Selective Search for Object Recognition

Hannan Syed, *McGill University, COMP 417*

***Abstract*—This paper evaluates the selective search algorithm implemented by J.R.R. Uijlings et al. The selective search algorithm addresses the problem of object recognition. In particular the selective search has emphasis on the inherit hierarchical structure of images. This is done by combining segmentation for object recognition with exhaustive search. The advantage of exhaustive search is that is aims to capture all object locations, and the advantage of segmentation is that it uses image structure to guide the search for object locations. The selective search results in a small set of data-driven, class-independent, high quality locations. The results of selective search have been outstanding with exceptional scores across the Pascal Image challenges. This paper evaluates external potential challenges where the algorithm may fail to recognize an object. These instances may include camouflaged object, which may be obvious to a human but not so much to the selective search algorithm.**

***Keywords*—Object recognition, selective search, segmentation, exhaustive search, hierarchical image structure.**

I. Introduction

The task of object recognition is the task of determining objects through the use of computer vision [4]. Objects are things with well defined boundaries and centers [14]. This means objects such as cars, cows, and spoons as opposed to background things like the sky or grass [14]. There are many challenges when trying to decipher an object in an image. Differences between the color of two similar objects, or a difference in texture of two objects of the same color, etc. For this reason the researchers have defined three characteristic used for determining objects. They argue that any object has at least one of three deterministic characteristics: (i) a well-defined closed boundary; (ii) a different appearance from their surroundings; (iii) sometimes it is unique within the image and stand out as salient [16].

However, even while trying to identify boundaries or textures, object recognition runs into problems surrounding humans. In specific, humans are a challenge because of the variety of poses that they can adopt as well as the diversity of human appearance [13]. One example concerning human recognition is a face over a sweater. This example showcases regions with different characteristics which can be combined into one object after the object is determined to be a human [16]. Dalal et al. agreed on the fact that the first step to be taken is creating 'a robust feature set that allows the human form to be discriminated cleanly, even in cluttered backgrounds...' [13]. This idea has lead to much progress for object recognition as a new approach to the task was adopted, where localisation was done through identification of an object [16].

Object recognition can be accomplished using a variety of techniques. In general the task of object recognition is completed by using a database of objects and images and applying a variety of algorithms to the image, such as matching, learning, or pattern recognition algorithms. Common techniques for these algorithms include edges, gradients, Histogram of Oriented Gradients (HOG), among others [4]. In particular the researchers for selective search combined the breadth of an exhaustive search as well as the specificity of segmentation. This combination decreased the number of locations in an image compared, to only running and exhaustive search, and allowed the Uijlings et al. to apply stronger learning algorithm techniques on a Bag-of-Words (BoW) model for object recognition.

Fig. 1: An image of a farm which show the hierarchical structure of images

Representing an image as a BoW model requires that an image be treated as a document. This is a three step process: feature detection, feature description, and codebook generation [3]. A technique used for generating the dictionary is using a histogram based representation on independent features. This is similar to the techniques used in the selective search research. The use of visual words is done by using jumping windows, which means that the relationship between the visual word and the object location is learned in order to predict test images [16].

It is also important to note the core idea surrounding segmentation for selective search in particular. This idea is the hierarchical property found in images. In the figure below we see that the the image has some hierarchical features such as the background with a barn and trees and the foreground with horses. This shows how images are built with objects overlaying one another. This leads to the idea that since the object to be recognized may be the barn rather than the horses, a segmentation procedure which uses multiple scales for segmentation is necessary.

With the progress that has been made with selective search for object recognition, there are still a few caveats when it comes to object recognition. Such as identifying objects through a silhouette or camouflage. We first outline the related works, then the algorithm and what it is comprised of, then evaluate it (what works and what it doesn't work on), and finally conclude.

II. Related Work

It is fair to explore the concepts of this paper through the two main perspectives the algorithm is comprised of. The first being exhaustive search and the second being segmentation. The reason for this is that these two concepts were combined together to create a novel selective search algorithm which had not previously been tested.

A. Exhaustive Search

Exhaustive search essentially relies on searching through every possible position in the images, because an object could lie at any point in the image [16]. However, the underlying problem with this type of search is that it is computationally expensive. Although eventually an object will be recognized, it is a strained search. It is especially futile to search through all of the objects possible locations if the object cannot be recognized in the end. This is a problem which can occur with images of camouflaged figures or silhouettes where the program will search through the entire image only to return the image itself rather than any specific objects within it.

Much of the related work to object recognition with respect to brute-force search is using the sliding window technique [12]. The sliding window algorithm for object detection or recognition is often used with histograms [8]. A histogram is a discretized distribution that measures the frequency of quantized visual features, i.e. pixels/colors [8]. Additionally, the sliding window is an objective function which 'slides' across the image. However, this method as with all exhaustive search methods have limitations when it comes to computational cost.

The problem with the sliding window is that for an image of size nxn, a window of size rxr and a histogram of dimension B, any generic algorithm will scan n^2 windows, and r^2 pixels per window [8]. This makes the complexity of the algorithms $O(n^2(r^2+B))$. Despite this, many solutions have been provided for creating effective Histograms to reduce the factor of r^2, therefore resulting in a complexity of $O(n^2B))$ [8]. The problem then lies with the size of B. When B is large, the sliding window technique does not scale as well and in addition high dimensional histograms have become the standard for solving computer vision problems [8].

A work related to object recognition is object localization, using the sliding window algorithm by Felzenszwab et al [17]. Their method uses a linear SVM and HOG features to perform an exhaustive search for object recognition. This combination results in an impressive object detection performance. Additionally, Wei et al [8] propose an efficient histogram-based sliding window method which aimed to reduce the high computational costs of high dimensional histograms. They did this by introducing a SVM with a non-linear kernel and a bag-of-words model. Their method impressively was helpful in object tracking, where tracking real-time running is necessary. However, it is not always the case that a linear versus non-linear SVM kernel perform better. Lambert et al [18] proposed another method which used an appearance model to guide the search. An appearance model is a computer vision algorithm for matching a statistical model of object shape and appearance to a new image [9]. However, despite Lambert and his team obtaining impressive results for linear classifiers, found that their method was not as effective for non-linear classifiers.

Nevertheless, the selective search method attempts to use the hierarchical structure of images to identify objects [16]. The selective search method also does not perform a complete exhaustive search because it does not brute force the image blindly. Also, since there are no sliding windows, it means that there is no need for a fixed aspect ratio or coarse grids, which may use weaker classifier to compensate for computational cost. Additionally, the selective search method is not limited to objects, but should also be able to recognize "grass" or "sand [16]. Finally, the selective search as opposed to exhaustive search methods is less computationally intensive which allows the use of stronger machine learning algorithms.

B. Segmentation

The approaches to object detection and recognition are filed into three categories: top-down, bottom-up, or a combination of the two [15].The top-down approach includes a training stage which acquires class specific features within the image in order to define the object. The bottom-up approach however starts from low-level image features, such as edges and segments. The method then builds up an object hypothesis from these features. It is ideal to attempt to use both a top-down and bottom-up approach, such as Borenstein et al. who attempts to enforce continuity along segmentation boundaries to align matched patches. However, there are problems with a combined top-bottom approach, where a promising hypothesis can result in false positives when features are locally extracted and matched [15].

The benefit of bottom-up approaches is that they follow a hierarchical build up of edges and gradients to identify objects. As such, both Carreira and Siminshisescu [19] and Endres and Hoiem [20] proposed using a bottom-up approach to segmentation, via class independent object hypotheses. An example of this comes simply from humans themselves. Humans, despite lacking knowledge of different types of objects, are able to successfully localize objects. The method put forth by Endres and Hoiem proposed to guide each step of the localization process with estimated boundaries, geometry, color, and texture [20]. These regions after a few more processes such as seeding, are put through learning techniques which rank the regions, such that the top-ranked regions are likely to be different objects.

The idea with the class independent object hypotheses is that the methods generate multiple foreground and background segments. They then learn to predict the likelihood that the foreground segment is in fact an object [16]. Additionally, both methods count on a single strong algorithm for identifying good segmented regions [16]. A major advantage of segmentation in these proposed methods is their ability to delineate objects within images. Contour detection is a technique that is extremely powerful and state-of-the-art when it comes to segmentation, but despite that, it is not ideal for selective search [16]. The reason for this is that a contour detector is computationally expensive [21]. Instead, the selective search method deals with a variety of of image conditions, using many different grouping criteria and representations.

Overall selective search as opposed to exhaustive search uses segmentation as selective search yielding a small set of class independent object locations. Additionally, instead of using the best segmentation algorithm, selective search opts to use a variety of strategies to deal with segmentation in order to reduce the computational complexity of the algorithm. Finally, with segmentation a bottom-up grouping procedure is used to generate object locations instead of learning the objectness of randomly sampled boxes [16].

III. Selective Search

This section outlines the details around the selective search for object recognition. As the algorithm utilizes segmentation as selective search, there are a variety of different strategies used in order to deal with variations in image conditions.

The three main categories are: Capturing all scales, diversification, and fast computability [16]. With respect to capturing all scales, because object within an image can occur at any scale and in addition these objects can have blurry boundaries, the selective search algorithm deals with them using a hierarchical algorithm. Diversification relates to the idea that there is no single optimal strategy for grouping regions together [16]. An object may appear differently due to texture, color, or lighting, among other conditions. Therefore, selective search employs various strategies which can conquer many situations instead of one which works in most situations. The final category is that selective search is fast to compute. As selective search hopes to be used in a practical framework, it is important to decrease computational costs where possible so that the task of object recognition is not a bottleneck for the algorithm.

A. Hierarchical Grouping

Selective search uses hierarchical grouping as the basis for its segmentation process. Hierarchical grouping has previously been implemented using a hierarchical region tree by Arbelaez and Fowlkes [22], however this was done by using a single state-of-the-art segmentation strategy as opposed to the variety of strategies used in selective search. The hierarchical concept represents a bottom-up approach to segmentation which is beneficial because the way it groups regions for segmentation is inherently hierarchical. The major advantage of hierarchical grouping is that is allows the search to capture all scales, because the bottom-up segmentation grouping naturally generates locations at all scales by continually grouping regions until the whole image becomes a single region.

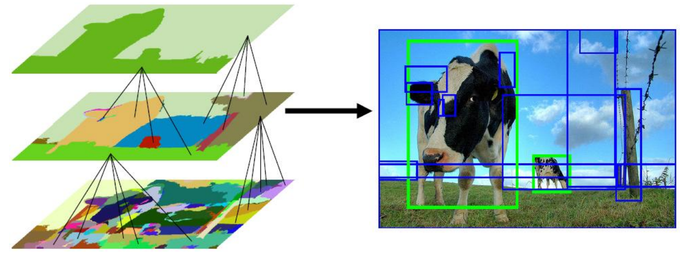

Fig. 2: An example of bottom-up segmentation

The grouping procedure, from Felzenszwab and Huttenlocher [23], is used to create initial regions as seen in the figure above. A greedy approach is used to iteratively group regions together where similarities between neighboring regions are computed, then the two most similar regions are grouped together, and new similarities are calculated between these combined regions. This process is continued until the whole image becomes a single region. When comparing similarities between regions a variety of techniques are used which all have one underlying component that they are fast to compute.

B. Diversification Strategies

Diversification of strategies allows selective search to deal with all cases of images, by diversifying the sampling of images and by using complementary strategies for segmentation whose locations are later combined. The first method of diversification is using a variety of color spaces with difference invariance properties. This means that the hierarchical grouping is done with multiple color spaces. The use of different color space accounts for changes in scene and lighting from image to image. The color spaces used increase in invariance: RGB, grey-scale, Lab, the Hue channel H from HSV, and several others. All of the invariance attributes are listed in the table below [16].

The second method of diversification is using complementary similarity measures. These measurements are all fast to compute and fit within a range of 0-1. The first of four measurements is the similarity of color. This measurement is compared by taking the color histogram for each color channel. The histograms are normalized and the intersection of the two histograms are compared. The second measurement is a similarity of texture. The texture is represented using a SIFT-like measurement. Scale-invariant feature transform (SIFT) algorithm which detects and describes local features in images [10]. One method of implementing SIFT is to transforms an image into a large collection of feature vectors. The benefit of SIFT is that each of these images is invariant to image translation, scaling, and rotation [10. The benefit of SIFT is that it acts like the inferior temporal cortex which is responsible for object recognition in primate vision. This gives rise to segmentation by creating class invariant hypotheses. The other two measurements are size and regional fit. The size measurement ensures that object locations at all scales are created for all areas of the image. The fit measures how well regions fit into each other. If two regions are close together and fill the holes in each other, then it is more plausible to merge these two regions.

Finally, the third diversification strategy is to vary starting regions. The algorithm provided by Felzenszwab and Huttenlocher [23] was used as it proved to be the most computationally efficient for yielding high quality starting locations.

C. Object Recognition using Selective Search

This section aims to outline how it all ties together for object recognition. A common feature used in object recognition is a HOG. However, related works with HOGs have been utilized with linear classifiers for exhaustive search. Selective search opts for a different approach, where and Bag-of-Words was used with the addition of some more powerful features such as a SIFT color descriptor and finer spatial pyramid division [16]. The classifier which is employed is an SVM with a linear histogram intersection kernel. An approximate classification strategy was taken from [24], which works fast and efficiently with a Bag-of-Words.

IV. Evaluation

In this section we will evaluate the selective search demo.m file. Attempts to implement basic improvements to the code provided by Koen et al. were to no avail as much of the free code was not open sourced. For this reason I decided to attempt an analysis discussing where this algorithm can potentially be improved by testing cases where selective search had not recognized an object which it should have.In the evaluation I aim to create create my own image set to challenge the selective search and evaluate its performance descriptively.

The descriptive analysis will test the potency of the selective search algorithm by identifying whether more challenging images outside of the 2010 and 2007 Pascal Image tests can be correctly identified. The reason for this unorthodox method of testing is because the original paper already covered a plethora of different testing strategies, proving that the algorithm was extremely robust and performed with high consistency, producing results of 99 percent accuracy [16]. Some of the evaluation methods originally used included testing individual and combinations of diversification strategies, testing a flat versus hierarchical approach to segmentation, and testing the quality of initial locations for segmentation, among some other tests.

A. The Code

It is essential to understand the code which was provided by Koen et al. for their selective search algorithm. There are three parts to the demo.m provided. The first is the added dependencies to run the algorithm, the second is the main part which runs the selective search with a variety of parameters, and finally the last portion opens up GUIs with sets of JPG images of the boxes and the rectangles of the image selection process.

The two main dependencies added are related to the methods developed by Felzenszwalb and the other is called anigauss, which is an anisotropic gaussian filter. Felzenszwalb's method

Algorithm 1 Segmentation as Selective Search [8]

$colorTypes$: Hsv, Lab, RGI, H, Intensity
$similarityFunctions$: ColorTextureSizeFill
k : Size of initial segmentation
$minSize$: k
$sigma$: 0.8
$images$: Image Set for Testing

Initialize:
  im=$images[1]$
for n = 1 to length($colorTypes$) **do**
  $colorType \leftarrow colorTypes(n)$
  [boxes blobIndIm blobDoxes hierarchy] ←Image2HierarchicalGrouping(im, sigma, k, minSize, colorType, simFunctionHandles)
  $boxes \leftarrow$ BoxRemoveDuplicates($boxes$)
end for

ShowRectsWithinImage($boxes, 5, 5, im$)
hBlobs←RecreateBlobHierarchyIndIm(blobIndIm, blobBoxes, hierarchy(1))
ShowBlobs(hBlobs, 5, 5, im)

run very efficiently, running in $O(nlogn)$ time [23]. The method is based on selecting edges from a graph, where each pixel corresponds to a node in the graph. The weights on the edges measure the similarity between pixels. This graph is used as a greedy segmentation technique. The code for this section is non-accessible from Koen's source code. Additionally, anigauss is a recursive anisotropic Gauss filtering. Anisotropic filtering is a method of enhancing the image quality of textures on surfaces that are far away and steeply angled with respect to the point of view [11].

The next part of the code applies a number of diversification strategies. The parameters for this method allow of a variety of colour space, similarity measures, and thresholds [16]. These parameters control the number of hierarchical segmentation's which are combined, which allows a variety of generated object hypotheses. The code allows different color types to be used, such as RGB and HSV. It allows several segmentation strategies to be used and the use of multiple thresholds for the Felzenszwalb segmentation algorithm. As referenced from the demoPascal2007.m code, which was also provided by Koen, the average minimum number of initial segmentation's was 200. The remainder of the code loads the images from a directory, then proceeds to perform the selective search using the multiple colors and merging parameters.

A difference between the demo.m code and the Demo2007PASCAL.m code is that the 2007 code also uses a learning algorithm with I had set up with the 2007 Pascal image database. However, the use of that is to time how long the image recognition process takes. This feature was not necessary for the evaluation of the selective search method.

B. Results

The tests were put together in two steps, the first being cases which are known to work, essentially creating a base case for each category of testing and the second being a cases which are try to break the algorithm. The categories include images which have camouflaged objects, silhouettes, or a cluster of similar objects such as food items.

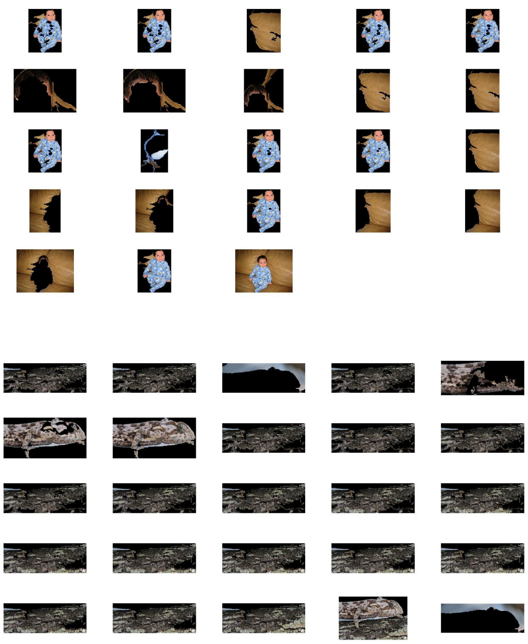

Fig. 3: Boxes and Blobs of Base Case Object Recognition

1) Base Cases: The images which were used in this section were a bike and a baby for general testing. For the camouflage base case two obvious images were used: a chameleon which was on a branch, and two people in camouflage gear lying on the ground. The reasons for using these images were that they provided equally difficult counter parts which attempted to break the selective search algorithm. The base case images can be seen in the the Appendix, or in the images folder provided with this report.

We can begin this descriptive analysis by first understanding the results of the selective search on the baby. The code returns images of the boxes which it created attempting to single out the object within the image. This print out allows us to verify the results of the search. This boxed print out result was a feature in the demo.m code which was not found in the demoPascal2007.m code. The demoPascal code was tested regardless, assuming an average size for the image being 500x500, the recognition was generally fast even for a practical setting, completing in around 1-2 seconds.

The boxes for the baby image correctly identified the baby by outlining its shape against the sofa. Similarly the other base images, such as the bike and the chameleon on the branch were correctly identified as can be seen in the figure above. The object recognition can be identified by looking at the black outlining of an object within an image. The algorithm was able to differentiate the objects from their surroundings using a variety of distinguishing features. The baby has both a different color and texture from the sofa so this task was easy, the chameleon was a little tougher however that too was correctly identified despite having a similar color, because the texture was differentiable.

2) Test Cases: Similarly the test images comprised of people and a chameleon which were camouflaged, a more clustered photo of food items, silhouettes of a bike, among other images. In the test cases the camouflage was the most difficult set of images for selective search to recognize. Secondly, were the silhouette images which posed a challenge to the algorithm. However, the cluster of objects fared well as many of the fruits in the images were identified by the algorithm.

A picture of a burger and fries were used in the base case tests to see if the objects would be correctly differentiated. The algorithm was able to separate the fries, meat patty, and the chicken nuggets, however it did not correctly identify the the burger as a whole. In order to add complexity to the cluster of food, an image of an array of fruits were used, however this image performed well as many of the fruits were correctly identified.

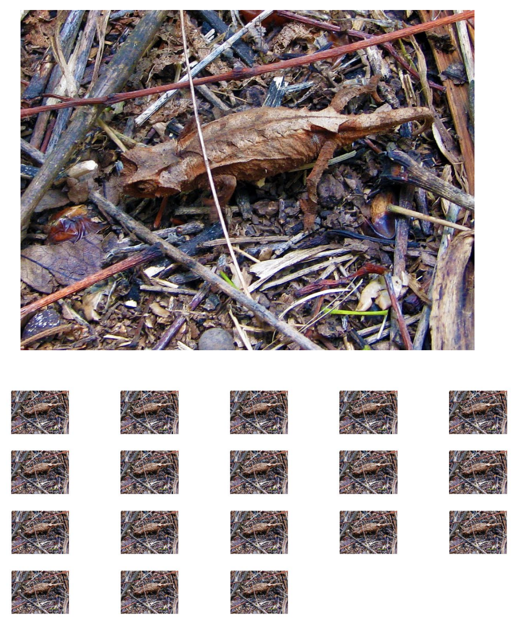

Fig. 4: Boxes and Blobs of Test Case for Camouflage

The other test cases however proved more challenging for the algorithm. With regards to silhouette images, two were taken into account. The first being a man walking against the sunset, and the second were two kids riding on a bike. These images were selected due to their similarity to the base case images. Despite a silhouette having a definitive outline, the algorithms both struggled to correctly identity an all black figure in the center of the image. While testing the human silhouette, I had realized that the HSV color filter was a lot stronger in recognizing the object in the image than were the remainder of the filters. However, with the image of the bike, even with the HSV filter, it failed to correctly separate the bike from the two kids sitting on them.

Finally, the toughest test for the algorithm was working with camouflage images. This is certainly understandable as even humans have some trouble recognizing camouflaged insects or people. The first image was a chameleon which was similar to texture and color to its surroundings on the ground of a forest. Generally the algorithms boxes which are printed out attempt to identify the object in the image, however in this case the boxes returned the entire image as seen in figure 4 above. Additional tests were done with camouflaged people, some obvious to the human eye and other not so obvious. None of the these images fared well at all. The entire image was returned without any indication of recognition attempt. These images posed the greatest challenge for the algorithm because of the challenge to differentiate between texture and color.

V. Conclusion

The main premise of the paper was to understand the benefits of segmentation as selective search and to test its implementation against a more challenging set of images. It is clear that selective search is a very high functioning algorithms capable of correctly identifying certain classifications of images. After understanding that the structures of images is inherently hierarchical selective search was implemented in order to correctly identify locations of objects using a diverse number of complementary and hierarchical grouping strategies for segmentation. After running the code for multiple test images, it seems that this method is very beneficial in identifying clustered objects within an image. The selective search strategy despite not being able to correctly identity a camouflaged chameleon, was able to cycle through the object locations an identify leaves and branches in the picture. The diverse grouping strategies allow for this by generating object hypotheses independent of object-class.

As a result of the the experiments run in the original paper we know that selective search was superior to other tested algorithms in terms of quality of object locations. The major benefit of selective search in addition to providing better object locations is that it also does this faster. This is a key element when working with object recognition softwares.Even while testing the challenging images, all of the results were found extremely fast, however it did take time for MATLAB to print out all of the boxes and blobs that the algorithm generated. On the other hand, if importance on the speed and computational cost of the algorithm was not as much of an issue, it is possible that using the state-of-the-art segmentation algorithms or even a non-linear kernel for the SVM learner could have improved the object recognition capabilities of the algorithm.

Nevertheless, in terms of object recognition, the algorithm in the original test proved that selective search can be used successfully for normal instances of object recognition. A difference between the original tests and the ones run in this paper were that the paper used a descriptive analysis of what the algorithm saw. A set of images which the object could learn from were not provided for the evaluation of this paper. Instead the algorithm was run directly for its object location potential. In any case, it performed well for all easily identifiable images. The algorithm struggled most with camouflaged images, the most surprising was the image of the chameleon which appeared to be easy to the human eye. With other tests such as the silhouette the algorithm provided better results depending on which individual color setting were used. The HSV color filtered images provided a better sense of the objects location. This would potentially mean that having an even larger variety or parameters for colors could improve the potency of the algorithm.

Appendix A
Image Set for Evaluation

This image set was complied using Google Images and is provided at this in the folder accompanied by this paper or at this website http://cs.mcgill.ca/ hsyed2/selectivesearch/Object-Recognition.

The images folder in the zip file are the original test images used. The rendered folder holds the image which were gathered after the code was run. The image and rendered names correspond to each other by name.

Acknowledgment

I would like to acknowledge Uijlings, J. R. R. et al. for their work on ”Selective Search For Object Recognition”, which was the basis for this paper.

References

[1] G. Hager, 2016. [Online]. [Accessed: 21- Apr- 2016].

[2] ”Object recognition for robots”, MIT News, 2016. [Online]. [Accessed: 21- Apr- 2016].

[3] ”Outline of object recognition”, Wikipedia, 2016. [Online]. [Accessed: 21- Apr- 2016].

[4] ”Object Recognition - MATLAB”, Mathworks.com, 2016. [Online]. [Accessed: 21- Apr- 2016].

[5] ”Object Recognition”, Colombia University. 2016. [Online]. Available: http://www.ee.columbia.edu, ee4830/notes/lec11.pdf. [Accessed: 21- Apr- 2016].

[6] ”Image segmentation”, Wikipedia, 2016. [Online]. [Accessed: 21- Apr- 2016].

[7] 2016. [Online]. Available: https://homes.cs.washington.edu/ bilge/remote.pdf. [Accessed: 21- Apr- 2016].

[8] Y. Wei and L. Tao, Efficient histogram-based sliding window, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[9] ”Active appearance model”, Wikipedia, 2016. [Online]. [Accessed: 21- Apr- 2016].

[10] ”Scale-invariant feature transform”, Wikipedia, 2016. [Online] [Accessed: 21- Apr- 2016].

[11] ”Anisotropy”, Wikipedia, 2016. [Online]. [Accessed: 21- Apr- 2016].

[12] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In ICCV, 2009.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.

[14] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In CVPR, 2010.

[15] L. Wang, J. Shi, G. Song, and I.-F. Shen, Object Detection Combining Recognition and Segmentation, Computer Vision – ACCV 2007 Lecture Notes in Computer Science, pp. 189-199.

[16] Uijlings, J. R. R. et al. "Selective Search For Object Recognition". International Journal of Computer Vision 104.2 (2013): 154-171. Web.

[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. TPAMI, 32:16271645, 2010.

[18] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. TPAMI, 31:21292142, 2009

[19] J. Carreira and C. Sminchisescu. Constrained parametric mincuts for automatic object segmentation. In CVPR, 2010.

[20] I. Endres and D. Hoiem. Category independent object proposals. In ECCV, 2010

[21] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. TPAMI, 2011.

[22] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In CVPR, 2009.

[23] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient GraphBased Image Segmentation.

[24] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In CVPR, 2008.