DETECT ANY OBJECT INSIDE THE SCENE WITH MINIMAL TRAINING

Justin Nguyen

Deep Learning Engineer – Computer Vision, May 2023

ABSTRACT

This report answers how a single object type can be detected within an image, without the time and cost of data collection, labeling, and training. The following report will delve into publicly available models tailored for image recognition, such as the Vision Transformer (ViT), OpenAI's Contrastive Language-Image Pre-training (CLIP), and Meta's Segment Anything Model (SAM).

Index Terms – deep learning, object detection, ViT, CLIP, SAM, pre-trained, transfer learning, zero-shot, one-shot.

1. INTRODUCTION

Object detection with deep learning involves collecting a large dataset of images, drawing an outline around the object of interest, and assigning a class label to each image [1]. This is a time-consuming task with a large labour cost attached, hence why many users opt for a traditional computer vision approach. Traditional computer vision and image processing techniques typically utilize OpenCV and colour spaces [2], however, there are also intricate hurdles to overcome, such as shadows. Additionally, some of these hurdles cannot be overcome while maintaining its accuracy and consistency. With traditional techniques, it is often trying to maintain a delicate balance of accuracy and robustness. Deep learning was introduced to solve this issue, albeit with trade-offs of its own.

The scientific challenge that this report will focus on is the use of deep learning models in the field of single object detection. The problem presented describes a scenario where a client or user desires a more efficient approach to collecting images, labeling, and training a model for a single object by requiring only one training sample. To alleviate this problem, there are existing models that are pre-trained on a large dataset that have the capability to utilize transfer learning [3]. The following pre-trained models are investigated in this report:

- Vision Transformer (ViT)
- Contrastive Language-Image Pre-training (CLIP)
- Segment Anything Model (SAM)

2. LITERATURE REVIEW

2.1 Vision Transformer (ViT)

Vision Transformer, or ViT, is a pre-trained model that utilizes the Transformer architecture, which is normally used for natural language processing [4]. A transformer is a deep learning model that uses the mechanisms of attention, which differentially weighs the significance of each part of the input sequence of data [5]. ViT leverages this self-attention mechanism to capture the relationships between different image patches. Briefly, each image is split into a sequence of same-sized patches. These patches are linearly embedded, and then fed into the Transformer encoder from which the output can be used for classification [4].

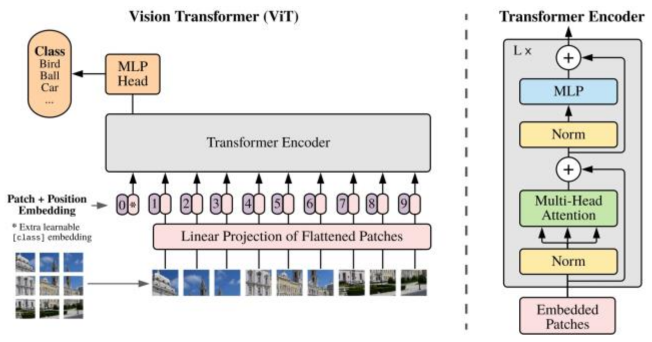

Figure 1 [4]: ViT architecture.

ViT is normally pre-trained on the ImageNet-21k dataset, containing 14 million images and 21,843 classes [6]. A pre-trained ViT may be able to solve the client or user's problem, although if the object is not observed in the pre-training dataset, transfer learning must be utilized. Transfer learning is the process of adapting a representation learned while solving one problem, to a different but related problem [7]. This technique is meant for situations where the dataset has too little data to train a full-scale model from. The problem presented requires only one training sample, which is not enough to utilize transfer learning for a robust model. Additionally, ViT is not designed for one-shot learning, where the model is able to learn from a single example of a class. Heavy modifications would need to be done to the model to make it capable of one-shot learning.

2.2 Contrastive Language-Image Pre-training (CLIP)

Another pre-trained model is Contrastive Language-Image Pre-training, or CLIP. CLIP is a neural network trained on 400 million image-text pairs and can be instructed in natural language to predict the most relevant text snippet given an image that the model has not been optimized for [8]. This concept is called zero-shot learning, wherein a learner encounters samples from classes that were not seen during training and must accurately predict the class to which they belong. This is done based on auxiliary information, such as textual description [9].

CLIP is a multi-modal vision and language model that adopts ViT-L as the image encoder and uses a casual language model for text features [10].

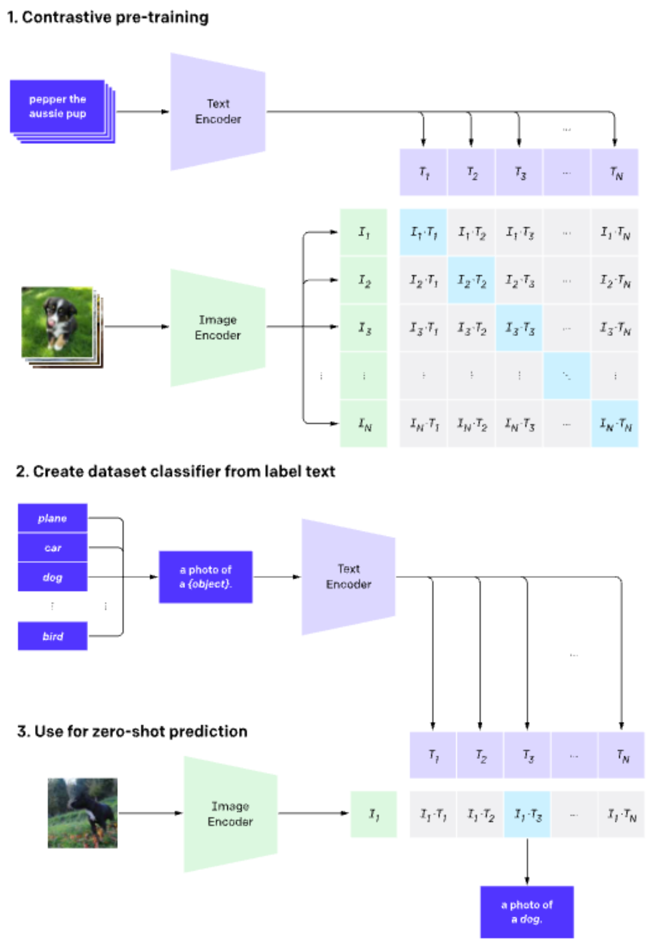
1. Contrastive pre-training
pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ... I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
3. Use for zero-shot prediction
Image Encoder
T_1 T_2 T_3 ... T_N
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
a photo of a dog.

Figure 2 [11]: CLIP architecture.

Despite CLIP performing well for recognizing common objects, it struggles with more abstract tasks such as counting the number of objects [11]. In addition, CLIP has poor generalization to images not well-represented in the pre-training dataset, and it requires an accurate wording or phrasing to perform well.

If the client or user is trying to detect an object bearing some resemblance to those present in the pre-training dataset, CLIP's zero-shot classifier capabilities could be enough to meet their demand. If it is an object that diverges further from what it is trained on, the classifier will perform worse [11]. CLIP is a zero-shot classifier, and like ViT, it is not designed for one-shot learning and would require modifications to the model [12].

2.3 Segment Anything Model (SAM)

The Segment Anything Model, or SAM, can generate accurate object masks based on input prompts like points or boxes. It excels at producing masks for multiple objects within an image. SAM has been trained on an extensive dataset consisting of 11 million images and 1.1 billion masks, as a result, it demonstrates impressive zero-shot performance on various segmentation tasks [13].

SAM uses a form of interactive segmentation with prompts (e.g., points, boxes, and masks). Unlike normal interactive segmentation that will eventually predict a mask after enough user input, SAM is able to predict a valid mask for any prompt even if it is ambiguous. This results in a robust model that performs well in use-cases that involve ambiguity. Additionally, this enables the model with the ability to respond accurately to any prompt at inference time. This means that the model can perform segmentation on an object that has not been pre-trained. As a result, SAM performs well in zero-shot classification as it can transfer zero-shot to new image distributions and tasks [14].

The architecture that SAM deploys has three components, as seen in Figure 3. SAM contains an image encoder, a flexible prompt encoder, and a fast mask decoder, which is built on Transformer vision models [14].

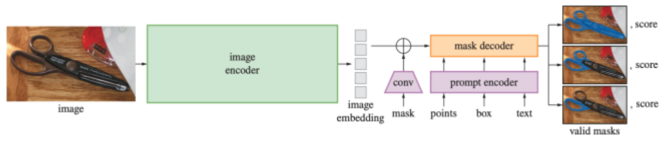

Figure 3 [14]: SAM architecture.

Between SAM and CLIP, SAM seems to be the most promising model in terms of zero-shot classification, however, the given problem is a one-shot problem. For classifying a new class in SAM, the user would be required to provide multiple samples and masks of the target object in different poses or contexts to the model for accurate and robust segmentation [15]. This would be a form of few-shot learning.

3. METHODOLOGY

3.1 Personalize Segment Anything Model (PerSAM)

Adapting ViT, CLIP, and SAM for one-shot classification is a time-consuming task that would introduce extra training time and computational resources. Through research, there are some implementations of CLIP for few-shot classification, however, there are no implementations that directly solve the given problem efficiently.

The problem with SAM in classifying a new class can be solved with a specialized SAM model, called Personalize Segment Anything Model (PerSAM) [15]. PerSAM is a training-free personalization approach for SAM, which customizes SAM using only one-shot data. The one-shot data is in the form of an image and a rough mask of the object. With PerSAM, the user is only required to give a single image with a reference mask, from which PerSAM can segment specific visual concepts [16]. Additionally, there is PerSAM-F which further enhances segmentation accuracy by fine-tuning two parameters within 10 seconds [15].

To solve the problem of a single type of object detection, the following steps can be done utilizing PerSAM with reference to the tutorial notebook [17].

1. **Setup**:
    a. Obtain any number of test images and a reference image, then create an accurate or rough mask for the target.
    b. Set up the environment and install the Transformers library.
    c. Load the PerSAM model from the Transformers library, then load the reference image, mask, and test image(s).
2. **Get target embedding** from the reference image and mask, which is a numerical representation of the target concept.
3. **Calculate cosine similarity**, to find how close the target concept is related to each part of the test image, creating a location confidence map.
4. **Obtain location priors**, which indicate where the target concept might or might not be in the test image.
5. **First step prediction**, which prompts SAM to generate a mask for the target concept in the test image.
6. **2-step cascaded post-refinement**, to improve the output segmentation mask.
7. **Visualize the mask** on the test image.

The steps outlined can be directly utilized to perform instance segmentation on a single type of object from only one training sample.

3.2 Limitation

One limitation inherent to the current PerSAM architecture is its ability to handle only a single instance of an object within an image. In Issue #9 of the PerSAM GitHub repository, the owner states that PerSAM can only handle a single instance of an object or multiple objects of different classes in an image [18]. It is not able to handle multiple instances of the same object in a single image. The owner explains the reason in Issue #11, with how SAM contains no semantics for visual categories, meaning the model is not able to understand the meaning and context of the objects beyond its visual features [19]. The owner then provides a theoretical solution; by incorporating CLIP, it may allow PerSAM to gain semantics for visual categories, thereby enabling PerSAM to recognize all objects of the same class.

CLIP bridges visual and textual understanding, allowing the model to understand images in context of natural language, thus embedding a semantic understanding in the visual recognition tasks. Incorporating CLIP into PerSAM and creating a cohesive model is a time-consuming and complex task though.

Another solution described in Issue #9; the owner provided four steps to find two objects of the same class within an image [18]:

1. “Use image encoder to obtain the image feature only once and calculate the feature similarity map.”
    i. We only need to obtain the image features once from the image.
    ii. The feature similarity map identifies areas of the image that are similar in terms of their features (calculating cosine similarity).
2. “Find the location prior and segment the first object by the decoder, obtaining mask M_1.”
    i. Get the location prior (the general area of where the object might be found) of the first object the model detects.
    ii. From the location prior, we can use the decoder to segment the first object by creating a mask (M_1) on it.
3. “Assign zeros to the pixels within M_1 on the feature similarity map, and then find the second location prior.”
    i. By assigning zeros to the pixels within M_1 (the first object detected), this effectively removes the object from the image and prevents it from being detected again.
4. “Segment the second object by the decoder.”
    i. Performing step 2 again to obtain the second object.

These steps could be adjusted to continue looping steps 2-3 until no additional objects of the same class can be found.

To determine whether there are more objects of the same class, we can calculate the similarity score between the mask of newly detected objects with previously detected ones. If the similarity score is low, then it is likely that the new object is of a different class. However, this method may not be the most accurate solution and fine-tuning the model with semantic information may be ideal for consistency and accuracy, especially for industrial usage.

4. CONCLUSION

This report investigated three different pre-trained models: ViT, CLIP, and SAM to solve the given problem scenario. In their base form, ViT, CLIP, and SAM are not designed for one-shot learning, however, they can be adapted for one-shot learning.

A modified version of SAM, called PerSAM, was chosen as the ideal model to perform instance segmentation on a single type of object as it is designed for one-shot learning. Most importantly, it is training-free, providing the user with a simple and efficient process.

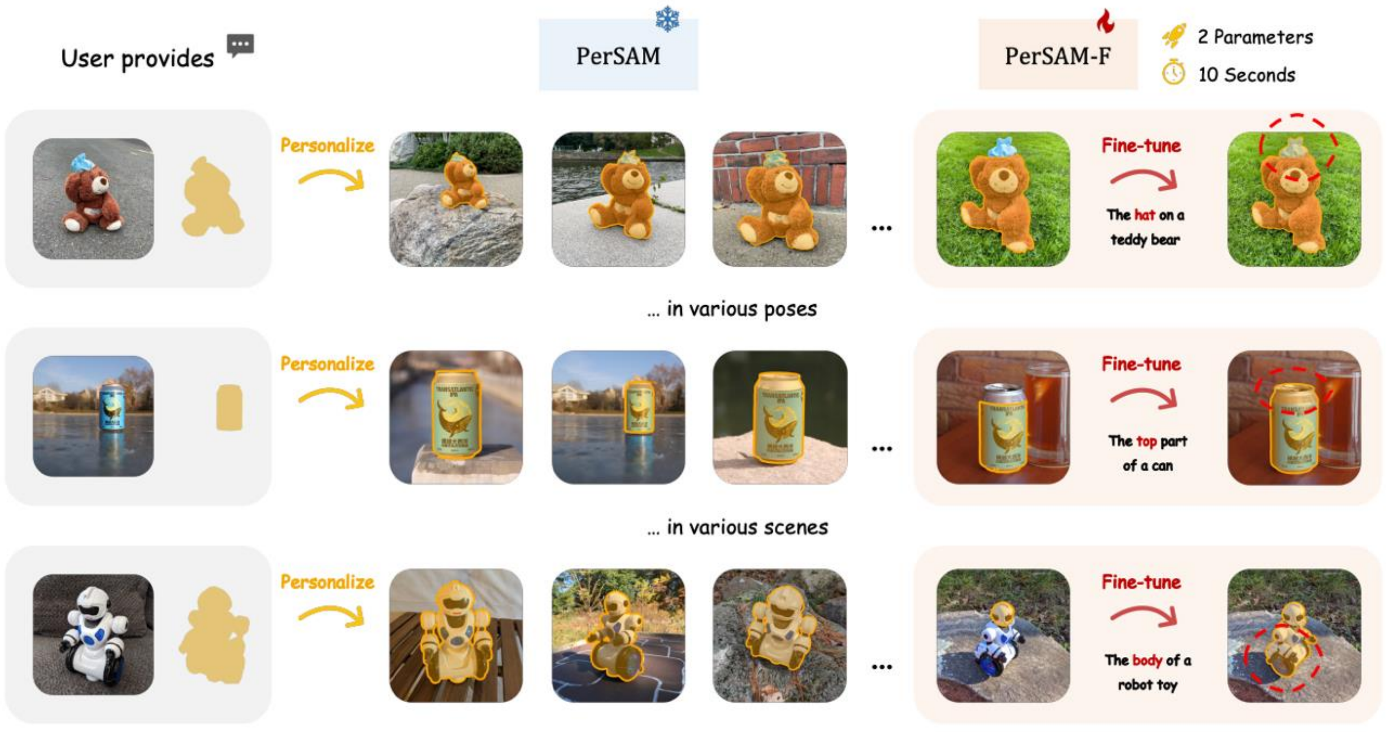

Figure 4 [15]: Personalization examples utilizing PerSAM.

5. REFERENCES

[1] J. Brownlee, “A gentle introduction to object recognition with deep learning,” MachineLearningMastery.com, https://machinelearningmastery.com/object-recognition-with-deep-learning/ (accessed May 25, 2023).

[2] J. Bullock, “Real-time object detection without machine learning,” Medium, https://towardsdatascience.com/real-time-object-detection-without-machine-learning-5139b399ee7d (accessed May 25, 2023).

[3] J. T. Raj, “Transfer learning : Why train when you can finetune?,” Medium, https://towardsdatascience.com/transfer-learning-picking-the-right-pre-trained-model-for-your-problem-bac69b488d16#:~:text=A%20pre%2Dtrained%20model%20is,was%20originally%20trained%20to%20solve. (accessed May 25, 2023).

[4] “Vision Transformer (VIT),” Vision Transformer (ViT), https://huggingface.co/docs/transformers/model_doc/vit (accessed May 25, 2023).

[5] G. Boesch, “Vision transformers (VIT) in image recognition - 2023 guide,” viso.ai, https://viso.ai/deep-learning/vision-transformer-vit/ (accessed May 25, 2023).

[6] “Google/VIT-large-patch32-384 · hugging face,” google/vit-large-patch32-384 · Hugging Face, https://huggingface.co/google/vit-large-patch32-384 (accessed May 25, 2023).

[7] J. Brownlee, “A gentle introduction to transfer learning for Deep learning,” MachineLearningMastery.com, https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (accessed May 25, 2023).

[8] Openai, “Openai/CLIP: CLIP (Contrastive Language-image pretraining), predict the most relevant text snippet given an image,” GitHub, https://github.com/openai/CLIP (accessed May 25, 2023).

[9] “Zero-shot learning,” Wikipedia, https://en.wikipedia.org/wiki/Zero-shot_learning (accessed May 25, 2023).

[10] CLIP, https://huggingface.co/docs/transformers/model_doc/clip (accessed May 25, 2023).

[11] “Clip: Connecting text and images,” CLIP: Connecting text and images, https://openai.com/research/clip (accessed May 25, 2023).

[12] N. Kafritsas, “Clip: The most influential AI model from openai- and how to use it,” Medium, https://towardsdatascience.com/clip-the-most-influential-ai-model-from-openai-and-how-to-use-it-f8ee408958b1#:~:text=CLIP%20is%20a%20zero%2Dshot,model%2C%20such%20as%20a%20ResNet.&text=CLIP%20significantly%20outperforms%20the%20other%20classifiers. (accessed May 25, 2023).

[13] Facebookresearch, “Facebookresearch/segment-anything: The repository provides code for running inference with the segmentanything model (SAM), links for downloading the trained model checkpoints, and example notebooks that show how to use the model.,” GitHub, https://github.com/facebookresearch/segment-anything (accessed May 25, 2023).

[14] A. Kirillov et al., “Segment Anything,” arXiv preprint arXiv:2304.02643, 2023.

[15] R. Zhang et al., “Personalize Segment Anything Model with One Shot,” arXiv preprint arXiv:2305.03048v1, 2023.

[16] ZrrSkywalker, “ZrrSkywalker/personalize-SAM: Personalize segment anything model (Sam) with 1 shot in 10 seconds,” GitHub, https://github.com/ZrrSkywalker/Personalize-SAM (accessed May 26, 2023).

[17] NielsRogge, “Transformers-tutorials/personalize_sam_with_one_shot_using_hugging_face.ipynb at master · nielsrogge/transformers-tutorials,” GitHub, https://github.com/NielsRogge/Transformers-Tutorials/blob/master/PerSAM/Personalize_SAM_with_one_shot_using_Hugging_Face.ipynb (accessed May 26, 2023).

[18] ZrrSkywalker, “Multipole objects · issue #9 · Zrrskywalker/personalize-sam,” GitHub, https://github.com/ZrrSkywalker/Personalize-SAM/issues/9 (accessed Jun. 3, 2023).

[19] ZrrSkywalker, “Multiple objects issue · issue #11 · Zrrskywalker/personalize-sam,” GitHub, https://github.com/ZrrSkywalker/Personalize-SAM/issues/11 (accessed Jun. 3, 2023).