Using RGB, Depth, and Thermal Data for Improved Hand Detection

Rachel Luo, Gregory Luppescu
Department of Electrical Engineering
Stanford University
{rsluo, gluppes}@stanford.edu

Abstract

Hand detection is an important problem in computer vision, as it has applications spanning areas from robotics to virtual reality to life logging. We examine hand detection performance on a novel egocentric, multi-modal video dataset of everyday activities. We contribute a new dataset of video sequences with RGB, depth, and thermal data along with thousands of ground truth bounding box hand annotations. We run selective search on all of our data to generate region proposals and finish a Tensorflow implementation of Fast R-CNN. We then test the hand detection performance for RGB, depth, and thermal video frames and find that the performance is best for the RGB data, followed by the depth data, followed by the thermal data.

1. Introduction

Many recent works have explored the use of egocentric camera systems for applications in fields such as augmented reality, medical rehabilitation, and robotics, where the majority of activity analysis is driven by both the user's hand pose and the way in which the hands interact with various objects. Thus, hand detection and pose estimation are crucial problems for making the aforementioned applications realizable. Unfortunately, this problem can be difficult because hands performing daily activities can be severely occluded due to a combination of object handling and limited field of view. Very recently, researchers have utilized RGB-D data to show that depth-based cues from an egocentric camera can supply more information to help ameliorate the many challenges faced when observing hands from a first-person viewpoint. However, detection results can still be significantly improved.

Within the past year, thermal cameras have become cheaper and more accessible for use in various applications. Thermal image data has not yet been used for hand detection. Thus, the inclusion of thermal data into hand detection and pose estimation analysis shows great promise for further improvements. For this research project, the ultimate long term goal is to leverage this novel thermal data to our advantage for a deeper insight into the properties of human motion for robotics applications. For instance, one practical application could be imitation learning, where a robot uses thermal hand detection data to learn by watching humans perform various tasks. Within the scope of this class project, we hope to implement the first step of this long-term research project: an improved hand detection mechanism using thermal data.

2. Related Work

In this section, we give an overview of various works that relate to and influence our project.

2.1. Hand Detection and Pose Estimation with Depth Data

Rogez *et. al.*[8] describe a successful approach to egocentric hand-pose estimation that makes use of depth cues, and also uses strong priors over viewpoint, grasps, and interacting objects. They implement a hierarchical cascade architecture to efficiently evaluate a large number of pose-specific classifiers, framing their learning task as a discriminative multi-class classification problem.

In another recent work, Rogez *et. al.* utilize egocentric RGB-D images to predict the 3D kinematic hand grasp, contact points, and force vectors of a person's hand during scenes of in-the-wild, everyday object interactions and manipulations [9]. They show that a combination of RGB and depth data is important for hand detection and pose estimation, where depth information is crucial for detection and segmentation, while the richer RGB features allow for better grasp recognition. These results provide a strong rationale for exploring new modes of data, as these new modes can add new dimensions and insights for classification.

2.2. Region Proposal Methods

Some works have also explored various region proposal methods for object recognition. J.R.R Uijlins *et. al.* introduce the method of selective search, which combines the strengths of both exhaustive search and segmentation. This method attempts to diversify the search for possible objects, yielding a smaller set of higher quality, data-driven, locations of interest [10]. Selective search reduces the number of locations when compared to exhaustive search, which enables the use of stronger machine learning techniques for object recognition. Other region proposal algorithms similar to selective search include RandomizedPrim's [5], Rantalankila [6], Objectness [1], and MCG [2]. We use selective search as our region proposal method since it is a very popular technique that is the method of choice for many state-of-the-art object detectors such as Fast R-CNN [4].

2.3. CNN Architectures

While there are a wide variety of CNN architectures available, we choose to use Fast R-CNN for our hand detection task. Girshick proposes this CNN architecture, which has several advantages over the traditional R-CNN, including higher detection quality and single stage-training using a multi-task loss [3]. Another more recent variant called Faster R-CNN [7] also exists; however, we use a combination of selective search and Fast R-CNN in order to see how well selective search does in comparison to the CNN used for region proposals in Faster R-CNN.

2.4. Our Approach

Since a combination of RGB and depth images can be successfully used for hand detection, we believe that adding thermal as a third mode of data can further improve results. We experiment with thermal videos as a new mode of data for hand detection.

3. Methods

3.1. Technical Overview

The general pipeline for our project can be seen in Figure 1. First, we acquire egocentric multi-modal video data of people performing various everyday tasks. After the data acquisition, we annotate ground truth bounding boxes for the RGB, depth, and thermal video frames. In parallel, we run selective search on those same video frames to obtain region proposals for possible objects in each image. The images, along with their ground truth bounding boxes and the selective search region proposals, are then fed into a Fast R-CNN architecture (which we finished implementing in Tensorflow) for training. Finally, the Fast R-CNN outputs predictions for bounding boxes around hands for a test data set, and we look at the results from each modality - RGB, depth, and thermal. The next section delves more deeply into the details of each step in the pipeline.

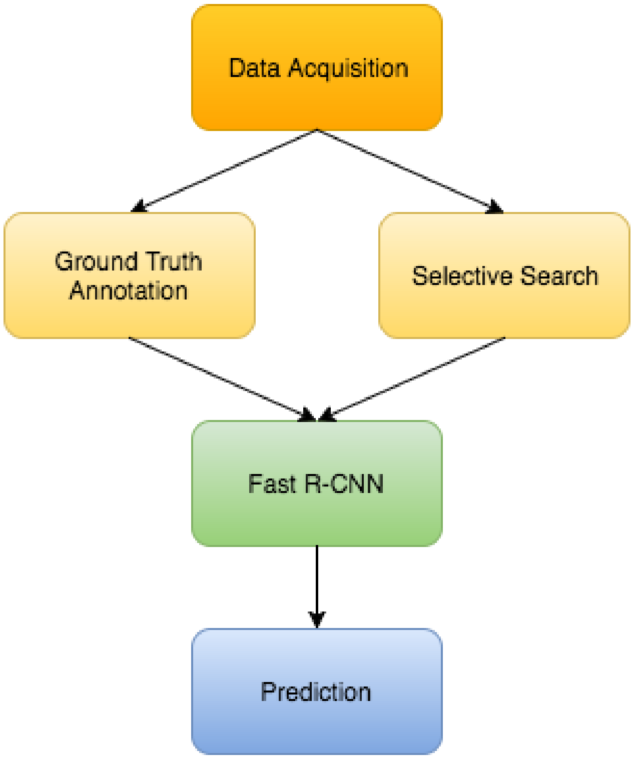


Figure 1. Pipeline for the hand detection project

3.2. Technical Details

Data collection

We built a chest-mounted egocentric camera setup consisting of a Flir One thermal and RGB camera and an Intel DepthSense depth and RGB camera attached to a GoPro chest harness, as shown in Figure 2, in order to collect multi-modal video data. We then recorded egocentric video sequences of different people performing various everyday tasks, such as washing dishes, opening doors, typing, and preparing food. By recording many different types of tasks for several different people, our training data can portray a more varied range of hand positions and gestures and be more robust and generalizable. A short sample of a few video frames from one video sequence in which the user heats up food in a microwave can be seen in Figure 3. From top to bottom, these rows correspond to the RGB (from the Intel camera), depth, RGB (from the Flir camera), and thermal video sequences. The RGB and depth video frames are 640x480 pixels each, and the thermal video frames are 160x120 pixels each.

Ground Truth Annotation

We needed ground truth bounding-box annotations for each image in order to train our Fast R-CNN model. To find these ground truth bounding boxes for all hands or partial hands,

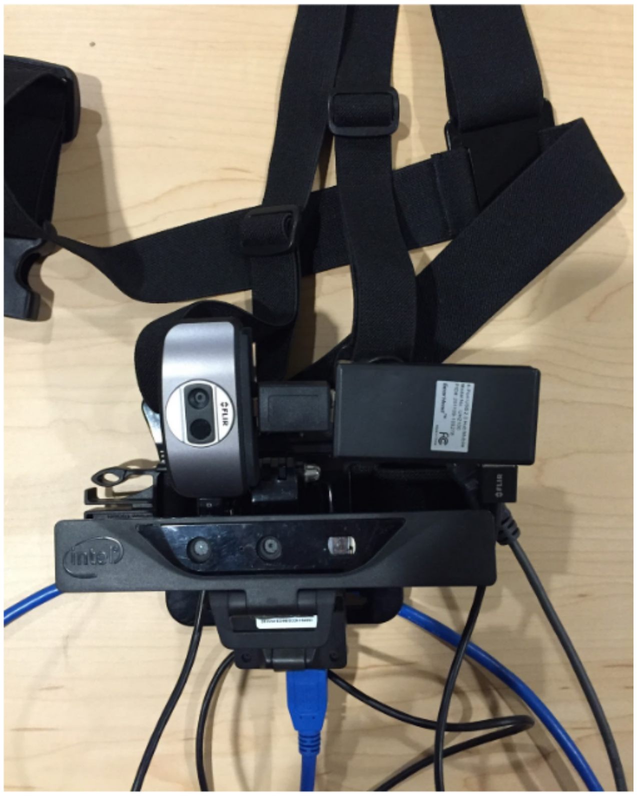


Figure 2. Chest-mounted egocentric camera setup with both RGB-T and RGB-D cameras attached to a GoPro chest harness

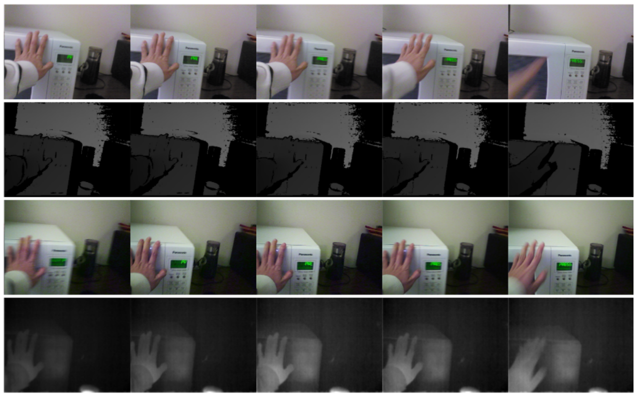
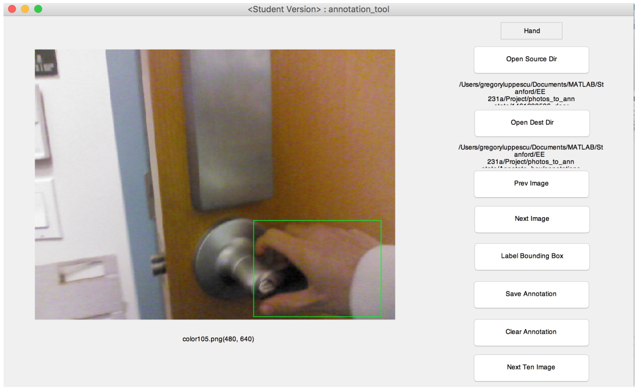


Figure 3. Multi-modal video data sequence. From top to bottom: RGB (from depth camera), depth, RGB (from thermal camera), and thermal video sequences

we manually annotated about 2000 RGB images, 1000 thermal images, and 1000 depth images. Each bounding box was defined by its top left corner and its bottom right corner, and we used a tool written as a Matlab GUI to expedite the annotation process. The annotations for each image were saved as .mat files, and we then wrote a Matlab script to save these files in the .txt format required by our Tensorflow implementation of Fast R-CNN. Our annotation tool is shown in Figure 4.



/Users/gregoryluppescu/Documents/MATLAB/Stanford/EE 231a/Project/photos_to_ann

/Users/gregoryluppescu/Documents/MATLAB/Stanford/EE 231a/Project/photos_to_ann

color105.png(480, 640)

Figure 4. Annotation tool used for capturing ground truth bounding box annotations

Selective Search

We implement and run the selective search algorithm for every video frame for every video sequence in our dataset. Selective search greedily merges superpixels based on engineered low level features [10]. It produces a bounding box around anything that might be an object in an image, and these bounding boxes can then be used as region proposals. An example of the top 20 bounding boxes returned from selective search can be seen in Figure 5. After obtaining all selective search results, we wrote a Matlab script to save these files in the .txt format required by our Tensorflow implementation of Fast R-CNN.

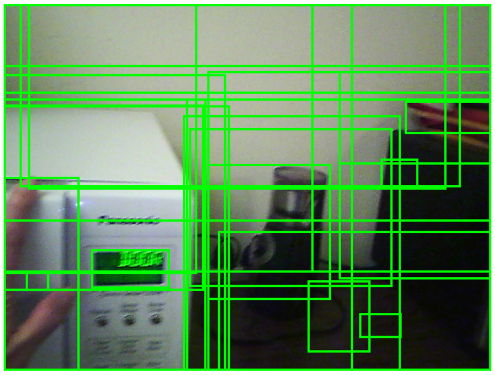


Figure 5. Visualization of the top 20 region proposals output by the selective search algorithm for one image

Fast R-CNN

Fast R-CNN is a method for efficiently classifying object proposals using deep convolutional networks [3]. It takes in an image and multiple regions of interest (RoIs) as inputs into a fully convolutional network. These RoIs are then pooled into fixed-size feature maps and mapped into feature vectors through fully connected layers. Fast R-CNN outputs two vectors for each RoI: softmax probabilities and per-class bounding-box offsets. There exist other CNN architectures for object detection such as Faster R-CNN; however, we chose to use Fast R-CNN because we wanted to see if selective search returns better region proposals for our purposes than the CNN used for region proposals in Faster R-CNN. In the future, we also plan to do hand detection on our dataset in Faster R-CNN and compare results from the two architectures.

The Fast R-CNN architecture is natively implemented in Caffe [3], and this code is open source. However, we wanted to use the Tensorflow framework in case we wanted to use RNNs with our videos in the future. Thus, we finished a Tensorflow implementation of Fast R-CNN, and this process comprised the majority of the time spent on this project. After finishing and debugging the implementation, we wrote a Python class for our ”hands” dataset, and divided our video sequences into training and test categories. We used five video sequences for training, and one video sequence for testing. Each video sequence includes a few thousand frames. We initialize the Fast R-CNN with pretrained weights from ImageNet, and then trained it for the RGB video sequences, the depth video sequences, and the thermal video sequences.

4. Experiments

We divided our dataset into five video sequences for training and one video sequence for testing, where each video sequence contained a few thousand frames. We then trained our Tensorflow implementation of Fast R-CNN for hand detection on the RGB video frames, the depth video frames, and the thermal video frames, and the results are below.

4.1. RGB Results

For the RGB video frames, we trained our Fast R-CNN implementation for 40,000 iterations with a learning rate of 0.001 (that decayed after 30,000 iterations), a momentum of 0.9, and a batch size of two images (with a few thousand corresponding region proposals). The training loss is shown in Figure 6 below.

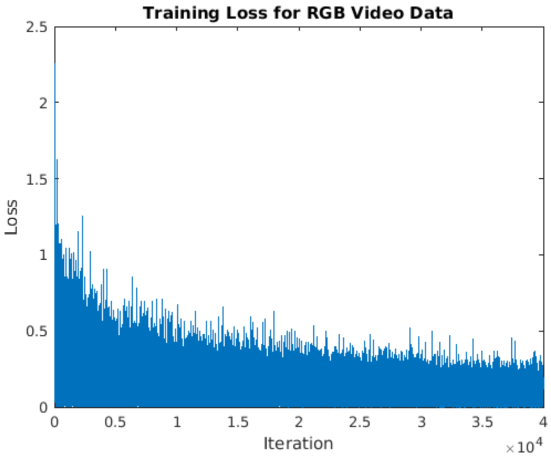


Figure 6. The training loss for the RGB video data over 40,000 iterations

We then tested our model on the remaining video sequence with non maximum suppression, and wrote a Matlab script to visualize the output bounding boxes. We found that our model returned many potential ”hands” bounding boxes, but most of them had very low confidence scores. When we used a threshold confidence level above which any regions output by Fast R-CNN could be considered ”hands” but below which the regions were considered ”not hands,” the results looked much better. Several examples of the hand detections for our RGB video frames are shown below in Figures 7, 8, 9. Three images show successful hand detections, one image that includes no hands shows a successful lack of hands detected, and one image shows a missed hand detection.

After testing our model with the test data, we wrote Matlab scripts to generate precision-recall curves and compute the average precision. For this test sequence, our model had an average precision of 45.1. The precision-recall curve is shown in Figure 10.

4.2. Depth Results

For the depth video frames, we again trained our Fast R-CNN implementation for 40,000 iterations with a learning rate of 0.001 (that decayed after 30,000 iterations), a momentum of 0.9, and a batch size of two images (with a few thousand corresponding region proposals). The training loss is shown in Figure 11 below.

We again found that our model returned many potential ”hands” bounding boxes, mostly with very low confidence scores. Thus, we thresholded the output regions by confidence level as before, and show several examples of the resulting hand detections below in Figures 12, 13, 14. We found that a lower confidence threshold was needed for the depth images than for the RGB images.

For this test sequence, our model had an average precision of 26.6. The precision-recall curve is shown in Figure 15.

4.3. Thermal Results

For the thermal video frames, we trained our Fast R-CNN implementation for 40,000 iterations with the same hyperparameters as before: a learning rate of 0.001 (that decayed after 30,000 iterations), a momentum of 0.9, and a batch size of two images (with a few thousand corresponding region proposals). The training loss is shown in Figure

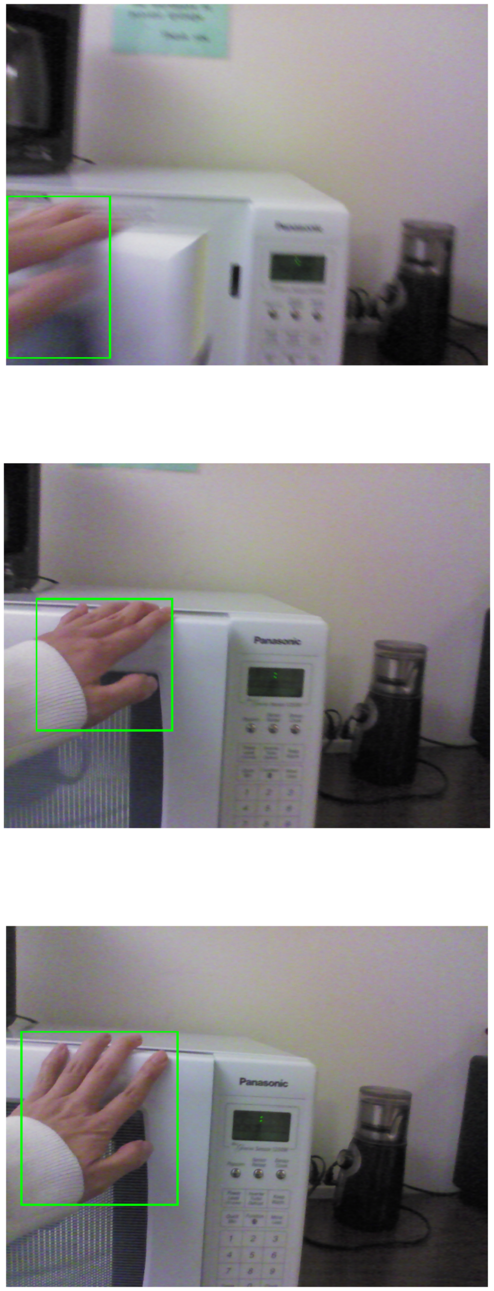


Figure 7. Three examples of successful hand detections for the RGB video data

16 below.

We thresholded the output regions by confidence level as before, and found that the thermal images required an even lower confidence threshold than either RGB or depth. Several examples of the resulting hand detections are in Figures 17, 18, 19.

For this test sequence, our model had an average precision of 12.7. The precision-recall curve is shown in Figure 20.

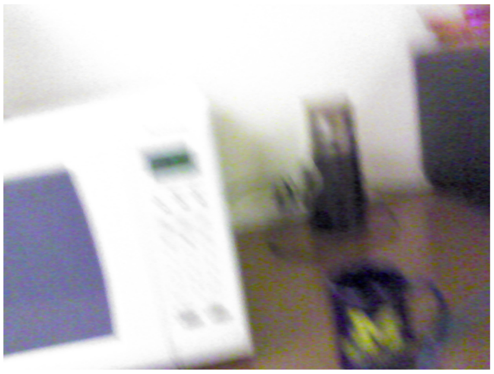


Figure 8. An example of a successful lack of hands detected for the RGB video data

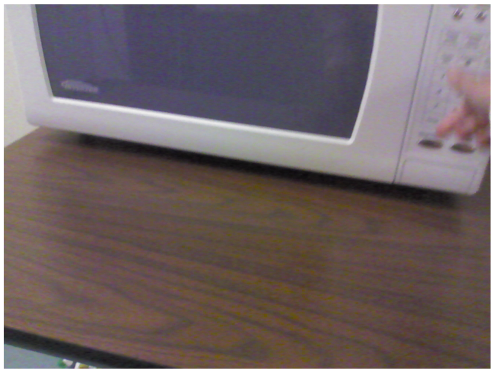


Figure 9. An example of a missed hand detection for the RGB video data

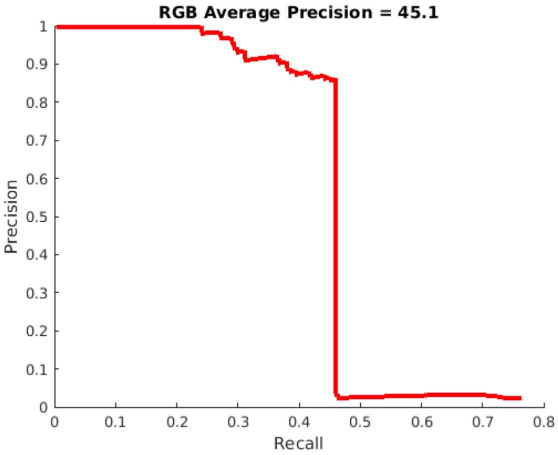


Figure 10. The precision-recall curve for the RGB video data

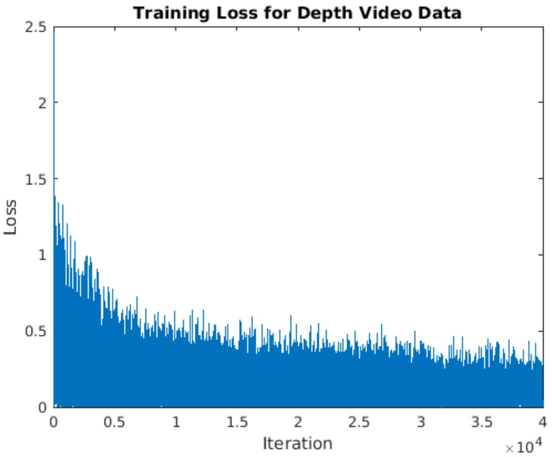


Figure 11. The training loss for the depth video data over 40,000 iterations

4.4. Discussion

Hand detection using the RGB video data seems to perform best on our dataset, both visually and in terms of the average precision. The depth video data seems to be the next best, followed by the thermal video data. One hypothesis for the relatively poor thermal detection results is that the thermal images are much smaller, with only 1/16 the pixels of the other images, and thus the hands are also smaller and harder to detect. Additionally, because the Fast R-CNN architecture includes RoI pooling layers, the already-small bounding boxes shrink even more. One way to test this hypothesis is by rescaling the thermal images and re-training our model - we initially only trained our model with the original size images due to time constraints. Another potential reason for the lower thermal performance is that our model for the thermal data may be more easily fooled by bright spots in images. Because the test data that we used was a video sequence in which the user heated up food in a microwave, there were several bright spots in each image that were often mistaken for hands - these false positives explain the low precision shown in the precision-recall curve.

The depth data falls between the RGB and the thermal data in terms of its performance, and the RGB data performs the best. One interesting item of note in the RGB precision-recall curve is the sharp drop in precision in the middle of the plot. This steep cutoff suggests that above a certain confidence threshold, the detection results are very good, but below that threshold, the results are very poor. This idea is also supported through visualizations of the hand detections.

5. Future Work

There are many directions that we can take with this project in the future. First, we could try to improve the thermal detection results, for instance by rescaling the thermal images. We could also implement hand detection in Faster R-CNN to compare results between these methods. Next, we could use all three data modalities together (RGB, depth, and thermal) to make improved hand detection predictions. One method that we had in mind for combining these data modalities is to concatenate the RGB, depth, and

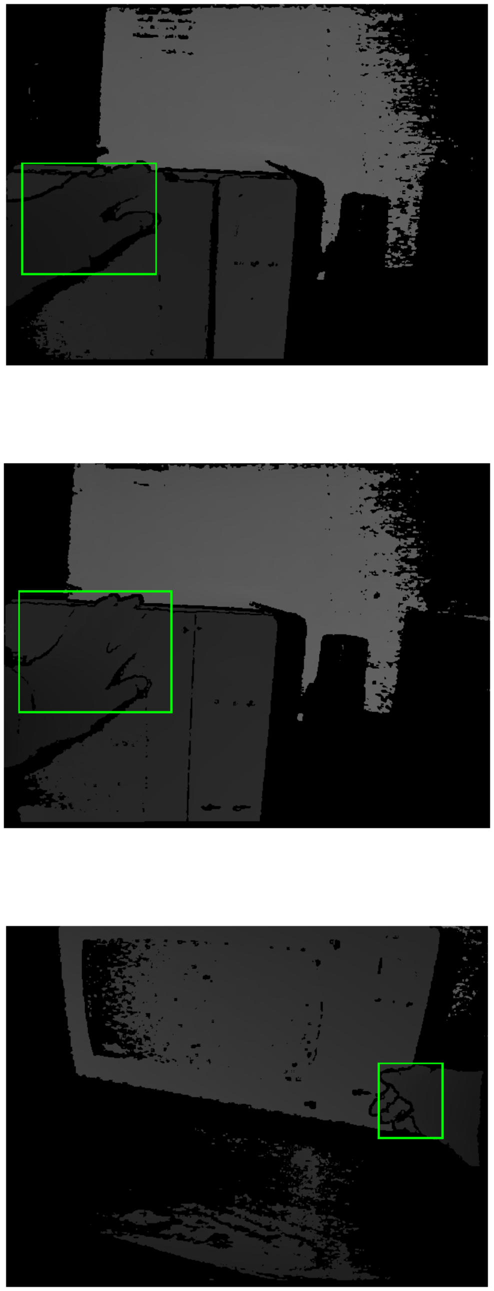


Figure 12. Three examples of successful hand detections for the depth video data

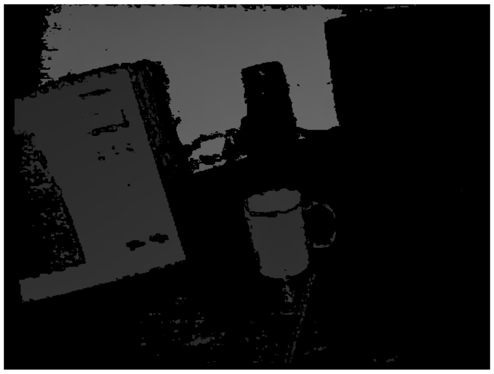


Figure 13. An example of a successful lack of hands detected for the depth video data

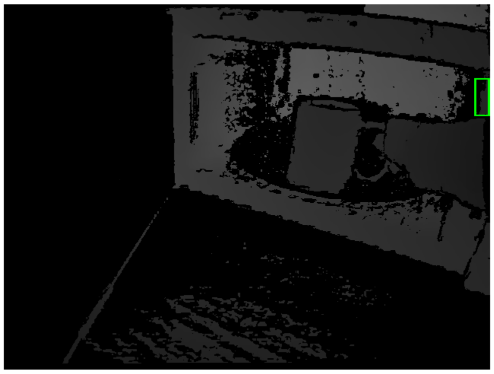


Figure 14. An example of a mistaken hand detection for the depth video data

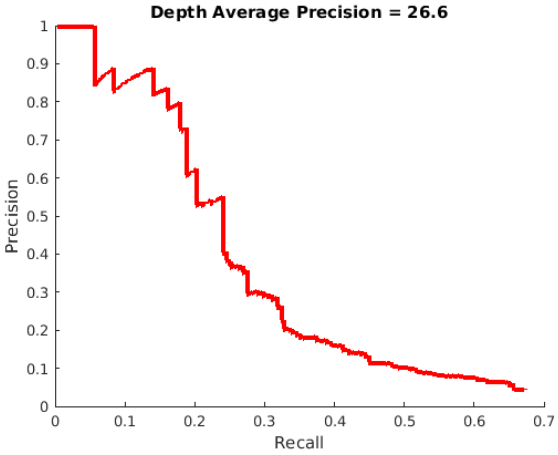


Figure 15. The precision-recall curve for the depth video data

thermal images into an input volume with a depth of five (instead of the usual three) to input into a CNN and change the architecture of the first layer to accept a depth of five.

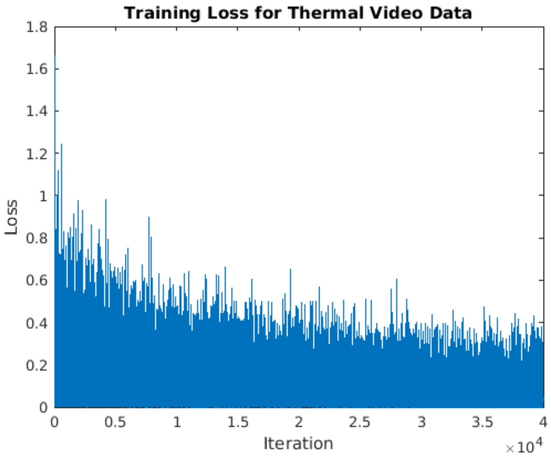


Figure 16. The training loss for the thermal video data over 40,000 iterations

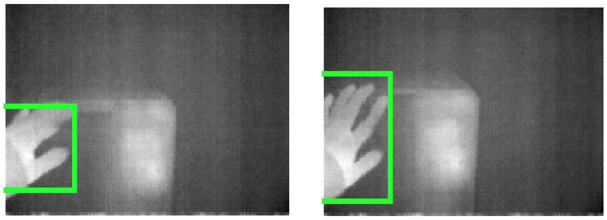


Figure 17. Two examples of successful hand detections for the depth video data

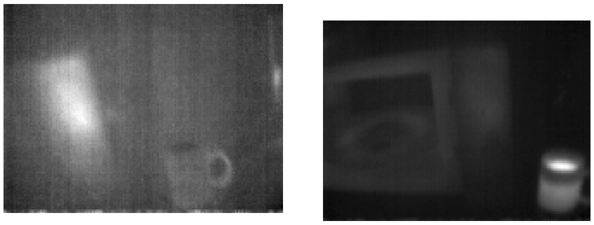


Figure 18. Two examples of successful lack of hands detected for the thermal video data

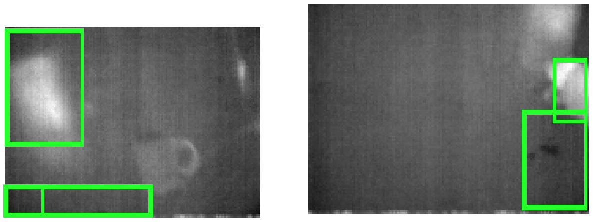


Figure 19. Two examples with too many hand detections for the thermal video data

To come up with the new region proposals, we could select

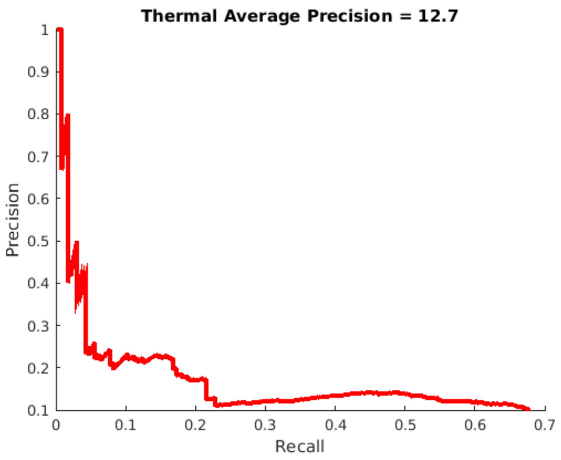


Figure 20. The precision-recall curve for the thermal video data

areas where the original region proposals for the different data modalities overlap significantly and take the smallest rectangular area that still includes all three original region proposals. Finally, we could extend the hand detections to hand pose estimations.

6. Conclusions

We implemented hand detection methods using novel egocentric, multi-modal video data of everyday activities. We built an egocentric, chest-mounted, multi-camera setup, recorded many video sequences of everyday activities, annotated thousands of video frames, ran selective search on all of our data, finished a Tensorflow implementation of Fast R-CNN, visualized our output hand detections, and provided some discussion of our results. We found that the hand detection performed best on the RGB data, followed by the depth data, followed by the thermal data; however, we hope to further improve the thermal hand detections in the future and combine all three modalities for even better results. Some of the issues that we faced during this project included debugging the Tensorflow code for Fast R-CNN, and using only very small thermal images. Code will be submitted privately to the TAs.

Acknowledgments

We would like to thank Professor Silvio Savarese and the TAs of CS 231A for providing valuable insights and suggestions throughout the course of this project. We would also like to thank Dr. Yu Xiang for his many useful explanations and guidance. Thanks for a great class!

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? *Proc. IEEE Int. Conf. Comput. Vis Pattern Recog.*, pages 73–80, 2010.

[2] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. *Proc. IEEE Int. Conf. Comput. Vis Pattern Recog.*, pages 328–335, 2014.

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

[4] J. Hosang, R. Benenson, D. Piotr, and B. Schiele. What makes for effective detection proposals. *Transactions on Pattern Analysis and Machine Intelligence*, 38(4), 2016.

[5] S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized prim's algoirthm. *Proc. IEEE Int. Conf. Comput. Vis Pattern Recog.*, pages 2536–2543, 2013.

[6] J. K. Rantalankila and E. Rahtu. Generating object segmentation proposals using global and local search. *Proc. IEEE Int. Conf. Comput. Vis Pattern Recog.*, pages 2417–2424, 2014.

[7] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[8] G. Rogez, J. Supancic, M. Khademi, J. M. M. Montiel, and D. Ramanan. 3d hand pose detection in egocentric rgb-d images. *International Journal of Computer Vision*, 2014.

[9] G. Rogez, J. Supancic, and D. Ramanan. Understanding everyday hands in action from rgb-d images. *ICCV*, pages 3889–3897, 2015.

[10] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2012.