On Localizing Urban Events with Instagram

***Abstract*—This paper develops an algorithm that exploits picture-oriented social networks to localize urban events. We choose picture-oriented networks because taking a picture requires physical proximity, thereby revealing the location of the photographed event. Furthermore, most modern cell phones are equipped with GPS, making picture location, and time metadata commonly available. We consider Instagram as the social network of choice and limit ourselves to urban events (noting that the majority of the world population lives in cities). The paper introduces a new adaptive localization algorithm that does not require the user to specify manually tunable parameters. We evaluate the performance of our algorithm for various real-world datasets, comparing it against a few baseline methods. The results show that our method achieves the best recall, the fewest false positives, and the lowest average error in localizing urban events.**

I. Introduction

This paper investigates social networks that carry pictorial information as a means to localize urban events of interest in time and in space. In turn, the ability to localize events gives rise to new search services that allow users to view important events matching a category of interest on a map, and remotely experience those events through the lenses of eye-witnesses. Since the majority of the world population lives in cities [1], we restrict ourselves to *urban* events.

The work is made possible thanks to the proliferation of picture-taking devices (e.g., over 2 billion smart phone users at present [2]) and picture-sharing media that offer a real-time view of ongoing events. We consider Instagram [3] as our social medium of choice. Instagram is a real-time picture sharing network, whose popularity has increased dramatically in recent years. At the time of writing, Instagram has more than 500 million users, who collectively upload 80 million pictures a day [4]. This is up from 400 million, 300 million, 150 million, and 30 million users in 2015, 2014, 2013, and 2012, respectively. Based on an experiment from a sample of images we collected that are publicly viewable, more than 15% contain location metadata, making it meaningful (given the large total volume) to consider Instagram as a tool for *localization*.

Localizing user-specified types of events based on pictures calls for a capability to associate the pictures with specific event keywords. Fortunately, Instagram users frequently associate customized metadata with images to identify what an image is of. Specifically, Instagram allows users to *tag* images they upload and also associate a spatial location based on the GPS. The followers of a user also have the option to like or comment on the image posts. This makes it possible to search Instagram images for those matching event-specific keywords. Instagram offers an application programming interface (API) that allows searching for images by using a tag keyword. Users can search for both current and previous images.

The above suggests that a text query for an event such as "*#JapanEarthquake*" or "*#ChicagoMarathon*" can retrieve pictures with annotations matching the query, from which the corresponding event can be localized. The manner in which pictures matching a set of keywords are identified is not the challenge addressed in this paper (It constitutes a standard database indexing problem). The challenge we address below is the way one might identify and localize events in space and in time *given* the set of retrieved pictures matching a query.

One of the prior works [5] explored the feasibility of using Instagram for identifying *points of interest* (POIs) such as tourist attractions within a city. It uses distance-based clustering techniques to group together images that fall within a threshold distance [6]. Generalizing the approach to enable localizing *events* is challenging for multiple reasons. First, events come and go. Hence, finding their signature requires detection not only in space but also in time. Second, different events can have a very different popularity and spatial signature. For example, a local marathon might engender a very different picture-taking response than a terror attack, which complicates the detection process. Instead, algorithms are needed that are capable of using the time and space properties of the data shared during an event without any labeled data.

The solution we propose is based on a technique that uses the distribution of pictures in the time domain along with a spatial range to observe the events to generate clusters followed by a false alarm elimination. We eliminate any manual inspection for parameter settings with the help of a self-evaluation scoring metric. In order to help us design an algorithm, we propose a set of assumptions that guide us in the derivation. Some of these assumptions fall within the scope of the design while the remaining are verified later during the evaluation with the help of collected datasets.

The rest of this paper is organized as follows. We first describe the state of art and related work in Section II. Section III describes the assumptions we make in order to derive the algorithm. We present the design of our system in Section IV. The collection of datasets, verification of assumptions, and algorithm performance results are discussed in Section V. Finally, we present the conclusion of our work in Section VI.

II. Related Work

A. Instagram: A popular image sharing social network

Due to an explosive increase in the user base over the past two years, Instagram has emerged as a popular platform among researchers to analyze social networks from a crowdsensing point of view. In [7], Instagram was studied as a social media visualization tool to identify cultural dynamics in major cities. The study particularly zoomed into the city Tel Aviv, Israel, for a period of two weeks collecting images shared on important national event days. In [8], an analysis was presented to identify different types of users on Instagram and the categories of pictures they take. The work characterized Instagram based on eight categories of pictures shared by five

distinct types of users. In [5], the authors have described about an approach that is capable of identifying important *tourist attractions* (POIs) with the help of Instagram. The idea of their approach is to discover places that are collectively geo-tagged using pictures by unique users. However the focus of this work is still limited to locations that are extensively visited by tourists. A very recent work [9] explores the capability of using Instagram pictures along with the metadata to find the correlation between obesity patterns and fast food restaurants located in few selected counties within United States. The focus of this work is specific to a particular category type, which is *food*, in order to find the trend of tags used by the people belonging to a certain region and compare with the corresponding health factors. In another work by Mejova et al. [10], a further analysis has been performed on the food habits of users on a global scale to answer questions related to health research. They identify the existence of emotions and health-related topics with Instagram pictures containing *#foodporn* as one of the tags. They also suggest that there is a social approval for users sharing healthy pictures compared to those sharing unhealthy pictures in terms of likes and followers. These works have provided a good indication of using Instagram for identifying popular and trending locations or topics among users.

B. Event Localization: Using geo-tagged data from social networks

The exploitation of social networks that expose *location information* has been studied in depth long before Instagram became popular. In [11], a study was reported on a popular location-based social network, Foursquare, to reveal user mobility patterns in urban spaces. Another work [12] focused on analyzing the mobility patterns of users to identify social ties based on co-location history, and determine the relation between location visits and network strength of a user. In [13], the authors have demonstrated the capability of using a popular social network, Twitter, to jointly localize events and sources. The focus was to make use of location affinities of users jointly with location references in tweets to infer location of events and sources in an iterative manner. In [14], the authors have presented a clustering technique for finding the dynamics of a city based on the check-ins posted by users on Foursquare. Noulas et al. [15] proposed a method that uses Foursquare check-ins to identify regions that are similar within a geographic area. In [16], an analysis of a photo sharing online network, Flickr, was presented to show the variation in the popularity of photos around a geographical location. Although most of these works have highly focused on using Foursquare or Flickr with geo-tagged data, but the purpose of using Foursquare or Flickr is completely different from that of Instagram among users. The latter is widely used to share images of an observed entity along with location information, while the former are used to post reviews and suggestions for a visited place. Detecting and localizing events using Instagram requires a different approach from the ones described above. This is mainly due to presence of several constraints in the way images and other data are shared by people for different types of events. Not all events are equally popular. For example, a Taylor Swift concert might have more observers at the event location as compared to an earthquake event. Also there is a high chance that several groups of users from different locations are talking about the same event. [17] is an early work that uses Instagram geo-tagged images to detect hyper-local events. Their method tries to identify any abnormal signal generated from a concentrated region followed by a classification technique to detect events. The authors of [18] have described about the implementation of a system capable of detecting events using geo-tagged data from networks such as Instagram. Their method determines the burst of keywords (tags) within a time interval, which is then modeled as Gaussians and events are detected based on mapping with bursty keywords. In [19] a method for event detection using Twitter has been described. Two classifier models are built based on text and image features that later decide the class of the geo-tagged tweet. [20] is another event detection work based on geo-tagged data from Flickr network. They focus on nine events using an online event directory to define a bounding box around venue using GPS data from Flickr images. The events are then detected using time-series analysis within the box based on a threshold. The authors of [21] describe a real-time detection of crash incidents using geo-tagged tweets. However their approach uses a classification model as well using a training dataset. The work presented in [22] is another event detection technique using geo-tagged images from social networks. A hybrid similarity graph is constructed based on tags and images to form clusters that are then classified using a trained model. In [23], the authors have presented an approach to detect events from social media with the help of geographical temporal pattern. Even though this work does not rely on any training dataset to build a classification model but it requires a few manual parameter tuning in order to achieve a good accuracy. There is also no discussion about removal of false alarms generated during event detection.

Contrary to the previous event detection and localization works using geo-tagged data from social networks, we provide a simple and robust approach that works online for streaming data feed without using any classification model. We do not rely on any training data to identify new events for localization and can easily adapt to any event type of varying degree of popularity. The novelty we introduce in this paper is an approach to find clusters of events (matching the user query) followed by false positive elimination without any manual parameter tuning. To our best knowledge, this approach has never been explored before for event localization in urban spaces.

III. ASSUMPTIONS

In this section we describe a set of assumptions that we use to design our localization algorithm. We divide our assumptions into two categories: *Category 1* is the set of assumptions that fall inside the scope of our algorithm design pattern and *Category 2* is the set of assumptions that can be verified using experiments. For the first category, the stated assumptions allow us to set the ground rules on top of which we design our system. We do not have to verify the validity of assumptions from this category as they must always hold true for the algorithm to work. For the second category, the stated assumptions describe the conditions for which the algorithm must work. The assumptions from this category need to be verified for validity which we provide later with the help of

randomly selected event samples from the collected datasets in the evaluation section.

A. Category 1 Assumptions

- Assumption 1: There can be only one event occurrence at a specific point location (latitude, longitude) during a particular time interval.
- Assumption 2: Two or more independent events can take place during the same time interval or have some overlap in their respective time intervals.
- Assumption 3: The users (sensors) generating the signals (pictures) are independent of each other. This means we do not consider the follower-followee relationship among the users for designing our algorithm. Every user has a range 'R', which determines the maximum distance of observing an event.
- Assumption 4: If there is a single user producing multiple signals over a period of time such that there is no other user supporting the observations within 'R' distance, then we mark that user as a false alarm.

B. Category 2 Assumptions

- Assumption 5: It is possible for users to post pictures of an event from a location (*False* location) different than the actual place of occurrence (*True* location). For example, multiple users watching a sports match at the stadium and those watching the same match in a group at a bar in a different city can post pictures for the same event indicating two different locations.
- Assumption 6: The number of users (sensors) generating signals (pictures) from the *True (Actual)* location of an event are always more than those located at any of the *False* locations for the same event. However, it is possible that the number of users from the *True* location of an event are less than the count from a *False* location of a different event.
- Assumption 7: The events generally belong to two major categories and the distribution of the signals (pictures) generated in the time domain follows a certain pattern for both the cases:
  a) *Planned Events* : The events that are scheduled well ahead in time, such as music concerts, generate attention from users much before the event begins but reaches a peak in the observations only after the event has actually started and then it gradually decreases over time.
  b) *Unplanned Events* : The events that are not scheduled ahead of their occurrence, such as an earthquake, generate a few posts at the actual time of occurrence but then reaches a peak in observation during the post event actions, such as relief or medical operations, and also gradually decreases over time.

  The peak in the observation is the mode (most frequent value) of the distribution. To handle both the cases we try to estimate the start time and the end time, so that we can localize the observed signals within this time frame. Since both the cases are skewed to some extent, the mean and standard deviation will be not good at predicting the spread of the distribution. Thus, we consider the first quartile as the estimated start, and the third quartile as the estimated end time.
- Assumption 8: There is some amount of prevalence in the popular tags used to describe an event regardless of *True* or *False* locations.

IV. System Design

The goal of our work is to identify the locations of real-world events in time and space based on the data shared by users on the Instagram social network. We derive an algorithm that is capable of detecting and localizing the events in physical space. Figure 1 is a visual representation of the design of our system indicating the flow between different components. In the following subsections, we describe the functionality of each component in the pipeline.

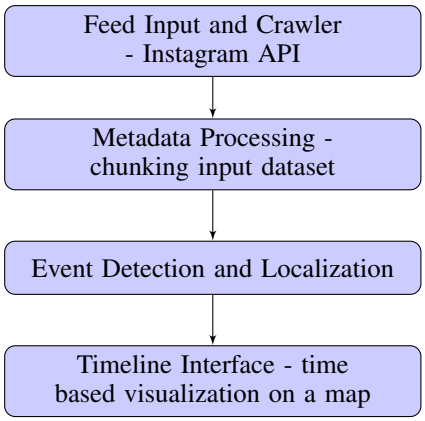

Fig. 1. Architecture Pipeline of the Localization Tool for Instagram

A. Feed Input and Crawler

Our system follows the feed subscription model as opposed to the search query model. This allows the user to monitor the events of different types on a timely basis in near real time. The user can view any of the subscribed events from the past or create a new subscription for an event with the help of a "tag" keyword. This tag can be anywhere between from being completely generic (*#Earthquake*) to completely specific (*#JapanEarthquake*). It is left upto to the user to decide about the amount of generalization required for the feed retrieval. Once an event has been subscribed the crawler service makes request to the Instagram API using the associated tag at an interval of 1 hour. This choice of request interval is based on two factors: (i) To avoid any spam detection, and (ii) A limitation of 5000 calls per hour set by the Instagram API. The API allows us to search for images based on a tag in two different ways; the first one is by popularity and the second one is by recency. Since we have a subscription model we use the most recent results during every interval. Every image has a tag ID in the metadata such that they are sorted in ascending order in the recency list. This allows our crawler service to identify the tag ID at which the call needs to be stopped for the current interval. The retrieved images along with metadata are then sent to the next component for further processing.

B. Metadata Processing

The image posts obtained from the crawler service are processed in this step to remove any noise present. Every image has a metadata component, which contains several fields. We make use of only *image id*, *image url*, *user id*, *created time*, *tags*, and *location*. We filter out any image for which the *location* field is empty. Next we make use of *created time* of the image post to divide the data feed into intervals. This step is repeated for every API call and the image is added to the corresponding interval. Any updated interval is then sent to the next component in the pipeline.

C. Event Detection and Localization

The main contribution of our work is a novel and simple algorithm to identify locations of physical events in urban spaces using Instagram. We make use of the assumptions described earlier in order to derive our localization algorithm. The following two subsections contain the problem definition and the details of our algorithm. In the problem definition section, we introduce all the variables that will be used during the derivation of our algorithm.

1) Problem Definition: Each signal (picture) generated by a user is a tuple of the format $< l, t, u, tag >$, where l - location, t - image post time, u - user id, tag - set of tags.

If for a selected time interval there are 'K' unique locations (l_1,l_2, ... , l_k) present then for each location $l_{k \in K} \rightarrow < t_{ik}, u_{jk}, tag_{ik} >$, where
$< t_{ik} > \rightarrow$ time instance of an image i at location 'k'
$< u_{jk} > \rightarrow$ user j posting an image i at location 'k'
$< tag_{ik} > \rightarrow$ tags of an image i at location 'k'

A sensing range 'R' is needed for the algorithm, which determines the upper limit until which an user can observe (or talk about) the event. We eliminate this parameter setting with the help of *silhouette score* in order to avoid any manual intervention for our algorithm. This metric computation is a three step process based on the following:

- Cohesion Factor (a_i): For the i^{th} data point, we find the average distance to all other data points within the same cluster.
- Separation Factor (b_i): For the i^{th} data point, we find the average distance from all the data points of another cluster to which it does not belong. Then we take the minimum of the average distances from all the clusters.
- Silhouette Coefficient: Finally, we assign the score to the i^{th} data point using the equation $s_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$.

The silhouette coefficient for any data point is in the range $[-1, 1]$. The ideal best case is when $a_i = 0$ for which the maximum value of 1 is attained. For our algorithm we vary the value of 'R' between 0.25 and 30 miles.

For the current time interval, we use a padding of previous 3 days to find and localize the events. All the events in current time interval are represented by a set of cluster ξ = [E_1,E_2,...,E_m], where $m \leq K$.

2) Localization Algorithm: The main intuition behind the derivation of our algorithm is that for any current time interval we try to find the estimated start and end time of the image sharing activity for all possible geo-tagged locations within that interval. We then group these locations into an event cluster based on time overlap and a sensing range parameter. Finally we eliminate any false alarm clusters generated with the help of similarity between popular tags present in each cluster. The algorithm is described below:

i) In the current time interval, we arrange $l_{k \in K}$ in descending order by length of associated u_{jk} (If there is a tie, then we use length of associated t_{ik}). Let this be the ordered list of locations. We make use of both Assumptions 1 and 6 for this step.
ii) Process the locations from the ordered list one at a time.
iii) For a selected location, use the $< t_{ik}, u_{jk}, tag_{ik} >$ including the padded intervals. Using Assumption 7, we find the estimated start and end time of the distribution. There are 4 cases possible:
   - Both estimated times are outside the current interval. This means that this event occurred in one of the previous intervals. Discard this location and move to next location.
   - Both estimated time are inside the current interval. Use the location for analysis with data within estimated time range.
   - One of the estimated time is inside the current interval. Use the location for analysis with data between boundary of interval and the estimated time.
   - Both the estimated times capture the boundaries of current interval. Use the location for analysis with all data within the interval boundary.
iv) Let the l_k for analysis from the current interval have t_{start_k} as start time and t_{end_k} as end time. If this is the very first location in the ordered list, then form a new cluster E_1 representing an event. This l_k is the prime location (l_{prime}) for this newly formed cluster indicating the most probable value.
v) If the l_k for analysis is not the first location from the ordered list, then scan through each event cluster from ξ for two conditions:
   - If the t_{start_k} and t_{end_k} have overlap with (t_{start_prime}, t_{end_prime}) the prime location of that cluster.
   - If l_k is within 'R' miles of distance from the prime location l_{prime} (using Assumption 3).

   If both conditions are satisfied, then l_k goes into the same cluster.
vi) If only the first condition is satisfied then either l_k is a false alarm location (Assumption 5) or another event location happening at the same time (Assumption 2). When both conditions are not satisfied then l_k is highly likely to be an undetected event location. In either case, we form a a new cluster with l_k as the prime location.
vii) We repeat the steps 5 and 6 for varying values of 'R' as indicated earlier and compute the silhouette score in each case. Finally, we select the range (R_{sel}) with the maximum score.
viii) Once all the locations from the ordered list are analyzed, we eliminate those clusters that have only a single user

inside with no support within R_{sel} distance according to Assumption 4.

ix) In order to eliminate the false alarm clusters, we use Assumption 8 to compute the similarity in the vectors formed by considering the top 10 commonly used tags from each cluster. We process the clusters in the order they were formed and check for identical clusters from the remaining. The similarity threshold setting is described later in the text.

x) The estimated location of an event is the weighted average of the 'l's inside the cluster. The weights are derived using the fraction of images posted from a location compared to images present inside the cluster.

For elimination of false alarm clusters, we first need to identify the type of event. Events can be broadly classified into two categories: (i) Single Entity (SE), and (ii) Multiple Entities (ME). For example, Taylor Swift being a single entity (person) can perform only at one valid location during a particular time interval. If there are several clusters identified for a SE event in the same interval then only one of them can be a true positive while the remaining are false alarm clusters. However, a ME event such as marathon or tornado can occur at several locations during the same interval. Based on the clusters generated, we looked at a few random samples and noticed that the size of the main cluster in case of SE events was always significantly large compared to the false alarms. At the same time, the clusters in case of ME events were comparable in size.

TABLE I. CLUSTER STATISTICS FOR EVENTS WITH SINGLE ENTITY

Event	Date	City	Top 5 cluster size
Taylor Swift	09/29/15	St. Louis, USA	[169, 27, 6, 5, 1]
Taylor Swift	10/03/15	Toronto, Canada	[940, 24, 6, 6, 5]
Maroon V	06/12/15	Milan, Italy	[134, 13, 8, 5, 5]
Maroon V	09/17/15	Manila, Philippines	[181, 11, 9]

TABLE II. CLUSTER STATISTICS FOR EVENTS WITH MULTIPLE ENTITES

Event	Date	Cities	Top 5 cluster size
Marathon	10/18/15	Columbus, OH, USA Detroit, MI, USA Toronto, Canada	[266,172,112,79,74]
Marathon	10/25/15	Washington DC, USA Frankfurt, Germany Jakarta, Indonesia	[153,134,88,58,46]

The cluster statistics for a random sample of events are provided in table I for SE and table II for ME. The last column in the tables corresponds to the top five clusters by size (number of data points) for each event type on a particular date. It can be clearly observed that in case of SE events the top most cluster by size is extremely dense in comparison to other clusters, whereas in case of ME events the true clusters don't have a huge difference. This can be attributed to the fact ME events attract the attention of people from all the ground truth locations at more or less the same rate.

Hence, we need to be careful while selecting the similarity threshold for these two types of events. In case of SE events, almost all the clusters can be expected to have high similarity among the popular tags, whereas the ME events may not share the popular tags across all the clusters. This means we might have to set a really low threshold value for SE events but a relatively higher threshold value for ME events. Based on the observations from tables I and II, we use the size of the ordered clusters as a function to determine the threshold value. A huge drop in size from first cluster (E_1) to second cluster (E_2) signifies a single entity event and thus $threshold = \frac{len(E_2)}{len(E_1)}$ assigns a really small score. At the same time this score will be much larger in case of multiple entities events due to comparable cluster sizes. This function for assigning the score also satisfies the bounding range for similarity score $[0, 1]$. Algorithm 1 is the pseudo code for the steps that have been described so far.

Algorithm 1 Localization Algorithm

procedure LOCALIZE($< l_k, t_{ik}, u_{jk}, tag_{ik} >, K$)
  $orderList \leftarrow sort(l_k, u_{jk}, t_{ik})$
  $events \leftarrow []$
  $count = 1$
  for $l_k \in orderList$ **do**
    $D \leftarrow distribution(l_k, t_{ik})$
    $t_{start\_k} = D(quartile_1)$
    $t_{end\_k} = D(quartile_3)$
    if $count == 1$ **then**
      $event[peak] = D(mode)$
      $event[start] = t_{start\_k}$
      $event[end] = t_{end\_k}$
      $event[prime] = l_k$
      $insert(events, event)$
    else
      for $event \in events$ **do**
        if $overlap(l_k, event, R) == True$ **then**
          $event[support] = l_k$
        else
          $event[peak] = D(mode)$
          $event[start] = t_{start\_k}$
          $event[end] = t_{end\_k}$
          $event[prime] = l_k$
          $insert(events, event)$
      $count = count + 1$
  $n \leftarrow len(events)$
  for $i \in (1, n)$ **do**
    $vector_i = get\_tags(events[i])$
    for $j \in (i + 1, n)$ **do**
      $vector_j = get\_tags(events[j])$
      if $similarity(vector_i, vector_j) > \theta$ **then**
        $remove(events[j])$

D. Timeline Interface

The last component in the pipeline is the timeline-based visualization on a map interface. A timeline slider is provided that lets the user to jump to a particular date in order to view the corresponding events. The interface shows pins for each event that have been localized by our algorithm. Figure 2 is an example map interface for localized *marathon* events on *10/18/15*.

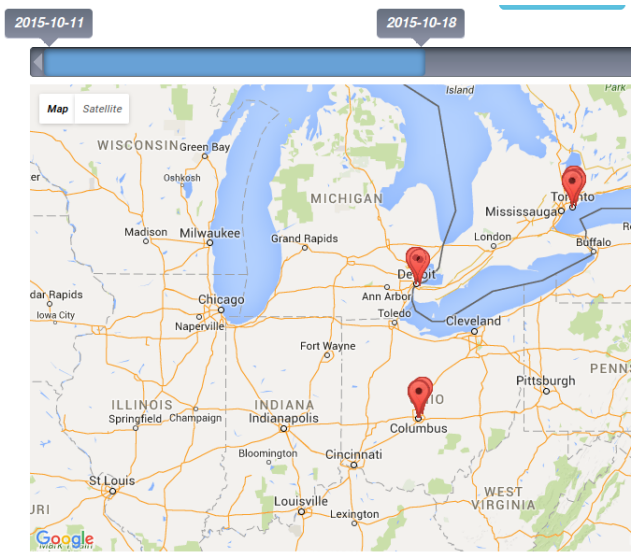

Fig. 2. Timeline map based interface

V. Evaluation

In this section, we first describe the various real-world datasets that we collected using the Instagram API. With the help of these datasets, we verify the assumptions from *Category 2* that were presented earlier in this paper and then finally show the comparison of the performance of our algorithm against a few baseline methods for localizing events.

A. Collected Datasets

1) Dataset 1 - Taylor Swift Music Tour: Taylor Swift, one of the most popular American singers, conducted a music concert tour called **The 1989 World Tour** in various cities across the world. We collected the complete set of Instagram posts related to this tour using the hashtag $\#1989worldtour$ starting from May 5, 2015, until December 12, 2015. We evaluate a total of 28 events spanning across the last three months of the event tour that happened in various cities in United States, Canada, Asia and Southeast Asia, and Australia. The ground-truth locations for all the events were obtained from the Wikipedia page [24] associated with the tour.

2) Dataset 2 - Maroon V Music Tour: The **Maroon V Tour** is a music concert tour by the popular American band Maroon V. We collected the Instagram posts related to this tour using the hashtag $\#maroonvtour$ starting from February 16, 2015, until October 4, 2015. We evaluate a total of 17 events from the months of September and October spanning different cities in south east Asia and Australia. The ground-truth locations for all the events were obtained from a Wikipedia page [25] associated with the tour.

3) Dataset 3 - Marathons: According to the 2014 annual marathon report [26], more than $1,100$ races were completed across the United States, making it one of the most popular urban sporting events. For the purpose of evaluating our work, we considered the top 30 cities in the United States, ranked by population [27] that hosted a popular marathon [28] during the fall of 2015. Based on this filtering, we identified five major events, listed in table III.

TABLE III. List of major US Marathons, Fall 2015

Event	City	Marathon	Date
1	Chicago	Bank of America Marathon	Oct 11
2	Baltimore	The Under Armour Marathon	Oct 17
3	Washington D.C.	Marine Corps Marathon	Oct 25
4	NY City	TCS Marathon	Nov 1
5	Las Vegas	Rock n Roll Marathon	Nov 15

Instagram posts related to marathon events were collected using the hashtag $\#marathon$. It is important to note that this search query tag is not targeted towards a particular entity such as a name (Maroon V) as in the case of previous datasets. Thus, this data set is much more "noisy" compared to others, making it a very interesting case to consider.

4) Dataset 4 - Tornadoes: The number of fatalities caused by **tornadoes** in the United States during the year 2015 [29] is estimated at 480, an exponential increase compared to 54 recorded during 2014. The strength of a tornado is computed using the *EF* scale ranging between $[0, 5]$ based on the damage caused. In 2015 (Jan - Oct), there were no *EF5* tornadoes, while the count of *EF4* was 2 and *EF3* was 8. The *EF3* tornadoes have mostly occurred in rural areas with populations less than $5,000$, except for one urban location. Instagram posts related to tornadoes were collected using the hashtag $\#tornado$. Table IV lists the filtered set of tornadoes that caused severe fatality in urban areas.

TABLE IV. List of Fatal Tornadoes, 2015 (Jan-Oct)

Event	City	EF	Date
1	Rochelle, IL, USA	4	April 9, 2015
2	Oklahoma City, OK, USA	3	May 6, 2015
3	Venice, Italy	4	July 8, 2015

B. Verification of assumptions from Category 2

Before we present the performance results of our localization algorithm, we demonstrate the validity of the assumptions that were made earlier while deriving the algorithm. Specifically, we focus on *Category 2* assumptions that can be verified with the help of experiments using the datasets collected.

1) Validation 1: In figure 3, we show the distribution of unique users present in *True* versus the *False* clusters for fifteen events that were randomly selected from the output of our localization algorithm using the collected datasets. The x-axis represents the event ID while the y-axis represents the fraction of users who posted images for that particular event. This figure validates two assumptions at the same time. Firstly, we can see that there are some groups of users who are located at places other than the actual event location (Assumption 5), and secondly, the fraction of users from the *True* location is always greater than the *False* location for the same event (Assumption 6).

2) Validation 2: In figure 4, we verify Assumption 8 using the same random fifteen events that were selected for validation 1. For each event, we first identify the top 10 commonly used tags according to frequency (we remove the tag word used for search query) from both *True* and *False* clusters. Next, we determine the similarity between the *True* cluster vector with

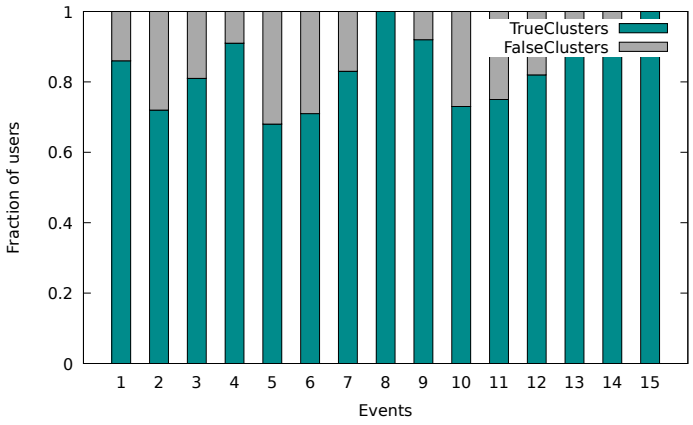

Fig. 3. True and False event clusters

each of the corresponding *False* cluster vector and take the average score. Figure 4 shows the boxplot representation for the average similarity scores that were obtained for the random samples. It is evident that the median of these scores is around 0.65 and the minimum score is well above 0.5. Thus, there exists some amount of prevalence of common tags between the *True* and *False* clusters of the same event.

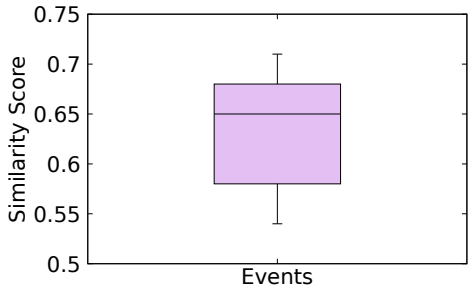

Fig. 4. Prevalence of Commonly Used Tags

3) Validation 3: Finally, we select four random event samples from each of the two categories (*Planned* and *Unplanned*) to plot the distribution of the frequency of images shared form the *True* location in order to verify our Assumption 7. For these plots, the x-axis represents the timestamp ID and y-axis represents the frequency of images that were shared for a particular time interval. Figure 5 consists of four subplots for planned events in which users start posting images well ahead of scheduled time and there is a peak around the time when the event actually takes place. Figure 6 consists of four subplots for unplanned events in which there is a peak observed right after the event and then it gradually falls down over a period of time. Hence, for either case, we established the fact that users tend to maximize the observation very close to the event occurrence.

C. Performance of our Localization Algorithm

With the establishment of the validity of the assumptions that we made in order to derive our localization algorithm, we now compare the performance of the results against a few baseline methods using different metrics. The baselines and the metrics are discussed in detail below followed by the comparison tables.

1) Baseline Method 1 - Tag Similarity Localization: The first baseline method is based on the intuition that all the observations for an event are closely linked to each other in terms of common tags used for description (this is according to our Assumption 8). We follow the same processing method for the incoming feed of data using the crawler. For any current interval, we consider all the unique K locations $(l_1, l_2, \ldots, l_k)$ along with the associated $< t_{ik}, u_{jk}, tag_{ik} >$. We then form a cluster by grouping all the l's for which the similarity score among the top 10 common tag words is at least $X\%$. We vary the value of X as 20, 40, 60, and 80 respectively. For each case, we use the same false alarm cluster elimination technique as described in our own algorithm. The higher the threshold for grouping locations, the better the results will be.

2) Baseline Method 2 - Geo Event Detection: For the second baseline method, we use the work described by the authors of [23] for geographical social event detection in social media. This work is very closely related to our motivation in terms of using geo-tagged data to detect events. We implement their algorithm as mentioned to detect the events on our collected datasets. Specifically, we do per day analysis for the four time slots on each geographic region present for that day. A region comprises of geo-coordinate with maximum number of users and all points within 30 miles of radius from it. There is a threshold requirement for abnormal geographic regions. We vary this θ value as 0.2, 0.4, 0.6 and 0.8 to see the effect on localization. The minimum number of observations required in a cluster is set as 3.

3) Baseline Method 3 - Points of Interest: For the third baseline, we use the work described by the authors of [5] in order to find points of interests using pictures shared by users on the Instagram network. This work can be very well applied to our interest of finding the locations of events. However, the authors conducted the experiments on very popular locations. Thus, we again set the minimum number of observation required in a cluster as 3 and use the approach as described in the paper.

4) Metrics for comparison: We use three metrics in order to compare the performance of our localization algorithm against the selected baseline methods:

- *Recall* : Determines the count of events that were detected and localized from the available set of events.
- *False Positives (FP)*: Determines the count of events that were falsely classified as positive.
- *Average Localization Error (ALE)* : Determines the average error in the estimated location from the actual ground truth for all the localized events.

Table V is the recall value comparison between our localization algorithm and the baseline methods under different settings. Our method performed consistently well in correctly identifying all the events. Baseline 2 method also gave a perfect recall.

Table VI is the false positives value comparison between our localization algorithm and the baseline methods under different settings. It can be clearly seen that our method

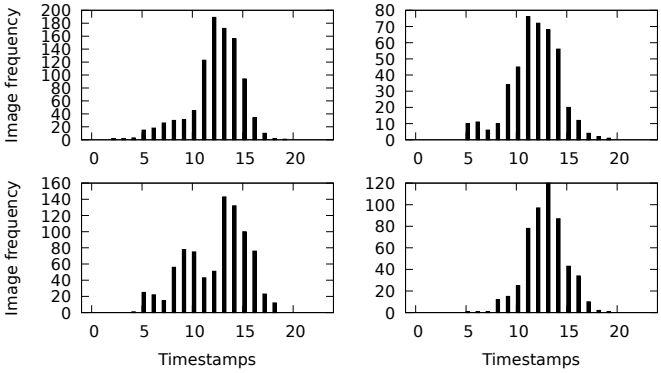

Fig. 5. Planned Events

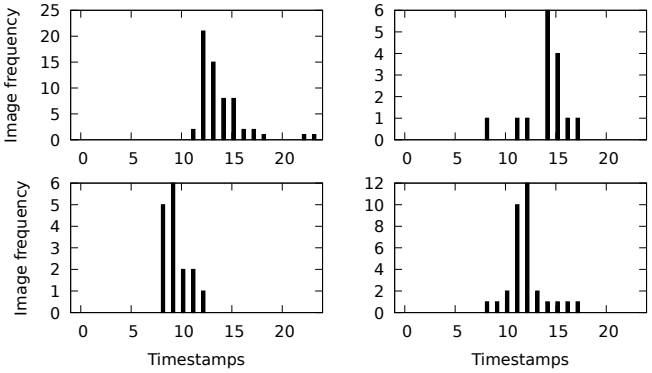

Fig. 6. Unplanned Events

TABLE V. RECALL

Dataset	Our Localization Algorithm	Tag Similarity Localization				Geo Event Detection [23]				Points of Interest [5]
		$X = 20\%$	$X = 40\%$	$X = 60\%$	$X = 80\%$	$\theta = 0.2$	$\theta = 0.4$	$\theta = 0.6$	$\theta = 0.8$	
Taylor Swift	28/28	24/28	25/28	26/28	26/28	28/28	28/28	28/28	28/28	27/28
Maroon V	17/17	13/17	15/17	17/17	17/17	17/17	17/17	17/17	17/17	17/17
Marathon	5/5	3/5	4/5	4/5	4/5	5/5	5/5	5/5	5/5	5/5
Tornado	3/3	2/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3

TABLE VI. FALSE POSITIVES

Dataset	Our Localization Algorithm	Tag Similarity Localization				Geo Event Detection [23]				Points of Interest [5]
		$X = 20\%$	$X = 40\%$	$X = 60\%$	$X = 80\%$	$\theta = 0.2$	$\theta = 0.4$	$\theta = 0.6$	$\theta = 0.8$	
Taylor Swift	2	18	10	5	4	35	16	9	9	26
Maroon V	0	5	4	4	2	19	8	8	8	14
Marathon	0	16	10	7	6	17	11	11	11	15
Tornado	1	3	3	3	2	6	6	6	6	19

generated the least number of false alarm clusters for any dataset.

Table VII is the ALE comparison between our localization algorithm and the baseline methods under different settings. It can be clearly seen that our method has the best average error rate for the estimated location from the actual ground truth. In case of first two datasets (which are immobile events), the average error is almost close to zero, but for the other two datasets (mobile events), the average error is close to 6 miles in worst case.

VI. CONCLUSIONS

This paper presents an algorithm for localizing urban events using geo-tagged media from the Instagram social network. The motivation for this work comes from the fact that the spatio-temporal behavior of events varies from one type to another leading to two main challenges : determining a way to identify events using the geo-tagged data within an interval and reducing false alarm indicators. The first one is solved with the help of a clustering technique based on the distribution of the images (observations) in both time and spatial domains. We provide an adaptive way to maximize the best set of clusters generated. The second challenge is solved by considering similarity between clusters to minimize the false positives. In order to derive an algorithm to solve these problems, we provide a set of assumptions, which are later verified using experimental results. For evaluation, we consider three baseline methods and compare the results with our localization algorithm. The results show that we outperform the baseline methods for all the three metrics considered for comparison. Also we achieved this result without the need to tune any manual parameter in our algorithm.

REFERENCES

[1] United nations - world population statistics. https://www.un.org/development/desa/en/news/population/world-urbanization-prospects.html.

[2] Smartphone user statistics. http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

TABLE VII. AVERAGE LOCALIZATION ERROR (MILES)

Dataset	Our Localization Algorithm	Tag Similarity Localization				Geo Event Detection [23]				Points of Interest [5]
		$X = 20\%$	$X = 40\%$	$X = 60\%$	$X = 80\%$	$\theta = 0.2$	$\theta = 0.4$	$\theta = 0.6$	$\theta = 0.8$	
Taylor Swift	0.03	102.87	25.06	2.86	1.02	0.78	0.78	0.78	0.78	0.17
Maroon V	0.12	75.34	32.78	10.23	2.67	1.23	1.23	1.23	1.23	1.32
Marathon	3.45	141.43	34.33	16.18	4.12	4.82	4.82	4.82	4.82	5.86
Tornado	6.02	40.23	25.23	11.34	11.34	9.06	9.06	9.06	9.06	8.47

[3] Instagram. https://www.instagram.com.

[4] Instagram statistics. http://mediakix.com/2016/03/top-instagram-statistics-you-should-know/#gs.GiCN2lw.

[5] T.H. Silva, P.O.S.V. de Melo, J.M. Almeida, J. Salles, and A.A.F. Loureiro. A picture of instagram is worth more than a thousand words: Workload characterization and application. In *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, pages 123–132, May 2013.

[6] Complete linkage clustering. http://nlp.stanford.edu/IR-book/completelink.html.

[7] Nadav Hochman and Lev Manovich. Zooming into an instagram city: Reading the local through social media. *First Monday*, 18(7), 2013.

[8] Yuheng Hu, Lydia Manikonda, Subbarao Kambhampati, et al. What we instagram: A first analysis of instagram photo content and user types. *Proceedings of ICWSM. AAAI*, 2014.

[9] Yelena Mejova, Hamed Haddadi, Anastasios Noulas, and Ingmar Weber. #foodporn: Obesity patterns in culinary interactions. *CoRR*, abs/1503.01546, 2015.

[10] Y. Mejova, S. Abbar, and H. Haddadi. Fetishizing Food in Digital Age: #foodporn Around the World. *ArXiv e-prints*, March 2016.

[11] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An empirical study of geographic user activity patterns in foursquare, 2011.

[12] Justin Cranshaw, Eran Toch, Jason Hong, Aniket Kittur, and Norman Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, pages 119–128, New York, NY, USA, 2010. ACM.

[13] P. Giridhar, Shiguang Wang, T.F. Abdelzaher, J. George, L. Kaplan, and R. Ganti. Joint localization of events and sources in social networks. In *Distributed Computing in Sensor Systems (DCOSS), 2015 International Conference on*, pages 179–188, June 2015.

[14] Justin Cranshaw, Raz Schwartz, Jason Hong, and Norman Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city, 2012.

[15] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. Exploiting semantic annotations for clustering geographic areas and users in location-based social networks. *The social mobile web*, 11:02, 2011.

[16] Roelof van Zwol. Flickr: Who is looking? In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 184–190, Washington, DC, USA, 2007. IEEE Computer Society.

[17] Ke Xie, Chaolun Xia, Nir Grinberg, Raz Schwartz, and Mor Naaman. Robust detection of hyper-local events from geotagged social media data. In *Proceedings of the Thirteenth International Workshop on Multimedia Data Mining*, MDMKDD '13, pages 2:1–2:9, New York, NY, USA, 2013. ACM.

[18] Pierre Houdyer, Albrecht Zimmerman, Mehdi Kaytoue, Marc Plantevit, Joseph Mitchell, and Céline Robardet. Gazouille: Detecting and illustrating local events from geolocalized social media streams. In *Machine Learning and Knowledge Discovery in Databases*, pages 276–280. Springer, 2015.

[19] Samar M Alqhtani, Suhuai Luo, and Brian Regan. Fusing text and image for event detection in twitter. *arXiv preprint arXiv:1503.03920*, 2015.

[20] Xueliang Liu, Raphaël Troncy, and Benoit Huet. Using social media to identify events. In *Proceedings of the 3rd ACM SIGMM International Workshop on Social Media*, WSM '11, pages 3–8, New York, NY, USA, 2011. ACM.

[21] Axel Schulz, Petar Ristoski, and Heiko Paulheim. *The Semantic Web: ESWC 2013 Satellite Events: ESWC 2013 Satellite Events, Montpellier, France, May 26-30, 2013, Revised Selected Papers*, chapter I See a Car Crash: Real-Time Detection of Small Scale Incidents in Microblogs, pages 22–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[22] Symeon Papadopoulos, Christos Zigkolis, Yiannis Kompatsiaris, and Athena Vakali. Cluster-based landmark and event detection for tagged photo collections. *IEEE Multimedia*, 18(1):52–63, 2011.

[23] Xingyu Gao, Juan Cao, Qin He, and Jintao Li. A novel method for geographical social event detection in social media. In *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*, ICIMCS '13, pages 305–308, New York, NY, USA, 2013. ACM.

[24] 1989 world tour wikipedia page. https://en.wikipedia.org/wiki/The_1989_World_Tour.

[25] Maroon v tour wikipedia page. https://en.wikipedia.org/wiki/Maroon_V_Tour.

[26] Marathon annual report. http://www.runningusa.org/index.cfm?fuseaction=news.details&ArticleId=332.

[27] Top us cities by population. http://www.infoplease.com/ipa/a0763098.html.

[28] Best fall 2015 marathons. http://dailyburn.com/life/fitness/best-marathons-fall/.

[29] Tornadoes 2015 wikipedia page. https://en.wikipedia.org/wiki/Tornadoes_of_2015.