Image Cropping Based on Saliency and Semantics

by

Jiang Lin

B.A.Sc., Simon Fraser University, 2014

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Applied Science

in the
School of Engineering Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2017

Approval

Name:	**Jiang Lin**
Degree:	Master of Applied Science
Title:	**Image Cropping Based on Saliency and Semantics**
Examining Committee:	**Chair:** **Andrew Rawicz** Professor
	Ivan V. Bajić Senior Supervisor Professor
	Jie Liang Supervisor Professor
	Parvaneh Saeedi Internal Examiner Associate Professor
Date Defended/Approved:	November 29, 2017

Ethics Statement

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

a. human research ethics approval from the Simon Fraser University Office of Research Ethics

or

b. advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

c. as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016

Abstract

This thesis proposes a new automatic image cropping technique and a platform for subjective image quality evaluation on mobile devices. Image cropping is a widely used technique in the printing industry, photography and cinematography. The proposed cropping method considers both the low-level pixel properties and high-level semantics. It is a combination of saliency-based and semantics-based image analysis. In the end, we compare the proposed method with a conventional saliency-based strategy. Furthermore, in order to simplify the final subjective test, we developed an iOS based mobile application for subjective image quality evaluation. The developed application implements two-alternative forced choice (2AFC) test methodology and further reduces the cognitive load of subjects performing the test by providing an easy-to-use, natural interface using the mobile device's touch screen. The test results show the proposed cropping technique performs significantly better overall compared to saliency-based cropping.

Keywords: image cropping; saliency-based; semantics-based; subjective test; iOS; two-alternative forced choice

Acknowledgements

I would like to express my gratitude to Dr. Bajić for his guidance, supervision, patience, and mentorship throughout the studies and research of my thesis. Working under his supervision was a confidence-building experience. He gave freely of his time and provided the strategic and knowledgeable support I required to complete this thesis. I couldn’t have done it without his constant support.

I would like to thank Dr. Saeedi for reviewing this thesis carefully and examining my thesis. I would also like to thank Dr. Liang for providing the valuable comments on this thesis, and Dr. Rawicz for chairing the thesis defense.

Table of Contents

List of Tables

List of Figures

List of Acronyms

2AFC	Two-Alternative Forced Choice
AWS	Adaptive Whitening Saliency
DAPC	Describable Attribute for Photo Cropping
DMP	Deformable Parts Model
GBVS	Graph-Based Visual Saliency
HDR	High Dynamic Range
IKN	Itti-Koch-Niebur
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
JPEG	Joint Photographic Experts Group
LDR	Low Dynamic Range
LED	Light-emitting diode
R-CNN	Region-based Convolutional Neural Network
SBPC	Sensation-Based Photo Cropping
SCSM	Sparse Coding of Saliency Maps
SVM	Support Vector Machine
TMO	Tone Mapping Operation
UHDTV	Ultra-High-Definition Television
YOLO	You Only Look Once

Chapter 1.

Introduction

Image cropping is widely used in the printing industry, photography, and cinematography. It is a well-known fundamental operation for improving the quality of photographs, by removing the distracting content from a photo, changing its aspect ratio, and enhancing the overall composition [1, 6]. In this thesis, a new automatic image cropping algorithm is presented, and a subjective test is presented to evaluate the performance of the new algorithm. A large number of image cropping methods have already been developed, but the proposed algorithm considers both bottom-up visual saliency and high-level semantics, which distinguishes it from existing cropping techniques.

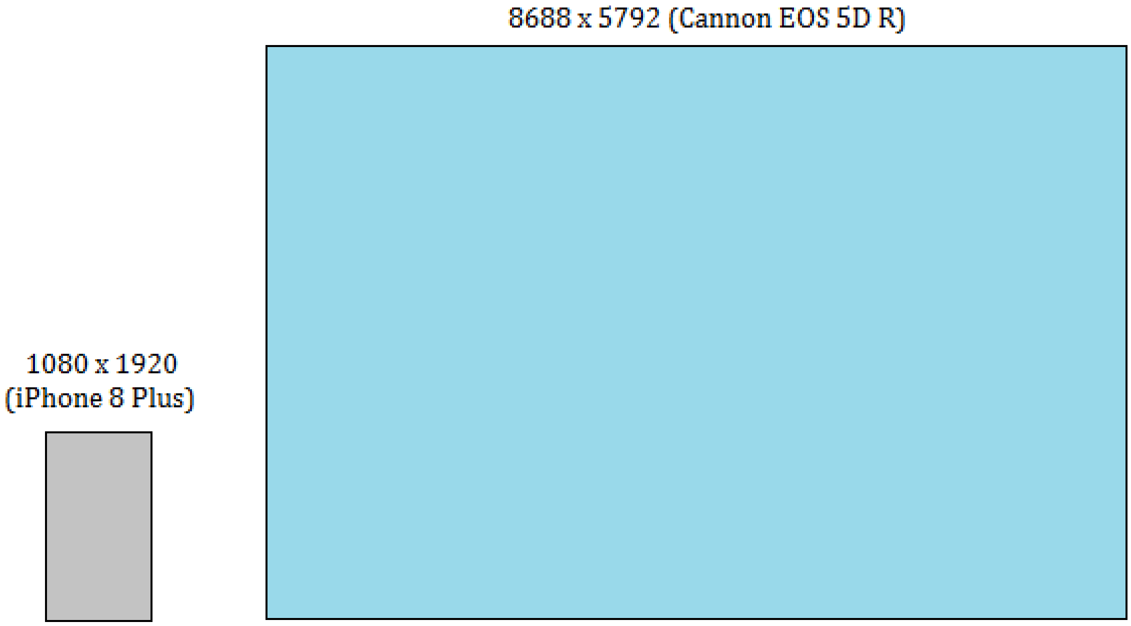


Figure 1: The resolution comparison between iPhone 8 Plus display and Cannon EOS 5D R imaging sensor

In recent years, the resolution of mobile device displays has been growing. Figure 1 shows the resolution, 1080×1920 pixels, of the latest iPhone (iPhone 8 Plus), which was released in September 2017. However, the resolution of imaging sensors is growing at an even faster pace. As Figure 1 shows, in 2015, Cannon released the EOS

5D R camera, which could take a photo with the resolution of 8688×5792 pixels. This illustrates a challenge where the image is too large to be displayed on a screen in its native resolution. Even if the native display resolution matches or exceeds image resolution, the small physical size of the mobile screen may reduce the size of important objects and features in the image to the point where they are not clearly visible. The same happens when a high-resolution image is downsized to match the screen resolution. Automatic image cropping techniques have been developed to address these challenges, as well as other applications such as thumbnail creation and image summarization [63, 64, 43]. The existing automatic cropping methods ([1-3, 41-43, 46-50]) were mostly developed and tested on low-resolution images, usually smaller than 1920×1080 pixels. Therefore, automatic cropping of high-resolution images is still a relatively unexplored topic.

Throughout this thesis, we have focused on the cropping of the high-resolution images. The lowest resolution of the image we collected in our dataset is 1920×1280 pixels, and the highest is 5760×3840 pixels. These resolutions are less considered in the literature on automatic image cropping. We implemented the proposed algorithm in MATLAB. First, we designed a system based on bottom-up saliency and high-level semantics. Then we developed multi-resolution techniques for these building blocks to deal with various sizes of objects of interest relative to the image resolution.

Second, we developed an iOS based mobile application [38] for subjective image quality evaluation using objective-C in Xcode to simplify the final subjective test.

Third, we tested the cropped images using the subjective quality evaluation mobile application. We compared cropped images produced by our method against those produced by a saliency-based method [20], and found that the proposed images were preferred by participants.

In this chapter, we will first review the existing automatic cropping techniques. Then we will give an overview of methods used in the proposed algorithm: saliency estimation and semantic image analysis. Finally, we will summarize the contributions of the thesis.

1.1. Existing Cropping Techniques

Many techniques have been developed for image resizing/retargeting. Image retargeting [56, 57] has been proposed to deal with the mismatch between the native display and image characteristics, by using scaling, seam carving and cropping techniques. Simple scaling could fix the mismatch, but some details may be lost when downsizing the image. Seam carving is also a popular technique in image retargeting. At first, each pixel of the image is assigned an energy value using energy function which can be visual saliency, entropy and so on. A seam is a path of low energy pixels in an image that connect the top boundary to the bottom boundary, or the left boundary to the right boundary. The image size can be changed to any size by repeatedly removing or inserting the seams [69]. Seam carving usually changes the structure of the object in the image, which sometimes makes the objects look unnatural. Therefore, in this thesis, our retargeting method is only concerned with how to crop the image, without scaling or deforming objects in it.

Many computational methods [1-3, 41-43, 46-50] have been proposed to crop an image automatically. Existing automatic image cropping algorithms can broadly be classified into aesthetics-based and attention-based [43]. Aesthetics-based methods attempt to preserve artistic intent in a cropped image, by following photographic composition rules such as the “rule of thirds” [44]. Examples of aesthetics-based approaches include [1, 41, 42]. In [1], the researchers proposed sensation-based photo cropping (SBPC), which is illustrated in Figure 2. SBPC trains a quality classifier, which gives a quality score to each candidate region, and the candidate region with the highest score is cropped. They trained the quality classifier on a large photo database. The database included photos from DPChallenge [7] and Photo.net [8]; the photos containing human faces were removed from the database. The remaining photos were associated with quality scores given by different people. The higher the quality score is, the higher the image quality is. The classifier is trained according to the quality scores. The input photo is trimmed into several candidate regions. The region to be cropped is determined by the quality score. The quality classifier could automatically rank all the candidate regions of a photo to ensure the region with the highest quality score is cropped.

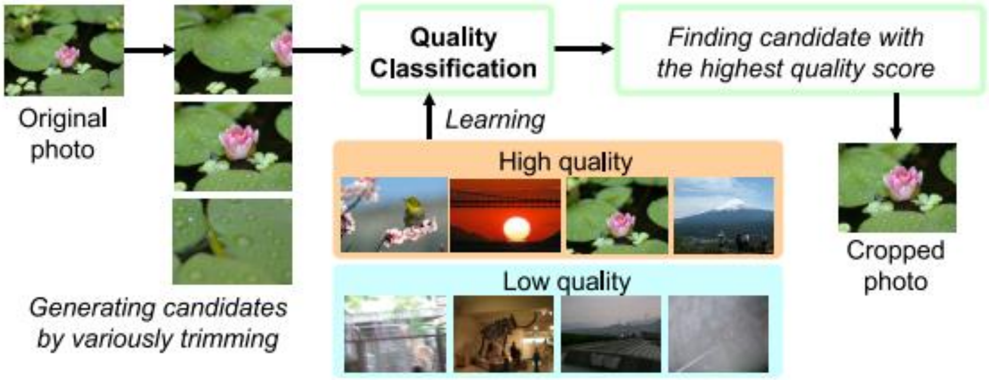

Original photo
Generating candidates by variously trimming
Quality Classification
Learning
High quality
Low quality
Finding candidate with the highest quality score
Cropped photo

Figure 2: The overview of SBPC algorithm [1]

In [41], a content preserving aesthetic image cropping method is proposed. The researchers trained a quality classifier to give a quality score to each crop candidate, and combine the quality scores with object boundaries simplicity and content preservation. Object boundaries simplicity is used to avoid cutting through the object. Visual saliency is used to preserve attention grabbing content.

In [42], composition rules are combined with region statistics to create a cropping method that takes into account aesthetic change between the original and cropped image. Crop-out and cut-through values are used to identify each crop candidates, and a quality classifier is trained to evaluate each candidate.

Attention-based methods attempt to preserve the most important content in a cropped image, usually by generating a bottom-up visual saliency map and choosing a crop from the region with the highest total saliency [46, 47]. In some cases, bottom-up saliency is replaced by the search for important high-level concepts such as faces [48] and human figures [49]. In [50], bottom-up saliency is supplemented by face and skin detection to create a cropping method that combines rudimentary bottom-up and top-down analysis.

In [2], an attention-based method, sparse coding of saliency maps (SCSM), is proposed for image cropping. SCSM first trains a classifier from the saliency map of training photos and then selects the candidate region with least error. The researchers first classify all training photos into 13 categories with a multi-class SVM classifier. The classifier is trained for 13 scene categories with more than 6000 photos. Then they use the Graph-Based Visual Saliency (GBVS) algorithm [19] to extract the saliency map of each training photo for each category. The saliency maps are used as the feature vectors. A sub-classifier for each category is trained based on these saliency maps and is called a “Dictionary” in [2]. At test time, they first classify the photo and calculate saliency maps. They then search the candidate crop regions that can be best decoded from the dictionary.

Reference [3] proposed another attention-based method, describable attribute for photo cropping (DAPC). DAPC picks an appropriate set of low-level features, and trains classifiers to predict the high-level attributes, then ranks the input images with the trained classifier. The researchers proposed a photo cropping method using an attribute classifier. To train the classifiers to predict the attributes from low-level features, a training dataset with ground-truth attributes is needed. The high-level attributes have never been provided by any image dataset. To get the training data, the researchers presented the photos and asked people to label the photos based on some attributes. People were asked if the image meets the low-level attributes, such as the “rule of thirds”, and people could answers “Yes”, “No” or “Not sure”. The positive images, which are consistently labeled as “yes”, are used to train the attribute classifier. They collected 1000 photos from DPChallenge [7] and Flickr [9] with quality scores given by different users. In Flickr [9], the quality score is called “interestingness.” The researchers also trained a quality classifier based on the high-level attributes. The quality score will be given for each cropping candidate with the trained quality classifier.

1.2. Purpose of the Study

The cropping method proposed in this thesis is conceptually similar to [50] in the sense that it combines bottom-up visual saliency and top-down semantic analysis. However, we take advantage of the recent progress on object classification and detection [51] and employ a deep convolutional neural network to detect the presence of 1000 different object categories in an image. We further classify the object categories according to importance and combine this semantic analysis with bottom-up saliency detection, as well as face and upper body detection. The result is a cropping system that is better able to understand the image content and produce a cropped image that better preserves important content.

1.3. Bottom-up Saliency

Visual attention has been investigated by scientists in different disciplines, such as cognitive psychology, neuroscience, and computer vision [10-13]. A bottom-up saliency-based is derived from low-level features and is independent of categories, or other properties of objects. Many bottom-up saliency models have been developed, such as feature integration theory (FIT) [14], guided search model [15], [16], Koch and Ullman [17] and Itti-Koch-Niebur [18]. Below, we review three popular saliency models: Itti-Koch-Niebur Saliency (IKN [18]), Graph-Based Visual Saliency (GBVS [19]) and Adaptive Whitening Saliency (AWS [20]).

IKN model [18] first decomposes the image into a set of feature maps based on colors, intensity, and orientations, and normalizes the feature maps. Then it uses across-scale processing to combine the feature maps to create three master “maps” corresponding to colors, intensity, and orientations. Finally, these three master “maps” are linearly combined to get the final saliency map. GBVS [19] first extracts feature vectors from the whole image and constructs activation maps based on feature vectors. Then, it normalizes these maps to highlight conspicuity and combines them into a single map. It defines Markov chains over various image maps, and computes the activation and saliency values from the equilibrium distribution over map locations.

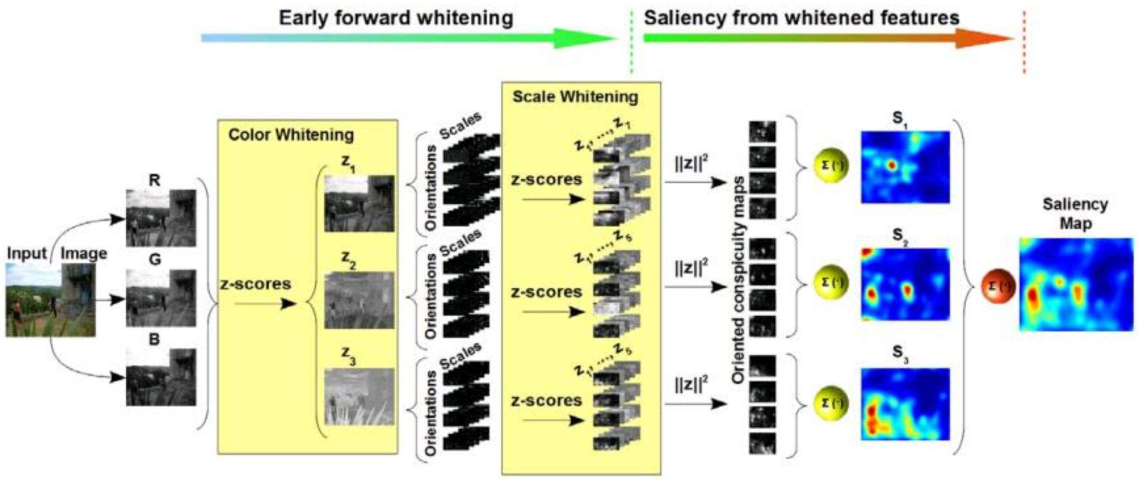


Figure 3: AWS saliency map computation process [20]

As Figure 3 shows, the AWS [20] process contains early forward whitening and saliency extraction from whitened features. In the early forward whitening stage, the researchers introduced a multi-scale multi-orientation decomposition to the input image, which contains the color whitening, chromatic scale orientation, and scale whitening. Then after the early whitening, conspicuity maps are computed using a simple squared vector norms computation and the final saliency map is the summation of these conspicuity maps.

These three saliency models have been widely used for computing bottom-up saliency maps. Each model generates a saliency map image with pixel values between 0-1. The higher the value is, the more salient the pixel is. If we display the saliency map as a grayscale image, the bright area is predicted to be more salient than the dark part. Figure 4 shows one sample image and the saliency maps computed by each model: IKN, GBVS and AWS.

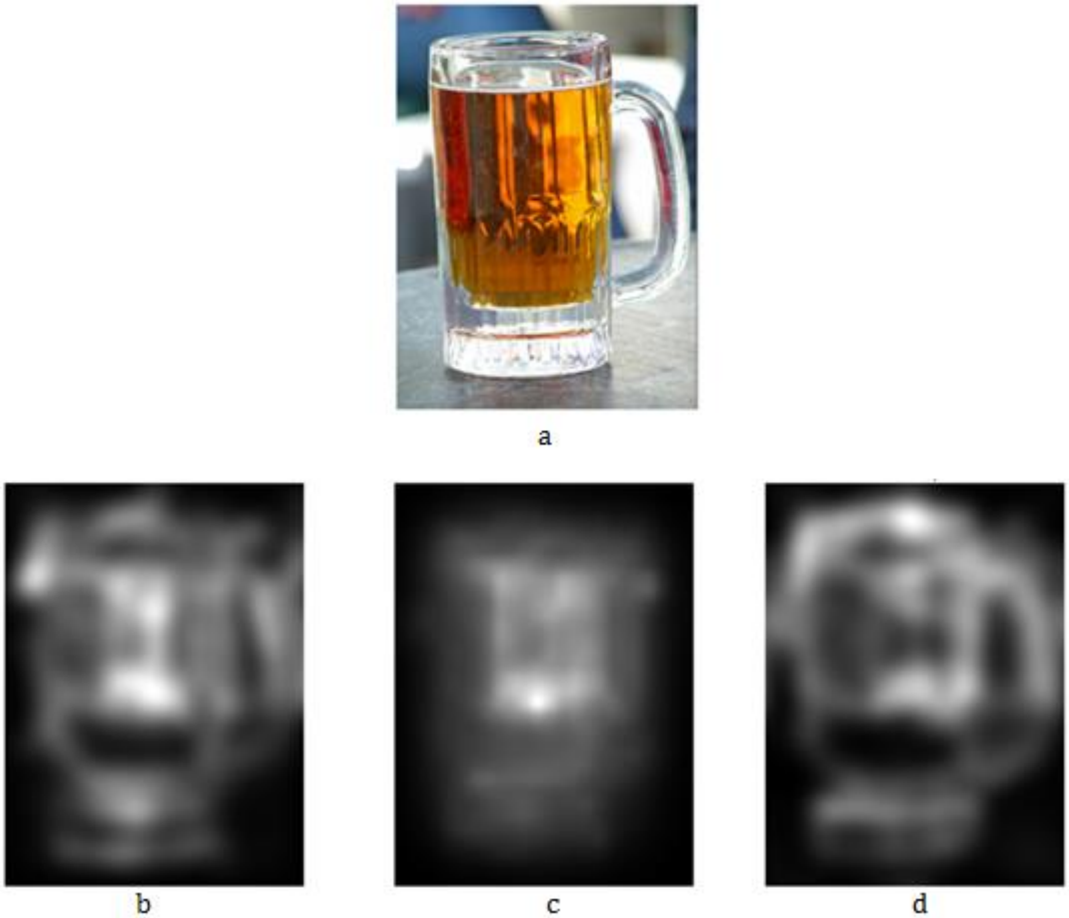


Figure 4: Saliency map computed by each saliency model: Original Image (a) vs. IKN saliency map (b) vs. GBVS saliency map (c) vs. AWS saliency map (d)

In order to find the best saliency model among these three to use in our application, we collect some test photos to evaluate the performances of these three models. We collected some photos from unsplash.com [21] (our dataset is described in more detail in Chapter 4), and then processed all the photos with these three models. With each saliency map we get, we will crop the most salient part of the original image. To find the most salient part, we create a search window, which has the same size as the final required image. The window will slide around the saliency map, and the summation of all saliency values in the search window will be calculated. The window that contains the largest summation (total saliency) is the most salient part of the image. Finally, we will cut this part from the original image as the final retargeted image result. Figure 5 shows one sample image and Figure 6 shows the result of cropping the most salient rectangle.

In Figure 6, the first row is the saliency maps computed by each saliency model. The second row is the cropped window with the largest value of total saliency, and the third row is the corresponding window from the original image, which is the cropped result.

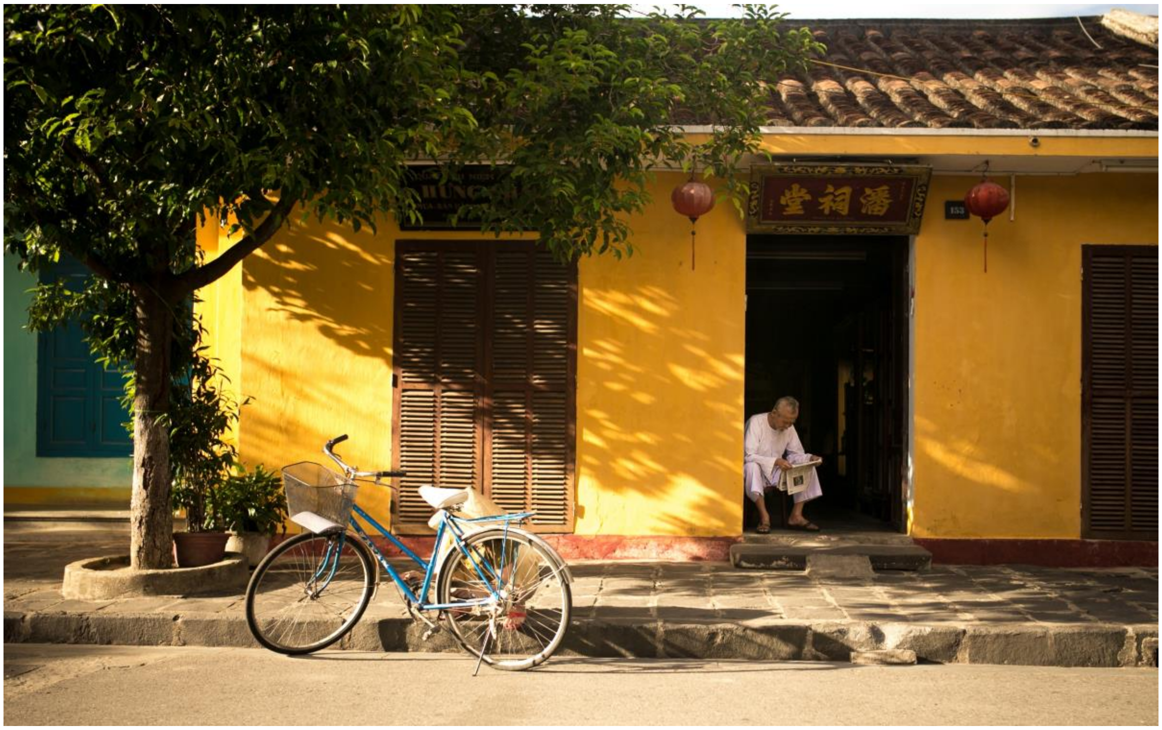


Figure 5: A sample image used for evaluation of three saliency models

After browsing through the results across all test images, we found the AWS produces the best crops, followed by IKN. GBVS has similar results in most cases, but in some cases it missed the important parts, especially near the boundary of the image, as GBVS concentrates its saliency map to the center of the image. As Figure 6 shows, the GBVS cropped image (g) misses the central part of the bicycle. The AWS cropped image (i) captures the bicycle. In addition to examples like this, AWS was the top-performing saliency model in a comprehensive study reported in [11]. Therefore, we choose AWS as the saliency model in our proposed algorithm, as well as the benchmark against which our model will be compared.

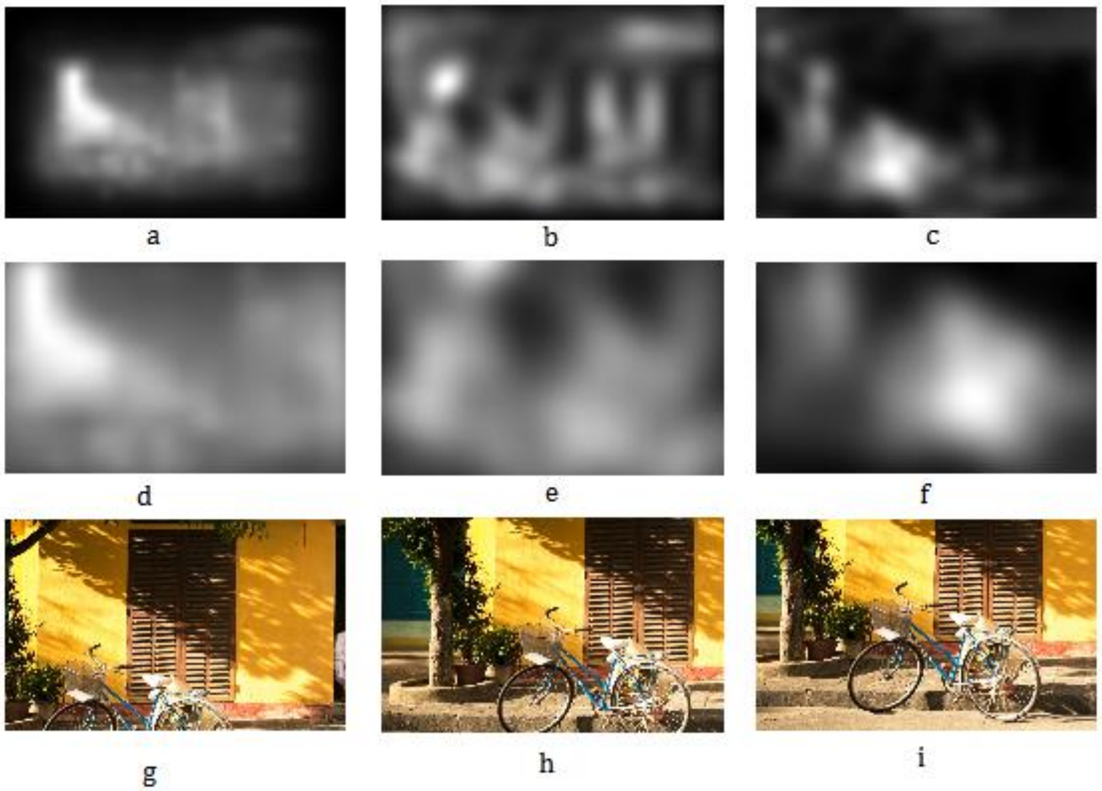


Figure 6: The process of cropping the image: GBVS saliency map (a) vs. IKN saliency map (b) vs. AWS saliency map (c); cropped GBVS saliency map (d) vs. cropped IKN saliency map (e) vs. cropped AWS saliency map (f); GBVS cropped image (g) vs. IKN cropped image (h) vs. AWS cropped image (i)

From the image in Figure 6, we also find that none of these three bottom-up saliency models indicate the old man as a salient object, which is the problem encountered with only using the bottom-up saliency model to perform cropping. That is why we need further semantic analysis to improve automatic cropping performance. The following section briefly discusses this issue.

1.4. Top-down Semantics

In order to incorporate semantic information into image cropping, we employ object detection and classification in our cropping system. The top-down semantic analysis in our system includes object detection and face/upper body detection. Object detection is the process of finding real-world objects such as trees, cars, and buildings in images or videos. The typical object detection algorithms include feature extraction and a machine learning algorithm to recognize an object. Face detection is a category under object detection, but considering the importance of human faces, we decided to separate face/upper body detection from general object detection.

The object detection and classification model we employed in our proposed cropping system is a Deep Convolutional Neural Network called GoogLeNet [22], the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 classification challenge. The challenge involved the task of classifying images into one of 1000 leaf-node categories in the ImageNet hierarchy. There are about 1.2 million images for training, 50,000 for validation and 100,000 images for testing. 1000 categories are labeled in these 1.2 million images. They include various animals, daily life objects, plants, and so on. A full list of categories can be found in [45]. The average image resolution is around 500×500 pixels. Each image is associated with one ground truth category, and performance is measured based on the highest scoring classifier predictions. 200 categories were chosen in the challenge, and GoogLeNet won 142/200 object categories, which was the most among all competing models, and also had the largest mean average precision. Most importantly, GoogLeNet had the lowest classification error [23] in all 1000 categories.

There are many other object detection and classification models, such as YOLO [58], R-CNN [59], Fast R-CNN [60], Faster R-CNN [61], and SSD [70]. In [58], the researchers indicated that YOLO could achieve 0.88 on top-5 accuracy on the ImageNet 2012 validation set, which is still slightly lower than the accuracy that GoogLeNet could achieve in ImageNet 2014, 0.89, shown in [66]. Many other new models are being developed, especially for object localization, but at the time of this research GoogLeNet is still considered a state-of-the-art solution for object classification. GoogLeNet does not give the location of the detected object, but it provides better classification accuracy than competing models, which is crucial for semantic analysis in our proposed system.

Since the GoogLeNet ImageNet model does not detect humans or faces, we use another model for face/upper body detection. Many face detectors have been developed over the years. As stated in [71], the face detection algorithm can be classified into three main families: boosting-based [24], convolutional neural network based (CNN-based) [59] model and Deformable Parts-based Models (DPMs) [72]. Boosting-based approaches contain two major procedures: feature extraction and a learning algorithm. Viola-Jones is one of the typical boosting-based models which uses the Haar-like feature and Adaboost Learning. CNN-based models are based on convolutional neural network which have been popular in the face detection field recently, and are currently considered state-of-the-art for this problem. DPMs model the deformation between facial parts, and usually contains two major procedures: facial part localization and face detection. A lot of extension works have been built based on these face detection techniques, such as emotion recognition, face tracking, and head pose estimation.

The detector we used is the well-known Viola-Jones [24] detector. It is one of the earliest and most popular object detection algorithms. While there are more advanced and accurate face detectors currently available, Viola-Jones is still used and one of the benchmarks and we felt it was sufficiently good for the purpose of demonstrating the principles behind our proposed approach. Our overall system architecture is very flexible, however, and it does not depend on any particular face detector. One could easily replace the Viola-Jones detector with any other face detector in our system without any other modification to the architecture or parameters.

The Viola-Jones face/upper body detector is implemented in MATLAB as `cascadeObjectDetector`. The `cascadeObjectDetector` function provides options for face detection, upper body detection, eye pair detection and so on. The most salient parts of an image are usually the faces and the human bodies, so we use both face detection and the upper body detection in the proposed algorithm.

1.5. Contributions

The contributions of this thesis are as follows:

1. A complete automatic image cropping method based on bottom-up visual saliency and top-down semantic analysis.
2. A comparison of the proposed algorithm against a prototypical attention-based cropping method on high-resolution images with a variety of content.
3. A platform for subjective image quality evaluation on mobile devices by providing an easy-to-use, natural interface using the mobile device's touch screen.

These contributions have also been reported in the following papers:

- J. Lin and I. V. Bajić, "A platform for subjective image quality evaluation on mobile devices," *Proc. IEEE CCECE'16*, pp. 1-4, May 2016. (Reference [38])
- J. Lin and I. V. Bajić, "Automatic Image Cropping based on Bottom-up Saliency and Top-down Semantics", presented at *IEEE PacRim' 17*, Victoria, BC, Aug. 2017. (Reference [52])

Chapter 2.

Proposed Automatic Cropping Technique

In this chapter, we present our proposed automatic image cropping technique. The four major parts, which are visual saliency estimation, multiresolution object detection and classification, multiresolution face/upper body detection and final map construction, are explained in detail, followed by some optimization solutions for practical implementation. The contents of this chapter have been presented in [52].

2.1. System Architecture

The architecture of our image cropping algorithm is shown in Figure 7. It consists of four major parts: the AWS saliency estimation, multiresolution object detection and classification, multiresolution face/upper body detection and final map construction (fusion). The last step is finding the maximum enclosing rectangle within the final map.

The AWS saliency estimation, object detection and human detection are processed in parallel. The input image $I(x,y)$ is subject to AWS saliency estimation to generate the AWS saliency map $S(x,y)$. Meanwhile, object detection and face/upper body detection are applied to the input image to find the object and human faces/upper bodies. Both the object detection and face/upper body detection are applied in a multiresolution manner to detect objects at multiple scales. Then, an object map $O(x,y)$ is constructed based on the multi-resolution object detection result, and the face/upper body map $H(x,y)$ is created based on the result of the face/upper body detection. Once we have all three maps, the AWS saliency map $S(x,y)$, object map $O(x,y)$, and face/upper body map $H(x,y)$, we fuse them together to create the final map $F(x,y)$. Finally, we find the maximum enclosing rectangle in $F(x,y)$ and crop the input image based on the result. Each part is described with more details in the following sections.

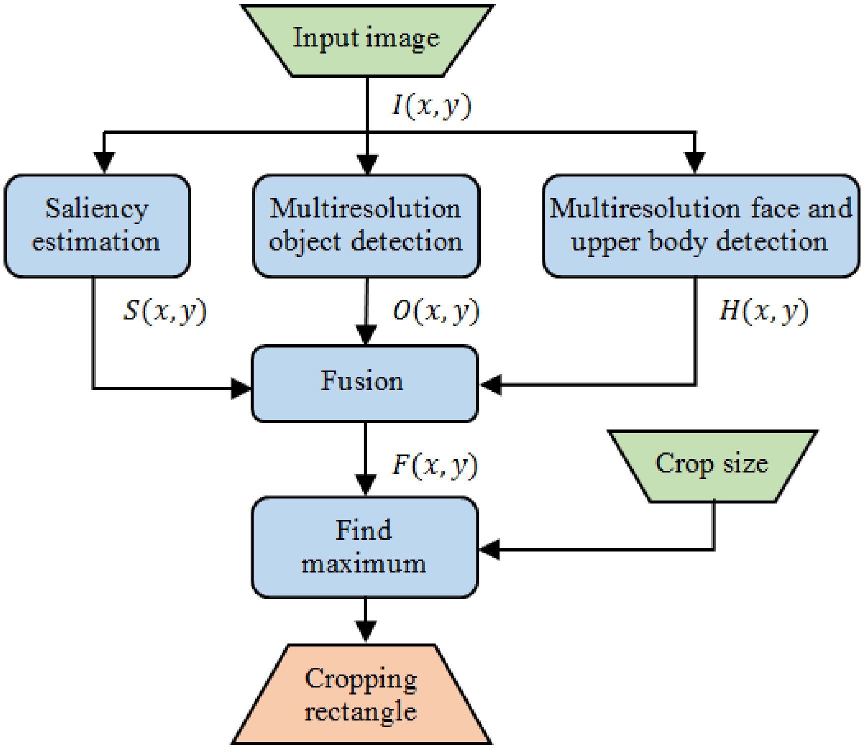


Figure 7: Overview of the proposing image cropping algorithm

2.2. Saliency Estimation

Great progress has been made on bottom-up visual saliency modeling in the last few decades [10]. As we mentioned earlier, the AWS saliency model [20] showed better performance compared to the GBVS and IKN models [19, 18]. In fact, AWS was the top performing model in a comprehensive comparison reported in [11]. Therefore, we choose AWS as the bottom-up saliency model in our system. The input image is processed with the AWS saliency estimation method to obtain the AWS saliency map $S(x,y)$.

An example of an image and its AWS saliency map is shown in Figure 8. As seen in the figure, the saliency map indicates the area near the bicycle as the most salient region of the image, due to the diversity of low-level features such as edges, colors and brightness in this part of the image. However, the higher-level semantic concept of the bicycle is not captured by the bottom-up saliency map, and neither is the human that sits in the doorway. Examples like this show us that bottom-up saliency alone is not sufficient to identify the most important part of the image, because it is missing out on the higher-level semantic concepts. This is why our image cropping system incorporates object detectors.

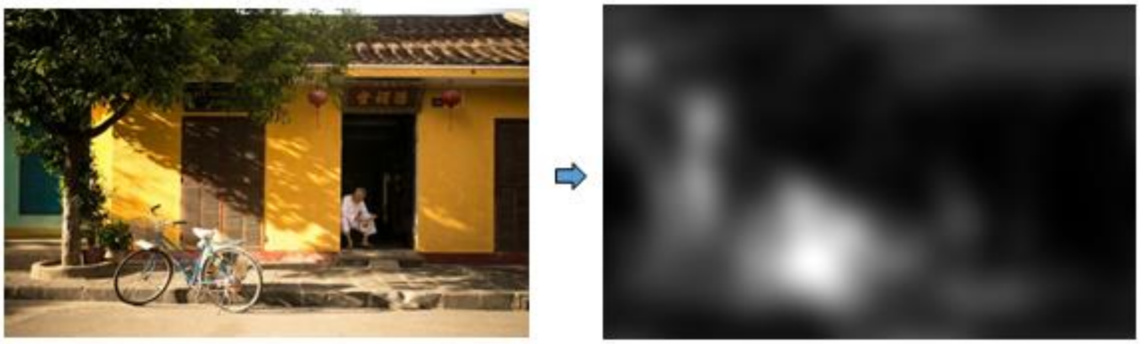


Figure 8: Input image $I(x, y)$ (left) and its AWS saliency map $S(x, y)$ (right)

2.3. Object Detection and Classification

We build our object detection around the GoogLeNet deep convolutional neural network (CNN) [22]. GoogLeNet was trained on 1.2 million images and 1000 object categories. However, straight-forward application of GoogLeNet and similar CNN-based object detectors to large images that are encountered in image cropping applications usually does not give good results. This is because, due to the complexity and memory requirements of CNN training, these detectors are trained on relatively small images, usually around 300×300 pixels in size. Specifically, GoogLeNet's input is 224×224 [22]. If one shrinks a large image to this size, the objects of interest often become too small for the detector to recognize.

For this reason, we develop a multiresolution approach for object detection in our system. The whole multiresolution approach for object detection in our system is shown in Table 1.

Function `object_importance` (whose details are described below) is applied to the input image I and results in an object importance map O_1 of the same size as the input image. If the width and height of the image are larger than 2000 pixels, the image is rescaled by ½ in each dimension (and referred to as I_1), `object_importance` function is applied again, and the resulting map (O_2) is upsampled to the size of the original image. The process is repeated until the dimensions of the downscaled image are less than 2000 pixels. Index k in the algorithm denotes the resolution level, with $k = 0$ being the original resolution. At the end, max-pooling is executed over all the object importance maps, and each pixel in the final map $O(x, y)$ obtains the maximum value found at location (x, y) across all object importance maps at various resolution levels. In the algorithm, the dimension threshold of 2000 pixels is set empirically. It gave reasonable results on our data, but it can easily be changed for other applications or datasets.

Table 1: Multiresolution object detection algorithm

```
$O_0 =$ object_importance($I$)
Let $k = 1$, $L_{max} = 0$, and $I_0 = I$
while width($I_k$) > 2000 or height($I_k$) > 2000 :
    $k = k + 1, L_{max} = k$
    Scale width and height of $I_{k-1}$ by ½, call the result $I_k$
    $O_k =$ object_importance ($I_k$)
    Upsample $O_k$ to the size of $I_0$ using bicubic interpolation
end while
$O(x, y) = \max_{k=0..L_{\max}} O_k(x, y)$
```

The function object_importance in Table 1 operates as follows. Image I_l at level l is subdivided into tiles. In our implementation, each tile is 500 × 500 pixels. The tiles on the right and bottom boundary of the image overlap their neighboring tiles in order to fit fully into the image. Each tile is rescaled to 224 × 224 pixels and input to the GoogLeNet object detector, which outputs a vector of 1000 real values indicating its confidence about the presence of 1000 different object categories in the tile. The confidence value is from 0 to 1, which indicates how confident the model is about the presence of a certain object category: the larger the value, the more confident the model is. Even though GoogLeNet sometimes suggests more than one object in the tile, we only take the object category with the largest confidence, denoted c. However, the detector's confidence c by itself is not a complete indication of the importance of the tile because different object categories may have different importance to the user. To account for that, we assign the object importance weight w as shown in Figure 9 and Table 2.

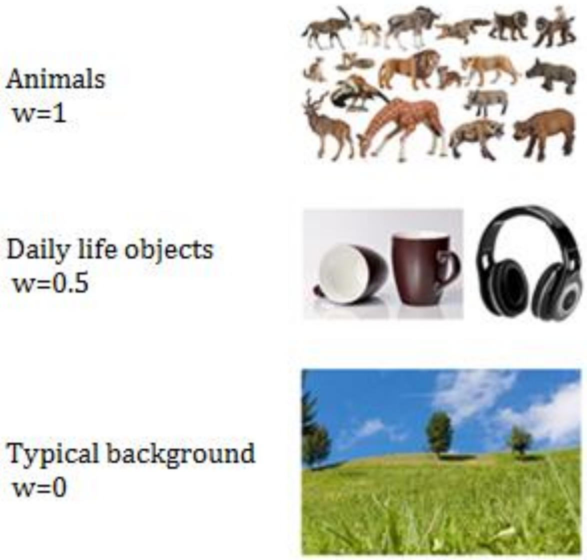


Figure 9: All 1000 categories are classified into three classes

Table 2: Object importance weights

Animals $\boldsymbol{w = 1}$	**Daily life objects** $\boldsymbol{w = 0.5}$	**Typical backgrounds** $\boldsymbol{w = 0}$
English setter	Umbrella	Picket fence
Egyptian cat	Soccer ball	Sliding door
Gazelle	Laptop	Dam, dike, dyke
…	…	…

We divided the 1000 objects categories that GoogLeNet detector can recognize into three groups: animals, daily life objects, and typical backgrounds. Table 2 lists some of the examples in each category, along with the weights w. While many different weight assignments are possible and, depending on the application, can also be personalized for each user, we settled for a simple and sensible assignment that we believe is sufficient to demonstrate the effectiveness of the cropping system: animals get the highest weight $w = 1$, daily life objects $w = 0.5$, and typical backgrounds $w = 0$.

$$w = \begin{cases} 1, & \text{if object belongs to animals} \\ 0.5, & \text{if object belongs to daily life objects} \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

Once the weight is set according to the object category, the pixels in the tile are assigned importance values using the scaled Gaussian function

$$O_l(x, y) = w \cdot c \cdot exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right) \tag{2.2}$$

where (x_0, y_0) is the center of the tile, and $\sigma_x = \sigma_y = 200$. In Equation (2.2), w represents object importance, which can be obtained from Equation (2.1), c represents detectors confidence about the presence of the object, and the exponential term represents uncertainty about the actual location of the object in the tile, since GoogLeNet does not provide the coordinates of the detected object. Figure 10 shows the cross-section and Figure 11 shows a surface plot of an example of an object map for two neighboring tiles with different detected objects.

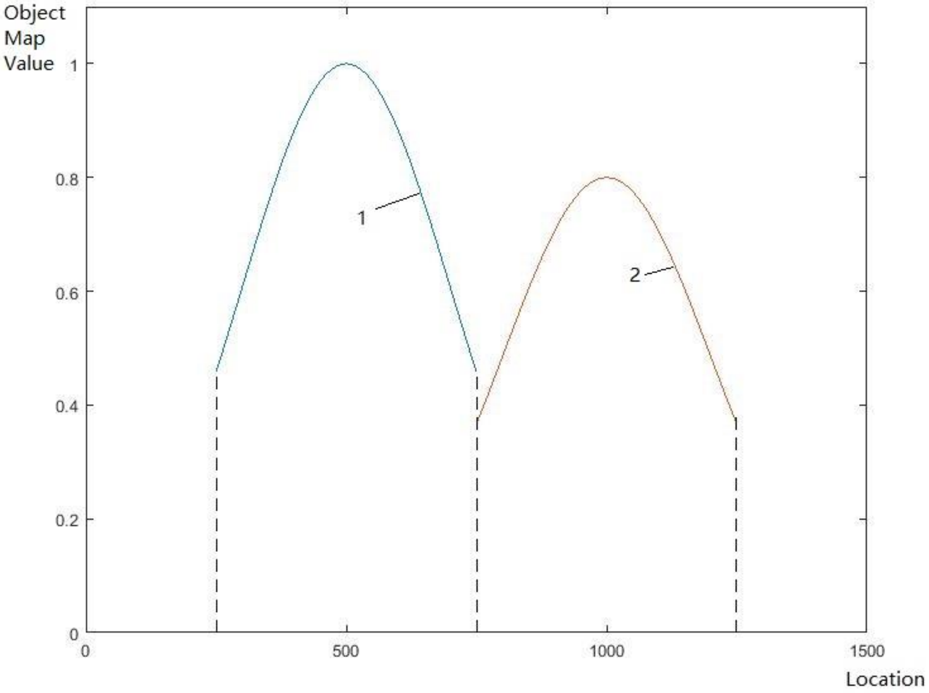


Figure 10: 2D cross-section of object map construction process for two tiles with different objects

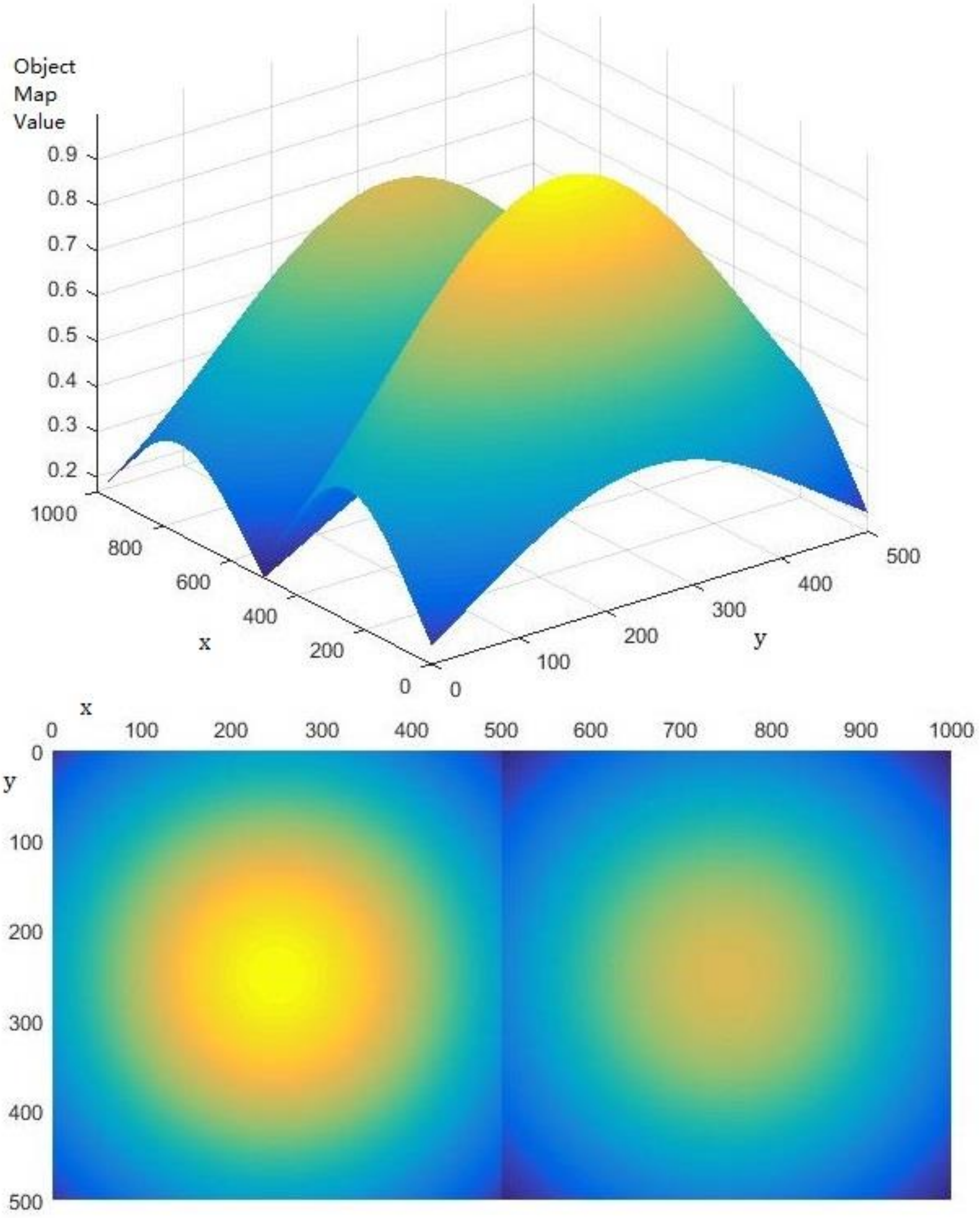


Figure 11: **3D Object map construction process for two tiles with different objects**

We use the center of each tile as the center of the Gaussian function, the weighted confidence as the peak value of the Gaussian function, and 200 as the standard deviation, which controls the width of the Gaussian function. Since the images we are targeting are pretty large, the object may be pretty large too. Suppose that we have an important object which is separated into two adjacent tiles. In order to have a smooth transition on the boundary between these two adjacent tiles, we need the Gaussian function to decay to half of the peak value at the boundary, which is 250 pixels away from the center, so that at the boundary the values of the two neighboring Gaussians would be equal for a Gaussian function. In the Gaussian function, the half maximum value will occur at $1.177\,\sigma$, where σ is the standard deviation. Therefore, we choose 200 as the standard deviation so that we have half maximum value at around 250. Figure 12 shows the cross-section of the object map. In Figure 12, the curves 1 and 2 are the object maps of two neighboring tiles detected as the same object, so they add up to curve 3. The value between two peaks is between 0.9 and 1.1, which is considered to be a smooth transition. Figure 13 shows a sample of a full object map construction process.

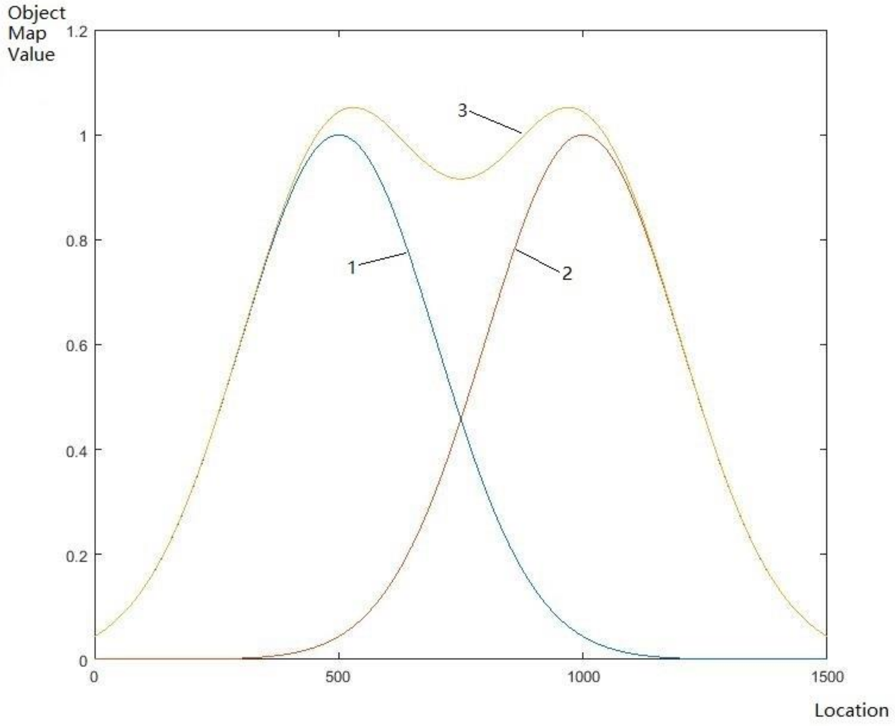


Figure 12: **2-D cross-section of object map construction process for an object separated into two tiles**

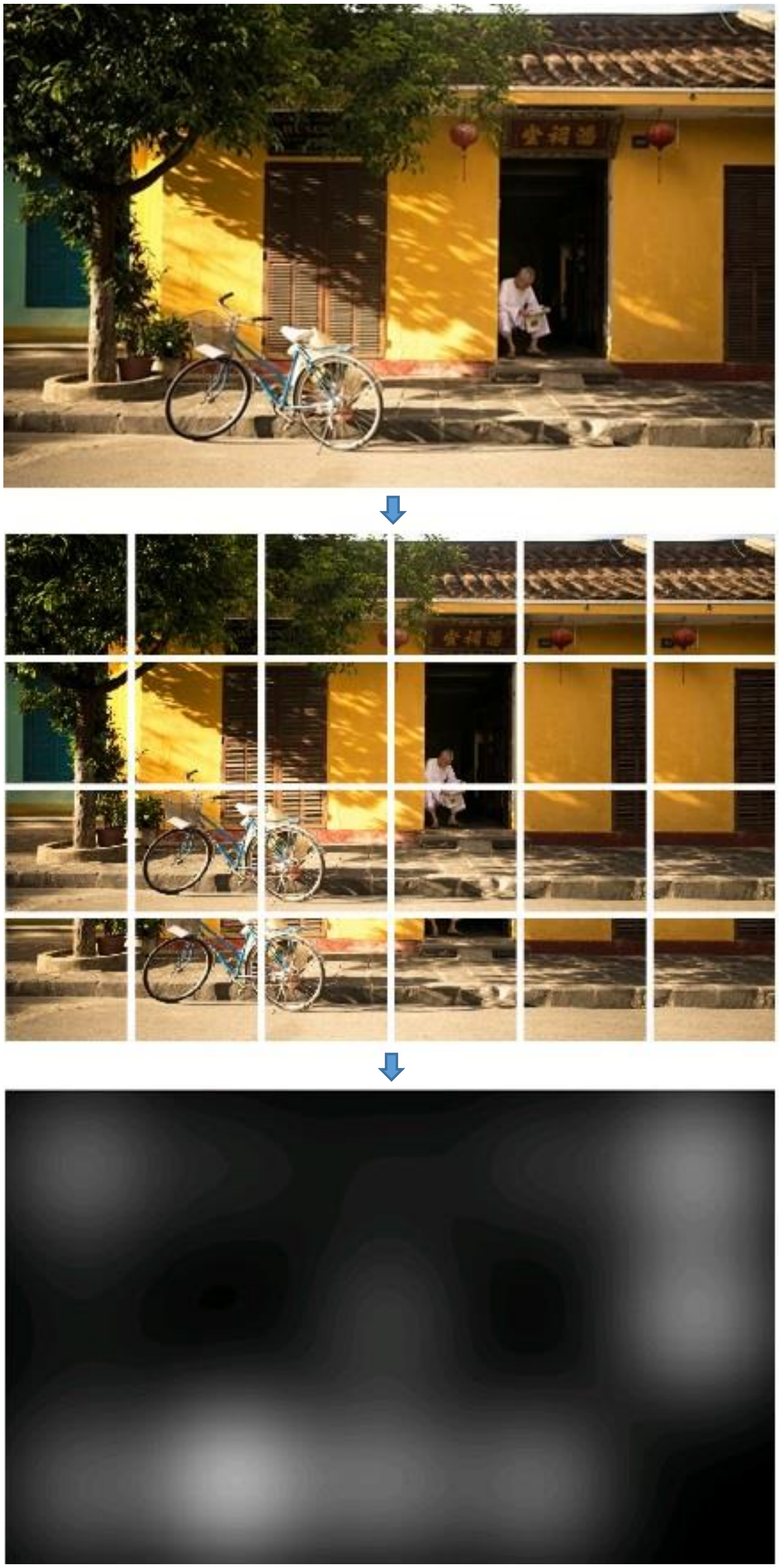


Figure 13: The object saliency map construction process

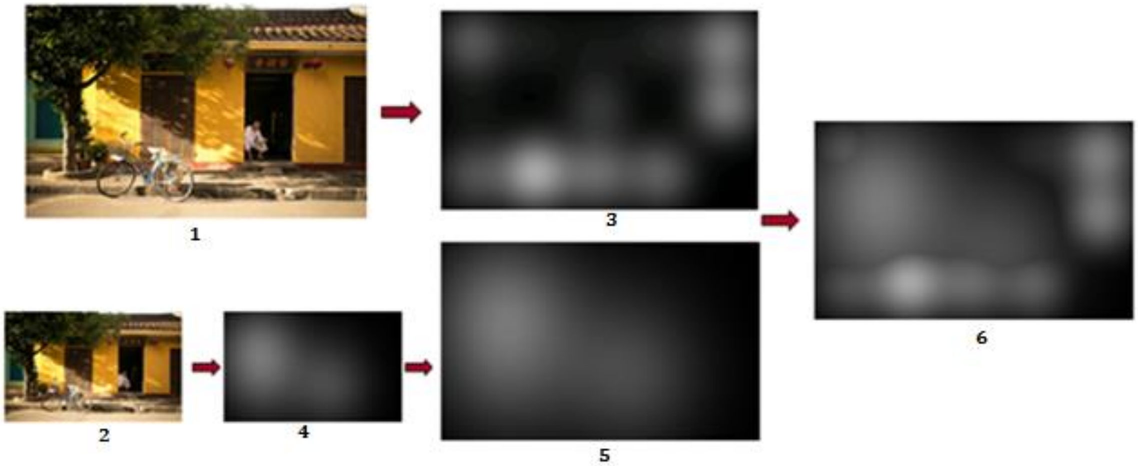


Figure 14: The multi-resolution object detection process for an image with resolution 2519×1581

An example of multi-resolution object detection is shown in Figure 14. The size of the input image ‘1’ is 2519×1581. We first compute the first level object map ‘3’ based on the input image ‘1’. And then, we scale down the input image to get the scaled image ‘2’. We compute the second level object map ‘4’ based on the scaled image ‘2’. After that, we combine the first level object map with the second level object map to get the final object map. For the combination, the second object map ‘4’ is scaled up to be of the same size as the first level object map, and it is shown as ‘5’. Finally, we combine these two maps, ‘3’ and ‘5’, by max pooling to get the final object map, ‘6’.

2.4. Face/Upper Body Detection

GoogLeNet does not detect humans or human body parts, so we use a separate detector for this purpose. Specifically, our face and upper body detection is built upon the well-known Viola-Jones detector [24]. It is implemented in MATLAB as function `cascadeObjectDetector`. The `cascadeObjectDetector` function provides options for face detection, upper body detection, eye pair detection and so on. Other than the classification model, there are also a few other parameters that could be customized in the `cascadeObjectDetector` function. We set all other parameters as default, except the `MergeThreshold`. The `MergeThreshold` is the threshold that defines the criteria to get the final result in an area where multiple detections are found around an object. Groups of detections that meet the threshold are merged to produce one bounding box around the target object. Increasing this threshold will help suppress false detections but runs the risk of missing more promising faces/upper bodies. We have experimented with various values and finally decide to set the `MergeThreshold` as 20.

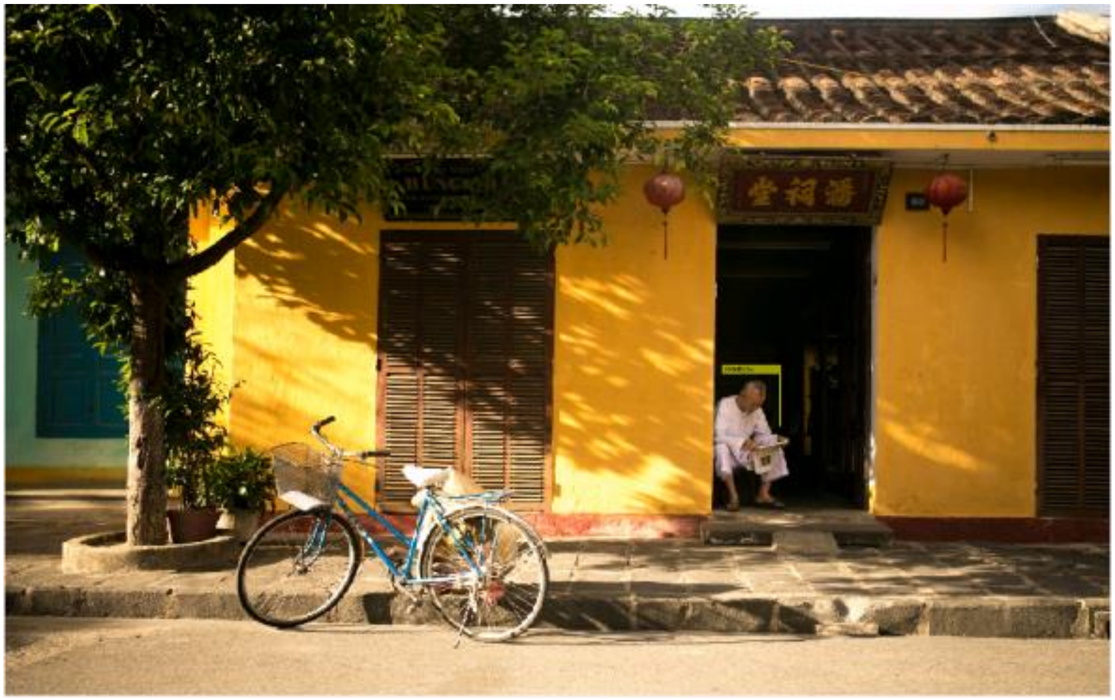


Figure 15: Human face and upper body detection result

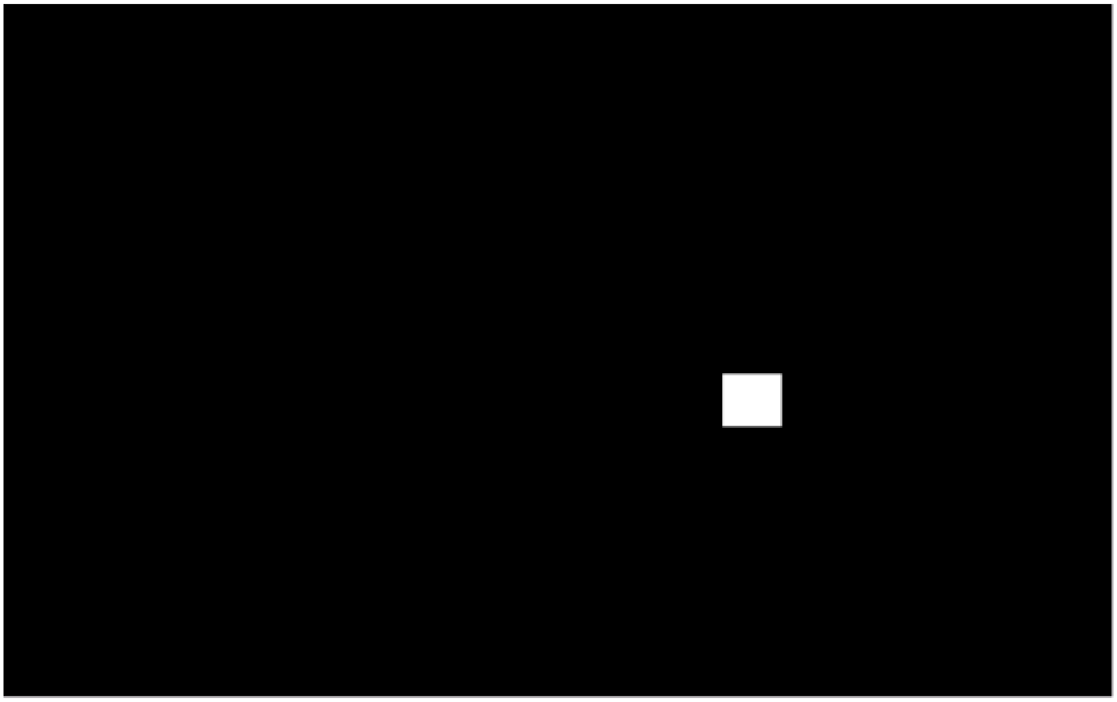


Figure 16: Human face and upper body detection saliency map $H(x)$

As Figure 15 shows, the man’s face in the image is not clear. In the experiment, the face detector misses the man’s face, but the upper body detector catches the man. Using both face and upper body detectors helps find humans in the input image.

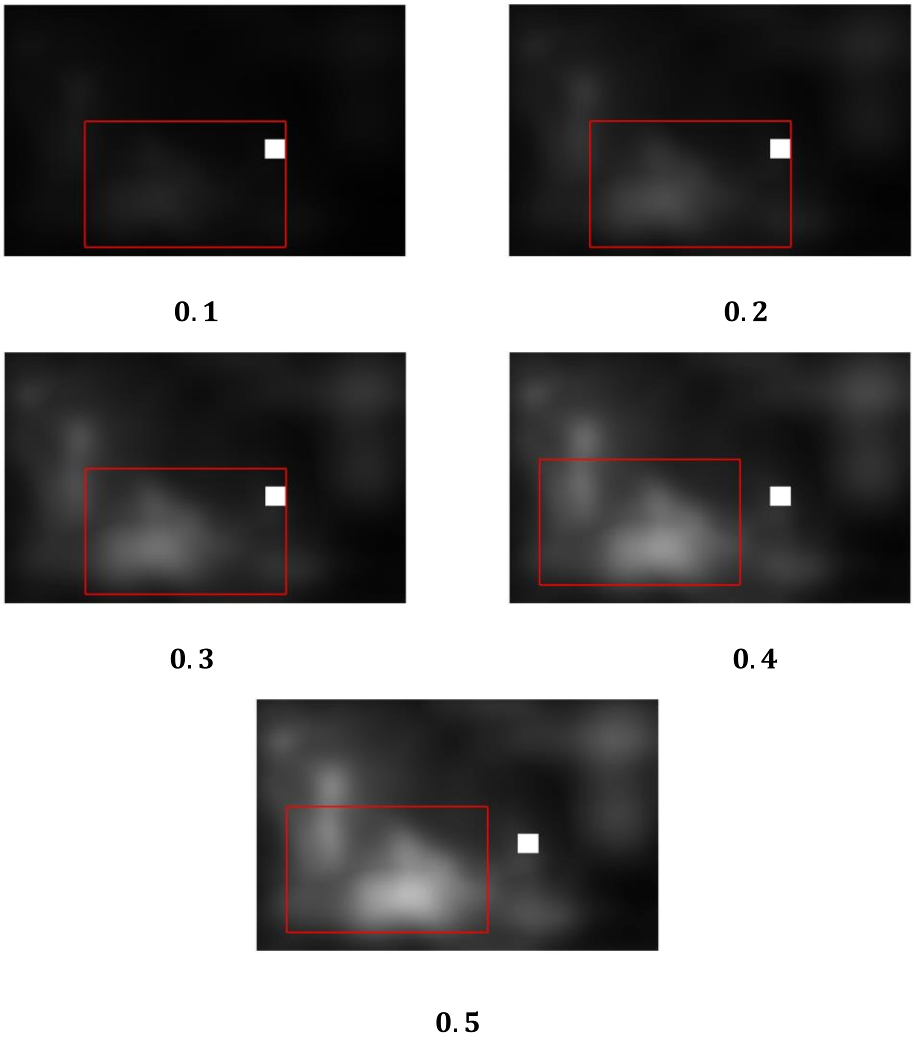
The multiresolution approach to face and upper body detection operates similarly to Algorithm 1. The main difference is that the `object_importance` function is replaced by the `face_detect` function, which runs the Viola-Jones detector on the input image I_l and returns a map H_l in which pixels in regions where faces/upper bodies are detected are set to 1 and others are set to 0. At the end (line 9 of the algorithm), maps H_l are max-pooled to create the final $H(x, y)$.

2.5. Final Map Construction

Once the three maps $S(x, y)$, $O(x, y)$, and $H(x, y)$ are created, they are fused as follows:

$$F(x, y) = \max(H(x, y), 0.3 \cdot S(x, y) + 0.3 \cdot O(x, y)) \tag{2.3}$$

The reasoning behind Equation (2.3) is as follows. If a pixel (x, y) belongs to a human face or upper body ($H(x, y) = 1$), then the final importance map will have the highest value at that position ($F(x, y) = 1$), because the second argument of the $\max$ function is upper bounded by 0.6. If there is no human at (x, y) (i.e., $H(x, y) = 0$), then its importance is determined by bottom-up saliency $S(x, y)$ and the possible presence of objects $O(x, y)$ each contributing an equal amount to the final importance. We experimented with various weights for these two terms. If the object weight is set to be very high, a large object with high confidence will be carrying a higher value, and it will potentially result in the final crop missing the human. But if the object is large enough, and sufficiently important (according to Table 2) while the human figure is very small, we want the final crop to give up the relatively small human in favor of the important object. Considering the trade-off between the large object and small human face/upper body, we experimented with different values, as shown in Figure 17. In this figure, the final crops labeled in red with value 0.4 and 0.5 missed the human face/upper body, and 0.1 or 0.2 may make the whole system only focus on the human face/upper body. Therefore, we finally chose 0.3 as the value for the object weight and used the same value for the saliency weight.



0.1

0.2

0.3

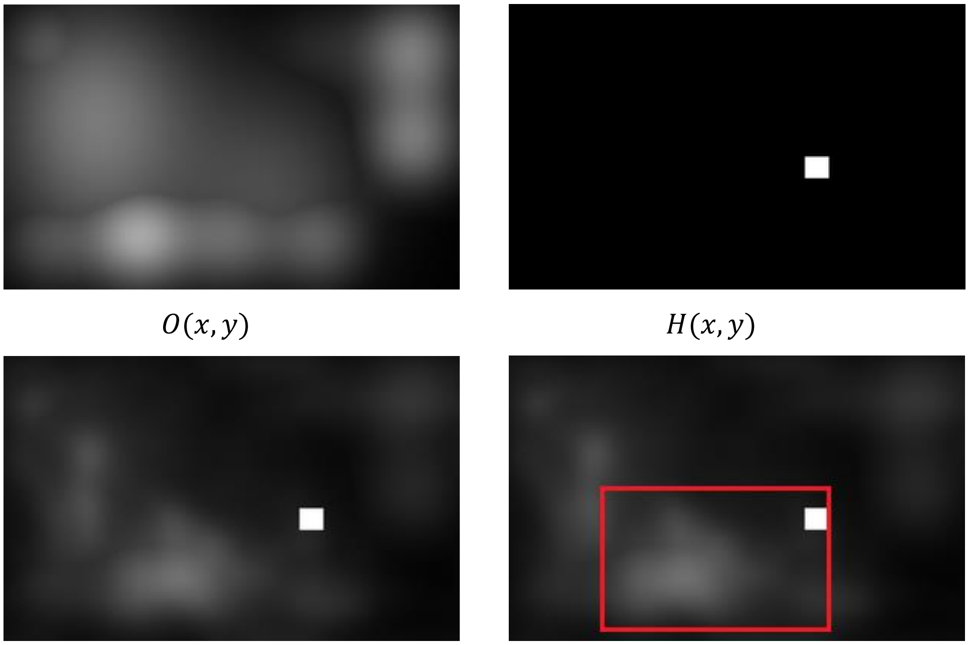
0.4

0.5

Figure 17: Final rectangle for choosing different object weight values in the fusion function

2.6. Image Cropping

Once the final importance map $F(x, y)$ is constructed, we find the rectangle of the desired size that includes the maximum total importance, i.e., the maximum sum of $F(x, y)$ within the rectangle. We create a search window with the same size as the final required image. The window will slide around the final map, and the summation of all values of $F(x, y)$ in the search window will be calculated. The window that contains the largest summation is the most important part of the image. Finally, we crop this part from the original image as the final cropped image. Figure 18 shows the various importance maps in the cropping system, along with the final selected rectangle.



$O(x, y)$ $H(x, y)$

$F(x, y)$ $F(x, y)$ with selected rectangle

Figure 18: Top: $O(x, y)$ and $H(x, y)$. Bottom: $F(x, y)$ and the maximum enclosing rectangle.

The above search procedure is inefficient and takes a long time to find the final rectangle. To speed it up, we use the Summed area table algorithm [25], also known as integral image. This is an algorithm for quickly and efficiently generating the summation of values in a rectangular subset of a matrix.

The value in the summed area table is the summation of all the importance values above and to the left. This is shown in Equation (2.4) where $F(x', y')$ is the value at the point (x', y') in the final importance map, and $T(x, y)$ is the value at the point (x, y) in the summed area table. Therefore, $T(x, y)$ is just the sum of all the importance values above and to the left of (x, y) in the importance map F:

$$T(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} F(x', y') \tag{2.4}$$

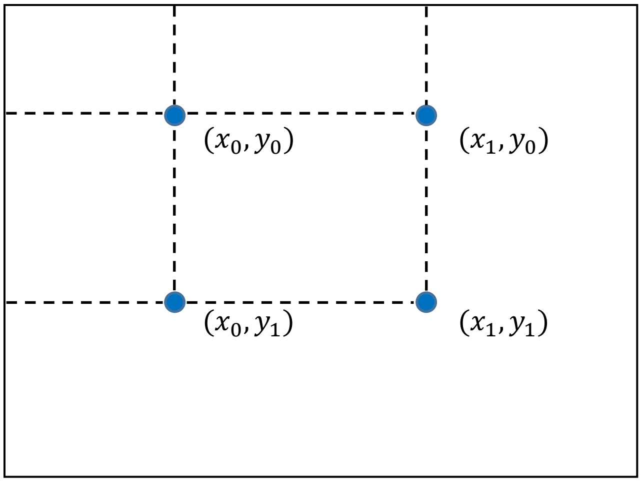


Figure 19: A example of computing a sum in the Summed Area Table algorithm (Adapted from [53])

Once the summed area table is computed, we can easily obtain the summation of values in any rectangular subset of the importance map. This takes a constant computing time that is independent of the size of the importance map and the size of the rectangle.

Figure 19 shows four points: (x_0, y_0), (x_1, y_0), (x_0, y_1), and (x_1, y_1). The summation over the rectangle in F enclosed by these points is

$$\sum_{\substack{x_0 < x \leq x_1 \\ y_0 < y \leq y_1}} F(x, y) = T(x_1, y_1) + T(x_0, y_0) - T(x_0, y_1) - T(x_1, y_0), \tag{2.5}$$

where $T(x_i, y_j)$ are the values in the summed area table at the corresponding points. The summed area table algorithm saves a lot of computing time. We found that with this algorithm, our system became more than ten times faster than before, when we used a straightforward search for the maximum enclosing rectangle.

2.7. Summary

In this chapter, we introduced the architecture of our automatic image cropping system. The system incorporates AWS saliency estimation, multiresolution object detection and classification, multiresolution face/upper body detection and final map construction. We use summed area table algorithm to speed up the search for the maximum enclosing rectangle in the final map.

In order to evaluate the performance of our automatic image cropping system, we designed a subjective image quality evaluation mobile application, so that we can easily perform a subjective quality test. The subjective image quality evaluation mobile application is introduced in Chapter 3.

Chapter 3.

Subjective Test Tool

Image quality is an important consideration for content creators, distributors, as well as for the consumer electronics industry that develops devices used for image display. While many “objective” image quality metrics have been developed in the last number of years [54, 55], subjective evaluation is still considered to be the ultimate test of image quality. This is especially evident when one considers the variety of subjective impressions that the same image can make when displayed on different devices.

In the engineering research community, the most popular set of protocols for subjective image quality evaluation comes from the ITU Recommendation BT.500 [29]. This chapter describes detailed experimental setup and methodology that can be applied to both still image and video quality assessment. In many test protocols described in [29], the subjects quantify visual quality in terms of numerical values.

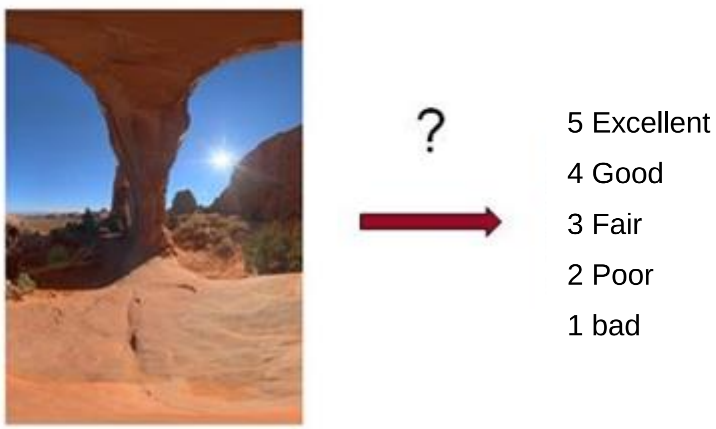


Figure 20: An example for a popular 5-point categorical quality scale

For example, as Figure 20 shows, a popular 5-point categorical quality scale allows the subject to describe the visual quality of the image or video in terms of the following five values, whose explanations are given in brackets: 5 (excellent), 4 (good), 3 (fair), 2 (poor), 1 (bad). Such an approach to quality assessment is known to create cognitive overhead, because subjects often spend considerable time deciding how to map their subjective impressions into the given categories. For example, an image whose quality is quite satisfactory may cause the subject to overthink – is this image “excellent” or merely “good?” Should I give it a 5 or 4?

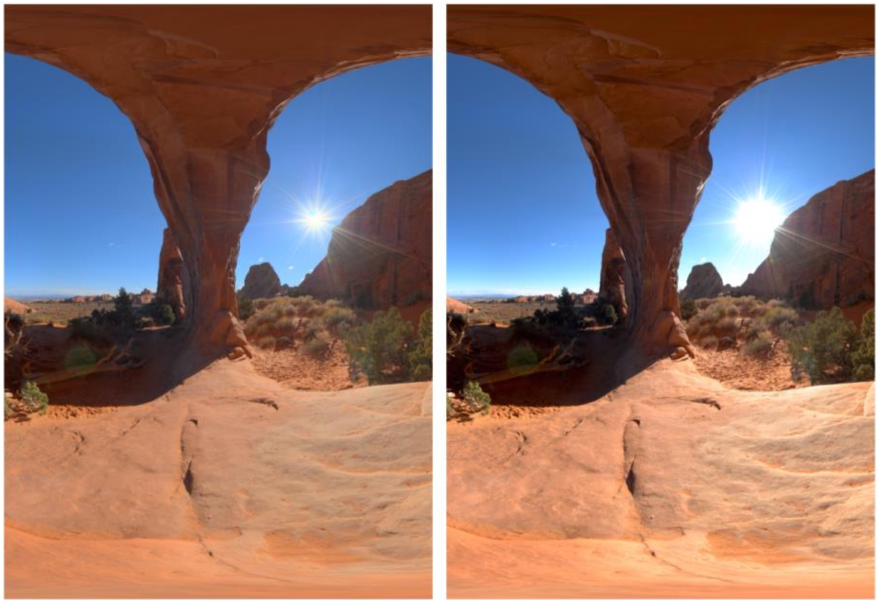


Figure 21: An example for a 2AFC approach

An alternate methodology that avoids the issues with categorical scales is the so-called two-alternative forced choice (2AFC) approach [30]. This methodology has long been used in the psychophysics community to measure detection thresholds for various psychophysical attributes. More recently, it has also been employed in image [31] and video [32] quality assessment. In this approach, test subjects directly compare qualities of two images or videos without having to map them to numerical values, which allows them to fully focus on quality assessment. For example, in Figure 21, two images are shown side by side. In 2AFC, the subject would simply be asked which of the two images looks better. 2AFC is the method of choice in the subjective image quality evaluation platform described here. We implemented this test platform as a mobile application on iOS. Coupled with an easy-to-use, natural user interface afforded by the mobile device’s touch screen, our test platform minimizes subjects’ cognitive load and permits them to devote full attention to the image quality assessment task. The content of this chapter has been presented in [38].

3.1.

3.2. Subjective Test Methodology

In this section, we first describe the methodology of Two-alternative forced choice (2AFC) for subjective image quality evaluation and the associated statistical test. 2AFC is an experimental methodology that has been used for a long time in psychophysics research [30]. It's most common use is related to measuring detection thresholds for various psychophysical attributes, whereby subjects would be presented with two slightly different auditory or visual stimuli, and would be asked whether they are able to detect the difference.

The same methodology can be used for subjective image/video quality evaluation [31, 32]. Suppose we have two different versions of the same image. For example, one version could be compressed, and the other uncompressed. The images would be presented to the subjects, who would be asked to select which of the two images looks better. They would be instructed to choose one of the images (randomly, if needed) even if they are unsure of which image they think looks better – hence the wording "forced choice" in 2AFC. If the two images look indistinguishable, which would force the subjects to choose randomly between the two, we expect about half the subjects to choose one of the images and the other half to choose the other image. This even distribution of votes is usually referred to as the chance level and is associated with the null hypothesis. Here, we use Pearson's chi-squared (χ^2) test to evaluate the test result. The chi-squared test is used to determine whether there is a significant difference between the observed result and the expected result. The expected result is formed under the null-hypothesis. The test provides the p-value, which represents the probability that the observed data can be generated under the null-hypothesis. If the p-value is too small, the observed data is unlikely to be generated under the null-hypothesis, so the hypothesis has to be rejected.

Suppose that after N subjects have voted, the first image has obtained n_1 votes and the second image has obtained n_2 votes, with $n_1 + n_2 = N$. If the images have the same subjective quality (which is the null-hypothesis in our case), we will expect each image to obtain $E_1 = E_2 = N/2$ votes. The Pearson's chi-squared (χ^2) test statistic is computed as [33]

$$\chi^2 = \sum_{i=1}^{2} \frac{(n_i - E_i)^2}{E_i}, \tag{3.1}$$

using which a p-value can be found from χ^2 distribution tables or graphs [34]. Figure 22 shows the relationship between the p-value and χ^2 for a chi-squared distribution with one degree of freedom, as is the case here.

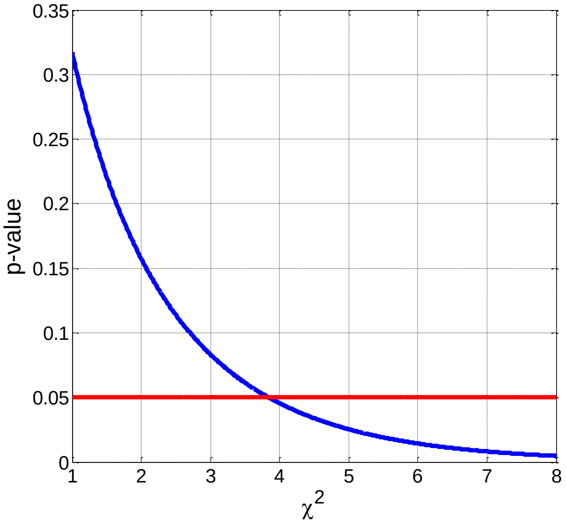


Figure 22: **p-value vs. χ^2 (blue) and the $p = 0.05$ level (red)**

In experimental sciences, the null hypothesis is usually rejected when $p < 0.05$. As seen in Figure 22, for this to happen, χ^2 needs to be sufficiently large, i.e., the observed votes n_i need to deviate sufficiently from their expected values E_i under the null hypothesis. If this happens to be the case, then the null hypothesis can be rejected, and we may conclude that the image that has received more votes has better subjective quality. Examples of chi-squared testing of subject responses are given in the last section of this chapter.

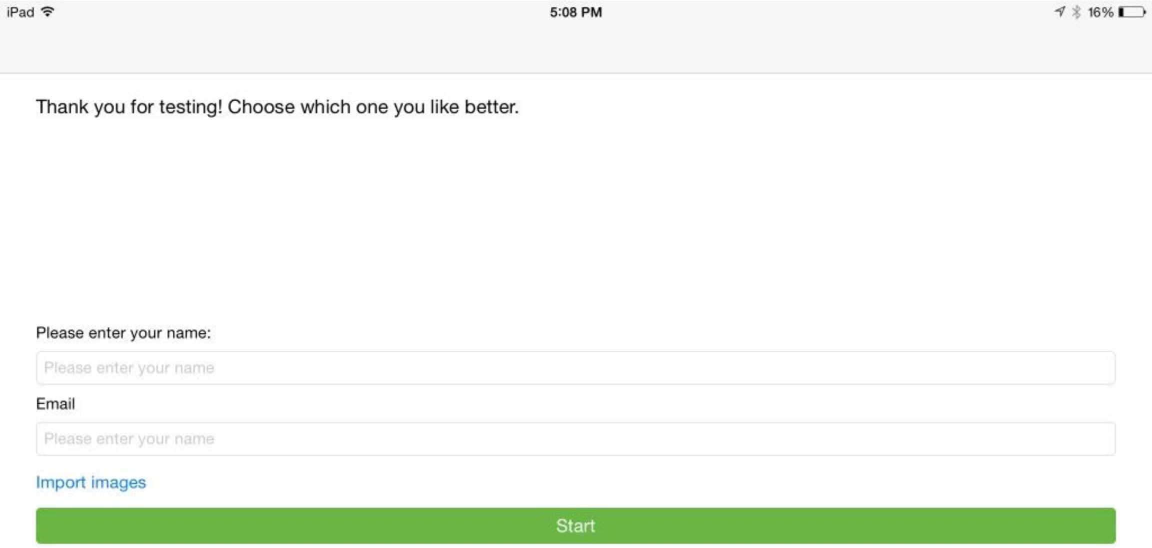
3.3. iOS Mobile App for Subjective Evaluation

The first step for the iOS mobile application development is to set up the development environment. The prerequisite for iOS development is a Mac workstation or Apple laptop, which runs Apple's operating system. Second, an iOS device is needed for debugging and testing purposes. We choose the iPad for this purpose, because the iPad has a larger screen than the iPhone, and it is better for the image quality test. The iOS

Simulator can also be used for the debugging purposes, but the test on the real device is still needed before the final deployment.

When the hardware is ready, the Xcode needs to be installed for development. To run an application on a real device, a registered Apple developer account is needed and iOS Developer Program needs to be subscribed to. Also, this gives us access to all beta software from Apple, related to iOS, which is pretty important from a developer's perspective.

Our app for subjective image quality evaluation is developed in Xcode and built on the iOS platform. When the app is started, brief instructions are shown on how to use it. The subject is asked to enter his/her name or other suitable ID, and an e-mail address where the test results should be sent – this e-mail address is provided to the subject by the experimenter, as Figure 23 shows.



Thank you for testing! Choose which one you like better.

Figure 23: The starting page of the subjective test application: The tester is asked to enter his/her name or other suitable ID, and an e-mail address where the test results should be sent – this e-mail address is provided to the tester by the experimenter.

The app searches the image gallery on the mobile device for images with filenames in a particular format. Specifically, the app looks for images whose filenames are 'A<index>' and 'B<index>'. These images are placed into the image gallery by the experimenter prior to the test. 'A' and 'B' indicate the two classes of images being compared, for example compressed and uncompressed, or images produced by two different image processing algorithms. Meanwhile, <index> is the index of the image pair that will be displayed to the subject: pair A1-B1 is displayed first, followed by A2-B2, and so on. Once the test starts, image pairs are displayed on the screen in sequence according to the index.

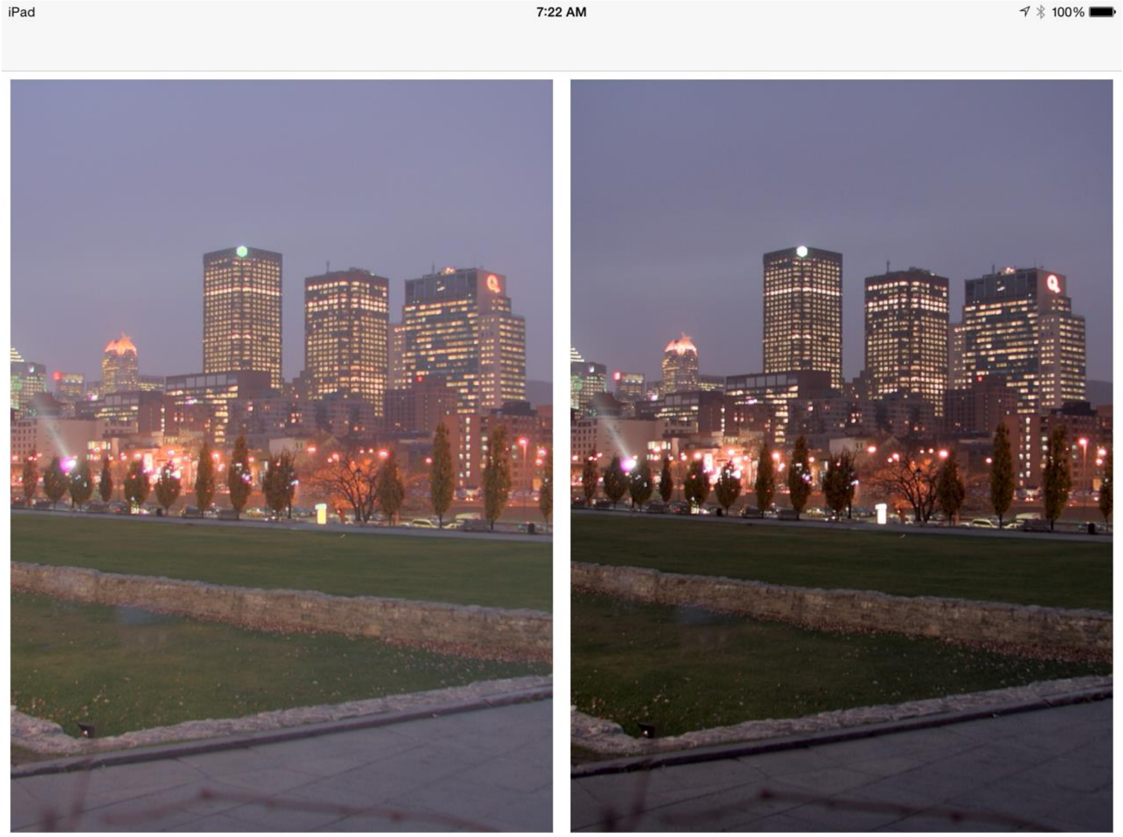
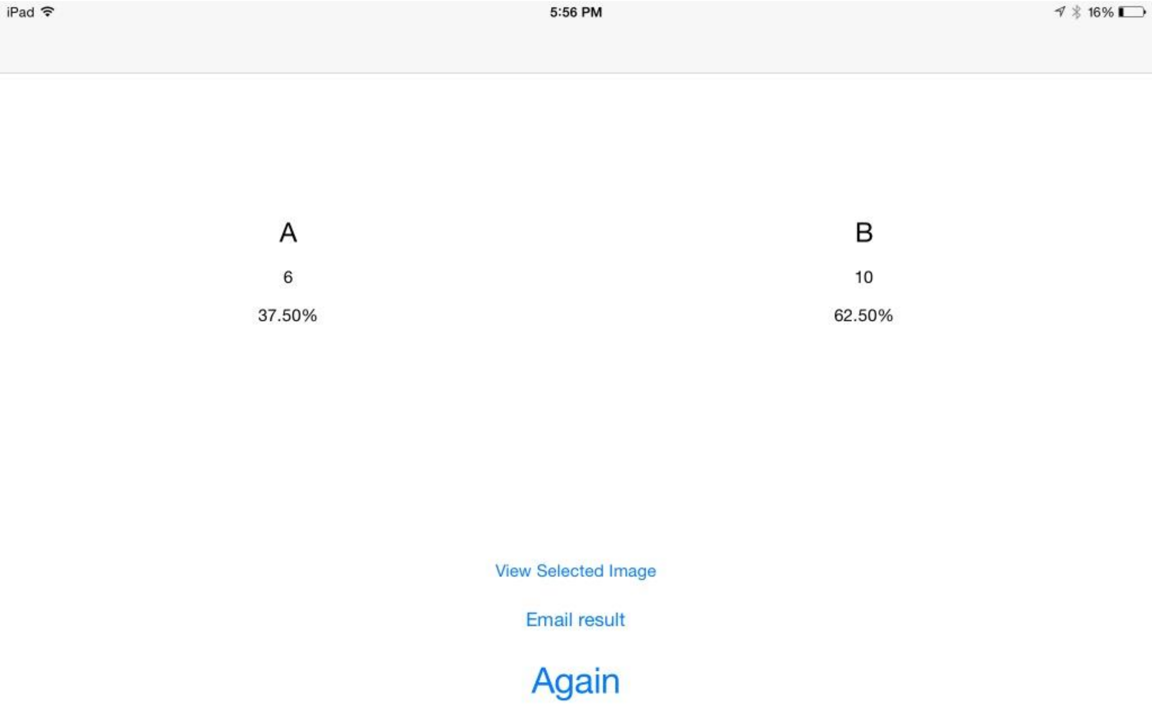


Figure 24: A screenshot of the image pair being compared

Figure 24 shows an example. Each pair is displayed for 10 seconds, during which time the subject can vote for the image that (s)he thinks has higher quality by touching that image on the touch screen. The vote is recorded and the app displays the next image pair. The 10 second interval comes from [30], which recommends visual stimuli be displayed to the subjects for 10 seconds. However, this can easily be modified to fit a particular experimental design. Within each pair, the app randomly chooses whether to display the image from class ‘A’ on the left or the right side (and thereby, the image from class ‘B’ on the opposite side). This is done to counteract side bias – a phenomenon whereby a subject may have a preference for the stimulus on one side of his/her field of view irrespective of the quality.

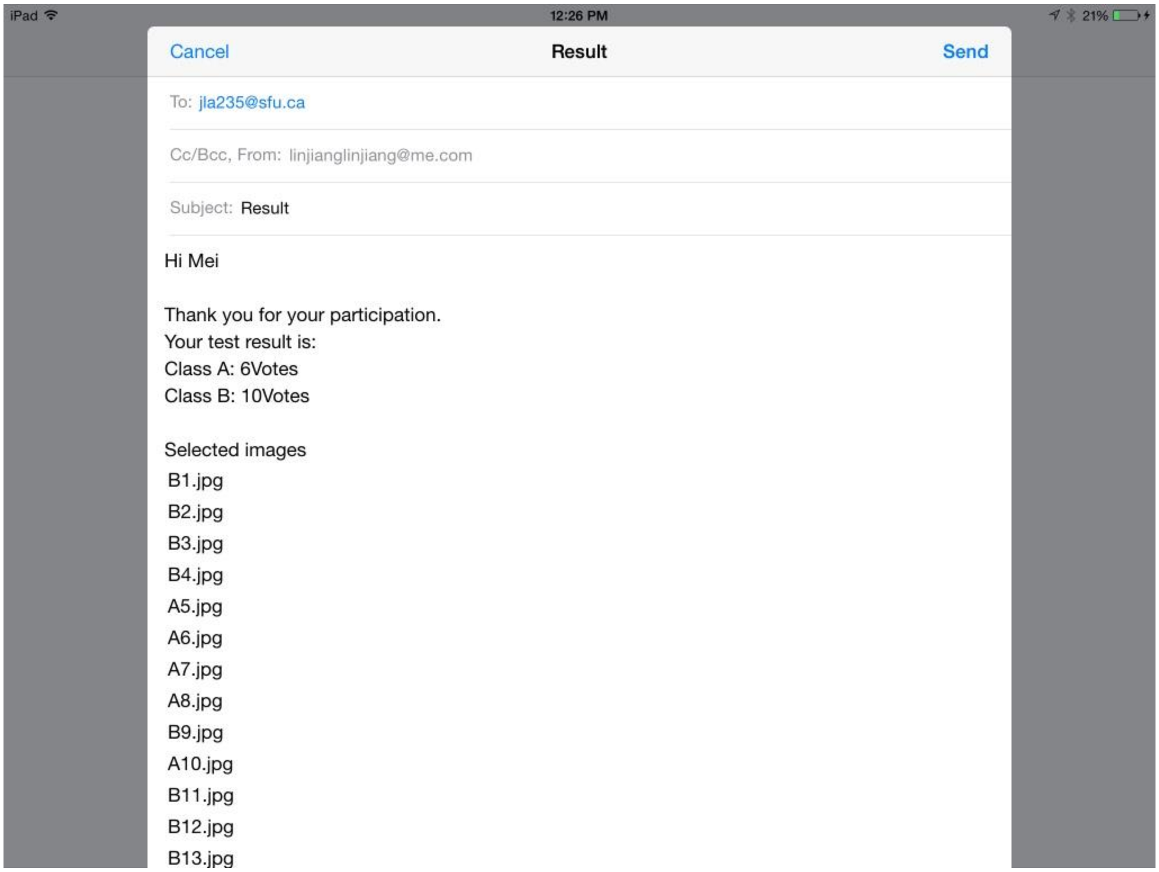
If the subject does not vote for any of the two images within the designated 10-second interval, the app randomly chooses one class (‘A’ or ‘B’) and records the random vote. This strategy enforces the forced choice aspect of 2AFC. If the images in a particular pair look so similar that the subjects cannot make up their mind as to which one looks better, the random voting will generate vote counts n_i that are close to the expected null-hypothesis vote count $N/2$. This will make the χ^2 test statistic small and thereby induce a large p-value, which will cause the null hypothesis not to be rejected. The same will happen if the subjects do vote within the designated 10-second period but split their votes nearly evenly between the images of the two classes. Once all image pairs are processed, the app provides a summary of the results for the subject and e-mails the complete results to the designated e-mail address.

Figure 25 shows a typical test ending screenshot. It shows a brief summary of the test. After selecting the “Email result” button, a summary email, as Figure 26 shows, pops up, and is sent to the email address set at the beginning of the test.



A	B
6	10
37.50%	62.50%

Figure 25: A Screenshot of the test ending page: a summary of the results is shown to tester



To: jla235@sfu.ca

Cc/Bcc, From: linjianglinjiang@me.com

Subject: Result

Hi Mei

Thank you for your participation.
Your test result is:
Class A: 6Votes
Class B: 10Votes

Selected images
B1.jpg
B2.jpg
B3.jpg
B4.jpg
A5.jpg
A6.jpg
A7.jpg
A8.jpg
B9.jpg
A10.jpg
B11.jpg
B12.jpg
B13.jpg

Figure 26: A typical email that contains the result will be constructed and sent to experimenter

3.4. Evaluation of the Test Application

To evaluate the developed test app, we collected a set of 16 High Dynamic Range (HDR) images from [37], comprising various indoor and outdoor scenes. Each HDR image was processed by Tone Mapping Operators (TMOs) from [35] and [36] with their default settings. This produced two Low Dynamic Range (LDR) images with 8 bits per color channel for each HDR image. Each LDR image was stored in a JPEG format with quality factor 100 and no subsampling. Images were assigned filenames according to the convention described in the previous section and copied to the image gallery on an iPad3 device. For the test, the screen resolution was set to 2548×1036 and the iPad was fully charged and with maximum screen brightness.

A total of 16 subjects (5 women, 11 men) took part in the experiment. All of them were between 20-30 years old, with normal or corrected-to-normal vision. They were naïve as to the purpose of the test, and were instructed to choose an image that they thought looked better in each image pair. The test took place in an office with conventional LED office lighting.

Table 3: Number of voters for the images produced by the two TMOs, along with the p-value of the corresponding χ^2 Test

Image pair	TMO [35]	TMO [36]	p-value
1	5	11	0.1336
2	8	8	1.0000
3	11	5	0.1336
4	8	8	1.0000
5	2	**14**	**0.0027**
6	7	9	0.6171
7	0	**16**	**6×10−5**
8	9	7	0.6171
9	10	6	0.3173
10	4	**12**	**0.0455**
11	11	5	0.1336
12	9	7	0.6171
13	11	5	0.1336
14	**12**	4	**0.0455**
15	8	8	1.0000
16	1	**15**	**0.0005**
Total	117	139	0.1691

Subjects' votes were sent by the app to the designated e-mail address, from where they were collected into an Excel sheet and statistically analyzed. The results are shown in Table 3, where the number of votes obtained for each image is given along with the p-value of the corresponding χ^2 test. Since there were 16 subjects in the experiment, under the null hypothesis (images of equal quality), each image is expected to obtain 8 votes, which is $E_1 = E_2 = 8$ in Equation (3.1). The p-values indicate that in most cases, the obtained votes did not deviate sufficiently from their expected values to reject the null hypothesis. Therefore, in most cases, the images produced by TMOs from [35] and [36] were of equal quality, in a statistical sense.

An example of the case where the votes were split evenly is shown in Figure 27. This figure shows image pair #2, where each image received 8 votes. It is evident that the images are different. The one on the right seems to have higher contrast, which is usually associated with high subjective quality. However, this comes at the expense of some overexposure, especially in the areas around and below the sun. In the end, the higher contrast did not help the image on the right – both images won the same number of votes, and are therefore considered to be, statistically speaking, of the same subjective quality.

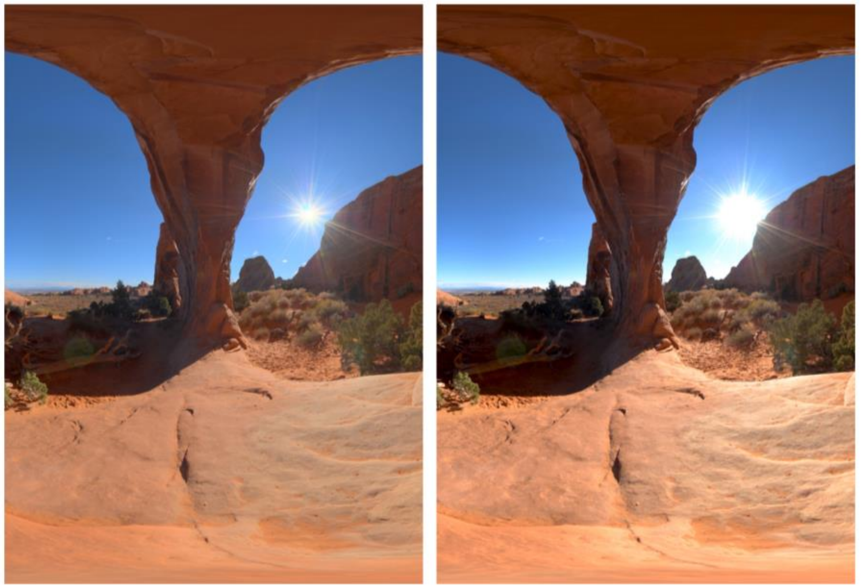


Figure 27: Image pair #2, where the votes were split evenly

There were 5 instances where $p < 0.05$ was obtained and the null hypothesis could be rejected. For these cases, the p-value and the statistically higher number of votes are indicated in bold in Table 3. In four of these cases, TMO from [36] produced the image that was judged to have better subjective quality, and in one case, TMO from [35] produced a better looking image. However, considering the total number of votes across the 16 images, the two TMOs were in a statistical tie – neither of them obtained a statistically significant advantage in the number of votes. Therefore, while each TMO may produce a better looking image in a specific instance, the results suggest that neither one can be considered better overall, at least in terms of the resulting image quality on mobile iOS devices.

This test was run simply to evaluate the mobile test app. In the next chapter, we employ the test app to compare the proposed image cropping approach against a benchmark that employs only bottom-up saliency.

Chapter 4.

Evaluation of the Proposed Automatic Cropping Technique

In this chapter, we will evaluate the proposed automatic cropping technique by comparing it against a suitably chosen benchmark. For the benchmark we choose a cropping system that bases its decisions only on the bottom-up saliency map produced by the AWS model. In other words, the benchmark system looks like Figure 7 with multiresolution object detection and multiresolution face/upper body detection removed. This can be considered as a representative of attention-based cropping methods [46, 47].

In [41], the researchers use the overlap percentage between the proposed crop and the “ground truth” to evaluate the performance of their cropping algorithm. The crop with the large overlap area was consider as the better crop. There are a lot of exceptions. For example, how to judge which one is better if two crops have the same overlap value. Also, the crop with the higher overlap value may be considered as the bad crop if it misses an important object. There is also a question of what is the “ground truth” in this case, because different people themselves may have different opinions on what is important in a given image. In this thesis, we use another way to evaluate the performance of the proposed cropping system, through subjective evaluation of final crop. This kind of methodology is commonly used in evaluating the image quality [2, 68].

4.1. Test Image Dataset

The image cropping literature generally uses small images (by today’s standards) for cropping. Even the latest work on the topic, [43], uses a dataset where the vast majority of images have width and height no larger than 1280. Since most of today’s displays, even mobile ones, have sufficient resolution to display such images, we believe image cropping should be tested on larger images. For this purpose, we selected a set of 20 high-resolution images from unsplash.com, shown in Figure 28. These images include a variety of content, from indoor to outdoor scenes, including humans. Most images have artistic flavor to them, which is a challenge for both bottom-up saliency models as well as object detectors, because the lighting and scene composition differ from “natural” images on which they are trained. Image resolutions in the dataset have a range from 1920×1280 to 5760×3840.

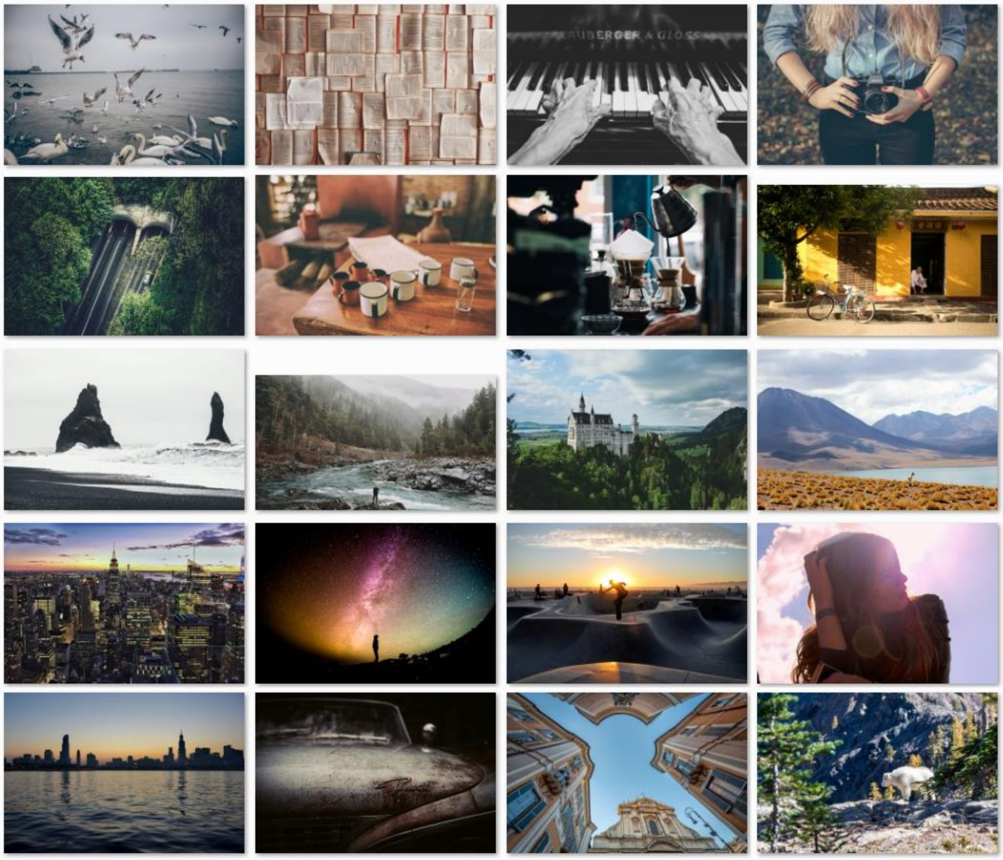


Figure 28: The sample images for testing

4.2. Test Setup

The target crop size for each test image was one quarter of its original size: half the width and half the height. The subjective test was carried out using our mobile test app. However, the test protocol is slightly different from the one described in the previous chapter. Since the crop is supposed to capture the most important part of the original image, the subjects need to know what the original image looks like. Hence, prior to showing each crop pair, we showed the original image to the subject.

The original image was shown for 5 seconds. After 5 seconds, the pair of cropped images, one produced by the benchmark and the other by the proposed method, was shown side by side for 10 seconds. This procedure was selected according to Section 4 in [29]. To counteract the side bias, the crops were randomly put on either side: the benchmark crop was sometimes on the left and sometimes on the right and vice versa for the proposed crop. During these 10 seconds, the subject was able to vote for whichever crop they think is a better representation of the original image by touching the corresponding image on the iPad. If the subject does not vote for any of the two images within the designated 10-second interval, the app randomly chooses one image and records the random vote. Once the subject finishes voting, the next original test image will show up, and the application will continue the process until all cases are tested.

A total of 22 subjects, 17 males and 5 females, took part in the experiment. All of them were between 20-30 years old, with normal or corrected-to-normal vision. They were naïve as to the purpose of the test, and were instructed to choose an image that they thought better describes the original image. Note that if the subjects had been asked a different question (e.g. which image looks better?), or if they had not seen the original image, the responses could have been different. The test took place in a room with conventional LED office lighting. Considering that the sizes of the tested images are very large, we projected the iPad screen on to a 55inch Samsung 4k TV and the subjects were looking at the TV instead of the iPad screen. Subjects' votes were sent by the app to the designated e-mail address, from where they were collected into an Excel sheet and statistically analyzed.

4.3. Test Result

Table 4: Number of voters for the images produced by the AWS based cropping and the proposed algorithm, along with the p-value of the corresponding χ^2 Test

Image pair	AWS Crop	Proposed Crop	p-value
1	5	17	0.0105
2	14	8	0.2008
3	14	8	0.2008
4	7	15	0.0880
5	11	11	1
6	17	5	0.0105
7	10	12	0.6698
8	0	22	2.73E-06
9	0	22	2.73E-06
10	12	10	0.6698
11	3	19	0.0006
12	0	22	2.73E-06
13	10	12	0.6698
14	15	7	0.0880
15	11	11	1
16	5	17	0.0105
17	3	19	0.0006
18	11	11	1
19	10	12	0.6698
20	9	13	0.3937
Total:	167	273	4.34E-07

Subjective test results are presented in Table 4, where the number of votes for each image crop produced by the benchmark and the proposed method is shown in the second and third column, respectively. The last column shows the p-value for the corresponding chi-squared (χ^2) test statistic [22]. In experimental sciences, the result is usually considered statistically significant when $p < 0.05$. There are eight such cases in Table 4, indicated in bold typeface. In seven of these cases (images 1, 8, 9, 11, 12, 16, 17) the proposed cropped image received a statistically higher number of votes, and these cases are shown in green. In one case (image 6, shown in red), the benchmark cropped image received a higher number of votes. In other cases the difference in the number of votes was not statistically significant. Overall, however, the proposed cropping method received a statistically larger number of votes, as indicated in the last row of the table.

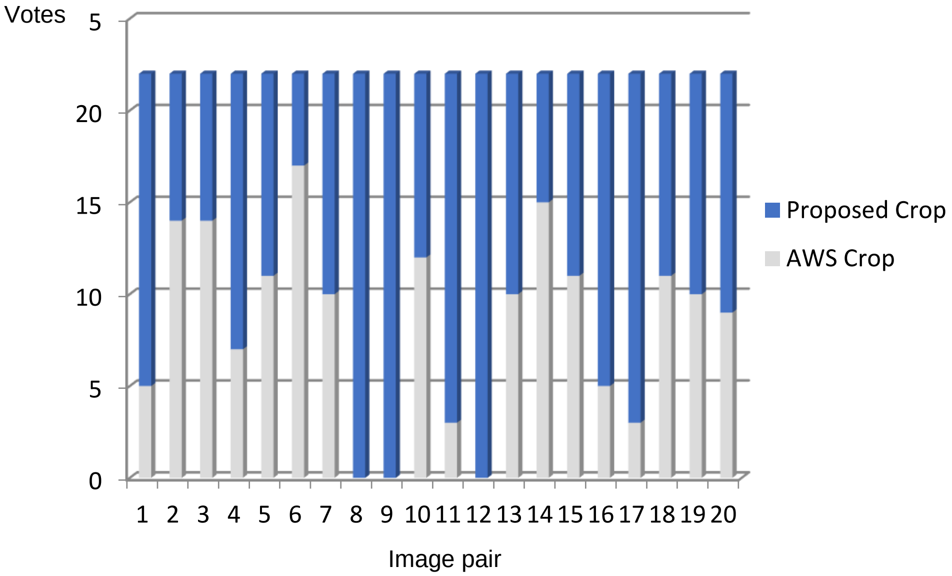


Figure 29: The chart showing now many votes each algorithm got (Orange for proposed; blue for AWS)

Another illustration of the results is shown in Figure 29. Here, the horizontal axis is the image pair index, and the vertical axis is the number of votes. The orange bars represent the votes for the proposed crop while the blue bars represent the votes for the benchmark crop. It is clear that the blue bar and orange bar are almost equal in most cases, but for image pairs 1, 8, 9, 11, 12, 16, and 17, the orange bar is much longer than the blue bar, which means the proposed crop was judged to be much better than the benchmark crop. Next, we show two illustrative examples.

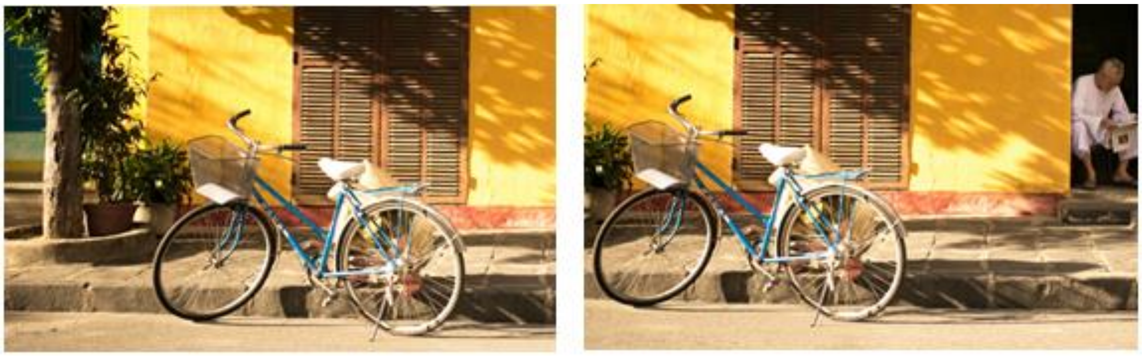


Figure 30: Image pair #8, where the proposed cropped image (right) got all the votes

Figure 30 shows an example of the case where the proposed crop got all the votes. The benchmark crop misses the human figure, while the proposed crop includes it.

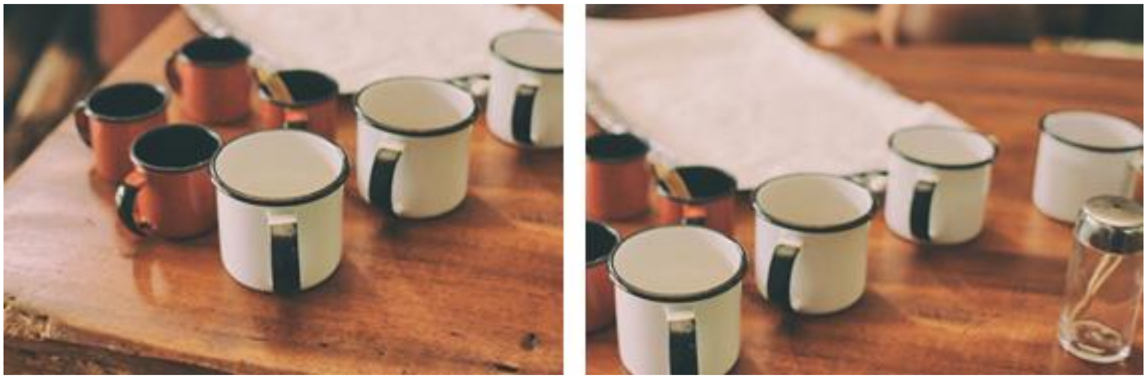
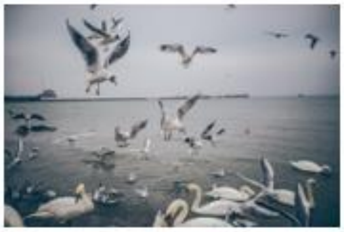
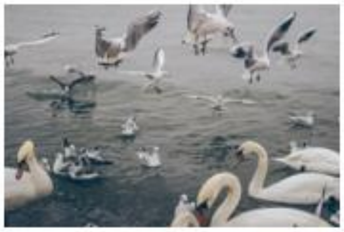
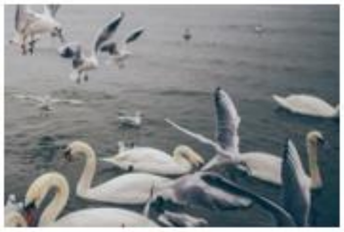
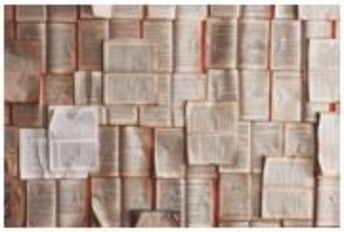
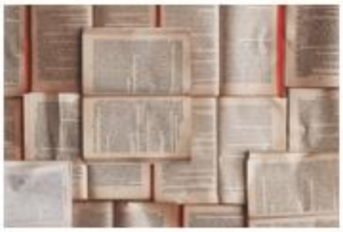
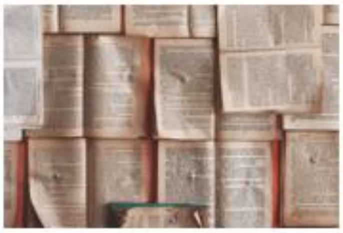
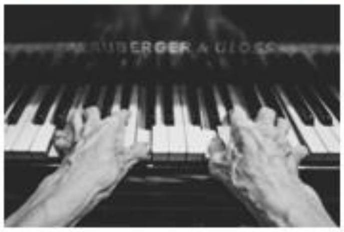
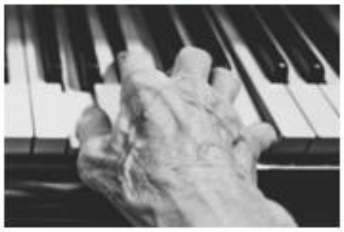
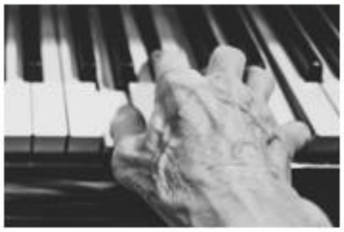
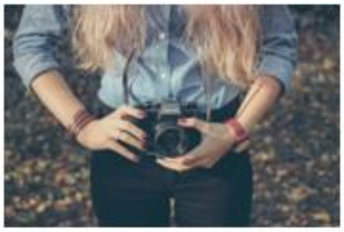
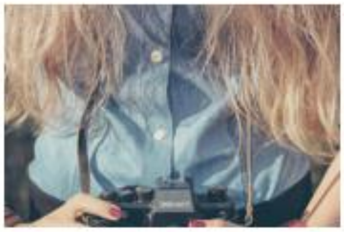
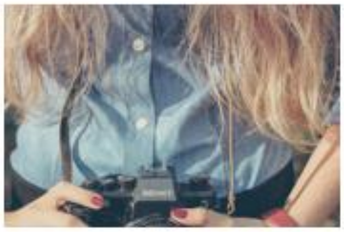
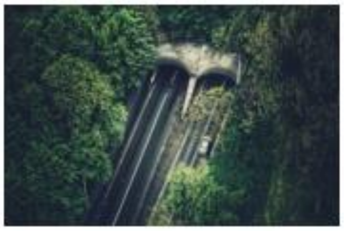
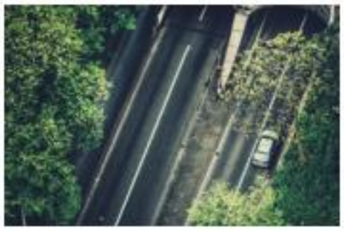
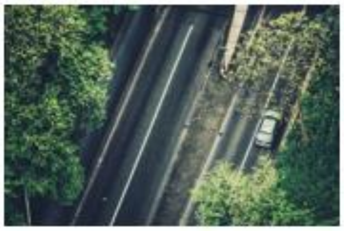
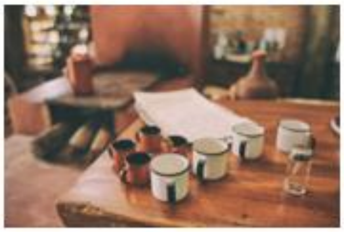
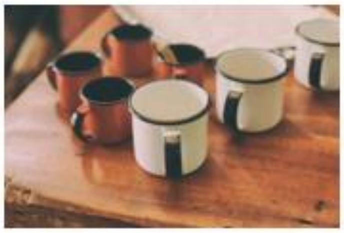
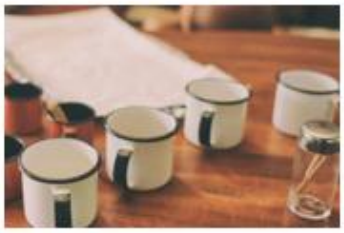


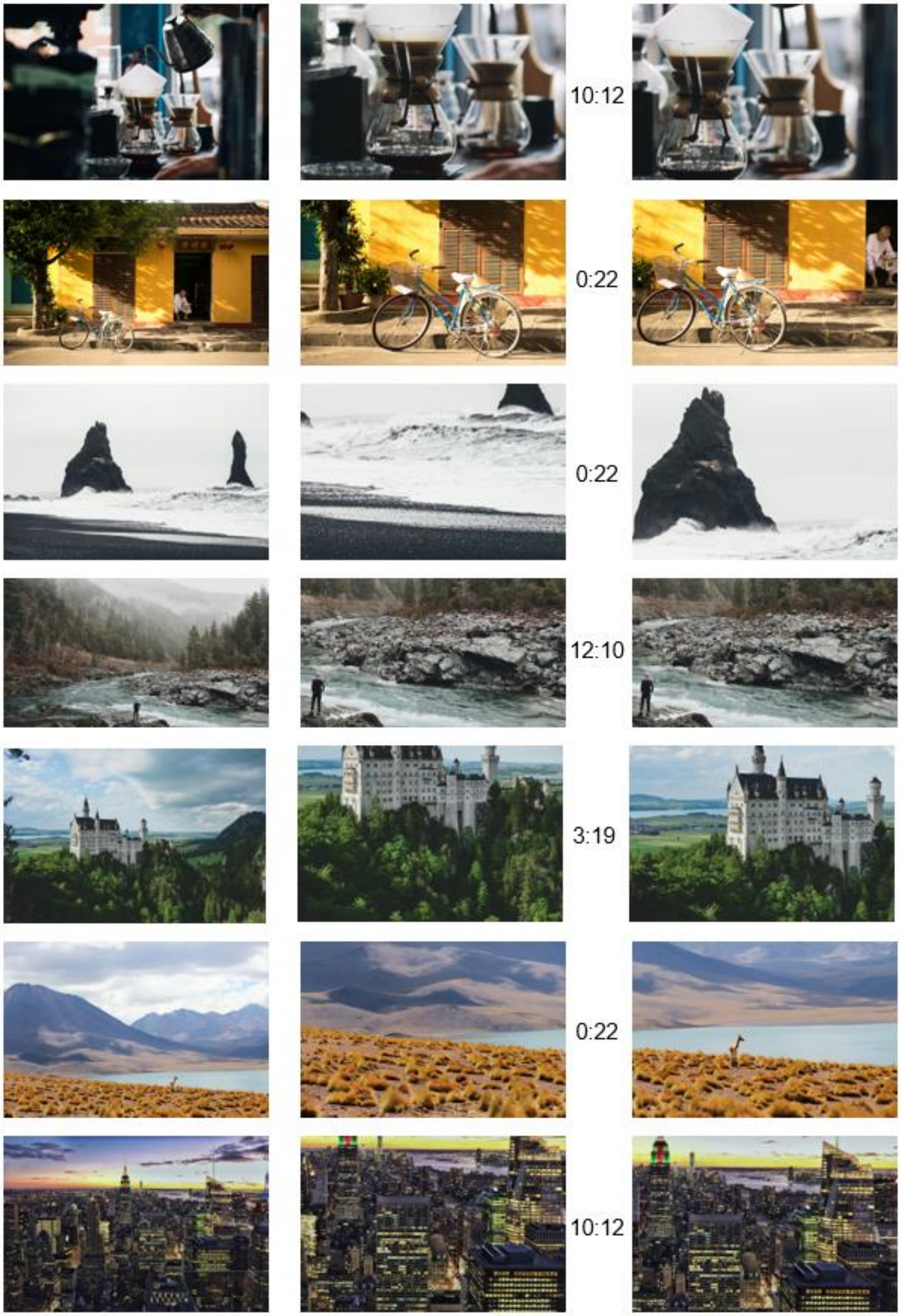
Figure 31: Image pair #6, where the proposed cropping image (right) got fewer votes than the AWS cropping image (left) and p-value is less than 0.05

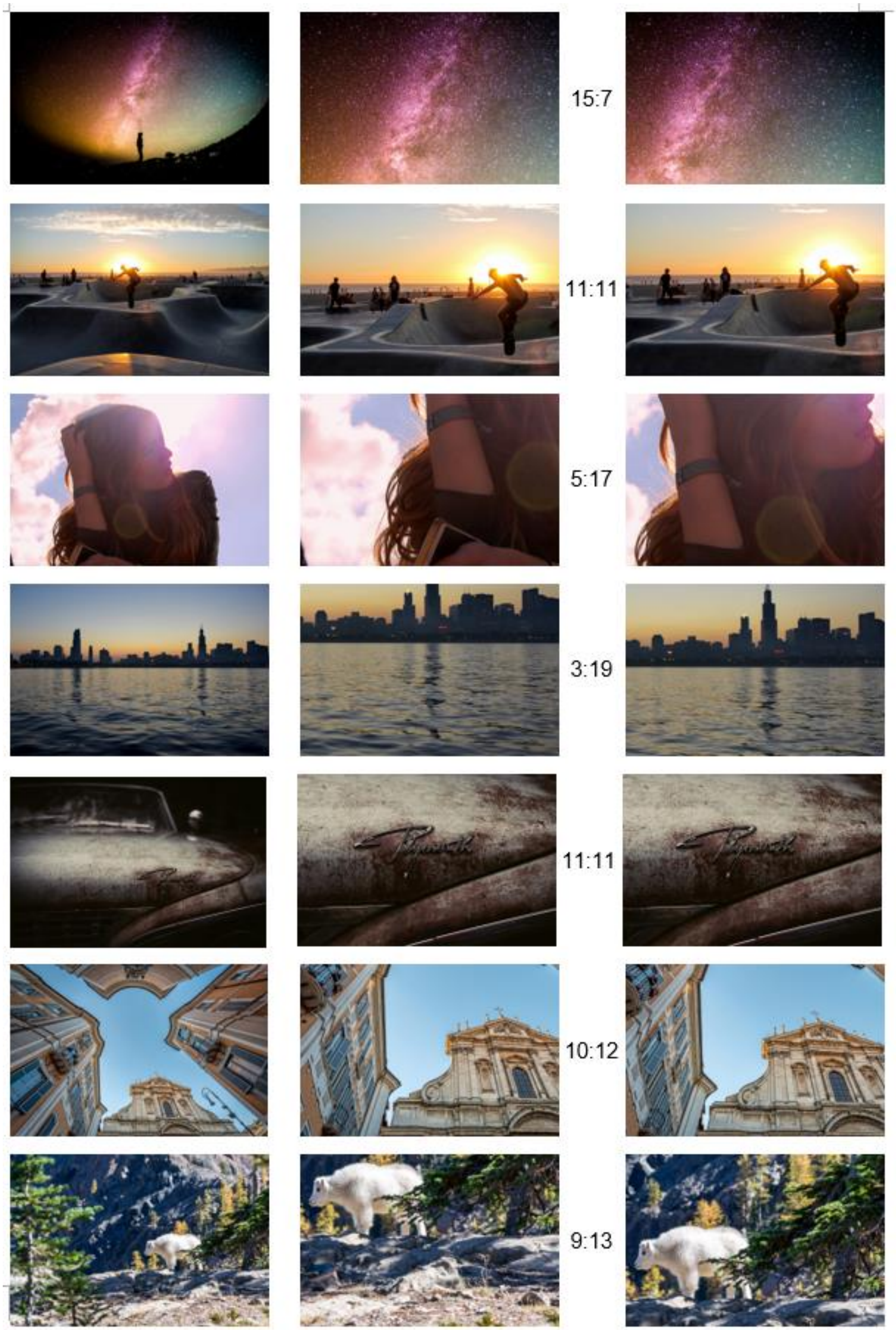
Figure 31 shows crops of image 6, where the benchmark crop was preferred (17 vs. 5). Even though the proposed crop includes more of the content of the original image (for example, the shaker in the bottom right, which is missing from the benchmark), the benchmark crop was preferred by the participants, presumably because of its higher aesthetic appeal. This illustrates the importance of aesthetic analysis for image cropping, especially in cases where the original image does not include humans and where aesthetic considerations may be considered more important than attention-related criteria.

Table 5 shows all the test images and also the voting results.

Table 5: All test images and results

Original	AWS	Votes	Proposed
		5:17	
		14:8	
		14:8	
		7:15	
		11:11	
		17:5	


10:12
0:22
0:22
12:10
3:19
0:22
10:12


15:7
11:11
5:17
3:19
11:11
10:12
9:13

Chapter 5.

Conclusions and Future Work

We presented an image cropping system built upon the principles of bottom-up saliency and top-down semantics. Taking advantage of recently developed high-performance object detection and classification, we were able to construct an importance map for the image, which assigns different importance to various classes of objects and combines this with bottom-up saliency. Images cropped using this approach were judged to be better overall by participants in a subjective test, compared to conventional saliency-based crops. However, there were cases where aesthetics-related factors led to saliency-based crops to be preferred over the proposed crops, despite the fact that they were missing out on some content.

A number of extensions of the proposed cropping system are possible. For example, different object groupings and greater differentiation of weights in Table 2 could lead to better importance maps. Weights can even be personalized if the user preferences are known, for example from their social media profiles. Also, importance map fusion in Equation (2.3) may be optimized. Another improvement could be the inclusion Recent object detectors, such as YOLOv2 [67] and Faster R-CNN [61], are getting better at classifying a larger set of objects and could potentially provide similar level of semantic interpretation together with better localization compared to our current object detection module. Our overall system architecture is flexible enough to allow replacement of an object detector on a plug-and-play basis.

Finally, a recent trend in many research areas has been to train end-to-end deep network models to perform certain tasks without separating them into smaller sub-problems, and this approach might work for image cropping as well. However, this would require much larger training datasets of “correct” crops that are currently not available.

References

[1] M. Nishiyama, T. Okabe, Y. Sato, and I. Sato, “Sensation-based photo cropping,” *Proc. ACM Multimedia*, pp. 669–672, 2009.

[2] J. She, D. Wang, and M. Song, “Automatic image cropping using sparse coding,” *in Proc. ACPR*, pp. 490–494, 2007.

[3] S. Dhar, V. Ordonez, and T. L. Berg, “High level describable attributes for predicting aesthetics and interestingness,” *Proc. IEEE CVPR’11*, pp. 1657–1664, 2011.

[4] Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, May 2001.

[5] F.-F. Li and P. Perona, “A Bayesian hierarchical model for learning natural scene categories,” *Proc. IEEE CVPR’05*, vol. 2, pp. 524–531, Jun. 2005.

[6] W. Fulton, *Image Resize: Cropping, Resampling, Scaling*, 2012. Available: https://www.scantips.com/lights/resize.html

[7] DPChallenge: http://www.dpchallenge.com

[8] Photo.net: http://photo.net.

[9] Flickr: http://www.flickr.com

[10] A. Borji and L. Itti, “State-of-the-art in visual attention modeling,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 35, no. 1, pp. 185–207, 2013.

[11] A. Borji, D. Sihite, and L. Itti, “Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study,” *IEEE Trans. Image Process*., vol. 22, no. 1, pp. 55–69, 2013.

[12] M. Hayhoe and D. Ballard, “Eye movements in natural behavior,” *Trends in Cognitive Sciences,* pp. 188–194, 2005.

[13] L. Itti and C. Koch, “Computational modelling of visual attention,” *Nature Reviews Neuroscience,* vol. 2, no. 3, pp. 194–203, 2001.

[14] M. Treisman and G. Gelade, “A feature-integration theory of attention,” *Cognitive Psychology,* pp. 97–136, 1980.

[15] J. M. Wolfe, K. R. Cave, and S. L. Franzel, “Guided search: an alternative to the feature integration model for visual search.” *J. Exp. Psychol. Human*., vol. 15, no. 3, p. 419, 1989

[16] J. M. Wolfe, “Guidance of visual search by preattentive information,” in *Neurobiology of Attention*, pp. 101–104, 2005.

[17] C. Koch and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry,” in *Matters of Intelligence*, pp. 115–141, 1987.

[18] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1154-1259, 1998.

[19] J. Harel, C. Koch, and P. Perona, “Graph-based visual saliency,” *Proc. NIPS’07*, pp. 545–552, 2007.

[20] A. Garcia-Diaz, “Modeling early visual coding and saliency through adaptive whitening: plausibility, assessment and applications,” Ph.D. Thesis, Higher Technical Engineering School, University of Santiago de Compostela, 2011.

[21] Unsplash.com: http:www.unsplash.com

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, “Going deeper with convolutions”, *Proc. IEEE. CVPR’15*, pp. 1-9, 2015.

[23] ImageNet Large Scale Visual Recognition Challenge: http://www.image-net.org/challenges/LSVRC/

[24] P. Viola, and M. J. Jones, “Robust real-time object detection,” *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[25] J.P. Lewis, “Fast template matching,” *Proc. Vision Interface*, pp. 120–123, 1995.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[27] W. Lin and C.-C. Jay Kuo, “Perceptual visual quality metrics: A survey,” *J. Vis. Commun. Image R.*, vol. 22, no. 4, pp. 297-312, May 2011.

[28] Q. Li, Y. Fang, W. Lin, and D. Thalmann, “Gradient-weighted structural similarity for image quality assessments,” *Proc. IEEE ISCAS’15*, pp. 2165-2168, May 2015.

[29] ITU-R BT.500-13: Methodology for the subjective assessment of the quality of television pictures, Jan. 2012.

[30] M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics*, Peninsula Pub, 1989.

[31] H. Hadizadeh, I. V. Bajić, P. Saeedi, and S. Daly, "Good-looking green images," *Proc. IEEE ICIP'11*, pp. 3177-3180, Sep. 2011.

[32] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19-33, Jan. 2014.

[33] J. T. McClave and T. T. Sincich, *Statistics*, 12th Edition, Pearson, 2011.

[34] *NIST / SEMATECH e-Handbook of Statistical Methods*, 2012, Chapter 1, [Online] Available: http://www.itl.nist.gov/div898/handbook/

[35] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph*., vol. 21, no. 3, pp. 267-276, Jul. 2002.

[36] R. Mantiuk, S. Daly, and L. Kerofsky, "Display adaptive tone mapping," *ACM Trans. Graph*., vol. 27, no. 3, pp. 68:1-68:10, Aug. 2008.

[37] HDR Labs - sIBL Archive: http://www.hdrlabs.com/sibl/archive.html

[38] J. Lin and I. V. Bajić, "A platform for subjective image quality evaluation on mobile devices," *Proc. IEEE CCECE'16*, pp. 1-4, May 2016.

[39] Camera photo resolution image: http://www.camerasunderwater.info/images/stories/info/imaging/video-vs-stills-resolutions_1500w_2.jpg

[40] J. Chen, G. Bai, S. Liang, and Z. Li, "Automatic image cropping: a computational complexity study," *Proc. IEEE CVPR'16*, pp. 507-515, 2016

[41] Fang, Z. Lin, R. Mech, and X. Shen. "Automatic image cropping using visual composition, boundary simplicity and content preservation models," *Proc. ACM Multimedia*, pp. 1105-1108, 2014.

[42] J. Yan, S. Lin, S. B. Kang, and X. Tang, "Learning the change for automatic image cropping", *Proc. IEEE CVPR'13*, pp. 971-978, 2013.

[43] Y.-L. Chen, T.-W. Huang, K.-H. Chang, Y.-C. Tsai, H.- T. Chen, and B.-Y. Chen. "Quantitative Analysis of Automatic Image Cropping Algorithms: A Dataset and Comparative Study," *Proc. IEEE CVPR'17*, pp. 1657–1664, 2017.

[44] S. Meech. *Contemporary Quilts: Design, Surface and Stitch*, London Batsford, 2005.

[45] ImageNet Large Scale Visual Recognition Challenge test image category labels list: http://image-net.org/challenges/LSVRC/2014/browse-synsets

[46] H. Suh, B. Ling, B. B., and D. W. Jacobs, "Automatic thumbnail cropping and its effectiveness," *Proc. ACM Symp. User Interface Software Tech. (UIST'03)*, pp. 95–104, 2003.

[47] F. Stentiford, "Attention based auto image cropping," *Proc. ICVS Workshop on Computation Attention and Applications (WCAA)*, 2007.

[48] M. Zhang, L. Zhang, Y. Sun, L. Feng, and W. Ma, "Auto cropping for digital photographs," *Proc. IEEE ICME'05,* pp. 1-4, 2005.

[49] J. Luo, "Subject content-based intelligent cropping of digital photos," *Proc. IEEE ICME'07*, pp. 2218-2221, 2007.

[50] G. Ciocca, C. Cusano, F. Gasparini, and R. Schettini, "Self-adaptive image cropping for small displays," *IEEE Trans. Consumer Electronics*, vol. 53, no. 4, pp. 1622–1627, 2007.

[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211-252, Dec. 2015.

[52] J. Lin and I. V. Bajić, "Automatic image cropping based on bottom-up saliency and top-down semantics", presented at *IEEE PacRim'17*, Victoria, BC, Aug. 2017.

[53] "Summed-area table," in *Wikipedia, The Free Encyclopedia*. Available: https://en.wikipedia.org/wiki/Summed-area_table

[54] H. Hadizadeh and I. V. Bajić, "No-reference image quality assessment using statistical wavelet-packet features," *Pattern Recognition Letters*, vol. 80, pp. 144-149, Sep. 2016

[55] H. Hadizadeh and I. V. Bajić, "Color Gaussian jet features for no-reference quality assessment of multiply-distorted images," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1717-1721, Dec. 2016.

[56] M. Rubinstern, D. Gutierrez, O. Sorkine and A. Shamir, "A comparative study of image retargeting," *ACM Trans. Graph.,* vol. 29, no. 6, 2010

[57] B. C. Das, V. Gopalakrishnan, K. N. Iyer and A. Gaurav, "Similarity and rigidity preserving image retargeting", *Proc. IEEE ICIP'16*, pp. 1584-1588, Sep. 2016

[58] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You only look once: Unified, real-time object detection,” *Proc. IEEE CVPR’16*, pp. 779–788, 2016.

[59] R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE CVPR’14,* pp. 580–587. 2014

[60] R. B. Girshick. “Fast R-CNN,” *Proc. IEEE ICCV’15,* pp. 1440–1448. 2015

[61] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards realtime object detection with region proposal networks,’’ *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[62] Result of LSVRC 2015:

http://www.image-net.org/challenges/LSVRC/2014/results

[63] L. Marchesotti, C. Cifarelli, and G. Csurka. “A framework for visual saliency detection with applications to image thumbnailing,” *Proc. IEEE ICCV’09,* pp. 2232–2239, 2009.

[64] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs. “Automatic thumbnail cropping and its effectiveness,” *ACM UIST*, pp. 95–104, 2003

[65] V. Jain, and E. Learned-Miller, “FDDB: A Benchmark for Face Detection in Unconstrained Settings,” Technical report, University of Massachusetts, 2010

[66] D. Mishkin. Models accuracy on imagenet 2012 val.
https://github.com/BVLC/caffe/wiki/Models-accuracy-on-ImageNet-2012-val

[67] J. Redmon, and A. Farhadi. “YOLO9000: Better, Faster, Stronger,” *Proc. IEEE CVPR’17*, pp. 6517–6525, 2017.

[68] Y.-M. Kuo, H.-K. Chu, M.-T. Chi, R.-R. Lee and T.-Y. Lee, “Generating Ambiguous Figure-Ground Images,” IEEE Trans. Visualization Comput. Graph. Vol. 12, no. 5, pp1534 -1545, May 2017

[69] S. Jadhav and P. Dighe, “Image resizing for thumbnail images by using seam carving,” *IEEE ICESA’15,* pp.636-640, 2016

[70] W. Liu, D. Anguelov, D. Erhan, C. Szegedy and S. E. Reed, “SSD: single shot multibox detector,” *arXiv:1512.02325*

[71] S. Zafeiriou, C. Zhang and Z. Zhang, “A survey on face detection in the wild: Past, present and future,” *Comput. Vis. Image Underst.* vol. 138 pp. 1-24, Sep 2015

[72] P.F. Felzenszwalb, R.B. Girshick, D. McAllester and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 32, no. 9, pp.1627–1645, 2010