FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Adéla Kostelecká

Content-Based Image Retrieval: from Primitive to Advanced Techniques

Department of Algebra

Supervisor of the master thesis: Ing. Adam Novozámský, Ph.D.
Study programme: Mathematics
Study branch: Mathematics for Information Technologies

Prague 2022

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

Firstly, I would like to thank my supervisor Ing. Adam Novozámský, Ph.D., for patient guidance and support. I would also like to express gratitude to doc. RNDr. Elena Šikudová, Ph.D., for her willingness to consultations and to doc. Mgr. Pavel Příhoda, Ph.D. for his kindliness and a piece of advice. I am thankful to my family and closest friends for motivating me throughout my journey of this thesis.

Title: Content-Based Image Retrieval: from Primitive to Advanced Techniques

Author: Adéla Kostelecká

Department: Department of Algebra

Supervisor: Ing. Adam Novozámský, Ph.D.,

Abstract: The Wienbibliothek im Rathaus, Vienna City Library, collected over 300 thousand posters scanned in high quality from the last 100 years. Browsing and searching in such a large dataset is beyond human power. Therefore, a project was set up in cooperation with the Technical University of Vienna to test the possibilities of automatic data annotation on a selected sample. One of the requirements was Content-based Image Retrieval - retrieving images based on their visual content. This thesis reviews these techniques that emerged over the last decades. We focus on simple techniques based on colour, texture, and shape, as well as more advanced algorithms using convolutional neural networks. We implement these methods and compare their retrieval effectiveness on particular image datasets. Finally, we describe the functionality of a developed web application.

Keywords: Content-Based Image Retrieval, Image Features, Convolutional Neural Networks, Transfer Learning.

Contents

Chapter 1

Introduction

1.1 Motivation

Due to the exponential growth of digital image capturing devices and extensive image collections, the amount of digitally produced images rapidly increases over time. Moreover, the increase in network speed has made it possible to store a large amount of images. Based on the difficulties users may encounter while organising and searching large databases, the need for efficient image retrieval emerged.

In order to respond to this need, researchers have tried various approaches in text retrieval. There are some efficient search and retrieval engines based on textual descriptions of the images. Such tasks require humans to annotate each image in the database by text manually. This approach is not scalable, and it has become deficient and tedious.

Furthermore, due to the subjectivity of the human annotation process, the textual description may not be consistent or complete, which can deteriorate the retrieval performance. However, researchers have come up with an idea to index images based on their visual content. In the last few years, researchers have developed many image features using information about colour, texture, and shape.

The importance of content-based image retrieval is motivated by the increasing desire to retrieve images from growing image databases. It became popular in many domains such as medical imaging, weather forecasting and crime prevention. It originates from many different fields such as statistics, pattern recognition, computer vision and more.

1.2 Outline

This thesis aims to investigate the techniques used in content-based image retrieval, from the earliest simple methods to state-of-the-art methods. At first, we research the methods of generating simple image descriptors based on colour, texture and shape. Afterwards, we explain the principle of neural networks and the known widely-used architectures. We implement several methods from both spheres of interest. We evaluate our experiments and compare the methods' behaviour and performance on different kinds of images. The main goal of this thesis is to develop an image retrieval system for Vienna City Library. In addition, we implement a web application that serves as an efficient tool for image retrieval on a particular image dataset of posters provided by Vienna City Library.

Chapter 2 introduces a task of content-based image retrieval. We explain the principle of a general content-based image retrieval system. We formulate a task mathematically and state definitions of metric space and distance measures. After all, we define evaluation metrics frequently utilised in retrieval systems.

We focus on research of primitive feature extraction techniques in Chapter 3. We name these approaches handcrafted features and divide them into three categories: colour, texture and shape. For each category, we define three approaches providing image features. These methods are not deployed in modern retrieval systems. However, the ideas behind them are beautiful and helpful to understanding more sophisticated approaches. Some of the techniques are based on image convolution, which forms a basis of convolutional neural networks (see Chapter 4).

Next, we explain deep neural networks in general and convolutional neural networks in Chapter 4 extensively used in applications of computer vision. Due to its tremendous success in computer vision applications during the last decade (mainly since the work of Krizhevsky et al. [2012]), these models replaced traditional approaches. A deep understanding of neural networks requires a strong mathematical background in mathematical analysis, linear algebra, numerical methods in optimisation and probability. However, we strive to clarify the fundamental knowledge to a common reader. We describe how to use these networks for image retrieval tasks and summarise existing pre-trained networks' architectures.

Dimensionality reduction techniques described in Chapter 5 are utilised in many applications. Principal component analysis represents one of the oldest such techniques. It employs linear projection on the lower-dimensional space preserving the maximum amount of information. Another described method called t-SNE is proper when visualising high-dimensional vectors in low-dimensional space. It employs advanced mathematics and tries to preserve the local and global structure of the data. It enables humans to understand the arrangement of the data in high-dimensional space. We utilised t-SNE in Chapter 6 to show retrieval results on a two-dimensional plot.

In Chapter 6, we summarise the results of our implementation and compare methods with various parameters on introduced datasets. We implemented both handcrafted features and models based on neural networks. Afterwards, we present a technique for improving retrieval effectiveness, widely known as fine-tuning neural networks. Moreover, we show t-SNE plots of high-level image descriptors.

We provide an overview of our web application implementation in Chapter 7. It is designed to perform image retrieval on an image chosen by a user and retrieves images from an image dataset collected from posters from the Vienna City Library. We explain its principle and briefly summarise the used technologies.

In the end, we summarise this thesis in Chapter 8. Moreover, we provide a list of suggestions for future work. Lastly, we refer to the work of state-of-the-art researchers using the most advanced existing approaches in content-based image retrieval.

Chapter 2

Content-Based Image Retrieval

This chapter focuses on the fundamental knowledge necessary to understand content-based image retrieval. We explain the principle of a typical content-based image retrieval pipeline and introduce a task with mathematical definitions of important notions included. In the end, we familiarize with distance measures and metrics used for evaluation.

2.1 Standard CBIR pipeline

Content-based image retrieval (CBIR) represents a technique to extract image features based on visual content. In other words, each image is indexed based on its visual properties, like colour, texture and shape. The main goal of CBIR is to find the most similar images to an image defined by a user from a given database. Therefore, the images need to be characterized efficiently to keep similar images close in terms of distance.

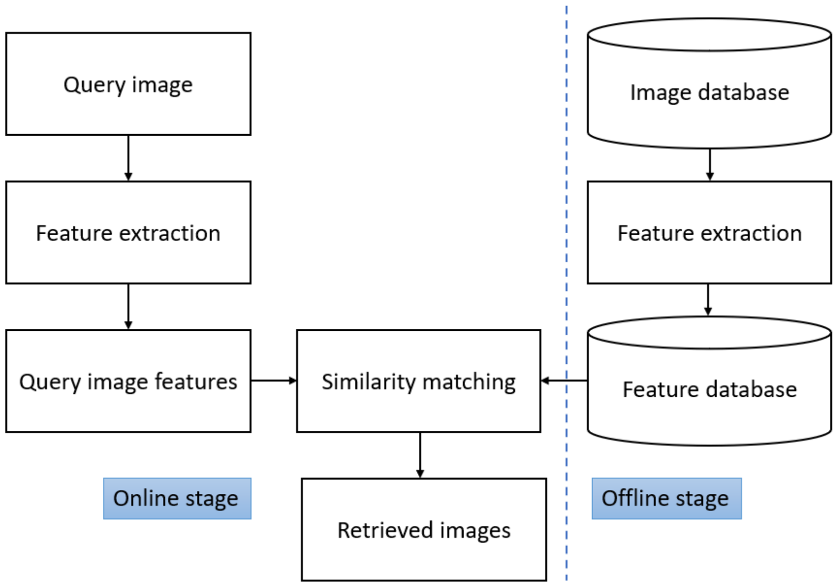


Figure 2.1: Typical CBIR pipeline. It consists of offline on right and online processing part on left and centre.

A standard CBIR system usually follows the pipeline shown in Figure 2.1.

Input is the query image defined by a user, and output is the most similar to a given query image. The architecture consists of the online and offline processing stages. In the offline stage, database features are pre-computed to get image descriptors of each image stored in the feature database. The online stage is composed of two parts. At first, the image features are extracted. Afterwards, similarity matching is performed on the query image descriptor and each database image descriptor.

A user typically inserts an image after the database image descriptors are pre-computed. The goal of the CBIR system is to retrieve the desired images similar to this query from the provided image database. All database images are ranked according to the similarity to the query image given by a chosen algorithm. The pipeline output is the first N images for the N defined by the user.

2.2 Task formulation

In order to describe the problem clearly, we need to state the elementary definitions.

Colour digital images are represented as 3-dimensional matrices with a given height, width and the number of colour channels. Image representation depends on the choice of the colour space, which is a specific organization of colours. In other words, it is a model used to represent as many colours as the human vision system can perceive. Many colour spaces have been developed (Ibraheem et al. [2012]). In this thesis, we represent images in trichromatic colour spaces such as RGB or HSV.

Since there is a finite number of intensity values for each colour channel c, a set of all possible images with a given height h and width w is finite. This set determined by a height h, a width w and a number of colour channels c is given by:

$$\mathcal{U}_{h,w,c} = \{0, 1, \ldots, 255\}^{h \times w \times c}$$

A digital image of a height h and a width w with three colour channels can be understood as an element of $\mathcal{U}_{h,w,3}$, abbreviated as $\mathcal{U}$. We define it as a three-dimensional matrix.

Definition 1 (Digital image)**.** *Digital image I of height h and width w under a given colour space with c colour channels is three-dimensional matrix of size $h \times w \times c$ with elements from set* $\{0, 1, \ldots, 255\}$.

The digital image usually has three colour channels. However, it may be in a grayscale image, which means that it has only one colour channel. Suppose we are given query image Q and a set of database images $\mathcal{D}$. As mentioned in the previous section, each database image corresponds to an image descriptor based on its content after feature extraction. The query descriptor is compared to each database image descriptor to measure the similarity between the query image and each database image. Images evaluated with the lowest distances are retrieved.

2.3 Feature space

The image features need to be captured effectively to describe the significant image properties. Good features have discriminating properties, meaning that they can distinguish one image from other dissimilar images. It also needs to be as robust as possible to prevent generating many different features for similar images with a similar object. A feature space $\mathcal{F}$ refers to the collections of features characterizing the image data.

Definition 2 (Feature space). *Feature space $\mathcal{F}$ is a vector space isomorphic to $\mathbb{R}^n$ for $n \in \mathbb{N}$.*

In the context of this work, by feature extraction, we mean extracting information based on the visual content. Feature extraction is a mapping that usually reduces the number of parameters. The main goal of feature extraction is to obtain the most relevant descriptors of the original data to represent that information in a lower dimensionality space. Efficient feature extraction is a research problem, and we describe some of the possible approaches in the following chapters.

In the following two sections, the definitions originate mainly from the master thesis of Bátoryová [2020]. We will define a notion of descriptor extraction function, a mapping from the image space $\mathcal{U}$ to the feature space $\mathcal{F}$.

Definition 3 (Descriptor extraction function). *Let $\mathcal{U}$ be the set of all possible images and $\mathcal{F}$ the feature space. A function $f_{ext} : \mathcal{U} \to \mathcal{F}$ is the descriptor extraction function. We say that vector $f_{ext}(I)$ is an image descriptor of image I with given descriptor extraction function f_{ext}.*

An image descriptor is extracted from each database image and is used to create an indexed database. The descriptor extraction function describes the properties of colour, texture or shape of an image and what is more interesting, we can obtain it by a neural network. We use various strategies to extract these image properties and diversely construct neural networks.

2.4 Distance and metric space

After feature extraction of the query image and the image database, database features are compared to the query feature to measure the dissimilarity between images. To do so, we need to define the distance space and the metric space properly. Comparing images means comparing their descriptors using distance measure d.

Definition 4 (Distance space). *A distance space is an ordered pair $(\mathcal{P}, d)$ where $\mathcal{P}$ is a set and d is a distance on $\mathcal{P}$, i.e., function $d : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_0^+$ such that for any $x, y \in \mathcal{P}$, the following holds:*

1. *$d(x, y) = 0 \iff x = y$ (identity of indiscernibles)*

2. *$d(x, y) = d(y, x)$ (symmetry)*

Definition 5 (Metric space)**.** *A metric space is an ordered pair $(\mathcal{P}, d)$ where $\mathcal{P}$ is a set and d is a metric on $\mathcal{P}$, i.e., function $d : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_0^+$ such that for any $x, y, z \in \mathcal{P}$, the following holds:*

1. *$d(x, y) = 0 \iff x = y$ (identity of indiscernibles)*
2. *$d(x, y) = d(y, x)$ (symmetry)*
3. *$d(x, z) \leq d(x, y) + d(y, z)$ (subadditivity or triangle inequality)*

In comparison to the distance space, metric space satisfies the triangle inequality. From the above three axioms of metric space, a condition $d(x, y) \geq 0$ for any $x, y \in \mathcal{P}$ is satisfied. The typical example of a metric space is the defined feature space considered to be isomorphic to $\mathbb{R}^n$. Measuring the similarity has the opposite effect of measuring the distance. Therefore, the most similar vectors are considered the closest ones in terms of distance.

2.5 Distance measures

In order to perform similarity matching, we define numerous metrics. There are two kinds of them: distances and similarities, depending on whether we consider smaller or larger values as similar. At first, we define an L_p norm, denoted by $\|\cdot\|_p$. A special case is commonly used L_2 norm and its notation can be abbreviated as $\|v\|$ instead of $\|v\|_2$.

Definition 6 (L_p norm)**.** *For a given $v \in \mathbb{R}^n$ we define the L_p norm of v as:*

$$\|v\|_p = \sqrt[p]{\sum_{i=1}^{n} v_i^p}.$$

Distance measures widely used in image retrieval are the following ones.

Definition 7 (Manhattan distance)**.** *For given $u, v \in \mathbb{R}^n$ we define the Manhattan distance as:*

$$d_{man}(u, v) = \|u - v\|_1 = \sum_{i=1}^{n} |u_i - v_i|.$$

Definition 8 (Euclidean distance)**.** *For given $u, v \in \mathbb{R}^n$ we define the Euclidean distance as:*

$$d_{euc}(u, v) = \|u - v\|_2 = \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}.$$

Definition 9 (Cosine similarity)**.** *For given $u, v \in \mathbb{R}^n$ we define the cosine similarity as:*

$$s_{cos}(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}.$$

We define the cosine distance using a transform $d = 1 - s$.

$$d_{cos}(u, v) = 1 - s_{cos}(u, v)$$

2.6 Evaluation metrics

In this section, we mean images similar to a given query by relevant images. In order to evaluate the retrieval performance, a set of query images is defined beforehand. We denote this set as a test set of queries.

The performance of the retrieval system can be evaluated in terms of offline metrics such as precision and recall. Precision is the ratio of the number of retrieved relevant images and the number of all of the retrieved images. In contrast, recall represents the ratio of the number of retrieved relevant images and the number of relevant images. Alternatively, the F_1 score is utilized, which combines precision and recall by computing a harmonic mean of them. However, for modern information retrieval, recall is no longer a meaningful metric since queries have thousands of relevant documents, and few users will be interested in reading all of them.

The precision at K represents a percentage of relevant images in the top K images. Since the precision at K depends on an application, it needs to be averaged on different values of K. In terms of average precision, these values of K are the positions where another relevant image appears. Let N be the number of retrieved images and R the number of relevant images that are retrieved. Then precision is defined as a fraction $\frac{R}{N}$. Equivalently, suppose we denote True Positives as TP, False Positives as FP, False Negatives as FN and True Negatives as TN. In that case, precision is computed as a ratio $\frac{TP}{TP+FP}$. Accuracy is calculated as a ratio $\frac{TP+TN}{TP+TN+FP+FN}$. Precision at K is $(P@K)$ is computed as a percentage of relevant images in the top K images.

Definition 10 (Precision at K)**.** *Let $R_K(I)$ be the number of relevant images in top K images for an image I. Precision at K $(P@K)$ is defined as:*

$$P@K(I) = \frac{R_K(I)}{K}.$$

The precision at K for a set of test queries is averaged precision at K over all test queries.

For systems that return a ranked sequence of images, it is desirable also to consider the order in which the returned documents are presented. Average precision (AP) summarizes precisions at K for different values of K at the positions of the relevant images.

Definition 11 (Average precision)**.** *Let R be the number of relevant images for an image I. Let $P@k$ be an precision at k for $k = K_1, \ldots, K_R$, where K_i is an $i - th$ relevant position. Average precision for an image I is defined as:*

$$AP(I) = \frac{1}{R}\sum_{i=1}^{R} P@K_i(I).$$

When it comes to measuring the performance of the whole retrieval system, the average precision is averaged through all the test queries Q. This leads us to a notion of the mean average precision (mAP), a widely used metric in information retrieval tasks.

Definition 12 (Mean average precision)**.** *Let Q be the number of test queries and $AP(I_i)$ be the average precision for query image I_i for each $i = 1, \dots, Q$.*

Mean average precision is defined as:

$$mAP = \frac{1}{Q} \sum_{i=1}^{Q} AP(I_i).$$

Chapter 3

Handcrafted features

This chapter introduces several image descriptors, widely used before the advent of the state-of-the-art techniques based on neural networks described in Chapter 4. Handcrafted features are image descriptors based on colour, texture, or shape properties. Its usage depends on a particular problem and dataset. In some applications, we want to retrieve images with similar colour properties. The spatial arrangement of pixel intensity values plays a vital role in texture analysis and retrieval. It is useful when dealing with greyscale images, where colour information is unavailable.

3.1 Related work

In the last few decades, many researchers have tried to find algorithms to extract features that describe an image efficiently (Stockman and Shapiro [2001]). Handcrafted methods consider colour, texture and shape information. Typical image descriptors depend on more of these categories. In this chapter, we summarise different approaches, and we discuss the implemented results in Chapter 6.

Colour features pose low-level visual features invariant to image size and orientation. It depends on a chosen colour space. Colour histograms represent the most essential colour features appeared in content-based image retrieval (Kumar and Saravanan [2013]). These can be global or local (Pant [2013]) depending on the size of the region of an image considered. It is commonly computed on the reduced set of image colours. This process is known as quantisation. Colour moments, simple statistical features such as mean, standard deviation, and skewness provide a measure of the degree of asymmetry in the distribution (Singh and Hemachandran [2012]). Colour coherence vector developed by Pass et al. [1997] generalises colour histogram by taking into account the coherency of each coloured pixel. It used in CBIR Al-Hamami and Al-Rashdan [2010]. Other methods include colour correlogram (Huang et al. [1997]), and colour co-occurrence matrix looking at co-occurrences of colours (Lin et al. [2009]). Dominant colour descriptor is determined by a prescribed number of dominant colours of an image and its corresponding percentage (yang Wang et al. [2011], Rashno and Rashno [2019]). Spatial chromatic histogram based on the work of Cinque et al. [2001] represents a generalisation of histogram specifying a relevant average position and a variance of each colour. The feature was employed in the context of content-based image retrieval (Gavrielides et al. [2006]).

Methods based on image texture are usually applied to an image with one channel. The image is converted to greyscale or uses only one component from colour channels. Texture analysis came up with one of the earliest methods based on grey-level co-occurrence matrix (GLCM) based on Haralick et al. [1973]. Its properties are known as Haralick features forming a feature vector. Next interesting texture features employed image filters, such as Gabor filters (Singh et al. [2018], Zhang et al. [2000]) or Discrete Wavelet Transformation (DWT) (Agarwal et al. [2013], Rashno and Rashno [2019]). Ojala et al. [2000] developed an approach called Local Binary Patterns (LBPs) describing the local pixel's neighbourhood. A simple generalisation of LBP is Local Tetra Pattern considering the character of the circular pixel neighbourhood concerning the centre pixel (Murala et al. [2012]). Other techniques look for the ordering of pixel values in pixel's neighborhood are known as Scan Pattern Co-occurrence Matrix (SPCM) (Rao et al. [2011]) or Scan Pattern Internal Pixel Difference (SPIPD) (Rao et al. [2011]).

Regarding shape methods, histograms of oriented gradients (HOG) Dalal and Triggs [2005] were developed to detect the pedestrians based on the orientation and magnitude of the image gradient. Discrete image derivatives provide meaningful image features. Local features such as SIFT and SURF were also used in CBIR Alkhawlani et al. [2015].

3.2 Convolution

The convolution of an image represents a significant milestone. It is computed using a kernel, a matrix of typically smaller size than the image size. An image is typically padded by zeros outside its border. In the following definitions, the sum is computed over the size of a kernel, and the output is of the same size as an original image.

Definition 13 (2D Discrete Convolution)**.** *Given an image I of size $h \times w$, padded with zeros, one colour channel, and a kernel K. Convolution of an image I with a kernel K is a matrix of type $h \times w$ defined as*

$$(K * I)_{i,j} = \sum_m \sum_n I_{i-m,j-n} K_{m,n}$$

Definition 14 (2D Discrete Cross-Correlation)**.** *Given an image I of size $h \times w$, padded with zeros, one colour channel, and a kernel K. Cross-correlation of an image I with a kernel K is a matrix of type $h \times w$ defined as*

$$(K \star I)_{i,j} = \sum_m \sum_n I_{i+m,j+n} K_{m,n}$$

Cross-correlation is a convolution without flipping the kernel, and this operation is applied to images. We follow the convention of calling both operations convolution and cross-correlation as convolution (see Goodfellow et al. [2016]). The first argument K is referred to as a kernel, the second argument I as an input image and the output $(K \star I)$ is called a feature map. When considering the convolution of an image with more colour channels, the convolution is applied to each colour channel separately, and we may define different kernels for different colour channels.

3.3 Colour methods

Colour perception depends upon both the physics of the light and complex processing by the eye-brain, which integrates properties of the stimulus with experience. Colour features are computationally the most simple and play an important role in image analysis, image retrieval and related fields.

The encoding of an arbitrary colour in the visible spectrum is made by combining and encoding three primary colours, red, green and blue (RGB) or other colour space such as HSV or L*a*b* compared in Figure 3.1.

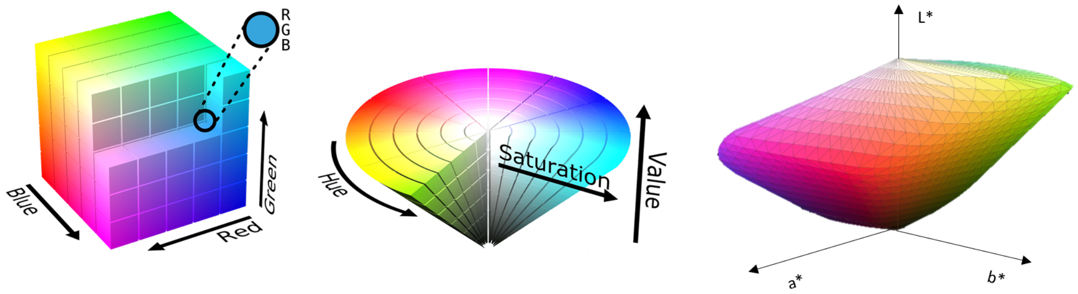


Figure 3.1: Colour systems: RGB, HSV and L*a*b* (Commons)

The RGB colour system is an additive colour system in which the red, green, and blue primary light colours are added together. HSV colour system is more adapted to human perception of colour. It has been reported that the HSV colour space gives the better colour histogram feature among the different examined colour spaces (Singha and Hemachandran [2012]). HSV is based on cylinder coordinates. Hue represents the dominant wavelength in the light; saturation represents the dominance of hue in colour, and value is defined as a relative lightness or darkness of a colour. Derivation of the transformation from RGB coordinates to HSV coordinates is based on the algorithm from the work of Stockman and Shapiro [2001]. Another colour space denoted by L*a*b* was intended as a perceptually uniform space, where a given numerical change corresponds to a similar perceived change in colour. However, it has a different range than RGB and HSV.

In a common image representation where each channel has 256 intensity levels, every pixel can have 256^3, over 16 million colours. Humans cannot distinguish between this large amount of various colours; they can see only a million colours. So it is not necessary to distinguish between every single colour. Moreover, considering fewer colours is sufficient and computationally more efficient. Thus we employ image quantisation to reduce the number of colours of the colour feature components. The general goal of quantisation is to reduce colour space without significantly affecting the visual properties of an image. It is applied to provide a trade-off between the accuracy of the image representation and memory requirements.

Quantisation is a reduction of the number of colours of an image. There are several commonly used quantisation techniques: predefined palettes (Macbeth or Fibonacci palette), uniform quantisation, median cut quantisation algorithm, quantisation computed by clustering, or octree quantisation. For the purpose of image retrieval, it makes sense to keep the same palette to compare corresponding colour features.

3.3.1 Colour histogram

Consider that we are given a colour space and quantised image. A colour histogram provides the occurrence of colour in an image I. It can be computed globally or locally in fixed regions or regions obtained by image segmentation. Their popularity stems from being computationally trivial and almost translation-invariant, meaning that small changes in camera viewpoint do not affect the computed feature. They are partially invariant to the rotation about the imaging axis, small off-axis rotations, scale changes, and partial occlusions (Stockman and Shapiro [2001]). It depends only on the colour properties of an image without providing any information on the spatial distribution of colours, so it merely describes which colours are present in the image. This chapter considers a digital image as a function from its domain to a set of colours.

Remark (Digital image as a function). *A digital image with three colour channels can be understood as a function as* $I : \{1, \dots, h\} \times \{1, \dots, w\} \to \{0, \dots, 255\}^3$.

Definition 15 (Global colour histogram). *Given an image I of size $h \times w$ with C possible image colours. Let k_c be an index of a colour c, so that $k_c = 1, \dots, C$. The global image histogram H_I of an image I is a C-dimensional vector defined as*

$$(H_I)_{k_c} = |\{I(i, j) = c \mid i \in \{1, \dots, w\}, j \in \{1, \dots, h\}\}|$$

A global normalised colour histogram forms a probability distribution of image colours. It can be visualised as a bar graph, in which each bar represents a particular colour density. Many different distances have been proposed to measure the similarity between two image histograms, such as intersection distance proposed by Swain and Ballard [1990], Euclidean distance, and histogram quadratic distance (van den Broek [2005]) incorporating the similarity matrix.

A local colour histogram divides an image into fixed blocks and calculates the colour histogram of each of those blocks. By concatenating local colour histogram, we obtain another colour feature. Its special case is a feature vector represented by the average colour in each block.

Definition 16 (Grid average colour). *Given an image I of size $h \times w$. Let $k', l' \in \mathbb{N}$ and $k = \lfloor \frac{h}{k'} \rfloor$ and $l = \lfloor \frac{w}{l'} \rfloor$. Grid average colour of an image I is defined as*

$$(G_I)_{m,n} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{l} I(i + m \cdot k, j + n \cdot l),$$

where $m = 0, \dots, k-1$ and $n = 0, \dots, l-1$.

A grid colour feature is a vectorisation of grid average colour. Since we assume that the number of grid blocks is significantly smaller than the size of an image, we can omit boundary pixels from the definition.

3.3.2 Colour coherence vector

Since global colour histograms lack spatial information, a more complex histogram-based approach was developed by Pass et al. [1997] named colour coherence vector (CCV). It is a refined colour histogram method providing additional information about colour coherence. A colour coherent pixel belongs to a large similarity coloured region, whereas an incoherent does not. Let us introduce a notion of a connected component of an image.

Definition 17 (Connected component). *A connected component C is a maximal set of pixels such that for any two pixels $p, p' \in C$, there is a path in C between p and p'. Path is a sequence of pixels of the same colour $p = p_1, p_2, \ldots, p_n = p'$, such that each pixel p_i is in C and any two sequential pixels p_i, p_{i+1} are adjacent to each other (meaning that if one pixels is among the eight closest neighbours of the other).*

Connected components of an image I can be computed in linear time with respect to the number of all image pixels. For each pixel, we remember the number of pixels in a connected component, and we call it the size of the connected component denoted by S_C.

Definition 18 (Coherent pixel). *A pixel is coherent with respect to a given threshold τ if the size S_C of its connected component C satisfies $S_C \geq \tau$. Otherwise, we say the pixel is incoherent.*

The Colour coherence vector generalises global colour histogram by computing colour histogram for one thing of coherent pixels and another of incoherent pixels. Although the following two definitions depend on τ, we consider τ as a predefined constant.

Definition 19 (Colour coherence vector). *Given a digital image I with V image colours. Let α_i be the number of coherent pixels of the i-th colour and β_i be the number of incoherent pixels of the i-th colour. The colour coherence vector C_I of an image I is a $2V$-dimensional vector defined by*

$$C_I = (\alpha_1, \beta_1, \ldots, \alpha_V, \beta_V)$$

Definition 20 (Colour coherence vector distance). *Let us assume that each image has the same number of pixels. Given an image I and I' together with their corresponding colour coherence vectors $C_I = (\alpha_1, \beta_1, \ldots, \alpha_V, \beta_V)$, $C_{I'} = (\alpha'_1, \beta'_1, \ldots, \alpha'_V, \beta'_V)$ we define a colour coherence vector distance as:*

$$d_{CCV}(C_I, C_{I'}) = \sum_{i=1}^{V} |\alpha_i - \alpha'_i| + |\beta_i - \beta'_i| = \|C_I - C_{I'}\|_1$$

The effectiveness of the Colour Coherence Vector was improved by Al-Hamami and Al-Rashdan [2010]. The original CCV method does not tell about the existence of dissimilarity between images. Therefore, it exploits the location information, such as the number of coherence regions of the same colour for each colour. Next possible improvements in colour coherence vector were investigated in the work of Al-Hamami and Al-Rashdan [2010], Singh et al. [2018].

3.3.3 Spatial chromatic histogram

Some histograms are based only on counting quantised colours; others incorporate spatial colour distribution information. Spatial chromatic descriptor generalised the traditional global colour histogram by taking into account another property of colours present in an image. An original idea of a spatial chromatic histogram was proposed by Cinque et al. [2001] and employs not only the occurrences but also the relative average position of every single colour and the standard deviation of each colour.

Definition 21 (Spatial chromatic descriptor)**.** *Let I be an image of size $h \times w$ and h_I be its normalised colour histogram. Let $a_I(c)$ be the absolute number of the pixels in an image I having c-th colour, $c = 1, \dots, C$.*

We define a barycenter of i-th colour $(b_I)_i = (\bar{x_i}, \bar{y_i})$, where

$$\bar{x_i} = \frac{1}{w}\frac{1}{a_I(i)} \sum_{I(x,y)=i} x$$

$$\bar{y_i} = \frac{1}{h}\frac{1}{a_I(i)} \sum_{I(x,y)=i} y$$

The standard deviation is defined as:

$$(\sigma_I)_i = \sqrt{\frac{1}{a_I(i)} \sum_{I(p)=i} d(p, (b_I)_i)^2},$$

where p denotes pixel in relative coordinates (in range $[0, 1]$), the the sum is over all pixels p having a colour i, $d(p, (b_I)_i)$ is Euclidean distance between a pair of pixels p. It gives a measure of how the pixel spreads around the barycenter. The spatial chromatic histogram is defined as:

$$S_I = (h_I, b_I, \sigma_I).$$

Note that the spatial chromatic histogram is a $3C$-dimensional vector. Distance measure was designed by Cinque et al. [2001] as in the following definition.

Definition 22 (Spatial chromatic distance)**.** *Given two images I_1, I_2. The spatial chromatic distance between images I_1 and I_2 is defined as:*

$$d_{sch}(I_1, I_2) = \sum_{i=1}^{C} \min((h_{I_1})_i, (h_{I_2})_i) \cdot \left(\frac{\sqrt{2} - d((b_1)_i, (b_2)_i)}{\sqrt{2}} + \frac{\min((\sigma_1)_i, (\sigma_2)_i)}{\max((\sigma_1)_i, (\sigma_2)_i)} \right)$$

Let us assume that each colour exists on one of the images at least in two pixels to avoid the expression of 0/0. The spatial colour descriptor was utilised for the purpose of content-based image retrieval in the work of Gavrielides et al. [2006]. Note that spatial chromatic histogram requires slightly different definition of digital image since it is defined on the CIELAB colour space, also referred to as L*a*b*.

3.4 Texture methods

Texture features give information about the spatial arrangement of the colours or intensities in an image (Stockman and Shapiro [2001]). It can be found in natural scenes and also artificial objects.

The widely known texture features are Haralick features describing a global representation of a texture computed from Grey-Level Co-occurrence Matrix based on the work of Haralick et al. [1973], and Local Binary Patterns invented by Ojala et al. [2000] based on local changes in intensity values. Extracted features describe the regularity, coarseness or geometric properties of an image. In order to build texture features, the image is usually converted into greyscale, but it may be applied to all colour components. For instance, some approaches extracted texture features from the value component of the HSV image.

3.4.1 Grey-Level Co-occurrence Matrix

Haralick et al. [1973] introduces Gray-Level Co-Occurrence matrix (originally defined as Grey-Tone Spatial Dependence Matrix) that became one of the earliest techniques used in texture analysis. It has been widely used in many applications in texture classification, and it poses a well-known textural feature used in image retrieval. Haralick's idea was to introduce a matrix to measure the occurrence of adjacent pixel values with a given direction forming a given angle and use it to extract a set of textural features. Haralick defined a set of 14 features. It was used to solve an image classification task performed on photomicrographs of sandstones (automated classification of rocks into six categories), aerial photographic and satellite imagery.

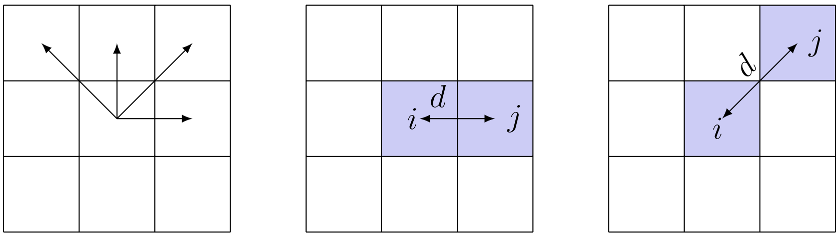


Figure 3.2: Four displacement vectors proposed by Haralick. The original GLCM was defined as a symmetric and it did not depend on the ordering of a co-occurrence.

A co-occurrence matrix is stored as a two-dimensional array C. The value of $C(i,j)$ indicates how many times value i co-occurs with value j in some designed spatial relationship. This relationship is given by a vector d specifying the displacement between these two pixels with values i and j.

Definition 23 (Grey-level co-occurrence matrix)**.** *Given a grey-scale image I with V grey levels. Let $d = (d_r, d_c)$ be a displacement vector where d_r is a displacement in rows (downward) and d_c is a displacement in columns (to the right). The grey-level co-occurrence matrix $C_{I,d}$ for image I is a matrix of type $V \times V$, in which each position (v_1, v_2) is defined by*

$$(C_{I,d})_{v_1,v_2} = |\{(i,j) \mid I(i,j) = v_1 \wedge I(i+d_r, j+d_c) = v_2\}|,$$

if $I(i,j)$ and $I(i+d_r, j+d_c)$ is defined. Grey-level co-occurrence matrix is commonly abbreviated as GLCM.

Definition 24 (Normalized GLCM)**.** *The normalized GLCM $N_{I,d}$ of type $V \times V$ of a given image I and displacement vector d is defined by*

$$(N_{I,d})_{i,j} = \frac{(C_{I,d})_{i,j}}{\sum_i \sum_j (C_{I,d})_{i,j}},$$

where $i, j = 1, \ldots, V$.

It can be thought of as a generalised histogram where each position represents not an occurrence density but a co-occurrence density. The normalised grey-level co-occurrence matrix values represent probabilities of pixel value co-occurrences.

Given the GLCM, its statistics are computed: contrast, dissimilarity, homogeneity, angular second moment, energy, correlation and entropy. Haralick presented a set of 14 features; we define just 5 of them, which we implemented in our experiments in Chapter 6. For more features, we refer to the original paper of Haralick et al. [1973].

Definition 25 (Haralick's features)**.** *Let $N_{I,d}$ be the normalized GLCM for an image I with V grey levels, $N_{I,d} = (a_{i,j})_{i,j=1}^{V,V}$. Then the features of GLCM $N_{I,d}$ are defined as:*
Angular second moment:

$$f_{asm} = \sum_{i=1}^{V} \sum_{j=1}^{V} a_{i,j}^2$$

Contrast:

$$f_{con} = \sum_{i=1}^{V} \sum_{j=1}^{V} a_{i,j} (i-j)^2$$

Dissimilarity:

$$f_{dis} = \sum_{i=1}^{V} \sum_{j=1}^{V} a_{i,j} |i-j|$$

Homogeneity:

$$f_{hom} = \sum_{i=1}^{V} \sum_{j=1}^{V} a_{i,j} \frac{1}{1+(i-j)^2}$$

Correlation:

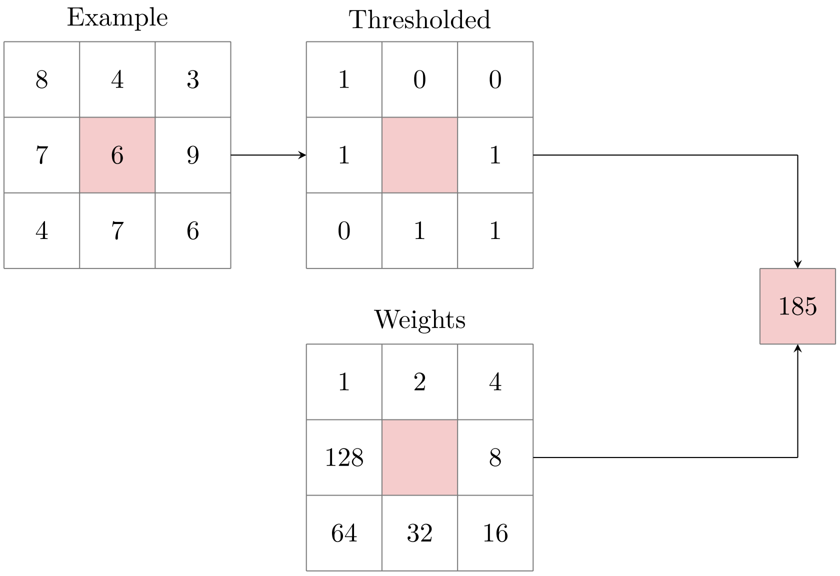
$$f_{cor} = \frac{\sum_{i=1}^{V} \sum_{j=1}^{V} ij a_{i,j} - \mu_x \mu_y}{\sigma_x \sigma_y},$$

where $\mu_x = \sum_{i,j} i a_{i,j}$, $\mu_y = \sum_{i,j} j a_{i,j}$, $\sigma_x = \sum_{i,j} (i-\mu_x)^2 a_{i,j}$ and $\sigma_y = \sum_{i,j} (j-\mu_y)^2 a_{i,j}$.

Haralick features were generalised (Vadakkenveettil [2012]) to the trace feature, which outperforms Haralick features in the context of the content-based image retrieval. The purpose of the trace feature is to identify constant regions in an image. The sum of its diagonal elements defines the trace of a GLCM.

3.4.2 Local binary patterns

Local binary patterns (LBPs) are computationally simple features used to describe a local representation of texture applied to a two-dimensional image. The original idea comes from the work of Ojala et al. [2000] and later generalized by Ojala et al. [2002]. They characterise the spatial configuration of local image texture by using the information in a small neighbourhood. Their advantage is invariance against monotonic transformations and robustness in the greyscale. The rotational invariance can be achieved by slightly modifying the most straightforward presented approach. It can be improved by choosing a limited subset of ”uniform” patterns instead of all rotation invariant patterns.



Example

8	4	3
7	6	9
4	7	6

Thresholded

1	0	0
1		1
0	1	1

Weights

1	2	4
128		8
64	32	16

Figure 3.3: Local Binary Pattern computation. Example: binary number 10111001 to decimal number as $1 + 8 + 16 + 32 + 128 = 185$.

Their disadvantage is ignoring the global spatial information of the image texture. Moreover, LBPs discard the contrast property and assume the independence of the central pixel value and its differences with neighbouring pixels, which is not warranted in practice. The local binary pattern values are stored in the two-dimensional array with the same width and height as an original image.

To explain the main idea, let us consider a 3×3 pixel neighbourhood and threshold each pixel against its given neighbourhood as shown in Figure 3.3. Thresholded neighbouring pixel values are concatenated to form a binary string. For each pixel, there are eight neighbouring pixels; thus, there are $2^8 = 256$ possibilities of a binary string. This binary string is converted to decimal to get a number between 0 and 255 for each pixel.

The LBPs were generalised to the regular circular neighbourhood of radius of any size and any number of points. Furthermore, a discrete image domain is expanded to a continuous image domain and the discrete set of grey values to the continuous interval $[0, 255]$ and bi-linear interpolation on pixel values (Ojala et al. [2002]).

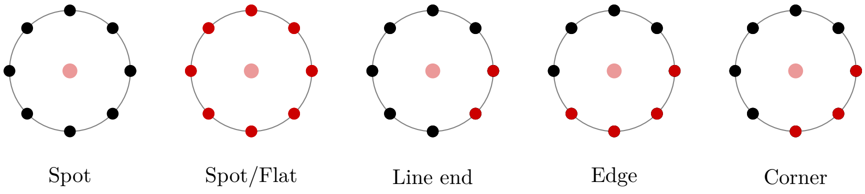


Figure 3.4: Local binary patterns: types of neighbourhood.

Definition 26 (Local binary pattern)**.** *Given an image I with continuous domain and range of grey-levels obtained by bi-linear interpolation. Let (i, j) be an image position, where $I(i, j) = g_c$, a neighbourhood size R, the number of examined points P and an angle $\alpha = 2\pi/P$. Let g_p, $p = 0, \ldots, P-1$ be computed by*

$$g_p = I(i + R\cos(p\alpha), j + R\sin(p\alpha))$$

The local binary pattern for an image I in a position (i, j) is defined as

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p$$

where $s(x)$ is the sign function given by

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The local binary pattern is computed for each position (i, j) of the image I.

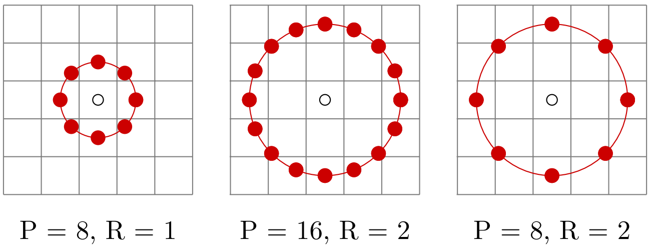


Figure 3.5: LBPs: Examples of a given number of points P and radius R.

In the context of image processing, monotonic transformation applied to an image can be a change of a contract or brightness of an image.

Theorem 1. *The $LBP_{P,R}$ is invariant against monotonic transformation.*

Proof. The sign function s of the difference of two neighbouring pixel values in definition of LBP threshold each pixel against its neighbours in a pre-defined neighbourhood. Assume that $g_p \geq g_c$ and let us consider a monotonic transformation f applied on g_p and g_c. Then from the definition of monotonic transformation it holds $f(g_p) \geq f(g_c)$, thus the value of $s(g_p - g_c)$ remains the same. Similarly, when $g_p \leq g_c$, then $f(g_p) \leq f(g_c)$. This holds for each $p = 0, \ldots, P-1$, therefore the value of $LBP_{P,R}$ remains the same. □

Rotational invariant local binary patterns are achieved via assigning a unique identifier to each rotation local binary pattern. Note that an exact rotation invariance is considered only rotations by an angle $2i\pi/P$, $i \in \mathbb{N}$.

Theorem 2 (Rotation invariant of local binary pattern)**.** *Given a local binary pattern of an image I. We define an operator $LBP^{ri}_{P,R}$ as*

$$LBP^{ri}_{P,R} = \min\{ROR(LBP_{P,R}, i) | i = 0, \ldots, P-1\}$$

where $ROR(x, i)$ performs a circular bit-wise right shift on the P-bit number x i times. Then $LBP^{ri}_{P,R}$ is a rotation invariant considering rotations by angle $2i\pi/P$, $i \in \mathbb{N}$.

Proof. The $LBP^{ri}_{P,R}$ is achieved by circularly rotating each bit $LBP_{P,R}$ until the minimum value is attained. The rotation by an angle $2i\pi/P$ produces local binary pattern values, which are shifted in a binary representation. The circular shift of binary numbers forms equivalence, in which each class representative is uniquely defined by a minimum. □

It was demonstrated that a special kind of LBPs called the "uniform" patterns achieve improved rotation invariance and provide better discrimination compared with the original rotation invariant LBP (Pietikäinen et al. [2000]).

Definition 27 (Uniformity measure of a binary number)**.** *Let g be an integer and $b(g)$ its binary representation. The uniformity measure of the integer g denoted as $U(g)$ is defined as the number of bitwise 0/1 changes in a $b(g)$,*

Definition 28 (Rotation invariant uniform local binary patterns)**.** *Given a local binary pattern of an image I. The rotation invariant uniform local binary pattern is defined as*

$$LBP^{riu2}_{P,R} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & U(b(LBP_{P,R})) \leq 2 \\ P+1, & \textit{otherwise} \end{cases} \tag{3.1}$$

where $b(x)$ is a binary representation of the number $LBP_{P,R}$.

For the image retrieval purpose the local binary patterns were applied in the work of Vatamanu et al. [2013] combined with colour coherence vector and there were modified by Martolia et al. [2020]. We utilised the rotation-invariant uniform local binary pattern feature and computed a histogram of this transformed LBP image as a feature in our experiments.

3.4.3 Gabor filter

Experiments on the mammalian vision system support the spatial-frequency analysis that maximises the simultaneous localisation of energy in spatial and frequency domains. Moreover, image analysis using Gabor filters is similar to perception in the human visual system (Rivero-Moreno and Bres [2003]), and that is why they represent a widely utilised technique to generate an efficient image descriptor.

These filters prove to be useful for texture analysis, image classification and also for image retrieval (Zhang et al. [2000]). Presented methods provide meaningful image descriptors by computing statistical features from a filtered image.

Definition 29 (Gabor function)**.** *Gabor function is a function over $\mathbb{C}$ with parameters $\lambda, \theta, \psi, \sigma, \gamma$ defined as:*

$$g((x,y); \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \lambda^2 y'^2}{2\sigma^2}\right) \exp\left(i(2\pi\frac{x'}{\lambda} + \psi)\right),$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\cos\theta + y\sin\theta$.

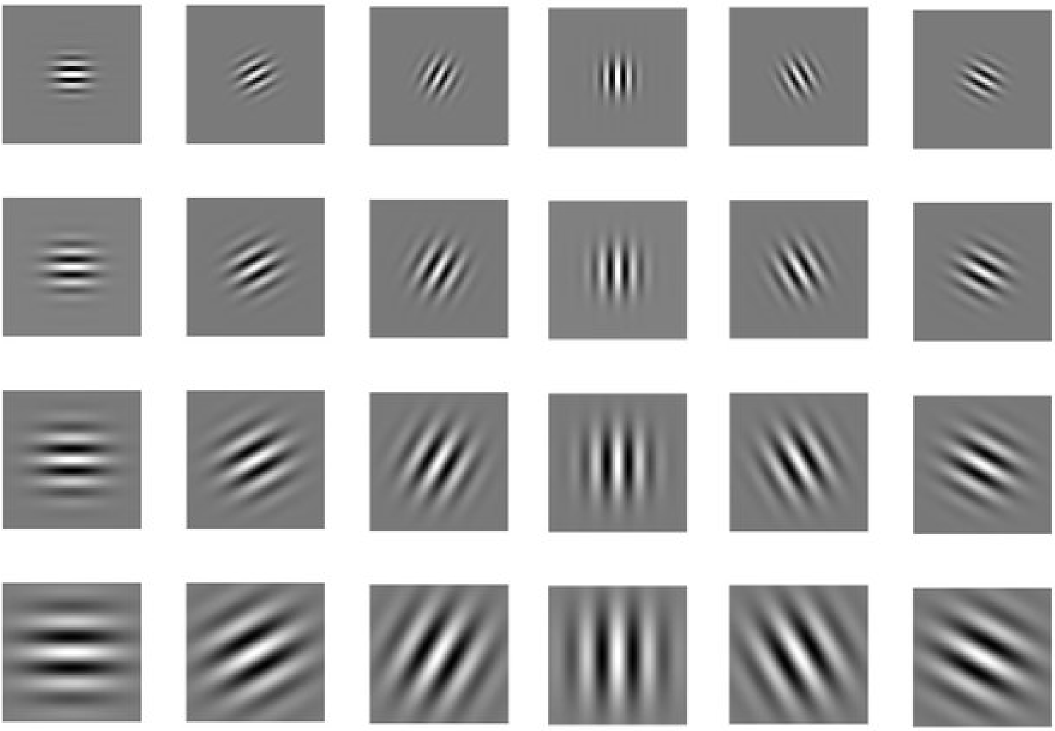


Figure 3.6: Gabor kernels using 4 scales and 6 orientations (Nurhadiyatna et al. [2015]).

The filter has a real, and an imaginary component representing orthogonal directions and only the real part is considered.

Definition 30 (Real part of a Gabor function)**.** *The real part of a Gabor function with parameters $\lambda, \theta, \psi, \sigma, \gamma$ is defined as:*

$$g((x,y); \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \lambda^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right),$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\cos\theta + y\sin\theta$.

Although the Gabor filter is defined on the entire $2D$ plane, its fixed subset is applied as a convolution filter. In order to determine a kernel size, we use one of the most common heuristic measures known as the 3-sigma rule providing a 99 % confidence interval $[\mu - 3\sigma, \mu + 3\sigma]$. Therefore, the kernel size is calculated according to standard deviation σ as $k = 2\lfloor 3\sigma \rfloor + 1$.

After convolving an image with Gabor filter with different parameters, we get resulting filtered images $I'(k, \lambda, \theta, \psi, \sigma, \gamma)$, a magnitude of each of them is computed. The magnitude of a filtered image by Gabor function is defined as:

$$E_G(k, \lambda, \theta, \psi, \sigma, \gamma) = \sum_x \sum_y |I'((x, y); \lambda, \theta, \psi, \sigma, \gamma)|$$

The resulting Gabor statistical features are mean and standard deviation for all combinations of different chosen parameters. Namely, we changed the parameter σ (determining the kernel size) in the experiments.

Definition 31 (Gabor statistical feature vector)**.** *Let I be an image of size $h \times w$ and $G_I(k, \sigma)$ be image I convolved with Gabor filter with parameter k and fixed parameters λ, θ, ψ, γ, σ, where σ is computed by the 3-sigma rule. Gabor statistical features are defined as:*

$$m_k = \frac{E_G(k)}{hw}$$

$$s_k = \frac{\sqrt{\sum_x \sum_y (|G((x, y); k)| - m_k)^2}}{hw}$$

Let k_i, $i = 1, \ldots, K$ be kernel sizes of Gabor kernel. Gabor statistical feature vector is defined as:

$$G_I = (m_{k_1}, s_{k_1}, \ldots m_{k_K}, s_{k_K})$$

The problem with such features is that it does not provide a rotation invariance (Zhang et al. [2000]). It is solved by a simple circular shift on a feature map. Total energy for each orientation is calculated, and the orientation with the highest total energy is considered the dominant orientation. Afterwards, the elements are shifted to obtain the dominant direction in the first position. Therefore, this leads us to potential improvement.

In our experiments, we compare Gabor's statistical features in terms of standard distance metrics defined in Section 2.5.

3.5 Shape methods

By shape methods, we mean techniques that generate features related to the discrete derivative of an image by convolving an image with a specific kernel. The most straightforward approach is a kernel $(-1, 0, 1)$ representing a discrete derivative in direction of x-co-ordinate and $(-1, 0, 1)^T$ in the direction of y-co-ordinate of an image. A convolution kernel such as Sobel or Robinson compass mask incorporates information about the neighbouring pixels with smaller weights. Histogram of Oriented Gradients additionally computes directions and magnitudes of an image. It divides them in an appropriate ratio into direction histogram bins and concatenates computed features into one feature vector.

3.5.1 Shape-based image filters

There are many possibilities for convolving an image with a filter. The Sobel filter corresponds to discrete derivatives in two directions, whereas the Robinson compass mask computes image derivatives in 8-directions. Filters can be applied to each channel separately, or an image can be converted to greyscale to obtain only one channel.

Definition 32 (Sobel operator)**.** *Sobel operator are matrices G_x, G_y is defined as:*

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad G_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Definition 33 (Robinson compass mask)**.** *Robinson compass mask are matrices G_i, $i = 1, \dots, 8$ is defined as:*

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad \begin{pmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

A shape-based feature is obtained by convolving an input image with these kernels and histogram computation.

3.5.2 Histogram of oriented gradients

It has been reported that the Histogram of Oriented Gradients (HOG) performs well in detecting an object of fixed size. It provides a simple feature vector describing the orientation and magnitude of the image regions.

Dalal and Triggs [2005] proposed HOG to address object detection, originally designed for pedestrian detection. These features are calculated by taking orientation histograms of edge intensity in a local region. It is designed to extract information about the edge's magnitude and orientation.

In the pre-processing part, an image should be resized or cropped to the fixed height-width ratio and converted to greyscale. The image with fixed width and height is divided into fixed blocks, and each block is separated into four sub-blocks. The gradient vectors of each block are computed by convolution with the discrete derivative kernels $D_x = [-1, 0, 1]$ and $D_y = [-1, 0, 1]^T$.

Definition 34 (Magnitude and orientation of the gradient)**.** *Let I be an image and I_x, I_y its discrete derivatives computed by convolution of an image I with kernels $D_x = [-1, 0, 1]$ and $D_y = [-1, 0, 1]^T$ respectively.*

The magnitude of the gradient is given by

$$|G| = \sqrt{I_x^2 + I_y^2}$$

The orientation of the gradient is given by an angle θ defined by

$$\theta = \arctan\left(\frac{I_y}{I_x}\right)$$

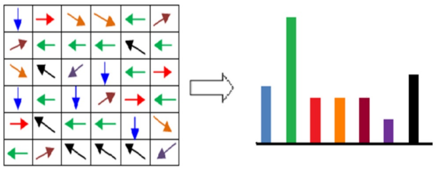


Figure 3.7: Histogram of oriented gradients computation of one subblock (Mokhtari et al. [2013]), after computing each subblock, concatenating histogram of whole block (4 subblocks), and then concatenating blocks into one feature vector.

Note that the magnitude and orientation are computed for each image pixel individually. An angle-based histogram is generated by taking a fixed number of bins and filling the resulting magnitudes into these bins proportionally, depending on the magnitude value. A feature vector is designed by concatenating oriented histograms of each block.

HOG was suggested in the field of content-based image retrieval (Halappa [2013], Halappa and Sudhamani [2015]) in order to detect objects. Alternatively, it was combined with a discrete wavelet transform (Vijendran and Kumar [2014]) by computing HOG of a transformed image.

Chapter 4

Deep Neural Networks

This chapter provides an overview of deep neural networks. At first, we discuss the main idea of its special kind employed on image classification tasks, followed by the mathematical background. We explain how these networks are used for other tasks, such as content-based image retrieval. Finally, we provide a summary of the existing neural network models.

A machine learning model is a mathematical model whose primary goal is to approximate a function f. The model parameters are trained to approximate a given function. Unlike optimization, where we take care of the available data, machine learning models strive to generalize well to reduce an error on the previously unseen data. Therefore, our data are usually split into train and test set. The machine learning model parameters are often gradually trained by optimizing given criteria called a loss function on a train set.

The notion of a Neural Network stands for a special kind of machine learning model inspired by the neurons in the human brain. A most common example of a Neural Network is visualized in Figure 4.1, widely known as a multilayer perceptron, a fully connected class of a feedforward neural network. It can be visualized as a graph $(\mathbf{V}, \mathbf{E})$ consisting of a set of vertices called neurons and a set of edges called weights. The neurons with the same horizontal coordinate in Figure 4.1 are called layers.

Each neuron's output is computed from connected neurons in the previous layer. It is calculated in terms of weights and biases. Finally, the neuron's output is evaluated with its corresponding activation function. The neural network has one input layer, several hidden layers and one output layer connecting neurons in consecutive layers in a fully-connected manner. The model is able to learn weights and biases adaptively by optimizing a loss function. When a model encounters the wrong classification, it updates the parameters to minimize a given loss function. This algorithm for training neural networks is called backpropagation.

Training a general Neural Network is done by minimizing an overall loss function by gradient-based algorithms, such as stochastic gradient descent or adaptive algorithms such as Adam, RMSProp etc. These gradient-based algorithms refer to the optimizer of a Neural Network. In order to minimize a loss function, the step size should be considered, called learning rate in the context of Neural Networks. The number of epochs corresponds to the number of forward and backward passes through the neural network.

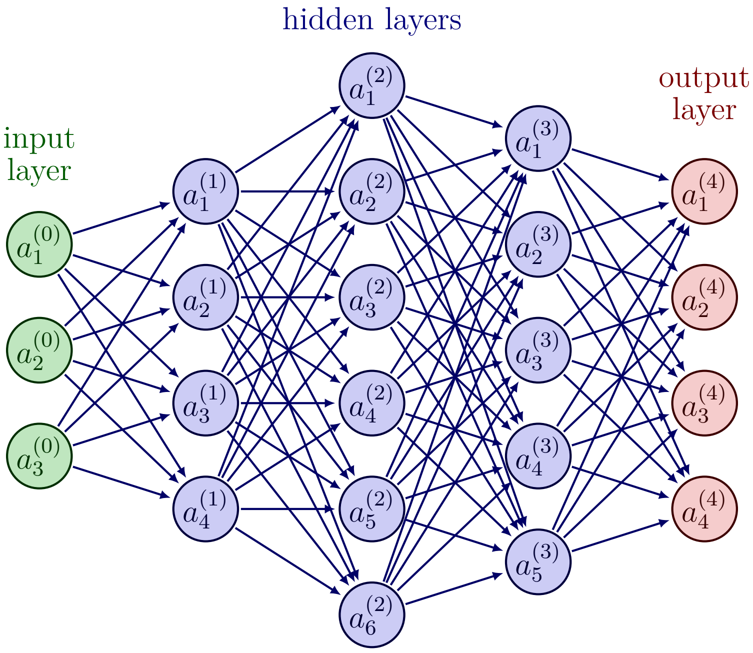


Figure 4.1: Multilayer perceptron (Neutelings [2022]).

More complicated models were invented, providing better accuracy in many tasks due to the increasing computational power. Also, scientists came up with better backpropagation algorithms. Deep Neural Network represents a special kind of machine learning model created by stacking many layers on top of each other. Because of its tremendous success in practical applications, deep learning models have replaced traditional machine learning approaches in many tasks (Goodfellow et al. [2016]).

When building a neural network, it is built from the bottom to the top. Therefore, by the top layer, we mean the last one. Since fully-connected layers in neural networks bring all the data information together, it incorporates a vast number of parameters. Therefore, the computation can be slow due to a large number of hidden layers. In some tasks, not all neurons are connected to each other, and that is why other kinds of layers have been invented, for example, Convolutional Neural Networks (CNNs).

As we already mentioned, the goal of Neural Networks is to achieve a suitable generalization property. Larger datasets help to train a model that generalizes better and hence reduces an error on previously unseen data, a test set. However, when working with a smaller dataset, the simpler model can be prefered to avoid overfitting. One possibility to deal with smaller image datasets is called dataset augmentation. Dataset augmentation stands for creating new data that resembles the original samples, for instance, by cropping an image, horizontal flip, or rotating it. In comparison with handcrafted features where we strive for invariance to these transformations, dataset augmentation solves invariance automatically. A choice of dataset augmentation depends on a specific application.

4.1 Convolutional Neural Networks

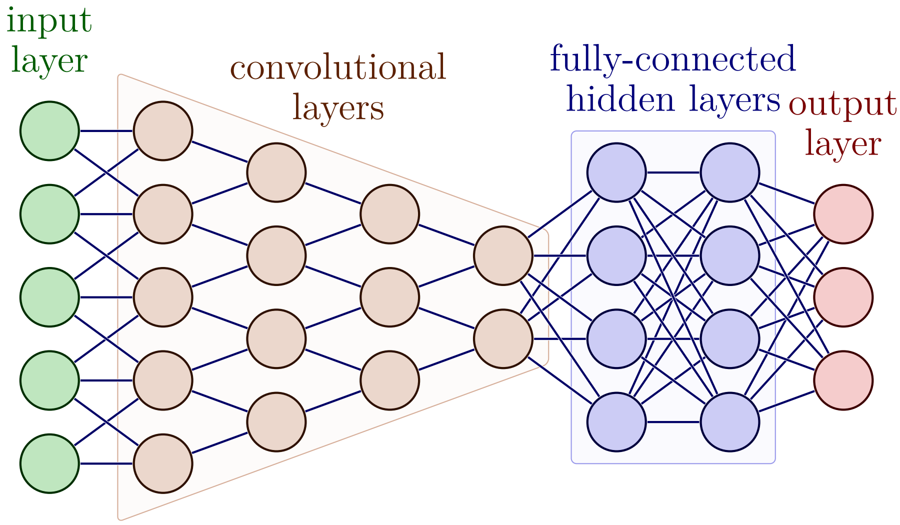


Figure 4.2: Deep convolutional neural network (Neutelings [2022]).

Convolutional Neural Networks (CNNs) (LeCun et al. [1989]) are a kind of neural network for processing data that has a spatial or temporal structure. Typical examples cover image data, which can be considered as a two-dimensional grid of pixels. It was demonstrated that CNN-based approaches achieve better results in image tasks. CNNs employ a linear operation called convolution, which we already used in handcrafted features, such as the Gabor filter or Sobel operator. The main difference between using predefined convolution filters and convolutional neural networks is that in these networks, a kernel plays the role of a trainable parameter to extract the crucial features describing an image accurately.

When solving an image-oriented task, the main focus is intended on local interactions because the individual image pixels nearby are more strongly correlated than the further ones. The parameters sharing provides shift-invariance in the spatial data. Thus, we get fewer parameters and features dependent on the spatial distribution of the data. Another advantage of convolution is that it provides a means for working with inputs of variable sizes.

Since the success of Krizhevsky et al. [2012] on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC, Russakovsky et al. [2014]), the current intensity of interest in deep learning and the potential of CNNs has rapidly increased. The network was trained on a huge annotated dataset ImageNet containing more than 14 million images with over 20 thousand categories. When networks' parameters were already optimized on ImageNet dataset, we say that the network is pre-trained. Therefore, when solving another image classification task, these models can be applied for inference.

Moreover, the same pre-trained network can solve an image retrieval task. The generated layers of such networks contain feature maps generated by the convolutional block, which represent high-level descriptors of an image. The bottom, i.e. the first layers, describes low-level descriptors similar to those gained by handcrafted features. When we go deeper, the layers represent more and more high-level features that appear and are widely used to solve related tasks.

4.2 Mathematical background

In this section, we provide a mathematical background of neural networks by first stating the Universal approximation theorem, followed by explaining convolution, pooling, batch-normalization and activation functions deployed in CNNs. Since a maximum likelihood principle is used to design a loss function, we derive the cross-entropy loss function.

4.2.1 Universal Approximation Theorem

Universal approximation theorems imply that neural networks can approximate any continuous function. It states that approximating function parameters is possible. However, it does not provide construction. We state one variant from 1989, and its proof can be found in the work of Hornik et al. [1989].

Theorem 3 (Universal Approximation Theorem)**.** *Let $\phi : \mathbb{R} \to \mathbb{R}$ be a nonconstant, bounded and nondecreasing continuous function. For any $\epsilon > 0$ and any continuous function $f : [0,1]^D \to \mathbb{R}$, there exists $N \in \mathbb{N}, v \in \mathbb{R}^N$, $b \in \mathbb{R}^N$, $W \in \mathbb{R}^{N \times D}$, such that if we denote $F(x) = v^T\phi(Wx + b)$, where ϕ is applied elementwise, then for all $x \in [0,1]^D$:*

$$|F(x) - f(x)| < \epsilon.$$

4.2.2 Convolution and cross-correlation

In order to define a generalized milestone of 2D convolution, let us recap the Definition 13. A convolution is a linear operation employing a kernel K representing the weights of pixels in a spatial configuration calculated over the whole image.

There are another three new parameters covered in the generalized definition of convolution: the number of channels C, the stride S and the number of output channels O. The result of the convolution is summer over multiple channels C.

The stride S represents the number of pixels shifts of a kernel K over the input image matrix. In other words, the output of a convolution is computed for each S-th pixel in every dimension. For instance, when the stride S is two, it decreases the size of the output twice in each spatial dimension. The number of output channels O adds another dimension to the resulting feature map meaning that convolution is applied multiple times with different kernels.

Definition 35 (2D Discrete Cross-correlation with C channels, O output channels and stride S)**.** *Let I be an input image of size $M \times N$ with C channels, the convolution layer is then parametrized by a kernel K of total size $W \times H \times C \times F$ and is defined as:*

$$(K \star I)_{i,j,o} = \sum_m \sum_n \sum_{c=1}^{C} I_{i \cdot S+m, j \cdot S+n, c} K_{m,n,c,o}$$

Local interactions are performed in the image spacial dimensions, width and height.

4.2.3 Pooling

Pooling is an operation that reduces the output dimension by computing a function, such as maximum or average, on a bunch of neighbouring neurons. Pooling over spatial regions produces invariance to translations and even to scaling and slight rotations. It summarizes the responses over a given neighbourhood to capture high-level features efficiently (Goodfellow et al. [2016]).

The pooling layer is given by a set size of a pooling window and has no learnable parameters. Suppose we are given a pooling window A. The max-pooling operation is computed as $max_{a \in A} a$ and the average-pooling as $avg_{a \in A} a$. Pooling represents a helpful tool for handling inputs of varying sizes. It can be designed so that the last classification layer receives the same feature size regardless of the input size.

The traditional structure of CNN has actually multiple blocks of convolution, activation and pooling. It contains a fully-connected layer directly before the top classification layer. A novel approach called global pooling was proposed in the work of Lin et al. [2013] to replace traditional fully-connected layers in CNNs in order to avoid overfitting. Instead of adding fully-connected layers on top of the feature maps, the average of each feature map is computed. It behaves more naturally and enforces correspondence between feature maps and categories. Since there is no parameter to optimize, it avoids overfitting in this layer, and it is more robust to spatial translations of the input.

4.2.4 Activation function

The choice of activation function in the hidden layer is important to control how well the network model learns. An activation function used in CNNs is usually a Rectified Linear Unit function (ReLU). It is a non-linear function which decides whether a neuron should be activated or not depending on a sign of an output. For more activation functions we refer to Goodfellow et al. [2016].

Definition 36 (ReLU)**.** $ReLU(x) = \max(0, x)$

4.2.5 Batch-normalization

A minibatch refers to equally sized subsets of the dataset over which the gradient is calculated and weights updated. Batch normalization represents a kind of model reparametrization that normalizes a minibatch output. Batch normalization is a layer that takes the hidden layer's outputs and normalizes them before passing them on as the input of the next hidden layer. More precisely, let A be a minibatch of neuron's output is transformed as

$$A' = \frac{A - \mu}{\sigma},$$

where μ is a vector of the neuron mean, and σ is a vector of the neuron's standard deviation.

4.3 Transfer learning

When a computer vision problem is solved for a different dataset, it may be advantageous to utilize stored knowledge for a given dataset. Transfer learning is a process that applies a model to solve different tasks, reducing the cost of the training.

We want to use a model trained on a different image dataset in practice. Therefore, the network is initialized with pre-trained parameters instead of random initialization. A pre-trained model trained on a large image dataset, such as ImageNet (Russakovsky et al. [2014]). ImageNet is a large database widely used in recognition software research. Since ImageNet networks perform classification tasks, the deep features can be obtained by dropping the last classification layer, and they are frequently used as general feature extractors.

Fine-tuning of the network is meant as initialization by a pre-trained classification network and then training it further. This kind of retraining may improve performance further by gaining the adaptation ability of the current dataset. It can be understood as a particular case of transfer learning. During this process, a lower learning rate is necessary because the original model probably finished training with a very small learning rate (Straka [2022]). In the context of image retrieval, the fine-tuning approach was proposed by Babenko et al. [2014] which enhanced the retrieval accuracy.

4.4 Pre-trained models

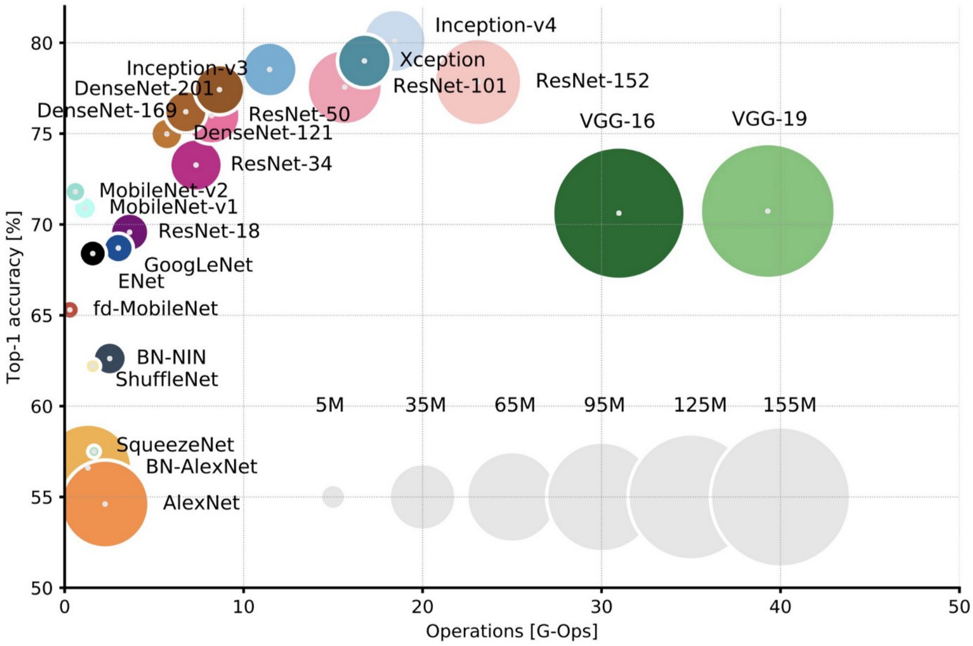


Figure 4.3: Comparison of pre-trained models (Canziani et al. [2016]).

Many pre-trained models have been used for competitions in ImageNet classification. By removing the classification layer and the fully-connected layers, it turned out that these feature maps obtained by pre-trained models can solve different tasks. These pre-trained convolution layers provide an excellent descriptor of an image.

4.4.1 AlexNet (2012)

AlexNet is a convolutional neural network architecture designed by Krizhevsky et al. [2012]. It won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC, Russakovsky et al. [2014]), a competition for image classification on the ImageNet dataset held in 2012. The network's input is an image of a fixed size 224 × 224 with 3 channels. The data were augmented using translations and horizontal reflections of random 224 × 224 patches from 256 × 256 images. It addressed overfitting by data augmentation and dropout.

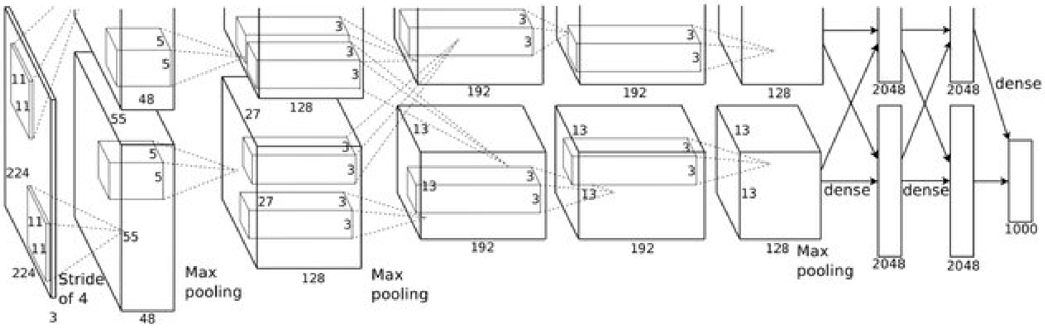


Figure 4.4: AlexNet architecture from the original paper of Krizhevsky et al. [2012], explicitly showing the responsibilities between two GPUs that communicate only at certain layers. The image is from the original paper.

Its architecture is designed in repeating layers in the following manner: convolution operation rectified linear unit activation (ReLU) and pooling with increasing the number of channels after it. At the top, there are several FC layers and then the top classification layer.

4.4.2 VGG (2014)

VGG neural network (Simonyan and Zisserman [2014]) is based on AlexNet architecture. It takes in 224 × 224 pixel RGB images. In comparison with AlexNet, it incorporates small kernels of size 3 × 3 and stride 1 and 1 × 1 convolution filters followed by a ReLU activation. The convolution stride is fixed to 1 pixel so that the spatial resolution is preserved after convolution. The last three layers are fully connected, and the last layer contains 1000 neurons.

4.4.3 ResNet (2015)

Since it was demonstrated that neural networks have a low ability to copy information, an innovative approach employing residual connections appeared. These connections are designed in a way that a feature map at a certain level is copied and added to another layer. They cannot be applied directly when the number of channels increases. The ResNet architecture was designed similarly to previous networks, using mainly 3×3 convolutions. Additionally, batch normalization was applied.

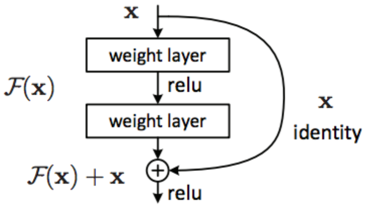

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4.5: ResNet block (He et al. [2015]).

In the case of ResNet or more advanced architectures, the last layer is a global pooling layer.

4.4.4 MobileNetV2 (2018)

Sandler et al. [2018] introduced MobileNetV2 with inverted residual structure. It is designed for mobile phones to reduce computational cost and space complexity. It incorporates bottlenecks (layers with fewer channels) connected by residual connections. It employs depthwise separable convolution acting on each channel separately and pointwise convolution acting on each position independently.

4.4.5 EfficientNet (2019)

One of the most efficient architectures for image recognition is EfficientNet, designed in the work of Tan and Le [2019] which was created to optimize both accuracy and computation complexity. The baseline network is denoted as EfficientNet-B0, and it is based on the inverted bottleneck residual blocks applied in MobileNetV2. For more details of its architecture, we refer to Tan and Le [2019].

An improved version of EffiientNet, called EfficientNetV2, was published in April 2021 by Tan and Le [2021]. The architecture is slightly different. It prefers smaller 3×3 kernel sizes, but it adds more layers to compensate for the reduced receptive field resulting from the smaller kernel size. Moreover, due to the slow training on large images, the maximum image size is limited to 480.

Chapter 5

Dimensionality reduction

Nowadays, most datasets have a vast number of variables. There is a high number of dimensions along which the data is distributed. These dimensions are given by the number of image feature components in our case. Visual data exploration can then become challenging. Sometimes it is even impossible to explore high-dimensional data manually. This problem leads us to understand how to visualise high-dimensional datasets. The dimensionality reduction can help us understand the data using fewer variables, retaining meaningful properties of the original data.

Principal component analysis (Hotelling [1933]) represents a technique that reduces the number of dimensions. It is a linear transformation preserving the maximum of the data variance. However, it assumes a linear relationship between features, which is not satisfied in most cases. When dealing with non-linear manifold structures, the linear algorithms do not yield satisfying results. A manifold can be intuitively understood as a geometric structure. For a proper definition, we refer to Hatcher [2000]. High-dimensional data lie on several different but related, low-dimensional manifolds, such as images of objects from multiple classes seen from multiple viewpoints. Therefore researchers have tried various approaches to visualise the data while retaining the local structure. One of the techniques is SNE (Hinton and Roweis [2002]); another one is t-SNE (van der Maaten and Hinton [2008]), which is based on SNE and offers better visualisation results. Since humans can imagine the data of two or three dimensions, dimensionality reduction methods convert the high-dimensional data set into two or three-dimensional data displayed in a scatterplot.

5.1 Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised machine learning algorithm typically used for dimensionality reduction. It can also be used for denoising. The idea is to reduce the dimensionality while retaining the variation present in the dataset as much as possible. It employs a linear projection from the original n-dimensional space to k-dimensional space, where $k < n$.

The first step requires to centre data to the origin. It means that data on all the dimensions are subtracted from their means. Next, the first principal component is computed to explain the most significant amount of variance in the original data. The second component is orthogonal to the first, and it explains the greatest amount of variance after the first principal component.

Generally, the n-dimensional data are linearly projected into k dimensions by maximising the data variance. Equivalently, it can be understood as minimising the sum of the projection distances of the data. PCA can be calculated from the centred data matrix's Singular Value Decomposition (SVD). For a detailed description, we refer to the original paper of Hotelling [1933].

5.2 t-Distributed Stochastic Neighbor Embedding

For high-dimensional data that lies on a non-linear low-dimensional manifold, keeping the low-dimensional representations of very similar data close together is necessary. This is typically not possible with a linear mapping. It is important to retain both the local and the global structure of the data. Such a valuable technique for visualising high-dimensional data is called t-Distributed Stochastic Neighbor Embedding (t-SNE). The t-SNE algorithm was designed by Hinton and Roweis [2002]. It represents a variation of Stochastic Neighbor Embedding and produces significantly better visualisation results.

The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional and low dimensional spaces. A good measure of the similarity of two probability distributions is the Kullback-Leibler divergence, known as KL divergence. Note that the KL divergence is not symmetric.

Definition 37 (KL divergence)**.** *For discrete probability distributions P and Q defined on the same probability space X, the KL-divergence of distributions P and Q is defined as:*

$$D_{KL}(P\|Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}.$$

t-SNE minimizes the KL divergence between a joint probability distribution P in the high-dimensional space and a joint probability distribution Q in the low-dimensional space. It employs computing pairwise similarities of the data-points in both high-dimensional and low-dimensional space. Although SNE used Gaussian distribution to compute the pairwise similarities of the data-points, t-SNE employs a t-Student distribution with one degree of freedom. The Student t-distribution has heavier tails than the normal distribution allowing better modelling of far apart distances. In order to optimise this distribution, t-SNE uses KL divergence between the probabilities p and q and the gradient descent algorithm is employed. For a detailed description of an algorithm, we refer to the work of van der Maaten and Hinton [2008].

Since t-SNE suffers from low computational speed, the authors of the original paper van der Maaten and Hinton [2008] recommended starting by using PCA to reduce the dimensionality of the data and employing t-SNE afterwards. We used this technique to visualise neural codes of images before and after fine-tuning in Chapter 6.

Chapter 6

Experiments

This chapter provides an overview of our experiments executed on image datasets. We implemented most of the presented feature extraction algorithms in Chapters 3 and 4. Afterwards, we tested and evaluated them with various parameters on datasets with annotated classes. Regarding the evaluation, we calculate the mean average precision (mAP) and the precision at K ($P@K$) (see Section 2.6). We compare the results of implemented techniques on the individual dataset classes. The primitive methods describing the properties of colour, texture, and shape are usually combined together to achieve finer results. However, there exists an immense number of combinations generating a new feature combining colour, texture, and shape properties. Thus, we rather compare them individually. Furthermore, neural network approaches are able to learn more significant features from images and outperform any combination of handcrafted features. We use the pre-trained networks described in Section 4.4 and fine-tune them in order to improve the retrieval effectiveness. Afterwards, we examine the effect of centring and normalisation for CNN-based methods. Additionally, we show the t-SNE visualisation described in Chapter 5 with high-level image descriptors. It employs high-level image descriptors obtained from feature maps of neural networks. t-SNE visualisation helps to understand high-dimensional feature spaces by approximating the coordinates to two-dimensional coordinates. The main goal of this work is image retrieval from the dataset, which is collected from posters without annotation. We will show a web application described in Chapter 7.

6.1 Datasets description

The experiments were performed on the following four datasets: the Wang image dataset (Wang et al. [2001]), the Patterns dataset (Cimpoi et al. [2014]), the GPR1200 dataset (Schall et al. [2021]) and the Posters dataset (Wienbibliothek im Rathaus). The first three datasets are annotated and used for evaluation. The Wang image dataset consists of 1000 images, Patterns image dataset has 1200 images (created as a subset of Describable Textures Dataset), and GPR1200 image dataset contains 12000 images.

However, the main purpose is to develop an image retrieval system for Vienna City Library. The original dataset comprises 300 thousand posters gathered in the Vienna City Library (Wienbibliothek im Rathaus). The Vienna City Library provided a sample of 5050 images to perform content-based image retrieval.

6.2 Evaluation

We evaluated the experiments on a test set containing 20 % of the dataset. This set is evenly distributed into all categories in terms of mean average precision and precision at ten. We utilised precision at ten in order to demonstrate how many images from the same category appeared in the first ten positions on average. However, the use of mean average precision is more frequent in retrieval systems. It employs the ordering of all relevant retrieved images. Since the model performance depends on a chosen distance metric (manhattan, euclidian or cosine distance), the results are compared for each metric separately. We test the retrieval performance of each method across various parameters, and then we recommend parameters giving the best results.

Moreover, we propose a feature vector normalization for CNN methods based on the work of Zouhar et al. [2022] developed in the context of natural language processing retrieval. The centering and normalization transformation is given by $x' = \frac{x-\bar{x}}{\|x-\bar{x}\|}$, where $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ for n feature vectors x_i, $i = 1, \ldots, n$. This preprocessing ensures both Euclidean distance and cosine distance giving the same retrieval results due to the same ordering of retrieved images, thus the same mean average precision.

Lemma 1. *Let $x, y \in \mathbb{R}$ normalized vectors (using L_2 norm). Then it holds:*

$$argmin_k \|x - y\| = argmax_k (x \cdot y).$$

Proof.

$$\begin{aligned} argmin_k \|x - y\| &= argmax_k - \|x - y\| \\ &= argmax_k - (x \cdot x)^2 - (y \cdot y)^2 + 2(x \cdot y)^2 \\ &= argmax_k (-1 - 1 + 2x \cdot y) \\ &= argmax_k (x \cdot y) \end{aligned}$$

□

6.3 Parameters

In this section, we focus on parameters of implemented feature extraction algorithms summarized in Table 6.1 that we run in our experiments.

Each colour method tested both RGB and HSV colour space. The colour histogram and CCV require the number of quantisation levels of each colour component. The colour grid employs a parameter of the number of grid blocks in x and y directions in which an average colour is computed. The spatial chromatic histogram is not implemented.

Texture methods require parameters such as pixel distance and angle parameters. GLCM computes co-occurrences of grey-level pixels based on the parameter of the distance and angle between two pixels. Each parameter combination is denoted by (d, a). An integer d represents a number of distances taking into account all integer pixel distances from one to d. Parameter a represents the number of evenly spaced angles. The resulting feature is concatenated into one feature vector. Results were evaluated for (d, a), such that $d = 1, 2, 3$ and $a = 4, 6, 8, 10$.

LBP method considers two parameters: the radius r and the number of points evenly spaced points p in the circular neighbourhood. We implemented the uniform patterns variant, ensuring the rotation invariance. A pair of parameters is denoted as (p, r). It is tested on parameters $(8, 1)$, $(8, 2)$, $(16, 2)$, $(16, 3)$ and $(24, 3)$. Gabor filter is applied on an image with multiple θ where θ determines an angle of the filter. A number of evenly spaced angles denoted by a is tested as a parameter. The kernel size of the Gabor filter is computed by 3-sigma rule. Therefore it is determined by the σ parameter of the filter. Tested parameters were (σ, a) for $\sigma = 1, 3, 10$ and $a = 4, 6, 8, 10$.

Parameters in implemented shape methods incorporate the number of histogram bins determining the length of the feature vector. Since the HOG feature requires a fixed size of an image, the image is downsampled to a fixed size 100×100, and it requires a parameter of the cell size c and the number of the angle bins b. Determining the HOG feature length is complex, and we use the cell size eight and eight angle bins. The resulting HOG feature is of length 3872.

Neural network approaches utilised pre-trained models on the ImageNet for the classification task. In order to solve the retrieval task, transfer learning is applied. The last classification layer is dropped to obtain a high-level feature vector describing image properties. The performance of different kinds of architectures is evaluated and compared. The networks chosen for a comparison are AlexNet, VGG16, MobileNet-v2, ResNet-152 and EfficientNet-b3. The pre-trained networks on ImageNet dataset work well on Wang dataset. However, their performance on Patterns dataset is slightly worse since it contains images different from typical images from ImageNet dataset. Therefore, we fine-tune these networks in order to improve the retrieval performance further. We tested the different optimiser algorithms and learning rate schedules. We monitor loss curves on the train and a validation set. Afterwards, we describe details of the implementation and training of neural networks.

	Method	Parameter	Feature length
Colour	Histogram	(q_1, q_2, q_3)	$q_1 q_2 q_3$
Colour	Grid	(x, y)	$3xy$
Colour	CCV	(q_1, q_2, q_3)	$q_1 q_2 q_3$
Texture	GLCM	(d, a)	$5ad$
Texture	LBP	(r, p)	$p + 2$
Texture	Gabor	(s, a)	$2a$
Shape	Sobel	q	$2q$
Shape	Robinson	q	$8q$
Shape	HOG	$(8, 8)$	3872
CNN	AlexNet	-	9216
CNN	VGG	16	25088
CNN	MobileNet	v2	62720
CNN	ResNet	152	2048
CNN	EfficientNet	b3	1536

Table 6.1: The summary of used parameters and feature length of each implemented method. Note that Gabor, LBP and Sobel feature are usually very small when considering the tested parameters.

6.4 Retrieval results

We present the retrieval results of each implemented technique and compare the evaluation metrics for each class. Pre-trained networks are usually not sufficient for the more challenging datasets. Therefore, we fine-tune the networks in order to improve the retrieval performance.

6.4.1 Wang image dataset

The Wang image dataset (Wang et al. [2001]) (also known as Corel-100 image dataset as a subset of Corel-10000 image dataset) consists of 1000 images in JPEG format. Each image size is 256 × 384 or 384 × 256. Dataset is grouped into 10 categories: Africans, beaches, monuments, buses, dinosaurs, elephants, flowers, horses, mountains and food. The dataset was extensively utilised to test the image retrieval effectiveness because the small dataset size and the availability of class information allow running performance evaluation. A sample of Wang's image dataset is shown in Figure 6.1.

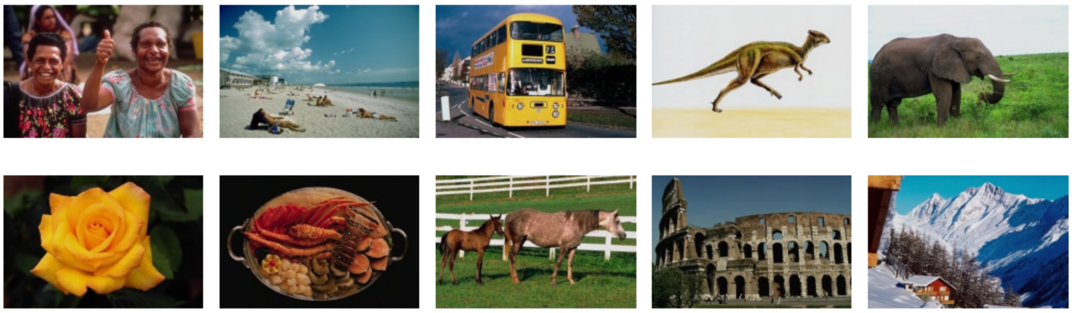


Figure 6.1: Examples of each category from Wang's image dataset (Wang et al. [2001]).

We run implemented methods on several parameters and metrics. The ones giving the best results in terms of precision at ten are summarised in Table 6.2. The colour-based approaches performed better than the texture and shape-based. Although the colour coherence vector was proposed as a generalisation of a colour histogram, its existing implementation does not outperform a colour histogram on the Wang dataset. Processing a CCV feature is computationally slow. Therefore, an image is down-sampled before the feature extraction part, which deteriorates the results. Moreover, CCV requires a threshold parameter τ that is set as one-hundredth of the number of pixels of an input image. The Gabor filter performed best from the texture methods and Robinson from the shape methods. Except for the Gabor filter, all handcrafted features performed better using the manhattan distance. For the methods using colour histogram, intersection distance results were similar to the manhattan distance (see Table 6.2). Thus we compare only the manhattan distance (L_1 norm of the difference) and cosine distance. All CNN-based approaches with cosine similarity outperformed other metrics. Therefore, we recommend using the L_1 norm for handcrafted features and cosine similarity for CNN approaches.

We calculated the mean average precision for each method on each image category. Mean average precision represents an average precision average over a set of all test queries. However, mean average precision on a given image category is averaged over a set of test queries on that category. The resulting table is shown in Figure 6.2.

Histogram works well on all image categories except for beaches, mountains and monuments, where it does not depend on colour. The grid colour approach performs well in the dinosaurs, horses and flowers category because of the spatial distribution of the colour.

Wang	Method	Parameter	P@10 L1	P@10 C	mAP L1	mAP C
Colour	Histogram	(8, 8, 8) H	0.765	0.669	0.528	0.459
Colour	Grid	(6, 6) H	0.620	0.603	0.425	0.412
Colour	CCV	(8, 8, 8) R	0.694	0.618	0.454	0.401
Texture	GLCM	(3, 4)	0.498	0.485	0.286	0.276
Texture	LBP	(2, 16)	0.577	0.567	0.367	0.358
Texture	Gabor	(3, 8)	0.587	0.5945	0.364	0.424
Shape	Sobel	128	0.5475	0.559	0.356	0.356
Shape	Robinson	128	0.579	0.566	0.374	0.361
Shape	HOG	(8, 8)	0.564	0.412	0.331	0.222
CNN	AlexNet	-	0.697	0.901	0.455	0.740
CNN	VGG	16	0.532	0.935	0.346	0.812
CNN	MobileNet	v2	0.785	0.926	0.620	0.816
CNN	ResNet	152	0.967	0.971	0.846	0.903
CNN	EfficientNet	b3	0.941	0.973	0.795	0.909

Table 6.2: The comparison of methods' performance on the Wang dataset. We denote HSV colour space by H, RGB colour space by R. The retrieval results are evaluated on both manhattan distance (denoted as L1) and cosine distance (denoted as C). Note that cosine distance works in general better for CNN-based approaches, whereas manhattan distance on handcrafted features. Eucliden distance performance was surprisingly poor, therefore we do not show it.

Texture and shape methods do not consider colour information; they perform substantially worse than colour approaches on colourful images like the Wang dataset. However, the texture and shape methods outperformed colour methods in retrieving flowers. Produced features are usually combined with colour features. However, we compared these methods individually. Gabor filter performs the best compared with other texture methods on categories without the object specification, such as beaches, foods and monuments. Although LBPs consider a local circular neighbourhood of size 2, it is able to retrieve well on buses, dinosaurs and flowers. GLCMs care about all co-occurrences of pixels within a prescribed distance, and they performed significantly worse in the dinosaurs category than other methods.

Since shape methods retrieve images only based on edges, these methods performed surprisingly well. Robinson's method outperforms Sobel in categories of objects with edges in horizontal, vertical, and diagonal directions, such as buses, elephants, dinosaurs, horses, and monuments. Since HOG was designed for object detection of pedestrians, it gives the best results for objects of the same size. Thus, it outperforms both shape methods in categories of animals such as elephants, horses and dinosaurs. Interestingly, HOG does not perform well on flowers. Furthermore, HOG includes spatial information of gradients; it also behaves well on categories such as beaches and mountains.

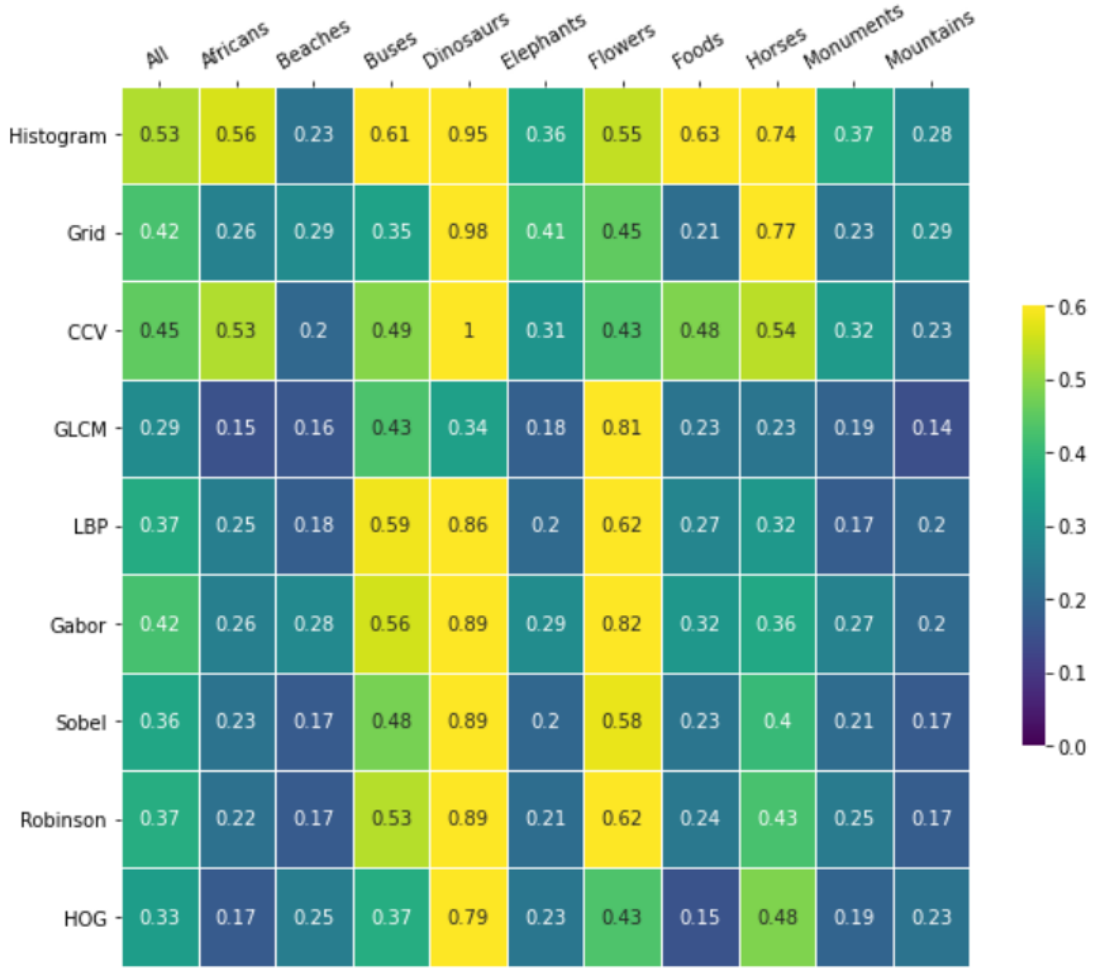


Figure 6.2: Comparison of mAP of handcrafted methods on different categories.

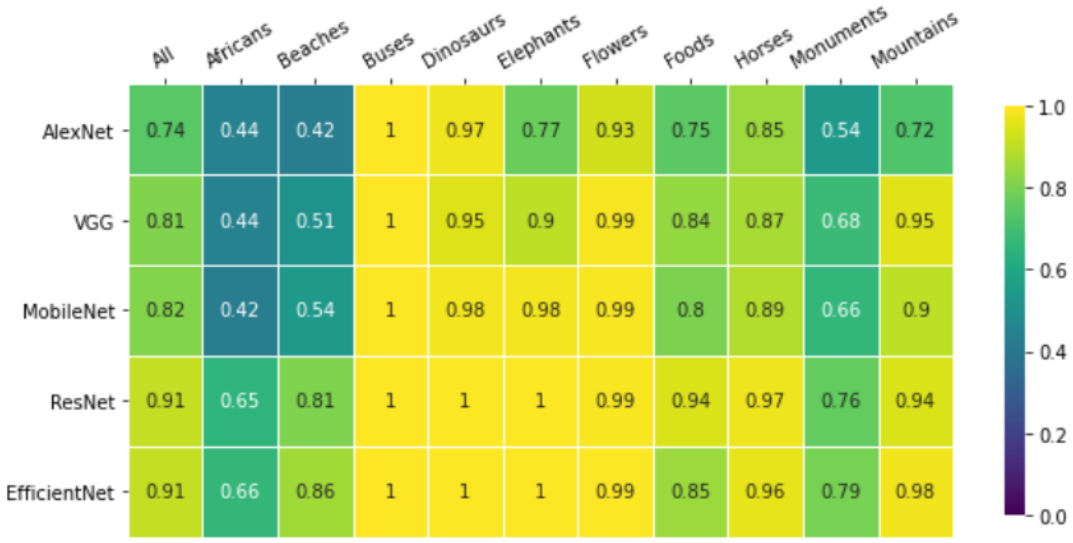


Figure 6.3: Comparison of mAP of CNN-based methods on different categories. Note the different colour scale than in Figure 6.2.

In comparison with handcrafted features, CNN approaches perform significantly better. The networks are ordered by the time of their invention. An architecture of EfficientNet-b3 gives the best results in all image categories except for foods in terms of mAP. The most challenging image categories are Africans and monuments. We consider the retrieval effectiveness sufficient. Thus we utilised a pre-trained EfficientNet-b3 model for image retrieval.

The t-SNE visualisation of Wang dataset is shown in Figure 6.4 to demonstrate that high-level image feature vectors of the images from the same class form clusters in the high-dimensional feature space.

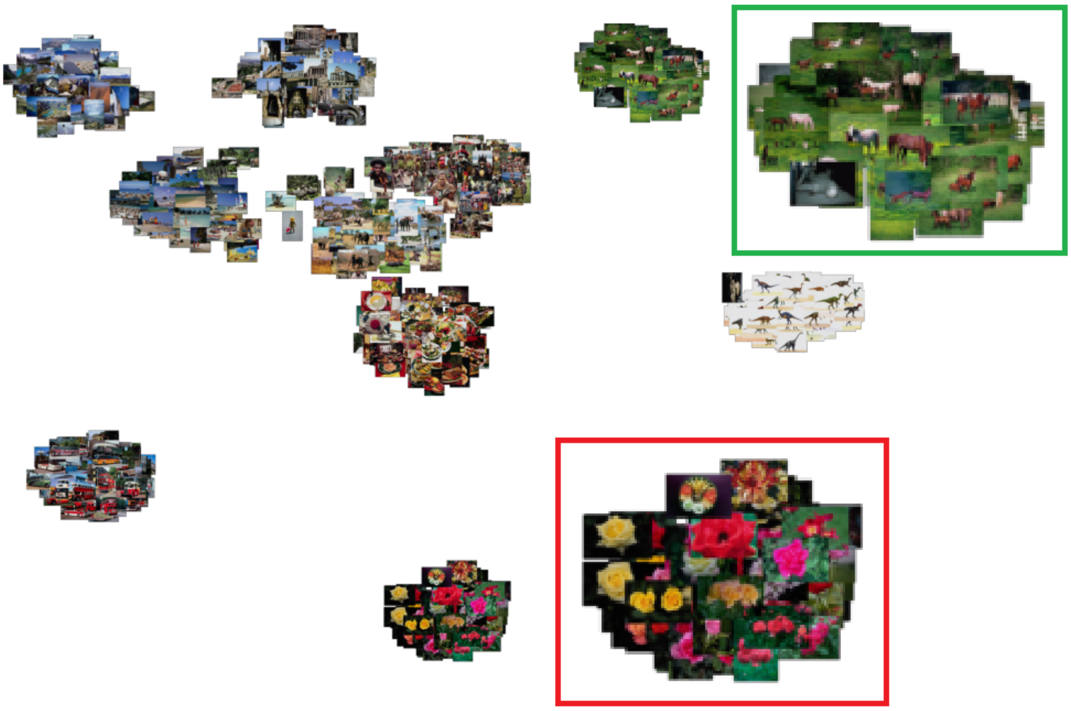


Figure 6.4: The visualization of the Wang dataset with t-SNE. We employed pre-trained EfficientNet-b3. Note the distant clusters of dinosaurs, flowers, buses and horses. The clusters of flowers and horses are zoomed.

6.4.2 Patterns dataset

The Patterns image dataset is formed by taking a subset of Describable Textures Dataset (Cimpoi et al. [2014]). It contains ten image categories: banded, chequered, cobwebbed, crystalline, dotted, fibrous, lacelike, potholed, stratified, and striped. We created this dataset to test the texture-based approaches and compare them with the state-of-the-art neural networks. Images in the same category have similar textures, but the colour usually differs. A sample of Patterns dataset is shown in Figure 6.5.

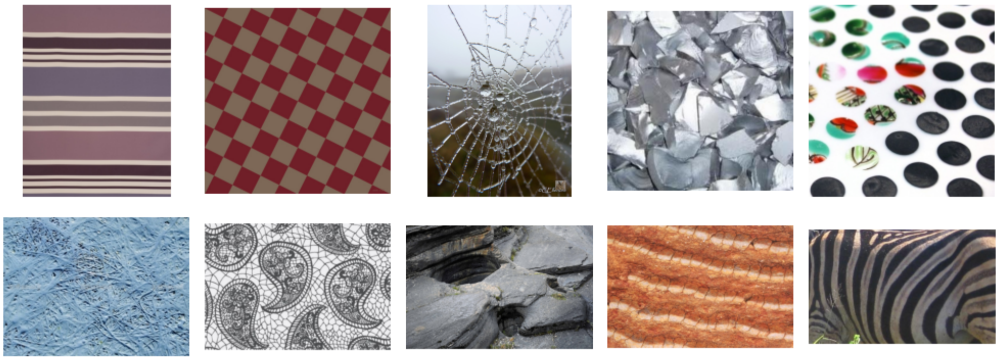


Figure 6.5: Example of each category of Patterns dataset, a subset of Describable Textures dataset (Cimpoi et al. [2014]). The colours over the images in each class differs in most cases, but the patterns are similar.

We tested implemented methods with several parameters and metrics. Evaluated results are summarized in Table 6.3. As expected, the texture and shape methods outperform colour methods on the Patterns dataset. Gabor filter achieved the best results from handcrafted features. HOG performs well in detecting objects. Therefore, it is deficient in this context.

The patterns dataset is less similar to ImageNet dataset than the Wang dataset. Since CNN-based approaches are trained on ImageNet, they achieve slightly worse results than the Wang dataset. Handcrafted features performed better using manhattan distance except for GLCM, Gabor, and Sobel, in which cosine similarity metric gives better results. We recommend using manhattan distance for handcrafted features and cosine similarity for CNN-based approaches.

Patterns	Method	Parameter	P@10 L1	P@10 C	mAP L1	mAP C
Colour	Histogram	(8, 8, 8) H	0.395	0.367	0.185	0.175
Colour	Grid	(4, 4) H	0.308	0.267	0.164	0.147
Colour	CCV	(8, 8, 8) R	0.417	0.395	0.201	0.196
Texture	GLCM	(3, 10)	0.397	0.479	0.180	0.251
Texture	LBP	(3, 24)	0.419	0.427	0.198	0.202
Texture	Gabor	(1, 4)	0.474	0.518	0.219	0.288
Shape	Sobel	8	0.489	0.495	0.265	0.273
Shape	Robinson	8	0.470	0.503	0.251	0.267
Shape	HOG	(8, 8)	0.360	0.351	0.229	0.204
CNN	AlexNet	-	0.618	0.851	0.303	0.594
CNN	VGG	16	0.612	0.915	0.299	0.707
CNN	MobileNet	v2	0.730	0.913	0.390	0.666
CNN	ResNet	152	0.917	0.935	0.683	0.736
CNN	EfficientNet	b3	0.923	0.939	0.655	0.688

Table 6.3: The comparison of methods' performance on the Patterns dataset. Note that colour methods are deficient. Texture methods performed on Patterns dataset employing a higher number of angles outperformed those using the small-angle parameter, except for Gabor filter. Not that Gabor filter outperformed all other handcrafted features with the feature length of eight.

All neural network approaches using the cosine similarity metric give the better results than using any other metric. We examined the effect of the centring and normalisation—the pre-processing slightly improved results by about one per cent for ResNet and EfficientNet and four per cent for MobileNet on Wang image dataset (see Figure 6.6). Similar results were obtained on Patterns dataset.

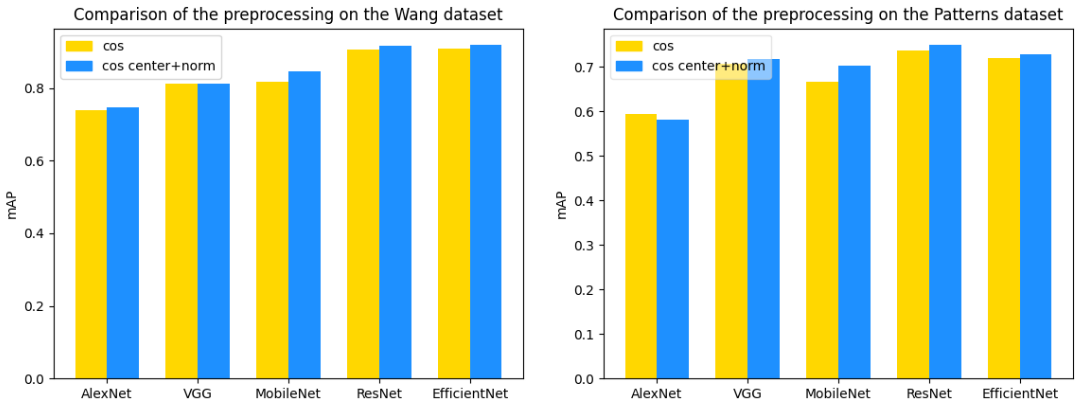
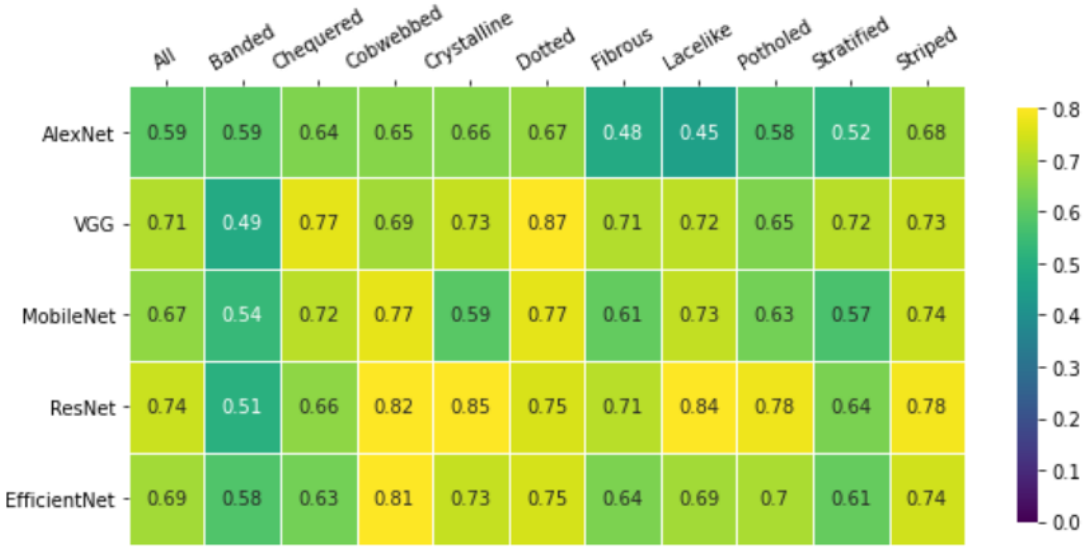


Figure 6.6: Comparison of the features pre-processing based on centring and normalisation on the Wang and the Patterns dataset. Mean average precision is evaluated for pre-trained networks. The results are compared in terms of cosine distance with and without described feature pre-processing. Note the different y-axes.

Since Patterns dataset is more challenging, the handcrafted features failed in the retrieval. Even though the texture-based and shape-based methods outperformed colour methods, the performance of neural networks is significantly better. However, when employing the pre-trained neural networks on Patterns dataset, the results are not satisfying. The results for each image category of CNN approaches are shown in Figure 6.7. Categories such as cobwebbed, crystalline, dotted and striped were retrieved with greater mAP than other categories because they contain real-world objects or at least commonly found textures on man-made objects (dotted and striped). Images from other categories were retrieved with a lower mean average precision. Figure 6.8 shows that some image categories barely form clusters.

Since the dataset is annotated, a fine-tuning of neural networks may help us. Fine-tuning is done by training a pre-trained neural network to adapt to the current dataset. We will consider fine-tuning the ResNet152 neural network since it performed the best in terms of mean average precision from the state-of-the-art neural networks (see in Figure 6.7).



	All	Banded	Chequered	Cobwebbed	Crystalline	Dotted	Fibrous	Lacelike	Potholed	Stratified	Striped
AlexNet	0.59	0.59	0.64	0.65	0.66	0.67	0.48	0.45	0.58	0.52	0.68
VGG	0.71	0.49	0.77	0.69	0.73	0.87	0.71	0.72	0.65	0.72	0.73
MobileNet	0.67	0.54	0.72	0.77	0.59	0.77	0.61	0.73	0.63	0.57	0.74
ResNet	0.74	0.51	0.66	0.82	0.85	0.75	0.71	0.84	0.78	0.64	0.78
EfficientNet	0.69	0.58	0.63	0.81	0.73	0.75	0.64	0.69	0.7	0.61	0.74

Figure 6.7: Comparison of mAP of CNN methods on different categories on Patterns dataset. Note that the colour scale range. A pre-trained ResNet152 architecture achieved a highest mean average precision 0.74 from the chosen neural networks. Dotted category was retrieved the best by VGG network. ResNet significantly outperformed EfficientNet on Lacelike category.

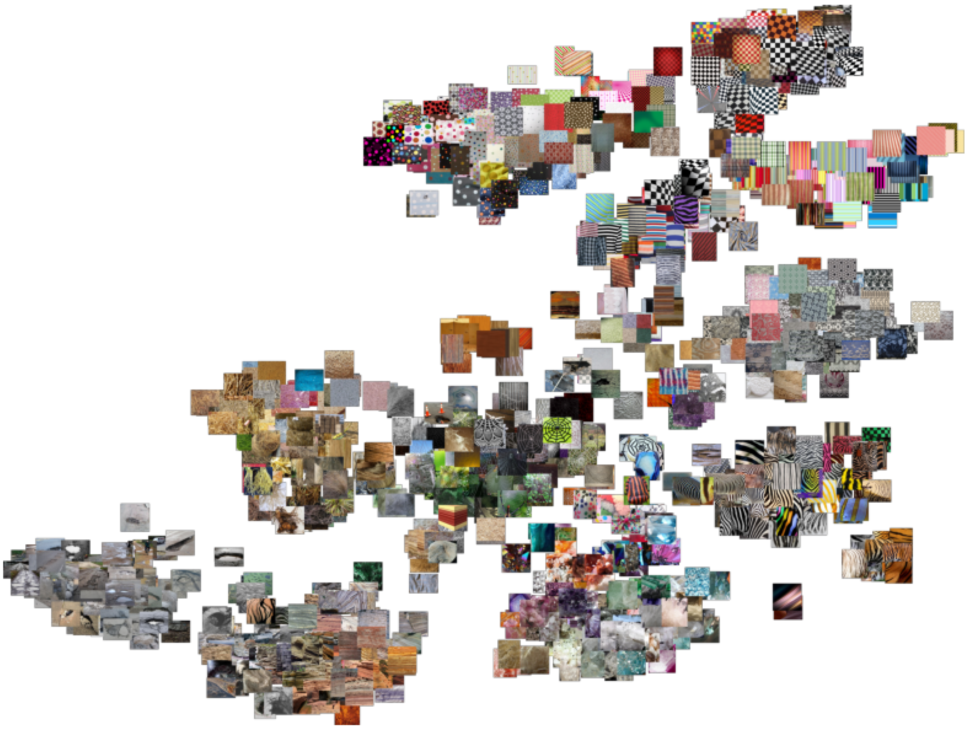


Figure 6.8: The visualization of the Patterns dataset with t-SNE. We used the pre-trained ResNet152.

Fine-tuning results

In the following, we describe some parts of our implementation. We fine-tune ResNet152 network. In other words, we take a pre-trained network with initialised weights and biases and set parameters such as learning rate, optimiser of neural network to train the network. Typically, learning rate is chosen smaller than usual because the parameters of the pre-trained network were already optimised after training on a huge dataset ImageNet. Since the dataset contains image class annotation, we can train it as a supervised classification task. As described in Chapter 4, we aim to minimise a given loss function (categorical cross-entropy) on a training set. At the same time, we monitor the validation loss of each epoch. When we encounter a validation loss smaller than ever before, we save the model as the model with the best loss. We stop the training part when a validation loss still increases for several epochs. The training and validation loss curves are shown in Figure 6.9, accuracy is shown in Figure 6.10.

The Python script `cnns_finetuning_patterns.py` is utilized when training neural network. We used a model `ResNet152`, optimizer `Adam` (an algorithm that modifies the attributes of the neural network, such as weights and learning rate) and initial learning rate `lr = 0.00001`. The scheduler is chosen as `ReduceLROnPlateau` (PyTorch documentation Paszke et al. [2019]) with parameters of `factor=0.9` and `patience=2`. Training of one epoch took approximately 13 minutes, therefore training of the model took for about a one day.

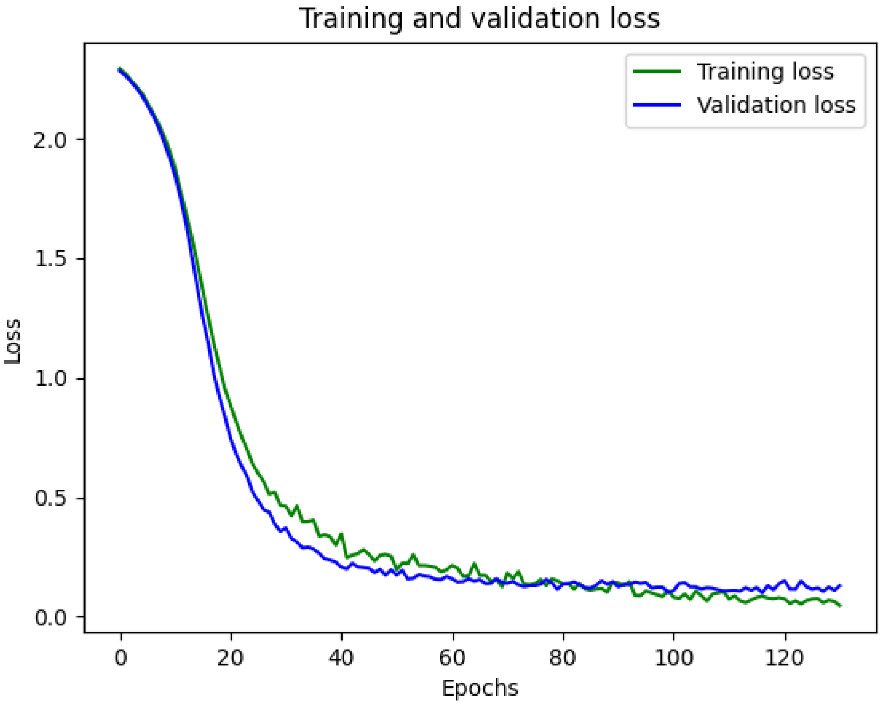


Figure 6.9: Comparison of training and validation loss while fine-tuning ResNet152 network on Patterns dataset. Note that the training loss is decreasing, wheras validation loss is decreasing and increasing from approximately epoch 90. This is phenomenon is known as overfitting. A model with the lowest validation loss is saved in order to generalise well.

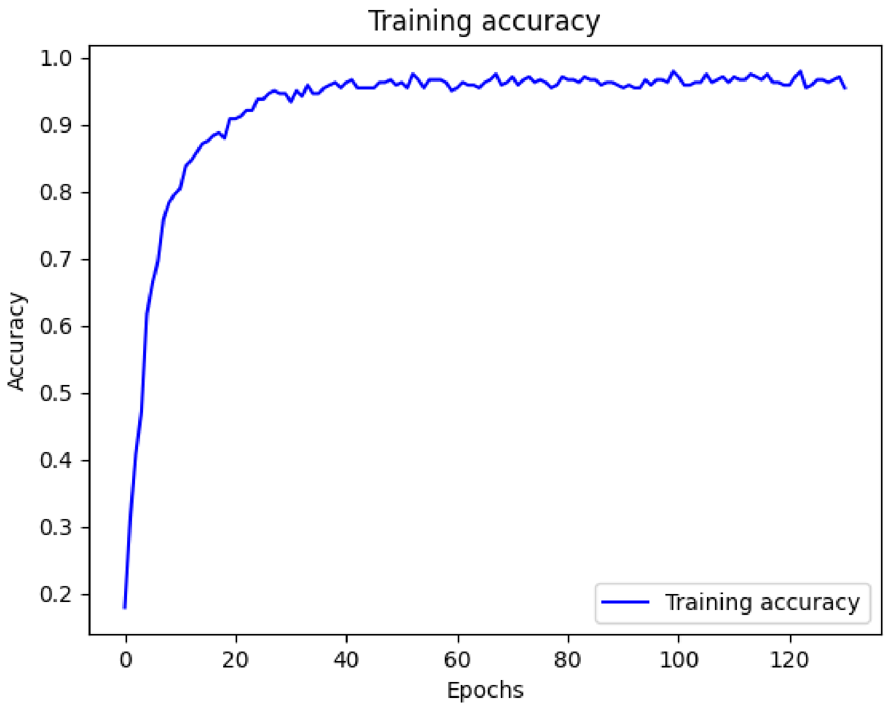


Figure 6.10: Accuracy fine-tuning ResNet152 network on Patterns dataset.

After training the neural network, we load a model with the smallest validation loss saved as a `.pth` file in the `feature_extraction_load_model.py` script. Based on this model, high-level image features are computed from a penultimate network layer. Therefore, each image from a dataset is described by its corresponding feature, and we save it as a `.npy` file.

The last step involves computing the overall mean average precision or the mean average precision computed on each class (based on a boolean parameter). The Python script `distances_evaluation.py` is used for this purpose, which loads a `.npy` file of image features and evaluates retrieval effectiveness in terms of mean average precision. Figure 6.11 is generated by comparing mean average precision on classes. t-SNE visualisation shown in Figure 6.12 proves that the image feature vectors in the high-dimensional space are clustered since they are clustered in two-dimensional space.

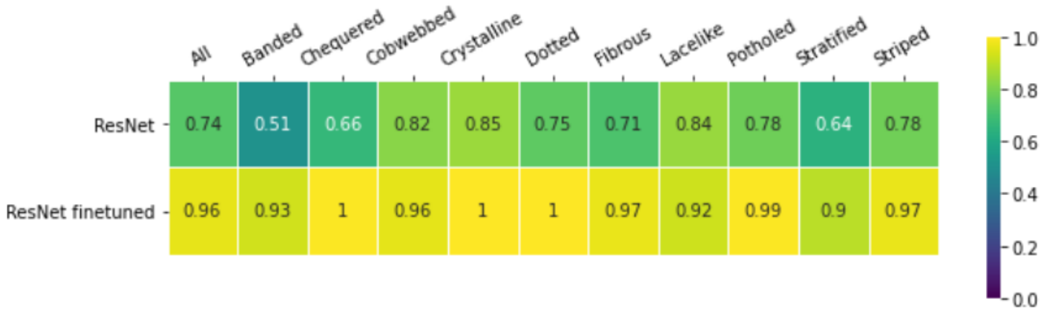


Figure 6.11: Comparison of mAP of ResNet152 and fine-tuned ResNet152 on different categories on Patterns dataset.

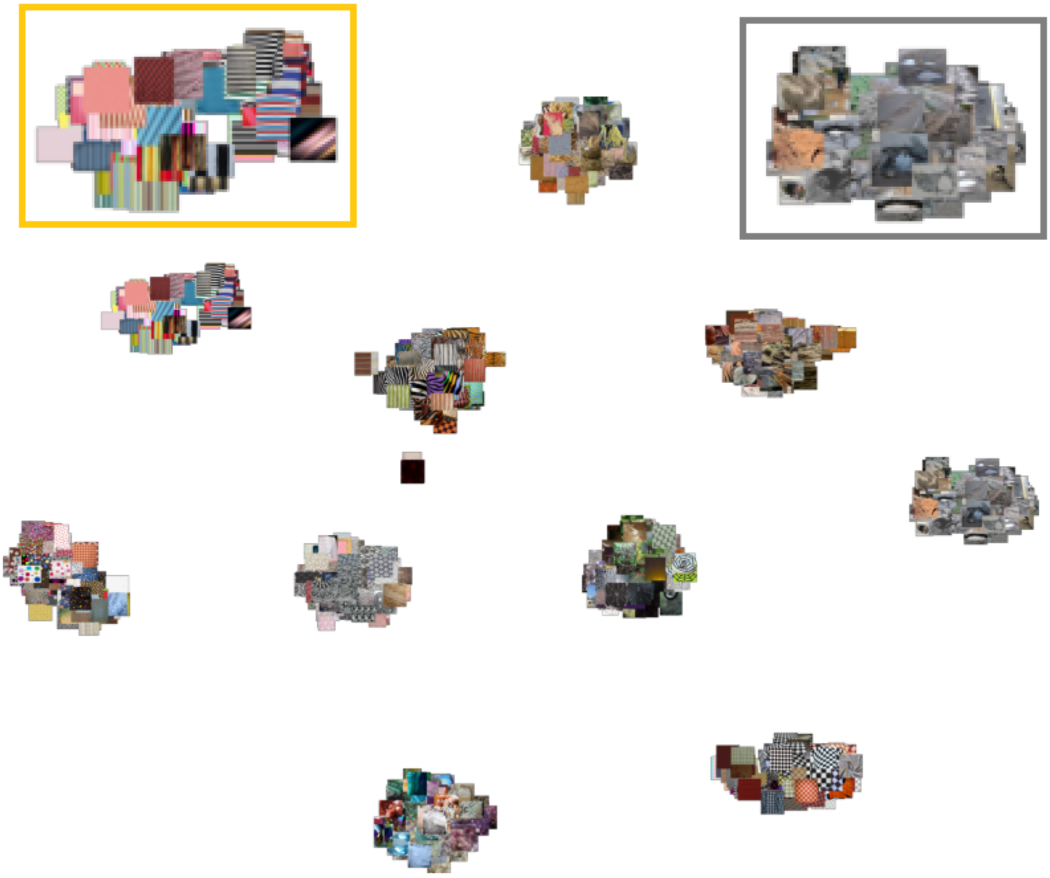


Figure 6.12: The visualization of the Patterns dataset with t-SNE. The clusters of potholed and banded categories are zoomed. We finetuned ResNet152 network.

6.4.3 GPR1200 dataset

Newly introduced GPR1200 dataset by Schall et al. [2021] includes 12 000 images from 1 200 categories, each category containing 10 images. It was collected from six different image domains from the listed collections and proposed as General-Purpose Retrieval Benchmark. GPR1200 dataset provides a suitable dataset covering many of the domains found in image collections.

1. Google Landmarks V2 (natural and architectural landmarks)
2. ImageNet Sketch (black and white sketches of animals and other objects)
3. iNat (plants, animals, insects and fungi)
4. INSTRE (planar images and photographs of logos and toys)
5. SOP (products and objects, partly isolated)
6. IMDB Faces (human faces)

State-of-the-art models trained on ImageNet (Russakovsky et al. [2014]) generating high-level image descriptors are fine-tuned. We used this dataset for evaluation purposes of deep learning methods. The sample of the GPR1200 dataset is shown in Figure 6.13.

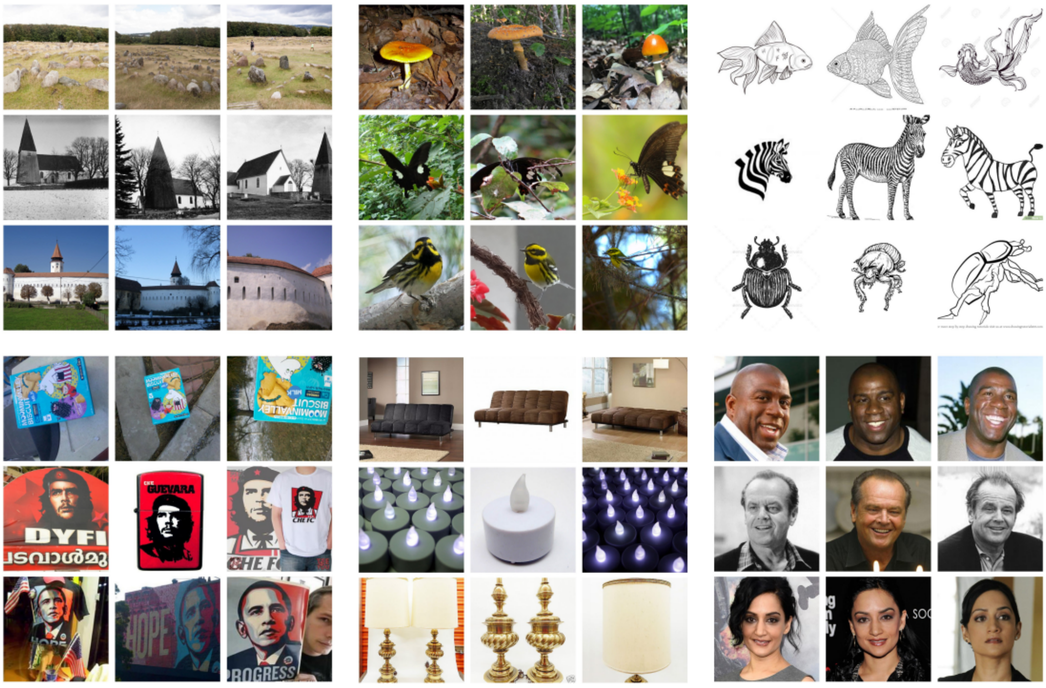


Figure 6.13: Example images from all of the subsets of the newly introduced GPR1200 dataset (Schall et al. [2021]). Domains from left to right, top to bottom: Landmarks, Nature, Sketches, Objects, Products, and Faces.

We employed the pre-trained architectures of ResNet152, EfficientNet-B0, EfficientNet-B3 and EfficientNet-B7 networks. Since the dataset consists of 1200 categories, we evaluate only the overall mean average precision. Based on the previous observations, we employed a cosine distance measure.

	Method	Parameter	mAP
CNN	ResNet	152	0.474
CNN	EfficientNet	b0	0.488
CNN	EfficientNet	b3	0.476
CNN	EfficientNet	b7	0.417

Table 6.4: Comparison of methods' performance of pre-trained state-of-the-art methods on the GPR1200 dataset. We used cosine distance since it provided the best results among other distance measures.

Fine-tuning results

Since this dataset is the most challenging, we fine-tune ResNet152 as on the Patterns dataset. The process of fine-tuning described in the previous section is similar; thus, we describe it briefly. We used `cnns_finetuning_gpr1200.py` for further training of the pre-trained network. Since GPR1200 is a large dataset of high-resolution images, we decided on a greater learning rate to speed up the early part of the training. We set the same parameters, except for the initial learning rate set as `lr=0.000065`. When training ResNet152, each epoch took approximately one hour and fifteen minutes.

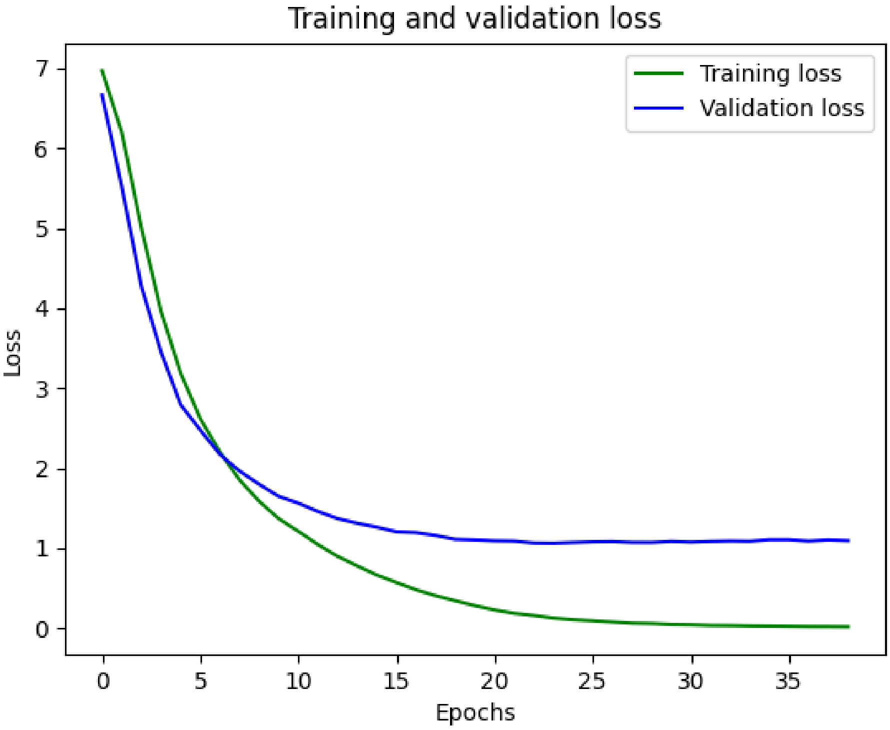


Figure 6.14: Comparison of training and validation loss while fine-tuning ResNet152 network on GPR1200 dataset.

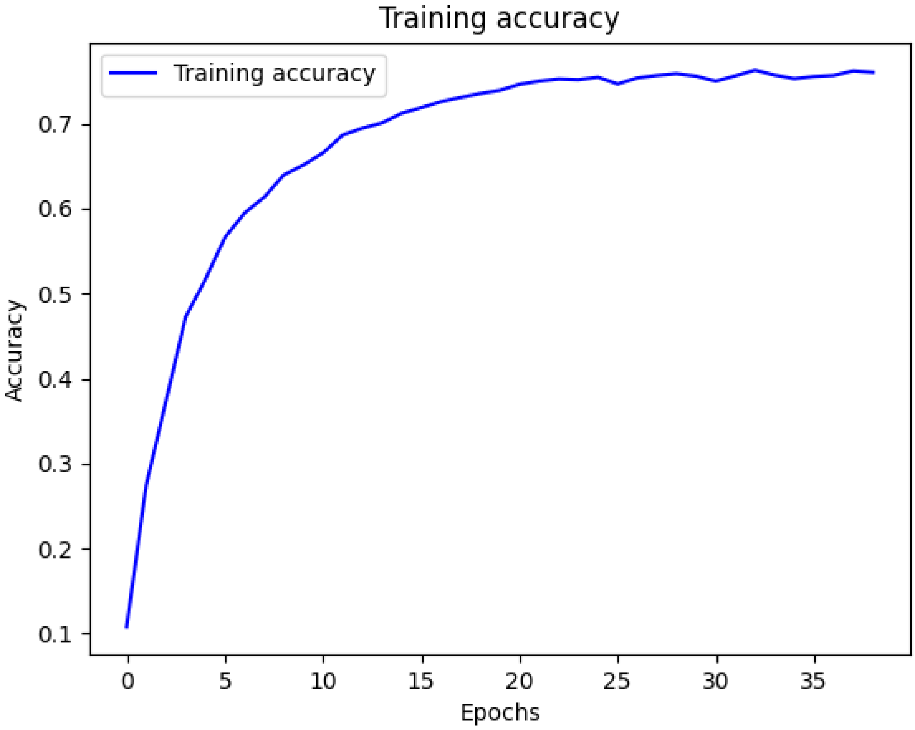


Figure 6.15: Accuracy fine-tuning ResNet152 network on GPR1200 dataset.

We also tried to fine-tune EfficientNet-b3, which is quite faster. However, by fine-tuning the ResNet we achieved saving a model with a smaller validation loss. The resulting loss curves are shown in Figure 6.14 and accuracy on 6.15. From Figure 6.14 we see that the initial learning rate is too big. Therefore, we recommend degrading it for training.

From fine-tuning performed in script `cnns_finetuning_gpr1200.py`, we continue with loading a saved model from the `.pth` file, process image features in the `feature_extraction_load_model.py` with appropriate parameter. The features are again saved in a `.npy` file. In the end, we evaluate the resulting mean average precision in `distances_evaluation.py` script.

The fine-tuned model did not improve the retrieval effectiveness. Since the GPR1200 dataset covers various scenes, it may more similar to ImageNet dataset, therefore parameters should be changed carefully. The work of Schall et al. [2021] provides the-state-of-the-art research together with implemented code. It shows retrieval results comparison over a wide range of models. The best model (ViT-L) tested in the work achieves mean average precision of 0.632, followed by the results of Swin-L achieving the value of 0.63 on a whole dataset. Mean average precision is also computed for each of the six dataset categories (see Schall et al. [2021]).

6.4.4 Posters dataset

The Vienna City Library (vie) contains important documents related to the history of Vienna, Austria. It preserves 500,000 books, 2,000 newspapers and magazines, 300,000 posters, 500,000 autographs. A sample of 5050 posters scanned in high-quality resolution is provided by the Vienna City Library in order to perform a CBIR task. There are advertisements, concert tickets, or political propaganda handouts from the war period within the dataset. Examples of images are shown in Figure 6.16.

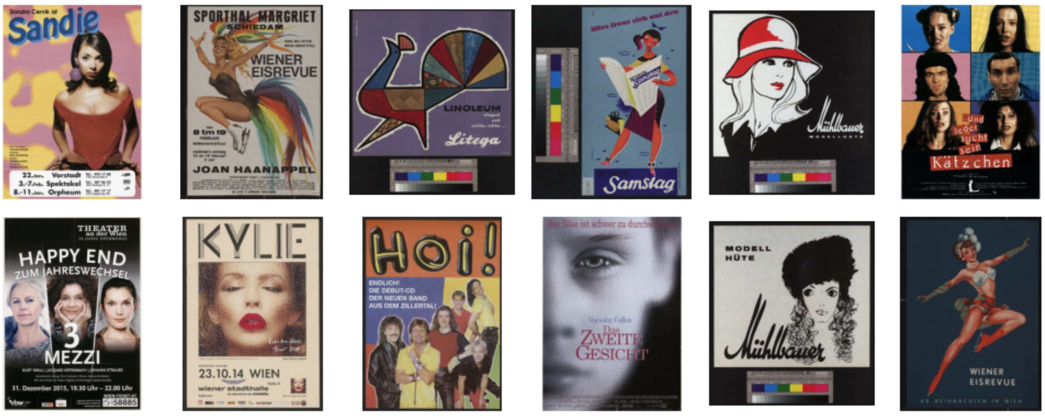


Figure 6.16: Examples of images from Posters dataset vie.

This dataset is not annotated and, therefore, not the retrieval performance is not evaluated. However, the pre-trained EfficientNet-b0 model is employed since it gave the best results on GPR1200 dataset. An inference is applied to each image, and the penultimate network's layer is used as an image descriptor. The retrieval results for a sample of ten queries are shown in Figure 6.20. t-SNE algorithm is applied to computed high-level image features, and the t-SNE plot is visualised in Figure 6.17.

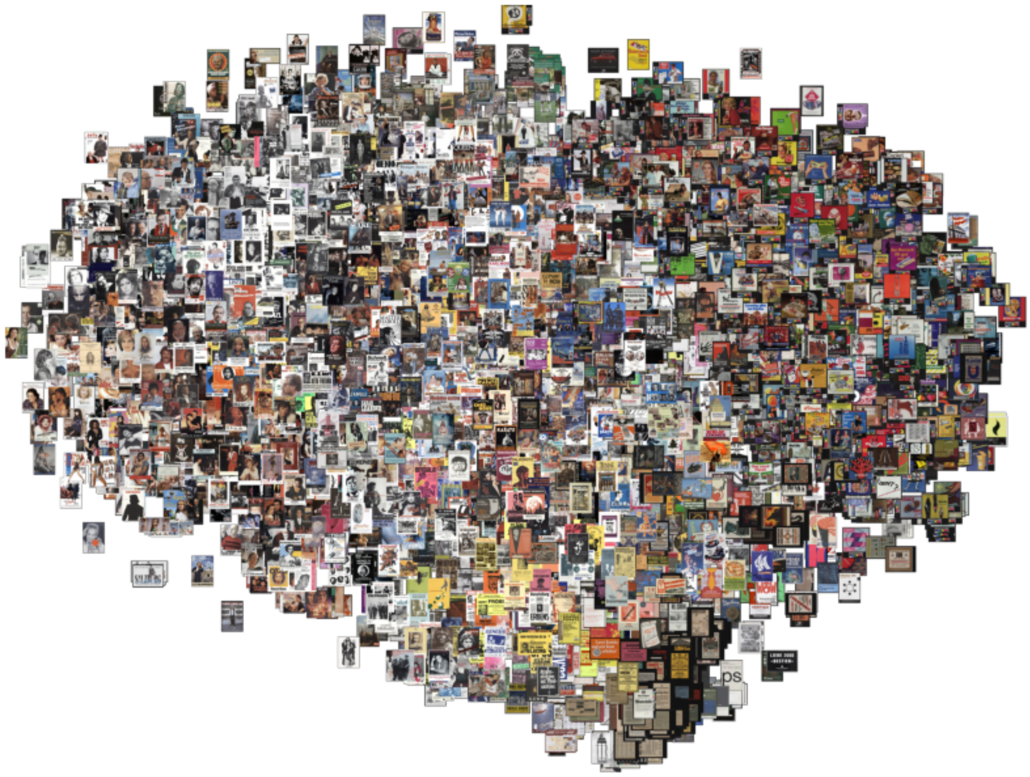


Figure 6.17: The visualisation of the Posters dataset with t-SNE. Since the dataset was not created artificially for computer vision task evaluation as in previous datasets, it is not divided into several clusters. We can observe that the text images are close to each other in the lower part. The images of women are accumulated on the left side.

6.5 Results on example queries

The retrieval results are shown in the following Figures 6.18, 6.19, 6.20 and 6.21. The visualisation is implemented in Python library matplotlib.

Running the `show_retrieved_images` function from the script `distances.py` chooses plots these figures, where there is a query image on the left. There appear the top K similar images for K defined by a user. The features for each dataset image are already pre-computed. Other parameters of the script are the dataset name, the number of retrieved images and the choice of a particular method, utilised parameter and metric. We chose a variant of the state-of-the-art EfficientNet or ResNet152 with cosine distance when generating these images. The ResNet152 is fine-tuned for Patterns dataset.

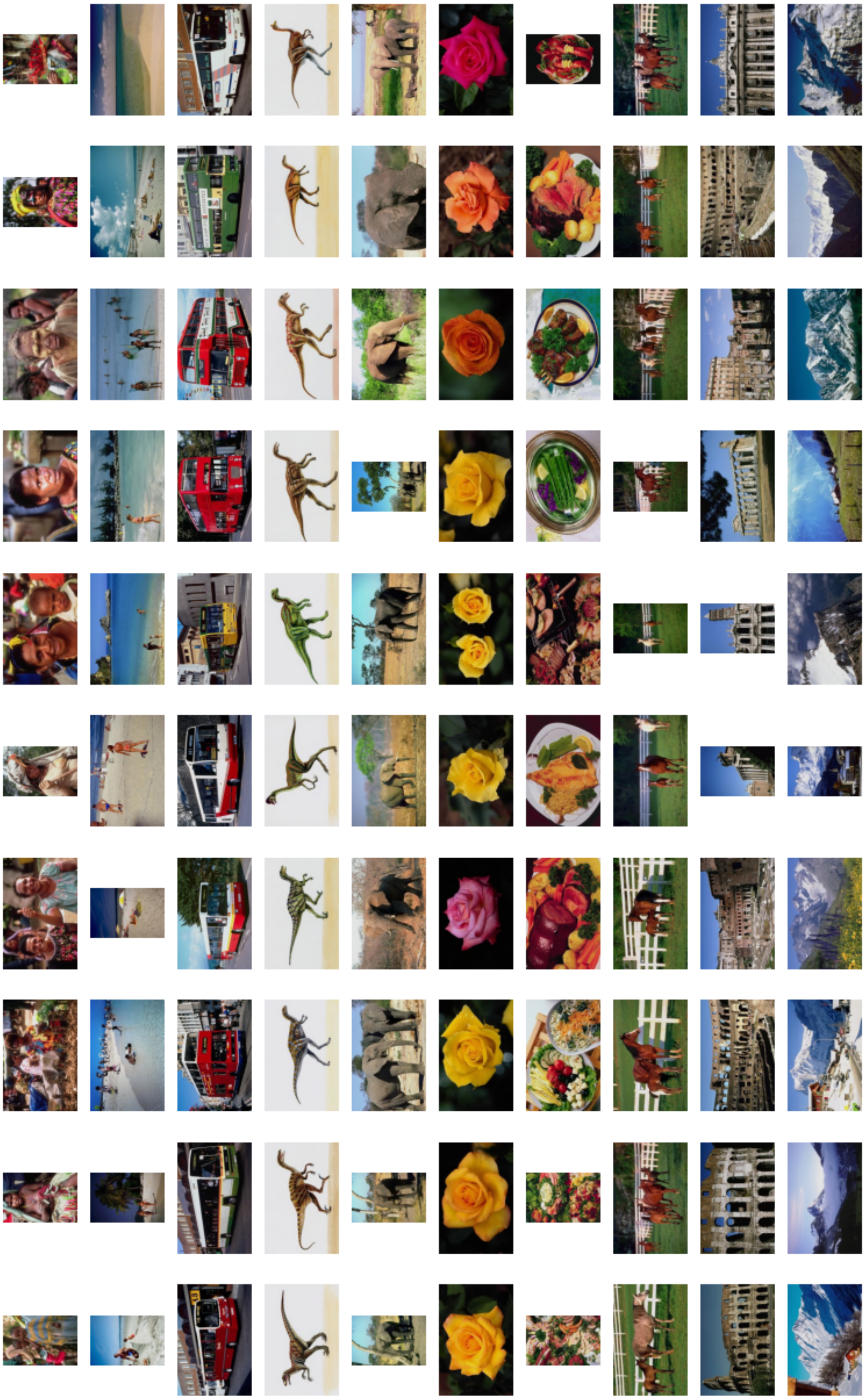


Figure 6.18: Retrieval results of the Wang dataset. One query image from each image category is on the left and retrieved images are in left-to-right order.

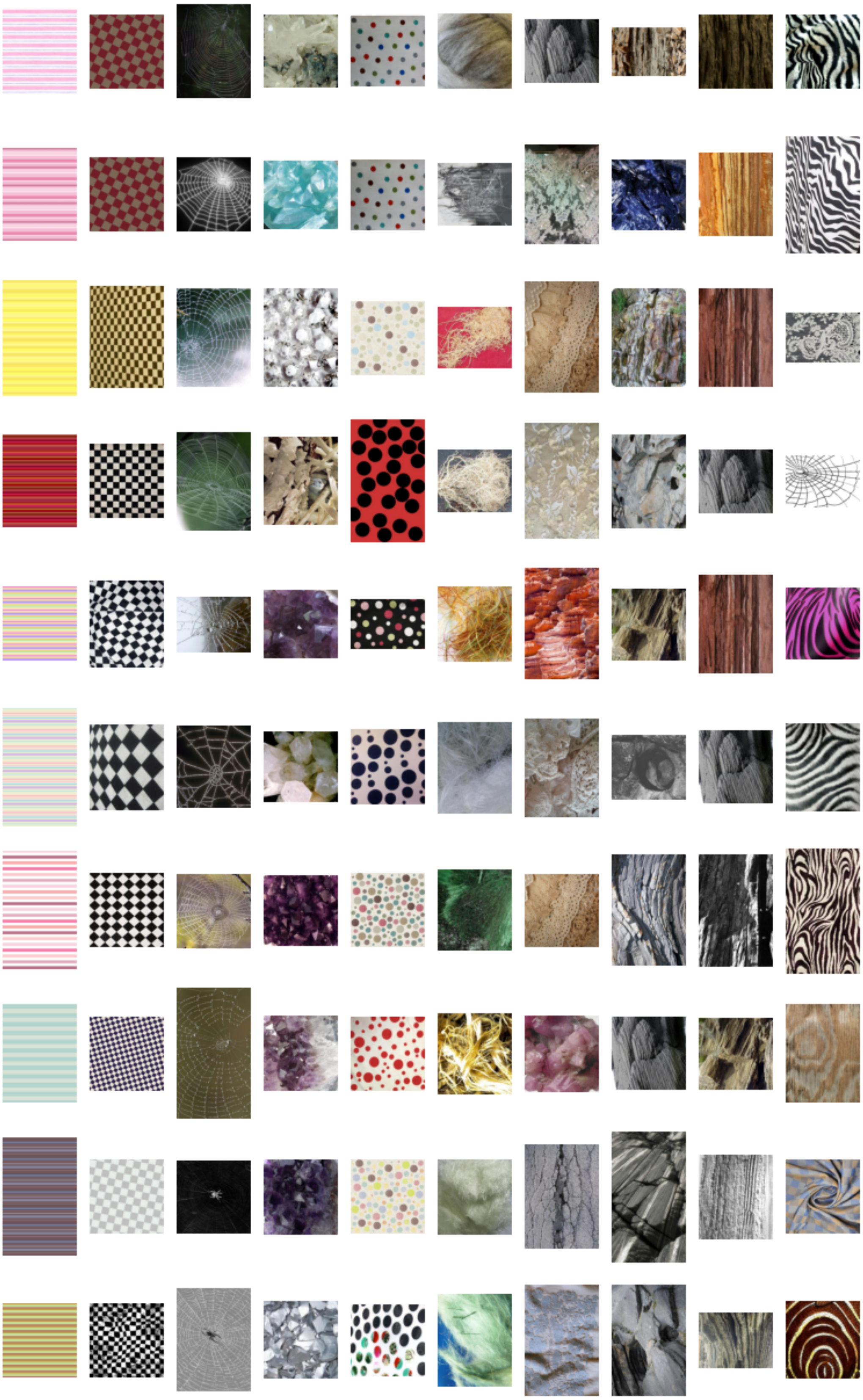


Figure 6.19: Retrieval results of the Patterns dataset. One query image from each image category is on the left and retrieved images are in left-to-right order.

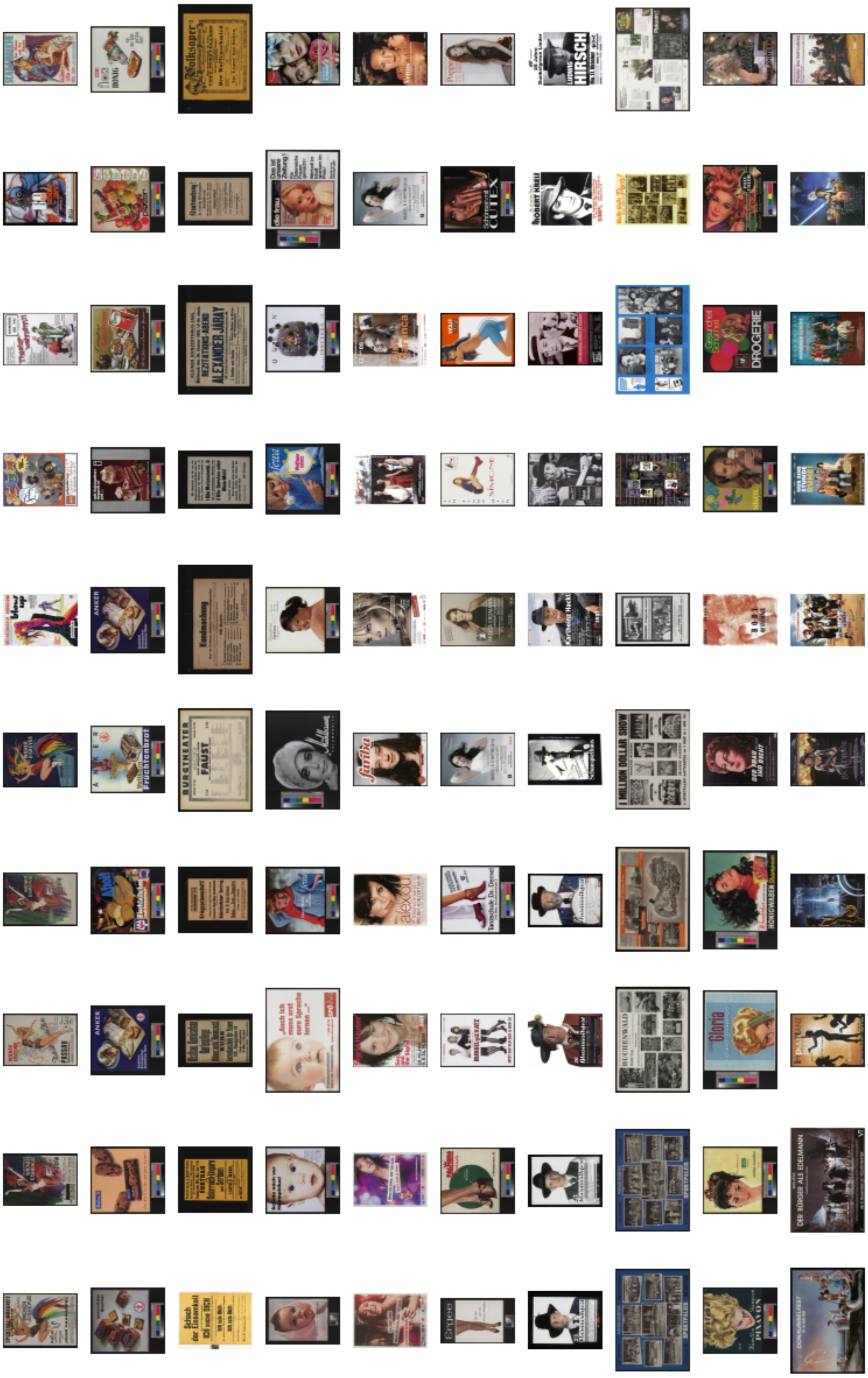


Figure 6.20: Retrieval results of the Posters dataset. Each query image is a Posters database image on the left and retrieved images are in left-to-right order.

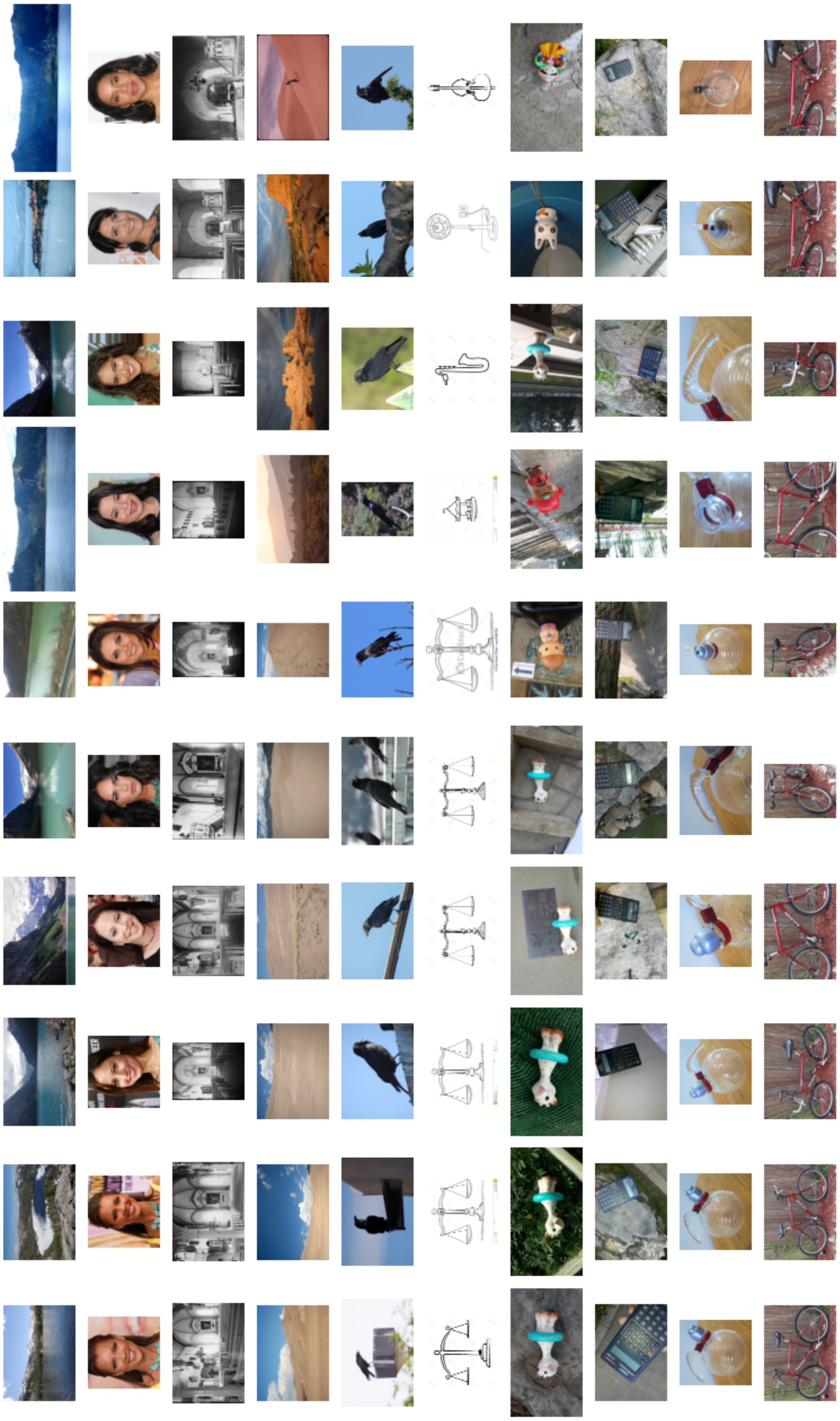


Figure 6.21: Retrieval results of the GPR1200 dataset. Each query image is a GPR1200 database image on the left and retrieved images are in left-to-right order.

Chapter 7

Web application

A web application is designed to serve a purpose: to solve an online image retrieval task. From the user's point of view, it is easy to use, fast and does not require any installation. An application was developed in order to retrieve images from the Posters dataset. This dataset contains images of posters gathered in the Vienna City Library.

7.1 Navigation

Users can simply access a web application by the web address. It requests a user to upload a digital image as a file from the computer. It can be a poster from the Posters database or any desired image. After choosing an image file and clicking on an upload button, the five most similar images to a given image appear. Since the dataset is relatively small and the features are computed efficiently, displaying similar images takes less than two seconds. Moreover, an application is designed so that a user can upload a new image repeatedly. The result of a user request for a poster of cake from the Posters dataset is shown in Figure 7.1.

7.2 Implementation details

The Graphical User Interface is based on a simple HTML webpage form. This form sends the selected image to the Flask server, which provides the service for counting and comparing image features.

For our backend, we used Python because it has an extensive machine learning community. We worked with standard Python libraries such as NumPy, Pandas, PyTorch, OpenCV and Flask. The search for the most similar images is designed the same way as the typical CBIR pipeline. The Posters database image features are pre-computed and stored in a file regarding its offline stage. An online stage consists of computing uploaded images' features based on pre-trained EfficientNet-b3 and then comparing this image feature to all database images with a cosine distance metric. The database images with the smallest cosine distance value are retrieved and displayed.

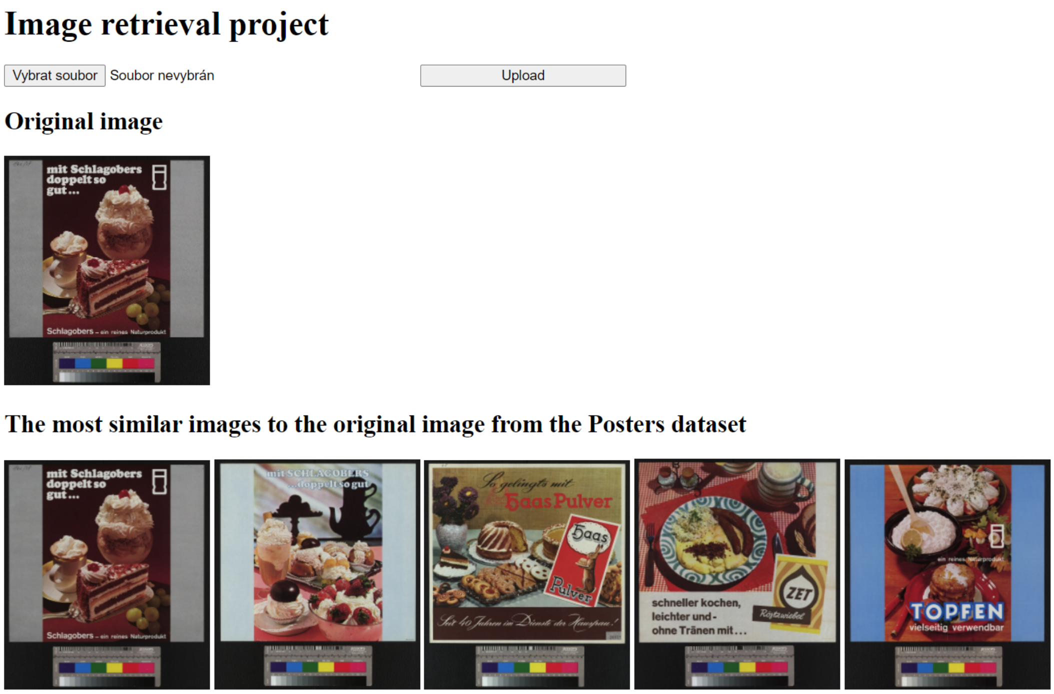


Image retrieval project

Soubor nevybrán

Original image

The most similar images to the original image from the Posters dataset

mit Schlagobers doppelt so gut...
Schlagobers – ein reines Naturprodukt
mit SCHLAGOBERS doppelt so gut
So gelingt's mit Haas Pulver
Haas Pulver
Seit 40 Jahren im Dienste der Hausfrau!
schneller kochen, leichter und - ohne Tränen mit...
ZET
Röstzwiebel
ein reines Naturprodukt
TOPFEN
vielseitig verwendbar

Figure 7.1: Web application example. A user chooses a query image, and the five most similar images from the Posters dataset appear. Notice that all of the retrieved images contain food.

Chapter 8

Conclusion

This thesis explored techniques to tackle the Content-Based Image Retrieval task. Its main advantage lies in the usage of images without textual description. Indexing images by meaningful feature vectors and returning the closest representations is an efficient image retrieval method. Since the design of valuable features of all database images is computationally slow, calculating the database image representations beforehand is key to providing a fast image retrieval.

8.1 Summary

We started by introducing the general knowledge of this topic. In detailed research on handcrafted image features, we defined image descriptors in mathematical language utilized before the advent of convolutional neural networks. Secondly, we explained a general overview of convolutional neural networks. Furthermore, we briefly described the architectures of the existing state-of-the-art pre-trained models and explained their application in the CBIR task.

We implemented various feature extraction algorithms and ran them on existing image datasets with several parameters. Afterwards, the methods' performance was explored by evaluating the mean average precision on different images. t-SNE algorithm provided excellent visualization of high-level image descriptors in two-dimensional space. On top of that, we fine-tuned the state-of-the-art neural network on a dataset with annotated classes and achieved better retrieval results. Dataset augmentation was employed. In order to estimate the performance of the image classification task, we monitored training and validation losses. We used transfer learning to address the image retrieval task.

We propose the use of fine-tuned neural networks for datasets with annotated classes. For datasets without annotations, we propose using the pre-trained networks. The use cosine distance metric for all CNN-based approaches is recommended. We also suggest considering features pre-processing. All presented experiments were tested on the pre-defined queries from each dataset with defined distance measures.

Based on the investigation of the effect of the centring a normalization of feature vectors, we recommend considering it as a potential improvement. Mainly, we contributed to the University of Vienna by designing and implementing an efficient image retrieval pipeline. In addition, we developed a functional, easy to use web application to solve the CBIR task on a dataset of posters from the

Vienna City Library. A user may upload their image file. By slightly modifying an application, we may utilize it for another dataset. When the image database features are pre-computed, we may use them internally on our image collections.

8.2 Future work

Unlike handcrafted features, convolutional neural networks offer strong adaptation abilities. The state-of-the-art approaches offer promising results on large datasets such as GPR1200, designed for general content-based image retrieval by Schall et al. [2021]. Due to their enormous potential, we propose to research their abilities further.

Although we researched CBIR and successfully deployed an efficient model on various datasets, we provide suggestions for future work:

- To provide an implementation for multiple queries.
- Exploration of other network's architecture.
- To improve a web application interface.
- To include user-defined parameters of the number of retrieved images and the type of counted features in the web application.
- Feature extraction and methods' comparison on larger image datasets.

8.3 State-of-the-art research

Content-based image retrieval is still a research problem, and many deep learning researchers try to invent more complicated models that achieve better results.

Image retrieval on the dataset of three-dimensional buildings was investigated in the work of Radenovic et al. [2018] on the Oxford and Paris dataset. Its annotation was introduced about ten years ago when the annotators had a different perception of the image retrieval limitations. Thus a new dataset annotation was introduced. Such image retrieval strives to retrieve images of buildings captured from different viewpoints on different scales. The research of Radenovic et al. [2018] provide an extensive evaluation of image retrieval methods. One proposed method was fine-tuned ResNet with presented Generalized-Mean pooling (GeM pooling). GeM pooling includes learnable parameters. Thus the standard non-trainable pooling layers, such as max pooling or average pooling, are special cases. It was demonstrated that GeM Pooling boosts the performance over standard pooling layers. Moreover, the triplet loss and contrastive loss were introduced and deployed (Radenović et al. [2019]). For further research the instance image retrieval we refer to related papers (see Radenović et al. [2016], Radenović et al. [2019]).

Furthermore, there already exists another state-of-the-art approach using an architecture called transformers, which researchers presently investigate in the context of content-based image retrieval in the work of El-Nouby et al. [2021] and Gkelios et al. [2021]. The retrieval effectiveness of transformers was investigated by Schall et al. [2021] on the GPR1200 dataset. It has been compared to the results of fine-tuned networks such as EfficientNet.

Bibliography

Wienbibliothek im rathaus. Wienbibliothek im Rathaus. URL `https://www.wienbibliothek.at/english`.

Swati Agarwal, A. K. Verma, and Preetvanti Singh. Content based image retrieval using discrete wavelet transform and edge histogram descriptor. In *2013 International Conference on Information Systems and Computer Networks*, pages 19–23, 2013. doi: 10.1109/ICISCON.2013.6524166.

Alaa Al-Hamami and Hisham Al-Rashdan. Improving the effectiveness of the color coherence vector. *Int. Arab J. Inf. Technol.*, 7:324–332, 2010.

Mohammed Alkhawlani, Mohammed Elmogy, and Hazem El-Bakry. Content-based image retrieval using local features descriptors and bag-of-visual words. *International Journal of Advanced Computer Science and Applications*, 6:212–219, 09 2015.

Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 584–599, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1.

Jana Bátoryová. Searching image collections using deep representations of local regions. 2020.

Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *CoRR*, abs/1605.07678, 2016. URL `http://arxiv.org/abs/1605.07678`.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Luigi Cinque, Gianluigi Ciocca, Stefano Levialdi, A. Pellicanò, and Raimondo Schettini. Color-based image retrieval using spatial-chromatic histograms. *Image Vis. Comput.*, 19:979–986, 2001.

Wikimedia Commons. Colour systems images. URL `https://commons.wikimedia.org/wiki/User:Datumizer`.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training vision transformers for image retrieval. *CoRR*, abs/2102.05644, 2021. URL https://arxiv.org/abs/2102.05644.

Marios Gavrielides, Elena Sikudova, and Ioannis Pitas. Color-based descriptors for image fingerprinting. *Multimedia, IEEE Transactions on*, 8:740 – 748, 09 2006. doi: 10.1109/TMM.2006.876290.

Socratis Gkelios, Yiannis S. Boutalis, and Savvas A. Chatzichristofis. Investigating the vision transformer model for image retrieval tasks. *CoRR*, abs/2101.03771, 2021. URL https://arxiv.org/abs/2101.03771.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Kavitha Halappa. Image retrieval using hog and edge features. 11 2013.

Kavitha Halappa and M.V. Sudhamani. Content-based image retrieval using edge and gradient orientation features of an object in an image from database. *Journal of Intelligent Systems*, 25, 01 2015. doi: 10.1515/jisys-2014-0088.

Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973. doi: 10.1109/TSMC.1973.4309314.

Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000. URL https://cds.cern.ch/record/478079.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Geoffrey E. Hinton and Sam T. Roweis. Stochastic neighbor embedding. In *NIPS*, 2002.

Kurt Hornik, Maxwell B. Stinchcombe, and Halbert L. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:498–520, 1933.

Jing Huang, S Kumar, Mandar Mitra, Wei-jing Zhu, and Ramin Zabih. Image indexing using color correlograms. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 05 1997.

Noor Ibraheem, Mokhtar Hasan, Rafiqul Zaman Khan, and Pramod Mishra. Understanding color models: A review. *ARPN Journal of Science and Technology*, 2, 01 2012.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in*

Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

A. Ramesh Kumar and Devaraj Saravanan. Content based image retrieval using color histogram. 2013.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

Chuen-Horng Lin, Rong-Tai Chen, and Yung-Kuan Chan. A smart content-based image retrieval system based on color and texture feature. *Image and Vision Computing*, 27(6):658–665, 2009. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2008.07.004. URL https://www.sciencedirect.com/science/article/pii/S0262885608001522.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network, 2013. URL https://arxiv.org/abs/1312.4400.

Mamta Martolia, Nilesh Dhanore, Anupam Singh, Vivek Shahare, and Nitin Arora. A modified local binary pattern (lbp) for content-based image retrieval. 29:1630–1644, 01 2020.

Maryam Mokhtari, Parvin Razzaghi, and Shadrokh Samavi. Texture classification using dominant gradient descriptor. pages 100–104, 09 2013. ISBN 978-1-4673-6184-2. doi: 10.1109/IranianMVIP.2013.6779958.

Subrahmanyam Murala, R.P. Maheshwari, and Balasubramanian Raman. Local tetra patterns: A new feature descriptor for content-based image retrieval. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 21:2874–86, 05 2012. doi: 10.1109/TIP.2012.2188809.

Izaak Neutelings. Neural networks, 2022. URL https://tikz.net/neural_networks/.

Adi Nurhadiyatna, Arnida Latifah, and Driszal Fryantoni. Gabor filtering for feature extraction in real time vehicle classification system. 09 2015. doi: 10.1109/ISPA.2015.7306026.

T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. doi: 10.1109/TPAMI.2002.1017623.

Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. volume 1842, pages 404–420, 06 2000. ISBN 978-3-540-67685-0. doi: 10.1007/3-540-45054-8_27.

Shilpa Pant. Content based image retrieval using color feature. *International journal of engineering research and technology*, 2, 2013.

Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *MULTIMEDIA '96*, 1997.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Matti Pietikäinen, T Ojala, and Zelin Xu. Rotation-invariant texture classification using feature distribution. *Pattern Recognition*, 33:43–52, 01 2000. doi: 10.1016/S0031-3203(99)00032-1.

F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016.

Filip Radenovic, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. *CoRR*, abs/1803.11285, 2018. URL http://arxiv.org/abs/1803.11285.

Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019. doi: 10.1109/TPAMI.2018.2846566.

M. Rao, Kavitha Chaduvula, B. Rao, and Dr Govardhan. *Content Based Image Retrieval Based on Dominant Color, Scan Pattern Co-occurrence Matrix of a Motif and Shape*, pages 353–357. 01 2011. ISBN 978-3-642-25733-9. doi: 10.1007/978-3-642-25734-6_53.

Abdolreza Rashno and Elyas Rashno. Content-based image retrieval system with most relevant features among wavelet and color features, 2019. URL https://arxiv.org/abs/1902.02059.

Carlos Rivero-Moreno and Stéphane Bres. Conditions of similarity between hermite and gabor filters as models of the human visual system. pages 762–769, 08 2003. ISBN 978-3-540-40730-0. doi: 10.1007/978-3-540-45179-2_93.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2014. URL https://arxiv.org/abs/1409.0575.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018. doi: 10.48550/ARXIV.1801.04381. URL https://arxiv.org/abs/1801.04381.

Konstantin Schall, Kai Uwe Barthel, Nico Hezel, and Klaus Jung. GPR1200: A benchmark for general-purpose content-based image retrieval. *CoRR*, abs/2111.13122, 2021. URL https://arxiv.org/abs/2111.13122.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. URL https://arxiv.org/abs/1409.1556.

Jyotsna Singh, Ahsaas Bajaj, Anirudh Mittal, Ansh Khanna, and Rishabh Karwayun. Content based image retrieval using gabor filters and color coherence vector. In *2018 IEEE 8th International Advance Computing Conference (IACC)*, pages 290–295, 2018. doi: 10.1109/IADCC.2018.8692123.

S. Singh and K. Hemachandran. Content-based image retrieval using color moment and gabor texture feature. *International Journal of Computer Science Issues*, 9:299–309, 09 2012.

Manimala Singha and K Hemachandran. Content based image retrieval using color and texture. *Signal Image Processing : An International Journal*, 3, 02 2012. doi: 10.5121/sipij.2012.3104.

George Stockman and Linda G. Shapiro. *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001. ISBN 0130307963.

Milan Straka. Deep learning, lectures. 2022. URL https://ufal.mff.cuni.cz/courses/npfl114/2122-summer.

M.J. Swain and D.H. Ballard. Indexing via color histograms. In *[1990] Proceedings Third International Conference on Computer Vision*, pages 390–393, 1990. doi: 10.1109/ICCV.1990.139558.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019. URL http://arxiv.org/abs/1905.11946.

Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training. *CoRR*, abs/2104.00298, 2021. URL https://arxiv.org/abs/2104.00298.

Bino Vadakkenveettil. Grey level co-occurrence matrices: Generalisation and some new features. *International Journal of Computer Science, Engineering and Information Technology*, 2:151–157, 04 2012. doi: 10.5121/ijcseit.2012.2213.

Egon L. van den Broek. Human-centered content-based image retrieval. *Neuroscience Research Communications - NEUROSCI RES COMMUN*, 01 2005.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Oana Astrid Vatamanu, Mirela Frandes, Mihaela Ionescu, and Simona Apostol. Content-based image retrieval using local binary pattern, intensity histogram and color coherence vector. In *2013 E-Health and Bioengineering Conference (EHB)*, pages 1–6, 2013. doi: 10.1109/EHB.2013.6707396.

Anna Saro Vijendran and S. Vinod Kumar. A new content based image retrieval system by hog of wavelet sub bands. *International journal of engineering research and technology*, 3, 2014.

James Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23:947–963, 10 2001. doi: 10.1109/34.955109.

Xiang yang Wang, Yong-Jian Yu, and Hongying Yang. An effective image retrieval scheme using color, texture and shape features. *Comput. Stand. Interfaces*, 33: 59–68, 2011.

Dengsheng Zhang, Aylwin Wong, Maria Indrawan, and Guojun Lu. Content-based image retrieval using gabor texture features. 2000.

Vilém Zouhar, Marius Mosbach, Miaoran Zhang, and Dietrich Klakow. Knowledge base index compression via dimensionality and precision reduction. 04 2022.

List of Figures

List of Abbreviations

CBIR	Content-Based Image Retrieval.
CCV	Colour Coherent Vector.
CNN	Convolutional Neural Network.
GLCM	Grey-Level Co-Occurrence Matrix.
HOG	Histogram of Oriented Gradients.
HSV	Hue Saturation Value.
LBP	Local Binary Pattern.
mAP	Mean Average Precision.
PCA	Principal Component Analysis.
P@K	Precision at K.
ReLU	Rectified Linear Unit.
RGB	Red Green Blue.
t-SNE	t-Distributed Stochastic Neighbour Embedding.