Multi-Scale Convolutional Neural Networks for Hand Detection

Shiyang Yan[1,2,*], Yizhang Xia[1], Jeremy S. Smith[2], Wenjin Lu[1], Bailing Zhang[1]

[1]Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, 215123, China

[2]Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK

[*]Shiyang.Yan@xjtlu.edu.cn

Abstract: Unconstrained hand detection in still images plays an important role in many hand-related vision problems, e.g., hand tracking, gesture analysis, human action recognition and human-machine interaction, and sign language recognition. Although hand detection has been extensively studied for decades, it is still a challenging task with many problems to be tackled. The contributing factors for this complexity include heavy occlusion, low resolution, varying illumination conditions, different hands gestures and the complex interactions between hands and objects or other hands. In this paper, we propose a multi-scale deep learning model for unconstrained hand detection in still images. Deep learning models, and deep convolutional neural networks (CNNs) in particular, have achieved state-of-the-art performances in many vision benchmarks. Developed from the Region-based CNN (R-CNN) model, we propose a hand detection scheme based on candidate regions generated by a generic region proposal algorithm, followed by multi-scale information fusion from the popular VGG16 model. Two benchmark datasets were applied to validate the proposed method, namely, the Oxford Hand Detection Dataset, and the VIVA Hand Detection Challenge. We achieved state-of-the-art results on the Oxford Hand Detection Dataset and had satisfactory performance in the VIVA Hand Detection Challenge.

Keywords: Hand detection; Multi-scale detection; Deep Convolutional Neural Networks; Region-based CNN

1. Introduction

Robust hand detection in unconstrained environments is one of the most important yet challenging problems in computer vision. It is closely associated with various hand-related tasks, e.g., hand gesture recognition, hand action analysis, human-machine interaction and sign language recognition. Hand detection is often the first step in the task of action recognition and is also one of the most difficult parts because the hand shapes or hand gestures can have great variability. For example, a hand may hold objects, hands may appear at different scales with closed or open palms, the hand may have different articulations of the fingers and the hand can also hold other hands. Moreover, the illumination variance and object occlusion also add extra difficulties to the task.

Hand detection has been intensely studied in the last decade. Encouraged by the success of Viola and Jones's face detection scheme [1] which combines rectangular Haar-like features and the AdaBoost classification algorithm to train a detector, similar methodologies have been researched for hand detection [2]. Though efficient in face detection, Haar-like features are not sufficient to represent complex and highly articulate objects like the human hand. As appropriate gradient histogram feature descriptors such as Histograms of Oriented Gradients (HOG) [3] have been extensively investigated for object detection, the same effort has also been made towards hand detection [4]. Despite achieving improvements, the performance is still far from satisfactory due to large variations in the appearance of hands in unconstrained settings.

Aiming to tackle the bottleneck of feature representation in object detection, a promising development, by exploiting a family of channel features, has achieved record performances for pedestrian detection [5]. Channel features compute registered maps of the original images like gradients and histograms of oriented gradients and then extract features on these extended channels. A variant of channel features, called aggregate channel features, has been adopted for hand detection in [6] where a two-stage scheme was designed for detecting hands and their orientations. Three complementary detectors were applied to propose hand bounding boxes and a second stage classifier learnt to compute a final confidence score for the proposals using these features. Based on the development of feature representation of images, various detecting schemes have been developed. Among them, a part-based model, i.e., Deformable Part Model (DPM) proposed by Felzenszwalb et al. [7] had been in the lead in objects detection before 2014. This method specially applied HOG features of images, with latent parts of objects forming a deformable graphical model of objects, and achieved promising results. Aiming to tackle the problem of hand detection, the authors of [8] also used DPM as the hands shape detector to detect hands in unconstrained images.

However, the aforementioned strategies for object detection in general, and hand detection in particular, exploited hand-crafted features which often have limited representational capability. Recently, Convolutional Neural Networks (CNN) [9] have been extensively studied in image recognition and other relevant tasks, often with state-of-the-art performance [10]. Girshick et al. [11] proposed the Region-Based Convolutional Networks (R-CNN) framework, in which the high-capacity convolutional networks were applied to bottom-up region proposals in order to localize and segment objects. More comprehensive evaluations of the R-CNN families have recently been published with different benchmarks [12], [13], [14]. An appropriately designed CNN model can learn multiple stages of invariant features of an image and a CNN based object detection is generally an end-to-end system that is jointly optimized for both feature representation and classification.

However, R-CNN also has drawbacks such as expensive multi-stage training and slow object detection as described in [15]. Recently, much research has tried to improve the R-CNN framework. Spatial pyramid pooling networks (SPPnets) [16] were proposed to speed up R-CNN by sharing computation but without improving the multi-stage training pipeline implemented in R-CNN. As a result, Girshick [15] proposed Fast R-CNN with multi-task learning and single-stage training.

How to faithfully describe an object at multiple scales is the core of a successful object detection system, which is particularly true when the objects are subjective to scale variations without restrictions. This is the precise situation of hand detection. R-CNNs are often applied to general purpose object detection, where the fixed filter receptive fields from the last layer of CNN could not match with the variable sizes of objects like hands. Some of the recent research has tried to find solutions for this. In [17], a multi-scale CNN was proposed, which comprises of two sub-networks to create complementary multiple detectors.

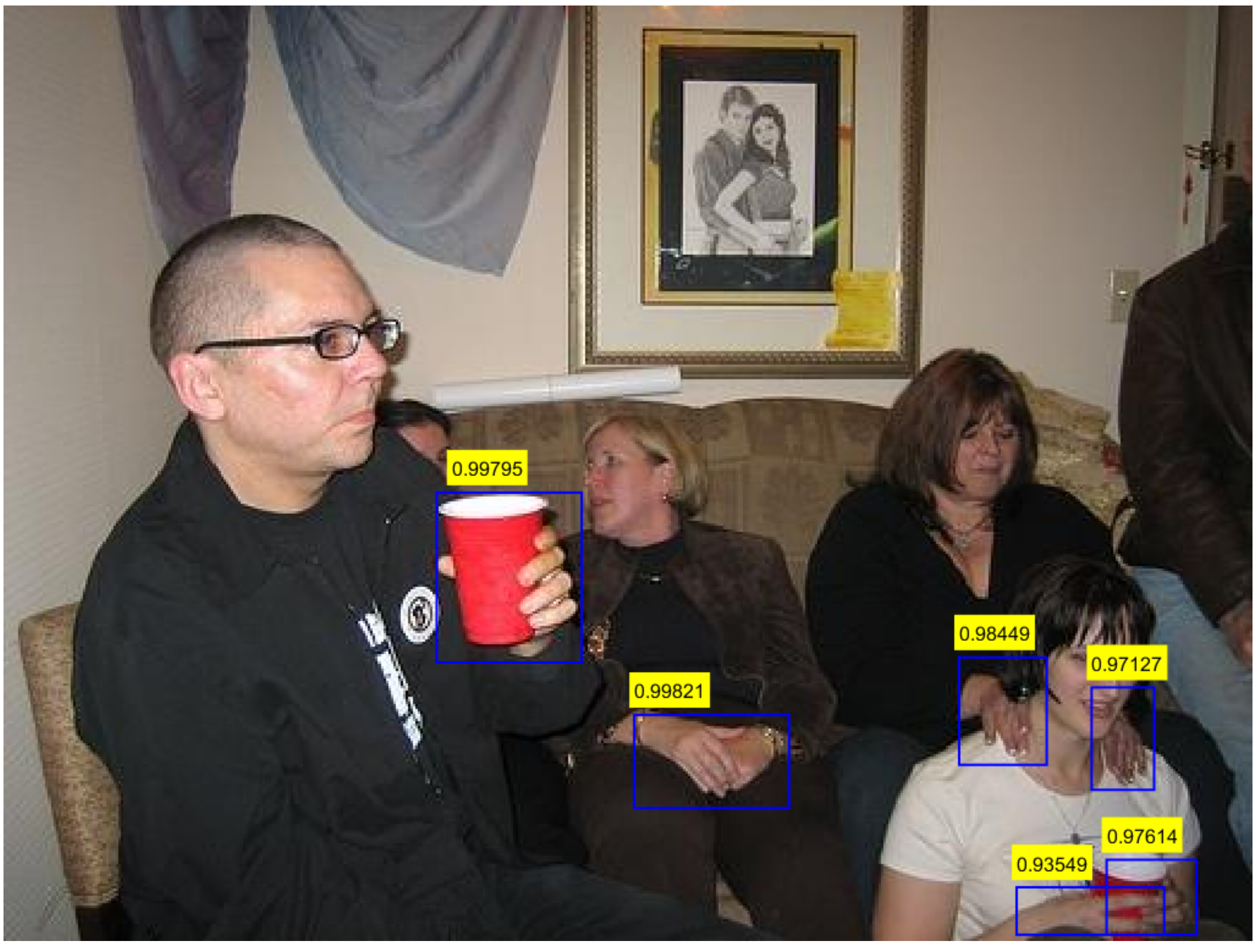


Fig. 1. *An example of the our hand detection scheme. Despite large occlusion, various scales of hands interacting with objects or other hands, the hands can be detected correctly.*

Rather than designing complex structures, as in [17], to fit the scale variations of objects, we propose a multi-scale detection system for hand objects by exploring the scale rich representations provided by a single CNN. As pointed out by Zeiler et al. [18], the information gathered in the different layers of a CNN model have different abstraction of features and scales. The last layer which is often applied in many recognition schemes [9], [15] is not sufficient to represent multi-scale objects such as hands in our system.

While the benefit of gleaning information from multiple layers of CNN has been discovered for image classification [19], our contributions lie in the integration of different features from intermediate layers to account for multi-scale hands, which has not been previously investigated.

To be more specific, our main contributions can be summarized as follows:

(1) To achieve multi-scale representation of hand objects, we propose a strategy to integrate the features from multiple layers of a CNN model.

(2) We verified the effectiveness of the proposed scheme through extensive experiments, with significantly boosted detection performance.

(3) We achieved state-of-the-art results on the Oxford Hand Detection Dataset [8] and competitive results on the VIVA Hand Detection Challenge [6].

Fig.1 shows one detection example of our methods in unconstrained environments.

The rest of this paper is organized as follows. In section 2, we briefly introduced previous research in hand detection, followed by our proposed approach explained in section 3. Section 4 details our experimental procedure and presents results from the two datasets used for hand detection. Conclusions are presented in section 5.

2. Related Works

2.1. Hand Detection

Inspired by the progress of object detection in the field of computer vision, many methods have been proposed for hand detection in the last decade. The simplest method [2] is based on the detection of skin color, which not only mixes up hands, faces and arms, but also has problems because of the sensitivity to illumination changes.

As Haar-like features and the AdaBoost classifier [20], [21], [22] have been extensively applied in many different object detection applications with outstanding successes, Mao et al. [21] proposed hand detection by improving Haar-like features with the restriction of asymmetric hand patterns. However, their experimental results demonstrated that the improvements might be marginal for complex backgrounds. Chouvavtut et al. [22] applied the use of the SAMME algorithm [23] instead of the decision tree as an estimator for the degree of orientation angles of the hands, mainly from the perspective of avoiding the over-fitting problem. Despite the achievements made, it is generally accepted that Haar-like features are not powerful enough to represent complex objects like hand due to the large variations in their appearance.

In [3], HOG was applied for human detection by Dalal and Triggs. HOG and a number of subsequent variants, have been extensively applied as an efficient feature representation in various vision problems. Felzenszwalb et al. [7] proposed the Deformable Part Model (DPM), which applied HOG features for image representation and made use of latent parts for object detection. The DPM won the championships in the VOC object detection challenge from 2007 to 2009. Recently, Mittal et al. [8] proposed to hand detection based on three types of detectors, namely DPM-based shape detector, color-based skin detector and detectors with contextual cues (context detector). Although the precision performance was satisfactory, the detection was extremely slow which prevent it from becoming a feasible real-time approach.

2.2. Region-based CNN

All of the methods mentioned above applied hand-crafted features before the classification. In recent years, there has been much progresses in CNN targeted at feature learning for object detection and other vision tasks. A typical CNN model can be illustrated by Fig.2, which consists of two convolutional layers, two sub-sampling layers and two fully connected layers. The model was proposed by LeCun et al. [24] to recognize handwritten digits, and has only recently gained popularity from the interest in deep learning [25]. The most remarkable success of CNNs is in large scale object recognition [9] in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

Szegedy et al. [26] applied separate CNNs for object detection, i.e., bounding boxes regression, and classification for the verification of whether the predicted boxes contain objects. Girshick et al. [27] proposed R-CNN, where the regions are generated by some over-segmentation algorithms such as the selective search [28] and the CNN is fine-tuned with these region proposals. With image features extracted by the trained CNN model, the system is further trained targeting at recognition with Support Vector Machines (SVM). R-CNN, the first generation of region-based CNN, has become a milestone for object detection, which also inspired a number of other superior methods [29], [15], [30], [31]. Amongst them, Fast R-CNN [15] features a joint training framework in which the feature extractor, classifier and regressor are trained together in a unified framework. Due to these advantages, Fast R-CNN is exploited as the main building block in our approach.

In many real world applications, some subtly different objects to be discriminated involve fine-grained details. As the differences between subcategories are small, ideal feature representations should take multi-scale image patches into account from different CNN layers. However, neither R-CNN nor Fast R-CNN considers the issue of information granularity with regard to fine-grained recognition. This is also one of the main limitations to many other CNN models which only target coarse-grained recognition problems. How to incorporate multi-scale features in fully convolutional neural networks to achieve improved performance has become an interesting research issue in computer vision research.

Bell et al. [32] proposed to account for the multi-scale information with an Inside-Outside Network (ION), which combines features at multiple scales and levels of abstraction with the aid of skip pooling and spatial recurrent neural networks. Recently, Zagoruyko et al. [33] further developed the idea of skip connections to extract features at multiple network layers and presented the MultiPath network to further improve the standard Fast R-CNN object detector.

Our work follows a similar strategy of gathering features from multiple layers by skip pooling for hand detection.

3. Our Methods

The proposed hand detection network is illustrated by Fig.3. Although our improvements upon the CNN architecture are not constrained by the type of models, our design is based upon the VGG16 model [34], a widely applied deep CNN model. The VGG16 network model consists of five convolutional blocks: Conv1 to Conv5. The Conv1 and Conv2 blocks each contain two

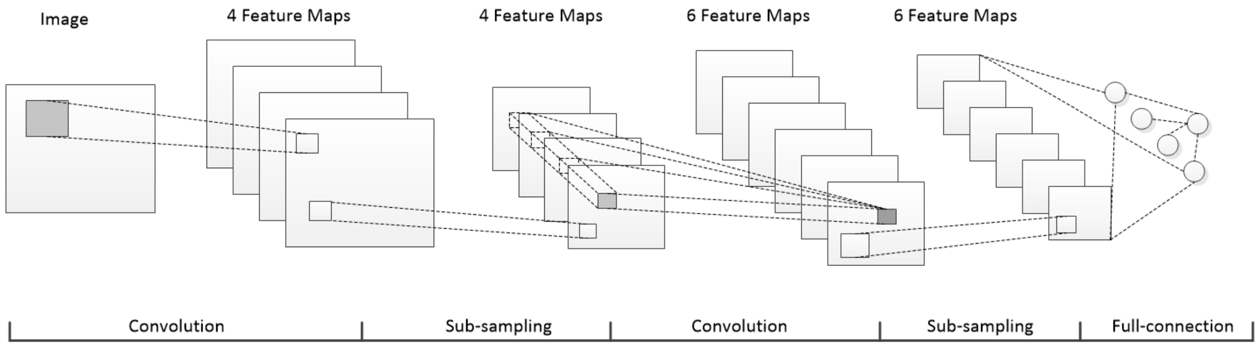


Fig. 2. *A common CNN architecture*

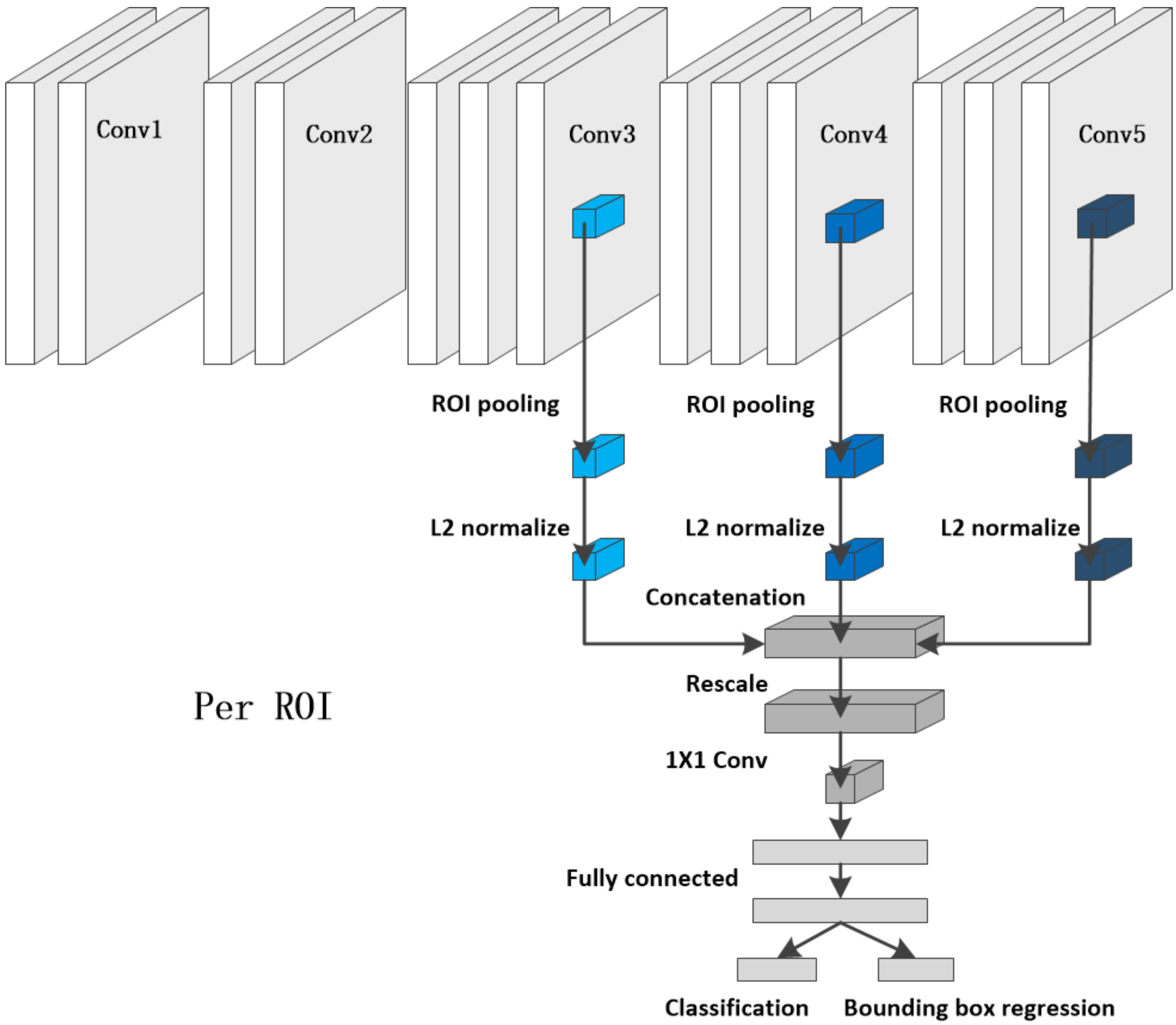


Fig. 3. *The model structure of the proposed networks.*

convolutional layers while there are three convolutional layers in Conv3, Conv4 and Conv5. Instead of pooling the Region of Interest (RoI) features only at the last convolutional layer, we add RoI pooling layers after Conv3, Conv4 and Conv5.

The Fast R-CNN [15] takes the whole image and sets of bounding boxes as inputs, and produces a feature map by convolutional and max pooling layers. Each bounding box will be initially projected to the feature map, followed by a pooling operation in a pooling layer, where RoI pooling, a special case of the spatial pyramid pooling layer in SPPnet [16], is adopted. As the most important component of Fast R-CNN, the RoI pooling layer enables the acceptance of different image sizes of the region proposal thus improving the R-CNN method. RoI max pooling first divides each RoI feature map into a fixed number of sub-windows and then applies max pooling in each window. As a result, different sizes of input can be pooled into fixed-lengths of feature representations.

As the different layers in Convolutional Neural Networks represent different abstraction for features, we implemented feature pooling from multiple layers [32],[33]. As previously explained, the paradigm has been generally acknowledged as an important improvement to earlier CNN models where only the last layer of the CNN is exploited for feature representation [15]. The information from the last single layer is only suitable when the task is to generate class labels to images or regions because the last layer is the most sensitive to semantic information [35]. When a task involves fine-grained information, which is the case of our work on hand detection, outputs from the last layer alone are not sufficient to represent the image features. The same statement can be applied to many other tasks such as image segmentation, pose estimation or fine-grained object recognition. As an efficient solution, features from shallow layers and deeper layers should be fused together to capture multi-scale information about a hand image.

Also, tiny hand objects will be difficult to identify based only on the last convolutional layers. Take the VGG16 model as an example where the last convolutional layer has an overall stride of 16. If a hand image is 16×16 pixels, the corresponding feature map in this layer would be only 1 pixel, which means the corresponding receptive field is too large to capture the essential information of the hand object. However, if features from multiple layers are aggregated, image representations from shallow layers will be retained which contain much more detailed information on tiny hand objects and accordingly facilitate multi-scale detection.

As previously explained, RoI pooling generates fixed length features. One potential problem for the pooled features is the wide range of attribute values as they vary widely in magnitude across different layers. The deeper layers often have much smaller values compared with shallower layers because of the convolution operation. This lack of feature normalization will cause convergence problems when training the CNN model. Also poor performance would be expected as the model will be biased by the larger features values. As a simple solution, we utilized L2 normalization after RoI pooling as suggested in [32] to normalize the features.

The L2 normalization is implemented after RoI pooling. The L2 normalization is conducted on all the pixels of the feature maps, and all the feature maps are treated independently, i.e.,

$$\hat{X} = \frac{X}{\parallel X \parallel_2} \tag{1}$$

$$\parallel X \parallel_2 = \left(\sum_{i=1}^{d} |x_i| \right)^{\frac{1}{2}} \tag{2}$$

where $\hat{X}$ represents the normalized features and X represents original features. In Equation 1, features are L2 normalized. In Equation 2, d represents the dimension of each entry of features.

The feature normalization step proposed in [32] also includes a re-scaling operation which is an important concept stemming from [36]. The scale factor can be a fixed value. We empirically set up the scale factor from experiments. Specifically, the mean scale of features pooled from the last convolutional layer (Conv5) on the training set was measured and set as the target scale. Then the mean scale of features from each convolutional layers are computed and the scaling factor can be consequently obtained by simple division.

To match the original shape of the RoI pooled features (512$\times$7$\times$7), we reduced the concatenated feature dimension using 1$\times$1 convolution. Hence, the outputs from our network architecture would be the same as the original VGG16 model. Subsequently, two fully connected layers are applied before the multi-task strategies, namely, feature classification and bounding box regression.

4. Experiments

In this section, we presented the results from our methods on two benchmark datasets: the Oxford Hand Detection Dataset [8] and the VIVA Hand Detection Challenge [6]. All the experiments were conducted using the Ubuntu 14.04 operating system. The CNN models were trained on the Caffe platform [37], a C++ deep learning library. The max iteration of training and learning rate were set as 40000 and 0.001, respectively. For the Oxford Hand Detection Dataset, we applied the PASCAL VOC evaluation toolkit for evaluation; for the VIVA Hand Detection Challenge, we submitted our results to the official evaluation server. All the data of the other participator's methods was obtained from the organizing committee.

4.1. Oxford Hand Detection Dataset

Mittal et al. [8] collected this dataset for hand and its orientation detection. This is a comprehensive dataset collected from a number of different public image resources. As illustrated in [8], no restriction was imposed on the pose or visibility of people, and there was no constraint placed on the environment.

The dataset is split into training (1844 images), validation (406 images) and testing sets (436 images). The details of the dataset can be found in [8]. However, the original annotations of the training dataset are not axes aligned, but placed according to the orientation of the hand's wrist. In our experiment, we re-allocate the bounding box annotations of the training set by making it align with the horizontal axis to facilitate the training of the deep learning model. These annotations are new in our research, which are consistent with locations and scales of the original bounding boxes. The testing set was applied in their original form, so as to compare with other methods.

For all the images and hand instances in the validation and testing dataset, we conducted comparison experiments with both the baseline approach and the proposed model. To compare with previously published methods, we also performed experiments using the original evaluation protocol of [8] so as to evaluate the detection performance of the big hand instances as in [8].

Fig.4 presents some image examples from the dataset and the corresponding annotations. As can be seen from the figure, there are large variations in the illumination conditions, scales, viewpoints and hands poses. Also, the dataset contains a number of small hands objects which

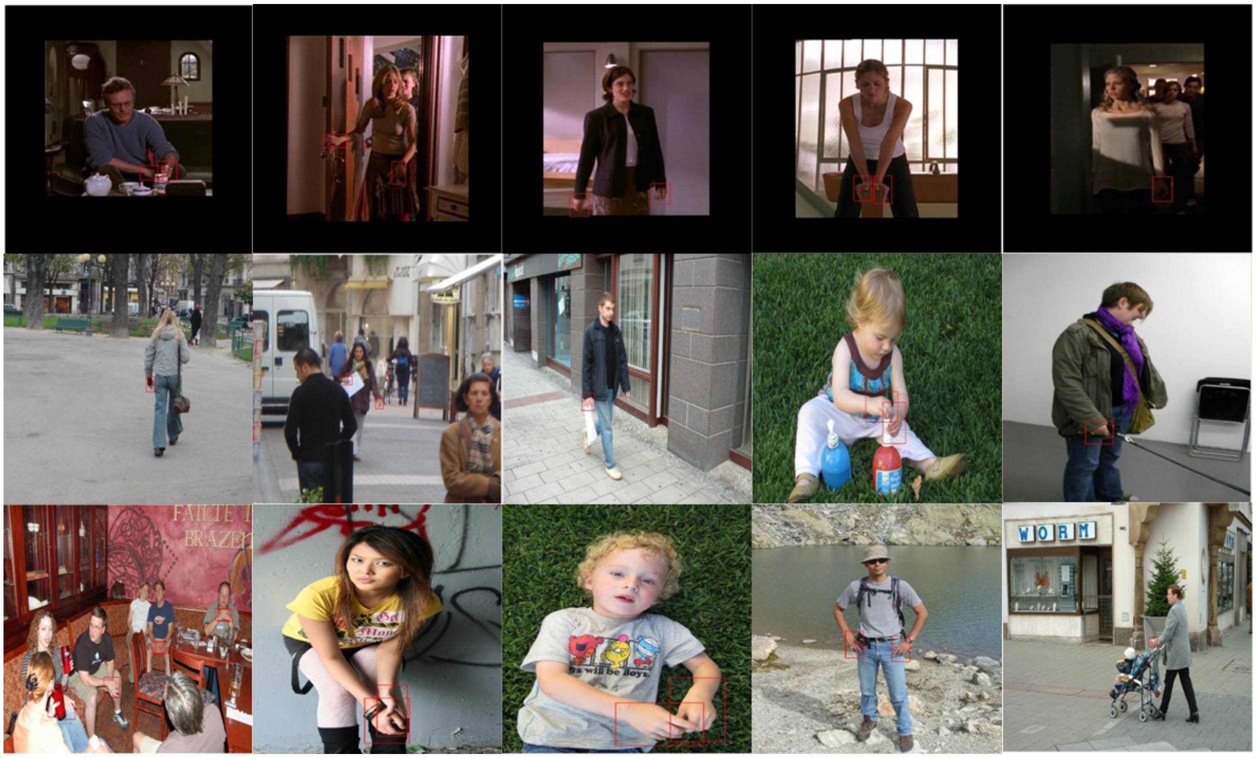


Fig. 4. Oxford hand detection Dataset

Table 1 The Average Precision (AP) on the Oxford Validation and Testing Set. All hand instances were used for evaluation.

Methods	Validation Set	Test Set
VGG16(baseline)	45.9%	47.7%
Our Model	**51.2%**	**49.6%**

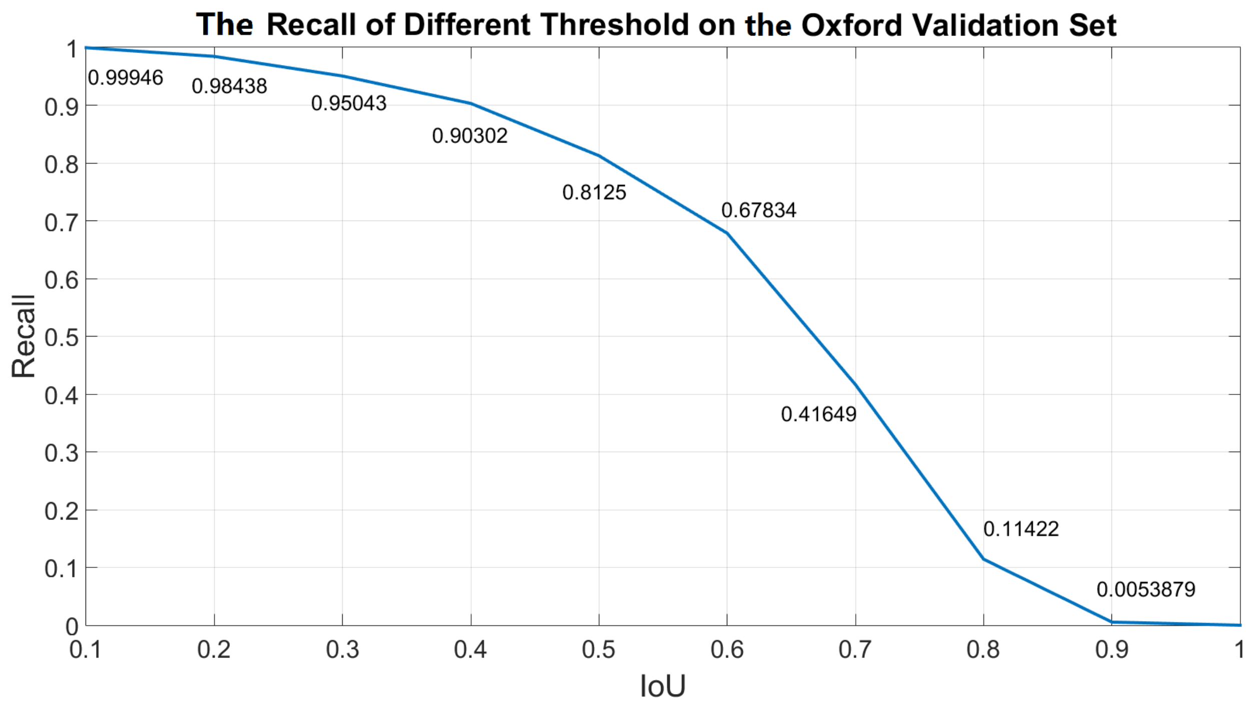
Table 2 The Average Precision (AP) on the Oxford Hand detection Dataset and comparison with previous methods. Only large hand instances (larger than a fixed area of bounding box) are considered in the evaluation.

Methods	AP
Multiple Proposals [8]	48.2%
VGG16(baseline)	56.8 %
Our Model	**58.4%**

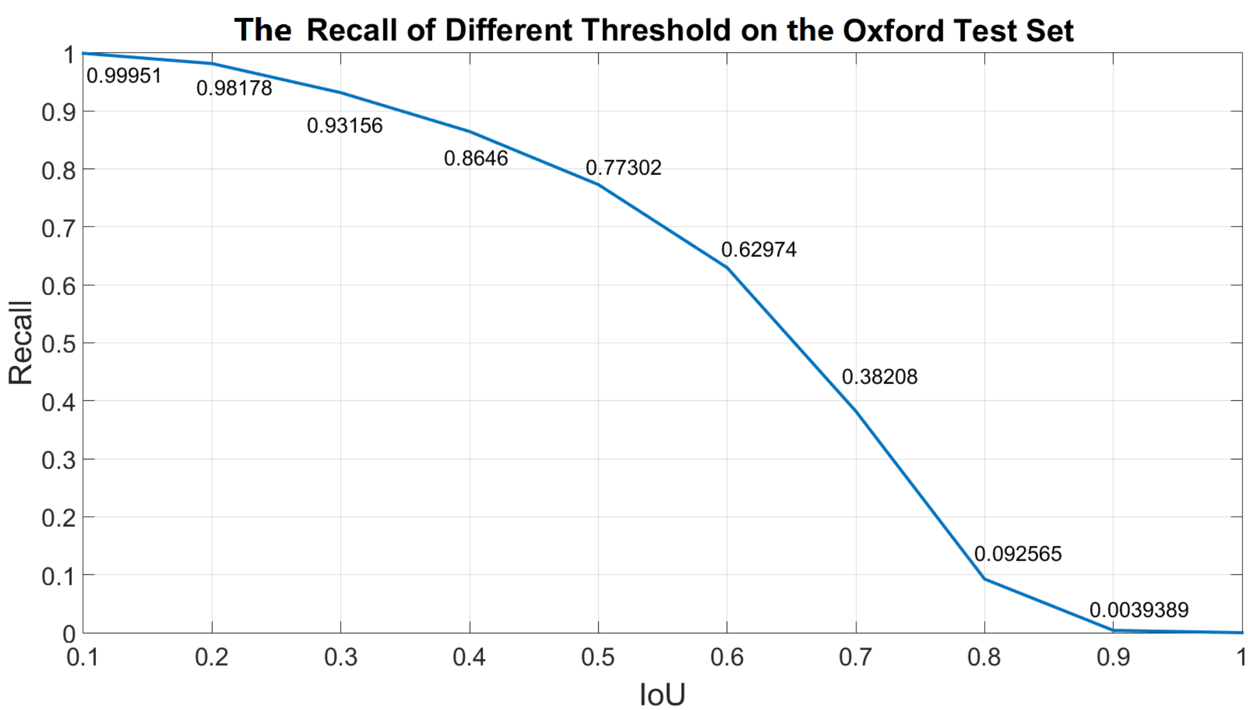
adds extra difficulties to the detection task.

The experimental procedure can be further explained as follows:

As a first step, a set of region candidates was generated by Edgeboxes [38] on the training set. We set the maximum number of candidates to 3,000. The Edgeboxes algorithm would generate bounding boxes according to the confidence values. The top 3,000 candidates have higher probabilities of containing objects. We then trained the proposed CNN model using ground truth



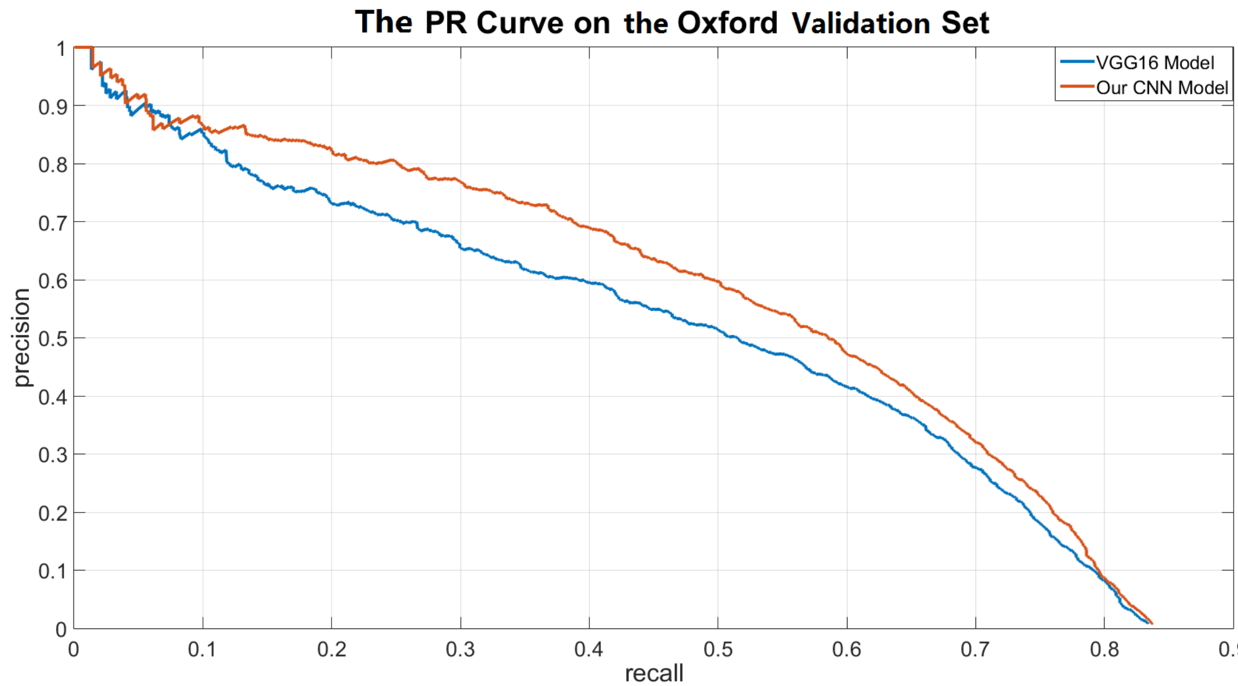
(a)



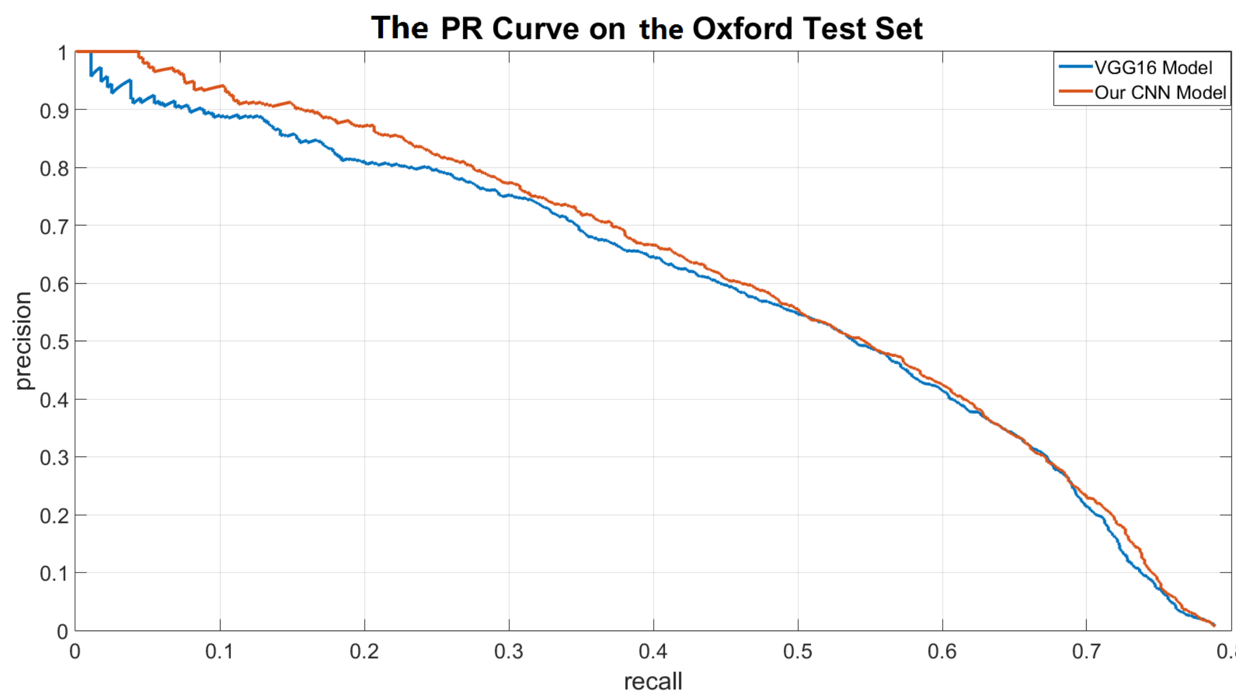
(b)

Fig. 5. *Recall of Edgeboxes algorithm on the Oxford dataset: (a) validation set. (b) test set*

annotations and the generated candidate regions. During training, positive samples were collected with a fixed overlapping ratio. If a candidate region overlaps more than 0.5 with the annotated



(a)



(b)

Fig. 6. *Precision-Recall curve on the Oxford dataset: (a) validation set. (b) test set.*

bounding box, it was considered as positive. Otherwise, the region was treated as a background. The percentages of positive samples and negative samples to all of the candidate regions are 25%

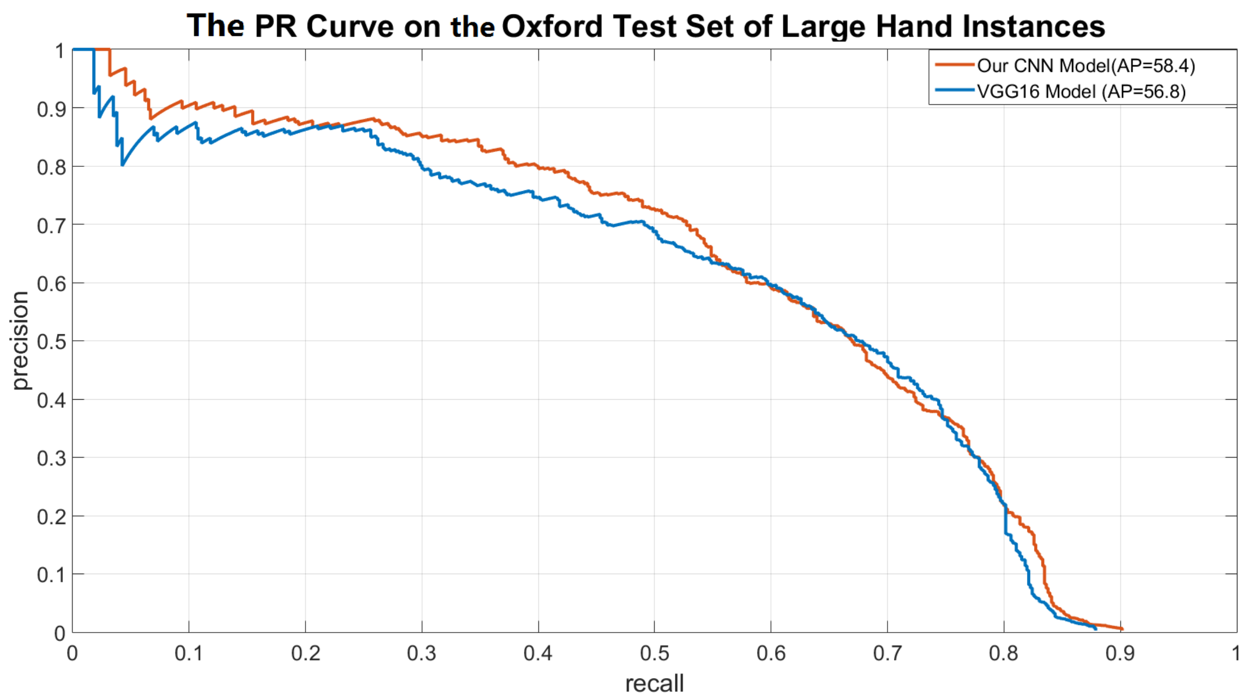


Fig. 7. *Precision-Recall curve on the Oxford test dataset with only large hand instances considered.*

and 75% respectively.

Following the common practice of applying CNN, the model was first pre-trained with ImageNet and then fine-tuned with the sampled candidate regions previously explained. The popular Stochastic Gradient Descent (SGD) algorithm was applied for the CNN training, with each SGD mini-batch size chosen as 128. As pointed out by Girshick [15], it is not necessary to fine-tune all the layers. In our experiments, we kept the Conv1 and Conv2 parameters unchanged, and fine-tuned the other layers with a maximum iteration of 40,000. During training, we encountered the under-fitting problem with the model training. In order to compensate for this, we removed all the drop-out layers of the model [32], and observed improved results.

After training, the methods were tested on the validation and testing sets separately. We firstly plotted the recall versus intersection over union (IoU) curve on both of the Oxford Validation set and Test set, as illustrated in Fig.5. The recall versus IoU curve was applied as the main evaluation metric for the region proposal algorithm in [39]. This figure indicates, that for certain overlap ratios (IoU) between detected boxes and ground-truth regions how many true positive samples can be fetched. Hence, in this paper, we also plotted this curve to evaluate the performance of the

Fig. 8. *Detected examples from the Oxford Hand Detection Dataset: The red boxes are the annotated hand positions. The blue boxes are the detected boxes with the corresponding label tags in yellow.*

Edgeboxes algorithm. The Edgeboxes algorithm achieved 81.25% and 77.30% recall rates when the IoU ratio is 0.5 on the validation set and test set, respectively. The recall rate is not very high due to the unconstrained settings of the dataset and the large variances of shape, pose, and the scale of the hands.

We then ran the CNN models using the generated candidate regions. To prove the capability of

Fig. 9. *Incorrect detected examples from the Oxford hand detection dataset*

the proposed model, we set the original VGG16 [34] model as the baseline. To keep the number of detected boxes limited, we applied Non-Maximum Suppression (NMS) with a threshold of 0.3 in the experiment to eliminate redundant bounding boxes. Following the popular Average Precision evaluation protocol, we applied the PASCAL VOC [40] evaluation tookit to calculate the Average Precision (AP). As pointed out by Provost et al. [41], simply using accuracy results can be misleading. A Precision-Recall (PR) curve is normally used as the evaluation metric for object detection [15]. Fig.6 shows the PR curve for the baseline method and our methods. The area below the PR curve is the AP value. We can see clear improvements on the AP results from the figure. Table 1 shows the AP values on the Validation and Test sets. On both of the validation and test set, our methods outperformed the baseline approach, with AP values of 51.2% and 49.6% on the validation and test set, respectively.

To compare with the previously published methods, experiments were also conducted with the same evaluation protocol of [8]. In [8], hand instances larger than a fixed area of the bounding box (1500 sq. pixels) are used in evaluation. [8] also applied the PASCAL VOC evaluation protocol for the evaluation. Hence, our experiments are consistent with the procedure in [8]. Fig.7 shows the PR curve of the proposed model and the baseline approach. From the figure, it is obvious that our method (red curve) has a higher AP value than the baseline method (blue curve). Table 2 shows the AP results of our method and comparisons with other published results. Our method achieved a state-of-the-art AP result of 58.4%.

Fig.8 illustrates some of the detected examples on this dataset. Despite the severe occlusion and small sizes of the hands in some images, the hands can still be correctly detected. Table 2 summaries the results of our approach and some of the previously published methods, confirming

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

Fig. 10. *Examples of the VIVA hand detection dataset: (a) different view point. (b) skin-like non-hand objects appear in the image. (c) occlusion example. (d) illumination variation.*

Table 3 Average Precision (AP) on VIVA L1 and L2 Dataset and comparison with previous methods.

Method	L1 Set	L2 Set
CNNRegionSampling [42]	66.8%	57.8%
ACF Depth4 [6]	70.1%	60.1%
YOLO [43]	76.4%	69.5%
FRCNN [44]	90.7%	**86.5%**
Our Model (Multi-scale Fast R-CNN)	**92.8%**	84.7%

the improved performance from our proposed method.

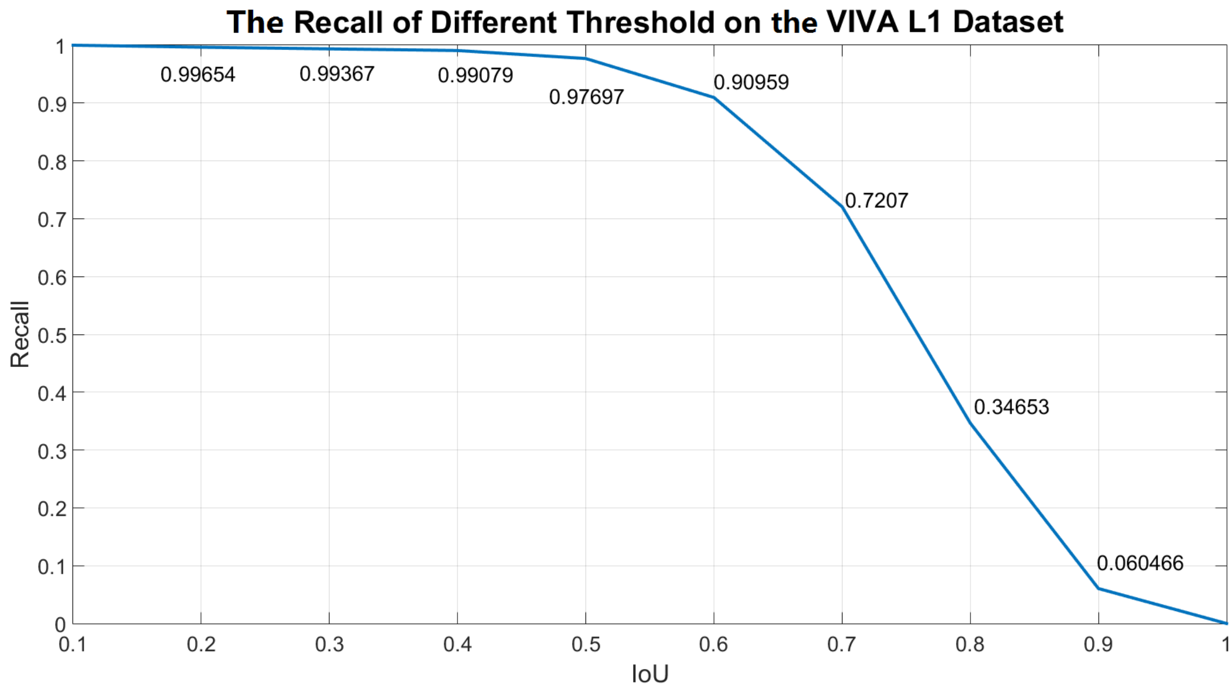
To investigate the situations where the proposed method was not successful, Fig.9 shows some examples of incorrectly detected images. In most of these instances, the mistake is misclassifying some other objects as hands. For example, feet, corsage or logos on T-shirts appearing in the image would be misjudged as a hand, as illustrated in the figure. This problem is not trivial and the solution may not be straightforward based on the current method. A possible approach to tackle the issue is to explore the contextual information in the discrimination of some hand-like objects and real hands.

4.2. VIVA Hand Detection Dataset

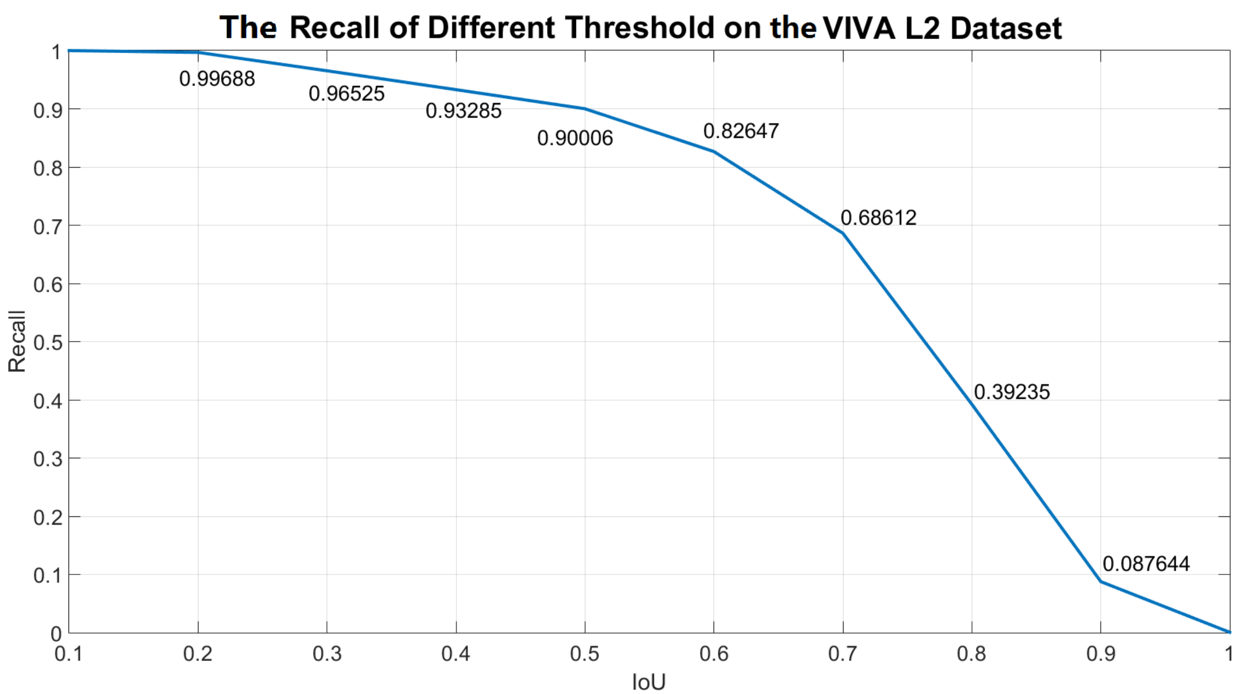
The University of California, San Diego [6] assembled an annotated dataset for hand detection under realistic driving conditions, with the objective of serving as a component in the Vision for Intelligent Vehicles and Applications (VIVA) challenge [1].

There are a number of challenges for the detection of a driver's hands in real driving conditions.

[1] http://cvrr.ucsd.edu/vivachallenge/index.php/hands/hand-detection/



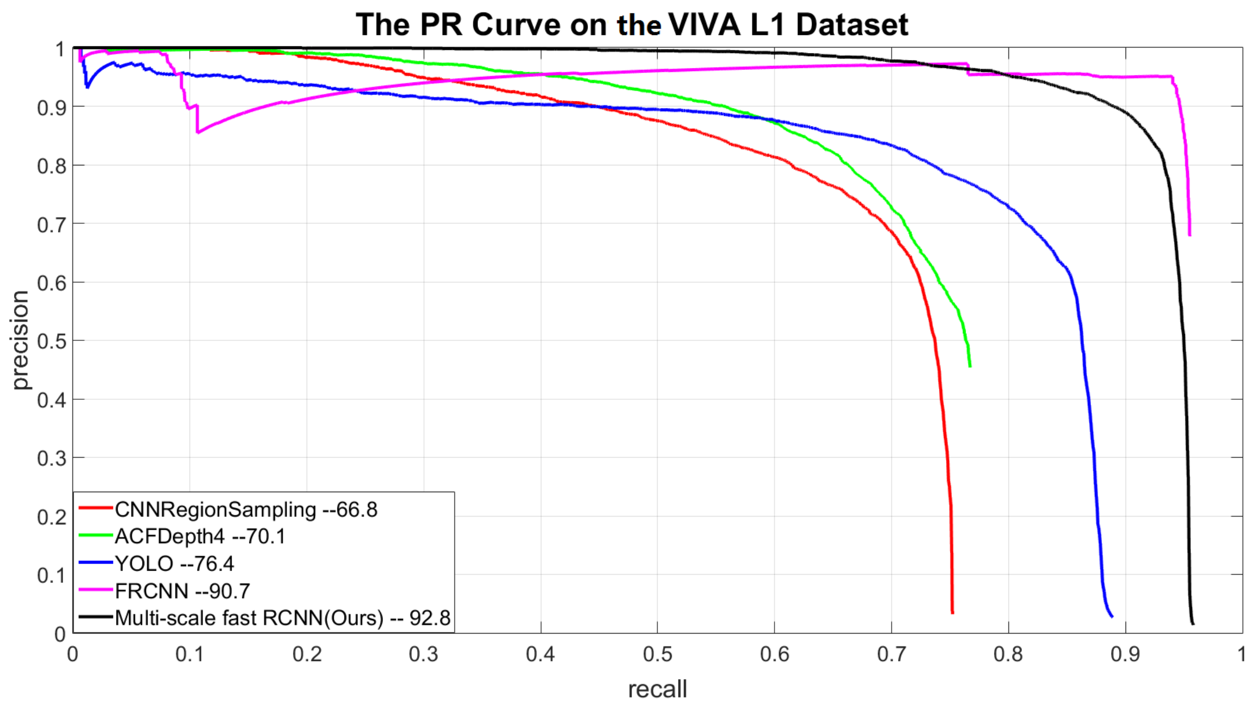
(a)



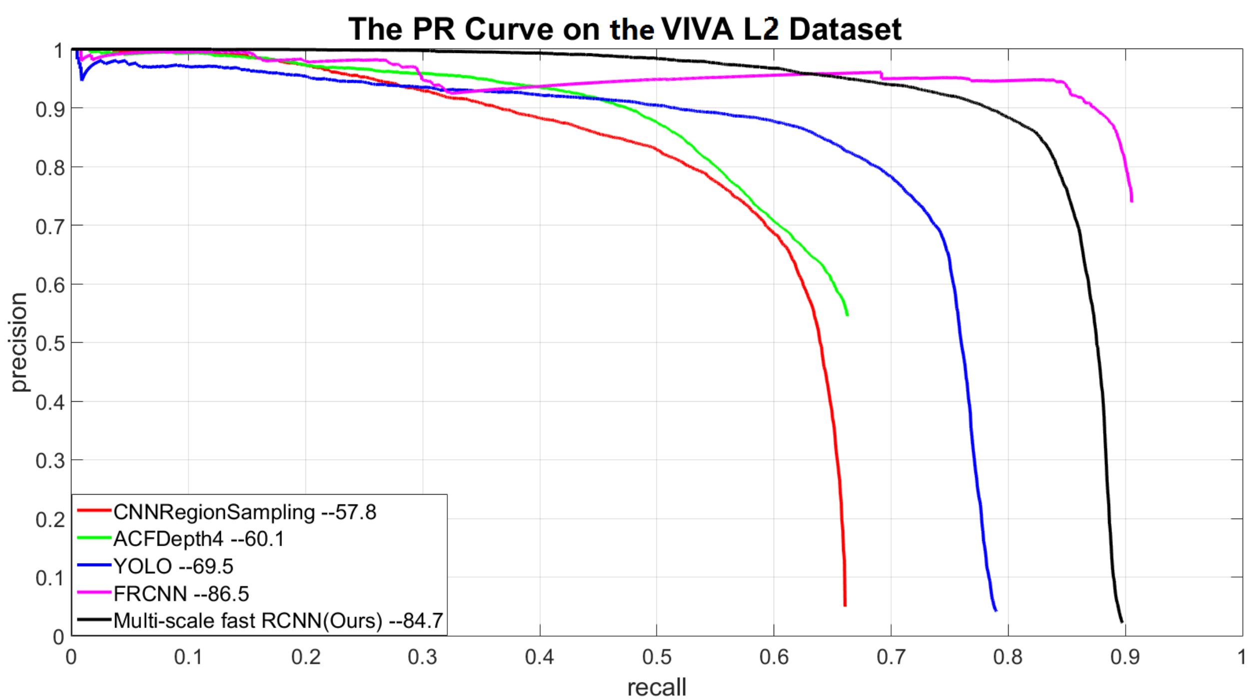
(b)

Fig. 11. *Recall of Edgeboxes algorithm on the VIVA hand detection dataset: (a) L1. (b) L2.*

To address these challenges, the dataset was designed to reflect variations in illumination, non-hand objects with similar color, occlusion and camera view-points. Fig. 10 (a) shows examples



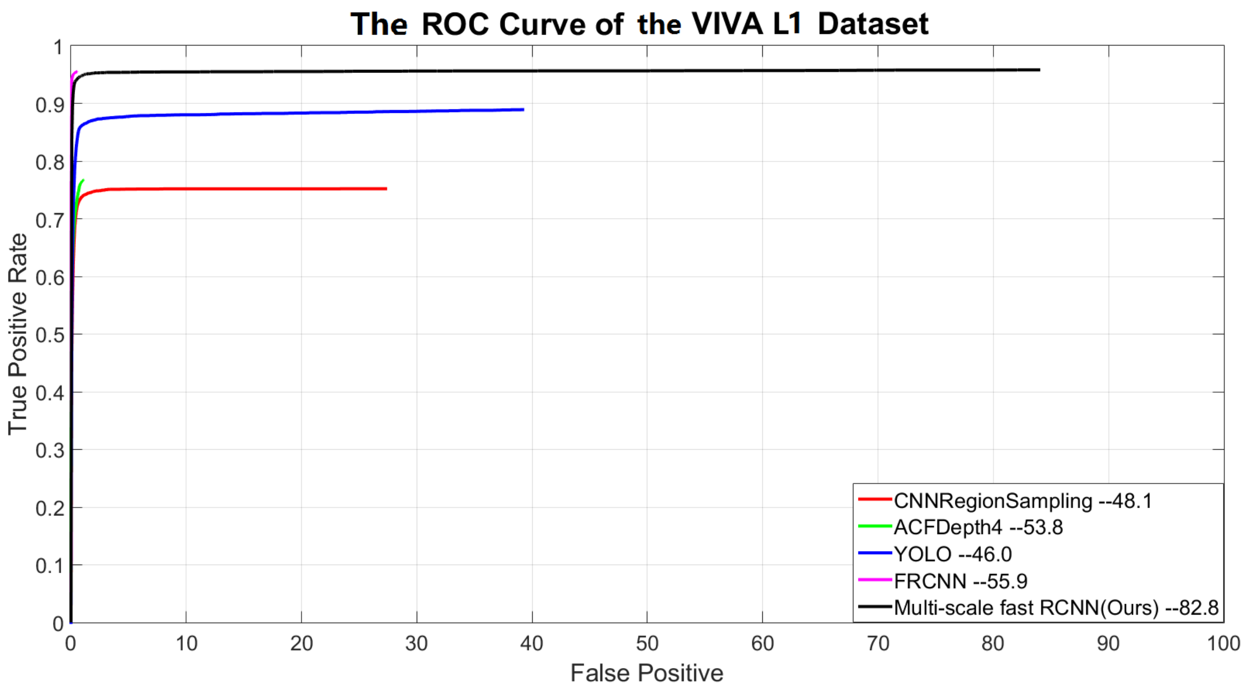
(a)



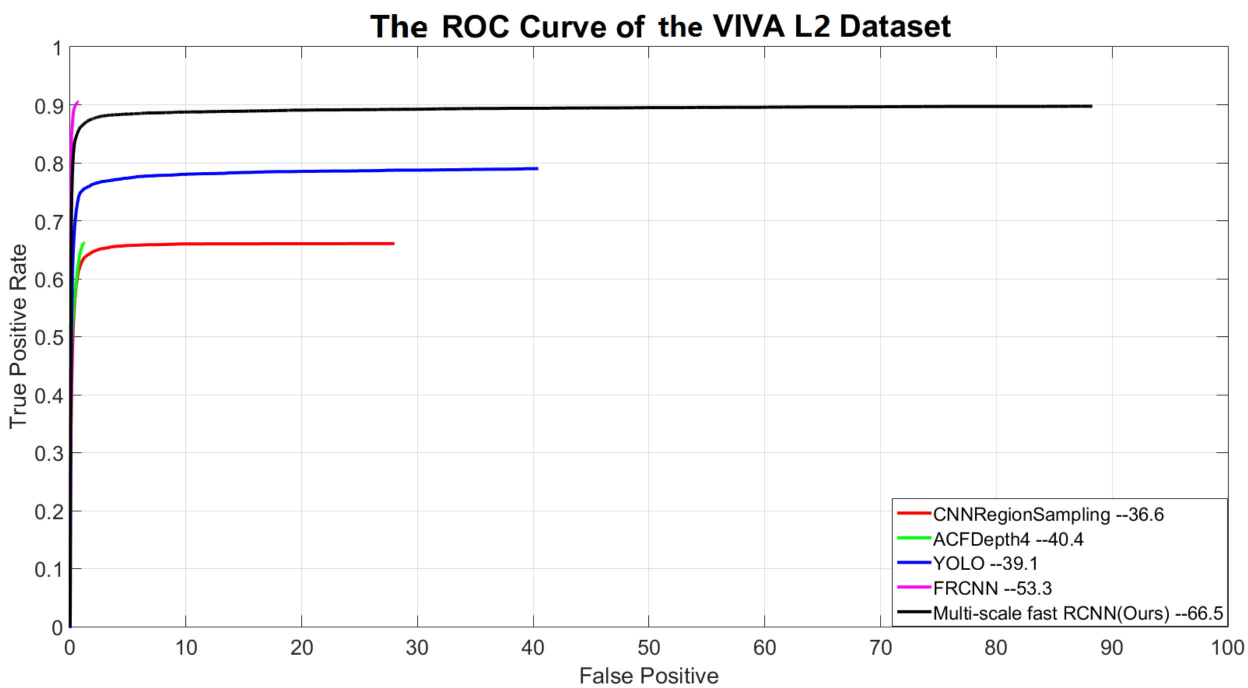
(b)

Fig. 12. *Precison-Recall curve on the VIVA hand detection dataset: (a) L1. (b) L2.*

of different view points, Fig. 10 (b) illustrates circumstances where skin-like non-hand objects appear in the image, Fig. 10 (c) demonstrates an occlusion example and Fig. 10 (d) is an example



(a)



(b)

Fig. 13. *ROC curve on VIVA the hand detection dataset: (a) L1. (b) L2.*

of illumination variation. The VIVA dataset is the first public dataset which can effectively evaluate the performance of a hand detection system inside a vehicle environment.

Table 4 Average Recall (AR) on VIVA L1 and L2 Dataset and comparison with previous methods.

Method	L1 Set	L2 Set
CNNRegionSampling [42]	48.1%	36.6%
ACF Depth4 [6]	53.8%	40.4%
YOLO [43]	46.0%	39.1%
FRCNN [44]	55.9%	53.3%
Our Model (Multi-scale Fast R-CNN)	**82.8%**	**66.5%**

The dataset includes two parts: the training set and the testing set, each with 5500 images. Whilst the annotations of training sets were released, we manually labelled the testing set for the subsequent experiments. The testing set can be further divided into two parts: Level-1 (L1) and Level-2 (L2). According to the dataset specification, L1 only includes the back view imagery and larger instances (above 70 pixels in height) while L2 comprises of imagery from all view points as well as instances larger than 25 pixels, which serves as a more difficult challenge. We will present results based on both of the subsets.

Similar to the experimental procedure in Section 4.1, after training of candidate regions generated by the Edgeboxes, during evaluation, we first generated a set of region proposals using the Edgeboxes algorithm and evaluated the performance by plotting the recall versus IoU curve, with the results shown in Fig.11. On the L2 dataset, the recall value is 90.0% with IoU 0.5, which is much smaller than the recall value of 97.7% on L1. This is consistent with the fact that L2 is more difficult than L1.

We then performed testing with our model. NMS with a threshold of 0.3 was also conducted to eliminate redundant bounding boxes. Fig.12 illustrates the PR curve for both of the L1 and L2 datasets. This PR curve indicates that our method (the black curve) ranks very highly in terms of the AP value (area under the PR curve). With AP values as the performance indicator, more comprehensive comparisons with results from applying other recently published methods are provided in Table 3. All the figures and values are from the official evaluation server. Among the compared methods, our approach (Multi-scale Fast R-CNN) showed satisfactory performance. Specifically, we achieved a state-of-the-art AP result on the L1 dataset, with a 92.8% AP value, and ranked second on the L2 dataset, with an 84.7% AP value.

As suggested by the challenge, we also utilized the Average Recall (AR) evaluation protocol [6],

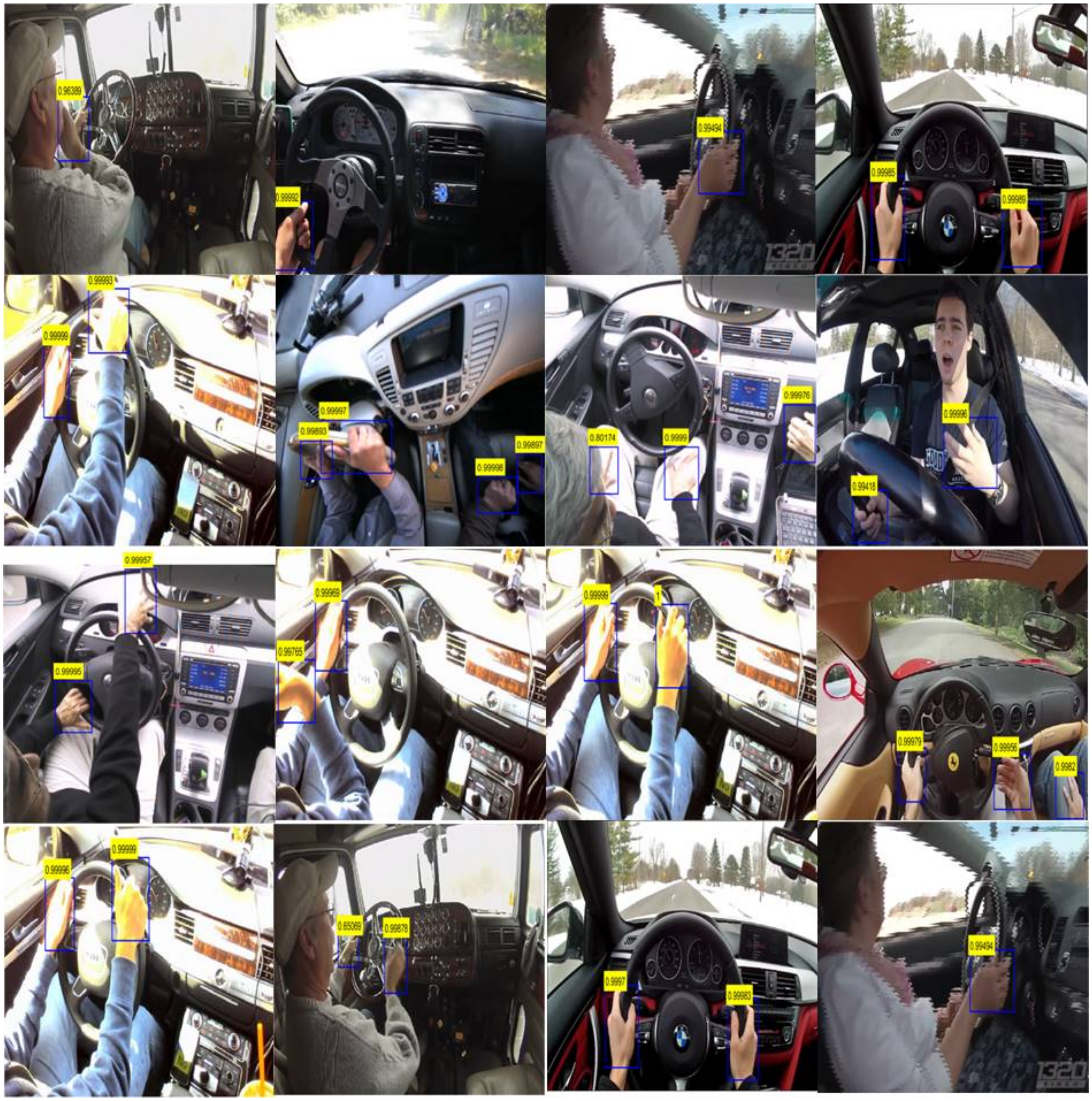


Fig. 14. *Correctly detected examples on the VIVA hand detection challenge: The red boxes are the annotated hand positions and the blue boxes are the detected boxes with corresponding label tags colored in yellow.*

AR was calculated from the ROC curve over 9 evenly sampled points in log space between 10^{-2} and 10^{0} false positives per image and suitable for summarizing the detection performance at lower false positive rates [6]. Fig.13 shows the ROC curve of our methods on the L1 and L2 datasets. From the figure, it is clear that the area under the curve of our method (black curve) ranks higher than other published results. Table 4 shows the AR results of our method and other participators'

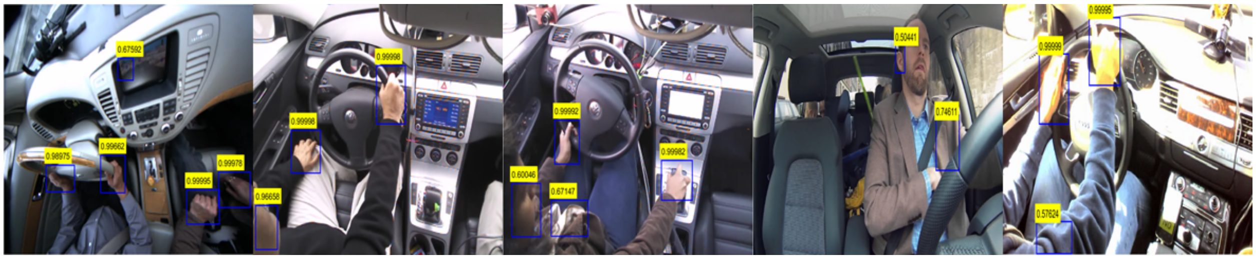


Fig. 15. *Incorrect examples on the VIVA dataset*

methods. Our method achieved 82.8% and 66.5% AR value on the L1 and L2 dataset, respectively, which are higher than all the other published results.

Fig.14 shows some of the correctly detected examples. Even with different types of variations including occlusions and re-scale, our proposed approach can correctly detect hands in most of the situations. Some unsuccessful examples are shown in Fig.15. Occasionally, certain kinds of cloth or part of the body such as an arm or face might be mistaken as hands. As we discussed at the end of section 4.1, this difficult task will be our next step in working towards developing a highly reliable hand detection system that is applicable in the real world.

5. Conclusion

This paper presented a multi-scale Fast R-CNN approach to accurately detect human hands in unconstrained images. By fusing multi-level convolutional features, our CNN model is able to achieve better results than the conventional VGG16 model. This method is especially efficient for small hand objects which are often hard to detect with conventional CNN models. Our methods have been validated on two benchmark datasets: the Oxford Hand Detection Dataset and the VIVA Hand Detection Challenge. On the Oxford dataset, we achieved state-of-the-art results with an improvement in performance by a significant margin; For the VIVA Hand Detection Challenge, our results have good performance as listed in the official website. Future work includes the fusion of contextual information to realize reliable hand detection, particularly for the environment inside a vehicle.

6. Statement

Statement: The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–511–I–518 vol.1, 2001.

[2] N. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *Instrumentation and Measurement, IEEE Transactions on*, vol. 60, pp. 3592–3607, Nov 2011.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893 vol. 1, June 2005.

[4] X. Meng, J. Lin, and Y. Ding, "An extended hog model: Schog for human hand detection," in *Systems and Informatics (ICSAI), 2012 International Conference on*, pp. 2593–2596, IEEE, 2012.

[5] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 8, pp. 1532–1545, 2014.

[6] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pp. 2953–2958, IEEE, 2015.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1627–1645, Sept 2010.

[8] A. Mittal, A. Zisserman, and P. H. Torr, “Hand detection using multiple proposals.,” in *BMVC*, pp. 1–11, Citeseer, 2011.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, pp. 1–42, 2014.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 38, pp. 142–158, Jan 2016.

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[15] R. Girshick, “Fast R-CNN,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 9, pp. 1904–1916, 2015.

[17] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*, pp. 354–370, Springer, 2016.

[18] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, pp. 818–833, Springer, 2014.

[19] S. Yang and D. Ramanan, "Multi-scale recognition with dag-cnns," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1215–1223, 2015.

[20] I. F. Ince, M. Socarras-Garzon, and T.-C. Yang, "Hand mouse: real time hand motion detection system based on analysis of finger blobs," *International Journal of Digital Content Technology and its Applications*, vol. 4, no. 2, 2010.

[21] G.-Z. Mao, Y.-L. Wu, M.-K. Hor, and C.-Y. Tang, "Real-time hand detection and tracking against complex background," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 905–908, IEEE, 2009.

[22] V. Chouvatut, C. Yotsombat, R. Sriwichai, and W. Jindaluang, "Multi-view hand detection applying viola-jones framework using samme adaboost," in *Knowledge and Smart Technology (KST), 2015 7th International Conference on*, pp. 30–35, IEEE, 2015.

[23] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[24] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[25] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[26] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou,

M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 2553–2561, Curran Associates, Inc., 2013.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 580–587, IEEE, 2014.

[28] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1879–1886, IEEE, 2011.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, pp. 1904–1916, Sept 2015.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv preprint arXiv:1506.02640*, 2015.

[32] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," *arXiv preprint arXiv:1512.04143*, 2015.

[33] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár, "A multipath network for object detection," *arXiv preprint arXiv:1604.02135*, 2016.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[35] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 447–456, 2015.

[36] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *CoRR*, vol. abs/1506.04579, 2015.

[37] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, pp. 675–678, ACM, 2014.

[38] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014.

[39] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What makes for effective detection proposals?," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[41] F. J. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms.," in *ICML*, vol. 98, pp. 445–453, 1998.

[42] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1949–1957, 2015.

[43] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[44] T. Zhou, P. J. Pillai, and V. G. Yalla, "Hierarchical context-aware hand detection algorithm for naturalistic driving," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1291–1297, Nov 2016.