Proceedings of 10th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2021 Month: October Volume: 10

One-shot Object Detection and Segmentation Mask using Attention Localization

Bhimesh Chandra Acharya [a], Hari Prasad Baral [b], Bikram Acharya [c], Asmita Kattel [d]

[a, b] *Department of Electronics and Computer Engineering, Paschimanchal Campus, IOE, Tribhuvan University, Nepal*
Corresponding Email: [a] acharyabhimys@gmail.com, [c] aacharya.bikram@gmail.com, [d] aakanshyaaashmi01@gmail.com

Abstract
Object detection is a part of research in computer vision that detects and recognizes object instances in images/video. During the development of object detection over an extended period of time, there have been numerous approaches and obtained some shots of promising results. Training a machine learning model requires a huge volume of data and computation power to classify an object, but if there is little data available, learning good features can be computationally expensive. Here we present a simple method for one-shot object localization and instance segmentation, which generates a segmentation mask for all objects within a complex scene (Reference/Target image), with respect to objects similar to the query image. To address this challenging task, the proposal is to use an attention-based transformer encoder and decoder with a Siamese network to predict similarity between the all-segmented mask object of target images to the underlying query image, irrespective of class, seen or unseen during training. Experimental evaluation shows that our proposed model achieves 68.8 AP50 for box prediction and also obtains 93% accuracy in similarity.

Keywords
Transformer encoder-decoder, multiheaded attention layer, siamese network, One-shot object detection

1. Introduction

Computer vision is a branch of artificial intelligence which has a wide research area in practical or real-world problem solving and analysis and also can be trained the computer to mimic human eyes/brain. With an increase in processing speed and different algorithms/methods, computers are now able to perform different applicable tasks, such as facial recognition, object detection, image manipulation, Natural Language Processing (NLP), pattern recognition, etc. In every application of computer vision, it requires a huge amount of wide categories of labeled dataset to analyze, and understand the environment in which it is to process in order to make intelligent decisions, which may be very labor-consuming to prepare. In this study, the focus to tackle this problem will be by using one-shot object detection and segmentation mask using attention localization as our object of interest.

Apart from using a large dataset, Classical object detection uses a lot of complexity in windowing and anchor boxes to generate RPN (region proposal networks), and then a post-processing step of NMS (non-max suppression) to clean up when multiple boxes are predicted for the same object, whereas one-shot object detection uses encoder and decoder with a multi-headed attention layer, and the Siamese neural network to perform direct set prediction of object detection and segmentation.

One shot object detection is a computer vision phenomenon that makes computers efficiently detect objects in cluttered, unseen images. Humans have strong performance in the task of recognizing (i.e. children can quickly learn to detect an object from very few examples, even if there are adverse conditions/variations in object appearances, color or viewing angle, lighting conditions. Such an ability to learn to detect an object from a few examples is also desired on a computer by using computer vision systems. One shot object detection is correlated with the human visual system to perform the task of localization and categorization of similar objects in an image. The goal of one shot object detection is to locate the query image's closest comparable appearances object in a reference/target picture using only a single query image and reference/target image.

2. Related Work

The ability to identify and localize objects based on oneshot learning is a very challenging task. To accomplish oneshot object identification tasks, several machine learning techniques have been utilized, such as Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen and Tyng-Luh Liu on their paper "One-Shot Object Detection with Co-Attention and Co-Excitation", used co attention and co excitation on siamese network to obtain relatively best output of 40.9 AP50 [1], Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge and Alexander S. Ecker "One-Shot Instance Segmentation", used Mask R-CNN with Siamese backbone encoding achieve 41.3 and 38.4 mAP50 for object detection and instance segmentation.[2], OS2D: One-Stage One-Shot Object Detection by Matching Anchor Features by Anton Osokin, Denis Sumin, and Vasily Lomakin proposed to use dense correlation matching and feed-forward geometric transformation model [3], "Fully Convolutional One–Shot Object Segmentation for Industrial Robotics" by Benjamin Schnieders, Shan Luo, Gregory Palmer and Karl Tuyls used DCNN network,This configuration leads to an overall accuracy of 73.1%, r4,"Tell Me What They're Holding: Weakly-Supervised Object Detection with Transferable Knowledge from Human-object Interaction" paper written by Daesik Kim, Gyujeong Lee, Jisoo Jeong, used RRPN at the top of Faster-RCNN model, model achives 17.19 mAP [4],"FCOS Fully convolutional one-stage object detection," for one shot detection task where they obtain feature map and calculate per pixel multi level prediction with fpn and use centreness and regression for object detection. and outperforms anchor based retinanet by more than 2% in AP obtaining 64.1 AP50[5] where most of the paper have implemented CNN architecture as main to obtain Region proposal network to eradicate hassle of convolution method different steps of anchor box generation to select RPN (region proposal networks) and againpost-processing step of NMS (non-max suppression) lead to development of direct set prediction approach using transformer decoder- encoder architecture paper [6, 7, 8].

"Finding your lookalike: Measuring Face Similarity Rather than Face Identity" paper written by amir sadovnik, Wassim Garbi, thanh vu and Andrew Gallagher uses triplet network with VGG face CNN achieves 78.1% accuracy to predict face similarity than original VGG embedding which achieves 66.43% [9], "Similarity Mapping with Enhanced Siamese Network for MutiObject Tracking " by minyung kim, Stefano alleto used a approach of enhanced Siamese network where basenetwork is siamese network and then it is extended with tracking system based on ESNN with geometric information when trained on market 1501 dataset achieves f1 score of 0.9814[10], "Fully convolutional instance-aware semantic segmentation, "written by Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei proposed to use FCN (fully connected layer) then second step to produce fixed sized ROI (region of interest maps) followed by one or more FCN to convert ROI feature maps to ROI maps. This method has won coco 2016 competition with 39.7% object detection accuracy [11].

3. Methodology

3.1 Image Dataset

Here for this method, the dataset used is the 21 classes of the MS COCO dataset for the training and validation process of Detr Architecture and a small set of images are assembled from different sources from the web to create 21 classes of image (i.e. person, bicycle, car, airplane, bus, fire hydrant, cat, dog, horse, cow, elephant, bear, cup, apple, orange, hot dog, pizza, chair, couch, keyboard, clock) for the training and validation process of the Siamese network.

In this study, the methodology is composed of two parts, i.e., where the target image is passed through the DETR Architecture and the output of DETR architecture with the query image is passed to the siamese network.

3.2 DETR Architecture

There are many algorithms developed to detect objects in an image, such as Fast R-CNN [12], Faster R-CNN [13], Mask R-CNN [14], Yolo, and others, which utilize lots of complexity in windowing and anchor box to generate RPN (region proposal networks) and again post-processing step of NMS (non-max suppression) to clean up when multiple boxes are predicted for the same object, but DETR [6] uses direct set prediction approach.

The Detr architecture is composed of three parts, including the feature extractor (backbone), the transformer encoder and decoder module, and the segmentation module. At the first target image $I \in R^{3\times H_0\times W_0}$ where H_0 represents height of image

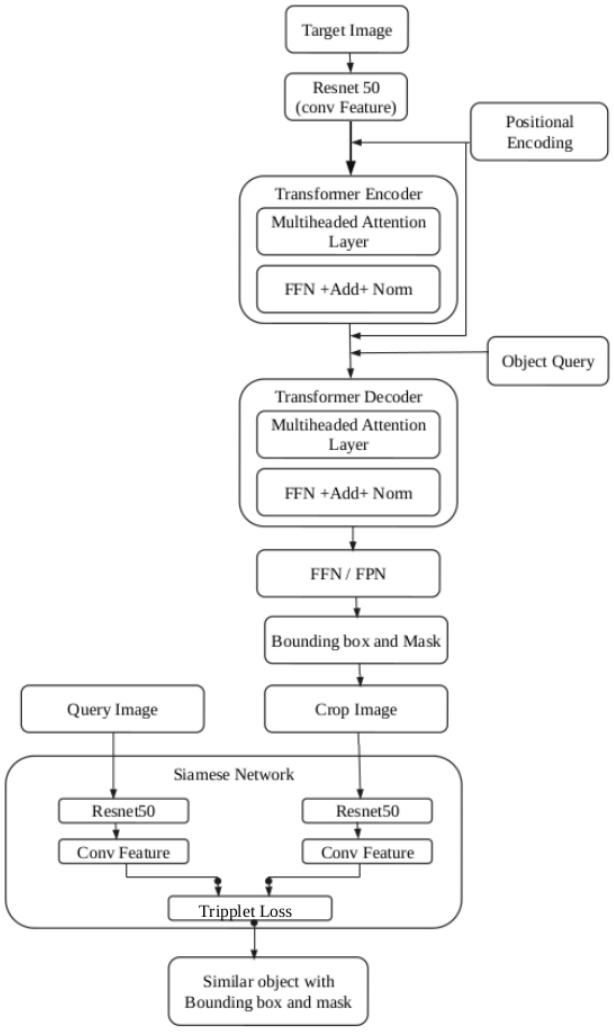

Figure 1: Research methodology

and W_0 represents width of image is passed to Resnet50 [15] to extract feature i.e., lower- resolution activation map $f \in R^{C \times H \times W}$ where f represents the backbone feature, C= 2048 and H and W represents height and width of image after convolution, learning rate of resnet is $1e^{-6}$ After feature extraction, again, 1x1 convolution is used to compress the channel dimension from 2048 to d_m=256. The extracted feature is flattened and positional encoding is done and fed to the transformer encoder. The Transformer encoder consists of a multi-headed self attention module and a fully connected feed forward network (FFN). We add residual connections around each of the two sub-layers and layer normalization is performed. The residual connection is added to minimize the vanishing gradient problem and to preserve important information that may be neglected during the multi-headed attention layer [7]. Number of parameters used is 41302368 and AP for small object is 18.4, AP for medium object obtained as 50 and AP for large object is obtained as 68.9. learning rate of transformer is $1e^{-5}$ The encoder is made up of six identical layers stacked on top of each other. The multiheaded attention layer contains a linear input layer with 256 channels, and these three linear layers are fed key, value, and query input, and each layer processes independently, with each output given to a scaled dot product attention layer, which calculates the attention score.

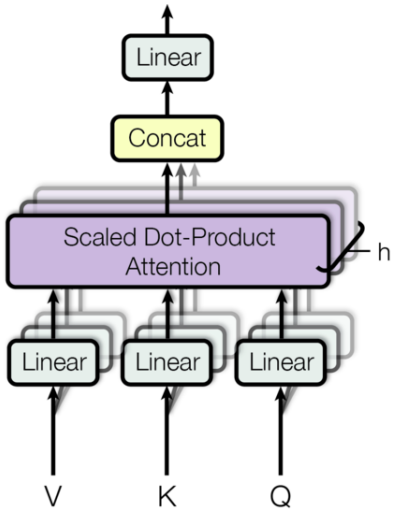

Figure 2: Multi-headed attention layer

Formula for calculating attention score

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Where q is a query matrix, k is a key matrix whose transpose is calculated and multiplied to q, and v is the value, whereas d_k represents the dimension of the key vector. Here, the softmax function is applied to the query and key which provides the attention filter and when multiplied with the value, we calculate the filtered value. The Query, Key, and Value parameters of the multi-headed Attention module are divided and processed through n_{heads} = 8 different attention heads, yielding multiple attention filtered values, i.e., encoder attention heads are 8 and decoder attention heads are 8. All the values from different multi-head attention layers are concatenated. Residual connection addition is performed on the output of the multi-headed attention layer followed by normalization, After residual connection addition and normalization, the output is provided to the feed forward network, which comprises of a linear layer followed by a dropout layer with a dropout of 0.1.

The transformer decoder has six identical layers, much like the transformer encoder. The Transformer Decoder consists of a multi-headed self-attention layer and a fully connected feed forward layer. We add residual connection around each of two sub-layer and layer normalization is performed to encoder but the difference here is the three inputs expected by the decoder as query, key and value of which key and value are provided by the output from the encoder but the query is passed from instance query, Where instance queries are provided to the decoder as embeddings, i.e., all input embeddings must be distinct in order to yield different results, and the number of instances detected is equal to the number of queries passed to the decoder, N= 100 also λ_{iou} = 5 and λ_{L1}=2,

Positional encoding is a technique for converting a finite-dimensional representation of the placement of items in a sequence into data that a neural network can understand and use. The positional encoding must be a tensor that we can feed to a model to inform it where a particular value is in the sequence. For positional encoding, a sinusoidal positional encoding matrix is utilized. Vectors are used to represent sequence places in the matrix. The decoder's output is now utilized to compute the segmentation mask. However, because computing the mask is computationally costly, we first get box predictions using a three-layer feed forward network with a Relu activation function and a linear projection layer. According to the FFN,The FFN layer predicts the box's normalized center coordinates, height, and breadth in relation to the input image [6]. After box prediction, we also predict binary masks in parallel for every predicted box. It takes learned query embedding of the transformer decoder and features from the encoder as input to the multi-headed attention layer to compute a multi head attention score, generating M attention heat maps per object. After M attention maps, then it passes through FPN architecture [16]. where FPN-style CNN up-samples produced features to obtain spatial masks [14] then the predicted masks are merged using pixel wise argmax.

Let us denote $\hat{y} = \{\hat{y}_i\}_{i=1}^{n}$ the predicted instance sequences, and y_i the ground truth set of instance sequences. In the learning process, the predicted instance sequence is matched with ground truth using a Hungarian algorithm. To construct a bipartite network matching between the two sets, we look for the lowest cost permutation of n element $\sigma \in S_n$. The matching process is as follows:

$$\hat{\sigma} = \underset{\sigma \in S_n}{argmin} \sum_{i}^{n} l_{match}(y_i, \hat{y}_{\sigma(i)}) \tag{2}$$

where $l_{match}(y_i, \hat{y}_{\sigma(i)})$ is the pairwise matching cost between y_i and instance sequence prediction with index $\sigma(i)$. now the loss function for DETR is formulated as:

$$\begin{aligned} L(y, \hat{y}) = \sum_{i=0}^{N} [& -log\hat{p}_{\hat{\sigma}(i)}(c_i) \\ & + \mathbb{1}_{c \neq \phi} l_{box}(b_i, \hat{b}_{\hat{\sigma}(i)})) \\ & + \mathbb{1}_{c \neq \phi} l_{mask}(m_i, \hat{m}_{\hat{\sigma}(i)})] \end{aligned} \tag{3}$$

Where $-log\hat{p}_{\hat{\sigma}(i)}(c_i)$ represents the negative log for class prediction, $l_{box}(b_i, \hat{b}_{\hat{\sigma}(i)})$ represents box loss and $l_{mask}(m_i, \hat{m}_{\hat{\sigma}(i)}$ represents mask loss

Box loss :

The second part of the Hungarian loss is the bounding box regression loss (l_{box}).where this loss calculates the center point and height difference between the predicted box and ground truth, including GIoU and L1 loss:

$$\begin{aligned} L_{box}(b_i, \hat{b}_{\sigma(i)}) = \lambda_{iou} L_{iou}(b_i, \hat{b}_{\sigma(i)}) \\ + \lambda_{L1} \| \hat{b}_{\sigma(i)}) - b_i \|_1 \end{aligned} \tag{4}$$

Here $L_{iou}(b_i, \hat{b}_{\sigma(i)})$ is GIoU Loss which evaluate accuracy of prediction and $\lambda_{iou}, \lambda_{L1} \in \mathbb{R}$ are Hyperparameters.

Mask loss:

$l_{mask}(m_i, \hat{m}_{\hat{\sigma}(i)})$ also known as mask loss, is defined by the combination of Dice and Focal Loss. The formula for mask loss is given as:

$$\begin{aligned} L_{mask}(m_i, \hat{m}_{\hat{\sigma}(i)}) = \lambda_{mask} [& L_{Dice}(m_i, \hat{m}_{\hat{\sigma}(i)}) \\ & + L_{focal}(m_i, \hat{m}_{\hat{\sigma}(i)}))] \end{aligned} \tag{5}$$

Dice loss:

A loss function for image segmentation tasks is the dice coefficient, which is simply a measure of overlap between two sets.The Dice coefficient runs from 0 to 1, with 1 denoting perfect and full overlap. To maximize the overlap between the two sets, dice loss is considered as a 1-Dice score coefficient (DSC).

$$l_{Dice}(m_i, \hat{m}) = 1 - \frac{2m\sigma(\hat{m}) + 1}{\sigma(\hat{m}) + m + 1} \tag{6}$$

Where $\hat{m}$ represent predicted mask, m binary mask and σ represents sigmoid function

Sigmoid focal Loss:

Focal loss is extremely beneficial in the training of tasks with a significant class imbalance.During training, focal loss reshapes cross entropy loss to down-weight the easy negatives and focus learning on hard examples.

The formula for calculation of focal loss is given below:

$$F_l(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t) \quad (7)$$

Where $log(p_t)$ is binary cross entropy loss and value of $\alpha_t = 0.25$ and $\gamma = 2$.

After segmentation is taken out, pixel wise argmax is performed to align the different masks into a single image. Every mask bounding box is predicted and all the images related to the bounding box are cropped. Images from crop operations have different sizes when resized to the input size for the Siamese network. They may lose the characteristics of the image, so we do vertical and horizontal padding if required to the image, and these images are fed to the Siamese network for similar object prediction to query image.

3.3 Siamese Network

The Siamese networks are made up of identical networks (such as CNN, resnet, and autoencoders), and the same networks share weight. Signature verification, face recognition, selective search, and other problems where a large library of datasets is not accessible have all been solved using the Siamese network. The best attribute of the Siamese network is its ability to handle sparse datasets and can be optimized with a loss function during training to obtain determine similarity.

In this study, the resnet50 architecture [15] is used as the convolution backbone to obtain the convolution features of the image. Images from the datasets are taken in such a way that images represent triplet formation of positive, negative, and anchor images. The anchor and positive image represent images from the same class but different images, and the negative image represents images from all other classes except the anchor/positive image class. As an input to the siamese network, the standard image size of 224*224 is used and padding is used if the image is smaller than the standard size. Data augmentation is also

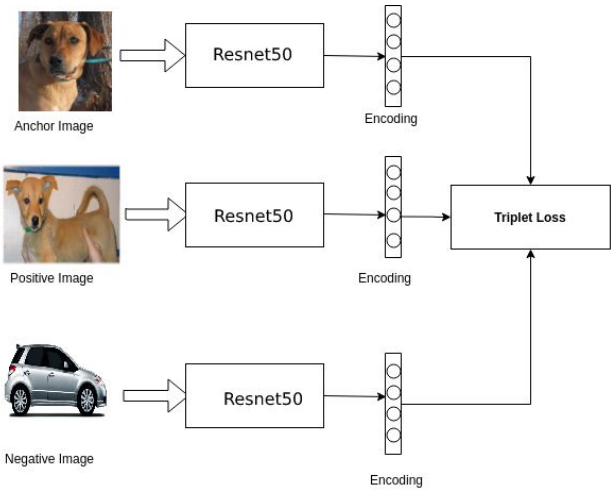

Figure 3: Siamese Network

applied on the datasets such as random-rotation, random horizontalflip, and random verticalflip. The activation function used is the Relu activation function with Adam optimizer and the loss function used is the triplet loss function. Here data splitting is based on one shot so all classes 10 images are taken for training, validation and testing, Random search technique as presented by James Bergstra and Yoshua Bengio [17] was used for finetuning hyper-parameter of optimizer. These three images are passed through the model and feature embedding of the anchor image as f (a), feature embedding of the positive image as f (p) and feature embedding of the negative image as f (n) are obtained. Then these embeddings are passed to the triplet loss function [18] with keeping margin equal to 2.

The triplet loss can be calculated using the following formula:

$$L(a,p,n) = max\{0, d(a_i,p_i) - d(a_i,n_i) + margin\} \quad (8)$$

Where L (a, p, n) represents the loss between the anchor, positive and negative image. $d(a_i,p_i)$ represents the cosine distance function between the anchor image and the positive image and $d(a_i,n_i)$ represents the cosine distance function between the anchor image and the negative image [18]. The cosine distance measure is mostly used to determine how similar two data points are.The goal of triplet loss computation is to reduce the distance between an anchor and a positive while increasing the distance between an anchor and a negative.

After training is completed, the model is fed with the query image and cropped images from the target image(i.e., croppped output of DETR architecture),

which are first resized and padding is applied if the image size is smaller than the standard image size (224,224). Query image and cropped Images from the output of detr are fed through convolution neural networks (resnet50) of the Siamese network to obtain the feature/encoding of the respective images. The feature vector of the query image F_q is used to find the similarity with each N cropped image feature vector $\hat{F}_i = \{\hat{f}_i\}_{i=1}^{n}$. Images which have a cosine similarity lower than a threshold value of 0.7 are discarded and those images whose values are higher than threshold=0.7 are selected. Those object images whose similarities are greater than the threshold, a bounding box with segmentation is plotted in the target image.

4. Result and Analysis

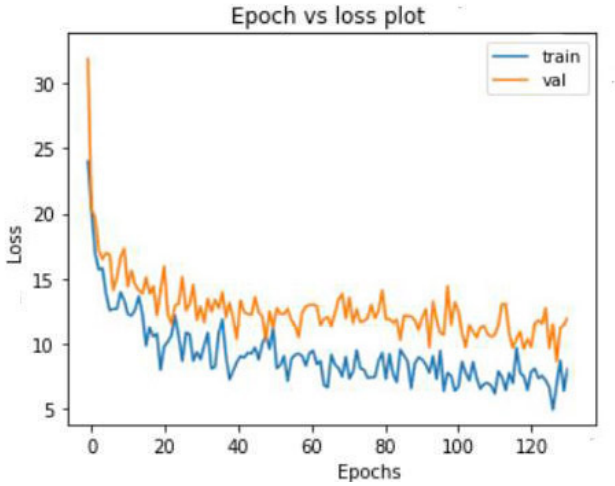

Figure 4: DETR Loss plot

The Detr network is trained with 21 classes of the image from the MS-COCO dataset. Above, fig 4 shows the loss curve constantly decreasing while training the network for 131 epochs and the best results are stored. When training for 131 epochs, Detr architecture obtained 68.8 $AP50$ for box prediction and 64.4 $AP50$ for segmentation mask.

Figure 5 shows the Siamese network training and validation loss curve while training a network for 80 epochs. From the loss curve, it can be observed that training loss continues to decrease until 80 epochs, but validation loss decreases to a point and begins to increase. The best results are stored during the training process which is at 42 epoch.Here standardizing and normalizing the data is performed, as in oneshot such plots are seen, which can be improved by increasing the number of epochs. Table 1

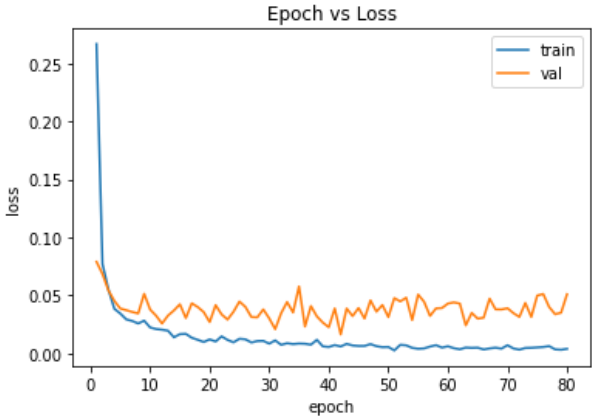

Figure 5: Siamese loss plot

shows the Siamese model's accuracy, recall, and F1 score.From the table, it can be observed that the Siamese network obtained 93 % accuracy.

Table 1: Accuracy, Precision, Recall and F1-score of Siamese network

	precision	recall	f1 score
0	0.91	0.97	0.94
1	0.97	0.90	0.93
accuracy			0.93
macro avg	0.94	0.93	0.93
weighted avg	0.94	0.93	0.93

Table 1, Here 0 means the negative samples and 1 means the positive samples. For class 0, recall of 0.97 means that among negative samples, 97% of the time we are able to correctly predict that the images are negatives.And the recall of 0.90 for class 1 means that 90% of the time the model is correctly predicting the samples as positive.

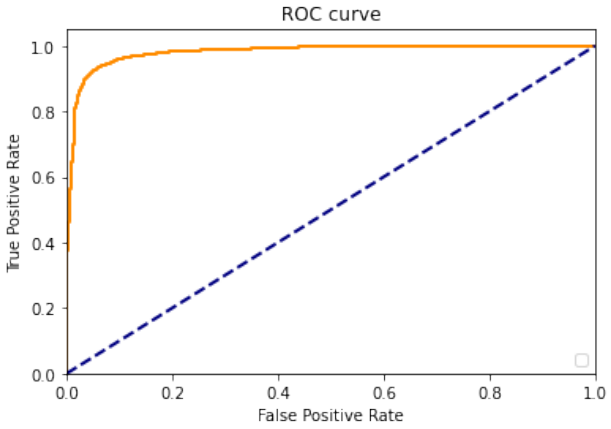

Figure 6: ROC curve

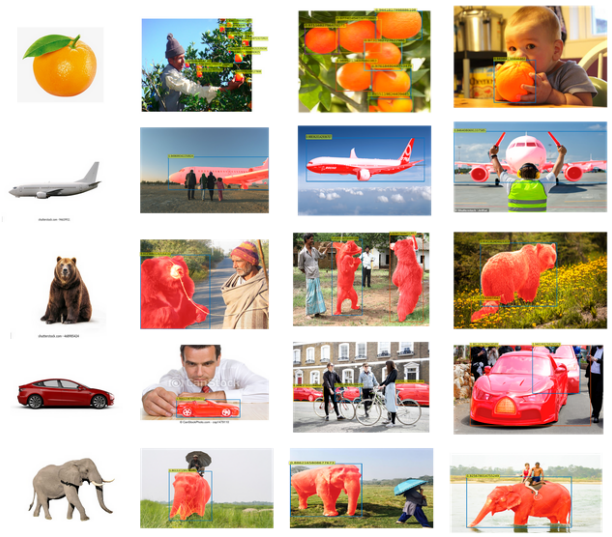

Figure 7: Qualitative results of our method where the leftmost column shows the query image and the other image on the right side shows the target image with segmentation mask.

5. Conclusion

In this paper, we present a one-shot object detection method that employs a transformer encoder and decoder architecture with a Siamese network. A Detr is proposed for generating an instance segmentation mask for each object present in the target image based on a multi-headed attention mechanism and a Siamese network for similarity metric learning between a query image and target images. This model can predict and segment objects from novel categories based on a single reference image.

This paper can be further extended by increasing the performance of transformer encoder decoder architecture by increasing the dataset and further optimize and training our method and for the Siamese network we can add some methods to extract the silent information of the image as well as an attention module inside the network to enhance or focus more on the salient features and be more discriminative.

References

[1] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. *CoRR*, abs/1911.12529, 2019.

[2] Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge, and Alexander S. Ecker. One-shot instance segmentation. *CoRR*, abs/1811.11507, 2018.

[3] Anton Osokin, Denis Sumin, and Vasily Lomakin. OS2D: one-stage one-shot object detection by matching anchor features. *CoRR*, abs/2003.06800, 2020.

[4] Daesik Kim, Gyujeong Lee, Jisoo Jeong, and Nojun Kwak. Tell me what they're holding: Weakly-supervised object detection with transferable knowledge from human-object interaction. *CoRR*, abs/1911.08141, 2019.

[5] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. *CoRR*, abs/1904.01355, 2019.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[8] 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. *CoRR*, abs/2010.11929, 2020.

[9] Amir Sadovnik, Wassim Gharbi, Thanh Vu, and Andrew C. Gallagher. Finding your lookalike: Measuring face similarity rather than face identity. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2408–24088, 2018.

[10] Minyoung Kim, Stefano Alletto, and Luca Rigazio. Similarity mapping with enhanced siamese network for multi-object tracking. *CoRR*, abs/1609.09156, 2016.

[11] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. *CoRR*, abs/1611.07709, 2016.

[12] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[13] Alexander Kirillov, Ross B. Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *CoRR*, abs/1901.02446, 2019.

[14] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. SOLQ: segmenting objects by learning queries. *CoRR*, abs/2106.02351, 2021.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[16] Alexander Kirillov, Ross B. Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *CoRR*, abs/1901.02446, 2019.

[17] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.