EECS 442 Final Project - Human Object Interaction

Ruihan Xu rhxu@umich.edu
Shiyu Liu lshiyu@umich.edu
Zichen Gai zgai@umich.edu
Chenglin Li lchengl@umich.edu

1. Introduction

HOI stands for Human-Object Interaction, a crucial concept in computer vision and artificial intelligence. It involves understanding the interactions between humans and objects in visual scenes, which is fundamental for a machine to comprehend human activities and the context in which they occur. This understanding is vital for applications such as image and video analysis, human-computer interaction, and autonomous systems. Recent research in HOI detection has focused on leveraging deep learning and specialized datasets to enhance model accuracy.[3] Some approaches aim to improve model generalization across diverse HOI categories and datasets,[6] while others explore techniques like Non-Interaction Suppression (NIS) and hierarchical learning. [1]

The problem addressed is the prediction of human-object interactions, where traditional methods might be slow due to exhaustive human-object pairings. The proposed solution focuses on using only human features for prediction, aiming to significantly speed up the process. This approach is particularly relevant in real-world scenarios like surveillance and human-computer interaction, where quick and accurate detection of interactions is essential. By streamlining the method, it promises more efficient and effective interaction prediction in various technological fields.

Our project, when integrated with additional models, has the potential to perform a diverse array of tasks within the field of computer vision. It significantly contributes to an enhanced comprehension of the ways in which machines can interpret and analyze visual information. Furthermore, its application in security domains promises to enhance performance and efficacy.

Our project proposes a pipeline for precise human-object interaction localization, involving stages like human presence identification, feature extraction, and point of interaction prediction. Various design choices were explored, revealing challenges with noisy data annotations and a workaround for images without interactions. While YOLO performed well in detecting humans, its limitations in recognizing diverse object categories prompted the need for future fine-tuning. To address challenges, future work includes exploring datasets with improved annotations, introducing a binary output for interaction presence, and fine-tuning YOLO for enhanced object recognition. Check out the GitHub repository here: `https://github.com/MultyXu/HOI_with_POI`

2. Related work

Previous research has explored various approaches to HOI detection. Yong-Lu Li and his team addresses the challenge of learning interactiveness knowledge across diverse HOI datasets.[6] The proposed interactiveness network, functioning as a transferable knowledge learner, introduces Non-Interaction Suppression (NIS) to improve HOI classification results. The hierarchical learning of interactiveness at both instance and body part levels, along with a consistency task can be an improvement upon our method. Comparative evaluations on HICO-DET, V-COCO, and PaStaNet-HOI datasets establish the superiority of their approach over existing state-of-the-art methods, emphasizing its effectiveness and versatility in HOI detection.

The conceptual foundation of our project is inspired by insights drawn from another research study: a study by Facebook AI Research team that is designed to identify human-object interactions by detecting triplets of ⟨human, verb, object⟩. They uses the appearance, action, and pose of a person to predict the location of the object they are interacting with, effectively narrowing down the search area for the target object. Implemented within the Faster R-CNN framework, this model employs action classification and density estimation focused on a person's region of interest (RoI). The density estimator predicts a 4D Gaussian distribution for each action, determining the likely position of the object relative to the person based solely on the person's appearance, enhancing the model's accuracy.[4]

3. Method

In order to tackle this complex HOI problem, we designed a pipeline that involves several different components shown in Figure 1 to facilitate the job. The pipeline mainly involves four stages.

1. Data Processing: Due to the training data not being in the desired format, we performed transformations on

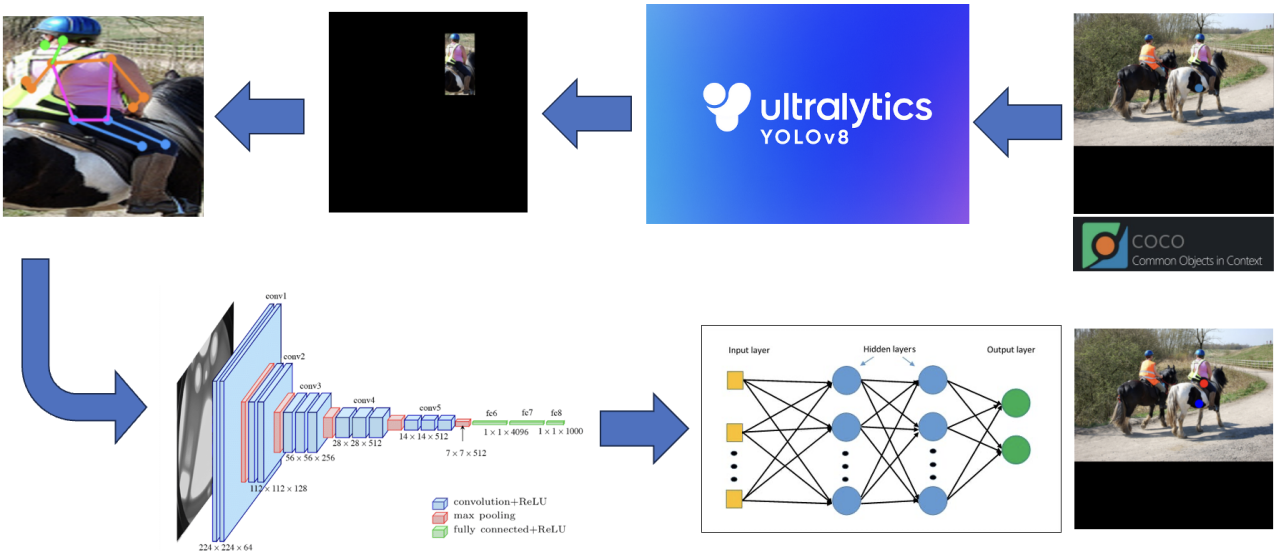

Figure 1. Our interaction detection pipeline: 1. Use YOLO to isolate human subjects. 2. Augment images with pose features. 3. Process through a VGG-like network for low-level feature extraction. 4. Apply an MLP to detect potential interaction points.

the input and label to achieve the desired structure.

2. Image Manipulation: YOLOv8 was employed to streamline the creation of suitable images for training and inference. YOLOv8 primarily focuses on detecting humans, human poses, and objects within the image.

3. Interaction Network: A neural network takes the human center image with pose as input and outputs a point of interest relative to the human. This point indicates the most likely location where an interaction with an object is occurring.

4. Object Grounding: This involves a straightforward algorithm that identifies the closest object in the image to the point of interest predicted by the neural network. The pixel distance is calculated and thresholded to 100 pixels.

3.1. Dataset

We experimented several choices and finally decide to use cropped human center image and training data and the relative position of the object to the human center as label.

3.2. Yolo

After reviewing different YOLO models and comparing their effects, we have selected YOLOv8n for detecting objects, recognizing objects' names and cropping their images. Additionally, we use YOLOv8n-pose to detect humans and crop their images.

3.3. Nerual Network

The final choice for the Neural Network architecture is a VGG-like structure [8]. Our objective is for the network to assimilate human features, incorporating additional information about human pose detected by YOLOv8. The desired outcome is that this network can predict the location of interactions in the image space based on human features such as pose, gesture, and facing direction. Intuitively, for instance, if the human image depicts a gesture of throwing something, the neural network should learn that it is likely the object being thrown is positioned several pixels in front of the person's hand. The architecture is shown in Figure 1

4. Experiments

Now, we delve into the details of this design process, taking you through each step of how we arrived at the final pipeline. We will also discuss the dataset we are using and the chosen evaluation method.

4.1. Data set

In our project, we used the V-COCO dataset, which is an extension of the Microsoft COCO object detection dataset, augmented with annotations of human-object interactions (verbs). The selection of the V-COCO dataset was influenced by its comparatively manageable size and its utilization in the reference paper for our study. Consequently, it was deemed advantageous to use this dataset for alignment with the methodologies outlined in the aforementioned paper.

First, we need to load images from the VCOCO dataset

into PyTorch tensors for subsequent training in the neural networks. Initially, our approach was to utilize the original images from the dataset. However, given that these images vary in size, we opted to standardize their dimensions to 640 × 640 by aligning the top-left corner and zero-padding all missing pixels. After experimenting this version, we observed that the accuracy did not meet our expectations. Upon reflection, we considered a potential drawback: the inclusion of background noise in the images. Consequently, we try to input the human-centric images exclusively. We extracted the portion of each image within the human bounding box, pasted it onto its original location in a black canvas with the same shape as the original image. In doing so, we aimed to direct the neural network's focus toward human-related features. While this version did yield better results compared to the previous one, the enhancement was not significant. We suspect that the limited information fed into the neural network may be attributed to the fact that the cropped human-centric images occupy only a small portion of the entire canvas. Determined that we are on the right path by emphasizing human-centric images, we resized all human bounding boxes to dimensions of 224 × 224 with no additional padding. As anticipated, this version produced the most accurate predictions.

As for the label of the dataset, we initially designated the label as the centroid of the object's bounding box since we aimed to predict the central point of the object interacting with the person depicted in the image. However, this method yielded unsatisfactory results. Subsequent research led to us using the coordinate relative to the center of the human's bounding box as the label. This modification proved to be effective, as the model began to generate outputs that were considered desirable.

4.2. Evaluation method and baseline

Typically, HOI projects use the VCOCO dataset repository benchmark for evaluating models. However, as this benchmark assesses various criteria such as IoU, object labeling, and interaction word accuracy, which may not align perfectly with our goal of detecting only the interaction point. So, we've developed our own evaluation metric. Focusing on identifying the correct object for human interaction, we measure the Euclidean distance between the predicted and ground truth interaction points in pixel space. We consider a detection correct if the distance is below a threshold of 100. This tailored metric better suits our model's intended integration into broader pipelines.

Moreover, use the techniques used by [4] as the baseline in order to compare with the recent research standing.

4.3. YOLOv8

Our training process involves utilizing YOLOv8n-pose. We feed the cropped images into this model and use it to detect and understand human poses. Once the human pose is detected, we overlay the pose detection graph onto the original image.

Our testing process uses both YOLOv8n and YOLOv8n-pose. We first crop the full-sized images to concentrate on the essential objects. This cropping is vital for isolating the key elements within each image. We then place these cropped images against a black background, ensuring that the focus remains solely on the subject, thereby facilitating a more precise assessment of the detection accuracy. Through our comparative analysis, we observed that YOLOv8n is highly effective in identifying non-human objects. With the outputs it generates, we stores the id for each object's name in a list, and make it one of our outputs of this function. However, YOLOv8n's performance in human detection in complex scenarios is less reliable since it might lead to unnecessary identifications. We tried to reset the confidence level of YOLOv8n in detecting hunans, but the improvement is limited. This observation prompted us to experiment with YOLOv8n-pose, which proved to be superior in human detection accuracy. Given its specific design for human pose analysis, YOLOv8n-pose consistently identifies humans accurately, even in complex scenarios. Consequently, we have integrated YOLOv8n for general object detection and YOLOv8n-pose for precise human detection.

4.4. Nerual Network and Hyperparameters

As mentioned in previous sections, the objective of this segment of the pipeline is to process the image and predict the point of interest where the target human is potentially interacting with an object. Initially, it appears to be a straightforward CNN detection problem, and there are several neural network architectures suitable for this task. Therefore, for the initial step, we opted for some pre-trained neural networks.

We initially considered ResNet [5], a widely-used neural network architecture still prevalent in various applications. Our choice was the pre-trained ResNet50 model in PyTorch. To adapt it for our task, which involved predicting xy coordinates in pixel space for a point of interest, we modified the output format. The original ResNet, designed for image classification, outputs scores for each class in the training set. To obtain features suitable for our needs, we applied the "create feature extraction" function in PyTorch before the fully connected layer.For regression of the xy coordinates, we designed a custom 3-layer multilayer-perceptron (MLP) [7]. During training, we kept the ResNet part frozen and updated only the weights and bias in the MLP. Our training data consisted of a 3x640x640 black canvas with the cropped human in the original pixel space. However, the performance was suboptimal, with the network often predicting (0,0) as the point of interest. Further investigation revealed that the ResNet50 in PyTorch, trained on Im-

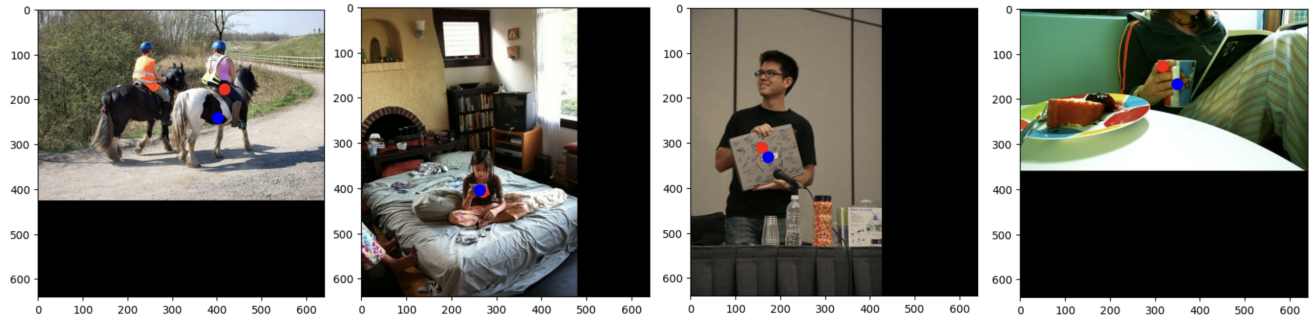

Figure 2. Some detection results from our model. The blue point is the ground truth interaction point and the red point is the predicted interaction point. The result looks good qualitatively and it can correctly capture the location of interacted object

ageNet, lacks the class "human." This limitation indicated the need for an alternative solution.

The next neural network we chose is a VGG-like architecture, similar to one we created in a previous homework assignment, and it proved effective. Additionally, this time, we used a cropped human center image with a size of 224x224 without zero padding, resized to a 640x640 size, and the label represented the relative pixel position of the object to the human. Following the architecture in Figure 1, we trained the model from scratch, fine-tuning it specifically for our application.

Ultimately, the network achieved an accuracy of about 30-40 percent. If human pose is incorporated into the image, the accuracy can be boosted to about 45 percent. But this value is still lower than the baseline [4], which is 50 percent. We will discuss this gap in the discussion section.

Furthermore, we experimented with L1 and L2 loss functions and found that L1 loss performed slightly better than L2 in this task.

Some of the detection result is shown in Figure 2

5. Conclusions

In this paper, we have devised a pipeline capable of detecting the precise location in the pixel space where a person is interacting with an object. The pipeline comprises several stages: initially identifying the human presence in the image, extracting features, including human pose, for inference, subsequently predicting the point of interaction, and ultimately pinpointing the object with which the person is interacting.

5.1. discussion and future work

Throughout the work, we have tried many different design choices of this pipeline in order to complete the work. However, there are still things that are not desired and can be improved in the future.

Upon investigating the factors contributing to low accuracy, we discovered that some data exhibit significant noise. For instance, as depicted in Figure 3, the left images suggest an interaction with the soccer ball in the upper right corner but fail to indicate an interaction with the ball at the center of the picture. On the right, the image suggests no interaction, yet a person is holding food in her hand. In the future, we can explore the more recent dataset HICO-DET [2] which has better data annotations.

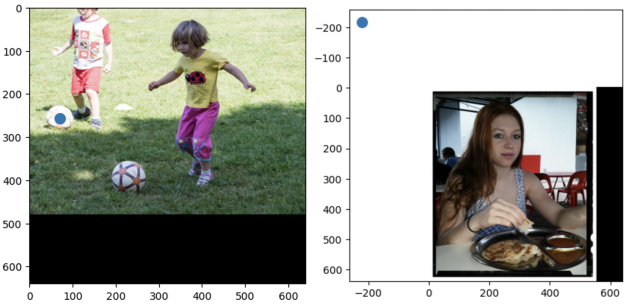

Figure 3. Example of bad annotation in the dataset.

During the data loading process, we applied a workaround to images without interactions: we set the point of interaction to (-500, -500) in pixel space. While this approach may divert the neural network from learning accurate information for predicting interactions, a potential improvement for the future could involve introducing an additional binary output from the neural network that explicitly indicates whether an interaction point is present or not.

While YOLO performs admirably in detecting humans and human poses, its lack of generality becomes apparent when attempting to recognize diverse object categories within the dataset. Some failures can be attributed to YOLO's inability to identify the object being interacted with. Therefore, for future endeavors, it would be beneficial to explore fine-tuning the YOLO model on our custom dataset to enhance its object recognition capabilities.

References

[1] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *Pro-*

ceedings of the IEEE Winter Conference on Applications of Computer Vision, 2018.

[2] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions, 2018.

[3] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[4] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions, 2018.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[6] Yong-Lu Li, Xinpeng Liu, Xiaoqian Wu, Xijie Huang, Liang Xu, and Cewu Lu. Transferable interactiveness knowledge for human-object interaction detection. *TPAMI*, 2022.

[7] Fionn Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5):183–197, 1991.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.