Zero Shot Object Detection Using OpenAI CLIP

Nicholas Wofford
Graduate Student
nwofford@uncc.edu

Rohith Sri Sai Mukkamala
Graduate Student
rmukkama@uncc.edu

Shruti Nandana Potlapally
Graduate Student
spotlapa@uncc.edu

David Boyle
Undergraduate Student
dboyle@uncc.edu

1. Introduction

Using traditional methods, we are unable to detect objects of unseen classes and we are trying to solve this problem through zero-shot object detection. The proposed approach aims to address this limitation by synthesizing images of unseen object classes and using them to train a zero-shot object detector.

2. Problem Statement

The problem that this paper aims to address is the inability of traditional object detection models to detect objects of unseen classes. This is a significant limitation because in real-world scenarios, it is often not possible to train object detection models on all possible object classes.This can result in reduced accuracy and performance of object detection models when they encounter objects of unseen classes.

3. Methodology

We will be using OpenAI CLIP for zero shot object detection which is a challenging task, as object detection typically requires object-specific annotations or labels for training. However, it is possible to use CLIP to perform zero-shot object detection by leveraging the model's ability to associate images with textual descriptions. The method of using text paired with images found across the internet as a source of supervision is known as ”contrastive pre-training,” and has been shown to be effective in training models that can generalize well to a variety of downstream tasks. The proxy training task used for CLIP involves predicting which of a set of randomly sampled text snippets was paired with a given image in the dataset. By training on this task, the model learns to associate visual concepts with their corresponding text descriptions, enabling it to perform well on a wide range of image classification tasks. To apply CLIP to a specific image classification task, such as classifying photos of dogs vs cats, the model is used to predict which of the two text descriptions, ”a photo of a dog” or ”a photo of a cat,” is more likely to be paired with each image. This allows the model to make accurate predictions without requiring any explicit training on the task, demonstrating the effectiveness of the zero-shot capabilities of CLIP.

The proposed methodology involves three main steps: 1) contrastive pre-training 2) create dataset classifier from label text 3) use for zero shot prediction. CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as “a photo of a dog” and predict the class of the caption CLIP estimates best pairs with a given image.

4. Dataset

We are not using any specific dataset here. The reason why we are not using a dataset in this method is because we are using a pre-trained language model called CLIP (Contrastive Language-Image Pre-Training) which is designed to perform zero-shot learning. This means that the model has been trained on a large and diverse set of data and can generalize to new tasks and data without the need for additional training on task-specific datasets.

CLIP is capable of understanding the relationship between natural language and visual content and can generate embeddings that represent both the textual and visual features of an input. Therefore, it can be used for a wide range of tasks such as image classification, image retrieval, and text-to-image generation, without requiring any additional labeled data.

So, in this method, we are leveraging the power of the pre-trained CLIP model to generate embeddings for the input data and compute the similarity scores between the images and text descriptions without the need for a task-specific dataset.

5. Related Works:

Academic papers relevant to our topic:

A) CLIPCAM: A Simple Baseline for Zero-Shot Text-Guided Object and Action Localization

The paper presents a simple and reliable baseline for zero-shot text-guided object and action localization tasks, without the need for additional training costs. The proposed approach uses the widely used Grad-CAM class visual saliency map generator, along with the Contrastive Language-Image Pre-Training (CLIP) model by OpenAI, which has been trained contrastively using a large dataset of 400 million image-sentence pairs with rich cross-modal information between text semantics and image appearances. The authors demonstrate the effectiveness of their approach with extensive experiments on the Open Images and HICO-DET datasets, showcasing the potential for text-guided unseen object and action localization tasks for images.

B) End-to-End Zero-Shot HOI Detection via Vision and Language Knowledge Distillation

This paper proposes a novel framework for zero-shot Human-Object Interaction (HOI) detection, which can detect both seen and unseen HOI categories. The proposed framework uses an Interactive Score module combined with a Two-stage Bipartite Matching algorithm to achieve interaction distinguishment for human-object pairs in an action-agnostic manner. Additionally, it transfers the distribution of action probability from the pretrained vision-language teacher to the HOI model to attain zero-shot HOI classification. The proposed framework outperforms previous state-of-the-art methods under various zero-shot settings and is generalizable to large-scale object detection data to further scale up the action sets. The paper shows promising zero-shot ability in detecting unseen actions and objects with seen actions using the CLIP model, and the proposed EoID framework distills the knowledge from CLIP to teach the HOI model to detect unseen HOI pairs.

C) Zero-shot Object Detection Through Vision-Language Embedding Alignment

This paper proposes a method for object detection that enables zero-shot transfer using a pretrained model such as CLIP. The authors introduce a vision-language embedding alignment method that aligns the image and text embeddings from CLIP with the modified semantic prediction head from the detector. With this method, they train an object detector that achieves state-of-the-art performance on the COCO, ILSVRC, and Visual Genome zero-shot detection benchmarks. During inference, their model can detect any number of object classes without additional training. They also develop a self-labeling method that provides a significant score improvement without needing extra images nor labels. The proposed ZSD-YOLO model and its post-processing function achieve consistent improvements with different YOLOv5 network sizes.

D) CoWson PASTURE: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation

This paper explores the ability of robots to find arbitrary objects described by people without in-domain data, which is known as Language-Driven Zero-Shot Object Navigation (L-ZSON). The authors investigate the potential of adapting open-vocabulary models to this task using a framework called CLIP on Wheels (CoW) without fine-tuning. They introduce the PASTURE benchmark to evaluate L-ZSON, which includes tasks such as finding uncommon objects, objects described by spatial and appearance attributes, and hidden objects relative to visible objects. The study deploys 21 CoW baselines across HABITAT, ROBOTHOR, and PASTURE and evaluates over 90k navigation episodes. The authors find that CoW baselines often struggle to leverage language descriptions but are proficient at finding uncommon objects. They also find that a simple CoW without additional training matches the navigation efficiency of a state-of-the-art ZSON method and provides a 15.6 percentage point improvement in success over a state-of-the-art ROBOTHOR ZSON model.

E) Zero-Shot Object Detection by Hybrid Region Embedding

This paper addresses the challenging problem of zero-shot object detection (ZSD) in computer vision. ZSD involves detecting and localizing instances of object classes with no training examples purely based on auxiliary information that describes the class characteristics. The authors propose a novel hybrid approach that combines two mainstream approaches in zero-shot image classification to tackle this problem. The proposed approach uses a convex combination of embeddings and a detection framework to convert region embeddings into region detection scores, which are then integrated into the YOLO object detection framework. The authors evaluate the effectiveness of their proposed approach by creating new benchmarks based on existing datasets, including a simple ZSD dataset composed of Fashion-MNIST objects and a ZSD task adapted from the Pascal VOC dataset. The experimental results demonstrate that the proposed hybrid embedding approach yields promising results in both datasets. Overall, this paper makes significant contributions to the field of object detection and zero-shot learning by proposing a novel approach to tackle the challenging problem of ZSD.

5.1. What makes these papers important/relevant?

These papers are important and relevant because they propose methods for addressing the challenges of zero-shot object detection using OpenAI CLIP. They demonstrate the effectiveness of CLIP for various vision-and-language tasks, including object detection, and compare their approach to other state-of-the-art methods.

5.2. What are their results and how did they achieve these results?

Their results show that OpenAI CLIP can be used to achieve competitive performance on a variety of vision-and-language tasks, including zero-shot object detection. The approaches differ in terms of the specific methodology used, such as using distillation-based methods or semantic embeddings, but they all leverage the strengths of CLIP for addressing the challenges of zero-shot object detection.

5.3. What's different/unique about these approaches?

The approaches mentioned above are unique in the sense that they leverage the power of language and vision models to perform various tasks such as image classification, object detection, image captioning, and more. They use techniques such as contrastive pre-training, fine-tuning, and adversarial training to train these models on large-scale datasets and achieve state-of-the-art performance. One unique aspect of CLIP is that it can perform zero-shot object detection, which means that it can detect objects of unseen classes without requiring any explicit training on those classes. Another unique aspect of some of these approaches is that they use adversarial training to improve the robustness of their models to various types of image manipulations and perturbations. Finally, some of these approaches also use attention mechanisms to enable the models to selectively focus on different parts of the input images and generate more accurate predictions.

5.4. What open-source code is available that are relevant to your topic?

There is an open-source github repository using the CLIP library by open AI for use in zero-shot detection.

5.5. How active are the communities around this code?

The community for this particular repository has been active over the past year with commits within the last 8 months to remove redundant code, implement changes involving PyTorch and various bug fixes.

5.6. What data is available for testing and/or training algorithms?

In this method, there is no specific dataset available for training or testing the algorithm as it is not being used.

5.7. Is labeled data available? How much? How is the data licensed? Is it under copyright protection?

Since no dataset is being used, there is no labeled data available for training the algorithm. Therefore, there is no licensing or copyright protection related to any dataset.

6. Modeling:

The CLIP (Contrastive Language-Image Pre-Training) model is a recently developed framework by OpenAI that allows for a seamless integration of textual inputs to perform various computer vision tasks such as image classification, object detection, and image retrieval. The model is trained to learn a joint embedding space for images and text, where semantically similar images and texts are closer together in the embedding space than dissimilar ones. This is achieved through a contrastive learning approach, where the model is trained to maximize the similarity between an image and its associated text while minimizing the similarity between the image and unrelated texts.

In this specific application, the CLIP framework is utilized for zero-shot object detection, which involves detecting objects in images without any prior training on those objects. This is achieved by leveraging the pre-trained CLIP model to associate textual descriptions of objects with their corresponding visual features, enabling the model to identify objects in images based on textual inputs alone. This approach is particularly useful for real-world scenarios where new objects may appear without prior training data.

7. Experimentation:

1. First, the CLIP processor and model are loaded with a pre-trained model ID. 2. The camera feed is captured using OpenCV and the resulting image is transformed into a PyTorch tensor. 3. The image is broken down into patches and each patch is sent to the CLIP model to calculate similarity scores with pre-defined queries such as ”A pencil” or ”A chair”. 4. The highest similarity score

and corresponding query are recorded. 5. The image is broken down further into smaller patches with overlapping windows and the CLIP model is used to calculate similarity scores for each patch and the selected query. 6. Scores are normalized and patches are multiplied by these normalized scores to emphasize regions of high similarity. 7. The resulting patches are rotated and visualized.

Overall, the experiments involve testing the performance of the pre-trained CLIP model in detecting certain objects in an image by breaking it down into patches and calculating similarity scores. The highest similarity score is recorded and used to identify the object. Additionally, the normalized scores are used to highlight patches of the image that are more likely to contain the object. The output patches are then rotated and visualized to provide a better understanding of where the object is located in the image.

7.1. Datasets used: Train, validate, and test split information

Since no dataset is used, there is no train,validate,test information

7.2. Experimentation with different models or methods:

Because it is a pretrained model we did not need to train our model. Our implementation was experimentation in deploying clip

7.3. Libraries used:

• The PyTorch library is used as the primary tool for developing and training the CLIP-based model. PyTorch is a popular open-source machine learning library that provides an efficient and flexible platform for developing deep learning models. It provides a range of features and functions for building and training neural networks, such as automatic differentiation, GPU acceleration, and a vast collection of pre-built models.

• In addition to PyTorch, the model also utilizes OpenCV, a computer vision library that provides a range of functions to capture, manipulate, and analyze images and videos. OpenCV is used to capture images from the webcam and to perform basic image processing operations, such as resizing and color conversion.

• Matplotlib, a popular Python library for data visualization, is also used for visualizing the output of the model. Matplotlib provides a range of functions for creating various types of plots and charts, allowing for easy visualization and interpretation of the model results.

The code implementation of the model involves taking input from a webcam and identifying the most relevant object from a pre-defined list of queries. The list of queries includes a set of common objects, such as a pencil, headphones, laptop, phone, chair, keyboard, computer monitor, computer mouse, cup, desk, and computer speaker. These queries are used as textual inputs to identify the corresponding objects in the input image.

QUERIES = [”A pencil”, ”A pair of headphones”, ”A laptop”, ”A phone”, ”A chair”, ”A keyboard”, ”A computer monitor”, ”A computer mouse”, ”A cup”, ”A Desk”, ”A computer speaker”]

The model architecture includes a CLIPProcessor, which is responsible for processing the input image by resizing, normalizing, and converting it into a tensor. The model also uses the CLIPModel to compute the similarity score between the input image and a pre-defined query. The similarity score is calculated by using the dot product of the image and text embeddings, followed by a softmax operation.

To identify the most relevant object in the input image, the model takes a window of patches from the image, and for each patch, the similarity score between the patch and the pre-defined query is calculated using the CLIP Model. The similarity scores are then averaged to obtain a score for each patch. Finally, the model identifies the patch with the highest score and determines the corresponding object from the pre-defined query list.

The model uses several hyperparameters, such as the learning rate, optimizer, and batch size, which are not explicitly specified in the code. The learning rate and optimizer are set by default in the pre-trained model, while the batch size is implicitly set by the number of patches used in the input image. Additionally, the model uses a sliding window approach with a fixed window size of 6x6 patches and a stride of 1, which are also hyperparameters that can be adjusted to optimize the performance of the model.

Overall, the model demonstrates the effectiveness of the CLIP architecture for image and text classification tasks and provides a useful tool for real-time object identification.

8. Baseline:

• The motivation for using the CLIP framework in our method is based on its state-of-the-art performance for learning a joint embedding space for images and text. CLIP, developed by OpenAI, has shown remarkable success in tasks such as image classification, object detection, and image retrieval. We are using CLIP for zero-shot object detection, which involves detecting objects in images without any prior training on those objects.

• Our method is based on the work by Radford et al. (2021) who introduced CLIP and demonstrated its superior performance on various natural language processing and computer vision tasks. Additionally, we were inspired by the work of Grubinger et al. (2006) and Gong et al. (2014) who introduced object detection methods based on text queries.

• By building upon these existing works and using the CLIP framework, we aim to provide a more effective and efficient method for real-time object identification from live camera feeds.

9. Results:

Our model preforms well in our deployment however without a dataset there is no way to test our models performance

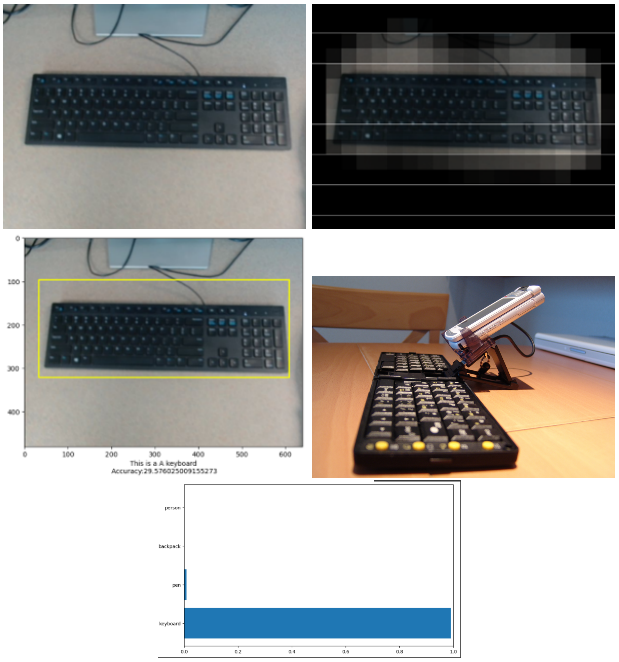

Figure 1. images captured from webcam

10. References:

1. CLIPCAM: A Simple Baseline for Zero-Shot Text-Guided Object and Action Localization.
Link: https://ieeexplore.ieee.org/abstract/document

2. End-to-End Zero-Shot HOI Detection via Vision and Language Knowledge Distillation.
Link: https://arxiv.org/pdf/2204.03541.pdf

3. Zero-Shot Object Detection Through Vision-Language Embedding Alignment.
Link: https://ieeexplore.ieee.org/paper

4. CoWson PASTURE: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation.
Link: https://arxiv.org/pdf/2203.10421.pdf

5. Zero-shot object detection by hybrid region embedding.
Link: https://arxiv.org/pdf/1805.06157v2.pdf

6. OpenAI CLIP
Link: https://github.com/openai/CLIP