A Multimedia Modular Approach to Lifelog Moment Retrieval

Maxime Tournadre, Guillaume Dupont, Vincent Pauwels, Bezeid Cheikh Mohamed Lmami, and Alexandru Lucian Ginsca

Atos BDS, Bezons, France
maxime@tournadre.org, guillaume.dupont.rm@gmail.com, vincentpauwels8@gmail.com, alexandru-lucian.ginsca@atos.net, bezeid.cheikhmohamedlmami@atos.net

Abstract. Lifelogging offers a mean for people to record their day to day life. However, without proper tools for retrieving specific moments, the large volume of collected data is rendered less useful. We present in this paper our contribution to the ImageCLEF Lifelog evaluation campaign. We describe a modular approach that covers both textual analysis and semantic enrichment of queries and targeted concept augmentation of visual data. The proposed retrieval system relies on an inverted index of visual concepts and metadata and accommodates different versions of clustering and filtering modules. We tested over 10 fully automatic combinations of modules and a human guided approach. We observed that with even minimal human intervention, we can obtain a significant increase of the F1 score, ranking fourth in the competition leader board.

Keywords: Deep Learning · Natural Language Processing · Life logging · Computer Vision · Multimedia

1 Introduction

Recently, interest in lifelogging appeared with the breakthrough of affordable wearable cameras and smartphones apps. It consists in logging the daily life of an individual using various data (i.e. image, biometrics, location etc.). These objects produce a huge amount of personal data, in various forms, that could be used to build great applications to improve the users life (help for the elderly or semantic video search engine). To research this field, a few datasets were constituted to allow researchers to compare methods through workshops, competitions and tasks.

Since 2017, IMAGEClef [1] hosts every year the Lifelog Task [2] [3] in order to compare different approaches on the same environment. This year, the Lifelog Moment Retrieval Task consists in returning a selection of 10 images for specific

topics. IMAGEClef Lifelog provides extracted visual concepts and many metadata for each image as GPS coordinates, the UTC time, the number of steps, the heart rate ect.

New approaches with beyond state-of-the-art performances are expected to be found during this challenge. This would allow, for instance, to build a powerful semantic search engine and to retrieve specific moments easier. Besides this, there are a wide range of possible application domains, which explains the popularity of related workshops. In this paper we detail multiple automatic approaches as well as a human guided one which give users possibility to choose between predefined filters.

2 Related work

The origins of image retrieval can be traced back to 1979 when a conference on Database Techniques for Pictorial Applications was held in Florence [4]. In the beginning, images were first annotated with text and then searched using a text-based approach. Unfortunately, generating automatic descriptive texts for a wide spectrum of images is not feasible without manual help. Annotating images manually is obviously very expensive and it thus limited the text-based methods. In 1992, the National Science Foundation of the United-States organized a workshop to identify new directions in image database management [5]. It was then recognized that indexing visual information on their inherent properties (shape, color etc.) was more efficient and intuitive. Since then, the application potential of image database management techniques has attracted the attention of researchers. In the past decade, many Content-based image retrieval have been developed.

For the Lifelog Moment Retrieval (LMRT) task of the LifeLog challenge, the two major criteria to achieve are relevance and diversity. To improve the relevance, most of the approaches relied on the extraction of concepts or features by existing pre-trained models. This is an important time-saver in these challenges. Some systems [6] chose to augment the data using other models [7] such as ResNet50 [8] trained on ImageNet for visual concepts, or VGG16 trained on Place365 to retrieve places.

Other approaches used one binary classifier per topic or classifiers with a minimum confidence level for each topic. Many teams also used a blur detector to filter the images before selection. To ensure diversity, a team [9] decided to cluster the hyper-features (such as oriented gradient or color histograms features obtained through various models) of the images. Last but not least, all participants used various NLP methods to interpret the topics. In this field, the most popular tools were WordNet [10] and Word2Vec [11].

In the multimedia research field, IMAGEClef is not alone. Many other related workshops and tasks occur on a regular basis such as NTCIR [12] or ACM Multimedia [13]. Launched in the late 1997, NTCIR hold its conference each year in Tokyo. It put an accent on east-Asian language such as Japanese, Korean and Chinese. Like the IMAGEClef competition, a fair number of tasks are

proposed, and among them is a LifeLog action retrieval. ACM Multimedia is an international conference which is held each year in a different country. Different workshops revolving around images, text, video, music and sensor data are proposed and a special part of the conference is the art program, which explores the boundaries of computer science and art.

3 Main components of a modular LMRT system

3.1 Data

Among the data delivered by the organizers, we have chosen to keep some of them and to add others. We kept the qualitative and persistent data and, therefore, did not take into account the following: lat, long, song, steps, calories, glucose, heart rate and distance. The metadata UTC-time was especially very useful and we will detail its contribution in the clustering part.

In the following, we will provide more details on the extraction of visual concepts. The user's images were transmitted with visual concepts extracted from several neural networks :

- The top 10 attributes predicted by using the PlaceCNN [14], trained on SUN attribute dataset [15].
- The top 5 categories and their scores predicted by using the PlaceCNN, trained on Place365 dataset [16].
- Class name, bounding box and score on up to the best 25 objects for each image. They are predicted by using Faster RCNN [17], trained on the COCO dataset [18].

In order to diversify the learning databases we added the following:

- The top 3 labels with their scores predicted by using VGG19. This architecture was created by VGG (Visual Geometry Group) from the University of Oxford [19] and trained on ImageNet [20].
- The top 3 labels with their scores predicted by using ResNet50 trained on ImageNet. We took an implementation of Microsoft which was the winner of ILSVRC 2015 [8].
- The top 3 labels with their scores predicted by using InceptionV3 implemented by Google [21] also trained on ImageNet.
- The top 10 labels with their scores predicted by using Retinanet [22] (object detection) implemented by Facebook.

As another approach, we also built one or two SVM (Support Vector Machines) [23] per topic thanks to its keywords. They have been created based on the FC2's layer of VGG16 implemented in Keras. We have two versions, the first one is fully automatic with an automatic scraping on the web and the second one is built with guided scraping in order to reduce the noise and to better specify the target.

3.2 Queries

The very first step of the online system is the interpretation of a topic. How could we define how an image corresponds to an undefined topic? We tried several techniques based on the idea to obtain a semantic representation of each topic. The way the query is interpreted has a direct impact on the ranking, as it will change the input of the ranking system. We chose to keep a set of keywords for each topic. In our toolbox, Python-RAKE [1] is a package used to extract basic keywords from a topic and WordNet allows to generate pre-defined synonyms. Finally, Word2Vec models provide another way to generate synonyms. It will be more detailed in the next section. The way we combined these tools will be detailed in the next section.

3.3 Ranking

For a given query, our system returns a ranking (full or partial) of the images. "How could we rank them?" is the general problem of this part. We explored several methods, from a basic counting of labels selected with a match, to the average of semantic similarities between labels and a topic. Each ranking is specific to a well-defined method and may be more efficient than another depending on the next processing (i.e. clustering, SVM weights...)

3.4 Filters

Another way to change the images selection, is to adjust some requirements. For instance, if we try to retrieve the images corresponding to "restaurant" in our index we would not find 'fast food restaurant' with an exact match criterion. It may look trivial, but changing the way an image is selected opens a wide range of possibilities, especially within the NLP field. We can easily recover images depending on their visual concepts and metadata. However, we may use some Word2Vec models to apply a similarity threshold or use a Levenshtein distance threshold and much more. In our guided method, we also used conditional filters. For instance, the topic "Driving home" where the user must be driving home from work, was treated with two conditions: the user must have been at work in the last hour and he must arrive at his home in the next fifteen minutes. Combined with a primary ranking, the filters allow to narrow down the selection to suppress the noise. Both precision and recall were usually improved thanks to these approaches.

3.5 Clustering

Since the evaluation method is F@10 - harmonic mean of precision and recall - it is important to augment the diversity of the selected images. That's the reason why it matters to focus on clustering in order to improve diversity without changing relevance. Consequently, we chose to explore two different approaches, one using temporal similarity and one using visual similarity.

[1] Python-RAKE : https://github.com/fabianvf/python-rake

4 Proposed Methods

We realized 12 runs (11 graded) with a combination of different methods for each run. This section aims at explaining these methods. Every run followed the global pipeline described below. The process will be detailed further in this section.

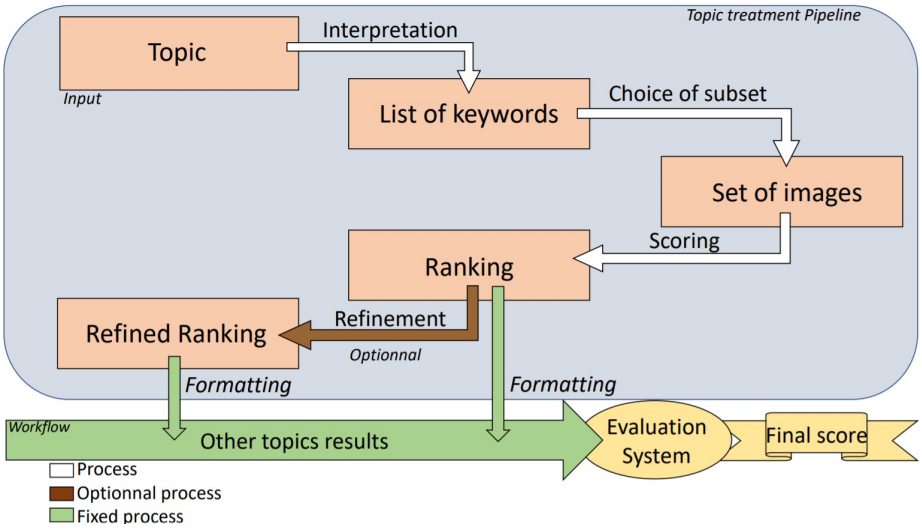

Fig. 1. General pipeline for the treatment of a topic

4.1 Interpretation

Keywords - For each topic, we extract the associated keywords with the Python-RAKE package. Python-RAKE cuts the sentence into words, filters with a list of stop-words and returns the remaining keywords.

Synonyms - This method helps to diversify the topic representation by adding synonyms to the basic extracted keywords. We use WordNet to generate predefined synonyms based on the WordNet hierarchy. Then, we compute the similarity between these synonyms and the basic keywords. This similarity score enables us to select the relevant synonyms. In our runs, the similarity was computed by the GoogleNews-vector-negative300 which is a Word2Vec model.

Another method to compute similarity uses the vector representation of each words in a Word2Vec model. This similarity is useful to filter irrelevant synonyms. Our final method consists of selecting 5 synonyms from WordNet with a threshold (0.5) on the Word2Vec model similarity. Furthermore we keep only synonyms which don't match partially to keywords.

4.2 Choice of subset

We tested different solutions to choose a subset to work on. We usually used one of the following approach to reduce the working set, but in a few runs we kept the whole dataset.

Match - We first create an inverted index. An inverted index in this case will take a label or a metadata as an input and will return all the images owning the input. Using an inverted index is time-saver and useful to interact with the dataset. Then, we recover through the index the images for which at least one label or metadata is part of the keywords.

Partial match - Similarly to match, we select labels that match partially at least one of our keywords. Two words are partial match if and only if one is a part of the other (i.e. "food" and "fast-food"). The verification was done through the clause 'in' in Python. This aims to increase the recall but it introduces some noise as well.

Similarity - Another approach consists in computing the semantic similarity between each key of the index and keywords. Only images that have a label similar enough to a keyword are selected. We didn't use this approach in our submitted run, however a similarity threshold at 0.6 should be efficient.

4.3 Scoring

Label counting - In this basic method, the score is the number of time the image was picked in the inverted index detailed earlier in section 4.2.

Topic Similarity - In this alternative method, we compute the similarity score between each label and keywords. We then pick the maximum similarity for each keyword as a score. The final score for an image is the mean keywords scores.

SVM - In terms of a visual approach, we used transfer learning to train a dedicated binary SVM for each topic. In the end, this provided us with a probability score for each image to be part of the topic based on visual features.

The first step to create our SVM was to establish a training set, so we used the Google Search API to get images from Google Images[2]. The negative examples are shared between the SVMs and these are images of scenes and objects from daily life. Here are the words we chose to use for the negative images : bag, car, cat, daily life photography, dog, fish, footballer, horse, newyork, python, smartphone and sunset . The positives were scrapped by querying the title of the topic and a combination of keywords and synonyms. After the scrapping, we filtered manually the positive to avoid the noise.

[2] https://developers.google.com/custom-search/v1/overview

Every SVM had around 2000 negative and 200 positive examples. Each image was then processed through VGG16 to extract the visual features, which are the input vectors of our SVM. As we intend to use the probability given by the SVM to establish a threshold in a few runs, we tried to use an isotonic calibration. However, as our training set wasn't big enough, we chose to keep the output probability.

SVM performances could be upgraded with a bigger training set, calibration and data augmentation. Unfortunately, we didn't have the time and the resources to deal with at this moment.

4.4 Refinement

Weighting - The refinement by weighting is based on the idea to combine different scoring methods. In fact, we define a new ranking based on the prediction of multiple approaches.

Thresholding - In a similar way, the thresholding considers a minimum score in other methods as a prerequisite to be kept in the ranking.

Visual Clustering - To maximize the diversity of returned images, one approach was to cluster the first 200 images with the highest score on the features extracted by our neural networks. We extracted these features with the second output layer of VGG16 as explained earlier. We clustered the images using the K-means algorithm to retrieve 10 clusters. It greatly improved the diversity of images (i.e. recall score) but inevitably reduced the precision score as only one of many correct images is returned sometimes. It did not improve the overall F@10 score. The second approach for clustering that we will present next proved to be more reliable.

Temporal Clustering - In order to find the maximum number of moments for a specified topic, the temporal approach seemed to be the most logical. We wanted to form different clusters spaced at least one hour, and to detect noise. Thus, we chose to use DBSCAN[3] (Density-based spatial clustering of applications with noise) on the first 250 images with the highest score.

This algorithm, created in 1996 just needs two arguments: a distance and a minimum number of points to form a cluster. Consequently, we converted the UTC times into minutes and then applied the algorithm implemented in Scikit-Learn. The best parameters on average for the topics were one hour for the maximal distance between two points to be considered as the same moment and 5 images minimum to form a moment.

In addition, we also used a blur filter from the OpenCV library to reject all blurry images. Once the blur filter and the clustering were completed, we had to select ten images. Finally, we chose to show the images with the best scores from

[3] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

different clusters. However, many other methods may be used to implement the image selection between clusters.

4.5 Formatting

The last process before sending the run to the evaluation system, is to normalize our score to obtain a confidence score ([0,1]) and combine the topics results.

4.6 Detailed runs

Automatic runs With these selected runs we wanted to establish a few comparisons between our methods. They will be analyzed in section 5. Table 1 shows the automatic runs.

Table 1. Correspondence between automatic runs and methods used

Run	Interpretation	Subset	Scoring	Refinement
1	Keywords	Entire dataset	Topic Similarity	Time Clustering
2	Keywords	Entire dataset	Topic Similarity	Visual Clustering
3	Keywords	Entire dataset	Topic Similarity	—
4	Keywords	Partial match	Topic Similarity	—
5	Keywords + Synonyms	Entire dataset	Topic Similarity	—
6	Keywords + Synonyms	Partial match	Topic Similarity	—
7	Keywords + Synonyms	Partial match	Topic Similarity	Weighting by SVM
8	Keywords + Synonyms	Partial match	Topic Similarity	Tresholding by SVM
9	Keywords + Synonyms	Entire dataset	SVM	—
10	Keywords + Synonyms	Entire dataset	SVM	Visual Clustering
11	Keywords + Synonyms	Entire dataset	SVM	Time Clustering

Interactive runs During our interactive runs, the user had to interpret a topic through filters. The subset selection is done by these filters. However, the keywords and synonyms are still extracted as they will be needed to compute the scoring of a few methods. Table 2 shows the interactive runs.

Table 2. Correspondence between interactive runs and methods used. * The method is picked from automatic runs and varies for each topic. The choice is done by a human

Run	Interpretation	Subset	Scoring	Refinement
12	Automatic*	Filters by Human	Automatic*	Automatic*

5 Results and analysis

Table 3 shows the results obtained by our different runs in the test set. The scoring is F@10, which is the harmonic mean of two measurements computed on each topic. The first one is the precision, whether the ten first are linked to the given topic. The second one is the proportion of different moments found on the given topic. Thus, good precision is not enough to achieve a high score; both recall and precision should be high to reach a great score.

From run 1 to 6, "Mixed run" means that additional concepts were extracted (visual) and that process were done on the label (textual). The "Mixed" aspect of run 7 and 8 is the same, but we add the refinement by the SVM, which rely on the visual approach. Run 9 to 11 do not use the textual approach during the process, however they were trained beforehand.

Table 3. Results obtained on test set

Run	Category	F@10 on test
1	Automatic — Mixed	0.077
2	Automatic — Mixed	0.036
3	Automatic — Mixed	0.036
4	Automatic — Mixed	0.078
5	Automatic — Mixed	0.053
6	Automatic — Mixed	0.083
7	Automatic — Mixed	0.101
8	Automatic — Mixed	0.068
9	Automatic — Visual	0.099
10	Automatic — Visual	*Not evaluated*
11	Automatic — Visual	0.116
12	**Human-Guided**	**0.255**

Table 4 presents the comparison that our runs allow us to do. Thus we are able to define which method worked best in this specific context. An attempt to generalize does not guarantee identical results. Despite the risk for Time Clustering to reduce the precision if there is not enough moments found in the K first images, its usage constantly increased the F@10 score. In a similar way, weight a ranking with SVM's prediction shows an increase in F@10. Combining these two refinement methods may be great.

However, the winner of these runs is clearly the human-guided method. Combining the human understanding and the mixed automatic runs, it reaches 0.255 at F@10 where our automatics runs didn't surpass 0.116. Then it would be interesting, in future works, to establish a hybrid framework which asks the operator to represent topics through filters and then make a ranking with the automatic approaches. It would use SVM to weight the confidence and Time Clustering to ensure cluster diversity. Figure 2 gives an example of how it works.

Table 4. Comparison of the methods used — Bold shows the cases where one method was significantly better than the other

Runs	Compared methods	Best result
1 — 2	Time vs Visual *clustering*	**Time Clustering**
3 — 4 , 5 — 6	Topic Similarity vs Selection	Selection
3 — 5 , 4 — 6	Keywords vs Keywords + Synonyms	Keywords + Synonyms
6 — 9	Classic vs SVM	SVM
7 — 8	Weighted vs Threshold *use of SVM result*	**Weighted**
9 — 11	SVM vs SVM + Time Clustering	SVM + Time Clustering
11 — 12	Automatic vs Human-Guided	**Human-Guided**

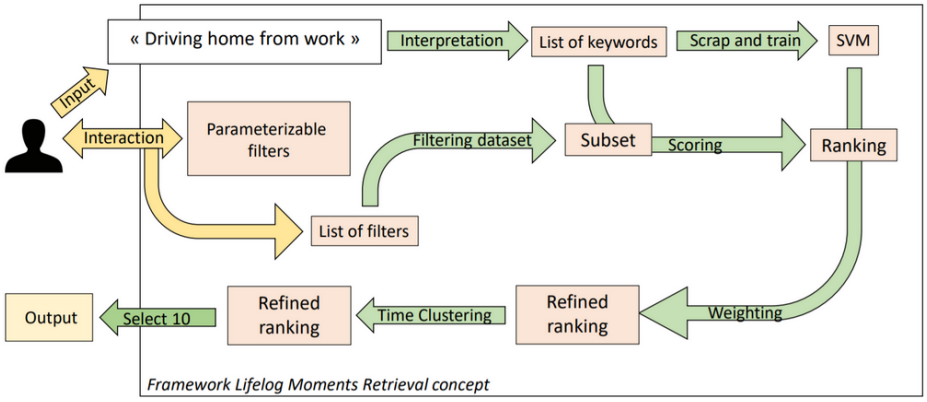

Fig. 2. Example of feasible framework for Lifelog Moments Retrieval

6 Conclusion

In this working-note we presented our approach to the LMRT task of the LifeLog IMAGEClef competition. We chose two frameworks: one fully automatic, and one with human assistance. We extracted feature vectors, and used meta-data on location and time for each image. Our approach relies on linking keywords and their synonyms from topics to our data. We made the human assistance framework available because some filters work better on some topics than others. To cluster the remaining images, we found out that the time clustering had better results than the visual clustering. The sole purpose of clustering was to improve the diversity of moments retrieved. It seems logical that the more images are separated in time, the more they can fit in different moment. Moreover, the DBSCAN algorithm selects automatically the number of clusters and identifies noise images. Therefore, it is superior to the k-means algorithm used in visual clustering. DBSCAN did not achieve great results on visual clustering because the distance between feature vectors is not well defined.

The main difficulty regarding the LMRT task was to process a great deal of multimodal data and find the optimal processing. Fine tuning parameters and thresholds by hand was often a good method but it limits the scalability of the system. As each topic requires slightly different approaches, there is still some work to do to achieve a fully automatic and adaptive system for moment retrieval such as improving the topics interpretation and the automatic setting of appropriate filters.

References

1. B. Ionescu, H. Müller, R. Péteri, Y. D. Cid, V. Liauchuk, V. Kovalev, D. Klimuk, A. Tarasau, A. B. Abacha, S. A. Hasan, V. Datla, J. Liu, D. Demner-Fushman, D.-T. Dang-Nguyen, L. Piras, M. Riegler, M.-T. Tran, M. Lux, C. Gurrin, O. Pelka, C. M. Friedrich, A. G. S. de Herrera, N. Garcia, E. Kavallieratou, C. R. del Blanco, C. C. Rodríguez, N. Vasillopoulos, K. Karampidis, J. Chamberlain, A. Clark, and A. Campello, "ImageCLEF 2019: Multimedia retrieval in medicine, lifelogging, security and nature," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, vol. CLEF 2019 Working Notes. CEUR Workshop Proceedings (CEUR- WS.org), ISSN 1613-0073, http://ceur-ws.org/Vol-2380/ of *Proceedings of the 10th International Conference of the CLEF Association (CLEF 2019)*, (Lugano, Switzerland), LNCS Lecture Notes in Computer Science, Springer, September 9-12 2019.
2. D.-T. Dang-Nguyen, L. Piras, M. Riegler, M.-T. Tran, L. Zhou, M. Lux, T.-K. Le, V.-T. Ninh, and C. Gurrin, "Overview of ImageCLEFlifelog 2019: Solve my life puzzle and Lifelog Moment Retrieval," in *CLEF2019 Working Notes*, CEUR Workshop Proceedings, (Lugano, Switzerland), CEUR-WS.org <http://ceur-ws.org>, September 09-12 2019.
3. D.-T. Dang-Nguyen, L. Piras, M. Riegler, L. Zhou, M. Lux, and C. Gurrin, "Overview of imagecleflifelog 2018: daily living understanding and lifelog moment retrieval," in *CLEF2018 Working Notes (CEUR Workshop Proceedings). CEUR-WS. org¡ http://ceur-ws. org¿, Avignon, France*, 2018.
4. A. Blaser, *Data base techniques for pictorial applications*. Springer-Verlag, 1980.
5. R. Jain, "Nsf workshop on visual information management systems," *ACM Sigmod Record*, vol. 22, no. 3, pp. 57–75, 1993.
6. E. Kavallieratou, C. R. del Blanco, C. Cuevas, and N. García, "Retrieving events in life logging,"
7. C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International Conference on Artificial Neural Networks*, pp. 270–279, Springer, 2018.
8. J. D. Kaiming Hewith Xiangyu Zhang, Shaoqing Ren and Jian, *Resnet winner of the ImageNet Competition 2015 at http://image-net.org/challenges/talks/ilsvrc2015_deep_residual_learning_kaiminghe.pdf*.
9. M. Dogariu and B. Ionescu, "Multimedia lab@ imageclef 2018 lifelog moment retrieval task,"
10. P. University, "About wordnet : https://wordnet.princeton.edu," 2010.
11. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

12. C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, and R. Albatal, "Ntcir lifelog: The first test collection for lifelog research," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 705–708, ACM, 2016.
13. C. Gurrin, X. Giro-i Nieto, P. Radeva, M. Dimiccoli, D.-T. Dang-Nguyen, and H. Joho, "Lta 2017: The second workshop on lifelogging tools and applications," in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1967–1968, ACM, 2017.
14. B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, pp. 487–495, 2014.
15. G. Patterson, C. Xu, H. Su, and J. Hays, "The sun attribute database: Beyond categories for deeper scene understanding," *International Journal of Computer Vision*, vol. 108, no. 1-2, pp. 59–81, 2014.
16. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.
17. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.
18. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.
19. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
20. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
21. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
22. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
23. A. Tzotsos and D. Argialas, "Support vector machine classification for object-based image analysis," in *Object-Based Image Analysis*, pp. 663–677, Springer, 2008.