Automatic Visual Analysis of Real-World Events Covered By Social Media Using Convolutional Neural Networks

Henning Hamer, Andreas Merentitis, Nikolaos Frangiadakis, and Sergey Shukanov

AGT International,
64295 Darmstadt, Germany
E-mail: {hhamer, nfrangiadakis, ssukhanov}@agtinternational.com, andreas.merentitis@ieee.org

***Abstract*—This paper investigates how well real-world events can be characterized by visual features detected in related images posted on social media, using state-of-the-art computer vision methods for object detection and classification. Over 48k images from four different events have been processed to detect objects of different types using convolutional neural networks (CNNs) and cascaded classifiers. Based on these object detections we train different classifiers to rank object types supporting the respective event and to discriminate images of an event from other images. Possible applications include: 1) finding images of a certain event in a semi-automatic way, and 2) classifying the type of an event.**

I. Introduction

Social media has long been recognized as a rich source of user-generated information containing valuable insights. Such information has been used for marketing (e.g., to assess the success of advertising campaigns) [16], crime prediction [29], and to find information relevant for news [17]. Microblogging service Twitter is one of the most popular sources of user-generated content, and it is used for scientific research as well as for commercial applications [19], [1]. While the rich, dynamic, and heterogeneous data provided by this service contains extremely useful information, Twitter streams at the same time can be noisy, irrelevant and even harmful. This fact imposes major challenges when analyzing such data and requires new strategies and approaches.

Recently, real-world event detection has become a major trend in the field of user-generated content analysis, the challenge being to summarize and categorize uncoordinated messages from various users. Political events, festivals, accidents, and natural disasters are all representatives of real-world events happening at a particular location and time. Rapid event discovery and analysis from social media can be of crucial importance, e.g., for emergency response.

In the past, most works in the area of event detection focused on the analysis of text (i.e. user messages) as well as meta-tags such as geo-locations, time-stamps, etc. Kaisser et al. [28] have detected real-world events in a given geographic region based on spatio-temporal cluster analysis. At first, tweets with similar geographical and temporal attributes are grouped together forming a set of candidate clusters, followed by a feature extraction and a classification step in which each cluster is classified as an event or non-event. A similar system was proposed by Becker et al. [4]. In [11] a scalable distributed event detection framework was proposed where events can be detected in high volume data streams in real-time. For this purpose a lexical key partitioning approach with K-Means Clustering was used.

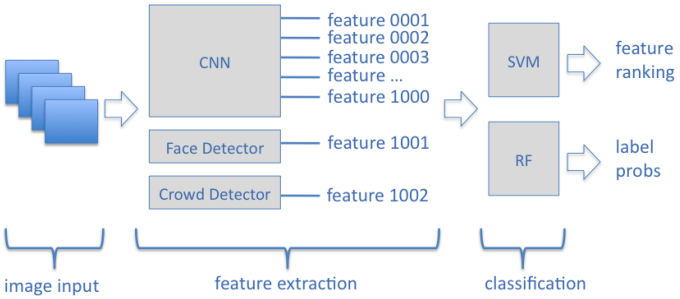

Fig. 1. Processing pipeline - The input is a number of images associated with real-world events. Object detections are extracted using a CNN and cascaded classifiers. These detections are then used as features during the classification stage. SVMs are applied to rank object types supporting events, random forests are used to discriminate images of an event from other images.

Less work has been done with respect to event detection based on the visual information in images attached to social media posts. There are several reasons for that: high processing power and memory requirements, high complexity of feature extraction algorithms, and the challenge of ambiguous interpretation of images. However, images usually reveal a lot of details not captured by text messages and meta-information.

Some works implement event detection by combining textual and visual information from tweets. In [3] the output of a text-based and an image-based detection system is fused to obtain their final results. For the textual event detection a bag-of-words approach with a weighting scheme is used. For the visual analysis Histogram of Oriented Gradients (HOG) descriptors, Grey-Level Co-occurrence Matrix (GLCM) and color histograms with support vector machine classifiers are applied. Recently, a clustering algorithm that employs both features extracted from photos and text was proposed in [2]. A kernel Canonical Correlation Analysis (CCA) is used here

to reduce the dimensionality of photos to be able to cluster these photos into different groups. Then text as well as meta-data features are employed to determine the best combination of feature sets.

Recently, a great effort with respect to visual event discovery was made by the organizers of the Social Event Discovery (SED) challenge as a part of the annual MediaEval benchmarking initiative. Over the past four years SED has uncovered different techniques for event detection and classification based on social media. In 2011 [13] a set of Flickr images with their meta-data was provided and the task was to discover particular events and to return related media items. Beside approaches focusing on the processing of meta-data and queries of additional data sources from the web, several image-based ones were also proposed. In [21] a visual pruning approach was used to filter out noisy and irrelevant items from a set of photos originally obtained by a text-based matching method. In [30] photos are filtered based on visual information obtained using the IBM Multimedia Analysis and Retrieval System [23]. This allowed to discard invalid event clusters. In 2012 [14] the challenge was similar to the one in 2011. Given a set of Flickr images with meta-data the task was to discover social events and detect related media items. As in 2011 most of the proposed methods exploited the timestamps and geo-tags of the images to performing some sort of clustering. Several image-based approaches including a bag-of-features model extracted from photos [31] and topic discovery using Latent Dirichlet Allocation with Gibbs sampling [26] were also presented. The dataset of SED challenge in 2013 [7] consisted of Flickr and Instagram images and Youtube videos, all with the respective meta-data. In [15] text and visual features were used to classify events versus non-events. Here, RGB-SIFT feature extraction was performed as in [22] and then an SVM was applied as a classifier. Along with text features also GIST features were extracted in [5] to train an SVM to determine whether an item belongs to an event or not. In 2014 [8] again a Flickr set of images was provided. A Ranking-based Clustering approach given a set of events was proposed in [24]. In [12] image descriptors similar to the one in [24] were used. Here, in addition a bag-of-visual-words (BoVW) framework was applied to generate a visual descriptor.

In this paper we are not concerned with the discovery/detection of events. Instead, we assume that the presence of an event to be analyzed is already known, e.g. from one of the event detection approaches above or from news on the media. We then work on tweets associated with such events based on the known time and geo-location. After this initial selection of tweets we use no meta or text information, but focus exclusively on the posted images. While e.g. van Kasteren et al. [25] have shown that there is information encoded in the image posting behavior of users (e.g. more images are posted during events, a lot of images are duplicates etc.) we focus on the pure image content. Our contributions are the following: 1) We introduce a processing pipeline combining convolutional neural networks and cascaded classifiers for feature extraction followed by a subsequent classification stage, 2) we conduct a visual inspection of object detections produced by a CNN applied to social media images 3) we present a ranking of object types with respect to different events as well classification results.

II. Processing Pipeline

Given a set of images associated with an event based on a certain time stamp and a certain geo-location, our goal is to find visual elements characterizing the event.

Figure 1 gives an overview of our processing pipeline. We follow a hybrid approach: one the one hand we exploit the discriminative power of a convolutional neural network (CNN) trained on a lot of object classes not specific to our task. On the other hand we introduce semantic knowledge specific to our problem via a face and a crowd detector based on the assumption that these detectors are relevant for events. Object detections from the CNN and the cascaded classifiers are then combined and used as features at the classification stage for the ranking of object types and for classification. The next sections introduce the individual processing steps of the pipeline.

A. Filtering of Input Images

At the beginning of our pipeline we filter out images not suited for our purposes. First we only consider images with a width and a height greater than 200 pixels. Images smaller than that (logos, profile images, etc.) are discarded. To focus on unique image content we then remove duplicate images 1) via their file size and 2) based on an image similarity measure: all images are converted to gray-scale, resized to a thumbnail size of 25x25 pixels, and compared in pair-wise manner. If the summed difference of the gray values of two thumbnails is smaller than an empirical threshold of 1000 we remove the potential duplicate. After this there may still be some similar images in the data set, e.g. due different crops of the same image, but we do not consider these as duplicates.

B. Extraction of CNN Features

CNNs are a special type of neural networks where the learned weights of the first layer correspond to image convolution kernels. A current trend in Computer Vision is to use pre-trained CNNs to extract features as input for other classification techniques. We follow this trend and extract CNN features from our images using the Caffe framework [18] and the BVLC CaffeNet Model. Caffe is a popular toolkit for training and applying deep CNNs with GPU support. The BVLC CaffeNet Model is a ready-to-use CNN (based on the AlexNet model [20]) that comes with Caffe and was trained on the ImageNet data set [9] to discriminate between a 1000 ImageNet object types in images. Given an input image the CaffeNet model has to be applied to candidate windows called *object proposals*. To generate such object proposals we use the selective-search method and implementation provided by [10].

A common procedure when using CNNs for feature extraction is to remove the last couple of layers of the trained CNN and to use the output of the last remaining layer as input for classification [6]. Instead we do not remove any layers and use the object detections of the last layer of the CaffeNet model as our features. This allows a seamless integration with additional object detections produced with cascaded classifiers.

To clarify the term *object detection*: when the Caffe framework assigns an ImageNet object type to an object proposal with a certain confidence (e.g. using the BVLC CaffeNet model), we call this an object detection (of the respective

object type). E.g. the assignment of the *Monkey* object type to an object proposal is called a *Monkey* detection at the position of the object proposal with a certain confidence.

In each image we find the top ten object types with regard to the greatest confidence of assignment across all object proposal. The feature vector of each image is a 1000-dimensional vector, where each element is set to 1 if the respective object type is within the top ten object types of the image and 0 otherwise. Two dimensions are then added to this vector as described in the next section.

C. Extraction of Features using Cascaded Classifiers

To detect objects of a specific type in images cascaded classifiers are trained with many images showing objects of exactly that type. Compared to CNNs these classifiers can be seen as highly specialized experts. E.g. [27] showed that cascaded haar-classifiers are well suited for face detection.

We employ cascaded classifiers to make use of our domain knowledge: since we assume that in the context of events the presence or absence of faces and crowds is significant we apply a cascaded face and crowd detector to all images. We then count the number of faces and crowds detected in each image and use these counts to augment the 1000-dimensional feature vector of the image, yielding a 1002-dimensional vector.

The implementation of cascaded classifiers we use is the one provided by OpenCV. For face detection we chose the readily available *haarcascade_frontalface_default.xml* cascade. For crowd detection we trained a HoG cascade with a set of crowd and non-crowd images.

D. Classification

To rank object types supporting an event and to discriminate images of an event from other images we use two of the most widely adapted classifiers, namely support vector machine (SVM) and random forest (RF). SVMs are among the top performing classification algorithms, basing their success on structural risk minimization (or margin maximization) to improve generalizability and on the kernel trick to efficiently separate the data in a higher dimensional space. On the other hand, random forest is a very popular ensemble classification method based on a multitude of weak learners (decision trees) in order to generate a strong learner. Among its main advantages is the ability to handle noise and outliers in both the samples as well as the labels, as well as the good scaling properties with respect to the number of training samples.

In this work we use *linear* SVMs, since the feature space is already high dimensional making a mapping to a higher dimensional space less relevant, while the high number of training samples of some of the experiments makes this choice also beneficial from a computational point of view. Linear SVMs are also easily interpretable with respect to the importance of each feature for the classes of interest. On the other hand, random forests are better suited for providing not only an estimated label but also an estimated posterior probability (Platt scaling can do something similar converting SVM distances from the hyperplane to posterior estimates but this requires careful balancing of the estimates). Our framework supports both classifiers (typically they perform very similar in terms of accuracy), using SVMs for feature ranking to find supporting object types per training set, and using the posterior from the random forest for cases that require a soft estimate (e.g. to find images of an event based on a confidence threshold).

TABLE I. DATA SET - IMAGES FROM FOUR REAL-WORLD EVENTS CONSIDERED HERE[1]

event	#tw	#days	#tw/#days	#img/#tw	#filt. img/#tw
Acapulco	202.915	12	16.910	0.170	0.068
Antwerp	110.161	6	18.360	0.168	0.088
Philippines	597.043	10	59.704	0.156	0.076
Zurich	41.102	4	10.276	0.207	0.100

[1]Columns correspond to the total number of tweets captured, the number of days captured, the number of tweets per day, the number of images per tweet, and the number of filtered images per tweet.

TABLE II. DATA SET DETAILS - STATISTICS OF THE DATA SET WITH RESPECT TO THE TIME OF CAPTURING OF THE TWEETS: SHORTLY BEFORE, DURING, OR AFTER THE RESPECTIVE EVENT[2]

event	time	#img	#filt. img	≈#event img
Acapulco	Before	7.296	3.808	
	During	7.560	2.255	659
	After	19.590	7.697	
Antwerp	Before	3.474	2.008	
	During	11.463	5.823	977
	After	3.601	1.854	
Philippines	Before	59.288	27.212	
	During	11.856	6.181	1100
	After	22.290	12.009	
Zurich	Before			
	During	2.715	1.515	181
	After	5.789	2.589	

[2]The last three columns correspond to the number of images, the number of filtered images, and the approximate number of event images.

III. DATA SET

As a basis for our experiments we use the data set provided by [25] consisting of tweets associated with 15 real-world events (crowd gatherings, natural disasters, accidents, and terrorist attacks). Those tweets were retrieved based on the known geographic location and time span once the respective event had been reported by the media. The data set also comes with over a million image links contained within the tweets. The amount of images per event varies largely and we discarded all events with an insufficient number of images. Also, we discarded the data from all accidents and terrorist attacks due to images of extreme violence. This reduction yields the subset of the data described in Table I covering the following events: the flooding disaster in Acapulco in September 2013, the Tomorrowland festival in Antwerp in July 2013, the earth quake near the Philippines in October 2013, and the ZuriFascht festival in July 2013. Other data sets (e.g. the SED data sets) do not contain images from natural disasters.

Event images are divided into images captured shortly before, during, and after the event (see Table II). We only consider the *Before* and *During* images. While post-event analysis (using the *After* images) is an interesting topic it is out of the scope of this paper. To obtain the approximate number of actual event images we manually annotated a randomly permuted selection of 6.121 images from *During* as either *event* or *non-event*

Fig. 2. IMAGES SHOWING FIREWORKS AND A SEA URCHIN. BOTH ARE SIMILAR FROM A VISUAL PERSPECTIVE.[1]

IV. RESULTS

Our classification framework addresses two question of interest: 1) can we find object types supporting a given event and 2) can images from an event be separated from other images based on object detections. To explore these questions we partition the images in two ways according to the time of capturing (*pos* and *neg* stand for positive and negative samples):

1) *Before partition* - pos: all *Before* images of a certain event, neg: all *Before* images from all other events
2) *During partition* - pos: all *During* images of a certain event, neg: all *During* images from all other events

Before we discuss the results of classifier training on these partitions we begin with a visual inspection of the Caffe detections.

A. Visual Inspection of Caffe Detection

Every object proposal is associated with a 1000-dimensional vector, each dimension corresponding to the confidence of the assignment of one of the 1000 object types. In each image we find the top ten object types with regard to the greatest confidence of assignment across all object proposal. We then examine the corresponding object proposals and their assignment, i.e. the corresponding object detections.

During visual inspection we observed that the object types assigned to objects in the images are visually very similar to the actual object types, demonstrating the expressive power of the BVLC CaffeNet model. To give just one example, in one of the images from Acapulco a military truck crossing a flooded street is detected as a *Garbage Truck*. These two object types look very much alike. Keep in mind that we are not concerned with the exact names of the ImageNet object types. Our primary interest lies on characteristic visual elements representing well images of the same event. False alerts (i.e. the assignment of wrong object types) during feature extraction can still be valuable to characterize events.

Most images from Acapulco show people, vehicles and buildings near streets flooded with brown water. The *Garbage Truck* detection mentioned above is one of the detections in these images, but we did not observe many detections of this particular type in the other images of that event. In contrast, we encountered multiple screen shots of web pages containing refuge information, so the respective *Web Site* detections could be more significant. Images of houses in the water (sometimes triggering *Boat House* detections) have also been observed repeatedly. In the case of Antwerp, there are many images of the main concert stage, resulting in a large amount of *Stage* detections. Since this main stage was designed to look like a smoking volcano, we also observed a number of *Volcano* detections. During the Philippines earthquake many people posted images of historic buildings which took damage, producing many *Monastery* and *Church* detections. Finally, many pictures from the ZuriFascht festival show the firework over the lake, often leading to *Sea Urchin* detections. Figure 2 illustrates this visual similarity.

[1]Renata Apanaviciene, Peter Leahy / Shutterstock.com

TABLE III. MOST FREQUENT CAFFE DETECTIONS SORTED BY FREQUENCY. FREQ. INDICATES THE PERCENTAGE OF IMAGES IN WHICH THE RESPECTIVE OBJECT TYPE WAS WITHIN THE TOP 10 CONFIDENCES.

object type	**freq.**	**object type**	**freq.**
(01) Theater Curtain	0.62	(06) Windsor Tie	0.28
(02) Velvet	0.54	(07) Dishwasher	0.23
(03) Window Shade	0.41	(08) Geyser	0.21
(04) Nematode	0.40	(09) Shower Curtain	0.19
(05) Milk Can	0.28	(10) Plate Rack	0.17

TABLE IV. SVM FEATURE RANKING

Acapulco	**Antwerp**	**Philippines**	**Zurich**
001: Mobile Home	001: Stage	001: Monastery	001: Sea Urchin
002: Snowplow	002: Toy Shop	002: Med. Chest	002: Alp
003: Dam	003: Volcano	003: Spatula	003: Pickelhaube
004: Moving Van	004: Ball Player	004: Gyromitra	004: Nail
005: Fox Squirrel	005: Beer Glass	005: Neck Brace	005: Dining Table
071: Crowd (casc.)	052: Face (casc.)		

B. Feature Ranking

Table III gives an overview of the most frequent Caffe detections of all *During* images of the entire data set. The frequency of an object type is defined by the percentage of images in which the object type ranks within the top ten (as described above). Some of these features can almost be seen as basic image components, e.g. very small "Theater Curtain" detections bear resemblance with haar-wavelets (two horizontal bars above each other).

The most frequent object detections are not necessarily the most characteristic/discriminative ones. Table IV shows the result of feature ranking when training a linear SVM on the *During partition* described above. The feature vector of each image is a 1000-dimensional vector, where each element is set to 1 if the respective object type is within the top ten object types of the image and 0 otherwise. Then two dimensions are added containing the count of face and crowd detections in the image, yielding a 1002-dimensional vector.

While some of the top ranking object types can nicely be interpreted when looking at the respective images it is more difficult for others. In the case of Acapulco, none of the *Mobile Home* detections actually correspond to mobile homes. Some are triggered by houses in the images, others seem arbitrary. The *Snowplow* and *Moving Van* detections correspond to various vehicles on the flooded streets and make more sense intuitively. The *Dam* detection trigger near the water in the images, although there are no actual dams. All in all the Acapulco data set seems to be the most difficult one in terms of finding characteristic object types. The Antwerp detections are much more easy to understand. Most of the *Stage*, and *Beer Glass* detections are true detections and they describe well the music festival. So do the *Volcano* detections: they also trigger on the stage designed as a smoking volcano. Also,

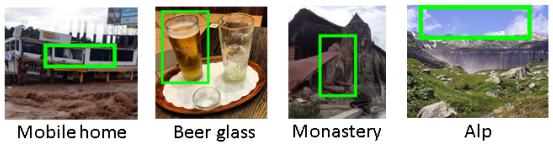

Fig. 3. OBJECT DETECTIONS OF HIGH RANKING OBJECT TYPES

the *Ball Player* detections are true detections, they originate from a soccer game going on in parallel in Antwerp. For the Philippines, the *Monastery* detections clearly correspond to the large amount of photos of damaged buildings. *Neck Brace* detections often trigger on faces. The other object types of Philippines are not straight-forward to interpret. For Zurich, as expected the *Sea Urchin* detections represent well a lot of the images of the firework over the lake. The *Alp* detections are also true detections, reflecting the fact that Zurich is surrounded by mountains. Figure 3 shows one detection of a high ranking object type per event.

Surprisingly the cascaded face and crowd detectors do not have high significance. For Acapulco and Antwerp the ranking of their detections is quite low, for the Philippines and Zurich they are not even reported in our statistics. The face detections could be redundant e.g. due to *Neck Brace* detections. Our crowd detector reliably detects crowds but also produces many false alerts, maybe the false alerts reduce its significance.

C. Classification

The images of the *Before* partition mostly reflect the location of the event (since the event itself has not yet started). Images of the *During* partition contain a mix of location and event specific image elements.

The performance (Accuracy) and the respective harmonic mean of precision and recall (F1-measure) of the two data partitions can be seen in Table V. Looking at the F-measure, classification works better for the *During partition* in the case of Acapulco and Antwerp, but not in the case of Philippines. Philippines performance is not better here because it was dominating the Before classification cases due to its large amount of images. For Zurich no images captured before the event were available. In some cases there is a tradeoff between accuracy and F-measure, meaning that the event with the highest accuracy does not also have the highest F-measure. This is due to the fact that the one-vs-all classification flows result in an unbalanced data set, where the class of interest is far less frequent compared to the majority class.

TABLE V. ACCURACY AND F-MEASURE FOR THE TWO DATA PARTITIONS

Acc/F-Measure	Before	During
Acapulco	0.87 / 0.46	0.83 / 0.51
Antwerp	0.93 / 0.52	0.62 / 0.57
Philippines	0.88 / 0.70	0.59 / 0.52
Zurich	-	0.89 / 0.46

D. Computation Time

In total we processed 48.802 images. With $\approx$ 623 object proposals and a processing time of $\approx$ 16.4 ms per object proposal (average numbers measured for a sample of 25 images) the total processing time of Caffe amounts to $\approx$ 139 hours of processing time on a standard PC (2.4 GHz Intel (R) Xeon (R) CPU) and a GeForce GTX 760. cuDNN did not seems to result in a speed-up. However, we were able to run three processes in parallel (each accessing the GPU) to reduce the processing time to of Caffe to $\approx$ 46 hours. Cascaded classifiers are quite fast with GPU support. Assuming 30 ms per image processing all images takes $\approx$ 30 min per cascaded classifier. Before training classifiers we convert all *h5* files delivered by Caffe to our own format. This takes $\approx$ 3 hours. Training all classifiers takes less than an hour.

V. CONCLUSION

In this work we have demonstrated that state-of-the-art, off-the-shelf CNNs have sufficient expressive power to visually characterize sets of images associated with real-world events covered by social media. We proposed a processing pipeline allowing the seamless integration of object detections from a CNN and several cascaded classifiers. Using linear SVMs we identified top ranking object types with respect to each of the four real-world events in our data set. With random forests we were able to classify images associated with real-world events, based exclusively on their visual content. Future work includes investing in more detail why the cascade classifiers had low impact. Also, it would be interesting to train a CNNs from scratch with images from social media, e.g. using accompanying hash tags as annotation.

ACKNOWLEDGMENT

The authors thank AGT International for funding this research. We also thank Tim Van Kasteren and Maria Niessen for data collection, and Nakul Gopalan for his support.

REFERENCES

[1] Apoorv et al. Agarwal. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38, 2011.

[2] U. Ahsan and I. Essa. Clustering social event images using kernel canonical correlation analysis. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 814–819, 2014.

[3] Samar M. Alqhtani, Suhuai Luo, and Brian Regan. Fusing text and image for event detection in twitter. *CoRR*, 2015.

[4] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, 2011.

[5] Markus Brenner and Ebroul Izquierdo. Mediaeval 2013: Social event detection, retrieval and classification in collaborative photo collections. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop, Barcelona, Spain*, 2013.

[6] Ali Sharif Razavian et al. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, 2014.

[7] David Corney et al. Socialsensor: Finding diverse images at mediaeval 2013. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop, Barcelona, Spain*, 2013.

[8] Georgios Petkos et al. Social event detection at mediaeval 2014: Challenges, datasets, and evaluation. In *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain*, 2014.

[9] J. Deng et al. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[10] J. R. R. Uijlings et al. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[11] Richard McCreadie et al. Scalable distributed event detection for twitter. In *Proceedings of the 2013 IEEE International Conference on Big Data, 6-9 October 2013, Santa Clara, CA, USA*, pages 543–549, 2013.

[12] Simon Denman et al. SAIVT-ADMRG @ mediaeval 2014 social event detection. In *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain*, 2014.

[13] Symeon Papadopoulos et al. Social event detection at mediaeval 2011: Challenges, dataset and evaluation. In *Working Notes Proceedings of the MediaEval 2011 Workshop, Santa Croce in Fossabanda, Pisa, Italy*, 2011.

[14] Symeon Papadopoulos et al. Social event detection at mediaeval 2012: Challenges, dataset and evaluation. In *Working Notes Proceedings of the MediaEval 2012 Workshop, Santa Croce in Fossabanda, Pisa, Italy*, 2012.

[15] Truc-Vien Nguyen et al. Event clustering and classification from social media: Watershed-based and kernel methods. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop, Barcelona, Spain*, 2013.

[16] Atefeh Farzindar. Industrial perspectives on social networks. In *EACL 2012 - Workshop on Semantic Analysis in Social Media*, 2012.

[17] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, 2007.

[18] Yangqing et al. Jia. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[19] Long et al. Jiang. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 151–160, 2011.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[21] Xueliang Liu, Raphaël Troncy, and Benoit Huet. Eurecom @ mediaeval 2011 social event detection task. In *MediaEval Benchmarking Initiative for Multimedia Evaluation*, Pisa, Italy, 2011.

[22] Riccardo et al. Mattivi. Exploitation of time constraints for (sub-)event recognition. In *Proceedings of the 2011 Joint ACM Workshop on Modeling and Representing Events*, pages 7–12, 2011.

[23] Apostol et al. Natsev. Ibm multimedia analysis and retrieval system. In *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval*, pages 553–554, 2008.

[24] Taufik Sutanto and Richi Nayak. Ranking based clustering for social event detection. In *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain*, 2014.

[25] Tim van Kasteren et al. Analyzing tweets to aid situational awareness. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, Proceedings*, pages 700–705, 2014.

[26] Konstantinos N. Vavliakis, Fani A. Tzima, and Pericles A. Mitkas. Event detection via LDA for the mediaeval2012 SED task. In *Working Notes Proceedings of the MediaEval 2012 Workshop, Santa Croce in Fossabanda, Pisa*, 2012.

[27] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[28] Maximilian Walther and Michael Kaisser. Geo-spatial event detection in the twitter stream. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, pages 356–367, 2013.

[29] Xiaofeng Wang, Matthew S. Gerber, and Donald E. Brown. Automatic crime prediction using events extracted from twitter posts. In *Proceedings of the 5th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 231–238, 2012.

[30] Yanxiang Wang, Lexing Xie, and Hari Sundaram. Social event detection with clustering and filtering. In *Working Notes Proceedings of the MediaEval 2011 Workshop, Santa Croce in Fossabanda, Pisa, Italy, September 1-2, 2011*, 2011.

[31] Matthias Zeppelzauer, Maia Zaharieva, and Christian Breiteneder. A generic approach for social event detection in large photo collections. In *MediaEval 2012 Multimedia Benchmark Workshop*. CEUR-Proceedings, 2012.