Hand detection and segmentation by implementing R-CNN and Meanshift

Asal Rangraziasl
Asal.rangraziasl@studenti.unipd.it

Abstract

Hand detection and segmentation is an important task in computer vision with numerous applications in areas such as human-computer interaction and robotics. In this work, we propose a combined approach for hand detection and segmentation using Region-based Convolutional Neural Networks (RCNN) and Mean Shift Segmentation. The pre-processing step involves applying bilateralFilter and erosion on the images to reduce noise. In addition, Mean Shift Segmentation is applied to segment the objects in the image. The segmented objects are then fed as input to an RCNN model for hand detection. Results demonstrate that this approach gain acceptable accuracy in hand detection and segmentation in image processing.

1. Introduction

In this project, I proposed a robust hand-detection technique in image processing using deep learning. A variety of applications such as automatic sign language analysis [4], fine-grained action recognition [9], movie interpretation, and even for understanding dance gestures [6] rely on hand detection in image processing. Note that in this project, I just focus on detecting hands, not gesture. There are different method for detecting the objects. A convolutional neural network (CNN) is a type of deep learning network that is designed to work well with image data. It consists of multiple layers of convolutional filters, activation functions, and pooling layers that extract features from the input image. The output of the last layer is then fed into a fully connected layer that makes the final classification decision. On the other hand, R-CNN is a specific type of CNN that is designed for object detection tasks. It first uses a selective search algorithm or a region proposal network (RPN) to generate a set of candidate regions (or proposals) that might contain an object of interest. Then, these candidate regions are fed into a CNN to extract features, and the features are used to classify the regions to refine the bounding boxes around the objects.
In this paper we implement R-CNN (Regions with Convolutional Neural Networks) which is a popular computer vision method for object detection and classification. It was first introduced by Ross Girshick et al. in 2014[5]. The original R-CNN algorithm uses a combination of selective search and Convolutional Neural Networks (CNNs) to detect objects in images. For hand detection,we use the R-CNN algorithm by training it on a dataset of hand images. The proposed network learned to identify the features and patterns that are associated with hands that allows it to detect hand in new images.1

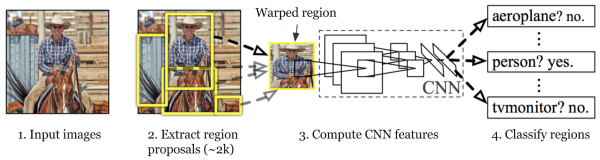


Figure 1: Structure of R-CNN

In addition, we used MeanShift, which is a non-parametric clustering technique that iteratively estimate the density of data points and to find the modes or cluster centers of the data distribution. MeanShift can be used to segment the foreground objects in an image based on color or intensity.[8, 7]
Thus we utilize combination of the MeanShift and RCNN to detect and segment hands in the images. In the proposed method MeanShift do the initial segmentation of the hand region based on color or intensity, and then I use RCNN to refine the hand segmentation by classifying the regions and refining the bounding boxes around the hands. This hybrid approach improves the accuracy and robustness of hand segmentation.

2. Related Work

Hand detection is a challenging task in computer vision, as the hands can appear in a wide range of poses, scales, and occlusions, and can be easily confused with other objects or parts of the body. Over the years, researchers have proposed various methods for hand detection.

Md Jahangir et al[2]resented a technique for hand detection which is Skin color methods. It use statistical models of skin color to identify hand regions in the image. These models typically assume that the skin color follows a Gaussian distribution and use a color space.
In addition,Xinyu et al.[10] try another technique. It is HOG-based methods. These methods use Histograms of Oriented Gradients (HOG) features to describe the local shape and texture of the hands and then use a classifier, such as a SVM or a Random Forest, to make the final detection decision.
In recent years, deep learning-based methods, especially RCNN (Regional Convolutional Neural Network) and YOLO (You Only Look Once)[3], have become increasingly popular for hand detection due to their ability to learn rich representations of hand appearance and shape from large datasets and to detect hands in real-time. These methods have achieved state-of-the-art performance on various hand detection benchmarks. In this paper, we will use RCNN which can be trained to detect hand regions and to segment the hands from the background by using the mean-shift for improving performance.

3. Dataset

For the dataset, we used EgoHands [1] which is A dataset for Hands in Complex Egocentric Interactions. The Ego-Hands dataset contains 48 Google Glass videos of complex, first-person interactions between two people. The main intention of this dataset is to enable better, data-driven approaches to understanding hands in first-person computer vision.
In order to train the neural network, the ground truth of the images is needed(coordinates of the bounding box around each hand). Since there are many identical images, 41 of the images are chosen from all kinds of hands, and XML files containing the coordinates of the hands in each image are created. Mention that, I first chose 483 images that contain hand, but after seeing the final results(images with bounding boxes), my results were too bad because of not appropriate pictures. So I have decided to re-select my images and have the selection of 41 of them.
In addition, a selective search algorithm is applied to each image.Selective Search is a computer vision algorithm used for object recognition and image segmentation. It is a bottom-up approach to object recognition that starts by dividing an image into smaller regions (called segments or regions of interest which is hand in our project) and then grouping the segments that are likely to belong to the same object. The result which has an enough IoU score is compared to the hand coordinates is labeled as “hand” 2 and the results which have an IoU score less than a Negative threshold are labeled as “no-hand” 3. Although exact hand images can be extracted from the full images with coordinates from ground truth, this step is done in order to label hands that are incomplete in the proposals and make a variety of no-hands images. Finally, our dataset consists of 179 images labeled as hand and 450 images labeled as no-hand. The size of all images are the same: (224, 224)

Figure 2: Examples of hands in dataset

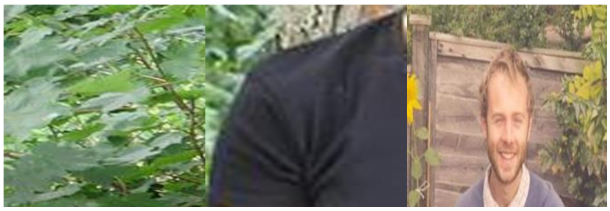
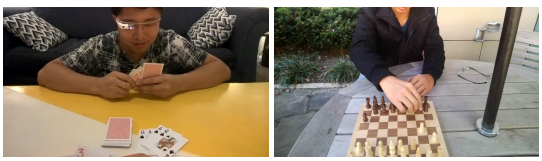


Figure 3: Examples of non-hands in dataset

4. Method

The proposed methods structure is to use the Ego-Hands dataset to train a Region-based Convolutional Neural Network (R-CNN) for hand detection. R-CNN is a popular object detection algorithm that uses a combination of region proposal and deep learning techniques to detect objects in images.
To train an R-CNN on the EgoHands dataset, the annotated hand instances would be used as positive samples, while the background regions would be used as negative samples. The network would be trained to minimize the classification error on these samples and learn to recognize hand instances in new images. Once the R-CNN is trained, it can be used to detect hand instances in new images.

As we mentioned in introduction, in the context of hand detection, MeanShift segmentation can be used to segment the hand instances from the background in an image. In comparision with other methods Meanshift has better per-



.

Figure 4: Examples of test dataset

formance due to its robustness to scale and orientation changes, handling of non-uniform illumination, ease of use, and real-time performance.
In the first approach, we use R-CNN with pre-trained weights. A simple R-CNN algorithm is choosing an array of proposals with selective search and then passing it to a convolutional neural network. The pre-trained part of the neural network architect used for this project is "VGG16". The VGG-16 is one of the most popular pre-trained models for image classification which contains 13 Convolutional Layers , 5 Pooling Layers and 3 Dense Layers also has 2 output in labeled "hand" and "no-hand".The architecture of the pre-trained models studied in this. link
We also create Augmented images from our dataset. In computer vision tasks such as object detection, augmentation can help to reduce overfitting, improve generalization, and increase the model's ability to handle variations in the image data. For example, by generating rotated or scaled versions of the original images, the model can learn to recognize objects from different viewpoints and scales.
When images are loaded for training the neural network, data augmentation is applied to the dataset. So, not only the pure image, but also oriented, shifted, flipped, and color changed versions of the images are passed through the neural network.
In the second approach we implement Meanshift. There are lots of popular algorithms for object segmentation include K-means clustering, Gaussian Mixture Models (GMM), and GrabCut but MeanShift is a fast and simple segmentation algorithm that can handle a variety of hand poses and scales and one of several algorithms that can be used for object segmentation in computer vision, including hand detection. It is suitable for real-time hand detection applications and is relatively robust to noise and clutter. Mean Shift is a non-parametric density-based clustering algorithm and the goal is to divide an image into regions such that pixels within each region are similar in terms of some similarity measure, such as color or texture. The basic idea behind Mean Shift is to shift the position of each data point (each pixel) towards the mean (center) of its local neighborhood, until convergence.

5. Experiment

In this section we will describe the detail of parameters for our setup and the achieved results. This project mainly focuses on object detection and segmentation. For better understanding we divided our work into different steps.

5.1. Object detection with selective search

As we mentined before selective search is a method for generating a set of candidate regions, or "proposals," that could potentially contain an object of interest. In addition we measure the object detection accuracy with Intersection over Union (IoU)5 which is a common metric used in computer vision and object detection to evaluate the overlap between a predicted bounding box and a ground truth bounding box. Selective search algorithm may choose several proposals from the same region of interest and the neural network detects all of them as hands. For this reason we apply non max suppression algorithm.

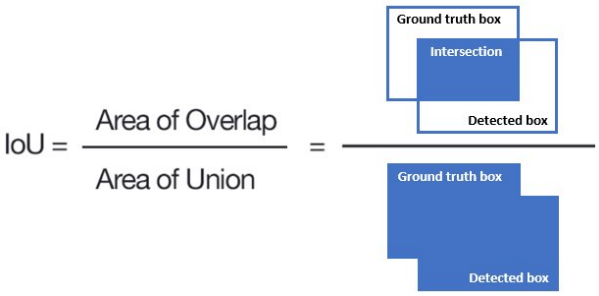


Figure 5: Intersection over Union

5.2. Fine-tune classification model on dataset:

in this part we are ready to fine-tune a classification CNN to recognize both of classes (hand, no-hand). We then establish training hyper-parameters including RMSprop learning rate (0.0006), number of training epochs (15), and batch size (128). Precision (1), Recall (2), and F1-score (3) are used to evaluate the classification model performance. A plot line which demonstrate the accuracy and loss performances of the model.6

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (1)$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (2)$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3)$$

	precision	recall	f1-score	support
hand	0.71	0.96	0.82	54
no_hand	0.98	0.85	0.91	136
accuracy			0.88	190
macro avg	0.85	0.90	0.86	190
weighted avg	0.91	0.88	0.88	190

Figure 6: Class, Precision, Recall, F1-Score

For implementation of model VGG16 (proposed method), we used RMSprop as optimizer with the learning

rate of 0.0006 and as a loss function we used Categorical Cross Entropy. In 11th epoch, our model achieved 93.23% and 91.05% train and test accuracy, respectively. As we can see from Figure 7.

Figure 7: Training and Validation accuracy

5.3. Hand segmentation part

In the segmentation part we develop a function to apply the meanshift to the dataset. The algorithm iteratively updates the mean color and intensity values of each cluster, until the mean values converge to a stable solution. In this model we have mutiple steps. First, The bilateral filter apply as a pre-processing step to reduce noise in the image and preserve important features, such as edges and fine details which are useful for object detection.
second, we apply the meanshift segmentation with the kernel size which defined in our model which are 5,45, and 65.

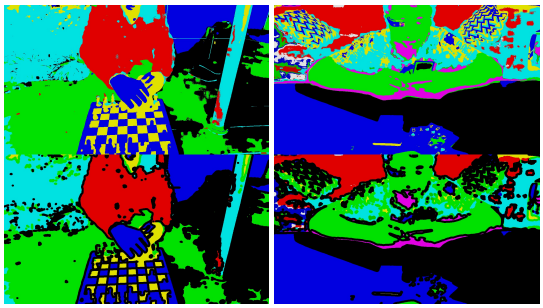


Figure 8: Example of Erosion output -kernel size 5

In most of the images after segmentation section, a lot of unmeaningful segments in the output picture appeared. As an example, a hand is segmented into multiple sub-segments which make too unusual proposals. For this problem, erosion could remove many of the small segments. The Erosion for hand detection can reduce the size of bright regions in an image and remove any small, noisy pixels that may be present. This can help to separate the hand from the background and improve the accuracy of hand detection.

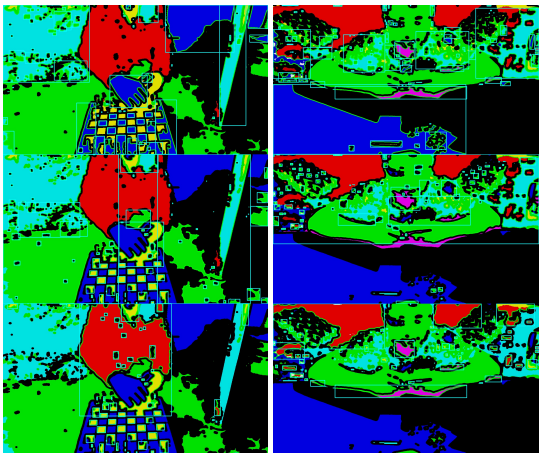


Figure 9: Example of Contour output -kernel size 5

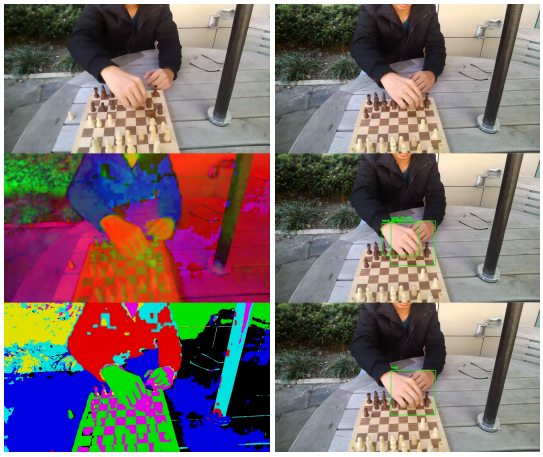
Finally, for finding boxes over the segments, we apply "find-Contours" which finds the edge line of each segment and by finding the minimum and maximum of coordinates of points of the edges with the kernel size which defined in our model are 5,45, and 65.

5.4. Return final detection

This is the final step in our project. Here we combine Mean Shift Segmentation and Region-based Convolutional Neural Networks (RCNN) to get the result of hand detection. We impelement Pre-processing by adding Gaussian blur to the input image to reduce noise. After that Perform Mean Shift Segmentation on the pre-processed image to segment the objects in the image. Then we apply a RCNN model to detect the hands in the segmented objects.10 and 11 are the output of final model.

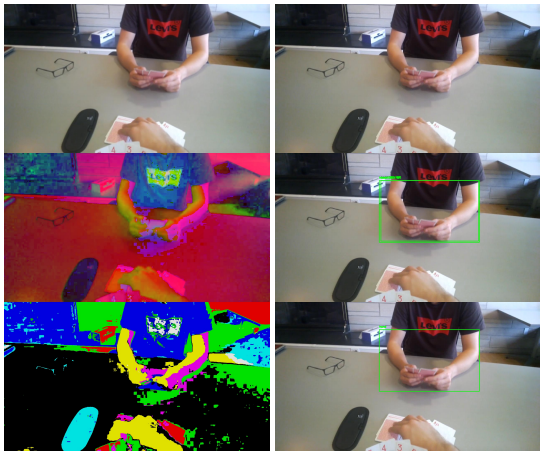
6. Concolusion

Combining Mean Shift Segmentation and Region-based Convolutional Neural Networks (RCNN) is a possible approach to detect hands in an image. This is just a general outline, and the specific implementation may vary depending on the particular use case and requirements. Also, this approach may not be ideal for every situation as it involves multiple steps and can be computationally expensive. How-



.

Figure 10: Final result of R-CNN by meanshift segmentation by kernel size 65 for hand detection



.

Figure 11: Final result of R-CNN by meanshift segmentation by kernel size 5 for hand detection

ever, it can provide good results for hand detection in certain scenarios. It's important to keep in mind that developing and training an RCNN model for hand detection is a complex task that requires a large amount of labeled data, computational resources, and expertise in deep learning. To use an RCNN model for hand detection, we train the model by using a pre-trained model VGG16 that has already been trained on a dataset.

References

[1] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[2] Ahmad Yahya Dawod, Junaidi Abdullah, and Md Jahangir Alam. Adaptive skin color model for hand segmentation. In *2010 International Conference on Computer Applications and Industrial Electronics*, pages 486–489. IEEE, 2010.

[3] Christine Dewi and Henoch Juli Christanto. Combination of deep cross-stage partial network and spatial pyramid pooling for automatic hand detection. *Big Data and Cognitive Computing*, 6(3):85, 2022.

[4] Ali Farhadi and David Forsyth. Aligning asl for statistical translation using a discriminative word model. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1471–1476. IEEE, 2006.

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[6] Aparna Mohanty, Alfaz Ahmed, Trishita Goswami, Arpita Das, Pratik Vaishnavi, and Rajiv Ranjan Sahay. Robust pose recognition using deep learning. In *Proceedings of International Conference on Computer Vision and Image Processing: CVIP 2016, Volume 2*, pages 93–105. Springer, 2017.

[7] Yainuvis Socarrás Salas, David Vázquez Bermudez, Antonio M López Peña, David Gerónimo Gomez, and Theo Gevers. Improving hog with image segmentation: Application to human detection. In *Advanced Concepts for Intelligent Vision Systems: 14th International Conference, ACIVS 2012, Brno, Czech Republic, September 4-7, 2012. Proceedings 14*, pages 178–189. Springer, 2012.

[8] Wenbing Tao, Hai Jin, and Yimin Zhang. Color image segmentation based on mean shift and normalized cuts. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1382–1389, 2007.

[9] Yezhou Yang, Cornelia Fermuller, Yi Li, and Yiannis Aloimonos. Grasp type revisited: A modern perspective on a classical feature for vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 400–408, 2015.

[10] Yanguo Zhao, Zhan Song, and Xinyu Wu. Hand detection using multi-resolution hog features. In *2012 IEEE international conference on robotics and biomimetics (ROBIO)*, pages 1715–1720. IEEE, 2012.