A Two-stage Detector for Hand Detection in Ego-centric Videos

Xiaolong Zhu
Tencent
lucienzhu@gmail.com

Wei Liu, Xuhui Jia, Kwan-Yee K. Wong
The University of Hong Kong
wliu, xhjia, kykwong@cs.hku.hk

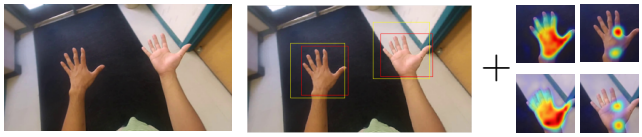
Abstract

We propose a two-stage detector that can not only detect and localize hands, but also provide fine-detailed information in the bounding box of hand in an efficient fashion. In the first stage, hand bounding box proposals are generated from a pixel-level hand probability map. Next, each hand proposal is evaluated by a Multi-task Convolutional Neural Network to filter out false positives and obtain fine shape and landmark information. Through experiments, we demonstrate that our method is efficient and robust to detect hands with their shape and landmark information, and our system can also be flexibly combined with other detection methods to handle a new scene. Further experiment shows that our Multi-task CNN can also be extended to hand gesture classification with a large performance increase.

1. Introduction

The advent of wearable cameras, such as GoPro camera, Google Glass, Microsoft SenseCam, presents us with a novel point-of-view of the world to understand users' activities in various applications [5, 26, 22]. Hands become the major objects in the resulting ego-centric videos. Detecting hands in these videos provides useful information for gesture analysis [1], hand-object manipulation [10], hand-eye coordination [3], *etc*. In contrast to third-person-view videos, where hands are usually detected upon estimating human full-body pose, hand detection in ego-centric videos has its own characteristics. Firstly, there are strong priors for hand sizes and viewpoints, so it is not necessary to perform an exhaustive search across all scales and positions to detect hands. Secondly, as hand gesture and interaction are usually defined by hand poses, these poses can be used explicitly to facilitate detection. However, generic object detection methods [25, 31, 24] cannot capture these properties both in practice and their formulation, because they do not explicitly handle the great appearance variance due to the hand articulation and viewpoint change.

To overcome the limitation of generic object detection approaches, we propose a two-stage detector that can not only detect and localize hands, but also provide fine-detailed information in the bounding box of hand. In the first stage, hand bounding box proposals are generated from a pixel-level hand probability map, which combines the hand size and position prior in ego-centric scenario. Next, each hand proposal is evaluated by a multi-task Convolutional Neural Network (CNN) to filter out false positives. The multi-task CNN models hand shape and landmarks information explicitly as its output, so it can be interpreted as a pose-aware model special for hand recognition. As shown in Fig. 1, a hand is detected with its location together with other information. We make the following contributions:



(a) (b) (c)

Figure 1: Introduction to our method. (a) Input image. (b) Detection results with ground truth bounding box (Red) and our prediction (Yellow). (c) Auxiliary output together with each bounding box: hand shape mask (left column) and heat-map of wrist and palm location (right column).

- We advance the pixel-level hand detection to a generic bounding-box based hand detection to facilitate subsequent hand pose analysis within the box.
- We augment the output bounding box by the results produced by multi-task learning which are more intuitive and informative.
- Our learned model can be flexibly and robustly combined with other detection methods for handling a new scene.

2. Related work

In the early 2000s, there were a number of research works using region-based methods to detect hands [8, 14, 20, 31]. Following the success of human face detection [24], bounding boxes were usually detected and used

as a representation of human hands. Features could be extracted from batches of training samples to train a Viola-Jones like boosted detector [8, 14] or an HOG-SVM detector [31]. Edge information was also used to form an ensemble representation to match the synthesized 2D projection from a 3D hand model [20]. However, these methods are often limited to certain applications.

In order to reduce the time used for hand detection, there comes another line of work trying to detect hands in pixel level. Early work [7, 15, 19, 9] used skin color as the cue to detect hands. In depth image sequence, hand region can be extracted using simple decision forests with pixel-wise depth comparisons [21]. When it comes to ego-centric videos, all kinds of color features were investigated under a random forest framework [29, 12] and the results showed that hand pixels can be classified by simply considering a small patch region. Serra *et al.* [18] further improved the precision of the prediction by using hand segmentation and removing small segments. Li and Kitani [11] formulated the problem of customized model prediction for specific user and scene as a recommendation problem. However, as the environment changes, it is still challenging to obtain an accurate hand mask for a new scenario [29]. It requires higher-level information to improve pixel-level approaches to narrow down the gap between pixel-level and hand-level prediction.

When it comes to two-stage detection-recognition framework, it first becomes popular in the realm of large-scale object detection recently. Among these successful models, *e.g.*, for R-CNN [4] and OCR [6], detection can be done in a two-step hypothesis-validation fashion. By proposing bounding boxes heuristically this approach does not only save time compared with brute-force sliding window approaches, but also enables a data-driven discriminative classification model that can bring higher-level information. Furthermore, the current progress in structured learning for face analysis indicates that the detection problem is highly related to other tasks such as pose estimation and landmark localization. Zhang *et al.* [28] augmented the facial landmark localization with other face attributes and improved the accuracy with a joint CNN model. Zhang and Zhang [27] extended face detection with facial landmark localization to improve the detection performance.

Our end-to-end hand detection system combines the efficient pixel-level approach with a highly discriminative hand model. It is well tailored for hand analysis in ego-centric videos.

3. System Overview

Our system consists of two stages: *hypothesis generation* and *bounding-box recognition*. The overall pipeline is illustrated in Fig. 2. During the first stage, a pixel-level hand pixel probability map is generated from an input image using structured random forests (SRF) [29], and box proposals are then generated based on this map. In the second stage, the cropped image patches are used as input for a Multi-task CNN, which will produce the detection scores together with shape masks and locations of wrist and palm. Finally, these proposals are ranked after a box-based Non-Maximum Suppression (NMS) to obtain the final detection result.

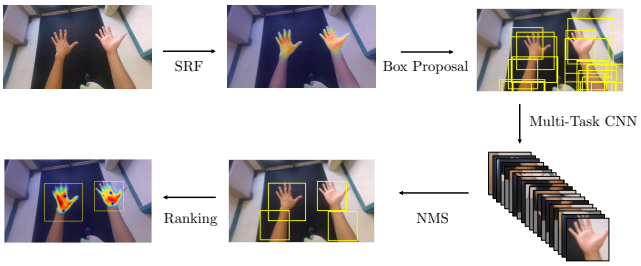


Figure 2: The pipeline of our system. Structured Random Forests (SRF) is first used to obtain a pixel-level hand probability map. Next, a set of bounding boxes is generated and fed to a multi-task Convolutional Neural Network. The network output scores will be used for Non-Maximal Suppression (NMS) and sorted as final system output.

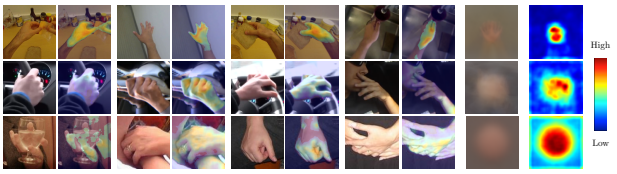


Figure 3: Illustration of the distribution of local maxima. (a) Image example and its copy overlaid with its pixel-level hand probability for three representative hand datasets. Top Row: GTEA-EDSH dataset [3, 12]. Middle: VIVA dataset [13]. Bottom: BMVC dataset [31]. (b) Average image of these datasets. (c) Spatial distribution of local maxima in hand image. The image center stands for hand central position.

3.1. Bounding Box Proposal Generation

Given an image $\mathbf{I}$, our goal in the first step is to propose a set of boxes $\{\mathbf{b}_i | \mathbf{b}_i = (x_i, y_i, w_i, h_i)\}$ [1] that cover the hands in a proper size. Ideally these boxes should tightly enclose the hands. In practice, we aim at proposing boxes with a high precision and recall of hand pixels. Previous pixel-level hand detection approaches [29, 12] enable us to obtain a probability map of hand pixels efficiently under a random forest framework. Based on these methods, we performed a set of experiments and plotted the distri-

[1] Without loss of generality, we refer to x_i as horizontal coordinate of the center of bounding box i, y_i as vertical coordinate, h_i as the height of the box and w_i as the width.

bution of the local maxima of the probability map in the hand images in three datasets, namely the GTEA-EDSH dataset [3, 12] as an example of hands in ego-centric scenario, the VIVA dataset [13] as an example in assistive driving scenario, and the BMVC dataset [31] as an example in unconstrained still images. The model for pixel-level hand detection was trained solely on GTEA-EDSH dataset and tested on the above three datasets. The results are shown in Fig 3. We observed that the local maxima of the hand probability map were always located around the hand centers. Even when we applied the model to a new scene, *i.e.*, VIVA and BMVC dataset, the local maxima of the hand probability map were still located around hand centers. This suggests that the hand center is likely located around the local maxima of the hand probability map. Consequently, we propose to randomly generate the center of a bounding box $\mathbf{b}_i$ around these local maxima. Meanwhile, the height and width of the box can also be sampled from a distribution conditional on the box center based on our statistics from the training data. We model this conditional probability as,

$$P(w,h|x,y) = P(w|x,y)\cdot P(h|x,y) \quad (1)$$
$$= \mathcal{G}(w;\mu_w,\sigma_w)\cdot\mathcal{G}(h;\mu_h,\sigma_h), \quad (2)$$

where $\mu_w = \mu_w(x,y)$, $\sigma_w = \sigma_w(x,y)$, $\mu_h = \mu_h(x,y)$, $\sigma_h = \sigma_h(x,y)$ are parameters for the Gaussian function $\mathcal{G}(\,\cdot\,;\mu,\sigma)$ and are estimated from the training data. It can be seen as a size prior at the center (x,y) that is specified for ego-centric videos. Normally, there are 20 local maxima after proper thresholding over probability map, and 100 proposals are generated in total for future evaluation.

3.2. Multi-Task CNN

At the second stage of our pipeline, the objective is to decide whether current proposal $\mathbf{P}_i$ defined by the bounding box $\mathbf{b}_i$ is a hand image or not. In order to improve the generalization performance of the recognition model, we extend the atomic recognition problem to a multi-task learning problem (MTL), which jointly learns several related tasks, *e.g.*, hand shape masks and hand landmark localization, as well. Formally, given a training data $(\mathbf{P}^{(k)}, y_0^{(k)})$, where $k \in \{1,...,N\}$ and N is the number of training samples, $y_0^{(k)}$ stands for whether $\mathbf{P}^{(k)}$ is a hand or not, we augment the output space with more tasks, *i.e.*, hand shape regression and hand landmark heat-maps regression for palm and wrist. As a result, a training sample becomes $(\mathbf{P}^{(k)}, y_0^{(k)}, \{\mathbf{y}_j^{(k)}\})$, where $j \in \{1,2,3\}$. The goal of our MTL is to minimize the energy function as follows,

$$\min_{\mathbf{W}_0,\{\mathbf{W}_j\}} \quad \sum_{k=1}^{N}\big(\mathcal{L}_0(y_0^{(k)}, f(\mathbf{P}^{(k)};\mathbf{W}_0)) + \sum_{k=1}^{N}\sum_j \lambda_j\mathcal{L}_j(\mathbf{y}_j^{(k)}, f(\mathbf{P}^{(k)};\mathbf{W}_j))\big), \quad (3)$$

where $\mathbf{W}_0$ are the parameters for main task hand detection, $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$ are the parameters for auxiliary tasks, which will be introduced later, $\mathcal{L}_k$ are the loss function for the prediction function $f(\mathbf{P};\mathbf{W}_k)$ for $k \in \{0,1,2,3\}$ respectively and λ_j controls the importance of different auxiliary tasks during training.

We adopt a CNN to solve the above energy minimization problem. The structure of the Multi-task CNN is illustrated in Fig 4. It contains two convolutional layers each followed by a Rectified Linear Unit (ReLU) layer and a 3×3 Max-pooling layer. The features are next connected to a fully connected layer. Finally, this feature vector is shared by the main task and its auxiliary tasks. We introduce three auxiliary tasks as follows:

- The first auxiliary task is to predict hand shape, which is the pixel-level mask inside the bounding box. The loss function for hand shape mask is a l_2-norm loss,

$$\mathcal{L}_1 = \frac{1}{n_s}\sum_{x,y}(S_{(x,y)} - \tilde{S}_{(x,y)})^2, \quad (4)$$

  where n_s is the total number of shape pixels and $S_{(x,y)} \in [0,1]$ is the value at (x,y) of ground truth shape mask and $\tilde{S}_{(x,y)} \in [0,1]$ correspond to the prediction of CNN.

- The second and third auxiliary tasks are to localize the hand landmarks: wrist and palm. Similar to [21], we also use an intermediate heat-map to represent each landmark location. It can be interpreted as a 2D truncated finite Gaussian distribution of a location of the landmark in the hand bounding box, whose pixel intensity represents the probability $P(x,y) \in [0,1]$ of the landmark occurring at position (x,y). We also minimize l_2-norm loss,

$$\mathcal{L}_j = \frac{1}{n_{lj}}\sum_{x,y}(P_j(x,y) - \tilde{P}_j(x,y))^2, \quad (5)$$

  where n_{lj} is the total number of heat-map pixels for hand landmark j and $\tilde{P}_j(x,y) \in [0,1]$ correspond to the prediction of CNN.

Equipped with the above three tasks, we solve the main classification task in our Multi-task CNN by a linear combination $\mathcal{L}_0 + \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_2 + \lambda_3\mathcal{L}_3$ according to Eq. (3) to Eq. (5).

3.2.1 Implementation Details

Our models are trained by stochastic gradient decent with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005. The learning rate is initialized as 0.01 and adapted during training. More specifically, we monitor the overall loss function. If the loss is not reduced for 5 epochs in a row, the learning rate is dropped by 50%.

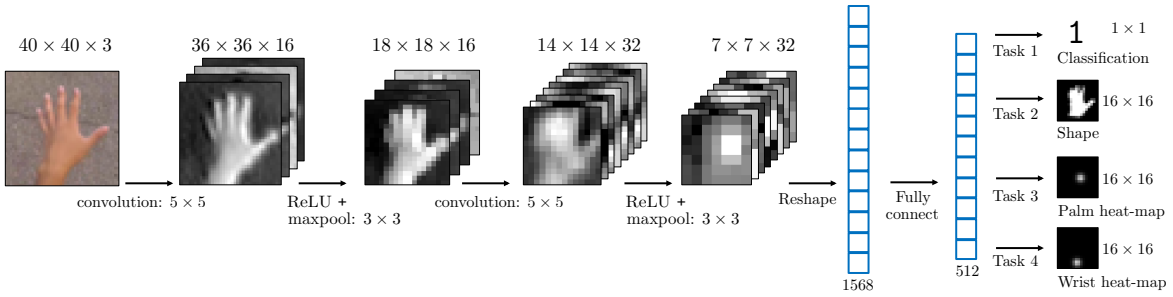


Figure 4: The structure of Multi-task CNN model.

For hand detection task, the overall number of training examples is about 30,000 positive patches and 70,000 negative patches. For recognition tasks, 100,000 samples of 24 classes are used for training. The data are randomly shuffled before sending to the network. Each epoch of training takes about 10 minutes on an iMac with 3.2GHz CPU and NVIDIA GeForce GT 755M by MatConvNet [23] implementation, and the network usually converges in 40 epochs.

3.3. Non-maximal Suppression

After all the boxes are evaluated, each box is assigned with a score. We also do this in a linear combination of the three tasks as follows,

$$s = w_0 y_0 + w_1 \sum_{x,y} \mathcal{G}_{(x,y)} S_{(x,y)} + \sum_j \sum_{x,y} w_j P(x,y) \quad (6)$$

and w_0, w_1, w_2 and w_3 are weights learned from a linear SVM and $G_{(x,y)}$ is a truncated Gaussian of the same size as the shape mask to assign high weights to the central region of the mask and low to its peripheral region. The we perform a box-based non-maximum suppression [2]. If the Intersection over Union (IoU) score for two boxes is larger than 0.5, we choose the one with a higher score. Finally, we will obtain a list of top-n candidates with descending scores as the final detection results.

4. Experimental Results

In this section, we explore our system in different aspects. First, we introduce our manually labelled GTEA [3] and EDSH [12] dataset for multi-task learning. Based on that, we evaluated each part of our pipeline in details, and then compared the detection results with atomic CNN. In order to show the generalization power of MCNN, we test our pipeline on VIVA datasets [13], with our model trained from GTEA and EDSH datasets. In addition, we show one possible extension of our system to hand posture recognition scenario.

4.1. GTEA and EDSH datasets

There are 1074 images in total for original GTEA and EDSH dataset with pixel-level ground truth mask of hands and arms. GTEA dataset consists of a subject performing several activities under the same indoor environment. It involves little camera motions. We down-sampled the image to 320×180. The original hand masks are quite noisy and sometimes confused with the objects in hand due to unsatisfactory segmentation. We used the masks obtained by GrabCut [17] instead as the mask of the hand and arms. EDSH dataset records a subject walking through different indoor and outdoor scene. Therefore it involves more camera motions and illumination changes. We also down-sampled the image to 320×180 and used the hand masks of [12]. Based on these available data and labelling, we implemented a labelling tool to draw bounding boxes of the hands in the image and pick the positions of wrist and palm so that for each hand we can obtain its shape mask excluding the arm and its wrist and palm positions. We only used these two points as hand landmarks because they are not occluded in these datasets. We used TEA, PEANUT sequences in GTEA dataset and EDSH1 and 70% of EDSH-Kitchen sequences in EDSH dataset for training, and used COFFEE, EDSH2 and the rest of EDSH-Kitchen for testing.

4.1.1 Box generation

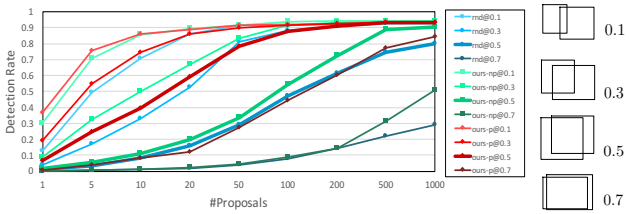
We begin by evaluating our box generator on the validation set. Intersection over Union (IoU) was calculated for each bounding box proposal against ground truth boxes. The evaluation was conducted over three sampling strategies:

1) The first approach (denoted as *rnd*) is based on pure randomness. It randomly sampled the (x, y, w, h) from a uniform distribution.

2) The second approach (denoted as *ours-np*) does not consider prior for hand size. It first predicted pixel-level hand probability map using [29]. Next, (x, y) were randomly sampled round the local maxima of the probability map, and (w, h) were sampled from a uniform distribution.

3) In the third approach (denoted as *ours-p*), we also used the pixel-level hand probability map to sample (x, y) and (w, h) were sampled according to Eq. (2).

The results are shown in Fig. 5. In practice, a detection at IoU level of 0.5 is acceptable for further classification. At this level, we can see that random sampling scheme achieved a satisfactory detection rate of 80% with 1000 proposals in the image, while our method with prior can achieved a good detection rate of 90% with less than 200 proposals and our method without prior can achieved a similar detection rate with 500 proposals. Our method with prior archived the best performance which covers all hands with 1000 proposals. At IoU of 0.7 (See Fig. 5(b)), our method with prior can achieve a detection rate of 80% with 500 proposals as well.



(a) (b)

Figure 5: A comparison of different strategies. (a) Detection rate *w.r.t.* the number of bounding box proposals at different Intersection over Union (IoU) level. The performance curves for the three settings with IoU of 0.5 are marked thicker. (b) Illustration of two bounding boxes with IoU of 0.1, 0.3, 0.5, 0.7.

4.1.2 MCNN results

In this section, we show the performance of multi-task CNN (MCNN) compared with atomic CNN. We augmented samples by randomly shifting or rotating the bounding boxes of the hands within their local neighborhood in a small scale and then cropping several sub-windows as the training data for the network. The negative samples were randomly sampled from other non-hand regions in the images. The sample predictions in the test set are shown in Fig. 6. We can see that the prediction of the hand shape and landmark locations are good enough in general, but the prediction of the fingers seems not very precise. This is probably because the fingers are thin and have varying poses and configurations, which required more training samples to cover more possibilities.

4.1.3 Detection results

Combining the first and second stage, we tested our system and compared it with an atomic CNN setting. F-score, *i.e.*, harmonic mean of precision and recall, after non-maximal suppression is shown in Fig 7(a). The CNN and MCNN achieve the best performance at top-2 level because there are at most 2 hands in the videos. Multi-task CNN further

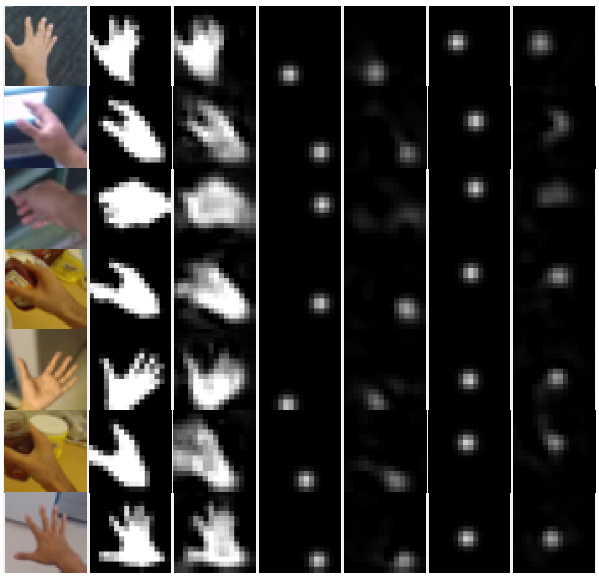
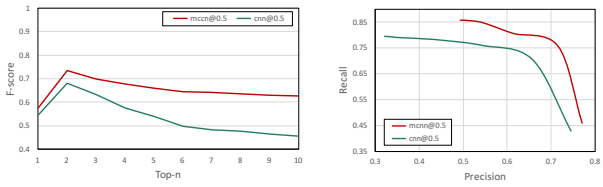


Figure 6: Some randomly selected examples. First column corresponds to the input images. Column 2-3 are the ground truths and predictions of the hand shape. Column 4-5 are the ground truths and predictions of wrist point heat-map. Column 6-7 are the ground truths and predictions of the palm point heat-map.



(a) (b)

Figure 7: Performance of CNN and MCNN setting on GTEA-EDSH test set. (a) F-score *w.r.t.* top-n. (b) Precision and recall curve.

improved the f-score by 0.05, which suggested that the axillary tasks actually help to improve the detection performance. Fig 7(b) shows the precision-recall curve, and it further confirms the superior performance of our Multi-task CNN. The time cost for each image in total is around 0.20s under `cpu` mode on an iMac with 3.2GHz CPU and 32 GB memory.

Fig. 8 shows the sample predictions by our system. For top-2 setting, our method can not only detect hand in a high IoU level but also provide useful information of hand shape and locations of wrist and palm points.

4.2. Generalization Power on VIVA Dataset

We tested our model learned from GTEA-EDSH dataset on a new dataset, VIVA dataset [13], where the subjects sit

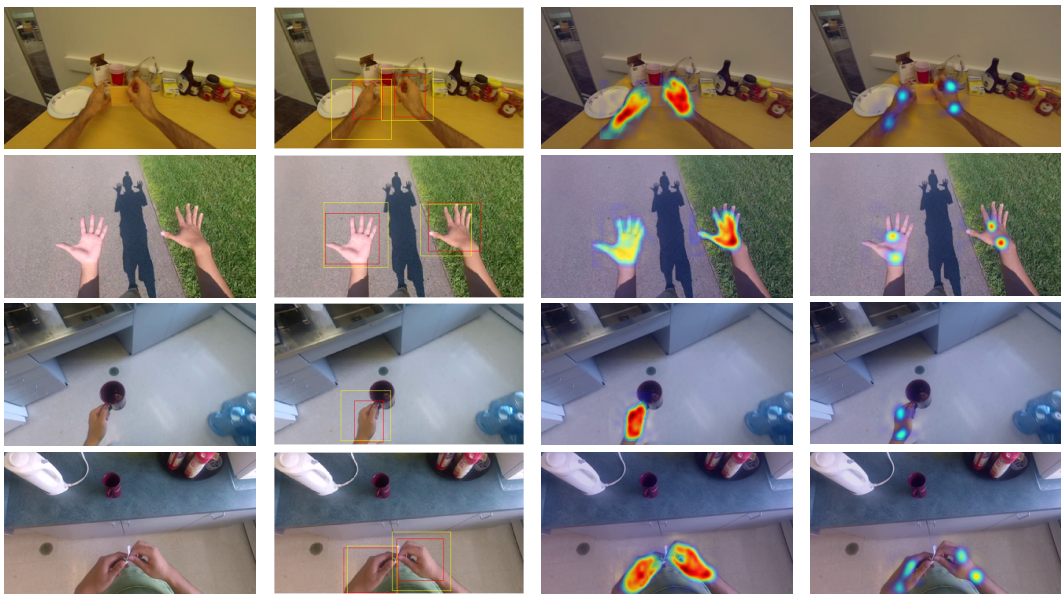
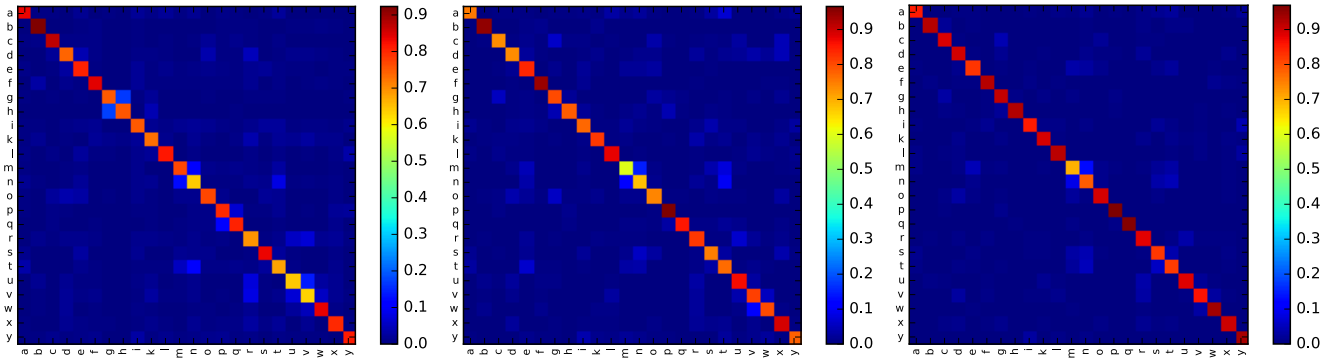


Figure 8: Some examples of our model on GTEA-EDSH dataset. From left to right: Original image, ground truth bounding box (red) and our detection (yellow), hand shape prediction in our detection box and wrist and palm heat-map in our detection box.

Figure 9: Samples for VIVA datasets. From left to right: Original image, ACF predictions (green), ground truth boxes (red) and our top-2 (yellow) results using ACF for box generation, hand shape in these boxes, landmark heat-maps in these boxes. Best viewed digitally at high zoom.

in a car with camera mounted at the back of shoulder. As only the ground truth for the bounding box are available. We only test our model in term of detection performance.

We compared our pipeline with the benchmark detection algorithm ACF [2]. The results at IoU level 0.5 are shown in Table 1. We can see that the performance of our model was not so good as the benchmark method. This is because our prior for hand size in box generator is not well generalized to this scenario. Interestingly, after observing the results of the benchmark method, we found that their method usually achieved a good recall with a low precision. This means that it can be a good box generation engine for our pipeline in this case, because the method is efficient to use and can recall the hands bounding boxes at a high IoU level. On the other hand, the boxes produced by ACF generator often tightly enclosed the hands, which cannot be directly



(a) (b) (c)

Figure 10: Confusion matrices on Finger-Spelling dataset. (a) Kernel descriptor [30]. (b) CNN. (c) MCNN.

	ACF [2]	MCNN Top-5	MCNN + ACF (Top-5)
F-score	55.5	39.9	**61.99**
Recall	79.72	49.6	**80.83**
Precision	42.5	33.3	**50.26**

Table 1: Comparison on VIVA dataset

used as the input of our MCNN as our model also considers the surrounding area of a hand for recognition. As a result, we enlarged the ACF output for our MCNN evaluation and shrinked the ones with high rank after the NMS step. Finally, our combined model achieved the best performance as shown in Table 1. We can see that it improved the f-score by 6 without any fine-tuning of the recognition model. This indicates the good generalization power of our Multi-task CNN model to the new dataset. Sample predictions are shown in Fig. 9. In general, the our detector helped to reduce the false positives of ACF method and provided reasonable hand shapes and landmark heat-maps.

4.3. Extension to Posture Recognition

Next, we trained an MCNN model for hand posture recognition to show the extensibility of our recognition model to multi-class prediction. Finger-spelling dataset [16] was used for evaluation. It consists of 5 subjects performing static postures of 24 letters from American Sign Language recorded by a Kinect camera. There are both color and depth images for each sample, and we use depth information to obtain its hand shape and the hand center as the landmark. The hand samples of Subject A, B and C were used for training, while those of Subject D and E were used for testing.

Notably, we extended the main task to multi-class recognition by changing the detection loss to softmax loss, and maintained the shape regression and hand center regression task. Data augmentation was also done before training. We compared CNN and MCNN models with Kernel Descriptor [30] classifier in Table 2. Both CNN and MCNN models were harvested after 10-epochs' iteration. MCNN outperformed the other two models by a large scale. Moreover, if we compared the confusion matrices of the three models in Fig. 10, we found that improvement was achieved in differentiating among r, u and v, and between g, h. The images of these postures often share similar gradients but differ in shapes. Our MCNN model with explicit shape regression helps to overcome the limitation of gradient based posture classifiers.

	KDes [30]	CNN	MCNN
Overall accuracy	77.31	80.54	**88.30**

Table 2: Comparison on Finger-Spelling dataset

5. Conclusion

In this paper, we present a pipeline to detect hands in ego-centric videos. We use a two-stage framework which saves time during the testing phase compared with sliding window based approach and benefit greatly from the Multi-task CNN to not only detect the hands but also give more information within that bounding box. This will be useful both for our observation and further analysis. One limitation of our method is that it requires a lot of training data to train the Multi-task CNN to improve its performance. We refer this as our future work to fine-tune the model with semi-supervised learning to make more use of our available data with different levels of labelling.

References

[1] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara. Gesture Recognition in Ego-centric Videos Using Dense Trajectories and Hand Segmentation. In *CVPRW*, 2014.

[2] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast Feature Pyramids for Object Detection. *TPAMI*, 2014.

[3] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011.

[4] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 2014.

[5] S. Hodges, S. Izadi, L. Williams, E. Berry, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. R. Wood. SenseCam: A Retrospective Memory Aid. In *Ubicomp*, 2006.

[6] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading Text in the Wild with Convolutional Neural Networks. *arXiv.org*, 2014.

[7] M. J. Jones and J. M. Rehg. Statistical Color Models with Application to Skin Detection. *IJCV*, 2002.

[8] M. Kölsch and M. Turk. Robust Hand Detection. *FGR*, 2004.

[9] M. Kolsch and M. Turk. Hand Tracking with Flocks of Features. In *CVPR*, 2005.

[10] N. Kyriazis and A. Argyros. Scalable 3D Tracking of Multiple Interacting Objects. In *CVPR*, 2014.

[11] C. Li and K. M. Kitani. Model Recommendation with Virtual Probes for Egocentric Hand Detection. In *ICCV*, 2013.

[12] C. Li and K. M. Kitani. Pixel-Level Hand Detection in Ego-centric Videos. In *CVPR*, 2013.

[13] E. Ohn-bar and M. M. Trivedi. Beyond just keeping hands on the wheel: Towards visual interpretation of driver hand motion patterns. In *ITSC*, 2014.

[14] E.-j. Ong and R. Bowden. A Boosted Classifier Tree for Hand Shape Detection. *FGR*, 2004.

[15] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. In *ECCV*. 2002.

[16] N. Pugeault and R. Bowden. Spelling it out: Real-time ASL fingerspelling recognition. In *ICCVW*, 2011.

[17] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 2004.

[18] G. Serra, M. Camurri, L. Baraldi, M. Benedetti, and R. Cucchiara. Hand segmentation for gesture recognition in EGO-vision. In *IMMPD*, 2013.

[19] L. Sigal, L. Sigal, V. Athitsos, S. Sclaroff, S. Sclaroff, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *TPAMI*, 2004.

[20] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical Bayesian filter. *TPAMI*, 2006.

[21] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. *ACM Trans. Graph*, 2014.

[22] M. Van Den Bergh and L. Van Gool. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In *WACV*, 2011.

[23] A. Vedaldi and K. Lenc. MatConvNet-Convolutional Neural Networks for MATLAB. *arXiv.org*, 2014.

[24] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 2001.

[25] C. Wojek, B. Schiele, P. Dollár, P. Perona, P. Dollár, P. Dollár, C. Wojek, C. Wojek, B. Schiele, B. Schiele, P. Perona, and P. Perona. Pedestrian Detection: An Evaluation of the State of the Art. *TPAMI*, 2012.

[26] R. Yang, S. Sarkar, and B. Loeding. Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. *TPAMI*, 2010.

[27] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *WACV*, 2014.

[28] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning Deep Representation for Face Alignment with Auxiliary Attributes. *arXiv.org*, 2014.

[29] X. Zhu, X. Jia, and K.-Y. K. Wong. Pixel-Level Hand Detection with Shape-Aware Structured Forests. *ACCV*, 2014.

[30] X. Zhu and K.-Y. K. Wong. Single-frame hand gesture recognition using color and depth kernel descriptors. In *ICPR*, 2012.

[31] A. Zisserman, A. Mittal, A. Mittal, and P. Torr. Hand detection using multiple proposals. In *BMVC*, 2011.