Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Selection of Representative Landmark Images

Pavel Gramovich

Supervisor: Prof. Jiří Matas
Field of study: Computer Vision
August 2020

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Gramovich Pavel**	Personal ID number:	**481728**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Open Informatics**		
Specialisation:	**Computer Vision and Image Processing**		

II. Master’s thesis details

Master’s thesis title in English:

Selection of Representative Landmark Images

Master’s thesis title in Czech:

Výběr reprezentativních obrazových prototypů

Guidelines:

The objective of the work is to select, given a large collection of related images, e.g. of a landmark, a subset of visually diverse images that “best” represents the collection. Definition of “best represents” is part of the assignment. For instance, each selected image should represent a group of visually similar photos and outliers - images that differ significantly from any other image, should not be output. The proposed method should be suitable for use in image retrieval systems for diversification of search results. The proposed method should thus be efficient for even large collections and be robust to small changes in initial image set.

The student will:

1. Review existing work on the problem and related problems.
2. Define a dataset for evaluating and comparison.
3. Propose a method for the above-defined problem and implement it.
4. Evaluate the performance of the proposed approach and compare it to a baseline solution.

Bibliography / sources:

[1] D. Dang-Nguyen, L. Piras, G. Giacinto, G. Boato and F. G. B. De Natale, "A hybrid approach for retrieving diverse social images of landmarks," 2015 IEEE International Conference on Multimedia and Expo (ICME), Turin, 2015, pp. 1-6.

[2] Dang-Nguyen, Duc-Tien, Luca Piras, Giorgio Giacinto, Giulia Boato and Francesco Natale. “Retrieval of Diverse Images by Pre-filtering and Hierarchical Clustering.” MediaEval (2014).

[3] Boato, G., Dang-Nguyen, D.-T., Muratov, O., Alajlan, N., and De Natale, F. G. B. “Exploiting visual saliency for increasing diversity of image retrieval results.“ Multimedia Tools and Applications, 2016.

[4] Ionescu, B., Gînscă, A. L., Zaharieva, M., Boteanu, B., Lupu, M., and Müller, H. Retrieving diverse social images at MediaEval 2016: Challenge, dataset and evaluation. In: Proceedings of MediaEval Benchmarking Initiative for Multimedia Evaluation, CEUR-WS.org, vol. 1739, 2016.

[5] Zhou, Z., Wu, Q. J., Huang, F., & Sun, X. (2017). Fast and accurate near-duplicate image elimination for visual sensor networks. International Journal of Distributed Sensor Networks.

Name and workplace of master’s thesis supervisor:

prof. Ing. Jiří Matas, Ph.D., Visual Recognition Group, FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **14.02.2020** Deadline for master's thesis submission: **14.08.2020**

Assignment valid until: **30.09.2021**

prof. Ing. Jiří Matas, Ph.D. Supervisor’s signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature

Abstract

The diversity of image retrieval results is an important feature that allows users to explore different aspects of the queried object. However, most of the works in this area are focused more on relevance rather than diversity. In this thesis, we are ameliorating this situation by proposing a new method for retrieving a diverse set of landmark images, which is based on recent advances in image retrieval area. The proposed approach consists of three phases. On the first one, irrelevant images are deleted from the input set using two detector networks. Then, the clustering phase follows, where landmark images are divided into groups by visual similarity based on distances between state-of-the-art image descriptors. Finally, from each cluster, a single representative located in the densest area of the cluster is chosen. For each phase, several alternative options are proposed, and the best combination is determined on the MediaEval dataset. Conducted experiments show that the proposed approach is superior to the current state-of-the-art, both in terms of diversity of the retrieved set and in terms of relevance.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodological instructions for observing the ethical principles in the preparation of university theses.

Prague, August 14, 2020

Contents

Figures

Tables

Chapter 1

Introduction

The amount of data available online today is unimaginably huge. To find the information of interest, one uses search engines such as Google or use structured data aggregators such as Wikipedia. The job of these engines is to select from a large collection a set of "documents" most relevant to the query and present them in a compact and user-friendly way. In addition to being relevant to the request, it is desirable for the retrieved set to be complete, which means that the queried concept is presented in all possible aspects. For example, for "Jaguar" query, a search engine should return not only articles about a car but also an article about the animal. On request "Yesterday", a music aggregator should find not only the original Beatles track but also all sorts of covers performed by different groups in different genres. On request "Colosseum", an image search system should find not only a photo of the main entrance, but also photos taken from different positions, at different times of day, and in different weather conditions. Apart from being relevant and complete, it is also desirable for the output of a search engine to be compact, as users can not spend hours browsing through thousands of returned "documents".

The properties of compactness and completeness of search engine output are difficult to combine together, nevertheless, both of them are important for users. This problem, therefore, attracts the attention of many researchers [2, 3, 5, 20, 24]. However, the topic raised is broad, since there can not be a common solution that would suit all search engines at once. Different types of search engines have different notions of similarities between documents and different sets of available information. Because of that, there is a necessity for individual solutions for each use-case.

This work is focused on one of these use-cases, specifically landmark image retrieval systems, but it is not limited by that. The proposed approaches could be used in general image retrieval systems such as Flickr or Getty to answer landmark-related queries. But they could also be used to provide a better landmark overview for Wikipedia articles about that landmark.

1.1 Landmark definition

The English language does not give a broadly accepted definition of a landmark. In some works, well-known lakes and parks can be considered landmarks, while in others it is prohibited. To avoid misunderstandings, it is necessary to give an exact definition.

For this work, the landmark definition from [10] was taken. There, a landmark is defined as an object that satisfies the following properties.

- Local uniqueness. Landmark should be distinguishable from its surroundings. For example, a tree in a forest cannot be a landmark.
- Global uniqueness. Landmark should be distinguishable from other landmarks. That property excludes objects such as traffic lights.
- Unchangeable position. According to this property, famous paintings cannot be called landmarks.

1.2 Problem description

Informally the problem addressed in this work could be formulated in the following way. Given a large collection of images, related to a certain landmark, select a subset of a given size that would describe the landmark the "best". A formal description would be hard to imagine, but a number of properties of the desired subset can be formulated.

1. Absence of near-duplicates in the returned subset. Near-duplicates are images that could be made almost identical by applying an affine transformation. Such images do not provide additional information.
2. Each image from the output represents a group of images from the input set, united by a common property. This property might be a point of view, weather or lighting conditions, special decorations, and so on. Examples of these groups could be found in figure 1.1.
3. Not more than one image from each group is allowed in the output. Near-duplicates should be in the same group, so this property is stronger than the first one, however, it is harder to check.
4. Only the largest groups are represented in the output subset.
5. Output images contain a landmark on the foreground. This property ensures that photos of people in front of a landmark or images that were taken close to landmark location, but do not depict it would not be present in the output. Such images are not relevant for people looking for a general overview of a landmark.

These properties might not be ideal for use in image retrieval systems, as in that case there is additional information such as tags and number of views for each image. And it would make sense to use this information in property formulations. For example, it would be a good idea to give higher priority to images with a bigger number of views in the fourth property or to define a group as a subset of images that share many tags in the second property. However, that would lead to a loss of generality.

Even though introduced properties are not ideal, still diversification of the output of an image retrieval system is the main use case for this work. And that imposes certain requirements on the proposed approaches.

1. Fast running time. The algorithm needs to answer user's queries in real-time.

2. Robustness to small changes in the input set. That means adding one or two images to the input set or changing the order of images should not affect the final result.

3. Memory efficiency. Modern retrieval systems contain billions of images, so the memory usage per image should be minimal.

4. Possibility to add new images into the system fast.

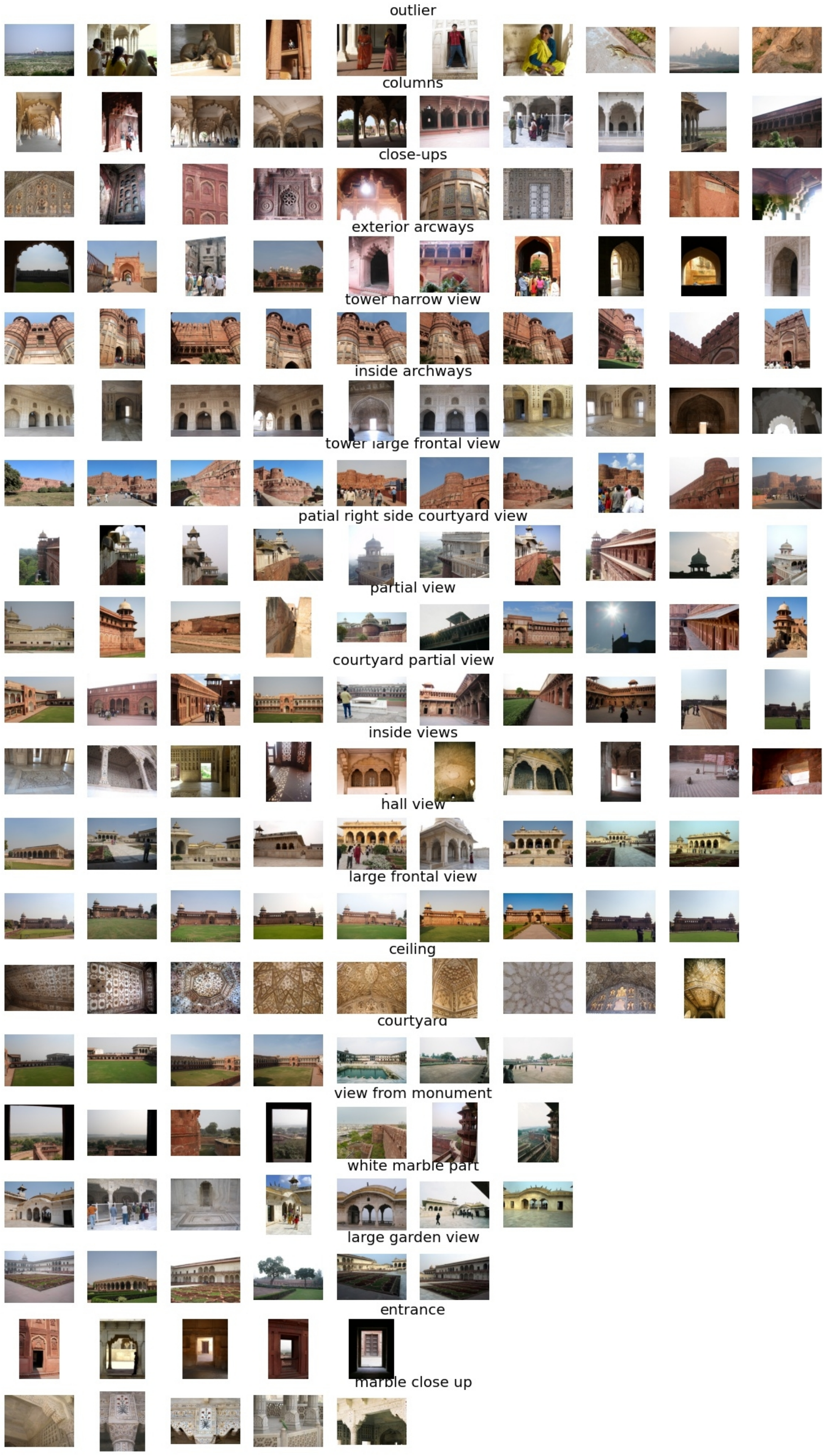


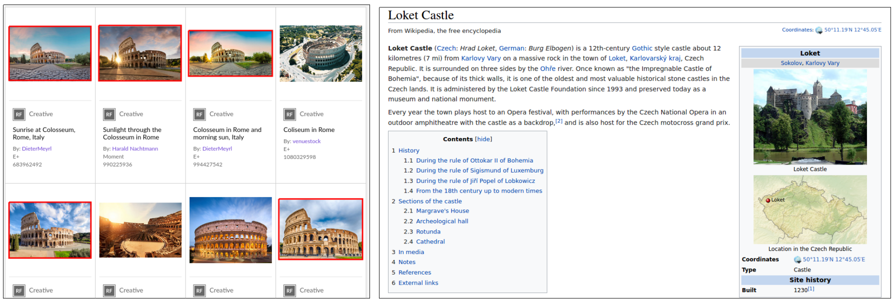
Figure 1.1: MediaEval dataset. Example of landmark images. Each row corresponds to a different landmark property

1.3 Motivation

Since the number of images uploaded to the web grows rapidly, the problem of selecting a concise visual representation of an object of interest becomes more and more important. This especially concerns image retrieval systems as many people use them. If retrieval results contain a lot of similar images, then a person who uses it might not find what he looks for. Example of such a situation is presented in figure 1.2a.

Approaches for the problem described above could also be applied to Wikipedia articles about landmarks. A typical example of such a page is given in figure 1.2b. On that page, there is only one photo of the castle, which does not give a complete picture of it. Using the algorithms suggested in this paper, it would be possible to automatically select the most representative set of images for all Wikipedia articles about landmarks, which would improve the quality of these articles.

Another application could be to remove redundant images from a collection to reduce memory consumption.



Loket Castle

From Wikipedia, the free encyclopedia

Coordinates: 50°11.19′N 12°45.05′E

Loket Castle (Czech: *Hrad Loket*, German: *Burg Elbogen*) is a 12th-century Gothic style castle about 12 kilometres (7 mi) from Karlovy Vary on a massive rock in the town of Loket, Karlovarský kraj, Czech Republic. It is surrounded on three sides by the Ohře river. Once known as "the Impregnable Castle of Bohemia", because of its thick walls, it is one of the oldest and most valuable historical stone castles in the Czech lands. It is administered by the Loket Castle Foundation since 1993 and preserved today as a museum and national monument.

Every year the town plays host to an Opera festival, with performances by the Czech National Opera in an outdoor amphitheatre with the castle as a backdrop,[2] and is also host for the Czech motocross grand prix.

Contents [hide]

1 History
  1.1 During the rule of Ottokar II of Bohemia
  1.2 During the rule of Sigismund of Luxemburg
  1.3 During the rule of Jiří Popel of Lobkowicz
  1.4 From the 18th century up to modern times
2 Sections of the castle
  2.1 Margrave's House
  2.2 Archeological hall
  2.3 Rotunda
  2.4 Cathedral
3 In media
4 Notes
5 References
6 External links

Loket

Sokolov, Karlovy Vary

Loket Castle

Location in the Czech Republic

Coordinates	50°11.19′N 12°45.05′E
Type	Castle
Site history	
Built	1230[1]

(a) : Example of retrieval results in Getty image database. 5 out of 8 images are almost identical

(b) : Example of Wikipedia article about a landmark. Only one picture of the landmark is provided, which is not enough for a full overview.

Figure 1.2: Possible applications

1.4 Approach outline

The structure of the proposed approach follows from the requirements for the output subset, mentioned above. The first step is to delete images that are not related to a landmark. This procedure should take place during the preprocessing phase in order to reduce the response time. The second step is clustering. This step follows from the condition that each image should represent some group. The last step is to select representatives from the found clusters. Since the number of found clusters may be larger than the required size of the subset and since the clusters themselves may contain errors, it is necessary to perform the procedure of selecting the most appropriate images. A schematic representation of the algorithm can be found in figure 1.3.

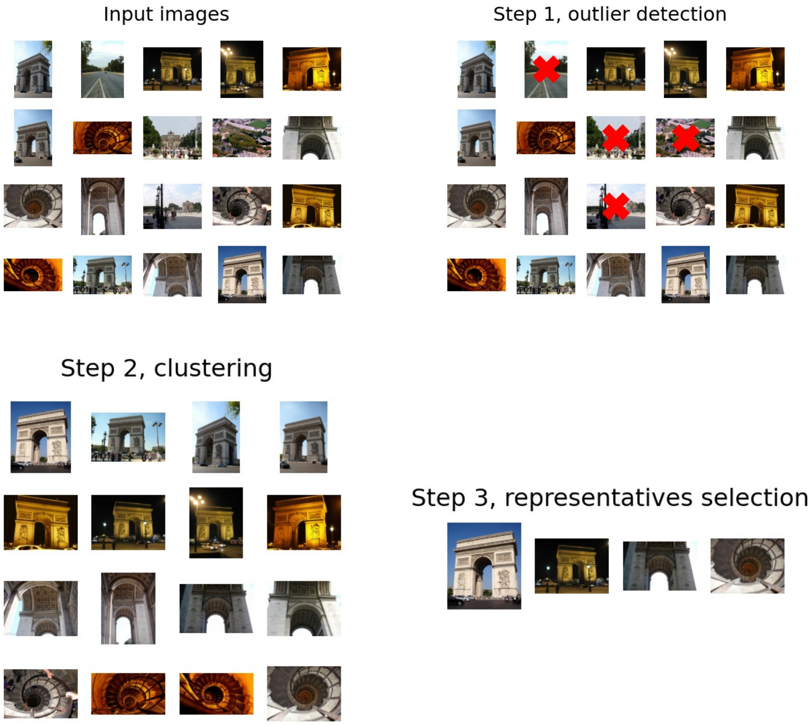


Figure 1.3: Approach scheme. Images from MediaEval dataset.

1.5 Structure

The rest of the work is organized as follows. In chapter 2, several related works are reviewed. The proposed approaches are briefly described and their drawbacks and limitations are outlined. Chapter 3 describes the datasets used to assess the quality of the proposed system and to justify design choices. In chapter 4, several image descriptors are introduced first. Then, three approaches for outlier detection are proposed. And finally, variants of clustering and representative selection algorithms are described. In chapter 5, all proposed methods are compared among themselves and with the algorithms of other authors. The analysis of operation time and stability of the suggested methods is also given. Finally, Chapter 6 summarizes the work done and indicates possible directions for future research.

Chapter 2

Related work

This chapter focuses on approaches for related problems from previous work. First, MediaEval competition is introduced. This competition has led to the emergence of many approaches to the diversification of landmark images, and datasets that were created to assess participant's submissions, have now become a standard for evaluating the quality of diversification algorithms in the area. After that, several approaches, presented at these competitions, are described. And finally, two more approaches are presented, which were proposed in research outside MediaEval competition.

2.1 MediaEval competitions

In 2013 - 2017 MediaEval was organising competitions devoted to retrieval of diverse social images [7–9, 18]. The task was to retrieve diverse landmark images from the output of the Flickr search engine. Participants were provided with the dataset (each year the dataset was a little different) consisting of several landmarks/locations each containing several hundred photos retrieved with Flickr. Each location had a name, a link to Wikipedia page, and up to 5 images from Wikipedia. Each image had a title, user tags, GPS data, Flickr rank, and a number of views. Given this data, participants had to choose a diverse subset of images for each landmark. In order to assess diversity, each image was assigned to a class. One of the classes consisted of outlier images - images that are not relevant to the landmark. Submissions were assessed using the following three metrics. The first one was called precision and showed a fraction of images in a subset that do not belong to the outlier class. The second metric was called cluster recall and it showed a fraction of ground-truth classes (excluding outliers) represented in an output subset. And the last metric was the F1-score, which was calculated as the harmonic mean of the first two metrics. F1@20 (F1-score for subset size of 20) was the main competition metric. All submissions were ranked according to it.

Participants were asked to provide several variations of their algorithm (several runs). In one of these variations (run1), it was allowed to use only visual information. That means the algorithm did not have access to image tags, titles, GPS coordinates, and landmark names (Wikipedia images were allowed). This variation is the most comparable to the problem set out in this work, so all algorithms from the MediaEval competition described in this chapter are using this set up (run1).

2.2 SocialSensor at MediaEval 2014

In 2014 competition the best approach (in run1 setting) was provided by SocialSensor team [20]. The key idea of the algorithm was to select a subset S of images that maximizes utility function $U(S) = w \cdot R(S|q) + (1 - w) \cdot D(S)$, where $R(S|q)$ is a relevance of selected subset S to a query image q, $D(S)$ is a diversity of subset S and w is some constant between 0 and 1. Finding an optimal subset might be tricky as there are many possible subsets. Instead, an approximate solution was found using a greedy strategy. At each step, M images are added to the solution, which maximizes utility. This step repeats until the required number of images is not reached.

Relevance of a set was calculated as a sum of relevances of each constituting image $R(S|q) = \sum_{i \in S} R(i|q)$. The relevance of a single image is calculated using logistic regression, which is trained for each landmark separately. To train this model for a landmark, ground truth relevance labels were used. All relevant images for this and other landmarks plus wiki images for this landmark served as positive examples and all outliers were used as negative examples. Wiki images received a higher weight. Relevance probability, predicted by trained logistic regression was used as $R(i|q)$ in the formula above.

Diversity of an image set was calculated as the minimum distance between pair of images of this set $D(S) = min_{i,j \in S} D(i, j)$, where $D(i, j)$ is a cosine distance between descriptors of images i and j. VLAD+CSURF [6,12] vectors were used as descriptors for diversity function and as features for logistic regression.

The proposed approach is not acceptable for use in an image retrieval system, as it requires training linear regression for relevance function. Not only it is time-consuming, but it also requires some positive examples (wiki images) which are not available in general.

2.3 TUW at MediaEval 2015

One of the best (run1) approaches at Mediaeval 2015 was proposed by the TUW team [19]. The output set S of size k was found by maximizing diversity function $D(S)$. Exact function was not specified, but it was probably either $D(S) = \sum_{i,j \in S} D(i, j)$ or $D(S) = \max_{i,j \in S} D(i, j)$ where $D(i, j)$ is a distance between images i and j. An interesting idea was used to calculate the distance. All landmark images were split/clustered into several groups in several ways. Each way might differ in the clustering algorithm, image descriptor, or distance type. Then for each pair of images, the number of times they occurred in different clusters is calculated. This number was used as a distance between a pair of images.

Such a definition of distance can leverage information from different descriptors and can result in a stable output (that doesn't change if a few images were added/removed). However, clustering has to be done several times, which might require too much time for the online use case. Also, storing several descriptors for each image increases memory consumption.

2.4 CFM at MediaEval 2017

The approach introduced by CFM team [16] used complete-link agglomerative hierarchical clustering to divide the set of input images into N different classes. Pairwise distances for the clustering algorithm were calculated as Euclidean distance between auto color correlograms [11] for corresponding images. From each cluster, a single representative image was chosen (N images in total). This choice was made using image ranks, provided by the Flickr search engine. The image with the highest rank was chosen as images with high ranks are likely to be relevant. Then from the list of N selected images, top $K < N$ images with the highest rank were returned as the system's output.

This approach is fast and memory efficient. However, it relies on retrieval ranks, which might not be available in all use cases. Furthermore, the image with the highest rank within the cluster does not necessarily represent this cluster the best. It may be that the highest-rank image is located on the edge of its cluster and in that case, an image from the cluster center would be a better choice.

2.5 NLE at MediaEval 2017

Another interesting idea was proposed by NLE team [17]. They tried to utilize a variant of pseudo-relevance feedback to re-rank input images. New image ranks $R'(i)$ were calculated using the following formula.

$$R'(i) = \frac{\sum_{j \in T} R(j) S(i,j)}{\sum_{j \in T} R(j)}$$

where T is a list of top t retrieved images, $R(i)$ is a Flickr rank and $S(i,j)$ is a similarity measure (e.g. cosine) between images i and j.

Then the diversity was achieved by using the Maximal Marginal Relevance [4] technique. It tries to find a subset of images with the maximum sum of ranks, where each rank is decreased by similarity to the most similar image from that subset (margin).

$$S = \arg\max_{|S|=k} \sum_{i \in S} R'(i) - \beta \max_{j \in S} S(i,j) \tag{2.1}$$

Here β is a parameter that controls the diversity of the resulting subset. If β inf, then the maximum similarity between a pair of images in the resulting subset would be as small as possible. However, image relevance would be ignored. On the other hand, if $\beta = 0$, then the relevance of the resulting set is the highest possible, but all selected images might be duplicates.

The exact procedure of solving optimization problem 2.1 was not specified, but a greedy approach, used in one of the previous sections could be adopted for this case.

The proposed approach is fast and efficient, but it still requires retrieval ranks. And it does not solve a problem of duplicate images. If top t images with the highest ranks are very similar, then pseudo relevance feedback would not work great.

2.6 A hybrid approach

Dang-Nguyen et al. [5] were working on a problem similar to the one set out at MediaEval competition. All data from the MediaEval 2014 dataset, including textual information and GPS coordinates, was used for landmark image diversification and ground-truth class labels were used for quality assessment. The same metrics as in MediaEval competition were used to select the best system parameters.

The proposed algorithm consisted of three stages.

1. Outlier filtering
2. Clustering
3. Summarization

At the outlier Filtering stage, several techniques were used to remove images that are likely to be irrelevant. First, images containing human faces were deleted, using a face detector. It was explicitly specified in the description of MediaEval competition, that images containing people on a foreground are marked as outliers, so deleting such images is a logical step. Then GPS coordinates were used to delete images that were taken far away from the queried landmark. Photos shot at a distant location are either do not contain landmark at all or landmark is barely visible on them. In both cases, such images should be considered outliers. The next technique tries to detect out of focus or blurred images using wavelet transformation. Photos of bad quality are considered outliers in the MediaEval dataset, so this step is justified. And the last method of detecting outliers deletes images with a few views. Even if such images represent a landmark, they are probably not interesting as people do not watch it and thus should not be present in the output of an image retrieval system.

Balanced Iterative Reducing and Clustering algorithm [25] was used to cluster the images. First, a CF tree is built using image textual descriptors using the following iterative procedure. Each new image is guided through the tree from the root to one of its leaves. At each node, an algorithm moves to the child closest to the image. If a leaf is reached and a distance from the image to that leaf is smaller than threshold T, the image is added to that leaf, otherwise a new leaf is added. If at some point number of children becomes larger than the branching factor B, that node is split. If the root is split, tree height increases by 1. After CF tree contains all images, it is refined using visual features. The exact procedure of refinement is not specified in the paper [5], but it was said that the BIRCH algorithm provides such an option. After refinement is done, clusters are extracted from that tree. Images withing resulting clusters are similar both textually and visually.

After clustering is done, a summarization stage follows. From each cluster, a few representatives are chosen based on user credibility information. The user credibility score is calculated in the following way. Up to 300 images, tagged by the user were chosen. Then for each tag relevance scores are calculated. User credibility is then calculated as the average relevance of tags, provided by that user. After credibility is evaluated for each user, representative images from each cluster are extracted. Clusters are sorted by a decrease in cluster size. Then from each cluster, a single image is selected. If more images are required, the second image is selected in each cluster, and so on. Selection is based on user credibility information. The image that was uploaded by the user with the highest credibility is selected as the first one. If there are several images uploaded by that user, the one closest to the cluster centroid is selected. If the second (or third, fourth, ...) image has to be selected from the same cluster, the image with the biggest distance to already selected ones is chosen.

The proposed algorithm combines visual, textual, and user credibility information to select a subset of diverse and relevant landmark images. Consequently, it requires lots of data for it to work properly, and thus use case of the algorithm is quite narrow. However, some ideas like outlier detection could be adopted for use in different settings.

2.7 Exploiting visual saliency

In work [3], authors were solving a problem of diversifying results of an image retrieval system. But unlike all previously discussed work, they were not restricted by landmark images only. The input set could contain images of any objects for example people, animals, vehicles, landscapes, and so on. The goal was to select a subset of a certain size that represents input set the best. That means that similar images are not desired in the output, as they provide redundant information. At the same time returned images should represent a significant group of input images.

To solve this problem authors proposed to use saliency map to distinguish between foreground and background. Examples of saliency maps are provided in figure 2.1. This allows assigning a higher weight for similarities between foreground objects and thus focus more on image contents.

Saliency maps were estimated for each image in the dataset. Then a set of low-level visual features was extracted. This feature set contained the size and location of the foreground object as well as the color and orientation of histograms for the foreground and background. These features were used to

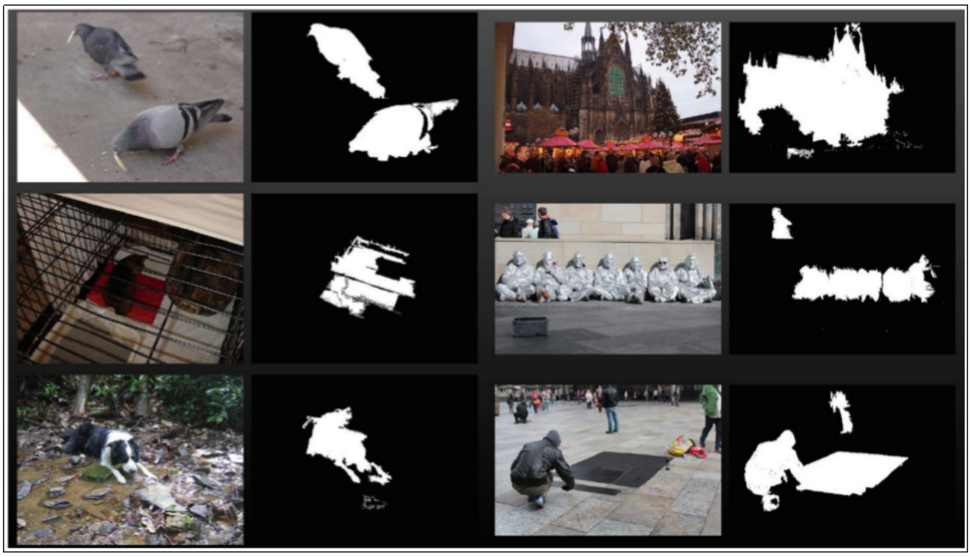


Figure 2.1: Examples of sailency maps. Illustration from original paper [3]

calculate a distance between a pair of images. $D(i, j) = \sum_k^N w_k cos(f_i^k, f_j^k)$, where N is a size of the feature set, f_i^k is kth feature for image i, cos is cosine between feature vectors and w_k is a weight of kth feature.

To select a diverse subset of images an iterative procedure was proposed. At each iteration and image which maximum ranking score $RS(i)$ is selected. $RS(i) = \beta_1 \sum_{j \in S} D(i, j) - \beta_2 \sum_{j \in A} D(i, j)$. Here S is a subset of currently selected images and A is a set of the rest images. The first sum represents the total distance to currently selected images. Maximizing this sum will lead to the choice of image that differs from selected images the most, and thus this sum promotes diversity. The second sum measures the distance to not yet selected input images and minimizing this sum would lead to the promotion of representativeness.

Techniques developed in this work could be adopted for many diversification setups, due to the generality of the addressed problem. Results from the paper show that distinguishing between foreground and background could improve the quality of retrieval algorithms.

Chapter 3

Dataset descriptions

This chapter introduces two image datasets that are used in this work.

3.1 Desired properties

In the problem description section, several properties for the output subset were formulated. These properties impose the following requirements for the dataset, that can be used for quality assessment.

1. Dataset should contain several thousand different landmarks, to properly measure and compare the quality of the system.
2. Landmarks in the dataset should satisfy the definition given in the introduction.
3. Landmark images should be obtained with the help of some actually used image retrieval system. And this system should not delete outliers or duplicates while forming the output.
4. Outlier images should be marked.
5. Landmark images that are not marked as outliers, should be divided into classes based on visual similarity.

3.2 Mediaeval Dataset

In 2014 MediaEval [9] has published a dataset of landmark social images, which was used to assess the quality of submitted results on their competition. Additionally, it was used in many independent studies in the area, so it is easy to compare approaches, proposed in this work with state-of-the-art. The MediaEval dataset best matches requirements introduced above, though not perfectly, it is the main dataset for this work. All design choices were made based on metrics calculated on this dataset.

3.2.1 Description

The MediaEval dataset contains approximately 45 thousand landmark images, which were retrieved using the Flickr search engine. These images are divided into 153 groups, each representing a different landmark. There are approximately 300 images for each landmark. Metadata was provided for each image containing GPS coordinates, title, tags, rank in search engine result page, and other data. Metadata for landmarks contained a name, GPS coordinates, link to Wikipedia article, and up to 5 representative photos from Wikipedia.

Images of a landmark were manually divided into groups by trained annotators. They were not restricted in time and had all information about a landmark that might be useful for clustering. One of the groups was dedicated to outliers and contained images which are not a common representation of a landmark, e.g. photos of people and animals, photos of bad quality, photos of different landmarks. Each of the rest of the groups represents a certain feature of a landmark. Images within such groups were usually made from similar viewpoints in similar light and weather conditions. The number of groups for each landmark is different and is around 20-25. Examples of group labels for the Eiffel Tower are presented in figure 3.1

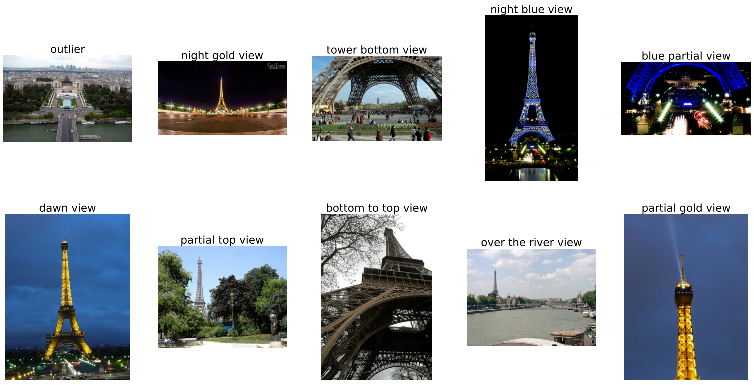


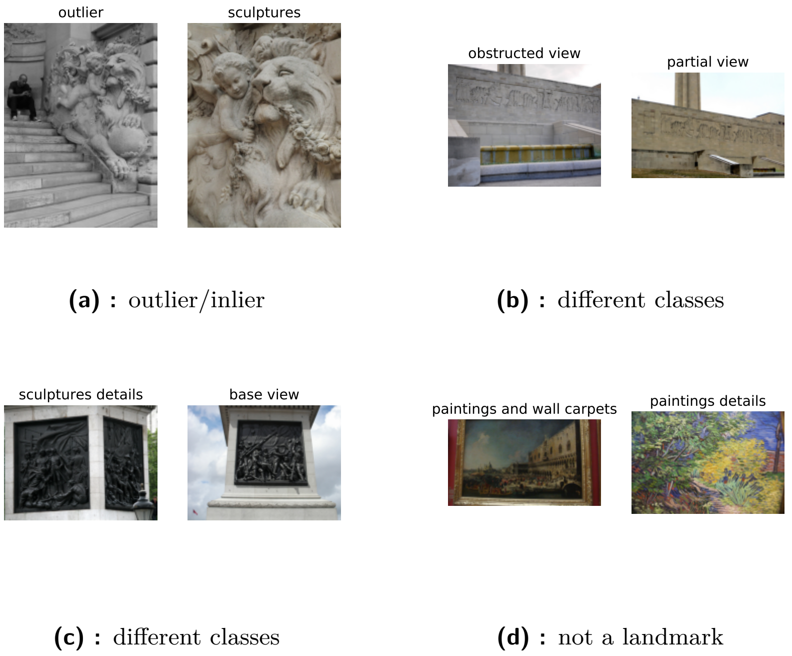
Figure 3.1: MediaEval dataset. Examples of group labels for the Eiffel Tower.

3.2.2 Issues

The MediaEval dataset does not satisfy all the requirements introduced above. First of all, it is not big enough. It contains only 153 different landmarks, therefore metrics calculated using it will have large uncertainty. On the other hand, 153 landmarks already contain 45 thousand images, each of which must be marked manually. Huge efforts have been made to create even such a dataset. Therefore, it is difficult to ask for a dataset with at least one thousand landmarks.

Second, image labels are not very accurate. There are lots of examples of visually similar images, that belong to different classes (figures 3.2b, 3.2c). Moreover, there are examples of similar images one of which is an outlier and the other is an inlier (figure 3.2a).

Third, there are photos that do not satisfy the landmark definition, which are not labeled as outliers (figure 3.2d).



(a) : outlier/inlier

(b) : different classes

(c) : different classes

(d) : not a landmark

Figure 3.2: Examples of questionable image labels. **(a)** image on the left is marked as outlier, while image on the right is not. **(b, c)** Visually similar images from different classes. **(d)** images that do not satisfy landmark definition (unchangeable position)

Even though this dataset is not ideal, it is the best option available. It has become a standard choice for research in this area and many studies use it to assess the quality of diversification algorithms [2, 3, 5, 20, 24].

3.3 Google Landmark Dataset

Another landmark dataset was provided by google - Google Landmark Dataset. It was first introduced in [14] and then used in Kaggle competitions.[1][2]. The dataset is focused on assessing the quality of landmark retrieval and recognition algorithms and it does not provide labels for assessing diversity. So its use in this work is limited to unsupervised training and visual assessment of the algorithm.

[1] https://www.kaggle.com/c/landmark-recognition-challenge

[2] https://www.kaggle.com/c/landmark-retrieval-challenge

Google Landmark Dataset contains around 5 million images of 200 thousand different landmarks. Landmark labels were assigned automatically using GPS data and visual similarity. The number of images for each landmark varies from a few to a few thousand. Landmarks presented in the dataset are from all over the world and represent both man-made objects and natural monuments.

For the purposes of this work, a small subset of the data was chosen. It contains 100 landmarks, that satisfy the definition from the introduction. Each landmark is represented by about 100 images. Preference was given to locations with the least amount of outliers.

Chapter 4

Proposed approach

In this chapter, an approach for selecting a diverse subset of landmark images is proposed.

4.1 Approach overview

The proposed approach consists of three steps: outlier detection, clustering, and representative selection. At the outlier detection stage, two detector networks are used to delete images that contain humans (and some other objects) and images that do not depict a landmark. Then, the clustering stage follows. Each image is assigned a visual descriptor of size 128. A matrix of pairwise cosine distances is calculated for these descriptors and this matrix is then used for clustering. If the number of clusters is greater than the desired size of the output subset, clusters with the smallest number of images are deleted. Several options for descriptor and clustering algorithm are considered. The best variation is determined in the experimental section. Finally, from each cluster, the best representative is chosen. The idea is to find an image in the densest area of the cluster. For that, three methods are proposed. The best one is determined in the experimental section.

4.2 Proposed visual descriptors

Quality of image visual descriptors is of great importance for the quality of the whole system. So the choice of descriptor should be made very carefully. Several options are suggested in this section.

In the last decade, neural networks have had a huge success in the computer vision area, especially image classification. This success was driven by an unprecedented increase in data availability and the invention of deep network architectures. Today there exist many pre-trained models that show state-of-the-art performance on the ImageNet image classification dataset. These networks have learned to extract highly-descriptive features from image patches, which could be used for the purposes of diversification.

Most of proposed descriptors are based on ResNext-50 [23] neural network architecture. This particular architecture consists of 4 convolutional blocks and a fully connected network at the end. Spatial and feature space sizes stay the same within each block and change only between blocks. The size of feature vectors doubles when moving to the next block while the number of features is reduced by 4 times. Each convolutional block consists of several residual blocks, whose architecture is depicted in figure 4.1. Two key features of this architecture are residual connections and split-transform-merge strategy. Residual connections connect the input of the block directly to the output skipping convolutions and non-linearities which improves the gradient flow and enables training of extremely deep networks. Split-transform-merge architecture serves as a regularizer and reduces computational complexity.

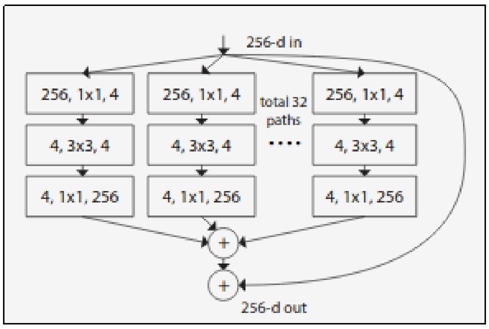


Figure 4.1: Architecture of ResNext residual block. Image from original paper [23]

4.2.1 Maximum activations of convolutions

The first proposed descriptor uses feature vectors from the output of ResNext convolutional blocks. For each feature dimension, a maximum value is found across all vectors in the output. This is equivalent to a max pool layer with the size of the pool equal to the size of spatial dimensions. Each feature dimension can be thought of as representing a certain template or object. If this template is present in the receptive field of a neuron, the feature value will be high. Finding maximum across spatial dimensions discards positional information and allows checking if an object or template is present in the image regardless of its position. The resulting vector of maximums represents a set of objects present in the image.

The proposed descriptor is robust to small shifts and rotations of the original image thanks to max-over-spatial-dimensions and properties of the network. To make it robust to scale changes, an image pyramid is built. Each image in the pyramid is scaled down by a factor of $\sqrt{2}$ with regard to the previous image and there are 4 images in total. A vector of maximums is built for each image as described above and then the maximum across scale dimension is found.

The size of the vector of maximum activations for the 3rd and the 4th convolutional blocks is 1024 and 2048 respectively, which is quite big for the use-case of this work. So the vector size is reduced with the following procedure. Each vector is l2-normalized, then its dimension is reduced to

128 with PCA, and then it is l2-normalized once again. PCA was trained on images from the GLD dataset.

4.2.2 TF-IDF for ResNext features

In image retrieval, Bag-Of-Visual-Words is a useful method of representing an image using its local descriptors. In this approach, the whole space of local descriptors is divided into groups of similar vectors - visual words, and the image is represented as a set of visual words present in this image. A bag of visual words could be used to build a global descriptor for the whole image.

One of the ways how to aggregate local descriptors was adopted from information retrieval. A document (image) is represented as a vector with the size the same as the number of (visual) words in the vocabulary (number of groups of local descriptors). ith element of this vector is equal to a product of term frequency tf_i and inverted document frequency idf_i. tf_i is the number of times a word w_i occurred in the document and $idf_i = ln\frac{N}{df_i}$, where N is the number of documents and df_i is a number of documents where word w_i is present. This gives higher weights to words that often occur in the document (tf) and to words that rarely occur in the whole document collection (idf).

Feature vectors from the 3rd or 4th block of ResNext were taken as local descriptors. All local descriptors from all images (including scaled images) of the GLD dataset were gathered and divided into 128 clusters using the k-means algorithm. Then for each group, a document frequency was calculated. The centers of clusters and document frequencies were then used to calculate tf-idf vectors for MediEeval (Div150Cred) dataset. An image pyramid as for the previous descriptor was used to make the resulting global descriptor robust to scale changes.

4.2.3 TF-IDF for DELF features

One potential drawback of two previous descriptors is that all features from a convolutional layer are taken, but not all of them are equally important. For example, a feature vector corresponding to a point on a tree or sky is less important than a feature vector for a door or a window. Also, feature vectors for neighboring points are likely similar and represent the same object, so using all of them is redundant. This problem can be tackled by using DEep Local Features introduced by Noh et al. [14]. Their idea was to use an attention layer to choose the most important features and delete redundant points by non-maximum suppression. They have trained a ResNext network along with the attention layer on Google Landmark Dataset and shared them with the community. These models were used to create global descriptors in this work.

Another improvement could be achieved by using a landmark detector proposed by Teichmann et al. [21]. They have created a landmark bounding box dataset and trained detector network on it. The detector model was shared with the community as well. The output of the detector is a list of bounding boxes along with probabilities of landmark detection. Given a bounding box of a landmark one can remove local features corresponding to points outside the box, as they do not represent the landmark. This additionally improves the quality of feature selection.

For each image, a landmark bounding box with the highest probability of detection was chosen. DELF features of a part of the image inside the bounding box were extracted. Then tf-idf vectors were created the same way as for the previous descriptor.

4.2.4 VLAD for DELF features

Vector of Locally Aggregated Descriptors [12] is another way of transforming a set of local descriptors into an image global descriptor. Local feature vectors from all images (image patches corresponding to bounding boxes) of a dataset are collected and divided into 100 clusters using the k-means algorithm. Then for each image residual vectors are introduced

$$f_i = \sum_{j \in J_i} (d_j - c_i)$$

where d_j is a jth local descriptor, J_i - indices of descriptors for which c_i is the closest cluster center. Resulting residual vectors are then concatenated into one large Dx100-dimensional vector, where D is a size of a DELF descriptor. This vector is then reduced by the PCA algorithm to a 128-dimensional vector and l2-normalized. K-means and PCA were trained on the GLD dataset.

4.2.5 VLAD for CSURF features

This is the only proposed descriptor that does not use neural network based features. It was used in the best solution [20] for The MediaEval Retrieving Diverse Social Images Task. It is based on CSURF [6] local descriptor which is a concatenation of SURF [1] descriptors for each color component of the image. CSURF descriptors are then aggregated into one global descriptor using VLAD the same way as for the previous descriptor.

4.2.6 Mean-normalization

Images returned by an image retrieval system are not independent. They all must be relevant to the query and that means their descriptors are similar as well. That is especially true if the same descriptors are used for retrieval and diversification. Similar vectors are harder to diversify as all pairwise distances are small. To address this problem a mean normalization technique is suggested. A mean descriptor vector of all images corresponding to one landmark/query is calculated and subtracted from these descriptors. Resulting vectors are l2-normalized. This simple technique could spread image descriptors more evenly and improve the quality of diversification.

4.3 Preprocessing step

The output of an image retrieval system such as Flickr contains lots of images that are either not relevant for the query landmark or not good enough to represent it (photos of people in front of a landmark, images of bad quality, obstructed views). Such images, called outliers, represent a serious obstacle to diversification algorithms and should be dealt with accordingly. The MediaEval Dataset contains 14500 (32%) images marked as outliers, so quite a significant part of the retrieval result has to be discarded.

During the preprocessing step, outlier images are detected using two detector networks. First detector is provided in ImageAI library [13]. It is able to detect 80 different types of objects including a "Person" class, which is the most abundant object type in the MediaEval dataset. Among these 80 classes, 10 most useful were chosen for outlier detection. These classes are "person", "car", "bus", "boat", "clock", "chair", "bird", "horse", "dog". If a significant part of an image is occupied by such objects, this image is likely to be an outlier and is deleted from the dataset. Implementation details, as well as experimental results, could be found in sections 5.1.1-2.

The second detector network was introduced and released by Teichmann et al. [21]. This network is able to detect landmarks on images with high accuracy. It was trained on a specially designed dataset of landmark bounding boxes based on Google Landmark Dataset. Using this detector, landmark bounding boxes for all remaining images were found along with the probability of detection. If several bounding boxes were found, only the one with the highest probability of landmark detection was stored. And if the highest probability of detection is less than a certain threshold, the image is regarded as an outlier and is removed from the dataset. This way of detecting outlier images is not particularly accurate as it is quite hard to define what is a landmark. The landmark definition that was used for creating the bounding box dataset included natural landmarks such as mountains, caves, lakes, and forests. That means that a photo of a tree or a rock could be detected as a landmark, while they are marked as outliers in the MediaEval dataset. But still, it is possible to use this detector to remove a significant portion of outliers and thus, improve the quality of diversification. Details are provided in section 5.1.3.

After detected outliers were removed, image local descriptors are calculated. Then, depending on the aggregation method, PCA, k-means, and document frequencies for visual words were learned on Google Landmark Dataset. And finally, global descriptors are created for the MediaEval Dataset.

Summary of preprocessing step

1. Outlier detection using ImageAI.
2. Finding landmark bounding boxes.
3. Deleting images without landmarks.
4. Local descriptors calculation.

5. Training of K-means and PCA.
6. Aggregating local descriptors.

4.4 Clustering algorithms

Each image in the output should represent a group of images united by a certain attribute. That could be a point of view or a lighting condition or special decorations or something else. This requirement naturally implies the use of clustering algorithms. Each image is assigned a global descriptor and images are divided into groups based on the mutual arrangement of these descriptors. If descriptor quality is good enough, images with similar contents will be close in the vector space and will be located in the same group.

A clustering algorithm should satisfy the following criteria.

- It should work fast as it is used in a real-time retrieval system and the number of images to cluster might be more than a thousand.
- It should not have a number of clusters parameter as this number cannot be known in advance.
- It should be robust to noise as outliers are still possible.

In this work, two clustering algorithms are considered. The first one is complete-link agglomerative clustering. It works as follows. First, each image is thought of as a separate cluster. Distances between each pair of clusters/images are calculated. The pair of clusters with the smallest distance is found and merged into a single cluster. Then distances from the new cluster to all other clusters are found. Then again, two clusters with the smallest distance are found and merged and this cycle repeats until the distance between merged clusters is bigger than a certain threshold. The only thing left to define is the distance between clusters. In the complete-link approach it is calculated as follows:

$$d(U, V) = \max_{u \in U, v \in V} d(u, v)$$

Where $d(U, V)$ is a distance between clusters U and V, $d(u, v)$ is a cosine distance between descriptors of images u and v.

The complete-link agglomerative clustering algorithm has the only parameter - distance threshold. By varying this parameter one can control the number of found clusters and their size. The bigger this threshold is, the fewer clusters there will be. Too big threshold could lead to clusters with images from different ground-truth classes and thus a bad quality. At the same time, too small threshold would result in images of the same class located in different clusters which is bad as well. So the threshold value must be selected wisely based on provided ground-truth labels. Details could be found in the experiments section.

The second suggested clustering algorithm is called DBSCAN. It divides all points into three classes - core points, edge points, and outliers. Two points are connected if a distance between them is less than ϵ. If a point is directly connected to at least n other points, it is called a core point. If a point is connected to less than n other points but is connected to at least on core point, it is called an edge point. All other points are outliers. This algorithm has two parameters n and ϵ which makes it harder to use than agglomerative clustering, but it is faster. Comparison of the two clustering algorithms are provided in the experiments section

4.5 Representative selection

After clustering is done, clusters are arranged in descending order of size. Idea is that bigger clusters are more likely to represent an important feature of a landmark than smaller clusters. After ordering, a representative for each group needs to be chosen. Three approaches are suggested. Tree-based, density-based, mean-based.

The tree-based approach is compatible only with the agglomerative clustering algorithm. It uses a tree structure of each cluster to locate the densest area. If a cluster contains one image, then there is only one choice of selecting. But if it contains at least two images, then this cluster is a result of a merge of two smaller clusters. From these constituting clusters, the biggest one is chosen and the algorithm repeats for that cluster. Descending to the biggest cluster hopefully leads to the denser area of the original cluster. This algorithm can be described by the following recursive procedure:

```
def Select(U):                # U - cluster index
  if Size(U) == 1:            # If cluster consists of one image,
    return U                  # return index of that image.
  else:                       # Otherwise,
    L, R = Children(U)        # cluster U is a result of
                              # a merge of two other clusters
    if Size(L) > Size(R):     # If the first cluster is bigger,
      return Select(L)        # select image from that cluster.
    else:                     # Otherwise,
      return Select(R)        # select from the second cluster.
```

Listing 4.1: Tree-based representative selection algorithm

The density-based approach also tries to select an image from the densest area of a cluster, but it does not rely on hierarchical structure and can be used with any clustering algorithm. In this approach, a density of each point is estimated using Gaussian Kernel.

$$d(i) = \sum_{j \in C} exp(-d(i,j)/h^2)$$

where $d(i)$ is a density at location of image i, C is a cluster, which contains image i, $d(i,j)$ is a cosine distance between descriptors of images i and j and h is a parameter of a Gaussian Kernel. This sum is calculated for each image in the cluster and the image with the highest sum is selected.

The last proposed approach of representative image selection is mean-based. Like the density-based approach, it can be used with any clustering algorithm, but it does not have any parameters. The idea is to choose the image closest to the cluster centroid. If the distribution of descriptors inside the cluster is similar to Gaussian Distribution, then the cluster centroid would be the densest area. The image closest to the centroid could be selected using this formula.

$$\arg\min_i ||d_i - \frac{1}{|C|} \sum_{j \in C} d_j||_2^2$$

where d_i is a descriptor of image i and C is a set of cluster images.

Chapter 5

Experiments

In this chapter, all the proposed system variants are thoroughly explored. The method of removing outliers is tested first. Then the diversity of output subsets for different system variations is studied. The best variation is compared with previous work. Finally, the speed and stability of the proposed system is examined.

5.1 Outlier detection

In this section, the proposed outlier detection methods are investigated.

5.1.1 Removing images with people

Images for the MediaEval dataset were collected from Flickr. Most of the photos of landmarks were uploaded to this database by tourists and such photos often contain people in the foreground. Such images usually do not describe the landmark and are not relevant for most users. So detecting and deleting photos where a significant part is occupied by a person is an important step towards decreasing the number of outliers in the output of the system.

To remove photos with people in the foreground ImageAI [13] library was used. This library contains a pre-trained neural network for detecting objects of several classes one of which is a human class. It provides bounding boxes for each detected object along with a probability of detection.

For each image from the dataset, a set of detections $\{(b_i, p_i)\}_{i=1}^{n}$ was found. b_i is a bounding box and p_i is a probability of detection. These detections were then used to calculate the overall score of "human presence" in each image.

$$s = \sum_{i=1}^{n} f(b_i)p_i$$

In this formula $f(b)$ is a non-negative function representing score of the bounding box b. If s is larger than a certain threshold t, the corresponding image is regarded as an outlier and is removed.

Setting threshold t to 0 would be a bad decision as there might be wrong detections. Additionally, landmarks are naturally crowded, so there could be very few images without people at all. Examples of images with people, which are not labeled as outliers could be found on picture 5.1. So the threshold should be chosen based on labeled data. For that, precision, recall, and f1 measures were considered.

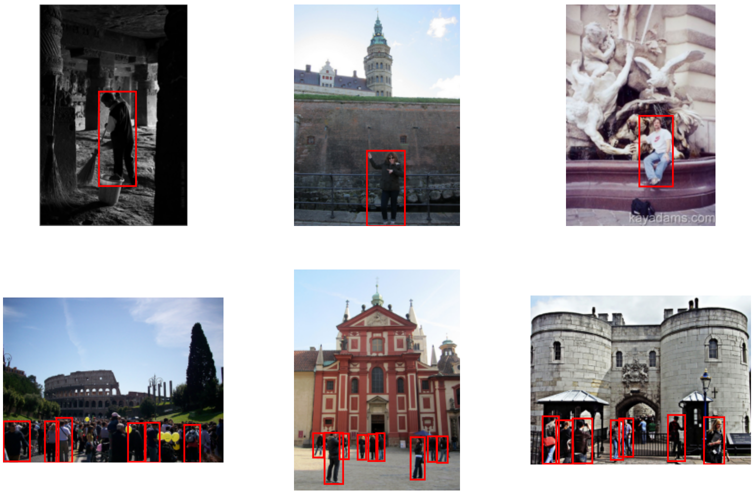
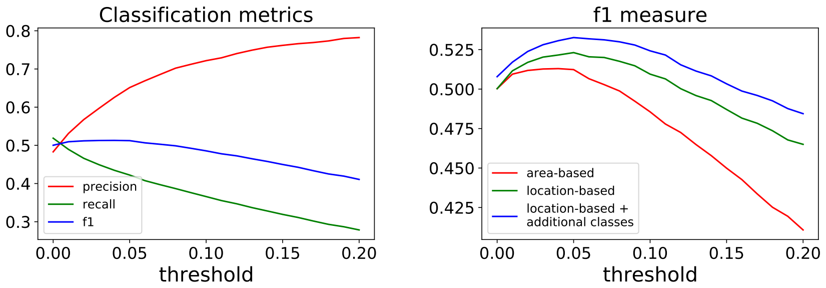


Figure 5.1: MediaEval dataset. Example of images with people, that were: labeled as outliers - top, labeled as relevant - bottom.

For score function f, two options were considered. The first one is the percentage of the image occupied by the bounding box. This corresponds to the following logic of outlier classification. If the total area of all bounding boxes is greater than a certain fraction of image size, then the image is considered as an outlier. Classification metrics for this score function are depicted on figure 5.2a. Maximum of f1 score is reached at $t = 0.05$.

Score function f, which takes into account only the size of the bounding box, has the following drawback. A Bounding box located in a corner of an image will have the same score as a bounding box of the same size located in the center. However, the latter has a bigger effect on image perception. In order to take this into account, a location-based score function is proposed. Score calculation is performed in the following way. The target image is resized to size 63x63. The bounding box is resized correspondingly. In the image space, the Gaussian distribution, centered in the middle of the image, is introduced. Then the score is calculated as a sum of pixel probabilities inside the bounding box. This way boxes at the center of the image get a higher score as center pixels have higher probabilities. The standard deviation for the Gaussian distribution was chosen to be 15, as this value maximizes f1 measure. The location-specific score function improves the quality of outlier detection as shown in figure 5.2b.



(a) : Classification metrics for area-based score function for different threshold values

(b) : Relationship between f1 measure and threshold for 3 different approaches. First approach uses area-based score function and "Pearson" class. Second uses location-based function and "Person" class. Last one uses location-based function and 10 most useful classes

Figure 5.2: Quality of outlier detection

5.1.2 Removing images with other objects

ImageAI [13] library allows detecting several classes of objects. "Person" is the most useful of them, however, it is worth considering other classes too. Figure 5.3 shows the number of images and outliers for the 20 most common classes. From this figure, it can be seen that class "Car" is the second most abundant class in the dataset and 40% of images containing a car are outliers. So deleting images with cars could improve the quality of outlier detection. 10 classes were chosen which improve f1 measure the most. These classes are "person", "car", "bus", "boat", "clock", "chair", "bird", "horse", "dog", and "cat". If the total score of bounding boxes corresponding to these classes exceeds a threshold, the image is regarded as an outlier.

Adding new classes additionally improves the f1 score of outlier classification. A comparison of different approaches could be found in figure 5.2b. From the figure, it is clear that the maximum f1 score is reached at a threshold value of around 0.05 for all three approaches, so this value will be used in the system.

5.1.3 Landmark detection

Another method to reduce the number of outlier images in the output is based on the use of landmark detector. Teichmann et al. in their work [21] introduced a dataset of landmark bounding boxes and trained a Region Proposal Network on that data. This network could be used to remove images where either no landmark was detected or the probability of detection was lower than a threshold. More formally, if p_i is a probability that ith bounding box contains landmark and $\max p_i < t$, then image is regarded as outlier.

It is impossible to perfectly split inliers and outliers with this approach, as there are many outlier images with a high probability of landmark detection. Examples are on figure 5.5. So as before, a compromise between precision

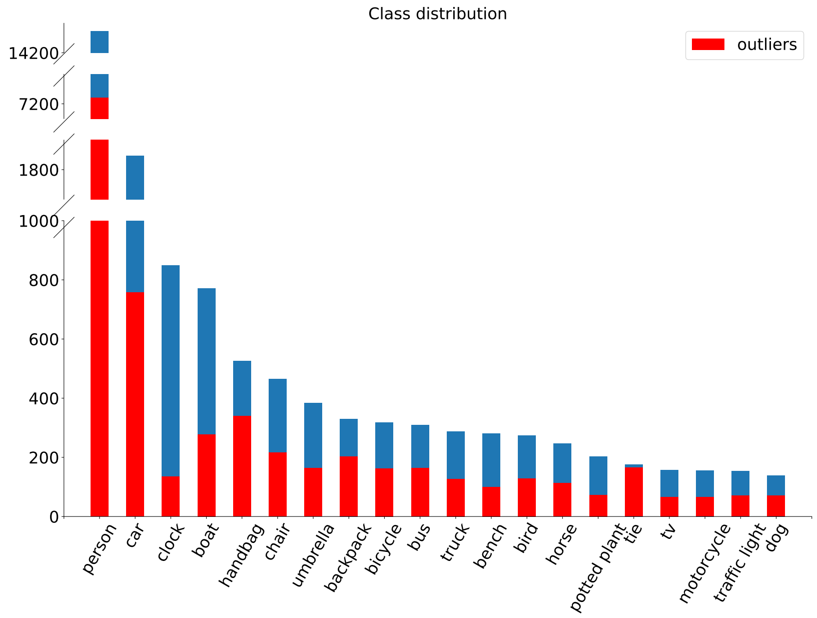


Figure 5.3: Top 20 most abundant classes in the dataset. For each class the number of images and the number of outlier images of that class is depicted

and recall is needed. Figure 5.4 shows dependence of classification metrics on threshold value. The maximum of f1 score is reached at threshold values close to 1. At this point, almost all images are rejected, so this threshold value cannot be accepted. On the other hand, precision stays almost the same on the interval from 0.6 to 0.9 and starts to decrease after the threshold becomes larger than 0.9. So setting the threshold to 0.9 would be a reasonable choice.

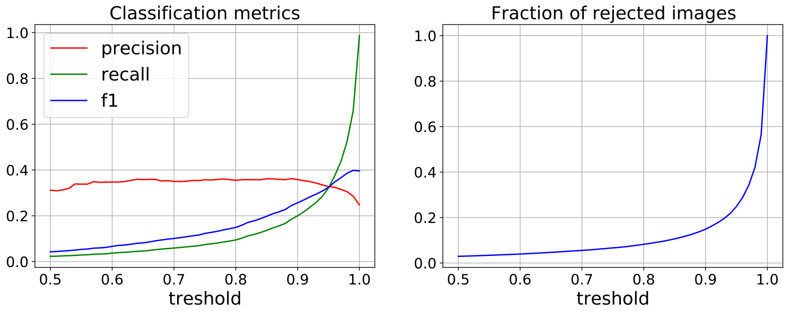


Figure 5.4: Quality of landmark detector. On the right is dependence of precision, recall and f1 scores on threshold value. On the left is dependence of the number of removed images on threshold value.

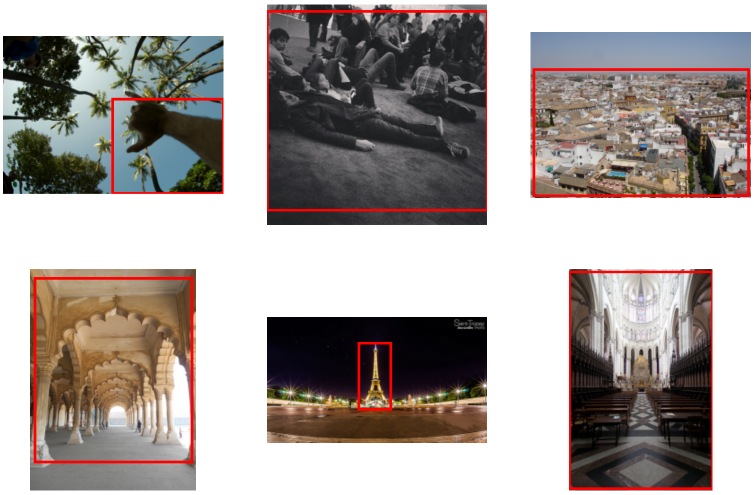


Figure 5.5: Problem with landmark detector. Top - outlier images with more than 95% probability of landmark detection. Bottom - inlier images with probability of detection less than 90%

5.1.4 Summary

Table 5.1 shows metrics for outlier detection of all proposed methods both separately and in combination. Using these methods, the number of outliers was reduced by 60%. Every second detection was wrong, but this is not a big problem as errors were made in images for which it is not obvious whether they are relevant. For example, the landmark detector often fails on interior photographs that are marked as inliers, which is acceptable because in this work interior photographs are considered as outliers according to landmark definition. Also, the number of deleted images is relatively small and the cluster structure should not be damaged.

method	**detected count**	**precision**	**recall**
human detection	9033	0.645	0.44
additional classes	1927	0.38	0.047
landmark detection	4955	0.349	0.096
all	15915	0.497	0.584

Table 5.1: Outlier detection quality

5.2 System parameters and variants

Several variants of diversification algorithms and image descriptors were proposed. Table 5.2 shows different choices for image descriptors. Table 5.3

summarizes two clustering algorithms and their parameters. Finally, table 5.4 presents different methods for selecting representative images.

Descriptor name	**Local features**	**Aggregation method**	**Size**
ResNext_X_MAC	Xth ResNext block	max + PCA	128
ResNext_X_TFIDF	Xth ResNext block	TF-IDF	128
DELF_TFIDF	DELF	TF-IDF	128
DELF_VLAD	DELF	VLAD	128
CSURF_VLAD	CSURF	VLAD	128
X_MN	Mean normalized descriptor.		128

Table 5.2: Image descriptor variants

Clustering algorithm	**Parameters**	**Complexity**
Agglomerative	Distance threshold θ	$O(n^2 log(n))$
DBSCAN	core point neighbour count n neighbour distance ϵ	$O(n^2)$

Table 5.3: Clustering algorithms

Representative choice	**Parameters**	**Compatibility**	**Complexity**
tree-based		agglomerative	$O(\|C\|)$
density-based	h - kernel size	agglomerative + dbscan	$O(\|C\|^2)$
mean-based		agglomerative + dbscan	$O(\|C\|)$

Table 5.4: Representatives selection algorithms

5.3 Diversity Analysis

In this section, all proposed system variations are compared between themselves and with the algorithms of other authors using the MediaEval dataset. The main metric used for comparison is referred to as ClusterRecall, which is a fraction of ground-truth classes represented in the system's output. This metric shows how diverse is the system output based on provided ground truth labels. It was used in the 2014 MediaEval Retrieving Diverse Social Images competition and thus allows comparing the quality of the proposed system to the quality of solutions from that competition. Another metric from the competition was the fraction of inliers in the output and is referred to as relevance (called precision in the competition).

5.3.1 Clustering algorithm choice

In this experiment, two clustering algorithms are compared. The best parameters for both algorithms were found maximizing ClusterRecall@20 on a parameter grid. Results for every possible combination of descriptor type and clustering algorithm choice are provided in the table 5.5. Agglomerative clustering is a superior choice for all suggested descriptors, so from now, DBSCAN will not be used in further experiments.

CR@20	**DBSCAN**	**Agglomerative**
resnext_3_mac	0.277	0.459
resnext_4_mac	0.397	0.471
resnext_3_tfidf	0.391	0.471
resnext_4_tfidf	0.363	0.471
delf_tfidf	0.39	0.469
delf_vlad	0.401	0.488
csurf_vlad	0.411	0.466

Table 5.5: Clustering algorithm comparison

5.3.2 Representative selection choice

Three representative choice algorithms are compared. ClusterRecall@20 for all combinations of descriptor - representative choice are provided in the table 5.6. The tree-based approach shows the worst performance for all descriptors and it will not be used in further experiments. Mean-based and density-based approaches show similar quality, but the density-based approach is more time-consuming. In addition, the density-based approach has a kernel size parameter, which makes the system more complex. Therefore, preference is given to the mean-based approach.

CR@20	**tree-based**	**density-based**	**mean-based**
resnext_3_mac	0.447	0.463	0.459
resnext_4_mac	0.464	0.47	0.471
resnext_3_tfidf	0.463	0.471	0.471
resnext_4_tfidf	0.464	0.477	0.471
delf_tfidf	0.464	0.467	0.469
delf_vlad	0.487	0.488	0.488
csurf_vlad	0.462	0.466	0.466

Table 5.6: Representatives algorithm comparison

5.3.3 Mean-normalization effect

Figure 5.6 shows the distribution of average vector lengths. As can be seen from the graph image descriptors for one landmark have a preferred direction, which means vector space is not used properly. Mean normalization can spread vectors more evenly and potentially improve quality. ClusterRecall@20 for raw and mean-normalized descriptors is presented in table 5.7. From the table, it is clear that mean normalization is a useful technique as it improves diversity for all proposed descriptors.

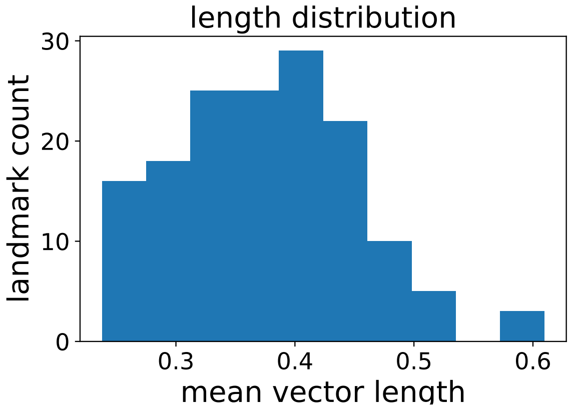


Figure 5.6: Mean vector length distribution for DELF_VLAD image descriptors

CR@20	**no centering**	**with centering**
resnext_3_mac	0.459	0.459
resnext_4_mac	0.465	0.471
resnext_3_tfidf	0.464	0.471
resnext_4_tfidf	0.458	0.471
delf_tfidf	0.467	0.469
delf_vlad	0.474	0.488
csurf_vlad	0.462	0.466

Table 5.7: Mean normalization effect

5.3.4 Descriptor comparison

From the tables 5.5 5.6 5.7 it can be seen that DELF_VLAD descriptor shows the best results with high margin, and ResNext_3_MAC performs the worst. At the same time, it is difficult to order the rest descriptors as the difference in ClusterRecall is not that big.

Graph 5.7 shows the relationship between ClusterRecall and top size for different descriptors. For each descriptor, the best system parameters were found using a grid search. Results show that ResNext_3_MAC, ResNext_3_TFIDF, ResNext_4_MAC, and DELF_TFIDF descriptors perform significantly worse than the rest when the top size is high.

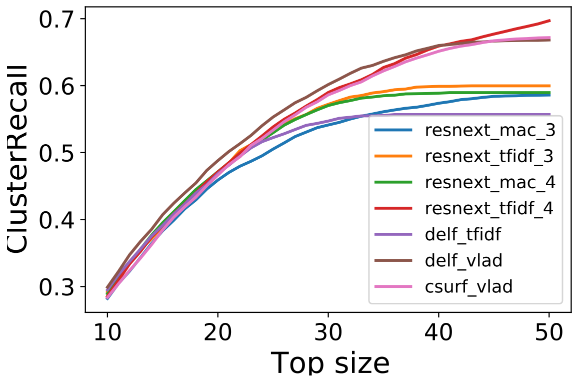
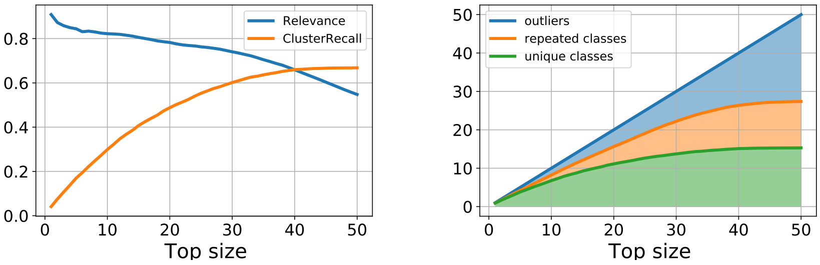


Figure 5.7: Relationship between ClusterRecall and top size for different descriptors

5.3.5 Different subset sizes

In figure 5.8a a ClusterRecall@X and Relevance@X for the best system configuration are plotted for different values of X. As can be expected, ClusterRecall@X increases with X, however it does not reach 1. That is because some ground-truth classes were never picked as representatives, most likely because there are too small and got mixed with some other classes, or because of labeling errors.

Figure 5.8b shows composition of output subsets for the best system configuration. The green area corresponds to the average number of unique classes. At some point, this number stops to grow, because of the limited number of classes (25 for each landmark on average). The orange area corresponds to the average number of repeated classes, which stops to grow as well. And finally, the blue area shows average number of outliers. At some point, only one-image clusters are left. These images are far from any other and most likely are outliers, so at this point number of outliers grows linearly.



(a) : ClusterRecall and Relevance for the best proposed system

(b) : Analysis of the output of the best proposed algorithm. Blue area - average outlier count, Orange area - average number of repeated classes, Green area - average number of unique classes

Figure 5.8: System quality depending on the size of the output

5.3.6 Comparison with the state-of-the-art

The following configuration of the system was used in the comparison. Visual descriptor choice was made in favor of DELF+VLAD with mean normalization. Agglomerating clustering with a distance threshold $t = 0.8$ was used to split images into groups. The mean-based algorithm was used to select the best representative for each group.

Proposed method was compared to state-of-the-art solutions from other authors. SocialSensor [20] and TUW [15] are two best solutions from MediaEval 2014 competition [9] (according to run1). HA [5], EVS [3], FCA [2] are independent researches that provided results of their algorithms on MediaEval dataset. Results of comparison could be found on table 5.8.

As can be seen from the table, the proposed method has the highest ClusteRecall@20 among all previous approaches. Relevance@20 is highest for HA method, however, HA is a hybrid approach and besides visual information it also uses tags, titles, GPS coordinates, and view counts, so this comparison is not fair. If the HA method is excluded from the results, the proposed method has the highest ClusterRecall and Relevance with a high margin.

Method	**ClusterRecall@20**	**Relevance@20**
SocialSensor	0.467	0.783
TUW	0.453	0.776
HA	0.479	**0.858**
EVS	0.419	0.773
FCA	0.414	0.684
Ours	**0.488**	0.825

Table 5.8: ClusterRecall and Relevance metrics for state-of-the-art. Ours - solution proposed in this work

5.4 Working time analysis

This section describes an experiment to measure the speed of the proposed algorithm.

5.4.1 Experimental setup

The proposed solution has been run K times on K different input sets of size N. These input sets were generated in advance so that the generation time is not included in the results. Furthermore, the images that the algorithm considers to be outliers were not used, so all N images were used for clustering. The average time of all K runs was measured and the whole experiment was repeated for different values of N. K was equal to ten thousand, which is a good compromise between the duration of the experiment and the accuracy of the determination of the mean. N ranged from one hundred to one thousand, which covers typical values of the number of images found in an image retrieval system.

This experiment was conducted on a single core of Intel Xeon CPU with a 2.10GHz clock frequency.

5.4.2 Expectations

Asymptotically the most time-consuming phase of the proposed algorithm is complete-link agglomerative clustering, which is $O(N^2 log(N))$. However, finding all pairwise distances might take more time on lower input sizes, as it requires multiplying two big matrices. Finding all pairwise distances is asymptotically $O(N^2)$, so the expectation is that running time will behave like $O(N^2)$ for small N and then smoothly move on to $O(N^2 log(N))$.

5.4.3 Results

Figure 5.9 shows experimental results. It can be seen that the proposed algorithm works reasonably fast even for large input sizes. In real-world scenarios, we can expect even faster results, because of the following reasons. First, only one CPU core was used in the experiment while usually several are available. Second, the actual image retrieval system's response will contain outlier images that will not be used in clustering. In the MediaEval dataset, around 30% of all images are outliers. Third, the proposed algorithm was implemented in Python which is not the fastest solution. C++ or Java is a more appropriate choice for such use cases. The combination of these three factors allows asserting with great confidence that in real-world applications the proposed algorithm can respond to 1000 requests per second, which is a very good indicator.

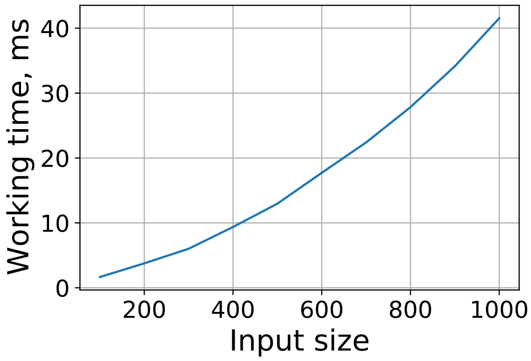


Figure 5.9: Working time of the proposed system for different input sizes

5.5 Stability of the system

In this section stability of the system is analyzed.

5.5.1 Experimental setup

The proposed system in its best configuration was run 20 times for each landmark. N random images were removed from the input, each run different images were chosen. After the run, it was checked if the top 5 images were the same as in the original version when the images were not deleted. This experiment was repeated for N ranging from 1 to 20.

The stability of the system's output mainly depends on the clustering and representative choice algorithms. DBSCAN works much worse in terms of ClusterRecall than agglomerative clustering, so it will not be tested for stability. At the same time, all representative choice algorithms show good quality, so all of them will be checked. Additionally, one more method of representative choice is added. It takes the image with the smallest index from the cluster. Low indices correspond to high Flickr ranks, so this method takes the most relevant images from clusters according to Flickr.

5.5.2 Results

The results of the described experiment are provided in figure 5.10. The X-axis corresponds to a number of deleted images and the Y-axis corresponds to a fraction of runs where the top 5 result changes. As can be seen from the graph even small changes to the input set can influence the result of the system. Min-index (highest relevance) strategy works a little better than representative choice algorithms proposed before, but still, there is a 40% chance that the output will change if only 5 images are deleted. This instability is most likely due to the fact that the cluster structure is not very pronounced. More sophisticated clustering approaches are needed to tackle this problem.

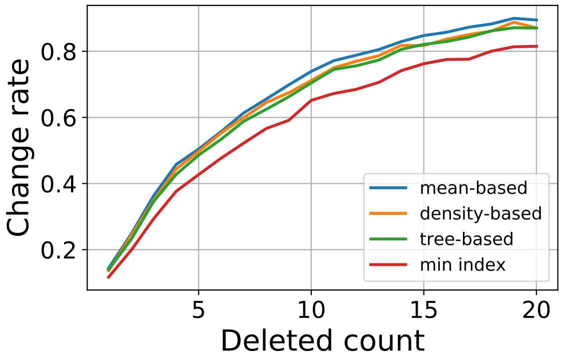


Figure 5.10: The probability that the result will change if a small number of images are removed from the input.

5.6 Examples of clusters

In this section, two examples of the system's output are discussed.

Examples are presented in figures 5.11 and 5.12. Each example consists of 10 largest clusters, located in 10 separate rows. Each cluster in turn is represented by 8 images closest to the cluster centroid. The first column corresponds to the system's output (a subset of diverse images). A ground-truth label from the MediaEval dataset is provided for each image.

5.6.1 Best score example

Figure 5.11 shows an example of clusters for "Louvre Pyramid". The system's output for this landmark has the highest number of unique ground-truth classes. Moreover, clusters found by the algorithm generally correspond to ground-truth clusters. There are, however, some labels that stand out from the others within the cluster (row 2 and 7), but these are most likely labeling errors. Also in rows 8 and 10 a nearby building is depicted, which is not related to the landmark. Still, this building often occurs in the input set and satisfies all landmark properties from the introduction, so these two clusters cannot be considered an error.

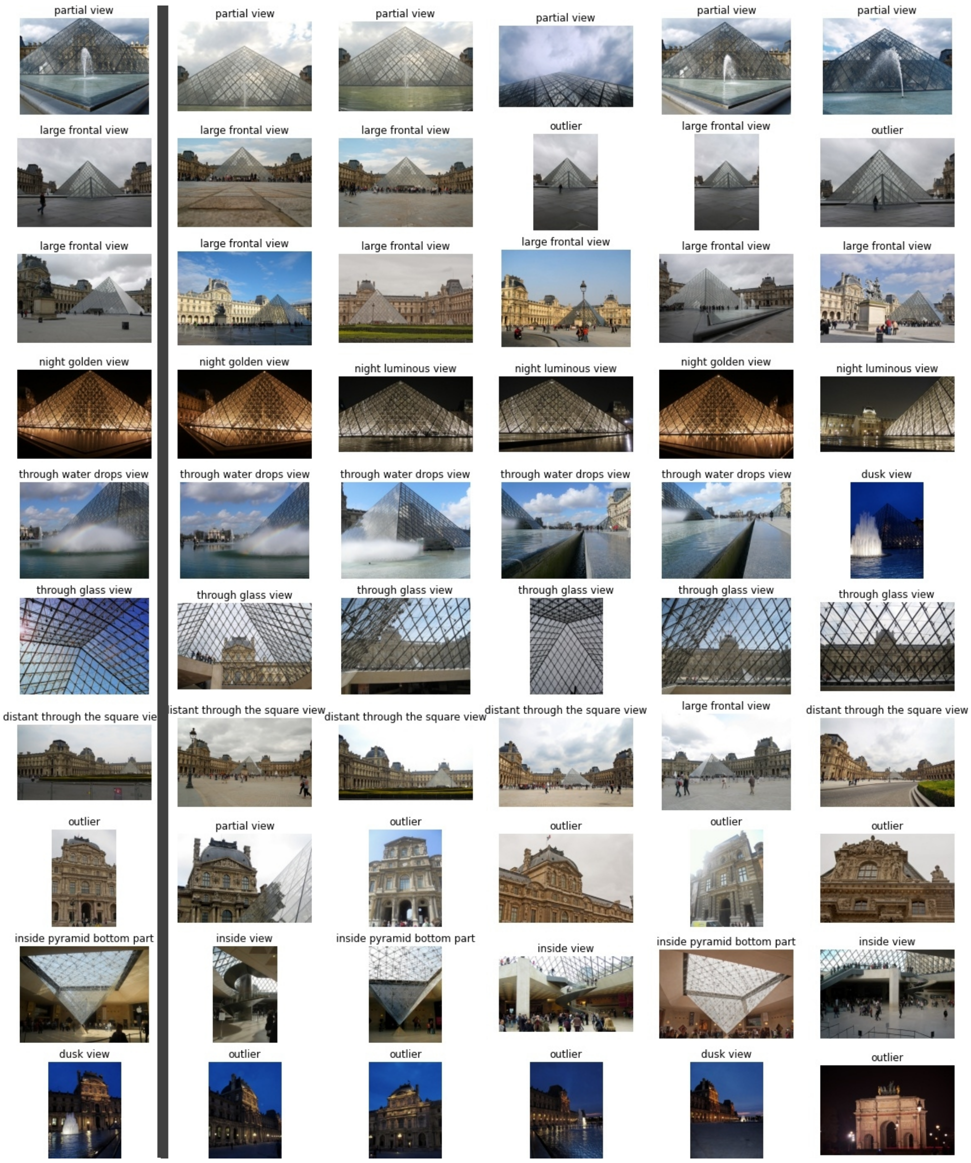


Figure 5.11: Examples of clusters for "Louvre Pyramid" found by the algorithm. Highest number of unique ground-truth labels

5.6.2 Worst score example

Figure 5.12 shows an example of clusters for "Tour Montparnasse". For this landmark, the number of unique classes in the system's output is the lowest. "Tour Montparnasse" is a skyscraper in Paris and the photographers related to it can be divided into two categories, photos of the building itself and photos of the city taken from the top. The photos from the latter category do not show the skyscraper, so they are marked as outliers in the dataset. However, the landmark detector still finds something there and therefore they are not removed. In the case of the images of the Eiffel Tower, this is justified. Thus, even if images that do not represent any landmark are removed, there is still no way to know which landmark is correct, Eiffel Tower or Tour Montparnasse, based on visual information only. Both landmarks often occur in the input collection, so this case is very difficult to solve without using textual or GPS information.

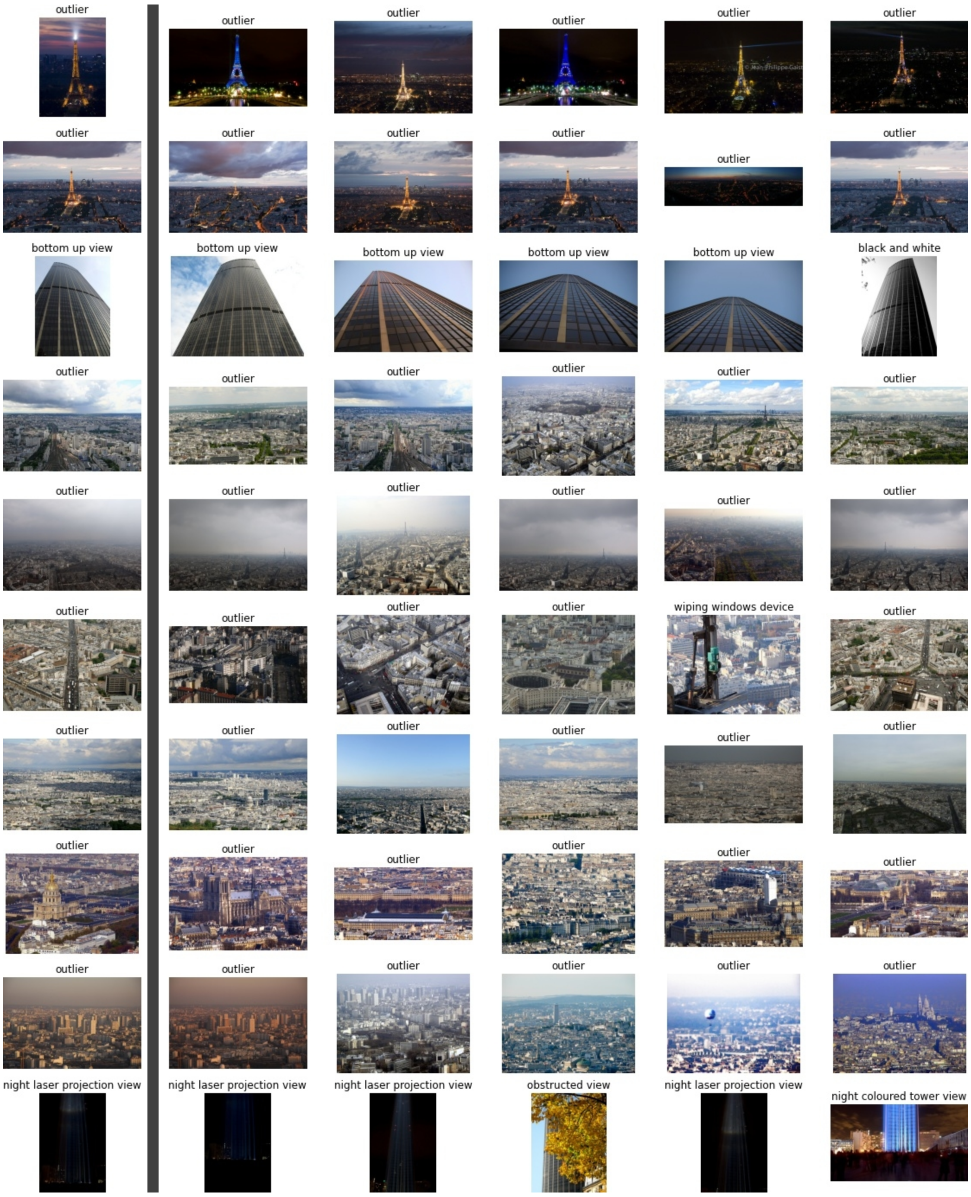


Figure 5.12: Examples of clusters for "Tour Montparnasse" found by the algorithm. Lowest number of unique ground-truth labels

Chapter 6

Conclusion

In this work, an approach for selecting a diverse subset of landmark images was proposed. The main use case for this approach is improving quality for image retrieval systems such as Google images or Flickr, but it can also be used for other problems connected to diversity due to the generality of proposed algorithms. For example, it can be used for selecting a representative set of images for Wikipedia articles or for reducing memory consumption of large image collections.

A method for detecting outlier images was implemented and tested. It is based on the use of two detector networks, one for detecting objects of several common classes, and the other is focused on detecting landmarks only. A location-based method for handling bounding boxes of detected objects was proposed. Its efficiency was proven on the labeled dataset. In total, using the proposed methods, it was possible to remove 60% of outlier images from the MediaEval collection.

Two options for clustering algorithm were considered, DBSCAN and complete-link agglomerative clustering. Experimental results showed that the latter resulted in much better diversity.

For selecting representative images from each cluster three methods were proposed. All of them are focused on finding images in the densest area of the cluster. The first method is based on density estimation with the Gauss kernel, the second one takes the image closest to the cluster centroid, and the last one uses the hierarchical structure of the cluster. In the experiment section, it was shown that the first two methods give the best results in terms of diversity.

Several types of image global descriptors were analyzed. One of them is based on CSURF local features and the rest are based on local features from ResNext neural network. All of them were tested and results on the MediaEval dataset were compared.

The best variation of the proposed algorithm was compared to several state-of-the-art approaches from previous work. The approach from this work was better than the others by a wide margin in terms of both ClusterRecall@20 and Relevance@20.

Finally, time requirements and stability of the proposed approach were investigated. It turned out that even not very optimal implementation of the approach can work quickly enough to be used in real-world search engines. However, proposed method is sensitive to small changes in the input set, which might be inappropriate for some image retrieval systems.

6.1 Future work

From the analysis of image clusters, found by the algorithm, it was clear that the landmark detector does not work as expected. That is because it was trained on images that do not always satisfy landmark properties given in this work. So one of the ways of improving the quality is retraining the landmark detection network on an appropriate image collection.

Another possible direction is to improve the stability of the algorithm result to removing/adding a small number of images to the input set. Currently, even small changes can affect the final result which might not be acceptable. A better clustering approach is needed. It is possible that combining results of different clustering algorithms, like it was done in [19], might be a good approach.

In this work, only two local feature aggregation methods were used. The best performance showed VLAD, which was invented in 2010. Since then better aggregation methods appeared, such as ASMK [22], and using them might further improve diversification quality.

Lastly, the proposed approach relies on visual information only, but for image retrieval systems more data is available, such as GPS, tags, titles, and views. Using this information a better approach can be invented.

Bibliography

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. volume 3951, pages 404–417, 07 2006.

[2] Xaro Benavent, Angel Castellanos, E. Ves, Ana Garcia-Serrano, and Juan Manuel Cigarran Recuero. Fca-based knowledge representation and local generalized linear models to address relevance and diversity in diverse social images. *Future Generation Computer Systems*, 100, 05 2019.

[3] G. Boato, Duc Tien Dang Nguyen, Oleg Muratov, Naif Alajlan, and Francesco Natale. Exploiting visual saliency for increasing diversity of image retrieval results. *Multimedia Tools and Applications*, 75, 03 2015.

[4] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA, 1998. Association for Computing Machinery.

[5] D. Dang-Nguyen, L. Piras, G. Giacinto, G. Boato, and F. G. B. De Natale. A hybrid approach for retrieving diverse social images of landmarks. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2015.

[6] Jing Fu, Xiaojun Jing, Songlin Sun, Yueming Lu, and Ying Wang. C-surf: Colored speeded up robust features. volume 320, pages 203–210, 01 2013.

[7] Bogdan Ionescu, Alexandru Ginsca, Bogdan Boteanu, Adrian Popescu, Mihai Lupu, and Henning Müller. Retrieving diverse social images at mediaeval 2015: Challenge, dataset and evaluation. 01 2015.

[8] Bogdan Ionescu, Alexandru Ginsca, Maia Rohm, Bogdan Boteanu, Mihai Lupu, and Henning Müller. Retrieving diverse social images at mediaeval 2016: Challenge, dataset and evaluation. 10 2016.

[9] Bogdan Ionescu, Adrian Popescu, Mihai Lupu, Alexandru Ginsca, and Henning Müller. Retrieving diverse social images at mediaeval 2014: Challenge, dataset and evaluation. volume 1263, 01 2014.

[10] Tomáš Jeníček. Canonical views extraction from multimedia databases using non-image information. Master's thesis, Czech Technical University, 2017.

[11] Jing Huang, S. R. Kumar, M. Mitra, Wei-Jing Zhu, and R. Zabih. Image indexing using color correlograms. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 762–768, 1997.

[12] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010.

[13] Moses and John Olafenwa. Imageai, an open source python library built to empower developers to build applications and systems with self-contained computer vision capabilities, mar 2018–.

[14] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features, 2016.

[15] João Palotti, Navid Rekabsaz, Mihai Lupu, and Allan Hanbury. Tuw @ retrieving diverse social images task 2014. 10 2014.

[16] Liang Peng, Yi Bin, Xiyao Fu, Jie Zhou, Yang Yang, and Heng Tao Shen. Cfm@mediaeval 2017 retrieving diverse social images task via re-ranking and hierarchical clustering. In *MediaEval*, 2017.

[17] Jean-Michel Renders and Gabriela Csurka. Nle@mediaeval'17: Combining cross-media similarity and embeddings for retrieving diverse social images. In *MediaEval*, 2017.

[18] Maia Rohm, Bogdan Ionescu, Alexandru Ginsca, Rodrygo Santos, and Henning Müller. Retrieving diverse social images at mediaeval 2017: Challenges, dataset and evaluation. 01 2017.

[19] Serwah Sabetghadam, João Palotti, Navid Rekabsaz, Mihai Lupu, and Allan Hanbury. Tuw @ mediaeval 2015 retrieving diverse social images task. 09 2015.

[20] Eleftherios Spyromitros-Xioufis, Symeon Papadopoulos, Ioannis Kompatsiaris, and I. Vlahavas. Socialsensor: Finding diverse images at mediaeval 2014. volume 1263, 10 2014.

[21] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search, 2018.

[22] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. Image search with selective match kernels: Aggregation across single and multiple images. *International Journal of Computer Vision*, 116, 03 2015.

[23] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2016.

[24] B. Yuan, X. Gao, and Z. Niu. Discovering latent aspects for diversity-induced image retrieval. *IEEE MultiMedia*, 25(4):19–33, 2018.

[25] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, page 103–114, New York, NY, USA, 1996. Association for Computing Machinery.