Improving Human Activity Classification through Online Semi-Supervised Learning

Hugo Cardoso* and João Mendes-Moreira

INESC TEC, Faculty of Engineering, University of Porto, R. Dr. Roberto Frias s/n, 4200-465 Porto, Portugal
hugo.l.cardoso@inesctec.pt; xkynar.github.io

Abstract. Built-in sensors in most modern smartphones open multiple opportunities for novel context-aware applications. Although the Human Activity Recognition field seized such opportunity, many challenges are yet to be addressed, such as the differences in movement by people doing the same activities. This paper exposes empirical research on Online Semi-supervised Learning (OSSL), an under-explored incremental approach capable of adapting the classification model to the user by continuously updating it as data from the user's own input signals arrives. Ultimately, we achieved an average accuracy increase of 0.18 percentage points (PP) resulting in a 82.76% accuracy model with Naive Bayes, 0.14 PP accuracy increase resulting in a 83.03% accuracy model with a Democratic Ensemble, and 0.08 PP accuracy increase resulting in a 84.63% accuracy model with a Confidence Ensemble. These models could detect 3 stationary activities, 3 active activities, and all transitions between the stationary activities, totaling 12 distinct activities.

Keywords: Human Activity Recognition, Machine Learning, Online Semi-Supervised Learning

1 Introduction

The goal of Human Activity Recognition (HAR) is to develop systems capable of recognizing the actions and goals of a human agent by automatically analyzing these ongoing events and extract their context from the captured data. The detection of human activities, such as walking, running, falling, or even cycling, has several applications, from surveillance systems to patient monitoring systems. Despite being a particularly active field of study in the past years, HAR still leaves many strategies left to explore and key aspects left to address.

There are two main approaches in terms of data extraction: Video and sensors. The sensor approach is, however, the most promising, due to its extreme

* This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project POCI-01-0145-FEDER-016883.

portability and unobtrusiveness. In particular, the introduction of these built-in hardware sensors in many of the modern smartphones, in association with their viral spread throughout the world, unlocked the possibility for the creation of applications based on the context perceived from the data they provide, in a way so vast that it could never have been envisioned a decade ago.

Most sensor-based HAR systems are trained in a static dataset with Supervised Learning (SL) techniques, generating a classification model with a relatively low error rate. However, these systems commonly ignore one of HAR's challenges, the difference of input signals produced by different people when doing the same activities. Consequently, as a user's movements drift from the generic, the system error increases. In fact, each user has his own unique signal, allowing the use of accelerometers to identify them [1]. The activity classification method should therefore be able to generate adapted results for each different user.

The ideal scenario for this problem would be the creation of a smartphone application capable from the beginning of classifying the user's activities with a certain error, and as the time passes and the user utilizes the application, without manual input, the classification error of the system would decrease autonomously until it is virtually insignificant for that specific user.

This document exposes a series of practical experiments performed in an effort to provide a solution to this problem, by using an under-explored technique named Online Semi-supervised Learning (OSSL), an incremental approach capable of adapting the classification model to the user of the application by continuously updating it as the data from the user's own specific input signals arrive.

First, a brief introduction to OSSL will be conducted. Afterwards, for each experiment, the goals, methodology, results and conclusions will be presented sequentially, as an effort to answer some of these undressed aspects. In the end, conclusions about both the advantages and drawbacks of this technique will be discussed.

2 Online Semi-Supervised Learning (OSSL)

Most common Data Mining approaches make use of static datasets. These datasets are collected and organized a priori, and only afterwards analyzed and processed. They also have the advantage of being traditionally labeled, i.e., the activity of the instances used for training and testing the models are known.

Labeled data is massively used by SL techniques. These approaches analyze the data and generate a model, capable of determining the class labels for unseen instances. Systems which perform a single training phase on a static dataset and whose models do not change afterwards are classified as offline.

As interesting as these concepts are, the resulting systems can only be generic. If applied in the intended HAR application, while they might yield decent results using static datasets, gathered from a sample population, they are far from perfect, since as the movements of a specific user drift away from the generic, the classification gets worse. Therefore, the need to explore different approaches increases. Adaptation must be taken into account.

Yet, data coming directly from the smartphone sensors would not be labeled. It is not possible to exploit the conventional supervised learning approaches. All these issues might be addressed by means of Online Semi-supervised Learning.

Semi-supervised Learning has the particularity of operating on both labeled and unlabeled data, typically a small amount of labeled versus a large amount of unlabeled one. The fact is that it is much easier to acquire unlabeled data than labeled one. Labeled data often requires a skilled human agent, that is, an entity which can accurately classify the data. On the contrary, the acquisition of unlabeled data is relatively inexpensive.

OSSL is capable of adapting a classification model, pre-trained by means of supervised learning, to the user of the application by continuously updating it as the data from the user's own specific input signals arrives. It is, therefore, a promising approach to solve the problem in question. If the HAR application learns in an online, semi-supervised fashion, it could start off as generic, but iteratively adapt to a new user, instead of being bound by a generic model, taught by static, pre-gathered data.

3 Empirical Research

After exposing the main architecture of the used dataset, the experiments will be presented in the exact order they were performed. This order was important to tackle the final problem incrementally, adding one layer of complexity to the solution at a time. The experiments were done using MOA - Massive Online Analysis framework [2].

3.1 Dataset

There is a significantly limited collection of activity datasets with easy access and availability to the public. The chosen dataset from within the unfortunately scarce possible solutions was created by *Smartlab*, and consists of experiments carried out by a group of 30 volunteers, within an age bracket of 19-48 years.

It is important to state that *Smartlab* made an invaluable contribution by creating such a complete and organized dataset. The reasons that render the dataset non-optimal are very specific to this particular project, and will be discussed later on.

Each volunteer performed a protocol composed of six basic activities: Standing, Sitting, Laying, Walking, Walking Downstairs, Walking Upstairs. The transitions between the static postures also count as activities: Stand-To-Sit, Sit-To-Stand, Sit-To-Lie, Lie-To-Sit, Stand-To-Lie, Lie-To-Stand. As such, there are a total of 12 activities.

The recording was performed at a constant rate of 50Hz by an accelerometer and a gyroscope of a Samsung Galaxy S II, which is very important since the project should be able to work with readings from a smartphone. Although the dataset comes with a processed version, only the unprocessed version composed of the original raw inertial signals from the smartphone sensors was utilized, since it provided more freedom of choice for custom features and data summarization.

Each person repeated the activity routine twice, totaling around 15 minutes of data per person. As such, the full dataset possess a few hours of activities, which ended up being less than the desired, but enough to achieve results.

3.2 Perfect Segmentation Online Supervised Learning (OSL)

Goals Perfect Segmentation is what we call knowing exactly where an activity starts and where it ends. Perfect Segmentation is virtually impossible in a realistic context, because it is impossible to separate a stream of sensor data into their respective activities automatically. If that was possible, there would be no need for human agents to spend so much time segmenting the collected data for labeling.

However, the goal of this first experiment was not to be realistic, but to serve as a test bed for finding the features that better identify the activity data. Also, it was a good way of finding out the maximum accuracy that could be expected, since it was very unlikely that higher accuracy could be achieved in a realistic context.

Methodology This very simple initial experiment consisted of splitting the dataset by a split factor (usually 50%), creating a train set and a test set, but taking into account that each set must have a fair amount of each activity, to avoid a training set to have no samples of a particular activity. Then, these sets would be pre-processed into ARFF files and fed to a classifier.

Results Numerous combinations of features were experimented, mostly inspired in already existing HAR Studies [3]. The best result was a combination of 20 features, 10 from the accelerometer and 10 from the gyroscope: 1) **Arithmetic Mean** of the X, Y and Z axis of both sensors, individually and also together, resulting in 4 features for each of the sensors; 2) **Standard Deviation** of the X, Y and Z axis of both sensors, individually, resulting in 3 features for each of the sensors. 3) **Pearson Correlation** of axis X and Y, Y and Z, and X and Z, for both sensors, resulting in 3 features for each of the sensors. The results are shown in Table 1.

Table 1. Results for Perfect Segmentation OSL with the final features

	Naive Bayes	VFDT	KNN
Accuracy	87.5 %	87.5%	100%

Conclusions Although incremental algorithms are usually not as powerful as their batch implementations, this experiment proves that even with just SL, it would be possible to achieve positive results if the activities were well separated. Although it is not a realistic situation, these features have demonstrated to characterize the data quite well.

3.3 Perfect Segmentation Online Semi-Supervised Learning (OSSL)

Goals After finding a good set of features, it was necessary to understand what to expect in terms of accuracy boost with the Semi-Supervised approach. Therefore, this second experiment served as an initial familiarization with this technique, in terms of both results and implementation. Perfect Segmentation was still used, since the experiment does not aim at realism, but at testing limits and data dynamics.

Methodology The first implemented OSSL algorithm was a combination of Democratic Co-Learning [4] and Tri-training [5]. As such, a new type of classifier was produced, an ensemble of three classifiers: Naive Bayes, VFDT and KNN.

After the initial supervised training, further training with an unseen instance was only performed if most classifiers agreed on the label they would classify the instance with. Since our ensemble had three classifiers, if at least two agreed, than that new instance would be used with the agreed label for training all the classifiers. If all disagreed, the instance would be discarded.

The testing was done by means of Leave-One-Out Cross-Validation, with the particularity that what is left out is not an instance, but a whole user. The reasoning behind this is due to the fact that an important goal of this entire project is to make the application adapt to a specific user not used for training.

Assuming that the model is trained with a sample as much representative of the population as possible, the data of 29 users are used to train the model and one user is used to test it in a hold-out fashion.

Since the same data cannot be used simultaneously to test the initial supervised classifiers and to train again all the classifiers, the data of the user left out can be: 1) split into two parts, one of them for testing the supervised model and the second one to test the semi-supervised approach, or 2) used according to Prequential Evaluation [6].

In this particular experiment, the data from the user left out was split, which might not have been optimal, but it was good enough to take the desired conclusions. However, later experiments made use of Prequential Evaluation.

Results With this experiment, we can see in Table 2 the Democratic Ensemble Classifier was capable of achieving a Cross-Validation accuracy average of 89.15% before the unlabeled data was presented, and 89.49% after the unlabeled data, that is, an increase of 0.34 percentage points (PP).

Table 2. Results for Perfect Segmentation OSSL

	OSL	OSSL	OSSL − OSL
Democratic	89.15%	89.49%	0.34 PP

Conclusions Taking into account the fact that a single individual's data is only around 15 minutes long, and the data was split in half for train and test, the results are actually motivating.

This experiment proves that OSSL can indeed be used to improve a model's accuracy. As such, it is now time to start looking at the data as a stream.

3.4 Fixed-Length Sliding Window Online Supervised Learning (OSL)

Goals Now that we have a general idea of what to expect in terms of results, it was time to start working on realistic case scenarios. In this experiment, the data is handled as a stream. As such, a fixed-length sliding window was used to iterate the data in the order it was recorded.

Methodology The window size was set to 200, due to not only showing up in several of the past HAR works, but because it indeed presented the most consistent results. Since the data was recorded at a frequency of 50Hz, this means that a window possesses 4 seconds of data.

Both non-overlapping and overlapping sliding windows were tested, with the overlapping having an overlap value of 70%.

Since there is no Perfect Segmentation now, a method of testing the classification accuracy of a model on a window had to be decided. Initially, the activity most present was defined as the label of the window. This meant that if a window has 90% "Walking" and 10% "Standing", the window should represent "Walking".

However, it turned out to be intolerant, since if an activity is just slightly less present in a window, for example 49%, it should still be a valid classification. As such, an updated method of validation stated that an activity should be a valid label if it appeared in a significant amount, such as at least 30%.

Still, in the end, after some deliberation, we defined a classification as correct if the predicted class is present in the window, since it makes sense in a realistic, practical context.

Other than the already employed classifiers, a variant of the previous ensemble classifier was created: Confidence Ensemble Classifier.

In MOA, every classifier has the *getVotesForInstance* method, which returns a list with all the votes the classifier assigned to each activity. As such, for each classifier in the ensemble, the votes were collected, equally scaled, and added. In the end, the final classification is, simply, the most voted activity.

This differs from the Democratic Ensemble Classifier because if a classifier is 99% certain of an activity label, but the other two agree on a different activity with just 30% certainty, then it is not absurd to infer that the first classifier's opinion should be taken into account, despite its numeric disadvantage. Also, this classifier always presents a classification, as opposite to the Democratic, which does not provide a classification when all the classifiers disagree.

Results Several cases were tested in this experiment. Both ensembles and their individual classifiers were tested in Non-overlapping and Overlapping Windows. The results are presented in Table 3.

Table 3. Results for Fixed-Length Sliding Window OSL

	Naive Bayes	VFDT	KNN	Democratic	Confidence
Non-Overlapping	81.22%	81.22%	82.64%	81.22%	82.05%
Overlapping	82.11%	83.33%	82.51%	84.33%	85.41%

Conclusions From the analysis of the obtained results, we can conclude that Overlapping Windows are significant and consistently better than Non-Overlapping Windows.

Another interesting observation is how the ensemble classifiers are being able to provide very competitive results in contrast to the individual classifiers. The best result was indeed obtained from the new Confidence Ensemble Classifier, with an accuracy of 85.41%. As this is now a realistic context, it is a satisfying result, and a good basis for the OSSL approach.

3.5 Dynamic Data Segmentation

Goals This experiment deviated from the path that the previous experiments were taking. In this case, an attempt at discarding fixed-length windows was performed by implementing the Dynamic Data Segmentation algorithm proposed in an article [7] by Kozina et al. Since 100% accuracy was achieved when the activities were perfectly segmented, trying to more accurately separate the stream of data was a logical and worthy effort.

Methodology Data was segmented when a descending acceleration peak higher than a continuously calculated threshold was found:

$$threshold = (avg_{max} - avg_{min}) \times C$$

where C is a constant used to mitigate the impact of noise in the data. It was firstly calculated with the method proposed in the article. However, since it was not providing very good results, the program tested several combinations of N (number of previous data points used to calculate the threshold) and C, in order to find the best possible values.

Results Unfortunately, the results were far worse than expected. As we can see on Table 4, Using the same features than up until now, with the Confidence Ensemble Classifier, an accuracy of only 67.2% was achieved. In an attempt of optimizing the results, we experimented with the same features used in the cited article. Although the results were better, the maximum accuracy achieved was of 75.1%, which was still much lower than desired.

Table 4. Results for Dynamic Data Segmentation

	Our features	Article features
Confidence	67.2%	75.1%

Conclusions There are many reasons why this approach might not have worked as intended. The resulting windows presented many inconsistencies. For instance, an activity like "Standing" could be split into notably varying intervals, ranging from 8 data samples up to 50. The summarization of these windows would therefore result in unpatterned metrics, which are prone to confusing the learning algorithms.

Also, because the algorithm is based on finding sudden descending pikes of acceleration, many situations in which a segment possesses two activities in almost the same quantities happen. The reason is, for instance, that the change between two activities is smooth enough to not be segmented into two windows by the algorithm, failing its main purpose.

It is also hard to define what is the ideal scenario about this kind of dynamic-length sliding window algorithm. If the segmentation is done only after the end of an activity and the start of another, in a real case scenario, this would mean we would only be able to process the window and know which activity we have been doing once we actually have finished it and moved onto the next one. This is an unacceptable user experience. If, however, the goal was to segment each activity in every acceleration drop, we would likely acquire windows too small to provide quality features.

Still, this is mostly speculation, and we actually believe that Dynamic Data Segmentation might have a strong role in solving the HAR challenge. The delayed response could always be fixed by presenting an estimate of the activity being done after the window has a minimum size. In the end, user experience can always be tweaked into feeling right, so it is always worth to further explore this promising approach. Still, since it did not provide satisfying results in our experiments, we embraced Overlapping Fixed-Length Sliding Windows for the remaining of the project.

3.6 Fixed-Length Sliding Window OSSL

Goals After all the previous checkpoint experiments, we are now finally ready to tackle head-on the concept of applying OSSL to improve the accuracy of a generic model.

As such, the goal of this experiment was, fundamentally, the goal of the entire project: to understand whether it is possible to use unlabeled data, acquired from the application's final user, to improve the generic model which composes the initial state of the application.

Methodology As in the previous experiments, a sliding window of size 200 with 70% overlapping was used. Testing was performed with Leave-One-Out Cross-Validation, with Prequential Evaluation. This means that each window of data from the user that was left out would first be used for testing (the model would try to classify the window), and only then for training, if the window was considered a good training sample. This technique allowed to use the data to its full potential.

One of the biggest obstacles that were faced was that several times, a window that was considered a good training sample was labeled incorrectly, which meant the model was being wrongfully trained.

To avoid these mistrainings, we resorted to very high thresholds of certainty. As such, a new instance was only used for further training if every classifier composing the ensemble had at least 99.9% of their votes in the same activity. It might seem too restrictive, but in the end, it was more desirable to discard an instance than to use it it wrongly.

This instance validation method also allow us to use independently the classifiers composing our ensembles, since each classifier is capable of voting. Therefore, although each classifier and ensemble has their own ways of classifying an instance, in this experiment they all used the same method to determine trainable instances.

Results Table 5 shows the results of every test performed in this experiment. For each classifier or ensemble, the model accuracy was tested before and after the application of the unlabeled data by the Semi-Supervised methods. The table also contains a column with the difference between the pos and pre accuracies, for easier interpretation.

Table 5. Results for Fixed-Length Sliding Window OSSL

	Naive Bayes	VFDT	KNN	Democratic	Confidence
OSL	82.58%	83.56%	76.29%	82.89%	84.55%
OSSL	82.76%	83.47%	72.72%	83.03%	84.63%
OSSL - OSL	0.18 PP	-0.09 PP	-3.57 PP	0.14 PP	0.08 PP

As can be seen from the analysis of the result table, the ensembles remain as the classifiers with the most consistent results. The Democratic Ensemble was capable of achieving an average improvement of 0.14 percentage points, and despite the Confidence Ensemble providing a lower average improvement (0.08 percentage points), its final accuracy is the highest (84.63%). Naive Bayes was also able to achieve positive results even independently, with an average accuracy gain of 0.18 percentage points for a final accuracy of 82.76%. VFDT and KNN behaved not so well as independent classifiers. In average, the SSL approach reduced the models performance. However, when working as an ensemble, their view of the data was beneficial at achieving consistently positive results.

Conclusions This experiment was very gratifying as it proves that OSSL can be used in practice to improve the accuracy of a model, achieving a better HAR system.

The small amount of data per person (15 minutes) is a strong obstacle that might explain the low increase of performance. However, increases were achieved with nothing but some unlabeled data.

As such, it is indeed possible that, by simply processing a stream of unlabeled data extracted from smartphone sensors, a generic model improves by itself, and classifications that were once incorrect become accurate.

3.7 OSSL Using The Author's Data

Goals With the desire to take one step further into proving that OSSL has a realistic place in achieving accurate and autonomous HAR systems, the first author himself decided to record some of his own data in similar conditions to that of the original dataset.

The goal of this experiment was to prove that a generic model, fully trained in a dataset built in lab conditions, can improve its performance and adapt to anyone, even to the first author of this paper.

Methodology With the help of a waist belt, a Smartphone Galaxy S3 and a stopwatch (for ease of labeling in order to find the accuracy improvements), the author recorded himself performing a routine containing the same activities used in this project:

Standing –> Sitting –> Laying –> Sitting –> Standing –> Laying –> Standing –> Walking –> Walking Upstairs –> Walking Downstairs

The routine was repeated three times, producing a total of roughly 50 minutes of data. Classifiers were trained in a supervised fashion in the entirety of the original dataset, resulting in generic models. These models were tested in the user's data to acquire an initial accuracy value. Then, the models employed the previous Semi-Supervised techniques to further train using the new, recorded data. Prequential Evaluation was used in order to acquire understandable metrics and make the most out of the author's data.

Results Due to the fact the original dataset and the author's data were collected in unequal conditions, especially in terms of the waist belt used, some inconsistencies in the data patterns were produced. These inconsistencies tended to confuse most classifiers, and because of that, the threshold for an instance to be considered training material had to be increased from 99.9% to 99.99%. With this certainty, Naive Bayes was the classifier which better adapted to these inconsistencies, producing an accuracy increase from 60.17% to 60.25%.

100% certainty threshold was also tested for this experiment, with the Confidence Ensemble Classifier providing an accuracy boost from 62.18% to 62.22%. However, in the end, all classifiers presented very low overall accuracies.

Table 6. Results for Fixed-Length Sliding Window OSSL with the author's data

	OSL	OSSL	OSSL − OSL
Naive Bayes (99.99% thld)	60.17 %	60.25%	0.08 PP
Confidence (100% thld)	62.18 %	62.22%	0.04 PP

Conclusions We can conclude from this experiment that even small variations in the data gathering conditions tend to affect the accuracy of the model. Most results from this experiment were adverse, due to the fact that a good generic model is essential as a base for the model to be capable of training itself. The role of OSSL in these systems should not be to turn a bad classifier into a good one, but to turn a good classifier into a better one.

Despite the results, OSSL was still capable of improving the accuracy of some models. It is very interesting to think that the 40 minutes of activities the author has performed were able to help a classification model to improve its own accuracy, even if only by a little. This experiment was very important to understand both the role and the limitations of OSSL.

4 Conclusion and future work

The empirical research was capable of demonstrating that OSSL may indeed provide improvements to a generic model, adapting it to a specific user. Accuracy gains of up to 0.18 PP were achieved with just 15 minutes of unlabeled data.

It was also concluded that OSSL works better when the base generic model has a good initial accuracy, since a competent classifier is more qualified for self-training. Therefore, it excels at making good classifiers even better.

However, a lot of research is still essential to even think about turning this technology into an everyday tool. In terms of the chosen dataset, its main inadequacy to the project was due to the fact that the project focused on proving that OSSL improved a generic model for a specific user. This means that while the generic model can be trained from the data of several individuals, the data used for testing and Semi-Supervised training must come from a single user. As such, it would be important to have recordings for each individual longer than 15 minutes.

The dataset was also recorded using a waist belt, which ends up being very obtrusive. It would be much more realistic whether the new dataset had the smartphone located in the user's front pocket. However, that approach should be further researched because orientation matters. Very different data is produced depending on the smartphone orientation even when inside the pocket, which will likely confuse the classification model and render the application useless. This is unacceptable in terms of user experience. As such, a method for converting sensor data from the smartphone orientation to a generic orientation should be developed. The paper [9] is an attempt to solve this issue, and it should serve as a basis for additional attempts, especially applied to HAR and OSSL.

The addressing of the stated considerations may or may not be enough to solve the massive challenge that is Human Activity Recognition, but we believe they are key steps in turning this technology into an everyday tool.

References

1. Pisani, P. H., Lorena, A. C., & De Carvalho, A. C. P. L. F. (2014). Adaptive algorithms in accelerometer biometrics. Proceedings - 2014 Brazilian Conference on Intelligent Systems, BRACIS 2014, 336–341. http://doi.org/10.1109/BRACIS.2014.67
2. Bifet, Albert, Holmes, Geoff, Kirkby, Richard & Pfahringer, Bernhard (2010). Moa: Massive online analysis. Journal of Machine Learning Research, 11, 1601-1604.
3. Cruz Silva, N., Mendes-Moreira, J., & Menezes, P. (2013). Features Selection for Human Activity Recognition with iPhone Inertial Sensors. Advances in Artificial Intelligence, 16th Portuguese Conference on Artificial Inteligence., 560–570.
4. Zhou, Y., & Goldman, S. (2004). Democratic co-learning. In 16th IEEE International Conference on Tools with Artificial Intelligence (pp. 594–602). IEEE Comput. Soc. http://doi.org/10.1109/ICTAI.2004.48
5. Zhou, Z. H., & Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering, 17(11), 1529–1541. http://doi.org/10.1109/TKDE.2005.186
6. Dawid, A. P. (1984). Statistical theory: the prequential approach. Journal of the Royal Statistical Society. Series A, 147, 278–292.
7. Kozina, S., Lustrek, M., & Gams, M. (2011). Dynamic signal segmentation for activity recognition. Proceedings of International Joint Conference on Artificial Intelligence, 1–12.
8. R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.
9. Tundo, M. D., Lemaire, E., & Baddour, N. (2013). Correcting Smartphone orientation for accelerometer-based analysis. MeMeA 2013 - IEEE International Symposium on Medical Measurements and Applications, Proceedings, (May), 58–62. http://doi.org/10.1109/MeMeA.2013.6549706