Let’s Find Momo

Yipeng Su(yipengsu@stranford.edu) Yun Kun(Wilson) Zhang(wyzhang@stanford.edu)

Stanford
Computer Science Dept

Introduction

Let’s find Momo is s a series of children’s hide-and-seek books in which MoMo, a border collie, hides in various scenes. The mission is to develop an object detection model to identify MoMo in these scenes. The problem is challenging because

- MoMo is hiding behind objects
- MoMo is small
- Training examples are limited

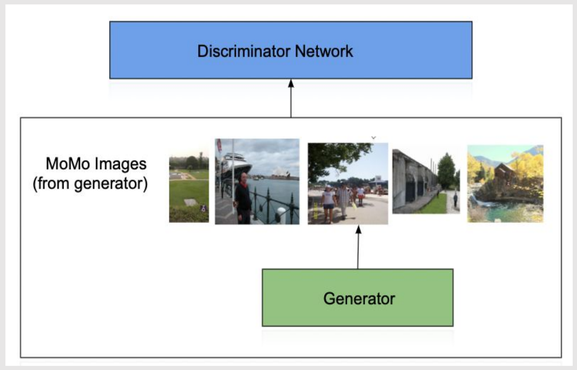
Can you identify MoMo below?

Problem Statement

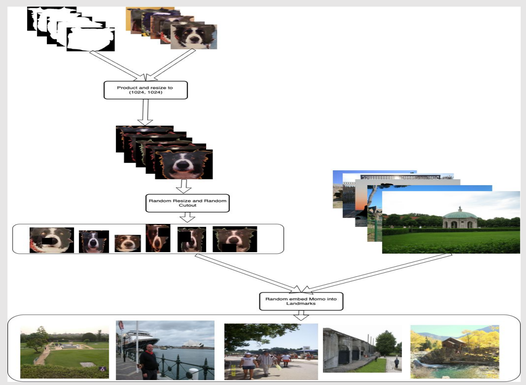
This problem is few shot object detection combined with small object detection. Prior related work includes various few shot object detection methods including using meta learning, attention-RPN and contrastive learning. We proposed a solution that is like GAN by building a generator generating candidate images to train the discriminator. We will use accuracy rate to evaluate the model.

Method

- Transfer Learning using RetinaNet as the pre-trained discriminator.
- Retrain the discriminator with images from the generator.



Dataset: Generator



Generator use Google Landmarks Dataset and Momo Avatars to generate training images to feed to our discriminator. Then the fine-trained discriminator will be used to detect Momo in the FindMomo Challenges.

Experiments & Results

Final Model Performance	
Metric	Accuracy
Top1	25.6%
Top3	39.2%
Top1 crop16	34%
Top3 crop16	54%

(a) Momo is hiding inside the bushes.

(b) Momo is hiding behind the block.

(c) Momo is hiding under the bridge.

(d) Momo is hiding inside the cherry yard.

Conclusion & Future Work

Overall in the experiment, we see the potential of leveraging GAN architecture to solve the few-shot object detection tasks, especially in dealing with the lack of labeled training data. We believe GAN + Object Detection worth more research and investment. In our experiment, we also identify some key ideas which need to be further investigated:

- Generalization. Building a good generator requires domain knowledge and current generator can not be directly applied to other applications.
- Overfitting. We identified the root cause of overfitting is the discrepancy between Test Momo Data and artificial training data created by generator. In future work, we need to populate the loss/gradient to generator and simultaneously improve the generator performance to create better images.