Mask R-CNN for templates matching

Code: github.com/raikilon/r-cnn-template-matching **Model:** https://mega.nz/folder/R9pjzIhA#1cwibVervgv0iofhHhyxzQ

Noli Manzoni, Michael Denzler
Università della Svizzera italiana

I. Introduction

We extended our SIFT [1] single template matching (assignment 2) to a multi template matching by using our algorithm once for each template. Our implementation is quite robust because it is able to detect the given templates in complex scenes and to retrieve the correct ones even when multiples templates are similar (Figure 1). Unfortunately, this system has some handcrafted parameters (template similarity, LOWE ratio, etc) and to work with different images it needs tuning and therefore it cannot be used as a fully automatic detection system. To try to solve this problem, we used a selective search [2] to get interesting area where to search for the given template but this resulted in even worse performance. For this reason, we decided to move to a deep learning model to see if it can solve the problem. The biggest challenge is that we had only a single image per template as input and usually deep learning models need a lot of samples. How could we solve this?

II. Method

A. Data augmentation

The approach we took was heavy use of data augmentation. Our aim was to create (1) augmented images as the neural network inputs and (2) image masks describing the positions of the different templates as the neural network target values. Each augmented image created was based on a background image and three different template types, each stitched onto the background randomly 1 to 3 times. To augment templates being in different locations, distance, angles, or perspective orientations on the image, we applied homographies on the templates to resize and rotate the templates, as well as giving them different viewing angles. Then we stitched the templates in random positions onto the image and applied further augmentation steps to include illumination through Gamma Correction and blur through Gaussian Smoothing. This allowed us to simulate varying lighting conditions and blurred images taken for example from mobile phones. The final augmentation step was to then rotate the stitched images to also adjust for images taken not fully horizontally. The exact same augmentation steps, except illumination and blur, we also applied to the image masks for all three templates. Through this approach we now could create data sets with input images and corresponding target masks (see Figure 2) that would allow our Deep Neural Network to learn the desired behaviour.

B. Architecture

Using the augmented image data sets now could learn to detect templates, define their class, and draw their masks using Mask R-CNN [3]. Mask R-CNN is based on top of Faster R-CNN [4] that can also predict both bounding boxes and class scores. Because of the small amount of data available, we decided to fine tune a pre-trained model on the COCO dataset [5]. More precisely, we use ResNet50 [6] as our Faster R-CNN backbone and we replaced the heads of the Faster R-CNN and Mark R-CNN with new ones to be compatible with the new number of classes. The chosen backbone is big (around 25 millions parameters) and therefore, the training and inference are slow (it cannot be used for real time detection). Fortunately, this backbone could be changed easily with a new smaller one like MobileNet [7] which should allow real-time inference but because of the small amount of time available for this project we could not verify this assumption. We trained the model for 50 epochs with Stochastic Gradient Descent (lr$= 0.005$, momentum$= 0.9$, weight decay$= 0.0005$ and a learning rate decay of $\times 10$ every 3 epochs) on a set of 200 images where 20% of them were used for validation. The average training time was of 4 hours on a NVIDIA GeForce GTX 1080.

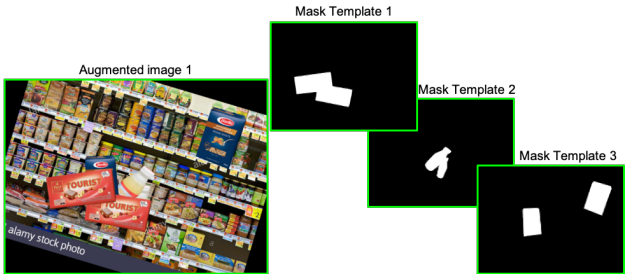
III. Results

To test our architecture we trained with different templates and different backgrounds (rooms or store shelves). In this report we report only the most interesting results. If you want to see more please consider looking at our repository under "images/results". To test the efficiency of our model trained on room backgrounds on real situation we trained it to recognise objects with different shapes that we had at home (Barilla pasta, Villar chocolate and Bauli pandoro). After some tuning of the parameters for the data augmentation (especially template size) we were able to achieve sufficiently good results. First of all, we checked if our model was able to recognise the templates in front-view like the synthetic images. As we can see from Figure 5 our model is able to detect the templates and to discard unknown objects. To see how well our model generalises, we took pictures with different viewing angles and illumination conditions. As we can see from Figure 4 is able to detect correctly all the templates, even the pandoro which has a curved shape. The only problem here is that it the top-view image it misclassifies parts the milk box as chocolate. In these situations with simple backgrounds our model behaves quite well. For this reason, we decide to train the network with more complex backgrounds (store shelves). Additionally, we wanted to see how good our model was in recognising similar templates (we used two chocolate boxes). Figure 3 shows the results of this training in a real and complex scenario. There we can see that the model is able to detect almost all the boxes (Frey Tourist and Frey blue Noxana) apart from the ones that are a partly occluded (action label) and one in the bottom. A problem of this model is, similar to the previous case, the misdetected false negatives.

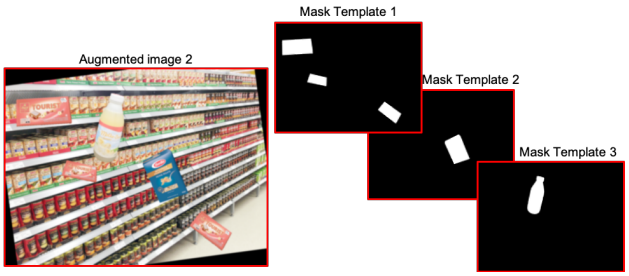
IV. Conclusion

Our model works well in simple and more complex situations and it is able to generalise on different viewing angles and illumination conditions. Moreover, compared to our SIFT multi-templates implementation, this method is much faster (SIFT takes around 77 seconds for 3 templates, our Mask R-CNN takes only 8 seconds) - and could be even faster if the backbone was replaced with a smaller one. Despite these promising results, our model's recall and precision rate are not yet optimal. They could presumably be solved by tuning the network and data augmentation hyper-parameters and training on a broader variety of images. Such enhancements would however go beyond the scope of this proof of concept and are left for future work.

Fig. 1: Multiple and similar templates detection with SIFT



(a) Medium illumination, low blur



(b) High illumination, high blur

Fig. 2: Examples of augmented image with corresponding image masks

Fig. 3: Similar templates detection in a complex and real situation

Fig. 4: Templates detection in real situations with different views and illumination

Fig. 5: Templates detection in a real situation similar to synthetic images

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, p. 91–110, Nov. 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[2] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: https://doi.org/10.1007/s11263-013-0620-5

[3] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[5] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861