Contextual Attention for Hand Detection in the Wild

Supreeth Narasimhaswamy[†,1], Zhengwei Wei[†,1], Yang Wang[1], Justin Zhang[2], Minh Hoai[1,3]
[†]Joint First Authors, [1]Stony Brook University, [2]Caltech, [3]VinAI Research

Abstract

We present Hand-CNN, a novel convolutional network architecture for detecting hand masks and predicting hand orientations in unconstrained images. Hand-CNN extends MaskRCNN with a novel attention mechanism to incorporate contextual cues in the detection process. This attention mechanism can be implemented as an efficient network module that captures non-local dependencies between features. This network module can be inserted at different stages of an object detection network, and the entire detector can be trained end-to-end.

We also introduce large-scale annotated hand datasets containing hands in unconstrained images for training and evaluation. We show that Hand-CNN outperforms existing methods on the newly collected datasets and the publicly available PASCAL VOC human layout dataset. Data and code: https://www3.cs.stonybrook.edu/~cvl/projects/hand_det_attention/.

1. Introduction

People use hands to interact with each other and the environment, and most human actions and gestures can be determined by the location and motion of their hands. As such, being able to detect hands reliably in images and videos will facilitate many visual analysis tasks, including gesture and action recognition. Unfortunately, it is difficult to detect hands in unconstrained conditions due to tremendous variation of hands in images. Hands are highly articulated, appearing in various orientations, shapes, and sizes. Occlusion and motion blur further increase variations in the appearance of hands.

Hands can be considered as a generic object class, and an appearance-based object detection framework such as DPM [9] and MaskRCNN [12] can be used to train a hand detector. However, an appearance-based detector would have difficulties in detecting hands with occlusion and motion blur. Another approach for detecting hands is to consider them as a part of a human body and determine the locations of the hands based on the detected human pose. Pose detection, however, does not provide a reliable solution by itself, especially when several human body parts are not visible in the image (e.g., in TV shows, the lower body is frequently not contained in the image frame).

Figure 1: **Hand detection in the wild**. We propose Hand-CNN, a novel network for detecting hand masks and estimating hand orientations in unconstrained conditions.

In this paper, we propose Hand-CNN, a novel CNN architecture to detect hand masks and predict hand orientations. Hand-CNN is founded on MaskRCNN [12], with a novel attention module to incorporate contextual cues during the detection process. The proposed attention module is designed for two types of non-local contextual pooling: one based on feature similarity and the other based on spatial relationship between semantically related entities. Intuitively, a region is more likely to be a hand if there are other regions with similar skin tones, and the location of a hand can be inferred by the presence of other semantically related body parts such as wrist and elbow. The contextual attention module encapsulates these two types of non-local contextual pooling operations. These operations can be performed efficiently with a few matrix multiplications and additions, and the parameters of the attention module can be learned together with other parameters of the detector end-to-end. The attention module as a whole can be inserted in already existing detection networks. This illustrates the generality and flexibility of the proposed attention module.

Finally, we address the lack of training data by collecting and annotating two large-scale hand datasets. Since annotating many images is a laborious process, we develop a method to semi-automatically annotate most of the data and we only manually annotate a portion of the data. Al-

together, the newly collected data contains more than 35K images with around 54K annotated hands. This data can be used for developing and evaluating hand detectors.

2. Related Work

There exist a number of algorithms for hand detection. Early works mostly used skin color to detect hands [5, 34, 35], or boosted classifiers based on shape features [19, 25]. Later on, context information from human pictorial structures was also used for hand detection [3, 18, 20]. Mittal *et al.* [24] proposed to combine shape, skin, and context cues to build a multi-stage detector. Saliency maps have also been used for hand detection [26]. However, the performance of these methods on unconstrained images is poor, possibly due to the lack of access to deep learning and powerful feature representation.

Recent works are based on CNN's. Le *et al.* [15] proposed a multi-scale FasterRCNN method to avoid missing small hands. Roy *et al.* [28] proposed to combine FasterRCNN and skin segmentation. Duan *et al.* [7] proposed a framework based on pictorial structure models to detect and localize hand joints from depth images. Deng *et al.* [6] proposed a CNN-based method to detect hands and estimate the orientations jointly. However, the performance of these methods is still poor, possibly due to the lack of training data and a mechanism for resolving ambiguity. We introduce here large datasets and propose a novel method to combine an appearance-based detector and an attention method to capture non-local context to resolve ambiguity.

The contextual attention module for hand detection developed in this paper shares some similarities with some recently proposed attention mechanisms, such as Non-local Neural Networks [32], Double Attention Networks [4], and Squeeze-and-Excitation Networks [16]. These attention mechanisms, however, are designed for image and video classification instead of object detection. They do not consider spatial locality, but locality is essential for object detection. Furthermore, most of them are defined based on similarity instead of semantics, ignoring the contextual cues obtained by reasoning about spatial relationship between semantically related entities.

3. Hand-CNN

Hand-CNN is developed from MaskRCNN [12], with an extension to predict the hand orientation, as depicted in Fig. 2a. Hand-CNN also incorporates a novel attention mechanism to capture the non-local contextual dependencies between hands and other body parts.

3.1. Hand Mask and Orientation Prediction

Our detection network is founded on MaskRCNN [12]. MaskRCNN is a robust state-of-the-art object detection framework with multiple stages and branches. It has a Region Proposal Network (RPN) branch to identify the candidate object bounding boxes, a Box Regression Network (BRN) branch to pull features inside each proposal region for classification and bounding box regression, and a branch for predicting the binary segmentation of the detected object. The binary mask is better than the bounding box at delineating the boundary of the object, but neither the mask or the bounding box encodes the orientation of the object.

We extend MaskRCNN to include an additional network branch to predict hand orientation. Here, we define the orientation of the hand as the angle between the horizontal axis and the vector connecting the wrist and the center of the hand mask (see Fig. 2b). The orientation branch shares weights with the other branches, so it does not incur significant computational expenses. Moreover, the shared weights slightly improve the performance in our experiments.

The entire hand detection network with mask detection and orientation prediction can be jointly optimized by minimizing the combined loss function $L = L_{RPN} + L_{BRN} + L_{mask} + \lambda L_{ori}$. Here, $L_{RPN}, L_{BRN}, L_{mask}$ are the loss functions for the region proposal network, the bounding box regression network, and the mask prediction network, as described in [12, 27]. In our experiments, we use the default weights for these loss terms, as specified in [12]. L_{ori} is the loss for the orientation branch, defined as:

$$L_{ori}(\theta, \theta^*) = |arctan2(sin(\theta - \theta^*), cos(\theta - \theta^*))|, \quad (1)$$

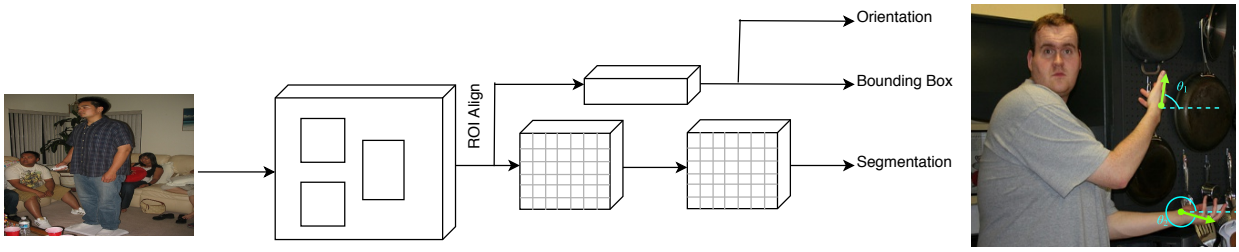
where θ and θ^* are the predicted and ground truth hand orientations (the angle between the x-axis and the vector connecting the wrist and the center of the hand, see Fig. 2b). We use the above loss function instead of the simple absolute difference between θ and θ^* to avoid the modular arithmetic problem of the angle space (i.e., 359° is close to 1° in the angle space, but the absolute difference is big). Weight λ is a tunable parameter for the orientation loss, which was set to 0.1 in our experiments.

3.2. Contextual Attention Module

Hand-CNN has a novel attention mechanism to incorporate contextual cues for detection. Consider a three dimensional feature map $\mathbf{X} \in \mathbb{R}^{h \times w \times m}$, where h, w, m are the height, width, and the number of channels. For a spatial location i of the feature map $\mathbf{X}$, we will use $\mathbf{x}_i$ to denote the m dimensional feature vector at that location. Our attention module computes a contextual feature map $\mathbf{Y} \in \mathbb{R}^{h \times w \times m}$ of the same size as $\mathbf{X}$. The contextual feature vector $\mathbf{y}_i$ for location i is computed as:

$$\mathbf{y}_i = \sum_{j=1}^{hw} \left[\frac{f(\mathbf{x}_i, \mathbf{x}_j)}{C(\mathbf{x}_i)} + \sum_{k=1}^{K} \alpha_k \; p_k(\mathbf{x}_j) \; h_k(d_{ij}) \right] g(\mathbf{x}_j).$$

This contextual vector is the sum of contextual information from all locations j's of the feature map. The contextual



(a) (b)

Figure 2: **Processing pipeline of Hand-CNN, and Hand Orientation illustration.** (a): An input image is fed into a network for bounding box detection, segmentation, and orientation estimation. The Hand-CNN extends the MaskRCNN to predict the orientation of hand by adding an additional network branch. The Hand-CNN also has a novel attention mechanism. This attention mechanism is implemented as a modular block and is inserted before the RoIAlign layer. (b): The green arrows denote vectors connecting the wrist and the center of the hand. The cyan dotted lines are parallel to x-axis, θ_1 and θ_2 denote orientation angles for the right hand and left hand of the person, respectively.

contribution from location j toward location i is determined by several factors as explained below.

Similarity Context. One type of contextual pooling is based on non-local similarity. In the above formula, $f(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}$ is a measure for the similarity between feature vectors $\mathbf{x}_j$ and $\mathbf{x}_i$. $C(\mathbf{x}_i)$ is the normalizing factor: $C(\mathbf{x}_i) = \sum_j f(\mathbf{x}_i, \mathbf{x}_j)$. Thus $\mathbf{x}_j$ provides more contextual support to $\mathbf{x}_i$ if $\mathbf{x}_j$ is more similar to $\mathbf{x}_i$. Intuitively, a region is more likely to be a hand if there are other regions with similar skin tone, and a region is less likely to be a hand if there are non-hand areas with similar texture. Similarity pooling can provide contextual information to increase or decrease the probability that a region is a hand.

Semantics Context. Similarity pooling, however, does not take into account semantics and spatial relationship between semantically related entities [13]. The second type of contextual pooling is based on the intuition that the location of a hand can be inferred by the presence and locations of other body parts such as wrist and elbow. We consider having K (body) part detectors, and $p_k(\mathbf{x}_j)$ denotes the probability that $\mathbf{x}_j$ belongs to part category k (for $1 \leq k \leq K$). The variable d_{ij} denotes the L_2 distance between positions i and j, and $h_k(d_{ij})$ encodes the probability that the distance between a hand and a body part of category k is d_{ij}. We model this probability using a Gaussian distribution with mean μ_k and variance σ_k^2. Specifically, we set: $h_k(d_{ij}) = \exp\left(-\frac{(d_{ij}-\mu_k)^2}{\sigma_k^2}\right)$. Some part categories provide more informative contextual cues for hand detections than other categories, so we use the scalar variable α_k ($0 \leq \alpha_k \leq 1/K$) to indicate the contextual importance of category k. The variables α_k's, μ_k's, and σ_k's are automatically learned.

The functions f, g, and p_k's are also learnable. We parameterize them as follows.

$$f(\mathbf{x}_i, \mathbf{x}_j) = \exp\left((\mathbf{W}_\theta \mathbf{x}_i)^T (\mathbf{W}_\phi \mathbf{x}_j)\right), \quad (2)$$

$$g(\mathbf{x}_j) = \mathbf{W}_g \mathbf{x}_j, \quad p(\mathbf{x}_j) = \text{softmax}(\mathbf{W}_p \mathbf{x}_j), \quad (3)$$

where $\mathbf{W}_\theta, \mathbf{W}_\phi, \mathbf{W}_g \in \mathbb{R}^{m \times m}$ and $\mathbf{W}_p \in \mathbb{R}^{K \times m}$. We set $p_k(\mathbf{x}_j)$ as k^{th} element of $p(\mathbf{x}_j)$. The above matrix operations involving $\mathbf{W}_\theta$, $\mathbf{W}_\phi$, $\mathbf{W}_g$, and $\mathbf{W}_p$ can be implemented efficiently using 1×1 convolutions. Together with μ_k's, σ_k's, and α_k's, these matrices are the learnable parameters of our attention module.

Comparison to non-local neural network [32]. The similarity context term $f(\mathbf{x}_i, \mathbf{x}_j)$ was first introduced by Wang *et al*. [32], but their work by itself is more suited for classification than detection tasks. The proposed attention module has an additional term $\alpha_k p_k(\mathbf{x})_j h_k(d_{ij})$ to capture semantically related entities and their spatial relationships. The proposed attention method, in addition to pooling similar features, provides information about other semantically related entities such as body parts as well as their locations. This is densely performed at every spatial location of an image, and is therefore suited for detection and localization tasks. For example, the proposed contextual attention can help distinguish between body parts with similar skin tones, while [32] may not.

4. Datasets

We aim to train a hand detector that can detect all occurrences of hands in images, regardless of their shapes, sizes, orientations, and skin tones. Unfortunately, there was no existing training dataset that was large and diverse enough for this purpose, so we collected and annotated some data ourselves. The data consists of two parts. Part I contains image frames that were extracted from video clips of the ActionThread dataset [14]. Part II is a subset of the Microsoft COCO dataset [22]. Images from Part I were manually annotated by us, while the annotations for Part II were automatically derived based on a hand pose detection algorithm

and the existing wrist annotations of the COCO dataset. We refer to Part I as the TV-Hand dataset and Part II as the COCO-Hand dataset.

4.1. TV-Hand Data

Data source. The TV-Hand dataset contains 9498 image frames extracted from the ActionThread dataset [14]. The ActionThread dataset consists of video clips for human actions from various TV series. We chose ActionThread as the data source because of several reasons. Firstly, we want images with multiple hand occurrences, as is likely with video frames from human action samples. Secondly, TV series are filmed from multiple camera perspectives, allowing for hands in various orientations, shapes, sizes, and relative scales (i.e., hand size compared to the size of other body parts such as the face and arm). Thirdly, we are interested in detecting hands with motion blur, and video frames contain better training examples than static photographs in this regard. Fourthly, hands are not usually the main focus of attention in TV series, so they appear naturally with various levels of occlusion and truncation (in comparison to other types of videos such as sign language or egocentric videos). Lastly, a video-frame hand dataset will complement COCO and other datasets that were compiled from static photographs.

Video frame extraction. Video frames were extracted from videos of the ActionThread dataset [14]. This dataset contains a total of 4757 videos. Of these videos, 1521 and 1514 are training and test data respectively for the task of action recognition; the remaining videos are ignored. For the TV-Hand dataset, we extracted frames from all videos. Given a video from the ActionThread dataset, we first divided it into multiple shots using a shot boundary detector. Among the video shots that were longer than one second, we randomly sampled one or two shots. For each selected shot, the middle frame of the shot was extracted and subsequently included in the TV-Hand dataset. Thus, the TV-Hand dataset includes one to two frames from each video.

We divided the TV-Hand dataset into train, validation, and test subsets. To minimize the dependency between the data subsets, we ensured that images from a given video belonged to the same subset. The training data contains images from 2433 videos, the validation data from 810 videos, and the test set from 1514 videos. All test images are extracted from the test videos of the ActionThread dataset. This is to ensure that the train and test data come from disjoint TV series, furthering the independence between these two subsets. Altogether, the TV-Hand dataset contains 9498 images. Of these images, 4853 are used as training data, 1618 as validation data, and 3027 as test data.

Notably, all videos from the ActionThread dataset are normalized to have a height of 360 pixels and a frame rate of 25fps. As a result, the images in TV-Hand dataset all have a height of 360 pixels. The widths of the images vary to keep their original aspect ratios.

Annotation collection. This dataset was annotated by three annotators. Two were asked to label two different parts of the dataset, and the third annotator was asked to verify and correct any annotation mistake. The annotators were instructed to localize every hand that occupies more than 100 pixels. We used the threshold of 100 pixels so that the dataset would be consistent with the Oxford Hand dataset [24]. Because it is difficult to visually determine if a hand region is larger than 100 pixels in practice, this served as an approximate guideline: our dataset contains several hands that are smaller than 100 pixels. Truncation, occlusion, self-occlusion were not taken into account; the annotators were asked to identify truncated and occluded hands as long as the visible hand areas were more than 100 pixels. To identify the hands, the annotators were asked to draw a quadrilateral box for each hand, aiming for a tight box that contained as many hand pixels as possible. This was not a precise instruction and led to subjective decisions in many cases. However, there was no better alternative. One option is to provide a pixel-level mask, but this would require enormous amount of human effort. Another option is to annotate the axis-parallel bounding box for the hand area. But this type of annotation provides poor localization for hands due to their extremely articulate nature. In the end, we found that a quadrilateral box had the highest annotation quality given the annotation effort. In addition to the hand bounding box, we also asked the annotators to identify the side of the quadrilateral that corresponds to the direction of the wrist/arm. Fig. 3 shows some examples of annotated hands and unannotated hands in the TV-Hand dataset.

The total number of annotated hands in the dataset is 8646. The number of hands in train, validation, and test sets are 4085, 1362, and 3199, respectively. Half of the data contains no hands, and a large proportion contains one or two hands. The largest number of hands in one image is 9. Roughly fifty percent of the hands occupy an area of 1000 square pixels or fewer. 1000 pixels corresponds to a 33×33 square, and it is relatively small compared to the image size (recall that all images have the height of 360 pixels).

4.2. COCO-Hand Data

In addition to TV-Hand, we propose to use images from the Microsoft's COCO dataset [22]. COCO is a dataset that contains common objects with various types of annotations including segmentations and keypoints. Most useful for us are the many images that contain people along with annotated joint locations. However, the COCO dataset does not contain bounding box or segmentation annotations for hands, so we propose an automatic method to infer them for a subset of the images where we can confidently do so.

Our objective here is to automatically generate non-axis

Figure 3: **Some sample images with annotated and unannotated hands from the TV-Hand dataset.** Annotators were asked to draw a quadrilateral for any visible hand region that is larger than 100 pixels, regardless of the amount of truncation and occlusion. Annotators also identified the side of the quadrilateral that connects to the arm (yellow sides in this figure). This is a challenging dataset where hands appear at multiple locations, having different shapes, sizes, and orientations. Severely occluded and blurry hands are also present. The blue boxes are some instances that were not annotated.

aligned rectangles for hands in the COCO dataset so that they can subsequently be used as annotated examples to train a hand detection network. This process requires running a hand *keypoint* detection algorithm (to detect wrist and finger joints) and uses a conservative heuristic to determine if the detection is reliable. Specifically, we used the hand keypoint detection algorithm of [30], which was trained on a multiview dataset of hands and annotated finger joints. This algorithm worked well for many cases, but it also produced many bad detections. We used the following heuristics to determine the validity of a detection as follows (see also Fig. 4).

1. Identify the predicted wrist location, called $\mathbf{w}_{pred}$
2. Calculate the average of the predicted hand keypoints, called $\mathbf{h}_{avg}$.
3. Considering $\mathbf{h}_{avg} - \mathbf{w}_{pred}$ as the direction of the hand, determine the minimum bounding rectangle that is aligned with this direction and contains the predicted wrist and all hand keypoints.
4. Calculate length L of the rectangle side that is parallel to the hand direction.
5. Compute the error between the predicted wrist location $\mathbf{w}_{pred}$ and the *closest* annotated wrist location $\mathbf{w}_{gt}$, $E = ||\mathbf{w}_{pred} - \mathbf{w}_{gt}||_2$.
6. Discard a detected hand if the error (relative to the size of the hand) is greater than 0.2 (chosen empirically), i.e., discard a detection if $E/L > 0.2$.

The COCO dataset also has annotations for the visibility of hands, and we used them to discard occluded hands. We ran the detection algorithm on 82,783 COCO images and detected 161,815 hands. The average area of the bounding rectangles are 977 pixels. Of these detections, our conservative heuristics determined 113,727 detections unreliable. A total of 48,008 detections survived to the next step.

The above heuristics can reject false positives, but it cannot retrieve missed detections (false negatives). Unfortunately, using images with missed detections can have an adverse effect on the training of the hand detector because a hand area might be deemed as a negative training example. Meanwhile, hand annotation is precious, so an image with at least one true positive detection should not be discarded. We therefore propose to keep images with true positives, but mask out the undetected hands using the following heuristics (see also Figure 5).

1. For each undetected hand, we add a circular mask of radius $r = ||\mathbf{w}_{gt} - \mathbf{e}_{gt}||_2$ centered at $\mathbf{w}_{gt}$, where $\mathbf{w}_{gt}$ and $\mathbf{e}_{gt}$ denote the wrist and elbow keypoint locations, respectively, as provided by the COCO dataset. We set the pixel intensities inside the masks to 0.
2. Discard an image if there is any overlap between any mask and any correctly detected hands (true positives).

Applying the above procedures and heuristics, we obtained the COCO-Hand dataset that has 26,499 images with a total of 45,671 hands. Additionally, we perform a final verification step to identify images with good and complete annotations. This subset has 4534 images with a total of 10,845 hands, and we refer to it as COCO-Hand-S. The bigger COCO dataset is referred to as COCO-Hand.

4.3. Comparison with other datasets

There exist a number of hand datasets, but most existing datasets were collected in the lab environments, captured by a specific type of cameras, or developed for specific scenarios, as shown in Table 1. We are, however, interested in developing a hand detection algorithm for unconstrained images and environments. To this end, only the Oxford Hand dataset is similar to ours. This dataset, however, is much smaller than the datasets being collected here.

5. Experiments

In this section we describe experiments on hand detection and orientation prediction. We evaluate the performance of Hand-CNN on test sets of the TV-Hand dataset and the Oxford Hand dataset. We do not evaluate the performance on the COCO-Hand dataset due to the absence of manual annotations. For a better cross-dataset evaluation, we do not train or fine-tune our detectors on the train data of the Oxford-Hand dataset. We only use the test data for

(a) (b)

Figure 4: **Heuristics for discarding bad detection on COCO.** (a): the hand keypoint algorithm is run to detect hands. The left hand of the man on the left is shown in (b). (b): black dot: predicted wrist $\mathbf{w}_{pred}$; cyan dot: closest annotated wrist $\mathbf{w}_{gt}$; yellow dots: predicted keypoints; green dot: center of the predicted keypoints $\mathbf{h}_{avg}$; blue-magenta box: smallest bounding rectangle for the hand keypoints; magenta side is the side of the rectangle that is parallel to the predicted hand direction, its length is L. We consider a detection unreliable if the distance between the predicted wrist and the closest annotated wrist is more than 20% of L.

(a) (b)

Figure 5: **Heuristics for masking missed detections on COCO.** (a): the hand keypoint algorithm failed to detect the left hand of the man. (b): A black circular mask centered at the wrist is added. The radius is determined based on the distance between the wrist and the elbow keypoints.

evaluation. The Oxford-Hand test data contains 821 images with a total of 2031 hands.

5.1. Details about the training procedure

We trained Hand-CNN and MaskRCNN starting from the GitHub code of Abdulla [1]. To train a MaskRCNN detector, we initialized it with a publicly available ResNet101-based MaskRCNN model trained on Microsoft COCO data. This was also the initialization method for MaskRCNN component of Hand-CNN. The contextual attention module was inserted right before the last residual block in stage 4 of ResNet101, and the weights were initialized with the Xavier-normal initializer.

5.2. Hand Detection Performance

Comparison to state-of-the-art. We used the TV-Hand dataset and COCO-Hand to train a Hand-CNN. Table 2 compares the performance of Hand-CNN with the previous state-of-the-art methods on the test set of publicly available Oxford-Hand data. We measure performance using Average Precision (AP), which is an accepted standard for object detection [8]. To be compatible with the previously published results, we use the exact evaluation protocol and evaluate the performance based on the intersection over the union of the axis-aligned predicted and annotated bounding boxes. As can be seen, Hand-CNN outperforms the best previous method by a wide margin of 10% in absolute scale. This impressive result can be attributed to: 1) the novel contextual attention mechanism, and 2) the use of a large-scale training dataset. Next we will perform ablation studies to analyze the benefits of these two factors.

Name	Scope	# images	Label
EgoHands [2]	Google glasses	4,800	Manual
Handseg [23]	Color gloves	210,000	Auto
NYUHands [31]	Three subjects	6,736	Auto
WorkingHands [29]	Three subjects	7,905	Man.+Syn.
ColorHandPose [36]	Specific poses	43,986	Synthetic
HandNet [33]	Ten subjects	212,928	Auto
GTEA [21]	Four subjects	663	Manual
Oxford-Hand [24]	Unconstrained	2686	Manual
TV-Hand	Unconstrained	9498	Manual
COCO-Hand-S	Unconstrained	4534	Semiauto
COCO-Hand	Unconstrained	26499	Semiauto

Table 1: **Comparison with other hand datasets.**

Method	AP
DPM [11]	36.8%
ST-CNN [17]	40.6%
RCNN [10]	42.3%
Context + Skin [24]	48.2%
RCNN + Skin [28]	49.5%
FasterRCNN [27]	55.7%
Rotation Network [6]	58.1%
Hand Keypoint [30]	68.6%
Hand-CNN (proposed)	**78.8**%

Table 2: **Comparison of the state-of-the-art** hand detection algorithms on the Oxford-Hand dataset.

Comparison to a heuristic based on 2D body pose. Given the success of 2D body pose keypoint estimation methods, one might wonder if we can detect hands by simply extending the direction from elbow to wrist, and guessing the extended vector from the wrist as the hand part. To compare with this heuristic baseline, we used [30] to obtain keypoints for elbows and wrists, and extend the vector from the elbow to the wrist to find the center of the hand. Suppose the distance between the elbow and the wrist is R, we set the extended distance to αR, with α being a controllable

parameter. The spatial extension of the hand is heuristically defined as a circular region with radius αR. Table 3 reports the APs of this method on Oxford data for various values of α, which are much lower than the AP of the Hand-CNN.

α	0.05	0.1	0.2	0.4	0.8	1.2	1.6
AP	28.27%	30.41%	33.56%	33.91%	24.22%	14.18%	9.29%

Table 3: **AP of the heuristic baseline**. The table reports the results on Oxford data as a function of the parameter α.

Benefits of contextual attention. Table 4 compares the performance of Hand-CNN with its own variants. All models were trained using the train set of the TV-Hand data and the COCO-Hand-S data. We did not use the full COCO-Hand dataset for training here, because we wanted to rule out the possible interference of the black circular masks in our analysis about the benefits of non-local contextual pooling.

On the Oxford-Hand test set, Hand-CNN significantly outperforms MaskRCNN, and this clearly indicates the benefits of the contextual attention module. MaskRCNN is essentially Hand-CNN without a contextual attention module. We also train a Hand-CNN detector without the semantics context component and another detector without the similarity context component. As can be seen from Table 4, both types of contextual cues are useful for hand detection.

The benefit of the contextual module is not as clear on the TV-Hand dataset. This is possibly due to images from TV series containing only the closeup upper bodies of the characters, and hands can appear out of proportion with the other body parts. Thus contextual information is less meaningful on this dataset. For reference, the Hand Keypoint method [30] also performs poorly on this dataset (38.9% AP); this method also relies on context information heavily.

Benefits of additional training data. One contribution of our paper is the collection of a large-scale hand dataset. Undoubtedly, the availability of this large-scale dataset is one reason for the impressive performance of our hand detector. Table 5 further analyzes the benefits of using more and more data. We train MaskRCNN using three datasets: TV Hand, COCO-Hand-S, COCO-Hand. The TV-Hand dataset has 4853 training images, the COCO-Hand-S has 4534 images, whereas COCO-Hand has 26,499 images.

A detector trained with the training set of TV-Hand data already performs well, including on the cross-data: Oxford-Hand dataset. This proves the generalization ability of our hand detector and the usefulness of the collected data. Table 5 also suggests the importance of having extra training data from Microsoft COCO. We see that using COCO-Hand data instead of COCO-Hand-S improves AP by 6.8% the Oxford-Hand and 3.6% on the challenging TV-Hand data. As explained in Section 4.2, COCO-Hand-S data was obtained from the COCO-Hand data by discarding images with even one unannotated hand without caring about the good hand annotations the image possibly contains.

Method	Oxford-Hand	TV-Hand
MaskRCNN	69.9%	59.9%
Hand-CNN	73.0%	60.3%
Hand-CNN w/o semantic context	71.4%	59.4%
Hand-CNN w/o similarity context	70.8%	59.6%

Table 4: **The benefits of context for hand detection.** The performance metric is AP. All models were trained using the train set of the TV-Hand and COCO-Hand-S. MaskRCNN is essentially Hand-CNN without using any type of context. It performs worse than Hand-CNN and other variants.

	Test Data	
Train Data	Oxford-Hand	TV-Hand
TV-Hand	62.5%	55.4%
TV-Hand + COCO-Hand-S	69.9%	59.9%
TV-Hand + COCO-Hand	76.7%	63.5%

Table 5: **Benefits of data.** This shows the performance of MaskRCNN trained with different amount of training data.

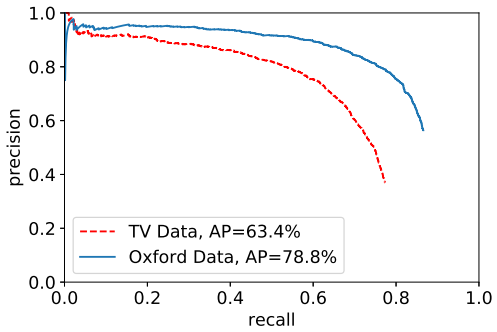


Figure 6: **Precision-recall curves of Hand-CNN**, trained on TV-Hand + COCO-Hand, tested on test sets of the Oxford-Hand and the TV-Hand data.

Whereas in COCO-Hand, we preserved images with good annotations by masking unannotated hands. The results of the experiments clearly show the benefits of doing so.

Precision-Recall curves. Fig. 6 plots precision-recall curves of the Hand-CNN on the test sets of the Oxford-Hand and TV-Hand datasets. The Hand-CNN was trained on the train sets of the TV-Hand and COCO-Hand datasets. The Hand-CNN has high precision values. For example, at 0.75 recall, the precision of Hand-CNN is 0.81.

5.3. Orientation Performance of the Hand-CNN

Tab. 6 shows the accuracy values of the predicted hand orientations of the Hand-CNN. We measure the difference in angle between the predicted orientation and the annotated orientation. We consider three different error thresholds of 10, 20, and 30 degrees, and we calculate the percentage of predictions within the error thresholds. As can be seen, the prediction accuracy is over $\sim$75% for the error threshold of 30°. Note that we only consider the performance of the

	Prediction error in angle		
Test Data	$\leq 10°$	$\leq 20°$	$\leq 30°$
Oxford-Hand	41.26%	64.49%	75.97%
TV-Hand	37.65%	60.09%	73.50%

Table 6: **Accuracy of hand orientation prediction** of the Hand-CNN on test sets of the Oxford-Hand and TV-Hand data. This table shows the percentage of correct orientation predictions for the three error thresholds of 10, 20, and 30°. The error is calculated as the angle difference between the predicted orientation and the annotated orientation. We only consider the performance of the orientation prediction for hands which have the intersection over the union greater than 0.5 with the corresponding ground truth.

Figure 7: **Some detection results of Hand-CNN**. Hands with various shapes, sizes, and orientations are detected.

orientation prediction for correctly detected hands.

5.4. Qualitative Results and Failure Cases

Fig. 7 shows some detection results of the Hand-CNN trained on both TV-Hand and COCO-Hand data, Fig. 8 compares MaskRCNN and Hand-CNN. MaskRCNN mistakes skin areas as hands in many cases. Hand-CNN uses contextual cues provided by the contextual attention for disambiguation to reduce such mistakes. Hand-CNN also predicts hand orientations, while MaskRCNN does not. Fig. 9 shows some failure cases of Hand-CNN. False detections are often due to other skin areas. Contextual cues help to reduce this type of mistakes, but errors still occur due to skin area at plausible locations. Missed detections are often due to extreme sizes or occlusions.

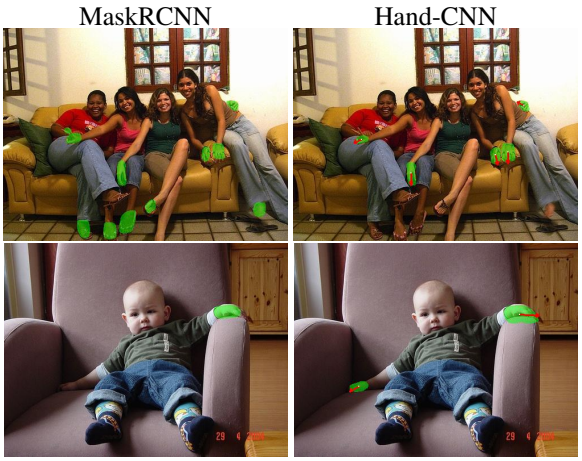


Figure 8: **Comparing the results of MaskRCNN (left) and Hand-CNN (right).** MaskRCNN mistakes skin areas as hands in many cases. Hand-CNN avoids such mistakes using contextual attention. Hand-CNN also predicts hand orientations, while Mask RCNN does not.

Figure 9: **Some failure cases of Hand-CNN.**

6. Conclusions

We have described Hand-CNN, a novel convolutional architecture for detecting hand masks and predicting hand orientations in unconstrained images. Our network is founded on MaskRCNN, but has a novel contextual attention module to incorporate contextual cues in the detection process. The contextual attention module can be implemented as a modular layer and is inserted at different stages of the object detection network. We have also collected and annotated a large-scale dataset of hands. This dataset can be used for training and evaluating the hand detectors. Hand-CNN outperforms MaskRCNN and other hand detection algorithms by a wide margin on two datasets. For hand orientation prediction, more than 75% of the predictions are within 30 degrees of the corresponding ground truth orientations.

Acknowledgements. This work is partially supported by VinAI Research and NSF IIS-1763981. Many thanks to Tomas Simon for his suggestion about the COCO dataset and Rakshit Gautam for his contribution to the data annotation process.

References

[1] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 6

[2] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proceedings of the International Conference on Computer Vision*, 2015. 6

[3] Patrick Buehler, Mark Everingham, Daniel P Huttenlocher, and Andrew Zisserman. Long term arm and hand tracking for continuous sign language tv broadcasts. In *Proceedings of the British Machine Vision Conference*, 2008. 2

[4] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ2-nets: Double attention networks. In *Advances in Neural Infor-mation Processing Systems*. 2018. 2

[5] Helen Cooper and Richard Bowden. Large lexicon detection of sign language. In *International Workshop on Human-Computer Interaction*, pages 88–97. Springer, 2007. 2

[6] Xiaoming Deng, Yinda Zhang, Shuo Yang, Ping Tan, Liang Chang, Ye Yuan, and Hongan Wang. Joint hand detection and rotation estimation using cnn. *IEEE Transactions on Image Processing*, 27(4):1888–1900, 2018. 2, 6

[7] Le Duan, Minmin Shen, Song Cui, Zhexiao Guo, and Oliver Deussen. Estimating 2d multi-hand poses from single depth images. In *The European Conference on Computer Vision (ECCV) Workshops*, September 2018. 2

[8] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 6

[9] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 1

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 6

[11] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 6

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision*, 2017. 1, 2

[13] Minh Hoai and Andrew Zisserman. Talking heads: Detecting humans and recognizing their interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 3

[14] Minh Hoai and Andrew Zisserman. Thread-safe: Towards recognizing human actions across shot boundaries. In *Proceedings of the Asian Conference on Computer Vision*, 2014. 3, 4

[15] T Hoang Ngan Le, Yutong Zheng, Chenchen Zhu, Khoa Luu, and Marios Savvides. Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection. In *CVPR Workshops*, 2016. 2

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. 2

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015. 6

[18] Leonid Karlinsky, Michael Dinerstein, Daniel Harari, and Shimon Ullman. The chains model for detecting parts by their context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2

[19] M. Kölsch and M. Turk. Robust hand detection. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 614–619, May 2004. 2

[20] M Pawan Kumar, Andrew Zisserman, and Philip HS Torr. Efficient discriminative learning of parts-based models. In *International Conference on Computer Vision*, pages 552–559. IEEE, 2009. 2

[21] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 6

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014. 3, 4

[23] Sri Raghu Malireddi, Franziska Mueller, Markus Oberweger, Abhishake Kumar Bojja, Vincent Lepetit, Christian Theobalt, and Andrea Tagliasacchi. Handseg: A dataset for hand segmentation from depth images. *ArXiv*, abs/1711.05944, 2017. 6

[24] Arpit Mittal, Andrew Zisserman, and P. H. S. Torr. Hand detection using multiple proposals. In *Proceedings of the British Machine Vision Conference*, 2011. 2, 4, 6

[25] Eng-Jon Ong and Richard Bowden. A boosted classifier tree for hand shape detection. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2004. 2

[26] Pramod Kumar Pisharady, Prahlad Vadakkepat, and Ai Poh Loh. Attention based detection and recognition of hand postures against complex backgrounds. *International Journal of Computer Vision*, 101(3):403–419, 2013. 2

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. 2015. 2, 6

[28] Kankana Roy, Aparna Mohanty, and Rajiv Ranjan Sahay. Deep learning based hand detection in cluttered environment using skin segmentation. In *ICCV Workshops*, 2017. 2, 6

[29] Roy Shilkrot, Supreeth Narasimhaswamy, Saif Vazir, and Minh Hoai. WorkingHands: A hand-tool assembly dataset for image segmentation and activity mining. In *Proceedings of British Machine Vision Conference*, 2019. 6

[30] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5, 6, 7

[31] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 2014. 6

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. 2, 3

[33] Aaron Wetzler, Ron Slossberg, and Ron Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. In *Proceedings of the British Machine Vision Conference*, 2015. 6

[34] Ying Wu, Qiong Liu, and Thomas S Huang. An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization. In *Proceedings of the Asian Conference on Computer Vision*, 2000. 2

[35] Xiaojin Zhu, Jie Yang, and Alex Waibel. Segmenting hands of arbitrary color. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2000. 2

[36] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the International Conference on Computer Vision*, 2017. 6