Can SAM Do Well in Counting? A Simple Few-Shot Counting Model Based on SAM

Zipeng Qiu
Fudan University
Shang Hai, China
20307130150@fudan.edu.cn

Abstract

Segment Anything Model (SAM) is an image segmentation model introduced by Meta AI[1]. It does an excellent job in segmentation tasks and can work as a pre-trained model in downstream assignments, which gives us a new way to solve problems. In this study, I introduce a few-shot counting model based on SAM. By connecting the SAM's decoder to a convolutional kernel, then just sum the output up as the counting result. The whole model is finished. Then we can train it just with the actual object quantity label. Finally, we can get competitive accuracy in the class-agnostic counting task. The model code can be found in https://github.com/TabibitoQZP/DIP_PJ

1 Introduction

The counting task is an important task in real life. Although it is a simple task for human beings. It is still time-consuming and boring for people to actually do. Especially if there are many objects in a scene to count, people could make mistakes due to the high-density influence on them. People are willing to invent some methods to help count. Therefore, in the computer vision (CV) field, scientists introduced many useful algorithms to detect objects and count them. And some of them can have a better performance than humans. However, most of them are developed for detecting one or a few specific objects like face count, car count, and so on. But as we all know, people can count on anything they want. We do not refer to a large number of labeled images to train ourselves to detect just one same object. Given a simple prompt like a coordinate or a wrapped box, we can instantly find all the required objects in an image. Even though counting still takes a lot of time. This kind of task is called the class-agnostic counting task. However, it is hard for the computer to count many different things. The class-agnostic counting task is still a tough problem in the CV field.

The big success of the transformer model in Natural Language Processing (NLP) inspires scientists to introduce it to the CV field. However, how to combine the transformer model with images is a big problem. By simply cutting an image into many patches, the image can be represented by these patch embeddings. After that, images can be processed like sentences. This novel method leads to the invention of the Vision Transformer (ViT). After that, many successful methods in the NLP field can be slightly adjusted and used in the CV field. The ViT is proven to have a strong ability to extract image features by training with these different methods. Its feature extraction ability can be used in many downstream tasks.

Recently Meta AI introduced a powerful segmentation model called the Segment Anything Model (SAM) and released the model for people to test. Meta said that this novel model could segment images with prompts[1]. According to many people's experience, this model does a great job in

Submitted to NeurIPS 2021 AI for Science Workshop.

segmentation tasks. The SAM is composed of an image embedding part and a decoder part as Figure 1. The image embedding part is trained in a corresponding dataset (SA-1B) of 1B masks and 11M images, which means that the image embedding can extract useful features relevant to object detection for downstream tasks. The class-agnostic counting task actually needs such important features to improve its accuracy. Therefore, in this study, I use pre-trained SAM and connect it to a convolutional kernel. And I simply add the kernel output up as the counting result. Then I fine-tune the model in the FSC-147 dataset. Finally, this simple model achieves competitive results like the SOTA model.

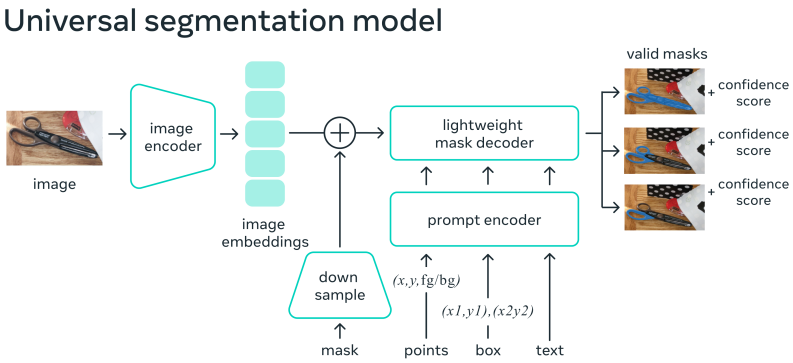

Figure 1: The main structure of the SAM. Since here we need to provide 3 box prompts. We will utilize the valid masks to connect to a 3-D kernel.

My key contributions are as follows:

- I introduce a really simple structure based on the SAM, which only has an extra convolutional kernel and a linear layer than the original one.
- The model training is based on the counting quantity. It does not refer to any density map when training. However, it can be trained to predict the density map-like array of an image.
- I get competitive results of MAE and RMSE in the validation dataset, which are better than many methods.

2 Related Work

2.1 Class-specific Counting

The class-specific counting has a long history. Even before the popularity of Neural Networks (NN) in the CV field, there are many useful counting methods introduced in the industry field and proved to be of high accuracy. However, these old methods are mainly designed for objects that have almost the same shape and size. Many objects like organisms may have different shapes even though they are of the same class. For these objects, we need to extract more detailed features to recognize them, which is hard to achieve before the NN. When Convolutional Neural Networks (CNN) were introduced in the CV field, people have a more powerful tool to extract features in images. Therefore, the more complex class-specific counting task can be solved successfully. In the meantime, many useful CNN structures are invented, which makes the counting model performs better.

Most counting models are trained by using density maps, which represent the density of the specific object. By pointing out the object in images and producing the corresponding density maps, people can train models by fitting outputs with them. It is not complex and has been proven to be an efficient method in training count models.

2.2 Class-agnostic Counting

The class-agnostic counting problem is introduced by a paper. In this paper, the group introduces the FSC-147 dataset and gives a basic method to solve it[2]. This problem is mainly concerned with the academic field. In the industry field, most people are more interested in specific objects. And they do care about the speed and accuracy of the counting process. Undoubtedly, class-agnostic counting models are hard to achieve both like class-specific counting ones.

However, studying the class-agnostic counting problem actually requires excellent feature extraction models. Therefore, the progress in this task represents the development of CV models. It is not hard to find that most of the time, the better CV models used in class-agnostic counting, the better results people will get.

The invention of the ViT model introduces some new training methods. For example, based on a given pre-trained ViT model, adding a linear layer after that will get a single output. If we utilize it as the counting number, we can train the model without the density map. And it can express a density map-like array representing the density map[3].

2.3 SAM

The SAM is recently introduced by Meta AI, Facebook. It is essentially a ViT model. With a given image and some prompts, it can automatically segment the image into different parts. It is composed of an image embedding part and a decoding part. The image embedding part is a huge ViT model. The training method of the SAM is nothing special. The Meta AI group just uses a mask method to train this ViT model. The most important contribution of this study is that they make a big dataset containing 1B masks and 11M images, which gives the SAM extreme power to extract image features for downstream tasks[1].

After that, many study groups are trying to utilize the SAM in other CV fields except segmentation. Without any training, if we give the SAM some boxes as prompts, it will segment every object out from the image. And this result can be used to count objects. There is a group that uses the SAM in the class-agnostic counting task. And they got 31.20 MAE, 100.83 RMSE in the validation dataset, and 27.97 MAE and 131.24 RMSE in the test dataset without any training[4]. Even though this result is not competitive with the SOTA model. It is still better than some earlier models.

Since the SAM gets a not-bad result without any extra training. In this study, I will try to fine-tune the SAM for a better result.

3 Methods

In my study, I just simply connect the SAM's image embedding output to a convolutional kernel. Then I just sum the output up as the counting result. The whole model is like Figure2. The aim of the convolutional kernel is trying to fit the density map. Although I will not actually provide it with any density maps, which means the model should learn this feature by counting labels.

It seems to be a weird method. Since it is not based on strong math knowledge. However, the study before has proved that this simple structure can work well with a proper pre-trained ViT model. This group linked a pre-trained ViT model with a linear layer. And they fine-tuned the model in the FSC-147 dataset without referring to density maps. Finally, they got an excellent result and the ViT backbone can output density-map-like arrays[3].

However, they do this without any prompts. They just simply input the image and get the counting number. It is not easy to add extra box prompts with many pre-trained ViT models. Since many ViT models only get the input of the images. If we want to add extra arguments, it may destroy the integrity of the ViT model. And the model will become much more complex. The too-complex model may need a larger dataset to train to adjust the new parameters. Therefore, this zero-shot method is not suitable to transfer into few-shot tasks.

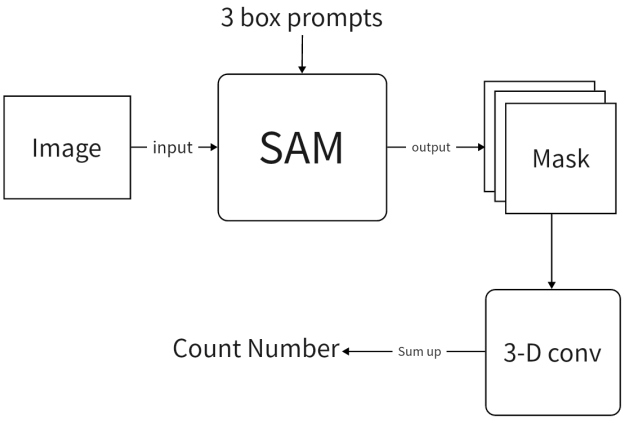

Figure 2: Input the 3 box prompts and an image, and the model will return the count number.

My model is like the model just introduced but slightly different. The SAM is designed to accept box prompts. If we give additional 3 box prompts for training, the result may become better. To strengthen the power of extracting the density map, I just add output elements to one value. Since if we assume the output is the density map. The sum of elements is proportional to the counting result.

4 Experiments

Since I did not find the test dataset. I just randomly split the whole dataset into two parts. The 80% is the training dataset and the rest is the validation dataset. After that, I finished the fine-tuning in 10 epochs. The result is as Table 1.

Table 1: The MAE and RMSE of Different Methods in the Validation Dataset

Methods	MAE	RMSE
Mean	53.38	124.53
Median	48.68	129.70
FR detector[5]	45.45	112.53
FSOD detector[6]	36.36	115.00
GMN[7]	29.66	89.81
MAML[8]	25.54	79.44
FamNet[9]	23.75	69.07
CFOCNet[10]	21.79	61.41
BMNet+[11]	15.74	58.53
SAFECount[12]	**15.28**	**47.20**
Ours	16.92	55.47

During the training, I found that even though the ViT-SAM-base model is the smallest SAM. It still costs a huge training burden. And I can not train this model directly. Since there are many parameters in the SAM. To make the training possible, I have to freeze the transformer layers in the SAM, which reduce the number of parameters needed to train. At the same time, the batch size can not be too big. In this paper, I choose 1 as the batch size. But do the optimize step every 8 batches. By doing all the above to reduce the training burden, I spend almost 4 hours finishing 10 epochs with a Titan XP. Therefore, in my experience, using the SAM for downstream tasks is of high requirement.

We can find that the result is more competitive than other models. My result is slightly worth than the SOTA method. However, it is competitive with the second-best method, which means my model is more suitable to count high-density scenes. Undoubtedly this result is quite acceptable. Since the model is trained without any density maps. But the accuracy is very close to the SOTA one.

Even though I did not utilize any density maps. However, instead of the counting result, people are more concerned about where objects are. Although the model can output a fairly accurate counting result. It should also output the density map of a given image. For this aim, I just visualize the kernel output and we can directly compare it with the given image as Figure 3.

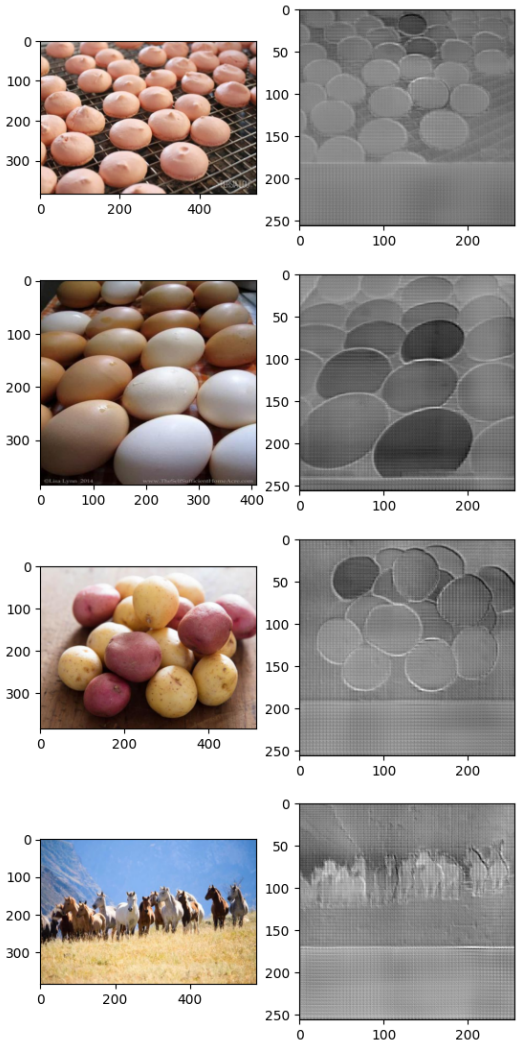

Figure 3: Randomly chosen images that processing through the SAM and the convolutional kernel. They are preprocessing into 1024×1024 pixels by lengthening the longest edge and padding another one with zero. Therefore the output is a square shape. The output shows clear edges that separate counting objects, which benefits from the strength of the segmentation ability of the SAM.

We can find that, just like the density map, the heat map can show where objects are. And it shows the edges even much clearer, which means even though the model did not refer to the density map when training. It still learned how to detect objects in the image. This result gives my model more interpretability.

Due to the time limit of the paper deadline. I just randomly chose the ViT-SAM-base model on the Hugging Face platform, which is released officially by Facebook. After that, I chose MAE as the loss function and Adam as the optimizer that selects 10^{-4} as the learning rate. There are many hyperparameters that are chosen by experience. And the whole model lacks more detailed fine tunes. Even though it gets a fairly excellent result. It can perform better with further studies.

5 Conclusion

In my study, I choose ViT-SAM as the backbone and connect it to a simple structure. Finally, I get a fairly good result compared with the SOTA one. This study proves that the pre-trained ViT model is suitable for the class-agnostic counting task. Without referring to the density map, it still can get good results after simple training. However, it is just a rough study of the SAM. Undoubtedly, the SAM has an extremely good feature extraction ability since it is trained in such a big dataset. Here we only add a 3-D convolutional kernel that has a few extra parameters. The final sum layer actually has no parameters, which means that we hardly add more parameters than the original SAM. We can try adding more layers after the SAM's output, like adding a simple CNN structure after it for a better fit of the density map. Or we can deprecate some layers of the SAM and connect it with other structures. In the meantime, in my study, I actually got 9.67 and 25.66 as MAE and RMSE, which means my model needs more fine-tuning. These experiments may result in a more accurate and more interpretable model.

References

[1] Alexander Kirillov et al. *Segment Anything*. 2023. eprint: arXiv:2304.02643.

[2] Viresh Ranjan et al. *Learning To Count Everything*. 2021. eprint: arXiv:2104.08391.

[3] Michael Hobley and Victor Prisacariu. *Learning to Count Anything: Reference-less Class-agnostic Counting with Weak Supervision*. 2022. eprint: arXiv:2205.10203.

[4] Zhiheng Ma, Xiaopeng Hong, and Qinnan Shangguan. *Can SAM Count Anything? An Empirical Study on SAM Counting*. 2023. eprint: arXiv:2304.10817.

[5] Bingyi Kang et al. *Few-shot Object Detection via Feature Reweighting*. 2018. eprint: arXiv:1812.01866.

[6] Qi Fan et al. *Few-Shot Object Detection with Attention-RPN and Multi-Relation Detector*. 2019. eprint: arXiv:1908.01998.

[7] Erika Lu, Weidi Xie, and Andrew Zisserman. *Class-Agnostic Counting*. 2018. eprint: arXiv:1811.00472.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1126–1135. URL: https://proceedings.mlr.press/v70/finn17a.html.

[9] Viresh Ranjan et al. "Learning To Count Everything". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[10] Shuo-Diao Yang et al. "Class-Agnostic Few-Shot Object Counting". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021, pp. 870–878.

[11] Min Shi et al. "Represent, Compare, and Learn: A Similarity-Aware Framework for Class-Agnostic Counting". In: June 2022, pp. 9519–9528. DOI: 10.1109/CVPR52688.2022.00931.

[12] Zhiyuan You et al. "Few-Shot Object Counting With Similarity-Aware Feature Enhancement". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2023, pp. 6315–6324.