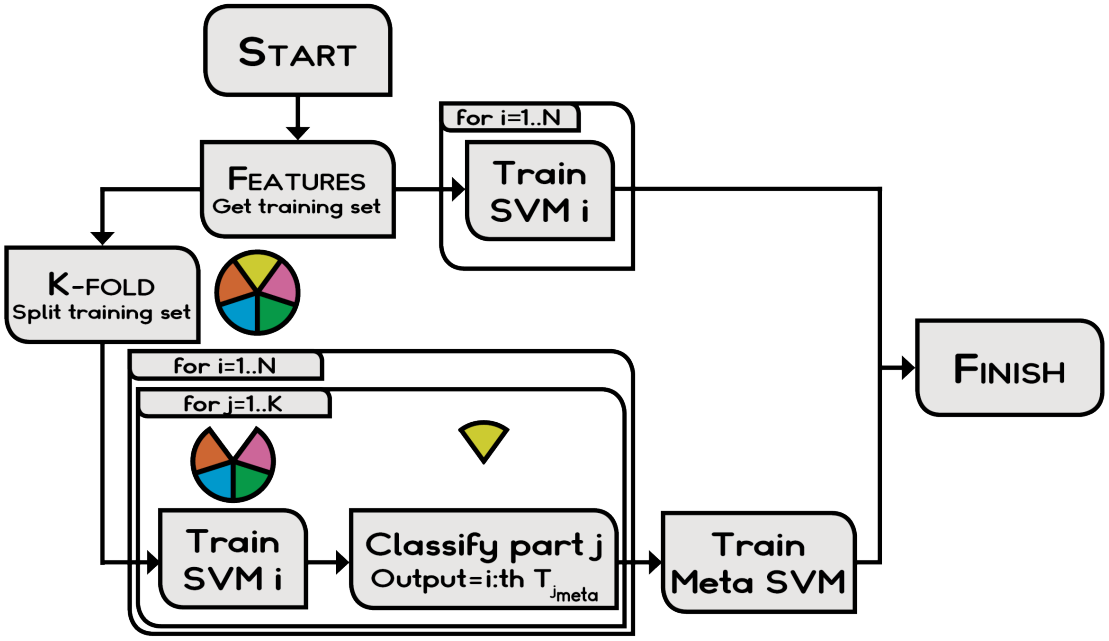
CHALMERS
UNIVERSITY OF TECHNOLOGY



Iterative interactive concept training on visual content

Master's thesis in Computer Science: Algorithms, Languages & Logic
Complex Adaptive Systems

GABRIEL ANDERSSON, MATS UDDGÅRD

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis 2017

Iterative interactive concept training on visual content

GABRIEL ANDERSSON
MATS UDDGÅRD

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Chalmers University of Technology
Gothenburg, Sweden 2017

Iterative interactive concept training on visual content

Supervisor: Josef Eklann, Safer Society Group
Examiner: Fredrik Kahl, Department of Signals and Systems

Master's thesis 2017
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Abstract

This thesis presents a novel method to quickly sift through the visual content (image material) of a database in order to retrieve as much relevant material as possible. The proposed model uses a combination of classification systems, image retrieval and relevance feedback. Five different feature descriptors, known to be useful within image retrieval, are extracted to later be inserted into a classification system. The material is presented to, and corrected by, a user and can therefore be used as training data in future iterations. The training data is inserted to a supervised learning classifier in order to search through the database. The most relevant material is passed through the feedback loop allowing the model to learn concepts in a fast manner.

The five feature descriptors that are commonly used within the field are the following: *histograms of oriented gradients*, *global color histograms*, *Haar wavelet transformations*, edge detections using a *Sobel filter* and the final activations of a *VGG-16* neural network.

In the classification system a classifier called *Deep SVM* (*Deep Support vector machine*) is used. In the proposed model it consists of 6 SVMs in order to create an ensemble, where one is used for each kind of feature descriptor and the last SVM is used to combine the result of the first order classifiers. Material in the search space is passed through the system and the most relevant material is presented to an expert user.

Evaluations and measurements were performed on the model in two settings. Firstly as a parameter benchmark in order to find the most appropriate setting for the intended use of retrieving all the relevant visual material in a crime investigation case. Secondly as an image retrieval comparison with other studies by using a small training set as query material. The parameter benchmark shows that the model is capable of retrieving the majority of relevant material within a small number of iterations. The study comparision shows that even though the model is designed to have sets of images as query data, the size of the sets does not have to be greater than 10 in order to outperform the related approaches.

The contributions of the thesis consist of the following: Using a Deep SVM in combination with relevance feedback to perform an image retrieval results in great performance and a complete retrieval within a low number of relevance feedback iterations. Content-based image retrieval has previously been performed with one image as query material while this thesis presents a method of using a set of images for the task in order to achieve a higher abstraction level.

Keywords: *Machine learning, Ensemble learning, Image analysis, Content-based image retrieval, Relevance feedback, Semantic gap, Feature extraction, Neural network, Support vector machine, Deep SVM*

Acknowledgements

Firstly, we want to reach out our uttermost thanks to our host company *Safer Society Group* for contributing with an office location where we could work and write a report for the entirety of the thesis.
Secondly a thanks to our supervisor, *Josef Eklann*. Thank you for providing with knowledge and bringing support.
Finally, a thanks directed to our examinator, *Fredrik Kahl*. This would not had been possible without your help.

Gabriel Anderson
Mats Uddgård

Gothenburg, May 26, 2017

Contents

1 Introduction

Generella kommentarer från examinatorn. Kolla innan vi skicka vidare rapporten

Digital video and image files are normally important evidence in criminal investigations, and the amounts of images and videos that constitute the evidence increases more now than ever. The computer forensics community has ever increasing problems with this continued growth of information and in investigations the amount of data to be examined can in the very least be huge [1][2][3]. In order to effectively help the investigators in their tasks of organizing and prioritizing evidence, methods to scrutinize the data in an efficacious manner is vital. In investigations pertaining digital information the need of a quick and effective way to handle large quantities of material is of importance as the evidence can be of abundance while the relevant part can be a trifle. Methods focused on grouping material by some collective attributes are often found to be efficient. There are several of these attributes that can be correlated to the in visual content, such as in images and video. Since which images that are relevant might differ from case to case, it would be a good praxis if the investigators could define their own respective grouping setting for each separate case. By letting the investigator define some form of concept by directly specify image examples as relevant and non-relevant an algorithm can be trained to recognize a concept queried by an investigator.

Ibland använder ni kursiverad font för "SVM", ibland inte. Var konsistenta - det inkluderar även andra förkortningar: CBIR, CNN, etc.

Något man saknar är "new contributions". Det vill man gärna ha reda på ganska tidigt, och hur det står sig relativt litteraturen. Det står lite om det i kapitel 5, men så länge vill man inte vänta som läsare. Ofta säger man något om det redan i abstract, och definitivt i introduction.

"eventhough" ska vara "even though". Likaså "atleast" -> "at least". (De finns på flera ställen).

long sentence

missing word?

or removal of words needed?

1.1 Problem definition

More like solution definition. Agree, something need to be rephrased or changed

The goal is to develop an algorithm that iteratively suggests new relevant material in line with the desired concept each new iteration and that improves while doing so. To achieve this a method called *relevance feedback* is incorporated, where a user interacts with the algorithm by being presented a couple of images each iteration and labels these in accordance with their relevance against the sought after concept. By using relevance feedback in the learning process a direct correction of the *false positives* and *false negatives* will give an increase in learning rate considered against not having a user which checks and tunes the algorithm. Each iteration will thus be made up of a selection of parts which constitutes; searching the image database for relevant content based on what the algorithm has learned in its previous stages. Presenting the most relevant ones to a user that corrects any errors in labeling and applies the desired labeling for the images. These newly labeled and checked images are then used to re-train and update the algorithm in each iteration.

Use case. Under problem definition

1.2 Goals

The aim of this project is thus to:

- create an algorithm that helps identifying relevant images in a database which should be subjected to a user defined concept, a dynamic general concept search engine.
- exhaust the database of relevant images faster than an independent user or random search.
- put more emphasis on trying to minimize the number of false negatives, than the false positives, in the search as to lower the risk of neglecting images that would be of importance to the operator. False negatives being images that are classified as non-relevant while being relevant, and false positives being non-relevant images but predicted as relevant.

more items?

1.3 Delimitations

In the scope of this project choices were made to be able to propose a functioning model with some limitations given the time frame of the thesis. The aim is to create a dynamic general concept search engine with following delimitations.

- Some parameters of the model need to be chosen emperically since all settings are missing support from previous papers. However there are some choices that are made more elaborately, e.g. parameter benchmarks are performed in order to find the optimal setting.
- The classification method chosen will be a binary one since this will be enough in the scope of this project. The classifier will not be tested towards other methods of classifying.
- Only five different feature descriptors will be used, as the proof of concept of a general learner is central and not how the addition or omission of certain parts changes this function. By using five different feature descriptors makes it possible to get the distinction and variation sought for in a general sense.

1.4 Contributions

Contributions of thesis

Using a classification system as a generic image retrieval system.

CBIR using more than one image as query

1.5 Organization of thesis

Chapter 2 **Theory** presents the basic and central concepts in the scope of the thesis such as content-based image retrieval and relevance feedback. The material used in form of datasets and information of the images used as well as in which color ranges that are used in this thesis.

Chapter 3 **Image analysis theory** describes the different parts that are relevant in the field of image analysis. The background of feature and the feature descriptors.

Chapter 4 **Machine learning theory** describes the methods used for supervised learning and classification of the datasets.

Chapter 5 **Method** explains in deatail how the proposed model is structured as well as how evaluations were performed in order to test the model. The evaluations are parameter benchmarks and study comparisons with other content-based image retrieval models.

needs rework and overview

Chapter 6 **Results** presents the obtained results of evaluations described in the chapter 5.

Chapter 7 **Conclusion** discusses the the results presented in chapter 6. From these discussions possible extensions on the model are presented as well as how some functionality of the model can be extracted for external usage.

2
Theory

In this chapter the theory of the key concepts behind this thesis are presented. Here content-based image retrieval is explained as well as the difficulty of closing the semantic gap. The idea of relevance feedback is introduced as it is a key module in the thesis. Lastly the different datasets used in evaluations are presented accompanied with a brief introduction to how images are stored digitally. The chapters *Image analysis theory* and *Machine learning theory* extends theory in their respective branches.

2.1 Content-based image retrieval

Content-based image retrieval (CBIR) is a term referring to techniques used in computer vision, where the goal is to find images with similarities in large databases. Given a query that is presented to the system, the output would be a set of images extracted from the total set which have the highest resemblance to the query. Content-based refers to the information stored in the image itself and not, as is the case in tag-based image retrieval (TBIR), the metadata of the image. CBIR is therefore a viable approach if either the metadata is non-existent or the classification is of some other variety than what the metadata can give. A situation when CBIR might be good to use is when the visual content of the semantic nature or there are reoccurring objects in several different images. In modern CBIR systems there are four reoccurring major parts: Feature extraction, where the raw features are recovered. Feature reduction, the recovered features are used to reduce feature dimensionality and storage space usage. Ranking, systemise the images so that the system can categorize the images in the dataset depending on resemblance to the query. Finally relevance feedback, the final feedback given by an expert user if needed correcting the algorithms predictions [4]. A large problem in image retrieval is that the query image might hold information easily perceived by a user but hard to concretize in pixel data and features, this problem is called the semantic gap. It can be said to be the difference that arises when two different linguistic representations try to describe the same object. This is a relevant issue whenever the perspective of a human is tried to be represented by a computer. The semantic gap is mounted by some function that can from the pixels representing an image make a computer understand and recognize what object that it is percieving [5].

2.2 Relevance feedback

A way to avoid the problems that arise when dealing with the semantic gap is to use relevance feedback. Relevance feedback can be said to be the direct interaction between a user and a machine in learning. The user reviews and corrects the predictions that the machine has made. The machine can in return use this information to re-evaluate the predictions that were made. The process is in general that a user

is presented with a number of images by a machine learning algorithm. Images that the algorithm has tried to label with the help, or occlusion, of some pre-training. These images are re-evaluated by the user and corrected by her if the corresponding label happens to be falsely assigned and acknowledged otherwise. With these adjustments to the data the to the data the retrieval process is refined in an attempt to make future classifications better. These two parts are then iteratively carried out as the algorithm keeps searching through the dataset for the required images [6]. There are different kinds of relevance feedback methods, the three most common are *explicit*, *implicit* and *blind (pseudo) feedback*. Using explicit feedback means that a user, knowingly of that her actions will affect how future material will be presented, indicates the relevance of the material presented to it. The first of the other two feedback variations is implicit which either means that the users behavior is observed or that the user is unknowing that the feedback are used as relevance feedback for the system. The other is pseudo relevance feedback which is a form of automated feedback that uses the first query as relevant results. In view of the situation of investigations having a need for the user to review all images in any case, explicit feedback is the best viable option. The explicit feedback is used to create a continuous data confirmation and thus creating more reliable data in each iteration.

2.3 Image formats

There are several file formats available and different ways to store images, such as storing images as as uncompressed and compressed raster formats as well as vector formats. When an image is stored with a raster format it is represented by a grid of pixels with a depth depending on the information of the image. The most common way to store image information is to use a 24-bit RGB pixel, where RBG is an abbrevation for red green and blue. Each 24-bit pixel has three equally sized channels of 8 bits, which makes each color channel range between 0 and 255. Resulting in approximately 16.8 million different combinations for colors, where a human can perceive about 10 million [7, p.388]. A computer screen usually operates at the described color setting. The size of the file simply depends on how many of these pixels that are stored, i.e. the size directly depends the dimensions (width and height) of the image. As the sizes may vary, some images can become expensive to process. Because of this it can be prudent to downsize the image to a smaller number of pixels and thus avoiding unnecessarily large amounts of data. Downsizing can be especially useful since high pixel information is not always equivalent to good performance [8]. There are several different color spaces and of which some are used in image analysis depending on the task. The most commonly known being RGB which is a color space that generated based on how colors are created based on mixing light. The use of additive color combinations of red green and blue became useful when the production television sets and computer screens rose. Though these three color channels are useful as lightsources, they are considered rather inept in image analysis. Humans tend to react more on the hue and saturation of an image than these color channels [9][10].

2.3.1 Hue, saturation, value

Hue, *saturation* and *value*, abbreviated as HSV, is a color space in line with RGB. The difference is that HSV is a cylindrical representation of RGB, where RGB is mapped as a cube where the channels r, g and b can be interpreted as the often named x, y and z axes. The HSV can be mapped cylindrically where hue is the degree position of the cylinder, saturation is the radius and value is the height. HSV is considered to be a color space that is a closer representation how a human perceives the world and thus also often used in the field of CBIR and image analysis as a whole.

2.3.2 YCbCr

YCbCr is a color space used mainly in color image pipeline for video and photography systems. The color space was invented during development of a digital video standard and set to lessen the total bandwidth required . This representation is composed to work towards human perception where the luminance component (Y) can be seen as analogous to the brightness or light component and the two chroma (Cb and Cr) filling out the color spectra [11]. The color information is not always as vital for human perception as the brightness is. The human retina has three types of photoreceptor cells where two are commonly referred to: The rods, that are very sensitive to light and can be triggered by a single proton, and the cones, that are less sensitive to light but reacts to differently to different wavelengths of light. A human has ≈ 120 million rods and ≈ 6 million cones. Then number of photoreceptor is somewhat of an indicator of which channel in YCbCr that affects human perception the most. The luminance is often used in edge detection since it conveys textures, illuminates the shapes of objects and portrays depth in images [12][13].

Sentence. Meaning?

required total bandwidth

One reason is that to have components that can described in some form of light, value (V) for HSV and luminance (Y) for YCbCr, component is separated from the color component(s) prominent in some color spaces, as the RGB color space where each channel have light built into it, which gives a more noisy interpretation of the light than if the channel were to be separated. :::: todo Was in Hsv... The entire sentence (starting at one reason): why is it here? Move to image format or remove.

2.4 Datasets

Datasets are used to evaluate and compare the performance of a proposed model with other studies. This is often done for CBIR systems as well as classification systems. Examples of evaluations that uses datasets are *plain recognition*, *image retrieval* or *image classification*. As a first means of both evaluation and with reference to several recent papers ([14], [15], [16] and [17]) focus one of the datasets used in the thesis is the dataset Corel-1000. This set comes with its limitations as it is relatively small in comparison with the huge datasets used to train deep neural networks. Neural networks such as *GoogLeNet* [18], *AlexNet* [19] and *VGG-16* [20] are designed to compete in the *ImageNet Large Scale Visual Recognition Competition* (ILSVRC) [21], a yearly object detection contest where 1000 object categories are present. Since the goal of the thesis is to learn general concepts and not to detect objects another dataset was used: The dataset Places205, designed by MIT, described in Section 2.4.2.

2.4.1 Corel-1000

The Corel set is an image dataset often cited and used in validation of different CBIR systems [22]. The dataset is a low resolution set composed of 80 classes (concepts) with 10.800 images in total. Due to the size of the dataset a subset, called the Corel-1000 dataset, is used to compare the proposed method with related CBIR approaches [14]. The Corel-1000 dataset is a subset of the Corel dataset which contains 1000 images, composed by 10 classes with 100 images in each class. The images in this dataset are of the sizes $64 * 96$ and $96 * 64$ pixels depending on their orientation. The 10 classes are referred to as *Africans*, *Beaches*, *Buildings*, *Buses*, *Dinosaurs*, *Elephants*, *Flowers*, *Horses*, *Mountains* and *Food*.

2.4.2 Places205

Places205 is a dataset produces by MIT and collaborators in the search for ever better human-reaching performance with machine-learning.
The MIT places205 is chosen to be part of the evaluation of the algorithm presented in this thesis since it has a large variety of images and classes and is a well-known dataset [23]. It is a repository of millions scene photographs, labeled with scene semantic categories and attributes. The dataset consists of 205 sceneries with an average of 12.000 images in each class. The different sceneries of the dataset places205 can be seen in Table A.1 in the appendices. The table lists all the names of the categories.

3
Image analysis theory

This chapter introduces and explains how images can be represented in more ways than the rasterized formats that simply consists of the pixel data as in compression standards such as .jpg and .png. It starts of by introducing image compression, then continue to describe what features and what feature descriptors are as well as the theory behind the feature descriptors in the scope of the thesis.

rephrase

scopr of the thesis, works?

3.1 Visual features

Features in image processing are embedded information in either the picture itself or the meta-data concerning the picture. These features are extracted and used to solve various problems in computer vision, machine learning and pattern recognition. In this thesis onlyfeatures embedded in images will be taken into consideration and the descriptors will be of the global type which emphasizes on the features relevant over the whole of image and not on the interest points as in local feature descriptors. There are myriads of methods to extract features from an image and which one that suits the problem at hand varies. Commonly used features for CBIR are those that describe color, texture and shape. When the features have been extracted there is the choiceto describe them in different ways and usuallythe output of these are so called feature vectors. The length of the feature vector varies, depending on the image being processed, the method used for description and the detail in which the features are extracted.

wording

theory?

there is the choice?

usually?

3.2 Feature detectors and descriptors

Feature detectors are usually built towards detecting either global features or local features. Global feature detectors are often color or texture oriented and tryto describe the picture as a whole. These are good at identifying similar images but have a hard time to distinguish between foreground and background of the image as they have focus on the total image qualities and usually fail to find local nuances and differences. They are often built to output a feature vector with focus on some property of the image involving all the pixels. In contrast the local feature detectors focus on keypoints in the image and try to describe these. This leads to local feature detectors having edge occluding parts of the image as objects and are often used to look for similar, or same, objects between images. The result is often several vectors representing the points of interest in the image. These attributes come with a cost, local feature detectors are often expensive in terms of computational power and data storage [24].

tries? är osäker

raphrase

Long sentence. Hard to maintain focus

this leads to... sentence strange.

A Feature descriptor is used when the interest points have been identified, or detected as previously stated. The descriptor creates a set of vectors based on this information which can be used in the proposed retrieval model. In image processing

and image analysis one uses feature descriptors to facilitate the transfer from an abundance of features that are derived from images to a subset or transformation of these features to create a more manageable dataset. Depending on the scope of the project the raw features one would get from the feature descriptors can become too large to handle and work with. To minimize this problem, the feature descriptors can be said to carry out dimensionality reduction on the data. Without any form of reduction of the actual information size there would be a large requirement on memory and computational power. A risk of overfitting also becomes relevant with to much focus on large swathes of features in the relevant images which makes it harder to find images that are alike.

rephrase, having same concept or objects but not alike

There is also a dividemade by the performance of the feature descriptors. If a feature descriptor has a precision just above random chance it is called a weak learner, this can be a descriptor that focuses on different color variation or edges in an image while trying to discern some higher grade of concept. A strong learner in contrast is a feature descriptor with a much higher precision which on its own can discern a majority of the images, this can e.g. be a deep convolutional neural network.

divide? Paint more.

3.2.1 Histogram of oriented gradients

more ref

Histogram of oriented gradients (HOG) is a feature descriptor used for object detection in the fields of image processing and computer vision. The idea being that a local object shape and appearance can be described by the distribution of intensity gradients, or edge directions. The HOG computes the first order gradient, as these capture contour and some texture information. This is all done on the locally dominant color channel which usually is the gray channel. Then a pooling method is used where the image is divided into a number of cells in where the 1-Dgradient orientation is acquired. The orientations are distributed in a fixed number of bins, or possible orientations. The different magnitudes determine the result of the gradient histogram. When these has been evaluated the cells are grouped within blocks which creates a film over the "surface" of the cells of the image thus creating a new normalized gradient oriented image. The block are set to overlap thus each cell are accounted for several different calculations over different blocks. The normalized blocks are referred as HOG descriptors. The descriptor is essentially the concatenation of these local histograms. The HOG is well suited for human detection according to [25].

1-D?

Det vore bra att illustrera vissa saker här med bilder. Kan ni inte t ex ge exempel på histogram

3.2.2 Global color histogram

A color histogram is a representation of the color distrubution in an image. The color histogram method counts the number of pixels with similar attributes and store them in a number of bins. There are no specific sizes for the bins, though there is a max size based on current color range used. The size used is dependent on the performance of using less bins in contrast to computational cost of using more bins. In image analysis it is common to use HSV for this histogram, which would result in three dimensions of bins, one for each color channel, and a max of 256

overfitting?

bins in each HSV color range. When calculating a global color histogram (GCH) one does not take smaller patches of the image and calculate the concentration of information locally, instead it is calculated over the image as a whole. This approach can be insensitive where the objects might be similar but the color characteristics are. Two completely different images can still contain the same GCH values due to that the images have similar color settings.

3.2.3 Wavelet transform

In this project a Haar Wavelet transform was implemented, the simplest of wavelets. Haar-like features are a type of digital image features which are used in object recognition [26]. The wavelet was implemented by first taking the differences and means of each pixel to its neighbors. This is done in a manner that for the difference each pixel is used once in horizontal calculations and once in vertical calculations, the values are then divided into the different rectangles. The signal is decomposed into a subset of signals representing different elements, approximations of the image and intensities in the different directions and orientations. So for each time the wavelet transform the image a couple of smaller ones are created with the information of with a dimensionality of 2. This makes it possible for a wavelet with an image of size $256 * 256$ to be minimized 8 times ($2^8 = 256$).

in a manner that for the?

Figure 3.1: Caption

write caption for figure.

3.2.4 Convolutional neural network activations

The field of Artificial neural networks emerged with the McCulloch and Pitts neuron in 1943 . The idea was to simulate the human brain, where there are about 10^{11} nerve cells, or neurons, that through a symphony of signals communicate information from and to other neurons. This is achieved by creating nodes for the neurons and edges interconnecting the different neurons to simulate the axons and dendrites. Dendrites works as a form of input to the neurons which have different intensities in their signals, in artificial neural networks modelled using weights. The neuron, or soma, then sums up the inputs into some outputwhich then the axon signals out to other neuron or as the final output of the system. From the basis of this simple adaptation the simulated neural network is created. In the simplest form a single layer of neurons with some form of inputs, valued with weights, which in the

Ge referens

flow, rephrase

split sentence in more lines

animalic brain is the intensity of the signal measured in frequency of the firing of axons, which then transforms the data into some form of response, or output. There are several methods that have been created in its wake with certain functionalities.

examples of this

3.2.4.1 Convolutional neural networks

In this thesis a subset of the field of artificial neural networks is used; convolutional neural networks (CNNs). A CNN is a type of a feedforward neural network that tries to simulate the animal visual cortex. A feedforward neural network is an artificial neural network wherein the connections between the units do not form any cycles or loops. A multilayer feedforward network is composed by, an input layer, an output layer and zero or more "hidden" layers. In this simple illustration the first layer, the input layer, receives the first number of inputs to be processed by the neurons. this is then sent to the next layer as inputs and the chain continues until the final layer outputs the final result for the whole neural network.

3.2.4.2 Network architecture

Is this a good title?

go on? point is missing.

The architecture varies a lot between different designs. Although, the building blocks are convolutional layers, pooling layers and (the more commonly used) fully-connected layers. They are variations of multilayer perceptrons which are designed to use minimal amounts of preprocessing. The CNNs are constructed by using multiple components of distinct design, some of the most commonly used ones are the following.

check this sentence again.

a

The convolutional layer is the backbone of CNN. They are composed of a set of learnable filters, aka kernels, with their distinct size of receptive fields. They traverse the whole area of the image, convolved across the height and width of the input, all

Partial sentence

the while computing the dot product. This produces a 2D activation map for each filter, which are all stacked along the depth which creates the output matrix, the

sentence. volume out of nowhere

volume for the convolutional layer.

The pooling layer is essentially a non-linear down-sampling of the input image. This is done by some algorithm where max pooling is one of the most commonly used ones. It splits the input into several smaller non-overlapping rectangles and then outputs the max value from these regions. The idea is to periodically insert pooling layers in between the convolutional layers, in so doing reducing the number

needed for the/of

of parameters as the size decreases and thus lessen the amount of computationin the network. This also helps to reduce the risk of overfitting.

the

The rectifier linear unit, known as ReLU, which applies a non-saturating activation function to increase the nonlinear properties of the network. This is done without directly affecting the receptive fields of the convolutional layer. The general idea is to reduce the time it takes to train the network while still retain the generalizing nature of the network.

add what it does. 0 if x less than 0, 1 otherwise.

The fully-connected layer is the last couple of layers are usually built as fully connected layers. Layers have full connection between the each other meaning that the activations in previous layers connects to all neurons in the next layer. These layers constitute the high-level reasoning of the network.

elaborate

strange sentence

rephrase

The softmax activation function is normally used to produce the final output of the CNN where a loss function is set to determine how to penalize the network if prediction deviate from actual labeling.

elaborate

One of the hardest parts is to configure a CNN based on these layers to create a well versed and functioning network. In this thesis the VGG-16 network is used. How the VGG-16 is built by using all these different layers in their convolutional neural network is shown to the left in Figure 3.2. A method called transfer learning is applied where one use pre-trainedCNN models and then just remove the last output layer, and extract the features directly from the fully-connected layers [27]. To be able to use the network for the purpose of a feature descriptor modifications were made. The the final fully-connected layer as well as the softmax layer, the loss layer, are both omitted as can be seen in Figure 3.2.

cite vgg paper

rephrase

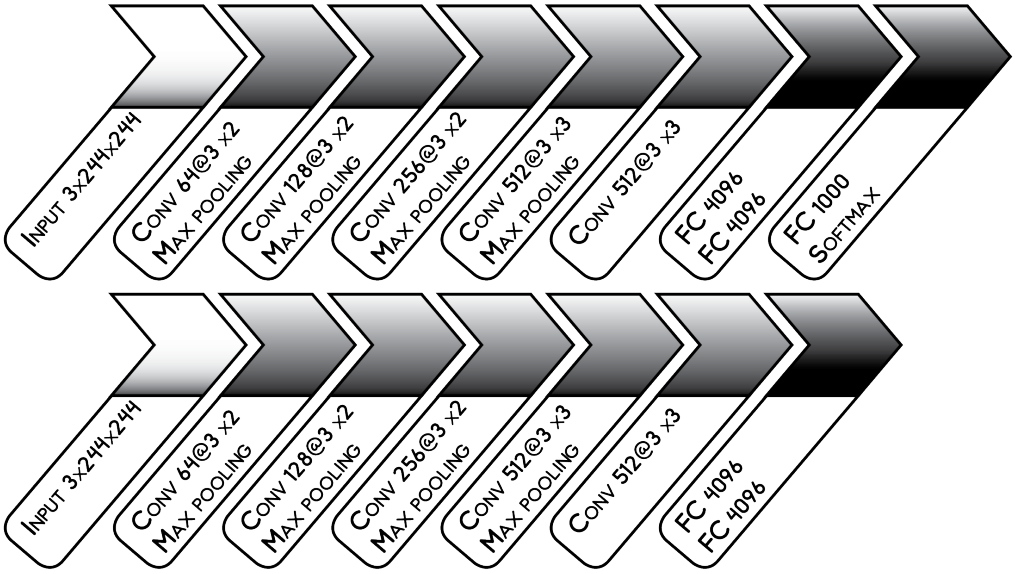


Figure 3.2: Top: A simplified visualization of the VGG-16 CNN. Bottom: The modification done in this thesis.

3.2.5 Edge detection histogram

The set of edge detectors is a group of different methods that aim to identify strong shifts in images, specifically of image brightness, which could signify discontinuities of some sort in the image [28]. The name edge detection is based on the relevant points that signify a discontinuity which are called edges. There are two distinct methods commonly used in edge detection which are the *search based* and the *zero-crossing based*. The first method, called the search based, looks for the local directional maxima of the gradient magnitude, often using a first-order derivative expression to compute this. Examples of these are the Roberts, the Sobel and the Prewitt operator. All edge detectors that all utilize convolutional masks in order to to calculate the gradient. Zero-crossing based, the other method, uses second order a derivative expression to find where there is a jump of values; the zero-crossing of

ref.

ycbcr is not mentioned here.

the image. Example of these are the Laplacian operator, which often is used with an approximate convolutional mask [29]. The identification of these edges can be used to find more relevant objects and shapes in the images, which can be used to build predictions on [30].

4
Machine learning theory

This chapter introduces key concepts of the computer schience subfield machine learning. The chapter start of with a brief introduction to machine learning, learning of the systemand classfication method used. Continues to present the idea of Support Vector Machine and its mechanics to endthe chapter with the focus on the SVM ensemble which is for getting a better performance.

vill vi formulera det så här?

Menar ni "Starts off"?

Ofullständing mening

Machine Learning is a field in computer science focused on methods where the computers learn without being explicitly programmed. Thus can be said to be the study and construction of algorithms that can learn from data and make predictions based upon it. In order for a person to learn about something new, the person looks back on previously learned knowledge and machine learning algorithms do not differ from this pattern. A famous quote that describes machine learning by Tom M. Mitchell is *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"* [31]. Machine learning is applied to several different subjects in a number of different fields. Even though there are several learning methods and utilities of machine learning this thesis will only handle the concepts of supervised learning and in extension classification. Supervised learning will be used as it is most applicable to the problem at hand as well as it is the learning method currently showing best results while one have "labeled data".

learning of the system?

method used?

partial sentence

strange. The chapter ends the chapter.

not really why it is used.

extension classification? What?

Supervised learning will be used as it is most applicable? Necessary?

and labeled data is often used to reward semi-supervised learning.

4.1 Supervised learning

Supervised learning is a training method for a computer program, where labeled data is explicitly used. Labeled data is a group of samples $\mathbf{x} \in \mathbf{X}$ composed of some form of information, or data, distinguishing the samples $\mathbf{x}_i$ where $i = \{1, ..., t\}$ and labels $y_i \in \mathbf{Y}$, or targets, corresponding each to one sample. The labeled data is usually split into a training set, a validation set and a test set. The training set is the data used to train some form of algorithm, so if presented unknown data, it should be able to determine which samples belong to which label. So presented with a set of samples $\mathbf{x_j}$ where $j = \{1, ..., s\}$ it should predict the correct labels y_j. The validation set is then applied to get an idea of how itsperforming and to see if further changes is needed to get an acceptable result. If one tries multiple approaches, this should determine which to use if the learning algorithm performs as anticipated. The last part of the labeled data, the test set, is then used to check what a possible expectation of the algorithm could be when presented unlabeled data. When these threeparts are completed the supervised learning is done.

Gap in sentence: algorithm, so if presented unknown data, it

check sentance flow

it is

what three parts?

4.2 Classification

Classification is a general problem in pattern recognition where some form of input value should result in an output value which is representing the label of the element. The research behind classification is extensive and many different fields are working with classification. The choice of which classification method that should be used varies depending on the data. It is of note that no one classifier suits all cases and no one classifier outperforms every other in every other case as per the "no free lunch" theorem from Wolpert and Macready [32]. The most basic form of classification is *binary classification* where the data is separated into two categories, for instance as relevant and non-relevant. The input to these algorithms are more commonly known as feature vectors, $\mathbf{x_i}$. Since the number of dimension of these vectors might vary inbetween different feature descriptors, the choice of classifier type is important. Since *Support vector machines* (SVMs) are good at handling feature vectors of both small and large numbers of dimensions, are fast at classifying and are relatively tolerant to noise, this type of classifier is used in this thesis [33][34].

4.3 Support vector machines

somewhere in this paragraph: Insert that it is a geometrical comparison.

A Support vector machine (SVM) is essentially a supervised learning model where data is analyzed for classification and regression analysis. Can be said to be a non-probabilistic binary classifier since new examples are assigned to either of two categories. The SVM model is a representation of the examplesas parts in space that are mapped in a way, as clear as possible, that separates the classes by a margin. A data point of a set is considered as a p-dimensional vector (composed of p numbers) and the goal is to be able to separate the data with a (p-1)-dimensional hyperplane.

does this sentence explain non-probabilistac-ness?

rephrase

of the examples?

SVMs have been found to function well on both small and large numbersof dimensions [35], but all datasets are not easily divided by a linear model. Due to this the *Kernel trick* was invented. A non-linear classification implicitly mapping their inputs into a different dimensional feature space.

size

SVMs can be used for classification, regression and outlier detection. But since the thesis has its focus within classification the theory covering the other two use cases will be omitted.

In order to classify an SVM constructs a hyperplane, or a set of hyperplanes in higher dimensional spaces, that can be used to classify data points depending on which side of the hyperplane they reside. Due to the limitation the number of sides that exist of an hyperplane this becomes a binary classification and therefore the label set becomes binarily defined as $\mathbf{Y} \in \{1, -1\}$. The optimal hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$, as seen in Figure 4.1, is found when the given training data is separated with an as large margin $\gamma = \frac{1}{||\mathbf{w}||}$ as possible. Which means that it will also be found when minimizing $||\mathbf{w}||$, as well as when minimizing $\mathbf{w}^T\mathbf{w}$. Given training data points, or

rephrase

and instead of all else

vectors, $\mathbf{x_i} \in \mathbb{R}^p, i = 1...n$ and the respective label $y_i \in \mathbf{Y}$ the primal (4.1)

$$\begin{aligned} &\min_{\mathbf{w}\in\mathbb{R}^p, b\in\mathbb{R}} \mathbf{w}^T\mathbf{w} \\ &\text{subject to} \\ &\forall j\colon (\mathbf{w}^T\mathbf{x}_j + b)y_j \geq 1 \end{aligned} \tag{4.1}$$

can be constructed.
As soon as a stable hyperplane has been found the test set can be classified by simply checking which side, of the hyperplane, the data points end up on by computing the label value (4.2)

$$y_i = sign\left(\mathbf{w}^T\mathbf{x}_i + b\right). \tag{4.2}$$

The larger the distance from the hyperplane to a point the more certain the prediction is that the data point belongs to a certain category. Hence the data points that are within then margin of the hyperplane have the most uncertain predictions. This can in fact be used in order to calculate some certainty that a data point belongs to a class or not. If the distance between two data points and the decision boundary compared is of different sizes, the point with the greater distance is more probable to be of the desired category [36].

rephrase

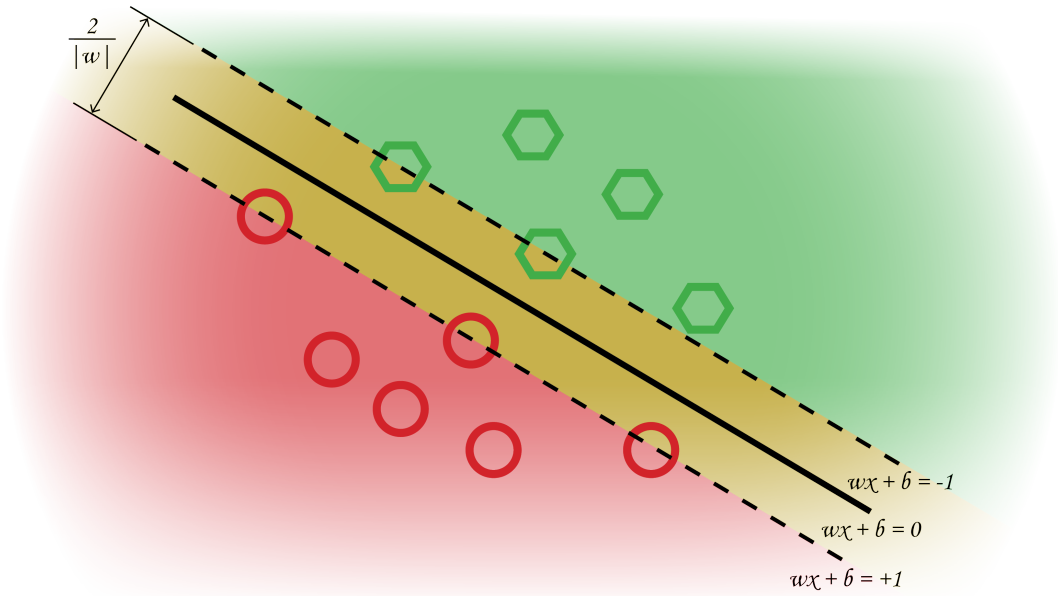


Figure 4.1: A simplified visualization of how data is linearly separable in a two dimensional space.

4.3.1 Kernels

Kernels define the cartesian product between vectors, which can be used to get the a real value. In order to create a kernel one defines a function K from the cartesian product of the feature space to a real value ($K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$). This real value can subsequently be used to evaluate a distance value. Depending on the chosen

kernel the distance have different attributes an thus the kernel can be selected to receive better results for different datasets. Kernels in SVMs are different methods of how the hyperplane is generated for the SVM and thus gives different ways of separating the data. The most common ones are the *linear* and the *radial-basis function* (RBF) kernels. The linear kernel is a straightforward approach which is as the name suggests a separates data in in a linear manner. But when data data is not linearly separable this kernel will not suffice. An example of a kernel that could solve solve this problem is the RBF. The RBF kernel is a fast approach that often work, as long as the feature space is not too large, and can be used to separate inliers from outliers. The kernel often uses a norm function between two points in order to separate them, e.g. the squared two-norm (4.3)

$$K_{RBF}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2^2 = \sum_{i=1}^{p} (\mathbf{x}_i - \mathbf{y}_i)^2. \tag{4.3}$$

Different kernels are used to make data points linearly separable in their own dimensional spaces, causing the decision boundary in the original feature space to be, and appear, non-linear.

4.4 Ensemble learning

Ensemble learning methods use a setup of different learning methods which are then combined in to achieve a better predictive behavior than if using a single one. There is no guarantee that the result will be better with only combining several differentmethods though. There are several things to be taken into consideration. The different classifiers have to show some form of diversity in their estimation otherwise it will be more like doing the same calculation over and over. If there would be one dataset and several classifiers that are designed in the same fashion they would have to be trained on different training sets to create some form of diversity [37][38]. A more applicable approach is to use different methods of classification. One can use the Condorcet jury theorem as an example of this methodology, *"If each voter has a probability p of being correct and the probability of a majority of voters being correct is P, then p>0.5 implies P>p. In the limit, P approaches 1, for all p>0.5, as the number of voters approaches infinity"* [39][40]. There come great potentials with the use of several different classifiers. The relevantspace have potential to be much more based on each andeveryone single one of the classifiers notation. This because a single classifier might get a part of the solution in linear case while the use of several can be dimensioned to be able to solve higher dimensional problems. The SVMs should not be over the same datasets since they in that case will have the same errors and thus the SVMs should focus on different targets and create their respective prediction spaces to reduce the possibility the errors are correlated. There are different ways how to [41].

wording

relevance? Does not support the deep svm.

rephrase

the relevant space have? what?

each and every one single one?

This is not in line with anything we do. They train on same data but with different learners. all have same targets.

4.4.1 Deep support vector machine

A *Deep support vector machine* (Deep SVM) is model aimed to enhance the performance by using multiple SVMs by positioning them in layers, inspired by deep belief

networks [42] and other stacking generalization approaches using SVMs [43]. The idea is to build layers of SVMs where the output of the previous layer becomes the input of the next layer. The use of more layers give new possibilities in classifications which cannot be achieved with single, however complex, kernel functions. One example of this is the XOR function which can not be solved with a single SVM, but can be when using layers of SVMs. The structure can be perceived in Figure 4.2, where the initial box, presenting the different feature vectors to the classifying system. The first layer of SVMs receives the feature vectors as input and the output of the first layer becomes the input of the following layer. The number of classifiers in the first layer is only restricted by the number of feature vectors presented to the system, which is an arbitrary number. The final layer, in the figure called "Meta SVM", presents the final result of the system.

elaborate?

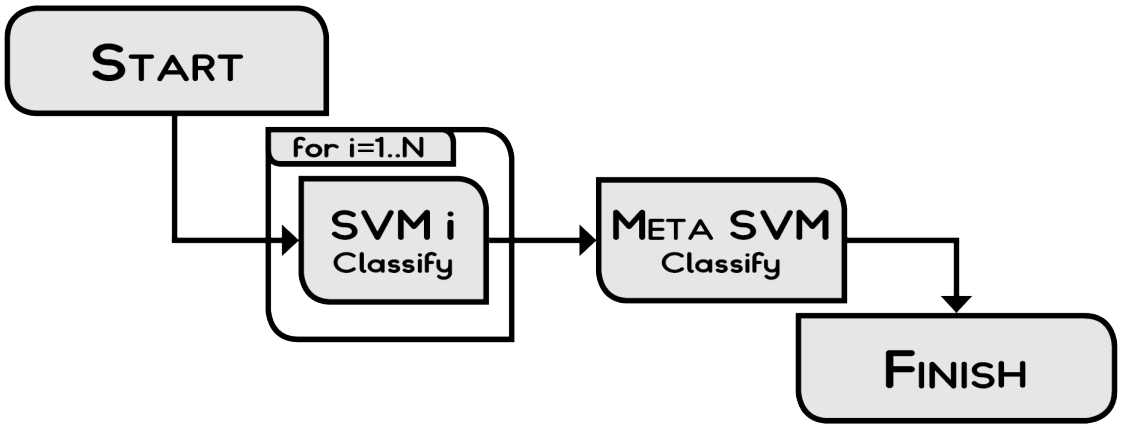


Figure 4.2: A simplified sketch of how a Deep SVM classifies data. The test data is passed through the first order classifiers and the output of those is the input for the Meta SVM. The output of the Meta SVM is the distance from the decision boundary that the entire classification system has created.

The training of a Deep SVM is performed in several steps, as presented in Figure 4.3. First step is to perform a K-fold split on the training data, $T = \{T_1 \cup ... \cup T_K\}$, that is applied to the classification system. All the first layer classifiers (first order classifiers) are then trained with the training subset $T_1^c = T \setminus T_1$, in order to use the remaining subset of the training set T_1 as a test set. The output of the first order classifiers then becomes the training subset for the second order classifier $T_{1_{meta}}$. This process is repeated K times to produce the full training set for the second order classifier T_{meta}. When the K-fold process has been completed the first and second order classifiers can be trained with the full training set T and T_{meta} respectively. The process of training a single SVM is described in Section 4.3.
This setup takes much more time to train than when just using a single SVM. The reward is an estimator that is capable of a more abstract level of classification.
Which kernel functions that the different classifiers have in the Deep SVM does not matter since each unit is independent of the other ones. The selection of kernels in the first order classifiers depend on which feature vectors that they have as input. The classifier at the second layer, however, separates an n number of dimensions if

there are n classifiers in the first layer since they have all produced an estimated distance to a decision boundary. Due to the low level of dimensions and the values of the input vector should be positive if a point is predicted correctly, a linear kernel is often possible to apply.

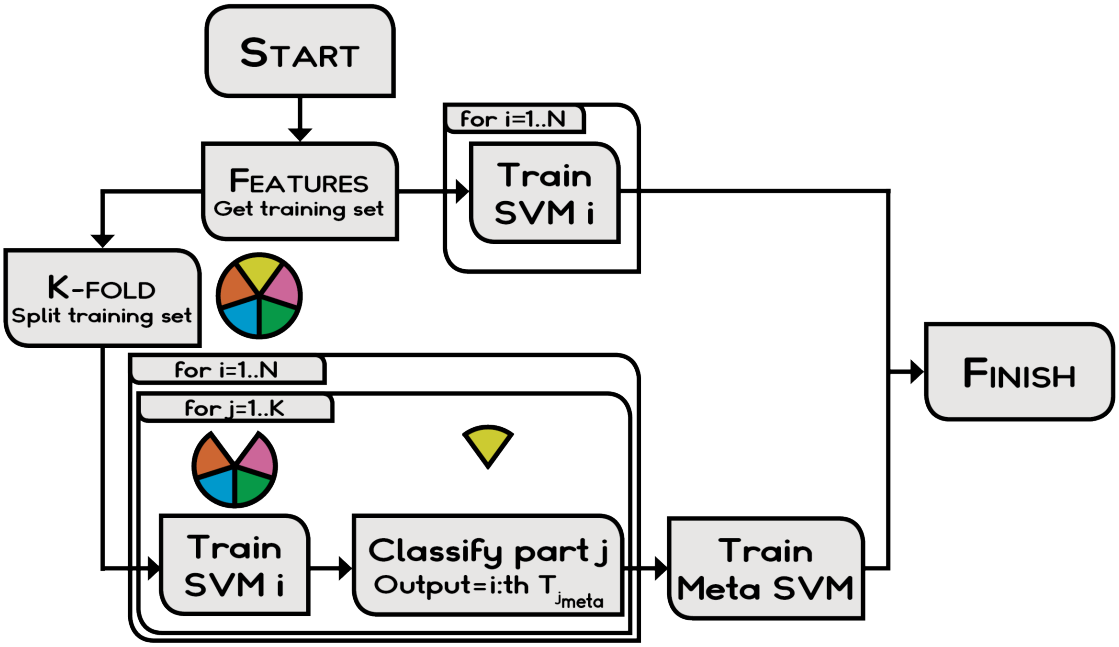


Figure 4.3: A simplified sketch of how a Deep SVM is trained. The Meta SVM needs approximations of how the first order classifiers treats the training data in order to fit its own decision boundary. Besides from that the layers can be trained in parallell.

5
Method

Herethe methodology of the thesis is presented. Related approaches are presented as well as how the proposed model is structured. In the end of the chapter there is a presentation of how evaluations are performed and which evaluations that were made.

This is where

what the evaluations are.

5.1 Related approaches

The approach of modeling a system presented in this thesis is, after checking numerous papers, still untested. There are however several papers that implement different components of the proposed model. Most content based image retrieval (CBIR) systems use a set of different feature descriptors that are proven to be good at finding equalities or similarities between images. The system is then presented with a single query image in order to find matches in a database. The images in the database are compared to this query image and the similarities are calculated in some way [14][15][16], e.g. calculating the Euclidean distance and sorting the data having the most similar first. Other CBIR systemshas created a feature vector from extracting certain feature descriptors, trained a neural network with a subset of the data and usesthe classifier to retrieve images [17]. There are other implementations where the use of relevance feedback is used in conjunction with the feature descriptors to even further increase performance, in light of the difficulty identify descriptors that are good at "understanding" concepts [44].

to our knowledge

Man brukar skriva "to our knowledge"

content-based

data points

have

have

use

to

feature descriptors

5.2 Proposed model

This thesis presents a model that uses relevance feedback and CBIR in order to categorize unlabeled data in an iterative and a more efficient manner. The material that has been recategorized by the user through relevance feedback in previous iterations can be used by the model to present better matches in future iterations. The unlabeled search space will in other words shrink as the labeled set for training will grow.

As mentioned in Section 5.1, the proposed model slightly deviates from other setups. Yet, the general structure is the same as most CBIR systems use. The proposed model consists of three modules; one for matching, one for feature extraction and one that handles relevance feedback. When a search iteration is initiated the matching module fetches a training set and information about the current search space from the feature extraction module, makes elaborate predictions about the material and passes the most accurate information to the relevance feedback module. The relevance feedback module processes the information, requests feedback from the user, passes the ground truths on to the feature extraction model, that updates the search space with new information, and then terminates the iteration which allows a new one tostart. This communication between the modules during a search

a search space of

verified or recategorized

to be started

The

iteration can be seen in Figure 5.1.

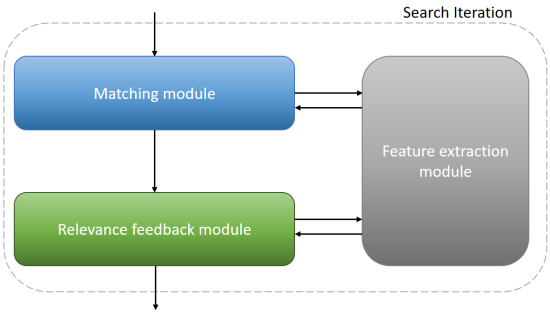


Figure 5.1: The system consists of three modules; matching, relevance feedback and feature extraction. Here the workflow for the model during a single search iteration is presented.

5.2.1 Relevance feedback module

There are a numerous ways of using relevance feedback in order to improve CBIR and categorization of material. In Section 2.2 the different ways of relevance feedback are divided into three categories and they are referred to as explicit, implicit and blind feedback. In order to make elaborate guesses the model needs to have validated data in its training set and as mentioned in Chapter 1 all the case material has to be handled by an investigator in order to build a case. The model has therefore been designed to use explicit feedback in the end of each search iteration.
The feedback that the model receives from the user gives the module information about which images that were correctly categorized and which images that were falsely categorized as negatives as well as positives. This information could be used explicitly to prevent similar mistakes to happen when categorizing the same images. This will however not happen because these images can from this point be used in the training set and thereby not be falsely categorized again. Due to the fact that the model is designed to deplete the search space of relevant images, the only information that is drawn from relevance feedback in this model is the knowledge of which images that are relevant and which are not for the specific case. This knowledge is passed on to the feature extraction module and the search iteration is then terminated as seen in Figure 5.2.

instead of found in the unlabeled search space

will

figure text. ...intermediator of the user...

figure text. ...feedback is given by the user the...

The information that the relevance feedback module receives from the matching module is overly simplified to reduce calculations. The material that is received is sorted to have the most relevant images first and the least relevant images last. The material is however only labeled as relevant and non-relevant. But to present all the material at once would be overwhelming for the user and the model would not need to learn iteratively. To present the *top-k* images is a common method within CBIR, and the number of images that are presented could make it easier for the user to oversee the material. The number 20 was arbitrarily chosen and was manageable. An extension to this *top-20* approach in order to improve classification was to also present some images from the other extreme: The *bottom-k* images. With the intent

empirically

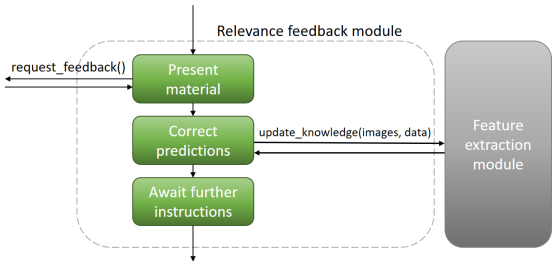


Figure 5.2: The relevance feedback module is the intermediator for the user and the rest of the model. When the relevance feedback by the user is given the search iteration can be terminated.

to avoid drowning the user with information it sufficed with 5 images. Resulting in that the user could quickly give feedback to 25 images in total every iteration.

5.2.2 Matching module

As a search iteration is initiated the matching module begins with retrieving training data from the feature extraction module. This training data consists of relevant and non-relevant images that can be used to fit the classifier of the model. When the classifier is set up the search space can be retrieved from the feature extraction module and then lateron be explored. The search space is processed in batchesat a time to avoid performing predictions on more images than necessary. When the exploration of material has resulted in an sufficient amount of material, the material is sorted from most to least relevant and then passed on to the relevance feedback module with labels of relevant or not. In Figure 5.3 there is a visualization of the workflow for the matching module during a search iteration.

remove “later on”

one batch at the time

labels of the material being relevant or not

of

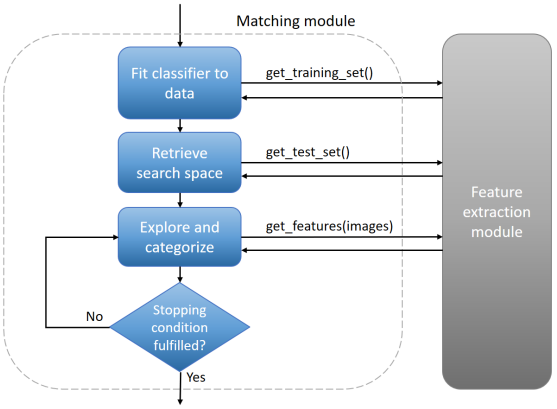


Figure 5.3: The matching module performs the initial work of a search iteration. At this point the feature extraction module provides with a training set and details about the search space.

5.2.2.1 Classifier

As mentioned in Section 1.3, there was no plan to compare how well different classifiers would perform in this thesis. But since having many dimensions can result in more general decisions, a classifier that scales well is to prefer. One classifier that is capable of handling a high number of dimensions is the support vector machine (an SVM), see Section 4.3. Comparative studies such as [45], [46] and [47] have deemed SVMs as classifiers that continuously show good results in different implementations. In the field of CBIR there are myriads of different feature descriptors that are used and the more feature descriptors one can combine, the more general the classifier can become. As mentioned in Section 1.3 the number of different feature descriptors used in this thesis is limited to 5. This solely because adding or removing feature descriptors could improve performance in a general sense, but there was not enough time to cover the subject. In order to combine these 5feature descriptors a tree of SVMs were created; one for every feature descriptor and one that treats the output of the different SVMs as input. Read more about the classifier structure *Deep SVM* in Section 4.4.1.

As mentioned in Section 5.1, most CBIR systems can quantify certainty of relevance by using a distance function in order to sort the material from most to least relevant. SVMs are geometrical tools that create a decision boundary in some dimensional space in order to say that all points on one side is one category and all points on the other side is another. Due to the fact that the relevance feedback module expects the material to be sorted the certainty of the classification needs to be quantified. A quantification method is mentioned in Section 4.3: Instead of using the *sign* function to decide a class, just use the distance that each data point has from the decision boundary. This gives the matching module the possibility to sort the data from the most to the least relevant and the categoryof the point can still be categorized by using the sign function later on.

5.2.2.2 Training data

The training data used to fit the classifier is mainly intended to have been classified by the user recently. As mentioned in Section 5.2 the search space (unlabeled material) shrinks as the labeled set grows for every iteration. The labeled data is used to create a training set in order to fit the classifier and make new predictions. In the first iteration however, there is no labeled set to work with and therefor no training set for the classifier. When the classifier can not be fit the exploring of the search space is not possible. Instead of locking the workflow of the proposed model the matching module instead randomly selects some material (without any predictions) to pass on to the relevance feedback module and in the following iterations there will be some labeled data to use.

In order to fit an *SVM*at least one relevant and one irrelevant data point is necessary and in order to fit a *Deep SVM* it is necessary to have two relevant and two irrelevant data points. When selecting material from the search space at random there is no guarantee that enough material is sampled in order to fit the classifier correctly within a certain number of iterations. To work around this issue the possibility to install an initially *predefined training set* was given. The initial training set is

used in combination with the labeled datathat will slowly grow with every search iteration. A justification to add such a training set to the model is that in most cases when a person knows what to look for it has some previous knowledge of how such material would appear. It would even be improbable that the user would not have some material at hand ready to be inserted into the model. When having a predefined training set with at least 2+2 relevant and irrelevant images, it is always possible to fit the classifier and the search space will be explored. Resulting in actual predictions and better odds of finding more relevant material to improve the training set even more.

Not having enough training data results in the incapability of fitting the classifier correctly, but having too much training data would make the classifier to take too much time to fit the best decision boundary for the data. Since the size of a search space could infinitely large, so could labeled set be after a large number of iterations. To prevent the labeled set to become to large an upper size limit was set to 500 data points. No evaluation or research was put into this number. Instead it was selected by the intuition of having a training set of that size it should be possible to present a small portion of material with some certainty. Having an upper limit of training material adds the possibility of two things: The time spent training the classifier is done in the same amount of time every iteration and by sampling a subset from a large labeled set allows the decision boundary to shift inbetween iteration.

add “,”

Ofullständing mening

force

possible

in-between

5.2.2.3 Exploring search space

After the search space has been received from the feature extraction module the exploration loop can begin. In this loop the classifier ofthe matching module is used to categorize the material and calculate their distances from the decision boundary. As mentioned in Section 5.2.2, this is done in batches. The batches of the search space are selected at random, i.e. a subset is easily sampled. The selection process is not covered by the scope of the thesis and is therefore performed as a random search. When thebatch of material has been selected the feature extraction module provides with the feature descriptors for the material in the batch. The material is run through the classifier and receives a calculated distance from the classifier.

In order to avoid continuesampling batches until the entire search space is depleted, three stopping conditions were introduced. The first rule was implemented to eliminate the risk of an infinite sampling loop. If a sampled batch only would contain material that already has been run through the classifier during the same iteration, the search is over. This stopping condition only exists because of how the batches are selected from the search space and will therefore not be evaluated in the same manner as the other two. The second rule is called *Early stopping*: if no relevant images seem to be found, stop the iteration and make the relevance feedback module present the least relevant images. When 200 data points has been passed through the classifier and none of them are categorized as relevant the material with the largest distance from the decision boundary is tallied up. Finally, the third rule is called *Threshold*: meaning that a number of images has to be further away from the decision boundary than a certain value. This value is initially set to 1 until atleast one image in a sampled batch already has been passed through the classifier during the same search iteration. As soon as this occurs the limit changes to a function

in

the

the data points to

is sampled from the search space

a

remove “continue”

every search iteration

if => If

have

meaning => Meaning

value depending on the search space size and how many unique images that has been sampled during the search iteration. The entire calculation for the *Threshold* stopping condition value is seen in Equation 5.1.

bake the equation

$$\begin{aligned} \text{Threshold} &= \begin{cases} \text{unique images only} & 1 \\ \text{otherwise} & \frac{n - x log(x)}{n} \end{cases} \\ \text{where } n &= \text{search space size} \\ x &= \text{numbler of sampled images} \end{aligned} \tag{5.1}$$

in the equation: add ". " after sampled images.

5.2.3 Feature extraction module

really not a

The feature extraction module is really a part of the workflow during a search iteration. It is, however, a necessary supporting module that centralizes the control of information regarding search space, predefined training sets and the feature descriptors that are extracted. It provides the matching module with information about the search space for the current iteration and delivers a training set in order for the classifier to work. The feature extraction module updates the search space with ground truths when the relevance feedback has been received and makes sure that the presented data can be used as a part of the training set.

space during search iterations and delivers

" , "

When the matching module is exploring the search space the feature extraction module ensures that the feature descriptors of material that is about to be classified is extracted. The process that occurs in the background is visualized in Figure 5.4.

data point

If the feature descriptors for a datapointis not extracted, the module extracts them and stores them in the databases for future use as well. As mentioned in Section 1.3, only 5feature descriptors are extracted and used.

five different feature descriptors

Why these descriptors.

why these.

The following Sections covers how these feature descriptors are extracted.

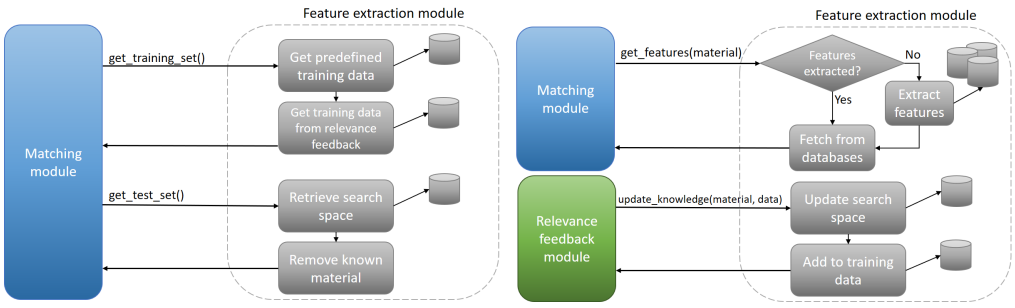


Figure 5.4: The feature extraction module centralizes the control of information sent to the other modules in the proposed model. The feature extraction module provides data to the matching module and updates the search space with information that is passed on by the relevance feedback module.

in the figure: "sent to the other modules of the proposed"

5.2.3.1 Histogram of oriented gradients

The histogram of oriented gradients (HOG) are implemented using scikit-learns algorithm [48]. The functionis presented with an image changed into a single color channel, gray is usedin this case, as well as parameters telling the function how thoroughit should work though the image. Number of pixels per cell and cells per block are selected to determine how large the resulting square shall be. The number of orientation bins are selected to set how many possible gradients should be in the output for each block as described in Section 3.2.1. In this implementation 8 orientations were used, with 32-by-32 pixels per cell and 4-by-4 cells per block.

5.2.3.2 Global color histogram

The Global color histogram is implemented as proposed in [44], with a few modifications. First the color range of the image is changed to Hue, Saturation and Valueor HSV, the color range is covered in Section 2.3.1, instead of any pre-existing color range, generally RGB. Each of thesecolor channels arethen summarized for their individual different number to sort the intensities of the values in a number of pre-determined bins. These numbers can be set to different values based on utility. In this project they are set as 24, 12, 6 respectively, instead of the 8, 4 and 2 which are used in Wang et al. implementation. These bins are divided by the total number of interest points, or maximum possible value to get the values in the range of the SVM as;

$$H(i) = \frac{n_i}{N} \text{ where } (i = 1, 2, ..., 1728), \tag{5.2}$$

where n_i number of interest points, N is the total number of interest points and iindicates the possible combinations of the bins. H are used to create the feature vector for the SVM of Section 3.2.2.

5.2.3.3 Wavelet feature transform

For the wavelet feature transform a Haar wavelet was used. The Haar waveletcompared with other wavelets still has a good performance at cheap computational cost explained in Section 3.2.3. In the proposed model 3 levels of transformation is used, resulting in a $2^3 = 8$ dimensions and a feature vector of length 144. Inthe end the values are normalized and placed in an array to form the feature vector, the implementation is based on the one featured proposed in [44].

5.2.3.4 Convolutional neural network activations

In this implementationan already trainedneural network is used, model of the 16-layer network used by the VGG team in the ILSVRC-2014 competition [20]. The network arepresented an 256∗256 image which is processed, the features are extracted from the VGG:s last activation layer before the last fully connected layer and loss layer, after the first fully-connected layers as per Section 3.2.4. The vectors are concatenated into a feature vector presentable to the SVM described in Section 4.3.

using the algorithms developed by *scikit-learn*

the HOG

an image that is converted into a

remove “is used”

as some parameters

thoroughly

work though => process

The parameters are the following: The number of

resulting square shall be => pooling squares become

As well as the number of

that

or HSV => (HSV).

The color channels are covered in Sec...

remove “, instead of any pre-existing color range, generally RGB”

the

is

strange sentence.

utility?

cite again.

Big leap. The reader needs help to go from 24,12,6 to 24*12*6=1728.

i is explained here and in equation.

something is missing.

replace following with: one of the feature descriptors used by the classifier described in Section 5.2.2.1.

heading: Haar wavelet transform

5.2.3.5 Sobel edge detection histogram

How? Does this work? is the result an edge image? Is the result a vector of numbers?

skriv "four". Direction edge detection? Kanske bättre med directions?

two

four

images? First mention of image here.

numbers

pixel counts of the images

...edges are near are numbered...

vector string?

The Sobel edge detector is implemented with 4 direction edge detection instead of the usual 2. The direction is as the 4major cardinal directions without the opposites, thus having the directions seen in the image. In addition a Gaussian center blur is added to create a overview feel to the edges. The pixels that have values are put into different numberof bins that represent the quantitative states of the images pixel counts. Also the number of places where no edgesare near are numbered into one last bin. In addition to this the number of edge points in each orientation are presented in one final vector string. The edges found in the image are binned as detailed in Section 3.2.5.

5.3 Evaluation of model

on a larger scale

","

and =>whilst

remove after

The feature extraction module, Section 5.2.3, is described as

from here to end of sentence => "the feature descriptors of the search space are extracted before all evaluations begin."

Performing evaluations in larger scalesoften demand that labeled datasets are usedandthe proposed model is designed to have a user standing by aftereach iteration. This section covers how these evaluations were performed in order to achieve the results presented in Chapter 6.

In Section 5.2.3 the feature extraction module is described as a system that will extract the necessary feature descriptors for material evaluations on the fly. But to reduce the time spent evaluating and to avoid removing all the feature descriptors prior to every new evaluation the features were extracted before the exploration of the search space began.

The evaluation is divided into two parts: How benchmarks are performed in order to achieve the optimal settings for the proposed model and how the model compares with other CBIR studies. Since the model requires that a user performs the arduous task of peer-reviewing the predictions of the model every iteration, the simulation of a user was created in order to retrieve data from evaluations.

5.3.1 Relevance feedback simulation

each image => the presented images each iteration

to make sure that the entire dataset is labeled correctly.

are run

be performed

The

This sentence does not work in our favour.

relevance feedback is, as the proposed model suggests in Section 5.2.1, used to help the model to mount the semantic gap. A user peer-reviews each image and makes corrections where necessary to make the entire data set labeled correctly. But to do this on large data sets is both time consuming and takes quite a toll on the user. There is also no guarantee that a user will label the same image as the same category in two separate settings but when searching for the same material. Since the evaluations will be run on datasets that already are labeled this risk of mistakes can be reduced by instantiating a user simulation. The relevance feedback simulation takes the role of a user that communicates with the relevance feedback module. This way time is saved and no user needs to be present in order for the evaluations to run.

Thissimulated user will not be misslabeling images because of negligence or exhaustion as a human would. A normal user would fail to label material correctly in the same extent as a simulated one would. However, the error rate for a "trained" human is

extremely low, comparable to the best neural networks that compete in the *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), e.g. GoogleNet [49]. In this case a "trained" person refers to a person that is aware of the situation and well versed in classifying images. The data sets presented in Section 2.4 are very diverse and in some cases a simulated user would categorize the material differently than an ordinary onwould. This is however not a problem rooted in how the simulated user works but how the datasets are designed and categorized to begin with.

Yet, this still means 4% misclassification compared to 0%.

one

5.3.2 Parameter benchmarks

The presentation of the proposed model in Section 5.2 covers the mostpart of how the optimal implementation is designed. What it does not cover is if some of the design decisions are good and not. In Section 5.2.1 that covers the relevance feedback module Theselection of images to present to the user is mentioned but not elaborated. In Section 5.2.2.3, that covers how the search space is explored, two stopping conditions are introducedand how they affect performance needs to be evaluated. In Section 5.2.3, that covers the feature extraction module, five different feature descriptors are extracted to be used by the classifier. The usage of these feature descriptors alone needs to be compared to when they are combined. Finally the Section 5.2.2.2, the training data for the matching module, mentions having different training set sizes but not which sizes that are expected to work the best. To summarize, the parameter benchmark will consist of the following four evaluations:

- The effect of presenting different sets of images to the user for relevance feedback (Section 5.3.2.2).
- The effect of proningthe search space eachiteration (Section 5.3.2.3).
- How well the deep SVMbehaves compared to its parts (Section 5.3.2.4).
- The effect of using different kinds of training sets (Section 5.3.2.5).

Before specifying which metrics that are intended to be used in these evaluation, the metrics need to be defined. The two more commonly used measures are *recall* (see Equation (5.3))and *precison* (see Equation (5.4)). Often presented in pairs due to the fact that recall can reveal if the matching module presents too many false negatives while precision can indicate if too many false positives are presented.

$$\text{recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \tag{5.3}$$

$$\text{precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}. \tag{5.4}$$

A metric that measures the models effectiveness is the so-called F1-measure[50], seen in Equation (5.5). The F1-measure is the harmonic mean of recall and precision, meaning that recall and precision are equally weighted.

$$\begin{aligned} \text{F1-measure} &= 2\frac{precision \times recall}{precision + recall} \\ &= \frac{\text{True positives}}{\text{True positives} + \frac{\text{False positives} + \text{False negatives}}{2}}, \end{aligned} \tag{5.5}$$

The final metric to present is the accuracy of the model (see Equation (5.6)). Simply

For example in

remove : "that covers the relevance feedback module"

the

In Section... Konstig mening. Omformulera.

and => , yet

replace sentence with: The five feature descriptors covered in Section 5.2.3, need to be evaluated. Not only in combination but how they perform on their own.

Entire sentence: As well as the effect of the training sets mentioned in Section 5.2.2.2. How much can the behaviour of the model change depending on training set sizes.

benchmark will consist of => benchmarks are

pruning

each iteration=> during the search iterations

the ensemble of learners performs

ekvationer del av meningen

Bake

Bake

remove so-called

Bake

cite here instead.

put the rate of correct predictions that are made.

$$\text{accuracy} = \frac{\text{True positives+True negatives}}{\text{True positives+True negatives+False positives+False negatives}}. \tag{5.6}$$

these metrics

Apart from the metrics the evaluations include how long the calculations of an iteration takes, how many images that are passed through the classifier in the matching module and how many of the presented images that are relevant.

In addition to the search space an evaluation set, in complete disjunction to the search space, is used.

In order to make sure that the search space is explored and categorized correctly and at the same time see how well the model performs at classification, two different methods are used to calculate performance. To do this an evaluation set is used and the evaluation set is completely disjunct from the search space. The datasets for the benchmarks and how the are constructed is presented in Section 5.3.2.1.

The benchmark evaluations are measured in two different ways.

1. How well the model classifies an evaluation set each iteration. For the evaluation set the following metrics are used:
   - Recall
   - Precision
   - F1-Measure
   - Accuracy
2. How well the model classifies the images that are presented to the user each iteration. For the search space the following metrics are used:
   - Accumulated recall
   - Accumulated precision
   - Accumulated F1-Measure
   - Accumulated accuracy
   - Retrieved images
   - Handled images
   - Time taken calculating

relevant

When measuring the performance of classifying the search space the metrics are the accumulated value of the performance so far during the search, meaning that after iteration 20 the performance is measured on the 500 images that have recieved an prediction by the proposed model.

In order to retrieve a general trend, each evaluation setting is run 5 separate times and the metrics during these runs are presented with the maximum, the minimum and the geometric mean of each iteration. This allows the graphs that are presented in this section to show how much the metrics varied depending on different factors, such as image selection or how the decision boundary was fitted.

In all benchmarks, except for the one in Section 5.3.2.4 where the ensemble is evaluated against its parts, the classifier is implemented as proposed in Section 5.2and will have use all five feature descriptors. In addition; all benchmarks, with the exception of the one described in Section 5.3.2.5 where training data is evaluated, the matching model will have a predefined training set of 5 relevant images and 50 non-relevant.

classifier section.

The evaluation of the model is intended to be fair and the performance to be measured in an as general way as possible but still possible to perform within a relatively small time frame. The datasets for the evaluation are therefore constructed specifically for this purpose.

5.3.2.1 Datasets for benchmark

Since scenery datasets such as the MIT Places205, presented in Section 2.4.2, have a broad base of image material and have a large variety of material within the categories, Places205 is perfect for a parameter benchmark. However, due to the size of the dataset, the decision to only use a subset of it was made. Instead of using all 205 different scenery categories a subset of 23 classes was cherry-picked: All categories with a name starting with *the letter B*. This gives a $\approx 4.3\%$ chance that a randomly selected image is relevant. But since only 25 images are presented per iteration using the entire subset as a search space would cause the number of iterations to be $\approx$ 11000, the search space was designed to a bit smaller. The search space during the benchmarks consists of 200 images of each category where one of the categories is marked as relevant and the others are not. The choice of just using a part of the MIT places205 dataset makes it hard to compare toother implementations of image recognition implemented on this dataset, as well as the fact that the algorithm, which is designed to be more lightweight and less time-consuming thenthe more usual approach of using large deep neural networks. In a world where machines keeps getting better at replicating and overachieving in fields that humans used to dominate it is important to accept the shortcomings and bring forward the potential that computers have [51][52][53].
In Section 5.3.2 an evaluation set is mentioned. The Evaluation set is constructed by using 50 images, that are not represented in the search space, of each category. Resulting in having an evaluation set of 1150 images with the same probability of randomly selecting a relevant image as in the search space.
In addition to the 250 images of every category used in the search space and the evaluation set, 250 images of every category were sampled in order to have material for different predefined training sets. The different sizes of the predefined training sets vary and how they are evaluated is presented in Section 5.3.2.5. The different training sets are simply constructed to have an as broad base on irrelevant images – sampling some images from all categories that are not relevant – and the smaller sets of relevant images are subsets of the larger sets of relevant images.
To not put all eggs in the same basket, the benchmarks have been performed with three different categories marked as relevant in three different evaluations. These three categories are *Bar*, *Baseball field* and *Bedroom*. The three categories does not have very much in common but some of the other categories in the dataset might have some similarities. Having three different categories results in having three different search spaces, three different evaluation sets and three different set ups of training sets. The data drawn from these evaluations will be presented in parallell.

Places205

remove “of it”

reduced a bit further

benchmark evaluations

remove “MIT places205”

with

. As well

the algorithm is designed

than

“in a world...”: I want to remove this sentence completely. /Gee

setups

categories

5.3.2.2 Classifier learning method

Since the proposed model is designed to acquire deeeper understanding fora concept for each iteration, it is important that the model learns from is chosen in an appropriate manner. 25 images are presented each iteration and this evaluation is designed to decide which setting that is the optimal one. The best setting in this benchmark is used in the evaluations that are performed later. The goal of the evaluation is to find a method that ensures that the model learns the concept as

of

for each => for every passing

that what the model

how these 25 images should be selected

fast as possible but still perform well enough in order to reduce work for the user. Four distinct settings were chosen that are representative of how the data can be selected and still work through the data in an efficient manner. The different settings that were used are

1. **Top20+Bottom5**: To present the 20 images that the classifier finds the most relevant and the 5 images that the classifier finds the most irrelevant. As mentioned in Section 5.2.1, this is the how the model is designed to be present material.
2. **Top25**: To present the 25 images that the classifier finds the most relevant.
3. **Top20+Middle5**: To present the 20 images that the classifier finds the most relevant and the 5 images that are the closest to the decision boundary of the classifier.
4. **Top5+Bottom20**: To present the 5 images that the classifier finds the most relevant and the 20 images that the classifier finds the most irrelevant.

In order to make the different settings deviate as much as possible, the model classified the entire search space every iteration. Thus making the selection of images each iteration more predictable and the measurementsare less diverged in between the five different runs.

5.3.2.3 Limiting search space

Evaluating the entire search space every iteration is not just time consuming but also unnessecary since only 25 images are presented at a time. The stopping conditions presented in Section 5.2.2.3 are therefore evaluated to measure their effect on performance. As previously mentioned the first stopping condition, if a sample ofthe search space only contains material that has been passed through the classifier the same search iteration the exploration is over, was always in use to prevent the possibility of an infinite loop while exploring.

The four different settings evaluated in this benchmark were

1. **All images:** To stop after the entire search space is evaluated.
2. **Threshold:** To stop after enough images are classified to be above a decision threshold specified in Equation 5.1.
3. **Early stopping:** To stop if 200 images have been sampled but none are on the positive side of the decision boundary. In this case present the bottom 25 images.
4. **Both rules:** The additive result of using setting *2* and remove emph*3*.

The intention of the evaluation is to measure the trade-of between correctness in classification during the search space exploration and time spent calculating istances to different data points.

5.3.2.4 Feature descriptors

To determine how well the classifier performs as an ensemble compared to only using its parts this evaluation has been constructed by using or not using the first order classifiers either by themselves or in unison. When only usings one classifier in the first order, the second order classifier is given a linearly separable value in one

dimension and thereby just passes on the information. This evaluation is intended to see which feature descriptors that work well on which categories in the data set. But the intention is also to evaluate if different combinations of feature descriptors will improve the performance of the model. Therefore the evaluation handles 7 different settings:

1. **HOG:** Only using the first order classifier for the HOG descriptors (see Section 5.2.3.1).
2. **GCH:** Only using the first order classifier for the GCH descriptors (see Section 5.2.3.2).
3. **WT:** Only using the first order classifier for the wavelet transformation descriptor (see Section 5.2.3.3).
4. **CNN:** Only using the first order classifier for the neural network activation vectors as descriptors (see Section 5.2.3.4).
5. **Edge:** Only using the first order classifier for the edge detection descriptors (see Section 5.2.3.5).
6. **All:** Combining all the first order classifiers as intended and proposed in Section 5.2.2.1.
7. **All-CNN:** Combining all the first order classifiers with the exception of the neural network activation vectors. This setting was added half-way through the evaluation since the setting **CNN** almost performed as well as the setting **All**.

Haar wavelet transform

refer to accordingly with section name

proposed in section => as in the proposed model described in Section 5.2.

5.3.2.5 Training data

Is it necessary to provide initial training data to the model in order to improve classification correctness before learning a specific concept? And if so, how much difference would it make? This evaluation is designed to answer these two questions. A comparison will be made between only having a predefined training set, only using the data that is provided from the user during iterations and a combination of the two. On top of this, the size of the pretrained data set was also evaluated.
The different settings during the training data evaluation were

1. to train the classifier once in the beginning with given data and use the same classifier until the search space is empty.
2. to train the classifier every iteration with data given by relevance feedback together with a predefined training set.
3. to train the classifier every iteration with data given by relevance feedback only.

before learning => in order to learn

And if => If

The predefined training sets were assembled by different number of relevant and non-relevant images. The different sets consist of

A. 5 relevant and 5 non-relevant images.
B. 5 relevant and 50 non-relevant images.
C. 22 relevant and 484 non-relevant images (thus giving it the same relevance ratio as the search space).
D. 250 relevant and 250 non-relevant images (thus making it contain more relevant images than the search space does).

Since there are no good and concise ways to refer to these different settings, they will in this section be referred to by their index in these two lists. The setting that only uses a predefined data set of 5 relevant images and 50 non-relevant images is referred to as *setting 1B* and the setting that solely uses data given by relevance feedback is referred to as *setting 3*.

As mentioned in Section 5.2.2.2 the model uses at most 500 images from the relevance feedback data. Those 500 images are intended to be as close to evenly divided between relevant and non-relevant images as possible. Since the search space consists of 200 relevant images and 4400 non-relevant images the training data in *setting 3* consists of at most 200 relevant images and 300 non-relevant while *setting 2D* will have at most 450 relevant images and 550 non-relevant images in its training set. This since the setting combines the two training sets.

remove emph

remove emph

taken from the relevance feedback as training data

remove emph

remove emph

This since => Since

5.3.3 Study comparisons

Eventhough the proposed model is defined to become better at retrieving images iteratively and continuely increasing the query set, there is still value in seeing how well a barely trained version of the proposed model compares with retrieval methods from other papers. Most CBIR system are designed to have a single image as a query, [14], [15] and [16] among others, which is not possible with the proposed model. A model using an *SVM* (see Section 4.3) as classifier would require atleast two (one relevant and one non-relevant) images in the query set in order to fit a decision boundary. The proposed model uses a *Deep SVM* (see Section 4.4.1), that uses a K-fold split in order to fit the decision boundary for the second order classifier, and does therefore require at least four (two relevant and two non-relevant) images in the query set. In order to have a fair comparison to the other studies the query set must be kept small. But there are also studies like [17] that use a training set to attack the challenges within CBIR. A too small query set would be unfair towards the proposed model and the query set size will therefore be balanced to compare with the studies at hand.

5.3.3.1 The Corel-1000 evaluation

The Corel-1000 dataset is presented in Section 2.4.1 and as mentioned it is often used in to compare different image retrieval methods. This evaluation is intended to measure how well the system handles the diversity of the different classes in a dataset. Given a query image, present 20 images that are similar and the precision on that set of images is evaluated. The test is intended to be run with every image in the data set as query image and an average for every class is taken. There are in other words 100 data points for each class. But, as previously described in Section 5.3.3, the proposed model requires atleast two relevant images and two irrelevant images in the query set. Which results in 4950^2 different combinations of query sets as well as having 10 relevant images and 10 irrelevant images produces $\approx (1.7 \times 10^{13})^2$ different combinations. Evaluating all combinations would be time consuming as well as pointless since the performance of the model can be observed in a much smaller number of evaluations. In order to test more combinations, the query set is sampled 500 times per class instead of only 100 times. Resulting in an

approximation based on around 10% of all combinations on the smallest query set. In order to calculate how much the different query sets differs during an evaluation the variance of the result is calculated as well. When a query set is sampled the system processes the rest of the search space and presents the $n = 20$ most similar images. The retrieval precison for n images (5.7)

$$P(Q_{k_i}, n) = \frac{1}{n} \sum_{e \in \xi(Q_{k_i})} \delta(\Phi(e), \Phi(r)) \Bigg|_{r \in Q_{k_i}, \delta(\Phi(r),k)=1, |\xi(Q_{k_i})|=n}, \tag{5.7}$$

where Q_{k_i} is the ith query set for category k, $\xi(y)$ is the retrieved set for query set y. $\Phi(x)$ is the category of image x, $\delta(\Phi(a),\Phi(b)) = \begin{cases} 1 & \Phi(a) = \Phi(b) \\ 0 & Otherwise \end{cases}$. In short $P(Q_{k_i}, n)$ is the number of retrieved images that are relevant is divided by the total number of retrieved images. The average retrieval precision for category k (5.8)

$$ARP_k = \frac{1}{m} \sum_{i=1}^{m} P(Q_{k_i}, n), \tag{5.8}$$

where n is the number of retrieved images, k is the desired category and m is the number of evaluations per category. As well as the total average retrieval precision (5.9)

$$ARP = \frac{1}{t \times m} \sum_{k=1}^{t} \sum_{i=1}^{m} P(Q_{k_i}, n), \tag{5.9}$$

where n is the number of retrieved images, m is the number of evaluations per category and t is the total number of categories in evaluation. If briefly described the ARP is simply what the average ratio is of the n retrieved images that are predicted as relevent and actually are of the intended category after m evaluations for every class.

In order to retrieve the variance for each evaluation the result is split up into 20 equally sized subsets of which ARP is measured on. The variance is then derived from those and finally an avarage is calculated over the 20 variances for each subset. Evaluation is run with $t = 10$ categories, $m = 500$ different query sets per category and number of retrieved images $n = 20$ resulting in that the model calculates distances to all the images in the dataset for atleast 5000 times.

As mentioned in Section 5.3.3 the number of query images that will be used is determined by how the well other papers perform at the task. The most common approach of CBIR is to use one query image and calculate a distance to other images in some dimension space. This is what is done in [14], [15] and [16]. But there are other approaches to the problem as well. In the case of [17] where the authors trains a machine learning classifier in order to find images of relevance. One might argue that the proposed model is a combination of these two, where a small query set is fitted onto a classifier and checks it towards the test set. Eventhough there are similarities to the other models a fair comparison can not be made. In [14], [15] and [16] the test set consists the Corel-1000 dataset minus the query image. In [17] the test set only consist of a tenth of the Corel-1000 dataset and in the proposed model the test set consists of the Corel-1000 dataset minus the query set.

6
Results

The proposed model as described in Section 5.2 is implemented and is presented in this chapter. The results of the parameter benchmarks and the CBIR evaluations that are introduced in Section 5.3 are presented here.

6.1 Parameter benchmarks

As mentioned in section 5.3.2 there are some parameters to evaluate in order to find the optimal setting for the proposed model. The evaluations that are presented in this section are:

- **Classifier learning method**: Described in Section 5.3.2.2 and presented in Section 6.1.1.
- **Limiting search space**: Described in Section 5.3.2.3 and presented in Section 6.1.2.
- **Feature descriptors**: Described in Section 5.3.2.4 and presented in Section 6.1.3.
- **Training data**: Described in Section 5.3.2.5 and presented in Section 6.1.4.

In Section 5.3.2 the different metrics used in this evaluations are mentioned. However, the decision to omit some of the graphs from this thesis was made because of inconclusive differences in some metrics and an intention to only present what is relevantin each evaluation. In most cases the information that is excluded by removing some figures from the thesis can be found by interpreting the results inbetween evaluations. But to do so should not be necessary.
Recall that in Section 5.3.2.1 the datasets used for the evaluation is a subset of 23 categories drawn from the set MIT Places205 (see Section 2.4.2) and all benchmark evaluations are performed three times with different categories as the target.

Flip order: The different... in Section ref.

of relevance

dataset Places205

five times with three different categories

6.1.1 Classifier learning method

The entirety of this evaluation is introduced in Section 5.3.2.2 and the results of the different settings are presented here. Recall the interface mentioned in Section **??** in Figure **??** where the user needs to correct the predictions that the model has presented. This benchmarks evaluates what happens when replacing the Top20+Bottom5 setup with something else. The four different settings that was used was to present are **Top20+Bottom5**, **Top25**, **Top20+Middle5** and **Top5+Bottom20**.
As mentioned in Section 5.3.2 the performance of the model is measured in two ways. The performance on an evaluation set as well the performance over the entire search space.

Refer to the proposed model instead.

of the evaluation are to present the

6.1.1.1 Evaluation set

When it came down to classifying the evaluation set each iteration the measurements

remove “it came down to”

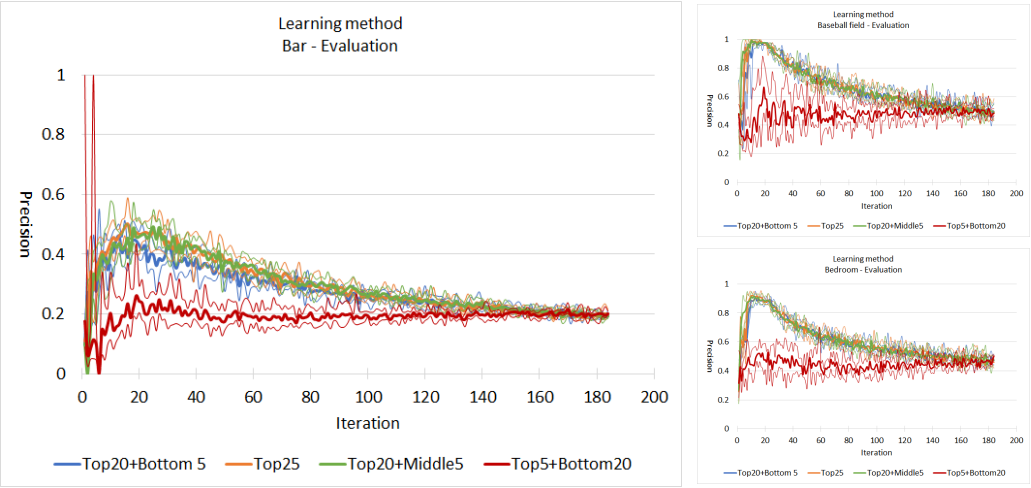


Figure 6.1: The precison that the different settings had on the evaluation set for the three different category searches. Note that the setting *Top5+Bottom20* deviates marginally from the other settings.

of the settings ended up to be very similar. The performance of the first three settings were almost identical. An initial performance peak in classifying the set that later on dropped off. While the performance of the fourth setting varied much more but the measurements were about the same throughout all the iterations.

...setting deviated from the performance of the others the measurements...

Independently of how images were selected the recall (see Figure 6.2) of the settings only differed marginally. With an exception of the fourth setting that in some manner deviated from the other ones. The precision of the fourth setting deviated a lot more which can be seen in Figure 6.1 . The precision of the fourth setting causes the F1-measure to go down as well. Which can be seen in Figure 6.3. The cause of this is that the fourth setting had an higher number of false positives than the other settings had.

Ofullständig mening.

Strange sentence.

connect the two sentences.

An important observation to make is that towards the end of the evaluations all four settings have the same training data and therefor classifies the evaluation set accordingly. Having the complete training set results in an F1-measure around 0.32 when retrieving the category *Bar* and around 0.65 when retrieving the other two from the remaining 22 categories in the evaluation set.

precision figure: remove “marginally”

The accuracy when using the different settings did not vary that much either. As seen in Figure 6.4 the accuracy of the settings is around 85% when classifying the category *Bar* and around 95% on the other two category evaluations. Just as with the F1-measure the accuracy measurements of the first three settings were higher in the first couple of iterations and then dropped off towards the end of the evaluation.

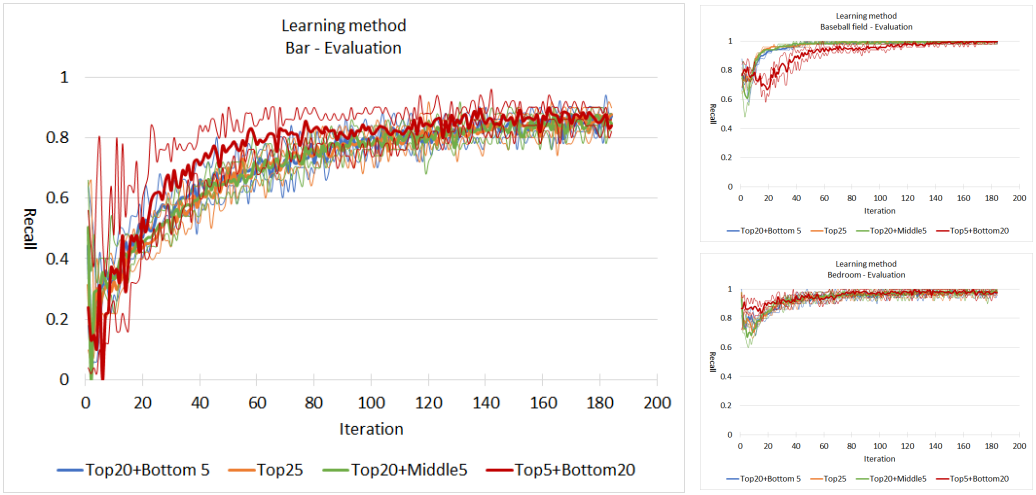


Figure 6.2: The recall rate on the the evaluation set.

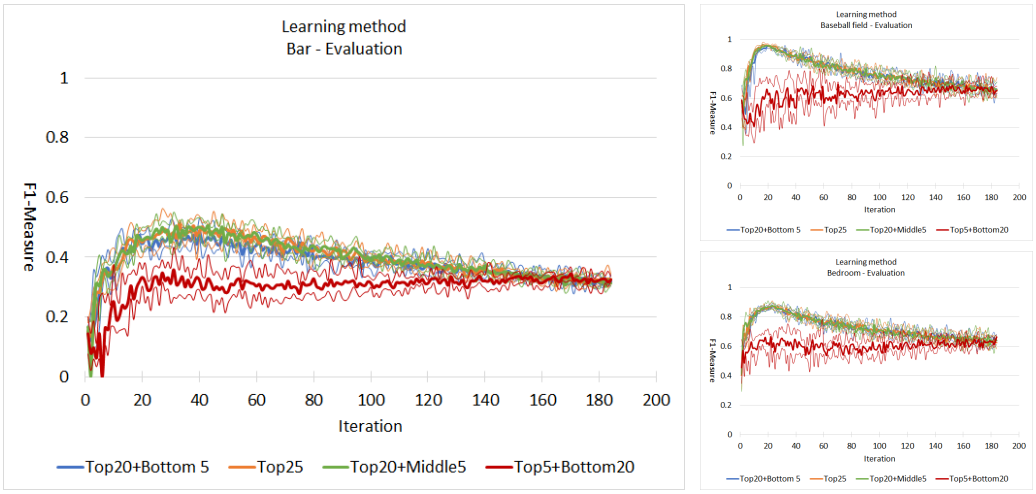


Figure 6.3: The harmonic mean of recall and precision, F1-measure, read on the evaluation set over iterations. The performance of the first three settings peak early and then drops towards the end of the evaluation.

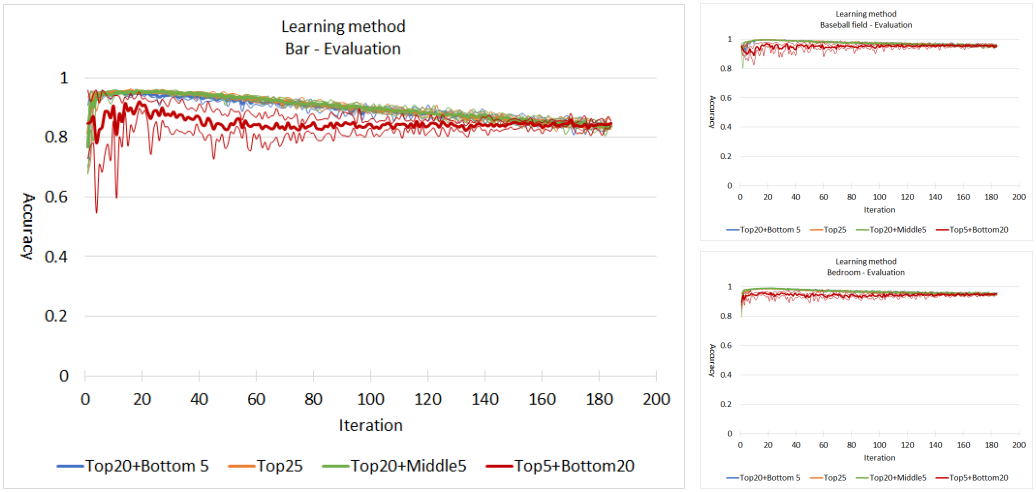


Figure 6.4: The accuracy of the different settings on the three evaluation sets. Note how the accuracy of fourth setting varied more in between runs than the accuracy of the other three did.

figure..evaluations: "between... runs"

figure..insert settings: "three... did"

6.1.1.2 Search space

compared to when the evaluation set was.

relevant

remove "rather"

At the end of the evaluations the fourth setting

strange sentence

sentence

over the search spaces?

third category? bar?

last relevant image

Rephrase the sentence.

the sentence requires a time perspective.

remove "of the setting"

The different settings deviated a lot more from each other when comparing how the search space was classified compared to the evaluation set. The first three settings stopped predicting images as truerather early on in comparison with the fourth setting, resulting in the precision presented in Figure 6.5. The forth setting had a precision of 15-25% in all three categories while the other three settings could maintain a precision of about 80% on the categories *Bedroom* and *Baseball field*. But when classifying the third category, *Bar*, all the settings had a rather low precision. In this category, the first setting was the setting with the lowest precision. While the fourth setting had a relatively low precision, according to the results it did produce the best recall over the search spaces as seen in Figure 6.6. But when searching for the third category it took until the final iterations before the last relevant images were retrieved. To compare this with the other settings, where the finalimage was retreived slightly after half of the evaluation.

In terms of performance as an image retrieval system the setting of presenting a majority of negatives is not to prefer. As seen in Figure 6.7, the number of retrieved images by the fourth setting is a lot lower than the number for the other settings. Looking at the performance on the *Bar* category there are some iterations where the fourth setting has retrieved fewer relevant images than when selecting images at random. This is only momentarily and the rate soon returns to being slightly above that. Reading into the F1-measure in Figure 6.8 and the accuracy in Figure 6.9 of the setting the data is out of favour of the fourth setting. But when inspecting the readings of the *Bar* category the *Top20+Bottom5* setting has a slightly worse F1-measure and accuracy than the three other settings.

The setting that will be used to select the set of images each iteration needs to be picked in order to continue the parameter evaluation. When working with

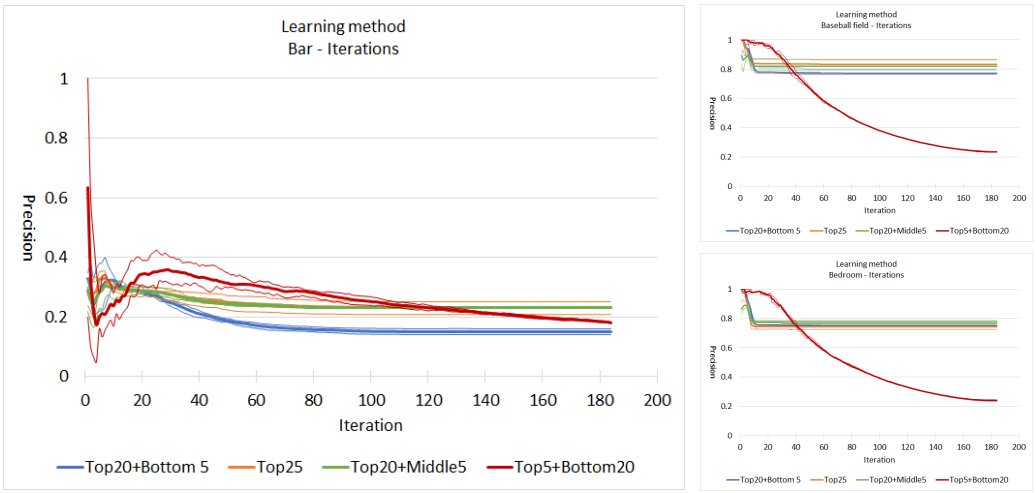


Figure 6.5: The precision that the different settings had classifying the search spaces for the three evaluation categories. Note how the setting *Top5+Bottom20* continued to predict images as positives when they indeed were negatives throughout the entire search.

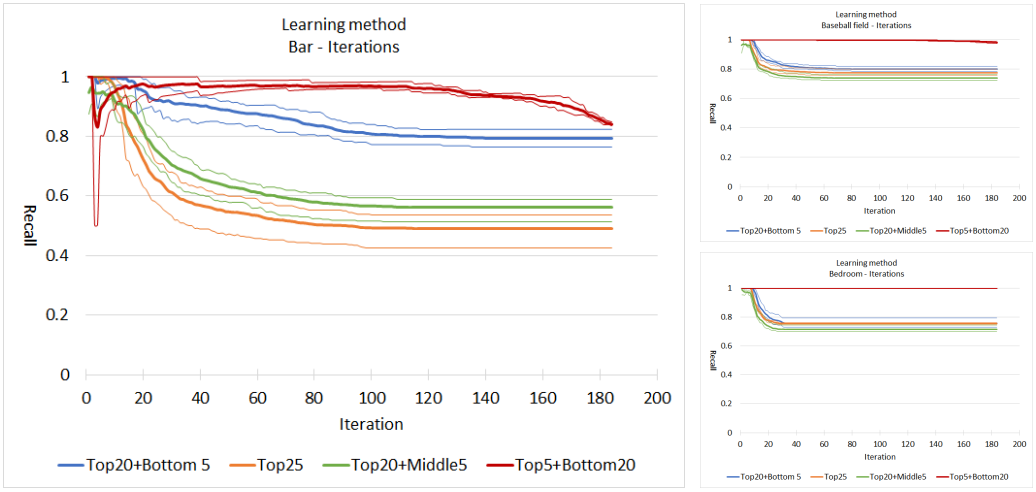


Figure 6.6: The recall rate on the three evaluation categories. The recall is slightly higher for those settings that continually present some of the *bottom* images in each iteration.

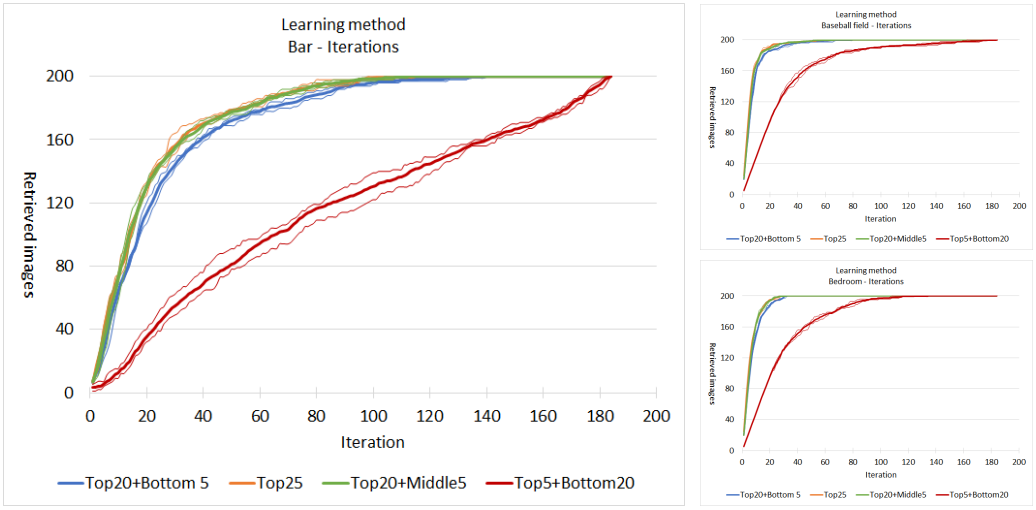


Figure 6.7: The number of retrieved images over iteration that the different settings had while classifying the search spaces for the three evaluation categories. Note how the *Bar* category is more difficult than the other ones to retrieve relevant images from.

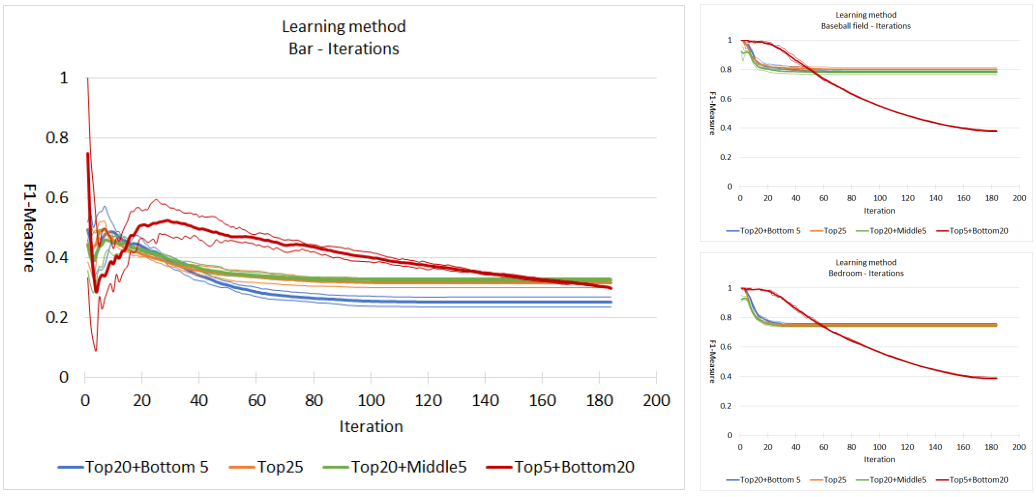


Figure 6.8: The F1-measure that the different settings had classifying the search spaces for the three evaluation categories. An harmonic mean of the precision and the recall.

image retrieval the model needs to have a high retrieval rate of relevant images early on which causes the *Top5+Bottom20* setting to not meet the preferences. To select between the remaining three settings one can recall what was mentioned in Chapter 1: Investigation material needs to be retrieved in a quick manner and labeled correctly. In other words as few false negatives as possible. In all three categories; the *Top20+Bottom5* setting had a higher recall rate on the search space throughout the evaluations (see Figure 6.6). Which means that the first setting is the most appropriate one to use in the following evaluations.

benchmarks

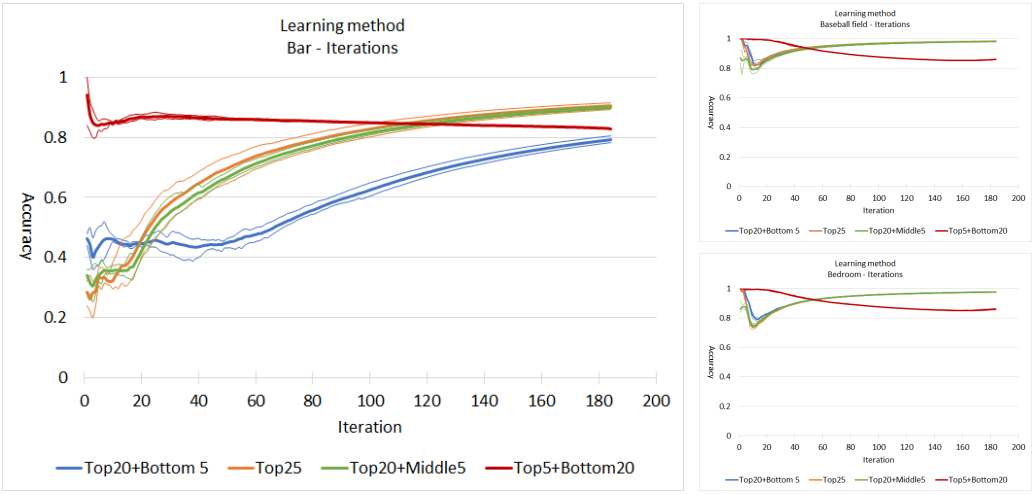


Figure 6.9: The accuracy that the different settings had classifying the search spaces for the three evaluation categories.

6.1.2 Limiting search space

This parameter benchmark is described in Section 5.3.2.3 and evaluates the settings **All images**, **Threshold**, **Early stopping** and **Both rules**.
The metrics used in this evaluation is presented in Section 5.3.2. Recall that the effectiveness of the model during search iterations is also measured in terms of time taken and how many images that are handled each iteration in order to find the best set of images to present to the user. Also that the performance of the model is measured on an evaluation set asover the entire search space.

four

processed

set as well as over

6.1.2.1 Evaluation set

When measuring the performance on the evaluation set, none of the stopping conditions affected the learning rate in any direction. The evaluation set was classified in a similar manner ny all the settings throughout the whole benchmark. Resulting in that all settings have about the same values on all metrics. Which is for instance visible when inspecting the F1-measure that is seen in Figure 6.3. Since all settings followed the same pattern the presentation of recall rateand accuracy is omitted in this section. However the general trend of the measurements can be observed in what is referred to as *Top20+Bottom5* in Section 6.1.1.1.

by

remove “that is seen”

rate, precision and

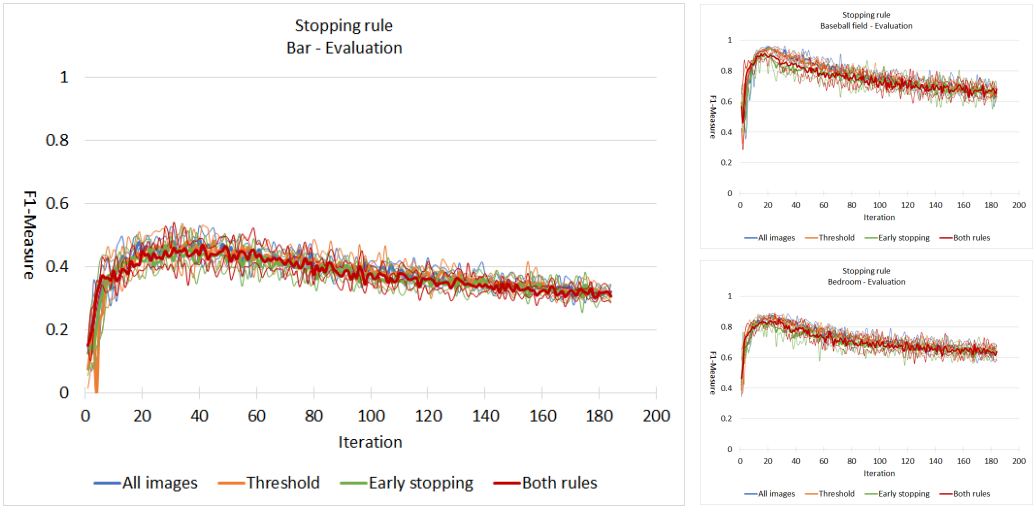


Figure 6.10: The harmonic mean of recall and precision, F1-measure, read on the evaluation set over iterations. All of the settings had performed similarly throughout the evaluation.

split up this sentence.

Since the concept was learned equally independently of which of these stopping conditions that were used, the classification performance on the search space becomes more important as well as how much time that is spent calculating distances to data points each search iteration.

6.1.2.2 Search space

check all the emphasises in this section.

...precision, when prediction the search space, using the stopp...

But since => Since

During the search iterations, the condition *Early stopping* introduced a notable trend compared to the condition *Threshold* in terms of correctness. As seen in Figure 6.11, the precisionwhen using the stopping condition *Early stopping* gradually decreased, causing the precision of using both rules to decrease as well. But since the selection of images near the decision boundary on the negative side is delayed when using the early stopping condition, the material near the decision boundary is predicted correctly due to more knowledge of the data on the negative side. Thus causing the recall to stay higher than the other settings, as seen in Figure 6.12.

in precision figure: ...different settings WHEN classifying the search...

in precision figure: remove emphasis.

Bake this equation.

rephrase the Which means part.

F1-measure levels?

If no stopping condition is used the total number of handled images, when the total search space is initially 4600 images, is 425500(see Equation 6.1). The total number of handled images for the different settings of stopping conditions in relation to this number can be seen in Table 6.1. Which means that the reduction of total handled images when combining the stopping conditions is $\approx 80 - 85\%$ on this dataset. Yet the F1-measurerecordings were only reduced with 20-25% according to the result in Figure 6.13.

$$\text{Handled images}_{max} = \sum_{i=1}^{4600/25} 25i = \sum_{i=1}^{184} 25i = 425500 \tag{6.1}$$

Table-caption: Equation bake

When inspecting the number of handled images by the different settings per iteration in Figure 6.14 it becomes clear how the reduction of handled images could vary so

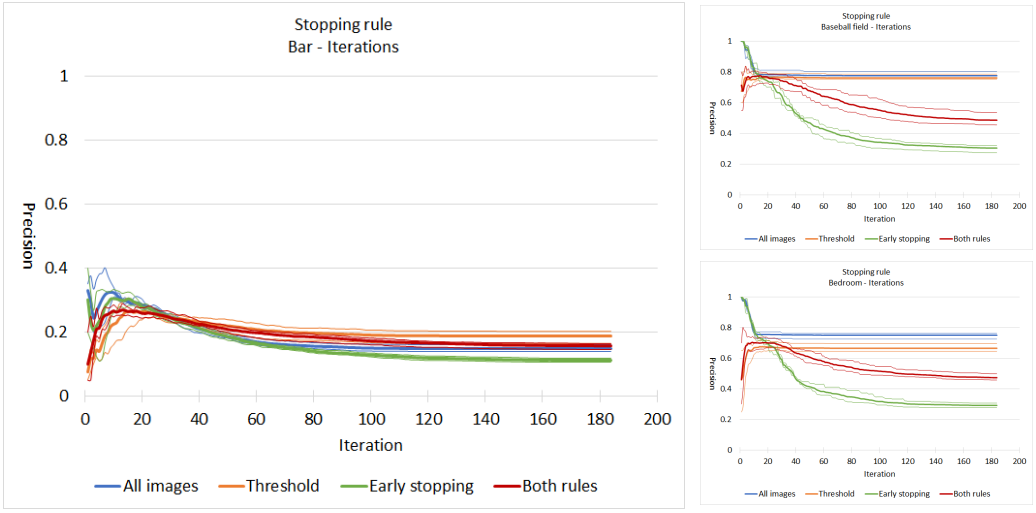


Figure 6.11: The precision of the different settings classifying the search spaces of the three evaluation categories. The *early stopping* condition appear to have a negative impact on precision.

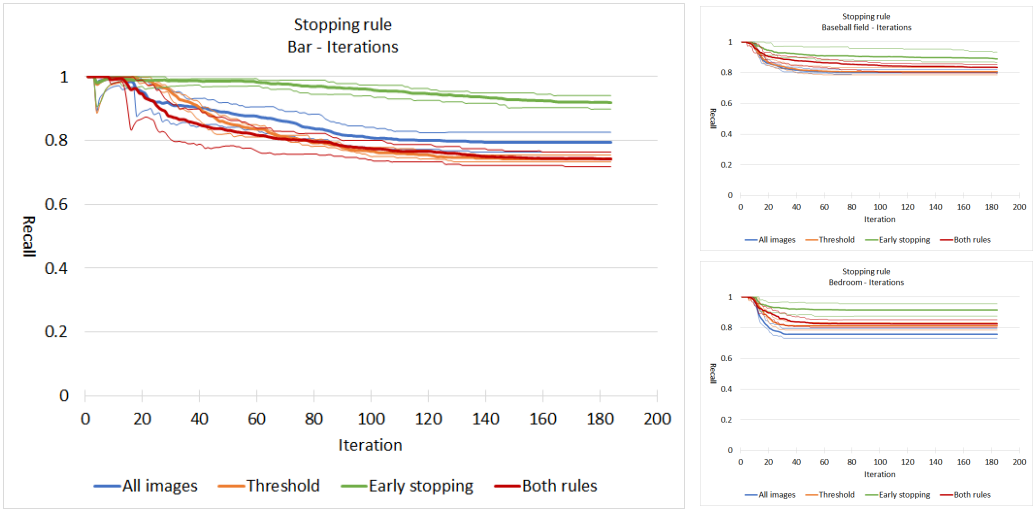


Figure 6.12: The recall rate of the different settings classifying the search spaces of the three evaluation categories. The condition *Early stopping* may have a positive effect on recall.

Set	All images	Threshold	Early stopping	Both
Bedroom	1	0.4168	0.2918	0.1522
Bar	1	0.2537	0.5717	0.1862
Baseball field	1	0.3843	0.2770	0.1491

Table 6.1: The average of total images that are handled during a search for each setting. The ratios of the value in Equation 6.1 are presented. The values are the average of five evaluations for each setting.

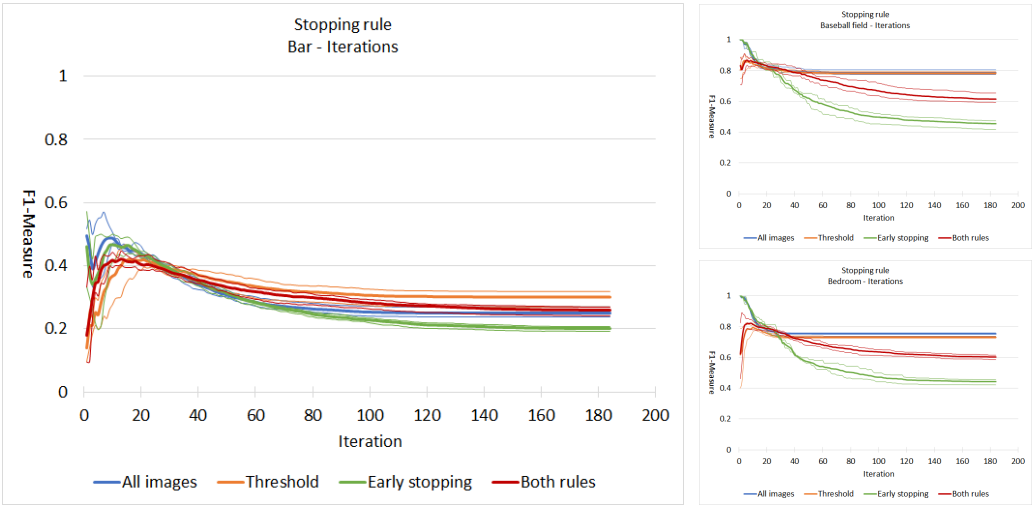


Figure 6.13: The F1-measure that the different settings had when classifying the search spaces of the three evaluation categories.

much between the classes when using each condition solely. In the evaluation of retrieving the category *Bar* the model finds more images that are predicted as relevant in a more dense manner. Which is why the early stopping condition does not evaluate as truefor most of the early iterations and why threshold setting stops exploring earlier every iteration than in the other two evaluation classes. The two stopping conditions are so disjunct in terms of when they cometrue and therefore the combination of the two actually reduce the number of handled images in the search of all three categories.

rephrase, is not fulfilled?

categories

rephrase: is fulfilled?

handled image Figure: fewer images handled => processing fewer images

notable

In terms of time taken there is a noticable improvement when using a stopping condition instead of handling all the images in the search space. The time that an iteration takes varies a lot for the condition *Early stopping*, see Figure 6.15, while the time taken for the threshold condition is just about the same throughout the entire search. But depending on which category that the algorithm is looking for the total time spent searching for the two conditions seems to vary compared to each other and combining the two conditions results in less time spent in all evaluations according to the measurements in Figure 6.16. On average, the total time reduction when using both stopping conditions compared with classifying all images every iteration was $\approx 60\%$.

time taken image: terms of time taken in all categories.

totaltime image: ratios of handled images => ratios of processed images

check empasises of section

In the same line as the F1-measure results, the number of retrieved images was a bit lower for the condition *Early stopping* (see Figure 6.17). When searching for the category *Baseball field* the early stopping condition seem to delay the full retrieval of the category by about 80 iterations. But when retrieving material from the category *Bar* the third setting follows the trend of the other ones and retrieves the relevant images at the same pace. Which could be caused by the same reason that was observed in Figure 6.14; the *Early stopping* setting often samples material until no more unique images are found. It is however clear that the setting *Both* follows the

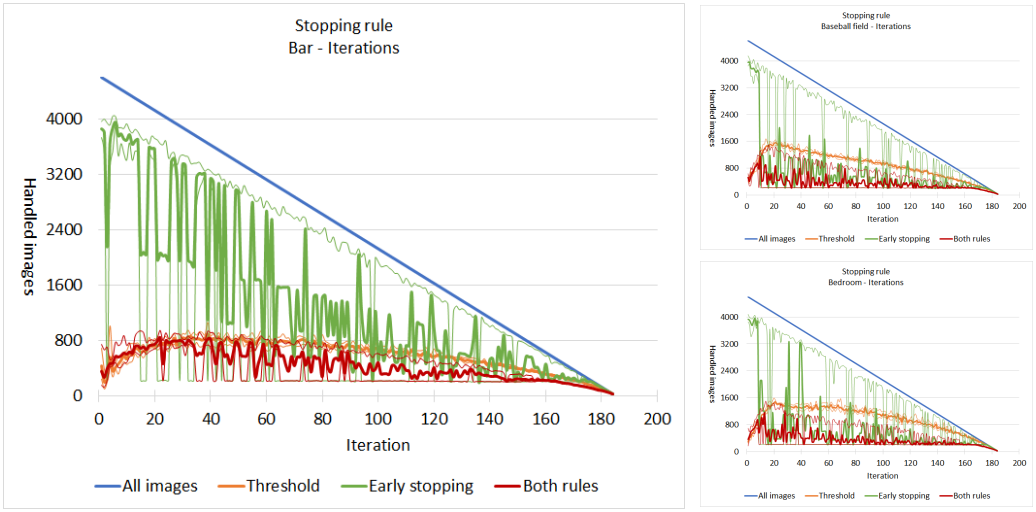


Figure 6.14: The number of handled images every iteration for the different settings. Having both stopping conditions results in fewer images handled in all three categories.

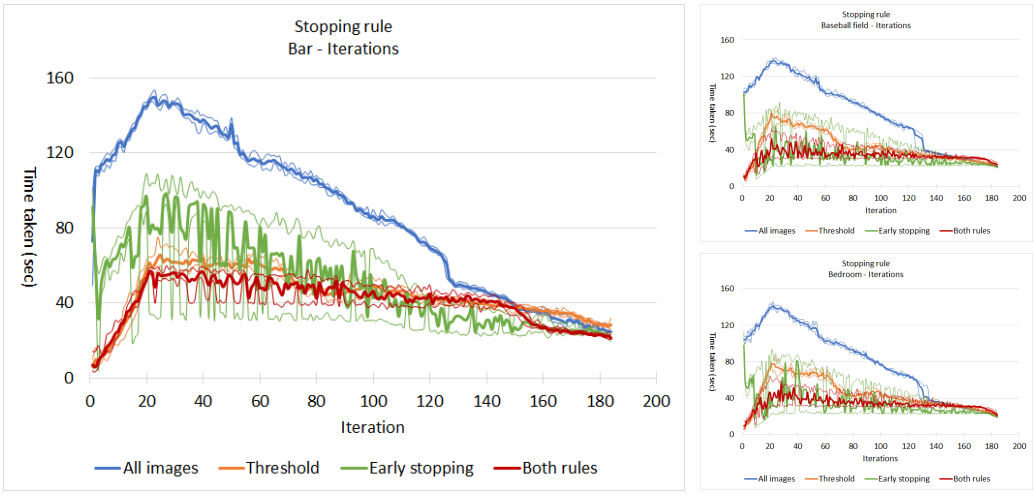


Figure 6.15: The time taken every iteration for the different settings. Having both stopping conditions results in the lowest peak in terms of time taken, in an iteration, in all three categories.

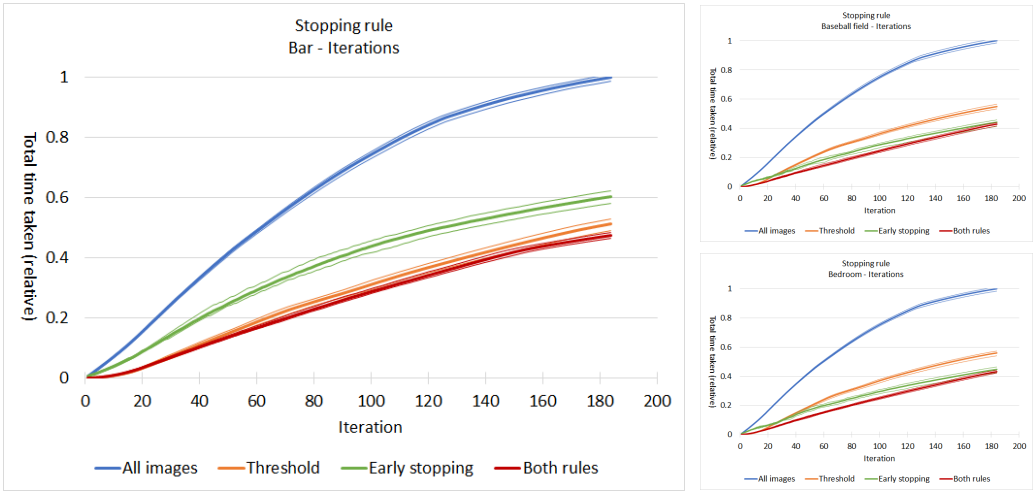


Figure 6.16: The total time taken for the different settings. There is a slight correlation with the ratios of handled images presented in Table 6.1 and the time taken ratios in these graphs.

relevant

setting *Early stopping* in terms of how manyimages that are retrieved at a certain iteration.

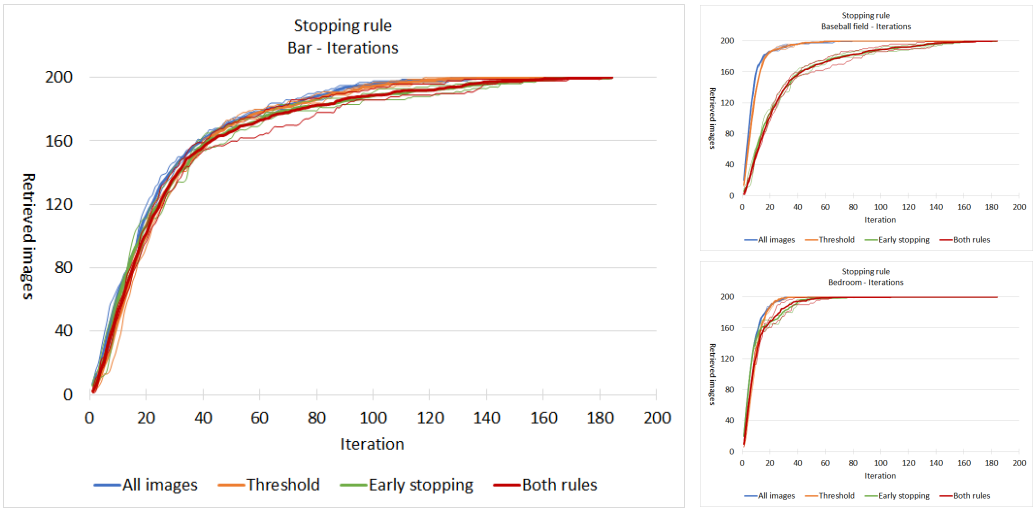


Figure 6.17: The retrieval rate that the different settings had when classifying the search spaces for the three evaluation categories. The retrieval rate of the setting with the *Early stopping* condition and the setting without stopping condition is higher than the other two settings. As seen in Figure 6.14, the two settings, *Early stopping* and *All images*, handled about the same amount of images in the initial rounds.

retrieved images
image: entire
caption is wrong.

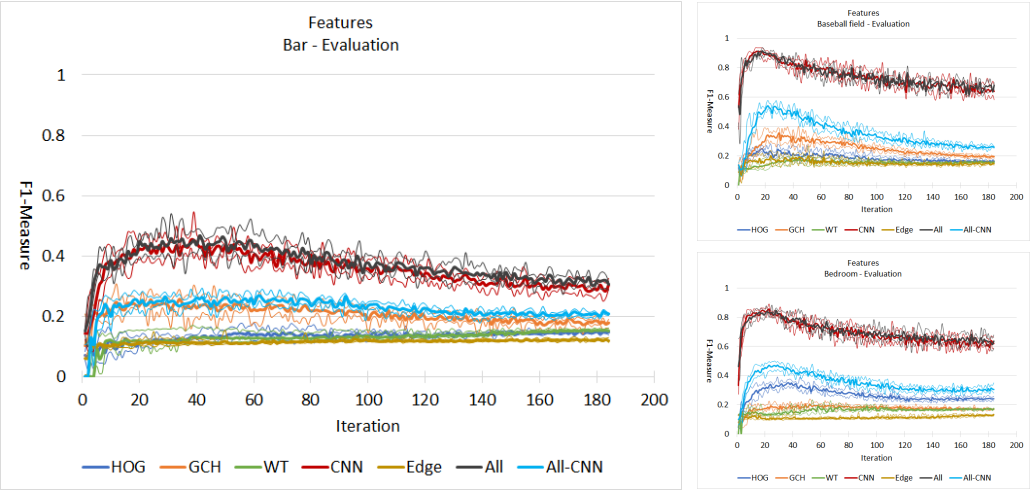


Figure 6.18: The F1-measure read on the evaluation set over iterations. Combining the information in different feature descriptors seems to give equally good or even better results.

6.1.3 Feature descriptors

This evaluation is described in Section 5.3.2.4 and a presentation of the evaluated settings can be found there. The settings described are **HOG**, **GCH**, **WT**, **CNN**, **Edge**, **All** and **All-CNN**.

Thisevaluation is performed with the complete set of stopping conditions that wasevalu in Section 6.1.2 and the material that is presented in each iteration is, as the evaluation in Section 6.1.1.2 suggests, a combination of the *top-20* images and the *bottom-5*.

Just like the previous parameter benchmarks the performance is measured over the entire search space as well as how anevaluation set is categorized.

The

with both of the stopping conditions

were

a separate

6.1.3.1 Evaluation set

Throughout all the metrics that were measured during the evaluation the settings that used *CNN* feature descriptors considerably outperformed the other settings. A trend that is distinguishable in both the F1-measure (Figure 6.18) and the accuracy (Figure 6.19) of the two settings that incorporate the *CNN* feature descriptor. There was a slight difference between using only the *CNN* feature vector as a descriptor and combining all the feature descriptors.

f1-measure caption: good or better results compared to the best single descriptor

The value of combining different feature descriptors become more clear when omitting the results of the fourth and sixth setting and only inspecting the remaining four feature descriptors and the combination of those. The F1-measure of the remaining five settings can be seen in Figure 6.20. Which single feature descriptor that performs the best depends on which category that isthe target. For the category *Bedroom* a better result is given when using the *HOG* descriptor while for the other two categories the target concept is more distinguishable with the *GCH* descriptor. But more importantly; the combination of the four descriptor performs better or just as

that the target is

when using the

check emphasis

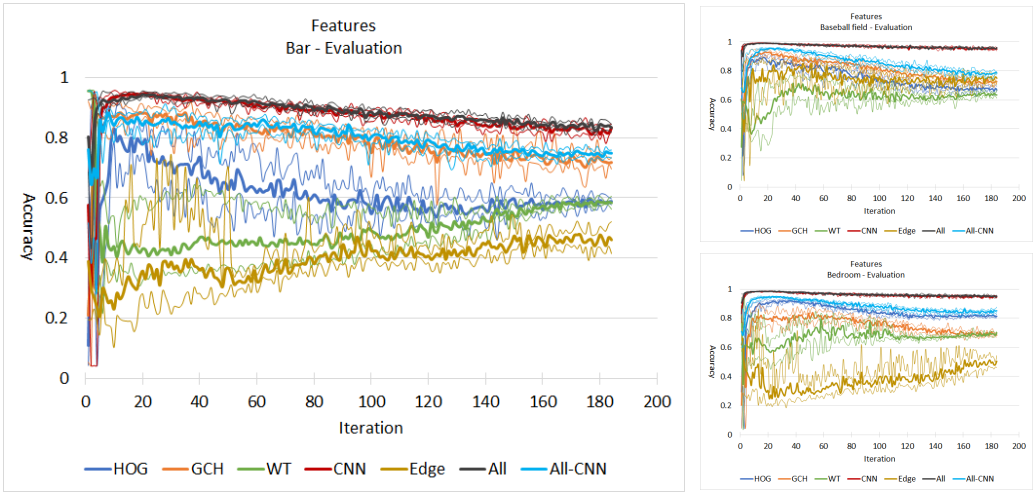


Figure 6.19: The accuracy of the different settings on the three evaluation sets. The *CNN* feature descriptor performs better as an information vector than the other descriptors.

good as the best single descriptor.

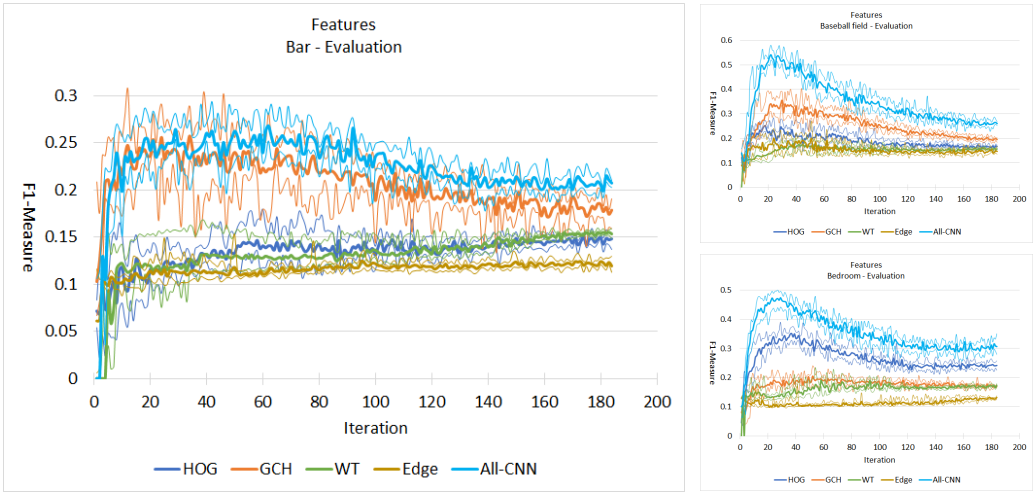


Figure 6.20: A closer look on the F1-measure of all the settings that are not handling feature vectors derived from a neural network. The combination of feature descriptors results in equally good or better results.

f1-measurenocnn caption: good or better results compared to the best single descriptor

6.1.3.2 Search space

As mentioned in Section 6.1.3.1 the different target categories of the data sets are more distinguishable when using the *CNN* activation vector during classification. This is also evident in the F1-measure (Figure 6.21) and accuracy (Figure 6.22) when classifying the search space. More interestingly the positive effect of combining the first order classifiers become more clear in these measurements. When inspecting

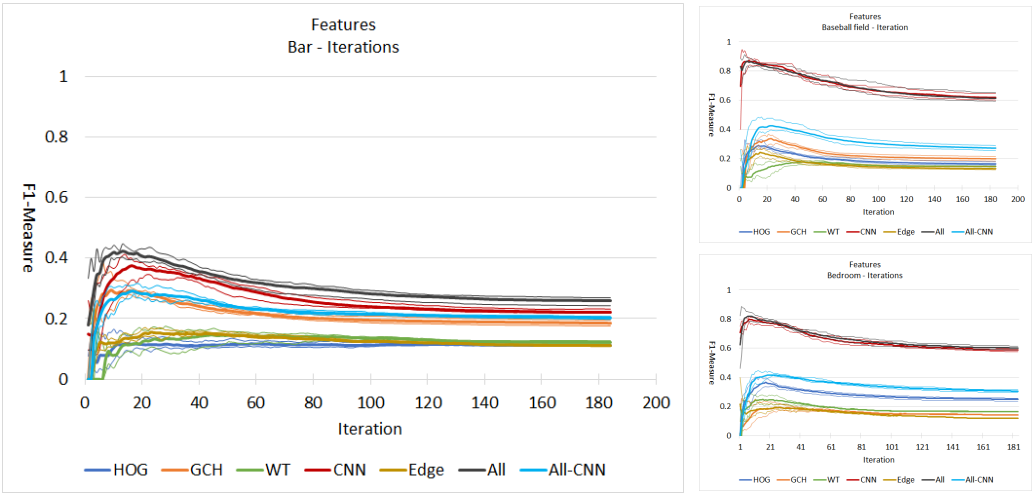


Figure 6.21: The F1-measure that the different settings had when classifying the search spaces of the three evaluation categories. The F1-measure on the *Bar* category is higher when combining all the feature descriptors compared with solely using the CNN feature descriptor.

the *Bar* category in the two figures the *All* setting outperforms the setting that only uses the *CNN* activation vector as a feature descriptor.

All the different settings of featuresdo seem to be able to present some distinguishability between the different categories. With the exception of the initial rounds of searching for the category *Baseball field* with the setting *WT*, where the results are in line with selecting images at random. The number of retrieved images per at a certain iteration is presented in Figure 6.23. Here the superiority of using the *CNN* activation vector becomes even more clear: When retrieving the bedroom class, the settings that use thisfeature descriptor have retrieved the entire target category at iteration 80 while the other settings can not retrieve the entire class until the search space is depleted.

feature descriptors

remove “per”

use the CNN feature descriptor

retrieval image: end: ...easier to retrieve.

The results of the setting that uses all the feature descriptors are considerably high in terms of pace of retrieving imagaes as well as with the F1-measure and the accuracy. But combining feature descriptors does come with the drawback of fitting more classifiers each iteration, causing the total time taken to grow accordingly. The total time spent computing predictions depended on how many and which classifiers that were being used and by using all six classifiers the time taken grew accordingly. In Figure 6.24 it is notable that combining all five descriptors compared to only combining four descriptors causes the computation time to grow to almost the double.

remove being?

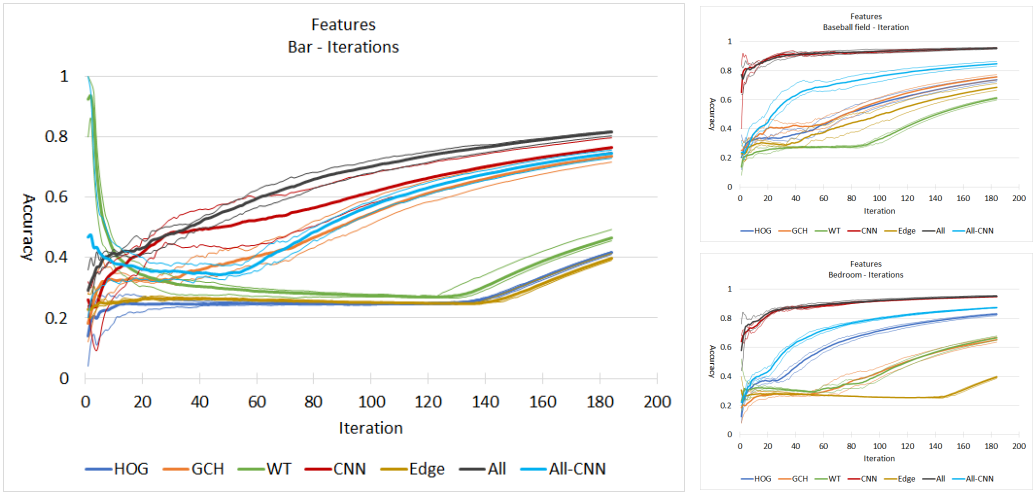


Figure 6.22: The accuracy of the different settings when classifying the search spaces of the three evaluation categories. As in Figure 6.21, there seems to be have a positive effect of combining feature descriptors.

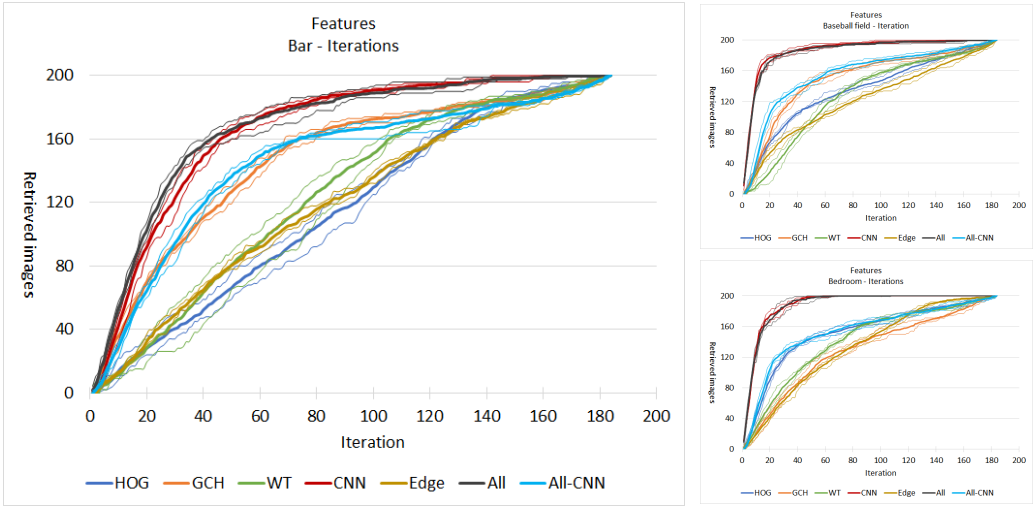


Figure 6.23: The number of retrieved images that the different settings had when classifying the search spaces. By introducing the *CNN* feature descriptors to the classifier the concepts of the categories seem to be easier to learn.

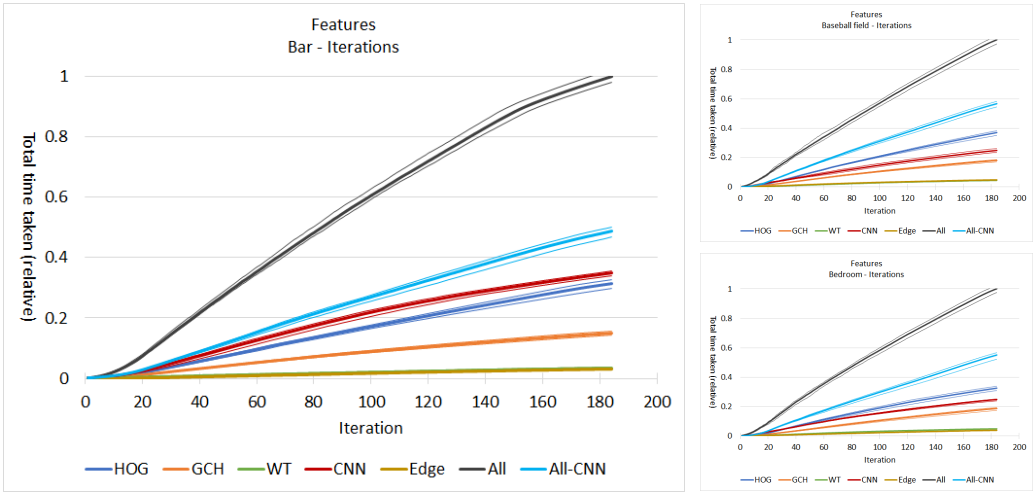


Figure 6.24: The total time taken for the different settings. The more feature descriptors that are used the more time it takes to fit all classifiers.

total time image: end: ... take to train the classifying system.

Given the performance of the *CNN* setting one might consider only using this setting in the final evaluation. But as stated in Section 5.3.2 the classifier is implementedas intended throughout the parameter benchmarks with the exception for this evaluation.

implemented => used

for => of?

6.1.4 Training data

The results of the evaluation described in Section 5.3.2.5 are presented here. The settings of the evaluation are as described in Table 6.2.

are listed in

Setting	**Training**	**Predefined training set** *(relevant+irrelevant)*
1A	Only the first iteration	5+5
1B	Only the first iteration	5+50
1C	Only the first iteration	22+484
1D	Only the first iteration	250+250
2A	Every iteration	5+5
2B	Every iteration	5+50
2C	Every iteration	22+484
2D	Every iteration	250+250
3	Every iteration	0+0

Table 6.2: The different settings evaluated in this benchmark. Deeper explanation found in Section 5.3.2.5.

The metrics used for this evaluation are presented in Section 5.3.2 and are as mentioned performed on an evaluation set and the search space. But the number of settings in this evaluation was higher than anticipated. Due to the number of different settings in this evaluation the presentation of the metrics will only consist the average value of the five runs of each setting. When the graphs included minimum and maximum values for each setting the data became much harder to understand and close to impossible to draw any conclusions from.

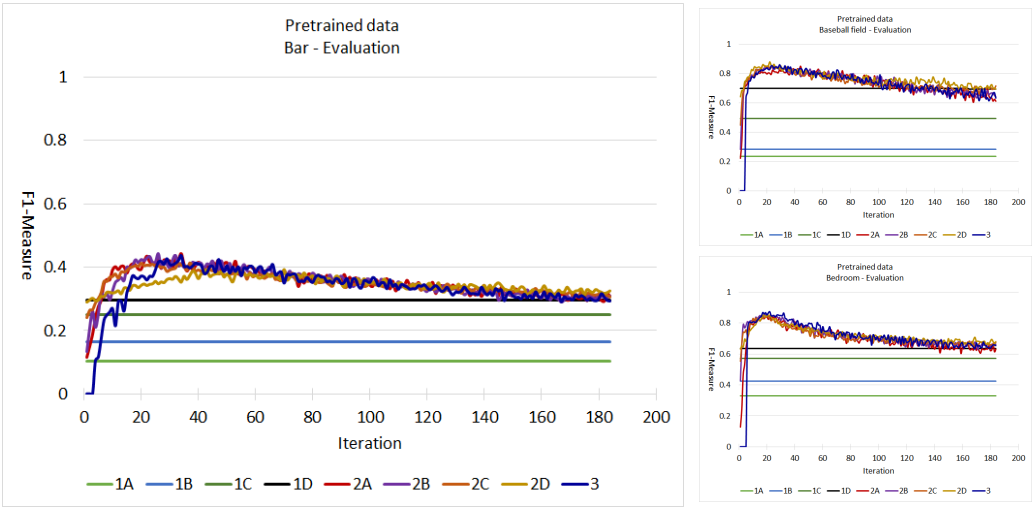


Figure 6.25: The F1-measure read on the evaluation set over iterations. Settings that use relevance feedback perform differently every iteration.

The model will during the evaluation use all of the stopping conditions described in Section 5.2.2.3, present the *top-20* images and *bottom-5* images in the end of every iteration as proposed in Section 6.1.2.2 and the classifier uses the full set of feature descriptors as described in Section 5.3.2.

6.1.4.1 Evaluation set

only use a

Naturally the performance of the settings that only usespredefined training data when classifying the evaluation set is same throughout all iterations. As expected the performance increases when having more data to begin with. In terms of F1-measure (seen in Figure 6.25) the performance of smaller training data sets are superseded by the performances of bigger training data sets.

use training set data

The settings that use dataretrieved by relevance feedback did however outperform the settings that did not. In the end of the search both the F1-measure and the accuracy where about the same for setting *3* as for the setting *1D*. But at around iteration 25 in the settings that use relevance feedback (settings *2A-2D* and *3*) have performance peaks that are considerably higher than at the end of the search, which

is noticeable

is visiblein Figure 6.25 and Figure 6.26.

f1-figure: remove last sentence (Settings that...)

The biggest difference between the settings *2A-2D* and the setting *3* was the initial three to six iterations when the setting with predefined data could actually perform a search. Causing the performance of setting *3* to be considerably lower than forthe other settings. But after those iterations the performance continuously rose up to be in level with the the settings *2A*, *2B*, *2C* & *2D*. Eventhough setting *2D* had a predefined data set of 250+250 relevant and non-relevant images their performance on the evaluation set turned out to be level. Yet it took the setting *2D* a few more iterations to be on par with the settings *2A-2C* and *3*.

insert “that during”

for => the performance of

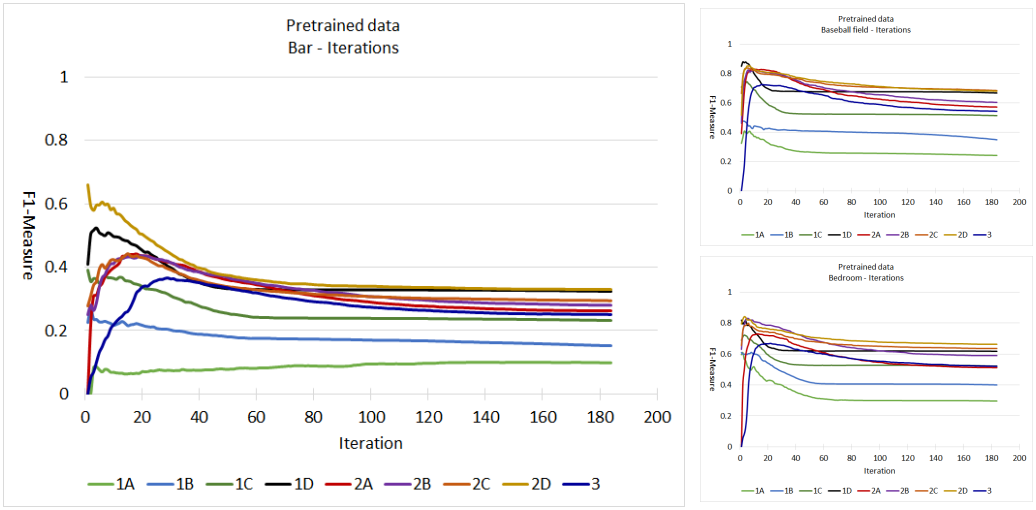


Figure 6.27: The F1-measure that the different settings had when classifying the search space. Having more data implies that the results become better.

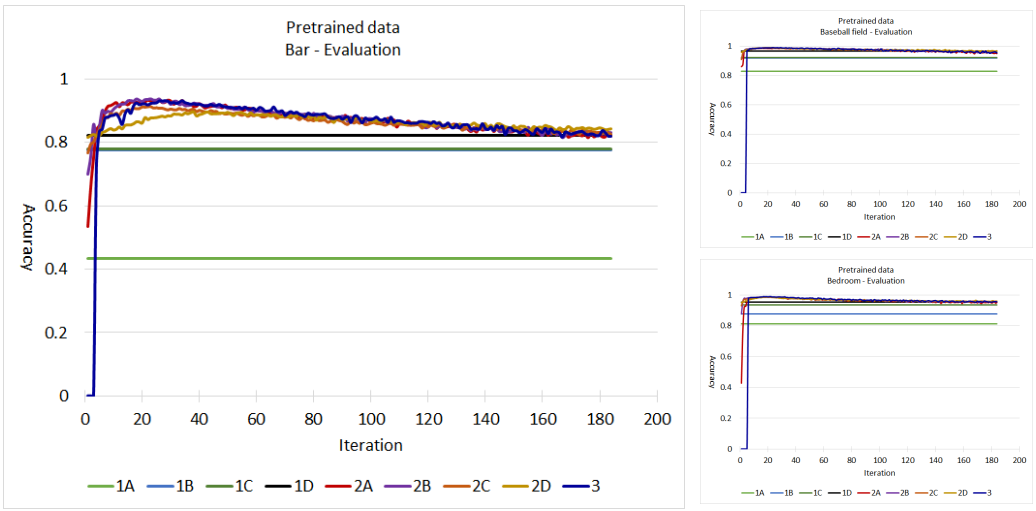


Figure 6.26: The accuracy on the evaluation sets. Using only relevance feedback (setting *3*) achieves higher results than having more relevant images in the predefined training set than in the search space (setting *1D*) between iteration 10 and 80.

accuracy figure: Using only training set data from relevance

accuracy figure: rephrase "than having more relevant images in the predefined training set than in the search space" => a predefined training set, that has more relevant images than the search space.. (1D)

is => as

results that were observed when classifying the evaluation set. (remove sec ref.)

use traning set data extracted

6.1.4.2 Search space

The results of measuring how the different settings classified the search space were not as indicating isthe results presented in Section 6.1.4.1. The metrics precision, recall, F1-measure and accuracy gave relatively inconclusive results. The values of the F1-measure of the evaluation (seen in Figure 6.27) do however imply some trends. In the first couple of iterations of the search the settings that use data extracted from relevance feedback begin to improve their results while the other settings begins with a high performance that shortly after decreases and then stabilizes.

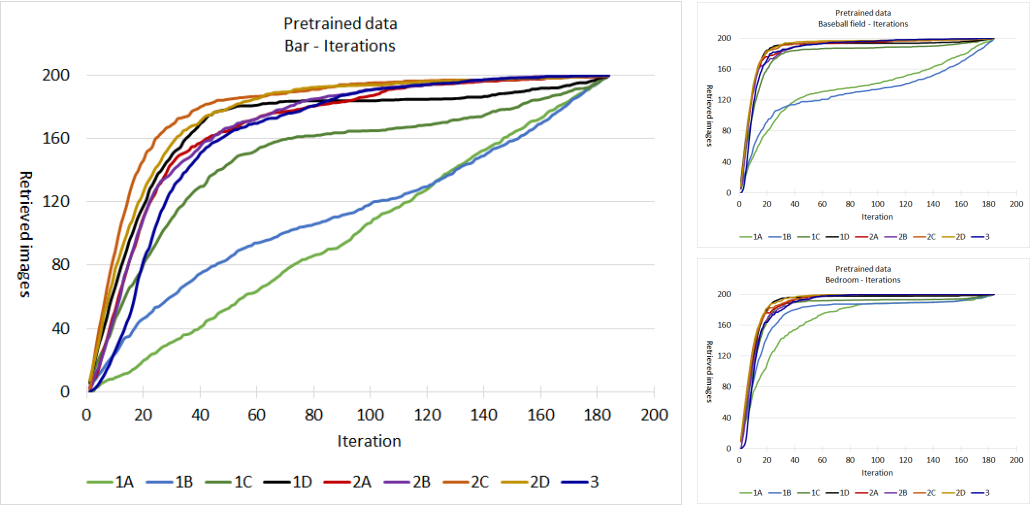


Figure 6.28: The number of retrieved imaged over the search iterations. Solely having 5 relevant images as training data (as in setting *1A* and setting *1B*) results in the incapability of retrieving the full content of the search space until it is depleted.

In terms of how well the different settings perform these measurements do not imply anything else than the more training data one has the better the result gets. The number of retrieved images after a certain iteration, as visualized in Figure 6.28, reflects the different sizes of traning data they had. By comparing the results of setting *1D* with setting *2D* when searching for the category *Bar* in this figure, one can see how the data from relevance feedback improved the result of retrieving relevant material.

In the intended scenario, the search space is unlabeled and is therefore categorized while the data is presented. In this setting the time taken to predict the category of the images should be relative to how long it takes to correct poorly predicted categories, which is not covered by the scope of the thesis. The time taken for each of the settings to categorize the search space relative to eachotheris presented in Figure 6.29. The impact of refitting the classifiers each iteration truly becomes clear in this figure as well as how the size of the training set causes fitting to take longer.

each other

training the classifier to take longer

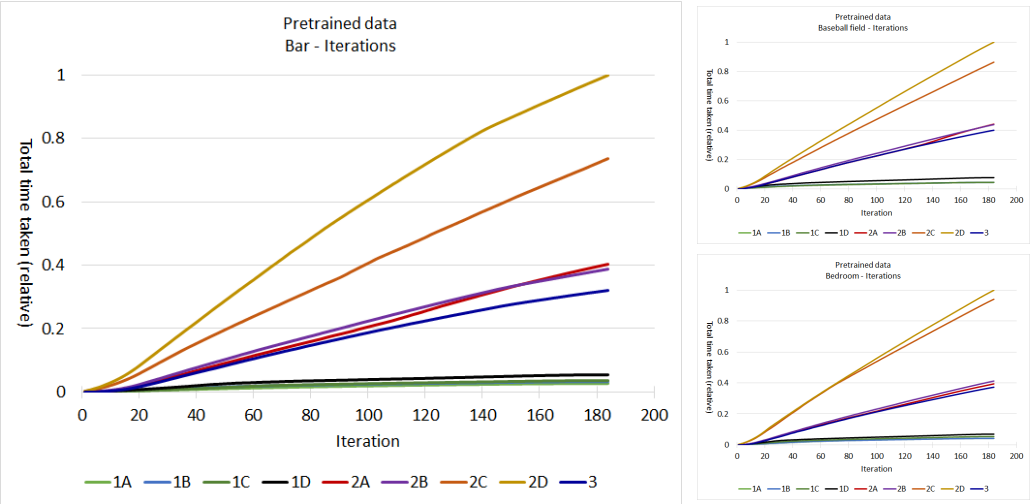


Figure 6.29: The total time taken for the different settings of this evaluation. There is a huge time difference between the refitting the controller once doing it every iteration.

time figure: the refitting => training

time figure: controller => classifier

time figure: once and doing it every iteration.

comparison evaluations

6.2 Study comparisons

This section covers the results of the study comparisonsdescribed in Section 5.3.3.

6.2.1 The Corel-1000 evaluation

The results presented in this section is in accordance with the evaluations described in Section 5.3.3.1. Recall that the evaluation was run with $t = 10$ categories, $m = 500$ different query sets per category and number of retrieved images $n = 20$.

are retrieved in accordance

is performed

The proposed model is adjusted to randomly sample the query data from the search space and makes sure that the query set and the test set are disjunct before an evaluation is performed. As mentioned in Section 5.3.3 the number of images that the proposed model has in the training set was not decided on beforehand but was instead selected by inspecting the results of the similar studies. To do so the same evaluation was run with 6different traningset sizes: $2+2$, $3+3$, $4+4$, $5+5$, $7+7$ and $10+10$ relevant and irrelevant images.

six

query

The results of the evaluation on different sizes of query sets for the thesis model is presented in Table 6.3 as well as in Figure 6.30 (*IICTVC* is short for the name of the thesis). When having the smallest number of query images and looking for the categories *Africans* and *Beaches* resulted in a precision that is below 10%. But when having the largest number of query images the precision on the same categories is above 80%.

an average retrieval precision

average retrieval precision

In both the Table 6.3 and the Figure 6.30 it becomes clear that the more query images the better the retrieval of the intended category becomes. This is somewhat in line with correlation that a separate image retrieval paper has found. The more

iictvc figure: final sentence. rephrase, "changes logarithmicly"

it is notable that

Category	Average precision (%); $n = 20$					
	$r + i$ (IICTVC query set)					
	$2 + 2$	$3 + 3$	$4 + 4$	$5 + 5$	$7 + 7$	$10 + 10$
Africans	05.66 ± 0.08	21.57 ± 0.31	48.03 ± 0.35	69.55 ± 0.49	84.59 ± 0.41	94.58 ± 0.06
Beaches	07.18 ± 0.12	14.63 ± 0.21	43.16 ± 0.41	48.62 ± 0.21	73.96 ± 0.26	81.57 ± 0.06
Buildings	11.53 ± 0.15	37.63 ± 0.70	68.10 ± 0.38	84.40 ± 0.25	95.77 ± 0.08	99.11 ± 0.01
Buses	61.64 ± 0.51	78.88 ± 0.65	97.65 ± 0.08	99.63 ± 0.01	100.0 ± 0.00	100.0 ± 0.00
Dinosaurs	99.06 ± 0.02	99.54 ± 0.02	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
Elephants	34.78 ± 0.49	63.56 ± 0.44	89.32 ± 0.17	93.45 ± 0.08	98.27 ± 0.01	99.24 ± 0.00
Flowers	49.79 ± 0.57	61.30 ± 0.64	96.54 ± 0.06	91.45 ± 0.12	99.66 ± 0.00	99.75 ± 0.00
Horses	63.02 ± 0.22	76.85 ± 0.57	92.50 ± 0.06	94.49 ± 0.05	97.73 ± 0.01	98.33 ± 0.01
Mountains	13.25 ± 0.15	41.41 ± 0.72	73.58 ± 0.69	87.83 ± 0.33	97.38 ± 0.02	99.02 ± 0.00
Food	10.01 ± 0.35	35.53 ± 0.42	68.73 ± 0.67	84.91 ± 0.17	93.97 ± 0.08	98.18 ± 0.01
Average	35.59 ± 0.01	53.09 ± 0.04	77.76 ± 0.02	85.43 ± 0.02	94.13 ± 0.01	96.98 ± 0.00

n - number of images retrieved.
r - number of relevant images in query set.
i - number of irrelevant images in query set.

Table 6.3: Precision when retrieving 20 images from the Corel-1000 set with different query sets. Note that the variance decreases as the query set increases from six images to more.

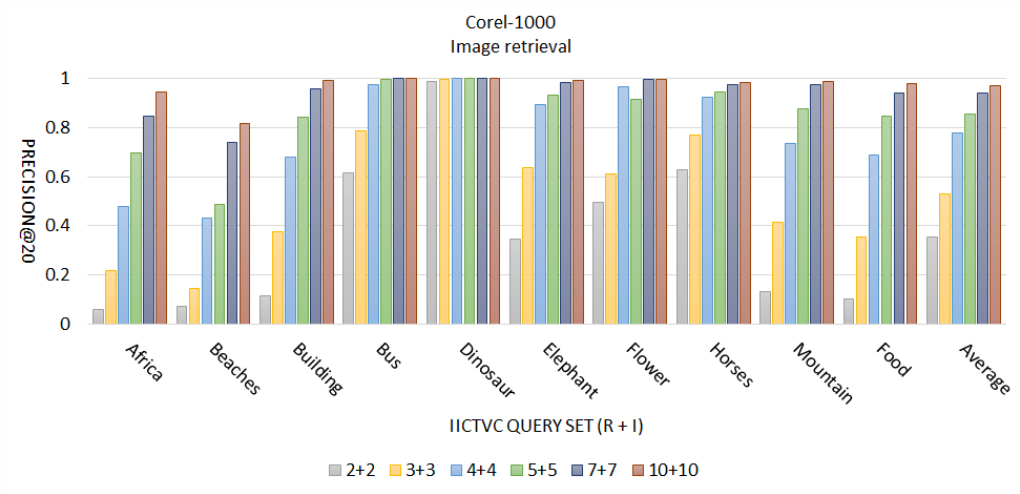


Figure 6.30: Precision when retrieving 20 images from the Corel-1000 set with different query sets presented in a graph. As the query sizes become larger than 4+4 the different precisions begin to increase much slower.

Category	Average precision (%); $n = 20$					
	Wang, James Z. et.al. [14]	Subrahmanyam, M et.al. [15]	Nagaraja, S et.al. [16]	ElAlami, M.A. [17]	IICTVC (3 + 3)	IICTVC (5 + 5)
Africans	47.50	69.75	56.00	**72.60**	21.57	69.55
Beaches	32.50	54.25	**60.00**	59.30	14.63	48.62
Buildings	33.00	63.95	58.00	58.70	37.63	**84.40**
Buses	36.30	89.65	94.00	89.10	78.88	**99.63**
Dinosaurs	98.10	98.70	98.00	99.30	99.54	**100.0**
Elephants	40.00	48.80	66.00	70.20	63.56	**93.45**
Flowers	40.20	92.30	88.00	**92.80**	61.30	91.45
Horses	71.90	89.45	78.00	85.60	76.85	**94.49**
Mountains	34.20	47.30	58.00	56.20	41.41	**87.83**
Food	34.00	70.90	54.00	77.20	35.53	**84.91**
Average	46.77	72.50	71.00	76.10	53.09	**85.43**

n - number of images retrieved.

Table 6.4: Precision when retrieving 20 images from the Corel-1000 the different studies and the proposed model with two different query set sizes.

query images they used the better results theyreceived until a certain point on their data set; using 6 images as a seed for their search resulted in a detection rate of 60%, with 10 images the rate turned out to be around 85% [54]. In the paper detection rate is described as the ratio of relevant images that are perceived by the model as relevant. Very much like how precision is described in this evaluation and the those values are somewhere near the result given by the test of different query sizes: Using 3 relevant and 3 irrelevant images as query set resulted in a precision of $\approx 53\%$ and using 5 relevant and 5 irrelevant images resulted in a precision of $\approx 85\%$. What was interresting with selecting 2 relevant and 2 irrelevent images as query images in a randomly generated manner was that in some cases the model could not find any categorial characteristics and therefor often found 0 relevant images during the retrieval. Yet once a good query set was constructed the number of relevant images that was retrieved rose up to 15-20 .

Given the results of the inspected papers, mentioned in Section 5.3.3.1, the query sizes that are used to compare with were set to 3+3 and 5+5 relevant and irrelevant images. The comparison of the *precision@20* evaluations can be seen in Table 6.4 and Figure 6.31. When the query size of the proposed model is set to $3 + 3$ the model performs slightly better than the model described in [14], yet slightly worse than the models in [15], [16] and [17]. But when the query set of the proposed model is set to $5 + 5$ the performance of the proposed model supersedes the other ones.

Seeing that the proposed model performs this well compared to other CBIR methods is a good result. Yet it is still important to remember that having more query images reduces the risk of only having outliers as trainingdata. This was noticed more often when the query set size was set to $2 + 2$ and $3 + 3$, but considerably less when the query sets were larger than that. The other studies handles all images in the same category as an equal influence which gives the proposed model too much of an upper hand in this evaluation. Which is interesting since the proposed model is designed to have much larger training/query sets in order to function as intended.

images that were uses

results were recieved

dataset

six

for a performed search

ten

Konstig och ofullständig mening.

average retrieval precision

"the those"=> the

":"=> "."

Intriguingly when selecting a query set of 2 relevant...

two

two

remove "as query images"

zero

But as soon as a good...

images. Formulera om meningen.

query

The effect of this was

remove "too much of "

But is still interesting since the proposed

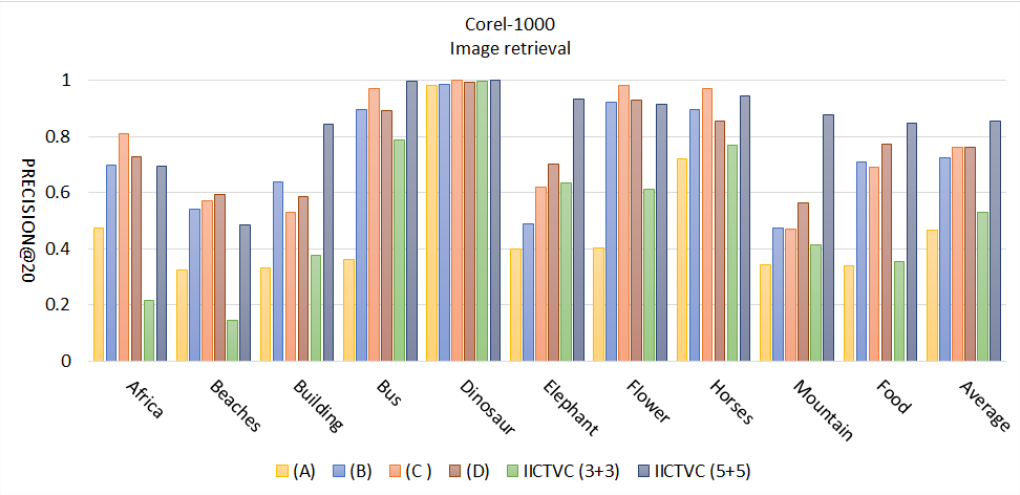


Figure 6.31: Precision when retrieving 20 images from the Corel-1000 the different studies and the proposed model with two different query set sizes had. **(A)**: Wang, James Z. et.al. [14], **(B)**: Subrahmanyam, M et.al. [15], **(C)**: Nagaraja, S et.al. [16] and **(D)**: ElAlami, M.A. [17].

7
Conclusion

The evaluations that werepresented in Chapter 6 were strictly based on how the solution was modeled in the beginning and was presented in Chapter 5. With this in mind the achieved results and measurements are discussed here. In this chapter the possible improvements on current model are discussed as well as how knowledge extracted from the results can be used in other system designs.

results of the

are

based on how the proposed model, described in Section 5.2, initially was designed.

of the proposed model

7.1 Discussion of results

Throughout the whole parameter benchmarking evaluation the model had the best performance in the first half of a search. This becomes the most clear when inspecting the metrics during the learning method evaluation, Section 6.1.1.1. If the training set that is acquired at the turnpoint – when the model stops to predict images as relevant – would the performance of the model on the entire search then become better? The difference would probably not be notablewhen inspecting the measurements on the the search space since the model has already started to predict the rest of the search space as irrelevant images. But this would however mean that training set behaves well on the independent evaluation set and the rest of the search space can be predicted as non-relevants without taking up additional computational power. In every evaluation there has however always existed outliers that were found after this point but in return became false negatives. Our iterative model didnot prevent this from happening. In the same section one setting had a lower count of false negatives; the setting called *Bottom20+Top5*. When evaluating the performance on the search space the recall for this method was higher than for the other ones. This would reduce the chanceof any false negatives, but using this setting would result in not reaching the turnpoint mentioned earlier due to positive predictions until the end of the evaluation. Meaning that the matching module has to keep on adapting to the given data. On top of that the retrieval rate of this setting will always be lower than for the other settingswhich makes it less of an appropriate image retrieval method. To search the entire search space, as the model in evaluation in Section 6.1.1 did, was never intended. But since the focus of the thesis was to create a light weight iterative model that can retrieve images, optimization became a second priority. Then again, the intention was to model something that reduced timeper image so the two stopping conditions were applied. Since the combined pruning of the search space barely affected how the pace of how the matching module learned, more work could be done to reduce the time spent. it would be interesting to see what would happen to the performance (as well as the total time taken) if the matching module would, as mentioned earlier, stop adaptingto data after a certain point. In Figure 6.10 this turnpoint is around iteration 20 and 40 depending on the category. And inspection the time taken seen in Figure 6.16 this could mean a reduction from 0.5 of the total to $\approx 0.05 - 0.1$ of the total time taken.
Due to that when only using the activation vector from a CNN as a descriptor resulted in time reduction of about 50% compared to when using all the different

flip order of content.

the most clear?

measurements

add “is persisted,”

noticeable

makes the model perform well

some outliers

the turnpoint described earlier

in most cases

could in other words not

section has lost focus... reformulate.

risk

rate of retrieving relevant images

add“,”

setting for image retrieval.

lightweight

time spent

it? what? Stor bokstav.

the classifier in the matching

its decision boundary to

make this value prettier.

the complete set of

descriptors as input to the deep SVM, it might seem tedious to use the full set of descriptors when the performance were about the same for the two settings. But it would however be interesting to see how well the model would perform if one would combine the activation vectors from multiple networks that have been trained for different purposes. The measurements presented in Section 6.1.3 do however indicate that there are some improvements when using an ensemble of classifiers compared to only using its parts. Especially in Figure 6.20 where the the performance was notable increase by using the feature descriptors in unison.

performs

in fact

noticeably increased

In the final parameter benchmark, the training set evaluation in Section 6.1.4, it became clear that if the refitting of the classifier was omitted the time taken was drastically reduced (see Figure 6.29). But if the model only used data that is retrieved from relevance feedback, the model would perform equally well – or even better – at the evaluation set as a model with a predefined training set with 250 relevant and 250 irrelevant images (see Figure 6.25). Could this mean that after a number of iterations the proposed model could be used to find a subset of a category that defines the essence of it? I.e. removing the outliers from a category that, in terms of the currently selected feature descriptors, lies closer to another category. This could be used to not only fine-tune a training set intended for some specific image retrieval, but also to manifest as the semantic gap between human and machine learning. The outliers that are not included in such a search defines what either the model could not see or what the person has put in a category that does not fit the description by mistake.

the category

The productivity of the proposed model was also shows the evaluation that was done on the Corel-1000 set in Section 6.2. Even though the model is designed to improve the training set in an iterative manner, it behaved surprisingly well with small query sets. This procedure was never tested on the same set that the benchmark was done but it might give some interesting results. But it was interesting to see that the proposed model could produce an similar precision as the model in [17] when having a query set of 5 relevant and 5 irrelevant image, while the model described in that paper uses a training set of 900 images. Unfortunately this comparison is unfair to all parts. Having a small query set allowed the deep SVM to fit immediatly and classifying the a dataset is done rather quick compared to calculating individual distances. But having more than one image as a query allows the model to ignore outliers and can still find images that are defined as matches. Outliers that the papers [14], [15] and [16] hade to include in their results. This evaluation will however give future studies the possibility to compare with a multi image-based CBIR approach on the Corel-1000 set.

shown in

performed

dataset

benchmarks

performed on

doing so

It

the neural network model proposed in

images while the

Mention why it is unfair to the ANN model (test set is 100 images for that model.)

7.2 Future work

Just as having

hade!

Some of the possible improvements might have been mentioned previously in this thesis, but in this section all future work is listed and discussed. The different ideas of improvements is split up into two parts; how the model can be improved and possible usage areas that were not intended initially.

multi image query-based CBIR

are

as well as

7.2.1 Model improvements

The different model improvements included in this section are time investments that could have been done within a short time period but were omitted since they were not covered by the focus of the thesis.

a near future but

7.2.1.1 Scaling to bigger datasets

A future feature for the system would be to create a search algorithm that improves the way images are selected for each iteration based on previous predictions. As of now the procedure to select images is to randomly sample a predefined set size from the search space. Problemarises as these dataset becomes larger,and some form of sorting can minimize the time spent searching for the relevant images as well as improve the rate in which they are located. The algorithm can for example, either sort the material after how relevant it was predicted as in previous iterations or in how the folder structure of the database is composed. This would thus minimizing the size of the search space and allow larger parts of the unknown set to be evaluated faster as well as the material that is located in the same folders is ensured to be examined at some point.

problems arise

the datasets become

", and"=> ". Some form"

to be

viable

7.2.1.2 Improvements to the relevance feedback loop

The number of images presented to the user in each iterations is in the proposed model set to 20 images perceived as relevant and 5 as non-relevant. This is in no way based on previous studies or knowledge and is set after what *felt right* to the authors after some testing. To optimize the performance of the algorithm in conjunction with the user, considerations should be made in how many images should be presented each iteration. The number of images presented might be decreased of increased, depending on empiricalstudies, to enhance the performance and enhance the user experience.

In the proposed model the relevance feedback module, described in Section 5.2.1, only updates the search space with which images that are relevant and which are notwhen knowing exactly which images that were mistakenly perceived as relevant and non-relevant by the matching module. If this information was used it would be possible to "move" the decision boundary. One example is that it is possible to weight each datapointin order to make them more decisivein how the classifier makes decisions. If an image is falsely categorized as a non-relevant one, it could be inserted into the dataset as two points. If the *two* data points are close to a decision boundary this would increase the cost of ending a training session at this point. By putting more energy into using relevance feedback more heavily, the semantic gap between man and machine might be reduced.

As can be seen in the results the models performance will decrease as the dataset is exhausted of relevant images. As this is a controlled environment this can not be taken as true for unknown datasets. If presented to unknown data, that the algorithm does not find any relevantin a couple of iterations it does not mean that relevant images are exhausted. There is however much to gain with cutting the retraining short as mentioned in Section 7.1. To stop the retraining and assume

In the proposed model the number of images predicted as relevant and irrelevant is set to 20 and 5 respectively.

add "empirically" somewhere

emperical =>"result of studies"

improve

when=> ". This when knowning"

data point

impactful

emph

data points instead of one

remove emph

remove "more heavily"

remove "can be"

the performance of the model

number of relevant images in the search space becomes exhausted.

that

seems to be the approach that is more time-efficient and maintains the performance of the model

all remaining images are non-relevant when the algorithm only finds non-relevant images is both time efficient and help to keep the performance. There need to be some feature that activates the retraining again if the user were to label images as relevant again as well.

7.2.1.3 Selection of feature descriptors

feature descriptors were chosen

based on

add ","

of the feature descriptor benchmark

setting that uses the

settings that use the

the margin is far more narrow when presented

. One could for example combine

that have been trained

remove "and purpose"

In the

classifiers

down

, in Section 6.1.3,

remove "information"

in this section.

In this thesis, not too much effort was put into selecting the best suited feature descriptors since the case is of the general nature and instead chose feature descriptors after the idea that in order to create a more general classifiermore feature descriptors would be added. There are however feature descriptors that perform better or worse in these situations.
In the results, Section 6.1.3, the CNN activation vector is seen outperforming the other feature descriptors by a large margin, but can be narrowly seen to receive some performance improvement when presented towards the *bar* category. Instead of mixing weak and strong learners,one could combine the activations of the final fully connected layers of several neural networks which were trained towards different datasets and purpose into an ensemble. The idea being to cover as many possible features as possible within the different networks.
In The proposed model, all classifiers are always present and their predictions are always taken into account. If there would be cases where some of these classifier do not contribute or even might reduce performance by misclassifying, it might be beneficial to shut them off. During the parameter benchmarksome of the feature descriptors were found lacking in performance on certain sets. The improvement being to reduce time calculating information that is of no use for the matching module by terminating certain parts of the classifier.

7.2.2 Miscellaneous usage areas

Besides from improving the proposed model by adding different features there are some possible usage areas outside of the domain of the thesis that were considered. These ideas are briefly described here.

7.2.2.1 Dataset improvement

previously described

the following two sentences are F-d.

As mentioned in Section 7.1 the images that represent the essential bits of a concept are selected early on during a search space exploration and the outliers that are selected later on seem ruin classification results. This information could be used to design a smaller training set specialized for a certain cause. In the scenario of a system that often uses a large dataset in order to solve some machine learning problem or an image retrieval problem, the feature found in this system could be applied. The possible functionality of the classifier could stop refitting the decision boundary after a point, described in Section 7.2.1.2, could be used to improve an already existing dataset. If the training set that achieves the best results is extracted it might perform just as well as the original large dataset.

Bibliography

[1] D. Rubino. The future of science as public service. [Online]. Available: https://www.dhs.gov/news/2011/03/14/future-science-public-service

[2] M. B. Kara Nance, Brian Hay, "Digital forensics: Defining a research agenda," in *Proceedings of the 42nd Hawaii International Conference on System Sciences*. IEEE, 2009.

[3] S. Garfinkel, "Digital forensics research: The next 10 years," in *The Digital Forensics Conference*, no. 7. Elsevier, 2010, pp. 64–73.

[4] M. K. Kundu, M. Chowdhury, and S. R. Bulò, "A graph-based relevance feedback mechanism in content-based image retrieval," *Knowledge-Based Systems*, vol. 73, pp. 254–264, 2015.

[5] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and J. Ramesh, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[6] S. R. Athira Mohanan, "A survey on different relevance feedback techniques in content based image retrieval," *International Research Journal of Engineering and Technology*, vol. 04, no. 02, pp. 582–585, 2017.

[7] D. B. Judd, G. Wyszecki *et al.*, "Color in business, science, and industry," 1975.

[8] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[9] A. Alzu'bi, A. Amira, and N. Ramzan, "Semantic content-based image retrieval: A comprehensive study," *Journal of Visual Communication and Image Representation*, vol. 32, pp. 20–54, 2015.

[10] H.-D. Cheng, X. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.

[11] S. Midha, R. Vijay, and S. Kumari, "Analysis of rgb and ycbcr color spaces using wavelet transform," in *Advance Computing Conference (IACC), 2014 IEEE International*. IEEE, 2014, pp. 1004–1007.

[12] Q. Su, Y. Huang, and J. Peng, "Coldimage: Contrast and luminance distribution for content-based image retrieval," in *Image Analysis and Signal Processing (IASP), 2011 International Conference on*. IEEE, 2011, pp. 143–146.

[13] N. Prajapati and A. K. Nandanwar, "Edge histogram descriptor, geometric moment and sobel edge detector combined features based object recognition and retrieval system."

[14] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

[15] M. Subrahmanyam, Q. J. Wu, R. Maheshwari, and R. Balasubramanian, "Modified color motif co-occurrence matrix for image indexing and retrieval," *Computers & Electrical Engineering*, vol. 39, no. 3, pp. 762–774, 2013.

[16] S. Nagaraja and C. Prabhakar, "Low-level features for image retrieval based on extraction of directional binary patterns and its oriented gradients histogram," *arXiv preprint arXiv:1503.03606*, 2015.

[17] M. E. ElAlami, "A new matching strategy for content based image retrieval system," *Applied Soft Computing*, vol. 14, pp. 407–418, 2014.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[22] D. Tao. The corel database for content based image retrieval, dct-research. [Online]. Available: https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval

[23] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," *arXiv preprint arXiv:1610.02055*, 2016.

[24] M. Hassaballah, A. A. Abdelmgeid, and H. A. Alshazly, "Image features detection, description and matching," in *Image Feature Detectors and Descriptors*. Springer, 2016, pp. 11–45.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[26] S.-K. Pavani, D. Delgado, and A. F. Frangi, "Haar-like features with optimally weighted rectangles for rapid object detection," *Pattern Recognition*, vol. 43, no. 1, pp. 160–172, 2010.

[27] M. Koskela and J. Laaksonen, "Convolutional network features for scene recognition," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 1169–1172.

[28] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[29] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*. McGraw-Hill New York, 1995, vol. 5.

[30] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International journal of image processing (IJIP)*, vol. 3, no. 1, pp. 1–11, 2009.

[31] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[32] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[33] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[34] M. A. Kadam, K. S. Kadge, S. S. Mane, S. P. Naikwadi, and M. V. Kulloli, "Study and review of various image classification methods for diabetes mellitus detection," 2016.

[35] J.-P. Vert, "Kernel methods in genomics and computational biology," 2005.

[36] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the ninth ACM international conference on Multimedia*. ACM, 2001, pp. 107–118.

[37] P. Cunningham and J. Carney, "Diversity versus quality in classification ensembles based on feature selection," in *European Conference on Machine Learning*. Springer, 2000, pp. 109–116.

[38] A. Krogh, J. Vedelsby *et al.*, "Neural network ensembles, cross validation, and active learning," *Advances in neural information processing systems*, vol. 7, pp. 231–238, 1995.

[39] M. Cord and P. Cunningham, *Machine learning techniques for multimedia: case studies on organization and retrieval*. Springer Science & Business Media, 2008.

[40] B. Grofman, G. Owen, and S. L. Feld, "Thirteen theorems in search of the truth," *Theory and Decision*, vol. 15, no. 3, pp. 261–278, 1983.

[41] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S. Y. Bang, "Constructing support vector machine ensemble," *Pattern recognition*, vol. 36, no. 12, pp. 2757–2767, 2003.

[42] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[43] J. Chen, C. Wang, and R. Wang, "Using stacked generalization to combine svms in magnitude and shape feature spaces for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2193–2205, 2009.

[44] X.-Y. Wang, H.-Y. Yang, Y.-W. Li, W.-Y. Li, and J.-W. Chen, "A new svm-based active feedback scheme for image retrieval," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 43–53, 2015.

[45] C. S. A. S. Mizanur Khondoker, Richard Dobson and D. Stahl, "A comparison of machine learning methods for classification using simulation with multiple real data examples from mental health studies," *International Research Journal of Engineering and Technology*, vol. 03, no. 01, pp. 814–820, 2016.

[46] S. S. M. S. P. N. M. V. C. K. Manaswi A. Kadam, Kiran S. Kadge, "Study and review of various image classification methods for diabetes mellitus detection," *Statistical Methods in Medical Research*, vol. 25, pp. 1804–1823, 2016.

[47] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, pp. 249–268, 2007.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn:

Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[49] O. R. et al, "Imagenet large scale visual recognition challenge," *arXiv:1409.0575v3 [cs.CV]*, 2015.

[50] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[51] C. R. Johnson, "Deep learning algorithms compete at imagenet challenge," February 2015, [Online; posted 19-February-2015 01:06 PM]. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1325712

[52] M. Thomsen, "Microsoft's deep learning project outperforms humans in image recognition," February 2015, [Online; posted 18-February-2015 08:15 AM]. [Online]. Available: https://www.forbes.com/sites/michaelthomsen/2015/02/19/microsofts-deep-learning-project-outperforms-humans-in-image-recognition/#6826a7ce740b

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[54] L.-J. Li and L. Fei-Fei, "Optimol: automatic online picture collection via incremental model learning," *International journal of computer vision*, vol. 88, no. 2, pp. 147–168, 2010.

A
Complete list of categories in the dataset MIT places205

Letter	Scenery name
A	Abbey, Airport terminal, Alley, Amphitheater, Amusement park, Aquarium, Aqueduct, Arch, Art gallery, Art studio, Assembly line, Attic, Auditorium, Apartment building
B	**Badlands, Ballroom, Bamboo forest, Banquet hall, Bar, Baseball field, Basement, Basilica, Bayou, Beauty salon, Bedroom, Boardwalk, Boat deck, Bookstore, Botanical garden, Bowling alley, Boxing ring, Bridge, Building facade, Bus interior, Butchers shop, Butte, Bakery**
C	Cafeteria, Campsite, Candy store, Canyon, Castle, Cemetery, Chalet, Classroom, Closet, Clothing store, Coast, Cockpit, Coffee shop, Conference center, Conference room, Construction site, Corn field, Corridor, Cottage garden, Courthouse, Courtyard, Creek, Crevasse, Crosswalk, Cathedral, Church
D, E	Dam, Dining room, Dock, Dorm room, Driveway, Desert (sand), Desert (vegetation), Dinette, Doorway, Engine room, Excavation
F	Fairway, Fire escape, Fire station, Food court, Forest path, Forest road, Formal garden, Fountain, Field (cultivated), Field (wild)
G, H	Galley, Game room, Garbage dump, Gas station, Gift shop, Golf course, Harbor, Herb garden, Highway, Home office, Hospital, Hospital room, Hot spring, Hotel room, Hotel (outdoor)
I, J, K, L	Ice cream parlor, Iceberg, Igloo, Islet, Ice skating rink, Inn, Jail cell, Kasbah, Kindergarden classroom, Kitchen, Kitchenette, Laundromat, Lighthouse, Living room, Lobby, Locker room
M, N	Mansion, Marsh, Martial arts gym, Mausoleum, Medina, Motel, Mountain, Mountain snowy, Music studio, Market, Monastery, Museum, Nursery
O, P, Q	Ocean, Office, Office building, Orchard, Pagoda, Palace, Pantry, Parking lot, Parlor, Pasture, Patio, Pavilion, Phone booth, Picnic area, Playground, Plaza, Pond, Pulpit
R	Racecourse, Raft, RailRoad track, Rainforest, Reception, Residential neighborhood, Restaurant, Restaurant kitchen, Restaurant patio, Rice paddy, River, Rock arch, Rope bridge, Ruin, Runway
S	Sandbar, Schoolhouse, Sea cliff, Shed, Shoe shop, Shopfront, Shower, Ski resort, Ski slope, Sky, Skyscraper, Slum, Snowfield, Staircase, Supermarket, Swamp, Stadium (baseball), Stadium (football), Stage, Subway station, Swimming pool
T	Television studio, Topiary garden, Tower, Train railway, Tree farm, Trench, Temple (east Asia), Temple/ (south Asia), Track, Train station
U, V, W, X, Y, Z	Underwater (coral reef), Valley, Vegetable garden, Veranda, Viaduct, Volcano, Waiting room, Water tower, Watering hole, Wheat field, Wind farm, Windmill, Yard

Table A.1: A complete list of all the categories in the set MIT places205. The sceneries in bolded text were used in the parameter benchmark.