Automatic Annotation of Lifelog Activities

Srijata Roy
17210122
srijata.roy3@mail.dcu.ie
MCM (DA) CA685
Supervisor(s): Dr. Cathal Gurrin & Dr. Duc-Tien Dang-Nguyen

A report submitted to Dublin City University, School of Computing for module CA685 Data Analytics Practicum, 2017/2018. I understand that the University regards breaches of academic integrity and plagiarism as grave and serious. I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references. I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work. By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name(s): Srijata Roy Date:10/08/2018

Automatic Annotation of Lifelog Activities

Srijata Roy
srijata.roy3@mail.dcu.ie
17210122

***Abstract*—This paper deals with the effective ways to index the lifelog activities. The multi-faceted system of lifelogging incorporates and proposes to document day-to-day mundane activities. The availability of various gadgets to record one's daily activities has increased the demand for the study of these unstructured egocentric data and auto typify them into specific frames of life. The main challenge is to develop algorithms which will automatically categorize everyday activities into labeled blocks of life. Our aim is to establish a comparative analysis of the existing algorithms.**

***Keywords*— lifelogging; indexing; egocentric data; daily activities.**

I. Introduction

Lifelogging, a self-explanatory term, comprising of images and videos describing diurnal activities captured over a prolonged duration of time. The purpose of such research is to establish a trend in the recorded activities which proves beneficial to health analysis for varied medical conditions. Next in line to blogging and vlogging, lifelogging holds a great number of solutions to the upcoming years of scientific research.

A lifelogger's moto is to create digital copies of memories owing to the temporary nature of human retention capabilities. The trend of lifelogging has gained popularity with the growing availability of wearable sensors which has paved the path for the potential research in the field. The expeditious advances made to alter the size of the wearable cameras to achieve full-time usage has exponentially increased the datasets of images and/or videos recorded. It is difficult to reach any conclusion with this amount of vast and varied data without pre-processing and categorically segregating it. A number of algorithms are designed to annotate and index these mass scale data to fit into specific life events. The analytical challenges posed by this activity is the main motivation behind choosing this topic.

II. Literature Review

A. Lifelog Annotation

The aim of lifelogging is to create a standby memory system which can be referred to when required. Gurrin et al. [1] state the possibility of segmenting the raw, unprocessed lifelog data into meaningful semantics which he defines as: "a temporally related sequence of lifelog data over a period of time with a defined beginning and end". The annotations, either manually created or generated through machine learning techniques, must be assessed for its efficacy. The quality of annotations has a direct impact on the information retrieval process. Gurrin et al. [1] propose the five aspects of human memory as recollecting, reminiscing, retrieving, reflecting and remembering, which forms a foundation for the search and retrieval process in lifelogging. Vuurpijl, et al. [2] stresses the importance of the domain knowledge when it comes to fulfilment of information of a user during the image extraction process.

B. Different Types of Annotations

The two broad categories of retrieval processes in life logs are text-based image retrieval (TBIR) and content-based image retrieval (CBIR). The former makes use of text tags whereas the latter extracts visual attributes such as colour, shape or texture through a series of queries. Hartvedt [3] states the benefit of TBIR over CBIR as the use of image retrieval based on high-level textual semantic concepts. Scells [4] argues about the machine learning capabilities due to the absence of a reference frame.

The retrieval process formulates the base for annotations. There are two common approaches for image annotation: tagging an image with a set of keywords and browsing a set of images against the applicability of the predefined keyword. Zarzour et al. [5] mention the drawback of the existing frameworks for video annotations as the incapability to simultaneously annotate videos during collaborative work.

Scells [4] presents the theory of outperformance of high quality textual descriptions in comparison to keyword or tag based annotation models and furthers his statement by stating the limitations of the latter as:

1. The relevance of a document is not automatically determined by the presence of a keyword in it
2. The exact word may not be present in the relevant document
3. The recall rate is lowered with the usage of synonyms of the query keywords
4. The precision rate is lowered with the usage of homonyms of the query keywords
5. Semantic relations such as hyponymy, meronymy, antonymy are not exploited

C. Annotation Evaluation

Text is the main source of providing semantic elucidation either in the form of short tags or long descriptions [6]. The focus on the accuracy and high standard of the annotation determines the quality of the search result. This data can either be generated by machine learning algorithms [7], [8] or manually created. Scells [4] points out the possibility of acquiring undesirable results when applying a trained domain model to a completely unknown field, because the algorithms learn from test data. He further goes on to stress on the importance of the evaluation of annotations, because the detailed image descriptions are more persuasive to humans and easier for them to query them.

Out of all the models available in the market to evaluate the quality of annotations, the most extensively used are BLEU, ROUGE and METEOR. BLEU is a Bilingual Evaluation Understudy and is based on the precision model. ROUGE is a Recall-Oriented Understudy for Gisting Evaluation and is based on the recall model. METEOR is a Metric for Evaluation of Translation with Explicit Ordering used for overall evaluation of annotation standards. Scells [4] states even though the above three forms are diametrically linked with the assessment of automatic summarisation and natural language processing, they still reference annotations from a framework model. Vedantam et al. [9] state the efficiency of human consensus through their method, CIDEr (Concensus-based Image Description Evaluation), a model which makes use of sentence similarities, notions of grammaticality, saliency, importance (precision) and accuracy (recall).

D. Lifelog Datasets Search

Gurrin et al. [1] reports the poor performance of the lifelog information retrieval system due to the absence of reference models until recent times, when there have been considerable development in the field such as TREC, CLEF and NTCIR. These evaluations are based on test collections which comprise of static database of information objects, a static set of topics representing information needs, and a static set of judgements of the relevance of the information objects to each topic [1]. The NTCIR model makes use of pre-trained image tagging algorithm which enables it to generate a set of tagged lifelog images from a lifelogging camera worn by researchers over a short period of time [10]. The lack of substantial growth in the establishment of retrieval methodologies in lifelogging hasn't deterred researchers from discussing about its aims and objectives. Bristow et al. [11] and Doherty et al. [12] endorse the advantages of detecting and interpreting the implicit semantics and context of lifelogging data from heterogeneous sources in explaining the Who?, What?, Where? and When? questions occurring in every day events. Ali et al. [6] point out the commonality of these questions among image searchers and the incapability of normal indexing methods solving the same. While there has been some research into tagging and annotating images, there has not been as much work in developing a model for searching these images within the context of lifelogging [1].

As stated earlier, image-based retrieval methods can be classified into two categories: text-based image retrieval (TBIR) and content-based image retrieval (CBIR). A CBIR system utilises image features such as grid colour movements, edge direction histogram, Gabor textual features, and Local binary pattern histograms. As described in work by Wu et al. [13]. These features (colour, texture, shape, SIFT key points) become a query to the search engine which match visually similar images. CBIR systems, although extensively studied for over a decade, are still limited in comparison to TBIR systems. The three reasons provided by Zhu et al. [14] for this are:

1. The semantic gap that exists between low-level visual features and high-level semantic concepts
2. The low efficiency due to high dimensionality of feature vectors
3. The query form is unnatural for image searching (appropriate example images may be absent)

The efficiency of TBIR can be explained when one considers that it can be formulated as a document retrieval problem and can be implemented using the inverted index technique. The downside to TBIR is that is highly expensive: experimental evidence by Wu et al. [15] shows that the performance of TBIR is highly dependent on the availability and quality of manual annotations. If this process can be automated and images can be automatically captioned, it would solve a fundamental issue that exists with TBIR systems.

III. Experiment

A. Task Overview

The classification task at hand is a subtask of Lifelog Moment Retrieval (LMRT) ImageCLEFlifelog 2018 competition [16]. The main task is described as the retrieval of specific moments from a lifelog by making use of the available multi-modal lifelogger information. To enhance the system of information retrieval, in the form of images, it is important to classify the different activities of the lifelogger.

Let us rewind and revise a few terms before we delve deeper into the system. The hierarchical model of retrievable lifelog data units, as defined by [17], are:

- **Item:** the smallest retrievable unit, the atomic unit of data, such as an image, temperature reading, location, etc. It is the unit of retrieval that was favoured in MyLifeBits.
- **Moment:** a fixed length of temporal unit, which has been considered as a minute. Hence, 1440 moments can be captured in a day and each moment is represented as a collection of all the (retrievable) items that take place within that minute. Moments were used as the retrieval unit in the NTCIR Lifelog comparative benchmarking exercises. It is described by the multi-modal atomic units.
- **Activity:** is defined as an un-interrupted sequential state of the individual in terms of their person or environment or stimuli. The activity is the indexing-time unit of retrieval that we define in this work and propose as the most appropriate indexing time unit. It represents a combination of sequential items whose size is dependent on the activities of the individual.
- **Event:** is a combination of moments or activities or experiences developed (up until now) at indexing time. It is the longest retrievable unit, whereby 2-4 units in any given hour (based on past research). The event has been the first unit of retrieval for lifelog data and was employed manually in the initial Sensecam image viewer tool as well as the early Doctoral Symposium Session work of Doherty et al. in the development of early lifelog search engines.

Nine activities are short-listed from NTCIR-13 Lifelog-2 after a careful study of the series of images in the lifelog. To enumerate the various problems affecting the efficiency of the retrieval system, these activities have been selected based on its occurrences, ranging from very high frequencies to very low ones. The classification of activities is a step towards the

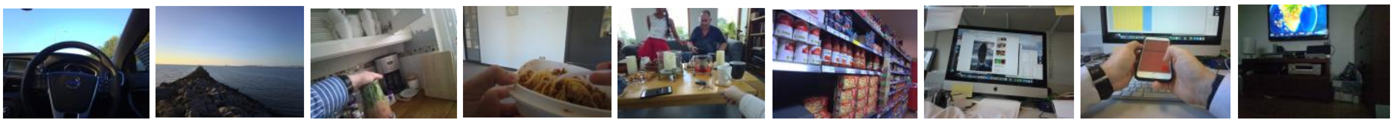

Figure 1: Images of nine lifestyle activities

refinement of the search systems implemented in Lifelogging Tools and Applications. Hence the most prominent activity is taken into relevance in each frame of data.

Commuting	Travelling	Preparing meals
Eating/drinking	Socialising / casual conversation	Shopping
Using desktop computer / laptop computer	Using mobile device / tablet	Watching television

Figure 2: The nine lifestyle activity labels

The activities chosen in order of occurrences from most to least are: "Using desktop computer / laptop computer", "Socialising / casual conversation", "Travelling", "Using mobile device / tablet", "Commuting", "Watching television", "Eating/drinking", "Shopping", "Preparing meals".

B. *Dataset*

As stated in my research plan earlier, I exploited the lifelog data from NTCIR and ImageCLEFlifelog, the two-benchmarking campaigns on lifelogging retrieval. For the subtask a subset of ImageCLEF2018lifeLog [16] image dataset is used. The images, captured by a wearable camera, define the lifeloggers daily activities right from dawn till after dusk. The dataset, better known as a concept file from a lifelogger perspective, is a list of all the images with 86-categories taxonomy. Microsoft Computer Vision API image categorisation is used to pre-process the images to generate this concept file which comprises of text-based tags and its corresponding confidence scores.

Number of Lifelogger(s)	1
Number of Days	6 (15th – 20th August 2016)
Size of the Collection	1.71 GB
Number of Images	7063
Number of NTCIR-13 Lifelog-2 Activities	10

Table 1: Statistics of subset of NTCIR-13 Lifelog Data

For the purpose of this experiment, the ground truth is defined by manual annotations. A single image may define multiple activities. But for the simplicity of the experiment the most prominent activity is given preference. This is a deviation from the prior definition of activity as stated in this paper. In order to focus on the specific activities, other activities of less priority go unnoticed in the captured data, which henceforth affects the classification of that particular activity. The gathered feature values are continuous while the targets are categorical.

A simple visualisation of the dataset shows the high levels of imbalance in the data. This factor is very crucial in classifying the activities as it might lead the data to overfit or underfit the model. Also, the dataset is quite sparse in nature with a maximum of 15 actual attribute values.

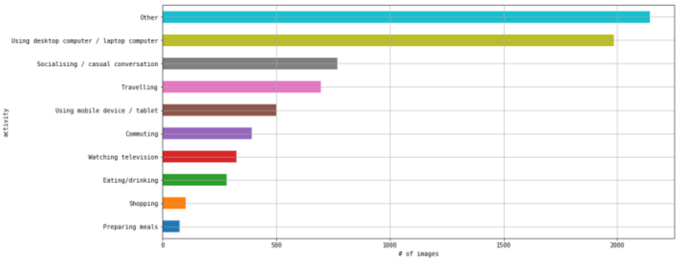

Figure 3: Number of images per activity

The “Other” content surpasses all other activity classes. “Using desktop computer / laptop computer” has the highest image count while “Preparing meals” ranks lowest. During manual annotation, it is noted, five out of six days are weekdays; which henceforth increases the chances of spotting the activity encompassing computers and laptops. “Socialising / casual conversation”, ranking second amongst all activities, prove that the lifelogger has an active social life despite only one weekend being recorded in this experiment.

C. *Data Preprocessing*

Microsoft Vision API, baselined in [18], is used to convert the images to computer compatible digits, where each component visible in the image is tagged and assigned a confidence score. The subset of concepts chosen for the experiment is transformed to form vectors for each image with the text-based tags as the attributes for each image and the confidence score as the values. The missing attributes for any image is assigned a confidence score of zero.

The data transformation results in the generation of feature attributes for each image, i.e. an image vector now comprises of 548 confidence scores defining it. Hence, the resultant high dimensional feature dataset requires further pre-processing in terms of feature reduction and feature selection. Both the processes are independent in nature, but their amalgamation can achieve better results.

1. *Feature Reduction*

The most common and widely used method of feature reduction is Principal Component Analysis (PCA). It is a dimensionality reduction technique where new features are created by combining principal components to retain the maximum variance. The data defined in the high dimensional space is reduced to a lower dimensional space such that the variance of the data is maintained. It makes use of linear algebra to transform the data into compressed form.

2. *Feature Selection*

On the other hand, feature selection method includes and exclude attributes present in the data without changing them. The two feature selection methods incorporated in this project are embedded and filter methods.

Recursive Feature Elimination (RFE), an embedded based method, recursively eliminates the lower weighted elements assigned to the base classifier. It first chooses a subset of features and prunes the same based on the weights assigned to the base classifier.

Feature Importance (FI), a filter-based method, is the other method used for feature selection. It is used in conjunction with the ExtraTreeClassifier ensemble approach to filter out the irrelevant features. It is a randomised process, hence, there is a new set of features that get importance using this algorithm.

The embedded feature selection method is used with three different base estimators: logistic regression, linear support vector classifier and support vector classifier with linear kernel. Both the feature selection methods are directly applied to the feature dataset and as well as the PCA reduced features. This combination of PCA with RFE or PCA with FI is conducted to compare results at the end of the experiment.

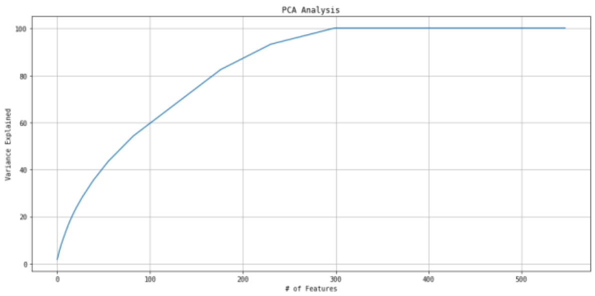

Figure 4: PCA Analysis

From the graph above, it is evident that to maintain maximum variance of all the features collectively, minimum of 300 features are to be used out of 548 totals. A further selection of 150 features is applied using RFE and FI. Also, 300 features are selected from the original feature dataset using both the selection techniques.

D. *Supervised Classification*

The data is split into train (75%) and test (25%) sets using a StratifiedKFold approach to preserve the percentage of samples for each class. Due to the high levels of data imbalance discussed previously, a step-by-step approach is adapted for supervised learning.

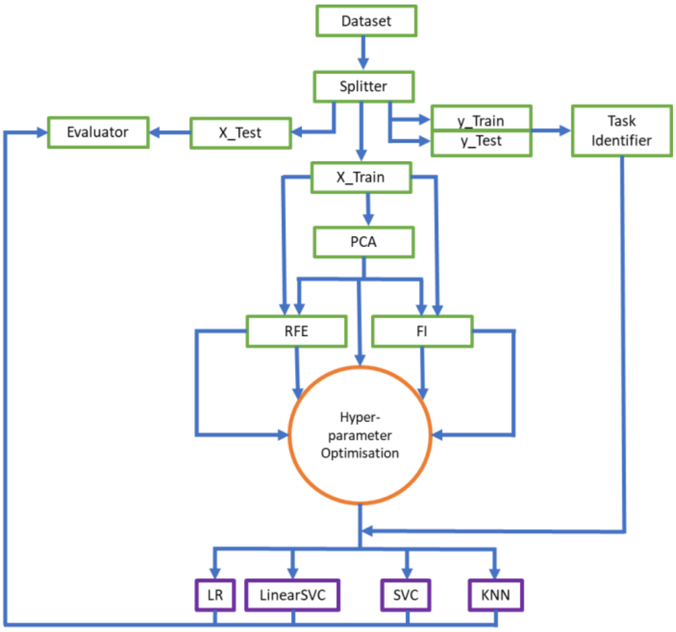

Figure 5: Modelling Schema

At first, a binary classification is performed on only three activities from the entire nine-activity list. These activities are selected based on its frequency: the most occurring “Using desktop computer / laptop computer”, the least occurring “Preparing meals” and one from the mid-range “Commuting”. This choice helps us analyse the problem that will be faced in the multi-class classification. The hyperparameter optimisation for the models is conducted using RandomizedSearchCV in contrast to the usual GridSearchCV. It implements a randomised search over parameters, where each setting is sampled from a distribution over possible parameter values [19]. This has two main benefits over an exhaustive grid search:

- A budget can be chosen independent of the number of parameters and possible values.
- Adding parameters that do not influence the performance does not decrease efficiency.

Specifying how parameters should be sampled is done using a dictionary. Additionally, a computation budget, being the number of sampled candidates or sampling iterations, is specified using the n_iter parameter. For each parameter, either a distribution over possible values or a list of discrete choices (which will be sampled uniformly) are be specified. For continuous parameters it is important to specify a continuous distribution to take full advantage of the randomisation. This way, increasing n_iter will always lead to a finer search. The search makes use of 5-fold cross-validation as well.

1. *Logistic Regression*

The model best known for binary classifications of categorical targets given continuous features is applied first. The hyperparameter optimisation is set for inverse regularisation strength parameter C and the regularisation penalty. The former is defined as a list, whereas the latter is defined as a continuous distribution. The ‘max_iter’ hyperparameter is set to 10000 for the estimator to converge.

$$\sigma(t) = \frac{1}{1+e^{-t}}$$

The above equation signifies the logistic sigmoid function which accepts all continuous real input and outputs a value between 0 and 1.

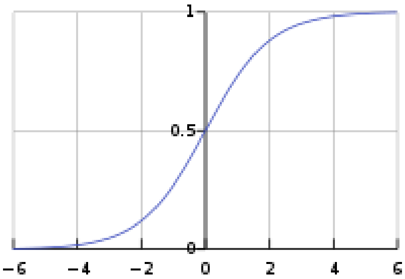

Figure 5 : The standard logistic function [20]

The moto is to model the probability of a random variable being 0 or 1 given experimental data. Hence this model is applied to the binary classification of the selected activities.

The Multinomial Logistic Regression (MLR) is applied to the final modelling where the images are classified into nine lifestyle activities. The hyperparameters ‘saga’ and ‘l1’ are selected for faster convergence on the huge dataset.

2. *Support Vector Machine*

The parametric model makes use of hyperplanes to efficiently classify the target classes. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p-1)-dimensional hyperplane. SVM is an optimal classifier in the sense that, given training data, it learns a classification hyperplane in the feature space which has the maximal distance (or margin) to all the training examples (except a small number of examples as outliers).

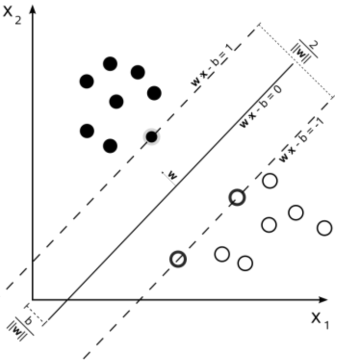

Figure 6: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors. [21]

There are several ways to implement Support Vector Classification. The two implementations used here are:

- ● LinearSVC
- ● SVC with rbf (radial basis function) kernel

The former, self-explanatory, converts the problem into a linear function and is a liblinear implementation. Whereas the latter converts the problem into a non-linear gaussian function using libSVM, a library which implements the Sequential Minimal Optimization (SMO) algorithm, for solving the quadratic programming (QP) problem that arises during the training of support vector machines. The RBF kernel can classify a non-linear function by creating a non-linear boundary using:

$$K_{RBF}(x, x') = e^{[-\gamma||x-x'||^2]}$$

However, libSVM is not ideal for large datasets as kernelized SVMs require the computation of a distance function between each point in the dataset, which is the dominating cost of O($n_{features}$×$n^2_{observations}$). The QP solver used in libSVM is targeted to work for both linear and non-linear kernel. The training time complexity is somewhere around O(n^2) to O(n^3). The solver in libLinear is targeted to primarily work with linear kernel and on top of that it offers several variations (regularization, loss etc.). The training time complexity is hence reduced to O(n), even though the same SMO algorithm is implemented. The intent was to try out the simplest linear model first before jumping into the complex non-linear model. Hence, the linear SVC (libLinear) is implemented before going forward with the gaussian SVC (libSVM) function.

Between SVC and LinearSVC, one important decision criterion is that LinearSVC tends to converge the large number of samples faster. This is due to the fact the linear kernel is a special case, which is optimized for in libLinear, but not in libSVM. The differences in results come from several aspects: SVC and LinearSVC are supposed to optimize the same problem, but in fact all libLinear estimators penalize the intercept, whereas libSVM ones do not. This leads to a different mathematical optimization problem and thus different results. There are other subtle differences such as scaling and default loss function. In multiclass classification, libLinear does one-vs-rest by default whereas libSVM does one-vs-one. For the current classification problem, due to the volume of the dataset, LinearSVC implementing libLinear functionalities has been chosen.

3. *K Nearest Neighbours*

The non-parametric model makes use of the nearest neighbouring data points to classify the object. It depends on the density of target clusters to classify the object into a class. It makes use of different distance measures for the classification. The most basic distance measure is Minkowski distance:

$$d_{(x,y)} = (\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p}$$

Test Data = 1454		Using desktop computer / laptop computer (=396)					Commuting (=78)					Preparing meals (=15)				
		PCA	RFE	PCA_RFE	FI	PCA_FI	PCA	RFE	PCA_RFE	FI	PCA_FI	PCA	RFE	PCA_RFE	FI	PCA_FI
LR	Precision	1.00	1.00	1.00	1.00	1.00	0.02	0.01	0.05	0.01	0.06	0.67	0.00	0.40	0.00	1.00
	Recall	0.81	0.80	0.80	0.79	0.79	0.04	0.03	0.04	0.03	0.12	0.13	0.00	0.13	0.00	0.13
	F1	0.89	0.89	0.89	0.88	0.88	0.03	0.01	0.04	0.01	0.08	0.22	0.00	0.20	0.00	0.24
	AUC	0.96	0.94	0.96	0.95	0.96	0.74	0.76	0.77	0.76	0.80	0.72	0.94	0.67	0.95	0.82
LnSVC	Precision	0.97	0.99	0.99	1.00	0.98	0.06	0.02	0.05	0.01	0.03	0.08	0.50	0.00	0.50	1.00
	Recall	0.82	0.76	0.80	0.79	0.80	0.06	0.06	0.12	0.03	0.06	0.07	0.07	0.00	0.13	0.07
	F1	0.89	0.86	0.88	0.88	0.88	0.06	0.03	0.07	0.01	0.04	0.07	0.12	0.00	0.21	0.12
	AUC	0.95	0.92	0.96	0.94	0.95	0.76	0.46	0.50	0.71	0.71	0.67	0.69	0.75	0.70	0.82
SVC	Precision	0.99	0.99	1.00	1.00	1.00	0.03	0.00	0.03	0.01	0.03	0.00	0.00	0.00	1.00	0.00
	Recall	0.83	0.80	0.82	0.78	0.80	0.05	0.00	0.05	0.03	0.06	0.00	0.00	0.00	0.07	0.00
	F1	0.90	0.88	0.90	0.87	0.89	0.04	0.00	0.04	0.01	0.04	0.00	0.00	0.00	0.12	0.00
	AUC	0.94	0.95	0.96	0.95	0.96	0.67	0.42	0.69	0.66	0.73	0.65	0.55	0.65	0.54	0.64
KNN	Precision	1.00	0.97	0.99	0.99	1.00	0.02	0.02	0.02	0.01	0.02	0.00	0.00	0.00	0.00	0.00
	Recall	0.84	0.79	0.84	0.81	0.85	0.05	0.04	0.04	0.03	0.05	0.00	0.00	0.00	0.00	0.00
	F1	0.91	0.87	0.91	0.89	0.92	0.02	0.02	0.02	0.01	0.03	0.00	0.00	0.00	0.00	0.00
	AUC	0.97	0.95	0.95	0.94	0.95	0.70	0.71	0.74	0.73	0.74	0.94	0.89	0.87	0.90	0.93

Table 2: Evaluation metrics for Activities

The distance metric used for the classification of the activities is a combination derived from the RandomizedSearchCV where p = 2 or 3. The element of similarity, derived from the distance metric, between the feature values is used to group the target classes together.

Since it is heavily dependent on the distance measures between the feature points, it is not highly recommended for the sparse and high dimensional dataset. However, this classification throws a comparative light on the approaches that can be adopted in the future for classifying the activities.

E. *Results and Evaluation*

The experiment was first carried out on binary targets:

1. *Activity: "Using desktop computer / laptop computer"*

This activity in total has 1984 images captured, 1588 used as train and 396 as test data. There is sufficient number of images to train the classifier to classify the activities. From table 2, we see, for PCA reduced components KNN classifier (k = 77) works best with AUC of 0.97. The feature selections applied on the reduced components also give better results for the other classifiers used.

2. *Activity: "Commuting"*

This activity in total has 392 images captured, 314 used as train and 78 as test data. This activity is segregated from Travelling based on the destination; i.e. if the destination is either home or office it will be classified as Commuting, else it will be classified as Travelling. Travelling in total has 697 images, which is little short of double the number of images for this activity. The best AUC scores are noted for Logistic Regression, with the features selected using FI from the PCA reduced list ranking the highest.

3. *Activity: "Preparing meals"*

This activity in total has 76 images captured, 61 used as train and 15 as test data. It has the least support amongst all the activities. On an average the AUC scores best for KNN for low support of data.

Overall, we see the AUC scores for the first activity is the highest. This is due to the maximum number of samples present in the dataset. The third activity also has good AUC scores for some classifiers, but this is mainly because an average of the binary classification is considered, i.e. the "Other" gets classified due to the support correctly and not the activity itself.

IV. Conclusion

Privacy is one of the primary concerns when it comes to lifelogging project as it directly deals with records of daily human activities. The privacy aspect of the datasets to be used for this research has already been established. The data ownership and access has been granted by the individuals to the owner of the datasets for the scope of research work. The ethical aspect of the activities recorded in the lifelogs is also a key issue to be considered.

The topic of lifelogging is vast and poses the potential for research in many directions. Existing research on manual annotation lays down the foundation for this research and the main aim is to identify best practices used in annotating and indexing of lifelogs. The medium for recording activities have improved over the years where extra care has been taken to design the wearable sensors in a robust and unobtrusive fashion. This has enabled the dataset of life activities and events to increase which broadens the scope of work. Manual annotations are still considered as ground truth for this field of research, which poses a huge cost on evaluation. The other factors to consider for this classification-based task is the unbalanced dataset and the sparsity of it.

References

[1] C. Gurrin, A. F. Smeaton, and A. R. Doherty, ‘LifeLogging: Personal Big Data’, *Found. Trends® Inf. Retr.*, vol. 8, no. 1, pp. 1–125, Jun. 2014.

[2] L. Vuurpijl, L. Schomaker, and E. van den Broek, ‘Vind(x): using the user through cooperative annotation’, in *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, 2002, pp. 221–226.

[3] C. Hartvedt, ‘Using Context to Understand User Intentions in Image Retrieval’, in *2010 Second International Conferences on Advances in Multimedia*, 2010, pp. 130–133.

[4] H. Scells, ‘Investigating Methods Of Annotating Lifelogs For Use In Search’, Queensland University of Technology.

[5] H. Zarzour and M. Sellami, ‘Using commutative replicated data type for collaborative video annotation’, in *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, 2014, pp. 523–529.

[6] D. A. Ali and S. A. Noah, ‘Semantically indexed and searched of digital images using lexical ontologies and named entity recognition’, in *2010 International Symposium on Information Technology*, 2010, vol. 3, pp. 1308–1314.

[7] A. Karpathy and L. Fei-Fei, ‘Deep Visual-Semantic Alignments for Generating Image Descriptions - IEEE Journals & Magazine’. [Online]. Available: https://ieeexplore.ieee.org/document/7534740/.

[8] I. Sutskever, J. Martens, and G. Hinton, ‘Generating Text with Recurrent Neural Networks’, in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, USA, 2011, pp. 1017–1024.

[9] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, ‘CIDEr: Consensus-Based Image Description Evaluation’, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4566–4575.

[10] 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*, Pisa, Italy, 2016.

[11] H. W. Bristow, C. Baber, J. Cross, J. F. Knight, and S. I. Woolley, ‘Defining and evaluating context for wearable computing’, *Int. J. Hum.-Comput. Stud.*, vol. 60, no. 5, pp. 798–819, May 2004.

[12] A. R. Doherty and A. F. Smeaton, ‘Automatically augmenting lifelog events using pervasively generated content from millions of people’, *Sensors*, vol. 10, no. 3, pp. 1423–1446, Feb. 2010.

[13] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu, ‘Distance Metric Learning from Uncertain Side Information with Application to Automated Photo Tagging’, in *Proceedings of the 17th ACM International Conference on Multimedia*, New York, NY, USA, 2009, pp. 135–144.

[14] G. Zhu, S. Yan, and Y. Ma, ‘Image Tag Refinement Towards Low-rank, Content-tag Prior and Error Sparsity’, in *Proceedings of the 18th ACM International Conference on Multimedia*, New York, NY, USA, 2010, pp. 461–470.

[15] L. Wu, R. Jin, and A. K. Jain, ‘Tag Completion for Image Retrieval’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 716–727, Mar. 2013.

[16] ‘ImageCLEFlifelog | ImageCLEF / LifeCLEF - Multimedia Retrieval in CLEF’. [Online]. Available: https://www.imageclef.org/2018/lifelog. [Accessed: 11-Aug-2018].

[17] R. Gupta, ‘Considering documents in lifelog information retrieval’, in *ICMR ’18 Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, Yokohama, Japan, 2018.

[18] R. Gupta and C. Gurrin, ‘Approaches for event segmentation of visual lifelog data’, in *24th International Conference on Multimedia Modeling (MMM 2018), Proceedings*, Bangkok, Thailand, 2018, vol. 10704.

[19] ‘3.2. Tuning the hyper-parameters of an estimator — scikit-learn 0.19.2 documentation’. [Online]. Available: http://scikit-

learn.org/stable/modules/grid_search.html. [Accessed: 09-Aug-2018].

[20] Qef, *LR*.

[21] Cyc, *SVM*.