# CLIPCAM: A SIMPLE BASELINE FOR ZERO-SHOT TEXT-GUIDED OBJECT AND ACTION LOCALIZATION

*Hsuan-An Hsia*[1⋆] *Che-Hsien Lin*[1⋆] *Bo-Han Kung*[1⋆] *Jhao-Ting Chen*[1⋆]
*Daniel Stanley Tan*[1] *Jun-Cheng Chen*[1] *Kai-Lung Hua*[2]

[1]Research Center for Information Technology Innovation, Academia Sinica
[2]National Taiwan University of Science and Technology

## ABSTRACT

The key for the contemporary deep learning-based object and action localization algorithms to work is the large-scale annotated data. However, in real-world scenarios, since there are infinite amounts of unlabeled data beyond the categories of publicly available datasets, it is not only time- and manpower-consuming to annotate all the data but also requires a lot of computational resources to train the detectors. To address these issues, we show a simple and reliable baseline that can be easily obtained and work directly for the zero-shot text-guided object and action localization tasks without introducing additional training costs by using Grad-CAM, the widely used class visual saliency map generator, with the help of the recently released Contrastive Language-Image Pre-Training (CLIP) model by OpenAI, which is trained contrastively using the dataset of 400 million image-sentence pairs with rich cross-modal information between text semantics and image appearances. With extensive experiments on the Open Images and HICO-DET datasets, the results demonstrate the effectiveness of the proposed approach for the text-guided unseen object and action localization tasks for images.

***Index Terms*—** text-guided, zero-shot, localization, CLIP, CAM

## 1. INTRODUCTION

With the recent rapid development of deep learning, deep learning-based algorithms for object and action localization and other vision tasks for images have achieved satisfactory performances. The key to success for these models heavily relies on the large-scale annotated datasets or pre-trained models trained on those annotated data. However, there are still an infinite amount of unlabeled data and classes around the world not covered by publicly available labeled datasets. It takes enormous costs and efforts to annotate all the data and to train the model. Furthermore, a successful prediction in computer vision tasks usually requires plenty of prior knowledge of interaction and a combination of different contextual information, especially for unseen scenarios. Therefore, unseen object and action localization with a limited amount of annotated training data for known classes and computational power is still a challenging and ongoing research area in the computer vision and machine learning communities with plenty of applications for visual surveillance, autonomous driving, healthcare, etc.

To address these challenges, we show a simple and effective baseline for zero-shot text-guided object and action localization tasks can be obtained by combining Grad-Cam and the recently released Contrastive Language-Image Pre-Training (CLIP) model [1] by OpenAI, where Grad-Cam proposed by [2, 3] is a widely used class visual saliency generation approach for visual interpretability, and CLIP is a strong textual-visual feature encoding method which is trained contrastively using a very large dataset of 400 million image-sentence pairs with rich cross-modal information between text semantics and image appearances. As shown in the work proposed by [1], with the rich cross-modal information encoded by CLIP, it can match the performance of the ImageNet pre-trained ResNet50 developed by [4] on the ImageNet dataset in the "zero-shot" manner without using any of the originally labeled examples. The proposed approach works in the way by first feeding the CLIP model with an input image and the query sentence for the corresponding image and text embeddings. Then, we can easily get a visual saliency map by applying Grad-CAM to the image branch of the CLIP model upon the output value of the cosine similarity between the image and text embeddings followed by a proper thresholding operation to generate the final bounding boxes of object and action localization tasks for images. With the help of CLIP, the proposed approach can also work effectively even in the unseen scenarios without introducing additional training costs. With extensive experiments on the Open Images and HICO-DET datasets, the results demonstrate the effectiveness of the proposed approach for the text-guided unseen object and action localization tasks for images. The code will be released upon acceptance.

## 2. RELATED WORKS

In this section, we briefly review the recent relevant works of zero-shot, one-shot, and few-shot detection as follows.

**Zero-shot Detection.** Early studies on zero-shot learning for object detection and visual recognition are achieved by encoding categories as vectors and learning the embeddings [5, 6]. Other approaches investigate label similarity and hierarchy for knowledge transfer [7, 8, 9], or simply set the weight of a specific layer from environment [10]. Recent studies focus more on semantic class representation. In terms of zero-shot object detection, Bansal *et al.* [11] improved this issue by adapting visual semantic embeddings and motivated background-aware approaches. Rahman *et al.* [12] reduced domain shift and model bias by a transductive approach. Li *et al.* [13] also considered textual description. Rahman *et al.* [14] presented polarity loss that handles class-imbalance and properly aligns the visual and semantic cues. Gu *et al.* [15] also incorporated the CLIP model to object detection via knowledge distillation by minimizing the distance of image and text embeddings and aligned region embeddings from CLIP. This is the closest work to ours. In contrast, the proposed method does not require any further training and can work directly for the unseen object and action localization tasks.

**One-shot and Few-shot Detection.** Early attempts on few-shot

⋆ The first four authors equally contribute to the work.

learning were mostly incorporating meta-learning with a two-stage Faster-RCNN object detector [16] or guiding models to gradually learns the rare category oriented network representation [17]. Recently, Wang *et al.* [18] proposed FsDet, a novel but simple approach for few-shot detection by fine-tuning the last layer of the detectors on rare classes. Hsieh *et al.* [19] explored the co-attention and co-excitation scheme for one-shot object detection. In addition, meta-learning for capturing the relations between classes also got increased attention recently for few-shot object detection, such as DCNet [20]. It exploited support features to cover all spatial locations in a feed-forward fashion using the meta-learning framework. Zhang *et al.* [21] proposed Meta-DETR to incorporate correlation aggregation for meta-learning into the DETR framework [22] to capture inter-class correlations among different classes.

## 3. THE PROPOSED APPROACH

The overview of our framework is shown in Fig. 1. We first introduce the CLIP model and Grad-CAM, which are the main components of our model in Section 3.1, and then build our CLIPCAM method in Section 3.2. To obtain the final result, we explain the post-processing in Section 3.3.

### 3.1. Preliminaries

#### *3.1.1. CLIP.*

CLIP [1] is a joint vision and language model trained using over 400 million images $\mathcal{I}$ and their corresponding captions $\mathcal{T}$. It can be used for zero-shot image classification and potentially for other vision tasks. CLIP is comprised with two networks, Image Encoder $E_I$ and Text Encoder $E_T$. For each image $I \in \mathcal{I}$ and caption $T \in \mathcal{T}$ pair, the training process of CLIP utilizes contrastive learning, which maximizes the cosine similarity score $y$ between the embeddings of image $e_I = E_I(I)$ and caption $e_T = E_T(T)$ pair and minimize the score between different images and captions

$$y = \gamma \frac{e_T^T e_I}{\|e_T\| \|e_I\|}, \tag{1}$$

where $\gamma$ is the scaling factor which is learnt during training. By this training scheme, in inference stage, given sentence $T$ based on ground truth classes such as "This is a photo of a dog.", the model can find the closest image $I$ fitting the description by computing the similarity scores of all images with the highest similarity score.

#### *3.1.2. Grad-CAM.*

Class activation mapping (CAM) proposed by [23] provides a simple approach to retrieve a visual saliency map. This method uses the weight in the fully connected layer corresponding to each average value of the feature map to calculate the weighting sum of all feature maps. However, this method has a strict restriction that requires the last pooling layer to be a global average pooling (GAP) layer. Grad-CAM [2] solves this problem by computing neural importance $\alpha_k$ using the partial derivatives of each feature map $A^k$ in the target layer $k$ where the partial derivatives were computed during backpropagation.

$$\alpha_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y}{\partial A_{ij}^k}, \tag{2}$$

where $y$ is the network output, and $Z$ is the total number of pixels in $A^k$.

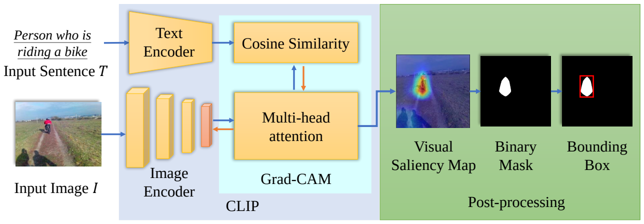

**Fig. 1**: Illustration of the proposed CLIPCAM model. We first pass the input image $I$ through CLIP image encoder and multi-head attention to get our image embedding $E_I$. The guiding text input is also passed through the text encoder to get the text embedding. Then, we can compute neural importance $\alpha$ based on the derivatives of cosine similarity score using backpropagation. The red arrows indicate the backpropagation we need to generate the saliency map $S$. After getting the saliency map, we use the binary mask threshold $\beta$ and mask out threshold $m$ to obtain the binary mask $B$ and the corresponding bounding box.

### 3.2. CLIPCAM

Given an image and a description sentence, our target is to find the area in the image that fits the text most. Since CLIP [1] maximizes the similarity between paired sentences and image embeddings during training, we made an assumption that CLIP has the capability of capturing various regions in the image based on different text descriptions. Therefore, our CLIPCAM applies Grad-CAM on CLIP and outputs a coarse heatmap that localizes the image regions.

For an image $I \in \mathcal{I}$, we first resize it to $224 \times 224$ to fit the input size of the CLIP model, and get the embedding $e_{\hat{I}}$ from multi-head attention of *A*. For the paired text, we use the same preprocessing procedures of the CLIP model and pass it to its text encoder to generate the text embedding $e_T$. We then calculate the cosine similarity $y$ between $e_{\hat{I}}$ and $e_T$ by equation 1, and utilize Grad-CAM to obtain the visual saliency map by computing neuron importance $\alpha_k$ corresponding to *k*th feature map. According to chain rule, we may expand $\alpha_k$ as

$$\alpha_k = \frac{\gamma}{Z\|e_T\|\|e_{\hat{I}}\|} \sum_i \sum_j e_T \frac{\partial e_{\hat{I}}}{\partial A_{ij}^k}, \tag{3}$$

where the backpropagation of text embedding indicates how different text embeddings contribute to distinct neuron importance, and the corresponding regions of the target objects can be attained by performing Grad-CAM on CLIP followed by additional thresholding operations.

### 3.3. Post Processing

To obtain a binary mask with the same size as the input image for the detection task, we can post-process the visual saliency map generated in Section 3.2. Since the resolution of the output saliency map is 224 × 224, we perform bilinear interpolation on it to obtain a resized map *S* which is the same size as the original image $I$. To obtain the binary mask *B*, we perform thresholding operation with a threshold $\beta$ upon the heatmap *S* as the following equation

$$B_{ij} = \begin{cases} 1 & \text{for} \quad S_{ij} > \beta \\ 0 & \text{for} \quad S_{ij} \le \beta, \end{cases} \tag{4}$$

where $\beta$ is set differently based on our target task. Then, we compute the bounding box for the target that covers the binary mask with minimum area.

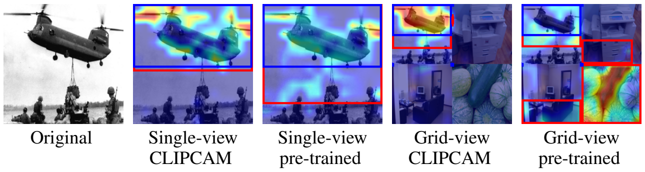

**Fig. 2**: Qualitative results for CLIPCAM and pre-trained model (ViT-B/16). Guiding text: “helicopter”. We see that for single-image evaluation, pre-trained model can localize the object well even with various background object. In grid-view, only CLIPCAM can consistently capture the image.

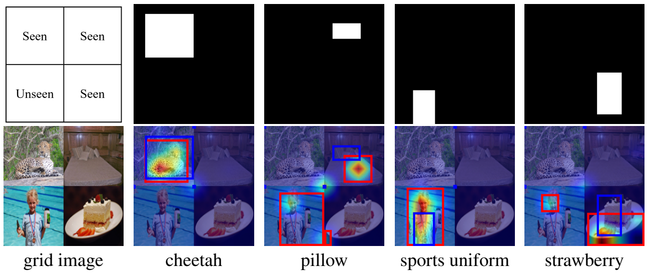

**Fig. 3**: An example for grid-view localization. The ground-truth bounding boxes are shown in the top row. The result of CLIPCAM (ViT-B/16) is in the bottom row. The red boxes are the predicted bounding boxes, and the blue boxes are the ground-truths.

## 4. EXPERIMENTAL RESULTS

In this section, we show extensive experimental results of the proposed approach to demonstrate its capability of the unseen object and action localization tasks for images.

### 4.1. Datasets and Evaluation Settings

In this work, we mainly used two datasets for evaluation: the Open Images dataset [24] for unseen object localization and the HICO-DET dataset [25] for unseen action localization. The details of both datasets are described as follows.

**Open Images.** The images in the Open Images dataset contains very diverse and many complex scenes with several objects in a single image (8.4 objects per image on average). The validation set of the Open Images V6 dataset is used in our experiments, which contains 41,620 images and 570 valid classes.

**HICO-DET.** The HICO-DET dataset contains 600 distinct types of human-object interactions. Each image contains the bounding boxes of the person and the object with which the person is interacting. There are 38,118 images in the training dataset and 9,658 images in the testing dataset.

Since CLIPCAM was trained with large amount of image-text pairs without any explicit categorical labels, all classes can be considered unseen for CLIPCAM since it has no direct supervision on these pre-defined classes. To compare with baseline models, we split the evaluation sets for each dataset into two groups: seen and unseen classes. For the zero-shot object localization task, we denote the evaluation samples of the Open Images dataset whose class is also in the ImageNet dataset as seen and otherwise as unseen, and select ImageNet-pretrained model as the baseline methods. For zero-shot action localization, we take half of the action classes in the HICO-DET dataset to fine-tune the ImageNet pre-trained models as the baseline methods. Those classes are denoted as seen and the rest as unseen.

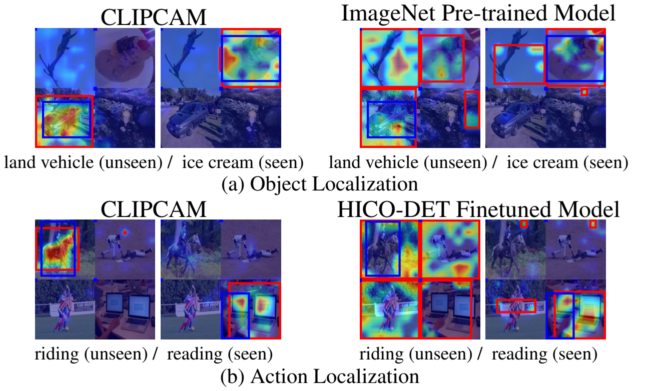
CLIPCAM
ImageNet Pre-trained Model
land vehicle (unseen) / ice cream (seen)
land vehicle (unseen) / ice cream (seen)
(a) Object Localization
CLIPCAM
HICO-DET Finetuned Model
riding (unseen) / reading (seen)
riding (unseen) / reading (seen)
(b) Action Localization

**Fig. 4**: Qualitative results of CLIPCAM for object and action localization using CLIPCAM with ViT-B/16. The red boxes are the predicted bounding boxes, and the blue boxes are the ground-truths.

We then construct the 2x2 image grid by concatenating 4 different images into a single image. Only 1 of the 4 images contain the object of interest, the other 3 images serve as distractions to make the task more challenging. This prevents the models from cheating wherein they unintentionally highlight the object by chance without actually recognizing it just because they are in the middle or occupies a large portion of the image, as demonstrated by Adebayo et al [26]. A demonstration of our grid-view evaluation and traditional single image evaluation is shown in Fig. 2. We then evaluate the performance by computing the mean intersection over union ($mIoU$) between the ground truth labels and the predictions of the models. This gives us a measure of how much overlap there is between the object of interest and our visual saliency map.

For zero-shot object localization tasks, we generate 500 grid-view images with OpenImage validation set as shown in Fig. 3, where we select 100 of the most dominant classes from seen and unseen categories, and then randomly select 10 images from each class as the test set. For zero-shot action localization tasks, we follow the zero-shot object localization settings for HICO-DET dataset. We discard those images comprising more than one pair of human object interaction, and those images with action “no interaction” to prevent ambiguity. Since HICO-DET is not designed for image-based action recognition and only contains the bounding boxes of persons and objects, this usually results in much lower mIoU values. Therefore, for effective comparisons, we also calculate the accuracies of the correct region localized by the proposed and other baseline methods to see if the model is focusing on the correct region in the grid.

### 4.2. Implementation Details

For the CLIP model, there are several pre-trained image encoders available, including the ResNet-based (RN50 for ResNet-50 and RN101 for ResNet-101) [4] and Vision-Transformer-based (ViT-B/16 and ViT-B/32) models [27], and we used all of them in our experiments where for ResNets, we selected the final layer for the heatmap generation with Grad-CAM; for ViTs, we selected layer-normalization in the final block. For the text encoder of CLIP, it is a transformer architecture model pre-trained using masked self-attention in generative pre-training framework (GPT) [28]. It performs lower-cased byte pair encoding and projects encoded texts into image-text embedding space. For Grad-CAM and its variants, we used the implementations from the *pytorch-grad-cam* package [29]. For the query sentences used by the proposed method, we adopt the sentence templates for object detection as “a photo of

| Model | backbone | grid-view mIoU | | | single-image mIoU | | |
|---|---|---|---|---|---|---|---|
| | | all | seen | unseen | all | seen | unseen |
| Pre-trained | RN50 | 0.191 | 0.276 | 0.106 | 0.447 | 0.502 | 0.444 |
| Pre-trained | RN101 | 0.198 | 0.290 | 0.105 | 0.445 | 0.502 | 0.443 |
| Pre-trained | ViT-B/16 | 0.174 | 0.237 | 0.110 | 0.450 | 0.506 | 0.448 |
| Pre-trained | ViT-B/32 | 0.218 | 0.328 | 0.109 | 0.438 | 0.501 | 0.435 |
| CLIPCAM | RN50 | 0.310 | 0.362 | 0.259 | 0.438 | 0.451 | 0.437 |
| CLIPCAM | RN101 | 0.263 | 0.320 | 0.207 | 0.438 | 0.451 | 0.437 |
| CLIPCAM | ViT-B/16 | **0.370** | **0.434** | **0.306** | **0.461** | **0.518** | **0.457** |
| CLIPCAM | ViT-B/32 | 0.363 | 0.427 | 0.298 | 0.446 | 0.498 | 0.443 |

**Table 1**: Localization performance of objects in the validation set of the Open Images dataset using Grad-CAM. CLIPCAM outperforms the baseline models for both unseen and seen objects in the grid-view setting, and the baseline models are incapable of recognizing unseen objects.

| Model | CAM Type | grid-view mIoU | | |
|---|---|---|---|---|
| | | all | seen | unseen |
| Pre-trained | Grad-CAM | 0.177 | 0.239 | 0.116 |
| Pre-trained | LayerCAM | 0.273 | **0.446** | 0.101 |
| Pre-trained | EigenGradCAM | 0.225 | 0.355 | 0.010 |
| Pre-trained | EigenCAM | 0.149 | 0.173 | 0.125 |
| Pre-trained | Grad-CAM++ | 0.177 | 0.241 | 0.113 |
| Pre-trained | XGradCAM | 0.123 | 0.130 | 0.118 |
| Pre-trained | ScoreCAM | 0.121 | 0.126 | 0.115 |
| CLIPCAM | Grad-CAM | **0.370** | 0.434 | **0.306** |
| CLIPCAM | LayerCAM | 0.247 | 0.316 | 0.178 |
| CLIPCAM | EigenGradCAM | 0.235 | 0.283 | 0.188 |
| CLIPCAM | EigenCAM | 0.153 | 0.173 | 0.133 |
| CLIPCAM | Grad-CAM++ | 0.142 | 0.147 | 0.137 |
| CLIPCAM | XGradCAM | 0.129 | 0.135 | 0.124 |
| CLIPCAM | ScoreCAM | 0.121 | 0.127 | 0.115 |

**Table 2**: CLIPCAM with different CAM variations. The performance on grid-view object localization is shown. We see that not all the CAMs are suitable for object localization.

[label]" and "[label]". For actions, we use a single verb "[action]" as the sentence template for action localization. Then, we tokenize and encode the sentences into text features. CLIPCAM takes in text features and encoded image features, then outputs the visual saliency maps of the image-sentence pairs. According to the four backbones used in the experiments (RN50, RN101, ViT-B/16, ViT-B/32), the $\beta$'s we used for CLIPCAM are $0.2, 0.2, 0.3, 0.3$ and for ImageNet pre-trained model are: $0.3, 0.3, 0.3, 0.25$ respectively. All the experiments were conducted using a single NVIDIA RTX 2080 Ti with Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz.

### 4.3. Evaluation Results

**Zero-shot Object Localization.** As shown in Table 1, the localization performance ($mIoU$) of the proposed CLIPCAM method on both seen and unseen objects outperforms ImageNet pre-trained models. We can find that the baseline models have difficulties localizing the unseen objects (with mIoU about 0.1), while our proposed CLIPCAM model can achieve much better performance with mIoU close to 0.3. CLIPCAM with the VIT-B/16 architecture achieves the best $mIoU$ scores in both grid-view and single-image settings. Some qualitative results are shown in Fig. 4(a).

**Zero-shot Action Localization.** In Table 3, we see that our text-guided CLIPCAM performs consistently well across all classes. The baseline models which are fine-tuned on action classes perform well on seen actions but get low accuracies on unseen actions. Some qualitative results are shown in Fig. 4(b), We see that our CLIPCAM can identify the person that is performing the action in the image, as well as objects that are related to the action. CLIPCAM can identify the correct region described in the given sentence.

| Model | backbone | Accuracy | | | mIoU | | |
|---|---|---|---|---|---|---|---|
| | | all | seen | unseen | all | seen | unseen |
| Pre-trained | RN50 | 0.351 | 0.395 | 0.249 | 0.152 | 0.168 | 0.114 |
| Pre-trained | RN101 | 0.320 | 0.335 | 0.284 | 0.151 | 0.165 | 0.118 |
| Pre-trained | ViT-B/16 | 0.294 | 0.326 | 0.221 | 0.070 | 0.080 | 0.049 |
| Pre-trained | ViT-B/32 | 0.318 | 0.351 | 0.240 | 0.094 | 0.104 | 0.072 |
| CLIPCAM | RN50 | 0.526 | **0.530** | 0.517 | 0.183 | 0.193 | 0.158 |
| CLIPCAM | RN101 | 0.392 | 0.450 | 0.259 | 0.137 | 0.145 | 0.121 |
| CLIPCAM | ViT-B/16 | **0.546** | 0.528 | **0.589** | 0.184 | **0.195** | 0.159 |
| CLIPCAM | ViT-B/32 | 0.513 | 0.492 | 0.562 | **0.269** | 0.186 | **0.161** |

**Table 3**: Localization performance of actions in HICO-DET dataset with the grid-view setting. Accuracy of correct predicted region and mIoU are shown. We see that CLIPCAM performs consistently well across all these classes.

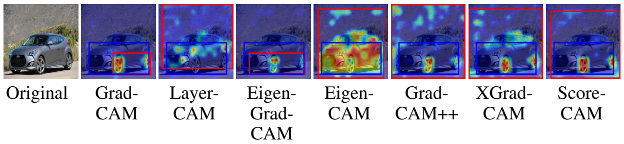

**Fig. 5**: Qualitative results of CLIPCAM (ViT-B/16) with different CAMs. Guiding text: "wheel".

**CAM Variations.** We also evaluate the proposed approach with different CAM methods implemented in the *pytorch-grad-cam* [3] package, including XGradCAM [30], ScoreCAM [31], EigenCAM [32], EigenGradCAM and LayerCAM [33]. From Table 2, we find that using Grad-CAM in our CLIPCAM framework results in the best localization performance. We conclude two explanations. First, some CAMs tends to generate more false positives at the backgrounds. This reduces their $mIoU$ despite them being able to also capture the object of interest. The other reason lies in the implementation of these CAMs. LayerCAM generates saliency map from different layers in the model, including the shallow layers which produce textural features in the final output, thus triggering the aforementioned problem. EigenCAM, EigenGradCAM and ScoreCAM depend heavily on the classification layer to be a supervised classification framework. However, the CLIP model is trained in a contrastive learning framework on image-text pairs, which may violate the initial premise of these CAMs. The qualitative results of different CAMs are shown in Fig 5.

## 5. CONCLUSION

In this work, we propose a simple and effective baseline called CLIPCAM for the zero-shot text-guided object and action localization task. With the assistance of the CLIP model, which encodes rich textual and visual information, with an input image and a query sentence, we can easily use the proposed method along with any gradient-weighted class activation mapping methods to parse and highlight the image according to the sentence. Thus, the proposed text-guided localization method can generate a heatmap to detect the relevant objects and actions described in the text. With extensive experiments on the OpenImage and HICO datasets for seen/unseen object and action localization, the results show that the proposed approach achieves satisfactory performances compared with other baseline models without introducing additional training costs.

## 6. ACKNOWLEDGEMENT

This work is supported by Ministry of Science and Technology, Taiwan (R.O.C), under Grants no. 110-2221-E-001-009-MY2, 110-2221-E-001-002-, 110-2634-F-002-051-.

## 7. REFERENCES

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[2] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017, pp. 618–626.

[3] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *WACV*, 2018, pp. 839–847.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[5] Dinesh Jayaraman and Kristen Grauman, "Zero-shot recognition with unreliable attributes," in *NeurIPS*, 2014.

[6] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid, "Label-Embedding for Image Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1425–1438, 2016.

[7] yunlong yu, Zhong Ji, Yanwei Fu, Jichang Guo, Yanwei Pang, and Zhongfei (Mark) Zhang, "Stacked semantics-guided attention model for fine-grained zero-shot learning," in *NeurIPS*, 2018.

[8] Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal, "Link the head to the "beak": Zero shot learning from noisy text description at part precision," in *CVPR*, 2017.

[9] Zeynep Akata, Mateusz Malinowski, Mario Fritz, and Bernt Schiele, "Multi-cue zero-shot learning with strong supervision," in *CVPR*, 2016.

[10] Bernardino Romera-Paredes and Philip Torr, "An embarrassingly simple approach to zero-shot learning," in *ICML*, 2015, pp. 2152–2161.

[11] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran, "Zero-shot object detection," in *ECCV*, 2018.

[12] Shafin Rahman, Salman Khan, and Nick Barnes, "Transductive learning for zero-shot object detection," in *ICCV*, 2019.

[13] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang, "Zero-shot object detection with textual descriptions," 2019.

[14] Shafin Rahman, Salman Khan, and Nick Barnes, "Improved visual-semantic alignment for zero-shot object detection," in *AAAI*, 2020, pp. 11932–11939.

[15] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui, "Zero-shot detection via vision and language knowledge distillation," *arXiv preprint arXiv:2104.13921*, 2021.

[16] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao, "Lstd: A low-shot transfer detector for object detection," in *AAAI*, 2018.

[17] Dawei Zhou, Jingrui He, Hongxia Yang, and Wei Fan, "Sparc: Self-paced network representation for few-shot rare category characterization," in *ACM SIGKDD International Conference*, 2018, pp. 2807–2816.

[18] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu, "Frustratingly simple few-shot object detection," in *ICML*, 2020.

[19] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu, "One-shot object detection with co-attention and co-excitation," in *NeurIPS*, 2019.

[20] Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang, "Dense relation distillation with context-aware aggregation for few-shot object detection," in *CVPR*, 2021, pp. 10185–10194.

[21] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, and Shijian Lu, "Meta-detr: Few-shot object detection via unified image-level meta-learning," *arXiv preprint arXiv:2103.11731*, 2021.

[22] YNicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020, pp. 213–229.

[23] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016, pp. 2921–2929.

[24] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy, "Openimages: A public dataset for large-scale multi-label and multi-class image classification.," *Dataset available from https://github.com/openimages*, 2017.

[25] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng, "Learning to detect human-object interactions," in *WACV*, 2018, pp. 381–389.

[26] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim, "Sanity checks for saliency maps," in *NeurIPS*, 2018, p. 9525–9536.

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[28] Alec Radford and Karthik Narasimhan, "Improving language understanding by generative pre-training," 2018.

[29] Jacob Gildenblat and contributors, "Pytorch library for cam methods," `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

[30] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li, "Axiom-based grad-cam: Towards accurate visualization and explanation of cnns," in *BMVC*, 2020.

[31] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *CVPRW*, 2020, pp. 24–25.

[32] Mohammed Bany Muhammad and Mohammed Yeasin, "Eigen-cam: Class activation map using principal components," in *IJCNN*, 2020.

[33] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei, "Layercam: Exploring hierarchical class activation maps for localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 5875–5888, 2021.