Automated Lifelog Moment Retrieval based on Image Segmentation and Similarity Scores

Stefan Taubert[1], Stefan Kahl[1], Danny Kowerko[2], and Maximilian Eibl[1]

[1] Chair Media Informatics,
Chemnitz University of Technology, D-09107 Chemnitz, Germany
[2] Junior Professorship Media Computing,
Chemnitz University of Technology, D-09107 Chemnitz, Germany
{stefan.taubert, stefan.kahl, danny.kowerko,
maximilian.eibl}@informatik.tu-chemnitz.de

Abstract. In our working notes we discuss our proposed strategies for the ImageCLEFlifelog LMRT task. We used word vectors to calculate similarity scores between queries and images. To extract important moments and reduce the image amount we used image segmentation based on histograms. We enriched the given data with concepts from pretrained models and got twelve concept types for which similarity scores were calculated and accounted. Furthermore, we used tree boosting as a predictive approach. Our highest F1@10 on the training queries was 27.41% and for the test queries we obtained a maximal F1@10 of 11.70%. All of our models were applicable to generic queries in a fully automated manner.

Keywords: Lifelog Moment Retrieval · Visual Concept · Image Segmentation · XGBoost · Wordembeddings · YOLO · Detectron

1 Introduction

A lifelog is a collection of structured data from the life of a person [12]. The data is collected through sensors, which take images or measure for example heart rate, electrodermal activity, blood sugar levels or geographic coordinates. These sensors can be found in mobile phones, wearable cameras or smart watches. Lifelogs can greatly differ from person to person because of different behaviours.

In most cases, a lifelog contains big amounts of data and can be used for multiple use cases. One example is lifelogging as memory extension for patients who suffer mild cognitive impairment and episodic memory impairment [25]. Furthermore, they are used to analyse people's behaviour, review important moments of the day or to find lost items [31].

As processing of the lifelogs by hand takes enormous effort, computer systems are deployed. The goal of the LMRT task is to create a system which processes shot images and meta data from the life of a person with predefined queries and return appropriate images [6, 14]. For this task lifelogs of two people are given which contain images recorded from a wearable camera and meta data like geographical coordinates and heartrate of each minute of the day.

In this paper we will briefly discuss existing solutions and will afterwards present our proposed strategies. We also review our experiments which we will summarize at the end. We decided to only focus on queries given for user 1 because the training queries only contain queries to user 1 and in the test queries only one query is on behalf of user 2.

In the last years, we worked with deep convolutional networks [23, 16, 24] and we participated in TRECVID challenges [30, 18, 17, 33] and LifeCLEF challenges [15] (BirdCLEF [19, 21, 11, 20], GeoLifeCLEF [32] and PlantCLEF [13]).

2 State of the Art

Lately the topic of lifelog moment retrieval gained increasing attention. In the last year's challenge six teams in total participated whereas in the year 2017 it was only one team [5, 7]. The difficulty in this sort of task lies in the extraction of information from the big amount of data and comparing these with main concepts from the queries, whereby the detection of concepts in the images works really reliable [28, 10, 35, 29].

In such tasks the metrics cluster recall, precision and F1 play a great role because as many different situations as relevant as possible needed to be found. Therefore, methods were found which work automated and some of them used human interaction with relevance feedback. The latter performs generally better than the automated one [36, 22].

The usage of image segmentation before the further query processing can result in better scores [36]. Methods for image segmentation are based on frames characterisation, event segmentation and key frame selection [2, 26, 9]. Frames were characterised through concepts predicted with CNNs [8]. Then agglomerative clustering were used but in regard with time coherence [26].

Furthermore, NLP approaches with word embeddings and LSTMs had been successfully used to compare query and image concepts [34, 1]. Vectors and relevance scores were used for the final comparison [34, 8].

Our approach combines segmentation, NLP methods and similarity scores without human in the loop.

3 Proposed Strategies

Our main strategy was to account multiple similarity scores which were calculated between images and queries. Therefore, we extracted concepts which describe different areas out of a lifelog and compared these with each token from

a query. Furthermore, we include an image segmentation for grouping similar moments and reducing the image amount by removing segments containing only one image. We also present a model based on supervised machine leaning which we trained with similarity scores to process the test queries.

3.1 Segmentation

We read all images which were shot with the wearable camera of a person. To recognise differences between two images we extracted the colour histograms of the images by the day. Then we perform agglomerative clustering by comparing the euclidean distances of the histograms. To remove the hierarchical structure of the clusters, we flattened them afterwards by merging clusters which had a distance lower than a certain threshold.

After those steps we had clusters which did not necessarily contained images from only one moment, for instance the moments user 1 is driving to work and back were in the same cluster. Because of that we separated the clusters into single segments out of coherent images. To reduce the total image amount, we removed segments which contained only one image because these images were mostly blurry or represented changes of the situation.

In the last step we combined segments cluster-wise which were maximum 15 images apart because they were most likely similar and represented the same moment. Finally, we took the segments out of the clusters because we did not needed the clusters anymore.

Results We performed experiments where we tried different thresholds and found a value with which we were able to extract 4302 segments containing in total 21643 out of the 63696 images of user 1. This corresponded to a reduction of 66%. Afterwards we took one image of each segment as representative and calculated the cluster recall for those 4302 images. As result we attained 88.50% cluster recall by anew reduction of 80%. The selection of the representative image did not decrease the cluster recall in comparison to taking all images.

3.2 Extracted Labels

We extracted multiple label types to describe moments from different views. Therefore, we used many of the given data but performed own detections with pretrained models as well. We used following types:

- Categories: We used the given categories which contained Places365 labels predicted through Place CNN. Some of the categories include information about indoor and outdoor, for example *museum indoor* so we decided to extract two extra label types from the categories: Base Categories and Indoor/Outdoor Labels. Those labels described the location of a moment.
- Attributes: We used the given attributes which contained labels from the SUNattribute dataset predicted on Place CNN. With those labels we described the environment of a moment.

- COCO Concepts 1: We used the given concepts which contained labels from the COCO dataset predicted on Faster R CNN. We used them to identify objects in the images.
- COCO Concepts 2: Besides the given concepts we used a pretrained YOLOv3 model to detect COCO labels.
- COCO Concepts 3: Because COCO labels describe common situations very well we extracted COCO labels a third time but with a pretrained Detetron model.
- Open Images Concepts: We used a pretrained model on YOLOv3 again for the detection of labels from the Open Images dataset.
- Image Net Concepts: We used a pretrained model on YOLOv3 again for the prediction of labels from the ImageNet dataset. These concepts were classification labels, they did not identify multiple objects in images.
- Daytimes: We mapped the recording times of the images into four daytimes: *morning*, *afternoon*, *evening* and *night*.
- Locations: We generalised the names of the locations by defining common locations like *store*, *bar* or *bakery*.
- Activities: We used the activities *transport* and *walking*.
- Cities: We extracted from the given time zones the city part.

Besides the raw labels we also calculated the IDF of each label to weight them later in the comparison. For each label type with scores a threshold could be defined at which the label is seen as accurate for a certain image. In those cases all other labels were ignored and the IDFs were adjusted. For labels which were not in the word vectors we tried to find a valid representation and if we could not find any, we ignored the label.

In the following table all label types are shown with their amount of valid labels and if they were generated from the given data.

Table 1. Overview of used label types including if they were generated from given data and their total amount of labels. Overall we got a maximum of 1191 labels.

Label type	Given data	Count of labels
Categories	yes	360
- Base Categories	yes	347
- Indoor/Outdoor	yes	2
COCO Concepts 1	yes	76
COCO Concepts 2	no	72
COCO Concepts 3	no	79
Open Images	no	53
Image Net	no	423
Daytimes	yes	4
Locations	yes	16
Activities	yes	2
Cities	yes	7

3.3 Image Processing

For every image/segment a vector is created for each label type whose size is equal to the amount of labels. There were two possible ways to create these vectors, one for the images and one for the segments.

Image based Vectors For the creation of an image vector we looked a every label of a label type if it occurred in the image. If this was the case, the score of the label was written into the vector. For label types without scores an one was written into the vector if the label occurred in the image. Labels which did not occur were represented by a zero.

Segment based Vectors For the segment based vector creation we either selected the first image of a segment and produced the vector the way described above or we combined the image vectors of all images of a segment by taking the maximum of each label value.

3.4 Query Processing

For the query processing we first had to define the query source which could be either the query description or query title. Then we removed punctuation from the query, lowercased and tokenized it. Afterwards we removed stop words as well as query typical tokens like *find*, *moment* or *u1* and duplicates.

3.5 Results

In the following tables the results of our tokenization process are shown. In Table 2 it is visible that the description token contain token which do not have a clear representation in any of the label types like *view*, *taking*, *using* or *beside*.

Table 2. The results of the tokenized training queries showed that the description token contains multiple token which do not have a clear representation in any of the label types like *view* or *taking*.

Topic	Tokens from description	Tokens from title
1	beside, eating, icecream, sea	icecream, sea
2	drinking, eating, food, restaurant	food, restaurant
3	devices, digital, using, video, watching	videos, watching
4	bridge, photo, taking	bridge, photograph
5	food, grocery, shop, shopping	grocery, shopping
6	guitar, man, playing, view	guitar, playing
7	cooking, food	cooking
8	car, sales, showroom	car, sales, showroom
9	countries, public, taking, transportation	public, transportation
10	book, paper, reading	book, paper, reviewing

The tokenization for the test queries shown in Table 3 also returned tokens which had no clear representations like *using*, *two*, *items*, *plaid* or *red*.

Table 3. The results of the tokenized test queries showed that even the title tokens contained tokens which do not had clear representations like *using* or *two*.

Topic	Tokens from description	Tokens from title
1	items, looking, toyshop	toyshop
2	driving, home, office	driving, home
3	home, inside, looking, refrigerator	food, fridge, seeking
4	either, football, tv, watching	football, watching
5	cafe, coffee	coffee, time
6	breakfast, home	breakfast, home
7	coffee, person, two	coffee, person, two
8	outside, smartphone, standing, using, walking	outside, smartphone, using
9	plaid, red, shirt, wearing	plaid, red, shirt, wearing
10	attending, china, meeting	china, meeting

We recognized differences in the query types comparing training and test queries, for example test query 10 contains a location but in the training queries no locations occurred. Furthermore, test query 7 asked for two persons but a special number of objects were not asked in any of the training queries.

Token Vectors For each query token a vector with the same size as the image vector is created. The entries of the vector were the cosine similarities between the token and each of the labels retrieved from our word vectors. These similarities were inside the interval -1 and 1, whereby 1 represented the highest similarity.

3.6 Vector Comparison

After we retrieved both image/segment vectors ($\boldsymbol{I}$) and token vectors we calculated a similarity score between them. Therefore, we first transformed the co-domain of the token vectors into the interval [0, 1] because we wanted to obtain scores in that range. We called the resulting vector $\boldsymbol{T}$. Then we calculated the norm of the difference vector of $\boldsymbol{I}$ and $\boldsymbol{T}$ and divided it by the square root of the amount of predicted labels n. We obtained a value between 0 and 1 which we subtract from 1 to get the similarity score:

$$\text{Similarity}(\boldsymbol{I}, \boldsymbol{T}) = 1 - \frac{\|\boldsymbol{I} - \boldsymbol{T}\|}{\sqrt{n}}$$

We implemented an option called *ceiling* which replaces the entries (scores) of the image/segment vector with 1. We also include a factor p which potentiate the difference of the vectors and an option *use_idf* which defines if an IDF vector

should be included in the calculation. It contained the IDF of the label if the difference of the vectors was greater than 0.5 and otherwise the reciprocal of the IDF. This way we could boost similarities and punish dissimilarities of rare labels.

3.7 Accounting of the similarity scores

After we calculated similarity scores between all images/segments and tokens for each label type we needed to account them in the next step. Therefore, we created a similarity matrix M_k for each image/segment k whose rows represented each label type j and whose columns represented each token of a query i.

We also included weights v_{ij} for each token and weights w_j for each label type. Weights v_{ij} were the maximum cosine similarity that a token i could have with any label of label type j. If the maximum was negative the weight was set to zero. Weights w_j were set manually.

$$(M_k)_{ij} = v_{ij} \cdot w_j \cdot \text{Similarity}(\boldsymbol{I}_k, \boldsymbol{T}_{ij})$$

Based on this matrix we calculated two different similarity scores. The first method was to calculate the mean of the matrix and the second method was to take the maximum similarities of each label type and calculate the mean. We called the first method *mean* and the second method *labelmax*. We also tried to take the maximum of the token and calculate the mean but the result on the training queries was always worse than that of both other methods.

Token Clustering As extension to the described accounting method we implemented a clustering of the query tokens to group similar tokens. We merged tokens which had a cosine distance lower than 0.5 because they were very similar.

Then we calculated the similarity score as described above but for each cluster. The resulting cluster scores were taken and the mean was computed.

The idea behind this clustering was that if a query contains different things all of them should be considered equally. For example the token *icecream* and *sea* were in different clusters whereby *food* and *restaurant* were in the same cluster. If the token *beach* would be added to the first query it would be added to the second cluster. Without clustering the images with the labels *icecream* and *sea* would be taken less into account than images with all tokens.

3.8 Postprocessing

After we calculated the similarity scores, they were sorted descending for each topic. From the segments we only extracted the first image because it was a valid representative for all other images as shown in 3.1.

To improve the cluster recall we added the option *sort_submission* which defines if only one image per day should be selected for the top submission entries per topic because many of the test queries contained situations which were likely occur only once per day like *driving home from office* or *looking for items in a toyshop*.

4 Resources

We only used resources which were open source. Our word vectors were pretrained GloVe vectors from Common Crawl which had 300 dimensions and a vocabulary of 2.2 million tokens [27].

Furthermore, we used real-time object detection system YOLOv3 in combination with pretrained models to detect labels from the Open Images dataset, ImageNet 1000 dataset and COCO dataset [28]. We also used Detectron with a pretrained Mask R CNN to predict labels from the COCO dataset [10].

4.1 Source Code

We published our source code on GitHub[3]. It was written in Python and uses different third party libraries. The project page provides instructions on how to execute the code. We used an Intel Core i5-7500T in combination with a NVIDIA GeForce GTX 1070 Mobile to run our code.

5 Experiments

In this section we will present our experiments which all based on one base model which combined multiple label types and calculated similarity scores regarding to them.

5.1 Optimization of label types

As we had many parameters we first optimized each label type for itself by running a grid search for different parameter settings. The best comparing settings of a label type were these which achieved the highest F1@10 on the training queries. We tried the following parameters and ran our model without segmentation, with *mean* comparing, unsorted, without label optimization and queries based on the description.

Table 4. To optimize each label type, we ran a grid search with following parameter variations on a model based only on one label type.

Parameter	Description	Variations
t	thresholds Places	0, 0.1, 0.05, 0.01
	other label types	0, 0.9, 0.95, 0.99
idf	use IDF	no, yes
p	exponentiation factor	1, 2, 3
c	use ceiling	no, yes

[3] https://github.com/stefantaubert/imageclef-lifelog-2019

Results We obtained the results shown in Table 5. We found out that the Places Labels achieved a maximal F1 of 23.21%. The label types from the minute based table on the other hand achieved really low scores.

Table 5. This table shows our obtained results for the label optimization experiments. We found out that the Places Labels achieved a maximal F1 of 23.21%. The label types from the minute based table on the other hand achieved really low scores.

Label type	t	idf	p	c	max. F1@10 in %	Count of labels
Places	0	no	1	yes	23.21	360
Raw Places	0	no	1	yes	21.16	347
Indoor/Outdoor	0	no	3	no	1.67	2
Open Images	0	yes	2	yes	6.50	53
COCO Concepts 1	0.9	no	1	no	10.89	72
COCO Concepts 2	0.9	no	1	no	11.57	64
COCO Concepts 3	0.95	no	1	no	14.69	68
Image Net	0.99	no	1	no	12.04	75
Attributes	-	no	1	no	7.98	97
Daytimes	-	no	1	no	3.00	4
Locations	-	no	1	no	0	16
Activities	-	no	1	no	0	2
Timezones	-	no	1	no	0	7

The best score of the 3 COCO Concepts was achieved by Detectron at a threshold of 0.95. The labels from ImageNet performed best using a threshold of 0.99. The IDF values were only useful for the Open Images labels which was really unexpected. The ceiling for the Places related labels was useful which was expected because the scores of the places labels were really low in comparison to the other label types.

5.2 Category Model

Because the category labels achieved such high F1 we decided to process the test queries with that model.

Table 6. The results of run 1 showed that the categories worked much better on the training queries than on the test queries.

Run	Segmentation	Sorted	F1@10 train. set in %	F1@10 test set in %
1	no	no	23.21	3.30

The resulting score was 3.3% which was much lower than expected. The reasons could be that the difference between the training and test queries were to big.

5.3 Visual Concept Model

Our second model was based on all visual concept labels: Attributes, Places and COCO Concepts 1. We ran four different settings. Run 5 was accidentally submitted with the same submission of run 4. The results showed that the score increased after sorting the submissions for the training queries but the test queries only benefited from it in the runs with segmentation. For the segmentation variants we considered all images of a segment.

Table 7. Results of run 2-6 showed that segmentation increased the test scores enormously. A reason could be that the limited amount of possible images increased the cluster recall.

Run	Segmentation	Sorted	F1@10 train. set in %	F1@10 test set in %
2	no	no	12.81	3.90
3	no	yes	17.20	3.00
4, 5	yes	no	12.25	8.60
6	yes	yes	12.67	9.00

It was noticeable that the run with segmentation achieved almost three times better scores than the normal runs. One reason could be the decreased amount of possible images which increased the probability of taking relevant images from different clusters. The difference between the training and test scores was this time lower than in run 1 but still high especially for run 2 and 3.

5.4 Predictive Model

We decided to create a model which uses tree boosting as predictive approach. We used the similarity scores from run 2 and 4 for training a classifier. Therefore, we took 20% of the irrelevant images and 20% of the images of each cluster for the evaluation set and the rest for the training set. Then we trained our model with XGBoost and binary logistic regression and depth three for the best log loss [4]. Afterwards we predicted the images for each test topic.

Table 8. Runs 7 and 8 showed that the predictive approach was not suitable for such a task because the similarities were simply memorized and could not be used to predict unknown queries.

Run	Segmentation	Sorted	F1@10 train. set in %	F1@10 test set in %
7	no	no	19.43	0.00
8	yes	no	11.78	4.60

Results of run 7 and 8 showed that the predictive approach was likely not suitable for such a task because the similarities simply were memorized and

could not be used to predict unknown queries. In run 7 no relevant images were found and in run 8 the found images could be just random images because in total there were just 4302 images to select from.

5.5 Visual Concept Metadata Model

The Visual Concept Metadata Model was an attempt to improve the Visual Concept model by adding the given metadata. We did not sort the submissions in these runs and switched over to the labelmax comparing method and optimized the labels for all following runs. We also included the weights for each token. For the segmentation variant we considered only the first image of a segment.

Table 9. The results of run 9 and 10 were really bad for both test and training queries. The main reason could be that the given metadata worsened the scores.

Run	Segmentation	Sorted	F1@10 train. set in %	F1@10 test set in %
9	no	no	12.27	1.70
10	yes	no	8.15	1.40

The scores were really low for both test and training queries. The main reason could be that the given metadata pulled the scores down because they were really unsuitable as found out in the label optimization experiments. We obtained mostly better scores on the training queries with the *labelmax* comparing method which was why we used this method. It could be guessed that an adjusting of weights for the metadata could be resulted in higher scores.

5.6 Extended Visual Concept Model

As the extension with metadata did not increased the score, we tried to extend the Visual Concept Model with the self predicted concepts COCO Concepts 2 & 3, Open Images and Image Net Concepts. This time, we decided to take the titles as queries to update the previous model.

Table 10. With run 11 we achieved our highest score with 11.70%. Run 12 showed that the segmentation decreased both scores.

Run	Segmentation	Sorted	F1@10 train. set in %	F1@10 test set in %
11	no	no	26.16	11.70
12	yes	yes	14.25	4.00

Using this model we achieved our highest score with 11.70% and we also had the highest score for the training queries with 26.16%. The segmentation decreased the score for both training and test queries. We had an increase of 7.8% F1 by taking the extra label types into account.

5.7 Extended Visual Concept Metadata Model

We combined the two last mentioned models into one big model which contained twelve label types. We also used token clustering in our models and added the common location *costa coffee* because two topics were asking for coffee. We obtained the results shown in Table 11.

Table 11. Runs 13, 14 and 15 showed that the extension with the metadata did not improve the score but worsened it.

Run	Segmentation	Sorted	Label weights	F1@10 train. set in %	F1@10 test set in %
13	no	no	no	21.62	6.20
14	yes	yes	no	13.01	1.10
15	no	no	yes	27.41	8.70

From the results we could read out that the score could be increased by weighting the label types. The segmentation caused a decrease in the score again like in the previous model. A new highest score could not be accomplished. The following images in Fig. 1 show the returned images for this run.


(a) Top 1: 1st image

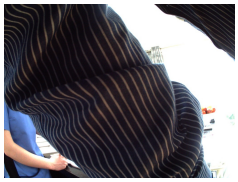

(b) Top 2: 1st image


(c) Top 3: 1st image


(d) Top 6: 5th image


(e) Top 7: 7th image


(f) Top 8: 2nd image


(g) Top 9: 1st image


(h) Top 10: 1st image

Fig. 1. The returned images from run 15 showed that our model had found some relevant images at top positions. Topics 1, 3, 6, 7 and 8 contained at least one relevant image whereby for topics 2, 4, 5, 9, 10 no relevant images were returned.

For topic 1 correct images from a toyshop were found. All top ten images for topic 2 showed images of the same day when u1 was driving to work but not back home. Three relevant images were returned for topic 3 but none were found

for topic 4 and 5. The reason for the latter may be that the uninformative token *time* was still in the query after the tokenization process.

Almost all selected images for topic 6 were relevant but were taken on the same day which led to a small cluster recall. For topic 7 we got two different relevant moments and for topic 8 half of the images were relevant. For the 9th topic we only got an image of u1 wearing not a plaid but a striped shirt and the images returned for the last topic contained the airport in Shanghai with many people but not a meeting.

6 Conclusion

We tried different strategies for the lifelog task. The best results could be achieved without using segmentation. One reason for that could be, that too much of the relevant images were removed or that the histogram based approach was not precise enough. Sorting of the submissions was only useful in the models with few label types. In our evaluation, we found out that the weighting of the tokens improved the scores immense, so did the use of thresholds for the label types. We could also achieve an improvement in the score by considering only the highest similarity score of all tokens per label type. Token clustering led to an increase of the score, too.

Our predictive model showed that this approach may not be suitable for such tasks. The big differences between the training and test queries led to great differences in the F1 scores. We found out that the usage of the title did improve the scores because less uninformative words were included.

We could have done more experiments which would have shown that some parameters had more influences for the resulting scores. Weighting of the label types might improve our best run but because of this great amount of parameters it was difficult to find the best experiments.

To improve our model, we could include dissimilarity scores and find labels which may not occur in the images even if the similarity scores were high. We also could include more of the metadata like the heart rate and more external data, for example poses from open pose [3] to detect hands in the images to prevent the detection of hands as persons.

References

1. Abdallah, F.B., Feki, G., Ezzarka, M., Ammar, A.B., Amar, C.B.: Regim lab team at imageclef lifelog moment retrieval task 2018
2. Bolanos, M., Mestre, R., Talavera, E., Giró-i Nieto, X., Radeva, P.: Visual summary of egocentric photostreams by representative keyframes. In: 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). pp. 1–6. IEEE (2015)
3. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In: arXiv preprint arXiv:1812.08008 (2018)

4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794. ACM (2016)
5. Dang-Nguyen, D.T., Piras, L., Riegler, M., Boato, G., Zhou, L., Gurrin, C.: Overview of ImageCLEFlifelog 2017: Lifelog Retrieval and Summarization. In: CLEF2017 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Dublin, Ireland (September 11-14 2017)
6. Dang-Nguyen, D.T., Piras, L., Riegler, M., Tran, M.T., Zhou, L., Lux, M., Le, T.K., Ninh, V.T., Gurrin, C.: Overview of ImageCLEFlifelog 2019: Solve my life puzzle and Lifelog Moment Retrieval. In: CLEF2019 Working Notes. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org <http://ceur-ws.org>, Lugano, Switzerland (September 09-12 2019), ISSN 1613-0073, http://ceur-ws.org/Vol-2380/
7. Dang-Nguyen, D.T., Piras, L., Riegler, M., Zhou, L., Lux, M., Gurrin, C.: 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 (September 10-14 2018)
8. Dogariu, M., Ionescu, B.: Multimedia lab@ imageclef 2018 lifelog moment retrieval task
9. Doherty, A.R., Smeaton, A.F., Lee, K., Ellis, D.P.: Multimodal segmentation of lifelog data. In: Large Scale Semantic Access to Content (Text, Image, Video, and Sound). pp. 21–38. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE (2007)
10. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. https://github.com/facebookresearch/detectron (2018)
11. Goëau, H., Glotin, H., Planqué, R., Vellinga, W.P., Kahl, S., Joly, A.: Overview of birdclef 2018: monophone vs. soundscape bird identification. CLEF working notes (2018)
12. Gurrin, C., Smeaton, A.F., Doherty, A.R., et al.: Lifelogging: Personal big data. Foundations and Trends® in information retrieval **8**(1), 1–125 (2014)
13. Haupt, J., Kahl, S., Kowerko, D., Eibl, M.: Large-scale plant classification using deep convolutional neural networks
14. Ionescu, B., Müller, H., Péteri, R., Cid, Y.D., Liauchuk, V., Kovalev, V., Klimuk, D., Tarasau, A., Abacha, A.B., Hasan, S.A., Datla, V., Liu, J., Demner-Fushman, D., Dang-Nguyen, D.T., Piras, L., Riegler, M., Tran, M.T., Lux, M., Gurrin, C., Pelka, O., Friedrich, C.M., de Herrera, A.G.S., Garcia, N., Kavallieratou, E., del Blanco, C.R., Rodríguez, C.C., Vasillopoulos, N., Karampidis, K., Chamberlain, J., Clark, A., Campello, A.: ImageCLEF 2019: Multimedia retrieval in medicine, lifelogging, security and nature. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 10th International Conference of the CLEF Association (CLEF 2019), LNCS Lecture Notes in Computer Science, Springer, Lugano, Switzerland (September 9-12 2019)
15. Joly, A., Goëau, H., Botella, C., Kahl, S., Poupard, M., Servajean, M., Glotin, H., Bonnet, P., Vellinga, W.P., Planqué, R., Schlüter, J., Stöter, F.R., Müller, H.: Lifeclef 2019: Biodiversity identification and prediction challenges. In: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (eds.) Advances in Information Retrieval. pp. 275–282. Springer International Publishing, Cham (2019)
16. Kahl, S., Hussein, H., Fabian, E., Schlohauer, J., Thangaraju, E., Kowerko, D., Eibl, M.: Acoustic event classification using convolutional neural networks. In: Eibl, M., Gaedke, M. (eds.) INFORMATIK 2017. pp. 2177–2188. Gesellschaft fr Informatik, Bonn (2017)

17. Kahl, S., Richter, D., Roschke, C., Heinzig, M., Kowerko, D., Eibl, M., Ritter, M.: Technische universitat chemnitz and hochschule mittweida at trecvid instance search 2017
18. Kahl, S., Roschke, C., Rickert, M., Richter, D., Zywietz, A., Hussein, H., Manthey, R., Heinzig, M., Kowerko, D., Eibl, M., Ritter, M.: Technische universitat chemnitz at trecvid instance search 2016 (11 2016)
19. Kahl, S., Wilhelm-Stein, T., Hussein, H., Klinck, H., Kowerko, D., Ritter, M., Eibl, M.: Large-scale bird sound classification using convolutional neural networks
20. Kahl, S., Wilhelm-Stein, T., Klinck, H., Kowerko, D., Eibl, M.: A baseline for largescale bird species identification in field recordings
21. Kahl, S., Wilhelm-Stein, T., Klinck, H., Kowerko, D., Eibl, M.: Recognizing birds from sound-the 2018 birdclef baseline system
22. Kavallieratou, E., del Blanco, C.R., Cuevas, C., García, N.: Retrieving events in life logging
23. Kowerko, D., Kahl, S.: Ws34 - deep learning in heterogenen datenbestnden. In: Eibl, M., Gaedke, M. (eds.) INFORMATIK 2017. p. 2141. Gesellschaft fr Informatik, Bonn (2017)
24. Kowerko, D., Richter, D., Heinzig, M., Kahl, S., Helmert, S., Brunnett, G.: Evaluation of cnn-based algorithms for human pose analysis of persons in red carpet scenarios. In: Eibl, M., Gaedke, M. (eds.) INFORMATIK 2017. pp. 2201–2209. Gesellschaft fr Informatik, Bonn (2017)
25. Lee, M.L., Dey, A.K.: Lifelogging memory appliance for people with episodic memory impairment. In: Proceedings of the 10th International Conference on Ubiquitous Computing. pp. 44–53. UbiComp '08, ACM, New York, NY, USA (2008). https://doi.org/10.1145/1409635.1409643, http://doi.acm.org/10.1145/1409635.1409643
26. Lin, W.H., Hauptmann, A.: Structuring continuous video recordings of everyday life using time-constrained clustering. In: Multimedia Content Analysis, Management, and Retrieval 2006. vol. 6073, p. 60730D. International Society for Optics and Photonics (2006)
27. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), http://www.aclweb.org/anthology/D14-1162
28. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 91–99. Curran Associates, Inc. (2015), http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf
30. Ritter, M., Heinzig, M., Herms, R., Kahl, S., Richter, D., Manthey, R., Eibl, M.: Technische universität chemnitz at trecvid instance search 2014. In: Proceedings of TRECVID Workshop, Orlando, Florida, USA. vol. 7 (2014)
31. Sellen, A., Whittaker, S.: Beyond total capture: a constructive critique of lifelogging. Communications of the ACM (2010)
32. Taubert, S., Mauermann, M., Kahl, S., Kowerko, D., Eibl, M.: Species prediction based on environmental variables using machine learning techniques. CLEF working notes (2018)
33. Thomanek, R., Roschke, C., Manthey, R., Platte, B., Rolletschke, T., Heinzig, M., Vodel, M., Kowerko, D., Kahl, S., Zimmer, F., et al.: University of applied sciences mittweida and chemnitz university of technology at trecvid 2018

34. Tsun-Hsien Tang 12, Min-Huan Fu, H.H.H.K.T.C., Chen, H.H.: Visual concept selection with textual knowledge for activities of daily living and life moment retrieval. In: 19th Working Notes of CLEF Conference and Labs of the Evaluation Forum, CLEF 2018. vol. 2125. CEUR-WS (2018)
35. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 487–495. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5349-learning-deep-features-for-scene-recognition-using-places-database.pdf
36. Zhou, L., Piras, L., Riegler, M., Boato, G., Nguyen, D., Tien, D., Gurrin, C.: Organizer team at imagecleflifelog 2017: baseline approaches for lifelog retrieval and summarization (2017)

Appendix

Table 12. This table shows the settings of the model parameter for each run.

Run	Query source	Comparing	Token weights	Token clustering	Label opt.	Max. train.	Max. test
1-8	description	mean	no	no	no	23.21%	9.00%
9, 10	description	labelmax	yes	no	yes	12.27%	1.70%
11, 12	title	labelmax	yes	no	yes	26.16%	11.70%
13-15	title	labelmax	yes	yes	yes	27.41%	8.70%

Table 13. In this table the weights of the label types were listed for each run. The weights for the last run were obtained through random search on the training queries.

Label type	Run 1	2-8	9,10	11,12	13,14	15
Places	1	1	0	0	0	0
- Raw Places	0	0	1	1	1	0.8
- Indoor/Outdoor	0	0	1	1	1	0.4
COCO Concepts 1	0	1	1	1	1	0.1
COCO Concepts 2	0	0	0	1	1	0.3
COCO Concepts 3	0	0	0	1	1	0.7
Open Images	0	0	0	1	1	0.2
Image Net	0	0	0	1	1	0.7
Attributes	0	1	1	1	1	0.8
Daytimes	0	0	1	0	1	1
Locations	0	0	1	0	1	1
Activities	0	0	1	0	1	1
Cities	0	0	1	0	1	1