Bachelor thesis
Computer Science

A deep residual network for recognizing cluster-based generic photo categories

Author:
Evi Sijben
s4476727

Internship supervisior:
L. Scholten (Luuk)

Second supervisor:
Prof. dr. T.M. Heskes (Tom)
tomh@cs.ru.nl

August 20, 2018

Abstract

There is a need for generic categories which categorize photos in a relevant way in relation to a photo collection. These categories can be used for analyzing the content of photo collections. This work provides a neural network for recognizing categories that are relevant to the photos in photo collections.

Contents

Chapter 1

Introduction

There is a need for generic categories which categorize photos in a relevant way in relation to a photo collection. Hereby, generic indicates that the categories can be used for multiple purposes.

The method I used for constructing categories is comparable to the method in [13]. This method consist of discovering concepts as illustrated in figure 1.1. Thereafter I will transform the found groupings of photos into categories in an experiment in which humans preform a task on the found results. The emerged categories are thereafter evaluated on their relevance and learnability by measuring the distribution of the photos over the categories, calculating the Fleiss' kappa and investigating the confusion between categories. After the evaluation some adjustments are made to the categories in order to improve relevance and learnability. To be able to recognize the constructed categories a data set is made on which a neural network is trained.

The categories are tools for analyzing photos in photo collections.

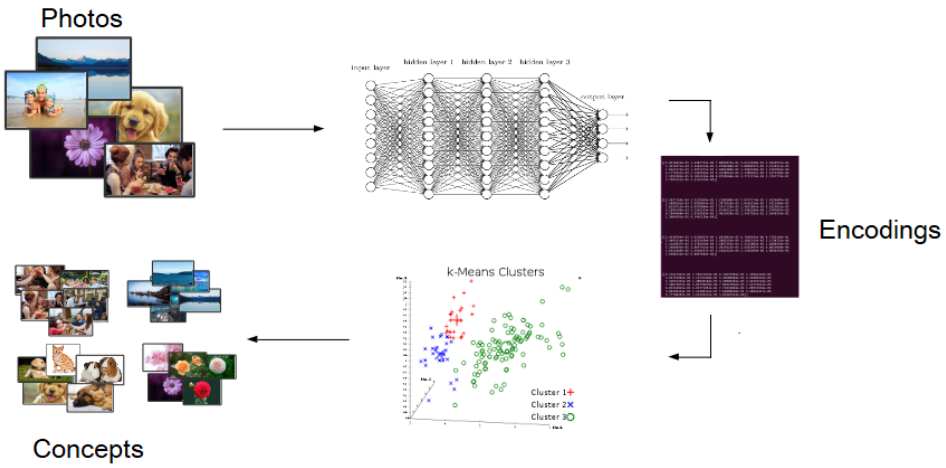

Figure 1.1: Total flow of discovering concepts. First, one selects photos. Second one encodes these photos using a neural network. At last, one tries to discover concepts by clustering these encodings.

Chapter 2

Preliminaries

This chapter gives details about theories and methods that were used for the research as described in the research chapter. In the first section I will elaborate on theory about concept learning and when categories are relevant. This will give a basis for the guidelines for evaluating generic categories. In the second section I will explain how several methods (input selection algorithms, representation learning, transfer learning and cluster methods) work. This will be used in order to learn concepts by a machine learner. In the third section I will give some background for evaluating clusters which will be taken into account when transforming the clusters to categories. In the fourth section I will elaborate on theories that can be used to evaluate categories. Section five will give background on neural networks. We will use this knowledge throughout the research but in particular for learning representations of photos and recognizing categories. In the sixth and final section, I will tell more about constructing a data set and how one can use active learning doing that. I will use this theory for constructing a data set which I will need to train a network.

2.1 Theory about relevant categories

2.1.1 Concept learning

We want to teach a machine to recognize categories that are deduced from concepts. In order to find a method to do this it is convenient to first learn some more about how humans learn concepts and what those concepts include. We will use this knowledge to set up some guidelines for evaluating what generic categories are.
Bruner defined concept learning as a search for and listing of attributes that can be used to distinguish exemplars and non exemplars of various categories [4]. Three intuitions about concepts wave throughout cognitive neuroscience literature [10]:

Definition 2.1.1. Concepts are mental representations that are used to discriminate between objects, events, relationships, or other states of affairs.

Definition 2.1.2. Concepts are learned inductively from the sparse and noisy data of an uncertain world.

Definition 2.1.3. Many concepts are formed by combining simpler concepts, and the meanings of complex concepts are derived in systematic ways from the meanings of their constituents.

2.1.2 Relevant categories

There are tons of photo categories to come up with, however it is more useful to have a limited set of relevant categories. According to Abhyankar et al. the relevance of categories depend on the entropy and the coverage of the categories in relation to the data [1]. The entropy is a way of expressing the information density. The higher the entropy over a set of categories, the more information these categories give over the data. The coverage of the categories is the percentage of data records that belong to one of the categories. We will use this theory to define guidelines on which we will thereafter evaluate the constructed categories.

2.2 Tools for concept learning

To let a machine learn concepts we will use some tools as elaborated below. In section 2.2.1 I will give background information for selecting photos. In section 2.2.2 I will elaborate on learning representations of data which will be used for encoding photos. In section 2.2.3 I will explain how transfer learning works which we will use in order to be able to apply representation learning. In section 2.2.4 I will elaborate why we will use clustering as basis for the categories.

2.2.1 Input selection algorithms

We will need to provide the learning algorithm data in order to let it learn concepts. This data will be made up of photos. There are a lot of photos from photo collections available which we can not all use because of time and computation costs. Therefore, we will need to find a strategy to choose the data we are going to use as input. We will base our strategy on the method of choosing an input selection algorithm (which chooses which input factors to use). While the problem at hand does not concern variable selection, we will use the knowledge about input selection variable selection as an inspiration for our problem about data selection. According to Fernando et al. four factors should be considered when choosing an appropriate input selection algorithm [8]:

1. The strength between candidate model inputs and outputs.
2. The algorithm should cater for redundancy in candidate model inputs.
3. There should be a stopping criterion that determines when to stop with adding or removing candidate model inputs.
4. The computation efficiency is predominant.

We will mainly use the first two factors as an inspiration. In section 3.2.1 I will further elaborate how we will use this inspiration.

2.2.2 Representation learning

In order to let a machine undergo the process of searching for and listing of attributes that can be used to distinguish exemplars and non exemplars, the machine needs to be able to compare the photos in a relevant manner. One can let the machine compare the pixels but that will not deliver a useful output in sense of concepts. The way that data is represented is important for the success of a learning algorithm in many cases [3]. Therefore we will learn the underlying representation of the objects in the photo and let the learner compare these representations. One can call this process representation learning. Representation learning is concerned with learning representations of the data so that it is easier to extract useful information for building classifiers (or other predictors). When working with probabilistic models the distribution of the underlying explanatory factors is often relevant. Some fields in which representation learning has yielded great success are: speech recognition and signal processing, object recognition and natural language processing.

2.2.3 Transfer learning

Because we do not know the concepts yet we will not be able to find a way to transform the photos into a representation of those concepts. But we can use knowledge gained from learning a similar problem. This is a way of transfer learning. Transfer learning is when one uses the knowledge gained from learning a similar problem when solving the current problem. Deep learning is attractive to utilize when applying transfer learning because it focuses on learning a rather abstract representation of the data, in which that representation needs to be able to differentiate on the variation of the input. I will elaborate more on deep learning in section 2.5. Other fields were transfer learning is used is in data visualization, creating auto-encoders or when denoising auto-encoders [2].

2.2.4 Cluster methods

We will need a way to compare the representations in order to find out which photos have similar and different attributes. For doing this clustering is a suitable approach because it focuses on assigning a number of observations into groups whereby the observations within each group are similar and the observations between groups are dissimilar.

K-means

In this project I will use K-means as clustering method. K-means, which was first defined by MacQueen, is a method whereby a population is partitioned into k sets so that the sets have a low within-class variance [20]. Figure 2.1 illustrates how this algorithm works.

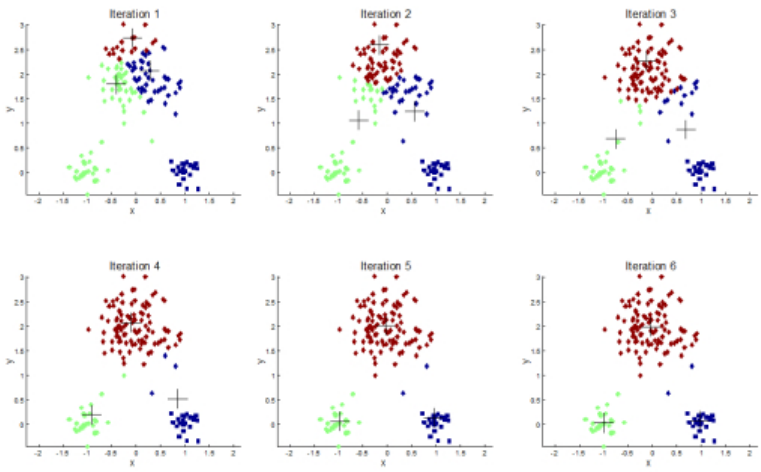

Figure 2.1: Example of K-means algorithm on two dimensional data. The colors illustrate to which cluster a data point belongs. Iteration 1: selecting three random data points and assign all the data points to the closest point. Iteration 2-5: compute the centers of the clusters and assign all the data points to the closest cluster center. Iteration 6: data points stay with the same cluster: converged. Source: `https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap8_basic_cluster_analysis.pdf`.

2.3 Evaluating clusters

Evaluating clusters is known to be a hard task [14]. However, Kovacs et al. listed different ways of measuring the validity of clusters [16] . Recurring aspects of these measurement techniques are measuring the cluster separation and the cluster cohesion. The concepts cluster cohesion and cluster separation are further explained in figure 2.2

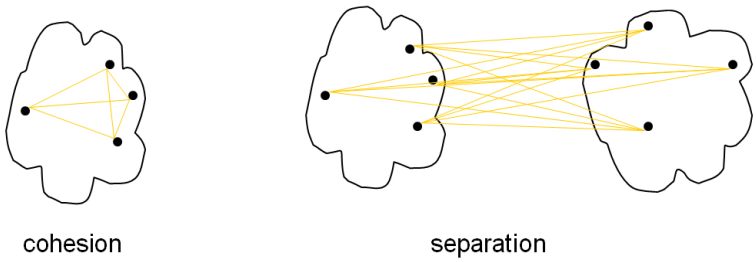

Figure 2.2: Cluster separation and cohesion. Cluster cohesion is a measure based on the distance between points within the clusters and cluster separation is a measures based on the distance between points from one cluster to an other. Source: `https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap8_basic_cluster_analysis.pdf`.

2.4 Category evaluation metrics

2.4.1 Measuring agreement: Fleiss' Kappa

The content of the categories need to be independent of the person who is attributing the photos into categories. This will be evaluated by measuring the agreement about several photos per category among different raters. I will measure this with Fleiss' Kappa, which is a scale to measure agreement among any number of raters, as long as these number of raters is the same for each subject [9]. The Kappa can be calculated as a value over all categories, or for a specific category, this will be further explained in the sections below. We will base the example calculations on the data from table 2.1.

Categories	**Photo 1**	**Photo 2**	**Photo 3**	p_j
Flowers	$n_{11} = 3$	$n_{21} = 0$	$n_{31} = 0$	0.33
Sunset	$n_{12} = 0$	$n_{22} = 0$	$n_{32} = 3$	0.33
Forest	$n_{13} = 0$	$n_{23} = 1$	$n_{33} = 0$	0.11
Mountain	$n_{14} = 0$	$n_{24} = 2$	$n_{34} = 0$	0.22
Total	3	3	3	

Table 2.1: Example of data on which one can calculate Fleiss' Kappa. In this example, $N = 3$ (number of subjects), $k = 4$ (number of categories) and $n = 3$ (number of raters). n_{ij} is the number of raters who assigned the i^{th} subject to the j^{th} category. And p_j is the chance that a photo gets assigned to j^{th} category, i.e. $p_j = \frac{1}{Nn} \sum_{i=1}^{N} n_{ij}$.

Overall agreement

P_i is the proportion of agreeing pairs out of the total possible pairs about the i^{th} subject:

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}(n_{ij} - 1)$$

The overall agreement purely at random would be expected to be:

$$\bar{P}_e = \sum_{j=1}^{k} P_j^2$$

The overall agreement corrected with the agreement at random can then be calculated as follows:

$$\kappa = \frac{(\frac{1}{N} \sum_{i=1}^{N} P_i) - \bar{P}_e}{1 - \bar{P}_e}$$

In the example above the kappa is:

$$\kappa = \frac{(\frac{1}{3}(\frac{1}{6} \times 6 + \frac{1}{6} \times 6 + \frac{1}{6} \times 0 + \frac{1}{6} \times 2)) - \frac{23}{81}}{1 - \frac{23}{81}} = \frac{\frac{63}{81} - \frac{23}{81}}{\frac{58}{81}} = \frac{40}{58} = 0.69$$

Agreement on a particular category

The probability that a second assignment is to the j_{th} category, given that the first assignment was to j_{th} category can be calculated as follows:

$$\bar{P}_j = \frac{\sum_{i=1}^{N} n_{ij}(n_{ij} - 1)}{\sum_{i=1}^{N} n_{ij}(n - 1)}$$

The chance that a subject gets assigned to the j^{th} category is p_j as described above. The agreement on a category beyond chance is then:

$$\kappa_j = \frac{\bar{P}_j - p_j}{1 - p_j}$$

In the example above the kappa for the category Flowers is:

$$\bar{P}_1 = \frac{3 \times 2 + 0 \times -1 + 0 \times -1}{3 \times 2 + 0 \times -1 + 0 \times -1} = \frac{6}{6} = 1$$

$$\kappa_1 = \frac{1 - 0.33}{1 - 0.33} = 1$$

interpretation of k-values

Landis and Koch give an idea on how to interpreted these κ-values [18] as one can see in table 2.2

κ	**Interpretation**
<0	Poor agreement
0.01 - 0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-1.00	Almost perfect agreement

Table 2.2: Interpretation of κ-values.

2.4.2 Confusion

When people do not agree on what the photo category should be, it is interesting to know between which categories the confusion is. Categories can be considered too similar if there is much confusion between categories. One can display the confusion between categories in a confusion matrix as for example figure 2.3.

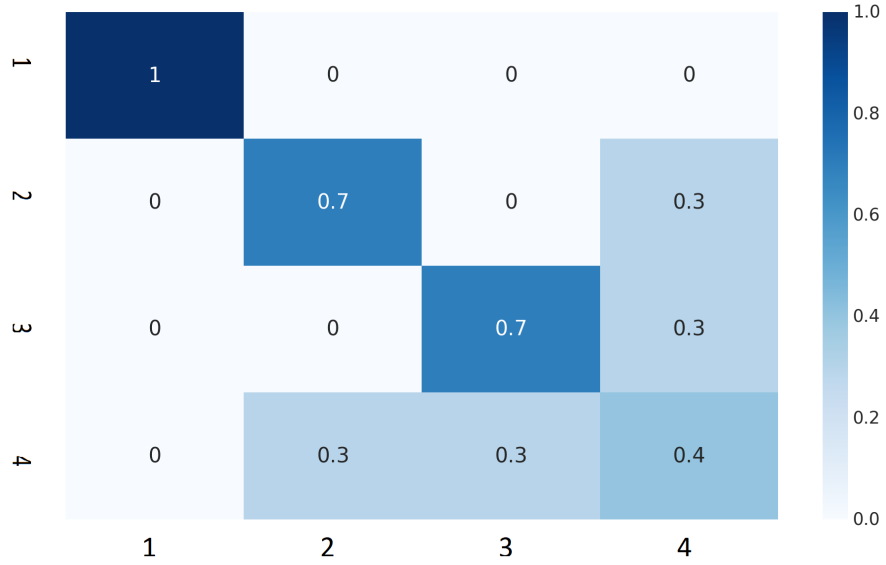

Figure 2.3: Confusion matrix. Every cell of the matrix gives the number of times when one rater assigned a photo to the i^{th} category and an other rater assigned that same photo to the j^{th} category divided by the total of times someone assigned a photo to the j^{th} category.

2.5 Neural networks

In this project I will make use of neural networks to learn the representation of photos and I will use them for recognizing the categories. In specific I will make use of deep residual networks (2.5.2) which are CNNs, convolutional neural networks, (2.5.1) with residual features.
A neural network is a network that serves as a machine learning model. It is a network with a layer of input neurons, one or more hidden layers and a layer of output neurons [24]. Figure 2.4 shows how such a neuron works.

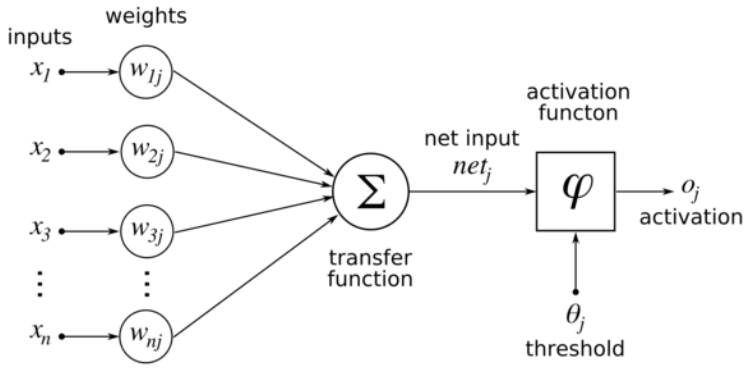

Figure 2.4: A visual representation of a neuron. The output or activation of a neuron can be defined as follow: $o_j = \varphi(\sum_{i=1}^{n} w_{ij}x_i + \theta_j)$. The activation function, $\varphi()$, can for example be the sigmoid function: $f(x) = (1 + e^{-x})^{-1}$. Source:`https://en.wikibooks.org/wiki/Artificial_Neural_Networks/Print_Version`.

Every couple of neurons that are connected is associated with a numeric number called weight. These weights are the trainable component of the model. They are trained by optimizing a loss function with a gradient descent approach. The number of hidden layers, the number of neurons per layer and the activation function of each layer can differ as much as needed. A basic neural network with one hidden layer is represented in figure 2.5.

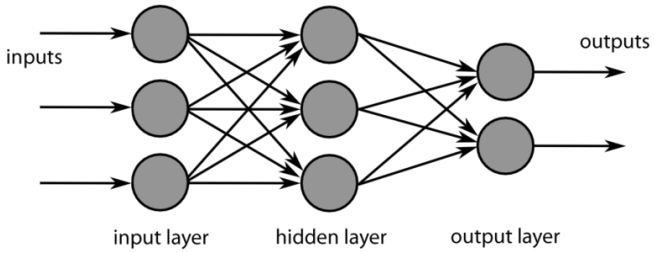

Figure 2.5: A basic neural network. Source:`https://commons.wikimedia.org/wiki/File:MultiLayerNeuralNetworkBigger_english.png`.

2.5.1 Convolutional neural networks

CNNs are often used for image recognition [17]. CNNs are useful for this because they can detect features of varying size and on varying places in photos. In figure 2.6 one can see how such a CNN works [19].

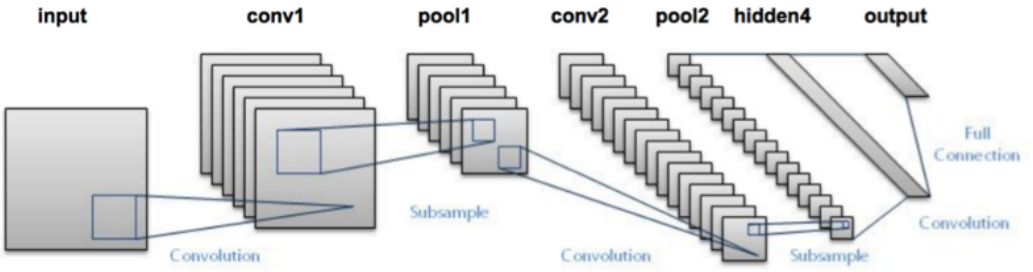

Figure 2.6: A CNN. Typical for CNNs is that they have local receptive fields. These fields can extract elementary visual features such as edges, end-points and corners. These features are then combined in the following layer in order to detect higher-order features. The local receptive fields, which produce the feature maps, perform the same operation on different parts of the images. In that way they can, besides extracting the features, also extract the position of the feature in the input. Sub-sampling is a method to reduce the resolution of the input. One can do this with max (or mean) pooling as described in figure 2.7. Source: `https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/`

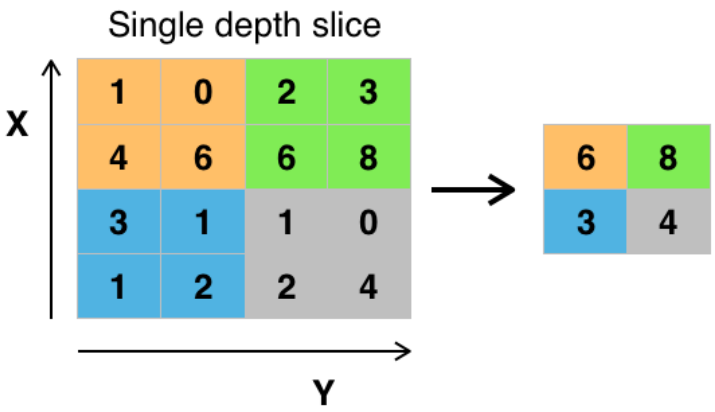

Figure 2.7: Max pooling. For every sub-field, the highest value will represent the sub-field in the output. Mean pooling works in the same way except that it takes the mean of every sub-field instead of the highest value. Source: `https://en.wikipedia.org/wiki/Convolutional_neural_network`.

2.5.2 Residual neural networks

A problem that arises when using very deep CNN's is that when the depth increases the accuracy degrades rapidly. According to He et al., one can

address this problem by adding residual learning to a CNN [12]. How this works is shown in figure 2.8. One can integrate these blocks on several layers of the network.

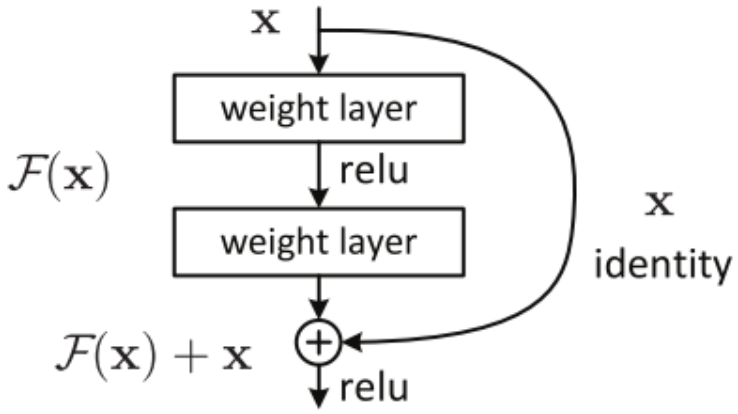

Figure 2.8: Residual learning, a building block. Instead of just passing through $\mathcal{F}(\mathbf{x})$, one passes the result of a relu on $\mathcal{F}(\mathbf{x})$ and x. Source: [12]

2.6 Constructing a data set

We need a data set on which we can train the network, before it can recognize the categories. We will first take a look at how Imagenet, a set of 3.2 million photos divided over 5247 sets, was constructed. According to Deng et al [5], this process consisted of the following steps.

1. Collecting a diverse set of candidate images to represent the selected categories, this was done by searching for the category labels on search engines;

2. Annotation of the collected images to obtain a clean data set, this was done by letting a human decide if the given category matches with the given photo.

2.6.1 Active learning

The key idea of active learning is that a classifier chooses its next pool-queries based on the previous answers. Tong et al. showed that one can use this for image retrieval [22]. We will apply this knowledge in the following way: we will use a neural network for predicting labels of images and so collecting candidate images and after cleaning those images we will train the same network on the clean data set.

Chapter 3

Research

This chapter gives details on the research. In the first section I will draw some guidelines on how to evaluate the performance of the categories. In the second section we will go into detail about the construction of concepts on the basis of clusters of encodings. In the third section I will elaborate on how I transformed these cluster into categories and if how I evaluated if the labels matched with the content they represent. In the fourth section I will show how I carried out the evaluation of the emerged categories and I will show the adjustments I made on the basis of these results and the guidelines. In the fifth and final section, we will go into detail on how I constructed the network and how the data set to train on.

3.1 Guidelines for evaluation of generic categories

This chapter will provide information on how we are going to evaluate the created categories. The main themes are the learnability and the relevance of the categories.

Learnable categories

After creation of the categories, I want to learn a neural network to recognize these categories. About learnable categories we can say the following:

1. In order to be able to learn a network to recognize the categories, they need to be objective and not subjective. This can be tested through measuring human agreement on the categories.

2. In order to promote the process of searching and listing of attributes that distinguishes photos that do and photos that do not belong to the different categories, (aspects of) the categories should not be too similar to each other. This can be tested through measuring confusion among human raters about the categories.

Relevant categories

The intention of the categories is that a category represents a photo in a relevant way in relation to the photo collection. In section 2.1.2 we have concluded that the relevance of the categories is dependent on the coverage and the entropy of the total of categories. Specifically, for the case of photo categories, this amounts to:

1. Adding a category were almost none of the photos of the data set belong will decrease the entropy of the categories as a total. One could argue that this category is not relevant and needs to be removed.

2. Adding a category were disproportionately many photos of the data set belong to a category will decrease the entropy. One could argue that that category is not specific enough and should be divided up in two or more categories.

3.2 Constructing concepts

By the use of a cluster algorithm on photos of photo collections concepts were constructed. First, I selected photos as elaborated in section 3.2.1. Next, these photos were encoded as described in section 3.2.2. Thereafter, concepts were discovered by using the encodings in a cluster algorithm as explained in section 3.2.3.

3.2.1 Photo collection

When looking at the theory in section 2.2.1, one can conclude that it is relevant to consider the following:

1. Adding photos that diverge from the photos already selected will have the most influence on the output clusters and will therefore be valuable input data.

2. (semi-)duplicates, which occur quite often in photo albums, cause redundancy because they do not provide new information.

When selecting photos as input to create categories I will pay attention getting a variety of photos and on avoiding having lots of duplicates in the input. As stated before we can not use all the photos available because of time and computation costs. So we will need a way to select photos. To illustrate the effect of the number of photos per collection and the number of collections on the previous two points, two visualizations with different data selection algorithms are given below.

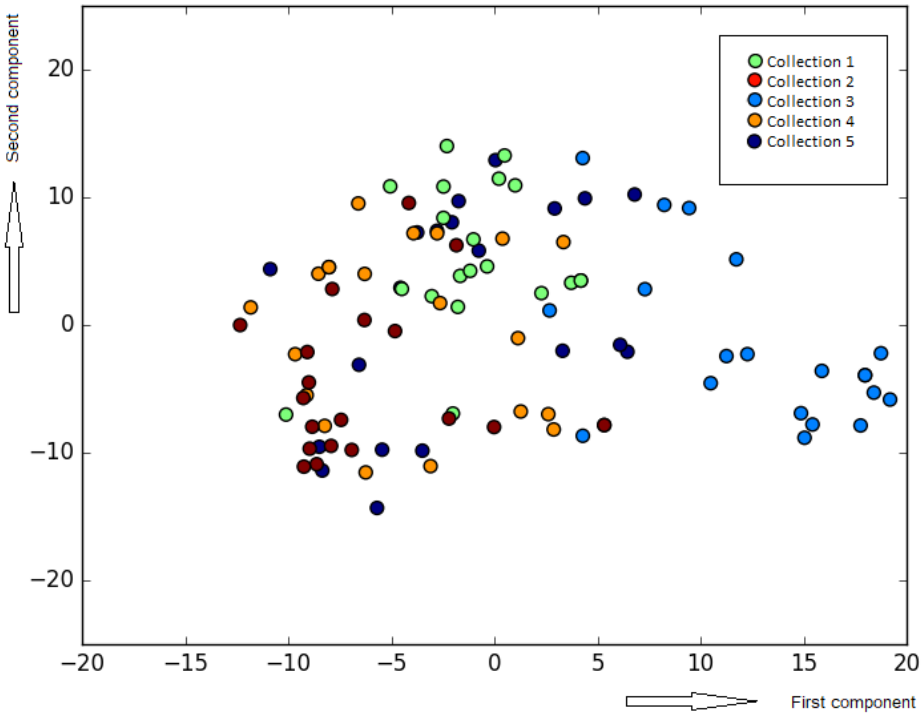

Figure 3.1: Visualization of 5 photo collections with 20 photos per collection (100 photos in total). The photos were first encoded as described in section 3.2.2 and afterwards they were compressed to two components with PCA. As one can see the photos are more concentrated whereas the photos in figure 3.2 are more dispersed.

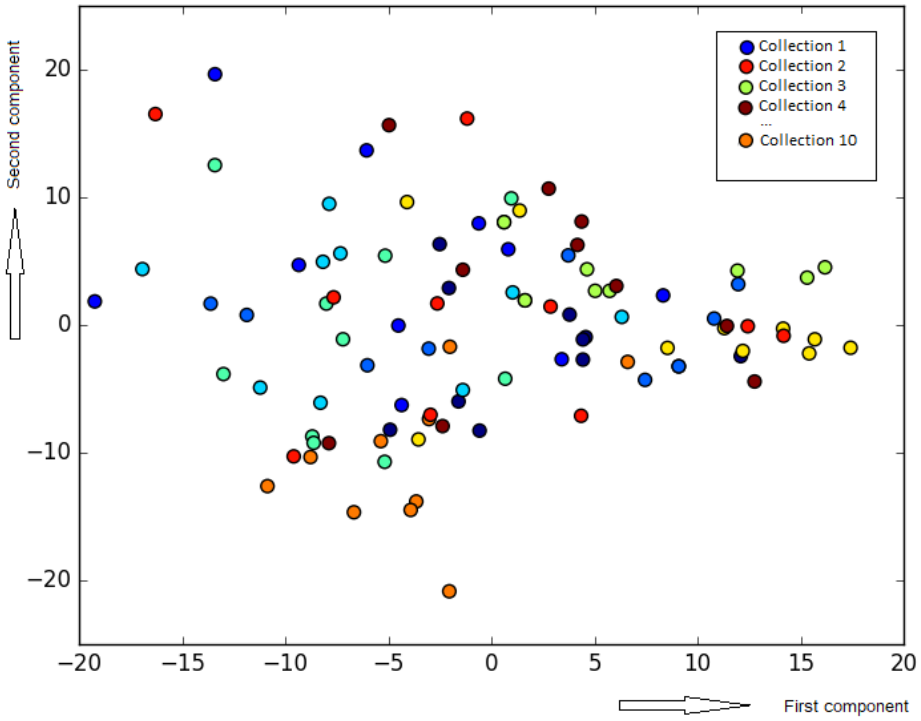

Figure 3.2: Visualization of 10 photo collections with 10 photos per collection (100 photos in total). As well as in figure 3.2, the photos were encoded and thereafter compressed. The photos in this figure show a more diverse representation than the photos in figure 3.2.

Photo selection

Eventually I selected 10 photos per collection at random from 1000 randomly selected photo collections. After the filtering step a set of 8600 photos resulted.

3.2.2 Encodings

Next, I encoded the photos that were selected in section 3.2.1. The photos were encoded by the use of a residual neural network, namely ResNet50 [12]. ResNet50 is trained on 1000 classes, which can be found on the ILSVRC website [5], and is a state-of-the-art network in object recognition (won 1st prize in the ILSVRC 2015 classification competition). By removing the fully connected layer from ResNet50, it gives as output a vector of size 2048. I used this vector as a representation for the objects in the photo. By using the output vector of ResNet50 as an encoding for the photos I applied

representation learning. By using the knowledge gained from learning the 1000 classes form ILSVRC competition in learning the representation of the photos I applied transfer learning.

3.2.3 Photo clustering

The encodings described in section 3.2.2 were used as input for the cluster algorithm. For clustering I used K-means from the scikit package [21].

Number of clusters

I decided to set the target number of categories at 20 to 25. In Hu et al. the number of categories was about halve of the number of clusters. Therefore, it seemed logical to apply K-means with 40 clusters.

3.3 Constructing generic categories

3.3.1 Transforming clusters into categories

To convert clusters into categories, Hu et al. submitted 200 photos divided into clusters to two examiners for assessment. In this experiment the examiners individually analyzed the affinity of the themes within the category and across categories. The examiners had to move a specific photo if another cluster fitted better and they had to merge two clusters if the themes overlapped. They did this individually. Hereafter they needed to resolve their conflicts through a discussion session and give the resulting categories a name. It is not possible to submit 8600 photos for judgment to examiners in the same setting as Hu et al., because examiners can not memorize 8600 photos at the same time. Therefore I did a different experiment as described below.

Experiment

The goal of this experiment was to secure cluster separation and cluster cohesion and also to give the found categories a representative label. First I selected twelve photos per category as further elaborate in figure 3.3. Two examiners were asked to fulfill the following tasks with these photos as input:

1. Remove the cluster pieces without a central theme (secure cluster cohesion);
2. Merge cluster pieces with overlapping theme (secure cluster separation);
3. Classify the 40 random and 40 farthest photos over the emerged categories (in order to represent the whole scale of photos).

4. Give the emerged categories a representative label.

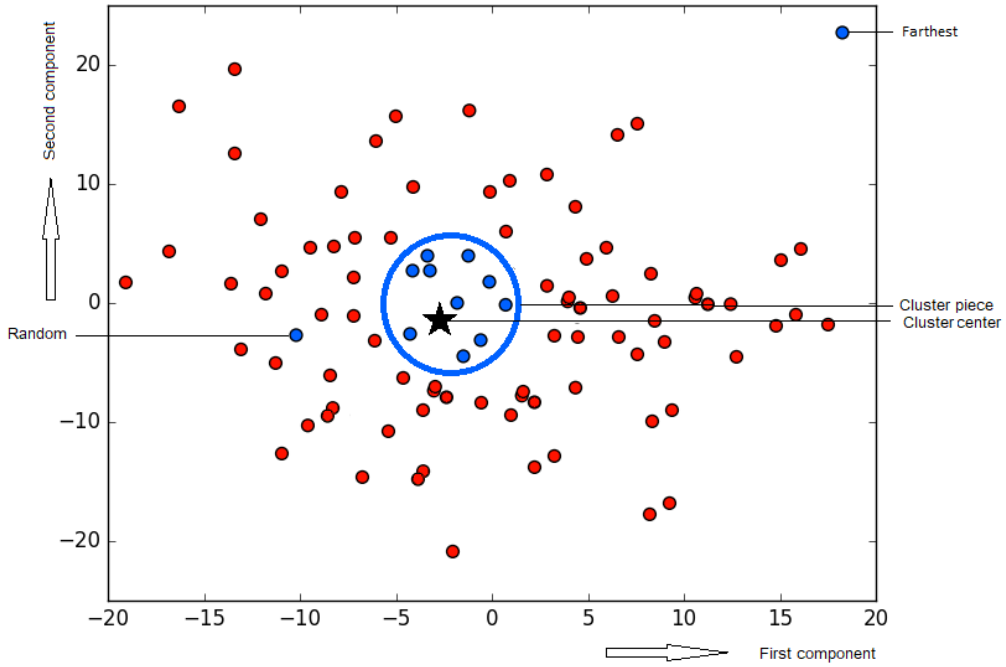

Figure 3.3: Example of photo collection for the set of photos per cluster. The photos were first encoded as described in section 3.2.2 and afterwards they were compressed to two components with PCA. For every cluster the ten photos with encoding closest to the cluster center were selected. In addition, the photo with encoding farthest from the cluster center and a random photo were selected. The ten photos with encoding closest to the cluster center, were seen as one piece and could not be separated, these pieces will be called cluster pieces.

3.3.2 Results experiment

In table 3.1 one can see the adjustments made in the experiment in step 1 and 2. In figure 3.4 one can see the distribution of the random and farthest photos over the emerged categories.

Merged and removed clusters

cluster nr	Adjustment	Reason
0	merged with cluster 8 and 9	themes overlapped
2	merged with cluster 27	themes overlapped
4	merged with cluster 13	themes overlapped
6	merged with cluster 39	themes overlapped
8	merged with cluster 0 and 9	themes overlapped
9	merged with cluster 0 and 8	themes overlapped
10	removed	photos with rotation
13	merged with cluster 4	themes overlapped
15	removed	photos with bad lightning
16	removed	photos with rotation
21	removed	photos with rotation
27	merged with cluster 2	themes overlapped
28	merged with cluster 32	themes overlapped
30	removed	dark photos
32	merged with cluster 28	themes overlapped
39	merged with cluster 6	themes overlapped

Table 3.1: Adjustments experiment (step 1 and 2 from experiment). The merged clusters and the removed clusters are stated with the reason why they were merged or removed.

Distribution

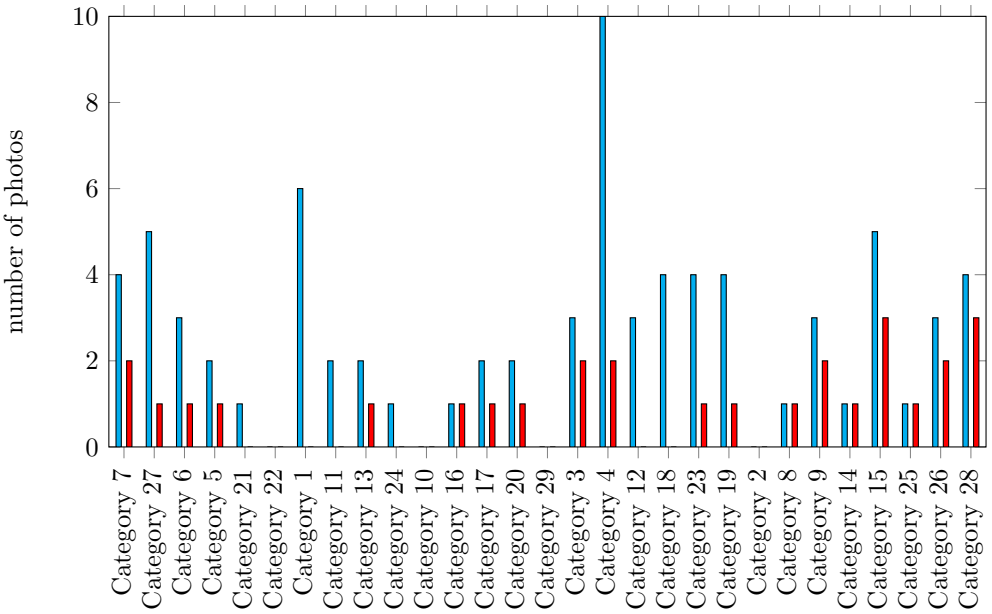

Figure 3.4: Overview in which category the 40 random and 40 farthest photos eventually ended up (step 3 from experiment). Blue: extra photos. Red: extra photos form original cluster.

Data set for visual representation

In order to make an accurate visual representation, I needed to be sure that every photo was assigned to the right category, also the photos from the cluster pieces. Therefore I decided to run over the photos from the cluster pieces and to move a photo to another category but only if it clearly belonged to a different category. Eventually I moved 36 photos from the original category to another.

3.3.3 Label evaluation

The goal of the label evaluation is to check whether the given label matches with the photos in the category. First I will elaborate on the method I used to do this. Thereafter I will give the results of this method. And finally I will discuss these results.

Method

For every category label, five Google images were selected. This was done by typing the category label into Google images[1] and selecting the first five relevant images. Thereafter I ranked the categories for every Google image photo on the basis of the Euclidean distance between the category center and the encoding of the photo.

Results

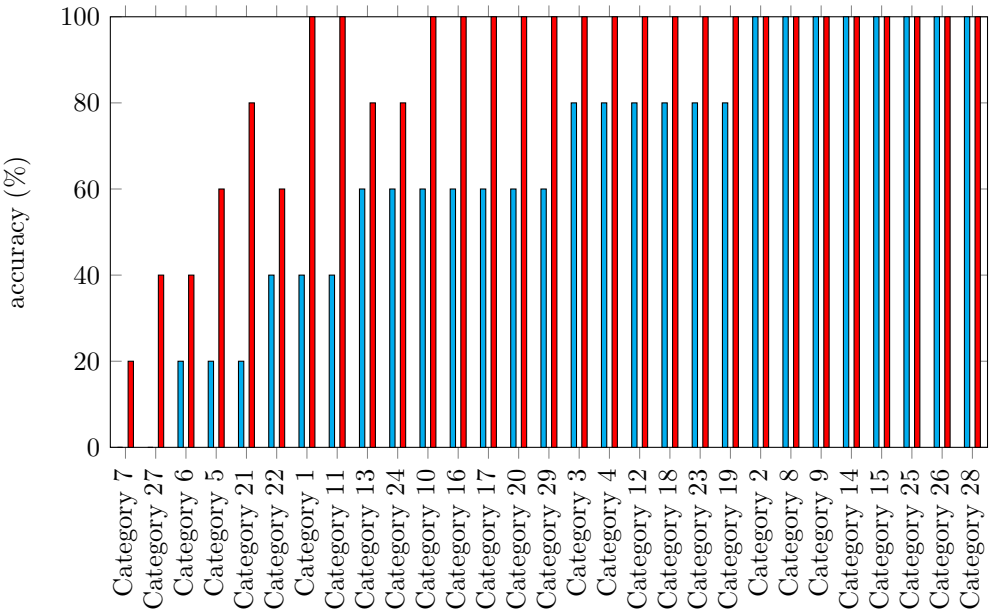

Figure 3.5: Performance of the labels. This figure shows the number of photos that was assigned to the right category according to the Google label defined in terms of top-1 accuracy (blue) and top-5 (red). About 66 percent of photos was assigned directly to the right category. About 89 percent of photos had the correct category in the top 5.

Discussion

For 3 of the 29 categories, namely Category 6, Category 7 and Category 27, it failed to have the correct category in the top 5 closest categories for most of the images of that category. The method I used to rank the categories is a naive approach, and therefore does not always necessarily deliver the correct ranking for the human eye.

[1] `https://images.google.com`. Accessed: 2018-02-01.

3.4 Categories evaluation

3.4.1 Survey

In order to evaluate the distribution, the confusion and the agreement of the categories, I did a survey among 12 people who all categorized 100 photos. Every photo was categorized by 3 different people, so a total of 400 photos were categorized. The participants needed to select the right category, out of the 5 with least Euclidean distance between the category center and the encoding of the photo, or choose to not select a category if they thought it did not belong to one of these categories. I chose to not let them select out of all 27 because then the task would become to complex.

3.4.2 Results

The results of the survey are further elaborated in the following sections.

Distribution

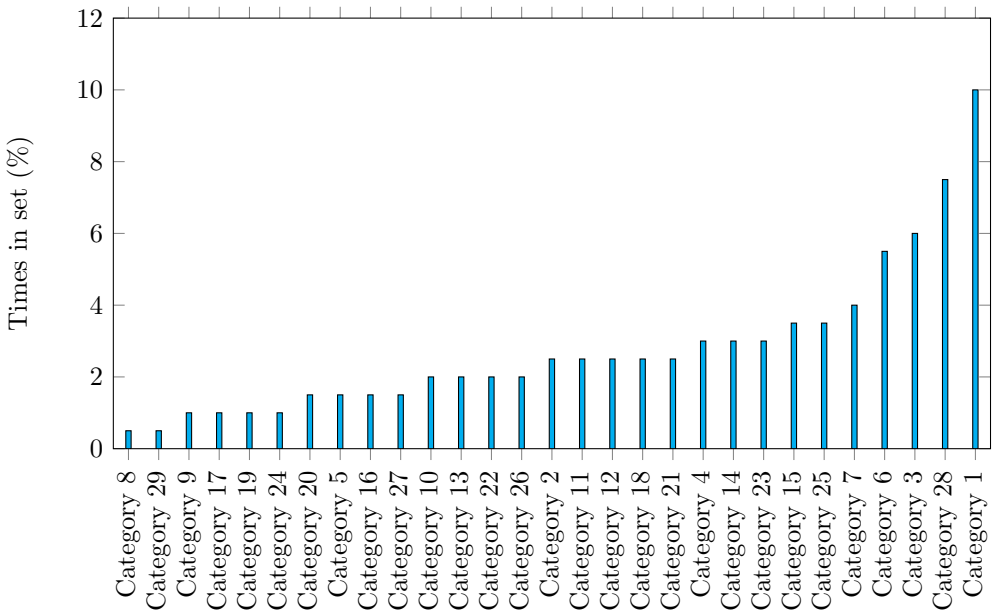

Figure 3.6: Estimation of the distribution on the basis of the survey (results rounded to 0.5%). Also for 13% of the photos the answers was that the photo did not belong to one of the given themes. It is possible that a part of these photos did belong to one of the 27 categories but that that category just was not given as an option in the survey.

Fleiss' Kappa

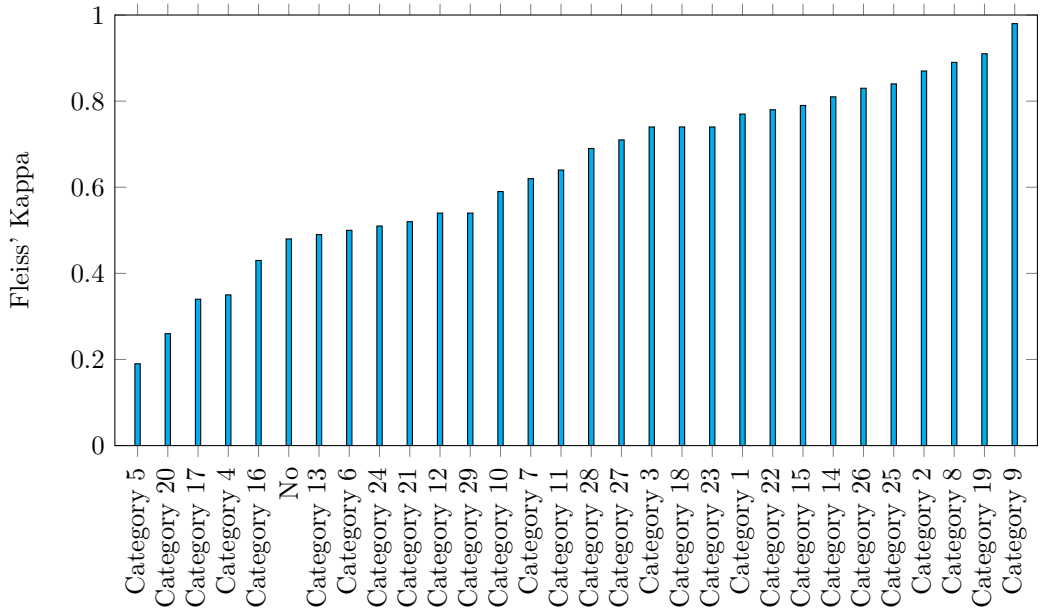

Figure 3.7: Agreement per category with Fleiss' Kappa based on the survey.

Confusion

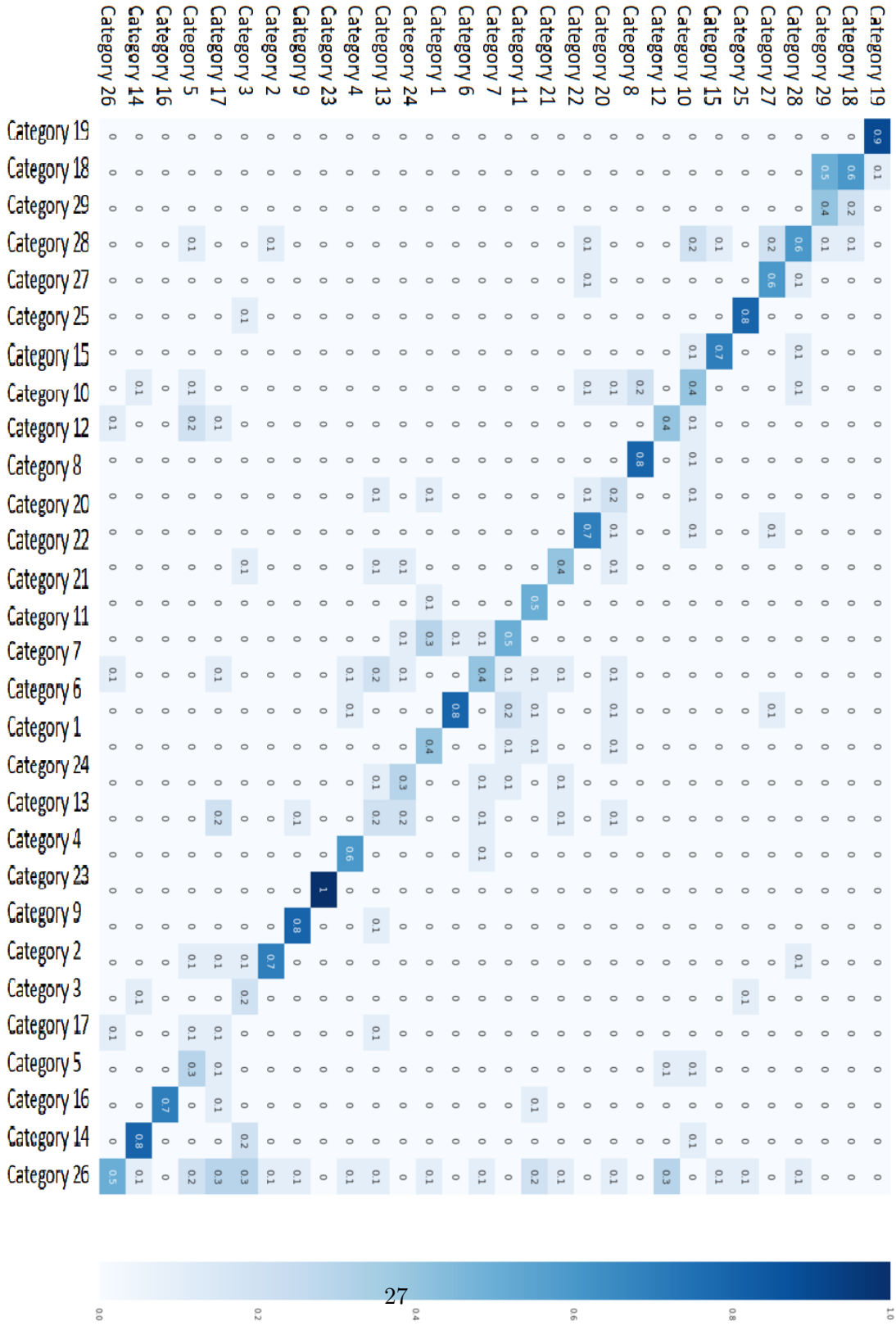

Figure 3.8: Confusion between categories on basis of the survey.

3.5 Recognizing categories

In order to be able to recognize the categories we will need a data set. For doing this we will make use of a method that is similar to the method used for creating Imagenet. Namely collecting a set of candidate images and then cleaning them by human input. Throughout this process we will make use of active learning. The total flow is shown in figure 3.9.

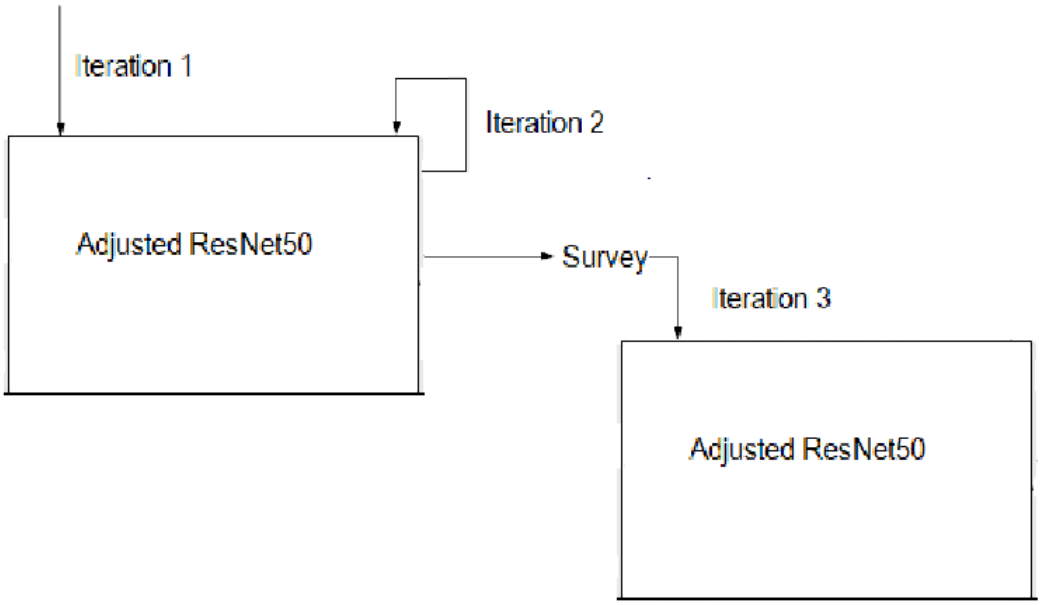

Figure 3.9: Total flow of training a network. Iteration 1: training on 409 photos from the resulting categories (section Data set for visual representation in 3.3.2). Iteration 2: training on 1500 produced by model and cleansed self afterwards (active learning). Iteration 3: training on 3109 photos produced by model and cleansed by survey(active learning).

3.5.1 Collect a set of candidate photos

Method to predict labels of photos (iteration 1)

Because I wanted in particular the photos of photo collections to be in the resulting data set, I needed a way to predict the category of a photo. I did this by training a model on the categorized photos from the resulting categories as described in section Data set for visual representation in 3.3.2. Hereby I used the Resnet50 network, but instead of the fully connected layer with 1000 outputs, I used a fully connect layer with 27 outputs (representing the different categories).

Training the network on more photos (iteration 2)

I decided to add more photos to the training set. Therefore I used an other 1500 photos that were produced by the previous model and were annotated by myself.

Size of the data set

Before starting to make a data set it is convenient to estimate how many photos are needed. The number of photos that is needed per category depends on several aspects as the number of categories, the learner which one wants to teach to recognize the categories, the relation between the categories (one can argue that distinguishing dogs from lions is much harder than distinguishing dogs from buildings, so in the first case more examples are needed of both concepts in order to learn the concept lion). In table 3.2 one can see other data sets and their size.

Name data set	**nr categories**	**nr photos per cat**
Imagenet [5]	>5000	500-1000
Stanford dogs [15]	120	180
city vs. landscape images [23]	2	1128-1588
natural scene [7]	13	151-410
Caltech-256 [11]	256	80-800
PASCAL VOC2012 [6]	20	303-4087

Table 3.2: Other data sets and their size. As it can be seen, the number of categories and the photos per category differ from data set to data set. But also the categories in relation to each other differ much. The Stanford dogs data set is very fine-grained whereas the city vs. landscape images data set is much more abstract.

Collection of the photos

Eventually I decided to collected 250 photos per category. According to the information in section Size of the data set, 250 photos per category should provide a starting point for training and it should be a feasible number of photos to label in the given time. Training on this data will give us more insight in the problem and will give us an idea if collecting data this way works out. If needed, it is possible to collect more data hereafter.

3.5.2 Annotation of the collected photos

Three different raters decided for every of the 250 photos if it had as main theme the given category (Yes-No question). On forehand the participants

received an instruction in which 5 exemplars and 5 non-exemplars were given per category.

3.5.3 Results of the labelling survey

Agreement

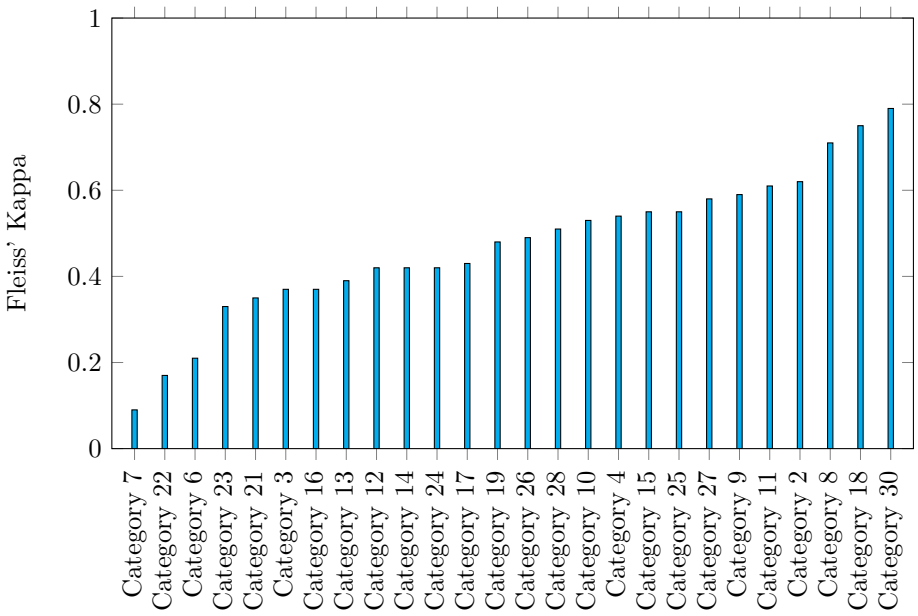

Figure 3.10: Overall agreement per category with Fleiss' Kappa based on the survey. Note that one can not compare these results to the results in figure 3.7. The survey from section 3.4.1 had a different question. This causes a different calculation: overall agreement instead of categorical agreement. In addition the question in this survey is perhaps more difficult because it ask for the main theme instead of if it fits within the theme. What also must be taken into account is that when someone has not understand the assignment right, the kappa for a certain category drops very fast. In appendix 5.1 one can see that the agreement about Category 7 between rater 1 and 2 and between rater 2 and 3 is 0. Probably rater 2 did not understand the task. This would explain the low overall kappa of Category 7.

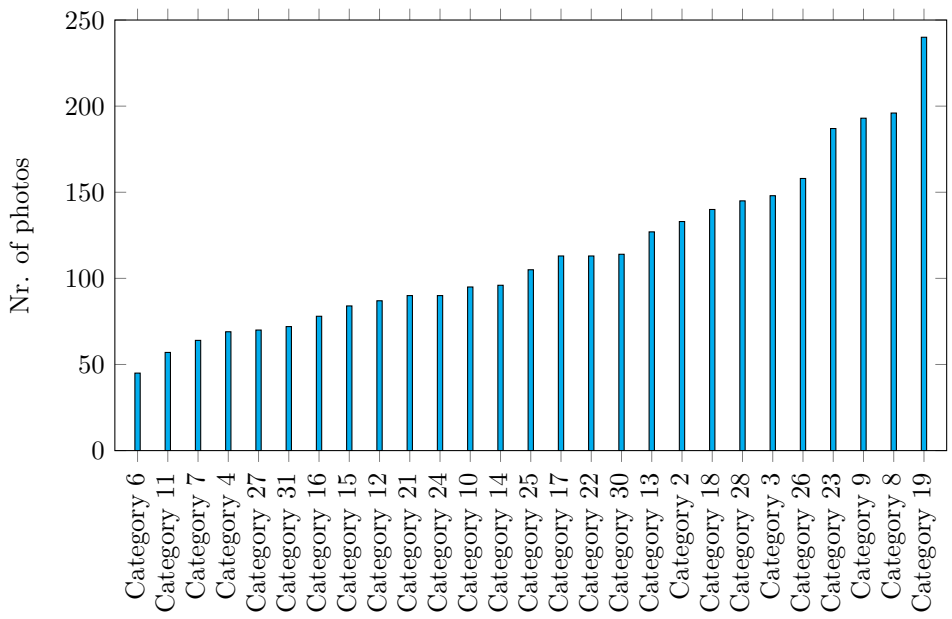

Figure 3.11: Number of photos per category in the resulting data set. I selected the photos were two or more of the raters agreed that it was the main theme of the category.

3.5.4 Training on the data set (iteration 3)

Subsequently, I trained on the resulting data set. Hereby I used the categorized photos from section Data set for visual representation in 3.3.2 to validate on. In the following sections I will elaborate on the performance of the resulting network. In figure 3.12 we can see the overall performance of the network. We will compare the overall performance of the network with the performance of ResNet50 in table 3.3. In figure 3.13 and figure 3.14 one can see the performance per category. In figure 3.15 we can see between which categories the network is confused.

Overall performance

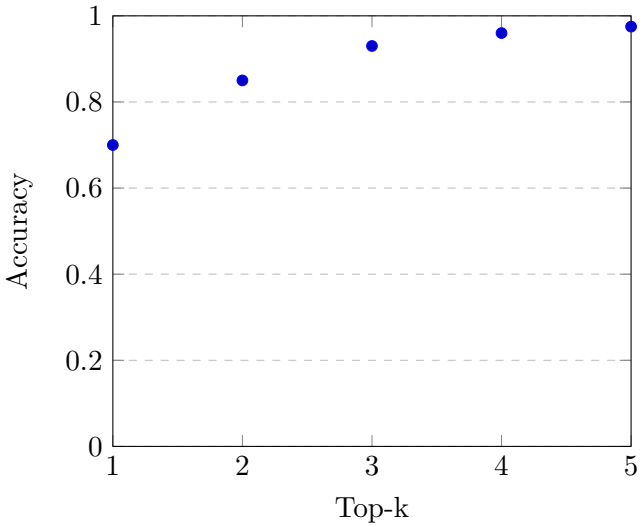

Figure 3.12: Top-1 accuracy till top-5 accuracy. The values lie between 0.70 and 0.975.

Network	**Top-1 error**	**Top-5 error**
ResNet50	22.85%	6.751%
My network	30.0%	2.5%

Table 3.3: Comparison between the performance of my network and ResNet50. My network is trained to distinguish between 27 categories whereas ResNet50 is trained to distinguish 1000 categories. In addition, the categories on their own also differ. More explicit, the categories I used are overall more broad than the categories that were used to train ResNet50 on. All though the problems that the networks are trained on are different, it gives us an indication of what is feasible since ResNet50 is state-of-the art in object recognition (won 1st place in the ILSVRC 2015 classification competition).

Accuracy per category

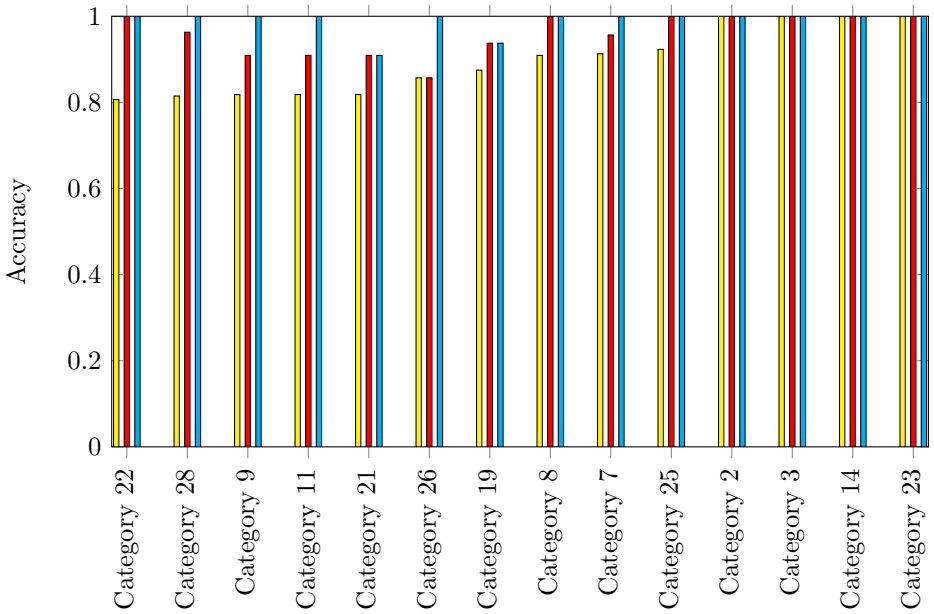

Figure 3.13: Accuracy per category. Blue: Top-1 accuracy. Red: Top-3 accuracy. Yellow: Top-5 accuracy.

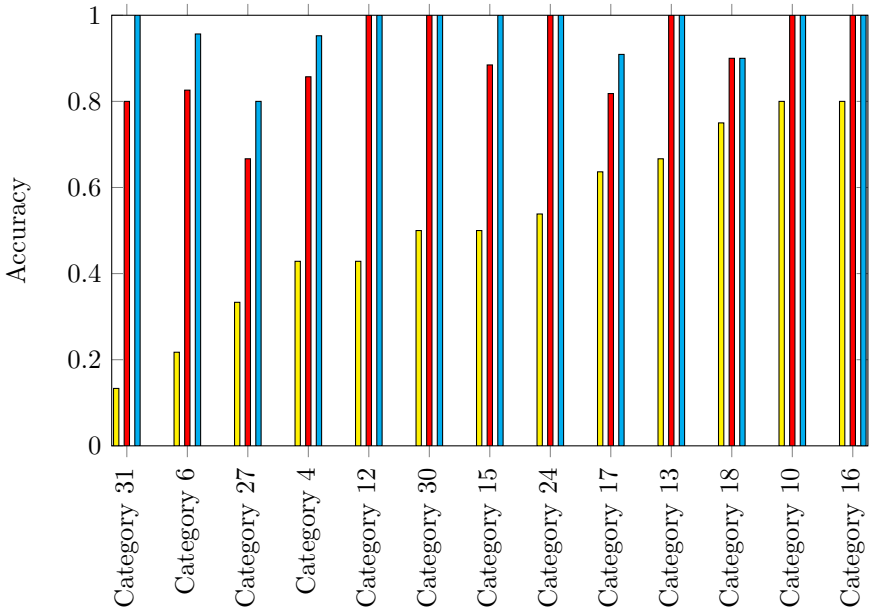

Figure 3.14: Accuracy per category. Blue: Top-1 accuracy. Red: Top-3 accuracy. Yellow: Top-5 accuracy.

Confusion

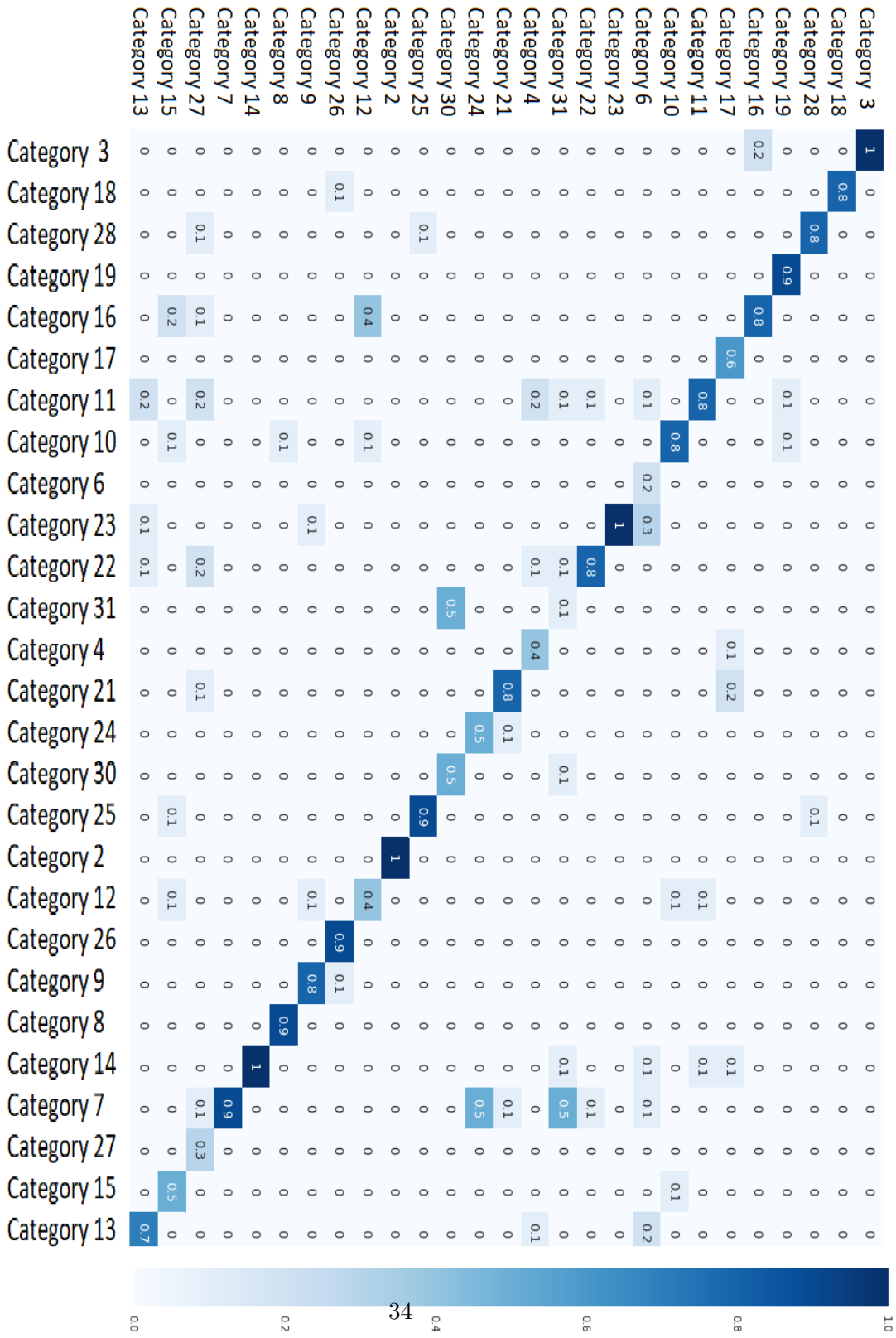

Figure 3.15: Confusion of network between categories. Some of the categories are not confused at all, for example Category 2. Some of the categories are confused a lot with a specific category, for example Category 31 is confused a lot with Category 7. At last there are also categories that are confused a bit with various categories.

Chapter 4

Conclusions

In this work I showed a way to construct generic categories for a specific photo collection and a way to recognize these categories. I leave it to the reader to judge how good the constructed categories are. Looking at earlier work, this work gives some additional thoughts on the following topics:

1. guidelines to evaluate the learnablity and relevance of the categories on. These guidelines say something about the agreement about the categories, confusion between the categories, the distribution of photos over the categories and possibility to categorize (all of) the photos.

2. how to select input data for the cluster algorithm. I focused on selecting a variety of photos and avoiding having lots of duplicates in the input.

3. a way of transforming clusters of photos into categories for a bigger set of photos and a bigger number of target categories. I did an experiment in which I selected a part of the photos. The goal of the experiment was to secure cluster separation, cluster cohesion and to give the emerged categories a representative label.

4. a way to construct a data set for these kind of categories. With active learning I selected candidate photos for every category, these photos were thereafter annotated by human input.

Acknowledgement

I want to thank Luuk for the close guidance during the process and for looking at the work I produced with a critical look. I want to thank Tom for always thing along with me during the process.

Bibliography

[1] Saurabh Abhyankar, Jean-luc Agathos, Virgile Chongvilay, Davor Cubranic, and Julian Lars Gosper. Apparatus and method for assessing relevant categories and measures for use in data analyses, March 1 2011. US Patent 7,899,832.

[2] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36, 2012.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[4] Jerome Bruner. *A study of thinking*. Routledge, 2017.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[6] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 2011.

[7] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[8] TMKG Fernando, HR Maier, and GC Dandy. Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. *Journal of Hydrology*, 367(3-4):165–176, 2009.

[9] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[10] Noah D Goodman, Joshua B Tenenbaum, Jacob Feldman, and Thomas L Griffiths. A rational analysis of rule-based concept learning. *Cognitive science*, 32(1):108–154, 2008.

[11] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Yuheng Hu, Lydia Manikonda, Subbarao Kambhampati, et al. What we instagram: A first analysis of instagram photo content and user types. In *Icwsm*, 2014.

[14] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[15] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2, page 1, 2011.

[16] Ferenc Kovács, Csaba Legány, and Attila Babos. Cluster validity measurement techniques. In *6th International symposium of hungarian researchers on computational intelligence*. Citeseer, 2005.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[18] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[21] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[22] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118. ACM, 2001.

[23] Aditya Vailaya, Anil Jain, and Hong Jiang Zhang. On image classification: City images vs. landscapes. *Pattern Recognition*, 31(12):1921–1935, 1998.

[24] Sun-Chong Wang. Artificial neural network. In *Interdisciplinary computing in java programming*, pages 81–100. Springer, 2003.

Chapter 5

Appendix

5.1 Cohen's Kappa of labbeling survey

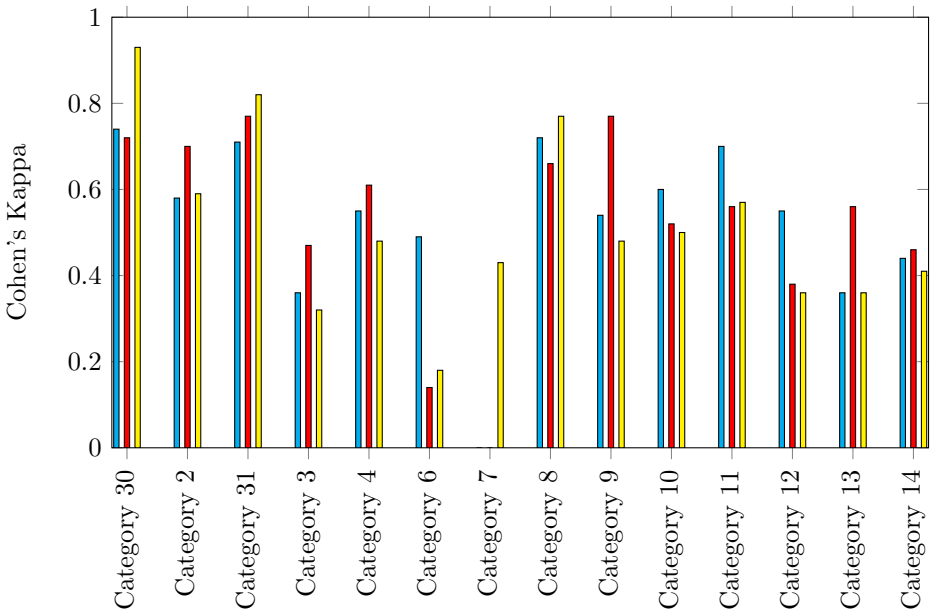

Figure 5.1: Blue: agreement rater 1 and 2. Red: agreement rater 2 and 3. Yellow: agreement rater 1 and 3.

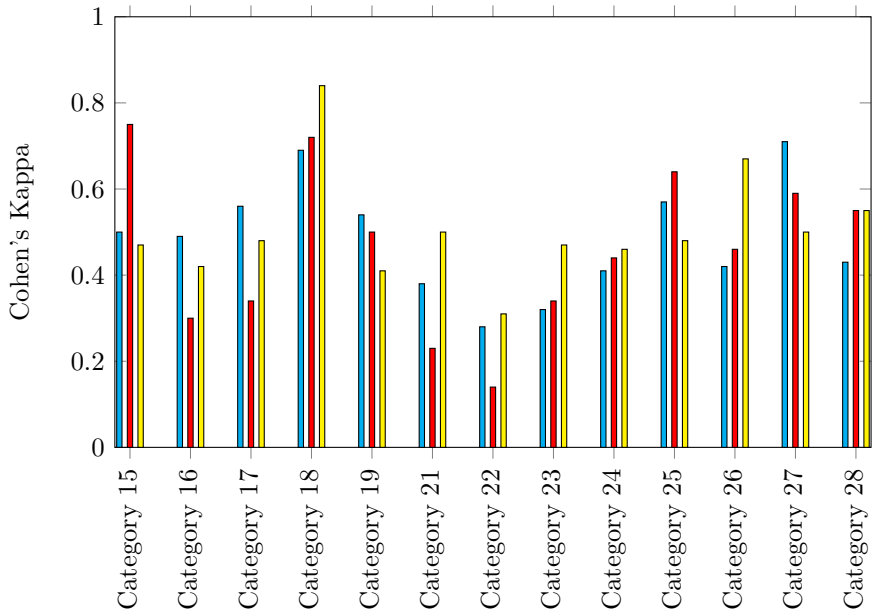

Figure 5.2: Blue: agreement rater 1 and 2. Red: agreement rater 2 and 3. Yellow: agreement rater 1 and 3.