Manifold Learning for Real-World Event Understanding

Caroline Mazini Rodrigues, Aurea Soriano-Vargas, Bahram Lavi, Anderson Rocha, Senior Member, IEEE, and Zanoni Dias

Abstract—Information coming from social media is vital to the understanding of the dynamics involved in multiple events such as terrorist attacks and natural disasters. With the spread and popularization of cameras and the means to share content through social networks, an event can be followed through many different lenses and vantage points. However, social media data present numerous challenges, and frequently it is necessary a great deal of data cleaning and filtering techniques to separate what is related to the depicted event from contents otherwise useless. In a previous effort of ours, we decomposed events into representative components aiming at describing vital details of an event to characterize its defining moments. However, the lack of minimal supervision to guide the combination of representative components somehow limited the performance of the method. In this paper, we extend upon our prior work and present a learning-from-data method for dynamically learning the contribution of different components for a more effective event representation. The method relies upon just a few training samples (few-shot learning), which can be easily provided by an investigator. The obtained results on real-world datasets show the effectiveness of the proposed ideas.

Index Terms—Manifold learning, event understanding and reconstruction, image representation, image components, digital forensics.

I. INTRODUCTION

NOWADAYS, millions of people are sharing vast amounts of information (e.g., texts and images), through social networking platforms such as Twitter, Facebook and Instagram [1]. This fact opens up an opportunity to obtain relevant insights to understand many events taking place around us. However, understanding events could be a difficult task, especially when the available data is not reliable. Consider an event (e.g., terrorist attack) taking place in a crowded place, e.g., a stadium or a theatre. Many persons equipped with mobile phones are likely to witness that event in person. In a few minutes, hundreds of texts, pictures, and videos would be shared on social media [2] depicting and describing what happened. These data might improve events understanding by establishing, for example, the order of incidents, objects position, and possibly people involved.

In fact, some works in the literature have studied textual content dissemination, as tweets, during emergency events [3]–[5]. The tweets behavior understanding could improve government response during crises, for example. In this line, a diversity of works explores the textual information [6] or the correlation of texts and images [7], [8] to deal with disasters, generally focusing more on textual features than on visual semantics. In contrast, our goal is to filter data relative to an event to understand and reconstruct the event, for forensic applications. Images are one of the most valuable sources of documentation on events [9], [10]. Therefore, here we focus on images and how to represent their semantics.

When dealing with pool of images from an event, filtering what is relevant is one major challenge. The crucial data, which could indeed represent the event, might be mixed with massive amounts of non-important data. Hence, the question becomes: How to automatically separate representative images from non-representative ones? In a previous work [11], we dealt with this problem by considering the lack of labeled images using Content-Based Image Retrieval (CBIR) approaches [12]. However, the existing semantic gap of what we represent and what we want to retrieve [13], [14] remained an issue. We noticed that conventional representations using global and local low-level image features (e.g., texture, color, shape) [15]–[17] were not enough. Figure 1 [11] illustrates an example of the problem of using conventional representations. Considering Figure 1(b) as a query image, Figure 1(d) seems to be the most similar. Nevertheless, only Figures 1(a) and 1(b) are from the same event, which means that their visual patterns could diverge from their semantic aspects.

To overcome this issue, we decomposed events into representative components, as a solution to improve the semantic representation in low dimensionality. Representative components intend to describe vital information to characterize the event such as people attending it (e.g., suspects or victims); objects that appear in the scene (e.g., cars, gun, backpack); and the place where the event unfolded (e.g., park, stadium, building). Nevertheless, the lack of minimal supervision to guide the combination and the proper weighing of the representative components limited the performance of the method.
Furthermore, when we analyze event-centric generated content on social networks, from different kinds of events – such as bombings, fires, shootings – the contribution of each component may vary. Learning this contribution is further complicated due to the lack of labeled training data.

In this work, we extend upon our previous work [11] and present a method for learning the contribution of different components that an event comprises. Our method relies upon a few-shot learning data-driven strategy that learns a manifold space where representative images are put together deviating from non-representative ones. An essential contribution of our methodology is the possibility of using previous knowledge obtained from other domains with larger datasets, only learning how to combine this knowledge. We explore three different loss functions using our proposed combination network for a representative image retrieval task.

Initially, to understand the behavior of our approach under different configurations, we study three datasets representing an event of general contexts such as “The Royal Wedding” (with over 32,000 images) and events of forensic contexts, such as the “Notre Dame Cathedral Fire” (with over 45,000 images). These datasets represent real-world scenarios in which the majority of images are non-representative to describe and understand the event of interest. Posteriorly, to validate the findings, we applied the chosen configuration to two other real-world events with a significantly smaller amount of training samples, “Museum” and “Bangladesh” (94 and 88, respectively), and we compare the results with state-of-the-art techniques for dealing with few training samples.

The experiments and results show: the learned manifolds provide more representative features in comparison with unsupervised methods [11] and even with features extracted from fine-tuned convolutional neural networks with the used datasets; when using the decomposition in representative components, combining the representations using shallower and wider networks is effective for both small and large datasets, and; the proposed approach provide competitive results with state-of-the-art techniques.

II. METHODOLOGY

Given an event \( \mathcal{E} \), which we want to represent and understand, and \( n \) available images, our task is to separate the ones related to the event (which we refer to as Representative Images) from non-related ones (which we refer to as Non-Representative Images).

Here we extend upon a prior work of ours [11] for this task. In that work, we decompose an event \( \mathcal{E} \) into semantic components related to \( \mathcal{E} \) – Places, Objects, People – and combined using some distance metrics in a semi-supervised way. In this work, we also approach the problem using the decomposition into components, but instead of manually combining the components, we rely upon a manifold learning technique that finds a latent space in which the components are dynamically weighted and combined.

A. Definitions

Formally, given the set of \( n \) images obtained for the event \( \mathcal{E} \), we need to define what we consider as Representative and Non-Representative images. Representative Images are the images that belong to the event \( \mathcal{E} \) and, to some degree, can help us to understand the event. Non-Representative Images, in turn, refer to the images that do not belong to the event \( \mathcal{E} \), but that could bear similarities to the Representative images in, at least, three different ways:

1) **Very Close Non-Representative Images (VC):** these images present close semantic similarities to the event such as depicting the same place or closely mimicking the dynamics of the event;
2) **Close Non-Representative Images (C):** these images present similar places and situations, which could be confused with the event;
3) **Far Non-Representative Images (F):** these images include more general topics with non-related event aspects.

B. Representing and Retrieving by Components

With the definition of representativeness in hand, we now need to consider the actual filtering process responsible for separating the groups. To perform this process, we followed a pipeline as composed in four stages: (a) Definition of Representative Components (guiding examples); (b) Feature Extraction; (c) Feature Combination Based on Components; and (d) Retrieval of Examples Based on Representativeness.

![Image 1](image1.png)

Figure 2 depicts this pipeline.

**Representative Components (RCs):** We may choose our representative components to reflect the main points, which could improve the event understanding – for instance, the place where the event happened; objects that appear in the scene; and people that could be involved. These components form a set represented by \( \mathcal{C} = \{c_k\}_{k=1}^m \) (Figure 2(a)).

**Feature Extraction:** After choosing the RCs, we need to represent each image with these components. The feature extraction could be performed with different techniques as long as they describe the RCs. Let \( \mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \) be a set of \( n \) images; we have \( m \) feature extractors \( f \) to represent the...
m RCs, and the $k$-th feature extractor obtains a representation $\mathcal{D}$-dimensional (in which the size of $\mathcal{D}$ depends on the adopted feature extractor) for each $\mathbf{I}_i$ image in the dataset: $f_k : \mathcal{X} \rightarrow \mathbb{R}^D$. Here, we propose the use of deep Convolutional Neural Networks (dCNNs) to perform the role of the feature extractors $f$ according to Figure 2(b).

Components Combination: After obtaining features for each RC, we seek to learn a manifold to properly combine them (Figure 2(c)). For this purpose, we propose a dense network, including two main tasks: Reduction and Combination. The Input layers of this architecture have variable size, according to the output of the CNN used as $k$-th feature extractor. The first task comprises a Reduction layer used to reduce the dimensionality of the features representing the $m$ components to the same vector length. Each component feature vector has its Reduction sub-layer with a size of $r$ neurons, which does not share weights with the other Reduction sub-layers. While the second task is the Combination. This part comprises layers $\mathcal{Y} = \{y_p\}_{p=1}^{n}$, with the first layer with size of $|y_1| = r/2$ neurons, reduced by half in each subsequent layer (i.e., $|y_p| = r/2^p$). The Combination part receives all reduced representations of the components to be combined (see Section II-C). The output is a unique vector that represents the images of an event more compactly and effectively. The training process is dependent on the designed and adopted loss function (see Section II-D).

Retrieval by Representativeness: To define the particular retrieval task we want to attack, we consider the set $\mathcal{X} = \{\mathbf{I}_i\}_{i=1}^{n}$ of $n$ images, which we need to classify as representative of the event or not; $\mathcal{Q} = \{\mathbf{q}_j\}_{j=1}^{l}$ the set of Representative images used as queries. To retrieve representative images from $\mathcal{X}$, we seek to obtain the least distant (most similar) images to a $\mathbf{q}_j$. To obtain these distances (dissimilarities) we need to define a descriptor function $h$ and a valid metric $\delta$ (e.g., Euclidean distance), such that $\delta(h(\mathbf{q}_j), h(\mathbf{I}_i)) = d_{ij}$ in which $d_{ij}$ represents a value of distance between the representations of $\mathbf{q}_j$ and $\mathbf{I}_i$. A final ranking $\mathcal{R}_{\mathbf{q}_j}$ is a permutation of the images in $\mathcal{X}$ in which the first position contains the least distant (most similar) image for $\mathbf{q}_j$, and so forth until all images are ordered according to their distances (dissimilarities) for $\mathbf{q}_j$ (Figure 2(d)). Here, the descriptor function $h$ is obtained by the Components Combination step (Figure 2(c)).

C. Manifold Learning

Our proposed methodology relies upon learning a manifold to combine features from different RCs dynamically. This process can be carried out in two different ways: using a Standalone process of features combination, or using a Joint optimization process of features extraction and combination. Both approaches have strengths and weaknesses, which will be discussed.

On the one hand, when dealing with a Standalone process, we extract features (Figure 2(b)) using transfer learning of CNNs pre-trained with computer vision tasks (often with thousands or millions of training examples) and, posteriorly we use these feature vectors to train (independently) the components combination network branch (Figure 2(c)) to learn the manifold space. The main advantage of this approach is that we need less training data to train the combination network branch and to finetune the overall architecture, as we deal with a smaller number of network parameters. Moreover, we could choose networks that were trained with large datasets as long as they reasonably describe the adopted components. In turn, the main disadvantage is the lack of specialization of the feature extractors to the events of interest.

On the other hand, when dealing with a Joint optimization process, we can train the feature extractors (CNNs) (Figure 2(b)) along with the components combination subnetwork (Figure 2(c)). In this way, the process of feature learning
and component definition is unified. The main advantage of this approach is the specialization of the feature extractors to the events of interest. In this case, the aim is to extract what best represents the RCs for each event. However, in this approach, there is a significantly bigger number of parameters to train and adjust, consequently needing to increase the number of training images.

As the lack of labeled images is an intrinsic characteristic of our problem, we opted by the Standalone approach, using the transfer leaning of pre-trained CNNs as feature extractors. In this context, we also need to define the sizes of Reduction and Combination parts of the dense network considering just a few training samples for each event (a few-shot learning strategy).

D. Loss Function

We propose to learn the manifold in two different ways, either by classification learning or by distance learning. For classification learning, we used a cross-entropy loss function. For distance learning, we used a contrastive and a triplet-loss function.

Cross-Entropy: The first loss function performs the combination presented in Figure 2(c) as a two-class classifier according to the Figure 3: Representative and Non-Representative. In this case, we adopted the cross-entropy loss function. After training, we obtained the representation from the last Combination layer. We named the learned space as Cross-Entropy space. We trained the network for 50 epochs using Adam optimizer [18] with a learning rate $\alpha = 0.00001$ and a decreasing rate of $1e^{-5}$.

The rationale for not using shared weights is that we first want to reduce each component representation so that possible cross-effects among different components are minimized. A shared weight representation here is straightforward, but we opted first to reduce and only then combine the reduced representations.

Contrastive: The second loss function we adopt is contrastive loss [19]. This loss function uses a siamese network comprising the base model (Figure 2(c)) in each branch, followed by a Euclidean distance operation as presented in Figure 4. This approach compares two image representations. For this reason, we use two feature vectors as input, one representing the image $I_a$ and another representing the image $I_b$. After training, we obtained the representation from the last Combination layer. We named this learned space as Contrastive space. We adopted the Adam optimizer [18] for training during 50 epochs with a learning rate $\alpha = 0.00001$ and decreasing rate $1e^{-5}$, using a margin equal to 1 for the loss.

For training the networks, we adopt batches of 128 pairs. Two images from the same class define a positive pair, while images from different classes denote a negative pair. For each image in the training subset, we generate ten positive and ten negative pairs.

Triplet Loss: The third loss function we adopt consists of a triplet loss formulation using the base model (Figure 2) and a triplet loss layer at the end [20]. This model relies upon three branches for training, each one with one group of features. The first group represents an image $I_a$ (the anchor image for the triplet); the second one, $I_p$, denotes the positive image for the anchor; and the third, $I_n$, denotes the negative image for the anchor. After training, we obtained the representation from the last Combination layer. We named this learned space as Triplet space. We used Adam optimizer for training during 50 epochs with a learning rate $\alpha = 0.01$ and a decreasing rate of $1e^{-5}$, using a margin equal to 1 for the loss.

We consider training batches of 128 triplets created considering an anchor, a positive sample from the same class, and a negative sample from the different class. The computer vision literature suggests combining ordinary triplets with hard ones. An ordinary triplet is one drawn randomly according to the formation just described. A hard triple consists of the one comprising positive and negative examples that are too close in the representation space. In our work, we consider 64 randomly selected triplets and 64 hard triplets. The hard triplets are selected each time from a pool of 512 random triplets and sorted according to their level of confusion in the feature space.
Fig. 5. Network for learning distance metrics using the triplet-loss formulation. To train the network, we use three branches with sharing weights for the features ($x_n$, $x_p$ and $x_a$) of images $I_n$, $I_p$ and $I_a$. The objective here is to reduce the final distance between features from images $I_p$ and $I_n$ and increase it for features of the images $I_n$ and $I_a$. The Input layer receives feature vectors with different dimensionality according to the CNNs used to describe the RCs; the Reduction layer reduces the features independently (without sharing weights among components vectors); Combination layers combine the features for a final representation. After the last Combination layer, we have a Triplet Loss layer to compute the loss value.

III. EXPERIMENTAL SETUP

To validate the proposed methods, we have performed experiments considering three datasets (For Ablation Analysis). In order to compare the best configurations of our methods with state-of-the-art methods, we have performed a final experiment, in which we consider the three initial datasets and two new ones (referred to as Not Analysed Before – NAB). We have divided each dataset into training, validation, and test subsets. In this section, we present the datasets, adopted feature extractors, the training protocol for learning the manifold spaces, the methods in the prior art used to the comparison, and the evaluation metrics.

A. Data Description

One of the main objectives of the proposed approach is to be applicable in real-world events. For this purpose, the literature lacked datasets that could reflect the difficulty of separating non-representative images with different closeness to the event. For example, when retrieving images from Notre-Dame Cathedral Fire (an event which will be described in the sequence), we may have images before the fire (very similar to the representative images), images from cartoons as the Hunchback of Notre Dame (which could be easier to separate but yet with a certain level of similarity), images from other gothic cathedrals (which present structural similarities) and, if we expand too much the retrieval, images that could not be related at all.

To evaluate the approaches in datasets as realistic as possible, we adopted the datasets proposed by Rodrigues et al. [11]: Wedding, Fire, and Bombing. Due to the difficulties of labeling images, these datasets were constructed following a guided manual approach. It was performed a search for videos with different levels of proximity to the event: from the event, very close to the event (VC), close to the event (C) and, far from the event (F). By doing this, we could easily watch the videos, analyze if they belong to the category, and extract the frames. After this process, it was necessary to clean some “class-dislocated” frames, for example, the journalist presenting the news in a video in which the event was also presented. This cleaning was manually performed.

The criteria used for including a video to a specific category (event, VC, C, and F) for each of the three datasets were:

- **Representative**: videos from the event and/or moments immediately before or after which could be considered to explaining the event;
- **Non-Representative / Very Close (VC)**: presenting the same place of the event and/or the same people;
- **Non-Representative / Close (C)**: presenting similar place and/or activities;
- **Non-Representative / Far (F)**: non-related videos that could not be placed in other categories.

After an ablation study with different possible choices of our methods, we compare our methods with the ones in the prior art considering the three datasets above and two new ones: Museum and Bangladesh. For these datasets, the process of collecting and labeling images was different, aiming to validate the proposed methods in different conditions. Images were collected during a week after the events using the open-source framework presented by Schinas et al. [21], which performs a monitoring and collection of data from multiple social media platforms. Only a small fraction of these datasets were manually labeled as representative or non-representative (without the subcategories CV, C and F). A short description of each of the five events is presented below.

1) **Wedding**: images from the Royal Wedding of Prince William and Catherine Middleton. The event happened on April 29th, 2011 at the Westminster Abbey in London, United Kingdom.

2) **Fire**: images from the Notre Dame Cathedral Fire in Paris. The fire broke out on April 15th, 2019, and the cathedral had its spire, and most of the roof destroyed.

3) **Bombing**: images from the detonation of two homemade pressure-cooker bombs in the Boston Marathon, which happened on April 15th, 2013. The explosions resulted in three casualties and approximately 264 injured.

4) **Museum**: images from the fire which destroyed part of the collection of the National Brazilian Museum in Rio de Janeiro, which happened on September 2nd, 2018.

5) **Bangladesh**: images from the fire in a densely populated district of Dhaka, Bangladesh, which happened on February 20th, 2019. The explosions resulted in 80 casualties and approximately 50 injured.

Table I presents the breakdown for the five adopted datasets into training, validation and test sets. Subsequently, Table II presents the number of images from the Representative and Non-Representative classes for the initial datasets Wedding, Fire and Bombing, focusing on the subclasses of Non-Representative images: Very Close (CV), Close (C), and Far (F). The number of test images in each subclass can be consulted in Table S1 (Supplementary Material).

In forensic scenarios, it is common to have access to only a limited set of images for a given event to start an investigation. Therefore, augmentation techniques play a crucial role in further expanding the capabilities of deep learning algorithms in these setups. In this paper, we have considered geometric
TABLE I
Wedding, Fire, Bombing, Museum and Bangladesh Breakdown into Training (Train.), Validation (Val.) and Test (Test) Sets.
For the Museum and Bangladesh We Also Count on 490 and 759 Unlabeled Images Respectively, for Further Explorations with Unsupervised Techniques for Those Interested

| Dataset       | Representative | Non-Representative |
|---------------|----------------|--------------------|
|               | Train. | Val. | Test | Train. | Val. | Test |
| Wedding       | 280    | 75   | 84   | 19,055 | 6,101 | 6,438 |
| Fire          | 507    | 76   | 390  | 28,222 | 7,465 | 8,420 |
| Bombing       | 2,153  | 709  | 320  | 32,936 | 5,858 | 6,430 |
| Museum        | 39     | 19   | 19   | 55      | 27    | 27    |
| Bangladesh    | 35     | 17   | 17   | 53      | 26    | 26    |

TABLE II
Wedding, Fire and Bombing Datasets Breakdown, Examples of Representative and the Non-Representative Images (Including the Subclasses Very Close (VC), Close (C) and Far (F) of the Latter One)

| Dataset       | Representative | Non-Representative |
|---------------|----------------|--------------------|
|               | VC   | C   | F   | Total |
| Wedding       | 439  | 2,272 | 16,165 | 13,157 | 31,594 |
| Fire          | 973  | 182  | 20,619 | 23,306 | 44,107 |
| Bombing       | 3,182 | 1,669 | 14,430 | 29,125 | 45,224 |

TABLE III
Final Number of Augmented Images of Each Dataset – Wedding, Fire, Bombing, Museum and Bangladesh – Inside Training (Train.) and Validation (Val.). the Test Set Remains the Same – Without Augmentation – As Table I Shows

| Dataset       | Representative | Non-Representative |
|---------------|----------------|--------------------|
|               | Train. | Val. | Train. | Val. |
| Wedding       | 4,426  | 1,188 | 19,055 | 6,101 |
| Fire          | 8,019  | 1,204 | 28,222 | 7,465 |
| Bombing       | 34,080 | 11,231 | 32,936 | 5,858 |
| Museum        | 624    | 304  | 880   | 432  |
| Bangladesh    | 560    | 272  | 848   | 416  |

augmentation techniques to improve the combination learning process (manifold learning stage). The used transformations are: crop, rotation, and zooming. We performed transformations before the training process. Five random crops were performed in each image, maintaining half of its original size. Five random rotations between $-30^\circ$ and $30^\circ$ were applied to each image. Finally, five random zooming transformations in the range of $1 \times$ to $5 \times$ were performed on each image. If the same transformation with the same parameters to an image is chosen, one of the augmentations is discarded to avoid repetitions. Table III presents the number of images in each subset of the dataset after augmentation. The augmentation was performed only in the training and validation sets.

B. Feature Extractors

For feature extraction, we chose three components to represent each events: Places, Objects, and People. As a first image characterization stage, we used Convolutional Neural Networks (CNNs) to extract features related to each of the components using the outputs of the layer before Softmax in each adopted network. For the Places component, we applied VGG16 [22] trained on the Places dataset [23] to extract features ($x_{places}$) obtaining an 1536-dimensional feature vector. For the Objects component, we used Inception-ResNet [24] trained on the Imagenet dataset [25] to obtain features ($x_{objects}$) generating a 4096-dimensional feature vector. Finally, for the People component, we applied a Re-Identification network named PCB [26] trained on the Market-1501 dataset [27] to describe features ($x_{people}$) obtaining a 12288-dimensional feature vector.

C. Learned Spaces

For initial comparisons, we adopted three baselines: final feature vectors extracted directly from the three pre-trained adopted CNNs [11] (Concatenated); Event Semantic Space features (ESS) as proposed by [11]; and final vectors extracted from the three adopted CNNs but finetuned for each used dataset. We refer to the last case as (Fine-Tuned). The CNNs finetuning were carried out for 50 epochs considering the Representative and Non-Representative image classes. We used the Adam optimizer [18] with $a = 0.00001$ and a decreasing rate of $1e^{-5}$ in the finetuning process.

We called our three combination networks as Cross-Entropy, Contrastive and Triplet, according to the loss function used in the training process. We divided the six approaches (three proposed + three baselines) into two groups: classification learning and distance learning. The classification learning methods used a classification network in some part of the features learning process and included Concatenated, ESS, Fine-Tuned, and Cross-Entropy. The distance learning approaches, although using a CNN to extract the original features, learned the manifold space using a comparison approach, and include Contrastive and Triplet.

D. Few-Shot Learning Method

Few-shot learning methods, as the ones we propose, intend to acquire the capability of learning new classes based on few labeled examples, which could be interesting in our real-world events application. A few-shot problem can be defined as a C-way K-shot, if we want to learn C classes with only K available labelled samples for each of the classes.

We adopted the method Relation Net proposed by Sung et al. [28] for comparisons with our approach. The authors presented a network comprising an embedding module to extract the samples representations and a relation module to compare each of the labeled samples representations with an unlabeled sample.

The Relation NET was trained in the miniImageNet [29] dataset composed by 100 classes with 600 examples each one, which we used as a pre-trained few-shot network to one of our experiments. For the miniImageNet task, from the total of 100 classes, the Relation NET was trained with 64 known training classes and 16 known validation classes (for the meta-learning process). After this, the remaining 20 classes,
E. Semi-Supervised Methods

Another set of methods we adopt for comparisons is semi-supervised with label propagation. These methods aims to learn a function $f$ from a small set of labeled data and further propagate the labels over the unlabeled set. Typically, we are given the set $X = \{x_i\}_{i=1}^n$ of $n$ images, and a function $h$ to represent each image as a feature vector $x_i \in \mathbb{R}^N$. If we consider the first $m$ points as labeled by $y_i \leq m = \{0, 1\}$ determining non-representative and representative images respectively, the intention is to propagate these labels to the remaining samples $x_u(m + 1 \leq u \leq n)$ which are considered unlabeled samples $y_i \geq m+1 = -1$. The feature vectors used are the concatenation of the extracted features $\mathbf{x}_{\text{places}}$, $\mathbf{x}_{\text{objects}}$ and $\mathbf{x}_{\text{people}}$.

For label propagation, the algorithms need an established neighborhood relationship. A pairwise relationship $x_{ij}$ could be defined as an edge of the a graph $G = (V, E)$ linking two nodes $i$ and $j$, where the nodes in $V$ represent the feature vectors for the images in $X$, and the edges in $E$ are weighted by the values of an the obtained by a kNN algorithm.

We selected two methods for propagation: Confidence Aware Modulated Label Propagation (CAMLp) proposed by Yamaguchi et al. [30] and Modified Adsorption (MAD) proposed by Talukdar and Crammer [31], which differ in the way they propagate the signals.

The CAMLP algorithm takes considers the prior beliefs (the initial labels) and the information propagated from its neighbors during the prediction process.

In turn, the MAD algorithm is based on the random-walk interpretation of the Adsorption algorithm [32]. When the walk reaches a node $v$, the next possible action depends on the probability to inject ($p^{\text{init}}_v$), which determines the chances of stopping the walk and returning the obtained labels until that moment; the probability to continue random walk $p^{\text{cont}}_v$, which indicates the chances of continuing propagating labels to neighborhoods; and the probability to abandon the walking and the labeling process $p^{\text{abnd}}_v$, returning the labels to the original values.

F. Generative-Based Method

Finally, another method we consider for comparisons relies on a semi-supervised learning generative adversarial networks (sGAN) proposed by Salimans et al. 2016 [33]. The main idea is to exploit GAN generators’ samples to boost the image classification performance by improving generalization. The network is trained playing the roles of an image classifier and a discriminator trained to distinguish samples produced by a generator. In our case, we extended that idea to differentiative representative and non-representative images for a real event.

G. Evaluation Metrics

We adopted different metrics for rankings evaluation purposes: Precision × Recall (and their harmonic mean, F1), Mean Average Precision (MAP), Mean Distance and Distance Variance. Consider $R^p = \{rp_i\}_{i=1}^V$ the set of Representative images in the dataset; $NR^p = \{np_i\}_{i=1}^V$ the set of Non-Representative images; $Re = \{re_u\}_{u=1}^V$ the set of retrieved images, in other words, images considered as representative; and $NR^e e = \{r^p \cup NR^p\} - Re$ the set of images considered non-representative.

Precision × Recall is calculated by obtaining the ranking Precision values — given by Equation 1 — for the set of Recall values $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ – given by Equation 2. We generated graphs based on the obtained values.

$$P = \frac{|R^p \cap Re|}{|Re|}$$

$$R = \frac{|R^p \cap Re|}{|R^p|}$$

The MAP metric is the mean of the Average Precision (AP) values – given by Equation 3 – obtained by considering all the rankings constructed by the query images ($Q$).

$$AP = \frac{1}{V} \sum_{u=1}^{V} (R_{u+1} - R_u) P_{u+1}$$

The Mean Distance and Distance Variance are the mean and variance of all the distances ($d_{ij}$) of images ($i$) from the same group — Representative or Non-Representative — in a ranking, considering a specific query ($q_i$).

More specifically, for the classification task, we used the Balanced Accuracy metric presented in Equation 4 and F1-Measure presented in Equation 5, in which $P$ and $R$ are the Precision and Recall (Equations 1 and 2), respectively.

$$B\text{Acc} = \frac{|R^p \cap Re| + |NR^p \cap NR^e|}{|R^p| + |NR^p|}$$

$$F1 = 2 \times \frac{P \times R}{P + R}$$

IV. RESULTS AND DISCUSSION

We have performed eight experiments to evaluate the proposed methods (Experiments I-VII) and to validate them through a comparison with methods in the prior art (Experiment VIII). The first five experiments represent ablation studies of the different options of the proposed methods and analyze their capability of creating good representations. In this vein, we generated rankings using the training subset as queries to retrieve similar images from the test set. Precision results are reported using the mean precision of the rankings. For the last three experiments, we analyzed the capability of using the generated representations in classification tasks. Balanced Accuracy, precision and recall values are reported. The last experiment (Experiment VIII) also include the two Not Analyzed Before datasets for validating the capability of generalization to unseen events.

A. Experiment I: Exploring Network Depth and Width

Our first experiment explores the network architecture depth considering four networks with different depth and width. As Figure 2 depicts, the proposed network backbone comprises
a dense layer for feature reduction for each feature vector of the input and a combination dense layer. The networks were named $N(512, 128)$, $N(512, 128, 64)$, $N(1024, 512)$ and $N(1024, 512, 128)$, in which the first value (within parenthesis) denotes the number of neurons in the Reduction layer and the other values denote the number of neurons in the Combination layer(s).

Network $N(512, 128)$ presents 512 neurons in each of the dense layers used for feature reduction and 128 neurons in the combination dense layer. Network $N(512, 128, 64)$ follows the initial configuration of $N(512, 128)$ but increases the number of layers by adding a layer with 64 neurons after the combination layer. Network $N(1024, 512)$ explores network width. This network has 1024 neurons in the dense layer used for feature reduction (instead of 512 from $N(512, 128)$) and 512 neurons in the combination layer (rather than 128 from $N(512, 128)$). Finally, network $N(1024, 512, 128)$ increases the network in depth and width by using the $N(1024, 512)$ as the basis and adding a dense layer with 128 neurons from where the features are extracted. This last network has 1024 neurons in the reduction layer, 512 neurons in the combination layer, and one extra layer of 128 neurons.

The experiment with the four architectures was performed using the features extracted from the layer before the output layer providing 128 features for the network $N(512, 128)$, 64 for the network $N(512, 128, 64)$, 512 features for the network $N(1024, 512)$ and 128 features for the network $N(1024, 512, 128)$. The architectures were trained using the three different losses adopted in this work, Cross-Entropy, Contrastive and Triplet, for the three datasets. As we aim at retrieving representative images (those of the event) in the test set based on the learned representation, we used a set of relevant images (the representative training set) to perform as queries. The results were evaluated using precision $\times$ recall points (10%, 20%, …, 90%, 100%).

Figure 6 presents results of for the methods trained with triplet loss without augmentation. Please refer to Figures S4-S6 in the Supplementary Material for cross-entropy, contrastive and triplet (with augmentation) loss results. Networks $N(512, 128)$ and $N(1024, 512)$ present the best precision rates in datasets with less training images as Wedding dataset, and also for the training sets without augmentation (first column of the figures). The networks trained with cross-entropy loss presented consistency in relation to the best performances of $N(512, 128)$ and $N(1024, 512)$, even for augmented datasets.

As we presented in Section III-A the size of the datasets are considerably different. The biggest dataset, Bombing, presents about three times more Representative images than Fire dataset and, about six times more representative images than Wedding dataset. As previously observed [34], deeper networks could present better results as long as a sufficient set of training images is provided. This fact leads us to believe that, for small training sets as in Fire and Wedding, increasing depth would be negative for learning data distribution, which is reflected by the results in Figures 6, S4 and S5 without augmentation. As Bombing is the biggest dataset, the mentioned difficulty of deeper networks with small training sets, does not have the same impact.

When we consider augmented data for training, we provide more variance to the network and this tends to regularize the training process, providing a way to overcome the challenge of training with small sets. The results presented in Figure S4, S5 and S6 with the augmented data confirm that more data, even obtained through augmentation, can be beneficial. We also performed experiments with and without augmented training sets for the classification task (Section IV-F) using the proposed the Contrastive and Triplet representations, and the results corroborate the benefits of using augmented data (see Figure S13 in the Supplementary Material).

Despite good representations learned by deeper networks with augmented data, the network $N(1024, 512)$ seems to perform better for small numbers of images (as the Wedding dataset, for example). Using this network for bigger datasets (as Bombing) showed that, increasing depth presented minor
improvements, evincing that shallower architectures could be sufficient also for bigger amounts of data.

To generalize the approach to other real-world problems, we took into consideration the uncertainty in the problem nature: in a real forensic application, it is hard to know how many images will be available, and in short periods of time after an event, probably the number will be relatively small, even after performing augmentation. Therefore, we chose to use shallower networks. However, our datasets could yet be relatively representative in comparison to real problems. Therefore, we need to understand if we reduce the number of labeled images (for training the best shallower network \((N(1024, 512))\), we will be able to obtain good representations or if we will need to reduce also the width.

To understand this point, we proposed Experiment II in which we compare \(N(512, 128)\) and \(N(1024, 512)\) (the shallower and tighter network and, the apparently best shallower network) using reduced training subsets.

### B. Experiment II: Varying Training Sets

This experiment was performed using augmented data of randomly chosen original representative images for different training set sizes. The number of original representative training images for the datasets were 10, 20, 50, 100, 200. The non-representative images were randomly selected in the same number of the final representative set (original + augmented images). The selection of training images determines the quality of the manifold space – e.g., if we chose to train with ten representative images too similar to one another, probably the manifold will not be able to represent the complexity of the event (not enough diversity). For this reason, we selected ten different training sets for each training set size and reported the mean results of them.

We generated the representative sets in an incremental way. For example, we randomly selected 10 images for 10 sets (10-size sets) from the set of non-augmented images (augmentation was included after the composition of training sets). To create the 20-size sets, we added 10 different images to the each initial 10-size set. We continued the process for the 50, 100, and 200-size sets. Using these selected representative training images with augmentation, and including the same number of non-representative, we trained a network for each set size to observe the behavior of changing the training images and present average results for each set size.

We trained the two shallower architectures (determined in the previous experiment) using the Cross-Entropy, Contrastive and Triplet losses. We compared the results of the different training set sizes with the ESS \([11]\) representation by retrieving the test images using the positive training images as queries. To generate the ESS representation, we used the same images from the different sizes of training sets as Event Representative Images (ERIs). We used the mean of the ten Mean Average Precisions (MAPs) of the generated rankings for each of the five sizes of training sets to measure the quality of the rankings. To the remaining experiments, we used only the best network found in this experiment.

Figure 7 shows results when training networks with small quantities of original representative images using triplet loss.

Please refer to Figures S7 and S8 in the Supplementary Material for cross-entropy and contrastive loss results. In this case, we compare \(N(512, 128)\) and \(N(1024, 512)\) to the semi-supervised method ESS \([11]\). Even with reduced quantities of training images, the components combination \(N(512, 128)\) and \(N(1024, 512)\) produced the best MAP results, resulting in an improvement of more than 10 percentage points in the smallest set (with only 10 representative images). We noticed that the wider network \((N(1024, 512))\) presented higher values of Mean Average Precision (MAP). In view of these results, we continue the experiments adopting the \(N(1024, 512)\) network as our standard architecture.

### C. Experiment III: Visualizing Classes Separation

In this experiment, we explore the feature discriminability for different feature spaces considering the setups with and without data augmentation. The representations of ESS, Fine-Tuned, Cross-Entropy, Contrastive and Triplet were generated.
using all positive training images (Tables I and III) and the same amount of negative training images.

For visualization, we project all feature vectors onto a two-dimensional space by using the dimensionality reduction technique Uniform Manifold Approximation and Projection (UMAP) [35]. This technique aims at establishing local structures preserving the global data structure. UMAP is similar to the t-SNE technique [36] but proposes a more general application for dimensionality reduction besides visualization. The number of neighbors selected for UMAP projection was 25, and the distance between samples was set to 0.01. As we are ranking the retrieved examples by Euclidean distance, the distance metric chosen was also Euclidean. The data was reduced to three dimensions. The first and second dimensions were used as $x$ and $y$ axes of the scatter plot while the third dimension was used to order the samples for plotting purposes. This ordering means that the most significant value of the third dimension is plotted first (away from the observer), and the smaller value is plotted lastly (closer to the observer).

The test images were also projected using UMAP (with the same mentioned parameters) with four-classes discrimination – Representative, Non-Representative Very Close (VC), Non-Representative Close (C) and Non-Representative Far (F).

As we chose the combination network $N(1024, 512)$, we intended to analyze how the images were projected onto a 2-dimensional space, verifying if the combination provides better separation between classes than the baseline approaches. Figure 8 presents the projection of all images of the datasets onto the learned spaces when adopting data augmentation techniques, using the Representative and Non-Representative classes. Please refer to Figure S9 the Supplementary Material for results without data augmentation techniques. We notice a similar behavior in the classification approaches — Concatenated, Fine-Tuned and Cross-Entropy — and even for ESS, presenting more dispersion of images from both classes. On the other hand, the distance-learning approaches — Contrastive and Triplet — present much better separations, tightly grouping images from each class. The augmentation techniques proved to reduce the mixture between classes, especially for the distance-learning approaches. The Bombing dataset seems to be the easiest one to separate when considering the 2-dimensional spaces.

In Figure 8, if we consider the different non-purple colors as the levels VC, C and F, we could perform an analysis of the different levels of non-representativeness with and without data augmentation. The manifolds constructed by Contrastive and Triplet approaches apparently better sep-
arate the Non-Representative images from the Representative ones. Please refer to Figure S9 in the Supplementary Material for additional results with different levels of non-representativeness. These two classes represent the semantics observed in the images, and we expected them to be more distant in the projection space.

Based on the visualizations, we could observe that possibly the distance-learning approaches are able to better separate Representative from Non-Representative images, especially when using the augmented data for training. To verify the performance in the retrieval task, we move now to the fourth experiment, by retrieving Representative images based on the different methods we assess in this paper.

**D. Experiment IV: Retrieving Representative Images**

We performed experiments with and without augmentation for training. We used all training representative images as queries to the retrieving method. The Euclidean distance of each test image to the query was calculated, and the distances were used to order the most similar images to the query. The precision of the generated rankings was evaluated per recall points (10%, 20%, …, 90%, 100%), and the mean and deviation values of precision were obtained over the rankings. We generate the Precision × Recall curve based on the mean and deviation values.

We also calculated the mean and variance values for each class — Representative and Non-Representative — based on the distances of the representative and non-representative images from the queries.

Figures 10 and 11 present the Precision × Recall curves for the different methods without and with data augmentation respectively. All three learned spaces (Cross-Entropy, Contrastive and Triplet) outperformed the simple concatenation approaches in all datasets. The proposed techniques also outperform the Fine-Tuned space (i.e., fine-tuned methods) and the Concatenated ones. Note that the distance-learning
approaches (Contrastive and Triplet) improved the retrieval task and increased the precision in more than 20% percentage points, for all datasets.

The results of the rankings are also supported by the mean distance of the images from Representative and Non-Representative classes. Figure 9 presents the mean and variance of the image distances to queries in the Wedding, Fire and Bombing datasets when adopting data augmentation techniques. Please refer to Figure S10 in the Supplementary Material for results without data augmentation methods. In this experiment, we expect Representative images to be closer to the queries (lower mean). For this reason, a good result here would show Representative images with lower mean and variance. As expected, the three proposed methods (Cross-Entropy, Contrastive and Triplet), show a lower mean distance for Representative images (with lower variance) than the mean distances for the Non-Representative images. The obtained results indicate that learning a proper combination manifold allows a better separation of representative and non-representative images of an event.

E. Experiment V: Ranking Visual Quality

This experiment aims at qualitatively comparing the top retrieved examples (top@5) of the rankings generated by the different feature spaces to analyze their quality.

For this qualitative analysis, we observed the obtained rankings presented in Figure 12 for the Bombing dataset. The first row denotes the query. It is followed by the retrieved images using Concatenated, ESS, Fine-Tuned, Cross-Entropy, Contrastive and Triplet in each column (top@5), respectively. The purple squares indicate the representative images while the green squares denote the non-representative ones. Note how the manifolds learned by the component combination (proposed methods) provide more discriminative features, improving diversity at top positions. Contrastive and Triplet present the best rankings in this case. Please refer to Figures S11 and S12 in the Supplementary Material for results in the Wedding and Fire datasets.

F. Experiments VI and VII: Data Augmentation and Reduced Training Sizes

The previous experiment results presented the basis for choosing and analyzing the configuration of our networks. Now we aim to compare the results of using the best obtained representations for a classification task observing the impact of data augmentation and variation in training size. We chose to use the representations generated by the combination networks with Contrastive and Triplet losses as they outperformed the other representations in the retrieval task.

We selected two simple classifiers for this experiment: a Multilayer Perceptron Classifier (MLP) [37] and a Support Vector Machine Classifier (SVC) [38]. For the MLP Classifier, we used one hidden layer with 64 neurons, tanh as activation function and adaptive learning rate with initial learning rate equal to 0.001. We trained until convergence or reaching 1,000 iterations. For SVC, we used the RBF kernel with parameter $C = 1.0$. For both methods, we used the same training sets used to train the manifolds previously presented.

Sections I-G and I-H in the Supplementary Material present results of Balanced Accuracy and F1-Measure for the
Fig. 13. *Wedding, Fire* and *Bombing*) results using the metrics Balanced Accuracy and F1-Measure. The proposed methods (green and blue) outperform the few-shot method, presenting similar Balanced Accuracy to semi-supervised methods, but superior F1-measure. sGAN presented good F1-Measure for the dataset (*Bombing*), which has more training samples.

augmentation analysis and training size analysis. According to these results, also during classification, the augmented training sets outperform the non-augmented ones. Moreover, even with a reduced training size for classification, the methods are able to maintain the levels of Balanced Accuracy with F1-Measure improving as training size increases.

Therefore, for both retrieval (Sections IV-A, IV-C and IV-D) and classification, augmentation is critical when few training samples are available. For this reason, the final experiments include data augmentation for training.

**G. Experiment VIII: Comparing Different Methods**

As a final experiment, we compared the performance of the few-shot learning, semi-supervised, and generated-based methods with our methods (the best losses: *Contrastive* and *Triplet*) for the classification task using MLP and SVM Classifiers. All the methods used the complete training set and we evaluated the Balanced Accuracy, Precision and Recall for the same testing set. This experiment considers all five datasets adopted in this work. As we present a retrieval analysis in the first five experiments for the three initial datasets, we present in Figure S17 (Supplementary Materials) the Precision × Recall curves of the new datasets. We use only the chosen methods (*Contrastive* and *Triplet*) to obtain the features for representing samples.

**Few-shot learning method**: As we deal with a binary classification, we define it as a two-way five-shot learning problem (following the configuration of 5-shot proposed in the original article [28]). We used the proposed *Relation NET*
pre-trained in the miniImageNet dataset (we called it Few-Shot - RelationNet Cross). This dataset presents a different domain of our event datasets, therefore, we also trained the networks during 500,000 episodes (different set of labeled 5 images) — for the meta-learning process — with all the available training images according to each dataset (we called it Few-Shot - RelationNet Trained). As we have of more than 5 labeled images per class, we performed the learning process by 100 iterations and obtained the average of the metric values.

**Semi-supervised methods:** We chose the value $k = 32$ for the kNN algorithm (through grid search) to construct the used affinity matrix. We reduced the dimensionality of the feature vectors using PCA (pre-processing step) to 128 elements and iterated each method 100 times to find the optimal solution for the label propagation.

**Generative-based method:** For the sGAN, each image is divided into three channels, R, G, B, using a resolution of $256 \times 256$. The results with sGANs tend to improve when they have many days of training (more than 50 thousand iterations, resulting in almost one week per dataset). For comparisons in this work, we adopted up to one day of training for the sGANs (approximately 1,000 iterations) for a fair comparison as more than this does not apply directly in a forensic setup in which we need fast response time. In comparison, our techniques, for augmented datasets as Museum and Bangladesh, train in less than 20 minutes.

Figure 13 presents the comparison for the datasets for ablation analysis: Wedding, Fire and Bombing. Few-Shot - RelationNet Cross presented the inferior results, which was expected due to the differences between the datasets domain. When training with images from the event (Few-Shot - RelationNet Trained), Balanced Accuracy and F1-Measure slightly increased. This could indicate that, if we have events from a specific domain, for example, a diversity of marathons, and we receive other two (or more) marathons to differentiate, the method may perform better. Unfortunately, in forensic contexts, we cannot guarantee to have data from similar events available.

One potentially powerful method also considered was the sGAN, which generates, during training, even more data on its way to augment training set. The results obtained for this method showed the best F1-measure for the biggest dataset Bombing, which could indicate the need of more data in order to improve the other datasets. If the method is trained for many more iterations, it can benefit from more generated images (potentially thousands) and, consequently, present improved classification. Nevertheless, in scenarios with limited response time as the forensics one, it would be impractical to train it for the required time.

For this three datasets, the best results were obtained by our proposed methods and the semi-supervised ones (CAMLIP and MAD). Although, even using simple classifiers as the MLP and SVM, our learned representations were capable to
outperform semi-supervised methods in Balanced Accuracy of the datasets Fire and Bombing, and present similar results for the Fire dataset. More important, all the F1-Measure results (indicating the capability of recognize the positive class – Representative images) presented considerable superiority with respect to the values presented by semi-supervised methods.

Finally, we present in Figure 14 the same compared methods, now for evaluating datasets: Museum and Bangladesh. According to the previous analysis, the best methods were the proposed and the semi-supervised ones. In the Museum dataset, MAD and CAMLP presented the best results. In the Bangladesh dataset, our proposed methods were superior, with emphasis for the Triplet representation. For further reference, we present the same results (Figures 13 and 14) in Table S2 in the Supplementary Materials.

We highlight that these are smaller datasets (94 and 88 training images) that have not been subjected to the previous set of experiments to validate the architecture choice without having to adapt the approach to each arriving dataset. Based on the results of this experiment, we noticed that through this shallower and wider network, we obtained a manifold in which representative images are easily separated from non-representative ones.

In Supplementary Material we present examples of rankings generated using the Contrastive and Triplet representation. We intend to retrieve the most similar images to a query (Figure S18) from the unlabeled test of Museum (Figure S19) and Bangladesh (Figure S20) datasets.

V. CONCLUSION AND FUTURE WORK

In this paper, we dealt with the challenging problem of understanding an event and its encompassing pieces from large collections of images available in social platforms.

In our previous work [11], the decomposition of events in representative components – Places, Objects and People – produced a low-dimensional representation to retrieve representative images. The results were competitive with the baseline approaches. However, the contribution of each component for the different kinds of events was not deeply analyzed mainly because of the lack of a minimal supervision.

The justification of using unsupervised methods was the difficulty of obtaining labeled data describing event representative images, especially in short periods. It is a known fact that deep neural networks require massive amounts of labeled data to be correctly trained [1]. Nevertheless, as we noticed in the already cited work [11], the lack of training restricts the performance of event representation.

Aiming to overcome this issue, in this work, we relied upon a limited number of images and leveraged some augmentation techniques to learn how to better generate these representations. The results of ranking using an Fine-tuned representation — the attempt to finetune the CNNs using the target datasets — evince the problem of small training sets and yield poor precision rates (Figure 10) even when considering augmented data (Figure 11). On the other hand, the proposed methods achieved high precision rates with the available small training sets. The key change for this high precision in the proposed method lies in the correct selection of backbone CNNs used for feature extraction, the proper combination of such features through a manifold learning strategy allied with loss functions aiming at better discriminating examples from different classes.

The analysis of classification approaches — including the original Concatenated features, the ESS, the Fine-Tuned, and the Cross-Entropy — shows the improvements provided by the combination learned using the proposed solution (Figure 2). We observed the best performance for Cross-Entropy features. This result attests the hypothesis that learning the combination of representative components features and their contributions can be effectively achieved with less data for training, especially due to the small network architecture. This latter result is corroborated experiments comparing different network depth (Figures S4, S5 and S6).

Even better results are obtained with the proposed distance-learning methods — Contrastive and Triplet — which generated more discriminative feature spaces (Figures S9 and 8), providing tighter groups of Representative images and also more proximity of the Representative images to the queries (Figures S10 and 9). The above conclusions are also evinced by the generated rankings (Figures S11, S12 and S13). The last two columns present more variability of retrieved images, avoiding the global similarity but also maintaining the precision of the top ranking images.

Finally, when comparing our learned representations with state-of-the-art methods for classification (in few training samples scenario) we observed that our distance-learning representations (Contrastive and Triplet) maintained superior Balanced Accuracy and F1-Measure (see Figures 13 and 14). The competitive results in the two new and unseen datasets (Museum and Bangladesh), which were not used for choosing our approach configuration, evince the great potential to generalize to other datasets. We refer the reader to additional results in the Supplementary Material comparing the rankings produced by our method in a different application — retrieval. In Figures S19 and S20, we show how our method can retrieve images in an unlabelled set proposing a ranked list to a human expert.

The results corroborate our conclusion: learning to combine features (and the contribution of the representative components) is an effective alternative to deal with the small number of training samples, specially in forensic setups in which we might have only a handful of initial examples for a given event of interest. Distance-learning methods — Contrastive and Triplet — probably led to the best results due to the nature of this task, which aims at learning to approximate images from the Representative group based on the components of the events. This combination also represents a possibility to prevent mistakes of global image similarity by using representative components concepts in a divide-to-conquer fashion.

Future work can be dedicated to analyze the context of the rankings to improve retrieval results. As our goal is to separate social media content related to events, we believe that helpful images to understanding events would retrieve other helpful ones, if they are used as entries in queries. This means that
the ranking of a given query might have important hints to be explored when in comparison with the ranking of another query image. For instance, if an image has an important set of neighbors and those neighbors are also neighbors of another query, there is a chance such images are of relevance to the query. Finally, adding some mechanism of relevance feedback in which the user can tag a few retrieved images in a first round of retrieval might also prove helpful in refining the results and providing further training examples to the proposed methods.

REFERENCES

[1] D. T. Nguyen, F. Alam, F. Ofli, and M. Imran, “Automatic image filtering on social networks using deep learning and perceptual hashing during crises,” CoRR, vol. abs/1704.02602, pp. 1–13, Apr. 2017. [Online]. Available: http://arxiv.org/abs/1704.02602
[2] M. Naaman and L. Gravano, “Beyond trending topics: Real-world event identification on Twitter,” in Proc. 5th Int. AAAI Conf. Weblogs Social Media (ICWSM), Barcelona, Spain, 2011, pp. 438–441.
[3] A. L. Kavanaugh et al., “Social media use by government: From the routine to the critical,” Government Inf. Qua., vol. 29, no. 4, pp. 480–491, Oct. 2017.
[4] A. L. Hughes and L. Palen, “Twitter adoption and use in mass convergence and emergency events,” Int. J. Emergency Manage., vol. 6, nos. 3–4, pp. 248–260, Feb. 2009.
[5] K. Starbird, L. Palen, A. Hughes, and S. Vieweg, “Chatter on the red: What hazards threat reveals about the social life of microblogged information,” in Proc. 10th ACM Conf. Comput. Supported Cooperat. Work (CSCW), Oct. 2010, pp. 241–250.
[6] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, “Processing social media messages in mass emergency: A survey,” ACM Comput. Surveys, vol. 47, pp. 1–37, Jun. 2015.
[7] T. Chen, D. Lu, M.-Y. Kan, and P. Cui, “Understanding and classifying image tweets,” in Proc. 21st ACM Int. Conf. Multimedia (MM), Oct. 2013, pp. 781–784.
[8] R. Peters and J. P. de Albuquerque, “Investigating images as indicators for relevant social media messages in disaster management,” in Proc. 12th Int. Conf. Inf. Syst. Crisis Response Manage. (ISCRAM), 2015, pp. 1–8.
[9] R. J. Topinka, “Terrorism, governmentality and the simulated city: The Boston marathon bombing and the search for suspect two,” J. Commun., vol. 15, no. 3, pp. 351–370, Aug. 2016.
[10] S. Saikia, E. Fidalgo, E. Alegre, and L. Fernández-Robles, “Object analysis of sparse features for effective content-based image retrieval,” in Proc. 19th Int. Conf. Image Anal. Process. (ICIAP), Cham, Switzerland: Springer, 2016, pp. 1–8.
[11] C. M. Rodrigues, L. Pereira, A. Rocha, and Z. Dias, “Image semantic content-based image retrieval: A comprehensive study,” J. Vis. Commun. Image Represent., vol. 23, pp. 20–54, Oct. 2012.
[12] A. Alzua’bi, A. Amira, and N. Ramzan, “Semantic content-based image retrieval: A comprehensive study,” J. Vis. Commun. Image Represent., vol. 32, pp. 20–54, Oct. 2015.
[13] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. M. Snoek, and A. D. Bimbo, “Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval,” ACM Comput. Surveys, vol. 49, no. 14, pp. 1–38, 2016.