Is that my hand? An egocentric dataset for hand disambiguation

Sergio Cruz[a,*] (Researcher), Antoni Chan[a,*] (Researcher)

[a] *City University of Hong Kong, Tat Chee Avenue, Kowloon Hong Kong SAR*

ARTICLE INFO

Keywords:
Egocentric perspective
Hand detection
MSC: 68T45
MSC: 92B20

ABSTRACT

With the recent development of wearable cameras, the interest for research on the egocentric perspective is increasing. This opens the possibility to work on a specific object detection problem of hand detection and hand disambiguation. However, recent progress in egocentric hand disambiguation and even hand detection, especially using deep learning, has been limited by the lack of a large dataset, with suitable variations in subject, activity, and scene. In this paper, we propose a dataset that simulates daily activities, with variable illumination and people from different cultures and ethnicity to address daily life conditions. We increase the dataset size from previous works to allow robust solutions like deep neural networks that need a substantial amount of data for training. Our dataset consists of 50,000 annotated images with 10 different subjects doing 5 different daily activities (biking, eating, kitchen, office and running) in over 40 different scenes with variable illumination and changing backgrounds, and we compare with previous similar datasets.

Hands in an egocentric view are challenging to detect due to a number of factors, such as shape variations, inconsistent illumination, motion blur, and occlusion. To improve hand detection and disambiguation, context information can be included to aid in the detection. In particular, we propose three neural network architectures that jointly learn the hand and context information, and we provide baseline results with current object/hand detection approaches.

1. Introduction

With the recent advancements of technology, new devices have been developed like wearable cameras. Wearable cameras, such as the GoPro cameras and Google Glass, have become more accessible to the public, which creates the demand for solutions to the challenges these cameras present, e.g., the egocentric perspective. People have used these cameras to record their daily activities, and especially to do outdoor sports like hiking, surfing and biking through different places, since some of these cameras can be waterproof. This creates the need to address the specific style of egocentric videos as well as the variable complex environments in which they are shot. The egocentric perspective provides interesting characteristics as the videos move quite swiftly, which makes the images and objects blurry and hard to analyze.

The hands are the most consistent objects in the egocentric perspective and they appear in a reasonable size that allows enough features to be extracted for detection when doing daily activities. However, the challenging characteristics of the egocentric perspective leads to people doing work on controlled environments by staying in a small set of rooms [5], but neglecting other daily activity conditions.

Hand detection is the first step for high-level analysis, such as activity recognition [24, 5], or providing the hand-pose [31, 21]. Hand detection is a special case on the object detection problem, as the shape variations of the hands are large – the many joints of the fingers change the appearance of the hand drastically, creating a challenging detection and disambiguation task. From the egocentric perspective, the hand detection problem has been addressed by focusing on hand properties more than the hands themselves [16, 27]. Some works [3, 15] choose to approach the problem using segmentation, which makes them able to detect hands and other body parts that resemble the skin.

Currently there only exist a few datasets for egocentric hand detection, but they are built for specific problems like illumination changes [16], where the goal is to segment skin from the background, or people's interaction [2], by having them playing games and recognizing the activity using the hands. These datasets are quite small (less than 5000 images) and consist of a limited variability in terms of people, gender, ethnicity, activities and scenes. Other databases focus on fine-grained actions, such as pouring water, open a container [17], or using using the washroom and doing house chores [22]. These activities however only stay indoors while doing the activities, reducing the variability on illumination, and have similarities with each other when it comes to the hand gestures. Other works [34] present a segmentation database aimed for real world videos by taking videos from YouTube of people doing different activities, indoors and outdoors, however there is no information on the activities, people, or places (see Table 1).

In this work, we propose a new dataset for egocen-

*Corresponding author
scruzgome2-c@my.cityu.edu.hk (S. Cruz); abchan@cityu.edu.hk (A. Chan)
ORCID(S): 0000-0001-9444-7038 (S. Cruz); 0000-0002-2886-2513 (A. Chan)

Table 1
Comparison of datasets for egocentric hand detection.

Dataset	# Images	# Hand Annotations	Resolution	# People	# Activities	# Places	# Ethnicity	# Gender
EDSH [16]	720	1,414	720p	1	2	4	1	1
EgoHands [2]	4,800	13,834	720p	4	4	3	1	1
ADL dataset, [22]	32,662	4,651	960p	20	18	20	-	-
EGTEA Gaze+, [17]	13,847	15,176	960p	106	32	-	-	-
EgoYouTubeHands, [34]	1,290	2,600	-	-	-	-	-	-
Epic Kitchens dataset [4]	11.5M	116	1080p	32	124	32	-	-
Ours	50,000	95,001	1080p	10	5	40	5	2

tric hand disambiguation that addresses the disadvantages of previous datasets. Our dataset consists of daily life activities, with variations of illumination and people from different cultures and countries, and increased variability of skin color and gender. Our dataset is substantially larger than the previous datasets, comprising 50 videos with 10 subjects from different ethnicity, performing daily activities in 40 different scenes. Our dataset has bounding boxes annotations for 50,000 images (over 95,000 annotated hands), which allows data-hungry methods like deep neural networks (DNNs) to be used. Here we focus on hand detection and disambiguation rather than hand segmentation, since we want to differentiate the hands from the arms and because hands are used to interact with objects. We also only focus on detecting the camera wearer's hands, as this provides the most usefulness for future work. Using our dataset, we provide statistical analysis of the hands location and size from the egocentric perspective.

In some cases, the hands by themselves may be difficult to detect due to occlusion, unseen shape variations, changes in illuminations, and motion blur. However, the hands tend to appear with consistent surrounding context. For example, the hands will always appear next to arms, they will sometimes grasp objects, and left and right hands tend to be on certain sides of the image. To exploit this context information, we propose three neural network architectures that jointly learn the hand and context information to improve hand detection/disambiguation performance. We perform benchmark experiments on our dataset using our proposed networks and other recent approaches in hand detection.

In summary, the contributions of our work are as follows: 1) we collect a large dataset for egocentric hand detection, containing large variations in people, activity, scenes, and ethnicity, and which is larger than the previous datasets; 2) we propose different neural networks architectures that jointly learn hands and its surrounding context features; 3) we conduct baseline experiments for benchmarking the current state-of-the-art in hand detection. The remainder of this paper is organized as follows. In Section 2 we discuss related work. We introduce our dataset in Section 3, and our hand detection framework in Section 4. Finally in Section 5 we present benchmark experiments.

2. Related Work

2.1. Egocentric Hand Datasets

Hand detection and disambiguation from the egocentric perspective is a relatively new problem with not much work on it, nor datasets that address it. The existing datasets focus on specific purposes and controlled environments. One of first datasets to address hand detection from the egocentric perspective is from [16]. Their dataset is based on hand segmentation, as a surrogate for the hand detection problem. Hence, their dataset focuses on illumination changes, by having a person going through different rooms, stairs and even outside, which provides a substantial amount of variability when it comes to color and texture. However, this dataset only contains one person's hands which suggests that methods developed on this dataset might have a tendency to overfit a specific person's hand color. Furthermore, while moving through the rooms, the person barely interacts with the world, e.g., only opening doors or using kitchen items.

The dataset proposed by [2] called "EgoHands" addresses the hand detection from egocentric perspective with an object detection dataset. This dataset focuses on human interaction and has two people in the video facing one another, and interacting by playing board games: playing cards, playing chess, solving a jigsaw puzzle and playing Jenga. This dataset contains overall four people, rotating between the videos to add variability, but all people are male with similar ethnicity, which creates similar data among them. The recordings were taken in three locations: a table on a conference room, a patio table, and a coffee table inside a home. This adds variability in terms of the backgrounds and illumination, as the recordings were taken on different days with people wearing different clothes. However, recordings were of a static nature, as they never move from the tables, which does not add the shaking and blurry artifacts that are unique for the egocentric perspective. Furthermore, while the dataset can be used to analyze people's interaction, it does not provide actions from daily activities. With the variability on places (3), on people (4) and on activities (4) the dataset contains 48 unique combinations of videos, with 4,800 annotated frames with pixel-level masks (15,053 annotated hands). However, due to the nature of the Google Glass cameras used, the dataset has a large percentage of hands that are not the camera wearer's hands, since the camera field-of-view is not wide enough to capture the wearer's hands. This leads to the majority of the hands to be the other actor's hands, which limits the possibility of analyzing the wearer's actions.

The Epic Kitchens dataset [4] addresses the action recognition and object detection problem. It contains people recording themselves inside a kitchen over three consecutive days, with only one person in the frame at all times performing fine-grained actions, e.g. put,

take, open, and close various objects. This dataset contains annotations of activity segments and generic object bounding boxes. However, it only contains 116 hand/finger bounding box annotations, which is small compared to the overall dataset size, as the dataset is not specific for hand detection. This dataset also contains only indoor activities (Kitchen), but they record at different hours with different lightings, which provide with illumination changes.

The ADL dataset [22] focuses on generic object detection and action recognition using the egocentric perspective, but it is not hand detection specific and the videos are indoors, limiting the illumination variability. This dataset has people doing unscripted everyday activities. One of the difference of this dataset from typical actions is that they can involve long-scale temporal structure like making tea that can take a few minutes, and complex object interactions like having a fridge looking different when its door is open.

The EGTEA Gaze+ dataset [17] focuses on action recognition using an egocentric perspective, by having a person interact with multiple objects. This dataset includes cooking activities from 86 unique sessions of 32 subjects. This dataset however stays indoors and the activities are restricted to fine-grained actions, such as "Cut bell pepper" or "Pour condiment (from) condiment container into salad".

Finally, the EgoYouTubeHands dataset [34] focuses on any hands in the egocentric perspective without any constrained daily settings. They took 3 different egocentric videos from YouTube in which the people are doing different activities and interacting with each other, yielding different hand characteristics (both the wearer's and others') from the egocentric perspective. The nature of the videos gives the database high illumination variability and background changes.

Table 1 presents a comparison with previous datasets and our proposed dataset. Our dataset contains a considerable increase in number of images, which is needed for data-hungry methods like DNNs. Our dataset also has increased resolution (1080p) and field-of-view, which ensures that the hands are almost always in the image. Our dataset has a variety of activities, locations, subject ethnicity and gender, which yields more daily hand gestures that are necessary for real-world applications.

2.2. Egocentric Hand Detection and Segmentation

One of the first to handle something close to hand detection was [27], which concentrates on any object that behaves like a hand. They approach this problem by segmenting the image using optical flow patterns, and detecting if the pattern corresponds to that of a hand, since there is a noticeable difference between the flow of the hands and the background. However, this makes any object that moves like a hand to be detected.

[16] analyzed what features are the best for segmenting hands from the background. They use a random forest to try different features and find the best combination that differentiates the skin color from the background. However, this approach segments all skin (including arms) from the background instead of only the hands.

[2] generated bounding box proposals using a probabilistic model with 3 aspects: the occurrence probability of the hands being in the picture, the probability of a bounding box having a specific size and location, and the probability of the center of the bounding box being of skin color. Then they train a CNN to classify hands from the background and another one to disambiguate the hands.

Hand detection can be used in a variety of higher-level analyses of egocentric video. For example, detecting when the hands are interacting with different objects can be used for object recognition and tracking [16, 13, 7, 27, 21]. Following this, works have focused on hand pose estimation [31] by locating the hand joints in 2D or 3D and recognizing hand actions [9]. Other works [24, 5, 6, 23, 28, 29] focus on object detection in general to perform activity recognition, where a person changes rooms and interacts with many different objects to represent the activity. Since the egocentric perspective is generated by the user, video retrieval and summarization [20, 14, 19, 30] can be used to obtain an overall information about the daily life of the wearer, by visualizing the most important parts of the egocentric perspective as the time passes, creating stories from them. Egocentric hand detection also can be used for virtual and augmented reality [11], and other works that get information out of this perspective [8].

3. Daily Egocentric Dataset

In this work we propose a dataset that keeps the key characteristics from previous datasets, and adds more variability on people, activities and places, to simulate daily life situations. We create challenging background and illumination changes, by having different rooms and different illuminations, indoor and outdoor environments, and static and moving cameras. We have released the datatset on the web [1] with ground truth using the Matlab file format.

3.1. Data Collection

In order to add variability on the people, in our dataset we select 10 different subjects for recording the videos. Our dataset contains people from different cultural ethnicity, to add variability in skin color and hand gesture. The dataset subjects consist of 1 subject of Hispanic descent, 1 subject of African/East Asian descent, 1 subject of European descent, and the rest from

[1] https://github.com/sercruzg/EgoDaily

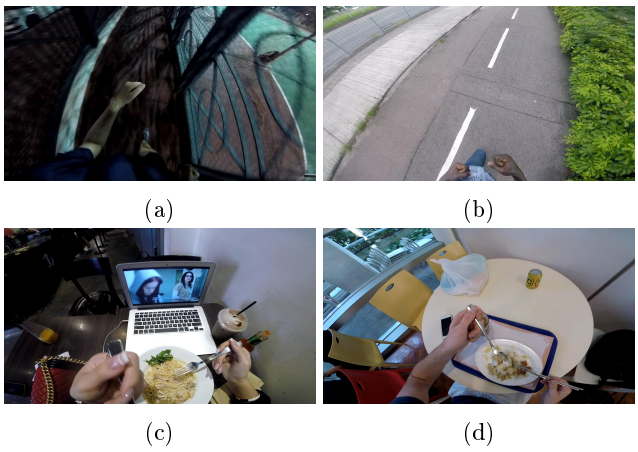


Figure 1: Dataset samples showing illumination variation provided by recording during the day (a) and night (b), outdoor (top) and indoor (bottom), and with different genders: woman (c) and man (d).

various Asian descents. We also add variability in gender by having 5 men and 6 women[2], as seen in Figure 1.

To add variability on the activities, our dataset introduces 5 different daily-life activities, which are representative of common egocentric videos: biking, eating, kitchen, office, and running. "Biking" has a person riding or interacting with a bike through real biking routes, which include passing through bridges, tunnels and streets. "Eating" has a person eating a dish using different kitchenware, such as fork, knife and chopsticks. "Kitchen" has a person interacting with the kitchen by making something to eat (sandwich or oatmeal) and some tea, and then doing doing the dishes. "Office" concentrates on the person interacting with office objects, such as typing on a computer keyboard, writing in a notebook with a pen, and using a stapler and a seal on the notebook. "Running" has the person running through different routes, including parks, streets and football courts. Figure 2 (top) showss samples images from the activities in the dataset, along with the cropped hand samples (bottom). We did not give detailed instructions to the people on how to do the activities, so that the videos contain real hand movements and gestures.

For each activity, we focus the camera view on the objects the person is interacting with: for "biking", the camera focuses on the handles; for "eating", the camera focuses on the dish the person is eating; for "kitchen", the camera focuses on the sink and the objects the person is handling; for "office", the camera focuses on the keyboard and the notebook; for "running", since the hands move a lot, the camera focuses as close to the body as possible to capture the hands the most (this makes the videos for "running" to partially neglect the road).

To have sufficient variability in background scenes, each recording is made in a different place as much as possible (up to the limit of availability): 9 places for "biking", 7 for "eating", 7 for "kitchen", 7 for "office", and 10 for "running". In total there are 40 different places for the recordings. As for the recordings that share the same place, we captured at different hours (day and night) to add variability in the background and illumination.

The videos were recorded by a GoPro Session 5 camera using the "superview" mode that resembles the egocentric perspective the most. This mode provides a wide field-of-view, which allows the camera to capture the wearers hands in most situations, allowing for a larger dataset. The resolution of the recorded video is 1920×1080 (1080p) at 60 frames per second. For this dataset we have 10 different subjects, doing 5 different activities, resulting in 50 unique combinations of videos. The average length of the videos is 7.8 minutes for each subject/activity combination, with average 470 seconds, and 29,010 frames. The dataset is constructed by selecting 1,000 images uniformly from each video, resulting in 50,000 images. The groundtruth hand bounding boxes were annotated manually containing the whole hand up to the wrist, resulting in over 95,000 annotated hands.

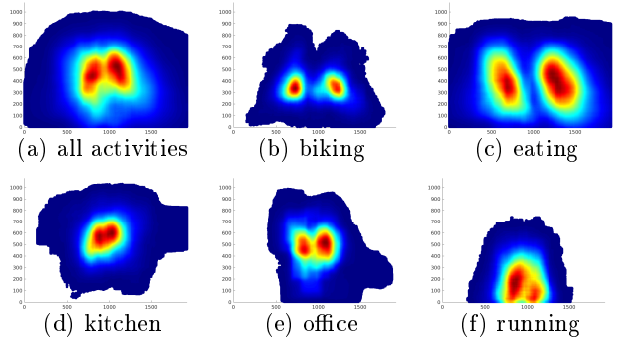
3.2. Dataset Statistics: Hand Location and Size

The egocentric perspective provides specific characteristics when it comes to hand detection, as the hands appear in specific regions of the image when interacting with objects depending on the activity the person is doing. Figure 3 shows a distribution of where the hands appear on the image in our proposed dataset for the different activities. Overall in the dataset, the hands are focused on the center as expected for the egocentric perspective. In "biking", the hands focus on grabbing the handles, and such their location is quite defined. As for"eating" the hands locate similarly with "biking" but with more movement, as they concentrate on bringing the food to the mouth and that has as consequence the hand going lower in the image. There is also a difference on the location between the right and left location – right-handed people move their right hands more often in this activity. For "kitchen", the hands are most concentrated in one location. This is due to the activity needing both the hands interacting with different objects, making both hands appear closer to each other than any other activity. Looking at "office", there is a less concentrated area for the hands as when you type on the computer or write on paper the hands do not need to be as close to each other as previous activities. The right-handedness of the people in the dataset also impacts the location, with the right hand being slightly more present in the image. Finally, "running" has the

[2]We have 2 people acting as one subject due to one person's availability and another person's health condition.

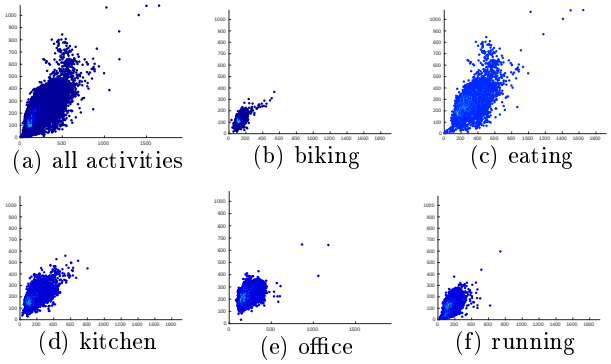
(a) biking (b) eating (c) kitchen (d) office (e) running

Figure 2: Samples from the proposed egocentric dataset: (top) scenes of 5 activities (a-e) showing the variation of background and illumination, along with hand samples (bottom) showing the annotations and the variation of hand gestures.



(a) all activities (b) biking (c) eating (d) kitchen (e) office (f) running

Figure 3: Probability of an image pixel to be contained within a hand bounding box (red color is highest probability, while dark blue is lowest), showing the location of the hands in (a) all activities, and (b-f) each activity separately.



(a) all activities (b) biking (c) eating (d) kitchen (e) office (f) running

Figure 4: Scatter plots of the width (horizontal axis) and height (vertical axis) of the hand bounding boxes, for (a) all activities, and (b-f) each activity separately.

most identifiable location in the dataset – the location of the hands tend to be in the lower part of the image, and also go outside of the picture the most in this activity,

Figure 4 plots the size of the hand bounding boxes over the whole dataset and for each activity. Both the location and size of the hands change throughout daily activities. Over the whole dataset, the hand size varies significantly. For “biking", the size is quite consistent as the person is grabbing the handles and the size changes little. “Eating" has the largest variation in hand size among the activities, since the person is interacting with the plates and bringing the food to the mouth, which makes the hand bigger as it gets closer to the camera. For “kitchen", the size is consistent as the person concentrates on different objects, and the hand stays on a relative same distance from the camera. “Office" has the most compact size overall as the hands remain on quite similar distance from the camera – the hands are interacting with the computer and the notebook, and do not move much or change distance from the camera. Finally, for “running", the hand boxes have small size as the hands tend to go out of the picture, resulting in only portions of the hands being in the image.

3.3. Experiment Protocol

The nature and size of our dataset allows for a robust analysis that can distinguish itself from other datasets. We suggest an experiment protocol consisting of 5-fold cross validation across the subjects, i.e., *leave-2-subjects-out testing*, where in each trial, 2 subjects are used for the testing and the rest for training. In this way, methods are tested on how they extrapolate to new people, and as people have different hand expressions and gestures, it forces methods to consider more abstract information about the hand to have higher performance. In order to make it as robust as possible, we split the dataset so that people from different ethnicity are in different splits, e.g., making the person with Hispanic descent to be in a different split than the person with European descent. Specifically, we split the 10 subjects into 5 pairs, with the 2nd pair having Hispanic descent, the 3rd pair having African/East Asian descent, and the 5th pair having European descent. This increases the challenge for detection, as well as evaluates methods based on extrapolation to hands that do not appear in the dataset, which is useful in real applications.

4. Hand Detection

Detection of hands in egocentric images is challenging, due to the many variations of poses, illumination variations, motion blur, and occlusion. While standard object detectors could be used, their performance will be limited by these factors. Although there are large variations in hand appearance, the hands typically appear in consistent context. For example, hands will appear attached to arms, hands will grasp rigid objects, such as cups, and left/right hands tend to appear in certain regions of the image. Hence, learning the context around the hands can help to detect the hands. In this section, we propose deep learning architectures to jointly learn the hands and the hand context for hand detection.

4.1. YOLO detector

The baseline detector that we use to address the hand detection problem is YOLOv2 [25] as it has good performance on generic object detection and has greater speed than other detectors. Here we will call it YOLO for simplicity. YOLO [25] contains 18 convolutional layers and 5 max-pooling layers, which we denote as the feature extraction section, followed by 8 convolutional layers, which we denote as the feature classification section, and a final convolution for the regression and object detection, as seen in Figure 5.

Other approaches like Faster R-CNN [26] and SSD [18] use finer-grained features by combining different resolution feature maps. YOLO uses this idea by adding a pass through layer that brings features from an earlier layer, and concatenates the higher resolution features with the low resolution features by stacking adjacent features into different channels. For example, a $26 \times 26 \times 512$ high-resolution feature map is converted into a $13 \times 13 \times 2048$ feature map, and then concatenated with the lower-resolution feature map.

YOLO outputs 5 values for each bounding box representing the x- and y- coordinates (t_x, t_y), width and height (t_w, t_h), and confidence (t_o). We set the number of classes for detection as $C = 1$ for the hand detection problem since we are detecting both hands as one object class, and $C = 2$ for the hand disambiguation problem since we are detecting left and right hands as separate object classes.

YOLO also provides a smaller version called Tiny YOLO which contains 7 convolutional layers and 6 max-pooling layers for the feature extraction section, followed by 1 convolutional layer for the feature classification section, and a final convolution for the regression and object detection. This makes Tiny YOLO version faster but with a decrease in detection accuracy.

4.2. Detection using context information

For YOLO, the object's features and some surrounding context are used to locate the object in the image. The hands co-occur with some of its surrounding context, e.g., arms, graspable objects, etc. Hence, the surrounding context could also be used to predict where the hand should be, even if the hand is occluded or has large shape deformations.

Using this information we propose to train a neural network focusing on the context alone, which we will denote as Context. During training, we mask out (set to zero) the regions of the convolution feature maps that contain the hands (using the ground-truth bounding box), as seen in Figure 6. This forces the network to learn the context around the hand that is predictive of the hand's location, but without using any hand features. Conceptually, masking the hand regions in the feature maps is analogous to using Dropout [32], where random nodes in a layer are set to zero in order to train the DNN to be robust to cases where some discriminative features are missing. The main difference is that here we selectively zero out the hand features, so that the trained network is robust when all hand features are missing. At test time, the test image is run through the Context network normally, and no masks are applied to the feature map.

The context mask is applied after each convolution in the feature extraction section of YOLO. Using a 416×416 image input, each of the grid cells in the YOLO architecture has a receptive field of 343×343. The average hand bounding box size is 38×40, while the largest is 230×229, showing that when the hand features are masked out, the network can still see the context around the hands within the receptive field.

We also propose to combine both networks, YOLO and Context, by creating a two-stream architecture: the 1st stream is the standard YOLO, and the 2nd is Context. After training each stream separately, we fuse them by concatenating the the features at a given level into one big feature map and fine-tuning the rest of the layers. We consider three levels of fusion, as seen in Figure 7. The "Early Fusion" architecture concatenates the features after the Feature Extraction section, and then fine-tunes the remaining Feature Classification and Object Detection sections. The "Late Fusion" architecture concatenates the features after the Feature Classification section, and fine-tunes the Object Detection section. The "Concat" architecture simply concatenates the detection bounding boxes produced by the separate YOLO and Context networks.

5. Hand Detection Experiments

In this section we present baseline experiments on our dataset, which we denote as EgoDaily, using current methods for egocentric hand detection, generic object detection, as well as our approach using context. We test three scenarios, hand proposal generation, hand detection, and hand disambiguation (left and right hand detection).

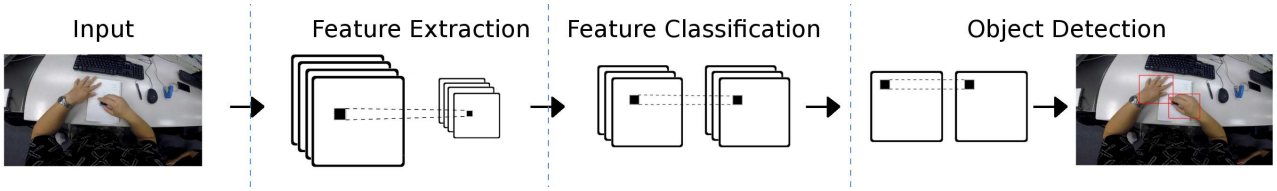


Figure 5: YOLO [25] neural network architecture sections.

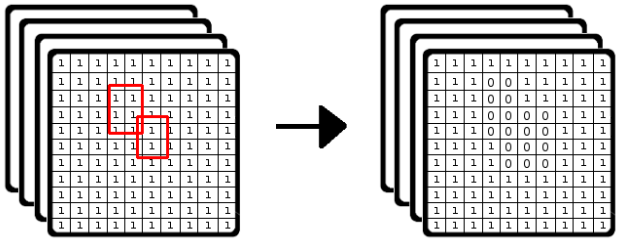


Figure 6: Output feature masking using the ground truth bounding boxes (red) by setting the feature values inside to zero.

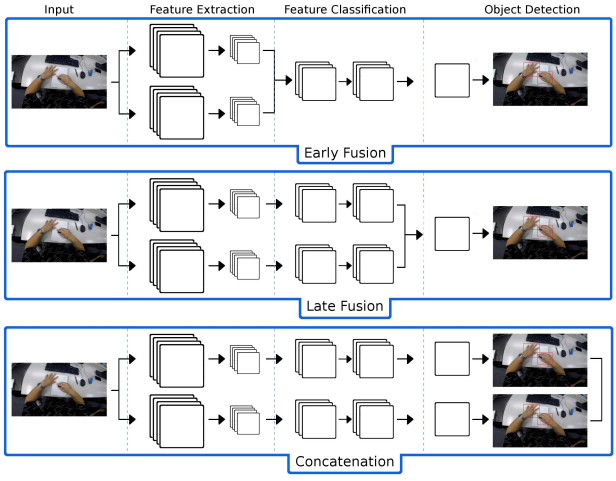


Figure 7: Proposed network architectures combining the already trained YOLO and Context streams.

5.1. Compared Methods

We compare existing object detection methods along with hand detection methods proposed from previous work. For object detection, the first works focused on generating bounding box proposals using various methods, and then use neural networks for disambiguation by classifying hands from background, and finally applying non-maximum suppression to obtain the final detection results. As neural networks perform well for classification [12], the burden falls onto generating the bounding box proposals for hand detection, especially since the egocentric perspective contains more variability than other. [2] showed that different neural networks performed almost identically using the same bounding box proposals. Hence, in this paper we present results on both hand proposal generation, hand detection and hand disambiguation.

We test different methods for hand proposal generation, and also compare their detection performance by applying the neural network proposed by [2], since it is hand specific. We consider 3 methods for generating bounding box proposals:

- **Selective Search** [33] is an unsupervised method, which segments the image using super-pixels employing multiple invariant color spaces, and then generates bounding boxes from the segmentation. We use the code provided by the authors and set the threshold $k = 50$.
- **Objectness** [1] is a supervised method, which combines color contrast, edge density and super-pixels straddling to generate bounding boxes proposals. We trained Objectness using at least one hand sample from each of person/activity in the training data, since the demo code provided by the author uses a small dataset for training. This guarantees that every combination of person/activity in the training data is taken into account.
- **Bambach, *et al.*** [2] generates proposals by considering three probability distributions. The first distribution is the hand occurrence probability, since their dataset has up to 4 hands and some appear more than others. The second distribution is the probability of the bounding box containing the hand being on a specific location (x, y) and size $(width, height)$, as in Figures 3 and 4. The third distribution is the probability of a pixel being skin color, which cannot be used in our dataset since there are no pixel-level annotations. Nonetheless, [2] showed that this third distribution was not necessary, as performance was similar without it.

Recent works have used DNNs for the proposal and classification stages, which are trained end-to-end. In order to compare the proposals results we generate the hand proposals using these methods without applying any probability threshold nor non-maximum suppression. For the detection results using these methods, we apply non-maximum suppression with overlap of 0.5. We consider 3 recent deep learning methods for hand proposal generation and detection:

- **Faster R-CNN** [26] is the combination of the Fast R-CNN [10] with the region proposal network (RPN) into a fully convolutional network trained end-to-end. We use the VGG-16 version

with default settings for the RPN. We change the maximum number of proposals from 300 to 2500 to get high recall.

- **SSD** [18] uses a deep neural network to detect objects in the image, and is trained end-to-end. It discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. We use the code provided by the author and the 300×300 resolution. We lower the base learning rate to 0.0001 for hand disambiguation as it had problems learning the hands. We keep the other settings as default.
- **YOLO** [25] takes the object detection problem as a regression problem to spatially separated bounding boxes and associated class probabilities. They use a single neural network to predict bounding boxes and class probabilities from the image, and they train the neural network end-to-end. We use the code provided by the author and 416×416 resolution, leaving the settings by default. We do not use any threshold to trim the bounding box proposals and we sort them by the probability of being a hand. We also test **Tiny YOLO** which has less convolutions, making it faster but decreasing its performance.

Finally, we consider our four proposed neural networks using context information (see Section 4.2):

- **Context** is a variation of the YOLO method that learns the hands only using context information.
- **EarlyFusion** combines the YOLO detector stream with the context stream (**Context**) after the "Feature Extraction" stage.
- **LateFusion** combines the YOLO detector stream and context stream (**Context**) after the "Feature Classification stage".
- **Concat** is the concatenation of the hand detections from the separate **YOLO** and **Context** methods.

We also replace **YOLO** with **Tiny YOLO** in the above four architectures, which are denoted as **Tiny Context**, **Tiny EarlyFusion**, **Tiny LateFusion**, and **Tiny Concat**.

5.2. Hand Proposal Results

We first test the performance of the methods on generating hand proposals on our dataset. The bounding box proposals are scored using the PASCAL VOC criteria where a proposal is considered correct if the intersection over union with a ground truth bounding box is over 0.5. We use recall as a measurement for comparison since it shows the method's ability to detect the hands regardless of the false positives, which translates to the upper limit of what an approach can achieve.

Figure 8 plots the recall versus the number of proposals, while Table 2 shows the breakdown of the recall

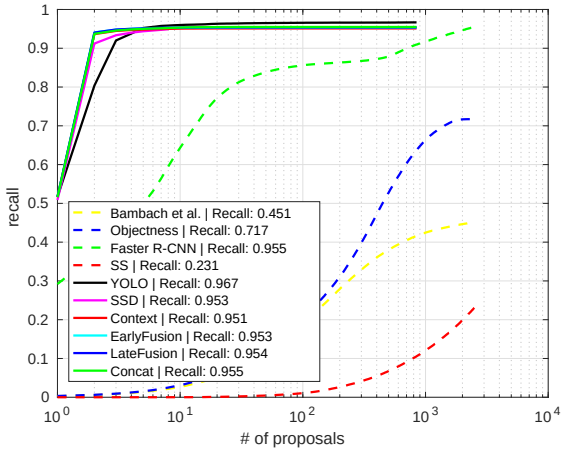


Figure 8: Method comparison using recall vs. number of proposals on the EgoDaily dataset.

results at 100 bounding box proposals for each activity. Selective Search (SS) does not perform well on all activities as this method is built on color variation – in our dataset, the arms share the same color features of the hand and there is high illumination variability, which makes them hard to segment. The method proposed by Bambach [2] performs quite differently from their dataset, which only consists of one activity. This is because of the variability in people and activities in our dataset – the different activities induce different locations and sizes of the hand, and people having different gestures from each other, which further makes the location and size more unpredictable. Hence, the location/size probability models are less specific (more spread out), and thus perform worse at generating proposals. The worst performance is on "eating", which has the highest variability in bounding box size and location of the hands due to their moving towards and away from the camera. Objectness has difficulty with hand shapes and with high illumination, with lowest performance on "biking" and "running", as the method uses color for the proposals.

Faster R-CNN shows the robustness of neural networks in this challenging task. The recall rapidly reaches 0.80 at 26 proposals, and then slowly increases from recall 0.85 for 100 proposals to its maximum level of 0.955 for 2,500 proposals. The recall is not perfect, and Faster R-CNN still has problems with some challenging scenes, such as "biking" and "running" that have camera blur, shaking, and color and illumination variations. Also, there are significant difference between the performances on "office" and on both "eating" and "kitchen", showing that the classification section of the neural network has problems with the high variability of the hand's shape. In addition, the increase in recall is gradual, which indicates that Faster R-CNN has problems with distinguishing the more difficult cases from the false positives.

Table 2
Recall results for activities and cross-validation splits (subject pairs). Recall is for 100 proposals.

Method	Activity						
	Biking	Eating	Kitchen	Office	Running	Mean	Std
SS [33]	0.016	0.136	0.103	0.057	0.049	0.072	0.047
Bambach, *et al.* [2]	0.406	0.206	0.661	0.600	0.585	0.492	0.186
Objectness [1]	0.155	0.579	0.343	0.497	0.212	0.357	0.180
Faster R-CNN [26]	0.617	0.856	0.897	0.959	0.558	0.777	0.178
SSD [18]	0.933	0.951	**0.970**	0.994	0.825	0.935	0.065
YOLO [25]	0.946	0.964	0.951	0.994	0.900	0.951	0.034
Context	0.949	0.962	0.946	0.959	0.895	0.949	0.036
EarlyFusion	**0.954**	**0.968**	0.949	0.994	0.892	0.951	0.037
LateFusion	0.949	0.964	0.952	**0.995**	0.901	0.952	0.034
Concat	0.951	0.964	0.952	**0.995**	**0.906**	**0.954**	0.031

SSD and YOLO show an improvement over other approaches as they use a single neural network to generate the proposals and classify them. Although the overall recall is similar to Faster R-CNN, both SSD and YOLO can reach their maximum recall with fewer proposals, which shows that the proposal generation is similar in all three neural network approaches but the single neural network trained end-to-end improves on the classification. SSD and YOLO have consistent performance over all activities, but they both have problems with "running" as it is the most variable with illumination and blurriness. Among all methods, the highest recall is on "kitchen", showing that the fine-grained features can find the hands even when they are occluded.

All our methods using context information (Context, EarlyFusion, LateFusion, and Concat) reach their highest recall with less proposals than SSD and YOLO. Our methods reach high recall with 20 proposals, while SSD and YOLO need 25 and 100 respectively, showing that context information can also be used for efficiency as it reduces the necessary number of proposals needed to find the hands. Interestingly, using context by itself (Context) has good performance, which demonstrates that information outside the bounding boxes is able to generate good proposals of hands, although it has problems when the arms are not consistently in the scene. Concat finds the most hands on "office" and "running", showing that separate NN streams work well when the hand gestures are not as complex. EarlyFusion performs best on "biking" and "eating", which have combination of multiple hand and arm gestures, suggesting that a combination of context and hand features are best suited for these complex actions.

Figure 9 shows a few examples of failure cases, where no method was able to detect both hands using 2,500 proposals. We visualize the false positives that have the highest intersection with the ground truth.[3] These example images have at least one ground-truth hand not found by every method, making them some of the most challenging cases. This shows the illumination changes are the most challenging scenarios for extracting information, as every method has issues regardless of what kind of features they use. After the illumination, the occlusion and hand shape follow as challenges, which makes the classification to encounter issues as even the neural networks based approaches cannot locate the hands.

[3] If there is no intersection, we take the highest score bounding box, and in the case of Selective Search we take the first bounding box proposal.

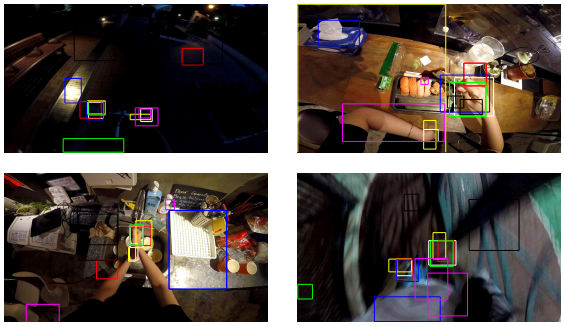


Figure 9: False positives bounding boxes on hard images generated by taking the closest false positive to the groundtruth using Selective Search (Yellow), Bambach, *et al.* (Red), Objectness (Blue), Faster R-CNN (Green), SSD (Magenta), YOLO (Black) and the ground-truth (White).

Table 3
Average Precision results for hand detection for each activity.

Method	Activity					
	Biking	Eating	Kitchen	Office	Running	All
SS [33]	0.090	0.098	0.097	0.090	0.090	0.091
Bambach [2]	0.217	0.118	0.508	0.490	0.218	0.308
Objectness [1]	0.090	0.280	0.174	0.339	0.053	0.193
Faster R-CNN [26]	0.417	0.749	0.787	0.901	0.216	0.641
SSD [18]	0.869	0.883	0.898	0.906	0.757	0.881
YOLO [25]	**0.907**	**0.876**	**0.905**	**0.907**	**0.821**	**0.894**

5.3. Hand Detection Results

We next present results on hand detection, where the left and right hands are treated as one class. This experiment will show how well a classifier can detect hands from the proposals generated in the previous section. Table 3 shows the breakdown on each activity using overall performance of the baseline methods for hand detection in terms of Average Precision (AP). The hand proposal methods that use color like Selective Search and Objectness, and bounding box features like Bambach have low recall, which acts as an upper limit for even a robust classifier like a neural network, making the overall performance low. Objectness has its best performance on "kitchen" and "office", which shows the approach cannot handle high variability in illumination and hand gestures, as both activities are indoors and have the least hand gesture variability.

The features a neural network extracts are more ro-

Table 4
Average Precision for hand detection for each activity for YOLO and the proposed context versions.

Method	Activity					
	Biking	Eating	Kitchen	Office	Running	All
YOLO [25]	**0.907**	**0.876**	**0.905**	**0.907**	0.821	0.894
Context	**0.907**	**0.876**	**0.905**	**0.907**	0.821	0.893
EarlyFusion	**0.907**	0.875	0.904	0.906	0.809	0.894
LateFusion	**0.907**	0.875	0.905	**0.907**	0.832	**0.895**
Concat	**0.907**	0.872	0.903	0.906	**0.856**	0.893
Tiny YOLO [25]	0.803	0.841	0.800	**0.907**	0.669	0.794
Tiny Context	0.800	0.823	0.787	0.905	0.604	0.790
Tiny EarlyFusion	**0.806**	0.844	**0.801**	0.906	**0.677**	**0.796**
Tiny LateFusion	0.805	**0.845**	0.800	0.906	0.673	0.795
Tiny Concat	0.804	0.835	0.797	0.906	0.661	0.793

bust than color, location and size from the previous methods, which allows a neural network to classify and detect the hands. SSD and YOLO shows the robustness of using end-to-end training of a single network for region proposals and classification, as opposed to separate networks for proposal generation and classification, as in Faster R-CNN (RPN). Faster R-CNN has its highest performance on "office", which is the least varying activity. It also has good performance on "eating" and "kitchen", showing that it is able to detect the hands even if they have high gesture variability, although it has problems with high illumination variability (e.g., "running" and "biking"). SSD has a higher accuracy when scenarios are not blurry and good illumination regardless if the hand shape is complex, but performs worse than YOLO when it encounters more challenging scenarios. This shows the fine-grained features make the neural network more robust to hand shapes, but is counter productive with blurriness and high illumination variability, which makes them both struggle with "running". There is still room for improvement on this dataset, in terms of both efficiency (reducing the number of proposals required) and effectiveness (increasing the recall upper bound and average precision).

We also compare YOLO with our proposed versions using context. Table 4 shows the breakdown of detection performance on each activity, and overall performance. For hand detection YOLO is able to detect the hands with high accuracy. However the Context architecture alone can be used for the detection even when the hand features are ignored. Context is negatively affected by the inconsistency of the arms, which are not always present in the image. This is shown on "eating" and "running", which do not contain as many visible arms as other activities. Using Context along with YOLO is able to detect hands more accurately on "running", as the arms can add information when the hands are blurry.

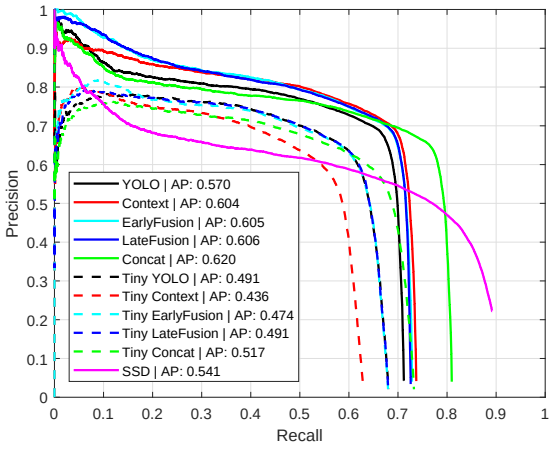


Figure 10: Results for Hand Disambiguation using YOLO and the proposed context versions on the EgoDaily dataset.

Table 5
Average Precision results of hand disambiguation for different activities.

Method	Activity					
	Biking	Eating	Kitchen	Office	Running	All
SSD [18]	0.606	0.454	0.666	0.758	**0.239**	0.541
YOLO [25]	0.689	**0.469**	0.744	0.638	0.160	0.570
Context	**0.788**	0.265	0.769	**0.867**	0.201	0.604
EarlyFusion	0.770	0.429	0.769	0.755	0.203	0.605
LateFusion	0.777	0.408	0.751	0.760	0.159	0.606
Concat	0.774	0.435	**0.810**	0.799	0.220	**0.620**
Tiny YOLO [25]	0.582	**0.319**	0.596	0.709	0.159	0.491
Tiny Context	0.612	0.127	0.586	0.718	0.197	0.436
Tiny EarlyFusion	**0.660**	0.203	**0.608**	0.730	**0.206**	0.474
Tiny LateFusion	0.651	0.240	0.606	0.709	0.164	0.491
Tiny Concat	0.637	0.260	0.592	**0.758**	0.203	**0.517**

5.4. Hand Disambiguation Results

We present results on hand disambiguation, which takes the left and right hands as separate object classes, leading to a more challenging detection problem. Figure 10 shows the overall performance of our proposed approaches and baselines methods on the disambiguation problem. The overall performance decreases significantly compared to the simpler hand detection problem in Section 5.3. For hand disambiguation, Context, EarlyFusion, LateFusion and Concat perform significantly better than the baseline YOLO, showing that context provide discriminative information on whether the hand is left or right. Among the different fusions the earlier combination of hand and context information leads to worse performance, showing that it is better to have the streams to focus on their own task and not interfere with each other.

Table 5 shows the breakdown of hand disambiguation performance on each activity and overall performance, in terms of average precision (AP). On "biking" and "office" we notice a big gap between the YOLO

Table 6
Average Precision results of Hand Disambiguation (left/right) for different activities on the EgoDaily dataset. Entries are bolded if one hand performance is 0.05 more than the other hand.

Method	Activity					
	Biking	Eating	Kitchen	Office	Running	All
SSD	0.620/0.624	0.439/0.468	0.657/**0.700**	**0.856**/0.748	0.277/0.259	0.563/0.558
YOLO	0.697/0.721	**0.541**/0.402	0.723/0.765	**0.658**/0.563	0.184/0.202	**0.601**/0.505
Context	0.790/0.783	**0.350**/0.223	0.760/0.774	0.799/**0.858**	0.177/**0.225**	0.611/0.596
EarlyFusion	0.709/**0.771**	**0.509**/0.359	0.739/**0.791**	0.711/**0.779**	0.159/**0.237**	0.614/0.592
LateFusion	0.711/**0.773**	**0.505**/0.323	0.729/**0.774**	**0.782**/0.726	0.167/0.171	0.618/0.589
Concat	0.782/0.766	**0.532**/0.377	0.795/0.828	**0.842**/0.740	0.220/0.265	0.631/0.606
Tiny YOLO	0.572/0.585	**0.366**/0.257	0.548/**0.633**	0.718/0.700	0.163/0.206	**0.516**/0.452
Tiny Context	0.620/0.603	**0.187**/0.101	0.560/**0.621**	**0.762**/0.634	0.199/0.195	**0.483**/0.401
Tiny EarlyFusion	0.659/0.653	**0.295**/0.191	0.585/**0.644**	**0.758**/0.659	0.176/**0.240**	**0.513**/0.446
Tiny LateFusion	0.651/0.652	**0.317**/0.212	0.557/**0.647**	0.733/0.693	0.167/**0.216**	**0.517**/0.448
Tiny Concat	0.643/0.635	**0.376**/0.239	0.571/**0.622**	**0.788**/0.696	0.192/0.232	**0.546**/0.472

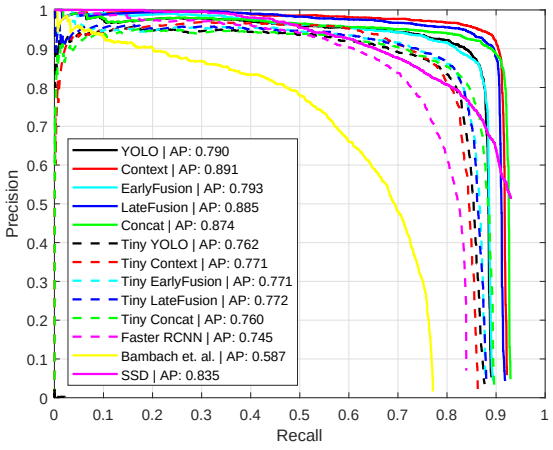


Figure 11: Results for hand disambiguation on the EgoHands dataset.

and our proposed approaches, with Context performing the best. This suggests the context (e.g., arms) can differentiate hands better than the hands themselves, since the activity is mainly grabbing the bike handle and thus the hand appearances are similar, and the arms are consistent enough. "Eating" is the only activity where YOLO performs better than our proposed context methods, with Context performing the worst – this is due to the arms not being as present in the image as in "biking" and "office". On "kitchen", context and the hand information complement each other, as this activity has the most occlusion, in which the context helps improving the performance. "Running" shows the lowest performance across all versions since there is no context information to be used.

Table 6 shows the left and right hand performance across activities using Average Precision (AP). We highlight the result where one hand had a higher performance by at least 0.05. Across different methods we see a consistent increase in performance on the "eating" and "office" activities on the left hand – people are often right-handed and the hand gestures vary more on the right hand as they interact with office tools or cutlery.

Table 7
Average Precision results of Hand Disambiguation (left/right) for both people (Other/Wearer) on the EgoHands dataset. The right column is the recall of the proposal generation after using Non-maximum suppression.

Method	Other		Wearer		All	Recall
	Left	Right	Left	Right		
Bambach *et al.* [2]	0.556	0.698	0.596	0.553	0.587	0.771
Faster [26]	0.754	0.809	0.681	0.582	0.745	0.839
SSD [18]	0.839	0.870	0.847	0.700	0.834	0.930
YOLO [25]	0.869	0.894	0.771	0.694	0.790	0.890
Context	**0.897**	**0.907**	**0.866**	0.774	**0.891**	0.922
EarlyFusion	0.871	0.897	0.817	0.706	0.792	0.894
LateFusion	0.892	0.904	0.722	0.726	0.885	0.918
Concat	0.872	0.900	0.848	**0.775**	0.873	**0.929**
Tiny YOLO	0.836	0.835	0.711	0.566	0.762	0.876
Tiny Context	0.815	0.812	**0.753**	0.601	**0.771**	0.862
Tiny EarlyFusion	0.847	0.840	0.719	**0.620**	**0.771**	0.877
Tiny LateFusion	0.853	0.835	0.722	0.600	**0.771**	0.880
Tiny Concat	**0.854**	**0.879**	0.720	0.607	0.760	**0.896**

5.5. Hand disambiguation on EgoHands

Figure 11 shows a comparison of the methods on the EgoHands [2] dataset, which focuses on two people's interaction while playing board games. In this task there are 4 classes: Wearer's left and right hands, and Other's left and right hands. Overall, the end-to-end neural networks (YOLO and SSD) are able to find more hands and more accurately than Bambach. Using context (Concat, Late Fusion, Context) significantly improves the detection of the hands in these heavily occluded scenarios, as it allows detection of the hands even when they are occluded by focusing on the arms or held objects. Similar to our proposed database, combining the hand and context stream earlier yields worst performance on the EgoHands dataset.

Faster RCNN finds less hands than YOLO but more accurately, suggesting focusing on the feature extraction can improve the overall performance. SSD [18] demonstrates the robustness of using fine-grained features, as they are able to find the most hands out of all the neural network approaches, however they have a lower accuracy as they find them.

Finally, in Table 7 we present the performance on each hand class (Other and Wearer) on the EgoHands dataset. For Bambach, their object proposal method is consistent with different hands as it is able to find them with similar performance, however their neural network has difficulties disambiguating. Faster R-CNN has difficulty detecting the Wearer's hands as they have small size and have fewer instances for training, leading to potential overfitting of the bigger neural network. SSD [18] and YOLO are able to find most hands, but have difficulty on the Wearer's right hand, since it appears with complex gestures while interacting with objects, but with few instances in the training set. The use of the context is able to detect more hands even without

combining it with the hand features. In particular, the most improvement is on the Wearer's hands, since they are usually occluded or grasping various objects.

5.6. Experiments using Tiny YOLO

We next present the hand disambiguation experiments using the Tiny YOLO architectures. Since Tiny YOLO has fewer layers, the accuracy is in general lower compared to the full YOLO.

Table 5 presents the hand disambiguation results on the EgoDaily dataset using the Tiny YOLO versions. Overall, the Tiny Concat fusion obtains the best performance among the Tiny versions. The Tiny versions are able to perform with a slight decrease to the full version on the "Office" and "Running" activities. This shows the Tiny versions are able to extract enough information from the challenging scenarios but encounter difficulty when dealing with complex hand gestures, suggesting the use of more layers on the classification section can detect more hands.

Table 6 shows the left and right hand performance using the Tiny YOLO versions on the EgoDaily dataset. Overall, the results using Tiny YOLO are consistent with the full YOLO fusion methods – performance is higher on the left hand, because the right hand has more complex gestures as it interacts with objects the most. "Kitchen" is the only activity where the left hand is often occluded by various objects while as it holds them, which has an impact on the disambiguation.

Finally, Table 7 shows the performance of the Tiny YOLO versions on the EgoHands dataset. The results of Tiny Concat show that a simpler neural network using context can perform similarly to the full YOLO (without context) in terms of recall. However, the AP for Tiny Concat is slightly lower, indicating that the classification module of this method is limited. This shows that the Tiny version is able to extract enough information from the image, but has problems with feature classification, suggesting an increase of the classification section can increase the performance for hand detection.

6. Conclusions

We have introduced an egocentric dataset with large variations in illumination, people and places, expanding the challenges from previous datasets that focus on hand detection. We also present an analysis of the dataset to show its nature and characteristics that can provide information for future work. We proposed three different joint neural network architectures to combine hand and context information to improve hand detection and disambiguation to be robust to occlusion, illumination and shape variations. We present a comparison with object/hand detection methods to benchmark our new dataset. We show that using context information can significantly improve the hand disambiguation task. We hope that our new dataset will help facilitate new research on hand detection, in particular deep learning methods. For future work we intend to expand the annotations to hand segmentation.

References

[1] Alexe, B., Deselaers, T., Ferrari, V., 2012. Measuring the objectness of image windows. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 2189–2202. doi:10.1109/TPAMI.2012.28.

[2] Bambach, S., Lee, S., Crandall, D., Yu, C., 2015. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions, in: IEEE International Conference on Computer Vision (ICCV).

[3] Betancourt, A., 2014. A sequential classifier for hand detection in the framework of egocentric vision, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, IEEE Computer Society, Washington, DC, USA. pp. 600–605. URL: http://dx.doi.org/10.1109/CVPRW.2014.92, doi:10.1109/CVPRW.2014.92.

[4] Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M., 2018. Scaling egocentric vision: The epic-kitchens dataset, in: European Conference on Computer Vision (ECCV).

[5] Fathi, A., Farhadi, A., 2011. Understanding egocentric activities, in: Proceedings of the 2011 International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA. pp. 407–414. URL: http://dx.doi.org/10.1109/ICCV.2011.6126269, doi:10.1109/ICCV.2011.6126269.

[6] Fathi, A., Hodgins, J.K., Rehg, J.M., 2012. Social interactions: A first-person perspective, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1226–1233. doi:10.1109/CVPR.2012.6247805.

[7] Fathi, A., Ren, X., Rehg, J.M., 2011. Learning to recognize objects in egocentric activities, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA. pp. 3281–3288. URL: http://dx.doi.org/10.1109/CVPR.2011.5995444, doi:10.1109/CVPR.2011.5995444.

[8] Finocchiaro, J., Khan, A.U., Borji, A., 2017. Egocentric height estimation, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1142–1150. doi:10.1109/WACV.2017.132.

[9] Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K., 2018. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[10] Girshick, R., 2015. Fast r-cnn, in: International Conference on Computer Vision (ICCV).

[11] GÃijnther, T., Franke, I.S., Groh, R., 2015. Aughanded virtuality - the hands in the virtual environment, in: 3D User Interfaces (3DUI), 2015 IEEE Symposium on, pp. 157–158. doi:10.1109/3DUI.2015.7131748.

[12] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Curran Associates Inc., USA. pp. 1097–1105. URL: http://dl.acm.org/citation.cfm?id=2999134.2999257.

[13] Lee, S., Bambach, S., Crandall, D.J., Franchak, J.M., Yu, C., 2014. This hand is my hand: A probabilistic approach to hand disambiguation in egocentric video, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 557–564. doi:10.1109/CVPRW.2014.86.

[14] Lee, Y.J., Ghosh, J., Grauman, K., 2012. Discovering

important people and objects for egocentric video summarization, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1346–1353. doi:10.1109/CVPR.2012.6247820.

[15] Li, C., Kitani, K.M., 2013a. Model recommendation with virtual probes for egocentric hand detection, in: Proceedings of the 2013 IEEE International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA. pp. 2624–2631. URL: http://dx.doi.org/10.1109/ICCV.2013.326, doi:10.1109/ICCV.2013.326.

[16] Li, C., Kitani, K.M., 2013b. Pixel-level hand detection in ego-centric videos, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 3570–3577. doi:10.1109/CVPR.2013.458.

[17] Li, Y., Ye, Z., Rehg, J.M., 2015. Delving into egocentric actions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 287–295. doi:10.1109/CVPR.2015.7298625.

[18] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. SSD: Single shot multibox detector, in: ECCV.

[19] Lu, Z., Grauman, K., 2013. Story-driven summarization for egocentric video, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA. pp. 2714–2721. URL: http://dx.doi.org/10.1109/CVPR.2013.350, doi:10.1109/CVPR.2013.350.

[20] Min, W., Li, X., Tan, C., Mandal, B., Li, L., Lim, J.H., 2014. Efficient retrieval from large-scale egocentric visual data using a sparse graph representation, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, IEEE Computer Society, Washington, DC, USA. pp. 541–548. URL: http://dx.doi.org/10.1109/CVPRW.2014.84, doi:10.1109/CVPRW.2014.84.

[21] Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C., 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[22] Pirsiavash, H., Ramanan, D., 2012a. Detecting activities of daily living in first-person camera views, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2847–2854. doi:10.1109/CVPR.2012.6248010.

[23] Pirsiavash, H., Ramanan, D., 2012b. Detecting activities of daily living in first-person camera views, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE.

[24] Poleg, Y., Arora, C., Peleg, S., 2014. Temporal segmentation of egocentric videos, in: CVPR.

[25] Redmon, J., Farhadi, A., 2016. Yolo9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242 .

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

[27] Ren, X., Gu, C., 2010. Figure-ground segmentation improves handled object recognition in egocentric video, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 3137–3144.

[28] Ryoo, M.S., Matthies, L., 2013. First-person activity recognition: What are they doing to me?, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 2730–2737. doi:10.1109/CVPR.2013.352.

[29] Ryoo, M.S., Matthies, L., 2016. First-person activity recognition: Feature, temporal structure, and prediction. International Journal of Computer Vision 119, 307–328. URL: https://doi.org/10.1007/s11263-015-0847-4, doi:10.1007/s11263-015-0847-4.

[30] Spriggs, E.H., De la Torre Frade, F., Hebert , M., 2009. Temporal segmentation and activity classification from first-person sensing, in: IEEE Workshop on Egocentric Vision, CVPR 2009.

[31] Spurr, A., Song, J., Park, S., Hilliges, O., 2018. Cross-modal deep variational hand pose estimation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[32] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[33] Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M., 2013. Selective search for object recognition. International Journal of Computer Vision 104, 154–171. URL: https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013.

[34] Urooj, A., Borji, A., 2018. Analysis of hand segmentation in the wild, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Mr. Sergio Cruz is a postgraduate student at the City University of Hong Kong. He recieved the M.Sc. and B.Sc. in Computer Science from Benemerita Universidad Autonoma de Puebla in 2014 and 2012. From 2010 to 2011 he was an exchange student at Bishop's University, Sherbrooke, Canada.

Dr. Antoni Chan is an associate professor at the City University of Hong Kong in the Department of Computer Science. Before joining CityU, he was a postdoctoral researcher in the Department of Electrical and Computer Engineering at the University of California, San Diego (UC San Diego). He received the Ph.D. degree from UC San Diego in 2008 studying in the Statistical and Visual Computing Lab (SVCL). He received the B.Sc. and M.Eng. in Electrical Engineering from Cornell University in 2000 and 2001. From 2001 to 2003, he was a Visiting Scientist in the Computer Vision and Image Analysis lab at Cornell. In 2005, he was a summer intern at Google in New York City. In 2012, he was the recipient of an Early Career Award from the Research Grants Council of the Hong Kong SAR, China.