Pixel-Level Hand Detection with Shape-aware Structured Forests

Xiaolong Zhu, Xuhui Jia, Kwan-Yee K. Wong

Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong
{xlzhu,xhjia,kykwong}@cs.hku.hk

Abstract. Hand detection has many important applications in HCI, yet it is a challenging problem because the appearance of hands can vary greatly in images. In this paper, we propose a novel method for efficient pixel-level hand detection. Unlike previous method which assigns a binary label to every pixel independently, our method estimates a probability shape mask for a pixel using structured forests. This approach can better exploit hand shape information in the training data, and enforce shape constraints in the estimation. Aggregation of multiple predictions generated from neighboring pixels further improves the robustness of our method. We evaluate our method on both ego-centric videos and unconstrained still images. Experiment results show that our method can detect hands efficiently and outperform other state-of-the-art methods.

1 Introduction

In recent years, we have witnessed great progress in object detection in the field of computer vision. Successful applications can be found in face detection [1], pedestrian detection [2], *etc.* Nevertheless, hand detection is still a challenging problem as the appearance of hands can vary greatly in images. For instances, the shape of a hand can change dramatically due to the articulation of fingers as well as changes in viewpoint. A hand can be (partially) occluded while interacting with other objects. The color of a hand can vary greatly under different illuminations, and a hand can even appear to be textureless under extreme illuminations. Generic object detection methods [7] based on gradients often have difficulties in representing the varying appearance of hands. Heuristic skin-color detection methods [3] may also fail in practice when skin color is not discriminative enough from the background.

Recently, ego-centric cameras have become more and more popular. Images captured by such cameras often have a dynamic background, which makes hand detection even more difficult. A pixel labeling approach recently proposed by Li and Kitani [15] has shown to be quite successful in hand detection in ego-centric videos. In this paper, we extend such a pixel labeling approach to a structured image labeling problem. Instead of assigning a binary label to every pixel independently, our method estimates a probability shape mask for a pixel using

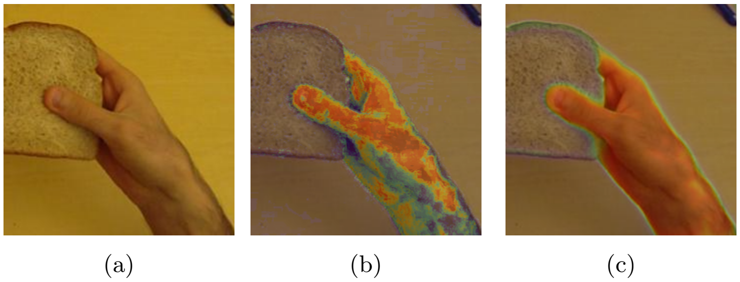


Fig. 1: Introduction to our method. a) Original image. b) Pixel-level hand detection by single pixel prediction. c) Pixel-level hand detection by structured label prediction.

structured forests. As shown in Fig. 1, our method can detect hand regions more robustly than the previous method which predicts pixel labels independently. In summary, our proposed approach has the following advantages:

- As hand pixel labels are not independent, our method explicitly models a pixel using a probability shape mask. This allows our method better utilize hand shape information in the training data and enforce shape constraints in the estimation.
- Since a probability shape mask is determined for a pixel, pixels in its neighborhood would also benefit from this prediction. Aggregation of multiple predictions generated from neighboring pixels further improves the robustness of our method.
- Our method is extremely efficient in generating a probability map of hands using random forest scheme.

2 Literature Review

Hand detection has been studied as part of human layout parsing and gesture analysis for many years. Hand detection methods can be categorized into three main approaches, namely 1) color-based methods, 2) model-based methods and 3) motion-based methods.

Methods based on skin color usually build a skin model in a color space for detecting hand regions. Mixture of Gaussians [3] is commonly used to model colors of skin and non-skin regions for hand localization [4] and hand tracking [5, 6]. Color-based methods often require a prior knowledge of skin color, extracted either from training data or from face detection, to build the skin model. These methods, however, often fail in unconstrained images and ego-centric videos where changes in illuminations cause a large variation in skin color.

Model-based methods usually model the appearance of hands using a hand template. They can be implemented as a Viola & Jones like boosted detector [1], or as a HOG detector built from a large number of images [7], or learned as an ensemble of edges [8, 9] from a set of $2D$ projections of a $3D$ synthetic hand model. Color information can also be used to further create more proposals to improve the detection performance [10]. However, their applications are often limited to a small number of hand configurations. To overcome this problem, these methods often exploit more training data in order to cover a larger configuration space. It is also possible to detect hands as part of human pictorial structure [11], which may bring more context information and allow inferring hand position via optimization. This is a common practice for still images, but usually it requires at least the upper body being visible for the inference of human layout.

Motion-based methods are mostly used for *ad-hoc* applications, *e.g.*, activity analysis and gesture recognition. They segment foreground hands from background by motion and appearance cues [12–14]. Hands can usually be tracked easily and it does not require a strong appearance model in most cases. However, motion-based methods are often not suitable for moving cameras which produce images with lots of background motion.

Recently, ego-centric cameras, *e.g.*, Google Glasses and GoPro cameras, have become more and more popular. A local-appearance-based pixel labeling method recently proposed by Li and Kitani [15] has shown to be quite successful in dealing with dynamic background and varying appearance of hands in ego-centric videos. However, their method only predicts the label of every pixel independently without considering any shape constraint. In this paper, we are going to investigate how local hand shape information can improve hand detection.

3 Shape-aware Structured Forests

In this section, we first briefly review the pixel-level hand detection using random forests. We then extend this framework by introducing a shape mask to represent the structure information. We refer to our method as *Shape-aware Structured Forests*. We also present an intermediate mapping that can accelerate the calculation of information gain during the training process. Finally, a multi-scale hand detection method will be illustrated in details.

3.1 Random Forests

Given a patch $\mathbf{I}_p \in \mathbb{R}^{w\times w\times 3}$ with a size of $w \times w$ centered at pixel p in a color image $\mathbf{I}$, we first extract a feature vector $\mathbf{x}_p \in \mathcal{X}$ using channel features [16]. A decision tree $f_{\mathbf{\Theta}}(\mathbf{x}_p)$, parameterized by $\mathbf{\Theta}$, is learned to map $\mathbf{x}_p$ to a binary label $y_p \in \{0, 1\}$, which indicates whether p is a hand pixel (i.e., $y_p = 1$) or not (i.e., $y_p = 0$). A random forest is a collection of such decision trees, each with an independent parameter $\mathbf{\Theta}_i$. The output of the random forest $F(\mathbf{x}_p)$ is the final class label y_p^*, which is obtained as the ensemble of the posterior distribution $P_i(y_p|\mathbf{x}_p)$ in the leaf node of tree i as,

$$y_p^* = \arg\max_{y_p} \frac{1}{T} \sum_{i=1}^{T} P_i(y_p|\mathbf{x}_p), \tag{1}$$

where T is the number of trees in the forest.

During the training process, each decision tree is constructed independently from a randomly sampled subset of the training data $S = \{(\mathbf{x}_p, y_p)\}$. For each decision node in a tree, one element of the feature vector $\mathbf{x}_p$ is selected for a binary test. Based on the binary test, a split function with parameter $\boldsymbol{\theta}$ is defined as

$$\Phi(\mathbf{x}_p, \boldsymbol{\theta}) = \begin{cases} 1, & \text{if } \boldsymbol{\theta}^\top [\mathbf{x}_p^\top \ \ 1]^\top \leq 0 \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where only the last element and one other element of θ are non-zero values. This function splits the training data in the current node into two subsets for its children. Usually the candidates of $\boldsymbol{\theta}$ are randomly generated and our goal is to find a candidate that maximizes the information gain, $\mathbf{G}(\boldsymbol{\theta})$, of the current split test. The information gain is defined as

$$\mathbf{G}(\boldsymbol{\theta}) = H(S_j) - \sum_{k \in \{L,R\}} \frac{|S_j^k|}{|S_j|} H(S_j^k), \tag{3}$$

where S_j denotes the set of training data in node j, $S_j^L = \{(\mathbf{x}_p, y_p) \in S_j | \Phi(\mathbf{x}_p, \boldsymbol{\theta}) = 1\}$ denotes the set of training data to be assigned to its left child and $S_j^R = S_j \setminus S_j^L$ denotes the set of training data to be assigned to its right child, $|\cdot|$ denotes the size of a set and $H(\cdot)$ denotes purity measurement *w.r.t.* y_p. As in our formulation, y_p is a binary variable, the purity can be measured by Shannon Entropy or Gini impurity [18]. A decision tree is constructed by splitting its nodes repeatedly until either the minimum number of training data in a leaf node or the maximum depth of a tree has been reached.

During the test phase, each tree will be evaluated on an input patch according to the split function in Eq. (2) iteratively until a leaf node is reached. The posterior distribution stored in that node will then be used for MAP estimation according to Eq. (1).

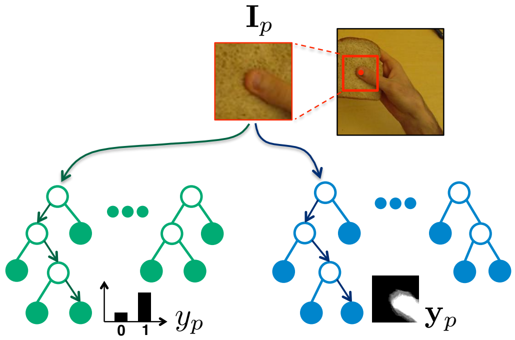


Fig. 2: The main difference between random forests and structured forests.

3.2 Learning Shape Maskes via Intermediate Mapping

For each pixel p' in the patch $\mathbf{I}_p$, similarly, a binary label can be determined by the random forests based on the feature vector $\mathbf{x}_{p'}$ extracted from the patch $\mathbf{I}_{p'}$. If these labels are considered as a whole, they form a shape mask of the hand for the patch $\mathbf{I}_p$. Obviously, pixel labels within such a shape mask should not be independent of each other. This suggests that it will be logical to learn a shape mask rather than just a single pixel label for an image patch. Learning a shape mask allows shape information from the training data be exploited to enforce shape constraints on hand detection. More precisely, the original output space $y_p \in \{0,1\}$ can be extended to a shape mask $\mathbf{y}_p \in \mathcal{Y} = \{0,1\}^{w \times w}$. Accordingly, the posterior of the shape mask is stored instead of that of the central pixel in the leaf nodes of the random forest (see Fig. 2). In this way, shape information is considered explicitly and represented as a local binary pattern in the new output space $\mathcal{Y}$. For a set of patches sharing similar appearance features in $\mathcal{X}$, our goal becomes to predict similar binary shape mask in $\mathcal{Y}$.

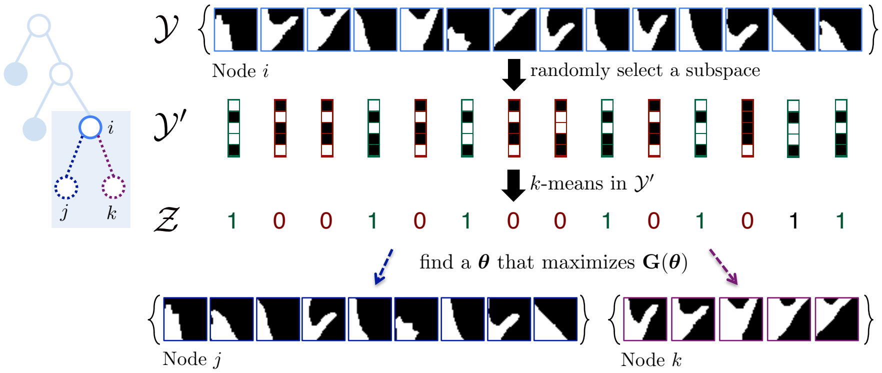

$\mathcal{Y}$
Node i
randomly select a subspace
$\mathcal{Y}'$
k-means in $\mathcal{Y}'$
$\mathcal{Z}$
1 0 0 1 0 1 0 0 1 0 1 0 1 1
find a $\boldsymbol{\theta}$ that maximizes $\mathbf{G}(\boldsymbol{\theta})$
Node j
Node k
i
j
k

Fig. 3: Intermediate mapping during node splitting. In the parent node, all mask images are firstly grouped into two clusters. Next, $\boldsymbol{\theta}$ for $\mathcal{X}$ is selected to maximize $G(\boldsymbol{\theta})$ calculated from cluster labels in $\mathcal{Z}$.

However, it will be extremely time consuming to calculate information gain in the new shape mask space as we need to enumerate all possible states. For instance, there will be $2^{16 \times 16}$ possible states for a patch with a size of 16×16. In order to speedup the calculation of information gain, an intermediate mapping is used during training to approximate the mask space by a lower dimensional space as illustrated in Fig. 3. In each node splitting, a subspace $\mathcal{Y}'$ is first randomly selected from the original mask space $\mathcal{Y}$ so that additional randomness can be injected into forest training to ensure the diversity of trees. This can be done by randomly selecting m elements from the mask $\mathbf{y}_p$. Next, k-means algorithm can be performed over the subspace $\mathcal{Y}'$ to group the training masks into 2 clusters. Either Shannon Entropy or Gini Impurity can then be used to compute information gain of a candidate split test. Finally, a standard procedure like in decision forest training can be applied to find an optimal split parameter $\boldsymbol{\theta}$ for the current node.

3.3 Implementation Details

Similar to [19], we apply the structured forest to the input image of multiple scales as illustrated in Fig. 4. We first rescale the input image $\mathbf{I}$ into several copies to form an image pyramid. In each level, color and gradient features are extracted as channel features. In particular, we extract $CIELUV$ channels as color features, which have been shown to be the most discriminative color features in many applications [2, 15, 17]. As for gradient features, we simply extract the magnitude and the orientation for each pixel and bin its orientation into a histogram followed by Gaussian smoothing among all bins. In order to describe the texture of a hand, we also include self-similarity features [20], which are pairwise differences between different cells that subdivide a patch like tiles. This will also help to differentiate the hand and non-hand regions in the patches. As all features are either channel features of order 1 or pairwise features of order 2, a lookup table can be built so that feature extraction can be done very efficiently during the test phase.

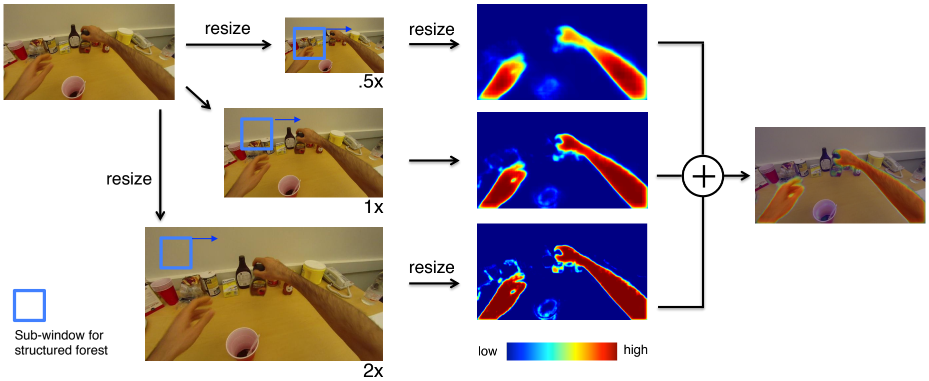


Fig. 4: Hand detection in multiple scales.

After features are extracted, a sliding window of width w can be used to apply our structured forests. For each tree i, the patch will pass through several binary tests until a leaf node is reached. In the leaf node, a posterior distribution of a mask is stored as a per-pixel posterior $m_i(x, y)$ at (x, y) as illustrated in Fig. 2 along with a per-pixel variance $\sigma_i(x, y)$. The output of the structured forest is defined as the weighted average of all these posteriors,

$$m^*(x, y) = \frac{1}{Z} \sum_{i=1}^{T} e^{-k\sigma_i(x,y)} m_i(x, y). \tag{4}$$

where $Z = \sum_i e^{-k\sigma_i(x,y)}$ is a normalization term, and k is a parameter for tuning the weights. If $k = 0$, Z will become the total number of trees T and $m^*(x, y)$ will simply be the average of all the posteriors.

The final probability map of pixel-level hand detection is obtained by averaging the results over different scales and sub-windows.

4 Experimental Results

We evaluate our structured hand detector on two types of data. The first type is characterized as the hands in ego-centric videos, where fine shape is always needed for further high-level analysis, such as action analysis or object recognition. The second type contains the hands in still images where the environment is much more unconstrained so the hand recall rate is always low using traditional object detection approaches. We first compare our methods with the state-of-the-art methods and analyze different factors that may affect the detection performance. We then conclude our observations for each type of data.

4.1 Hands in Ego-centric Videos: GTEA and EDSH dataset

We first test our approach on Geogia Tech Ego-centric Activity dataset (GTEA) [13]. The GTEA dataset involves little camera motion and is taken under the same environment as it is primarily recorded for activity recognition. Similar to the experimental setup in [15], all video clips are firstly down-sampled to 720p. There are 7 actions for 4 subjects, one of which the ground truth labels are available for evaluation. The original hand masks are quite noisy and sometimes confused with the objects in hand due to unsatisfactory segmentation. We turned to the masks made available in [15] obtained using GrabCut [21]. We performed three experiments that use Coffee sequence for training and Tea and Peanut sequences for testing, and use Tea sequence for training and Coffee sequence for testing.

We also test our approach on publicly available EDSH dataset [1], which involves more illumination changes and camera motion. EDSH1 and EDSH2 record both hands of a subject walking through different indoor and outdoor scenes in order to capture the changes in skin color. EDSH-Kitchen records a subject performing different activities in a kitchen, where there are great ego-motion and hand deformations. These are typical scenarios for hand detection in daily life and all these videos are recorded in 720p, and 442 labeled frames are used for training our shape-aware structured forests.

As far as evaluation is concerned, the F-score, *i.e.* harmonic mean of precision-recall rate, is used to measure the detection performance. We compare our result with [15] which uses a random forest of depth 10 for single pixel prediction.

Comparison We perform the same experiment using the public code [15] extracting 9 × 9 patches using color and HOG features and also include their reported best results for all 5 experiments. The F-scores are shown in Table 1. Our

[1] http://www.cs.cmu.edu/˜kkitani/perpix/

	Li *et al.* [15]	Li *et al.* (Color+HOG) [15]	Ours
GTEA-Coffee	88.8	78.05	**90.19**$_{\pm 1.07}$
GTEA-Tea	**88.0**	72.53	84.30$_{\pm 1.12}$
GTEA-Peanut	76.4	74.71	**84.37**$_{\pm 2.11}$
EDSH2	76.8	72.31	**80.43**$_{\pm 3.10}$
EDSH-Kitchen	80.5	74.37	**92.11**$_{\pm 1.41}$

Table 1: Comparison on F-score.

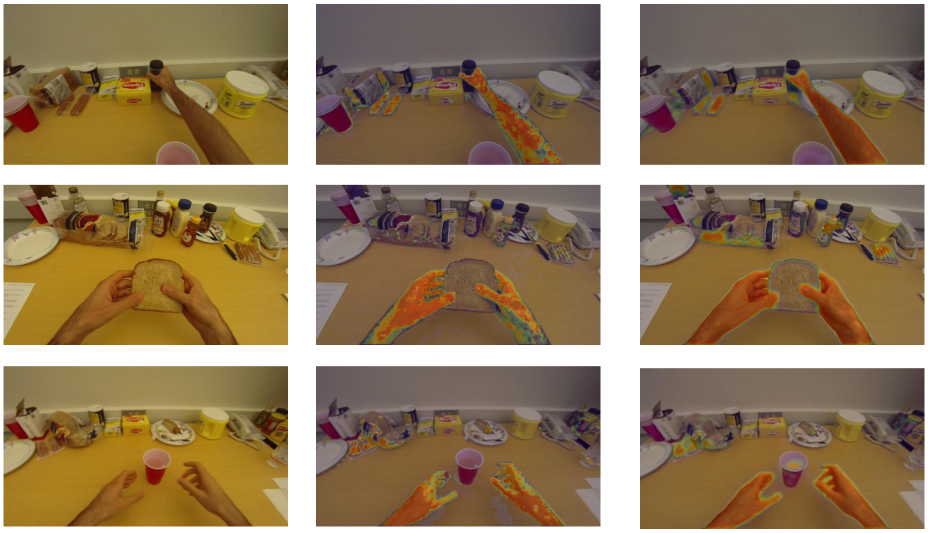


Fig. 5: Sample Images of GTEA dataset: (left column) original images, (second column) results of Li's Method [15], and (last column) our results. From top to bottom: GTEA-Coffee, GTEA-Peanut and GTEA-Tea. Best view in color.

approach outperforms the results produced by their code in all 5 experiments by a large scale.The improvement mainly happens in some cluttered regions because our method can filter out the noise and also smooth the prediction of hand region by averaging.

Figure 5 and Figure 6 show some sample images overlaid by per-pixel probabilities produced by their public code and our method in GTEA and EDSH dataset.

In Fig. 5, we find that single pixel hand prediction always fails at the edge of hand and fingers. This is because the local neighborhood of these pixels vary a lot when the hand is moving and deforming. Therefore it cannot collect a strong evidence saying that the central pixel belongs to a hand. On the contrary, our method can provide partial support from the neighborhood of edge pixels via

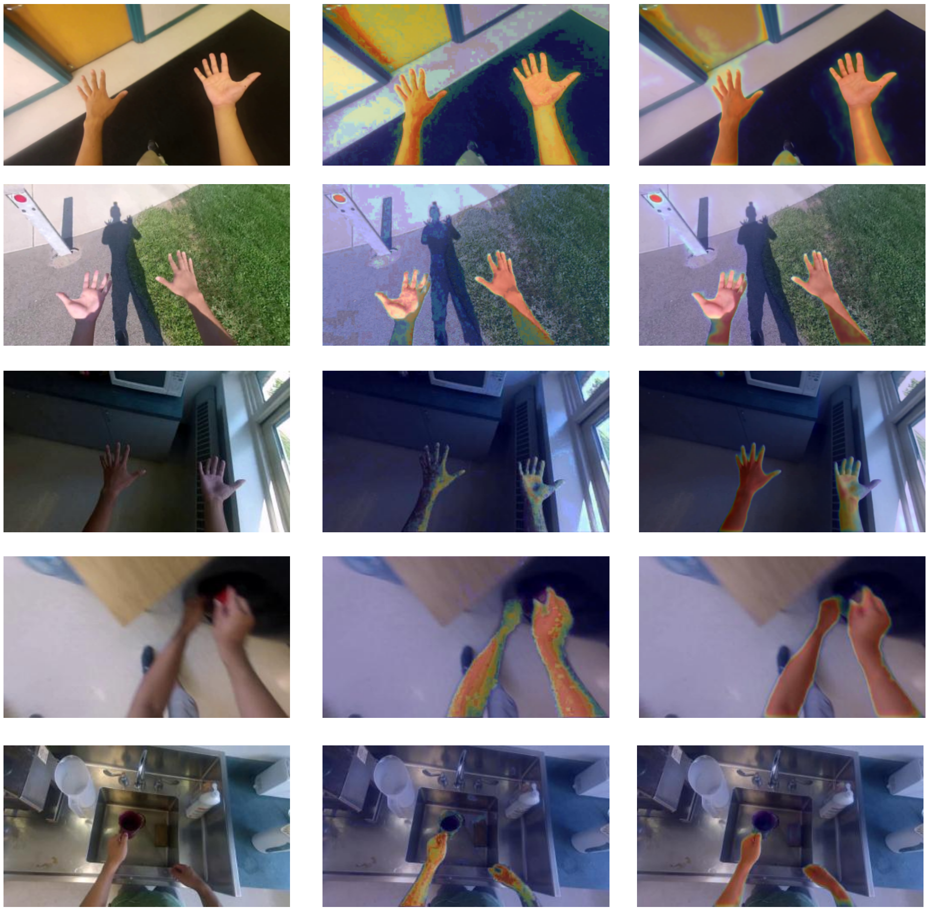
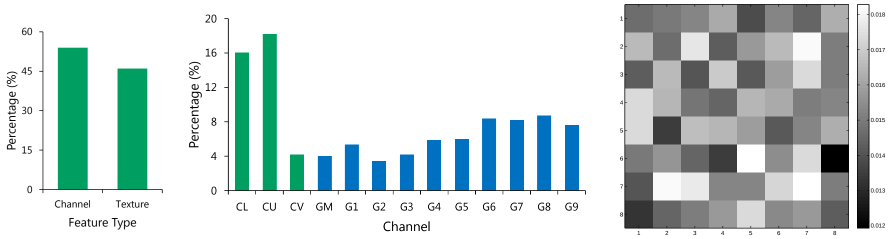


Fig. 6: Sample Images of EDSH data set: (left column) original images, (second column) results of Li's Method [15], and (last column) our results. From top to bottom: confusion with the door in EDSH1; shade on the hand in outdoor in EDSH2; poor light condition in EDSH2; motion blur in EDSH-Kitchen; confusion with sink in EDSH-Kitchen. Best view in color.

structured label predictions because these partial contributions will aggregate into the edge pixel such that the ambiguity along the edge can be removed.

In Fig. 6, both our method and single pixel prediction may cause confusion with certain textureless objects, *e.g.*, doors in 1^{st} row. However, our method smooths these regions so that it could be easily removed by post-processing. Meanwhile, our method is more robust to the incorrect labels in training samples due to improper segmentation. These labels often appear along edges of hand or suppose to be the objects in hand. For single pixel prediction, these incorrect labels will be directly treated as negative samples during training, so they will affect the prediction in a fundamental way. However, it is not common in our approach as ours is based on patch observation that is robust to pixel-level noise.



(a) (b) (c)

Fig. 7: Usage of features in the structured forest. (a) Feature of channel and similarity features. (b) Feature in different channels. The first 3 channels are *CIELUV* color channels, the 4th are magnitude of gradient and the rest are gradients in different orientations. (c) Spatial distribution of the features rasterized in a 8 × 8 grid.

Feature We investigate the contribution of different features by checking its usage in the structured forests. All selected features are aggregated from all of non-leaf nodes in the forests. First in Fig. 7(a), we show the ratio of channel features and texture features. They are almost equally important so the pairwise texture descriptor is essential in mask prediction. If we remove these features, the overall F-score will generally drop by 5%. Fig. 7(b) shows that the color features (first 3 channels) are the mostly used ones among all channel features. This means that color is still the most discriminative feature for hand detection. Moreover, the orientations of the gradients are more often used than their magnitudes, which suggests that the edge orientation of a hand is more informative in determining its shape mask. Fig. 7(c) shows the spatial distribution of selected channel features in a 32 × 32 patch, we can see that most of the places are used for predicting the hand shape mask.

Size We next examine the effect of increasing the size of patches used for our structured forests. In order to examine the hand in different resolution for egocentric videos, we down-sample the EDSH dataset from 720p to 320p. For both datasets, we increase the size of training patches from half the finger size to twice the palm size. Both in Fig. 8 and Fig. 9, there are two phases in the F-score curve. During the increasing phase, it brings more spatial context for shape mask prediction so the F-score will increase dramatically. In the decreasing phase, the structured forests will suffer from two limitations. First, it will over-smooth along the hand contour which makes it sensitive to the threshold

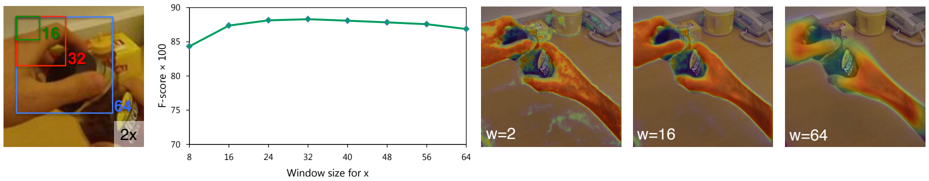


Fig. 8: Performance of our structured forests of different patch sizes for GTEA dataset.

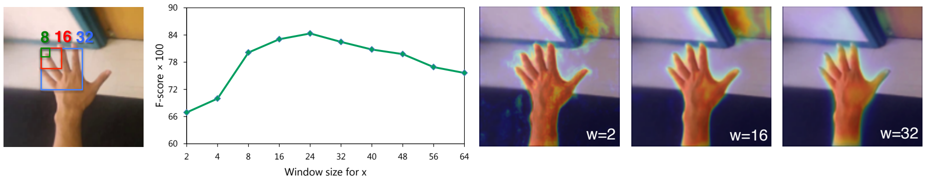
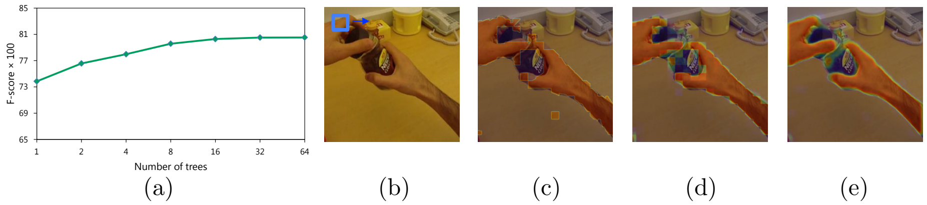


Fig. 9: Performance of our structured forests of different patch sizes for EDSH dataset.

for detection. Second, there will not be sufficient training samples for the exponentially increased output space. Thus the forests will probably overfit the training data. From our observation, more than half the palm size is suitable for a robust hand detector.

Number of trees As for the common smoothing effect introduced by our structured forests, we further examine the contribution of different number of trees to shape detection. Figure 10(a) shows the performance of structured forests of different number of trees. We use a forest trained from 16 × 16 patches to observe the shape mask prediction. In Fig. 10(c), we can see that a single tree can outline the shape but the shape contour is not smooth enough. This can be improved either by increasing the number of trees T as shown in Fig. 10(d) or reducing the stride width d as shown in Fig. 10(e). Both can accumulate more spatial context in order to obtain a better shape mask.

Timing Table 2 records the time cost for evaluating a 720 × 405 image. We compare the public code provided by the author [15] with our `MATLAB` implementation. We use 9 × 9 patch to train the single pixel predictor, and use the same size in our implementation. s and m stands for single and multiple scale detection. d is the stride width. Most of their time is spent on feature extraction compared with ours. Moreover, the time cost can be reduced greatly for multiscale implementation by increasing the stride width with a little performance drop.



(a) (b) (c) (d) (e)

Fig. 10: Performance of different number of trees and stride width. (a) Overall F-score *w.r.t.* different number of trees. (b) Original image. (c) Prediction by forest ($d = 16$, $T = 1$). (d) Prediction by forest ($d = 1$, $T = 16$) (e) Prediction by forest ($d = 1$, $T = 1$).

	Li *et al.* [15]	Ours (s, d=1)	Ours (m, d=1)	Ours (m, d=2)
Time (ms)	2138	664	3277	1129
F-Score (100×)	–	79.84	81.83	80.27

Table 2: Comparison on time cost.

4.2 Hands in Still Images: BMVC dataset

We evaluate our approach on a publicly available image dataset for hand detection in still images. All images are collected from various sources as cataloged in [10], of different resolution and imaging condition. Since there is only a bounding box available for a hand in each image, we manually annotated 3904 hands, of which 2806 are large instances. All hand images are rescaled to 128 × 128 size for sampling positive patches during training. Additional negative patches are sampled from the rest non-skin area and the images from natural scenes. Finally, our implementation is evaluated on 100 test images.

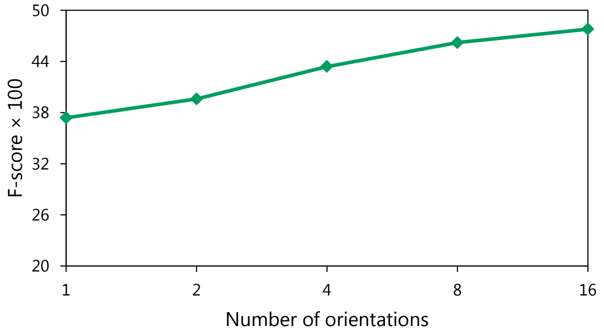


Fig. 11: Performance of F-Score *w.r.t.* different orientations used for prediction.

Hand orientation As the number of training samples is much bigger for training, we adopt a heuristic training method. All the hand images are rotated to the same orientation, and then the structured forests are trained from the samples of each orientation. Next, all predictions are merged by averaging the probability map over different orientations similar to Eq. (4).

Figure 11 shows the performance of final ensemble model *w.r.t.* the number of orientations of samples used for training. We can find the more orientations we use, the better it performs. This is because the hands in still images appear in arbitrary directions, so we need to enumerate all possible orientations during training. Otherwise, this has to be done in testing phase, which becomes a drawback in [10] which predicts an image in a long time.

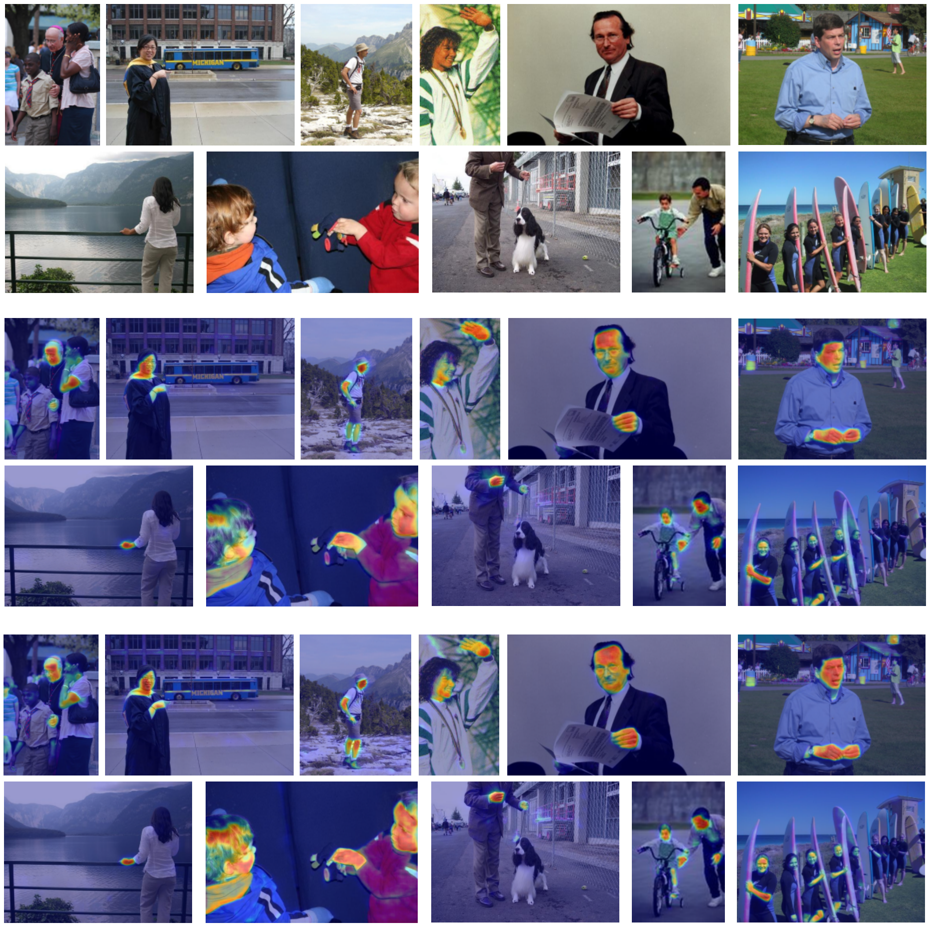


Fig. 12: Sample Images. From top to bottom: original images, predictions by a detector training from the samples with orientation downwards, predictions by a detector training from the samples with 16 orientations.

Figure 12 shows several sample images predicted by single direction and 16 orientations. The later can predict hands with stronger confidence than the former. Meanwhile, the overall F-score is quite low in this dataset, as the face regions and other skin-like regions are also confused in both experiments. Thus higher-level analysis is required in differentiating hands from other parts of human body in still images.

5 Conclusions

Our approach can detect a hand up to pixel-level accuracy in an efficient way while achieving the state-of-the-art performance. The structured output considers the information of neighboring labels so actually each pixel is evaluated much more than once during testing. Through the experimental results, we also find that the color information is critical in predicting the patch while the gradient and texture information are also relevant to the shape of the mask. This suggests that further work can be done to explore the relation between gradients, hand shapes and hand postures, which bridges the appearance with the semantic meaning of the hand.

References

1. Viola, P., Jones, M.: Robust Real-time Object Detection. International Journal of Computer Vision **57** (2004) 137 – 154
2. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian Detection: An Evaluation of the State of the Art. IEEE Trans. on Pattern Analysis and Machine Intelligence **34** (2012) 743 – 761
3. Jones, M., Rehg, J.: Statistical color models with application to skin detection. International Journal of Computer Vision **46** (2002) 81 – 96
4. Sigal, L., Sclaroff, S., Athitsos, V.: Skin Color-Based Video Segmentation under Time-Varying Illumination. IEEE Trans. on Pattern Analysis and Machine Intelligence **26** (2004) 862 – 877
5. Kolsch, M., Turk, M.: Hand tracking with flocks of features. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2005)
6. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: Proc. 7th European Conference on Computer Vision. (2002) 661–675
7. Dalal, N., Triggs, B., Europe, D.: Histograms of Oriented Gradients for Human Detection. In: Proc. IEEE International Conference on Computer Vision and Pattern Recognition. Volume 1., Ieee (2005) 886–893
8. Thayananthan, A., Torr, P.H.S., Member, S., Stenger, B., Cipolla, R.: Model-based hand tracking using a hierarchical Bayesian filter. IEEE Trans. on Pattern Analysis and Machine Intelligence **28** (2006) 1372–84
9. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Markerless and Efficient 26-DOF Hand Pose Recovery. In: Proc. 10th Asian Conference on Computer Vision. (2010) 744–757
10. Mittal, A., Zisserman, A., Torr, P.: Hand detection using multiple proposals. In: Proc. British Machine Vision Conference, British Machine Vision Association (2011) 75.1–75.11

11. Buehler, P., Everingham, M., Huttenlocher, D.P., Zisserman, A.: Upper Body Detection and Tracking in Extended Signing Sequences. International Journal of Computer Vision **95** (2011) 180–197
12. Sheikh, Y., Javed, O., Kanade, T.: Background Subtraction for Freely Moving Cameras. In: Proc. IEEE 12th International Conference on Computer Vision, Ieee (2009) 1219–1225
13. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Ieee (2011) 3281–3288
14. Trinh, H., Fan, Q., Gabbur, P., Pankanti, S.: Hand tracking by binary quadratic programming and its application to retail activity recognition. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2012)
15. Li, C., Kitani, K.: Pixel-level hand detection in ego-centric videos. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2013)
16. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral Channel Features. In: Proc. British Machine Vision Conference. (2009) 1–11
17. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. In: IEEE International Conference on Computer Vision Workshops, Ieee (2013) 397–403
18. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. Chapman and Hall/CRC, New York, NY, USA (1984)
19. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. on Pattern Analysis and Machine Intelligence **32** (2010) 1627–45
20. Bagon, S., Brostovski, O., Galun, M., Irani, M.: Detecting and sketching the common. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Ieee (2010) 33–40
21. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics **23** (2004) 309 – 314