



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

MASTER THESIS

Patrik Veselý

Similarity Models for Content-based Video Retrieval

Department of Software Engineering

Supervisor of the master thesis: Mgr. Ladislav Peška, Ph.D.

Study programme: Computer Science

Study branch: Software and Data Engineering

Prague 2023

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

I am grateful to my loving family and fiancé for their unwavering support through my studies. They always found a way to cheer me up and motivate me to keep going. Also, I would like to thank my supervisor Mgr. Ladislav Peška, Ph.D., for the guidance, valuable lessons, and patience with which he listened to my ideas and helped me to transform them into this thesis.

Title: Similarity Models for Content-based Video Retrieval

Author: Patrik Veselý

Department: Department of Software Engineering

Supervisor: Mgr. Ladislav Peška, Ph.D., Department of Software Engineering

Abstract: Multimedia retrieval is increasingly important with the skyrocketing multimedia volumes produced every day. Therefore many image and video retrieval tools are being developed utilising visual similarity modelling algorithms for similar image retrieval or various visualisations. As such, the quality of the similarity modelling is crucial for these tools. This thesis explores diverse similarity models, their agreement with human perception of similarity and possible improvements of these models. The examined similarity models consisted of colour-based, SIFT-based, and DNN-based models. For the purpose of model evaluation, a user study was conducted to create a dataset of relative image similarity comprising both generic images as well as two compact domains. In this study, the participants were asked to state which of the candidate images was more similar to the query image. The collected data showed the superiority of DNN-based models compared to other evaluated variants. Nonetheless, all similarity models performed significantly better than a random guess. In order to further enhance the performance of the similarity models, we fine-tuned the best-performing model (W2VV++) with the collected dataset and achieved significant improvement in some areas.

Keywords: multimedia retrieval, similarity models, deep learning, user study

Contents

Introduction	3
1 Preliminaries	6
1.1 Colour	6
1.2 Image matching	6
1.3 Neural network	7
2 Related work	9
2.1 Image datasets	11
3 User study	14
3.1 Dataset	14
3.2 Extractors	15
3.2.1 Colour-based extractors	15
3.2.2 SIFT-based extractors	16
3.2.3 DNN-based extractors	17
3.3 Triplet selection	19
3.4 Web application	21
3.4.1 Gamification	23
3.4.2 Implementation	24
4 Dataset analysis	31
4.1 Device types	31
4.2 Demographic attributes	33
4.3 Gamification	34
4.4 Annotations	35
4.5 Triplet decisivness	38
4.6 User agreement	41
4.7 Model agreement	44
5 Similarity model improvment	49
5.1 Preprocessing	49
5.2 Setup	49
5.3 Results	50
Conclusion	53
Bibliography	55
List of Figures	63
List of Tables	65
A Attachments	66
A.1 Feature extractor - User documentation	66
A.1.1 Manual	66
A.1.2 Docker pipeline	68

A.2	Image similarity app - User documentation	68
A.2.1	Manual	68
A.2.2	Docker	69
A.3	Dataset evaluation and fine-tuning - User documentation	69
A.3.1	Requirements	70
A.3.2	Preprocessing	70
A.3.3	Evaluation	70
A.3.4	Fine-tuning	71

Introduction

We live in a time when everybody with a phone can produce tons of multimedia content daily. With the spread of the Internet of Things (IoT), the production of information and multimedia is accelerated. The overall trend is clear – the size of multimedia created grows every day with astronomical speed, and we need powerful tools to work with that. The automatic processing and distribution are already done in publicly and commercially available software such as Youtube and Vimeo. On top of that, they usually implement some recommendation system [9] to ease the use of their software and sometimes omit the need to search for something. However, the need to search in multimedia, especially videos, persists and is not fully solved. Remember how many times you had searched for a specific video or photo you wanted to show someone and weren't able to find it? It could be a video from some streaming platform, social media platform, or even from your phone.

The information overload ¹, particularly multimedia overload, is apparent, and there is active research to cope with that. There are regular competitions to compare state-of-the-art tools such as TREC Video Retrieval Evaluation (TRECVID [4]), Video Browser Showdown (VBS [26]), and Lifelog Search Challenge (LSC [22]). One of the competitions in multimedia retrieval is TRECVID, which focuses mainly on ad-hoc video search (AVS), video-to-text, disaster scene description and indexing over the past years. The AVS task evaluation is done automatically with predefined queries and ground truth. In contrast, the competition VBS focuses on multimedia retrieval but with the user aspect. There are three main tasks: textual known item search (TKIS), visual known item search (VKIS) and AVS. The goal of the known item search (KIS) tasks is to find a single shot from a video from the dataset. In contrast to the AVS, only one correct answer exists in the KIS. Both these competitions for the video search use the V3C1 dataset [6]. The next competition uses quite a different dataset. It is LSC, and the participants search for pictures in lifelog image collection. This data is collected using lifelog device worn by a person and captures a photo every 30 seconds.

The retrieval tools from the interactive competitions commonly utilise more retrieval methods and try to make the best use of their combinations. The combinations of the methods differ among tools but usually share at least some types of retrieval methods. Most utilised types are based on text-to-image similarities, image-to-image similarities, additional metadata (e.g. time of a day, optical character recognition (OCR), successions of frames in a video), and combinations of them.

One of the most common retrieval methods is using text search. In the past years, text-to-video retrieval was more successful than the previously used automatic tag annotation. One of the text and image joint embedding models was W2VV++ [40] employed in SOMHunter [74]. Recently, a new CLIP model was used with superior performance to the W2VV++, e.g., in VIRET [55] on VBS 2021.

Another retrieval methods use image-to-image similarity models. These similarities are computed from embeddings in latent space. These embeddings can

¹<https://dictionary.cambridge.org/dictionary/english/information-overload>

be handcrafted low-level features as used in HTW [27] on VBS in 2021 or represent high-level concepts from some deep neural networks (DNNs). High-level features can be obtained from the last pooling layer of a deep convolution neural network pre-trained on a classification task. These features were employed in VERGE [3]. SOMHunter used high-level features from W2VV++, which were the embedding vectors from the image. Computing similarity among images gives the system ability to reflect relations beyond pixel-wise comparison. The image similarity can be used in a simple k nearest neighbours (k-NN) algorithm, where the user displays the most similar images from the dataset. This querying algorithm is a powerful exploitation technique but lacks exploration. Therefore some systems employed machine learning algorithms to work with relevance feedback and tradeoff exploitation vs exploration. For instance, Exquisitor [31] utilises Linear Support Vector Machines, and the user can provide positive and negative feedback. SOMHunter uses Bayesian-like relevance feedback where the user only provides positive feedback; everything seen and unasserted is assumed to be soft negative feedback.

Image similarity can be used not only for querying but for different exploring visualisations. One of these visualisations was employed in HTW and used self-organising maps (SOM) for visualising the whole collection or only the top results in hierarchical SOM. Another system SOMHunter used SOM fitted on the probability relevance distribution and provided an overview of the top-scored images.

The efficiency of the above-described image-to-image similarities relies on efficient similarity models. However, it is often not clear which particular model to choose in what situation. In this thesis, we will closely investigate the current state-of-the-art (SOTA) possibilities for automatic image-to-image similarity prediction and try to improve these results. To achieve these goals, an image similarity dataset will be created. This dataset will consist of triplets representing a query image and two candidate images. Then the user study’s participants will be asked to select a more similar candidate to the query image. The collected dataset will provide the base for model comparison and an approximation of image-to-image similarity prediction upper bound (w.r.t. human-human consistency). Additionally, the impact of motivational elements, i.e. gamification, will be explored. The dataset will provide not only a benchmark but also training data for additional fine-tuning and improvement of the state-of-the-art similarity models.

The key contributions are:

- An easy-to-use feature extraction framework will be created.
- A web application for image similarity annotations will be created.
- A dataset with human judgments will be collected and made available.
- Vast variety of image similarity models will be compared w.r.t. their consistency with human similarity judgments.
- A new SOTA model will be created.
- The impact of gamification principles on participants’ motivation will be evaluated.

The remainder of the thesis is organised as follows. In the first chapter, a brief refresh of the preliminaries for this thesis is described. The preliminaries contain an introduction to colour science, a brief description of computer vision techniques used in this thesis, and an introduction to neural networks. The following chapter summarises related work to this thesis and in which ways this thesis is novel. The third chapter describes the whole user study, beginning with the video dataset, continuing with image feature extractors, and finally describing the triplets selection and annotation. The next chapter analyses the results of the user study and answers the questions about what image similarity models agree the most with human annotators. Additionally, it explores the impact of gamification and connections among demographic groups. The final chapter describes how the top-performing model can be improved with the help of our dataset.

1. Preliminaries

In this thesis, three categories of preliminaries will be revised for ease of understanding in the following chapters. These sections will summarise the background of the image similarity models which were developed over the years. The first section describes the colour, their modelling and phenomena. The colour similarity models were one of the oldest and the simplest ones. The second section describes selected feature extraction and matching algorithms. The image matching is not the same as the image similarity modelling. However, their outputs can be transformed to achieve similarity modelling. The last section describes neural networks and their applications. Similarity modelling with the neural networks is another possible option, and thus it will be examined in this thesis.

1.1 Colour

The colour can be defined in many ways [54]. In physics, it is a specific electromagnetic radiation within a visible spectrum. In more general terms, it is an attribute of an object. People perceive colour through their eyes. Moreover, the human brain can process and think about the colour of all visible things. Therefore colour reproduction is a well-known task in human history.

One of the early colour models in modern history was proposed by Smith and Guild [67]. These models, CIERGB and CIEXYZ, are additive colour models consisting of three additive units. The RGB colour model is widely used in most digital devices (e.g. PC, laptops, smartphones) to store and reproduce images. This model's units are red, green, and blue; thus, the colour values are easily interpretable. However, the downside of both colour models is their perceived non-uniformity [50]. MacAdam [50] conducted twenty-five thousand colour-matching trials with their chromaticity discrimination apparatus to show that the difference of the colours does not linearly correspond to the perception of an average human observer.

The perceptual non-uniformity had been an issue that CIELAB colour space tried to cope with [52]. This colour space transformed the previous CIEXYZ model nonlinearly, scoring the highest correlation between Euclidean distance and psychological values.

1.2 Image matching

Image matching is a task where a query image is processed, and images of the same object or scene are retrieved. The image-matching algorithms can usually cope with some noise, 3D rotation, or linear transformation, e.g. image scaling, brightness and contrast change, and shear. One of the widely used image-matching algorithms is Scale Invariant Feature Transform (SIFT) [47].

The SIFT are invariant to scaling and rotation and partially invariant to the brightness and view changes. The process of feature extraction can be divided into five steps. The first step is to find significant points in the image. The author suggests computing multiple magnitudes of the Gaussian blur of the image and

then computing the differences between each two closest magnitudes (Difference of Gaussians). Then the same process is done on the different sizes of the images, i.e. scaling up and down. Lastly, local extrema on the Difference of Gaussians and including the other sizes are chosen. The author discusses that these points are adequate candidates for scale and rotation invariant keypoints. The second step is to filter these keypoints. The first filter approximates the keypoint stability, and those under a threshold are discarded. The second filter removes keypoints on the edges because noise heavily affects them. The third step is to assign an orientation of the keypoint. The region around the keypoint determines the orientation and magnitude. In the next step, a feature vector is computed for each keypoint. This feature vector is determined by the region around the keypoint and its subregions' magnitudes and orientations. The author suggests a setting that results in a 128-dimensional feature vector. The last step is the matching procedure. The query image keypoints are extracted and matched with the keypoints from the training set with the nearest neighbour search. Then the pairings are validated, and those with a high probability of false positives are discarded. A clustering algorithm is then applied to find clusters of keypoint matches which are additionally validated with a geometric verification procedure.

Even though the SIFT can find near duplicates, it has two downsides. The first downside is more general. It is its computational complexity. As the author mentions, the usual photo of size 500x500 yields around 2000 keypoints. This can become quickly computationally infeasible. Secondly, visually similar images could be rejected by the feature and geometric validations and thus yielding a negative result. Therefore vector of locally aggregated descriptors (VLAD) was proposed by Jégou et al. [30]. This representation creates a single feature vector for each image from a variable length of feature vectors. Thus it can be used on the SIFT keypoints to produce a single feature vector. At first, it creates a visual dictionary similar to the bag of features (BOF). Then each feature vector is mapped to a visual word from the dictionary, and a difference between them is computed. The differences are summed, resulting in a 128-dimensional vector. A "power-law normalization" was proposed by Delhumeau et al. [11] to improve the original feature vector. Finally, the vector is L2 normalized. The resulting vector is a compact image representation.

1.3 Neural network

Neural networks are a family of algorithms for learning and storing that knowledge from the real world. They have been motivated by higher organisms which can perceive the world through their sensory systems and process the stimuli. The processed stimuli can influence their behaviour and be remembered for later use. One of the first learning algorithms was Perceptron [62]. This algorithm iteratively finds a hyperplane that separates data points into two groups based on a target attribute. However, this algorithm will converge only with linearly separable data. Moreover, this algorithm can only produce predictions without any additional information about probability.

Another breakthrough was achieved by Rumelhart et al. [64]. They introduced a backpropagation algorithm that iteratively updates the weights of a neural network accordingly to decrease a defined loss function on the training data. The

training data consists of input values and expected output values. This algorithm propagates the input data through the network and receives the predicted output. Then the predictions are compared to the expected outputs, and the error value is computed. The update process goes backwards through the whole network, and the weights are updated to minimize the error value. The advantage of this algorithm is that any feedforward network with semilinear units can be learnt.

One of the earliest applications of backpropagation was made on handwritten digit recognition by LeCun et al. [38] in 1989. The proposed network contained three hidden layers. The first two layers consisted of kernels with learnable weights called feature maps. These feature maps are similar to the convolutional filters in the more recent deep convolution neural networks like ResNet [24], and EfficientNet [70]. These feature maps aimed to learn local patterns and decrease the number of trainable parameters. The last hidden layer is a fully connected one. More interestingly, with less than ten thousand trainable parameters, the researchers achieved a 0.14% error rate on the test set.

2. Related work

The image-to-image similarity plays one of the prominent roles in content-based image retrieval (CBIR), and many state-of-the-art systems use representations from deep neural networks trained on image classification or text-to-image retrieval [74, 44, 28, 68]. These image features can be useful; however, they could perform better as suggested in previous studies [57, 61, 25, 73].

In the first study conducted by Peterson et al. [57], they collected images from six categories and then asked participants to assess similarity on a scale from 0 to 10. Each pair was annotated by 10 participants. With created dataset, they improved the quality of the image features in every domain. However, the improvement did not generalize inter-domain well. Moreover, the absolute scale can introduce unwanted biases, where the users can unintentionally use different neutral points or can introduce assimilation and contrast effects [2, 81].

Assimilation, contrast and assimilation-contrast are well-known theories in psychology [2]. Initially, these theories were observed on consumer satisfaction with the product based on their prior beliefs and experience. Zhang et al. [81] showed that the same theories could be applied to the user ratings from the Internet for product domains such as movies, books, electronics, and clothes. The assimilation theory states that satisfaction with the next product tends to be more similar to the previous one. The opposite direction is the contrast theory. If the quality of the previous product was worse than expected, then the consumer will more likely be more satisfied with the next product. It also applies the other way around. If the previous product quality is better than expected, satisfaction with the next one will be lower. Lastly, the assimilation-contrast theory combines both of the previous theories. If the difference between the quality of the successive products is similar, the assimilation theory occurs. However, if the difference is more extensive than some threshold, then the contrast effect applies. Even though in the image-to-image similarity studies, the participants did not buy any products, the users' satisfaction with the image similarities may influence the subsequent annotations.

In the following study, Roads and Love [61] used the widely popular dataset ImageNet [65] and collected human similarity judgments with 8-rank-2 trials. In the trials, nine images were shown to the user. They were aligned in a grid; the middle image was the query, and the other eight images were to be assessed by the user. The user had to choose two images, the most similar and the second most similar. This technique produced more combinations for triplet creation and triplet loss [17] calculation. However, the user was challenged to assess the similarity of nine images and the sessions were designed to complete 50 trials in 10 minutes. This is a challenging task and thus can induce some noise in the judgments. Some of the noise can even come from human biases, such as contextual or positional bias [29]. The used dataset ImageNet is widely used and contains a lot of diverse images across many classes. However, nearly every image dataset for image classification tasks (incl. ImageNet) has specific types of images where a single object dominates the image. Thus assessing the similarity of images on these datasets is close to one-shot learning [75] and does not have to generalize well on completely general images and/or frames from videos.

Křenková et al. [34] in a recent study tried to cope with similarity modelling with the Distance density model [37]. They collected human judgments with a triplet schema, where a web application presented a query image in the first row and two options in the second row. Then the user was asked to express the similarity score between those two options in the range of 0-100. Profiset dataset [7] was used as the underlying image dataset. These images are photos from a stock image bank and provide only some basic metadata, e.g., title and keywords. The dynamic metric allowed researchers to capture both similarity judgments and user disagreements. They found that on some triplets, the standard deviation was low, and such triplets had consistent judgments. On the other hand, some triplets had high standard deviations, and thus, the users did not agree with each other. However, the scale system lacks more meaningful interpretability, where only the middle, far left, and far most points are interpretable, and the rest of the scale is purely subjective to the user. Besides this dataset being a class-less one, the stock images usually share similar characteristics with ones from ImageNet, e.g. single dominant object. Thus it may also not generalize well on more general images and video frames.

In Rossetto [63] thesis, they designed a user similarity judgment study with an absolute scale with four options, i.e. not similar, slightly similar, very similar, and nearly identical. Two images were presented to the user, and the user was asked to choose from the scale and thus give feedback on the pair similarity. The presented images were collected from three different public image sources to achieve a more diverse set of images. The author examined many low-level feature extractors and (dis)similarity functions. The results showed that the fusion of the image features achieved significantly better results than the individual features. One of the two images was so-called a reference image, and the author chose to use only 14 of those. Therefore there is a considerable risk of some overfitting on these reference images and lack of generalization. Furthermore, the absolute scale has some drawbacks. One of the drawbacks is that the user is asked to assess similarity without any context and can suffer from assimilation and contrast effects. The next drawback is inconsistent image pair generation and the scale labels, where according to the author, more than half of the similarity judgements were "Not similar" options. In addition, 91% of the pairs had been assessed with the option "Not similar" or "Slightly similar", leaving us with only 9% of the pairs with more significant similarity, e.i. "Very similar" and "Nearly identical".

Preceding this thesis, we conducted a preliminary study Veselý and Peška [73] in a similar setting to this thesis. The study aimed at a more general similarity among video frames; thus, we used the V3C1 dataset [6]. The dataset contains many trivial frames, e.g. single-colour or entirely blurred frames, which have been manually excluded. A total count of 38 participants annotated 4394 triplets. The triplet consisted of a query image and two option images from which participants were asked to choose the more similar image to the query image. These triplets were generated to cover a wide range of similarity possibilities, i.e. both options are highly similar, only one option is highly similar, and none of the options is highly similar. The authors compared a wide range of image features extraction models, including SOTA deep neural networks and pre-deep learning era colour and SIFT-based extractors. Quite surprisingly, smaller models of the same architectures performed better in general. However, the preliminary study had two



Figure 2.1: Selected images from ImageNet on the left and V3C1 on the right.

main limitations. The first limitation was the very limited options when the participants were forced to choose from either of two images, even in the cases where they were completely unsure. This could lead to noise in the data and unnecessary pressure on the participants. The second limitation is related to the first one, that the participants could not distinguish the annotation certainty.

2.1 Image datasets

As it was discussed before, dataset selection is an important part of the whole process of understanding image similarity as humans do. There is plenty of image datasets, some with annotation, i.e. classes or object position, and some with only weak annotation. This weak annotation is usually some metadata, e.g. title of the image or keywords, and does not have to be accurate.

One of the most popular go-to datasets in computer vision and machine learning is the ImageNet dataset. ImageNet contains 14,197,122 images annotated with 21,841 synsets. There is always exactly one label for each image. The popularity of the dataset and wide use is not directly explainable, but there are some clues which are widely accepted. The first huge advantage is its size. Dataset size is essential for deep neural networks because the larger the train set size is, the better performance the models have [49]. The number of classes is enormous compared to other image classification datasets, which is probably why learnt models on this dataset generalize well [33]. However, this dataset is not suitable for general image similarity for more reasons. Generally, image classification datasets lack images with no clear dominant object. Example images from the ImageNet are depicted in figure 2.1 on the left. It is understandable that without any dominant object, there cannot be an assigned class, and thus, it is not suitable for the classification task. Moreover, the dominant object is usually nicely aligned in the centre, and consequently, these images do not represent well enough frames from videos and any image in general. The dominant object is not the only issue with the object classification datasets. Another issue is with the final and limited number of classes. Some other image classification datasets (e.g. CIFAR-10 [35], CIFAR-100 [35], CALTECH-101 [39], etc.) contain only tens or small hundreds of classes. Even ImageNet contains only 21,841 synsets which is a small part of the whole WordNet [53] with 117,000 synsets. The last and the most crucial

reason is that the performance in image classification does not generalize well for image similarity, as shown in Roads and Love [61].

The issue with predefined and fixed classes in image classification datasets can be overcome with more general class-less datasets such as Flickr30k [80], Profiset [7], and GPR1200 [66]. The last one is not a true class-less dataset and contains 1200 classes. However, it is not a traditional image classification dataset because the number of classes is disproportionately larger than the number of images per class. Especially there are only 10 examples per class, and the dataset was created from six different datasets to achieve generalization and uniformity per class. The Flickr30k dataset contains 30 thousand images from Flickr ¹ and 150 thousand descriptive captions. The Profiset dataset was already previously discussed, and it contains 20 million images from stock image bank Profimedia ². Each image is annotated by a title and 20 keywords on average. All of these datasets avoid the class issue with diverse images which were not collected from predefined classes. Nonetheless, the main objectness of the image remains, and thus, it does not represent a diverse sample of a frame in a video.

Achieving desired generality and robustness is challenging. Hence, the video dataset comes into play. A video, in simplification, is just a sequence of images and an audio track. When we disregard the audio track and focus on the image set, we get an enormous amount of images. These images in the sequence are nearly duplicates of their neighbours. Therefore, we can drop many of these duplicates, and what we get is an image dataset that represents both nicely aligned images with a main object and images with no clear main object or meaning. The second type of image is important because even on these images, the similarity can be assessed by humans, although they are not usually represented in the classical image datasets.

There are many video datasets to choose from. One of the datasets is called ActivityNet [16], which is a benchmark for activity recognition. This dataset contains 648 video hours and annotations for 200 activities. The next dataset is YouTube-8M [1], which is significantly larger with its 350,000 hours of video and 3862 classes. Both these datasets were collected from YouTube³. The dataset used in this thesis is the V3C1 [6] dataset, which serves as an evaluation basis for the VBS and TRECVID video search tasks. It contains about 1000 video hours and comes with a predefined master shot reference and its representative keyframes. These videos were collected from Vimeo⁴. Each video contains additional information from Vimeo metadata, e.g. title, short description, and associated tags. This dataset was chosen over the previous two because three main reasons. The first is similar to an attribute of the class-less image classification datasets; hence, the V3C1 dataset lacks a fixed number of predefined classes. The second one is for the purpose of easy reproducibility. This thesis will use the master shot reference and its keyframes. Doing so makes the input data easily obtainable from the official source because this dataset cannot be directly shared. Thus providing the reference keyframes leads to higher reproducibility without breaking the dataset terms. See figure 2.1 for a comparison of ImageNet (on the

¹<https://www.flickr.com/>

²<https://www.profimedia.com/>

³<https://www.youtube.com/>

⁴<https://vimeo.com/>

left) and V3C1 (on the right).

3. User study

Image similarity modelling and approximation require a lot of data with multiple levels of diverse images. We conducted a user study to collect human judgments and explore the resulting data. In order to minimize potential biases and facilitate data interpretation, the study employed a judgment paradigm consistent with our previous research [73]. The original paradigm was based on a query image, and the user was asked to select from two options the one most similar to the query image. Forcing the users to choose one of the two images was occasionally a challenge from their perspective. Sometimes they could not decide and had to select a random option which could lead to some noise in the data and extra effort. In this thesis, a modification was made in the response format, allowing users to select from a range of five options based on two accompanying images. This approach resulted in more granular data that was easier to interpret.

The next goal of the user study was to gather user judgments on different diversities of the underlying data. These different levels of diversities reflect different stages of an interactive search. The exploration stage [69] usually requires assessing similarity on less similar images overall; on the other hand, the exploitation stage requires fine-grained similarity distinction among similar images.

Lastly, the context of the images can take part in the decision process, especially if the context is the same for all the images. If the context is the same (e.g. wedding images), then only minor differences play a crucial part in the decision process. However, they might be ignored by the image similarity models.

This chapter describes all the steps of the user study, beginning with the used image dataset. The description follows with feature extractors and their employed variants. Then the triplet generation algorithm is defined with respect to the previously stated conditions. Lastly, the web application for the participants' judgment collection is presented. The steps are depicted in figure 3.1.

3.1 Dataset

The underlying image dataset consists of v3c1 keyframes. There are 1 082 659 keyframes, some of which are hardly interpretable or trivial, e.g. single colour images, entirely blurred. These images were semi-manually filtered in our previous study [73]. The number of filtered images is small, and the resulting image set consists of 1 010 398.

Additional distinctions were made in the image selection process to gather judgments on both general and contextually similar images. Accordingly, three image sets were created: general, wedding, and scuba. The general set was composed of all filtered keyframes, while the wedding and scuba subsets were selected based on videos that had the "wedding" tag or mentioned "wedding" in the video title or description and similar criteria for scuba videos, respectively. It should be noted that these subsets represent a relatively small proportion of the entire dataset. Specifically, the wedding and scuba subsets comprised 69,388 and 3,803 keyframes, respectively.

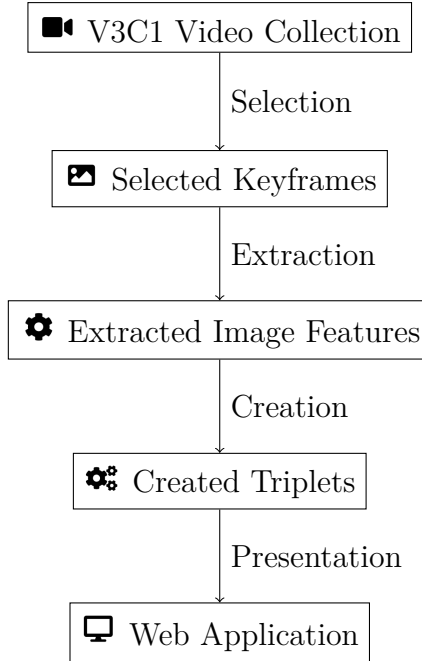


Figure 3.1

3.2 Extractors

As described in the previous chapter, current state-of-the-art image similarity models are mostly vector spaces on top of the deep features from deep neural networks (DNN). These neural networks are usually trained on different domains, e.g. image classification or text-to-image retrieval. This thesis studied various state-of-the-art DNNs and even included simpler colour-based and SIFT-based models. These methods were widely used before the massive success of the DNNs in other domains. The extractors comprise an underlying model and return a feature vector for each input image. The complete list with their embedding dimension and trainable parameters is shown in table 3.1.

3.2.1 Colour-based extractors

Colour-based extractors compute similarity based only on values of pixels from images with no additional training data. This approach was successful in some well-defined domains [48]. Three types of these extractors are covered in this thesis: RGB histogram, LAB k-means, and LAB positional.

The first method, RGB histogram, is the most simple one. The RGB colour space is the most used colour space for work with images in computer science. This method computes the histogram for each channel, i.e. red, green, and blue, concatenates the histograms and normalizes it with the L2 norm. It is computationally lightweight and captures a general colour layout. However, the RGB colour schema is not perceptually uniform [50], and even though the extractor does not use the RGB colour space directly for computing distances, the resulting histogram can still be affected. The number of bins for the histogram computation is an adjustable parameter. The higher number is, the better distinction between colours can be achieved. However, the lower number of bins can balance

the imperfection of the colour space and assign a higher similarity score for closer colours. In this thesis, the studied number of bins is 256 and 64.

The previous method has some drawbacks, some of which were discussed, e.g. perceptually non-uniformity. Additionally, the user's perception of the image does not always have to use all colour pixels and can only focus on the most dominant colours. In the LAB k-means method, both of these drawbacks were addressed. This method computes k-means [23] on the set of pixels present in the image. Then the centroids from the k-means are sorted by their hue from the HSV colour space. In this order, the centroids' pixel values are concatenated. In contrast to the previous method, the pixels are in CIELAB colour space [52] instead of RGB. Based on psychological experiments, CIELAB colour space was specifically designed to reflect colour similarities. Thus, the first drawback is addressed using the said colour space. The second drawback is handled with the k-means algorithm. The centroids reflect the high-level aggregation of the present pixel colours. The centroids primarily represent the most significant clusters and thus represent the dominant colours. The number of centroids is a parameter of this method, and this thesis investigates settings with four centroids. The higher number of centroids can be computationally demanding for a large number of input images.

Another drawback yet to be mentioned is that the previous two methods lack information about the position of pixels. The position of the colour pixels can be a decisive element. The last colour-based extractor is LAB positional. It uses the same colour space as the previous one. However, the difference is in how the representatives are selected from the image. The input image is divided into $n \times n$ chessboard-like sections. Then, a representative colour is computed with an arithmetic average for each section. The representative colours are then concatenated row-wise and create the resulting feature vector. The n is the parameter of this method, and this thesis works with values two, four, and eight.

3.2.2 SIFT-based extractors

Beyond-colour image analysis was another method widely used for various computer vision and CBIR tasks, e.g. Canny edge detector [14], Histograms of oriented gradients [10], and SIFT [47]. The SIFT features are widely used for image matching and object recognition. They are designed to be robust for resizing, cropping, change of illumination, or viewing angles. However, this method creates a dynamic number of keypoints; thus, complicated and computationally exhaustive feature matching for every two images would have to be implemented. In this thesis, an aggregated version of these features will be used. Delhumeau et al. [11] proposed a bag of features (BOF)-like SIFT feature aggregation called VLAD. The BOF aggregation requires a dictionary of features. Then for each feature from the set of features, the method computes the nearest neighbour from the dictionary and increments the counter for the dictionary feature vector. The resulting aggregated feature vector is the counted histogram. The aggregation proposed in Delhumeau et al. [11] does not create a histogram of dictionary features but computes deviations between the input feature vector and the nearest dictionary feature vector. Then it sums all the deviations. The sum is then normalized using "power-law normalization", which is applied component-wise and

should reduce the influence of the bursts in the natural images. The final step is L2 normalization. The vocabulary size is a parameter of this method, and in this thesis, a vocabulary size of 64 will be used.

3.2.3 DNN-based extractors

In recent years, the state-of-the-art in many domains (e.g. image classification, text-to-image retrieval, image generation) was dominated by DNNs [36, 59, 60]. The architectures of the DNNs have been changing over the years. One of the simplest is a deep multilayer perceptron, which is just an extended multilayer perceptron [51] with an enormous amount of hidden layers and parameters. Deep convolutional neural networks (CNN) surpassed the fully connected models in image processing [36, 24, 77]. This network architecture is based on small filters applied to the input as a sliding window. Some pooling layers or normalization layers usually accompany the convolutional filters. In Vaswani et al. [72], a novel architecture called Transformers was proposed and has achieved state-of-the-art in the natural language processing (NLP) domain. This architecture was adapted for the computer vision domain in Dosovitskiy et al. [15]. This thesis will focus on various models from the two last-mentioned architectures due to their superior performance in other computer vision tasks.

The extractors with an underlying DNN are relatively straightforward. The input image is passed to the neural network, and the features are activation values from a hidden layer. A last hidden layer will be used in this thesis if not stated otherwise.

This thesis will use six different types of CNN-based extractors with various sizes. One of the models is ResNet introduced in He et al. [24]. This architecture was one of the first which overcome the issue of learning deep neural networks. Prior studies could not achieve better results with deeper networks. This study presented skip connections which improved identity mapping throughout multiple layers and improved backpropagation efficiency. This thesis will use ResNets with layers 50, 101, and 152.

Tan and Le [70] presented an easily scalable CNN architecture. The researchers proposed width and resolution scaling hand-to-hand with traditional depth scaling. Additionally, they performed a multi-objective neural network search to optimize ImageNet accuracy and floating point operations per second (FLOPS). The FLOPS metric is there to reduce computational complexity. The found model was used as a base model B0; then, the optimal scaling was employed. It was shown that it outperformed the state-of-the-art regarding ImageNet top-1 accuracy and image size at that time.

A network for AVS tasks was proposed by Li et al. [40]. It consists of two encoders which embed videos and text queries into the same feature vector space. The text encoder embeds the text query using a bag of words, word2vec and GRU layers. The text encoder will be omitted in this thesis. The video encoder uses the CNN feature extraction consisting of ResNet and ResNeXt [77]. This encoder embeds all the sequence frames and then applies a mean pooling layer on the feature vectors. We can use this video encoder as a simple image encoder. This network was trained on MSR-VTT [79] and TGIF [41] datasets.

In the natural language processing (NLP) domain, novel architecture Trans-

formers surpassed previous state-of-the-art [72]. Dosovitskiy et al. [15] succeeded in using this architecture in the image domain. The use of the Transformer architecture was not straightforward and required some modifications. One of the changes was to create a sequence of tokens from an image. It was achieved by splitting the input image into patches, which were treated as words. According to their research, this novel architecture outperformed previous state-of-the-art CNNs on the ImageNet dataset and even in transfer learning on different datasets.

Another Transformer architecture called CLIP was proposed by Radford et al. [59]. It consists of image and text encoders, and its purpose is zero-shot classification. This architecture was trained on 400 million image-text pairs collected from the Internet. Then the contrastive learning technique was used to learn weights for the text and image encoders. The researchers showed that the CLIP model performed better as a zero-shot classification model than another zero-shot classification model or linear probes on some state-of-the-art classification models. The "data-less" approach in zero-shot classification is a huge advantage compared to the standard image classification approach. However, the authors discussed that the zero-shot classification would require much more computational power to achieve the state-of-the-art performance of the supervised networks.

The last Transformer used in this thesis was introduced by Chen et al. [8] and called ImageGPT. This network was inspired by the success of GPT-2 [58]. It uses self-supervised methods with autoregressive and BERT objectives. According to the study, the network produces high-quality image representations and surpasses some networks in some domains with only linear probing. The pre-trained model is also performant in fine-tuning.

Despite the success of the Transformers architecture used in computer vision, heavy criticism of the architecture in this domain appeared [43, 20]. As the authors of the studies describe, this architecture has three main drawbacks. The first one is the spatial limitation caused by treating the 2D image as a list of tokens. The next drawback is a quadratic complexity with respect to the input. It can be discussed that most test benchmarks scale down the images to small resolutions (e.g. 224x224), so the image size does not play a role in the computational complexity. However, it is not always the case, and it takes itself out from possible candidates for a general deep neural network backbone. Lastly, one of the advantages of the Transformers is their spatial adaptability, but it lacks channel adaptability. The channels are also important [21] and carry important information about objects in the image.

In a recent study, Liu et al. [43] tried to fix the addressed issues and introduced a novel architecture, Visual Attention Network (VAN), and a new type of layer, Large Kernel Attention (LKA). The LKA combines the self-attention mechanism with convolution filters. At first, it computes attention values using depth-wise convolution, depth-wise convolution with dilatation, and 1x1 convolution. In contrast to the self-attention mechanism previously used, it does not apply any activation function, e.g. sigmoid. The input values are then multiplied by the computed attention values and forwarded to the feed-forward network. Traditionally, the authors added skip connections and batch normalizations to the basic VAN stage. It was shown that this novel backbone outperforms most of the SOTA backbones with similar GFLOPs on different domains, e.g. image classification on ImageNet and CUB200 [76] datasets, object detection and semantic

segmentation on COCO [42] dataset.

A more radical approach was employed by Guo et al. [20], where the success of the Transformers architecture was primarily attributed to their superior ability to scale at the expense of the inductive bias of CNNs. The authors revisited the scaling capabilities of ResNet architecture. They performed many modification trials where different macro and micro changes, kernel sizes, inverted bottlenecks, and more groups (as introduced in ResNeXt [78]) were applied to different sizes of ResNet. As a result, a novel architecture ConvNet was introduced. This architecture outperformed SOTA Transformers architectures on the ImageNet classification benchmark with the same or fewer GFLOPs.

3.3 Triplet selection

In this thesis, the similarity will be assessed relatively in triplets rather than absolutely in pairs, which were employed in some mentioned studies in the previous chapter. This way, some problems will be avoided, e.g. lack of interpretability or different neutral points per user.

This thesis aims to explore the image similarity for CBIR. In this field, the demands on similarity modelling can differ because the CBIR tools usually start with exploration, and in the later stages, exploitation comes to play. The tools handle it differently, and it is referred to as the exploration-exploitation trade-off [32]. Therefore it is required to assess similarity credibly both on highly dissimilar and highly similar images. The similarity of highly dissimilar images is usually computed in the exploration part of the process. In this part, the user starts providing some input to the system, e.g. text query or image relevance feedback. The system tries to utilise the inputs the user provides and provide the most relevant results. This part is referred to as exploitation. During this stage, the results are usually highly similar; thus, similarity modelling must handle these images.

The process of triplet generation has to cover both of these stages. If we ignore a single stage, we could end up with a similarity model, neglecting a part of the search process. Simple random triplet generation would not suffice because the V3C1 dataset is large, and the number of highly similar frames for a random frame is significantly lower than the dataset size due to the high diversity of the dataset [6]. Thus we need to create triplets based on their similarities. However, the chicken and egg problem applies here. To model image similarities, we need to gather similarity data to find how to model the similarities the best. Therefore in this thesis, every extractor discussed in the previous section will be used to create some triplets. This method was already used by Veselý and Peška [73]. But first, some assumptions need to be taken into count. The first assumption is that some extractors model similarity better than others. This assumption is trivial and expected due to the different nature of the extractors. The next assumption is that at least some extractors model the similarities credibly enough. This has been proven to some extent in the previously mentioned competitions [26, 22]. Therefore with this in mind, we will be able to create triplets with some baseline similarities taken into count.

The second assumption might lead us to a quick question, why don't we just use one of the similarity models from the CBIR tools? We could do it; however,

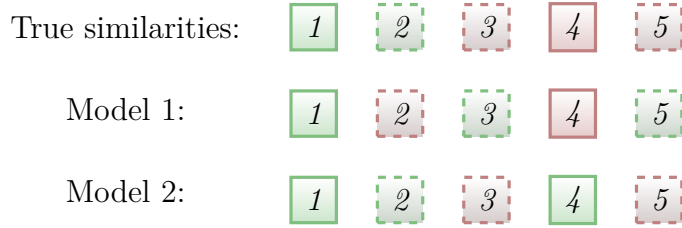


Figure 3.2: A depiction of presentation bias. The squares represent an image from a dataset; green ones depict highly similar images to a query image, and red depict dissimilar images. Images 1 and 4 are selected by model 1 and will be chosen as a triplet.

the results could be influenced by the presentation bias [5]. The presentation bias in information retrieval and recommendation systems is a kind of bias which is induced by the combination of items that are presented to the user. If the items presented to the user are generated by an algorithm and not at random, then the resulting item interactions or annotations will be on a specific subset. This can lead to misleading results where the performance of other algorithms is inferior to the prior one. A small example is depicted in figure 3.2. Let’s say we will be going to model the similarities with Model 1. We select a random query image, and we want to select one highly similar and one dissimilar image for the triplet. The Model gives us three images as highly similar and two dissimilar images. From both classes, we sample randomly and get images 1 and 4. When we evaluate the performance of these models on this triplet, we get a 100% success rate for Model 1 and 0% for Model 2. That indicates that Model 1 is superior to Model 2; however, the opposite is true. Model 1 has a 40% true success rate, and Model 2 has a 80% success rate. Using more image similarity models does not prevent presentation bias from occurring entirely but greatly minimizes it.

Sometimes the CBIR systems can be used only on a specific domain. The closeness in the similarity model does not have to mean the same domain; thus, this thesis will examine two subdomains of the V3C1 dataset separately, i.e. wedding and scuba videos. These subdomains were selected for their specificness and their size. Both domains are relatively small. The wedding domain is larger than the scuba with 66373 keyframes, which is about 6.6% of all V3C1 keyframes. The scuba domain comprises 3530 keyframes, about 0.3% of all V3C1 keyframes. The keyframes were selected according to the video metadata provided by the V3C1 dataset. The video title, description, or tag had to contain the word wedding (scuba) to be included in the wedding (scuba) domain. From now on, the whole V3C1 dataset will be referred to as the general domain.

From these three domains, triplets will be generated with respect to the exploration/exploitation stages and their transition. For this purpose, the triplet generation algorithm is described in algorithm 1. Firstly, for each domain, we need to compute an image features matrix and select an appropriate (dis)similarity function. The (dis)similarity functions widely used are cosine similarity or L_2 distance. Both of these functions will be used for triplet generation, each for half of the triplets. The next step is to define what kind of similarity between triplets is needed and how to achieve that. The absolute criteria on the (dis)similarity functions are somewhat unreliable and complicated. It would require an analysis

of each feature matrix and its similarities among the rows. A different approach was chosen to cope with this issue. The whole domain will be sorted based on the similarity to a query. The query image is selected randomly from the entire domain. Then this sorted list is then divided into bins of fixed sizes. The bins are $[2^4, 2^8, 2^{12}, 2^{16}, 2^{20}]$ for the general domain. For the wedding domain, the bins $[2^4, 2^8, 2^{12}, 2^{17}]$ were used. The smallest bins $[2^4, 2^8, 2^{10}, 2^{12}]$ were used for the scuba domain. The bins should always reflect multiple levels of similarity, e.g. near duplicate, similar, somewhat similar, or completely irrelevant. Then more similar and at least the same or less similar bins are selected. One image is then randomly chosen from both bins. Then each image from the option pair is assigned either side, i. e. left or right. Therefore this algorithm does not implicitly favourite either side, and the triplet is presented in the same layout. The process repeats until the algorithm satisfies the number of images per bin.

The last step of the triplet generation was to add additional metadata for each triplet. It consisted of similarities based on all similarity models and their relative ranks. Three similarity models (W2VV++, RGB histogram with 64 bins, and VLAD) were chosen for a special role in gamification, described in the next section. For the three chosen models, an additional simplified binarized similarity decision was stored.

The feature extraction and cleaning framework is attached to this thesis. The user documentation is in appendix A.1, and the programmatic documentation is present in the project directory in the `docs/html/index.html`.

3.4 Web application

The user interface for the study participants was created as a web application. The web application, rather than a desktop or mobile application, was chosen for its easy distribution and overall familiarity of the users with the web applications. The study distribution was as easy as sending a single Uniform Resource Locator (URL). The goal of this study was to make it as accessible as possible. Thus two language mutations were implemented; Czech and English. Moreover, the intermediate results were immediately available.

The first thing the user saw on the landing page (see figure 3.3) was the user information form. This form was optional to be filled out by the users. They could insert their email to receive the study results, their nickname for gamification purposes, an age group, highest achieved education, and familiarity with machine learning. Following the form, a Google reCaptcha¹ validation was applied to avoid some automatic abuse of the input form and prevent bots from getting the credentials and to the annotation screen. Lastly, informed consent was shown to the participants.

After filling out the user information form, an information screen was shown; see figure 3.4. Credentials were displayed to the user. They could return anytime with these credentials and continue with the study where they left off. Then a brief guide through the study was presented. Also, the participants were introduced to the gamification element of the study. The users continued on the annotation screen by clicking the "Continue" button.

¹<https://www.google.com/recaptcha>

Algorithm 1 A triplet generation algorithm.

Require:Domain D Image features matrix F Bins B Similarity function $sim : F \times F \rightarrow \mathbb{R}$ Number of query images per bin pair N

```
1:  $triplets \leftarrow []$ 
2: for  $i \in \{1, \dots, N\}$  do
3:   for  $b_1 \in B$  do
4:     for  $b_2 \in B$  do
5:       if  $b_1 \leq b_2$  then
6:          $q \leftarrow rand(size(D))$   $\triangleright$   $rand(L)$  returns random number from
            $\{1, \dots, L\}$ 
7:          $similarities \leftarrow []$ 
8:         for  $j \in \{1, \dots, numOfRows(F)\}$  do
9:            $similarities.add(sim(F[q], F[j]))$   $\triangleright$  adds a similarity value
           to the similarities list
10:        end for
11:         $sortedIndices \leftarrow argSort(similarities)$   $\triangleright$   $argSort(list)$ 
           returns a descending list of sorted indices; e.g.  $[2,3,1]$  yields  $[2,1,3]$ ; the first
           element of the list is the index of the biggest number, the second element is
           the index of the second biggest number and so on...
12:         $offset_1 \leftarrow rand(b_1.end - b_1.start)$ 
13:         $offset_2 \leftarrow rand(b_2.end - b_2.start)$ 
14:         $query \leftarrow D[q]$ 
15:         $choice_1 \leftarrow D[sortedIndices[b_1.end + offset_1 - 1]]$ 
16:         $choice_2 \leftarrow D[sortedIndices[b_2.end + offset_2 - 1]]$ 
17:         $triplets.add((query, choice_1, choice_2))$   $\triangleright$  add the created triplet
           to the result
18:       end if
19:     end for
20:   end for
21: end for
22: return triplets
```

The annotation user interface originates from the preliminary study [73]. The goal was to make the study as accessible as possible and motivate the participants to annotate more than the asked minimum. See the figure 3.5 for an example. Three images were dominant on the annotation screen. The top image was the query image, and the two bottom ones were the choice one and choice two images. The text at the top of the screen said: "Which image is more similar to the one on the top?". At the bottom, there were five options for the users to choose from: Left, Maybe left, I don't know, Maybe right, and Right. Under the choices, a submit button is present to send the annotation. At the top right corner, there was a Help button to show a Help modal window. The last UI element on the screen is in the top left corner, and it is level progress, which is part of the gamification element.

The triplets, created by the previously introduced algorithm, were imported into a database. Then, a random category was selected for each trial, and from the chosen category, a random triplet was presented to a participant. The participants' agreement on a single triplet would be hard to estimate with random triplet sampling. Therefore an additional meta-category was introduced. Random 500 triplets from the general category were selected and marked as a repeating category. In this category, an exception from random sampling was made. The triplets were not chosen randomly but sequentially to get the maximal number of annotations for as many triplets as possible. The maximal number of annotations per triplet was twenty.

The main differences between the preliminary study and the study in this thesis are three. The first difference is the different choice possibilities. In the preliminary study, the user was forced to choose either the left or right images. There was no possibility of expressing uncertainty. Secondly, the user incentive was purely their will to continue. This study added an external motivation to encourage the user to annotate more triplets. Lastly, the preliminary study contained only the general set compared to this study, which distinguished among general, scuba, weeding, and repeating subsets.

3.4.1 Gamification

The participants of this study, as for the preliminary study, were recruited by the author and his supervisor. The participants were recruited primarily from friends, colleagues, acquaintances, or transitively through social groups. There were no money or price incentives applied. Therefore the primary motivation of the participants was doing a favour to whom they were invited. An additional external stimulus was implemented to gather more annotations.

A gamification was chosen as an external stimulus, which has been used in software design for more than a decade [13]. As defined by Deterding et al. [13], gamification is the use of game design elements in a non-game context. This method is often embedded in software designs to encourage users to use the software more or in a more enjoyable way.

The game elements in this study were inspired by role-playing games² (RPG). Every player of an RPG controls their own character, which they try to improve. The improvement is made by doing quests and other tasks. The better the

²<https://encyclopedia.pub/entry/1583>

character, the more likely the player will win some tournaments, gain better rewards, or be higher in the leaderboards.

One of the game elements adopted in this study is levels. These were not just simple numbers, but they were inspired by a fantasy medieval society. A participant's character started as a beggar, and for each 20 annotated triplets, the character would level up. The level-up meant that the character became a higher member of the fantasy society. The current level was displayed as a moving picture with a name in the top left corner of the screen. The moving images of levels were provided under CC-4.0 by chierit³. The highest distinct level was king with guards. From this level, only the number of guards was increasing to motivate the participants to continue beyond this highest level.

The next game element was the leaderboard (see figure 3.6). In serious games, the leaderboards or scoreboards add a competitive part. This way, the players can compete among themselves and try to achieve the top of the leaderboard. The leaderboard in this study was sorted by the number of annotated images. The participants could see the leaderboard with the top five participants and their rank at every level-up. At the start, they could choose a nickname under which they were presented on this leaderboard.

The last gamification implemented in the study was statistics. The statistics in serious games often show some interesting facts about the gameplay. In this study, agreement statistics were primarily displayed to the participants. First was an agreement with W2VV++, RGB histogram, and VLAD models. These values were precomputed in the triplet selection described in the section 3.3. Moreover, agreements with the other users were computed, and the nickname of the most agreeing user was shown.

3.4.2 Implementation

The annotation application was implemented in Javascript with Node.JS⁴ as a runtime environment on the server. Express⁵ was used as a web application framework for its flexibility, robustness, and ease to use. PostgreSQL⁶ database was used for storing information about triplets and their annotations. PostgreSQL is a relational database with many years of active development, active community, performant and easy to set up. The front end was written in Embedded Javascript templating⁷ (EJB) and styled with Bootstrap⁸.

The whole framework stack was chosen based on ease of use, effectiveness, and our previous experiences. Other back-end framework stacks were also taken into account, e.g. C# + ASP.NET core⁹, Java + Spring¹⁰ ecosystem, or PHP + Laravel¹¹. The main requirements for the web application were:

- Easy deployable on Debian

³<https://chierit.itch.io/lively-npcs>

⁴<https://nodejs.org/en>

⁵<https://expressjs.com/>

⁶<https://www.postgresql.org/>

⁷<https://ejs.co/>

⁸<https://getbootstrap.com/>

⁹<https://dotnet.microsoft.com/en-us/apps/aspnet>

¹⁰<https://spring.io>

¹¹<https://laravel.com/>

- Straightforward data persistence
- Support for internationalization
- Ability to secure communication with Transport Layer Security (TLS)
- Manage access of the users
- Ability to handle at least a few (less than a hundred per second) concurrent annotations

These requirements would be satisfied by all of the previously mentioned framework stacks, and therefore the ease of use, preference and previous experience came to play. The application deployment was as easy as installing Node.JS from the standard package manager, then pulling the repository, installing dependencies with a single command using the Node Package Manager¹² (npm) and then starting the web server with a single command. The database read and write operations were handled using node-postgres¹³ module, which allows quickly inserting SQL queries and executing them with sanitized arguments. Module i18n-express¹⁴ provides internationalization support with a single configuration line. The Node.JS HTTP server supports adding the TLS layer by just adding a configuration line with the path to the server key and public certificate. The user authorization is managed with an HTTP authorization [19], and the credentials are validated against those stored in the database. Users can't change their credentials to avoid obtaining some sensitive information because they tend to use the same login for more online services. The users weren't expected to return very often, so the automatically generated passwords were a tiny inconvenience. The selected frameworks and database should easily handle more than a hundred HTTP requests per second. Additionally, Peška et al. [56] already used a similar framework stack and was proven satisfactory. More detailed user documentation can be found in appendix A.2 and the programmatic documentation in the project directory in docs/index.html.

¹²<https://www.npmjs.com/>

¹³<https://node-postgres.com/>

¹⁴<https://www.npmjs.com/package/i18n-express>

Feature extractor	Feature dimension	Trainable parameters
RGB Histogram 256	768	-
RGB Histogram 64	192	-
LAB Clustered 4	12	-
LAB Positional 2×2	12	-
LAB Positional 4×4	48	-
LAB Positional 8×8	192	-
VLAD	8192	-
ConvNeXt Tiny	768	28M
ConvNeXt Small	768	49M
ConvNeXt Base	1024	88M
ConvNeXt Large	1536	196M
EfficientNetB0	1280	5.3M
EfficientNetB2	1408	9.2M
EfficientNetB4	1792	19M
EfficientNetB6	2304	43M
EfficientNetB7	2560	66M
ResNetV2 50	2048	26M
ResNetV2 101	2048	45M
ResNetV2 152	2048	60M
VAN Tiny	256	3.8M
VAN Small	512	13M
VAN Base	512	26M
VAN Large	512	44M
W2VV++	2048	152M
CLIP patch16	768	86M
CLIP patch32	768	87M
ImageGPT small	512	76M
ImageGPT medium	1024	455M
ViT Base	768	86M
ViT Large	1024	307M

Table 3.1: Full list of feature extractors with their embedding dimension and trainable parameters.

Uživatel: - Nápověda

Uživatelské informace

Czech English

Email

Pokud není e-mail uveden, nebude možné provést výsledky studie a není povinný.

Přezdívka

Přezdívka může být viditelná ostatním účastníkům studie. Nevyplňujte vaše jméno nebo jiné osobní údaje. Používejte pouze písmena bez diakritiky nebo čísel.

Věková skupina:


Nespecifikováno

Nejvyšší dosažené vzdělání:

Nespecifikováno

Jste obeznámen s oborem [strojového učení](#)?

Nespecifikováno

Nejsem robot 

Informovaný souhlas

S čím byste měli souhlasit, pokud chcete pokračovat (informovaný souhlas):

Seznámil jste se s cíli výzkumu (viz Nápověda vpravo nahoře) a nevádí vám se na něm podílet (např. váš neuráží, není rozporu s vašim přesvědčením apod.). Souhlasíte s tím, že můžeme použít vaše (anonymní) odpovědi a vaše demografická data k prezentaci například vědeckých zpráv. Pokud nebudeme zveřejňovat vaše osobní údaje, ty zůstanou neznámé a ani je nechochceme; nebudeme ani zveřejňovat váš e-mail, pokud nám ho poskytnete. V práci, která na základě výzkumu vznikne, bychom chtěli zveřejnit dataset s výsledky v následujícím rozsahu:

- anonymizovaný, náhodně vygenerovaný identifikátor účastníka (něco jako UID = 535466)
- demografická data účastníka (věková skupina, vzdělání, znalost strojového učení)
- odpovědi účastníka

Výzkumu se účastníte bez nároku na honorář (kromě naší vděčnosti a dobrého pocitu ze pomůlky vědy).

Výzkum můžete kdykoliv přerušit (prostě zavřete okno prohlížeče). Pokud chcete svůj souhlas se zpracováním vašich odpovědí odvolat, napište nám e-mail (kontakt je v okně Nápověda). Pokud jste nevyplnili e-mail, budeme potřebovat trochu více informací, abychom mohli vaše interakce zobrazit (například kdy jste začali s experimentem, nebo vaši IP adresu).

Pokud jste vyplnili e-mail, ale nechcete, abychom vás nadále kontaktovali, prostě nám to napište...

[Získat přístup](#)

User: - Help

User information

Czech English

Email address

The email will be used only for browsing the study results and it is not mandatory.

Nickname

The nickname may be visible to the other participants. Avoid using your name or any personal information. Use only characters without diacritics and numbers.

Age group:


Not specified

Highest achieved education:

Not specified

Are you familiar with [machine learning](#)?

Not specified

Nejsem robot 

Informed consent

Before continuing with the research, you should be familiarize yourself and agree with the following statements:

I familiarize myself with the aim and targets of the research project (see Help at the top right) and I do not mind to contribute to it (i.e., the research topic does not go against my beliefs etc.). I agree that authors of the project may utilize my (anonymous) responses and my demographic data while presenting research outcomes - e.g. in scientific papers. I do not require any monetary compensation for my participation in the research. (Nonetheless, we will really appreciate your help! Plus, there is a good feeling for helping the science!)

We will never publish data that could breach your identity (we do not have such data anyway). We also neither share your e-mail address to third parties nor disclose it publicly (if you opt to provide it) in the prospective paper; we would like to publish an anonymized dataset of responses. The datasets should contain following information

- randomly generated ID of the participant (i.e. UID = 468201)
- demographic data of the participant (age group, education, machine learning familiarity)
- participant's responses to individual tasks

In the dataset, there will be no mapping between the ID of the user and his/her true identity (we do not have such data anyway)

You can stop your participation at any time (just close the browser windows). You can always revert your consent to use your responses - just write to us with your details (submitted e-mail, time when you started participation etc.). If you did submit your e-mail, but do not want to receive any new messages from us, just write it to us...

PS: any disputes (hopefully, non should arise...) will be governed by the law and jurisdiction of Czech republic.

[Get login](#)

Figure 3.3: UI. Top: A screenshot of the web application with a user form in Czech. Bottom: A screenshot of the web application with a user form in English.

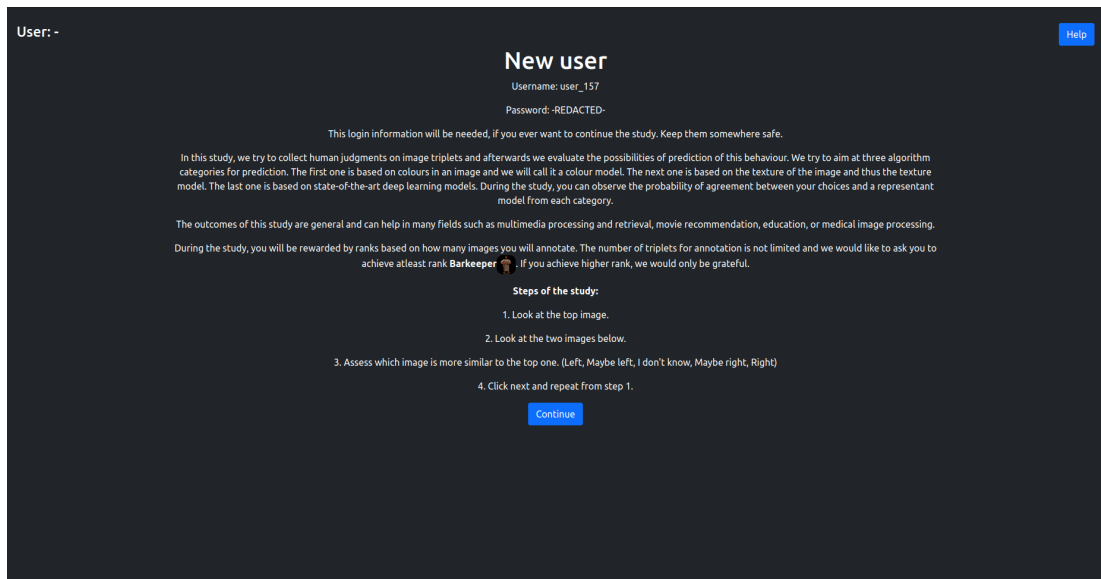
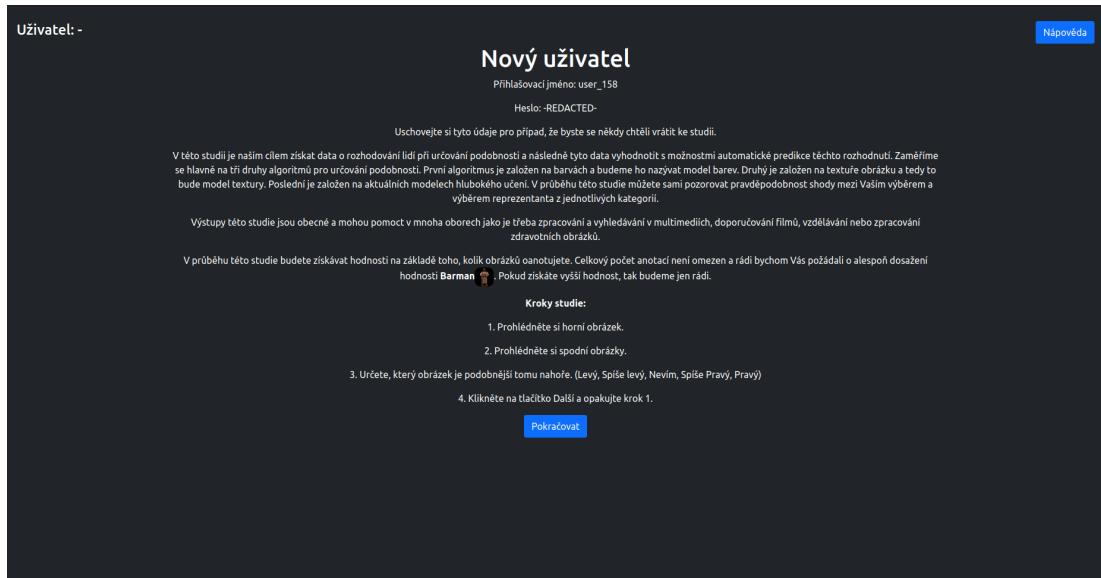


Figure 3.4: UI. Top: A screenshot of the web application with credentials and a short study briefing in Czech. Bottom: A screenshot of the web application with credentials and a short study briefing in English.

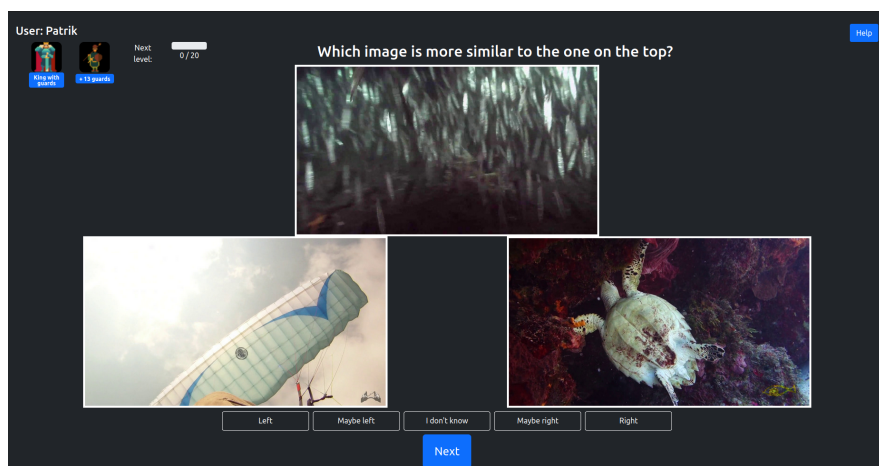
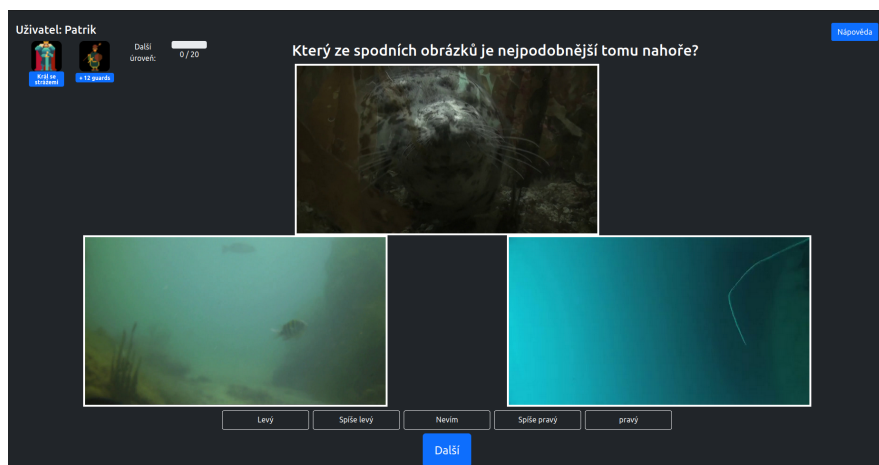
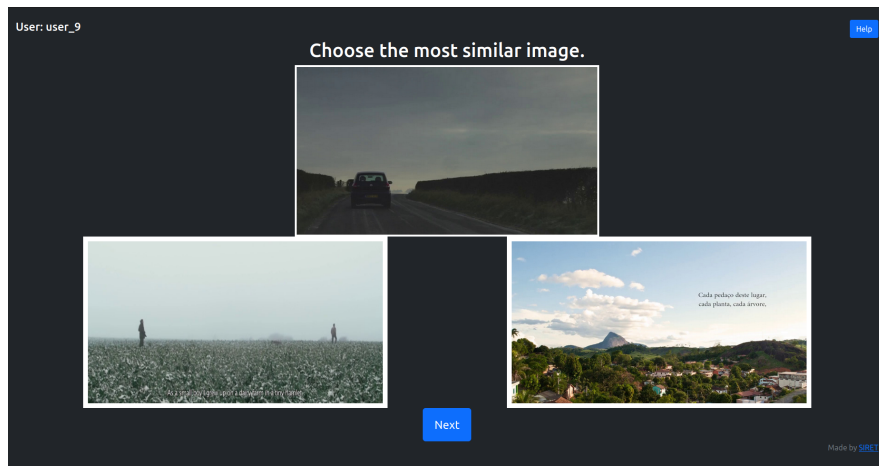


Figure 3.5: UI. Top: Legacy UI from the preliminary study[73] Middle: A screenshot of the web application in Czech. Bottom: A screenshot of the web application in English.

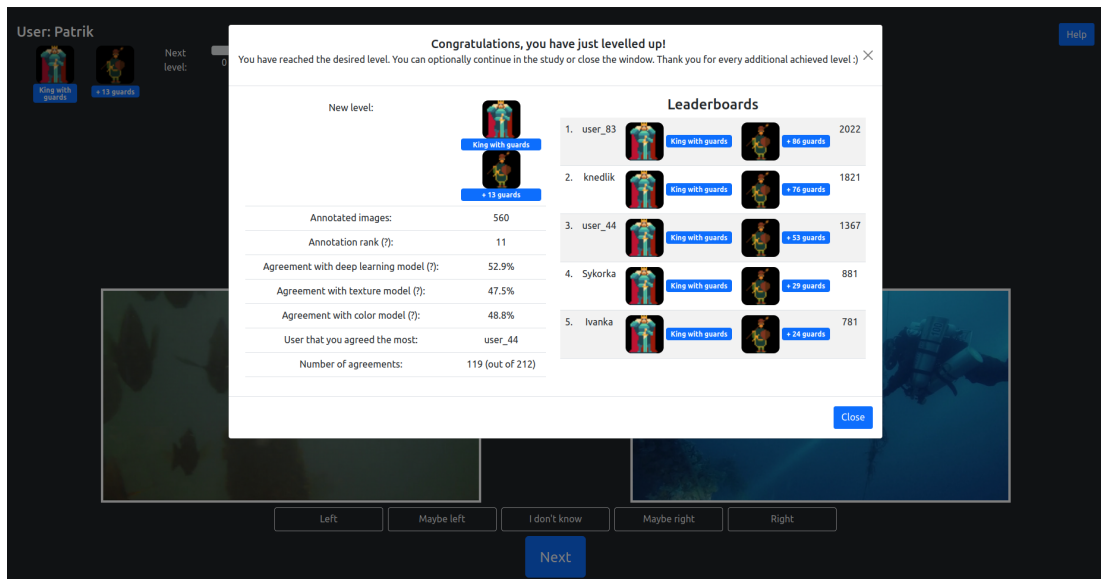
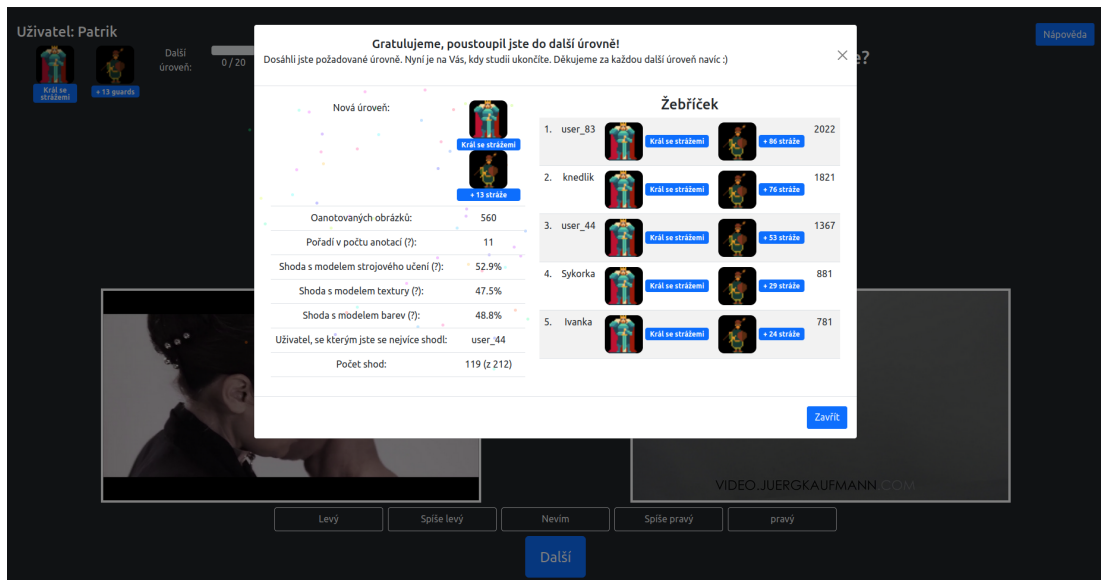


Figure 3.6: UI. Top: A screenshot of the web application with a level-up screen in Czech. Bottom: A screenshot of the web application with a level-up screen in English.

4. Dataset analysis

The user study results were stored in the database in the table `triplets_annotation`. The values stored for each annotated triplet are the user’s choice, decision time, window size, and the browser User-Agent[18]. In the postprocessing, some additional fields were computed. One of the fields was a handheld flag. This flag was set when the User-Agent contained a substring "iPhone" or "Android". The following fields were similarities among the query, options one and two.

This section will analyse the image similarity assessment capabilities modelled by the previously explained algorithms and limitations that can be observed by agreement among the participants. But at first, the overall dataset features will be investigated. Overall 17365 unique annotations were received from 84 participants.

4.1 Device types

The participants were not instructed to use any particular device, e.g. PC, tablet, or smartphone. Therefore, they could participate in the study on a device based on their preference. A brief device specification is stored in the columns window width, height, user agent, and a derived column handheld. Most users did not use any handheld device and probably used a laptop or PC; see left figure 4.1. The number of annotations on a handheld device was lower by 24.4% on average in comparison to other devices. However, the median annotation count for handheld devices was 89.5, which was slightly higher than the median of 80 on other devices. Only two participants used a handheld and non-handheld device; the others probably stuck with a single device.

The screen’s resolution can also impact the annotation quality; thus, it was logged by the web application’s front end. The logged resolution was the size of the inner browser window consisting of width and height in pixels. The sizes vary a lot, and hence the screen size was precomputed to the diagonal size in pixels. From these values, three major screen sizes come as a result. Small resolution screen with less than 1468.6 pixels in diagonal, corresponding to 720p (1280×720).

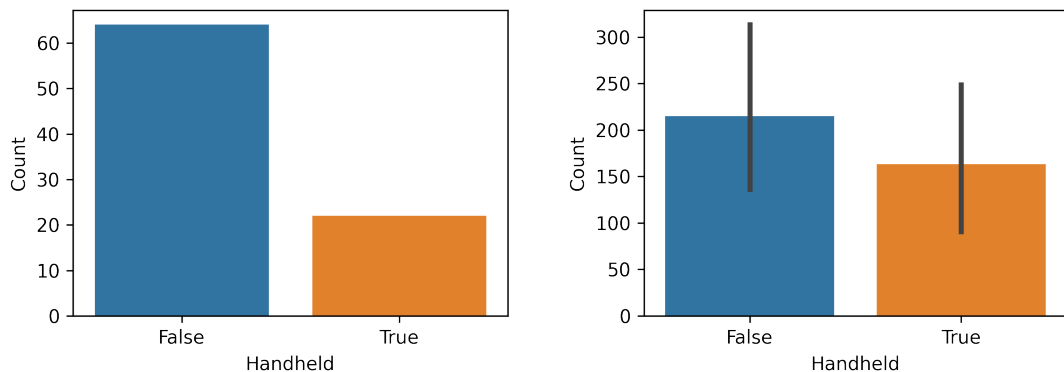


Figure 4.1: Left: Number of users that used a handheld or another device at least once. Right: Average volume of per-user annotations for the device type.

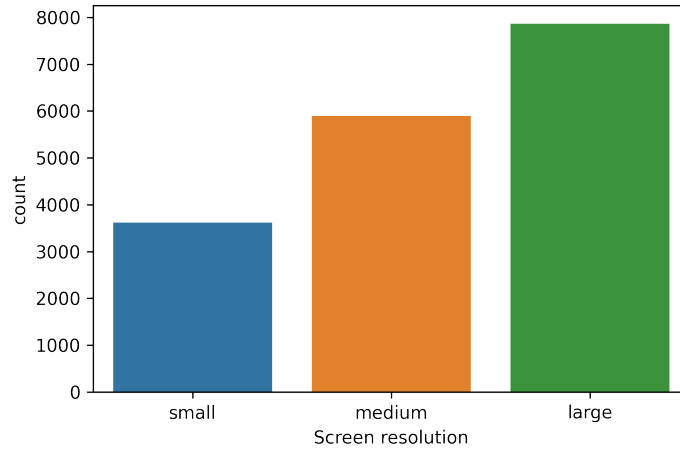


Figure 4.2: Distribution of the count of annotations with respect to the screen resolution.

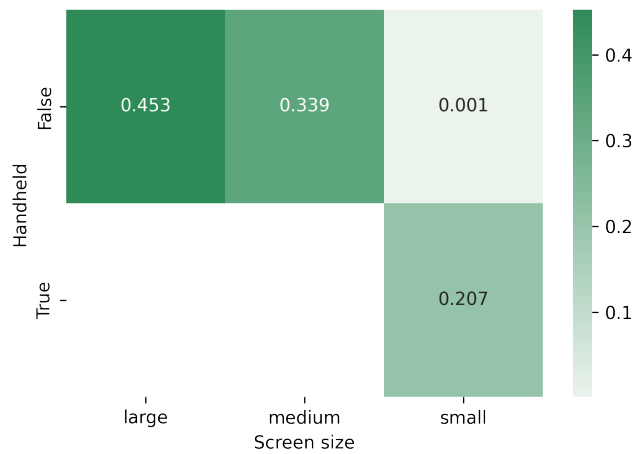


Figure 4.3: Distribution of the count of annotations with respect to the screen resolution.

The medium-resolution screens were classified as those with at least 1468.6 pixels in diagonal but less than 2202.9. The upper bound corresponds to the 1080p (1920×1080) resolution. The distribution of annotations count can be seen in figure 4.2. Most of the annotations (45.3%) were collected on a high-resolution screen. The second most used resolution was medium, representing 33.9% of all annotations. The last and least used resolution was the smallest one. This one was used in 20.8% cases.

An additional observation was made that annotations from small resolutions were primarily done on a handheld device. In figure 4.3, it can be seen the distribution of the annotation counts with respect to the handheld type and resolution type. The large and medium screen sizes were from non-handheld devices. This difference can be caused by the HW capabilities of the respondents or their browser usage (fullscreen or windowed mode).

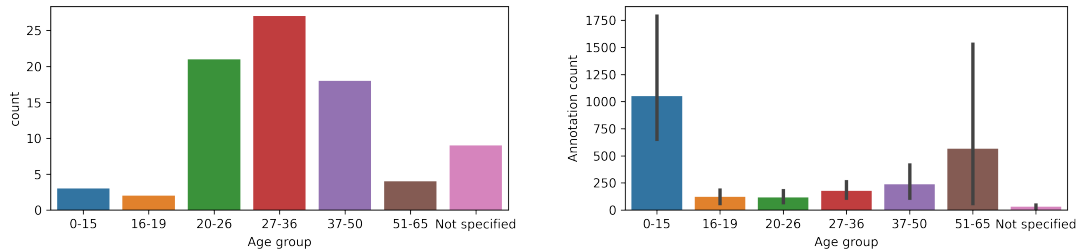


Figure 4.4: Left: The distribution of users with a given age group. Right: Average volume of per-user annotations for the age group.

4.2 Demographic attributes

The age group is one way to observe generational differences and, at the same time, not breach privacy. We created the groups artificially, and the participants chose them by themselves. These groups were split by the years: 0-15, 16-19, 20-26, 27-37, 37-50, 51-65, and 65+. The group intervals are closed intervals. They should reflect some development and stands of the individuals. Image similarity can be subjective, and it might change during human maturation.

This study covered nearly all age groups except the 65+ group. For age group distribution, see figure 4.4, left. The dominating age groups were 20-26, 27-36, and 37-50, with a total user share of 78.6%.

In figure 4.4 on the right, it can be observed that the groups 0-15 and 51-65 have the highest average annotation counts. However, the large error bars and the low number of participants shown on the left graph indicate that some outliers skew these averages. Interestingly the differences in average annotation count among the dominant groups are evident. The group 20-26 has, on average, 116.1 annotated triplets. The two following groups combined, 27-36 and 37-50, have higher average annotated triplet counts of 200.6 (one-sided Student’s t-test p-value = 0.0459). Their individual means are 175.5 and 237.0, respectively.

The following demographic attribute collected from the participants was their highest achieved education level. All the groups and their distribution can be seen in figure 4.5, except the group with no previously finished formal education, which did not include any participants. The group with primary school seems to be underrepresented. The large error bars indicate that most groups’ annotation counts are highly diverse. There is a vast overlap between this education group and the age group 0-15; thus, the means and error bars are similar.

Familiarity with machine learning algorithms is a specific attribute. However, the experience gained by knowing and using machine learning algorithms might induce some unintentional bias favouring them over other users. The distribution of the machine learning expertise is shown in figure 4.6. The dominant groups are those with no or small machine learning experience. This is expected given that it is a specific field of computer science. Similarly to the education level, the error bars of an average annotation count are relatively large, and thus there is a difference among participants in each group (see figure 4.6 on the right).

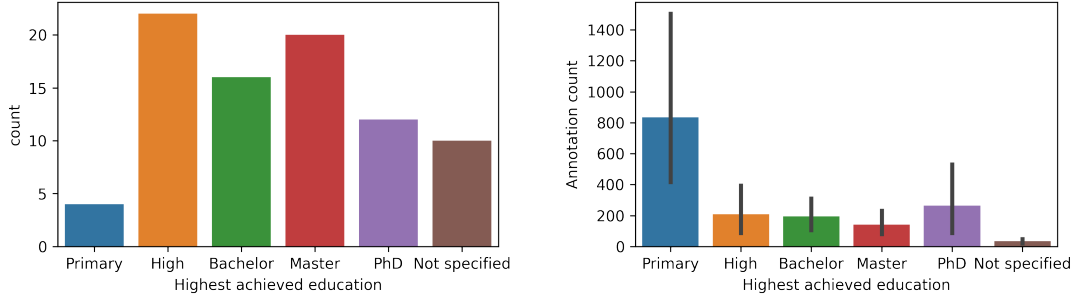


Figure 4.5: Left: The distribution of users with a given highest achieved education. Right: Average annotation count per group with a given highest achieved education.

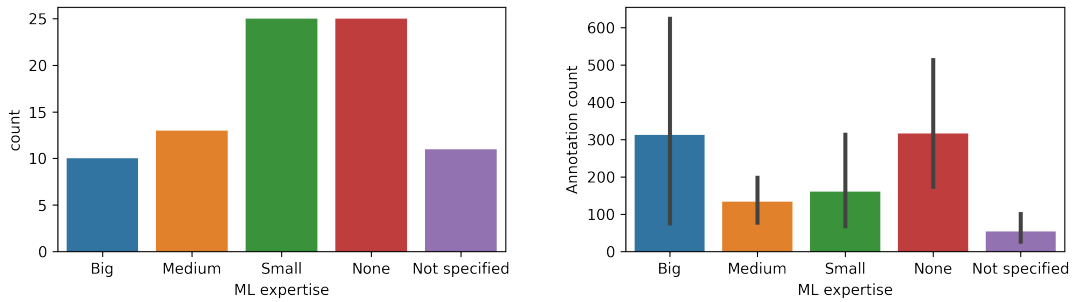


Figure 4.6: Left: The distribution of users with given machine learning expertise. Right: Average volume of per-user annotations for the machine learning expertise.

4.3 Gamification

The gamification elements in this study were employed to motivate participants to make more annotations. This change is one of the significant differences to the preliminary study [73]. Moreover, the annotators were given more choices which can lead to making the task harder.

The difference between this study and the preliminary is depicted in figure 4.7. The mean number of annotations per user was nearly twice as much. The mean annotation count in the preliminary study was 111.6, and in this study, it was 206.7 triplets. The median has also increased, even though the increase is not as significant as with the mean. The maximum increased excessively by nearly four times from 570 annotations in the preliminary study to 2021. This indicates that the gamification elements increase participants' motivation and keep them continuing even in a possibly more challenging task. This claim is supported by the Student's t-test between the preliminary and this study and yields $p = 0.0296$, and thus the gamification impact is statistically significant. However, the results show that some participants were motivated more than others. In 85.7% cases, the participants stopped right after a level-up screen. This may indicate that they were motivated to finish at least their current level after deciding to end the study.

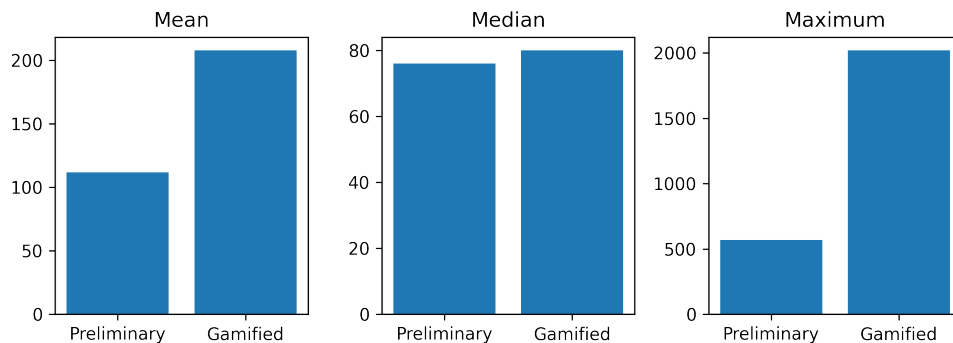


Figure 4.7: Difference of the annotation counts between this study and our preliminary study [73].

4.4 Annotations

As described before in section 3.4, the participants had five options to assess the triplet similarity. The options were: Left, Maybe left, I don’t know, Maybe right, and Right. Moreover, in section 3.3, the triplet selection was described with the goal of creating different levels of similarity. Given that, the number of responses for each judgment type should be somewhat uniform.

Figure 4.8 depicts the distribution of user choices. The figure is divided into subfigures, each depicting distribution in a different subset. The General and Repeating subsets are nearly identical and close to a uniform distribution. The participants tended to be more unsure about their Wedding subset responses. The most frequent responses were the Maybe left and Maybe right answers. This may indicate that the participants were more likely to favour one option over the second, even though they were not that sure. The middle option and the edge options were balanced. The Scuba subset seems to be a somewhat more decisive one. The participants used the I don’t know option the least from all three subsets. However, the other four choices are balanced. Overall all the subsets have mostly balanced answers and do not have any options entirely neglected.

The users can introduce some biases to the dataset during annotation. They were described in chapter 2. This study can be affected by the positional [29], assimilation, and contrast [81] biases. The presence of these biases can affect results if not appropriately addressed.

The positional bias does not seem to affect the results overall in the subsets based on figure 4.8. The Fisher’s exact test [71] was performed on a contingency table, where the first row represents the sum of Left and Maybe left option annotation counts and the Right and Maybe right annotation counts. The second-row values are both an average from the first row. The null hypothesis is that the user selections (left options vs right options only) correspond to a uniform distribution. The test yielded a p-value equal to 0.079, which is greater than 0.05; thus, the null hypothesis cannot be rejected with statistical significance.

The positional bias does not have to be observable directly in the overall statistics but can be present in the per-user distribution. For per-user statistics, we will omit the users with less than 30 annotations to avoid outliers caused by randomness. In figure 4.9, the users are depicted as dots, and their position is determined by how they respond. The y-axis represents part of the left options,

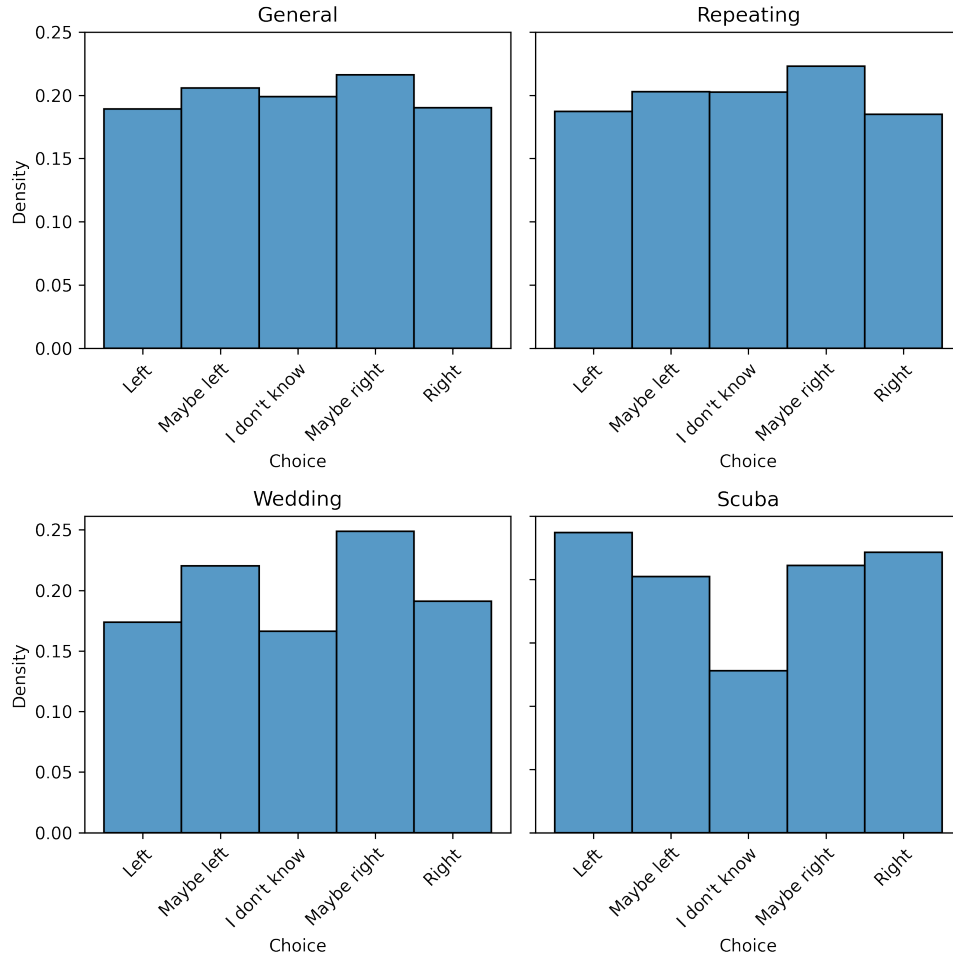


Figure 4.8: Distribution of annotations per subset.

and the x-axis represents the part of the right options from all of the annotations from the user. The size of the data point means their total annotation count. It can be observed that most of the data points are around the dashed red linear functions $y = x$ and thus have mostly balanced left/right options. The Fisher's exact test, computed analogously as previously, showed that for any participant, the null hypothesis could not be rejected, and even for 9 out of 69 participants, the p-value was greater than 0.95. Moreover, 32 of the participants had more left options selected, the next 36 participants had more right options, and the last remaining had the same left and right options. Therefore, there is likely no systematically significant positional bias in terms of left and right options.

However, another bias can be observed from figure 4.9. The willingness to select the neutral I don't know option differs highly among the users. Even though the users differed in what they wanted to decide and what they wanted to leave with the neutral option, this tendency is an attribute of the subjective study and should be kept.

The decisiveness of each participant varied. In figure 4.10, the differences among participants are depicted. The decisiveness is represented by the ratio of certain answers (Left or Right) and less certain answers (Maybe left or Maybe right). The participants depicted in the top left corner were more likely sure of

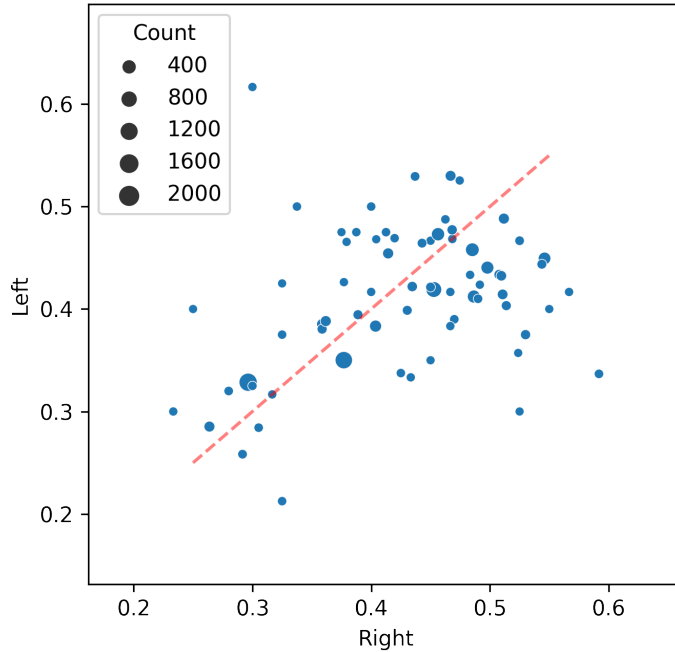


Figure 4.9: Distribution of annotation biases. The data points are participants; the x and y axis is part of the answers from them with Left/Maybe left and Right/Maybe right options, respectively. The size of the data point depicts the annotation count. Only participants with at least 30 annotations are shown.

their choices than the participants in the bottom right corner. The participants closer to the bottom left corner used the I don't know option more frequently. About half (34 out of 69) of the participants used the decisive options relatively same frequently as the less decisive options, meaning that the difference between the decisive part and the less decisive part of the annotations was less than 0.2. This indicates that some participants were less sure in general, and their annotations might be less accurate. This bias will be addressed simply by giving certain answers (Left and Right) more weight.

To detect the assimilation and contrast [81] bias, some preprocessing needs to be done. Simply comparing the previous and next annotations would not suffice. The reason is the last bias decisiveness of the users. There would be a high number of successive same decisive annotations (Left then Left, Maybe right then Maybe right, etc.) even though this does not have to mean the assimilation bias but rather the participants' decisiveness. After filtering participants with a low number of annotations, the lift score was computed for each successive annotation per user

$$lift(i, j) = \frac{P(i|j)}{P(i)}$$

. The $P(i|j)$ denotes the probability of the participant selecting the i-th option when the last option was j-th, and $P(i)$ denotes the probability of the participant selecting the i-th option without knowing the previous answer. The lift scores higher than two and lower than 0.5 were picked, and their counts are depicted

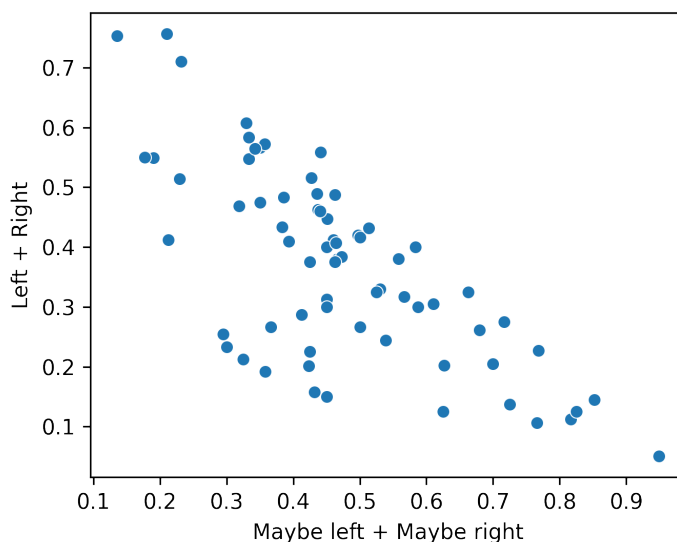


Figure 4.10: The participants’ decisiveness distribution. The data points are participants; the x and y axis is part of the answers with less decisive (Maybe left + Maybe right) and more decisive (Left + Right) options, respectively. The size of the data point depicts the annotation count. Only participants with at least 30 annotations are shown.

in 4.11. In this statistic, 31 participants were studied, and 26 of them did not have any lift scores out of the selected bounds. One participant had some values higher than the upper bound. Three participants had some values lower than the lower bound. Lastly, one participant had some values higher than the upper bound and some values lower than the lower bounds. The lift scores of the last participant are depicted in figure 4.12.

From the summarization counts in figure 4.11 and from the lift score of the most extreme participant in terms of lift scores in figure 4.12, it is evident that a few participants might be influenced by the assimilation and contrast bias. However, the number of participants affected is relatively small and more importantly, the bias affects solely the sureness of the annotation and not the actual direction (left vs right options). Therefore the impact is insignificant for the similarity models evaluation.

4.5 Triplet decisivness

Image similarity is presumably subjective, and therefore it is desired to know the limitations of image similarity modelling. The repeating subset was designed to answer that and obtain multiple annotations from distinct participants for each triplet. These annotations give us some insight into the triplet similarities and hardness to asses the similarity.

The agreement on what is similar in the triplet varied. The triplets with at least four annotations were taken in count. The participants agreed on 71.5% of the triplets, meaning the part of votes for the left or right options was higher than 0.5. The rest of the triplets did not have enough votes for any of the two

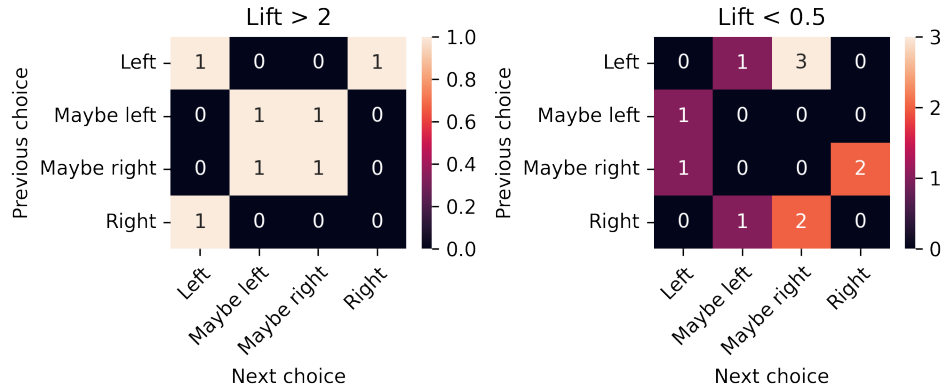


Figure 4.11: Left: Number of successive annotations with the lift score higher than 2. Right: Number of successive annotations with the lift score less than 0.5.

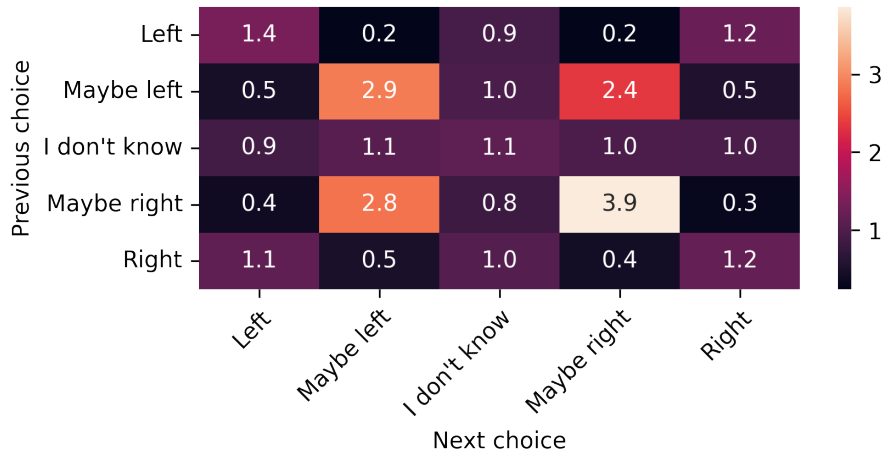


Figure 4.12: Lift scores for the participant with the highest number out of bound lift scores.

options, and thus the participants could settle on neither. However, it did not always mean that the participants were going for the I don't know or the unsure options. Therefore the rest were also additionally divided into two categories. The first one was an uncertain category. These triplets had more than 25% I don't know annotations and made 21.7% of these repeating annotations. The last part of 6.8% triplets did not get enough votes on either side and simultaneously did have less than 0.25 part of I don't know annotations. These triplets had rather polarizing opinions on what is actually similar. The distribution is depicted in 4.13.

To better understand how the similarity works and gain some intuition, some samples will be examined closely. The first examined triplet is shown in figure 4.14. This triplet is an example of images that usually do not appear in the standard image dataset (described in 2.1) because they capture some moment or action. Even though the objects in the images can be uncertain and the captioning may be challenging, the participants did agree convincingly. The participants might decide on the similarity of the "bubbly" part of the images. However, few participants voted for the Maybe right option. The presence of a

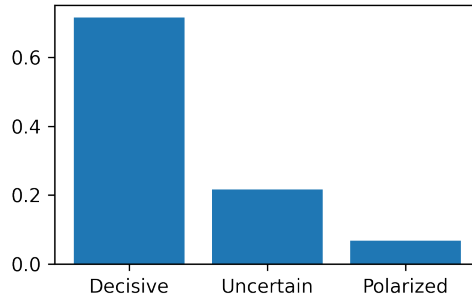


Figure 4.13: Distribution of different decisiveness classes

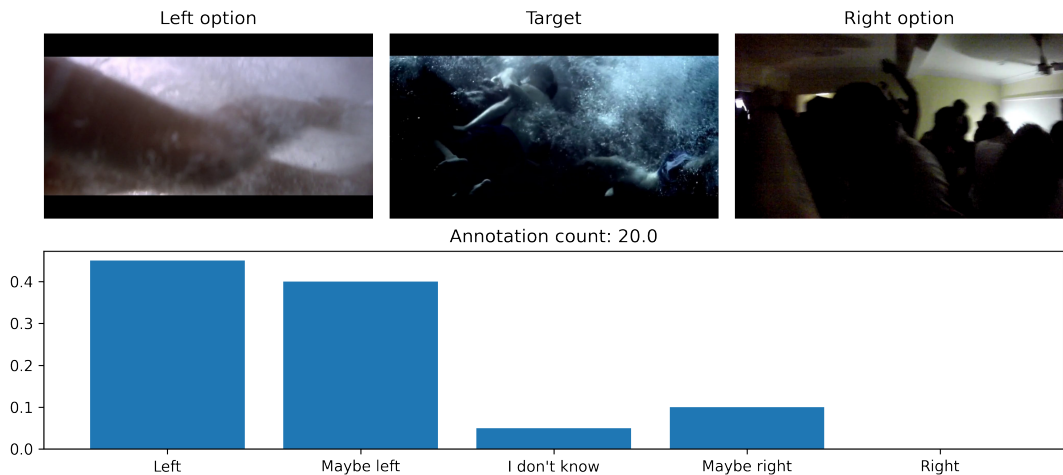


Figure 4.14: Top: An example of a decisive triplet. Bottom: Distribution of annotations.

similar dark part of the images might explain this.

On the other hand, the left option and target image in figure 4.15 are more easily describable and could even be present in the standard image datasets compared to the previous triplet and the right option of this triplet. However, in this case, the participants could not decide if a sports car was more similar to a raised view of a beach or a blurry moving hand over a paper. Slightly more participants voted for the Maybe left than the Maybe right option. The slight shift could mean that some participants may use low-level visual features such as blurriness as a minor deciding criterion. Even though these images may seem too random to determine a similarity between them, they may represent an initial part of the exploration process.

The last triplet displayed in figure 4.16 is polarizing. The participants could settle on neither option and moreover, they did not go for the neutral option in most of the cases. These three images were presumably from the same motocross racing video and thus semantically genuinely close. This triplet is a fine example of pictures from the exploitation part. Therefore the similarity uncertainty can sometimes be hidden even from human users who are sure of their choice. Moreover, this is an example of the subjectivity of image similarity and affirms the potential limitations of the image similarity approximation.

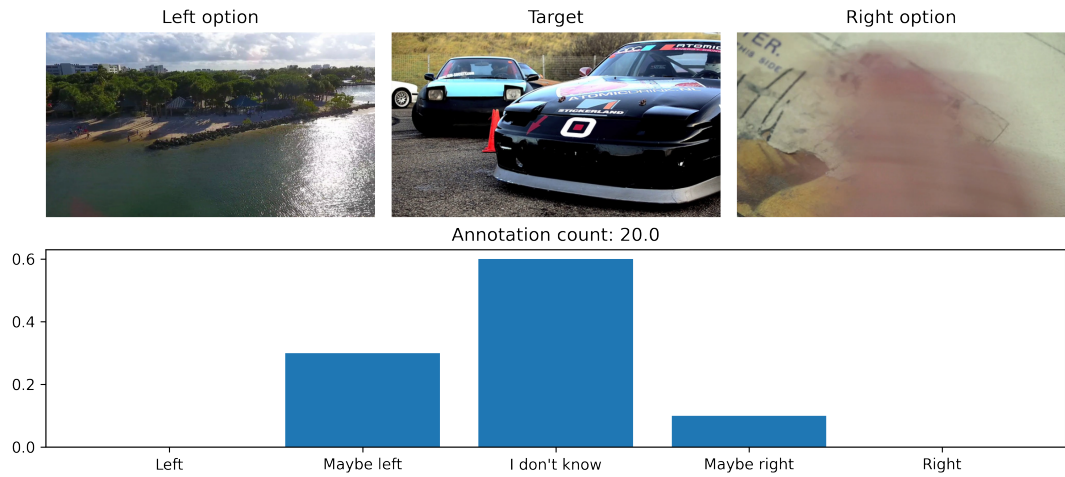


Figure 4.15: Top: An example of an uncertain triplet. Bottom: Distribution of annotations.

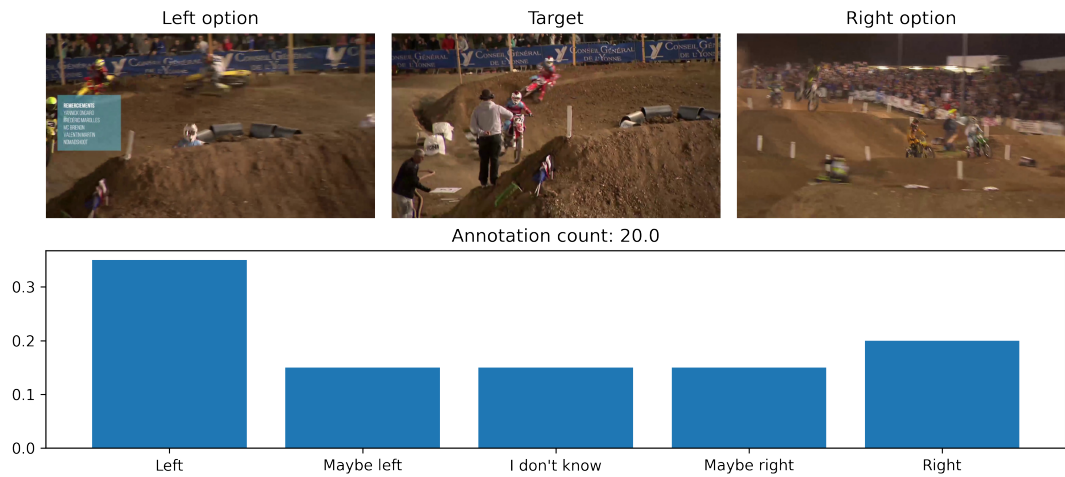


Figure 4.16: Top: An example of a polarized triplet. Bottom: Distribution of annotations.

4.6 User agreement

As explored in the previous section 4.5, the participants did not always agree with each other on every triplet. This is understandable, given the subjective aspect of the image similarity. However, the previous section examined agreement from a triplet perspective. This agreement can also be examined from the perspective of the participants.

The agreement depicted in figure 4.17 shows soft agreement for each participant. The y-axis represents a soft agreement with the other users on triplets that both users annotated. The users' agreement refers to choosing the same option on the same triplet. However, this "hard" agreement says that two users, one selecting the Left option and the second selecting the Maybe left options, do not agree with each other. Intuitively, this does not seem right, and those participants would agree with each other in reality. Therefore this thesis works with a soft agreement, which treats the more and less certain options as equals

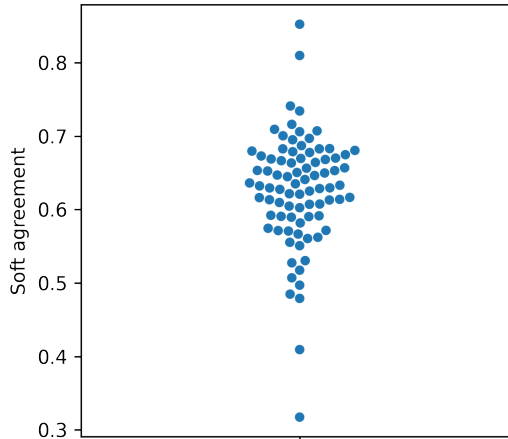


Figure 4.17: Soft agreement among users on the same triplets. The less and more certain options were merged (Left == Maybe left; Right == Maybe right) for the soft agreement computation.

(Left == Maybe left; Right == Maybe right). The mean agreement is 0.626, and the median is 0.633. There are four visible outliers, two with relatively high and two with a relatively low soft agreement with the other participants. The two outliers with high soft agreements are those with only 20 annotations. Therefore the value was computed on a relatively small number of triplets. The responses from the participants with the lowest agreement were validated manually to detect possibly unreliable annotations. The outlier with the lowest agreement was probably not paying attention to the study after a few annotations and started randomly selecting Maybe left and Maybe right options only. After a thorough inspection of the annotations and finding at least three triplets where the target image and one of the options were near-duplicates, and the other option was utterly different, it was concluded that this participant’s annotations were unreliable and would not be taken into count while evaluating similarity models. The second outlier from the bottom seems to be reliable because the responses did not show any objectively unreliable annotations. However, this participant did submit quite a lot I don’t know options (36.7%), together with a relatively low number of annotations (60). Therefore their soft agreement was low.

The overall participants’ soft agreement being far from the 100% and the presence of some polarizing triplets discussed in section 4.5 may lead us to a question, what are the most common disagreements? A disagreement matrix is depicted in figure 4.18, inspired by a confusion matrix in machine learning. On the left side, a disagreement matrix is depicted, and on the right side is a variant with soft disagreement. It is evident that the participants mostly agreed on the more certain options, which might be due to the existence of decisive triplets. The most common disagreement is between more and less certain options. This disagreement rate is likely to be caused by the different decisiveness among the participants. Interestingly the annotations with the neutral or less certain options have disagreed the most. The soft disagreement matrix shows that only about in 14% annotation pairs the participants disagreed with opposite options.

The agreement among participants differs to some extent. The differences might be explained by task difficulty or differences among the participants. The

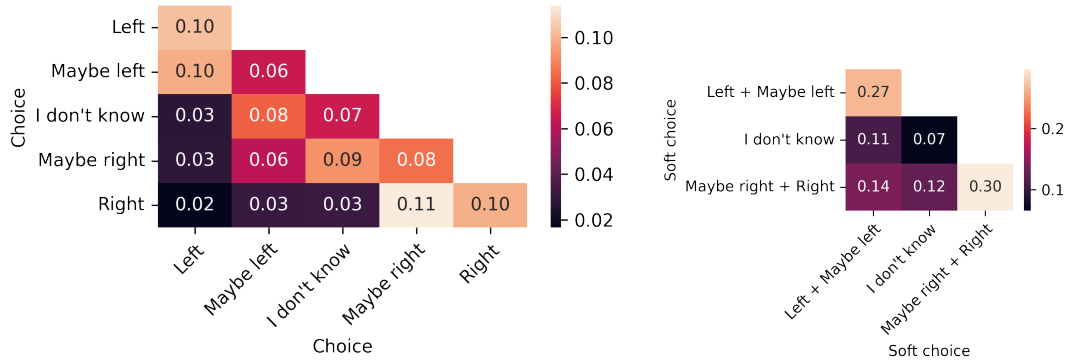


Figure 4.18: Left: Disagreement matrix of participants. Right: Soft disagreement matrix of participants.

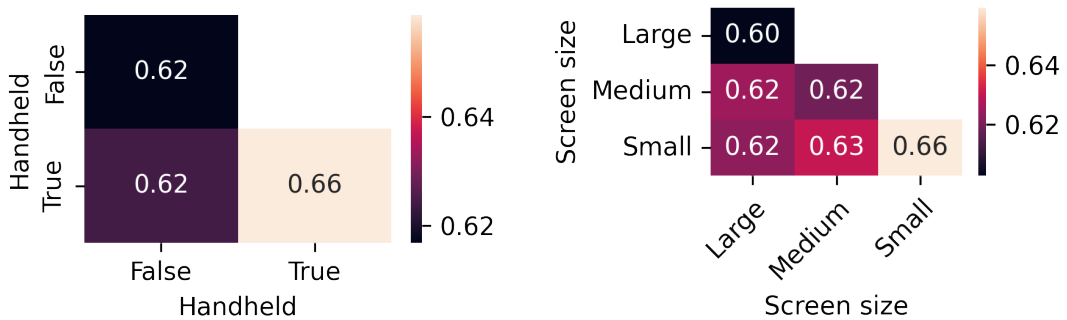


Figure 4.19: Soft agreement among different device types. papers with code speech to text

differences might be explained by different demographic attributes or different device types.

The agreements among different demographic groups are depicted in figure 4.20. Missing values are caused by the lack of the same annotated triplets by participants from both groups. In the first heatmap, the participants are grouped by their education. The highest agreement is between participants with primary school and PhD. This observation may be caused by a low number of participants in the primary school education group. The second highest agreement is in the group with a PhD. Other agreements are generally close to the overall mean agreement of 0.62.

The second demographic attribute is age group. The highest agreement is among participants in the group 37-50. The lowest agreements are with groups 51-65 and any other. Again the low number of participants in this group probably caused the slight shift. However, the other agreements are still close to the overall mean.

The last demographic attribute is machine learning expertise. The largest extreme is in this demographic group, with those participants with a big machine learning expertise. This agreement is about 12% points higher than the average. This might indicate a bias among these participants that was induced by their specialization.

The different device types and their impact is depicted in figure 4.19. The

device type does not seem to have any significant effect on the agreement among the participants. The most extreme one is the soft agreement among those who used a handheld device. However, this extreme is rather insignificant in comparison to the differences among demographic attributes. Moreover, this leads to the conclusion that the screen size and device type do not significantly impact the soft agreement.

4.7 Model agreement

In this thesis, 30 similarity models were challenged to model image similarity as accurately as possible to the human annotations. The models were presented in the section 3.2. This thesis included low-level extractors based on colour models or SIFT features, which were popular in the pre-deep learning era, and state-of-the-art deep learning models trained on various domains.

Two minor distinctions to the soft agreement will be made in this section. The first one is because the preprocessing, which will be described in the following chapter 5, will merge annotations of the identical triplet to a single data with their annotation averaged with the following weights: Left = -1 , Maybe left = -0.5 , I don't know = 0 , Maybe right = 0.5 , and Right = 1 . Secondly, all those data points with average annotation = 0 will be omitted because the image similarity models, by default, can't predict the I don't know option. This metric will be called binary agreement (BA).

The second metric used for the model comparison is a weighted version of the binary agreement. The weighted binary agreement (WBA) uses the same form as the BA, except each data point is weighted by the absolute value of the average annotation. This metric prioritizes the triplets that were more agreed on.

Both metrics are depicted in equation 4.1. The $S \mapsto [-1, 1]$ denote the similarity function, which returns the similarity score. The cosine similarity on the feature vectors was used. The D represents the set of annotated triplets. Each element in the D is represented by four values: q is the query image, $o1$ is the left option, $o2$ is the right option, and a is the averaged annotation value.

$$\begin{aligned}
 BA(S, D) &= \frac{\sum_{(q,o1,o2,a) \in D} 1 + \text{sign}(S(q, o2) - S(q, o1)) \cdot \text{sign}(a)}{2 \cdot |D|} \\
 WBA(S, D) &= \frac{\sum_{(q,o1,o2,a) \in D} |a| \cdot (1 + \text{sign}(S(q, o2) - S(q, o1)) \cdot \text{sign}(a))}{2 \cdot \sum_{(q,o1,o2,a) \in D} |a|}
 \end{aligned} \tag{4.1}$$

The BA and WBA matrices are depicted in figures 4.21 and 4.22, respectively. The figures show the metric value in a heatmap, with rows representing models and columns representing a specific subset. The first column is triplets from the general category, the second from the repeating category, the next two are from subsets scuba and wedding, and the last one shows the metric on the whole dataset. The W2VV++ feature extraction model dominates both metrics. This model even achieved the highest metric values in all subcategories. The metric differences between the W2VV++ and the EfficientNetB0 are significant in comparison to most of the differences among the other top models. Different training data might explain the difference.

The deep-learning models performed significantly better than the low-level models. However, none of the tested models was worse than a random guess. The best-performing low-level model was LABPostional 4×4 . In scuba subset achieved even a higher WBA than both variants of ImageGPT. The low-level models were generally much more successful in the scuba dataset. The higher agreement level may indicate that the participants relied on simple visual attributes in this subset more often than in the others.

ImageGPT performed significantly worse than other deep-learning models. Moreover, this model achieved similar or even worse agreement in the scuba subset than some low-level models. This network’s authors claimed that the model produces high-quality vector representations of images, and thus this result was quite surprising. The Transformers architecture is often more successful than deep CNNs in many domains, as discussed in section 3.2. However, the highest agreement models were mostly CNN-based, even when trained on the same datasets.

The second phenomenon is that the deeper models of the same architecture are usually more precise. However, in image similarity, the opposite is more likely. The largest EfficientNet achieves one of the worst deep-learning BA and WBA results. Yet, every smaller version of this architecture achieves better results than any larger one. Furthermore, the smallest version of this network is the second-best performant model on the whole dataset. A similar pattern can be observed in the ResNet architecture, where the largest ResNet 152 performs the worst results of these three. The Transformers follow the same arrangement, with the smaller version of ImageGPT being slightly more performant than the larger version and the ViT base model being superior to the large one.

The previous section 4.6 shows us that there is an upper bound in terms of the maximal agreements. However, the soft agreement is not directly comparable to the binary and weighted binary agreements. Therefore the upper bound for the models will need to be computed based on the user annotations. Nevertheless, every subset except the repeating does not have enough annotations on the identical triplets for the evaluation. Therefore similarly to the previous section, only annotations from the repeating subset will be considered. Those filtered annotations evaluate the participants’ choices as they were the models’ similarities. The participants’ binary agreement is 0.881, and the weighted binary agreement is 0.946.

The approximated upper bound shows a possibility of improving both agreements. The best-performing model lags behind the participants by 6.6 per cent points on the binary agreement and 6.5 per cent points on the weighted binary agreement.

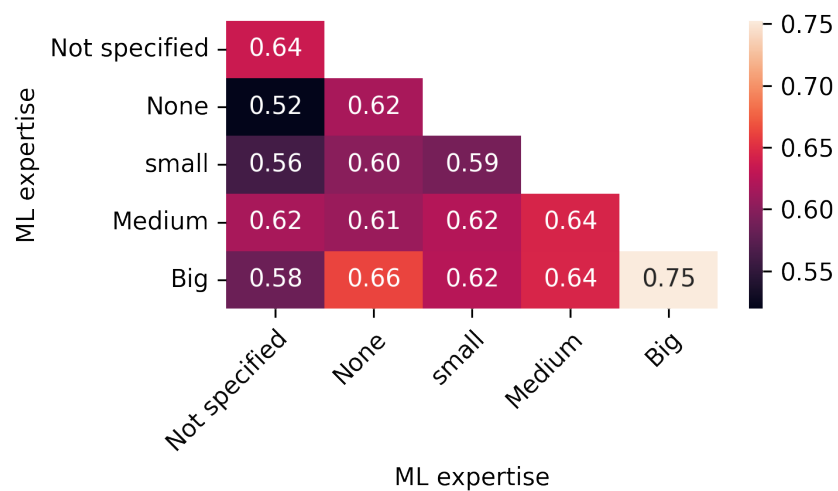
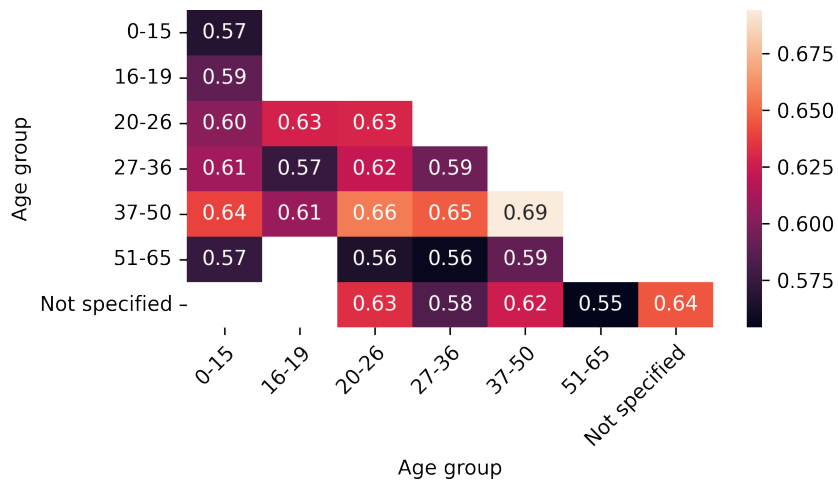
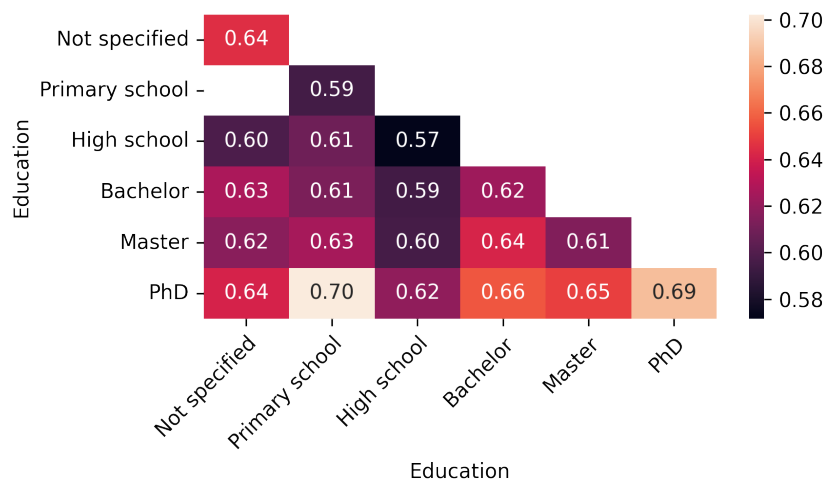


Figure 4.20: Soft agreement among demographic groups.

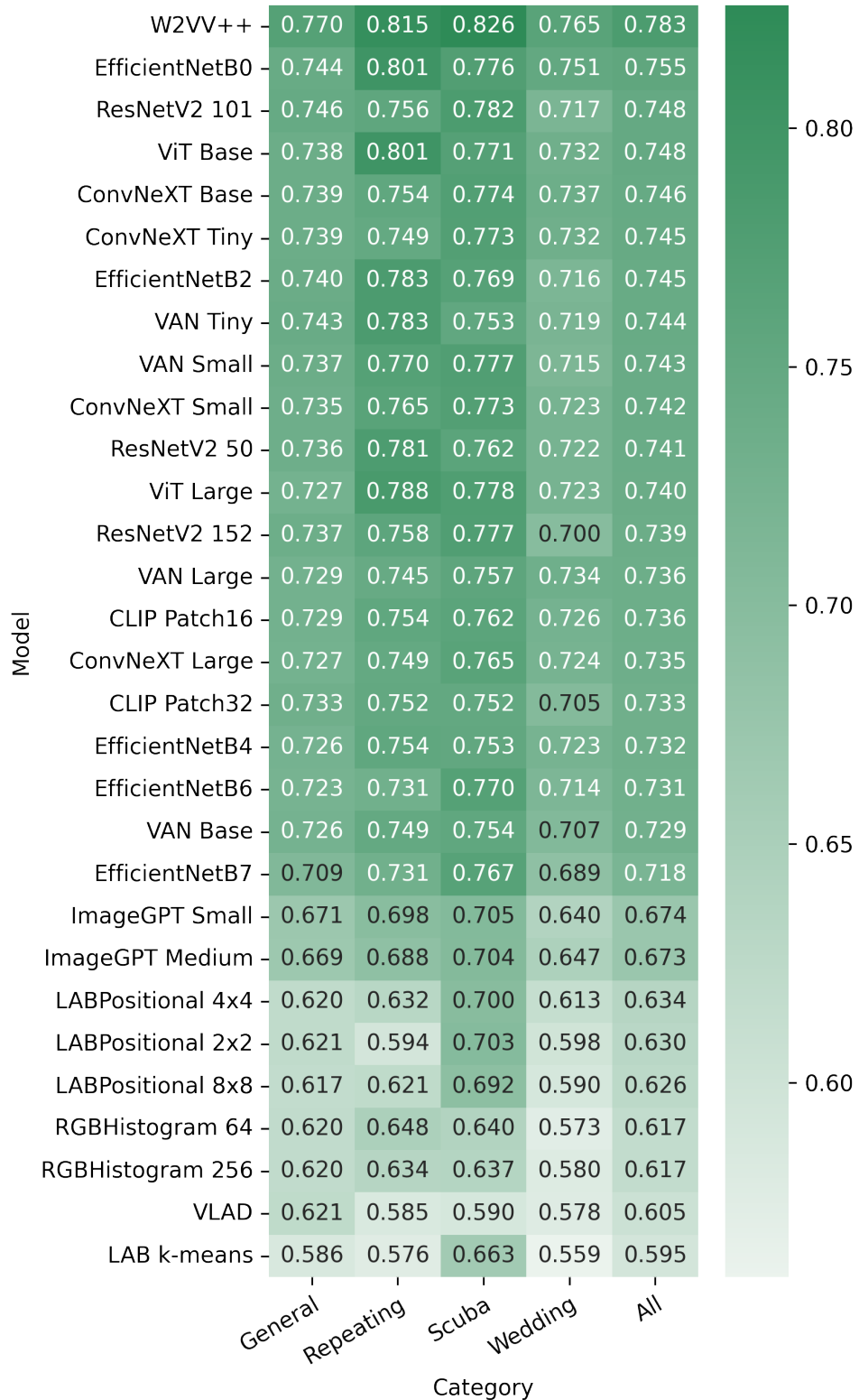


Figure 4.21: Similarity models compared by binary agreement with the human annotations. The agreements are divided into five groups on the x-axis. The models are sorted by agreement in the All column.

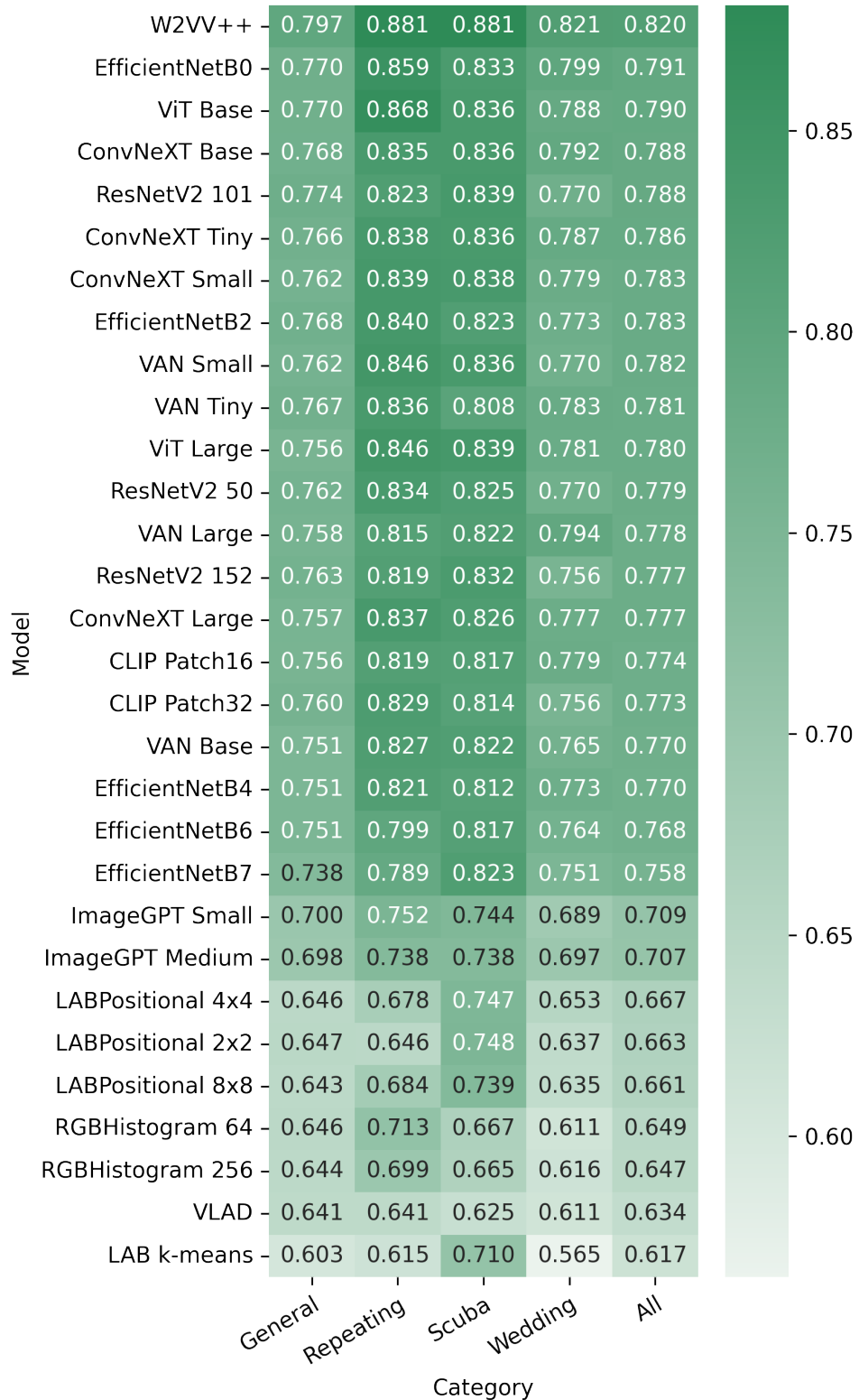


Figure 4.22: Similarity models compared by weighted binary agreement with the human annotations. The agreements are divided into five groups on the x-axis. The models are sorted by agreement in the All column.

5. Similarity model improvement

The previous chapter 4 explored the relations among the participants' choices by their device types, demographic attributes and impact of the gamification elements. The primary outcome built on in this chapter is the performance of the deep learning models. Even though they provide plausible similarity feature vectors for the images, they still fall behind the performance of the human annotators.

5.1 Preprocessing

The annotation paradigm used in the user study allowed, and even in the repeating subset encouraged, multiple annotations by different participants on a single triplet. Thus straightforward usage of these annotated triplets would create an imbalance, and some triplets would be trained for more time than others. Moreover, the same triplet could appear in the training set with different annotations, and this could cause an issue with the training convergence. Therefore the annotations for each triplet were average with assigned values: Left = -1 , Maybe left = -0.5 , I don't know = 0 , Maybe right = 0.5 , and Right = 1 . This preprocessing provides additional information on how similar the favoured option is.

Using this preprocessing, a different metric was proposed in section 4.7. The binary agreement and weighted binary agreement were created, and their formula is in equation 4.1. The models do not provide the ability to assess the I don't know option, and thus the triplets with an average of zero rating would always be incorrect. Therefore, these metrics are evaluated on the triplets with an average rating different from zero.

5.2 Setup

With the prepared dataset, the loss function needs to be defined prior to the training. The fine-tuned network will be used to feed forward all three images from each triplet and then compute the cosine similarity of the query feature vector and each option feature vector. One of the widespread loss functions is, for example, mean square error or mean absolute error. This may provide some results because the differences between the cosine similarities could be the output, and therefore it could be interpreted as a regression task. However, the goal is not to model the user's exact selection but rather estimate what is more similar. For example, the mean square error for a triplet with an average rating of 0.25 and the two guesses -0.2 and 1.0 would be lower for the first guess, even though the first one is incorrect. Therefore a better-suited loss function needs to be selected.

Triplet ranking loss [17, 40] was proven to achieve state-of-the-art in the image domain. In contrast to the mean square error, this loss tries to move positive examples closer to the target and the negative examples further. Nonetheless, this dataset's annotations are rather fuzzy and thus, some triplets need to be adjusted with greater weight. Even some triplets may need to move both options

to a closer relative distance to adequately reflect the similarities. Therefore some adjustments were made to address these points. The triplet fuzzy ranking loss is depicted in equation 5.1. The y is an absolute average participants' choice, and the s_c and s_f are cosine similarities of query image and image which should be closer and further, respectively. The m is a margin that should divide the two similarities, and it was set to 0.2.

$$\mathcal{L} = \max(0, (m + s_f - s_c) * y) + \max(0, (|s_f - s_c| - m) * (1 - y)) \quad (5.1)$$

The collected dataset is rather small compared to the other datasets, e.g. ImageNet [12]. Therefore 5-fold cross-validation was employed on the training set to get more reliable results. Moreover, four different training sets were used for 5-fold cross-validation to observe a possible generalization. The first training set included only the general subset (denoted as G in the results), and the rest was used only for testing. The second and third training sets included the general subset with the wedding (G + W) and scuba (G + S) subset, respectively. Again the rest was used for testing. The last training included general, wedding, and scuba subsets (G + S + W).

The overfitting can poorly influence the training results, and thus some regularization techniques were employed. Random zoom with maximal $\pm 20\%$ possible change, random horizontal flip, and random rotation with maximal $\pm 20\%$ change were employed. These augmentations were modest because more aggressive changes could influence the overall image similarity. Therefore, randomizing brightness, contrast, saturation, and hue leads to poor results.

The fine-tuned model was the best performing one, i.e. W2VV++. This model consists of ResNet-152 and ResNeXt-101, a concatenation layer of their results, and a dense layer with tanh activation resulting in a 2048 dim. vector. Most of the part of the W2VV++ was frozen during the training, and only the last layer was updated. Some tests with unfrozen deeper layers were made; however, they overfitted quickly and performed poorly. The test was made on a single fold with a solely general subset as the training set, and the model with unfrozen deeper layers achieved 0.4% lower binary agreement and 0.2% lower weighted agreement.

The finetuning was done in Tensorflow 2 framework¹. The used optimizer was AdamW [46] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay equals to 0.0001. The cosine decay [45] scheduling function was used as the learning rate with the starting learning rate equal to 0.0001 and parameter $\alpha = 0$. The training was run for a maximum of 200 epochs, and the model with the highest validation WBA was selected. The number of needed epochs for the best model varied between a few epochs (less than 10) and, at most, 166 epochs.

The whole evaluation and fine-tuning pipeline is attached to the electronic version, and the user manual is in appendix A.3.

5.3 Results

The training results on the left folds from the cross-validation increased significantly. They are depicted in table 5.1. The four train sets were employed. The

¹<https://www.tensorflow.org/>

	Fine-tuned W2VV++	Baseline W2VV++	
Train set	BA	BA	p-value
G	0.788	0.771	0.054
G + S	0.799	0.785	0.007
G + W	0.789	0.769	0.007
G + S + W	0.797	0.781	0.019
Train set	WBA	WBA	p-value
G	0.816	0.798	0.034
G + S	0.83	0.816	0.02
G + W	0.82	0.802	0.003
G + S + W	0.832	0.817	0.015

Table 5.1: Mean binary agreement (BA) and weighted binary agreement (WBA) on the left one fold out from the cross-validation.

	Fine-tuned W2VV++	Baseline W2VV++	
Training set	BA	BA	p-value
G	0.81	0.814	0.209
G + S	0.817	0.814	0.497
G + W	0.814	0.814	0.921
G + S + W	0.817	0.814	0.433
Train set	WBA	WBA	p-value
G	0.875	0.88	0.197
G + S	0.881	0.88	0.869
G + W	0.877	0.88	0.403
G + S + W	0.882	0.88	0.428

Table 5.2: Mean binary agreement (BA) and weighted binary agreement (WBA) on the repeating set.

most significant improvement can be observed in the general and wedding sets, where the binary agreement increased by 0.02 and the weighted binary agreement by 0.018 with statistical significance. The Student’s t-test was used as a statistical test, and all the cross-validation results except one (binary agreement on the general set) had a p-value lower than 0.05.

The Fine-tuned network on the repeating set did not achieve any major improvement. See table 5.2. A slight improvement was achieved on the general + scuba and general + scuba + wedding training sets. However, this improvement is not statistically significant. The results were the same or slightly lower in the other training sets. Again without statistical significance.

The binary and weighted binary agreements were significantly improved on the wedding set, shown in the table 5.3. The binary agreement increased by 0.022 and 0.023 with the training set general and general + scuba, respectively. Similar increases, 0.017 and 0.018, can be observed in the weighted binary agreement. All the increases have a p-value lower than 0.05. Interestingly, even though the scuba

	Fine-tuned W2VV++	Baseline W2VV++	
Training set	BA	BA	p-value
G	0.787	0.765	0.012
G + S	0.788	0.765	0.003
Train set	WBA	WBA	p-value
G	0.838	0.821	0.015
G + S	0.839	0.821	0.006

Table 5.3: Mean binary agreement (BA) and weighted binary agreement (WBA) on the wedding set.

	Fine-tuned W2VV++	Baseline W2VV++	
Training set	BA	BA	p-value
G	0.829	0.827	0.463
G + W	0.829	0.827	0.405
Train set	WBA	WBA	p-value
G	0.885	0.882	0.223
G + W	0.885	0.882	0.234

Table 5.4: Mean binary agreement (BA) and weighted binary agreement (WBA) on the scuba set.

set is unrelated to the wedding set, including it in the training data resulted in slightly better improvement.

The scuba set was more challenging for the fine-tuned network. The results can be seen in the table 5.4. The improvements are minor and thus not statistically significant. In this set, adding the wedding set to the training did not increase performance as in the previous results.

Overall the fine-tuned network had 11 statistically significant improvements, nine minor improvements, one same result, and three minor deteriorations.

Conclusion

Similarity modelling is an essential part of many state-of-the-art image and video retrieval tools. There is a plethora of research in this domain, including competitions, e. g. VBS, LSC, and TRECVID. These retrieval tools are focused on retrieving a known scene or satisfying a user’s need to find any scene fulfilling their criteria. The search stages and usage of the tools’ features may differ. The early stage is called exploration, and the users often use some kind of text search or overview visualisations exploiting image similarity information. In the later stage, called exploitation, the system presents more focused visualisations, exploiting the gained knowledge about the task, and users usually use some kind of relevance feedback or nearest neighbour search. Therefore both these stages rely on trustworthy image similarity approximation techniques.

This thesis explored the possibilities of automatic similarity modelling and their possible improvements. To accomplish this goal, a comprehensive user study was conducted. The base data for the study were generated by a complex and yet simple-to-use pipeline. The data consisted of triplets with different expected levels of similarity. These different levels mirrored the various stages of the search process, i.e. exploration vs. exploitation. On top of that, two subsets of the whole dataset were examined separately. These were wedding and scuba videos. Additionally, some triplets from the general category were intentionally sampled more often than others to get more detailed similarity information on some of the triplets. These triplets were referred to as the repeating subset.

A total number of 84 participants took part in this study and resulting in 17365 unique annotations. The participants in each trial were asked to select the most similar candidate to the query image. Most participants used a laptop or PC, yet more than twenty operated a handheld device. The participants came from various age groups, education levels, and machine learning expertise. Employment of the gamification elements in this study proved to help receive more annotations with statistical significance. Moreover, the study did not indicate any major biases that would change the relative outcome of each annotation. However, this thesis shows strong evidence that some triplets were hard to predict, even for human annotators. The hardness can be perceived by the participants most of the time, but in some cases, the participants were unaware of the hardness and thus, some triplets were polarising. The disagreement among the users was relatively small, and they mostly agreed on one option.

The agreement between similarity models and the participants differed in many ways. One of the primary outcomes confirms the overall superiority of the deep learning models against the low-level models. However, the low-level models performed better than a random guess by a substantial margin. Also, the difference among the deep learning models was substantial in some cases. Moreover, smaller models of the same architecture often performed better than the larger ones. The subset categories, i.e. general, repeating, scuba, and wedding, introduced a second type of difference. This difference was observed in agreements for each model. Moreover, it even caused an incomparability of some model pairs, where one achieves better results in one category and the second one in another category. However, the results of W2VV++ were superior in every category.

Despite the W2VV++ performance being better than the others, there seems to be a space for improvement, given the results of the inter-user agreement. Therefore, we fine-tuned the W2VV++ using cross-validation, and we managed to improve the results in some areas significantly. For instance, in the general category, the improvement was 1.7% points in the mean binary agreement and 1.8% points in the mean weighted binary agreement.

However, some limitations were observed in this thesis. The first one is the number of annotated triplets. Despite the improvements shown, a humongous amount of data is needed for a more thorough fine-tuning of deep neural networks (e.g. incorporating more layers). This was out of our possibilities even with motivational techniques, e.g. gamification. The second limitation is the number of participants and their composition. Some social groups were partially neglected in the study. Therefore conducting a large-scale user study in this form is one of the main future work directions.

Another possible enhancement of the similarity modelling and future work may be done via the employment of a model ensemble. This may increase the agreement with the human annotations. Moreover, the multi-valued annotations could be used in finer detail and train the models to predict both similarity and the hardness of the task.

The created dataset may serve additional purposes that do not enhance the image similarity models directly. One of the common postprocessing of the feature vectors is some dimensionality reduction. This dataset may be employed as a dimensionality reduction benchmark or be directly included in the dimensionality reduction algorithm. Furthermore, it might serve as a benchmarking and validation for relevance feedback algorithms or visualisations.

This thesis's outcomes can be directly applied to the image and video retrieval tools. They can help researchers choose from evaluated feature extractors, add more similarity models for evaluation, or utilise the fine-tuned W2VV++ with improved performance. Note that due to licencing of W2VV++, the fine-tuned weights will be available upon request.

Bibliography

- [1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark, 2016. URL <https://arxiv.org/abs/1609.08675>.
- [2] Rolph E Anderson. Consumer dissatisfaction: The effect of disconfirmed expectancy on perceived product performance. *Journal of marketing research*, 10(1):38–44, 1973.
- [3] Stelios Andreadis, Anastasia Moutzidou, Konstantinos Gkountakos, Nick Pantelidis, Konstantinos Apostolidis, Damianos Galanopoulos, Ilias Gialampoukidis, Stefanos Vrochidis, Vasileios Mezaris, and Ioannis Kompatsiaris. Verge in vbs 2021. In *MultiMedia Modeling*, pages 398–404. Springer, 2021. ISBN 978-3-030-67835-7.
- [4] George Awad, Keith Curtis, Asad A. Butt, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Yvette Graham, , and Georges Quénot. An overview on the evaluated video retrieval tasks at trecvid 2022. In *Proceedings of TRECVID 2022*. NIST, USA, 2022.
- [5] Judit Bar-Ilan, Kevin Keenoy, Mark Levene, and Eti Yaari. Presentation bias is significant in determining user preference for search results—a user study. *Journal of the American Society for Information Science and Technology*, 60(1):135–149, 2009. doi: <https://doi.org/10.1002/asi.20941>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20941>.
- [6] Fabian Berns, Luca Rossetto, Klaus Schoeffmann, Christian Beecks, and George Awad. V3c1 dataset: An evaluation of content characteristics. In *ICMR’19*, pages 334–338. ACM, 2019.
- [7] Petra Budikova, Michal Batko, and Pavel Zezula. Evaluation platform for content-based image retrieval systems. In Stefan Gradmann, Francesca Borri, Carlo Meghini, and Heiko Schuldt, editors, *Research and Advanced Technology for Digital Libraries*, pages 130–142, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-24469-8.
- [8] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML’20*. PMLR, 2020.
- [9] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.
- [10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, volume 1, pages 886–893. Ieee, 2005.

- [11] Jonathan Delhumeau, Philippe-Henri Gosselin, Hervé Jégou, and Patrick Pérez. Revisiting the vlad image representation. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 653–656, 2013.
- [12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR'09*, pages 248–255. IEEE, 2009.
- [13] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining "gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15, 2011.
- [14] Lijun Ding and Ardeshir Goshtasby. On the canny edge detector. *Pattern recognition*, 34(3):721–725, 2001.
- [15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL <https://arxiv.org/abs/2010.11929>.
- [16] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.
- [17] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.
- [18] Roy T. Fielding and Julian Reschke. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. RFC 7231, June 2014. URL <https://www.rfc-editor.org/info/rfc7231>.
- [19] Roy T. Fielding and Julian Reschke. Hypertext Transfer Protocol (HTTP/1.1): Authentication. RFC 7235, June 2014. URL <https://www.rfc-editor.org/info/rfc7235>.
- [20] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network, 2022.
- [21] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, 8(3):331–368, 2022.
- [22] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Andreas Leibetseder, Liting Zhou, Aaron Duane, Dang Nguyen, Duc Tien, Michael Riegler, Luca Piras, et al. Comparing approaches to interactive lifelog search at the lifelog search challenge (lsc2018). *ITE Transactions on Media Technology and Applications*, 7(2):46–59, 2019.

- [23] John A Hartigan, Manchek A Wong, et al. A k-means clustering algorithm. *Applied statistics*, 28(1):100–108, 1979.
- [24] 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.
- [25] Martin N. Hebart, Charles Y. Zheng, Francisco Pereira, and Chris I. Baker. Revealing the multidimensional mental representations of natural objects underlying human similarity judgements. *Nature Human Behaviour*, 4(11): 1173–1185, Nov 2020. ISSN 2397-3374.
- [26] Silvan Heller, Viktor Gsteiger, Werner Bailer, Cathal Gurrin, Björn Jónsson, Jakub Lokoč, Andreas Leibetseder, František Mejzlík, Ladislav Peska, Luca Rossetto, Konstantin Schall, Klaus Schoeffmann, Heiko Schuldt, Florian Spiess, Duyen Tran, Lucia Vadicamo, Patrik Veselý, Stefanos Vrochidis, and Jiaxin Wu. Interactive video retrieval evaluation at a distance: comparing sixteen interactive video search systems in a remote setting at the 10th video browser showdown. *International Journal of Multimedia Information Retrieval*, 11, 03 2022. doi: 10.1007/s13735-021-00225-2.
- [27] Nico Hezel, Konstantin Schall, Klaus Jung, and Kai Uwe Barthel. Video search with sub-image keyword transfer using existing image archives. In *MultiMedia Modeling*, pages 484–489. Springer, 2021. ISBN 978-3-030-67835-7.
- [28] Nico Hezel, Konstantin Schall, Klaus Jung, and Kai Uwe Barthel. Efficient search and browsing of large-scale video collections with vibro. In Björn Þór Jónsson, Cathal Gurrin, Minh-Triet Tran, Duc-Tien Dang-Nguyen, Anita Min-Chun Hu, Binh Huynh Thi Thanh, and Benoit Huet, editors, *MultiMedia Modeling*, pages 487–492, Cham, 2022. Springer International Publishing. ISBN 978-3-030-98355-0.
- [29] Katja Hofmann, Anne Schuth, Alejandro Bellogín, and Maarten de Rijke. Effects of position bias on click-based recommender evaluation. In *Advances in Information Retrieval*, pages 624–630, Cham, 2014. Springer. ISBN 978-3-319-06028-6.
- [30] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR’10*, pages 3304–3311. IEEE, 2010.
- [31] Björn Þór Jónsson, Omar Shahbaz Khan, Dennis C Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. Exquisitor at the video browser showdown 2020. In *International Conference on Multimedia Modeling*, pages 796–802. Springer, 2020.
- [32] Maryam Karimzadehgan and ChengXiang Zhai. Exploration-exploitation tradeoff in interactive relevance feedback. In *19th ACM International Conference on Information and Knowledge Management, CIKM ’10*, page 1397–1400. ACM, 2010. ISBN 9781450300995.

- [33] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019.
- [34] Markéta Křenková, Vladimír Mic, and Pavel Zezula. Similarity search with the distance density model. In *SISAP’22*, pages 118–132. Springer, 2022.
- [35] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- [36] 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_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
- [37] Carol L Krumhansl. Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. 1978.
- [38] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [39] Fei-Fei Li, Marco Andreeto, Marc’Aurelio Ranzato, and Pietro Perona. Caltech 101, Apr 2022.
- [40] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. W2v++ fully deep learning for ad-hoc video search. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1786–1794, 2019.
- [41] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *CVPR’16*, pages 4641–4650, 2016.
- [42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- [43] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022.
- [44] Jakub Lokoč, František Mejzlík, Tomáš Souček, Patrik Dokoupil, and Ladislav Peška. Video search with context-aware ranker and relevance feedback. In Björn Pór Jónsson, Cathal Gurrin, Minh-Triet Tran, Duc-Tien Dang-Nguyen, Anita Min-Chun Hu, Binh Huynh Thi Thanh, and Benoit Huet, editors, *MultiMedia Modeling*, pages 505–510, Cham, 2022. Springer International Publishing. ISBN 978-3-030-98355-0.

- [45] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. URL <http://arxiv.org/abs/1608.03983>.
- [46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.
- [47] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [48] Tzu-Chuen Lu and Chin-Chen Chang. Color image retrieval technique based on color features and image bitmap. *Information Processing & Management*, 43(2):461–472, 2007. ISSN 0306-4573.
- [49] Chao Luo, Xiaojie Li, Lutao Wang, Jia He, Denggao Li, and Jiliu Zhou. How does the data set affect cnn-based image classification performance? In *2018 5th International Conference on Systems and Informatics (ICSAI)*, pages 361–366, 2018. doi: 10.1109/ICSAI.2018.8599448.
- [50] David L. MacAdam. Visual sensitivities to color differences in daylight*. *J. Opt. Soc. Am.*, 32(5):247–274, May 1942. doi: 10.1364/JOSA.32.000247. URL <https://opg.optica.org/abstract.cfm?URI=josa-32-5-247>.
- [51] James L McClelland, David E Rumelhart, PDP Research Group, et al. *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models*, volume 2. MIT press, 1987.
- [52] K. McLaren. The development of the cie 1976 (l*a*b*) uniform colour-space and colour-difference formula. *J. Soc. Dye. Colour.*, 92:338 – 341, 10 2008.
- [53] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.
- [54] Kurt Nassau. colour. In *Encyclopaedia Britannica*. 2022-06-28, <https://www.britannica.com/science/color>.
- [55] Ladislav Peška, Gregor Kovalčík, Tomáš Souček, Vít Škrhák, and Jakub Lokoč. W2vv++ bert model at vbs 2021. In *MultiMedia Modeling*, pages 467–472. Springer, 2021.
- [56] Ladislav Peška, Marta Vomlelová, Patrik Veselý, Vít Škrhák, and Jakub Lokoč. Evaluating a bayesian-like relevance feedback model with text-to-image search initialization. *Multimedia Tools and Applications*, Nov 2022. ISSN 1573-7721. doi: 10.1007/s11042-022-14046-w. URL <https://doi.org/10.1007/s11042-022-14046-w>.
- [57] Joshua C. Peterson, Joshua T. Abbott, and Thomas L. Griffiths. Evaluating (and improving) the correspondence between deep neural networks and human representations. *Cognitive Science*, 42(8):2648–2669, 2018.
- [58] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

- [59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL <https://arxiv.org/abs/2103.00020>.
- [60] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. URL <https://arxiv.org/abs/2102.12092>.
- [61] Brett D Roads and Bradley C Love. Enriching imagenet with human similarity judgments and psychological embeddings. In *CVPR’21*, pages 3547–3557. IEEE/CVF, 2021.
- [62] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [63] Luca Rossetto. *Multi-modal video retrieval*. PhD thesis, University_of_Basel, 2018.
- [64] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [65] 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. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252, 2015.
- [66] Konstantin Schall, Kai Uwe Barthel, Nico Hezel, and Klaus Jung. Gpr1200: A benchmark for general-purpose content-based image retrieval. In *MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6–10, 2022, Proceedings, Part I*, page 205–216, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-030-98357-4. doi: 10.1007/978-3-030-98358-1_17. URL https://doi.org/10.1007/978-3-030-98358-1_17.
- [67] T Smith and J Guild. The c.i.e. colorimetric standards and their use. *Transactions of the Optical Society*, 33(3):73, jan 1931. doi: 10.1088/1475-4878/33/3/301. URL <https://dx.doi.org/10.1088/1475-4878/33/3/301>.
- [68] Florian Spiess, Ralph Gasser, Silvan Heller, Mahnaz Parian-Scherb, Luca Rossetto, Loris Sauter, and Heiko Schuldt. Multi-modal video retrieval in virtual reality with vitrivr-vr. In Björn Þór Jónsson, Cathal Gurrin, Minh-Triet Tran, Duc-Tien Dang-Nguyen, Anita Min-Chun Hu, Binh Huynh Thi Thanh, and Benoit Huet, editors, *MultiMedia Modeling*, pages 499–504, Cham, 2022. Springer International Publishing. ISBN 978-3-030-98355-0.

- [69] Nicolae Suditu and François Fleuret. Iterative relevance feedback with adaptive exploration/exploitation trade-off. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, page 1323–1331, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311564. doi: 10.1145/2396761.2398435. URL <https://doi.org/10.1145/2396761.2398435>.
- [70] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML'19*, pages 6105–6114. PMLR, 2019.
- [71] Graham JG Upton. Fisher’s exact test. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 155(3):395–402, 1992.
- [72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL <http://arxiv.org/abs/1706.03762>.
- [73] Patrik Veselý and Ladislav Peška. Less is more: Similarity models for content-based video retrieval. In *Accepted to: Proceedings of MultiMedia Modeling 29th International Conference, MMM 2023, Bergen, Norway, January 9–12, 2023*, 2023.
- [74] Patrik Veselý, František Mejzlík, and Jakub Lokoč. Somhunter v2 at video browser showdown 2021. In *MultiMedia Modeling*, pages 461–466. Springer, 2021. ISBN 978-3-030-67835-7.
- [75] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.
- [76] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [77] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016. URL <http://arxiv.org/abs/1611.05431>.
- [78] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.
- [79] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR'16*, pages 5288–5296, 2016.
- [80] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

- [81] Xiaoying Zhang, Junzhou Zhao, and John C.S. Lui. Modeling the assimilation-contrast effects in online product rating systems: Debiasing and recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 98–106, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346528. doi: 10.1145/3109859.3109885. URL <https://doi.org/10.1145/3109859.3109885>.

List of Figures

2.1	Selected images from ImageNet on the left and V3C1 on the right.	11
3.1	15
3.2	A depiction of presentation bias. The squares represent an image from a dataset; green ones depict highly similar images to a query image, and red depict dissimilar images. Images 1 and 4 are selected by model 1 and will be chosen as a triplet.	20
3.3	UI. Top: A screenshot of the web application with a user form in Czech. Bottom: A screenshot of the web application with a user form in English.	27
3.4	UI. Top: A screenshot of the web application with credentials and a short study briefing in Czech. Bottom: A screenshot of the web application with credentials and a short study briefing in English.	28
3.5	UI. Top: Legacy UI from the preliminary study[73] Middle: A screenshot of the web application in Czech. Bottom: A screenshot of the web application in English.	29
3.6	UI. Top: A screenshot of the web application with a level-up screen in Czech. Bottom: A screenshot of the web application with a level-up screen in English.	30
4.1	Left: Number of users that used a handheld or another device at least once. Right: Average volume of per-user annotations for the device type.	31
4.2	Distribution of the count of annotations with respect to the screen resolution.	32
4.3	Distribution of the count of annotations with respect to the screen resolution.	32
4.4	Left: The distribution of users with a given age group. Right: Average volume of per-user annotations for the age group.	33
4.5	Left: The distribution of users with a given highest achieved education. Right: Average annotation count per group with a given highest achieved education.	34
4.6	Left: The distribution of users with given machine learning expertise. Right: Average volume of per-user annotations for the machine learning expertise.	34
4.7	Difference of the annotation counts between this study and our preliminary study [73].	35
4.8	Distribution of annotations per subset.	36
4.9	Distribution of annotation biases. The data points are participants; the x and y axis is part of the answers from them with Left/Maybe left and Right/Maybe right options, respectively. The size of the data point depicts the annotation count. Only participants with at least 30 annotations are shown.	37

4.10	The participants' decisiveness distribution. The data points are participants; the x and y axis is part of the answers with less decisive (Maybe left + Maybe right) and more decisive (Left + Right) options, respectively. The size of the data point depicts the annotation count. Only participants with at least 30 annotations are shown.	38
4.11	Left: Number of successive annotations with the lift score higher than 2. Right: Number of successive annotations with the lift score less than 0.5.	39
4.12	Lift scores for the participant with the highest number out of bound lift scores.	39
4.13	Distribution of different decisiveness classes	40
4.14	Top: An example of a decisive triplet. Bottom: Distribution of annotations.	40
4.15	Top: An example of an uncertain triplet. Bottom: Distribution of annotations.	41
4.16	Top: An example of a polarized triplet. Bottom: Distribution of annotations.	41
4.17	Soft agreement among users on the same triplets. The less and more certain options were merged (Left == Maybe left; Right == Maybe right) for the soft agreement computation.	42
4.18	Left: Disagreement matrix of participants. Right: Soft disagreement matrix of participants.	43
4.19	Soft agreement among different device types.	43
4.20	Soft agreement among demographic groups.	46
4.21	Similarity models compared by binary agreement with the human annotations. The agreements are divided into five groups on the x-axis. The models are sorted by agreement in the All column. . .	47
4.22	Similarity models compared by weighted binary agreement with the human annotations. The agreements are divided into five groups on the x-axis. The models are sorted by agreement in the All column.	48

List of Tables

3.1	Full list of feature extractors with their embedding dimension and trainable parameters.	26
5.1	Mean binary agreement (BA) and weighted binary agreement (WBA) on the left one fold out from the cross-validation.	51
5.2	Mean binary agreement (BA) and weighted binary agreement (WBA) on the repeating set.	51
5.3	Mean binary agreement (BA) and weighted binary agreement (WBA) on the wedding set.	52
5.4	Mean binary agreement (BA) and weighted binary agreement (WBA) on the scuba set.	52

A. Attachments

A.1 Feature extractor - User documentation

Feature extractors project is a project for easy feature extraction and dataset cleaning. This project was created for user study [73] and this thesis.

This project handles:

- Image feature extraction
- Dataset cleaning
- Triplet generation
- Converting triplets to SQL insert statements

There are two ways of running the project. First is manually installing dependencies and running the individual programs. The second way is running the whole processing pipeline in Docker¹. The second approach is much easier; however, it will not be accelerated by a GPU and lacks dataset cleaning step.

A.1.1 Manual

There are prerequisites that need to be prepared prior to the installation. Prerequisites:

- Python ≥ 3.8 && < 3.10
- Python-venv
- Pip ≥ 21

Installation - Linux

```
python3 -m venv ./venv
source venv/bin/activate
pip install -r ./requirements.txt
```

Installation - Windows

```
py -m venv venv
.\venv\Scripts\activate
pip install -r /path/to/requirements.txt
```

Extraction

Feature extraction can be done in two ways. The first way is by using the classes directly through API. The second method uses a CLI tool that takes a list of images and saves Numpy matrices into the output directory.

¹<https://www.docker.com/>

Extraction - Python

Direct usage can be used in cases that require some additional postprocessing, generating an image list dynamically, or in a real-time application.

An example of the usage can be found in `extract_images.py` or here:

```
from extractors import ResNetExtractor # Import any extractor

images_paths = []
with open("imagelist.txt") as file: # Load list of files
    images_paths = file.readlines()

extractor = ResNetExtractor("50") # Create extractor instance

image_features = extractor(images_paths) # Extract image features
# image_features = (M,N)
# M - number of images
# N - features dimension
```

Extraction - CLI

The CLI usage is suitable for easy one-time feature extraction.

```
python3 extract_images.py -e 'CIELABKMeansExtractor(k=8)' \
    'CLIPExtractor(size="small")' \
    -i ./imagelist.txt -o ./output --batch_size 16 -ev
```

Dataset cleaning

The dataset cleaning can be done using the `Dataset` class in the `manipulators` package. This class provides a plethora of techniques to visualize and delete images from the dataset and save the result.

An example of dataset cleaning can be seen in the `dataset-cleaning.ipynb`.

Triplet generation

The main method of the triplet generation reads the configuration file. Then the triplets are created accordingly. The attributes of the configuration file:

- `input_dir` - The directory with the txt and npy outputs from the feature extraction implemented in `extract_images`.
- `output_file` - Name of the output CSV file
- `targets` - Number of distinct target images. The targets will be the same for all the extractors.
- `distance_measures` - List of distance measures for the triplet generation.
- `distance_classes` - Distance classes for the triplets. Each distance class is defined with its end index. The start index is computed as the previous end index + 1.
- `videos_filter` - (Optional) Path to a file with identifications of videos.

A.1.2 Docker pipeline

The docker pipeline provides a quick way of running defined feature extractors, creating triplets and transforming triplets into the SQL insert statements. It takes a directory with images and an image list as input, and it outputs the feature matrices, triplets, and SQL insert statements into that directory. Sample images generated from Stable diffusion² with the image list are provided in the directory samples.

Prerequisites

- Docker³ \geq 20.10.17

Run

Simply build the container:

```
docker build -t feature-extractor .
```

Then run the container:

```
docker run -v 'realpath ./samples':/data feature-extractor
```

The resulting image features, triplets and SQL statements will be saved in the `./samples/features` directory.

A.2 Image similarity app - User documentation

This application provides a simple web interface for image similarity study. There are two ways of running the project. First is manually installing dependencies and running the Node server and PostgreSQL database alone or running the whole application at once using the docker-compose⁴. The second way is more convenient for running the application unchanged.

Live application is running here: <https://otrok.ms.mff.cuni.cz:8031/user>.

A.2.1 Manual

The manual installation is more convenient for developing new features and bug fixing.

Prerequisites

- NodeJS \geq 18.15.0
- npm \geq 9.5.0
- PostgreSQL⁵ \geq 14.3

²<https://huggingface.co/spaces/stabilityai/stable-diffusion>

³<https://www.docker.com/>

⁴<https://docs.docker.com/compose/>

⁵<https://www.postgresql.org/>

Install

Install dependencies using npm.

```
npm install
```

Execute `database/full_init.sql` in the PostgreSQL database.
Import triplets from the feature extractor project.

Run

Run command:

```
npm run start
```

A.2.2 Docker

The docker installation is an easy way of getting the application up and running on the sample data or with your own data preprocessed by Feature extractors in appendix A.1.

Prerequisites

- Docker⁶ \geq 20.10.17
- Docker Compose⁷ \geq 1.29.2

Run

A simple preview of the application can be run using docker-compose.

The sample data and their triplets are the same as the feature extractor project.

```
docker-compose up
```

The application will be ready on URL: `https://localhost:3000/user`.

A.3 Dataset evaluation and fine-tuning - User documentation

This contains a pipeline for fine-tuning and evaluation. The whole pipeline is presented in four Python scripts. Three scripts (`evaluate_dataset.py`, `prepare_dataset.py`, and `train_model.py`) are executable and provide a help option (e.g. `train_model.py -help`).

⁶<https://www.docker.com/>

⁷<https://docs.docker.com/compose/>

A.3.1 Requirements

This project expects these dependencies to be installed prior to running:

- Python ≥ 3.8 && < 3.10
- Tensorflow ≥ 2.9
- TensorFlow Addons $\geq 0.19.0$
- Numpy
- Pandas
- Pillow

A.3.2 Preprocessing

Firstly, the raw dataset has to be prepared for training. For this purpose, the `prepare_dataset.py` was created. It takes three arguments:

- `--triplets` - Triplets CSV file
- `--judgements` - Triplet judgements CSV file
- `--output` - Output directory

The resulting CSV files will be placed in the output directory.

A.3.3 Evaluation

The second Python script evaluates the user agreement on the dataset for reference. In default settings, it evaluates the binary agreement on ResNet-50. You need to modify the `feature_extraction(img_path)` function on line 25 for different models. It takes three inputs. These are:

- `--dataset` - Dataset CSV file (output from `prepare_dataset.py`)
- `--images` - Root directory for V3C1 images
- `--output` - Name of the output CSV file with results

The human agreements will be present in the output directory.

A.3.4 Fine-tuning

The last executable Python script starts fine-tuning. There are many arguments described in `train_model.py -help`. It takes the preprocessed dataset and V3C1 keyframes and starts model fitting with the given dataset part, optimizer, learning, fold size, fold index, number of epochs, etc. The Tensorflow logs and the best-performing model will be saved in `{base_path}/logs/fit/{generate_name_of_run}.|`

For fine-tuning the W2VV++ model, you need to obtain the network weights first from <https://github.com/xuchaoxi/video-cnn-feat>. The obtained model needs to be converted into the Tensorflow 2 Model. The TF2 Model has to be returned from function `create_w2vv_model` in `finetuning_lib.py` on line 413. Then the training script needs to be run with these parameters:

```
train_model.py -p /path/to/dataset \  
--epochs 200 \  
--seed 42 \  
--folds 5 \  
--fold_n N \  
--batch 32 \  
--yes-general_train \  
--no-scuba_train \  
--no-wedding_train \  
--learning_rate_class 3 \  
--optimizer adamw \  
--model w2vv \  
--loss TripletFuzzyLoss \  
--loss_margin 0.2 \  
--dropout_rate 0.5 \  
--weight_decay 0.0001 \  
--no-train_whole \  

```

For each cross-validation step, the parameter N needs to be changed accordingly.