Object Localization and Recognition Using ResNet and Edge Box Proposals

Serhat Aras
Dept. of Computer Engineering
Bilkent University
Ankara, Turkey
serhat.aras@ug.bilkent.edu.tr

Berat Biçer
Dept. of Computer Engineering
Bilkent University
Ankara, Turkey
beratbbicer@gmail.com

Alper Şahıstan
Dept. of Computer Engineering
Bilkent University
Ankara, Turkey
alpersahistan@gmail.com

***Abstract*—This paper studies a method for accurate object localization and recognition using edge box proposals and multiple SVM classifiers for specific object types. We use a subset of ImageNet dataset: 400 train images from 10 categories and 100 test images, 10 per category. After preprocessing, samples are vectorized using ResNet implementation by PyTorch. For validation, we compare test results with truth values and correct object proposals provided to us. Experiments showed that edge box proposals give accurate localizations within reasonable time, and time complexity can be further optimized by reducing number of edge boxes obtained for each image.**

Keywords—Object localization, object classification, edge boxes, SVM, one-vs-all, ImageNet, ResNet

I. INTRODUCTION

This project aims to detect accurate object boundaries and classify them into 10 categories, which are chosen from ImageNet**[1]** database. These categories are eagle, dog, cat, tiger, starfish, zebra, bison, antelope, chimpanzee, and elephant. The data we use are preprocessed by the instructor in such a way that objects are tightly cropped from the original image. In Figure 1, we display samples from selected categories.

Figure 1. Examples from dataset.

We use a similar architecture to R-CNN[2] model for classification. The pretrained ResNet model provided by instructor is used to vectorize the data. For classification, we trained one-vs-all style binary SVM classifiers for each object type where each train image has a positive label if it belongs to the class SVM is trained to recognize and negative else.

For testing, we extracted edge boxes[3] from test image and vectorized them using ResNet. Then, we tested SVMs trained before using this data for confidence in positive class, and selected the SVM with highest confidence as predicted label. We then proceed to calculate performance metrics and report them.

For the rest of the paper we first describe our methodology in detail, then share detection results and finally finish with a discussion.

II. METHODOLOGY

A. Preprocessing

In preprocessing we normalize train samples for vectorization. Training data provided is cropped tightly from object boundaries, however image dimensions are not guaranteed to be equal. Since ResNet requires RGB images of 224 x 224 pixels, we transform each image by first apply padding with minimum number of zeros so that it becomes a square and then resizing the image to 224 x 224. Lastly, we normalize pixel intensities as follows: First, divide the image by 255 to reduce pixel values to range [0, 1]. Then, subtract mean values 0.485, 0.456, 0.406 from red, green, and blue channels respectively. Lastly, divide each channel with standard deviation 0.229, 0.224, 0.225 respectively.

B. Vectorization

After preprocessing, each train image is given to ResNet model for vectorization. After this step, samples are represented by a 2048 dimension feature vector.

C. Training SVM Classifiers

In this step, we train one-vs-all style SVM classifiers where images belonging to the class SVM is trained to recognize are labelled as positive, and the rest are labelled as negative. This way, we obtained 10 SVM classifiers for each object class.

D. Testing

For testing, each test sample is subjected to edge box method for object proposals. We then obtained the best 50 proposals and apply preprocessing and vectorization steps on them, which corresponds to obtaining tightly cropped objects similar to train samples. Next, these images are vectorized as before and given to SVMs for predictions. Note that for each image, we use 50 proposals and vectors, and obtain 2 confidence values for each vector. We repeat this process for each test image and select best prediction. Best prediction corresponds to class label and localization result, so we obtain these results as well.

E. Validation

Correct labels and localizations for test images are compared to the ones we obtain in previous step in validation part. We then compute confusion matrix, precision, recall, and f-score for each SVM, overall class prediction accuracy, and localization accuracy for each test image.

III. RESULTS

You can see the edgeboxing results for some of the test data in the figure X. We used 50 edge boxes per image with default parameters of the openCV. Also the Pollar edge's were created using the openCV model that was available online. 92% overall classification accuracy was achieved. And 43% localization accuracy was achieved overall the test images.

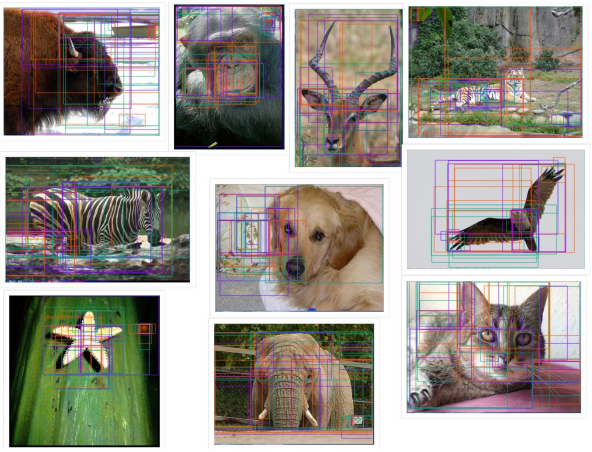

Figure 2. Edge Boxing results on test data(object)

The following enumeration is used for the classes: 0-eagle, 1-dog, 2-cat, 3-tiger, 4-starfish, 5-zebra, 6-buffalo, 7-antelope, 8-monkey 9-elephant. The corresponding results were achieved.

TABLE I. CONFUSION MATRICES

Object Type	Confusion Matrix
Starfish	'tp': 10, 'tn': 89, 'fp': 1, 'fn': 0
Elephant	'tp': 10, 'tn': 88, 'fp': 2, 'fn': 0
Zebra	'tp': 9, 'tn': 90, 'fp': 0, 'fn': 1
Dog	'tp': 10, 'tn': 89, 'fp': 1, 'fn': 0
Bison	'tp': 10, 'tn': 89, 'fp': 1, 'fn': 0
Monkey	'tp': 8, 'tn': 88, 'fp': 2, 'fn': 2
Antilope	'tp': 8, 'tn': 90, 'fp': 0, 'fn': 2
Cat	'tp': 9, 'tn': 90, 'fp': 0, 'fn': 1
Eagle	'tp': 10, 'tn': 90, 'fp': 0, 'fn': 0
Tiger	'tp': 8, 'tn': 89, 'fp': 1, 'fn': 2

TABLE II. CLASSIFICATION STATISTICS

Object Type	Precision	Recall	F-Score
Starfish	0.90	1	0.95
Elephant	0.83	1	0.90
Zebra	1	0.90	0.95
Dog	0.90	1	0.95
Bison	0.90	1	0.95
Monkey	0.80	0.80	0.80
Antilope	1	0.80	0.89
Cat	1	0.9	0.95
Eagle	1	1	1
Tiger	0.89	0.80	0.85

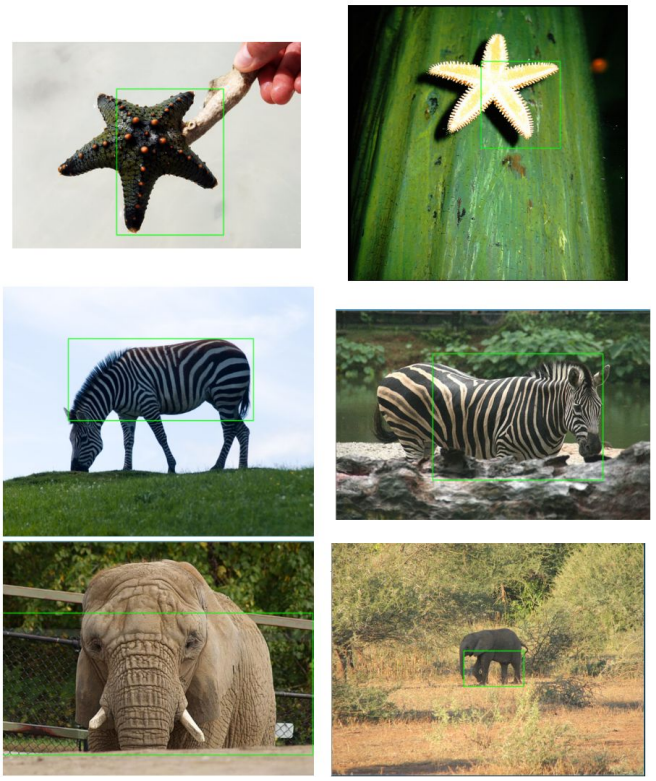

Figure 3. Localization o11

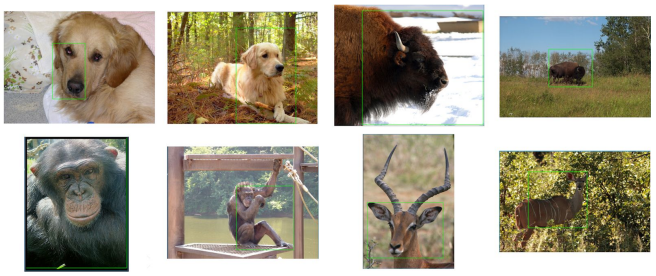

Figure 4. Edge Boxing results on test data2

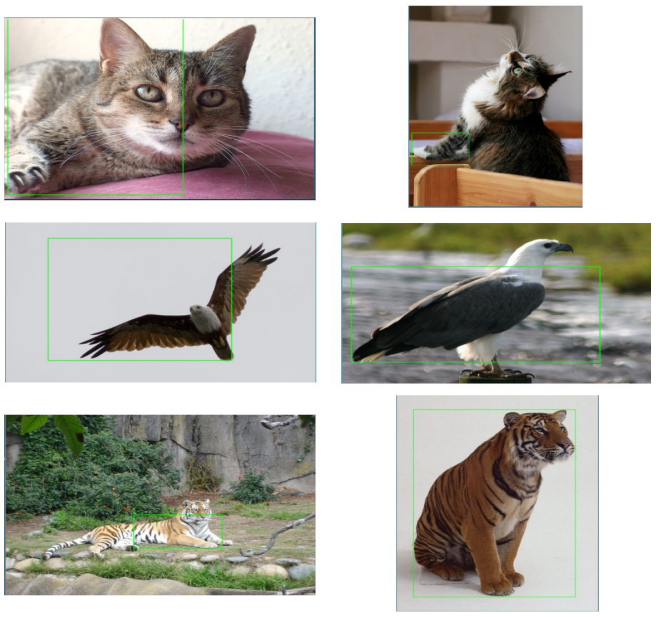

Figure 5. Edge Boxing results on test data3

IV. Discussion

Edgeboxe method requires a hyperparameter for number of rectangles created for each image. Since we have limited computational power and time, we chose to obtain 50 edgeboxes for each test image. However, one might observe that this is not enough, since some proposals are not accurate, which affects localization accuracy. Therefore, one way to increase recognition and localization accuracy is to increase number of edgeboxes created for each image.

Regarding classification, we did not experiment with SVM parameters, namely kernel, gamma, and cost of misclassification; because overall accuracy we obtained was satisfactory enough. One way to improve classification accuracy might be to experiment with these hyperparameters and the kernel to be used. Also, when an image is vectorized by ResNet, we did not apply feature selection and directly used the entire feature vector for either classification or testing. This might be problematic, since the curse of dimensionality might occur. Therefore, another way to improve classification accuracy might be to apply feature selection on ResNet features before testing. Lastly, we did not apply validation on the models because the number of samples were small. One might argue that cross validation can be applied, however, our program is already computationally expensive. Applying cross validation could cause a single execution to take hours. Therefore, another way to improve classification accuracy is to apply validation on SVMs, however, this requires high computing power and time. Another issue with overall accuracy is that since number of classes in our dataset is small, choosing inaccurate proposals did not result in significant classification error. For example, we observe that some proposals captured only part of object of interest yet that image was classified correctly. In case our dataset contains more object types, we may observe such errors leading to misclassification.

In classification, we observed that zebras and eagles were mixed by our models. This may be since objects of these types have similar color schemes. Another problem is with tigers, which had inaccurate object proposals and thus had low localization accuracies.

Lastly, it is important to note that image normalization was a problem we encountered. Without normalization, we obtained extremely low confidence values from classifiers. We suspect that since feature vectors are obtained from ResNet, without normalization intensity values do not correspond to accurate vectorization of images.

V. Division of labor

We divided the work equally amongst ourselves.

References

[1] L. Fei-Fei, J. Deng and K. Li, "ImageNet: Constructing a large-scale image database", *Journal of Vision*, vol. 9, no. 8, pp. 1037-1037, 2010. Available: 10.1167/9.8.1037.

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik, \Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," IEEE Conference on Computer Vision and Pattern Recognition, 580-587, June 23-28, 2014.

[3] C. L. Zitnick, P. Dollar, "Edge boxes: Locating object proposals from edges," European Conference on Computer Vision (ECCV), pp. 391{405, 2014.