optimization

The accuracy of coordinate will strongly affect the quality of animation, as the previous example, I expect to put the demonhunter to the right side of the sunwell, so I have to calculate the coordinate of the sunwell for each frame,

Figure 8. Screenshot of Final Output Video

Figure 9. Sunwell Location Change

then, put demonhunter to the background video according to the coordinate. As a result, even the location of the sunwell have changed in the background video, the demonhunter still stands on the right side of the sunwell as figure 9 showed.

Figure 10. Unstable Coordinates

However, the YOLO object detection result is not stable even the location of the sunwell does not change. so, I have to evaluate whether the location of the sunwell has really changed or the change in coordinates is due to the detection error. As figure 10 showed 2 detection results, the coordinates of the sunwell (xmin, ymin, xmax,ymax) is slightly different even the location of the sunwell does not change, if I directly put demonhunter to the background video based on these coordinates, the animation is terrible, the location of the demonhunter is rely on the sunwell coordinates, when the coordinates is changing, the location of the demonhunter is changing as well, it is not acceptable as the coordinates change is caused by detection error. Therefore, these unstable coordinates should be optimized. In order to make the coordinates more stable, for any 2 consecutive frames, I will calculate the squared difference between the previous frame coordinate and the current frame coordinate. If the difference is too small, I will set the current frame coordinates equal to the previous frame coordinates, so, it can make the coordinates more stable when the location of the sunwell does not change, in addition, when the location of the sunwell has really changed in the background video, the difference will significantly increase,then, I will not modify the coordinates.

$$Difference = (Coordinates_{cur} - Coordinates_{pre})^2$$

I have done a test, 20000 is the critical point, if the difference is less than 20000, it means the change in coordinates is caused by detection error, so, I set the current coordinates equal to the previous coordinates, it can make the animation more stable. When the difference is greater than 20000, it means the location of the sunwell has really changed, so, the current coordinates will not be modified.

4. Conclusion

In conclusion, this paper provides a simple method to convert text to animation by using object detection algorithm and edge detection algorithm, the main limitation of this method is that the text is limited by the video dataset, user cannot provide a text which includes the information that can not be found in video dataset, for example, 'Demonhunter is standing on the grassland', this method will not work as there is no grassland video. So, in future, I will try to use the neural network to analyze different background feature and generate new background video to expand video dataset.

References

[1] S. Bird and E. Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004.

[2] J. Canny. A computational approach to edge detection. In *Readings in Computer Vision*, pages 184–203. Elsevier, 1987.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[4] R. Maini and H. Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11, 2009.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.