Single Gaussian Maximum Likelihood Estimation for Color Segmentation - A Project Review

Lien-Hsi Lu
Department of Electrical Computer Engineering
University of California, San Diego
llu@eng.ucsd.edu

***Abstract*—This paper is about an implementation of using Single Gaussian Maximum Likelihood Estimation for Color Segmentation problem and the aim is to detect blue barrels in the picture.**

I. Introduction

Living in this fast-pacing era, technology improves everyday. The power of machine learning gradually influence our lives, for example Autonomous Vehicles with a variety kinds of sensor such as camera doing computer vision has already being the hottest topic in the industry. In addition, household robots with sensors like camera and Lidar also needs the power of this technology to detect things in houses for the robot to grasp.

Based on the examples above, object detection is an important application of machine learning. In this project, we aimed to detect blue barrels in the picture and circle them out. This is any important technique for an Autonomous Vehicle or even a robot to know the environment. We can assume that the pictures in the training set are different inputs for them to learn and make decisions. So, the work this project is really a practical problem for us to solve.

To complete this task, we need to label the training set ourselves and eventually use a single Gaussian probability distribution to do Maximum Likelihood Estimation to determine each pixel's meaning to recover a mask. After segment the color, we can use some contour finding function to draw the rectangle bounding box to finish the task.

II. Problem Formulation

A. Gaussian Probability Distribution Model

In our project, we assume the probability is distributed in Gaussian Distribution. Due to our data is 3-dimension, we need to compute the parameters for conditional probability in multivariate Gaussian.

$$f(x) = \frac{1}{\sqrt{(2\pi)^k \mid \Sigma \mid}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$$

$\Sigma : covariance \quad matrix$
$\mu : mean$
$k : dimension$(1)

B. Maximum Likelihood Estimation

In our case, our label Y={blue, dark blue, bright blue, other} and X is the RGB value of each pixel {B,G,R} First, we derive the log-likelihood:

$$\begin{aligned} i^* &= argmax P_{X|Y,T}(x|i, D_i) * P_Y(i) \\ &= argmax log P_{X|Y}(x|i) + log Pr(i)) \\ &= argmax log \frac{1}{\sqrt{(2\pi)^k \mid \Sigma \mid}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)) \\ &= argmax -\frac{1}{2}(x-\mu)^T)\Sigma^{-1}(x-\mu) - \frac{k}{2} log 2\pi - \frac{1}{2} log \mid \Sigma \mid \\ &+ log Pr(i) \end{aligned} \tag{2}$$

After getting the log-likelihood, we obtain the derivative of μ=0 and the derivative of Σ=0:

$$\frac{\partial i^*}{\partial \mu} = \frac{\partial -\frac{1}{2}(x-\mu)^T)\Sigma^{-1}(x-\mu)}{\partial \mu} \tag{3}$$

$$\frac{\partial i^*}{\partial \mu} = \frac{\partial -\frac{1}{2}(x-\mu)^T)\Sigma^{-1}(x-\mu) - \frac{1}{2} log \mid \Sigma \mid}{\partial \Sigma} \tag{4}$$

As a result, we obtained the estimated parameters:
μ = sample mean of x
Σ = sample variance of x

C. Bayes Decision Rule

The Bayes Decision Rule will help us to classify certain pixel into certain classes.
Start from Bayes Rule:

$$P(\theta|\mathbf{D}) = P(\theta)\frac{P(\mathbf{D}|\theta)}{P(\mathbf{D})} \tag{5}$$

In our case, D will be the data and θ will be the classes we labeled.
Using Bayes Decision Rule to determine the class of the pixel, take the example of classifying blue class and non-blue class, like below:

$$i^* = argmax P_{X|Y,T}(x|i, D_i) * P_Y(i) \tag{6}$$

If the pixel wants to be classify into blue class, the Bayes Decision Rule should be like:

$$PY|X(y = blue|x) > PY|X(y = non-blue|x) \tag{7}$$

By bayes rule:

$$PX|Y(x|y = blue) * PY(y = blue) > PX|Y(x|y = nonblue) * PY(y = non - blue) \quad (8)$$

If the pixel wants to be classify into non-blue class, the Bayes Decision Rule should be like:

$$PY|X(y = nonblue|x) > PY(y = blue|x) \quad (9)$$

By bayes rule:

$$PX|Y(x|y = nonblue) * PY(y = nonbule) > PX|Y(x|y = blue) * PY(y = blue) \quad (10)$$

III. Technical Approach

A. Labeling

In order to do supervised learning, we use the function **roiply** to label the training set images. In addition, to make the classifier to distinguish not only blue and non-blue case, I do labels for dark blue area and bright blue area in the images. In this case, the training set has the label of barrel blue, dark blue, bright blue and the others. After labeling, I save the image into masks in the data type of numpy array.

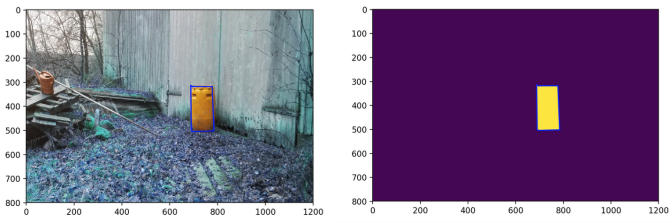

Fig. 1. original image and the mask after labeling

B. Parameters Computing

In order to plug in the parameters to Gaussian Distribution, we need to compute the mean and covariance. As mentioned before in the labeling part, there are 4 classes which are barrel blue, dark blue, bright blue and the other color class. So we need to get the index in the labeled images to obtain the RGB data of these certain index value for each classes. What interesting is I combine the barrel blue, dark blue and bright blue classes into a big class called general blue. At last, I get five means and five covariance matrix for barrel blue, dark blue, bright blue, other and general blue.

C. Gaussian Function

I built a Gaussian Probability Distribution Function from scratch with four inputs: pixel index, mean, covariance matrix and dimension. In our case, the dimension is 3 for the color space RGB. The formula is in the problem formulation section.

D. Bayes Decision Rule

The implementation of this Bayes Decision Rule is to first classify the pixel in the general blue class or other color class. With the result of classified the pixel in to general blue class, I do the classification again for determining which kind of blue (dark,bright,barrel blue) this pixel is.

The probability can be computed by conditional Probability times prior Probability. We get the conditional Probability by plugging the parameters into Gaussian Distribution. On the other hand, I set the prior probability of general blue class into 0.25 and other color class into 0.75. Both dark blue and bright blue class are set into 0.15 and the barrel blue class set into 0.7.

After the classification finished, we obtained a True-False image that the barrel part is True and the other pixels are False.

E. Bounding Box

To circle the barrel in the image, we need to draw bounding box around it. In order to obtain the bounding box, I use the function in openCV called **findContours** to find the contours in the True-False image and use the function **boundingRect** to draw the bounding box on the original image. We will obtain four parameters from it which is x (x axis of left-up point),y (y axis of left-up point),w (width),h (height). To make the bounding box more precise, I add some condition that certain area will not being classify into barrel blue class to make the result better. Updated successfully at the last day of deadline: I have implemented the method to erode and dilate the final mask and plug into the bounding box function, this improve the bounding box result.

Fig. 2. example of bounding box on the image

F. Training Set Enlargement

In the original training set, there are some pictures that include the color of dark blue and bright blue. However, there are very few of them. To make the training process to learn more about the dark blue and the bright blue part, I enlarged

training set by copying those pictures into new images to get better training model. As a result, the training set has 75 images and the segmentation accuracy achieved more than 98%.

IV. Result

A. Successful Cases

In general, while the blue barrel is not been put in a strange angle or in a very dark place (This make the barrel looks very dark.), my result of color segmentation and circle with the bounding box will succeed. See examples from Fig.3 and Fig.4.

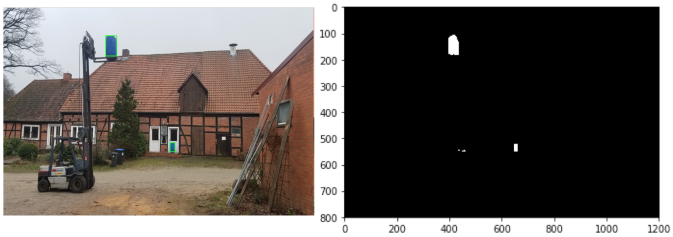

Fig. 3. general cases that succeed in both segmentation and bounding box and the coordinates [x1,y1,x2,y2] are:[395, 107, 438, 185], [644, 519, 663, 554]

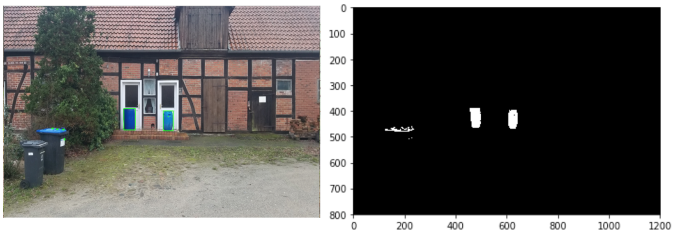

Fig. 4. general cases that succeed in both segmentation and bounding box and the coordinates [x1,y1,x2,y2] are:[456, 387, 498, 467], [606, 396, 642, 470]

There are some images that include different kinds of blue objects, which is a harder problem than the general cases, the result show that it still succeed in the bounding boxes although some result of the color segmentation looks not that well. See examples from Fig.5 to Fig.7.

Some case seems there are some dark shadow on the barrels, but the bounding box successfully circle out the right barrel although there are some failure in segmentation. See examples from Fig.8 to Fig.11.

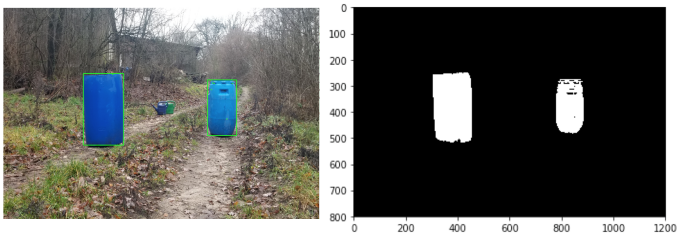

Fig. 5. include other blue objects but still succeed, bounding box successfully choose the right barrel and the coordinates [x1,y1,x2,y2] are:[304, 250, 457, 514], [777, 289, 885, 482]

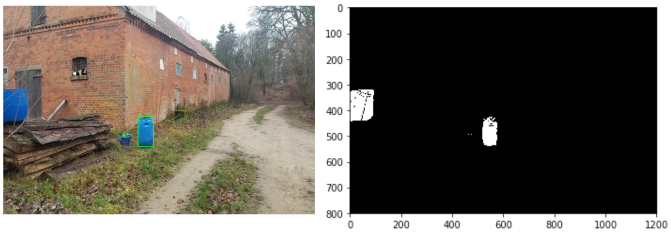

Fig. 6. include other blue objects but still succeed, bounding box successfully choose the right barrel and the coordinates [x1,y1,x2,y2] is:[518, 425, 578, 540]

Fig. 7. include other blue objects but still succeed, bounding box successfully choose the right barrel and the coordinates [x1,y1,x2,y2] is:[589, 413, 638, 486]

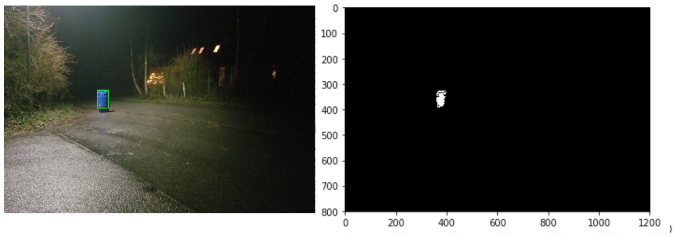

Fig. 8. bounding boxes are correct but with some failure in segmentation [x1,y1,x2,y2] is:[360, 326, 402, 395]

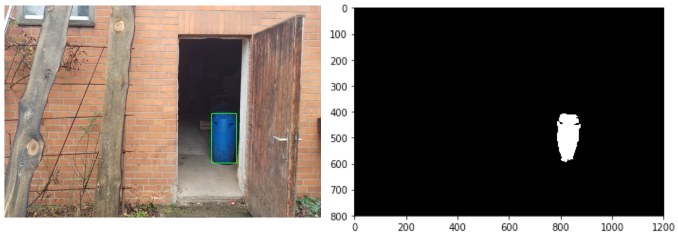

Fig. 9. bounding boxes are correct but with some failure in segmentation and the coordinates [x1,y1,x2,y2] is:[784, 407, 879, 594]

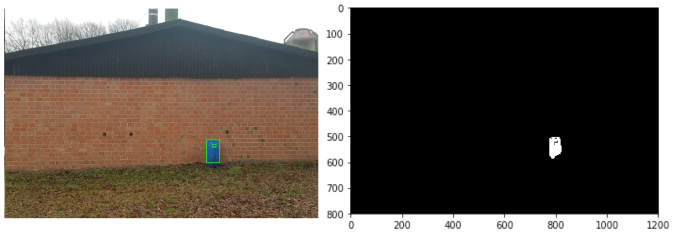

Fig. 10. bounding boxes are correct but with some failure in segmentation and the coordinates [x1,y1,x2,y2] is:[772, 501, 822, 588]

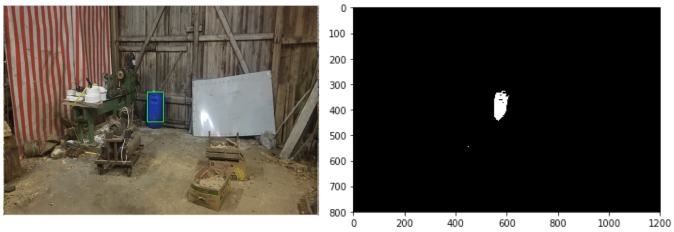

Fig. 11. bounding boxes are correct but with some failure in segmentation and the coordinates [x1,y1,x2,y2] is:[548, 330, 610, 444]

Some cases are successful in the bounding box but occurred some noise and the segmentation seems really bad because of the light makes the barrels look white. See examples in Fig.12.

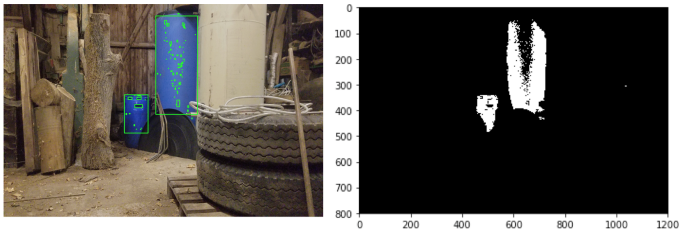

Fig. 12. the bounding box the light make the segmentation failed but succeed in bounding box with some noise and the coordinates [x1,y1,x2,y2] are: [453, 341, 542, 485],[571, 45, 728, 415] and with a lot of bounding boxes I define them as noises

B. Failure Cases

Some barrels are been put in a place that have some dark shadow on it, this makes the blue barrel has some dark area in the image that the classifier can not know it is the blue barrel color. In addition, a barrel been put too far from the camera will make the ratio of the width and the height of the barrel looks strange. There is an example of a picture with three barrels: one was normal, one was been put in a place with dark shadow and one was too small. See examples in Fig.13.

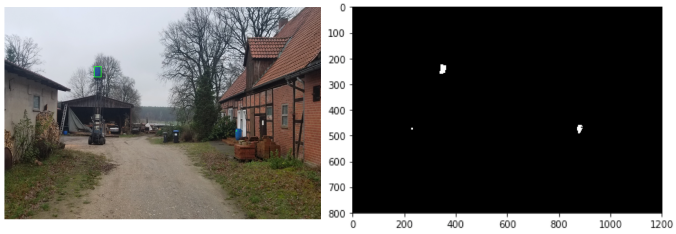

Fig. 13. three barrels in the picture but only one was circled and the coordinates [x1,y1,x2,y2] is:[338, 224, 364, 263]

Some cases are the blue barrel was been put in a very dark place that makes the blue barrel are covered by dark shadows or looks in a strange color. Some still can succeed because the shadow didn't cover too much of the barrel. However, if the environment was too dark the segmentation fails, so do the bounding box. See examples from Fig.14 to Fig.16.

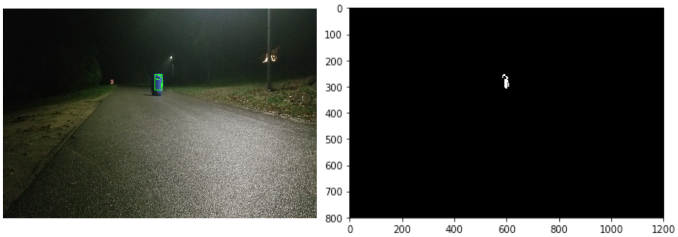

Fig. 14. the barrel is been put in a dark place and was covered by its shadow, fails a bit and the coordinates [x1,y1,x2,y2] is:[581, 251, 609, 310]

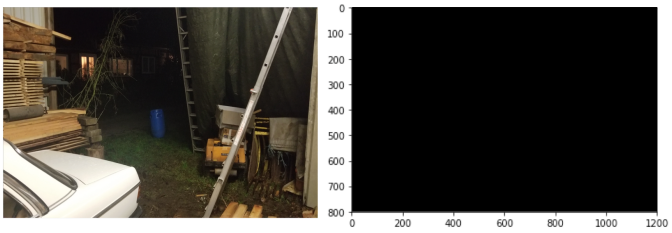

Fig. 15. the barrel is been put in a very dark place that fails and there are no bounding box appear

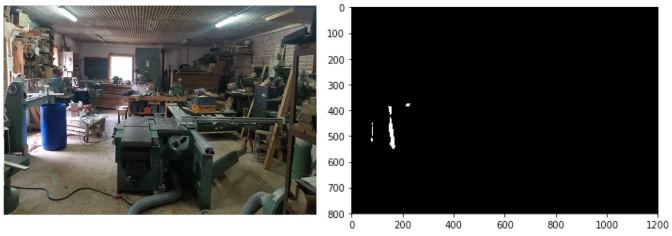

Fig. 16. the barrel is been put in a very dark place that fails and there are no bounding box appear

Some cases are that the blue barrel was blocked by certain obstacle in the image. In this case, although the segmentation still looked successful, the bounding box failed badly. See examples from Fig.17 to Fig.21.

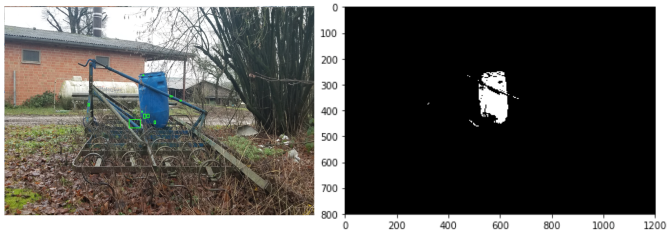

Fig. 17. the barrel is been block by a metal frame and the coordinates of bounding boxes are meaningless [x1,y1,x2,y2]: [531, 260, 536, 263], [590, 266, 594, 268], [550, 278, 551, 280], [590, 286, 611, 296], [610, 306, 612, 307], [595, 319, 597, 320], [636, 340, 645, 345], [530, 343, 531, 345], [645, 344, 673, 357], [529, 346, 530, 348], [527, 349, 529, 353], [674, 357, 680, 361], [320, 370, 326, 380], [527, 399, 530, 404], [547, 411, 556, 425], [534, 411, 543, 427], [479, 432, 522, 464], [575, 437, 581, 448]

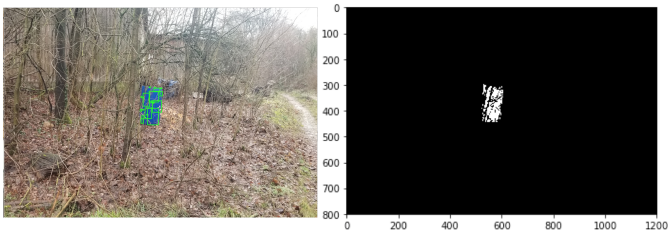

Fig. 18. the barrel is been block by a tree and the coordinates of the bounding boxes just circle out every part of the barrel behind the tree [x1,y1,x2,y2]: [555, 301, 560, 304], [575, 303, 580, 306], [599, 305, 604, 308], [573, 306, 596, 320], [555, 318, 592, 359], [597, 319, 606, 336], [532, 325, 557, 375], [524, 337, 527, 342], [531, 340, 534, 345], [523, 343, 527, 352], [568, 354, 603, 398], [562, 354, 574, 362], [568, 361, 573, 364], [539, 381, 564, 422], [549, 383, 591, 446], [521, 386, 526, 394], [529, 387, 534, 390], [563, 393, 565, 397], [545, 406, 550, 409], [525, 407, 529, 415], [527, 411, 535, 423], [535, 418, 547, 444], [524, 423, 533, 440]

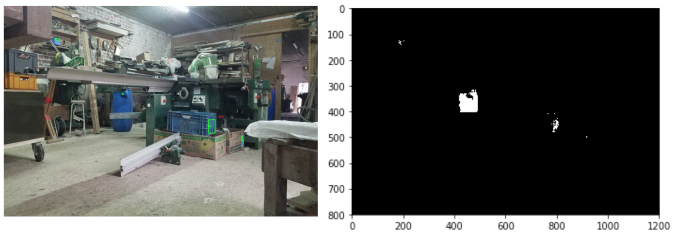

Fig. 19. the barrel is been block by a some laboratory tool and the coordinates of the bounding boxes are meaningless [x1,y1,x2,y2]: [184, 122, 195, 145], [199, 123, 212, 130], [194, 123, 199, 126], [188, 126, 190, 127], [472, 319, 473, 321], [477, 320, 481, 322], [453, 331, 456, 336], [468, 332, 469, 334], [439, 333, 441, 334], [481, 335, 482, 337], [474, 339, 475, 341], [462, 367, 463, 369], [782, 432, 807, 483], [792, 436, 793, 438], [790, 439, 791, 441], [787, 442, 789, 443], [798, 443, 800, 444], [796, 448, 797, 450], [792, 449, 793, 451], [788, 449, 789, 451], [780, 451, 783, 456], [801, 452, 802, 454], [796, 454, 797, 456], [792, 456, 793, 458], [787, 457, 790, 459], [796, 458, 797, 460], [790, 461, 791, 463], [796, 464, 797, 466], [909, 494, 912, 499], [909, 504, 914, 513]

Fig. 20. the barrel is been block by two water cans and there are no bounding boxes appear

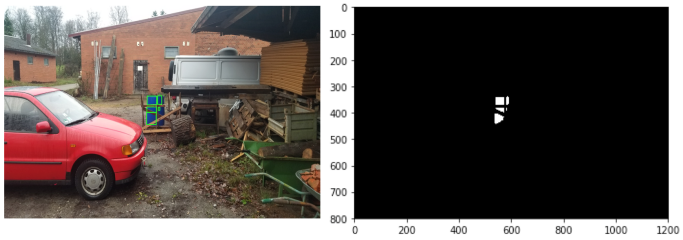

Fig. 21. the barrel is been block by a wood frame and the coordinates of the bounding boxes circle out most part of the barrel behind the wood frame, except some dark shadow area [x1,y1,x2,y2]: [[577, 338, 593, 370], [538, 339, 574, 372], [537, 380, 573, 401], [580, 381, 586, 393], [536, 400, 572, 446]]

Some cases are that there were a group of blue barrels been put together, so in the image we saw is a group of blue objects. Although we can know which one is the blue barrel through our eyes but in the computer it seems really hard to do that. Eventually, although the segmentation knows where the barrel blue color is, the bounding still fail to circle out the right things. See examples in Fig.22.

Some case is that the barrel was been put in a strange direction, for example a tilted position. Although the segmentation will succeed but the bounding box will not know what it is. What worse is, the example in this tilted

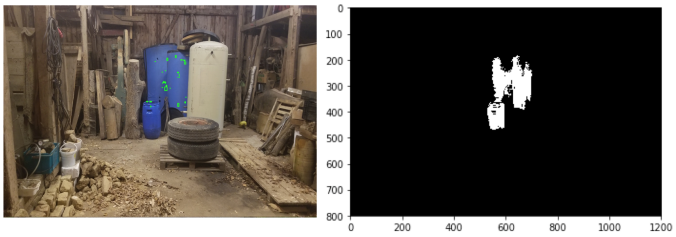

Fig. 22. a group of blue barrel stick together that fools the bounding box and the coordinates of the bounding boxes are meaningless [x1,y1,x2,y2]: [652, 190, 656, 198], [676, 193, 682, 197], [680, 197, 684, 203], [620, 200, 624, 209], [673, 205, 678, 208], [684, 207, 689, 210], [575, 212, 577, 213], [687, 214, 692, 223], [673, 216, 679, 225], [686, 222, 692, 231], [599, 233, 605, 237], [598, 235, 602, 241], [597, 240, 601, 246], [602, 246, 604, 249], [614, 247, 616, 248], [602, 252, 603, 254], [660, 253, 665, 256], [666, 261, 667, 263], [667, 264, 668, 266], [667, 271, 668, 273], [662, 271, 663, 273], [610, 281, 613, 283], [611, 288, 613, 291], [607, 288, 610, 293], [613, 290, 616, 295], [605, 292, 607, 293], [603, 294, 605, 295], [633, 300, 635, 303], [617, 302, 620, 307], [612, 309, 621, 323], [602, 311, 604, 312], [632, 313, 633, 315], [591, 314, 593, 317], [692, 347, 693, 349], [692, 351, 694, 354], [555, 363, 562, 366], [543, 363, 548, 366], [688, 365, 691, 370], [661, 371, 666, 383], [535, 375, 538, 377], [586, 376, 588, 379], [558, 381, 560, 385], [561, 383, 564, 385]

barrel was been put in a very dark place. This made both the segmentation and bounding box failed. See examples in Fig.23.

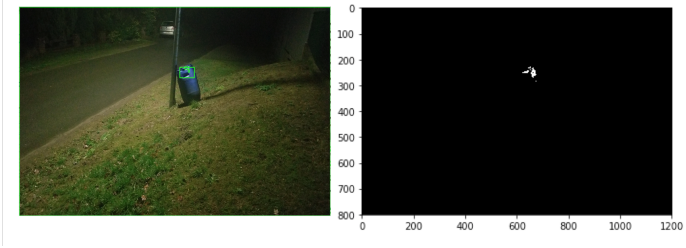

Fig. 23. a tilted blue barrel in a dark place that fools the classifier and the coordinates of the bounding boxes are meaningless [x1,y1,x2,y2]:[638, 229, 654, 238], [651, 241, 656, 250], [649, 243, 652, 248], [662, 251, 663, 253], [664, 255, 666, 259]

C. Discussion

As the result by seeing above of the successful cases and failure cases, some hard problems I have solved it, but some other remained unsolved.

1) Dark Shadow and Environment
   Some barrel in the images was affected by the light that it caused some shadow on the it. This caused the barrel blue looked too dark that the classifier will easily think this is not a barrel color. Eventually, the classification failed and of course the bounding box would failed. I have thought of a solution might work is when labeling the training image, we might need to label the real barrel blue and the dark barrel blue. In this case, the classifier will learn to know that both the real barrel blue and the dark barrel blue can be classified into the barrel blue class. This might improve the result.
2) Blocking Objects
   Some barrel in the pictures are being blocked by variety kinds of object, such as tree, frame...etc. In this case, although the classifier can still classified the pixels into right classes. But the biggest problem will be the bounding box wouldn't know the area is the barrel. I have tried to get the ratio of width and height of the barrel and some area conditions to obtain better result. However, I failed at last. I thought of a solution that might improved is to merge those small bounding box into a big one to achieve success.
3) A Cluster of Barrels
   Some image include a group of barrels that the classifier will think there is a big blue strange object. If the cluster of barrels are not overlapped with each other, then I can easily use the ratio of width and height of the barrel to detect the barrels. However, because of the overlapping barrels. The bounding box never succeed in this case.
4) Strange Pose barrels
   If a image has a barrel that are tilted, the segmentation would absolutely succeed but the bounding box would not. I have tried to implement the **rotated rectangle** function in the picture, but it still failed. I think there are two problems inside, the first is the picture on test on it is too dark that the segmentation didn't succeed. The second is although the titled barrel have a bounding box on it but it didn't match the tilted angle of the barrel. Maybe need some brighter image with the tilted angle to make more tests.

REFERENCES

[1] Richard O. Duda, Peter E. Hart, and David G. Stork, "Pattern Classification," Second Edition.