Image Classification and Object Localization

Ezgi ÇAKIR
Dpt. of Computer Science
Bilkent University
Ankara, Turkey
ezgi.cakir@ug.bilkent.edu.tr
ID : 21402334

Hüseyin Eren ÇALIK
Dpt. of Computer Science
Bilkent University
Ankara, Turkey
eren.calik@ug.bilkent.edu.tr
ID : 21402338

Metehan KAYA
Dpt. of Computer Science
Bilkent University
Ankara, Turkey
m.kaya@ug.bilkent.edu.tr
ID : 21401258

Abstract

The goal of this paper is to develop methods for image classification and object localization in an image. We developed these methods as the term project CS-484 Image Analysis.

Keywords—image, classification, object, localization, training

I. Introduction

In this paper, we put an effort to develop a method for classify the images. As we were classifying, also we tried to locate a bounding box of an object in the image that we try to classify. We are given 500 images from 10 different classes: eagle, dog, cat, tiger, starfish, zebra, bison, antelope, chimpanzee, and elephant. 400 of them are for training our model and the rest 100 of them are for testing the classification and localization model. In the rest of the paper, we will provide example resulting images that we obtained after testing. Furthermore, it will be discussed that the parameters chosen while testing data.

II. Pre-processing

A. *Data Normalization*

Since each image may has different aspect ratio and size, we normalized each image as the project description instructed. As a result of this process, we obtained images that has normalized RGB values and are in the size of 224 x 224 x 3.

B. *Feature Extraction*

2048-dimensional representations from Resnet-50 are extracted.

III. Training the System

We made a used a MATLAB implementation of SVM classifier in order to train the system. We trained our system with 400 different images belonging to 10 different classes. We classified images with respect to extracted features from ResNet-50. We decided to use linear Kernel function because the other MATLAB implementation Kernel functions were not appropriate for our training. Addition to that, we implemented 10-fold cross validation method in order to decide which parameter to use for SVM. Then, we run 10 iterations per each classifier. We trained our dataset with different hyperparameters of SVM such as KernelScale and BoxContraint. For example, when we set KernelScale as 0.4 and BoxConstraint as 0.1, focus of bounding box was for some images narrower, such as eyes and noises and some larger. This results in incorrect results in terms of localization and classification accuracy. After trying various set of parameters, we set KernelScale as default(1) and BoxConstraint as 1000 because these parameters gave the most correct results for our training dataset. Finally, we standardized the predictor data.

IV. Testing the System

We tested 100 different images in the test dataset. 50 different windows for each image were created as candidate bounding box in order to localize an object. After that, each image was classified with respect to 50 bounding boxes, then we selected the window that has the maximum classification score as the predicted bounding box of an image.

We used Edge Boxes method implemented in MATLAB. We test our system with different parameters. Those parameters were alpha which indicates the step size of sliding window search and beta which stands for non-maxima suppression for object proposal. Further explanation will be given for these parameters below. After many trials, we decided to set alpha as 0.65 and beta as 0.75.

In testing the data, we aimed to locate an object to be classified in an image and classify it. The classification accuracy and localization accuracy will be discussed in the rest of this paper.

A. *Classification Accuracy*

When we set alpha as 0.75 and beta as 0.85, classification accuracy decreased to 95%. We could obtain 100% classification accuracy when we set alpha as 0.65 and beta as 0.75.
You can find Table I that shows the predicted and correct classification of images.

# of the image	*Predicted Classification*	*Correct Classification*
0	*n02317335*	*n02317335*
1	*n02317335*	*n02317335*
2	*n02317335*	*n02317335*
3	*n02317335*	*n02317335*
4	*n02317335*	*n02317335*
5	*n02317335*	*n02317335*
6	*n02317335*	*n02317335*
7	*n02317335*	*n02317335*
8	*n02317335*	*n02317335*
9	*n02317335*	*n02317335*
10	*n02504458*	*n02504458*
11	*n02504458*	*n02504458*
12	*n02504458*	*n02504458*
13	*n02504458*	*n02504458*
14	*n02504458*	*n02504458*
15	*n02504458*	*n02504458*
16	*n02504458*	*n02504458*
17	*n02504458*	*n02504458*
18	*n02504458*	*n02504458*
19	*n02504458*	*n02504458*
20	*n02391049*	*n02391049*
21	*n02391049*	*n02391049*
22	*n02391049*	*n02391049*
23	*n02391049*	*n02391049*
24	*n02391049*	*n02391049*
25	*n02391049*	*n02391049*
26	*n02391049*	*n02391049*
27	*n02391049*	*n02391049*
28	*n02391049*	*n02391049*
29	*n02391049*	*n02391049*
30	*n02099601*	*n02099601*
31	*n02099601*	*n02099601*
32	*n02099601*	*n02099601*
33	*n02099601*	*n02099601*
34	*n02099601*	*n02099601*
35	*n02099601*	*n02099601*
36	*n02099601*	*n02099601*
37	*n02099601*	*n02099601*
38	*n02099601*	*n02099601*
39	*n02099601*	*n02099601*
40	*n02410509*	*n02410509*
41	*n02410509*	*n02410509*
42	*n02410509*	*n02410509*
43	*n02410509*	*n02410509*
44	*n02410509*	*n02410509*
45	*n02410509*	*n02410509*
46	*n02410509*	*n02410509*
47	*n02410509*	*n02410509*
48	*n02410509*	*n02410509*
49	*n02410509*	*n02410509*
50	*n02481823*	*n02481823*
51	*n02481823*	*n02481823*
52	*n02481823*	*n02481823*
53	*n02481823*	*n02481823*
54	*n02481823*	*n02481823*
55	*n02481823*	*n02481823*
56	*n02481823*	*n02481823*
57	*n02481823*	*n02481823*
58	*n02481823*	*n02481823*
59	*n02481823*	*n02481823*
60	*n02422699*	*n02422699*
61	*n02422699*	*n02422699*
62	*n02422699*	*n02422699*
63	*n02422699*	*n02422699*
64	*n02422699*	*n02422699*
65	*n02422699*	*n02422699*
66	*n02422699*	*n02422699*
67	*n02422699*	*n02422699*
68	*n02422699*	*n02422699*
69	*n02422699*	*n02422699*
70	*n02123159*	*n02123159*
71	*n02123159*	*n02123159*
72	*n02123159*	*n02123159*
73	*n02123159*	*n02123159*
74	*n02123159*	*n02123159*
75	*n02123159*	*n02123159*

76	*n02123159*	*n02123159*
77	*n02123159*	*n02123159*
78	*n02123159*	*n02123159*
79	*n02123159*	*n02123159*
80	*n01615121*	*n01615121*
81	*n01615121*	*n01615121*
82	*n01615121*	*n01615121*
83	*n01615121*	*n01615121*
84	*n01615121*	*n01615121*
85	*n01615121*	*n01615121*
86	*n01615121*	*n01615121*
87	*n01615121*	*n01615121*
88	*n01615121*	*n01615121*
89	*n01615121*	*n01615121*
90	*n02129604*	*n02129604*
91	*n02129604*	*n02129604*
92	*n02129604*	*n02129604*
93	*n02129604*	*n02129604*
94	*n02129604*	*n02129604*
95	*n02129604*	*n02129604*
96	*n02129604*	*n02129604*
97	*n02129604*	*n02129604*
98	*n02129604*	*n02129604*
99	*n02129604*	*n02129604*

TABLE I. CLASSIFICATION RESULTS OF IMAGES

B. *Localization Accuracy*

When we set alpha as 0.75 and beta as 0.85, localization accuracy decreased to 48%. When we set them as 0.65 and 0.75, it is increased to 49% and it was the highest localization accuracy that we could obtain.

Most of the predicted bounding boxes were located smaller than the correct bounding boxes. On the other hand, if there are more than 1 object in an image, it is also located in the predicted bounding boxes. Nevertheless, they were correctly classified. For each image, we drew the given bounding box in red and the predicted bounding box in green. Moreover, the predicted and correct classification and overlap ratio of predicted and correct bounding box were written on the corresponding image. In addition to these, we provided an image from each class in which all candidate windows are overlaid in the corresponding image. Detailed analysis for each class can be found below.

I. *Class of n02317335*

The average overlap ratio of this class is 0.6198. This was the second highest overlap score. Thus, we can claim that locating objects in this class was respectively easier. In this class, predicted bounding boxes tend to be smaller than the correct ones. This can be observed in Figure 1 and Figure 3. On the other hand, in some images, such as in Figure 2, predicted bounding boxes are not only larger than the correct one but also include other objects in the image. In Figure 4, a starfish and another object are found. Although they have different sizes, it seems that nearly same number of candidate windows are detected.

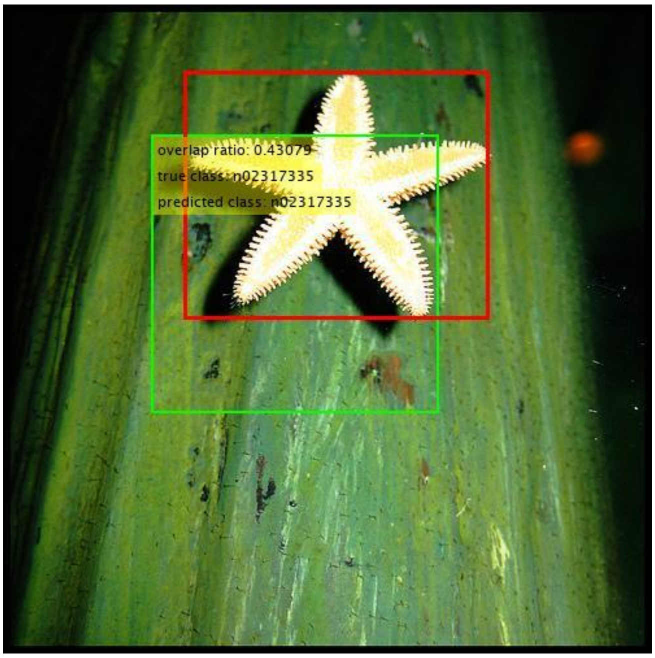

Fig. 1. Bounding box result of 0.jpeg

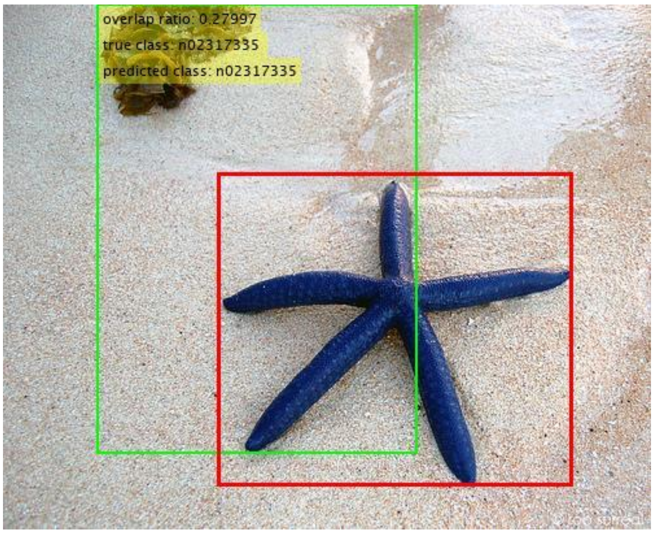

Fig. 2. Bounding box result of 3.jpeg

Fig. 3. Bounding box result of 5.jpeg

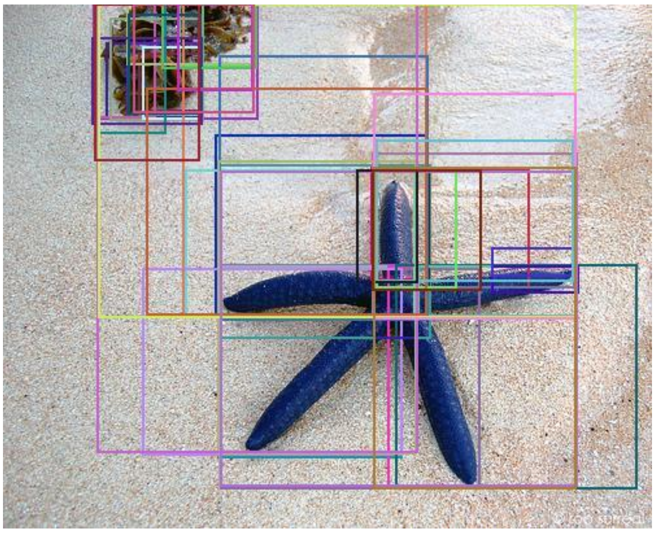

Fig. 4. All bounding box result of 3.jpeg

II. *Class of n02504458*

In this class, the predicted bounding boxes tend to be restricted heads of elephants rather than including the whole body of the elephant. Therefore, the average overlap ratio of this class is 0.4841 which can be accepted as lower. Bounding boxes can be seen in Figure 5 and Figure 7. Furthermore, if various objects exist in an image, the predicted bounding boxes tend to include those objects as well. This can be observed in Figure 6. As it can be seen in Figure 8, candidate windows are detected which lay between consecutive legs of the elephant. It shows that there is no rule such that background only appears at outside of the object.

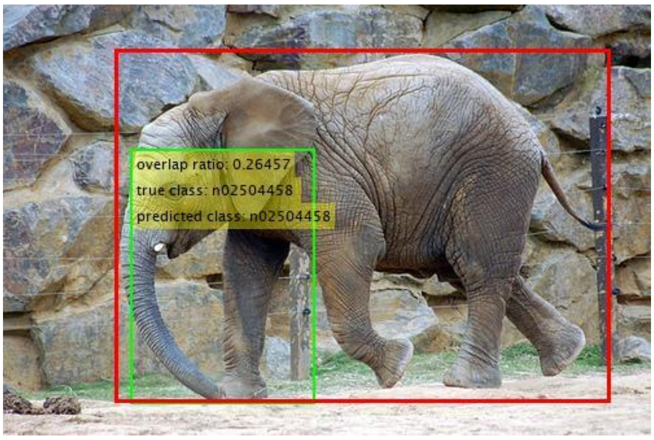

Fig. 5. Bounding box result of 12.jpeg

Fig. 6. Bounding box result of 14.jpeg

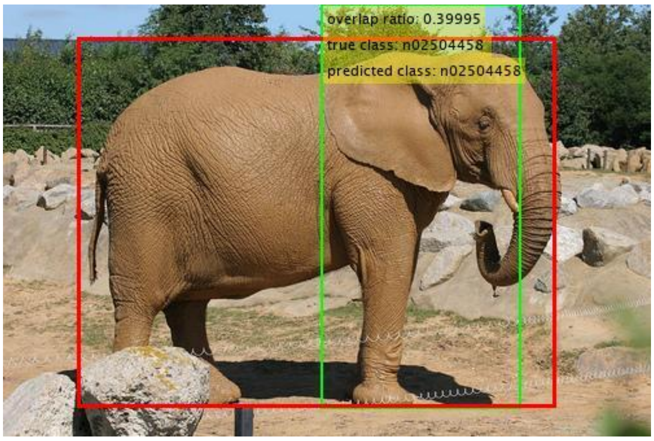

Fig. 7. Bounding box result of 19.jpeg

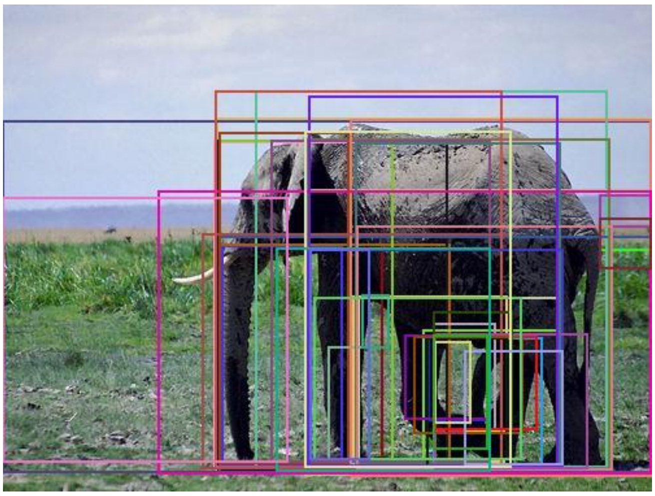

Fig. 8. All bounding box results of 13.jpeg

III. *Class of n02391049*

In this class, the predicted bounding boxes tend to include mostly the head and a part of the body of zebras rather than taking in the whole body. Nevertheless, average overlap ratio was higher than the previous class which is 0.5276. This is respectively high ratio, thus we may claim that localization of objects of this class was easy. In this class, most of the candidate windows are focused on the zebra where a few of them appears on some of the trees. It can be observed in Figure 11.

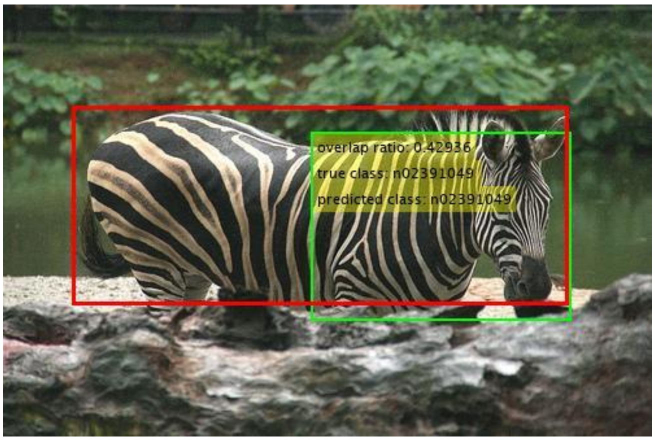

Fig. 9. Bounding box result of 20.jpeg

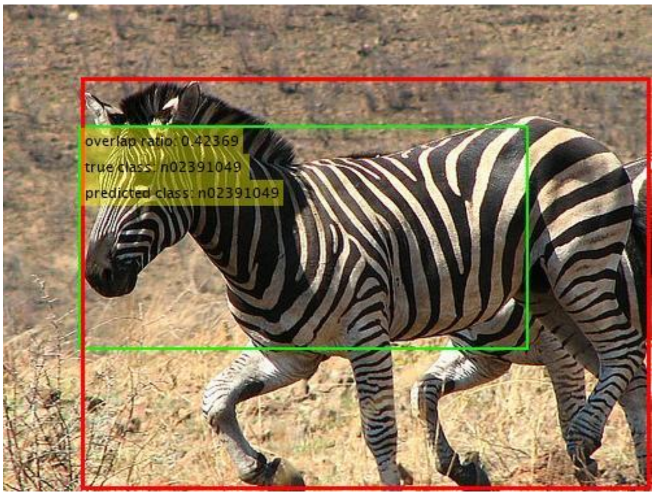

Fig. 10. Bounding box result of 24.jpeg

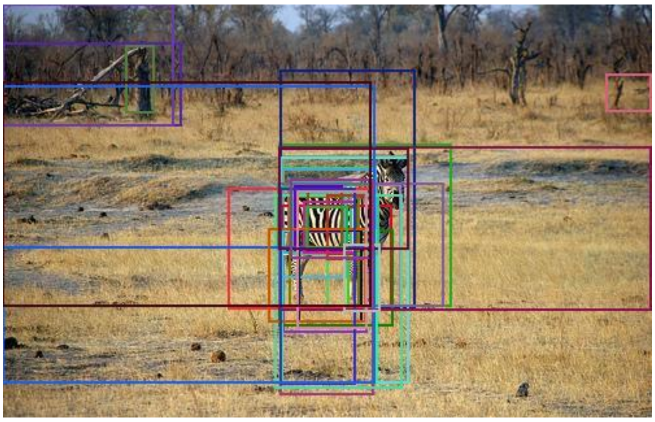

Fig. 11. All bounding box result of 29.jpeg

IV. *Class of n02099601*

The average overlap ratio of this class is 0.3545 which is the second lowest ratio among all classes. The reason for that can be that: the predicted bounding boxes tend to take in only faces of dogs rather than the whole body. Bounding boxes can be seen in Figure 12 and 13. Moreover, rather than the actual object, approximately all of the candidate windows are focused on next to the dog. To observe this, Figure 14 can be refered. Thus, this class was the worst among all of the classes in terms of localization accuracy. To sum up, localization whole object in images belonging to this class was difficult.

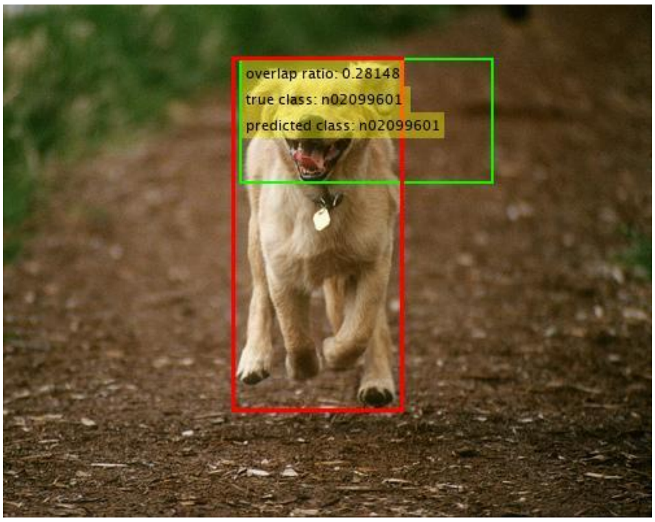

Fig. 12. Bounding box result of 37.jpeg

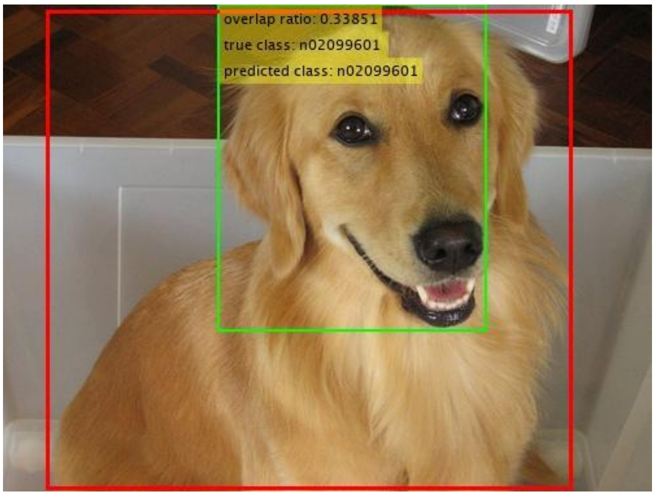

Fig. 13. Bounding box result of 36.jpeg

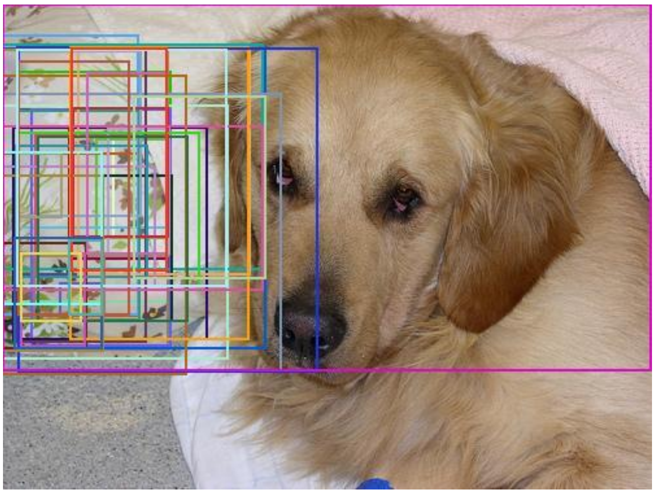

Fig. 14. All bounding box results of 30.jpeg

V. *Class of n02410509*

The average overlap ratio of this class is 0.6450 which is the highest score among the all classes. Therefore, the predicted bounding boxes were closer to the correct bounding box, considering the similarity between the correct and the predicted bounding box in the other class. Although most of the candidate windows are focused on the background, including trees, classification is done right. This can be observed Figure 17. As a result, we can say localization objects belonging to this class was the easiest.

Fig. 15. Bounding box result of 43.jpeg

Fig. 16. Bounding box result of 47.jpeg

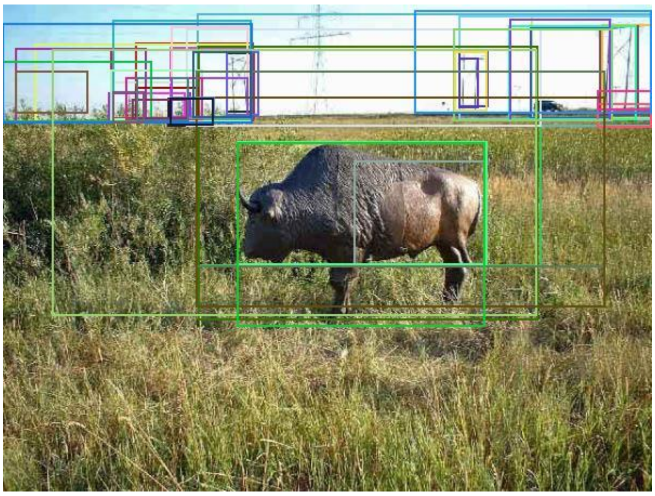

Fig. 17. All bounding box result of 47.jpeg

VI. *Class of n02481823*

In this class, the predicted bounding boxes tend to focus on the face of animals. This can be seen in Figure 18. Furthermore, if more than 1 object exist in an image, the bounding box tends to include those objects. This can be observed in Figure 19. Nevertheless, the average overlap ratio of this class is 0.5811 which is the third highest score. As it can be seen in Figure 20, although there are 3 big objects and candidate windows are distributed evenly which may lead to a wrong classification, importance of the training shows itself, and classification is done right. Thus, we may claim that it was easy to detect objects for this class.

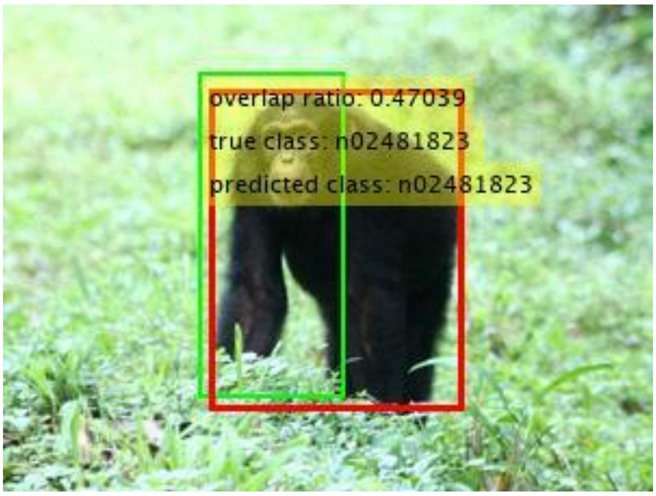

Fig. 18. Bounding box result of 52.jpeg

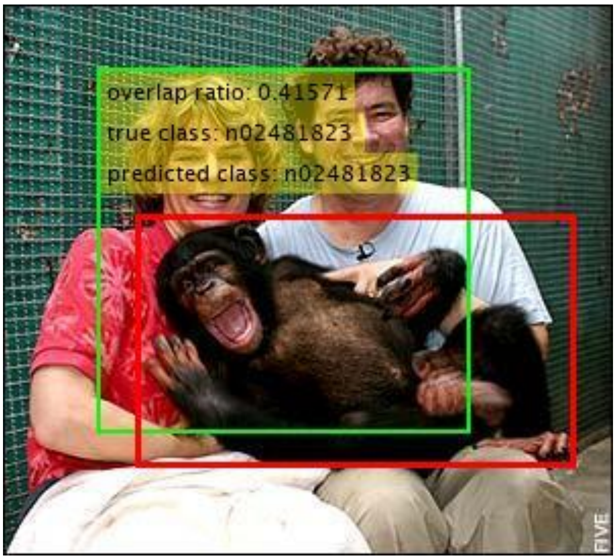

Fig. 19. Bounding box result of 55.jpeg

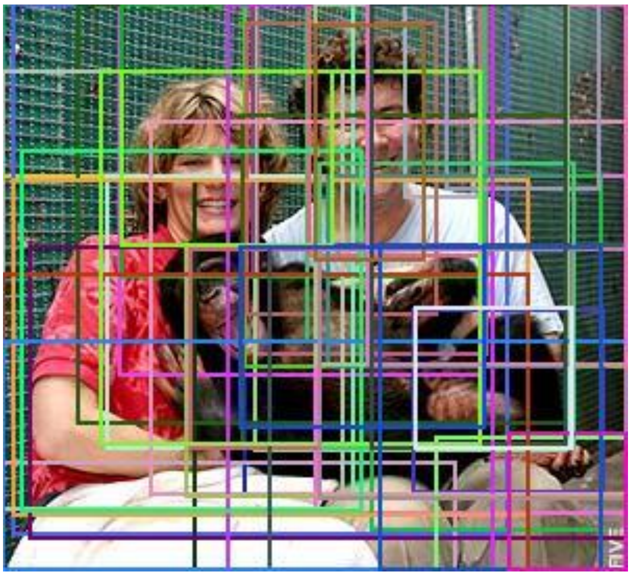

Fig. 20. All bounding box results of 55.jpeg

VII. *Class of n02422699*

The average overlap ratio of this class is 0.5567 which can be respectively high considering other classes. Figure 21 shows that for this class, if there exists another object in an image, it was not included in the predicted bounding box. Moreover, the bounding boxes tend to take in the whole body if the object is obvious to detect as in Figure 22. As it can be seen in figure 23, although a candidate window which best fits to the actual bounding box, since the one with highest score is selected, a worse candidate window is chosen. Nevertheless, localization of objects for this class can be seen as moderately easy.

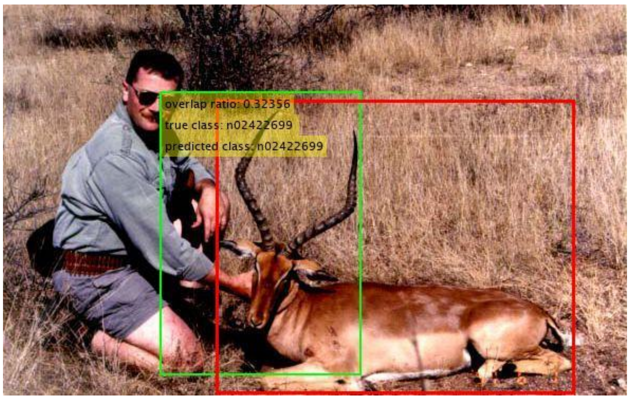

Fig. 21. Bounding box result of 64.jpeg

Fig. 22. Bounding box result of 66.jpeg

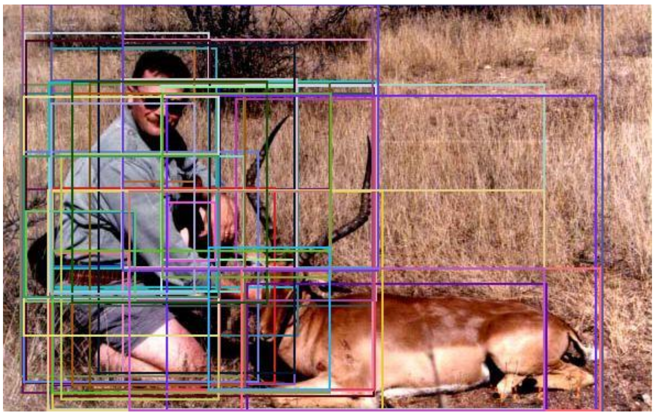

Fig. 23. All bounding box result of 64.jpeg

VIII. Class of n02123159

In this class, the bounding box tend to focus on stripes of the body of the objects in the images. This can be observed in Figure 24. Besides that, some images only focus on the faces which consist of mostly stripes. This can be seen in Figure 25 as well. The average overlap ratio of this class emerges as 0.3947 which is the third lowest score. In Figure 26, a similar situation to Figure 23 can be observed where a smaller candidate window is chosen. Also, it is observed that a small object which has the same color with the floor is covered by a blue candidate window. Consequently, detecting the whole object belonging to this class was difficult.

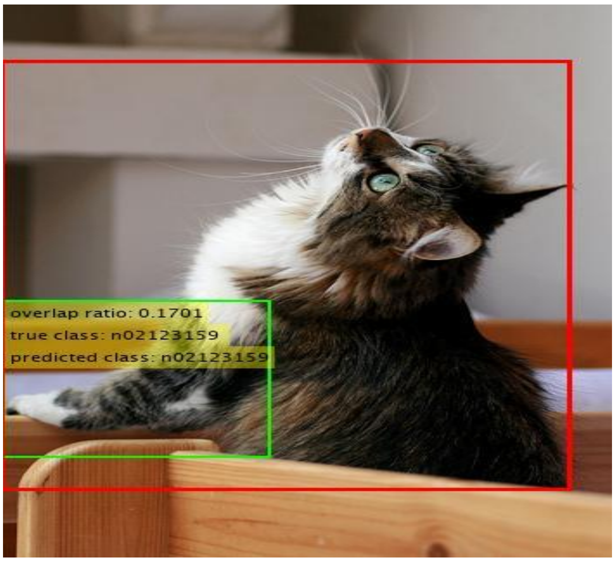

Fig. 24. Bounding box result of 71.jpeg

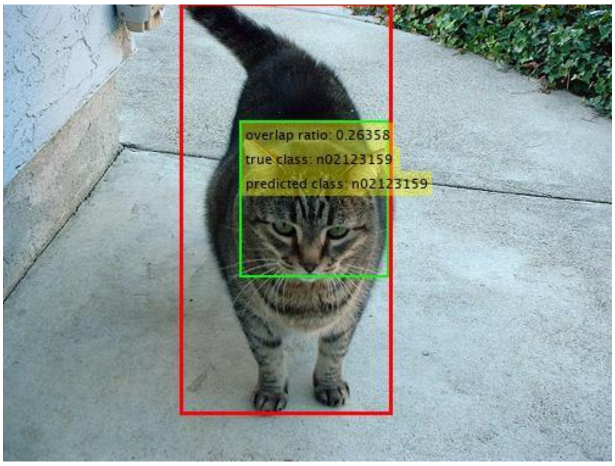

Fig. 25. Bounding box result of 73.jpeg

Fig. 26. All bounding box results of 73.jpeg

IX. Class of *n01615121*

In this class, the predicted and correct bounding boxes appeared respectively closer since the average overlap ratio of this class is 0.5966. This is the second highest score. This can be observed in Figure 27 and 28. As it can be seen in Figure 29, most of the candidate objects are focused on a small part of the eagle, on feet. It shows that there is no a common rule that states there is a strong positive correlation between size of parts of an object and the number of candidate windows. To sum up, it can be argued that localization of objects belonging to this class was respectively easier than the most classes.

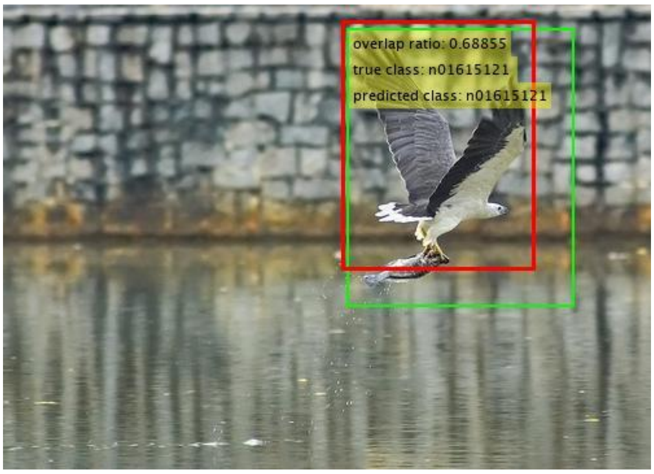

Fig. 27. Bounding box result of 81.jpeg

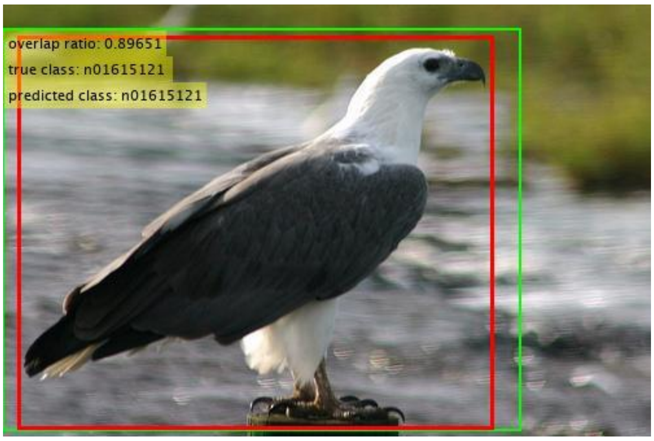

Fig. 28. Bounding box result of 82.jpeg

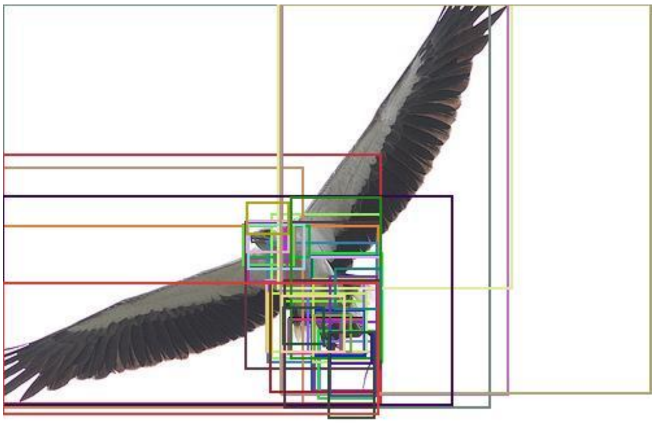

Fig. 29. All bounding box results of 85.jpeg

X. *Class of n02129604*

The average overlap ratio of this class is 0.2626 which is the lowest score among the all classes. The reason of that may be that the predicted bounding boxes tend to focus on only stripes like class of n02123159 because it can be said that the most important feature of this class are stripes. Furthermore. This can be observed in Figure 30 and 31. Furthermore, as in Figure 32, many different objects are detected as candidate object in the image which contributes to low localization rate. Consequently, it is obvious that localization of objects in this class was the most difficult among the all classes.

Fig. 30. Bounding box result of 94.jpeg

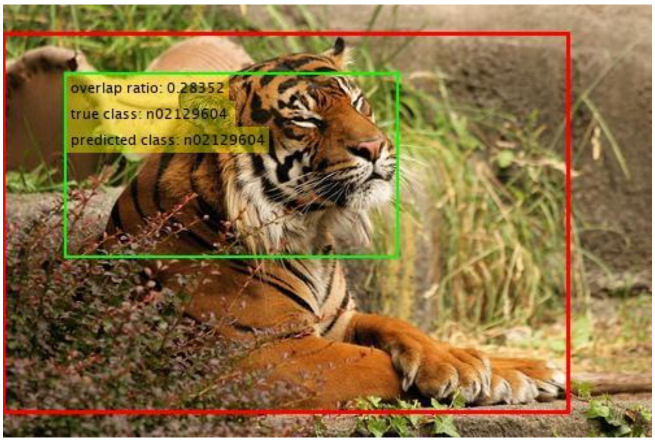

Fig. 31. Bounding box result of 99.jpeg

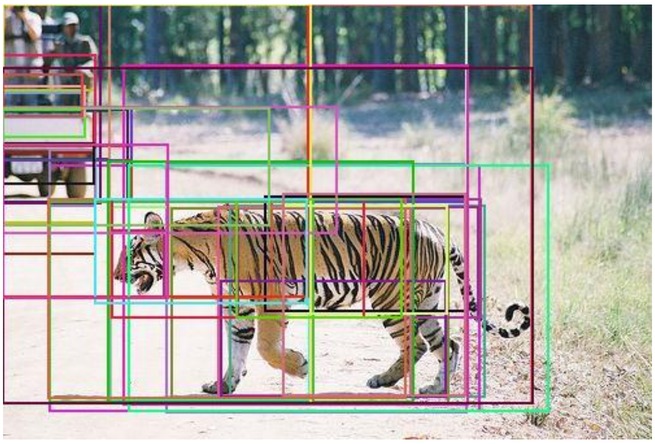

Fig. 32. All bounding box results of 97.jpeg

V. Suggestions for improvements

It can be stated that the more data for training exist, the more localization accuracy can be obtained. Moreover, if backgrounds of images in a class are consistent and flat, extraction of features can be more specific to object classes.

The more features by using different methods addition to ResNet-50 can be extracted so that more specialized features indicate classes.

If there would have been more time, parameters can be changed and tested since testing the who test dataset takes approximately 400 minutes.

VI. How to make run the system

Python’s pillow, glob, numpy, scipy, resnet, torch libraries must be installed into a Python environment with version at least 3.6.x. in order to run our system.

Conda environment cannot be used as we call python code from MATLAB.

https://github.com/pdollar/edges and https://pdollar.github.io/toolbox/ must be installed according to the instruction on their website. Matlab Computer Vision System Toolbox add-on must be installed.

Firstly, all codes must be in the same directory with test and train folders. proje.py and edgeBoxesDemo.m must be run respectively. To get the results and the resulting images scoreEval.m and boundedBoxes.m needs to be executed.

APPENDIX

CONTRIBUTION OF GROUP MEMBERS

Ezgi Çakır
She contributed to write the code of the system and write the report.

Hüseyin Eren Çalık
He contributed to write the code of the system and write the report.

Metehan Kaya
He contributed to write the code of the system and write the report.

All members completed their responsibilities and share the workload uniformly.