

**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  

---

**SINGAPORE**

**GENERATING BOUNDING-BOX  
ANNOTATIONS FOR  
LARGE-SCALE IMAGE DATASET**

**ZHAO WEIMING**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**A DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE IN DIGITAL MEDIA TECHNOLOGY**

**2017**

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgement</b>	<b>iv</b>
<b>Symbols</b>	<b>v</b>
<b>Lists of Figures</b>	<b>vi</b>
<b>Lists of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Challenges and Objectives . . . . .	3
1.3 Organisation of the Dissertation . . . . .	3
<b>2 Literature Review</b>	<b>5</b>
2.1 Saliency Maps . . . . .	5
2.2 Co-segmentation . . . . .	6
2.3 Co-localization . . . . .	6
<b>3 Bounding-Box Propagation Approach</b>	<b>8</b>
3.1 Overview . . . . .	8
3.2 Sample Selection Scheme . . . . .	9
3.3 User Annotation . . . . .	9
3.4 Object Proposals Generating . . . . .	10
3.4.1 Generating Region Proposals . . . . .	10
3.4.2 Reducing Region Proposals by Saliency Map . . . . .	11
3.5 Bounding-Box Propagation . . . . .	14
3.5.1 Propagation Algorithm . . . . .	14
3.5.2 Propagation Examples . . . . .	16

<b>4</b>	<b>Bounding-Box Propagation Tool</b>	<b>17</b>
4.1	Overview . . . . .	17
4.2	File Selection . . . . .	18
4.3	Image Showing and Annotating . . . . .	18
4.4	Information Display . . . . .	20
4.5	Help Document . . . . .	21
4.6	Buttons and Output Files . . . . .	22
<b>5</b>	<b>Evaluation</b>	<b>23</b>
5.1	Experimental Setup . . . . .	23
5.2	Overall Testing . . . . .	23
5.2.1	Visual Results in Bounding Objects . . . . .	24
5.2.2	Time Consumption . . . . .	24
5.2.3	Accuracy . . . . .	25
5.2.4	Failures . . . . .	25
<b>6</b>	<b>Discussion and Conclusion</b>	<b>27</b>
6.1	Discussion on Algorithm . . . . .	27
6.1.1	Advantages . . . . .	27
6.1.2	Drawbacks . . . . .	27
6.2	Summary . . . . .	28
6.3	Future Work . . . . .	28
	<b>References</b>	<b>30</b>

# Abstract

Bounding-box in images is becoming rather useful and helpful today. We introduce an approach the propagation of bounding-box annotations for large-scale image data set.

There are two challenges for achieving the elapsed time can be extremely long when the time for processing every image accumulated, and the quality of processing cannot reach rather high.

The user would annotate some of the images selected from the dataset through GUI. Then, we use Randomized Prim's (RP) algorithm to generate many bounding-box on each picture and then get saliency map to reduce the number of bounding-box for saving time. As for comparing and propagating, we use difference hash algorithm (dHash algorithm) with Hamming distance to generate the hash set of user's annotations and find the best bounding-box. We also have the GUI tool for general users.

We evaluated our method reduce the processing time of images to about one second, and the quality is competitive with other methods. Finally, we conclude our work that it could meet the satisfaction of user for propagating bounding-box in large-scale image sets, and we give our recommendation for future works.

**Keywords:** bounding-box propagation, saliency map, dHash algorithm.

# Acknowledgement

When I was sitting in my home and finishing this dissertation, I could always remember the days I spent at Nanyang Technological University.

Whatever that day is quick or slow, a full load of working or just resting, I am always memorizing and those days were encouraging by my supervisor, dearest families, and friends.

I would like to thank my supervisor Prof. Cai Jianfei, who shows a lot of penitence and enthusiasms to me. He guides me in finishing the whole project and dissertation, and give me a lot of his suggestions.

And, I would also like to thank my friends in NTU, who give me ideas and help in reading papers and proposing new methods.

Also, I would like to thank my best roommate Hu Yuedong who was living for a whole year with me in Singapore; he gives me encourages to help me to study harder.

Then, I would like to thank my friends from "Zhihu" in Xi'an, my hometown, they give me ideas and encourages when I am feeling tired.

Finally, I would especially thank my dearest parents and families, who are caring my living and supporting my studying in Singapore.

Especially, I would like to thank Mr. Yang Jianfei, Mr. Chen Shangyu, Mr. Chen Jianda, Mr. Su Peifeng, Mr. Zhang Hongwen and Mr. Wu Tianze who give me support while finishing this dissertation.

Time runs fast; it is time to say goodbye.

Thank you, NTU.

# Symbols

$I$	The whole image-set to be tested.
$I_{RS}$	Random selected image-set.
$n_{RS}$	The number of selected images.
$n_I$	The number of whole image-set.
$\eta$	Partition for selecting images.
$n_{RS,\min}$	Minimum number of selected images.
$n_{RS,\max}$	Maximum number of selected images.
$SM$	Saliency map.
$p$	Region proposal.
$E_p$	Energy of proposal.
$I \setminus p$	The complement set of image $I$ on proposal $p$ .

# List of Figures

1.1	Bounding-box example of planes, cars and horses. . . . .	1
3.1	Overview flowchart. . . . .	8
3.2	GUI for user annotation. . . . .	9
3.3	User annotating examples. . . . .	10
3.4	Example of generating region proposals. . . . .	11
3.5	Generating saliency map. . . . .	12
3.6	Reducing bounding-box. . . . .	13
3.7	dHash algorithm step-by-step example. . . . .	14
3.8	Computing difference and assigning bits of image. . . . .	14
3.9	The Hamming distance examples. . . . .	15
3.10	Propagating example. . . . .	15
3.11	User defined ground-truth. . . . .	16
3.12	dHash algorithm propagating result examples. . . . .	16
4.1	The GUI of this tool. . . . .	17
4.2	The Evaluation part of this tool. . . . .	18
4.3	The File selection part of this tool. . . . .	18
4.4	The image showing and annotating part of this tool. . . . .	18
4.5	Instruction messages of processing. . . . .	19
4.6	User annotating examples. . . . .	19
4.7	User annotating examples. . . . .	20
4.8	The information display part while propagating. . . . .	20
4.9	The information display part when finished. . . . .	21
4.10	The help web page. . . . .	22
5.1	Part of bounding-box result. . . . .	24
5.2	Some failures. . . . .	26

# List of Tables

5.1	Time Consumption on Each Dataset. . . . .	24
5.2	Time Consumption of Processing Images . . . . .	25
5.3	Comapring accurarcy with other methods. (Jaccard score) . . . . .	25



# Chapter 1

## Introduction

### 1.1 Background

The word “bounding-box” in the digital image processing means the smallest rectangle containing the region [1]. In our dissertation, we define that region is a part of the image which includes object(s) of interest. In Fig.1.1, we show some bounding-box examples, which bounding airplanes, cars, and horses.



Figure 1.1: Bounding-box example of planes, cars and horses.

Bounding-box plays a significant role in image processing and machine learning, and there are numerous of relevant applications and researchers using bounding-box in their approaches. Like “*Rich feature hierarchies for accurate object detection and semantic segmentation [2]*” and

"Fast R-CNN [3]" by Ross Girshick *et al* using the bounding-box to finding objects and define the class of objects by their trained object-set. The biggest problem is that the method they used for generating objects' bounding-box could be rather slower than the recognition procedures. Although the quality of the final bounding-box could be relatively high, the time consumption could become long when there is a large image dataset to be processed.

Propagating of the bounding-box is passing through bounding-box(es) from few images to all remains by an automatic or semi-automatic algorithm, and it would help us to solve the problem as we mentioned above. This approach could annotate object(s) in the same dataset or category with a few or without users' annotation. After the propagation, we can get a set of objects with the same label or in the same category, and their primary objects are bounded. Thus, people could see the object of interest more directly and computers could also process these objects by bounding-box information in the image processing and machine learning field.

There are some state-of-art researches about propagating bounding-box, like "*Image Co-segmentation via Saliency Co-fusion*" [4] by Jerripothula *et al*, "*Co-localization in real-world images*" [5] by Tang *et al*, and "*Large-scale knowledge transfer for object localization in ImageNet*" [6] by Guillaumin *et al*. Their proposed methods of propagating bounding-box could not only eliminate human working in these applications, but also makes the processing more efficient and relatively accurate, especially in large image dataset.

The large dataset always includes a huge number of images, and they are classified into different categories. There are some classic large dataset like ImageNet [7], which include 3,264 classes and about 940,000 images. It is a hierarchically structured image database of images to illustrate each concept or word in WordNet [8]. Images of each concept are quality-controlled and human-annotated. So it will offer tens of thousands cleanly sorted images. And there are also some other datasets like:

- **Object Discovery Dataset by MIT** [9], which is an image dataset collected from Internet search vary considerably in their appearance and typically include many noise images that do not contain the object of interest.
- **iCoseg** [10], which contains 38 groups with 17 images per group on average, that is 643 images in total, and pixel-wise hand annotated ground-truth. It is a dataset (and annotations) available to the public for facilitating further work and allow for easy comparisons.
- **MSRC** [11], which is the only object recognition dataset with dense labeling (almost every pixel in each image is labeled) and a large number of object categories [12], and *etc.*

It is meaningful for developing a tool of this propagating of bounding-box for large-scale dataset for saving time and improving accuracy. Thus, we propose a bounding-box propagation scheme inspired by image searching in time saving and quality.

Since we are using some large datasets, we intend to decrease users work as much as possible. After computer randomly select images, the user would have their annotation on these images. Thus, we create a new approach in choosing images from the whole set of images.

## **1.2 Challenges and Objectives**

Propagating bounding-box in a large set of images is difficult. There are two challenges in achieving our approach. First, time-consuming could be hugely significant. Because processing of each image may contain several steps, even if the processing time is a little bit longer on each image, the accumulated time cannot be ignored. Thus the processing of the whole dataset would be prolonged. Second, the quality or accuracy, could not achieve relatively high in automatic propagation schemes comparing to manual annotation. Thus, we might introduce some information by users to get over it.

We build an image processing framework and use some basic concepts for propagating methods on bounding-box propagating. And we also implement a friendly graphical user interface (GUI) tool for user interaction with our proposed method. This tool would reduce users' workload and improve the accuracy and reduce time-usage of the whole processing period. And we believe users would be satisfied with our tool for propagating bounding-boxes.

## **1.3 Organisation of the Dissertation**

In Character One, we introduce the main idea of our dissertation, which includes the background, objectives, and challenges of our work.

In Character Two, we give a review of other relative researchings which include the state-of-art approaches in saliency map, co-segmentation, and co-localization.

In Chapter 3, we describe our method in detail, which includes the overview of the whole approach and explains all four main parts.

In Chapter 4, we introduce the design of our GUI tool for users and briefly introduce every part of it.

In Chapter 5, we test our method on large image dataset and give out our result on visual, time usage and accuracy. We also compare our results with other methods we mentioned in

Chapter 2 and analyze some failures of our approach.

In Chapter 6, we discuss the advantage and drawbacks of our approach in general.

In Chapter 7, we conclude the main idea, approach and test result. Then, we give our recommendations for further researches.

# Chapter 2

## Literature Review

Our method for propagating of bounding-box is associated with saliency maps, co-segmentations, and co-localization.

### 2.1 Saliency Maps

The saliency map is a topographically arranged map that represents visual saliency of a corresponding visual scene [13]. The saliency map of an image could engage two kinds of information - one is low-level contributed by contrast like color, orientation, size, motion and depth of an object in the picture and another is high-level which contains information like textures [14]. These features are rather helpful in achieving our approach, and there are many methods to generate saliency map of an image.

There are some state-of-art methods like Itti's approach [15], frequency-tuned [16], the graph-based visual saliency (GBVS) model [17], the context-aware saliency detection [18], and the minimum barrier salient object detection (MBS) [31] *etc.* [15] is a method combining multi-scale image features into a single topographical saliency map, and using a dynamical neural network then selects attended locations in order of decreasing saliency. It could improve the efficiency for it solved the complex problem by rapid selecting. [16] is an approach exploits features of color, luminance and spatial frequency information to get saliency map. [17] is another classic approach based on the method of Itti's. It forms activation maps on particular feature channels and then normalizes them in a way which highlights conspicuity and admits combination with other maps. [18] aims at detecting the image regions that represent the scene that is different from previous definitions whose goal is to either identify fixation points or detect the principal object. [31] is a highly efficient and powerful salient object detection method based

on the Minimum Barrier Distance (MBD) Transform. It is robust to pixel value fluctuation and thus can be effectively applied to raw pixels without region abstraction. They present an approximate MBD transform algorithm with 100 times speedup over the exact algorithm, and the score of the result is extremely high.

Thus, the different quality and efficiency could help us to decide which could be used later in our approach.

## 2.2 Co-segmentation

Co-segmentation was first introduced in paper "*Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs*" by Rother *et al* [19] in 2006. They used histogram matching at the same time to segment the common objects in a pair of image.

After the work of Rother *et al* in [19], many co-segmentation methods start to improve segmentation quality, like iCoseg [10]. [10] introduces an algorithm allows users to decide foreground, and guide the output of co-segmentation by users' scribbles. Meanwhile, some other groups tried to speed up the processing, like method with an optimized Markov Random Field [20] and approach for video [21]. And some other techniques enhanced the scale of simultaneously process images, like an energy-minimization approach that can handle multiple classes and a significantly larger number of images [22].

Recently, there comes up some method using saliency map to enhance the quality. One of them is the co-segmentation via saliency co-fusion by Jerripothula *et al* [4]. It uses the fused saliency information to co-segment remain images and gets competitive performance even without parameter fine-tuning.

## 2.3 Co-localization

Co-localization is a part of our work since they have the same input and output. There are many thesis and algorithms to solve this problem.

"*Large-scale knowledge transfer for object localization in ImageNet*" by Guillaumin *et al* [6], introduces an automatically knowledge transferring method in ImageNet with many more bounding-boxes. By transferring knowledge from related source classes with available annotations, they could pass this information to all ancestors and siblings. Another method is "*Co-localization in Real-World Images*" by Tang *et al* [5]. They use a joint image-box formulation for solving the co-localization problem and the convex quadratic program which can

be efficiently solved. "*Image co-segmentation via saliency co-fusion*" by Jerripothula *et al* [4] gives out a solution by saliency co-fusion that also works for co-localization, and it enhances the quality of results.

Therefore, we could refer to these methods and their ideas for achieving our approach later in our dissertation.

# Chapter 3

## Bounding-Box Propagation Approach

### 3.1 Overview

In this approach, we divide the processing to four main steps - sample selection, user annotation, bounding-box generating, and bounding-box propagation as shown in Fig. 3.1.

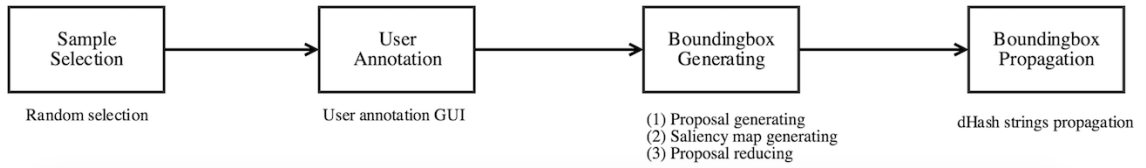


Figure 3.1: Overview flowchart.

First, we would do the sample selection right after user choosing the image folder of the dataset to be processed. We randomly select some of the images from the whole image-set. Then, our method would get user's annotation of these images by their input device, which need user draw bounding-box using mouse or trackpad to annotate the position and size of the object they want in pictures.

Then, we generate bounding-box for each image in image-set. This step include proposal generating, saliency map generating and proposal reducing, and we would discuss these in details in later this chapter. The last step is the propagation of bounding-boxes, we choose the Hamming distance of dHash strings distance comparison here to find the best bounding-box in each image.



### 3.2 Sample Selection Scheme

For selecting set of images to be annotated by user,  $I_{RS}$ , we would like to use random selecting method for the whole dataset. The number of selected images  $n_{RS}$  fits Eq. 3.1 below.

$$\begin{cases} n_{RS} = \eta n_I \\ n_{RS} \geq n_{RS,\min} \\ n_{RS} \leq n_{RS,\max} \end{cases} \quad (3.1)$$

where  $\eta$  is a selection coefficient decides how much of the whole image dataset will be selected to be annotated later, and  $n_I$  stands for the number of images in the annotated image-set.  $n_{RS,\min}$  and  $n_{RS,\max}$  represent the minimum and maximum number of selection from image dataset, respectively.

It is necessary to setting selection coefficient since users could not always know the number of images they want to process, they could control the number of images they would like to annotate by setting  $\eta$ . And it is also necessary to set the minimum, and the maximum number of user annotating image to avoid too less information except by algorithm, and on the other hand user could annotate as fewer images as possible to save their time. After we get the selected image-set, the user could draw bounding-boxes on these images to annotate the object or region of their interests.

### 3.3 User Annotation

User annotation is the second step of our method. After we get the image set  $I_{RS}$  by random selection in last procedure, we would give the user an easy-using graphical user interface (*i.e.* GUI) for their bounding regions by mouse or trackpad.



Figure 3.2: GUI for user annotation.

While annotating the images, the text beside image would tell the user how many images should be annotated. It will help the user to know the processing of whole annotating progress, and avoid that they may find this work is endless and tedious.

Operating this annotating procedure is quite easy - when selecting the region of interested, the user just needs to start from a point and drag their mouse to the end point to draw a rectangle to highlight where they would like to process. Images in Fig. 3.3(a) show the circumstance that(?) user's annotation of airplane.

However, sometimes a big image-set might have a small number of images are less or even not relevant to the keywords like in Fig. 3.3(b)., users could just to click their mouse on any pixel in that image, then our approach would not consider this image in later procedures. We would introduce the design of GUI in details in Chapter 4.

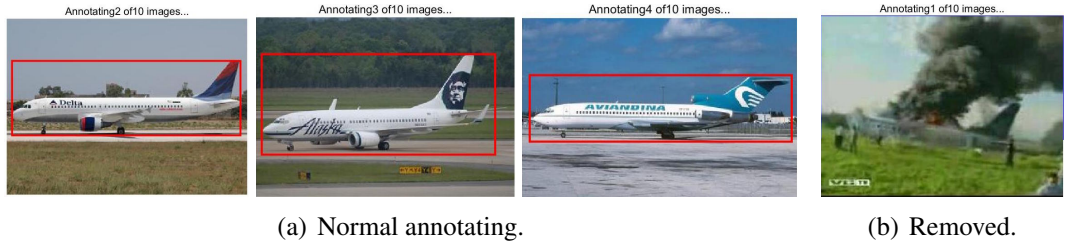


Figure 3.3: User annotating examples.

## 3.4 Object Proposals Generating

Object proposals generating is a very important procedure in our method, and it contains three main steps: generating all possible region proposals, getting saliency map of proposals and reducing proposals.

### 3.4.1 Generating Region Proposals

Object proposal generation is the first operation in this section. And there are a lot of methods to generate possible object regions in an image [23], like Objectness [24, 25], EdgeBoxes [26], Selective Search [27], Randomized Prim's [28], Bing [29] and *etc.*

We try out these approaches and find Randomized Prim's (RP's) algorithm by Santiago Manen *et al* [28] is an effective and efficient method for generating region proposals.

The Randomized Prim's algorithm using the connectivity graph of an image's superpixels, with weights modeling the probability that adjacent superpixels belong to the same object. The

algorithm generates random partial spanning trees with large expected sum of edge weights, and object localizations are proposed as bounding-boxes of those partial trees [28].

This method could produce numerous possible bounding boxes of an input image. For a 100 by 200 pixels image, it would create about 1,000 region proposals, which include the possible object as shown in Fig. 3.4(b).

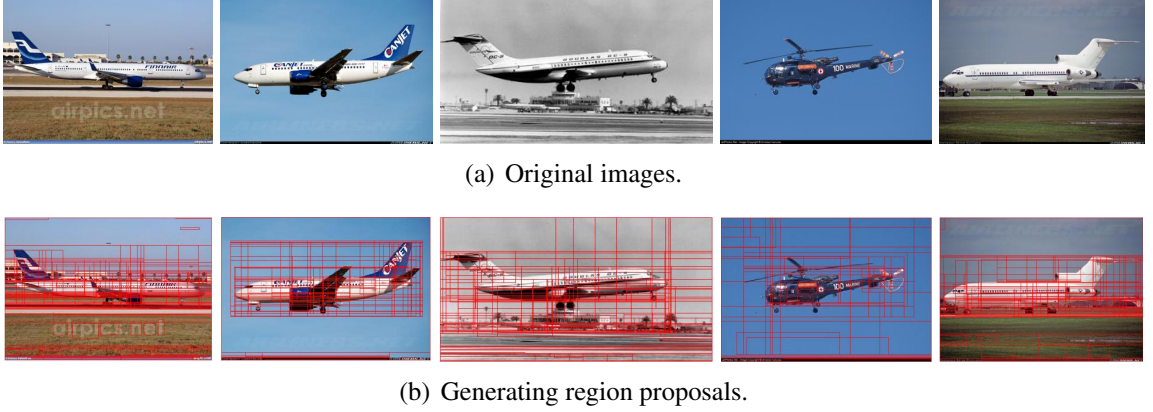


Figure 3.4: Example of generating region proposals.

From the example above, we could find that these region proposals could almost cover all of the pixels in images. Due to the enormous number of the region proposals, the computation and comparison could take very long time to each image, and the elapsed time would accumulate to rather long in processing some large scale image datasets. So we would reduce the number of region proposals from about 500 to 1000 in each image to its about 20% (*i.e.*  $\sim 100$  to 200 proposals) to reduce time consumption in our proposed method.

### 3.4.2 Reducing Region Proposals by Saliency Map

Since we want to reduce a great number of generated region proposals from generating process, we would use saliency map as an auxiliary tool to find the most important part of the image and then reduce these proposals.

The saliency map is the image that shows each pixel's unique quality. It aims to simplify or change the representation of an image into something that is more meaningful and easier to analyze. The result of saliency map is set of contours extracted from the image. Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture [30].

There are many approaches to generate saliency map of an image like Itti's method [15], frequency-tuned [16], context-aware saliency detection [18], GBVS model [17] and MBS object detection [31] as we introduced in Chapter 2. We would like to choose MBS object detection

method to produce saliency map here since it is a fast algorithm could achieve about 80 saliency maps per second.

MBS object detection at 80 FPS by Zhang *et al* [31] is a very efficient and robust salient object detection method based on the Minimum Barrier Distance Transform. Zhang *et al* speed up original algorithm 100 times to make it a rapid method. Also, the quality of generated saliency maps perform excellently in our test using quality constrained co-saliency estimation (QCCE) by Jerripothula *et al* [32] .

For decreasing the number of proposals, we generate its saliency map  $SM$  by the approach in [31] for each image to get their most salient part.

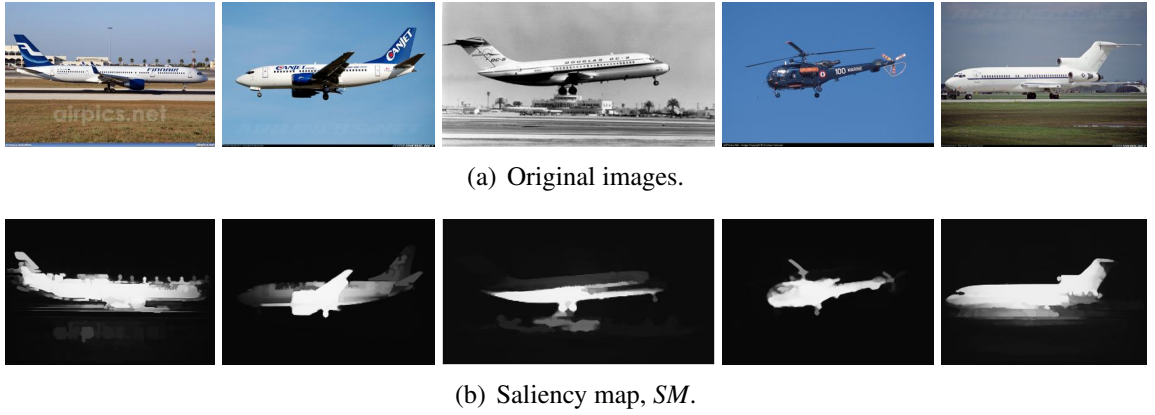


Figure 3.5: Generating saliency map.

We originally introduce the equation to calculate the sum of saliency energy as Eq.3.2 , or the average sum of the energy to the area as Eq.3.3:

$$E_p = \sum_p SM \quad (3.2)$$

$$E_p = \frac{\sum_p SM}{A_p} \quad (3.3)$$

where  $SM$  is the saliency map of the image, and  $A_p$  stands for the region area (pixels) of the region proposal.

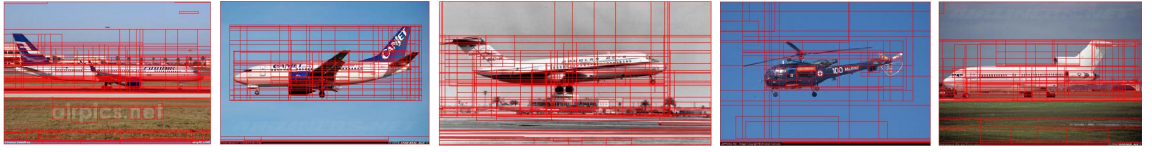
The problem is that when we use Eq.3.2, the greatest part of saliency belongs to the bounding-box who bounds the biggest area in the image, while the average energy equation (Eq.3.3) could only get the most concentrated but tiny part of the saliency map.

Thus, we compute its proposal energy  $E_p^*$  defined by Eq. 3.4. And we choose top 20% of all region proposals which contains the greatest energy. Then we could get regions bounded bounding boxes around these most salient parts in the image.

$$E_{p^*} = \frac{\sum_p SM - \sum_{I \setminus p} SM}{A_p} \quad (3.4)$$

where  $p$  is the region of proposal,  $I \setminus p$  is the relative complement of  $p$  with respect to set  $I$ . And the  $A_p$  stands for the area of region  $p$ .

We get an image with dense region proposals at first, as we could see in Fig. 3.6 (a). So, we generate the saliency map ( $SM$ ) by MBS method as we mentioned in this section as shown in 3.6 (b). Then, for each proposal, we calculate the energy using formula Eq.3.4. Thus, we can sort energy of proposals descendingly to get the top 20% energized regions. In Fig. 3.6 (c) and (d), we draw the reduced region proposals on saliency maps and images.



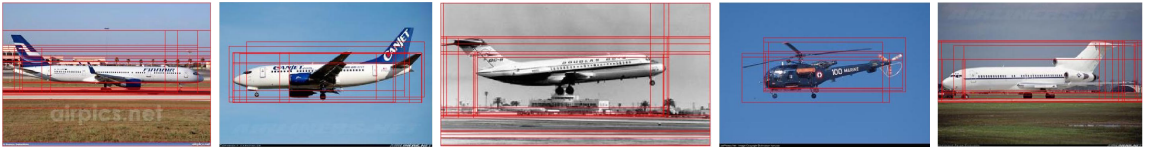
(a) Region proposals.



(b) Saliency map,  $SM$ .



(c) Reduced region proposals on  $SM\tau$ .



(d) Reduced region proposals on images.

Figure 3.6: Reducing bounding-box.

We could see from Fig. 3.6 that this method could successfully find the region proposal tightly bounding the object. Instead of selecting the meaningless biggest part or some tiny parts of the object, using this subtraction in Eq.3.4, we could get neither too big nor too small region of the image to proceed to next step - bounding box propagation.

## 3.5 Bounding-Box Propagation

### 3.5.1 Propagation Algorithm

Propagating of bounding-box is a step needs numerous of comparison and calculation. For a large image-set, it would take an extended period in processing comparing of proposals and user defined regions. Thus, we use difference hash algorithm (*i.e.* dHash algorithm) [33–35] which is fast in comparing and searching for large number of images.

The dHash algorithm has four steps - (1) reducing color, (2) reducing the size, (3) computing the difference and (4) assigning bits. We give an example of the image of a plane in Fig. 3.7 below.

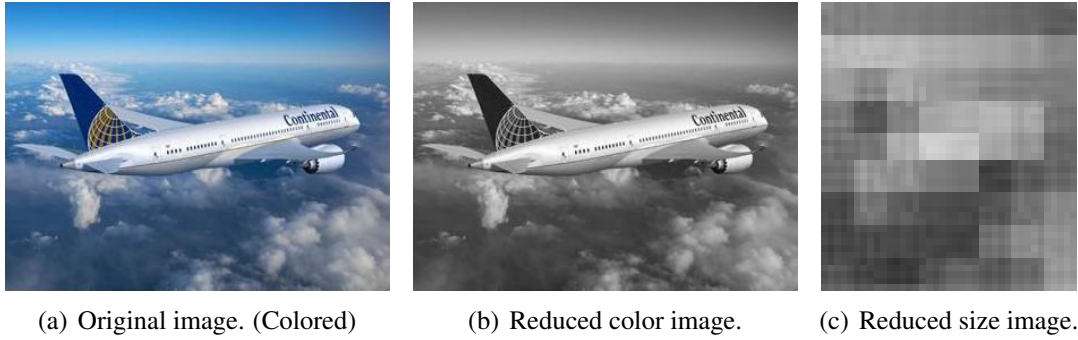


Figure 3.7: dHash algorithm step-by-step example.

(1) Reducing color. It converts the image to a gray-scale picture from Fig.3.7(a) to (b). This step changes the hash from 72 colored pixels to a total of 72 gray-scale colors.

(2) Reducing size. It could remove high frequencies and detail is to shrink the image as fast as possible. We resize it to 9x8 so that there are 72 total pixels. By ignoring the size and aspect ratio, this hash will match any similar picture regardless of how it is stretched.

(3) Computing the difference. The dHash algorithm works on the difference between adjacent pixels. It could identify the relative gradient direction. The 9 pixels per row yields eight differences between adjacent pixels. Eight rows of eight differences become 64 bits.

(4) Assigning bits. Each bit is simply set based on whether the left pixel is brighter than the right pixel. We use "1" to indicate that pixel  $P_x$  is smaller than pixel  $P_{x+1}$  and set the bits from left to right, top to bottom using big-endian as shown in Fig 3.8.

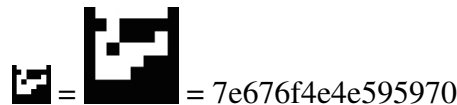


Figure 3.8 shows a small 9x8 pixel grayscale image on the left, followed by an equals sign, then a larger 9x8 pixel grayscale image on the right. Below the images is the hexadecimal hash value 7e676f4e4e595970.

$$\text{Image} = \text{Image} = 7e676f4e4e595970$$

Figure 3.8: Computing difference and assigning bits of image.



After we get the dHash strings or string-set, we could compare them using the Hamming distance. The Hamming distance between two strings of equal length (*e.g.* a pair of dHash strings) is the number of positions at which the corresponding symbols are different [36].

The Hamming distance between  
 01011101 and 01001001 is 2.  
 11000011 and 11101110 is 4.

Figure 3.9: The Hamming distance examples.

There is an example of the Hamming distance in Fig.3.9, it tells the distance between two strings of equal length and the different characters have been annotated in blue and red color.

The advantage of using dHash algorithm with the Hamming distance to propagate bounding-box is (1) Increasing or decreasing the brightness or contrast, or even altering the colors cannot dramatically change the hash value. Even complicated adjustments like gamma corrections and color profiles would not impact the result. (2) It is rather fast to process an image so that we could save time in large image dataset processing.

After we get a set of hash value of user annotations, we calculate every region of each image, and then compare the Hamming distance of every two regions as we explained in Fig.3.10 below.

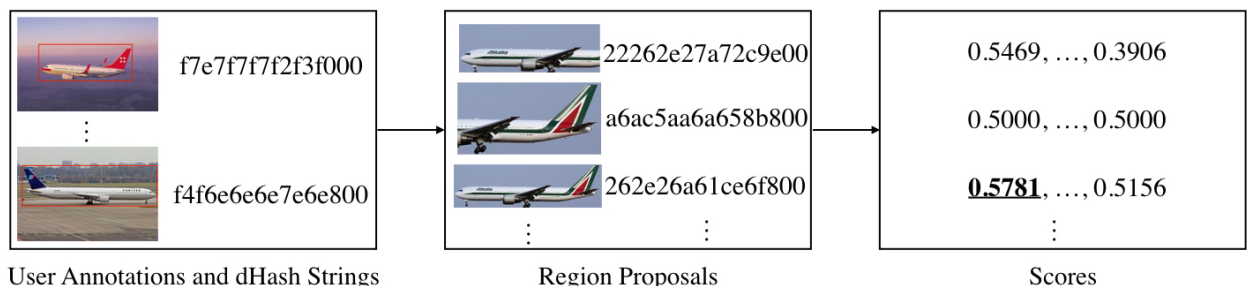


Figure 3.10: Propagating example.

For each image, we would find the best-scored proposal in all region proposals. Thus, if user annotated any object, this method would find the most match part in the other images. So, we could propagate user annotations to rest of all images in processing image-set. Due to the small computation is needed, it can be done very quickly for pairwise comparing of all region proposals to all user annotated ground-truth.

### 3.5.2 Propagation Examples

We would show an example in propagating bounding-box by our method. As you could see ten images in Fig. 3.11 and there are some user's annotation to the object shows on each image. Meanwhile, some of the images are not closely related to the class we are processing. Thus they are removed like Fig. 3.11(a) and Fig. 3.11(g).

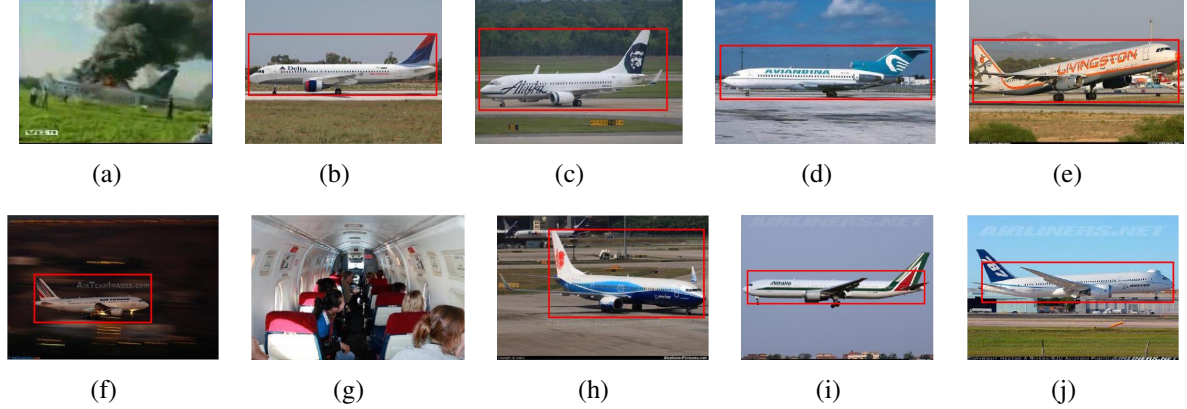


Figure 3.11: User defined ground-truth.

From user annotated images, we could get dHash strings and compare with remain images in the image-set. We could find the best-matched proposal from reduced region proposals from the last step as shown in Fig. 3.12.

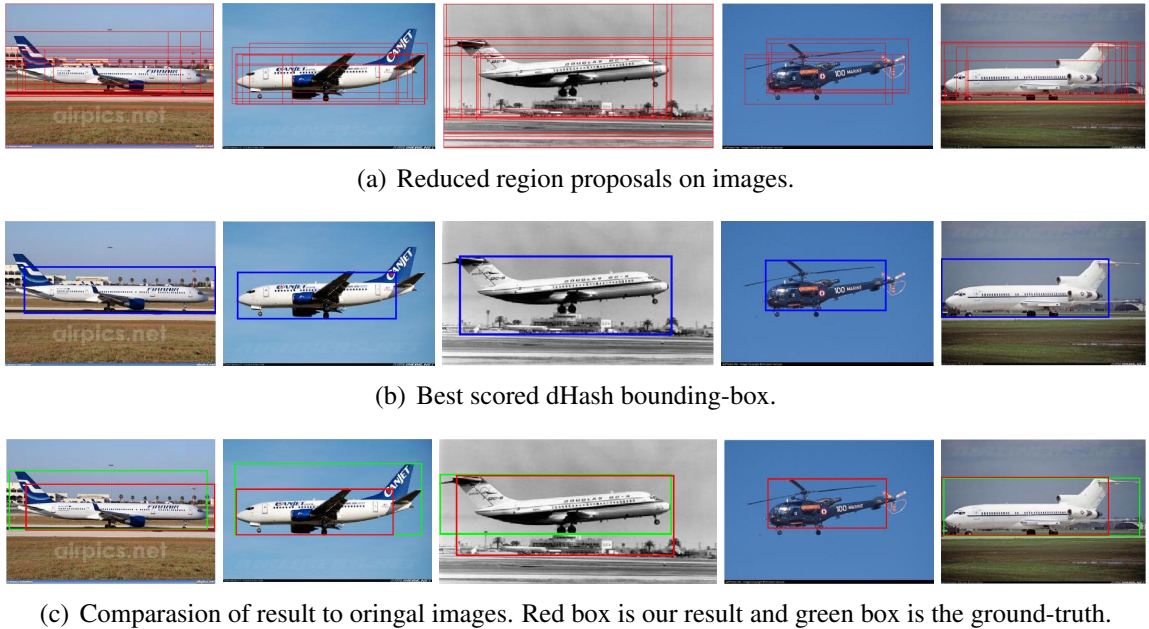


Figure 3.12: dHash algorithm propagating result examples.

We could see some of the results in this propagation, and the boxes are bounded to the airplane. And we would evaluate our method and analyze our result in Chapter 5.



# Chapter 4

## Bounding-Box Propagation Tool

### 4.1 Overview

We implement a fully designed and user-friendly MATLAB GUI based bounding-box propagation tool for users to run our algorithm. It mainly contains three parts - the file selection part, the image showing and annotating part, and the information display area. The GUI of our tool is shown below as Fig. 4.1. And we would introduce each part of our tool in details in following sections.

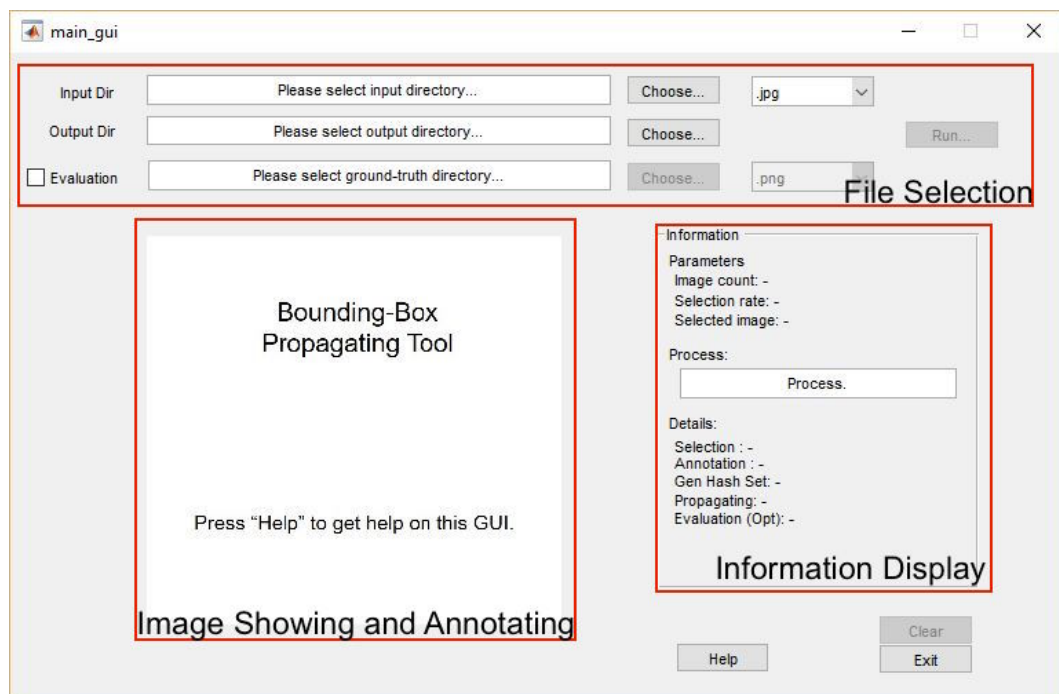


Figure 4.1: The GUI of this tool.

## 4.2 File Selection

File selection part is located at the top of our proposed GUI; the user could select the input folder of image-set, output folder to save result files. Optionally, they could also click the check box to choose the ground-truth folder for evaluating the quality of output as shown in Fig.4.2.

Fig. 4.3 demonstrates that our tool could support different formats of images in dataset like JPG, JPEG, BMP and PNG files, users could select the corresponding format of the dataset image and ground-truth image format.

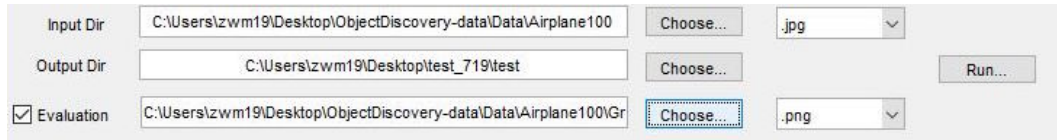


Figure 4.2: The Evaluation part of this tool.

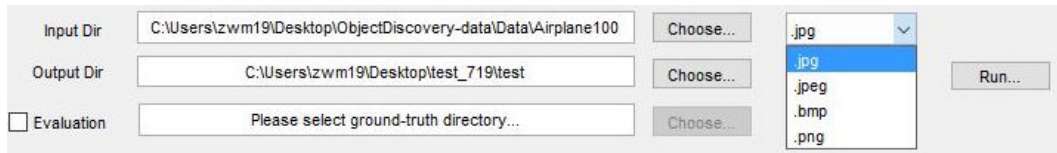


Figure 4.3: The File selection part of this tool.

## 4.3 Image Showing and Annotating

Image showing and annotating part is located at the left-bottom of our proposed GUI as shown below in Fig. 4.4 .

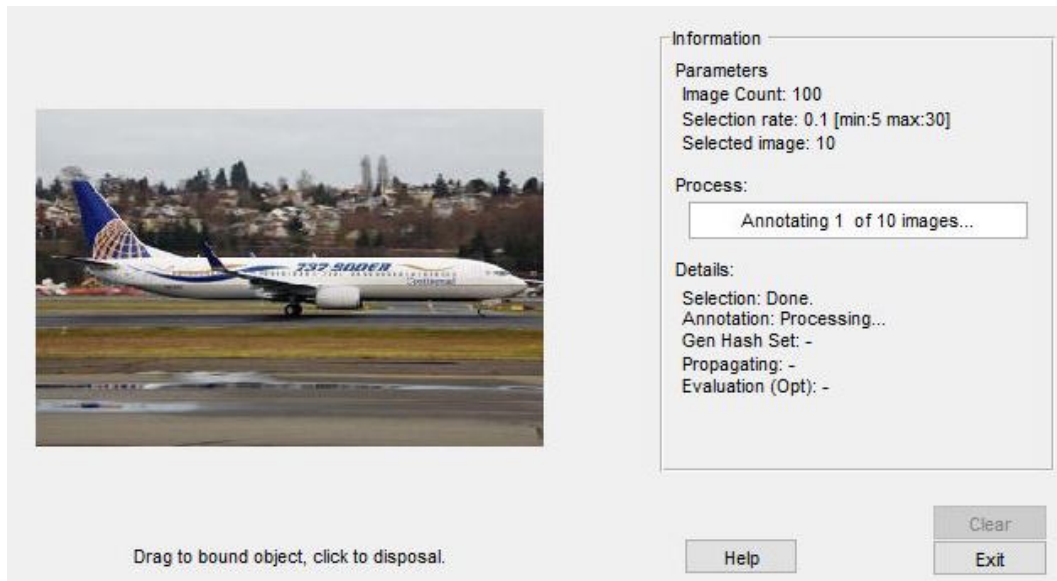


Figure 4.4: The image showing and annotating part of this tool.

Meanwhile our algorithm processing, we would show some instructions to make interaction with users better. Fig. 4.5(a) is a welcome message for users, and it would show after the initialization of our tool. And (b) is an image to confirm user has successfully annotated the pictures. (c) and (d) are the end of propagation message and the end of evaluation message, respectively.



Figure 4.5: Instruction messages of processing.

In this part of the user could also annotate or discard image by dragging a box or click, respectively.

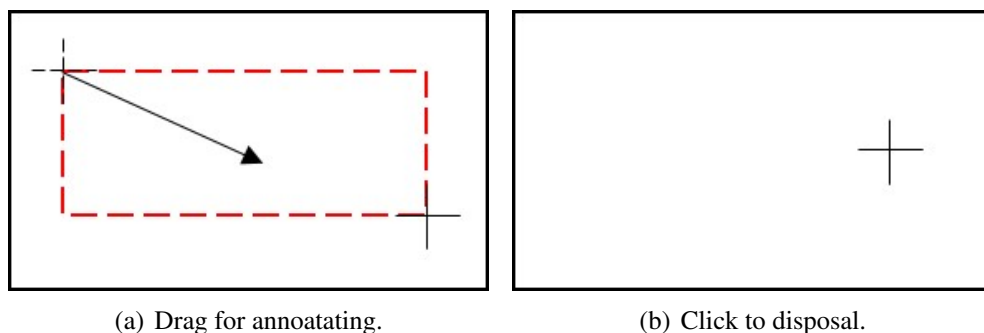


Figure 4.6: User annotating examples.

Depending on users' need, they could annotate any region for our following calculation. Fig

4.7 is an example of our annotating while using the tool.

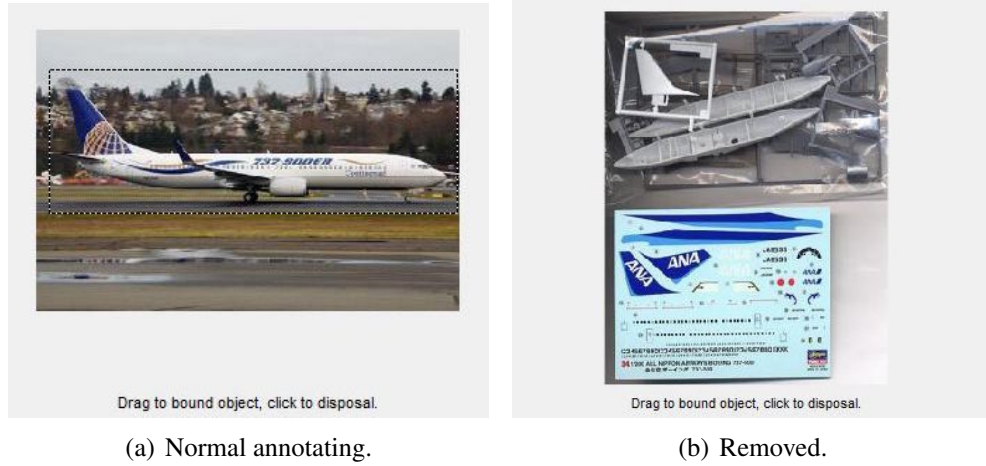


Figure 4.7: User annotating examples.

## 4.4 Information Display

Information display part is located at the right bottom of the GUI, and it would show the information as in three parts - parameters settings, processing status, and overall process in details. Parameter settings include the number of images, selection rate, the minimum and the maximum number of selected images and the final number of selected images. Processing status would show status when the algorithm is running, and it is helpful for users to know the progressing of each task. Overall process in details part shows the progress in selection, annotation, generating a dHash set, propagating and evaluation (optional), and it could help users to know which step we are running now.

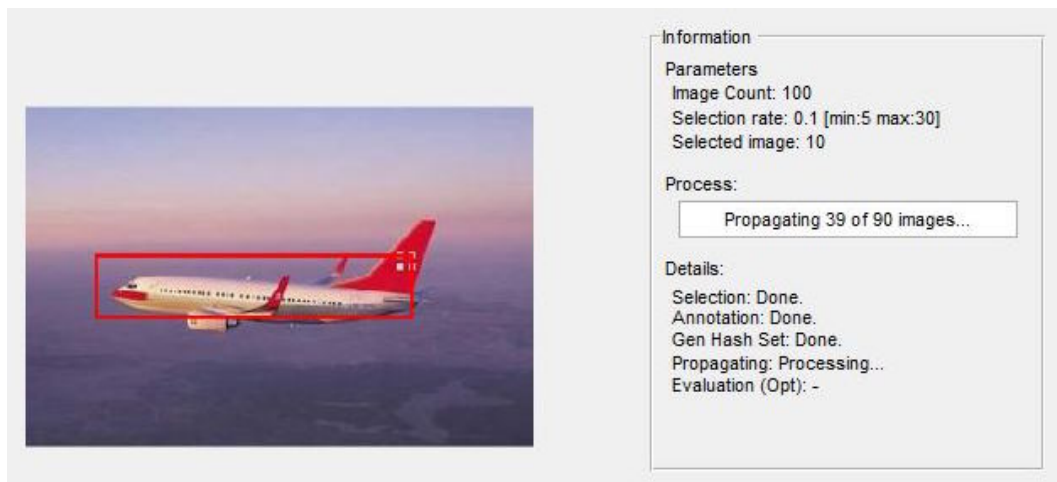


Figure 4.8: The information display part while propagating.

When finished, it will also show the results in Jaccard score if users choose to evaluate our

results as in Fig.4.9.

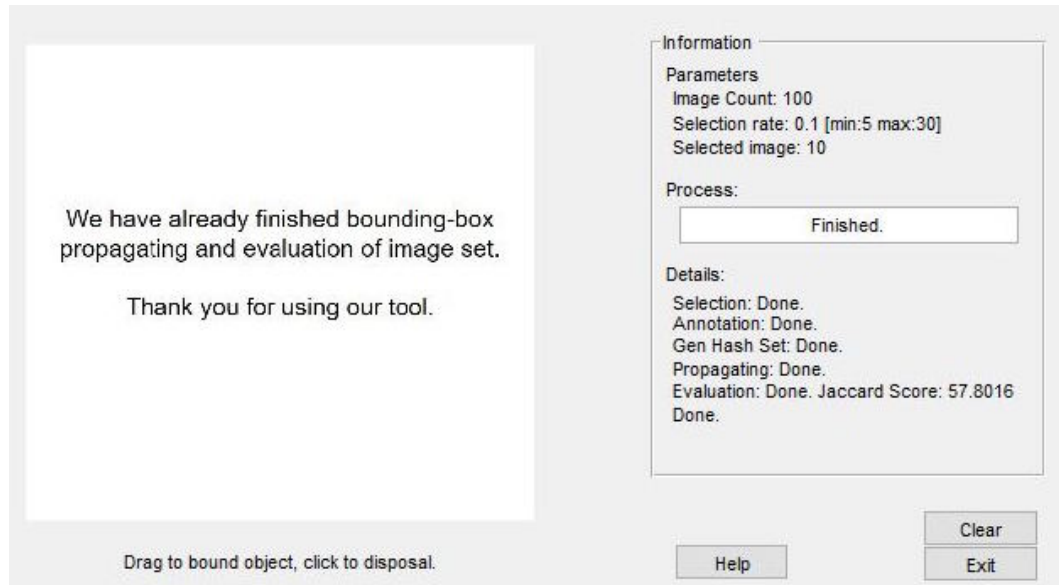


Figure 4.9: The information display part when finished.

## 4.5 Help Document

We publish the help document on the Internet. When the user clicks the "Help" button, an internal web browser of MATLAB will open the page. In this web page, we introduce how to compile the tool before using and give the instructions to users while using step by step as Fig. 4.10 shows.

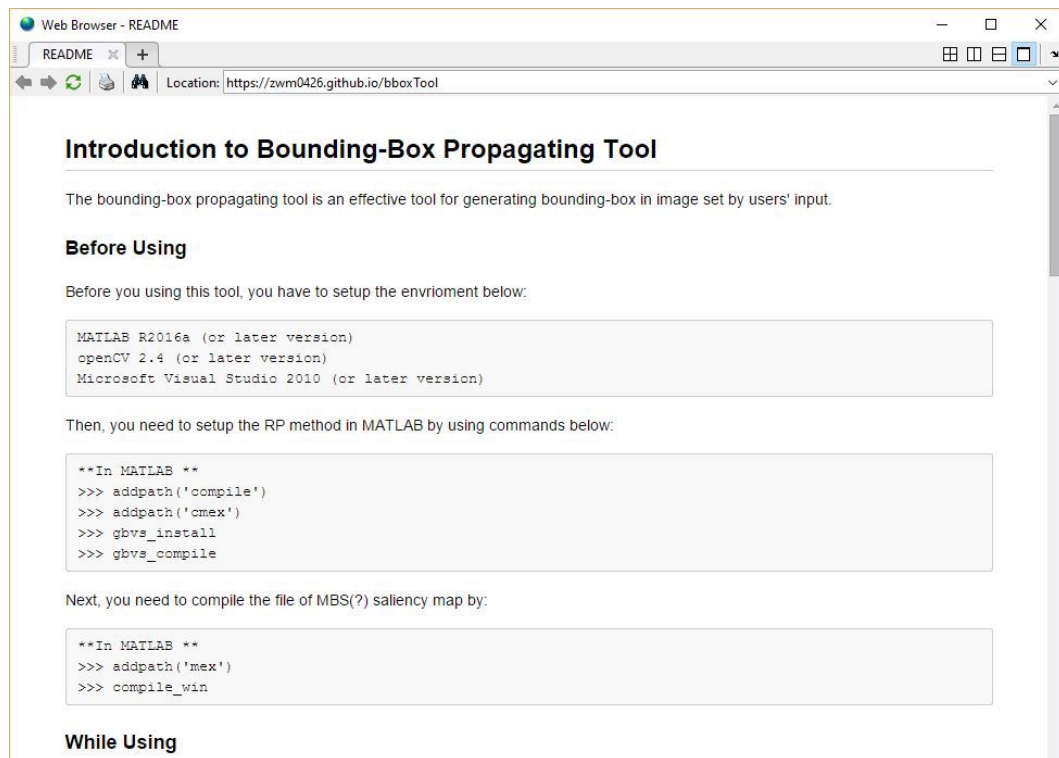


Figure 4.10: The help web page.

## 4.6 Buttons and Output Files

There are three buttons controlling the whole GUI - "Run", "Clear" and "Exit".

- "Run" button would be inactivated unless you finish choosing all of the folders.
- "Clear" button would clean all the selected folders and users could start to bounding a new set.
- "Exit" button would stop the bounding-box propagating tool and exit the GUI tool.

After processing, we would get the result in the output folder. And if the "Evaluation" is checked, the comparison between our results and ground-truth could be saved in folder "EVALUATION". The filename is set with prefix like "ANNO\_", "RESU\_" and "GT\_", which stand for annotated images, propagating result images, and evaluation with ground-truth images, respectively.

# Chapter 5

## Evaluation

### 5.1 Experimental Setup

We evaluate the performance of our proposed method on Object Discovery dataset on our PC, which includes an Intel Core i5-4278U processor 2.6GHz and 8GB RAM. We run our program on MATLAB 2015b and support with Microsoft Visual Studio 2015 Community and OpenCV version 2.4.9.

We set the parameters of our method to be  $\eta = 10\%$ . Thus we would select 10% images from the whole image-set to get user's annotations.  $n_{RS,\min} = 5$ ,  $n_{RS,\max} = 30$ , the minimum and maximum number of selected images as 5 and 30, it means we could get at least 5 images to annotate by user to ensure the effect of user and the number of images is less than 30 could make user finish the annotation in few minutes instead of very long time.

### 5.2 Overall Testing

There are some well-known image-sets like Object Discovery Dataset by MIT which include image class of plane, car, and horse as we introduced in Chapter 1. We test our program on propagating bounding-box in these datasets and compare with other methods.



### 5.2.1 Visual Results in Bounding Objects

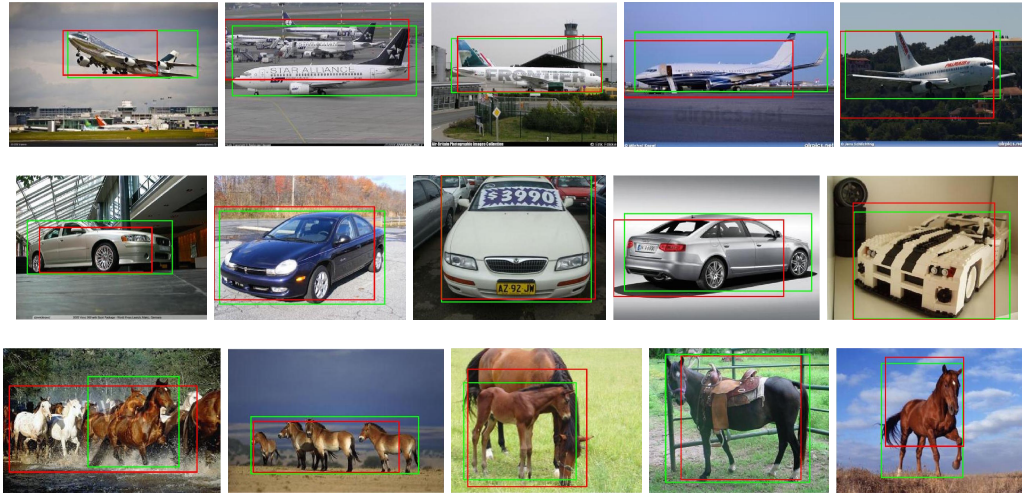


Figure 5.1: Part of bounding-box result.

We could see some of our result from Fig. 5.1 include three classes from the Object Discovery Dataset by MIT, the bounding-box of each image here is nicely bounded to the objects we want by comparing to the ground-truth they provided. In Fig. 5.1 red box is our result, and green box is ground-truth.

### 5.2.2 Time Consumption

When testing with images in each set of Object Discovery dataset by MIT (*i.e.* plane, car and horse), we record the time usage of every step in our method. Then we calculate the average time of every step for each image in Table 5.1. and Table 5.2.

Table 5.1: Time Consumption on Each Dataset.

Steps	MIT dataset(subset)			MIT dataset(full)		
	Airplane	Car	Horses	Airplane	Car	Horses
# Images	82	89	93	470	1208	810
# Annotation Images	10	10	10	30	30	30
Annotation Time	30s	26s	32s	1m 45s	2m 21s	2m 17s
Propagation Time	1m 10s	1m 24s	1m 22s	8m 23s	28m 35s	19m 12s
Evaluation Time	36s	36s	34s	4m 35s	11m 26s	6m 40s
Total Time	2m 16s	2m 26s	2m 28s	14m 43s	42m 22s	28m 9s



Table 5.2: Time Consumption of Processing Images

Item	Average Time Costs (ms)
Generating proposal regions.	~ 250ms
Generating saliency map.	~ 500ms
Generating dHash strings for each proposal and comparison between annotations and proposals.	~ 300ms

From the table above, we could see that although every processing step depends on the complexity of the image, we could reduce the time to about 1050ms for each image to find the final proposal.

### 5.2.3 Accuracy

We get the result of accuracy from comparing with the ground truth, and it is shown below in Table 5.3.

Table 5.3: Comparing accuracy with other methods. (Jaccard score)

Method	MIT dataset(subset)			MIT dataset(full)		
	Airplane	Car	Horses	Airplane	Car	Horses
# Images	82	89	93	470	1208	810
Joulin <i>et al</i> [37]	15.36	37.15	30.16	NA	NA	NA
Joulin <i>et al</i> [22]	11.72	35.15	29.53	NA	NA	NA
Kim <i>et al</i> [38]	7.9	0.04	6.43	NA	NA	NA
Rubinstein <i>et al</i> [9]	55.81	64.42	51.65	55.62	63.35	53.88
Chen <i>et al</i> [39]	54.62	<b>69.2</b>	44.46	60.87	62.74	<b>60.23</b>
Suyog <i>et al</i> [40]	58.65	66.47	<b>53.37</b>	<b>62.27</b>	<b>65.3</b>	55.41
Ours*	<b>59.91</b>	63.83	52.20	57.47	62.56	53.77

\* Results may vary from different users' input.

We could see the average accuracy of our processing is competitive to methods in Table 5.3, especially with method [9] by Rubinstein *et al*, [39] by Chen *et al*, and [40] by Suyog *et al*.

### 5.2.4 Failures

We could also find there are some failures when we are processing images in various datasets. In Fig. 5.2 we show some failures in our processing.

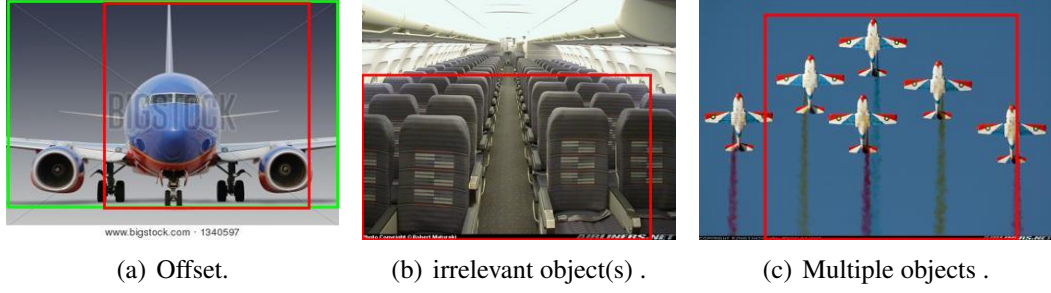


Figure 5.2: Some failures.

Failures are varied in different images, some of them are extremely poor by comparing with the ground-truth. There are some possible reasons that we could not achieve good results.

The most probable reason is that RP's method could not give the correct region proposals at first. Thus we cannot give out more accurate bounding-box even we go through all the bounding-boxes. It might cause a significant offset of the final bounding-box, like the example in Fig. 5.2(a).

As for the irrelevant object could be bound like Fig. 5.2(b), we check other's thesis to optimize this problem and find it is caused by the different approaches and features we have. For our approach, it thinks that there is not any noise in the dataset as in [40]. But, other methods like [4], they would minimize the bounding-box from a bigger image comparing their co-saliency map. When there is not a relevant object, the bounding-box would be eliminated after its size is smaller than their threshold.

Fig. 5.2(c) contains multiple objects, and that could cause the defect in our method since many objects in the image could affect the comparison between objects and users' ground truth in dHash strings set. Thus, that might cause an offset of the final bounding-box.

# Chapter 6

## Discussion and Conclusion

### 6.1 Discussion on Algorithm

#### 6.1.1 Advantages

Our propagating algorithm performs well in cutting down the time consumption since we use a rather fast procedure in every step of processing. And the quality of the bounding-box could be compatible with other state-of-art methods. There are some advantages of our proposed method:

- Time consumption is significantly reduced. Due to the past knowledge, processing on images especially pixel level can be prolonged, and when image-set is large, thus the time usage would accumulate dramatically. Since we embedded fast procedures in our approach, we could decrease the processing time of each image to about only 1 second on average.
- Accuracy is competitive. By comparing with other methods, we could see our method accuracy is increased in some aspect. It could be helpful to some other application like object recognition, face detection and other fields of processing these images.

#### 6.1.2 Drawbacks

Although there are some advantages of our method, we could still find some drawbacks:

- Although the quality is competitive with other methods, it is still not too high in value. Since there are some images not interested by the user, but we give bounding-box on that picture, this can be a great negative effect on the quality, Jaccard score, since the ground-truth we compared is null.

- Due to its limitation of our algorithms, numerous of possible region proposals produced by RP's algorithm cannot produce the closest bounding-box at the first stage. So, we cannot find the best bounding-box even our algorithm propagate user's annotations as we analyzed in the last chapter.
- Our approach depends on users annotation, it may cause the result is different every time. Due to this feature, the user could have some errors in drawing bounding-box, then this flaw could be exaggerated. Finally, it may contain great errors and deviations that we could not expect.

## 6.2 Summary

In this dissertation, we proposed a method to propagating bounding-box in large image-set and implement a useful tool for general users.

There are four main steps of our method, they are sample selection to get images to be annotated, user annotation to get ground-truth, region proposal generating to produce possible bounding-box, and bounding-box propagation to ensure the final bounding-box on the object.

Then, we implement a well-designed GUI tool for running our algorithm in a more user-friendly way. Later, we evaluate our method on each class in Object Discovery dataset by MIT, and the result of testing is processing time only 1 second on average for each image and processing quality is competitive with other methods like [9], [39], and [40].

Thus, our method is fast in processing and could get better quality on large image-set bounding-box propagation.

## 6.3 Future Work

From our discussion in Chapter 5, we have some recommendations for future work on achieving bounding-box propagating method:

- For achieving better quality in propagating bounding-box, we have to achieve an approach, which is more accurate than RP's algorithm to generating possible bounding-box at the first step to ensure the quality in later steps.
- We could also enhance the condition of the result in following procedures, like take a better use of saliency map of images.

- We could give more attention to our failure analysis; it could be solved when we found a high quality and efficient way to get other image features and running comparison with users' ground-truth.
- While we shorten time usage, we sacrificed some of the accuracies in processing. It is because we need to pursue the time-quality balance. We could add some feedback from the result to expand a new loop of low-score matching images, to calibrate them for a better result. So for other projects, we could check the requirements and conditions to re-balance time and quality.

# References

- [1] Minimum bounding box - wikipedia. [https://en.wikipedia.org/wiki/Minimum\\_bounding\\_box](https://en.wikipedia.org/wiki/Minimum_bounding_box). Accessed February 27, 2017.
- [2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [3] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [4] Koteswar Rao Jerripothula, Jianfei Cai, and Junsong Yuan. Image co-segmentation via saliency co-fusion. *IEEE Transactions on Multimedia*, 18(9):1896–1909, 2016.
- [5] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1464–1471, 2014.
- [6] Matthieu Guillaumin and Vittorio Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3202–3209. IEEE, 2012.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.
- [8] Adam Kilgariff. Wordnet: An electronic lexical database, 2000.
- [9] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1939–1946, 2013.

- [10] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3169–3176. IEEE, 2010.
- [11] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *Computer Vision–ECCV 2006*, pages 1–15, 2006.
- [12] Tomasz Malisiewicz and Alexei A Efros. Improving spatial support for objects via multiple segmentations. 2007.
- [13] Saliency map - scholarpedia. [http://www.scholarpedia.org/article/Saliency\\_map](http://www.scholarpedia.org/article/Saliency_map). Accessed July 3, 2017.
- [14] Introduction of saliency map - disp lab. <http://disp.ee.ntu.edu.tw/class/Saliencymap.ppt>. Accessed July 3, 2017.
- [15] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.
- [16] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, pages 1597–1604. IEEE, 2009.
- [17] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2007.
- [18] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012.
- [19] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 993–1000. IEEE, 2006.
- [20] Dorit S Hochbaum and Vikas Singh. An efficient algorithm for co-segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 269–276. IEEE, 2009.

- [21] Junsong Yuan, Gangqiang Zhao, Yun Fu, Zhu Li, Aggelos K Katsaggelos, and Ying Wu. Discovering thematic objects in image collections and videos. *IEEE Transactions on Image Processing*, 21(4):2207–2219, 2012.
- [22] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 542–549. IEEE, 2012.
- [23] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really? *arXiv preprint arXiv:1406.6962*, 2014.
- [24] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.
- [25] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2189–2202, 2012.
- [26] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.
- [27] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.
- [28] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim’s algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2536–2543, 2013.
- [29] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3286–3293, 2014.
- [30] Saliency map - wikipedia. [https://en.wikipedia.org/wiki/Saliency\\_map](https://en.wikipedia.org/wiki/Saliency_map). Accessed June 25, 2017.
- [31] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Minimum barrier salient object detection at 80 fps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1404–1412, 2015.



- [32] Koteswar Rao Jerripothula, Jianfei Cai, and Junsong Yuan. Qcce: Quality constrained co-saliency estimation for common object detection. In *Visual Communications and Image Processing (VCIP), 2015*, pages 1–4. IEEE, 2015.
- [33] Github - maccman\_dhash\_ compare image similarity with a dhash. <https://github.com/maccman/dhash>. Accessed June 19, 2017.
- [34] Detecting duplicate images using python - the iconfinder blog. <http://blog.iconfinder.com/detecting-duplicate-images-using-python/>. Accessed Jun 19, 2017.
- [35] Kind of like that - the hacker factor blog. <http://www.hackerfactor.com/blog/?/archives/529-Kind-of-Like-That.html>. Accessed June 19, 2017.
- [36] Hamming distance - wikipedia. [https://en.wikipedia.org/wiki/Hamming\\_distance](https://en.wikipedia.org/wiki/Hamming_distance). Accessed July 5, 2017.
- [37] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1943–1950. IEEE, 2010.
- [38] Gunhee Kim, Eric P Xing, Li Fei-Fei, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 169–176. IEEE, 2011.
- [39] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2027–2034, 2014.
- [40] Suyog Dutt Jain and Kristen Grauman. Active image segmentation propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2864–2873, 2016.