LEARNING TO COUNT OBJECTS IN NATURAL IMAGES FOR VISUAL QUESTION ANSWERING REPRODUCTION

Ziyang Wei & Riling Wei & Tianyi Gao
Department of Electronic and Computer Science
University of Southampton
{zw1y18,rw3a18,tg1m18}@soton.ac.uk

ABSTRACT

In this report, we justify the reproducibility of the counting component. The paper proposed an approach for counting the number of object proposals. We divide our report into 5 parts. In the first part, we describe the target question. In the next step, we perform additional experiments. In the third part, we visualized the procedure of the counting module. In the forth part, we discuss our result. In the final part, a conclusion is given.

1 TARGET QUESTION

Overlapping is the main problem in counting network as overlapping always causes double-counting of object proposals. For solving this problem, graph has been introduced. Then, the algorithm removes and scales object edges to estimate the number of targets.

Piecewise linear functions f1, , f8 are used for activation functions to give model perfect attention maps and bounding boxes to solve overlapping problem. These functions are with domain and range in [0,1] with monotonically increasing. In this algorithm, pre-trained model is used to obtain locations of targets. Then an attention mechanism has been introduced for counting. Attention weight(Anderson et al., 2018) and coordinates of boxes are used as input of counting network. The output is a vector of counting feature which used to count the number of the objects given in the image.

2 EXPERIMENTS

The paper uses toy task to evaluate the performance of the counting ability of the model and compares counting part with a baseline algorithm which are trained for 1000 iterations using Adam. The learning rate is 0.01 as well as a 1024 batch size.

In addition to the experiments performed in the paper, we perform more experiments. We create the given data by random value rather than the real images. Alter the original code to fulfill the inputdata. The additional experiments are as followed:

1. Change the number of weights of piecewise linear activation function to 32 instead of 16 performed in the paper.

2. Increase the number of targets to 30 rather than 10 in the paper.

3. Use random length of targets rather than the invariant length of data which is the same as realistic objects.

4. Add a visualize tool to track the calculation of the graph.

5. Change the confidence value to 0.1 rather than 0.5 performed in the paper.

6. Obtain a loss curve by changing epochs.[1]

[1] Our implementation is available at https://github.com/COMP6248-Reproducability-Challenge/vqa_counting.

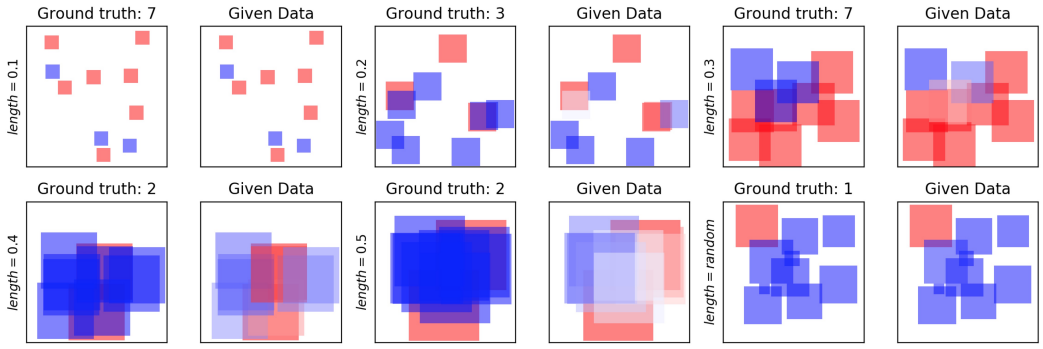


Figure 1: The object number is 10, we generate 6 different groups data depending on the length of the object

3 IMPLEMENTATION

In this part, we implement a visualization procedure for adjacent matrix.

Pre-trained model such as R-CNN(Ren et al., 2015) has been used to obtain object targets. Then object targets are translated into vector which is one of input of count component. Another input of counting session is the attention weights for those targets.

For simply problem, there are two assumptions. Firstly, the attention weights only have the value 1 (when object is target) or 0 (when object is not target). Secondly, any two object targets either overlap completely or having zero overlap.In count session, attention weights and the corresponding bounding boxes are the input. Exacting duplicates to prevent double-counting of objects is our main target. Graph is the main idea.

The first step is translating the vector into graph by generating an attention matrix:

$$A = aa^T \tag{1}$$

Then we compute the number of vertices $|V| = \sqrt{|E|}$, E can be computed by $\sqrt{E} = \sum_i a_i$.

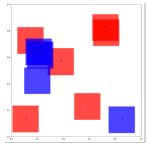


Figure 2: One of the input data which has 6 red boxes and 4 blue boxes. The box 0 and box 2 have an intersection area which will be calculate in equation 2. The distance matrix D also has two white nodes which shows that the IoU of it is large.

There are two different edge types should be eliminated: intra-object edges and inter-object edges. Intra-object edges can be eliminated by the following algorithm:

Firstly, intersection-over-union (IoU) metric are introduced to compute distance matrix D by the following equation:

$$D_{ij} = 1 - IoU\left\{b_i, b_j\right\} \tag{2}$$

Then, the attention matrix without self-loops are computed to remove intra-object edges:

$$\widetilde{A} = f_1\left(A\right) \odot f_2\left(D\right) \tag{3}$$

After eliminating intra-object edges, inter-object edges will be eliminated. A similarity function between proposals i and j are introduced:

$$Sim_{ij} = f_3\left(1 - |a_i - a_j|\right) \prod_k f_3\left(1 - |X_{ik} - X_{ij}|\right) \tag{4}$$

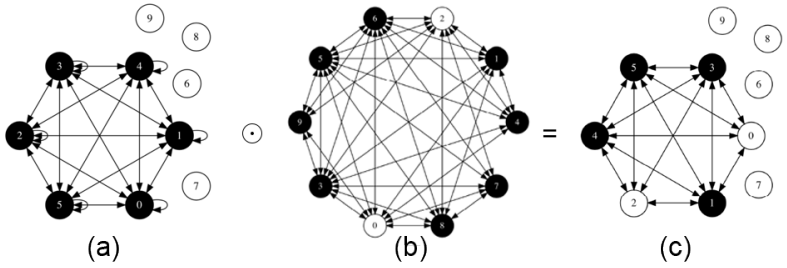


Figure 3: Relate to the equation 3, (a) is attention matrix A, (b) is the distance matrix D (bounding box), (c) is the attention matrix $\widetilde{A}$ without self-loop. In this figure object 0 is wrongly classified.

This similarity function is robust to inaccurate bounding boxes. Now we know how similarity between two proposals. Then, we compute a scaling factor:

$$s_i = 1/\sum_j Sim_{ij} \tag{5}$$

In the next step, we compute count matrix C with self-loops by the following equation:

$$C = \widetilde{A} \odot ss^T + diag\left(s \odot f_1\left(a \odot a\right)\right) \tag{6}$$

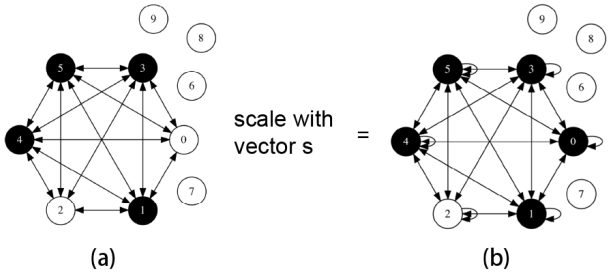


Figure 4: (a) is the attention matrix $\widetilde{A}$ without self-loop, (b) is the output computed from count matrix C. In this figure, after similarity function and count matrix, object 0 is correctly counted.

After computing count matrix C, the number of object proposals can be computed by the equation:

$$C = |V| = \sqrt{E} \tag{7}$$

The other parts are the same procedures as Zhang et al. (2018).

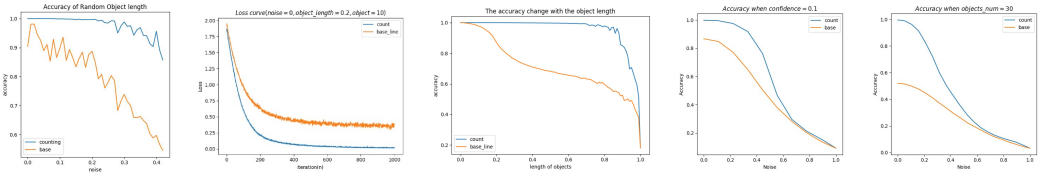
4 DISCUSSION

In order for evaluating robustness of the counting component, we add random noise to the model. For comparison, Baseline model (Kazemi & Elqursh, 2017) has been introduced. This model is based on a simple baseline architecture.

From figure 5(a), we can see clearly that the accuracy of counting component is constantly higher than Baseline method. If noise is lower than 0.25, the accuracy of counting component can remain at 1.On the other hand, we use loss curve to evaluate the performance of the counting model without noise. Figure 5(b) shows that counting model is better than baseline method as the loss of the counting model is constantly lower than baseline method. In addition, counting model convergences faster than the baseline. After about 400 iterations, the loss of counting model is almost at 0. Figure 5(c) shows the relationship between length of objects and the accuracy of the counting model and baseline method. From figure we can see clearly that counting component is constantly higher than baseline model and the accuracy is remained stable at 1 when length of object is less than 0.8 while the accuracy of baseline model decreased significantly. Figure 5(d) shows the accuracy of confidence

is 0.1, we can see there is not significantly different with confidence is 0.5. Figure 5(e) shows the accuracy when the objects are increased to 30 with noise. We can see clearly that although counting component works better than baseline method, the accuracy of counting component is worth than the number of objects is 10.

There are some reasons why counting component works better than Baseline method. The main reason is the use of deduplication. Baseline model use soft attention approach. This approach has some problems. For example, after weighting sum, the feature vector is always the same in different images and some information is lost about a possible count from the attention map (Zhang et al., 2018). By contrast, counting component has more advantages. For example, graph help counting module reduce overlapping problem, then the module removes intra-object and inter-object edges by distance matrix and similarity function. Those steps help counting module distinguish true targets without overlapping so the accuracy of the module is higher.



(a) (b) (c) (d) (e)

Figure 5: (a) Accuracy of two models with random noise. In this section, we use random length of objects. The number of dataset is also smaller than the dataset used by the paper, so the curve is fluctuated. (b) Loss curve with different iterations, (c) the relationship between length of objects and accuracy, (d) confidence is 0.1, (e) objects is increased to 30.

5 Conclusion

The paper proposed an approach for counting components. The paper written well. It provides the algorithm, parameters as well as results in detail.

Our task is to judge the reproducibility of the model by changing parameters. We change the number of weights of the activation functions, increasing the number and length of object, change the confidence value as well as increasing a visualization procedure. In addition, in order for reducing computational demands, we generate database by ourselves and do not use R-CNN for finding object proposals in images.

Our result obtained from experiments shows that this model is robust to the choice of the above parameters as the application of graph theory and the method for feature extraction.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6077–6086, 2018.

Vahid Kazemi and Ali Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. *arXiv preprint arXiv:1802.05766*, 2018.