VC-I2R@ImageCLEF2017: Ensemble of Deep Learned Features for Lifelog Video Summarization

Ana Garcia del Molino^{*,1,2}, Bappaditya Mandal^{1}, Jie Lin^{1}, Joo Hwee Lim^{1},
Vigneshwaran Subbaraju^{3} and Vijay Chandrasekhar^{1},

^{1}Visual Computing Department, Institute for Infocomm Research, A*STAR, Singapore,
^{2}School of Computer Science and Engineering, NTU, Singapore,
^{3}Singapore Bioimaging Consortium, A*STAR, Singapore,
{stugdma*, bmandal, lin-j, joohwee, vijay}@i2r.a-star.edu.sg;
Vigneshwaran_Subbaraju@sbic.a-star.edu.sg
*Contact author

Abstract. In this paper we describe our approach for the ImageCLEF-lifelog summarization task. A total of ten runs were submitted, which used only visual features, only metadata information, or both. In the first step, a set of relevant frames are drawn from the whole lifelog. Such frames must be of good visual quality, and match the given task semantically. For the automatic runs, this subset of images is clustered into events, and the key-frames are selected from the clusters iteratively. In the interactive runs, the user can select which frames to keep or discard in each interaction, and the clustering is adapted accordingly. We observe that the more relevant features to be used depend on the context and the nature of the input lifelog.

1 Introduction

With the rising availability of affordable wearable recording devices in the market (e.g. SenseCam or Narrative Clip), as well as the presence of countless mobile apps for fitness and lifestyle tracking, one may resort to personal Life-logging solutions to create memory collections or monitor their own life. However, little support is available for the browsing of such digital memories, and as a result, our phones and computers can get filled with personal information we may never revisit or analyze.

To solve this problem, ImageCLEF LifeLog Task [6, 13] aims to bring attention of researchers from diverse fields to study, evaluate and propose new methodologies to address the challenging problems in lifelog video summarization tasks. This rigorous comparative benchmarking would help the various research groups to evaluate their existing methodologies among each others on a common platform and also spur new thinking for solving the long standing key problems. In the following section we discuss these key challenges and related work in the literature.

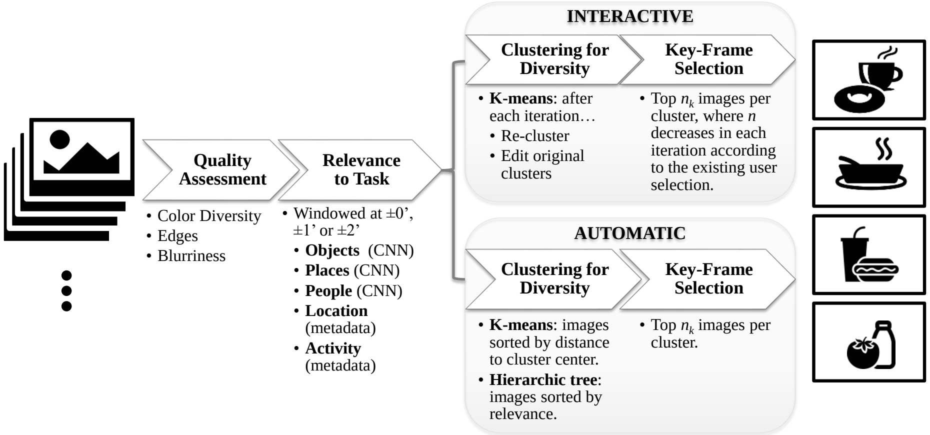

Quality Assessment
• Color Diversity
• Edges
• Blurriness
Relevance to Task
• Windowed at ±0', ±1' or ±2'
• Objects (CNN)
• Places (CNN)
• People (CNN)
• Location (metadata)
• Activity (metadata)
INTERACTIVE
Clustering for Diversity
Key-Frame Selection
• K-means: after each iteration…
• Re-cluster
• Edit original clusters
• Top n_k images per cluster, where n decreases in each iteration according to the existing user selection.
AUTOMATIC
Clustering for Diversity
Key-Frame Selection
• K-means: images sorted by distance to cluster center.
• Hierarchic tree: images sorted by relevance.
• Top n_k images per cluster.

Fig. 1: Pipeline of our runs. The input image stream is first analyzed to remove bad quality images. Then, the remaining frames are assessed in terms of relevance to the task. The top images are selected to be clustered into different events. Finally, at least one key-frame is selected from each event.

1.1 Related Work

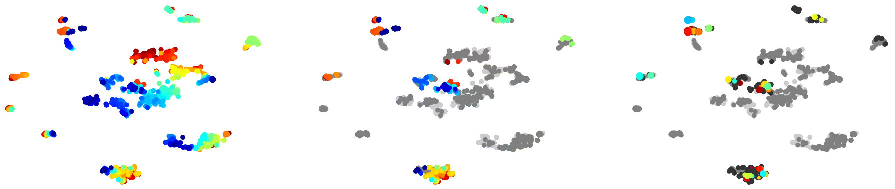
Summarization of egocentric videos has become a problem of much interest. In a recent comprehensive survey on the summarization of high temporal resolution videos [19], we note the importance of context dependency to generate better summaries. For low resolution videos, two recent surveys review the methods for better summarizing Lifelogs for memory augmentation [11] and storytelling [4]. The authors of [11] observe that our memory recall accuracy is directly related to how different that episode is from the rest of memories.

In [4] four steps in the process are identified: informative pictures filtering (removal of blurred or dark images and those with useless content [24]), episode segmentation (using low level features k-means, eigenvalues or graphs, and time-dependent methods [5, 9]), summarization (based on representativeness, or relying in the presence of important people/objects [16]) and retrieval (by means of encoded context, people, objects or activities [10, 15]).

All the aforementioned surveys conclude that richer semantic-level features are needed to encode the different episodes. Moreover, the key-frames included in a summary should be diverse, informative, and good memory triggers.

2 Proposed Methodologies

We compare a series of summarization methods that *(i)* filter out uninformative images; *(ii)* rank the remaining images according to how well they match the given query; *(iii)* cluster the top ranked images into a series of events; and *(iv)* select, in an iterative manner, as many images per cluster as to fill the length budget. Fig. 1 shows the flow of our proposed approaches.



(a) Input Stream (b) Relevant Images (c) Final Summary

Fig. 2: Flow of summary generation in the proposed methodology. Each scatter plot represents the frames in the lifelog for task 10. (a) Input data. Colors correspond to temporal evolution. (b) Remaining images after quality preprocessing (bad quality images filtered out in light gray) and relevance assessment (irrelevant images filtered out in darker gray). (c) Final summary. The keyframes are represented with different colors to map the clusters, and larger size to represent higher positions in the final summary ranking. (Best viewed in color)

2.1 Pre-processing Techniques

The incoming frame stream is pre-processed to evaluate the quality and informativeness of the images. All frames below a certain quality threshold are then discarded. The quality rate is obtained by combining the following scores:

Blurriness assessment We use two different methodologies:

Modified Laplacian: This method applies a non-linear filter operation on an image and filters out the prominent edges from the input image using a Gaussian kernel along the x- and y- directions. The edge score is taken as the mean value of all absolute values.

Variance Of Laplacian: The input image is convolved with the Laplacian operator (3×3 kernel). Then, the variances (*i.e.* standard deviation squared) of the response are computed. All values over a certain threshold are averaged to obtain a blurriness score.

Color diversity Images with very homogeneous colors are deem to be uninformative, since this usually means there are few different objects in the image. We compute the histograms of the quantized RGB values, and find the color diversity score based on the frequency of the predominant color.

2.2 Task Relevance Retrieval

Since the summaries are query-driven, we first need to evaluate the relevance of each image to the given task. For each task, we define a set of objects and places to be found or avoided in the target images (as listed in Table 1). Additionally, from the location and activity information available in the metadata, we define the relevant locations and activities, and the ones to avoid strictly.

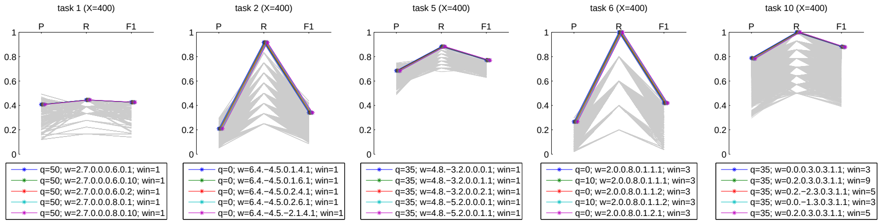


Fig. 3: Weight learning from the development set. Description of the parameters: q = quality threshold; w =[w_{coco}, w_{objy}, w_{objn}, w_{ply}, w_{pln}, w_{loc}, w_{act}, w_{ppl}]; win = size of the smoothing window. (Best viewed in color.)

	Objects		Places		MSCOCO
Task	Relevant	Avoid	Relevant	Avoid	Relevant
1	computer group meeting	-	computer group meeting *etc.*	-	laptop keyboard
2	television food glass	computer group meeting	living room television room *etc.*	conference room lecture room *etc.*	tv remote *etc.*
3	computer group meeting	office	coffee shop living room *etc.*	conference room office *etc.*	laptop keyboard
4	computer pencil notebook	office	living room hotel room *etc.*	conference room office *etc.*	laptop book *etc.*
5	food glass	drum white goods menu’	food court restaurant *etc.*	-	fork sandwich *etc.*
6	drink glass beverage	computer	bar pub etc	home	bottle wine glass
7	-	public transport	temple palace *etc.*	residential neigh. bus *etc.*	-
8	public transport	cab car seat taxi	bus interior subway station *etc.*	car interior	bus train
9	food cooking utensil white goods	-	pantry kitchen *etc.*	living room	oven refrigerator *etc.*
10	shopping shop	-	supermarket store *etc.*	shopfront shopping mall *etc.*	-

Table 1: Semantic queries for the retrieval task: Concepts to search in WordNet to find all related (and to avoid) ImageNet classes, manual selection of relevant (and to avoid) *places* classes, and objects to detect.

For each image, a relevance score is given by the presence of such key objects, places, locations and activities, and the number of people in tasks *In a meeting* and *Social drinking*. To fuse all these aspects, each one is given a different weight, as described in section 3.1. The N images with higher relevance score are drawn and used in the following steps.

Image Features Used: Image understanding is crucial for lifelog data analysis, of which, what objects are present in the images and where are the images taken is capable of linking lifelog images to certain topics/events. Here, our objective is to estimate "what" and "where" of the lifelog images, using deep convolutional neural networks (DCNN) [14].

Objects and Places: We use DCNN respectively trained on ImageNet1K [8] and Places365 [3] to identify objects and places depicted in the Lifelog images. The ImageNet1K training set contains 1000 object categories from WordNet and 1.2 million images, the Places365 train set has 365 place categories and around 1.8 million images. A separate ResNet152 [12] is pre-trained on each dataset (termed as ResNet152-ImageNet1K and ResNet152-Places365). During test, by passing a lifelog image through ResNet152-ImageNet1K (ResNet152-Places365), a 1000-D (365-D) probability vector is extracted from the last layer (after Softmax). As in [20], data augmentation is performed to generate scaled and rotated versions for each lifelog image. The maximum activation value (instead of the average) is chosen for each class. These probability vectors serve as object and place features for the retrieval stage.

Besides image-level object recognition, we also perform object detection to locate objects in lifelog images. A Faster R-CNN [22] is pre-trained on MSCOCO [17] training dataset, containing bounding box annotations for more than 200K images over 80 object categories. Most of the categories are common in lifelog images (e.g. laptop and tv). Given a lifelog image as test input to Faster R-CNN, we compute maximum probability for each category over the top 20 detections, empirically. The maximum probabilities serve as detection features for the subsequent relevance stage.

Human Detections & Counting: Detecting and counting the number of persons in an image may provide vital information which may be useful to determine the relevance of the image for a particular query. The most popular method for detecting people in an image is by using the histogram-of-gradients (HOG) approach. We tried this approach, however, due to the lack of sufficient and representative number of training samples from this database and possible involvement of huge manual efforts, good training cannot be achieved.

Several commercial entities also provide cloud-based APIs that perform the task of detecting and counting the humans in an image. Many of these entities use proprietary deep learning based approaches to perform the task of human detection. We selected the person detection API provided by *Sighthound, Inc.* [2], as it provided convenient features such as detecting and counting people, as well as providing the coordinates of the bounding box of the detected people. The performance of the Sighthound API on several computer vision tasks on benchmark datasets has been studied and a superior performance has been reported [7, 18]. The pre-trained model used by this API requires that the person in the image should occupy at least 96×40 pixels for upper/full body detection and at least 72×64 pixels for head and shoulders. In general we found the API to be more accurate on images with good lighting conditions and when the head and shoulders of the person are clearly visible.

2.3 Event Clustering

The N images with best relevance score are then described with the concatenation of all the available features. Each feature (deep learned features, locations, activities, day and time) is given a weight which can be 0, 1, the feature frequency score tf (for the deep learned features), or the inverse of its maximum (for the metadata).

The images are then clustered into events using either k-means or a hierarchical tree.

Image Features Used:

Objects and Places: A part from the object and place features described in section 2.2, we also test describing the images with low-level deep descriptors. Using the widely used VGG16 architecture [23] pre-trained on ImageNet1K data set, we extract 512 feature maps from the last pooling layer (i.e. pool5) for each augmented image, followed by nested invariance pooling over all feature maps [21]. This results in a 512-D global descriptor representing each lifelog image. Post-processing techniques (e.g. PCA whitening) can be applied to further enhance the discriminative power of the pooled descriptors.

Locations and Activities: Each location in the user's lifelog is given a unique id. Same process is done for the activity tag.

Day and Time: From the image TimeStamp, we extract the day of the month and the hour it was taken. Alternatively, we quantize the hour into morning, afternoon, evening or night.

2.4 Key-Frame Selection

To select the key-frames, all frames in each cluster $c_i = \{f_{i_k}\}$ are ranked according to distance to the cluster center (for k-means clustering) or relevance score (for hierarchical trees), so that $c_i = [f_k]_i$, and the summary is initialized empty, $S = []$. The following process is repeated iteratively until reaching the desired summary length X: The first available image in each cluster is selected to be part of the final summary $s = \{f_k \mid \forall_i, k = 0\}$, and discarded from the bag of available frames. Then, the selection is sorted according to each frame's relevance score, so that the most relevant are first in the generated summary. The sorted sequence is added at the end of the summary, $S = S^\frown s$. Note that to force that each event will be represented if selecting a summary shorter than X, the sequence to be sorted is the newly drawn s, and never S. If, at the end of the drawing process, the cadence of S is greater than X, the last elements of S are discarded.

3 Experiments

For this task, we submit two different sets of runs: automatic and interactive. In the interactive, we give the user the opportunity of removing, replacing and adding frames from the automatically generated summary. The bag of frames from where the user can replace images is the same set of relevant images as in the automatic approach. In this section, we will first present the parameters used to find the set of relevant images, and then explain in more detail the two types of submitted runs.

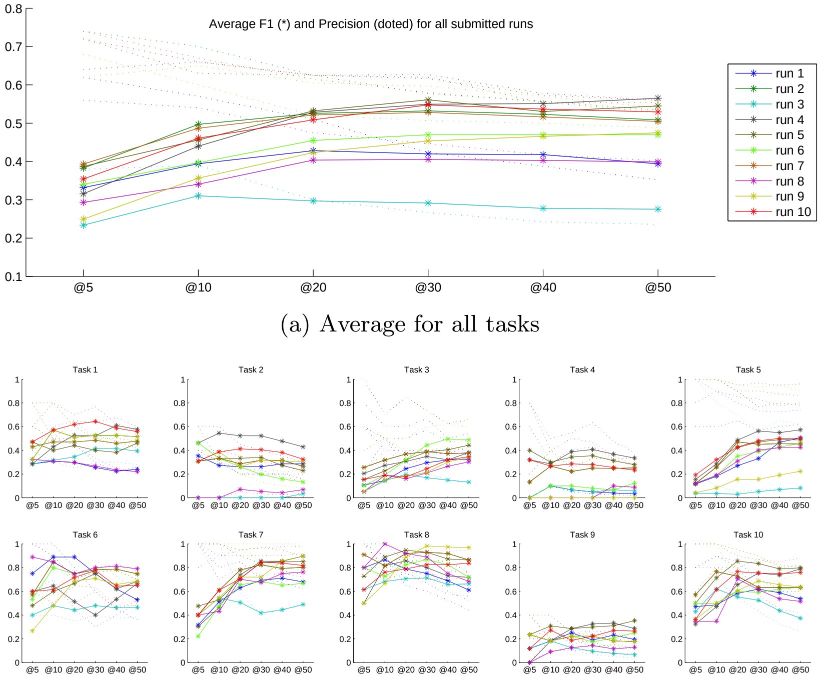
3.1 Learning the Best Parameters

We use the development set [6] to train the relevance weights and select the best features to fuse into the image descriptor. For the five test tasks that do not match any of the development ones, we have split the test set into two parts, and used the smaller of them as training data. All parameters are learned for each run and task separately.

Quality Filtering For most tasks, the quality threshold is defined by the 35 or 50 percentile, depending on whether the run uses only image features, or also metadata (note that since the quality assessment is visual, the threshold is set to zero when only using metadata). For task 6 (*Social Drinking*), where images are usually taken in dark places and are thus of bad quality, the threshold is set to be of 10 percentile.

Computing the Relevance Score The relevance score is a fusion of three modules: the visual score, obtained from the DCNN activations; the location and activity relevance from tags; and the locations and activities to remove.

The visual relevance score for each frame is the dot product between its descriptor and the reference query descriptor. For this purpose, the frame descriptor is defined as the $1365D$ vector of activations. To define the reference query descriptor, relevant object classes are found by using the WordNet [1] structure on two or three key concepts, and places are selected manually (Table 1). The reference query descriptor is initialized to zeros. Then, all classes present among the *wanted objects* are given a constant value w_{objy}, the *objects*



(a) Average for all tasks

(b) Results for each task individually

Fig. 4: Results for all the submitted runs: F1 score plotted in solid line, and precision doted. (Best viewed in color)

	Visual	Metadata	Obj. det.	Interactive	Description
run 1	✓				Parameters learned for maximum F1 score.
run 2	✓	✓			Parameters learned for maximum F1 score.
run 3		✓			Parameters learned for maximum F1 score.
run 4	✓	✓		✓	Re-clustering in each iteration; 20% extra clusters.
run 5	✓	✓		✓	No re-clustering. 100% extra clusters.
run 6	✓	✓	✓		Parameters learned for maximum F1 score.
run 7	✓	✓	✓		Parameters learned for maximum F1 score, w/ and w/o object detection. Summaries almost identical to run 2.
run 8	✓		✓		Parameters learned for maximum F1 score.
run 9	✓	✓			Parameters learned for maximum precision.
run 10	✓	✓		✓	No re-clustering. 20% extra clusters.

Table 2: Description of the different submitted runs.

to avoid are set to w_{objn}, and the same is done for *places*. Additionally, we perform object detection as described in section 2.2, and a weight w_{coco} is applied over the score of the relevant items.

Following, a value w_{loc} is added to the relevance score of all these images with a relevant location label, and the score for frames matching the relevant activity is increased w_{act}.

Finally, all the frames with location or activity label to avoid are given a relevance score of 0, and thus removed from the pool of frames.

Additionally, for tasks 1 (*In a meeting*) and 6 (*Social Drinking*), where the presence of other people is a task-relevance indicator, we count people as described in section 2.2, and increase the relevance score of those images with enough people by w_{ppl}. We observe that, given that relevant images in task 1 have many occlusions, and that images in task 6 have poor lightning conditions, the performance of the people detector for such tasks is not good enough.

The final relevance value is smoothed using a triangular window of size win, which ranges between 1 (0 extra frames) and 11 (5 frames to each side). The optimal values of w_{coco}, w_{objy}, w_{objn}, w_{ply}, w_{pln}, w_{loc}, w_{act}, w_{ppl} and win are found heuristically by analyzing the retrieval performance in the training data, as shown in Fig. 3. The objective is best recall at $X = 400$, to have the greatest number of events brought forward for the next step in the summarization process.

3.2 Automatic Runs

We submitted seven different automatic runs. Such runs are compiled in Table 2, and are defined by the range of features used: only image (with and without object detection), only metadata and mixed.

Clustering the Lifelog Images into Events The weights for each feature in the image descriptor for each task are defined by the best combination in the test set. The images are then clustered into M events using k-means, or hierarchical trees when only using metadata. The number of events M is set to be equal or smaller than the summary length budget. When using k-means, the frames with relevance below the 50 percentile for each cluster are discarded. The selection and ranking of keyframes from the clusters is described in section 2.4.

3.3 Interactive Runs

We submitted three different interactive runs, as compiled in Table 2. The interaction time per task is of $3'$ on average, ranging between $1'30''$ and $4'40''$.

Clustering the Lifelog Images into Events For the interactive runs, k-means is chosen for clustering the relevant images into M events, where M is greater than the summary length budget. In each iteration, the user can select which images to preserve for the final summary, which frames to remove from the bag of relevant images, and whether all other images in the cluster should

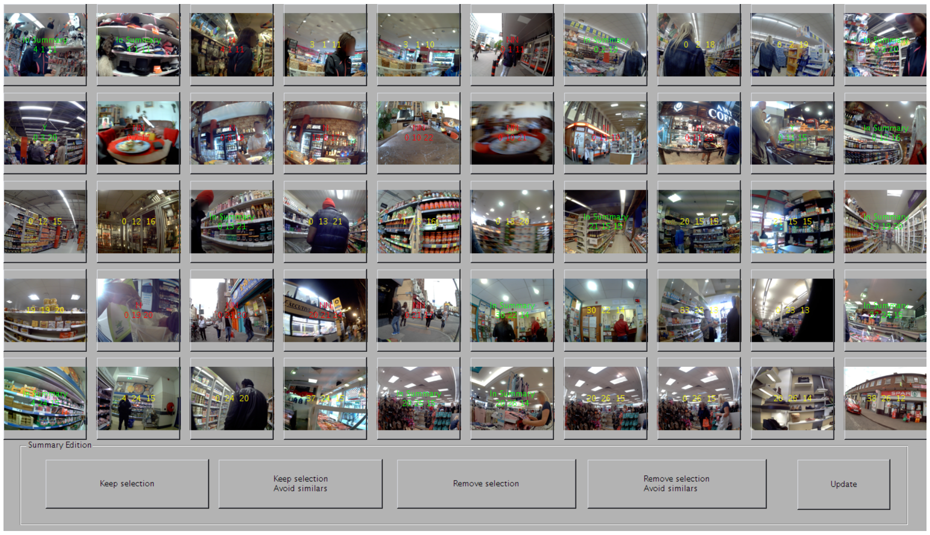


Fig. 5: GUI for Interactive Summarization. Frames are shown with information of their location and timestamp, and those already selected for the final summary are marked "In Summary". The user can select the frames to be included (green Y), to be removed (red N), or to remove all frames in the same cluster ($*N$).

be removed (Fig. 5). Two methodologies for updating the summary are proposed: First, re-clustering the remaining frames. Second, using the same initial clustering for all iterations.

In the first approach, the frames are clustered into a 20% more clusters than additional keyframes needed (note that this number changes at each iteration). For the second approach, two configurations are tested: clustering into either a 20% or 100% more frames than the length of the final summary. Once clustered, the frames closest to each cluster center are chosen as candidates to be added to the summary. The most relevant ones (as many as needed to fill the summary budget length) are then included in the proposed summary.

3.4 Discussion

By looking at the results in Fig. 4, we can observe that the use of mixed features generally improves the performance. Using only metadata (*run* 3) is, in average, worse than using only visual features (*runs* 1 and 8). We noted that the location metadata was not very precise in terms of starting and finishing times. It is interesting to see how interactive approaches (*runs* 4, 5 and 10) do not necessarily perform better than automatic ones for low values of X. This may be due to the subjectivity of such kind of retrieval task. One may think that, being the images handpicked (and sorted in drawing order), the precision score should be much higher than the automatic runs, and close to 1 for low values of X (being those the first selected keyframes). However, this only happens for some tasks.

Analyzing each task independently, we observe that visual features are not very precise for tasks 1 (*In a meeting*) and 4 (*Working at home*). Surprisingly, using only visual features yield the best results for task 6 (*Social drinking*) for low values of X, specially if also using object detection, which performs best for larger X. An outstanding retrieval performance (precision greater than 80% for all X) is achieved with a mix of visual and metadata features for tasks 7 (*Sightseeing*), 8 (*Transporting*) and 10 (*Shopping*), possibly due to the quality of the metadata (although the performance of using only visual features is competitive with the mixed approaches). The best results for tasks 2 (*Watching TV*), 4 (*Working at home*), 5 (*Eating*), 9 (*Preparing meals*) and 10 (*Shopping*) are obtained with the interactive approaches, since recall can be improved when manually selecting images from different clusters.

4 Conclusions

In this paper, we have presented a generic framework for the summarization of lifelogs given a target task. It can be used both in an automatic or interactive way, with the user providing feedback on the retrieved frames. The proposed approaches require that the user selects the relevant locations for each task. In order to ease this forced manual input, the metadata obtained with lifelog apps could contain additional info of, *e.g.* , the nature of each location.

We have observed that different tasks require different summarization methodologies (*e.g.* different weights), which may not be completely consistent when changing the lifelog input. Trained on the development set, we have obtained a 0.497 best averaged F1 score @X = 10 (the averaged F1 for the best run for each task is 0.563), meaning there is still a lot of room for improvement. We encourage other researchers to participate in future such competitions.

References

1. Wordnet. Princeton University (2010), `http://wordnet.princeton.edu`
2. Sighthound detection api. Sighthound, Inc. (2017), `https://www.sighthound.com/docs/cloud/detection/`
3. B. Zhou, A. Khosla, A.L.A.T., Oliva, A.: Places: An image database for deep scene understanding. In: arXiv:1610.02055 (2016)
4. Bolanos, M., Dimiccoli, M., Radeva, P.: Toward storytelling from visual lifelogging: An overview. IEEE Trans. HumanMach. Syst. 47(1), 77–90 (2017)
5. Bolanos, M., Mestre, R., Talavera, E., Giró-i Nieto, X., Radeva, P.: Visual summary of egocentric photostreams by representative keyframes. In: IEEE ICME Workshops. pp. 1–6 (2015)
6. Dang-Nguyen, D.T., Piras, L., Riegler, M., Boato, G., Zhou, L., Gurrin, C.: Overview of ImageCLEFlifelog 2017: Lifelog Retrieval and Summarization. In: CLEF 2017 Labs Working Notes. CEUR Workshop Proceedings, CEUR-WS.org, Dublin, Ireland (September 11-14 2017)
7. Dehghan, A., Ortiz, E.G., Shu, G., Masood, S.Z.: Dager: Deep age, gender and emotion recognition using convolutional neural network. arXiv preprint arXiv:1702.04280 (2017)

8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
9. Doherty, A.R., Conaire, C., Blighe, M., Smeaton, A.F., O'Connor, N.E.: Combining image descriptors to effectively retrieve events from visual lifelogs. In: Proceedings of the 1st ACM MIR. pp. 10–17. ACM (2008)
10. Elsweiler, D., Ruthven, I., Jones, C.: Towards memory supporting personal information management tools. J. Assoc. Inf. Sci. Technol. 58(7), 924–946 (2007)
11. Harvey, M., Langheinrich, M., Ward, G.: Remembering through lifelogging: A survey of human memory augmentation. Pervasive Mob Comput. 27, 14–26 (2016)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
13. Ionescu, B., Müller, H., Villegas, M., Arenas, H., Boato, G., Dang-Nguyen, D.T., Dicente Cid, Y., Eickhoff, C., Garcia Seco de Herrera, A., Gurrin, C., Islam, B., Kovalev, V., Liauchuk, V., Mothe, J., Piras, L., Riegler, M., Schwall, I.: Overview of ImageCLEF 2017: Information extraction from images. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction 8th International Conference of the CLEF Association, CLEF 2017. Lecture Notes in Computer Science, vol. 10456. Springer, Dublin, Ireland (September 11-14 2017)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
15. Lee, M.L., Dey, A.K.: Providing good memory cues for people with episodic memory impairment. In: ACM SIGACCESS ASSETS. pp. 131–138. ACM (2007)
16. Lee, Y.J., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: CVPR vol. 2, p. 6 (2012)
17. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV (2014)
18. Masood, S.Z., Shu, G., Dehghan, A., Ortiz, E.G.: License plate detection and recognition using deeply learned convolutional neural networks. arXiv preprint arXiv:1703.07330
19. Garcia del Molino, A., Tan, C., Lim, J.H., Tan, A.H.: Summarization of egocentric videos: A comprehensive survey. IEEE Trans. HumanMach. Syst. 47(1), 65–76 (2017)
20. Garcia del Molino, A., Xu, Q., Lim, J.H.: Describing lifelogs with convolutional neural networks: A comparative study. In: LTA Workshop. pp. 39–44. ACM (2016)
21. Morère, O., Lin, J., Veillard, A., Duan, L.y., Chandrasekhar, V., Poggio, T.: Nested invariance pooling and rbm hashing for image instance retrieval. In: ACM ICMR (2017)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS (2015)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
24. Xiong, B., Grauman, K.: Detecting snap points in egocentric video with a web photo prior. In: ECCV pp. 282–298 (2014)