Universiteit Leiden
The Netherlands

Opleiding Informatica

Segmentation based Image Retrieval using
Low level image features

Andrew Huang

Supervisors:
Prof. dr. Michael S. Lew

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

02/07/2021

Abstract

Retrieving images based on content is increasingly becoming more useful due to the volume of images available and the absence of human labour to correctly label these images. The alternative is to use image descriptor that describe attributes, such as shape, color and texture, and these low level features perform reasonably well. A user submits a query image to the database, the image is described using the image descriptor and it returns the images that are most similar to the given image. Unfortunately, the low level features are prone to noise and often include information that is not useful. To mitigate this, a semantic segmentation step has been proposed in the retrieval pipeline. Mask Region-based Convolutional Neural Network (Mask R-CNN) is used to semantically extract subregions from a query image, from which we extract Local Binary Patterns (LBP), Histograms of Oriented Gradients (HOG), color and Scale Invariant Feature Transform (SIFT) features. We compare the retrieved results against an unsegmented query image using the Mean Average Precision (mAP). We show that the retrieval performance of scale and rotation invariant features (RGB histograms, HOG, SIFT), is improved by using segmentation and that performance for features that are not invariant (LBP, COLTEX) is decreased when using segmentation.

Contents

1 Introduction

In the past two decades, storing and searching images in online databases has become a part of daily life. When a user wants to search for images, different methods are used to efficiently and accurately retrieve a collection of relevant images. Such methods include things as entering keywords to compare with annotations or keywords of each image and returning the images that have the most overlap. This has progressively become a more difficult and expensive way to search, due to the amount of images available and the human labour of describing each image.

A solution is to search for images not based on annotations, but by a query image, where a user gives an image to a database and it will return the images that are most similar to the query image through the use of a similarity measure, such as Euclidean, Manhattan and Square Chi-squared[1]. Many Content Based Image Retrieval (CBIR) systems have been proposed with earlier systems indexing images based on single attributes[2] to indexing using features extracted by complex neural networks[3]. Since images are composed of objects, many have proposed that using semantic segmentation could improve the retrieval performance of an image descriptor by allowing the extraction method to focus on the objects itself.

One type of image descriptors that could benefit from such segmentation is the low level image descriptor. These features are computationally very efficient at describing an image and are more general than higher level features. The downside to this, is that with images that contain different objects of varying size and shape, they often include noisy information around the objects of interest. Segmentation would solve this weakness and should produce a more robust image descriptor.

Over the years, many segmentation methods have been proposed[4]. One way that has become more prevalent over the years is to use machine learning to determine where the Region of Interest (ROI) of an image are. Fully Convolutional Networks or FCN[5][6] use multiple layers of convolution to preprocess an image before it is inputted to the classifying neural network. This allows for varying sizes of input to be fed to the network. which is not possible with standard CNNs. These FCNs have proven to perform well on standard segmentation datasets[7][8], such as PASCAL and ImageNet. In this research we will propose an image retrieval pipeline that uses a FCN to semantically segment the image before extracting features. We then compare these to the features of non-segmented images using mAP to show that segmentation does improve retrieval accuracy. For the segmentation we will use FCN based network, Mask R-CNN[9], which can accurately segment objects in general images. Our assumption is that semantic segmentation will improve retrieval accuracy for low level image descriptor, such as color and texture histograms. We will compare five different image descriptor: Local Binary Patterns[10], RGB color histograms, HSV color histograms[11], Histogram of Oriented Gradients[12] and Scale Invariant Feature Transforms[13] and compare retrieval results using the mAP score between a pass where the query image is segmented before feature extraction and a pass where no segmentation is done.

2 Definitions

Acronyms

CBIR Content Based Image Retrieval. 1, 4, 5, 20, *Glossary:* CBIR

HOG Histograms of Oriented Gradients. 2, 8, 9, 15, 16, 19, *Glossary:* Histograms of Oriented Gradients

LBP Local Binary Patterns. 2, 7, 9, 10, 15–17, 19, 20, *Glossary:* Local Binary Patterns

mAP Mean Average Precision. 1, 2, 12, 16, 17, 19, 20, *Glossary:* Mean Average Precision

Mask R-CNN Mask Region-based Convolutional Neural Network. 1, 2, 5, 6, 10, *Glossary:* Mask Region-based Convolutional Neural Network

MS COCO Microsoft Common Objects in Context. 10, 15, 19, *Glossary:* Microsoft Common Objects in Context

ROI Region of Interest. 1, 6–10, 15, *Glossary:* Region of Interest

SIFT Scale Invariant Feature Transform. 2, 9, 10, 16, 17, 19, 20, *Glossary:* Histograms of Oriented Gradients

Glossary

CBIR A Content Based Image Retrieval (CBIR) is a method of retrieving images from a database using the image properties instead of external information, such as user annotations. 1

COLTEX COLTEX is an image descriptor that calculates the distribution of the relationship between hue and intensity with three relational matrices.. 2, 8, 9, 15, 16, 19, 20

Histograms of Oriented Gradients Histograms of Oriented Gradients is an image descriptor to describe the structure or shape of an image by extracting the gradient and orientations of the edges.. 2

image descriptor An image or feature descriptor is a value that describes the content of an image. Descriptors can be calculated from different aspects of an image and allows for comparisons between images.. 1, 2, 4, 5, 7–10, 15–17, 19, 20

keypoint A keyoint is a spatial location in an image that defines a point of interest.. 9

Local Binary Patterns Local Binary Patterns is an image descriptor that describes textural features of an image, using pixel neighbours to calculate the distribution of intensity changes over an image.. 2

Mask Region-based Convolutional Neural Network A Region-based Convolutional Neural Network that in addition to classifying objects in an image, also produces bounding boxes and masks for each classified object.. 2

Mean Average Precision Mean Average Precision is an evaluation metric used in computer vision tasks. It is a single value score of the average of Average Precision.. 2

Microsoft Common Objects in Context MS COCO is a large image dataset used for detection, keypoint, segmentation and retrieval tasks. It contains 80 object classes and has annotations for each image in its dataset.. 10

precision Precision measures the ratio of relevant images over retrieved images.. 11

recall Recall measures the ratio of relevant images over all images that were retrieved. 11

Region of Interest Regions of interest are regions of an image that contain key features.. 1

3 Related Works

Intuïtively, segmenting an image into smaller sub-images and feeding those sub-images to the database should retrieve more accurate results than not. Guo et al.[14] shows an outline of such an image retrieval pipeline using segmentation, as well as providing the possible benefits of using segmentation. These benefits include such things as computational efficiency and an improvement in accuracy by removing background noise. It also outlines some possible weaknesses of using segmentation in retrieval tasks.

The use of segmentation in CBIR systems is almost as old as the introduction of the system itself. Williams et al. 1998[15] introduces an algorithm for segmenting color images into homogeneous regions and extracting low level features from them.

More recent research has focused more on the feature extraction methods that introduce some form of segmentation. For example, using the global distribution of color to segment the image (Borah et al., 2005)[16] and extracting size and shape histograms has been tested against standard image descriptors such as color chromaticity moments[17] and have retrieved more accurate images compared to the unsegmented image descriptor.

The combination of low level features to describe image segments[18], produced better results than other existing methods, especially for occluded images. This seems to suggest that low level features do indeed benefit from semantic segmentation. Yuvaraj et al.[19] uses color histograms and residual values which segments the input images and extracts features from those segments before indexing them in the database.

These methods unfortunately require that special modifications be made to the feature extraction algorithm that makes use of the segments. This makes a direct comparison between the retrieval performance of an unsegmented image against the retrieval performance of a segmented image, something that this paper aims to answer.

Recent methods in the semantic segmentation of images have focused on using deep neural networks. Since these networks are typically trained on large amounts of labeled training data[20][21], there are unsupervised models that use pretrained networks to extract mid-level deep features for segmentation prediction[22], or clustering based on convolutional features[23]. The downside to unsupervised models is that these models come at the cost of accuracy[24]. Girschik et al. proposed a supervised pre-trained CNN that extracts features from a set number of category-independent region proposals[25] and then predicts bounding box values. An improvement on this R-CNN has achieved 66% accuracy on PASCAL VOC 2012 with faster computation[26]. These Region based CNNs perform well on datasets with a wide variety of classes and are able to extract many instances from an image.

4 Methodology

Our goal is to show that low level feature extraction on segments will improve retrieval results compared to retrieval on full images. To show this we will choose a number of image descriptors, retrieve images using an arbitrary query image and its segmented image and compare the retrieved images against each other. Figure 1 shows the image retrieval pipeline of an image that is to be segmented as well as unsegmented. It generally consists of six steps:

1. The query image that is given to the CBIR system
2. Segmentation and preprocessing.
3. Feature Extraction on image segments.
4. Feature Comparison.
5. Image Ranking.
6. Image Retrieval.

In this chapter, we will briefly go over the components of the image retrieval pipeline.

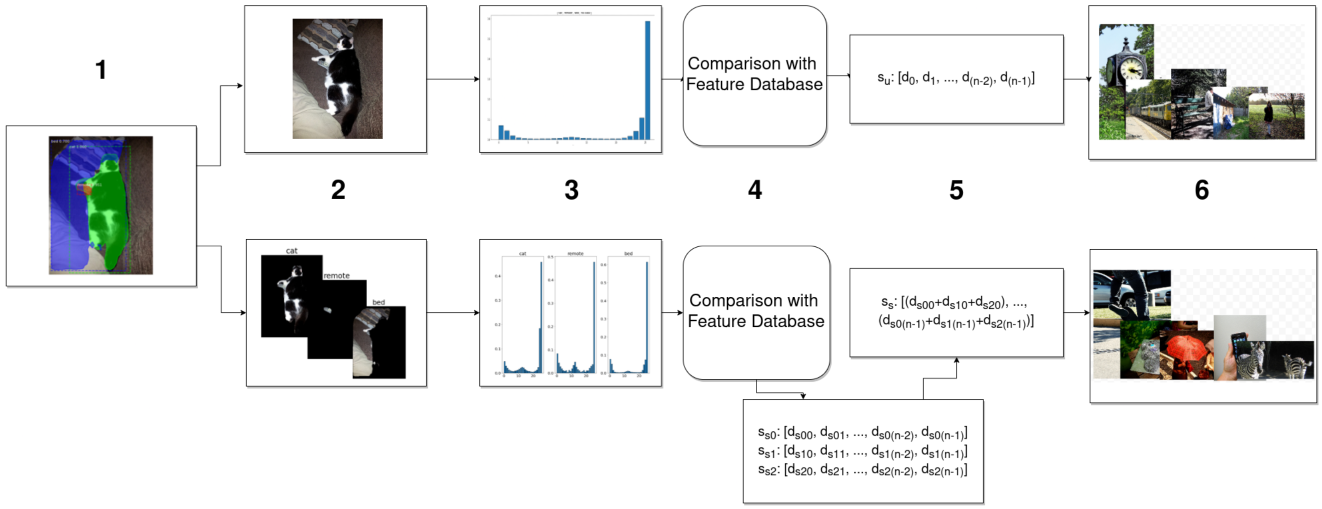

1
2
3
4
5
6
Comparison with Feature Database
s_u: $[d_0, d_1, ..., d_{(n-2)}, d_{(n-1)}]$
cat
remote
bed
Comparison with Feature Database
s_s: $[(d_{s00}+d_{s10}+d_{s20}), ..., (d_{s0(n-1)}+d_{s1(n-1)}+d_{s2(n-1)})]$
s_{s0}: $[d_{s00}, d_{s01}, ..., d_{s0(n-2)}, d_{s0(n-1)}]$
s_{s1}: $[d_{s10}, d_{s11}, ..., d_{s1(n-2)}, d_{s1(n-1)}]$
s_{s2}: $[d_{s20}, d_{s21}, ..., d_{s2(n-2)}, d_{s2(n-1)}]$

Figure 1: Flow chart of the image retrieval pipeline

4.1 Segmentation

For this research, we have chosen for Mask R-CNN in our segmentation process, which is an extension to Faster R-CNN[27] that introduces an object mask prediction branch along with Faster R-CNN's bounding box recognition branch. This particular method was chosen for its performance in handling instance segmentation. Instance segmentation, on top of identifying semantically what objects are present in an image, can also identify the object outlines of each instance of that object at a pixel level as Figure 2 illustrates. For this research, this means that features can be extracted from each object instance, rather than object regions, which allows for effective denoising.
We use Matterport's implementation[28] of Mask R-CNN, which consists of 3 parts:

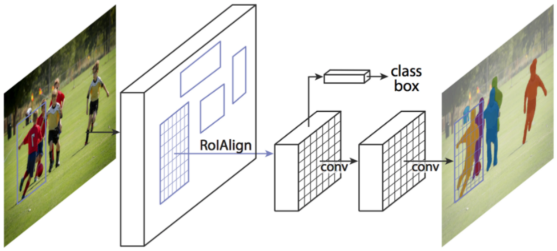


Figure 2: Mask R-CNN framework

1. Feature Extraction Network, which consists of a ResNet101[29] CNN to extract low and higher level features into a feature map. A Feature Pyramid Network (FPN)[30] is used in addition to pass features from higher down to lower layers.

2. This feature map is then fed to the Region Proposal Network (RPN). The RPN scans the feature map for ROIs, or anchors. For each anchor, RPN gives the class of the anchor (either foreground or background) and the bounding box refinement. At the end the top anchors are chosen as the final ROIs.

3. Finally, the generated ROIs are then classified through a deep CNN which classifies the specific object classes of each region. Masks are also generated in this step from the ROIs.

These masks can then be used to extract specific objects from the image for feature extraction, as shown in Figure 3.

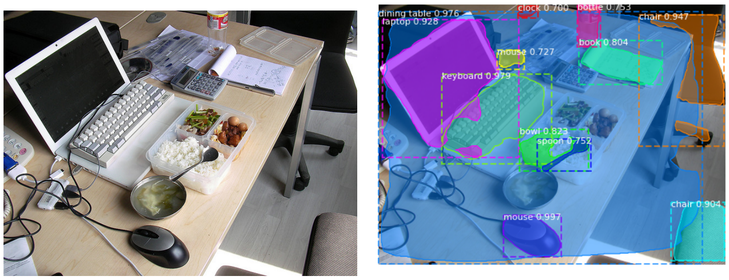


Figure 3: Example of the ROI masks obtained from the MRCNN.

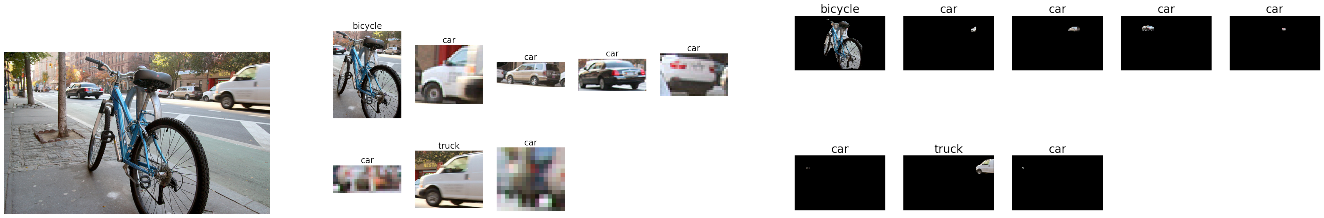
4.2 Segmented Images

Using the obtained masks from the Mask R-CNN, we can effectively isolate objects within an image and feed it to the extraction algorithm. Since different extraction algorithms require different preprocessing, we will use two ways to segment the query image:

1. Using the bounding boxes to crop the image into a MxN subimage.

2. Using the roi masks to extract the object from the query image and use padding to isolate it.

Figure 4 shows an example of the segmentation preprocessing. Which preprocessing is used will be discussed below.



(a) The original unsegmented query.

(b) The cropped sub images using bounding boxes.

(c) The extracted subimages using Mask R-CNN rois.

Figure 4: Example of segmentation preprocessing for the different feature extraction methods.

4.3 Feature Extraction

Images are commonly described using color, shape, texture, keypoints and there are many methods to obtain descriptors for these attributes. We will focus on five descriptors that describe these attributes and give a short overview of each.

4.3.1 Local Binary Patterns

LBP[10] are used to detect intensity changes over an image and is frequently used to detect textural features. It does this by calculating the LBP value of a pixel by comparing the intensity of its N neighbours with itself. The intensity value is obtained from the gray-scaled pixel. From this we obtain a bit string B for pixel P, where the binary value at the i-th position is:

$$B_i = \begin{cases} 0, & \text{if } intensity(P) < intensity(N_i) \\ 1, & \text{otherwise} \end{cases}$$

The neighbourhood does not have to be adjacent and its size is variable, thus the parameters for LBP are:

- P points in a circular symmetric neighbourhood around the current pixel.
- Radius R, that denotes the distance at which to sample from.

We will use LBP uniformity to obtain our desired histogram image descriptor. We consider an LBP value to be uniform if the bit string contains at most two 0-1 or 1-0 transitions. From this we can calculate a histogram where each bin is one bit string that is uniform, which will be $P + 1$ uniform patterns. The last bin that is added is all nonuniform values, thus our final image descriptor for LBP is a histogram of size $P + 2$. The advantages of uniformity are that the number of bins in the histogram is significantly reduced, $P + 2$ over 2^P, and that it is rotation invariant.
For the segmented image, we will use the generated masks to discard pixels that are not part of the ROI.

4.3.2 RGB Histograms

Color histograms are commonly used as image descriptor due to their fast calculation. We will use a simple color histogram to describe the RGB color distribution of an image. To generalize the distribution, we will use a bin size of N. Generated masks by the segmenting stage are used to discard pixels outside of the ROIs.

4.3.3 COLTEX

Instead of the traditional RGB values, the Hue, Saturation and Intensity (HSI or HSV) color space is also commonly used to represent the colors of an image. One of the advantages over RGB is that the intensity is separated from color information. COLTEX[31] is an extraction method that uses the hue and intensity information together to capture the color distribution and texture content. Between hue and intensity, the dominant component can be determined using a hard threshold.[11] in saturation. COLTEX then calculates the weights of the true color (hue dominance) and true gray (intensity dominance) to capture the distribution in relation to the current pixel. These relations are stored in three NxN matrices:

- Diagonal, pixels to the bottom right
- Vertical, pixels to the bottom
- Horizontal, pixels to the right

where $N = N_1 + N_2$ and:

$$N_1 = \frac{2\pi}{Q_H} + 1 \tag{1}$$

$$N_2 = \frac{255}{Q_I} + 1 \tag{2}$$

The parameters for COLTEX are Q_H and Q_I, the quantization levels of hue and intensity, respectively. Our final image descriptor is the flattened diagonal, vertical and horizontal matrices of size $3N^2$. For the segmented image we use the bounding box to crop the object to obtain a MxN sub image, which will be fed to the COLTEX algorithm.

4.3.4 Histograms of Oriented Gradients

To detect shape features we will use HOG[12]. HOG features are able to detect the structure in an image by extracting the gradient and orientation of each edge.

The image is first divided into cells, a region of NxN pixels. For each ell, the gradients of each pixel is calculated in the horizontal and vertical direction, resulting in two NxN sized matrices. Using these matrices, we can calculate the magnitude and orientation of each pixel. Local histograms are then generated from the magnitude and orientation for each cell. Since each cell is evenly sized, the input image size must be divisible by NxN. To do this, the image is first resized to MxM where $N|M$.

For better invariance in lighting, the orientation histograms of each cell are concatenated with the surrounding cell histograms into blocks and then normalized using the root of sum squares, resulting in the normalized histogram for a block of cells.

All blocks are then concatenated, which produces the final HOG image descriptor. The input parameters for the HOG algorithm are K the number of bins of the orientation histogram, the number of pixels NxN per cell and cells HxH per block.

Lastly, for the segmented image, we will extract the ROI from the input image using the masks obtained from the segmentation step. This provides us with a list of segmented object images.

4.3.5 Scale Invariant Feature Transforms

Lastly, SIFT[13] is used to extract distinct keypoints from an image that are scale and rotation invariant. There are four major stages:

1. **Scale space extrema detection**. Using different scales or octaves of the input image and Gaussian blurring to find potential keypoints. In each octave the image is increasingly blurred. Each adjacent image is then subtracted to generate the Difference of Gaussians (DOG). To obtain a potential keypoint, each pixel in the image is compared to its 8 neighbours and to the 9 pixels in the next and previous scale. If the pixel is a local extrema then it is a keypoint. Because differing scale levels are used in the detection step, the resulting keypoints are scale invariant.

2. **Keypoint Localization**, by eliminating keypoints that have low contrast or lie along an edge, we can reduce the large amount of keypoints produced in step 1.

3. **Orientation assignment**. Rotation invariance is achieved by assigning a consistent orientation to each keypoint, to which the descriptor can be represented relative to this orientation. For each keypoint, the gradient magnitude and orientation is computed. These are then collected into an orientation histogram and the peaks within 80% of the highest peak are selected to create a keypoint with that orientation. Thus we end with multiple orientations for the same location and scale.

4. **Descriptor generation** is done from the obtained location, scale and orientation of each keypoint. A 16x16 window around the keypoint is sampled around the keypoint and divided into 4x4 subregions. Then 8 bin orientation histograms are created for each 4x4 subregion. These orientation histograms are then concatenated to obtain the final $4x4x8 = 128$ element feature vector for each keypoint.
   Lastly to reduce the effects of illumination, the values in the descriptor are thresholded to be no larger than 0.2, after which the resulting normalized feature vector is our image descriptor for a keypoint.

For the feature extraction of the segmented image, we extract the ROI using the segmentation masks and run the SIFT algorithm on that extracted image.
Since an image can have an arbitrary number of keypoints, it makes direct comparisons between two SIFT descriptors not as direct as the other descriptors. This will be discussed in the next section.

4.4 Feature Comparison and Ranking

To determine for images X, Y, Z if X is more similar to Y than Z we will compare the image descriptor of X against both Y and Z. Since LBP, RGB histograms, COLTEX and HOG image descriptors are a single $1D$ feature vector, we the Euclidean Distance distance function to assign a score to an image in relation to X. For two feature vectors P and Q of size N:

$$d(P, Q) = \sqrt{\sum_{i=0}^{N} (Q_i - P_i)^2} \tag{3}$$

Then if $d(X, Y) < d(X, Z)$, we can conclude that image Y is more similar to X than Z. In this research we extract features from the unsegmented images to index our feature database. Then we compare the image descriptors of both systems with the image descriptors in the database to obtain the distance score for each image in the database.
Unfortunately, for SIFT, the shape of the image descriptor is $Nx128$, where N is the number of keypoints in the image. Since N is variable, we cannot directly compare two SIFT descriptors using the distance function above.
Instead, we compare each keypoint, a $1D$ vector, in X against each keypoint in Y and Z, using the distance function above. To determine if a match between two keypoints is good, we will apply the ratio test to the match. If the distance ratio is less than the threshold, we will accept the match. Lowe[13] suggests using a threshold of 0.75. Using the ratio test, we can count the number of correct matches between X and Y and between X and Z.
This is then the final score for each image in relation to X. We can then conclude that X is more similar to Y than Z when $|matches(X, Z)| < |matches(X, Y)|$.

In all cases, each image in the database is assigned a distance score, where lower scores denote a higher similarity to the query image. The image database is then sorted in ascending order to obtain our ranked results. Figure 7 shows an example of retrieved results using the LBP image descriptor. We retrieve a subset of the image database to evaluate the performance of the system.

4.5 Dataset

Many image datasets are available for both segmentation and retrieval tasks. For this research, we wanted to measure the performance of the image descriptors on a wide variety of images. These had to contain multiple different objects of varying sizes, shape and colors. Datasets such as COREL dataset contains 10,800 images of 80 different classes or COIL100, which contains 100 different classes with varying angles, were considered. However, we have ultimately chosen for the Microsoft Common Objects in Context (MS COCO) dataset[21]. The MS COCO dataset consist of over 200,000 annotated images, with 80 different object classes. MS COCO provides an annotation file that provides the labels of each object in an image, provides the ROI and corresponding bounding box coordinates and easily allows for the cross referencing of classes. MS COCO also is used in a wide range of computer vision tasks, such as object detection, keypoint detection, image segmentation and retrieval tasks.
Lastly, although the segmentation algorithm, Mask R-CNN, that we use in the segmentation step can segment arbitrary images, because Mask R-CNN is trained on this dataset, we can limit the noise that poor segmentation can produce, which will lead to a more pure comparison between unsegmented and segmented images.

4.6 Performance Evaluation

For a query image, we will do two passes, unsegmented and segmented, through the image retrieval pipeline as shown in Figure 1, where the unsegmented image will skip the segmentation step. Each pass will result in a list of N images sorted by relevance to the query image.
We will use binary classification in our evaluation of the retrieved images. If the retrieved image

contains a class that is also in the query class, then we consider the retrieved image to be correct. Let C_i be the set of classes for image i. For query image P and retrieved image Q, Q is relevant if:

$$|C_P \cap C_Q| > 0 \tag{4}$$

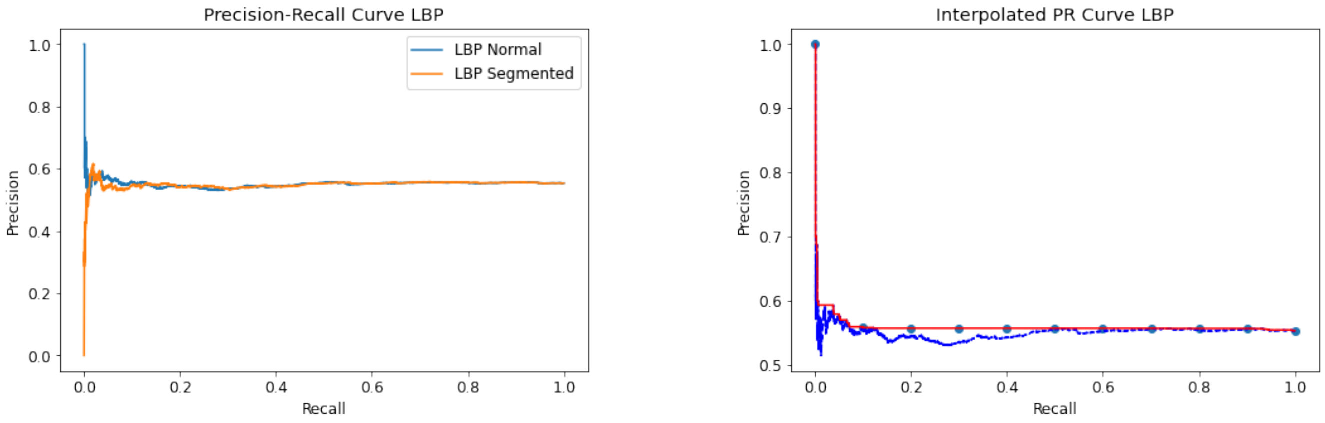
Using this classification, we can calculate the TP and $FP = N - TP$. To determine if the segmentation improves the image retrieval performance for a feature extraction method, we calculate the precision and recall for both segmented and unsegmented passes. The precision is calculated as follows:

$$P = \frac{TP}{TP + FP} \tag{5}$$

And recall:

$$R = \frac{TP}{N} \tag{6}$$

For each retrieved image, we calculate the precision and recall until we retrieve N images. Since the precision tends to decrease as recall increases, we can represent the precision as a function of recall. This can then be plotted to obtain a Precision-Recall curve as seen in Figure 8.
Through interpolation, we can calculate the precision at an 11-step recall interval, ranging from 0.0 to 1.0. The interpolation is done by searching for the highest precision to the right of the recall interval. Figure 5b shows the interpolated PR curve for one pass.



(a) PR curve for LBP.

(b) Interpolation of the PR curve

Figure 5: Example of the Precision-Recall curve for LBP and its interpolated curve.

To directly compare the unsegmented and segmented retrieval results, we calculate the average precision (AP) for a system by first calculating the precision and recall for each retrieved image, as Figure 6 illustrates. For the first N retrieved images and m relevant images, the AP is then calculated as follows.

$$AP = \frac{1}{m} \sum_{i=1}^{N} P(k) xrel(k) \tag{7}$$

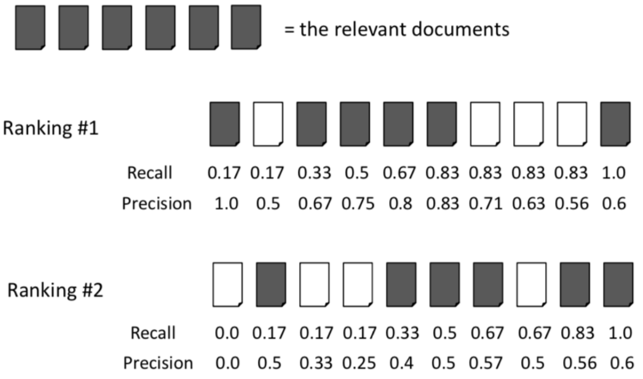

= the relevant documents
Ranking #1
Recall 0.17 0.17 0.33 0.5 0.67 0.83 0.83 0.83 0.83 1.0
Precision 1.0 0.5 0.67 0.75 0.8 0.83 0.71 0.63 0.56 0.6
Ranking #2
Recall 0.0 0.17 0.17 0.17 0.33 0.5 0.67 0.67 0.83 1.0
Precision 0.0 0.5 0.33 0.25 0.4 0.5 0.57 0.5 0.56 0.6

Figure 6: Example of precision and recall calculation per retrieved image. To obtain the average precision for a query, we calculate the mean of precision where a relevant image is returned.

where $P(k)$ is the precision value at the k^{th} retrieved image and

$$rel(k) = \begin{cases} 1, & \text{if } k^{th} \text{image is relevant (see eq 4)} \\ 0, & \text{otherwise} \end{cases}$$

The average of APs over Q queries is called the mAP, a score between 0.0 and 1.0, and it gives us a single value to compare an unsegmented system with a segmented system, where a score of 1.0 denotes a system that only retrieves relevant images.

4.7 Statistical Analysis

For the unsegmented system U and segmented system S, in order to show that an improvement of S over U is significant, we will use the statistical significance test.
Our null hypothesis (H_0, in all feature comparisons, is that U and S are equal in terms of retrieval performance. Since we calculated the average precision for each step until N, we can compare the single valued AP of U and S at each retrieval step i and assign a sign to it:

$$Sign_i = \begin{cases} +, & \text{if } mAP(U_i) < mAP(S_i) \\ -, & \text{otherwise} \end{cases}$$

Then our finalized null hypothesis will be:

$$H_0 : Pr(+) = Pr(-) = 0.5 \tag{8}$$

Under the assumption that the null hypothesis is true, there is a probability of 0.5 that the sign is either + or −. Thus, we use a binomial probability with a $p = 0.5$:

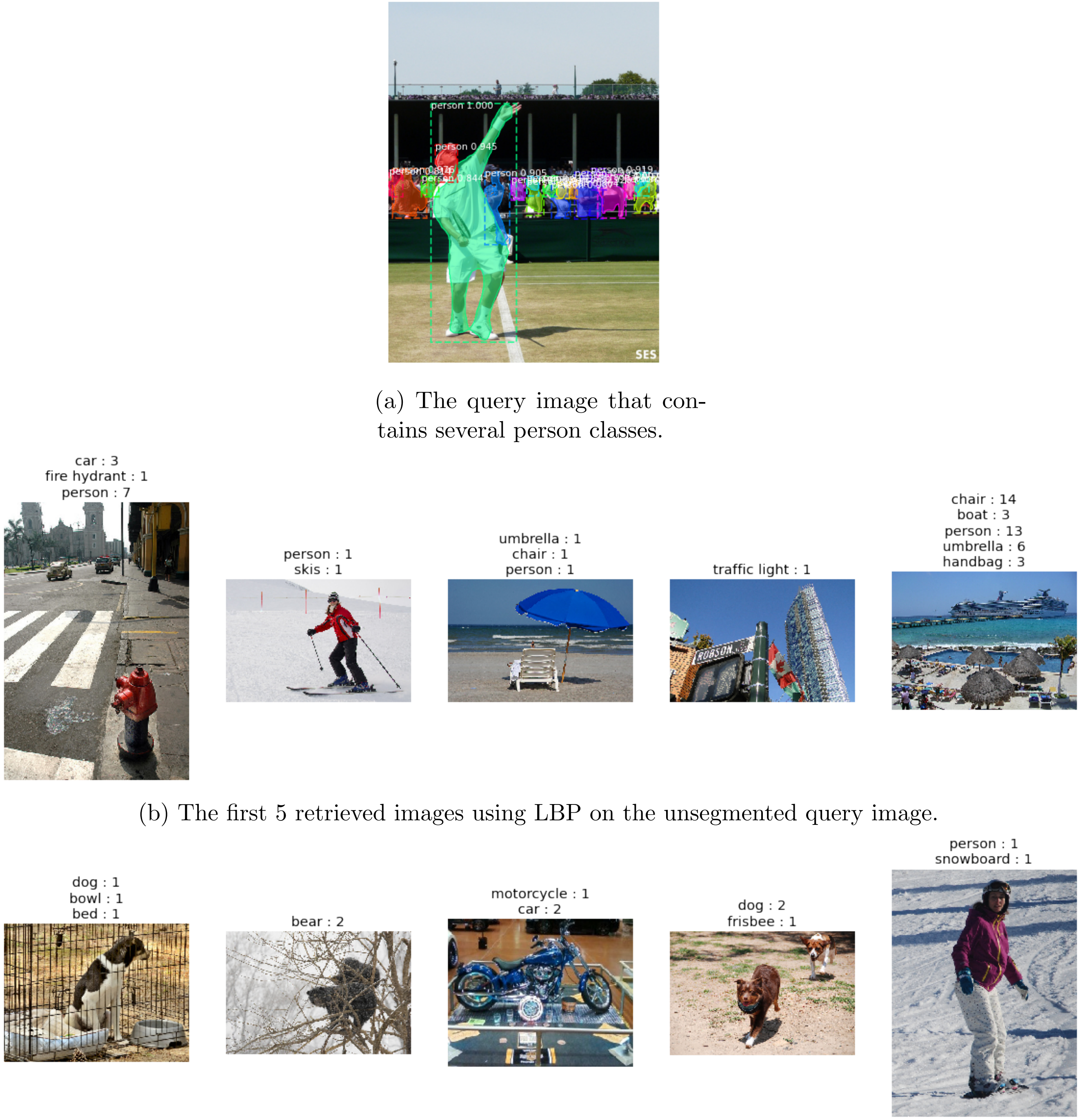
$$Pr(x, p, n) = \frac{n!}{x!(n-x)!}(p)^x(1-p)^{n-x} \tag{9}$$

$$Pr(x, n) = \frac{n!}{x!(n-x)!}(\frac{1}{2})^n \tag{10}$$

We want to calculate the probability of observing i or more positives. Thus, our final probability function is :

$$Pr(x,i) = \sum_{n=i}^{x} \frac{n!}{x!(n-x)!}(\frac{1}{2})^n \tag{11}$$

Where i is the number of positives and x the total number of signs.
If the probability is lower than our significance threshold or alpha value, then we can reject H_0 and conclude that the improvement is indeed significant. For our alpha value, we will choose a value of 0.5.



(a) The query image that contains several person classes.

(b) The first 5 retrieved images using LBP on the unsegmented query image.

(c) The first 5 retrieved images using LBP on the segmented query and combining the distances of each segment.

Figure 7: Example of retrieved results using LBP. The labels above the images show the classes and occurrences in the image.

5 Experiments

In this chapter, we will give the parameters used in each image descriptor, as discussed in the previous chapter. As explained in chapter 4.6, we will do multiple passes through the unsegmented and segmented systems, using arbitrary images from the MS COCO validation dataset. It consists of 5000 images and contains 80 classes. 200 queries were done on both systems and each pass returned the first 1000 images from the ranked image list.

5.1 Local Binary Patterns

For LBP, we use a 24 sampling points in a radius of 3 to calculate the LBP of our image. From this we obtain a 26 bin histogram.
In the segmented system, we isolate the objects using the ROI masks obtained and feed the individual sub images to LBP. The images are also gray scaled, since LBP only calculates intensity differences. Lastly, Euclidean distance was used as the distance metric to rank the image database, as described in section4.4.

5.2 RGB Histograms

We choose a bin size of 15 for the number of bins of the RGB histogram. This gives us a 45 element feature vector, 15 for each color channel. The segmented system will include the masks in the histogram calculation. This will discard pixels that fall outside of the masks. The 45 element feature vector is compared against the image database using Euclidean distance to retrieve a ranked list of images.

5.3 COLTEX

As discussed in section 4.3.3, the input parameters for COLTEX are the quantizations of the hue and intensity. The COLTEX paper suggests that a quantization hue of 4 and quantization intensity of 4 produces better early performance then other suggested levels. Thus our inputs are also 4 and 4 for the hue and intensity, respectively. Using function 1 and 2, the COLTEX algorithm gives us a feature vector of size 13872. Euclidean distance is used to score each image by comparing the 13872 element feature vectors.

5.4 Histogram of Oriented Gradients

Since HOG uses uniform sized blocks to generate the orientation histograms and our dataset contains different sized images, we resize each image to $256x256$, use a cell size of $16x16$ and a block size of $2x2$. For an image of size $256x256$, 255 blocks of size $16x16$ will fill the image. With a orientation histogram size of 9, each $16x16$ block contains a $4x9 = 36$ element feature vector, thus our final HOG is a 8100 element feature vector. This vector is compared to the HOG vectors in the image database using Euclidean distance.

5.5 Scale Invariant Feature Tranforms

The last image descriptor has n number of layers per octave used in the keypoint detection step. The SIFT paper suggest using a layer number of 3, which is shown to obtain the highest repeatability, thus we will also use 3 as our number of layers.
We use the modified distance function for SIFT to retrieve similar images as described in section 4.4. Our threshold to determine whether a match is rejected will be 0.75.

5.6 Precision-Recall Curves

We averaged the PR curves for 200 queries for each system, as shown in Figure 8 and compared the precision value for each recall level in table 1. For the COLTEX image descriptor, we see a clear domination of the unsegmented system. This indicates that segmentation, in the case of COLTEX, does not improve the retrieval performance and instead is detrimental to it. In all image descriptors, the unsegmented system has a better retrieval performance at recall 0.0, which suggests more relevant images were found at higher ranks than the segmented system. However in all image descriptors, excluding COLTEX, we observe that there is a definitive threshold where the segmented system outperforms its unsegmented counterpart. For RGB, HOG and SIFT, this is at a very early recall of 0.1, after which there is a parallel-like descent difference between both systems. Overall, unsegmented LBP consistently has the highest precision across all recall levels, compared to other image descriptors.

	LBP		**RGB**		**COLTEX**		**HOG**		**SIFT**	
Recall	U	S	U	S	U	S	U	S	U	S
0.0	0.657	0.601	0.583	0.582	0.608	0.584	0.488	0.427	0.594	0.585
0.1	0.431	0.412	0.394	0.403	0.389	0.365	0.329	0.336	0.373	0.387
0.2	0.406	0.396	0.380	0.392	0.372	0.357	0.326	0.331	0.365	0.378
0.3	0.392	0.389	0.374	0.386	0.365	0.352	0.324	0.329	0.362	0.375
0.4	0.384	0.384	0.370	0.381	0.360	0.346	0.323	0.327	0.360	0.373
0.5	0.379	0.380	0.368	0.378	0.357	0.343	0.321	0.326	0.359	0.372
0.6	0.375	0.376	0.365	0.375	0.354	0.340	0.320	0.325	0.358	0.371
0.7	0.371	0.373	0.363	0.373	0.351	0.338	0.320	0.324	0.357	0.370
0.8	0.367	0.369	0.360	0.370	0.349	0.335	0.319	0.323	0.356	0.369
0.9	0.364	0.366	0.358	0.368	0.347	0.334	0.318	0.322	0.355	0.368
1.0	0.361	0.363	0.355	0.364	0.344	0.331	0.318	0.322	0.353	0.365

Table 1: Precision values of each image descriptor at an 11-step Recall interval. These are the mean interpolated precision values over 200 queries with 1000 retrieved images per query. A higher value denotes a higher retrieval accuracy at a particular Recall level. The colored cells indicate the Recall threshold at which the segmented system performs better than the unsegmented system.

5.7 Mean Average Precision

Finally, we calculate the AP for each pass and average the score to obtain our final single-valued mAP score for an image descriptor. Table 2 shows these scores for both the segmented and unsegmented system. We observe from these results that three out of five image descriptor show an improvement using segmentation. The difference in most cases is small, to the third decimal place, with the only exception being SIFT, at a mAP of 37.6%, which showed a positive difference of 1.3%. Overall, based on the mAP score, unsegmented LBP had the best retrieval performance of 39.1%.

Image Descriptor	Unsegmented mAP	Segmented mAP	Difference
LBP	0.391	0.387	0.004
RGB	0.354	0.363	+0.009
COLTEX	0.365	0.350	-0.015
HOG	0.324	0.329	+0.005
SIFT	0.363	0.376	+0.013

Table 2: Mean mAP scores over 200 queries

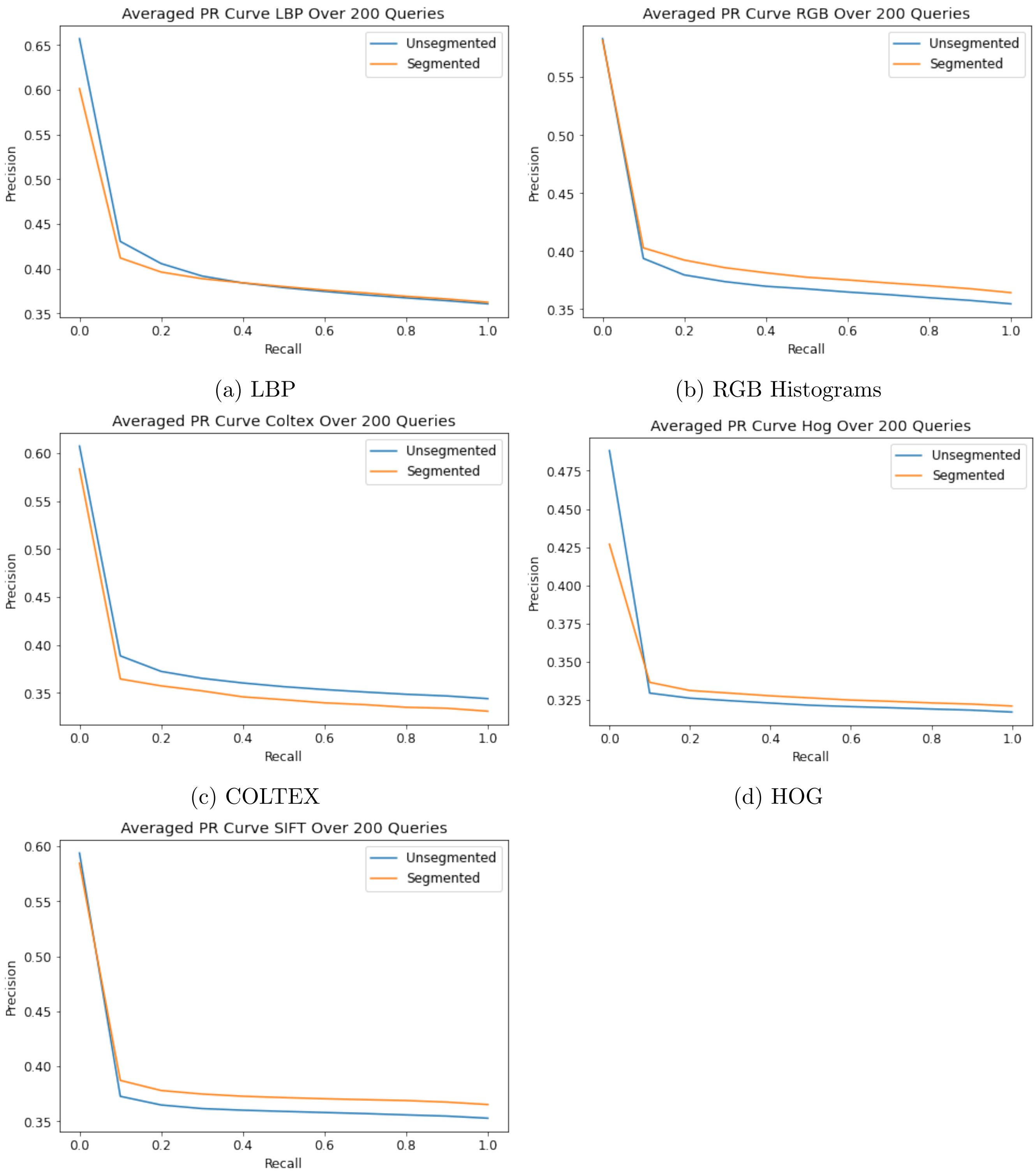
5.8 Significance

We will briefly report the significance result for each image descriptor where the mAP of the segmented system is higher than the unsegmented system.. Our null hypothesis for each image descriptor is that both systems are equivalent, as discussed in section 4.7 and we will reject this hypothesis if the p-value is less than the alpha-level of 0.05. Table 3 shows the number of queries where the segmented system achieved a higher AP score than the unsegmented system. These values, with the sample size of 200, are used to calculate the p-values.

Image Descriptor	Number of +
LBP	88
RGB	111
COLTEX	74
HOG	97
SIFT	131

Table 3: Sign Test results of each image descriptor. The results show the number of queries where the mAP of the segmented system was higher. All image descriptor systems were run with 200 queries.

- RGB histograms showed a difference of +0.009 and a p-value of 0.068. We cannot refute the hypothesis and therefore a +0.009 increase is not significant.
- HOG showed a +0.005 mAP difference and had a p-value of 0.689. Thus, the segmented system is not significantly better than the unsegmented system.
- SIFT had a difference of +0.013 and a p-value of 0.000006. We can then conclude that the segmenting is significantly better than not.



(a) LBP

(b) RGB Histograms

(c) COLTEX

(d) HOG

(e) SIFT

Figure 8: Averaged PR curves for each feature extraction method over 200 queries.

6 Discussion

Our aim in this research was to find if segmentation overall improves image retrieval by using low level image descriptors, such as color, shape, and texture. We used mAP scores for the five image descriptors to show that segmentation for LBP and COLTEX were outright a detriment to the retrieval performance. RGB, HOG and SIFT, on the other hand, showed improvements, however during the significance test, we found that the improvements of the RGB and HOG segmented system were insignificant. The SIFT image descriptor is the only feature that significantly benefits from a segmentation step in the retrieval pipeline.

We observed in all cases, as table 1 shows, that, between recall 0.0 and 0.1, the unsegmented system always outperforms the segmented system, which indicates that more relevant images were retrieved at higher ranks. Especially in the case of LBP and HOG, we see that the unsegmented system performed 5.6% and 6.1% better at recall 0.0. After that, from recall 0.1 to 1.0, we observe that both systems follow each other in parallel. The exception to this trend is LBP, where the segmented system starts to outperform at recall 0.5.

The results also show that segmentation improves the retrieval performance for non-textural features when the recall is higher than 10%. This means that the segmented systems tend to retrieve more relevant images at later ranks. LBP at recall 0.5 to 1.0 seems to perform equally, with a difference of at most 2%. SIFT, out of all tested image descriptors, benefits the most from segmentation, with a 3.5% increase in mAP score and it consistently has a better precision at recall 0.1 to 1.0. In terms of overall performance, table 2 shows that unsegmented LBP has the best average retrieval performance on the MS COCO dataset, with a mAP of 39.1%. The segmented system that has the highest retrieval accuracy is LBP, with a mAP of 38.7%.

One of the potential benefits of segmentation is the ability to remove unwanted information before feature extraction. However, it may also produce some invariance, where depending on the feature, the extracted feature does not generalize well enough to describe a general image. Segments could also be too small to extract features from. In the case of LBP and COLTEX, both are not scale invariant and perform poorly when textures are scaled differently, whereas RGB, HOG and SIFT are scale invariant and, as we have showed, perform better when segmentation is used. Thus we conclude that features, such as RGB, HOG and SIFT, that are more invariant to transform changes are better suited for segmentation.

Some limitations of this research are the use of a diverse image dataset. Since MS COCO contains a wide variety of object classes in varying shapes, sizes and textures, some image descriptor are not suitable for such a task. The ranking score could be negatively impacted by the inclusion of objects that are too small to extract any meaningful features from and thus, more incorrect images could be returned. This skews the score since the objects are evenly weighted. In this case, it was the textural features that were most affected by the lack of similar textured images. Another limitation was that it was not always possible to fully discard noisy information from the feature extraction method. This was true in the case of COLTEX, where we input the bounding box sub-image into the algorithm. In spite of these limitations, we have removed as many factors as possible to conclude that the segmentation step in the image retrieval pipeline does not improve the retrieval performance of low level features. Further research could be done on the types of image descriptor. Such as if there is a difference between local and global image descriptor when using segmentation. This could show that features like SIFT perform better in all circumstances using segmentation than global features like LBP. The comparison between unsegmented and segmented retrievals using an unsegmented and segmented database, respectively, can also be further researched. Since this research only focused on the retrieval performance of a feature database that is indexed using unsegmented features, further research could index the database using segmented based features and the performance against our results.

7 Conclusion

This research aimed to show whether segmentation in a CBIR system would improve retrieval performance using low level image descriptor. The results showed that based on the mAP score, segmentation does not improve retrieval accuracy of LBP and COLTEX, where they had a mAP of 39.1% and 36.5% for the unsegmented system, compared to 38.7% and 35.0% for the segmented one. SIFT showed the most improvement using segmentation, with a mAP of 37.6%, over the unsegmented mAP of 36.3%. This was a significant increase of 3.5%. We concluded that segmentation benefits invariant features and decreases performance of features that are not.

References

[1] Sanjay Patil and Sanjay N. Talbar. Content based image retrieval using various distance metrics. In Rajkumar Kannan and Frédéric Andrès, editors, *Data Engineering and Management - Second International Conference, ICDEM 2010, Tiruchirappalli, India, July 29-31, 2010. Revised Selected Papers*, volume 6411 of *Lecture Notes in Computer Science*, pages 154–161. Springer, 2010.

[2] Koji Wakimoto, Mitsuhide Shima, Satoshi Tanaka, and Akira Maeda. Content-based retrieval applied to drawing-image databases. In Carlton W. Niblack, editor, *Storage and Retrieval for Image and Video Databases, San Jose, CA, USA, January 31 - February 5, 1993*, volume 1908 of *SPIE Proceedings*, pages 74–84. SPIE, 1993.

[3] Yikun Yang, Shengjie Jiao, Jinrong He, Bisheng Xia, Jiabo Li, and Ru Xiao. Image retrieval via learning content-based deep quality model towards big data. *Future Gener. Comput. Syst.*, 112:243–249, 2020.

[4] Song Yuheng and Yan Hao. Image segmentation algorithms overview. *CoRR*, abs/1707.02051, 2017.

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[6] Kai Kang and Xiaogang Wang. Fully convolutional neural networks for crowd segmentation. *CoRR*, abs/1411.4464, 2014.

[7] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk, and Quoc V. Le. Rethinking pre-training and self-training. *CoRR*, abs/2006.06882, 2020.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask r-cnn, 2018.

[10] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[11] Shamik Sural, Gang Qian, and S. Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. volume 2, pages II–589, 02 2002.

[12] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893. IEEE Computer Society, 2005.

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

[14] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S. Lew. A review of semantic segmentation using deep neural networks. *Int. J. Multim. Inf. Retr.*, 7(2):87–93, 2018.

[15] P.S. Williams and M.D. Alder. Segmentation of natural images for cbir. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, volume 1, pages 468–470 vol.1, 1998.

[16] Bhogeswar Borah and Dhruba K. Bhattacharyya. Image retrieval by content using segmentation approach. In Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas, editors, *Pattern Recognition and Machine Intelligence, First International Conference, PReMI 2005, Kolkata, India, December 20-22, 2005, Proceedings*, volume 3776 of *Lecture Notes in Computer Science*, pages 551–556. Springer, 2005.

[17] George Paschos, Ivan Radev, and Nagarajan Prabakar. Image content-based retrieval using chromaticity moments. *IEEE Trans. Knowl. Data Eng.*, 15(5):1069–1072, 2003.

[18] E. R. Vimina and K. Poulose Jacob. Image retrieval using low level features of object regions with application to partially occluded images. In Luis Alvarez, Marta Mejail, Luis Gomez, and Julio Jacobo, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 422–429, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[19] Duraisamy Yuvaraj and Shanmugasundaram Hariharan. Content-based image retrieval based on integrating region segmentation and colour histogram. *Int. Arab J. Inf. Technol.*, 13(1A):203–207, 2016.

[20] M. Everingham, L. Gool, Christopher K. I. Williams, J. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2009.

[21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[22] Sudipan Saha, Swathikiran Sudhakaran, Biplab Banerjee, and Sumedh Pendurkar. Semantic guided deep unsupervised image segmentation. In Elisa Ricci, Samuel Rota Bulò, Cees Snoek, Oswald Lanz, Stefano Messelodi, and Nicu Sebe, editors, *Image Analysis and Processing – ICIAP 2019*, pages 499–510, Cham, 2019. Springer International Publishing.

[23] Qin Huang, Chunyang Xia, Siyang Li, Ye Wang, Yuhang Song, and C.-C. Jay Kuo. Unsupervised clustering guided semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1489–1498, 2018.

[24] Olga Russakovsky, Amy L. Bearman, Vittorio Ferrari, and Fei-Fei Li. What's the point: Semantic segmentation with point supervision. *CoRR*, abs/1506.02106, 2015.

[25] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[26] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[28] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[30] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[31] Vadivel Ayyasamy, S. Sural, and Arun Majumdar. Color-texture feature extraction using soft decision from the hsv color space. pages 161 – 164, 11 2004.