

UNIVERSITY OF PADUA

INFORMATION ENGINEERING DEPARTMENT (DEI)

MASTER'S DEGREE IN COMPUTER ENGINEERING

# Computer Vision Report

Prof.  
Stefano Ghidoni

Students:  
Francesco Caldivizzi  
Simone D'Antimo  
Daniela Cuza

# Contents

<b>1</b>	<b>Hand Detection</b>	<b>2</b>
<b>2</b>	<b>Hand Segmentation</b>	<b>4</b>
2.1	Data Augmentation . . . . .	4
2.1.1	Contrast and Motion Blur . . . . .	4
2.1.2	Shadows . . . . .	5
2.2	Unused segmentation . . . . .	6
<b>3</b>	<b>Results</b>	<b>10</b>
3.1	Ouput Images . . . . .	10
3.2	Intersection Over Unions . . . . .	11
3.3	Pixel Accuracy . . . . .	14
<b>4</b>	<b>Contribution</b>	<b>15</b>
4.1	Hours of work . . . . .	15
<b>5</b>	<b>Conclusions</b>	<b>16</b>
5.1	Results Discussions . . . . .	16
5.2	Future Works . . . . .	16

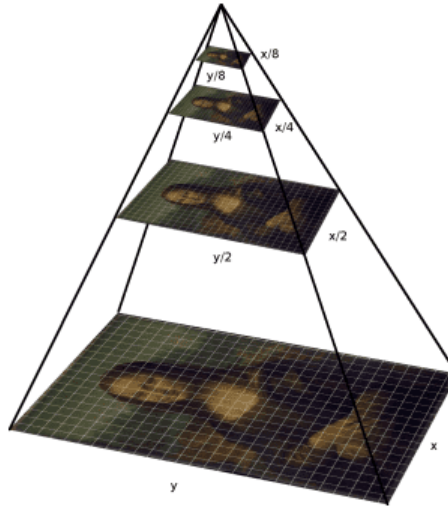
# Chapter 1

## Hand Detection

In this project, the key idea is to convert an image classifier into an object detector. In particular, we explore a pre-trained ResNet50V2 CNN trained on the ImageNet dataset. Moreover, we make use of image pyramid and sliding windows. Regarding image pyramid, it consists of different layers, each one representing an image at a different scale and usually a smoothing filter is applied. In our code, we implement *getGaussianPyramid(cv::Mat image)* that computes image pyramid by employing a gaussian filter.

Another essential tool in our work is sliding windows. A sliding window is composed by a rectangle of a given size that we translate from left-to-right and top-to-bottom into an image. For each window, we compute the subsequent steps:

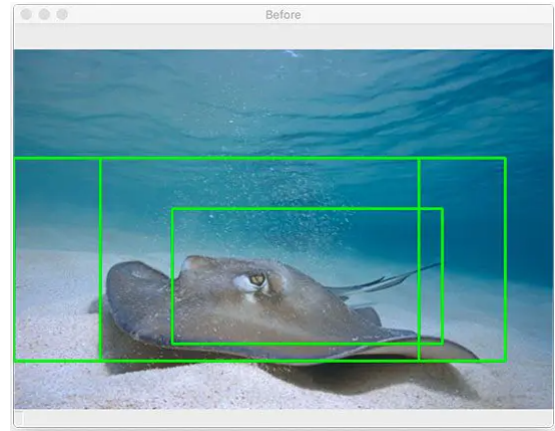
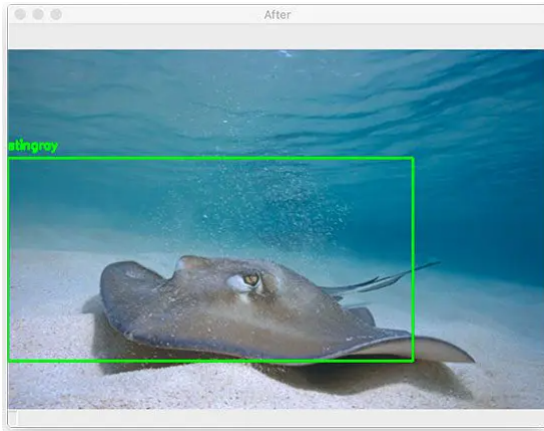
1. Obtain the region of interest
2. Apply the image classifier to the ROI
3. Get the final predictions, together with the confidence score



Thanks to image pyramid and sliding windows, we can consider a rectangle at various locations and various scales of the input image. The problem with this procedure are the different bounding boxes surrounding the same object. To deal with this issue, we apply non-maxima suppression. This method allows to maintain only the stronger bounding box, eliminating the extraneous ones. Consider the set  $P$  of all the predicted bounding boxes, comprehensive of the confidence score and the overlap threshold, the non-maxima suppression approach goes as follow:

1. Delete from  $P$  the bounding box  $S$  with the top confidence score, and add it to a list  $F$ , initially empty, representing the predictions that we maintain

2. Compute IoU between S and any other prediction T present in P. Delete T from P only if each resulting IoU is greater than the overlap threshold.
3. If P is empty return F and stop, otherwise compute step 1 and step 2



Finally, the last step applied is one that is used to remove occlusions. Such step consist of removing the bounding boxes obtained after Non Maxima Suppression that may not contain an hand. In particular such step consist of, for each bounding box after Non Maxima Suppression (notice that, this step can be only applied if the original image was not a grayscale image) :

1. Get the ROI identified by selected bounding boxes
2. Convert the ROI to an image into the color space YCrCb
3. Threshold the ROI with a certain range of colors (range of the skin color)
4. Compute  $percentage = \frac{\#pixels\ equal\ to\ 255\ in\ thresholded\ image}{image\ size}$
5. Reject the bounding box if  $percentage \leq threshold$

The detection task is quite time-consuming, principally do to the sliding window method. As future work, we suggest to reduce the computational complexity by using an alternative approach of sliding window.

# Chapter 2

## Hand Segmentation

We based our segmentation approach on DeepLabV3+ using ResNet18 as a backbone and perform the training in matlab. We combine four existing datasets: hand gesture dataset, hand over face dataset, egohand dataset and GTEA dataset. The resulting dataset was cleaned by removing images containing arm parts and test set images. Our final dataset is composed by 1003 images. The loss function employed in this project is the dice loss.

### 2.1 Data Augmentation

We use a data augmentation technique with the aim of increasing the size of the starting dataset and therefore increasing the performance of the classification system. We use both shape-based and color-based transformation and in particular eleven artificial images are created for each image in the dataset. The operations performed are:

1. The image is displaced to the right or the left.
2. The image is displaced up or down.
3. The image is rotated by an angle randomly selected from the range  $[0^\circ 180^\circ]$ .
4. Horizontal or vertical shear by using the function *randomAffine2d*.
5. Horizontal or vertical flip.
6. Change in the brightness levels by adding the same value to each RGB channel.
7. Change in the brightness levels by adding different values to each RGB channel.
8. Add speckle noise by using the function *imnoise*.
9. Conversion of the truecolor image RGB to the grayscale image.
10. Application of the technique “Contrast and Motion Blur”, described below.
11. Application of the technique “Shadows”, described below.

#### 2.1.1 Contrast and Motion Blur

This transformation is the composition of two alterations: firstly, it is necessary to modify the contrast of the original image, increasing or decreasing it, then a filter that simulates the movement of the camera is

applied. Two functions to modify the contrast are implemented, but only one randomly chosen between the two is applied to the image. The first contrast function is based on the following equation:

$$\frac{(x - \frac{1}{2})\sqrt{1 - \frac{k}{4}}}{\sqrt{1 - k(x - \frac{1}{2})^2}} + 0.5, k \leq 4$$

The parameter  $k$  controls the contrast, there is an increase in contrast if  $k < 0$ , a decrease in contrast if  $0 < k \leq 4$ , the image remains the same when  $k = 0$ . In the code,  $k$  is chosen randomly from a specific range, randomly chosen among the following four:

- $U(2.8, 3.8) \rightarrow$  Hard decrease in contrast.
- $U(1.5, 2.5) \rightarrow$  Soft decrease in contrast.
- $U(-2, -1) \rightarrow$  Soft increase in contrast.
- $U(-5, -3) \rightarrow$  Hard increase in contrast.

The second contrast function is based on the following equation:

$$y = \begin{cases} \frac{1}{2}(\frac{x}{0.5})\alpha, & 0 \leq x < \frac{1}{2} \\ 1 - \frac{1}{2}(\frac{1-x}{0.5})\alpha, & \frac{1}{2} \leq x \leq 1 \end{cases}$$

The parameter  $\alpha$  controls the contrast, there is an increase in contrast if  $\alpha > 1$ , a decrease in contrast if  $0 < \alpha < 1$  and the image remains the same when  $\alpha = 1$ . This parameter is chosen randomly from four possible ranges:

- $U(0.25, 0.5) \rightarrow$  Hard decrease in contrast.
- $U(0.6, 0.9) \rightarrow$  Soft decrease in contrast.
- $U(1.2, 1.7) \rightarrow$  Soft increase in contrast.
- $U(1.8, 2.3) \rightarrow$  Hard increase in contrast.

## 2.1.2 Shadows

The final image is obtained by applying a shadow to the left or the right of the original image. In particular, the intensities of the columns are multiplied by the following equation:

$$y = \begin{cases} \min \left\{ 0.2 + 0.8\sqrt{\frac{x}{0.5}}, 1 \right\} & direction = 1 \\ \min \left\{ 0.2 + 0.8\sqrt{\frac{1-x}{0.5}}, 1 \right\} & direction = 0 \end{cases}$$



Figure 2.1: The original image (top left) and the artificial images.

## 2.2 Unused segmentation

Before we decide to implement the segmentation using a CNN we try to investigate how to solve the problem using only the OPENCV library method. We spent a lot of time trying to implement such a solution, so we want to dedicate a little section in our report to details how far we were able to arrive, and the problems that made us change direction.

After many tries we found a solution that basically consist in:

1. Application of bilateral filter to the image to smooth the image
2. Applying a single pixel (threshold) operation to remove pixels that we are sure to not be skin pixels, and color them black. If you are outside this range we discard this pixel and color it black.

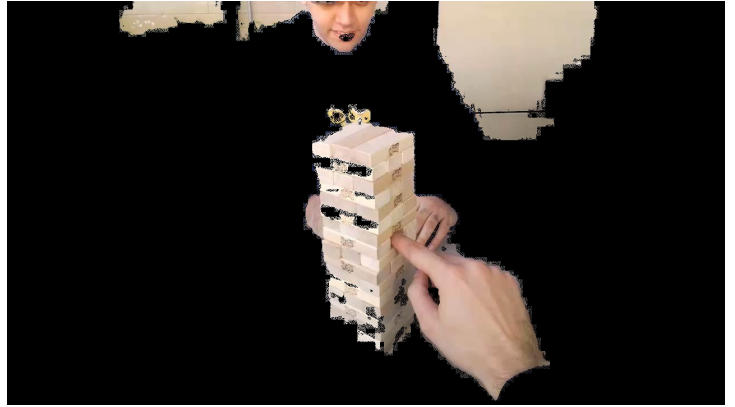
```
if ((R > 95 && G > 40 && B > 20 && (max_value - min_value > 10) && abs(R - G) > 10 && R > G && R > B)
|| (R > 220 && G > 210 && B > 170 && abs(R - G) <= 30 && R > B && G > B)) {
```

where B is the blue channel, R is the red channel and G is the green channel. max value and min value are respectively the max and min among the R,G,B values.

All the details and the motivation behind this choice are explained here

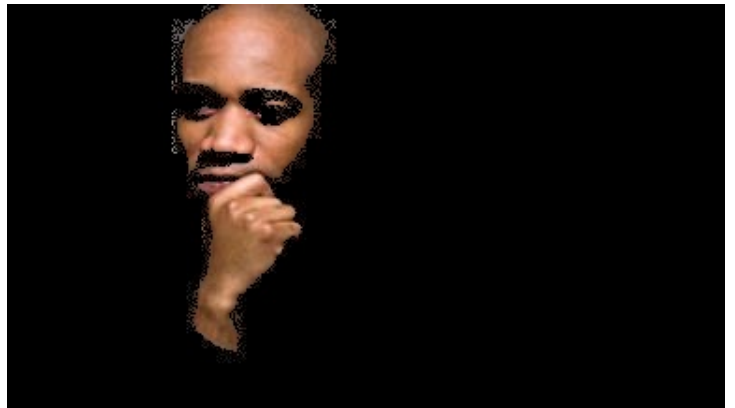
3. From the filtered image, we apply a findContours function to find areas (not too big) surrounded by black pixels and discard also the pixel contained in the area.

Since this section wants to give an idea of what we have done, the function that eliminates the background can work on images of arbitrary size, so we may have a more precise result applying the filter on just the portion of image inside the bounding boxes where hands are located, but to understand better the results given by the filter we are just applying it to the whole image. We now display and comment the results obtained: First of all, we can say that for an image with a background with color different from skin the filter works well.



Applying the filter to this image creates a very good result, applying the image only on bounding box portion would create an almost perfect segmentation of the hand (Based obviously on the accuracy of the bounding box, but we can state that with a good “cut” we can create an image with the full hand colored plus some piece of “Jenga” contained in the , which is perhaps impossible to eliminate since is almost the same color of the hand).

Another important pros of our filter is that it works well with different skin tone:





Talking about cons of our application we have a strong sensitive to shady portion of hands:



As you can see clearly from the image above, basically all the parts of the hand that are “in the shade” are eliminated, and unfortunately, using a more permissive threshold would allow shady part of images but also other parts of the background which would be labelled wrongly. The threshold used is one of the best trade-off between false positive and false negative that we were able to find. The last but not least big problem that we recognized, and you can also see from the images above is the impossibility to eliminate the other “skin part” of a person, like arms or faces:



As you can see here the filter application (even in the portion of bounding box) would color not only the hand but also all the other portion of the skin inside the image, and this lead to a poor result for many of the “Hand over faces” images and for the images where arms is not covered.

Now to let the reader understand what would be an hypothetical result applying our filter to a bounding box we display a sample using a ”perfect” bounding-box from “Hands over face” dataset, and one can note that it doesn’t lead to a good solution. This time, instead of displaying the discarded pixel as black, we will display the image segmented coloring the pixel belonging to the mask (In red)



The problems related to shady parts and other skin parts of the image plus the fact that we cannot guarantee that we would apply the segmentation on a good bounding box let this solution being too situational and doesn’t guarantee any “lower bound” of quality for the image segmentation. This forces us to search for a better solution.

# Chapter 3

## Results

### 3.1 Ouput Images

Output Images can be downloaded with this **link**

## 3.2 Intersection Over Unions

Here will be shown the values of IOUs computed for each image :

Name Image	Intersection Over Union		
01.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 667 Y: 249 W: 166 H: 166	X: 631 Y: 318 W: 217 H: 217	0.428918
	X: 504 Y: 252 W: 168 H: 168	X: 453 Y: 308 W: 175 H: 175	0.367544
02.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 588 Y: 252 W: 168 H: 168	X: 641 Y: 322 W: 133 H: 133	0.355330
	X: 504 Y: 336 W: 168 H: 168	X: 518 Y: 339 W: 201 H: 201	0.702322
03.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 504 Y: 252 W: 168 H: 168	X: 727 Y: 322 W: 153 H: 153	0.000000
	X: 504 Y: 252 W: 168 H: 168	X: 493 Y: 328 W: 162 H: 162	0.452597
04.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 667 Y: 333 W: 166 H: 166	X: 682 Y: 347 W: 138 H: 138	0.530846
	X: 504 Y: 252 W: 168 H: 168	X: 489 Y: 331 W: 172 H: 172	0.430163
05.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 672 Y: 252 W: 168 H: 168	X: 669 Y: 316 W: 151 H: 151	0.529127
	X: 504 Y: 336 W: 168 H: 168	X: 483 Y: 312 W: 201 H: 201	0.712781
06.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 672 Y: 252 W: 168 H: 168	X: 620 Y: 414 W: 235 H: 235	0.012332
	X: 750 Y: 249 W: 167 H: 167	X: 730 Y: 273 W: 181 H: 181	0.508381
	X: 420 Y: 252 W: 168 H: 168	X: 437 Y: 224 W: 125 H: 125	0.492529
07.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 756 Y: 420 W: 168 H: 168	X: 740 Y: 413 W: 400 H: 400	0.283374
	X: 588 Y: 168 W: 168 H: 168	X: 627 Y: 221 W: 164 H: 164	0.439881
	X: 333 Y: 166 W: 167 H: 167	X: 365 Y: 190 W: 154 H: 154	0.606553
08.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 420 Y: 336 W: 168 H: 168	X: 627 Y: 314 W: 247 H: 247	0.000000
	X: 420 Y: 336 W: 168 H: 168	X: 430 Y: 339 W: 187 H: 187	0.663680
09.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 672 Y: 504 W: 168 H: 168	X: 668 Y: 533 W: 109 H: 109	0.341481
	X: 325 Y: 324 W: 166 H: 166	X: 406 Y: 490 W: 98 H: 98	0.000000
10.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 650 Y: 243 W: 166 H: 166	X: 608 Y: 276 W: 211 H: 211	0.595996
	X: 406 Y: 243 W: 166 H: 166	X: 421 Y: 234 W: 143 H: 143	0.813092
11.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 756 Y: 252 W: 168 H: 168	X: 557 Y: 545 W: 276 H: 276	0.000000
	X: 756 Y: 252 W: 168 H: 168	X: 734 Y: 325 W: 121 H: 121	0.297203
	X: 756 Y: 252 W: 168 H: 168	X: 598 Y: 302 W: 121 H: 121	0.000000
12.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 583 Y: 499 W: 167 H: 167	X: 68 Y 469 W: 375 H: 375	0.000000
	X: 583 Y: 499 W: 167 H: 167	X: 536 Y: 430 W: 316 H: 316	0.311860
	X: 583 Y: 499 W: 167 H: 167	X: 525 Y: 93 W: 162 H: 162	0.000000
	X: 336 Y: 84 W: 168 H: 168	X: 385 Y: 96 W: 113 H: 113	0.408376

13.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 333 Y: 333 W: 167 H: 167	X: 285 Y: 611 W: 229 H: 229	0.000000
	X: 588 Y: 420 W: 168 H: 168	X: 554 Y: 365 W: 226 H: 226	0.502155
	X: 333 Y: 333 W: 167 H: 167	X: 299 Y: 305 W: 161 H: 161	0.533938
14.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 336 Y: 420 W: 168 H: 168	X: 256 Y: 477 W: 235 H: 235	0.273295
	X: 650 Y: 486 W: 166 H: 166	X: 659 Y: 473 W: 202 H: 202	0.564895
	X: 756 Y: 336 W: 168 H: 168	X: 757 Y: 391 W: 186 H: 186	0.536001
	X: 420 Y: 336 W: 168 H: 168	X: 459 Y: 312 W: 153 H: 153	0.583333
15.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 672 Y: 252 W: 168 H: 168	X: 617 Y: 366 W: 214 H: 214	0.153860
	X: 924 Y: 336 W: 168 H: 168	X: 893 Y: 337 W: 158 H: 158	0.522181
	X: 672 Y: 252 W: 168 H: 168	X: 654 Y: 271 W: 152 H: 152	0.512800
16.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 917 Y: 166 W: 166 H: 166	X: 264 Y: 670 W: 61 H: 61	0.000000
	X: 1008 Y: 336 W: 168 H: 168	X: 873 Y: 142 W: 405 H: 405	0.159837
	X: 917 Y: 166 W: 166 H: 166	X: 633 Y: 219 W: 162 H: 162	0.000000
	X: 420 Y: 84 W: 168 H: 168	X: 440 Y: 127 W: 160 H: 160	0.507814
17.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 650 Y: 486 W: 166 H: 166	X: 654 Y: 482 W: 432 H: 432	0.262064
	X: 333 Y: 166 W: 167 H: 167	X: 389 Y: 156 W: 122 H: 122	0.599011
	X: 81 Y: 162 W: 166 H: 166	X: 97 Y: 181 W: 181 H: 181	0.567167
18.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 250 Y: 499 W: 166 H: 166	X: 150 Y: 430 W: 360 H: 360	0.269250
	X: 500 Y: 499 W: 167 H: 167	X: 542 Y: 456 W: 255 H: 255	0.280827
	X: 924 Y: 252 W: 168 H: 168	X: 917 Y: 192 W: 207 H: 207	0.543218
	X: 924 Y: 420 W: 168 H: 168	X: 880 Y: 346 W: 261 H: 261	0.356891
19.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 672 Y: 420 W: 168 H: 168	X: 619 Y: 390 W: 295 H: 295	0.324321
	X: 756 Y: 420 W: 168 H: 168	X: 785 Y: 447 W: 130 H: 130	0.580357
	X: 168 Y: 504 W: 168 H: 168	X: 187 Y: 479 W: 174 H: 174	0.600541
20.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 672 Y: 420 W: 168 H: 168	X: 686 Y: 413 W: 266 H: 266	0.367250
	X: 672 Y: 336 W: 168 H: 168	X: 729 Y: 342 W: 77 H: 77	0.270089
	X: 672 Y: 336 W: 168 H: 168	X: 544 Y: 348 W: 22 H: 22	0.000000
21.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 231 Y: 112 W: 76 H: 76	No Detections	0.000000
22.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 0 Y: 100 W: 100 H: 100	X: 145 Y: 41 W: 38 H: 38	0.000000
	X: 0 Y: 100 W: 100 H: 100	X: 15 Y: 168 W: 62 H: 62	0.194777
23.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 50 Y: 100 W: 100 H: 100	X: 94 Y: 96 W: 81 H: 81	0.418931
24.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 100 Y: 50 W: 100 H: 100	X: 110 Y: 81 W: 76 H: 76	0.421815
25.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 100 Y: 100 W: 100 H: 100	X: 89 Y: 115 W: 141 H: 141	0.581674
	X: 150 Y: 100 W: 100 H: 100	X: 132 Y: 106 W: 133 H: 133	0.622640

26.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 100 Y: 50 W: 100 H: 100	X: 73 Y: 71 W: 125 H: 125	0.382170
27.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 100 Y: 50 W: 100 H: 100	X: 135 Y: 64 W: 55 H: 55	0.441437
	X: 100 Y: 50 W: 100 H: 100	X: 230 Y: 100 W: 82 H: 82	0.000000
28.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 198 Y: 99 W: 99 H: 99	X: 188 Y: 115 W: 114 H: 114	0.638462
29.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 100 Y: 100 W: 100 H: 100	X: 95 Y: 124 W: 167 H : 167	0.492675
30.jpg	Bounding Box Detected	Bounding Box Ground Truth	IOUs
	X: 89 Y: 90 W: 68 H: 68	No Detections	0.000000

### 3.3 Pixel Accuracy

Here will be shown the values of Pixel Accuracies computed for each image :

Name Image	Pixel Accuracy
01.jpg	0.992141
02.jpg	0.995846
03.jpg	0.995306
04.jpg	0.994716
05.jpg	0.993231
06.jpg	0.98774
07.jpg	0.986778
08.jpg	0.987347
09.jpg	0.993686
10.jpg	0.993523
11.jpg	0.988424
12.jpg	0.974098
13.jpg	0.988409
14.jpg	0.975737
15.jpg	0.988674
16.jpg	0.955056
17.jpg	0.983872
18.jpg	0.962644
19.jpg	0.980672
20.jpg	0.987248
21.jpg	0.996202
22.jpg	0.972186
23.jpg	0.988329
24.jpg	0.980469
25.jpg	0.971161
26.jpg	0.984074
27.jpg	0.967496
28.jpg	0.989969
29.jpg	0.977913
30.jpg	0.994816

# Chapter 4

## Contribution

### 4.1 Hours of work

- Francesco Caldivizzi : In this project I have worked the following amount of hours :
  - Roughly 60 hours to develop the C++ code
  - Roughly 50 hours to develop the python and matlab code and write the README file
  - Roughly 30 hours to find a suitable solution for the project after the first several trials
  - Roughly 15 hours to write the report
- Daniela Cuza :
  - Roughly 30 hours to find and try various ideas (most of them didn't work)
  - Roughly 15 hours to search the datasets, combine and clean them (also from "black masks"), resize and convert the images into the correct format
  - Roughly 15 hours to develop the data augmentation technique
  - Roughly 20 hours to compute various tests by changing the backbone, data augmentation method and loss function, in order to find the approach with the best performance
  - Roughly 15 hours to write the report
- Simone D'antimo :
  - Roughly 20 hours to search for a suitable solution for project, dataset etc..
  - Roughly 2 hours for cleaning data
  - Roughly 30 hours to try to implement a CNN based solution for Segmentation problem (using U-Seg) (that wasn't supported by c++ OPENCV library)
  - Roughly 50 hours to develop C++ code
  - 5 hours to write and correct typo of the report



# Chapter 5

## Conclusions

### 5.1 Results Discussions

For what concerns the detection part, we can immediately observe that, some images have bounding boxes well detected, others instead not or they do not have any detections at all. The main reason why there are those problems are :

- Some of the images contains hands with different point of view and, by using a window with static size for all the images (different size for images from 01 to 20 and from 21 to 30) there are multiple detections for the same hand because it is too big compared to the window size. (This can be seen in images like 17,18,19,20 etc.)
- Moreover, other images have no detection at all, and this may be cause by the fact that some of the hands have very strong illumination like images 21 and 30.

Therefore, even if the result is not 100% accurate, still it's quite good, because there are only 14 values of the IOU out of 68 that have value equal to 0. So, we can say that the main issue of this solution is the time taken for processing a single image which is very high.

For what concerns instead the segmentation part, we can observe that, most of the hands are segmented quite well, in fact, all the values of pixel accuracy are above 0.95. However, some different hands have been interpreted (colored) as a single one. (For example in image 01, 02, 14, 15, 18 etc.). This is due to the fact that, when two hands are overlapping one to another the mask produced by the model is connected instead of being separated.

### 5.2 Future Works

To solve the problems of the detection solution proposed, we can adopt more sophisticated techniques like ad-hoc Neural Networks that are able to recognize objects, by fine-tuning them, in order to recognize only hands (i.e. Yolo, Fast-RCNN, Faster-RCNN, SSD).

To solve the problem of the segmentation solution proposed, we can proceed by applying an erosion operation on the mask, however if we do so, we need to apply a very strong erosion, with the problem of reducing drastically the size of the mask and therefore the pixel accuracy value will become noticeably small compared to the ones used now. Another way to solve this problem is to use again, ad-hoc / state of the art segmentation methods that, instead of producing the masks as output they will produce the polygons which identify different objects, which, in this case are hands.