

# Feature extraction for image selection using machine learning

**Matilda Lorentzon**

Master of Science Thesis in Electrical Engineering  
**Feature extraction for image selection using machine learning**

Matilda Lorentzon  
LiTH-ISY-EX--17/5097--SE

Supervisor: **Marcus Wallenberg**  
ISY, Linköping University  
**Tina Erlandsson**  
Saab Aeronautics

Examiner: **Lasse Alfredsson**  
ISY, Linköping University

*Computer Vision Laboratory  
Department of Electrical Engineering  
Linköping University  
SE-581 83 Linköping, Sweden*

Copyright © 2017 Matilda Lorentzon

## Abstract

During flights with manned or unmanned aircraft, continuous recording can result in a very high number of images to analyze and evaluate. To simplify image analysis and to minimize data link usage, appropriate images should be suggested for transfer and further analysis. This thesis investigates features used for selection of images worthy of further analysis using machine learning. The selection is done based on the criteria of having good quality, salient content and being unique compared to the other selected images. The investigation is approached by implementing two binary classifications, one regarding content and one regarding quality. The classifications are made using support vector machines. For each of the classifications three feature extraction methods are performed and the results are compared against each other. The feature extraction methods used are histograms of oriented gradients, features from the discrete cosine transform domain and features extracted from a pre-trained convolutional neural network. The images classified as both good and salient are then clustered based on similarity measures retrieved using color coherence vectors. One image from each cluster is retrieved and those are the resulting images from the image selection. The performance of the selection is evaluated using the measures precision, recall and accuracy. The investigation showed that using features extracted from the discrete cosine transform provided the best results for the quality classification. For the content classification, features extracted from a convolutional neural network provided the best results. The similarity retrieval showed to be the weakest part and the entire system together provides an average accuracy of 83.99%.



## Acknowledgments

First of all, I would like to thank my supervisor Marcus Wallenberg at ISY for expertise and support throughout the thesis work. I would also like to thank my examiner Lasse Alfredsson at ISY for valuable feedback. Also thanks to my supervisor Tina Erlandsson for the opportunity to do my thesis work at Saab Aeronautics, as well as for showing great interest in my work.

Last but not least, I would like to thank my family and friends for love, support and coffee breaks.

*Linköping, 2017*  
*Matilda Lorentzon*



---

# Contents

<b>Notation</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Aim . . . . .	1
1.3 Limitations . . . . .	2
<b>2 Related theory</b>	<b>3</b>
2.1 Available data . . . . .	3
2.2 Machine learning . . . . .	4
2.3 Support Vector Machines . . . . .	5
2.4 Histogram of oriented gradients . . . . .	7
2.5 Features extracted from the discrete cosine transform domain . . . . .	9
2.6 Features extracted from a convolutional neural network . . . . .	13
2.6.1 Convolutional neural networks . . . . .	13
2.6.2 Extracting features from a pre-trained network . . . . .	15
2.7 Color coherence vector . . . . .	16
<b>3 Method</b>	<b>17</b>
3.1 Feature extraction . . . . .	18
3.2 Predictor . . . . .	19
3.3 Similarity retrieval . . . . .	19
3.4 Evaluation . . . . .	20
3.5 Generation of training and evaluation data . . . . .	21
<b>4 Results</b>	<b>25</b>
4.1 Quality classification . . . . .	25
4.2 Content classification . . . . .	28
4.3 Similarity retrieval . . . . .	30
4.4 The entire system . . . . .	34
<b>5 Discussion</b>	<b>35</b>
5.1 Results . . . . .	35

---

5.1.1	Quality classification . . . . .	35
5.1.2	Content classification . . . . .	37
5.1.3	Similarity retrieval part . . . . .	37
5.1.4	The entire system . . . . .	38
5.2	Method . . . . .	39
5.3	Possible improvements . . . . .	39
<b>6</b>	<b>Conclusions</b>	<b>41</b>
	<b>Bibliography</b>	<b>43</b>

---

# Notation

## ABBREVIATIONS

<b>Abbreviation</b>	<b>Meaning</b>
DCT	Discrete cosine transform
SVM	Support vector machines
HOG	Histogram of oriented gradients
RGB	Red, green, blue
SSIM	Structural similarity
ROC	Receiver operating characteristic



# 1

---

## Introduction

### 1.1 Motivation

The collection of image data is increasing rapidly for many organisations within the fields of for example military, law enforcement and medical science. As sensors and mass storage devices become more capable and less expensive the data collection increases and the databases being accumulated grow larger, eventually making it impossible for analysts to screen all of the data collected in a reasonable time. This is why computer assistance becomes increasingly important and when searching by meta-data is impractical, the only solution is to search by image content. [5]

During flights with manned or unmanned aircraft, continuous recording can result in a very high number of images to analyze and evaluate. The images are assumed to be evaluated by automatic target recognition functions as well as image analysts on the ground and also by pilots during missions. The images may contain interesting objects like vehicles, buildings or people but most contain nothing of interest for the reconnaissance mission. A single target can often be found in multiple images which are similar to each other. The images can also be of different interpretation quality, meaning that properties like different lightning conditions and blur affect the user's ability to interpret the image content. To simplify image analysis and to minimize data link usage, appropriate images are suggested for transfer and analysis.

### 1.2 Aim

The aim of the master's thesis is to investigate which features in images that can be used to select images worthy of further analysis. This is done by implementing two classifications, one regarding quality and one regarding content. In the first classification images will be binarily classified as either good or bad depending on the image quality. In this report *good* and *bad* refers to the two quality classes. The images classified as *good* will

continue to the next classification where they will be binarily classified as either salient or non-salient depending on the image content. In this report *salient* and *non-salient* refers to the two content classes. The images classified as *salient* will continue to the next step where the final retrieval will be done depending on similarity measures. In the case where there is a set of images that are almost identical the image with the highest certainty of being *good* and *salient* will be retrieved. What is interesting content in an image depends on the use case and data set.

The master's thesis will answer the following questions:

- Can any of the provided feature extraction methods produce features useful for differentiating between good and bad quality images?
- Can any of the provided feature extraction methods produce features useful for differentiating between salient and non-salient content in images?
- Is it possible to make a good image selection using machine learning classifications based on both image content and quality, followed by a retrieval based on similarity measures?

### 1.3 Limitations

The investigation is limited to an example data set which is modified to fit the task. Bad quality images are limited to the distortion types described in section 3.5, which are added to the images. Similar images are retrieved synthetically from one image. The investigation is limited to only using one classification model for all classifications. The classifications and retrievals are done using one *salient* class at a time.

# 2

---

## Related theory

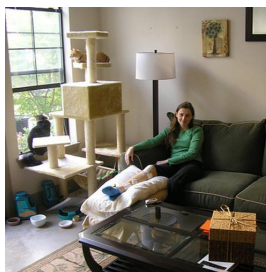
This chapter covers the related theory which supports the methods used in this thesis. Unless anything else is specified, the content of a paragraph is supported in the references specified at the end of the paragraph, without case specific modifications.

### 2.1 Available data

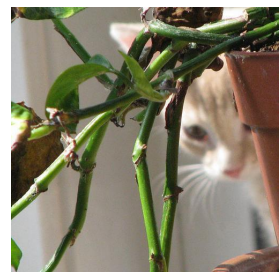
The data used is the COCO - Common Objects in Context [10] data set which contains 91 different object categories such as food, animals and vehicles. It contains many non-iconic images of the objects in their natural environment as oppose to iconic images which typically have a large object in a canonical perspective centered in the image. Non-iconic images contain more contextual information and the object in non-canonical perspectives. Figure 2.1 shows examples of iconic and non-iconic images from the COCO data set.



(a) Iconic image



(b) Non-iconic image

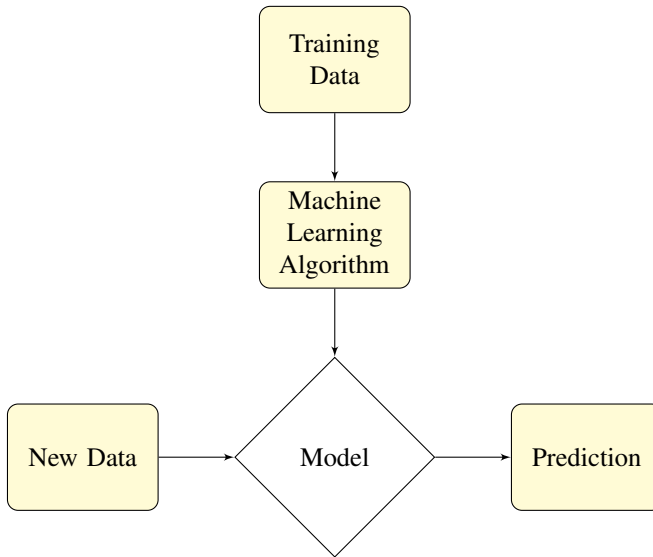


(c) Non-iconic image

**Figure 2.1:** Examples of images from the data set containing the object cat, (a) is an iconic image while (b) and (c) are non-iconic.

## 2.2 Machine learning

Machine learning is the concept of learning from large sets of existing data to make predictions about new data. It's based on creating models from observations called training data for data-driven decision making. The concept is illustrated by a flow chart in figure 2.2 where the vertical part of the flow is called the training part and the horizontal part is called the evaluation part. [18]



**Figure 2.2:** The concept of machine learning where a machine learning algorithm creates a decision model from training data. The model is then used to make predictions about new data. (Flow chart drawn according to [18])

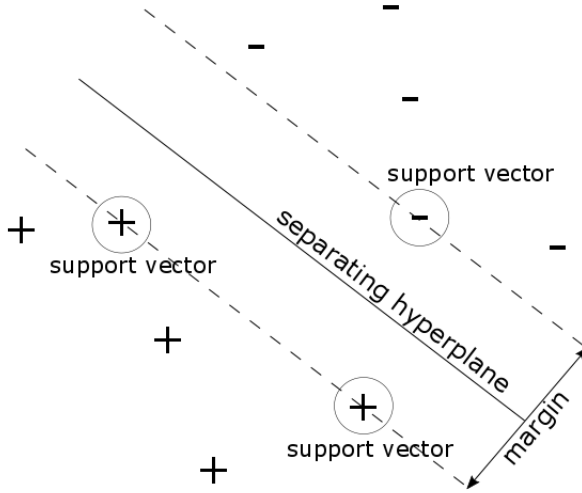
There are different types of machine learning models, this report focuses the one called supervised learning. In supervised learning the input training data have corresponding outputs and the goal is to find a function or model that correctly maps the inputs to the outputs. That is in contrast to unsupervised learning for which the input data has no corresponding output. The goal of unsupervised learning is to model the underlying structure or distribution of the input data to create corresponding outputs. [18] A common use of supervised machine learning is classification where the observations are labelled with classes and the prediction outputs are different classes. It can be described in a simple manner as finding the function  $f$  that fulfills  $Y = f(X)$ , where  $X$  contains the input observations and  $Y$  the corresponding output classes. With  $X$  and  $Y$  as matrices, the description becomes as follows:

$$\begin{bmatrix} \text{class}(\text{observation1}) \\ \text{class}(\text{observation2}) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = f \begin{bmatrix} \text{observation1} \\ \text{observation2} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad (2.1)$$

$Y$  is a column vector where each row contains the class of the corresponding rows in  $X$ . Each row in  $X$  corresponds to an observation which is represented by the values, also called features in its columns. These values can be measurements such as weight and height but when it comes to images, the compilation of the values in  $X$  becomes more complex. [14] Raw pixel values can be used as features for images but for other than simple cases, the representation is not descriptive enough, specially when working with natural images. The aim is to represent an image by distinctive attributes that diverse the observations from one class from the other. Therefore an important step when using machine learning on images is feature extraction. [7] In figure 2.2 the feature extraction is a big part of the first step in both the training part and the evaluation part. There are many methods for feature extraction, this thesis covers three of them: histogram of oriented gradients in section 2.4, features extracted from the discrete cosine domain in section 2.5 and features extracted from a pre-trained convolutional neural network in section 2.6

## 2.3 Support Vector Machines

Support vector machines (SVM) is a form of supervised machine learning model. By learning from provided examples -the training data- the model finds a function that couples input data to the correct output. The output for novel data can then be predicted by applying the retrieved function. SVM is often used for classification problems for which the correct output is the class the data belongs to. The model works by creating a hyperplane that separates data points from one class from those from the other class, with a margin as high as possible. The margin is the maximal width of the slab parallel to the hyperplane that has no interior data points. The support vectors which give the model its name are the data points closest to the hyperplane and therefore determine the margin. The margin and the support vectors are illustrated in 2.3.



**Figure 2.3:** Illustration of the hyperplane separating data points from two classes shown as + and -. The support vectors and the margin are marked. Figure drawn according to [11].

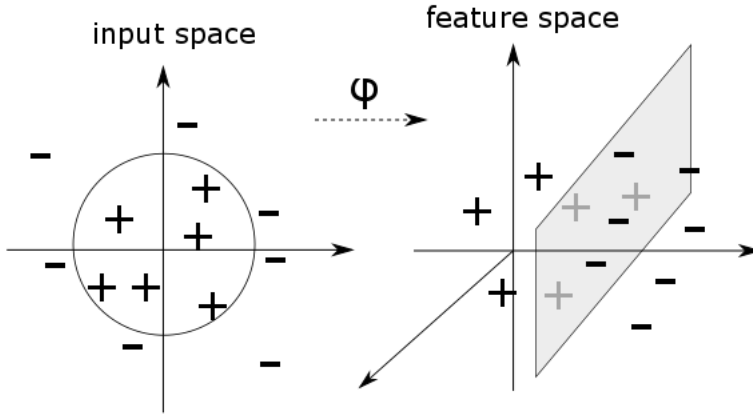
The data might not allow for a separating hyperplane, in that case a soft margin can be used which means that the hyperplane separates many, but not all data points. The data for training is a set of vectors  $x_j$  along with their classes  $y_j$ , where  $j$  is a training instance  $j = 1, 2, \dots, l$  and  $l$  is the number of training instances. The hyperplane can be created in a higher dimensional space if separating the classes requires it. The hyperplane is described by  $w^T \varphi(x_j) + w_0 = 0$ , where  $\varphi$  is a function that maps  $x_j$  to a higher-dimensional space and  $w$  is the normal to the hyperplane. The SVM classifier satisfies the following conditions:

$$\begin{cases} w^T \varphi(x_j) + w_0 \geq +1 & \text{if } y_j = +1 \\ w^T \varphi(x_j) + w_0 \leq -1 & \text{if } y_j = -1 \end{cases} \quad j = 1, 2, \dots, l \quad (2.2)$$

and classifies according to the following decision function

$$y(x) = \text{sign} [w^T \varphi(x_j) + w_0], \quad (2.3)$$

where  $\varphi$  non-linearly maps  $x$  to the high-dimensional feature space. A linear separation is then performed in the feature space which is illustrated in 2.4.



**Figure 2.4:** Illustration of the non-linear mapping of  $\phi$  from the input space to the high-dimensional feature space. The figure shows an example which maps from a 2-dimensional input space to a 3-dimensional feature space, but the resulting feature space can be of higher dimensions. In both spaces the data points of different classes, shown as + and - are on different sides of the hyperplane but in the high-dimensional space they are linearly separable. Figure drawn according to [2].

If the feature space is high-dimensional, performing computations in that space is computationally heavy. Therefore a kernel function is introduced which is used to map the original non-linear observations into higher dimensional space more efficiently. The kernel function can be expressed as a dot product in a high-dimensional space. Through the kernel function all computations are performed in the low-dimensional input space. The kernel function is

$$K(x, x') = \varphi(x)^T \varphi(x'), \quad (2.4)$$

which is equal to the inner product of the two vectors  $x$  and  $x'$  in the feature space. Using kernels a new non-linear decision function is retrieved:

$$y(x) = \text{sign} \left[ \sum_{j=1}^l y_j K(x, x'_j) + w_0 \right], \quad (2.5)$$

which corresponds to the form of the hyperplane in the input space. [2] [11]

## 2.4 Histogram of oriented gradients

Histogram of oriented gradients (HOG) is a commonly used feature extraction method for machine learning implementations for object detection. It works by describing an image as a set of local histograms which in turn represent occurrences of gradient orientations in a local part of the image. The image is divided into blocks with 50% overlap, each block is in turn divided into cells. Due to the overlap of the blocks one cell can be present in

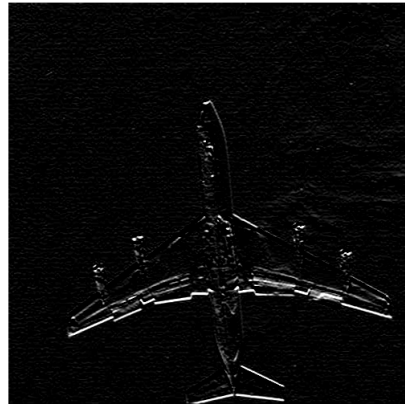
more than one block. For each pixel in each cell the gradients in the x and y directions ( $G_x$  and  $G_y$ ) are calculated. The gradients represent the edges in an image in the two directions and are illustrated in image 2.5.



(a) Original image



(b) Gradient in the x direction  $G_x$



(c) Gradient in the y direction  $G_y$

**Figure 2.5:** An image and its gradient representations in the x and y directions.

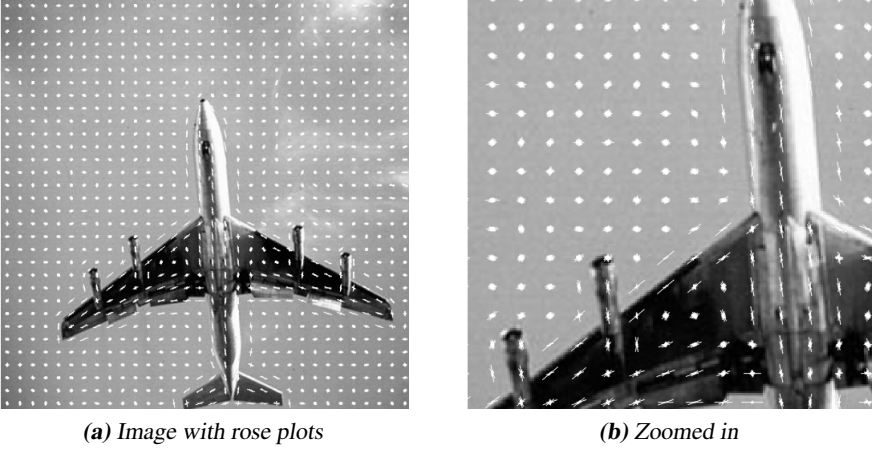
The magnitude and phase of the gradients are then calculated according to:

$$r = \sqrt{G_x^2 + G_y^2} \quad (2.6)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.7)$$

For each cell a histogram of orientations is created. The phases are used to vote into bins which are equally spaced between  $0^\circ - 180^\circ$  when using unsigned gradients. Using unsigned gradients means that whether an edge goes from dark to bright or from bright

to dark does not matter. To achieve that, angles below  $0^\circ$  are increased by  $180^\circ$  and angles above  $180^\circ$  are decreased by  $180^\circ$ . The vote from each angle is weighted by the corresponding magnitude of the gradient. The histograms are then normalized with respect to the cells in the same block. Finally, the histograms for all cells are concatenated into a vector which is the resulting feature vector. [20] [8] The resulting histograms for all cells in an image is shown as rose plots in figure 2.6.



**Figure 2.6:** The histograms of each cell in the image is visualized using rose plots. The rose plots shows the edge directions, which are normal to the gradient directions used in the histograms. Each bin is represented by a petal of the rose plot. The length of the petal indicates the size of that bin, meaning the contribution to that direction. The histograms have bins between  $0^\circ - 180^\circ$  which makes the rose plots symmetric. [12]

## 2.5 Features extracted from the discrete cosine transform domain

Representing an image or an image patch  $I$  of size  $M \times N$  in the discrete cosine domain is done by transforming the image pixel values according to:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_{mn} \cos\left(\frac{\pi(2m+1)p}{2M}\right) \cos\left(\frac{\pi(2n+1)q}{2N}\right) \quad (2.8)$$

where  $0 \leq p \leq M-1$ ,  $0 \leq q \leq N-1$ ,

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \leq p \leq M-1 \end{cases} \quad (2.9)$$

and

$$\alpha_q = \begin{cases} 1/\sqrt{N}, & p = 0 \\ \sqrt{2/N}, & 1 \leq p \leq N - 1 \end{cases} \quad (2.10)$$

As seen in equation (2.8), the image is represented as a sum of sinusoids with varying frequencies and magnitudes after the transform. The benefit of representing an image in the DCT domain is that most of the visually significant information in the image is concentrated in just a few coefficients which represent frequencies instead of pixel values. [13]

It has been shown that natural, undistorted images exhibit strong structural dependencies. These dependencies are local spatial frequencies that interfere constructively and destructively over scales to produce the spatial structure in natural scenes. Features that are extracted from the discrete cosine transform (DCT) domain are defined by [19], which represent image structure and whose statistics are observed to change with image distortions. The structural information in natural images can loosely be described as smoothness, texture and edge information.

The features are extracted from an image by splitting the image into equally sized  $N \times N$  blocks with two pixel overlap between neighbouring blocks. For each block, 2D local DCT coefficients are calculated using the discrete cosine transform described in equation (2.8). Then a generalized Gaussian density model shown in equation (2.11) is introduced and used to approximate the distribution of DCT image coefficients.

$$f(x|\alpha, \beta, \gamma) = \alpha \exp(-(\beta|x - \mu|)^\gamma), \quad (2.11)$$

where  $x$  is the multivariate random variable,  $\mu$  is the mean,  $\gamma$  is the shape parameter,  $\alpha$  and  $\beta$  are the normalizing and scale parameters given by:

$$\alpha = \frac{\beta\gamma}{2\Gamma(1/\gamma)} \quad (2.12)$$

$$\beta = \frac{1}{\sigma} \sqrt{\frac{\Gamma(3/\gamma)}{\Gamma(1/\gamma)}} \quad (2.13)$$

where  $\sigma$  is the standard deviation and  $\Gamma$  is the gamma function given by:

$$\Gamma(z) = \int_0^{\infty} t^{z-1} \exp(-t) dt \quad (2.14)$$

The generalized Gaussian density model is applied to each block of DCT components and to special partitions within each block. An example of a  $5 \times 5$  sized block and its partitions are illustrated in figure 3.2a. One of these partitions emerge when each block is partitioned into three radial frequency sub-bands which are represented as different levels of shadings in figure 2.7b. The other partition emerge when each block is split directionally into three oriented sub-regions which are represented as different levels of shadings in figure 2.7c.

DC	C <sub>12</sub>	C <sub>13</sub>	C <sub>14</sub>	C <sub>15</sub>
C <sub>21</sub>	C <sub>22</sub>	C <sub>23</sub>	C <sub>24</sub>	C <sub>25</sub>
C <sub>31</sub>	C <sub>32</sub>	C <sub>33</sub>	C <sub>34</sub>	C <sub>35</sub>
C <sub>41</sub>	C <sub>42</sub>	C <sub>43</sub>	C <sub>44</sub>	C <sub>45</sub>
C <sub>51</sub>	C <sub>52</sub>	C <sub>53</sub>	C <sub>54</sub>	C <sub>55</sub>

(a) A  $5 \times 5$  block in an image on which the parameters  $\gamma$  and  $\zeta$  are calculated

DC	C <sub>12</sub>	C <sub>13</sub>	C <sub>14</sub>	C <sub>15</sub>
C <sub>21</sub>	C <sub>22</sub>	C <sub>23</sub>	C <sub>24</sub>	C <sub>25</sub>
C <sub>31</sub>	C <sub>32</sub>	C <sub>33</sub>	C <sub>34</sub>	C <sub>35</sub>
C <sub>41</sub>	C <sub>42</sub>	C <sub>43</sub>	C <sub>44</sub>	C <sub>45</sub>
C <sub>51</sub>	C <sub>52</sub>	C <sub>53</sub>	C <sub>54</sub>	C <sub>55</sub>

(b) A  $5 \times 5$  block split into radial frequency sub-bands  $a$  on which  $R_a$  is calculated

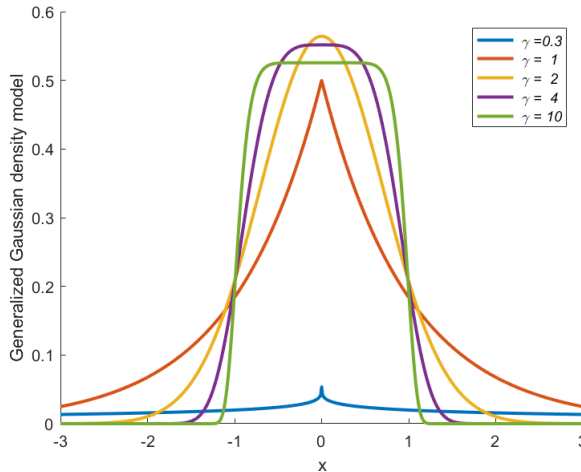
DC	C <sub>12</sub>	C <sub>13</sub>	C <sub>14</sub>	C <sub>15</sub>
C <sub>21</sub>	C <sub>22</sub>	C <sub>23</sub>	C <sub>24</sub>	C <sub>25</sub>
C <sub>31</sub>	C <sub>32</sub>	C <sub>33</sub>	C <sub>34</sub>	C <sub>35</sub>
C <sub>41</sub>	C <sub>42</sub>	C <sub>43</sub>	C <sub>44</sub>	C <sub>45</sub>
C <sub>51</sub>	C <sub>52</sub>	C <sub>53</sub>	C <sub>54</sub>	C <sub>55</sub>

(c) A  $5 \times 5$  block split into oriented sub-bands  $b$  on which  $\zeta_b$  is calculated.

**Figure 2.7:** Illustrations of the dct components in a block which an image is split into and the partitions created in each of the blocks. (Image source: [19])

Then four parameters derived from the generalized Gaussian model parameters are computed. These four parameters make up the features used for each image. The retrieved values of each parameter is pooled in two different ways, resulting in two features per parameters. The parameters are as follows:

- The generalized Gaussian model shape parameter  $\gamma$  seen in equation (2.11) which is a model-based feature that is retrieved over all blocks in the image. The parameter  $\gamma$  determines the shape of the Gaussian distribution, hence how the frequencies are distributed in the blocks. Figure 2.8 illustrates the generalized Gaussian distribution in equation (2.11) for different values of the parameter  $\gamma$ .



**Figure 2.8:** Generalized Gaussian distribution for different values of  $\gamma$ .

The parameter  $\gamma$  is retrieved by inserting values in the range 0.3-10 in equation

(2.11) to find the distribution which best matches the actual distribution of DCT components in each block. The resulting features are the lowest 10th percentile of  $\gamma$  and the mean of  $\gamma$ .

- The frequency variation coefficient  $\zeta$ ,

$$\zeta = \frac{\sigma_{|X|}}{\mu_{|X|}} = \sqrt{\frac{\Gamma(1/\gamma)\Gamma(3/\gamma)}{\Gamma^2(2/\gamma)} - 1}, \quad (2.15)$$

where  $X$  is a random variable representing the histogrammed DCT coefficients,  $\sigma_{|X|}$  and  $\mu_{|X|}$  are the standard deviation and mean of the DCT coefficient magnitudes of the fit to the generalized Gaussian model,  $\Gamma$  is the gamma function given by equation (2.14) and  $\gamma$  is the shape parameter. The feature  $\zeta$  is computed for all blocks in the image. The ratio  $\zeta$  has shown to correlate well with subjective judgement of perceptual quality. The resulting features are the highest 10th percentile of  $\zeta$  and the mean of  $\zeta$ .

- The energy sub-band ratio, which is retrieved from the partitions emerging from splitting each block into radial frequency sub bands. The three sub bands are represented by  $a$ , where  $a = 1, 2, 3$  which correspond to lower, middle and higher spatial radial frequencies, respectively. The average energy in sub band  $a$  is defined as its variance, described by

$$E_a = \sigma_a^2. \quad (2.16)$$

The average energy up to band  $n$  is described by

$$E_{j<a} = \frac{1}{n-1} \sum_{j<a} E_j \quad (2.17)$$

The energy values are retrieved by fitting the DCT histogram in each band  $a$  to the generalized Gaussian model and then taking the  $\sigma_a^2$  from the fit. Using the two parameters  $E_a$  and  $E_{j<a}$ , a ratio  $R_a$  between the components and the sum of the components, according to:

$$R_a = \frac{|E_a - E_{j<a}|}{E_a + E_{j<a}} \quad (2.18)$$

This ratio represents the relative distribution of energies in lower and higher bands, which can be affected by distortions. A large ratio value is retrieved when there is a large disparity between the frequency energy of a band and the average energy in the bands of lower frequencies. Since band  $a = 1$  does not have any bands of lower frequency, the ratio is calculated for  $a = 2, 3$  and the mean of the two resulting ratios  $R_1$  and  $R_2$  is the feature used. The feature is computed for all blocks in the image. The resulting features are the highest 10th percentile of  $R_a$  and the mean of  $R_a$ .

- The orientation model-based feature  $\zeta$ , which is retrieved from the partitions emerging from splitting each block into oriented sub-regions to capture directional information.  $\zeta_b$  is defined according to equation (2.15), from the model histogram fits

for each of the three orientations  $b = 1, 2, 3$ . The variance of each resulting  $\zeta_b$  from all the blocks in an image is calculated.  $\zeta_b$  and the variance of  $\zeta_b$  are used to capture directional information from images since image distortions often affect local orientation energy in an unnatural manner. The resulting features are the 10th highest percentile and the mean of the variance of  $\zeta$  across the three orientations from all the blocks in the image.

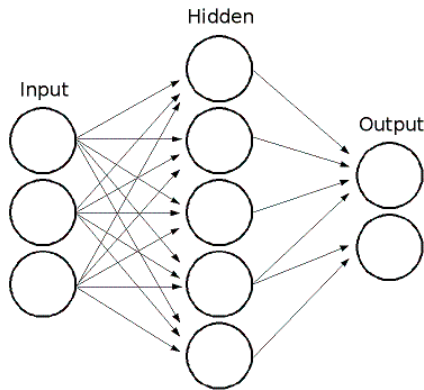
The features are extracted and the feature extraction is repeated after a low-pass filtering and a sub-sampling of the images, meaning that the feature extraction is performed over different scales. The above eight features are extracted on three scales of the images to capture variations in the degree of distortion over different scales. The low-pass filtering and sub-sampling provides coarser scales on which larger distortions can be captured since the entire image is briefed on fewer values, as if it was a smaller region. The low-pass filtering is with a symmetric Gaussian filter kernel and the sub-sampling is done by a factor of 2.

## 2.6 Features extracted from a convolutional neural network

### 2.6.1 Convolutional neural networks

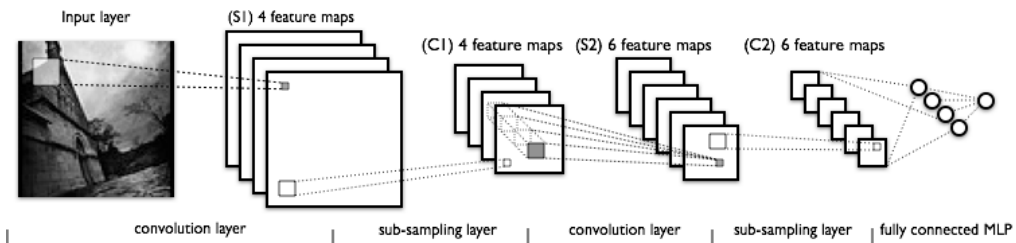
Convolutional neural network (CNN) is a machine learning method which has successfully been applied to the field of image classification. The structure roughly mimics the nature of the mammalian visual cortex and neural networks in the brain. It is inspired by the human visual system because of its ability to recognize and localize objects within cluttered scenes. That ability is desired within artificial system in order to overcome the challenges of recognizing objects in a class despite high in-class variability and perspective variability. [4]

Convolutional neural networks is a form of artificial neural networks. The structure of an artificial neural network is shown in figure 2.9.



**Figure 2.9:** The structure of an artificial neural network. A simple neural network with three layers; an input layer, one hidden layer and an output layer. (Image source: [15])

An artificial neural network consists of neurons in multiple layers; the input layer, the output layer and one or more hidden layers. Networks with two or more hidden layers are called deep neural networks. The input layer consists of an input data and the output layer consists of a value indicating whether the neuron is activated or not. In the case of classification, the neurons in the output layer represent the different classes. Each of the neurons in the output layer results in a soft-max value which describes the probability of the input belonging to that class. The input to a neuron is the weighted outputs of the neurons in the previous layer, if a layer is fully connected it consists of the output from all neurons in the previous layer. The weight controls the amount of influence the output of a neuron has on the next neuron. The hidden layers each consists of different combinations of the weighted outputs of the previous layers. That way, with increased number of hidden layers, more complex decisions can be made. The method can simplified be described as composing complex combinations of the information about the input data which correctly maps the input data to the correct output. In the training part, when the network is trained, those complex combinations are formed, which can be thought of as a classification model. In the evaluation part, that model is used to classify new data. [15] Convolutional neural networks is a form of artificial neural networks which is applied to images and has a special layer structure which is shown in figure 2.10.



**Figure 2.10:** The structure of a convolutional neural network. A simple convolutional neural network with two convolutional layers, each of them followed by a sub-sampling layer and finally two fully connected layers. (Image source: [1])

The hidden layers of a CNN are one or more convolutional layers each followed by a pooling layer in succession, followed by one or more fully connected layers. The convolutional layers are feature extraction layers and the last fully connected layer act as the classifier. The convolutional layers in turn consist of two different layers; the filter bank layer and the non-linearity layer. The inputs and outputs to the convolutional layers are feature maps represented in a matrix. For a 3-color channeled RGB image the dimensions of that matrix are  $W \times H \times 3$ , where  $W$  is the width,  $H$  is the height and 3 is the number of feature maps. For the first layer the input is the raw image pixel values for each color channel. The filter bank layers consist of multiple trainable kernels, which are convolved with the input to the convolution layer, with each feature map. Each of the kernels detects a particular feature at every location on the input. The non-linearity layer applies a non-linear sigmoid activation function to the output from the filter bank layer. In the pooling layers following the convolutional layers, sub-sampling occurs. The sub-sampling is done for each feature map and decreases the resolution of the maps. After the convolutional layers the output is passed on to the fully connected layers. In the connected layers different weighted combinations of the inputs are formed which in the final step results in decisions about which class the image belongs to. [9]

## 2.6.2 Extracting features from a pre-trained network

Using features extracted from pre-trained neural networks trained on large and general tasks have been shown to produce useful results which outperforms many existing methods and clustering with high accuracy when applied to novel data sets. It has shown to perform well on new tasks, even clustering into categories on which the network was never explicitly trained.[6] These features extracted from a deep convolutional neural network (CNN) are retrieved from the VGG-F network provided by MatConvNet's archive of open source implementations of pre-trained models. The network contains 5 convolutional layers and 3 fully connected layers. The features are extracted from the neuron's activity in the penultimate layer, resulting in 1000 soft-max values. The network is trained on a large data set containing 1.2 million images used for a 1000 object category classification task. The features extracted are to be used as descriptors applicable to other data sets. [3]

## 2.7 Color coherence vector

A color coherence vector consists of a pair of measures for each color describing how many coherent pixels and how many incoherent pixels there are of that color in the image. A pixel is coherent if it belongs to a contiguous region of the color, larger than a preset threshold value. Therefore, unlike color histograms which only provide information about the quantity of each color, color coherence vectors also provide some spatial information about how the colors are distributed in the image. A color coherence vector for an image consists of:

$$\langle (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n) \rangle \quad j = 1, 2, \dots, n$$

where  $\alpha_j$  is the number of coherent pixels,  $\beta_j$  is the number of incoherent pixels for color  $j$  and  $n$  is the number of indexed colors.

By comparing the color coherence vectors of two images, a similarity measure is retrieved. The similarity measure between two images  $I$  and  $I'$  is then given by the following parameters:

$$\text{differentiating pixels} = \sum_{j=1}^n |\alpha_j - \alpha'_j| + |\beta_j - \beta'_j| \quad (2.19)$$

$$\text{similarity} = 1 - \frac{\text{differentiating pixels}}{\text{all pixels} * 2} \quad (2.20)$$

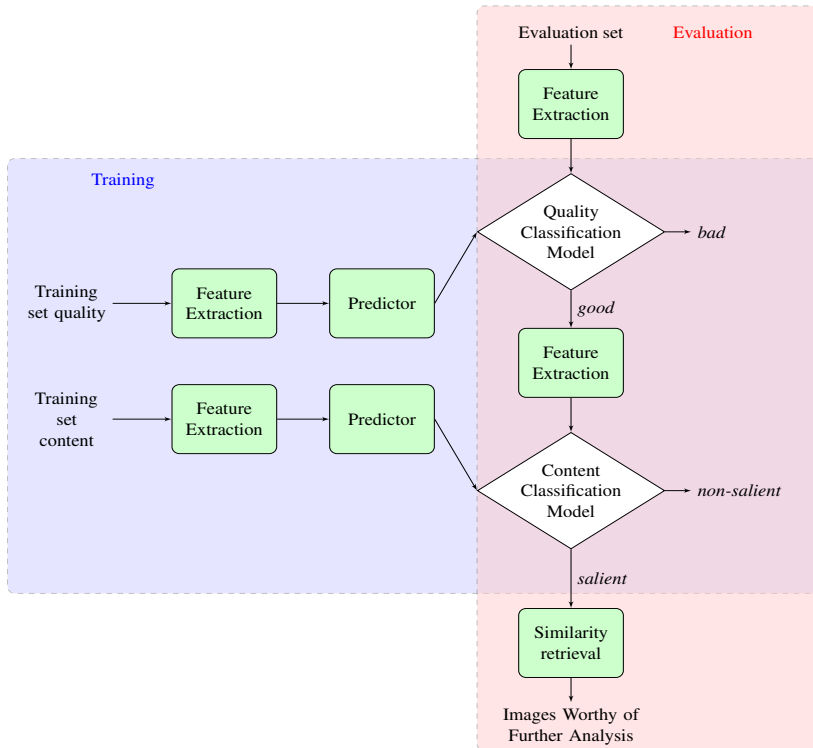
[17]

# 3

---

## Method

This chapter includes a description of how the different parts of the system are implemented. A flowchart of how the different parts of the system interrelate is shown in Figure 3.1. The implementation is divided into two parts: a training part and an evaluation part. For both parts the first step is feature extraction from the images which is described in section 3.1. In the training part, features are extracted from one content training set containing examples of images with *salient* and *non-salient* images and one quality training set which contains examples of images with *good* and *bad* quality. The features are sent to the predictor which creates a classification model for each training set, one quality classification and one content classification model. The predictor is described in section 3.2. In the evaluation part, features are extracted from an evaluation set. The features are used to classify the images according to the classification models retrieved in the training part. Images that are classified as both *good* and *salient* will continue to the final step in the evaluation part. The final step is a retrieval step where one image is selected from a cluster of images that are very similar to each other. The retrieval step is described in section 3.3. After passing through the three selection steps the images that are left are classified as *good*, *salient* and unique which means that they are worthy of further analysis.



**Figure 3.1:** Flow chart of implementation. The system is trained on two different input sets which leads to two classification models: one for quality and one for content. The evaluation set is classified using the two models, the images that are classified as both good and salient will be sent to the retrieval part. In the retrieval part, a selection will be made from sets of images that are similar, so that only one will be retrieved. The resulting images are good, salient and unique which means that they are worthy of further analysis.

### 3.1 Feature extraction

Three different methods of feature extraction are performed which leads to three different results for each classification, which are compared against each other. The best feature extraction method for each of the two classifications is used for that part and the entire system is put together. The methods that are used are the following: histogram of oriented gradients (HOG) [20], features extracted from the discrete cosine (DCT) domain [21] and features extracted from a pretrained convolutional neural network (CNN) [3]. The feature extraction methods have different advantages which are the reasons for why they are chosen. HOG is often used for object detection, it uses gradients to describe images. Since gradients provide information about edges and corners in an image, HOG is favorable when describing content in an image. The method of extracting features from the DCT domain on the other hand is chosen because the features are produced to describe quality

parameters in an image. The last method, using features extracted from a CNN, where the network is trained on a large set of images in an object recognition task to be able to generalize to other tasks and data sets for which the network has not been trained. The method is chosen because of its ability to perform well on generic tasks.

## 3.2 Predictor

The predictor used is an SVM as described in section 2 using the MATLAB implementation [11]. The model is trained on labelled examples of images of *good* and *bad* quality to retrieve a quality classification model. Another SVM model is trained on labelled examples of *salient* and *non-salient* images to retrieve a content classification model. When using a model to classify new data, the resulting output for each image is a class label and a certainty score matrix. The score matrix contains the scores for each image being classified in the negative class and the positive class respectively. The predictor SVM is chosen because of its advantages, one of them being not having the problem of over-fitting. Over-fitting occurs when a model has too many features relative to the number of observations and results in poor predictive performance. The problem of over-fitting is relevant to take into account when working with machine learning on images because the number of features extracted from an image is often very large. [16] SVM has previously been used in many image classification tasks with good results [20] [19].

## 3.3 Similarity retrieval

The retrieval step is performed on images that are classified as both *good* and *salient*. On those images, pairwise similarity measures is done based on difference in color coherence vectors of the images, according to [17]. The difference in color coherence vectors of two images consists of difference in number of coherent pixels and number of incoherent pixels of each color. The threshold value that determines whether a contiguous area is coherent or not is 2500 pixels which corresponds to 10% of an image. The images are first low-pass filtered using a local averaging filter of size  $5 \times 5$  pixels. The images are then converted from RGB valued to indexed valued with 128 different colors using the colormap jet.

The images are then clustered based on the similarity measures. The pairwise similarity measures from all images in a set form a similarity matrix which is then clustered. The clustering is done by placing an image in a cluster if it has an average similarity above 87% to that cluster. The average similarity between an image and a cluster is the mean value of the pairwise similarity measures between an image and all images in the cluster. From each cluster only one image is retrieved and that is the one with the highest sum of the score for being classified in the *good* quality class and the score for being classified in the *salient* class. The result is a set of images which are all unique compared to each other.

### 3.4 Evaluation

The system is evaluated using the results from the evaluation part and how well it conforms with the ground truth for the evaluation set. Each of the classifications and the retrieval is evaluated separately. For binary classification the resulting output for every image is either the positive or the negative class which is either true or false. This means each image can be described as a true/false positive/negative.

For the retrieval part, the resulting output for each image is whether it should be retrieved or not, which is either true or false. This means that every image can be described as a true/false negative/positive.

After evaluating each part separately the system is put together. For each of the classifications, the feature extraction method which provided the best resulting average accuracy is used. The results of the entire system is then evaluated. That is done by describing which images are retrieved as worthy of further analysis and how well it conforms with which images that should be. Images that are worthy of further analysis are images that are *good*, *salient* and unique with respect to the other retrieved images. The final output for an image is whether its retrieval is true or false, the same way as for the retrieval part. That way, true/false negatives/positives are achieved.

All results will be evaluated using the measures precision, recall and accuracy which are defined as:

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \quad (3.1)$$

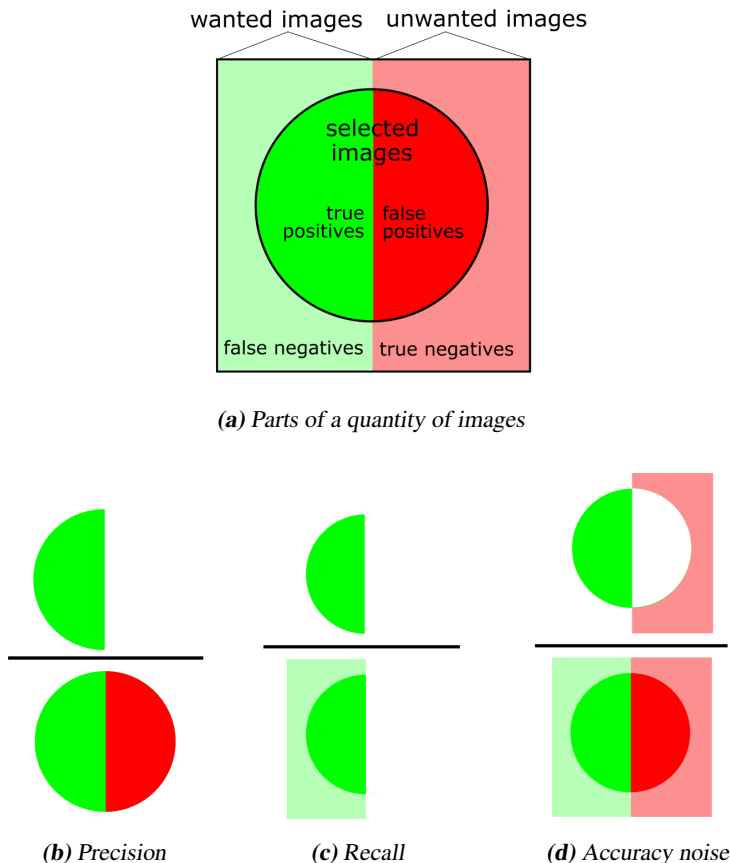
which describes how many of the retrieved images which should be retrieved.

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \quad (3.2)$$

which describes how many of the images that should be retrieved that are retrieved.

$$\text{Accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{all samples}} \quad (3.3)$$

which describes how many classifications that are out of all classifications made. The concept of true/false negatives/positives and the measures are illustrated in the in figure 3.2.

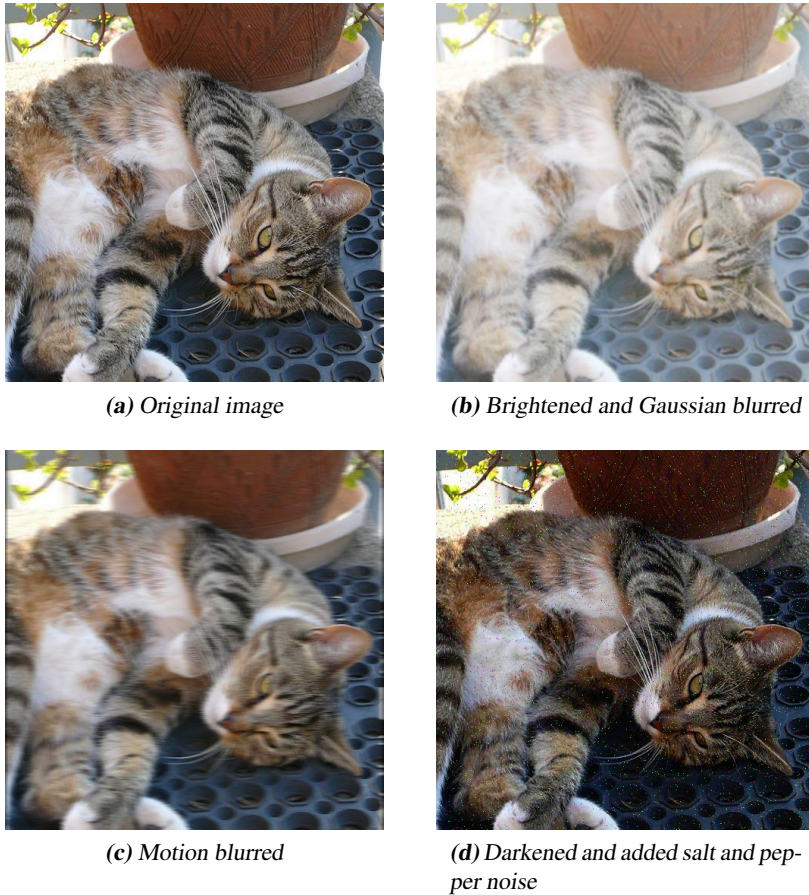


**Figure 3.2:** An illustration of the concept used in the definition of the measures precision, recall and accuracy. Out of a quantity of images some are selected which are noted positives and can be either true or false. The non-selected images are called negatives which can be either true or false. The different concepts are illustrated in (a) and how they define the measures is illustrated in (b), (c) and (d).

### 3.5 Generation of training and evaluation data

The COCO data set consists of objects sorted into 91 different categories, to fit the task new categories are formed. One category is set to form the *salient* class, the investigation is performed multiple times with different objects as *salient*. The *non-salient* class contain images which are randomly selected from other categories than the one chosen as *salient*. The images have been manually weeded by removing non-representative images such as animated images, collages and images of questionable quality. After the weeding it is assumed that the images are of good quality to begin with and are placed in the *good* class. The data is modified to fit the task by modifying quality parameters to degrade the image quality in the following way: brightening, darkening, adding salt and pepper-noise,

adding Gaussian noise, adding Gaussian blur and adding motion blur. To avoid the alterations counteracting each other they are divided into the two groups light and noise/blur. The modification is done randomly and one image can be subject to one alteration alone or a combination of two alterations. To one image at most one alteration from each group is applied. The degree of the degradation is randomized and the degraded image is then compared to the original using the structural similarity (SSIM) index introduced in [21]. SSIM provides an objective measurement of the quality of an image compared to a reference image. The measurement focuses on comparing how well the structures in the image are preserved and considers image degradations as perceived changes in structural information. The images that have an SSIM value above 65% have more than 65% of their structures preserved and are set to belong to the *good* class. The images that have SSIM value 65% or less are assumed to be of bad quality and make up the *bad* class. Examples of images which have been degraded to SSIM = 65% are shown in figure 3.3.



**Figure 3.3:** An image and examples of degraded versions of it, the original is seen in (a) and the degraded versions are seen in (b), (c) and (d). The degraded images have been subjects to different degradation methods and have the same SSIM index  $\approx 65\%$ .

Each class is divided into a training part and an evaluation part. The images are divided into approximately 80% training data and 20% evaluation data. The number of training images in the *salient* class is approximately 2000 but varies slightly depending on which object is set to salient. The number of training images in the *non-salient* class is approximately the same as the number of training images in the corresponding *salient* class. The number of images in the evaluation data set from the two classes are 920 for all different salient objects. The number of images in the classes *good* and *bad* differs in both the training set and the evaluation set. The quality training set consists of the content training set and modified versions of them and the quality evaluation set consists of the content evaluation set and modified versions of them. The *good* class consists of all images in the *salient* and the *non-salient* class and the modified versions of them having

an SSIM value above 65%. The *bad* class consists of the modified versions of the images in the *salient* and *non-salient* class that have an SSIM value less than or equal to 65%. Therefore the number of *bad* images are always less than the number of *good* images. The modification is done randomly which means that the number of *bad* images varies depending on what object is set to salient.

The data is modified to fit the task also by creating images that are very similar to each other. That is done by applying one or more rigid transformations to an image and therefore creating different versions of it. That is done without changing the saliency of the images, meaning that the salient object is present in all versions of the images. Images that originate from the same image are assumed to be similar and belong to the same cluster. Examples of images that are set to similar are shown in image 3.4. All images have been resized and cropped to obtain the size  $500 \times 500$  pixels.



**Figure 3.4:** Examples of similar images that originate from the same image and belong to the same cluster.

# 4

---

## Results

### 4.1 Quality classification

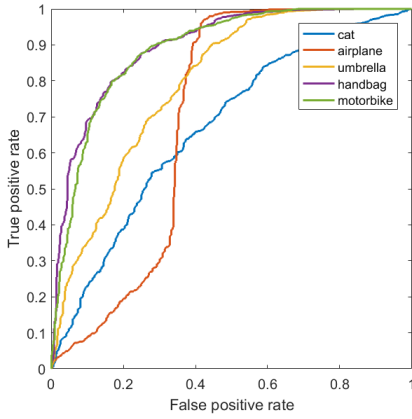
The evaluation of the quality classification is done for each of the salient objects. For each salient object a set of 1840 images is used for evaluation. Each set consists of both *salient* and *non-salient* images. 920 images have been modified randomly, as described in section 3.5 and 920 images have not. The images that have an SSIM value above 65% should be classified as *bad* and the rest as *good*. Since the degradation is done randomly the number of *good* and *bad* images in the evaluation set varies with the salient objects. The number of images in the *good* class is always larger than the number of images in the *bad* class and therefore classifying all images as *good* gives a recall value of 100%, a precision value same as the classification accuracy which is equal to the proportion of *good* images. If the difference in number of images in the two classes is large enough, classifying all images as *good* might lead to a false perception of good results. Therefore the proportion of *good* images needs to be considered when interpreting the results. The proportion of *good* images for the different salient objects is shown in table 4.1. The results of the quality classification are shown in table 4.2. The results are visualized using receiver operating characteristic (ROC) curves shown in figure 4.1. The ROC-curves shows the relation between true positive rate (recall) and true negative rate.

**Table 4.1:** The proportion of *good* images for the different salient objects

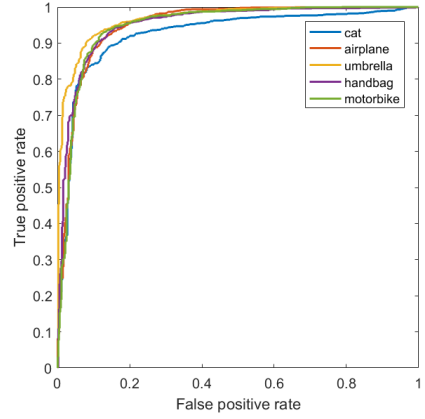
Proportion good images	Salient object
0.6951	cat
0.7288	airplane
0.6935	umbrella
0.6821	handbag
0.6902	motorbike

**Table 4.2:** Results from the evaluation of the quality classification for the different feature extraction methods and for different categories as salient.

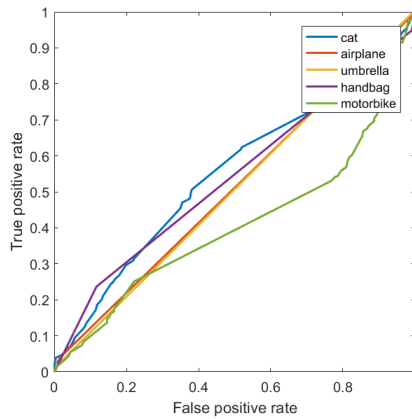
Feature extraction method	Precision	Recall	Accuracy	Salient object
HOG	0.8399	0.939	0.8332	cat
HOG	0.8544	0.9799	0.8636	airplane
HOG	0.8018	0.9702	0.813	umbrella
HOG	0.8333	0.9442	0.8332	handbag
HOG	0.8506	0.9236	0.8353	motorbike
HOG	0.8360	0.9514	0.8357	<b>average</b>
Extracted from the DCT domain	0.9196	0.9116	0.8832	cat
Extracted from the DCT domain	0.9292	0.9500	0.9109	airplane
Extracted from the DCT domain	0.9348	0.9444	0.9158	umbrella
Extracted from the DCT domain	0.9348	0.9251	0.9049	handbag
Extracted from the DCT domain	0.9308	0.9425	0.9120	motorbike
Extracted from the DCT domain	0.9298	0.9347	0.9054	<b>average</b>
Features extracted from a CNN	0.6951	1	0.6951	cat
Features extracted from a CNN	0.7288	1	0.7288	airplane
Features extracted from a CNN	0.6935	1	0.6935	umbrella
Features extracted from a CNN	0.6821	1	0.6821	handbag
Features extracted from a CNN	0.6902	1	0.6902	motorbike
Features extracted from a CNN	0.6979	1	0.6979	<b>average</b>



(a) HOG features



(b) Features extracted from the DCT domain



(c) Features extracted from a CNN

**Figure 4.1:** ROC-curves for the quality classifications. The curves show the relation between true positive rate (recall) and false positive rate (false positives/all negatives). (a) shows the results from using HOG features, (b) shows the results from using features extracted from the DCT domain and (c) shows the results from using features extracted from a CNN. The different salient objects are shown as different colors.

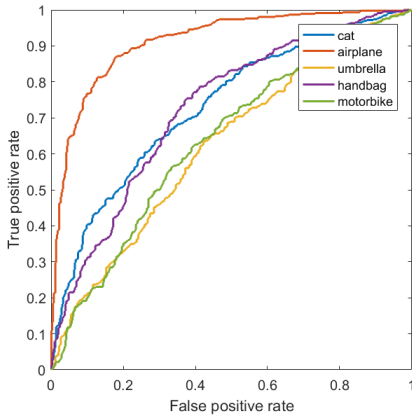
Features extracted from the DCT domain has the highest accuracy for all *salient* objects. Therefore this is the feature extraction method used for the quality part when putting the entire system together.

## 4.2 Content classification

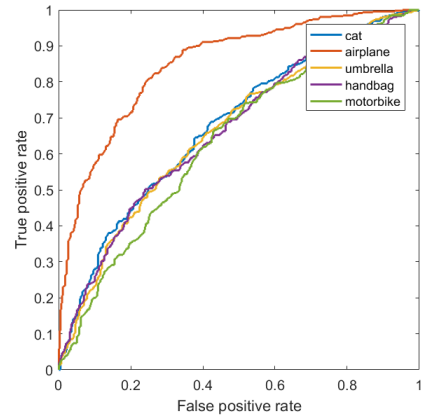
The evaluation of the content classification is done for each of the salient objects. For each salient object a set of 920 images without modifications is used for evaluation. 460 of those images are *salient* containing the *salient* object and 460 are *non-salient* containing random images from other categories. The number of images in the two categories are equal which makes the values for precision, recall and accuracy easy to interpret. The guess of placing all images in one class would lead to an accuracy of 50% and one of the values for precision or recall to 100% and the other to 50%, depending on which class the images are placed in. The results of the content classification are shown in table 4.3. The results are visualized using ROC-curves shown in figure 4.2. The ROC-curves shows the relation between true positive rate (recall) and false positive rate.

**Table 4.3:** Results from the evaluation of the content classification for the different feature extraction methods and for different categories as *salient*.

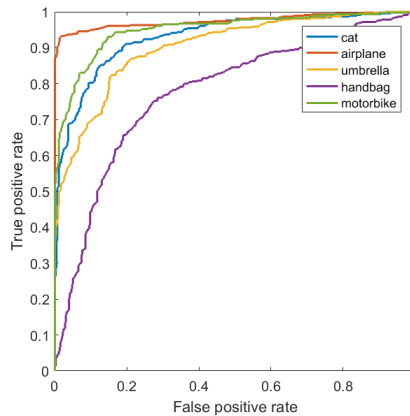
Feature extraction method	Precision	Recall	Accuracy	Salient object
HOG	0.6631	0.6717	0.6652	cat
HOG	0.8645	0.8043	0.8391	airplane
HOG	0.5959	0.5739	0.5924	umbrella
HOG	0.6759	0.6348	0.6652	handbag
HOG	0.5758	0.7348	0.5967	motorbike
HOG	0.6750	0.6839	0.6717	<b>average</b>
Extracted from the DCT domain	0.6253	0.6239	0.6250	cat
Extracted from the DCT domain	0.8182	0.6457	0.7511	airplane
Extracted from the DCT domain	0.6223	0.6196	0.6217	umbrella
Extracted from the DCT domain	0.6256	0.5630	0.613	handbag
Extracted from the DCT domain	0.5881	0.7326	0.6098	motorbike
Extracted from the DCT domain	0.6559	0.6370	0.6441	<b>average</b>
Features extracted from a CNN	0.9038	0.7761	0.8467	cat
Features extracted from a CNN	1	0.6935	0.8467	airplane
Features extracted from a CNN	0.8155	0.8457	0.8272	umbrella
Features extracted from a CNN	0.7560	0.6804	0.7304	handbag
Features extracted from a CNN	0.9242	0.8217	0.8772	motorbike
Features extracted from a CNN	0.8799	0.7635	0.8256	<b>average</b>



(a) HOG features



(b) Features extracted from the DCT domain



(c) Features extracted from a CNN

**Figure 4.2:** ROC-curves for the content classifications. The curves show the relation between true positive rate (recall) and false positive rate (false positives/all negatives). (a) shows the results from using HOG features, (b) shows the results from using features extracted from the DCT domain and (c) shows the results from using features extracted from a CNN. The different salient objects are shown as different colors.

Features extracted from a CNN has the highest accuracy for all *salient* objects. Therefore this is the feature extraction method used for the content part when putting the entire system together.

### 4.3 Similarity retrieval

The evaluation of the retrieval part of the system is done for each of the salient objects. For each salient object a set of 360 salient images are used for evaluation. 180 images are unique and 180 images belong to a cluster of similar images. Each set contains 62 clusters of varying sizes with 2-6 images in each cluster. The ideal output from the retrieval part is one image from each cluster. The scores that determine which image from each cluster that should be retrieved are results of the classifications. When investigating only the retrieval part the results from the classifications should not affect the outcome and therefore all images are set to have the same score. Hence, the results of the evaluation of the retrieval depends solely on the clustering based on the similarity measures. Examples of images from the similarity retrieval with the *salient* object cat and their color coherence vectors are shown in figure 4.4. The similarity matrix containing the pairwise similarity measures of all images in the similarity set with the *salient* object cat is shown in figure 4.5a. Also shown is a binary similarity showing the true clusters as yellow in 4.5b. The results from the retrieval part is shown in table 4.4.



(a)

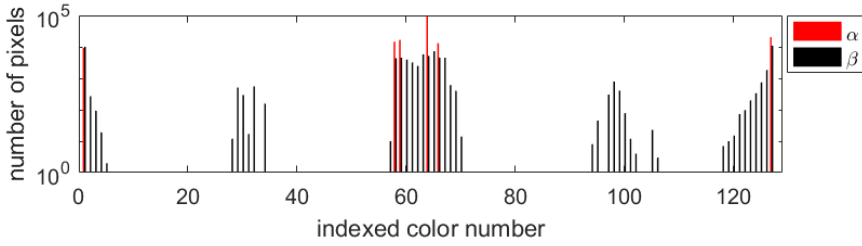


(b)

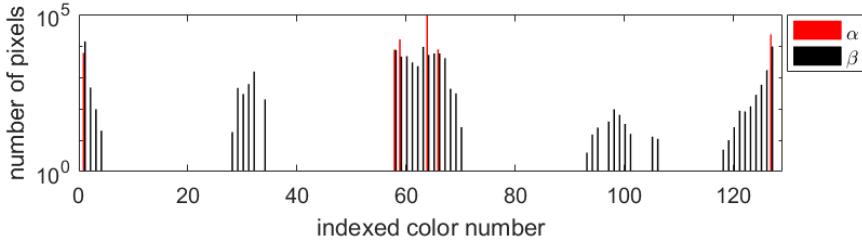


(c)

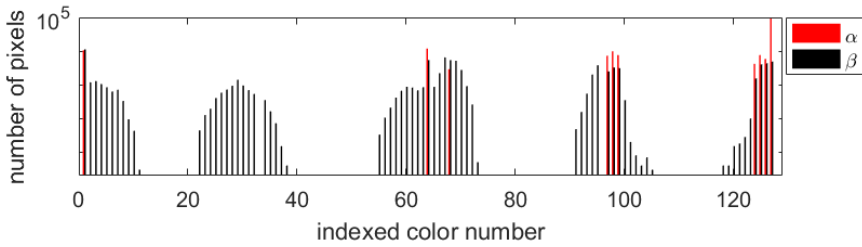
**Figure 4.3:** Examples of images that are clustered as similar and images that are not. Images (a) and (b) are placed in the same similarity cluster with similarity 91.18%. Image (c) is not placed in the same cluster and have resulting similarities 32.46% to (a) and 32.06% to (b).



(a) Color coherence vector of image 4.3a.

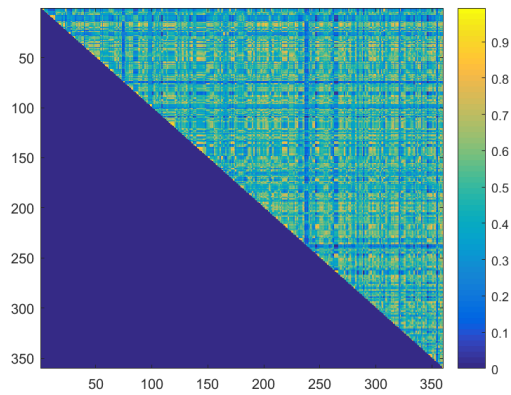


(b) Color coherence vector of image 4.3b

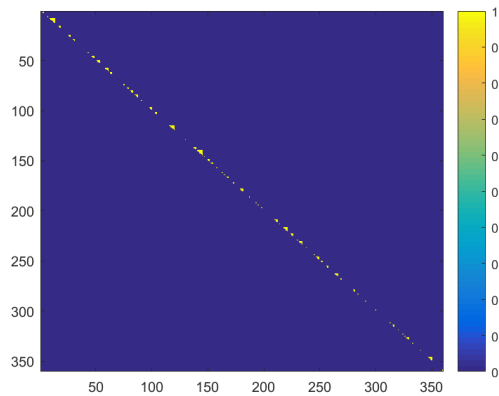


(c) Color coherence vector of image 4.3c

**Figure 4.4:** Color coherence vectors of images in figure 4.3. The x-axis are the indexed colors and the y-axis are the number of pixels in logarithmic scale. The red bars represent  $\alpha$  which is the number of coherent pixels for each color. The black bars represent  $\beta$  which is the number of incoherent pixels for each color.



(a) Resulting similarity matrix



(b) Binary similarity matrix showing images that originate from the same image

**Figure 4.5:** Matrices of pairwise similarity measures for the images in the similarity sub-set of the category *cat*. (a) is the resulting similarity matrix and (b) is a binary matrix showing the true similar as 1 and the rest as 0. Filling an entire similarity matrix would mean calculating the similarity measures between two images twice, which is avoided and results in upper triangular matrices.

**Table 4.4:** Results from the evaluation of the retrieval part for different categories as salient.

Precision	Recall	Accuracy	Salient object
0.7782	0.9421	0.7806	cat
0.8071	0.8471	0.7611	airplane
0.7698	0.8843	0.7444	umbrella
0.7537	0.8471	0.7111	handbag
0.7935	0.9050	0.7778	motorbike
0.7805	0.8851	0.7550	<b>average</b>

## 4.4 The entire system

The entire system is put together using the quality classification models retrieved using features extracted from the DCT domain. It is the feature extraction method which provided the best results when investigating the quality classification in section 4.1. The models used for the content classifications are the ones retrieved using features extracted from a CNN. It is the feature extraction method which provided the best results when investigating the content classification in section 4.2. The evaluation of the entire system is done for each of the salient objects. The evaluation is performed on the same sets as the evaluation of the quality classification which contains the evaluation sets from the content classification and the similarity retrieval. The output from the quality classification is input to the content classification and the output from the content classification is input to the similarity retrieval part. The results from the similarity retrieval part are the images that are evaluated compared to the images which are wanted. The images that are wanted are the ones which are actually *good*, *salient*, unique and best from its cluster. There are fewer images that are wanted than images that are not since half of the images are *salient* and some of them are almost duplicates and/or *bad*. There are 342 wanted images out of the total 1840 images which makes the proportion of wanted images 0.1859. The results of how the entire system works together is seen in table 4.5

**Table 4.5:** Results from the evaluation of the entire system for different categories as salient.

Precision	Recall	Accuracy	Salient object
0.5944	0.6813	0.8543	cat
0.6890	0.5117	0.8663	airplane
0.5055	0.6696	0.8168	umbrella
0.4717	0.5117	0.8027	handbag
0.6169	0.6404	0.8592	motorbike
0.5755	0.6029	0.8399	<b>average</b>

# 5

---

## Discussion

### 5.1 Results

#### 5.1.1 Quality classification

The evaluation of the quality classification shows that features extracted from the DCT domain gives the best results. Features extracted from the DCT domain gives an average accuracy of 90.54% compared to 83.57% for HOG and 69.79% for features extracted from a CNN. When taking the proportion of good images into account it appears that the accuracy values for features from a CNN matches the proportion values exactly. The fact that the precision values for the method also follows the proportion values and that the recall is always 1 implies from equations 3.1-3.3 that there are no true negatives or false negatives. The SVM was not able to create a good classification model using this method but simply classifies all images as *good*. This can be seen in the ROC-curve in figure 4.1c where all curves are very close to where the true positive rate equals the false positive rate, which is retrieved when placing all images in one class when the proportion of good images is 0.5. The slight differences are due to the proportion of good images not being 0.5 and small variations in the retrieved scores, although all scores are above the threshold for being *good*. The method of using features extracted from a CNN was chosen because of its ability of performing well on new data sets, however this task may differ too much from the task for which it was trained to be able to provide separating features. For HOG the recall is overall very high and the precision is lower and almost equal to the accuracy which implies that most images are classified as *good* with quite high number of false positives. So although it actually finds a classification model it is not a very good one. HOG is often used for object detection where it often is desired to disregard quality parameters such as lightning and blur. Therefore it is no surprise that it does not lead to great result when investigating quality. Since gradients describe difference in intensity, darkening or brightening entire images should not change the gradients unless edges disappear, and the histograms of oriented gradients are normalized which can explain why modifications

in lightning are hard to detect using HOG. Noise and blur should affect the histograms of oriented gradients. Noise should lead to many small, intense edges in spread directions, Gaussian blur should lead to fewer and weaker edges and motion blur should lead to fewer and weaker edges along the moving direction and many short edges orthogonal to the moving direction. However, no connection between modification types and images that are classified as *bad* is found. Features extracted from the DCT domain result in good values for precision, recall and accuracy, which shows that the SVM was able to find a good classification model. This is also seen in the ROC-curve in figure 4.1b. Ideal results are shown in a ROC-curve as following the left and the top borders, the results from features extracted from the DCT domain are quite close to that appearance. The features were extracted to describe quality parameters in images which makes it reasonable to find that that method gives the best result when investigating quality. Its features describe smoothness, texture and edge information which should be affected by noise and blur. None of them should however be directly affected by different lightning conditions. Despite that, no connection between modification type and images that are falsely classified is found.

Although the proportion of good images varies slightly between the different *salient* objects it is at most 3.09 percentage units from the mean value. The variation in accuracy values for the different sets of *salient* objects overall matches the variation in proportion in *good* images, meaning that the *salient* objects with slightly higher proportion of *good* images also have slightly higher accuracy. Therefore it is possible to interpret the results from the quality classification as being general and not varying remarkable with the different *salient* objects. This can be seen in the ROC-curves in figure 4.1b and 4.1c as the different colored curves being similar, the difference in proportion of *good* between the different *salient* objects however causes slight variations. In the ROC-curve for HOG features in figure 4.1a the curves are not very similar, which is partly because the different proportions of *good* images but mostly because it does not provide a good quality classification model. HOG provides a poor classification model from which the results varies between the different *salient* objects.

The number of *good* and *bad* training images varies with the salient object. Partly because the modification is done randomly but also because the number of images being modified varies. The largest *good* class consists of 6588 images and the smallest 4817. Although the number of training observations for each salient object is quite large the variation may impact the capacity of the resulting quality classification models. The small variations in the quality classification results is however more likely caused by the different context in the images.

The ROC-curves describe the trade-off between the true positive rate and the false positive rate, which is basically two different types of errors: letting too many images pass as *good* or finding too few *good* images. Following a curve gives the resulting true positive rate and false positive rate when changing how tolerant or strict the threshold for classifying images as *good* is. In this case where one class is retained and the other is not, it might be more important not to discard too many *good* images than to discard all *bad* images. Then, the threshold can be changed and the new rates can be retrieved from the ROC-curves in figure 4.1.

### 5.1.2 Content classification

The evaluation of the content classification shows that features extracted from a CNN gives the best results. Features extracted from a CNN gives an average accuracy of 82.56% compared to 67.17% for HOG and 64.41% for features extracted from the DCT domain. The accuracy values have variances 31.55% for features extracted from a CNN, 100.05% for HOG and 65.71% for features extracted from the DCT domain. Those numbers are all quite high and implies that the content classification is not general and varies significantly with the different *salient* objects. That can also be seen in the ROC-curves in figure 4.2 as the different colored curves representing different *salient* objects are differing. Figure 4.2b, which shows the results from using features extracted from the DCT domain, shows that the curves for the different *salient* objects are quite similar except for the category airplane. All curves are rather close to the line where the true positive rate equals the false positive rate except for airplane. Being close to that line, for this case where each of the two classes contain half of the images, corresponds to simply classifying all images in the same class. That means that the category airplane is the only one for which a decent classification model is retrieved. The bad performance of features extracted from the DCT domain for content classification for the majority of the different *salient* objects is not astonishing since it uses very few features describing statistics in images associated with quality. The decent result for the category airplane however, is more astonishing since it is able to differ somewhat between *salient* and *non-salient* images only described by smoothness, texture and edge information. Features extracted from a CNN are trained on a large set of images for an object classification task. The task is similar to this content classification and the features seem to fulfill their purpose of performing well when applied to new data sets. HOG are often used for content classification tasks and performing well. However, this shallow feature extraction method is outperformed by features extracted from a deep architecture.

The number of *salient* and *non-salient* training images is approximately 2000 for each salient object but it varies slightly. The largest *salient* class consists of 2418 images and the smallest 1700. Although the number of training observations for each salient object is quite large the variation may impact the capacity of the resulting content classification models. The variations in the content classification results is however more likely caused by the different content in the images.

As described for the quality classification in section 5.1.1, if one type of error is preferred over the other. In this case where one class is retained and the other is not, it might be more important not to discard too many *salient* images than to discard all *non-salient* images. Then, the threshold can be changed and the new rates can be retrieved from the ROC-curves in figure 4.2.

### 5.1.3 Similarity retrieval part

The similarity retrieval part gets an average accuracy of 75.50% with the best result being 78.06% and the worst 71.11%. The result varies with a few percentage points between the different salient objects and the variance in accuracy is 8.13%. That is most likely caused by the context of the salient objects rather than the objects themselves. That is because majority of the images consists of mostly context and the color coherence vectors

are calculated over the entire images. Applying a transformation to an image with a homogeneous background, still having the salient object present does not cause a change in the color coherence vector as big as it would be if the background were changing. This might explain why the two sets with the lowest resulting accuracy have the salient objects handbag and umbrella which are typically found in varying contexts such as crowds of people. The sets with the salient objects cat, motorbike and airplane has the best resulting accuracy. Those salient objects are often found in relatively homogeneous context such as indoor environment, roads and sky.

The similarity threshold was chosen from testing because it gave the best resulting accuracy on average for the different salient objects. As shown in the resulting similarity matrix for the sub-set of the category cat in figure 4.5, the resulting similarity values are dispersed across the spectrum. Therefore the results are very dependent on which threshold value is set. The value 87% is quite high which is why the recall value is in every case higher than the precision value. In this case where almost-duplicates are removed that means rather keeping a few similar images than risking the removal of unique images.

#### 5.1.4 The entire system

The evaluation of the entire system gives an average accuracy of 83.99% with the best result being 86.63% and the worst 80.27%. The result varies with a few percentage points between the different salient objects and the variance in accuracy is 7.99%. The classifications both have overall high precision values which means that they do not falsely classify many images as *good* or *salient*. That, and the proportion of wanted images being only 0.1859, together with the fact that most of the images should be removed during the classification steps is a probable cause for the high number of true negatives. For all sets most of the correct classifications are true negatives which, as shown in equations 3.1-3.3 affects the accuracy but not the precision and recall which explains why the accuracy is severely higher than the precision and recall. The accuracy values are also higher than the accuracy values for some of the content classification part and all for the similarity retrieval part separately. That is also most likely caused by the high number of true negatives when evaluating the entire system. The variance in accuracy being lower for the entire system than for the separate parts is probably another consequence of the high number of true negatives. One cause for the overall low precision and recall is that in the similarity retrieval part there is one more error cause when the system is put together. The image that is retrieved from each cluster is the one with the highest score from the classifications. All images in a cluster are thought to be equally *salient* since they all contain the salient object. The quality of the images are decided based on the SSIM values and since unmodified images have SSIM =1, only unmodified images retrieved are correct. In many cases an image retrieved from a cluster is modified to have SSIM slightly lower than 1 and is therefore counted as falsely classified. Although the quality classification scores lead to good classification result they might not correlate well enough to give an image of for example SSIM =0.99 lower quality score than an image of SSIM =1. Accepting any image being both *good* and *salient* being retrieved from each cluster would probably increase the precision and recall values.

## 5.2 Method

The biggest weakness in the system is the similarity retrieval part which resulted in lowest overall accuracy of the three parts of the system. The similarity retrieval method is relatively simple and if the thesis work would have been of bigger extent, a more advanced method could have been chosen. For the classifications, at least one feature extraction method provided good results for each part. Different feature extraction methods and predictor might have provided better results but when choosing such it is not often the case that one method is always outperforming the others but instead it varies much with data sets and tasks. Therefore the biggest remark in methods chosen is the data set. The data set used in this investigation is an example data set which differs in many ways from the data sets for which the system is supposed to be used. The images in the data set used are not automatically taken and are not part of the same continuously recorded set. One big difference between the data set used and a set of images that belong to a continuously recorded series, is that the background is typically more predictable in the latter. For images continuously recorded during a flight, the background may roughly consist of land, water and sky from afar in all images, meaning that the context is similar for all images. For the data set used however, the context in the images varies between indoor and outdoor scenes, in different places in the world and from different views. In the content classification, since entire images are set to *salient* or *non-salient*, it is much likely harder for the predictor to create an accurate classification model of saliency for the data set used where both objects and context varies much, compared to a data set where the context is more similar. That might explain why the category airplane shows better results in the content classification for all feature extraction methods. Airplanes which are typically found in more homogeneous context than the other categories, such as sky and airplane runways. The problem with the variety in context in the data set also affects the similarity retrieval part. If the context would be similar, the variety in objects present would have the major impact in the similarity measures, which is desired. Instead, with the data set used the context varies much and lower similarity measures are very often caused by variation in context rather than the salient object. Since so little is known about the data sets for which the system is supposed to be used, the investigation is very general. The more that is known about a problem, the more can the approach be specialized to solve it. Better results can probably be achieved when investigating quality if it is known what quality distortion types are prevailing, since methods can be chosen with more consideration.

## 5.3 Possible improvements

If one knows more about the data sets for which the system is supposed to be used many improvements are possible. For example, if it is known what kind of context that is typically prevailing during a flight, that information can be used to advance the similarity retrieval part. The color coherence matrix can be weighted so that colors typically appearing in the context of a planned flight can get a lower weight, giving a similarity measure which is less dependent on the context. The images might be processed by an automatic target recognition system during flights when collecting data but is not available for this study. Taking advantage of the results from such a system, the position of objects can be

found in images. That way, instead of investigating entire images only the parts where a potential salient object is found can be investigated.

The feature extraction method that provides the best results in the content classification is the one using features extracted from a pre-trained convolutional neural network. The network is not trained for the task on which it is evaluated but still outperforms the other methods used. That forebodes that using a convolutional neural network trained on the intended task might provide even better results in the content classification.

# 6

---

## Conclusions

Using features from the DCT domain together with the SVM classifier provided very good results in differentiating between *good* and *bad* quality in images. Using features extracted from a CNN together with the SVM classifier provided good results in differentiating between *salient* and *non-salient* content in images. The classifications together with the similarity retrieval part form the image selection system. The entire system provided acceptable results, but holds for improvement.

The results are acceptable for a selection system containing many steps but for the intended purpose they are however not good enough. Discarding an important image due to a false classification can result in fatal consequences if an important target is captured but dismissed. Even when changing the threshold in the classifications to prioritize avoiding the error of discarding too many images, higher accuracy is desired. Since the result varies with the sets having different salient objects it is much likely that it varies with data sets as well. The data set differs much from the data sets for which it is intended. A data set containing automatically taken flight data does not to the same extent have the problem of varying context which causes difficulties for some parts of the system. Therefore using the system on the intended data set might lead to substantially better results. For better results, more information than the raw pixel values should be used, for example what context is prevailing during a recording and where in the image a potential salient object is.



---

## Bibliography

- [1] Convolutional neural networks (lenet). URL <http://deeplearning.net/tutorial/lenet.html>. Cited on page 15.
- [2] B.H. Boyle. *Support Vector Machines: Data Analysis, Machine Learning, and Applications*. Computer science, technology and applications. Nova Science Publishers, 2011. ISBN 9781612093420. URL <https://books.google.co.uk/books?id=T7tAYgEACAAJ>. Cited on page 7.
- [3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. Cited on pages 15 and 18.
- [4] Dan C. Cireşan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two, IJCAI'11*, pages 1237–1242. AAAI Press, 2011. ISBN 978-1-57735-514-4. doi: 10.5591/978-1-57735-516-8/IJCAI11-210. URL <http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-210>. Cited on page 13.
- [5] R.L. Delanoy. Machine learning apparatus and method for image searching, August 11 1998. URL <https://www.google.com/patents/US5793888>. US Patent 5,793,888. Cited on page 1.
- [6] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013. URL <http://arxiv.org/abs/1310.1531>. Cited on page 15.
- [7] Eren Golge. How does feature extraction work on images? URL <https://www.quora.com/profile/Eren-Golge/Machine-Learning/How-does-feature-extraction-work-on-images>. Cited on page 5.
- [8] L. Greche and N. Es-Sbai. Automatic system for facial expression recognition based histogram of oriented gradient and normalized cross correlation. In *2016 International Conference on Information Technology for Organizations Development*

- (IT4OD), pages 1–5, March 2016. doi: 10.1109/IT4OD.2016.7479316. Cited on page 9.
- [9] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *ISCAS*, pages 253–256. IEEE, 2010. ISBN 978-1-4244-5309-2. URL <http://dblp.uni-trier.de/db/conf/iscas/iscas2010.html#LeCunKF10>. Cited on page 15.
- [10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL <http://arxiv.org/abs/1405.0312>. Cited on page 3.
- [11] MathWorks. Support vector machines for binary classification, . URL <https://se.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>. Cited on pages 6, 7, and 19.
- [12] MathWorks. ExtractHogFeatures, . URL <https://se.mathworks.com/help/vision/ref/extracthogfeatures.html>. Cited on page 9.
- [13] MathWorks. Discrete cosine transform, . URL <https://se.mathworks.com/help/images/discrete-cosine-transform.html>. Cited on page 10.
- [14] MathWorks. Supervised learning workflow and algorithms, . URL [https://se.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html?s\\_tid=conf\\_address\\_DA\\_eb](https://se.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html?s_tid=conf_address_DA_eb). Cited on page 5.
- [15] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. Cited on page 14.
- [16] Parul Parashar and Er. Harish Kundra. Comparison of various image classification methods. *International Journal of Advances in Science and Technology (IJAST)*, 2 (1), 2014. Cited on page 19.
- [17] Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM International Conference on Multimedia*, MULTIMEDIA '96, pages 65–73, New York, NY, USA, 1996. ACM. ISBN 0-89791-871-1. doi: 10.1145/244130.244148. URL <http://doi.acm.org/10.1145/244130.244148>. Cited on pages 16 and 19.
- [18] Srinu Penchikala. Big data processing with apache spark - part 4: Spark machine learning, May 2016. URL <https://www.infoq.com/articles/apache-spark-machine-learning>. Cited on page 4.
- [19] M.A. Saad, A.C. Bovik, and C. Charrier. Blind image quality assessment: A natural scene statistics approach in the dct domain. *IEEE Transactions on image processing*, 21(8), August 2008. Cited on pages 10, 11, and 19.

- 
- [20] F. Suard, A. Rakotomamonjy, and A. Bensrhair. Pedestrian detection using infrared images and histograms of oriented gradients. In *in IEEE Conference on Intelligent Vehicles*, pages 206–212, 2006. Cited on pages 9, 18, and 19.
- [21] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 13(4): 600–612, April 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861. URL <http://dx.doi.org/10.1109/TIP.2003.819861>. Cited on pages 18 and 22.