Physica A 540 (2020) 123151

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Early warning system: From face recognition by surveillance cameras to social media analysis to detecting suspicious people

Salim Afra [a], Reda Alhajj [a,b,*]

[a] Department of Computer Science, University of Calgary, Calgary, Alberta, Canada
[b] Department of Computer Engineering, Istanbul Medipol University, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 27 May 2019
Received in revised form 30 August 2019
Available online 16 October 2019

Keywords:
Surveillance
Security camera
Monitoring
Early warning
Social media
Intelligence service

ABSTRACT

Surveillance security cameras are increasingly deployed in almost every location for monitoring purposes, including watching people and their actions for security purposes. For criminology, images collected from these cameras are usually used after an incident occurs to analyze who could be the people involved. While this usage of the cameras is important for a post crime action, there exists the need for real time monitoring to act as an early warning to prevent or avoid an incident before it occurs. In this paper, we describe the development and implementation of an early warning system that recognizes people automatically in a surveillance camera environment and then use data from various sources to identify these people and build their profile and network. The current literature is still missing a complete workflow from identifying people/criminals from a video surveillance to building a criminal information extraction framework and identifying those people and their interactions with others We train a feature extraction model for face recognition using convolutional neural networks to get a good recognition rate on the Chokepoint dataset collected using surveillance cameras. The system also provides the function to record people appearance in a location, such that unknown people passing through a scene excessive number of times (above a threshold decided by a security expert) will then be further analyzed to collect information about them. We implemented a queue based system to record people entrance. We try to avoid missing relevant individuals passing through as in some cases it is not possible to add every passing person to the queue which is maintained using some cache handling techniques. We collect and analyze information about unknown people by comparing their images from the cameras to a list of social media profiles collected from Facebook and intelligent services archives. After locating the profile of a person, traditional news and other social media platforms are crawled to collect and analyze more information about the identified person. The analyzed information is then presented to the analyst where a list of keywords and verb phrases are shown. We also construct the person's network from individuals mentioned with him/her in the text. Further analysis will allow security experts to mark this person as a suspect or safe. This work shows that building a complete early warning system is feasible to tackle and identify criminals so that authorities can take the required actions on the spot.

* Corresponding author.
E-mail addresses: salim.afra@ucalgary.ca (S. Afra), alhajj@ucalgary.ca (R. Alhajj).

https://doi.org/10.1016/j.physa.2019.123151
0378-4371/

1. Introduction

Video surveillance systems are installed and used almost everywhere nowadays for the purpose of recording, monitoring and reviewing incidents that may happen around from permitting only certain persons to enter a building to identifying potential suspicious criminals as early and preventive as possible. Several applications are associated with video surveillance systems such as traffic monitoring, security systems, incident recording, etc. Images from surveillance security cameras/closed-circuit television (CCTV) are used as an important evidence during crime investigations to identify key persons who are involved in the crime. In theory, using CCTV images to identify people involved in a crime scene and compare these collected face images to gallery images of criminals should be a straightforward process for police officers and crime forensic experts. This might be true and affordable for limited cases. However, the current era of globalism and the associated big data turns manual analysis infeasible and hence pushes hard towards more effective automated systems capable of supporting investigators in their duties.

Many researchers, e.g., [1–4], however, have shown that identifying unfamiliar faces and comparing CCTV images with mugshot gallery images is very difficult and challenging even for humans and police officers. Bruce et al. [1] performed several experiments aimed at testing the ability of people to identify faces in mugshot images. People were shown a person's face and then shown 10 other target images of the same person and other people for the purpose of matching the shown face with one of the 10 candidate images. The subjects were then asked to decide whether or not the shown face was present in the 10 other images; and if it is present, to pick the correct match. The results of this experiment showed that people performed poorly. They picked the correct person only about 70% of the occasions. The department of psychology at University of Glasgow [5] did a research work on the ability of individuals and police officers to identify target people captured by a surveillance security camera. They performed experiments to answer questions about the performance of people to identify familiar and unfamiliar people in a video surveillance environment. The first experiment examined whether personal familiarity with people in the video affects recognition rate. They did the experiments with 20 students who knew people in videos, 20 other students unfamiliar, and 20 police officers experienced in the field of forensic investigations but are unfamiliar with the subjects. They concluded through their experiments that individuals who are familiar with the targets performed very well at identifying them, while individuals unfamiliar with the targets performed very poorly along with police officers who performed as poorly as unfamiliar students.

Due to recent advances in technology and machine learning models being proposed, face recognition using a machine outperforms in many cases the performance of humans in the ability to identify people using face images [6]. This certainly helps in automating the identification process in recognizing face images collected using surveillance cameras and solves a problem that many current surveillance systems have, i.e., they are mostly used as recording machines. Such that if an incident occurs, cameras are used for analysis after and not as part of an integrated warning system if an unusual behavior occurs in the image frames. A modern surveillance system is expected to do real time analysis on images it get and not just do basic object detection and tracking. But also to interpret object behavior and warn security officials of any security breach on the spot, and hence avoiding any more danger.

Research in video surveillance systems took a step towards making these systems automated in analyzing and processing video images in real time, overcoming the manual monitoring of security personnel process. Identifying people in a surveillance video camera is an important task of face recognition where many institutes need systems for the purpose of access control, security monitoring, etc. Identifying a person's identity using surveillance cameras is challenging due to the variety of factors involved in the identification process, including the background environment, person's motion, variable lightening, and face visibility and detail exposure.

Face recognition has been extensively studied over the past decade to improve the performance and applicability of face recognition. Where early research efforts on face recognition [7,8] focused on identifying people in frontal face images taken in a controlled environment where the background, pose, illumination are all pre-defined and set. These methods extract local descriptors of the image based on pixel intensity. Several other methods [9–11] were later designed to improve performance accuracy of face recognition in frontal controlled environment setting. The problem of face recognition then shifted to identifying people in uncontrolled environments (wild) where face images are collected outside a lab environment. Recent research work [6,12–14] applied face recognition by first extracting features of given faces using convolutional neural networks (CNN) and then applied distance measures to compare face images for identification.

Identifying people's images to check if the person is suspect or not is one of the most important tasks for an automated surveillance system that applies face recognition. However, in real world scenarios such as at airports, military areas, diplomatic and official regions, street blocks, etc. people with criminal intent can pass and perform criminal activities in a location where the criminal's face may be detected in the surveillance cameras but authorities do not have this person marked as a suspect. It is important for security surveillance systems to identify unknown people and try to get more information about a given person to try and predict if he/she is safe or dangerous to take appropriate precautions, as many crime incidents happen where authorities do not have the person marked as a suspect.

In this work, we propose a security surveillance system that acts as an early warning system to detect from camera images not only suspects and known people, but also to apply further investigation on unknown people passing though a location that might be dangerous. In our system, we collect more information about unknown people passing through a scene by matching the person under investigation with his/her social media profile and then applying further analysis on his/her posts to conclude if the person is a potential suspect or not. The goal of this system is to help security officers in crime forensic to get more information about people to assist them so that they can take decisions on the spot.

The contributions of this paper are:

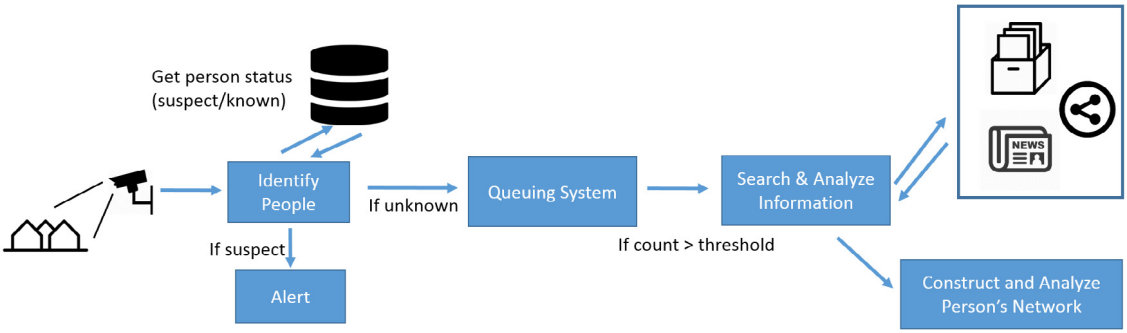

Fig. 1. The overall methodology.

a. An early warning system capable of highlighting potential threat and hence avoiding, preventing or at least reducing the possibility of attacks leading to disaster.
b. Capturing a face image and locating related information from available sources, including social media, archive, traditional media, etc.
c. Capturing and analyzing text, whether tweets or traditional documents, to identify some relevant terms, keywords, entities, etc.
d. Improving face recognition rates by re-training models with pre-processed face images
e. Employing machine learning, image processing, text analysis, and social media analysis techniques in a fully working system to help authorities in handling cases related to terror and criminology.

The rest of this paper is organized as follows. Section 2, covers our proposed security surveillance system with detailed information about how each part works so that we can perform person identification on images and identify potential suspects who were unknown before. We describe the dataset we used to evaluate the performance of the system to correctly identify people in Section 3. Section 4, provides experimental results. Section 5 is conclusions.

2. Methodology

The overall methodology of the proposed surveillance system is shown in Fig. 1. First, security cameras are used to monitor an area of interest whether it is a campus location, military base, street, etc. The collected images from these cameras are then fed to the "Identify People" process which applies machine learning techniques to extract face images which will then be compared with a database to check whether the person in the image is known or is a suspect. If the person's identity was inferred to be a suspect by the face recognition model, an alert will be raised by the system for a security officer to be warned in order to take a corresponding action. If the person is identified as known (safe) then no further action is taken by the system. Whereas if the person's face image was not matched with the known/suspect database, the face image is added to the "Queue". The idea behind the queue system is that not for every person passing through a scene the system should apply further investigation to collect information about him/her. Instead, an image is added to the queue. Every person in the queue has a counter which is incremented every time the person passes through the scene. More details about the implementation of the queue will be explained in Section 2.4 The "Search for Information" module will be used to further investigate every person whose counter passes a certain threshold. This module takes the person's image and compares it with existing social media profiles in order to match the person with his/her profile. If the social media of the person is successfully matched, analysis of the social media posts and the person network is applied to classify this person as a potential suspect or as a safe person. Details related to the implementation and functionality of each module are explained below.

2.1. Camera system

In our system, we make it possible to connect a local security camera on the network for live analysis. We also provide a functionality to get already recorded camera feed and import it to our system so that we can apply the same analysis on a crime incident.

Fig. 2 shows how the user can use the system to upload the content of a camera or connect to a live camera. The officer can upload several video sets or connect to a live camera feed from the network. The officer can also go back and check previously analyzed video stream where the video sequence is shown and people passing are labeled with their face image and detected identity. For our evaluation purposes we use the Chokepoint dataset which contains images collected using surveillance cameras.

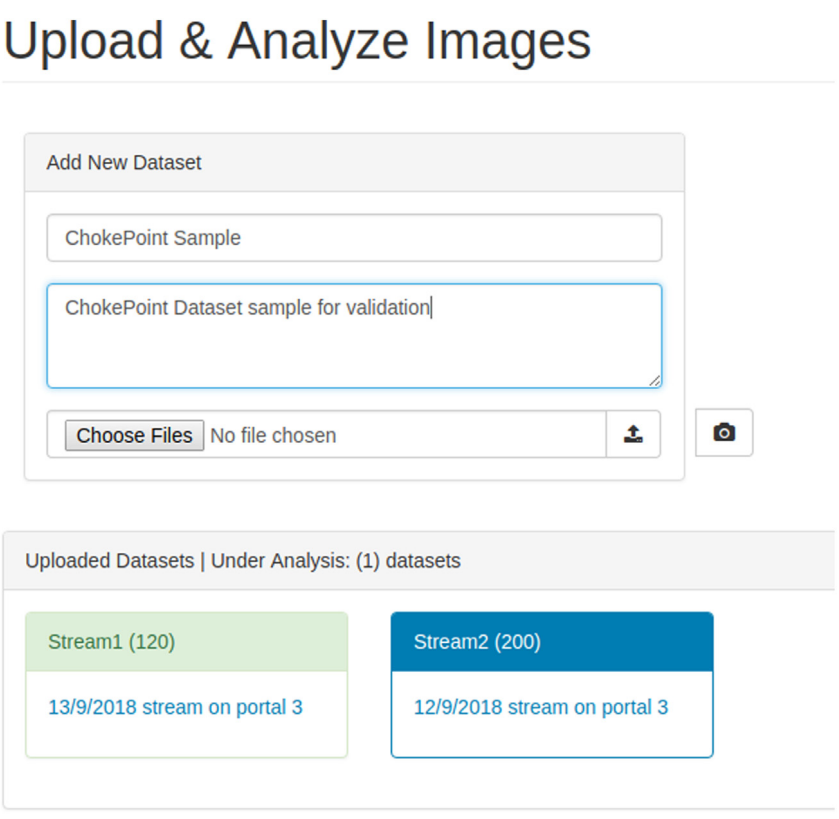

Fig. 2. Interface for security officers to upload or analyze a video stream.

2.2. Identify people

After connecting a camera to the system or uploading camera images to the system, the system will then run through the images to identify people present in the frames. There are two processes in identifying people, first the face image should be extracted and located in the frames, this process is referred to as face detection. The second process is to use the extracted face to match with other faces of suspects and known people available in the gallery. Matching face images is known as face recognition. Details on what is used for both face detection and recognition is described in the sub sections below.

(1) Face Detection: Face detection has been extensively studied by the research community for the past two decades. Early face detection methods such as Viola and Jones [15] and HOG [16] provided fast and accurate face detection for faces taken in a controlled and frontal environment. Viola and Jones method uses boosted cascade detectors and Haar features to locate face regions. While the Histogram of Oriented Gradient (HOG) [16] works by first dividing an image into grid cells. It then computes the feature vector of the image using the gradient descriptor. However, these methods fail to detect faces in a multi face environment where faces are collected from surveillance cameras and can be shown in any pose and in a complex background.

Many recent methods for face detection, however, make use of the recent advances in the graphic processing units (GPU) to learn complex models that better represent a face. Recent techniques for face detection use deep convolutional neural networks as the architecture to detect and extract features of faces. The success of using deep neural networks in speech recognition and image classification motivated their application for face detection. One of the first to use deep convolutional neural networks (DCNN) for face detection was the work done by Zhang et al. [17], where they collected many different face images from a variety of datasets containing different poses of faces. Around 120,000 faces were collected and then they trained a DCNN with 4 layers on these datasets. Farfade et al. [18] then fine-tuned the AlexNet [19] convolutional neural network (CNN) by training the network with face images. Their method recorded a receiver operating characteristic (ROC) value of around 80% on the FDDB dataset [20]. Other researchers worked on general object detection by using region proposal systems to get an object candidate image which is then used by the CNN model to verify if it is a face [21–23].

The work described in [21] proposed a region-based CNN called R-CNN. The first step of this method is to extract regions where each region may contain an object desired to be detected using a region proposal method such as Selective

Fig. 3. Image manipulation for face detection training. (a) shows an original image in the WIDER dataset. (b) shows a rotated image of (a). (c)–(g) shows the image at gamma levels (0.5, 1.5, 2, 2.5, 3).

Search [24] or EdgeBox [25]. Each extracted region is then warped and fed into the trained CNN model which decides whether the region contains the desired object or not. Then the method was later improved and called the Fast R-CNN in [22] to make the detection process faster than the original R-CNN by forwarding the entire test image only once to the CNN instead of every extracted region. Faster R-CNN was then proposed in [23] to make the whole process even faster by incorporating the region proposal method into a convolutional layer. This way, they will not use any external method that will slow down the process. Faster R-CNN has been applied on the face detection problem in [26].

For our system, face detection should be a fast process such that it can be applied it in real time analysis on a camera stream. It should be also accurate in a multi-view environment as people passing through can have their faces in different poses. For this purpose, we trained a CNN model to perform face detection on the WIDER face dataset [27]. The WIDER dataset contains thousands of face images collected under extreme cases varying scale, pose, occlusion and illumination of faces. For the CNN model, we used the recently proposed MobileNet-v1 [28] network architecture for training using the WIDER dataset. MobileNet-v1 is designed using depth-wise separable convolutions providing drastic decrease in model size and training/evaluation times while performing better in detection making it a perfect architecture for our purpose. To get better detection accuracy, we pre-process every image in the training data of the WIDER dataset before we train our detector model. We generate four different pictures for every image in the dataset and then feed the images to the learning model. The four different image types we generate from every image are shown in Fig. 3.

(2) Face Recognition: After the face location is extracted by the face detection process, the system is expected to identify the person by comparing his/her image with those in the database. Face recognition is done by first extracting from the face image relevant features which preserve the identity of the person such that two face images of the same person have similar values in their feature vector. Many feature extraction techniques have been proposed for face recognition. Early methods are classified as hand-crafted features because these algorithms follow predefined steps to locate and extract features from part of the image. Feature extraction methods such as SIFT [29], SURF [30], and LBPH [8] have been mentioned in several papers, e.g., [31–34], to extract features of faces that are then used to identify person's identity. Recent methods for feature extraction are based on training a model to automatically extract relevant features of faces; these methods are categorized as learned features.

Taigman et al. introduced DeepFace [35] which is a deep face recognition system developed by the research group at Facebook. They make use of a neural network model to learn face representations from large training datasets (order of millions). Before training their model on the images, they implement 3D modeling for all faces as a pre-processing step to

align faces to get a better face representation. Schroff et al. created the FaceNet [12] system designed by the research group at Google. Like DeepFace, their model is trained on millions of private images where faces are taken in an uncontrolled environment. The difference however is that FaceNet does not use any kind of 2D or 3D alignment. Instead, they applied simple scaling and translation techniques on the images. The method incorporates the triplet loss learning technique on each learning step of the neural network so that the representation is a vector where the Euclidean distance between the vectors is the distance between the images. Both DeepFace and FaceNet achieve a state-of-the-art accuracy results on face identification on the LFW dataset $\tilde{9}7\%$. For this work, we opted to use two different feature extraction techniques based on training CNN and compare which method is better for face recognition in surveillance camera type of images. The first extraction technique we use is from the popular OpenFace's [36] implementation of the FaceNet feature extraction technique. While FaceNet have trained their neural network model with over 200 million private images not available for the public, OpenFace trained their model with around 500 thousand images from public datasets and they provide their trained model for research purposes. OpenFace implements the triplet loss learning suggested by the FaceNet work in the feature learning process. The second feature extraction technique we use is the one we created by training our own CNN. The model architecture we used is the Inception-Resnet-v1 [37] network architecture and trained on the MS-Celeb-1M [38] face dataset. The training implementation also follows the method as in FaceNet [12] using the triple loss learning technique for our training.

2.3. Alert

After the person face is identified from the previous step, then the person will be classified as either known, unknown or a suspect. If the person is known then the system does no further action. But, if the person is unknown then the face image is sent to the Queue System for further analysis. An alert is generated by our system to the security personnel to act on the spot in case the person is identified as a suspect for being listed in the database.

2.4. Queuing system

When a person face image is classified as unknown, his/her face image is added to our queuing system. The purpose of the queue is to monitor the trend in which random people are arriving to the queue. For a random/unknown person who passes a number of times in a scene, we should apply further investigation to gather more information about the person to check if he/she is safe or not. The number of passes of the same person is deemed suspicious depends on the situation and specific circumstance where the surveillance camera is set. For example. if the surveillance camera is deployed in a military base, then the threshold for the number of times a random person passes should be lower than when a camera is deployed on a public street. In general, this threshold is set by security experts depending on the location.

Not every unknown person will have his/her face analyzed because in real life scenarios many people pass via a scene and never appear again. That is why a threshold is defined by a security expert to set a reasonable number of times that a person has to pass in order for our system to apply further analysis. Also, the queue cannot store every person face image that passes especially if the surveillance system is deployed in a busy area where large number of people pass. In such cases most people in the queue only pass one time and take unnecessary space in the queue. The queue size should have an upper bound set by the security expert depending on the monitoring location.

When the queue is full, a face replacement policy should be implemented such that the new face can enter the queue and one face instance will be removed from the queue. The decision of which queue element to discard is up to the replacement policy to decide on. Many replacement policies have been proposed for queuing systems coming from web cache replacement policies which are mainly used to manage cache content for web pages. Cache is an important aspect of the web to reduce loading times for web pages. A cache server stores Web objects such as HTML pages, images, and other files locally to be used for future requests. As the cache size of a browses is finite, a cache replacement policy is needed to manage cache content. The goal of the replacement policy is to make the best use of available resources such that we do not want popular items to leave the queue. Even recently added items might become popular in the near future. In our case, we treat face images like cache objects of a web page. Traditional replacement policies such as least recently used (LRU) and least frequently used (LFU) were proposed. More recent proposed solutions [39–41] provide only slight improvements and variations of these early methods. But, actually there is no single policy that performs best in all environments. It depends on the application in place [42]. The LFU method is a frequency-based policy which uses the count of an object solely to decide where the item will rank in the queue. The higher the count of an item is the higher it is in the queue. Items with the lowest count will leave the queue when new items arrive. The other type of cache replacement policy is LRU which where items that have been used least recently will be removed from queue regardless of how popular they were.

There are problems with both LRU and LFU. LRU does not take into account the usability of the item where the most accessed object can be evacuated from the queue. While the problem with LFU is that it ignores the latest item accessed which can be evacuated right after its addition because of its low frequency and may not take the chance to increase its value. A better approach will be combining both the frequency and the recency of an item for the removal policy. For this purpose, we implemented our own replacement policy for our queuing system to consider both the recency and the frequency of a person passing. This way, every time a person passes we increase his/her frequency and note down the

time of the passage. When the queue is full, the removal policy is not only based on the person with least frequency score but based on the time period the item has been in the queue unreferenced. For every x mns passes (ex: 30 to be set by security expert), the item loses one frequency score so that the most recent item will not be removed.

The algorithm to add a new face item to the queuing system is shown in Fig. 1. We describe the variables and functions in the list below.

- *x*: Face image of the unknown person.
- *element*: An element in the queue refers to a face image of an unknown person who already has a count and time of arrival in the queue.
- *getDist(image, element)*: A method that takes as input a face image and an element from the queue to calculate the distance between the two feature vectors of the face images.
- *distThreshold*: A variable that decides whether a face image belongs to the same person or not. If the distance between two face images is less than distThreshold, then the two images belong to the same person. (we set the threshold to 1.1 in our experiments)
- *getCurrentTime()*: A method that gets the current time, used to record the entry of a face image or to update the last time an element got referenced.
- *addCount(element, time)*: A method that takes as input an element and increments the hit counter of that element. The method also takes as input the current time to update the last time this element was referenced.
- *maxCount*: A variable defined by the security expert depending on the environment where the surveillance camera is installed. Further investigation is applied on an element if its count is greater than this variable.
- *investigate(element)*: A method that applies further investigation to collect information about the input element. The element is also removed from the queue to make space for new elements. Further details on the investigation process can be found in Section 2.4.
- *removeElement()*: Removes an element from the queue to make space for a new item. The process of removal takes into account the count of an element
- *addElement(image, time)*: A method that adds a new element to the queue with count equal to 1 and time equal to the current time.

Algorithm 1 Add Face to Queue

```
1: procedure AddFace(x)
2:     for element in queue do
3:         dist ← getDist(x, element).
4:         if dist < distThreshold then
5:             currentTime ← getCurrentTime().
6:             count ← addCount(element, currentTime)
7:             if count = maxCount then
8:                 investigate(element).
9:             return
10:    currentTime ← getCurrentTime().
11:    if queue is full then
12:        removeElement().
13:    addElement(x, currentTime)
```

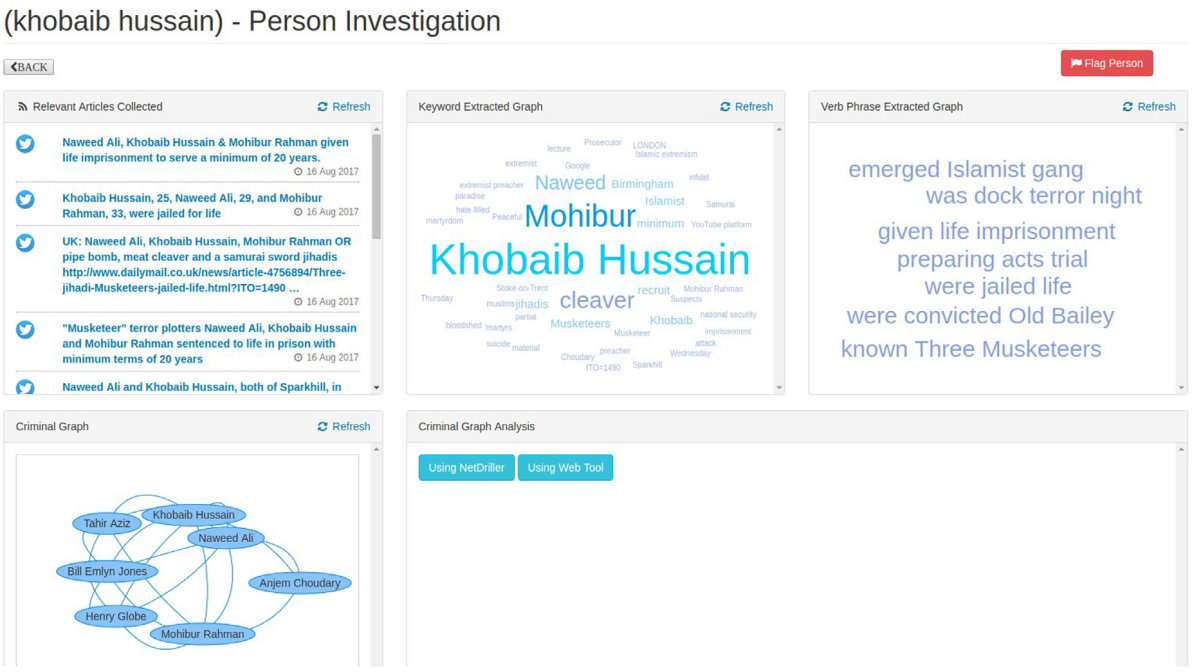

Fig. 4. Person Collected Information Details.

2.5. Search for information

If an unknown person in the queue reached maximum hits, his/her image is then used to identify the person using the social media. Terrorists and criminals has been shown over several studies [43,44] to use social media accounts to plan or to discuss criminal activities. For our system, we have collected thousands of Facebook profiles associated with people's profile picture. Facebook's Graph API[1] provides several functions to access Facebook's social graph. We used the API and generated random Facebook user IDs which gave us access to information of random people from Facebook. From this data of social network profiles who we collected, we built an SVM classifier model based on the collected images. Thus, when an unknown person image is collected from the surveillance camera and passed thorough the queue system, we compare the collected face with the social media profiles we have. If the face is matched with the social media profile, then we use the name of the social media profile to search more about the person using Twitter and traditional news. Fig. 4 shows an example of the page that an analyst is given once an unknown person has exceeded the pass number. We then get the person name from the social media profiles collected and then extract tweets, posts and news articles that mention this person name. We then automatically apply text analysis on the collected information to provide the analysts with relevant keywords and verb phrases that this person is mentioned in.

Keywords are extracted from the text using a method proposed by Mihalcea et al. [45] where they developed a term extractor called TextRank, which is a graph ranking based method applied on words as vertices in order to determine the importance of the words. For the tweets, however, we also used the hashtag as a topic of the tweet because social media posts are limited in number of words. After providing the analyst with the person name, articles mentioned with links, keywords, verb phrases and criminal network, the analyst can finally decide whether to flag the person. This will lead to the person being added to the suspect list if deemed so.

2.6. Construct and analyze person's network

After deducing the name of the unknown person from the search for information process, we can build a social graph of the person to check his/her network if it is safe or not. A social graph is defined such that nodes in the graph represent people and edges between these people indicate an interaction or relation between them. Research efforts, e.g., [46,47], have been done to populate a criminal graph of a person from his/her name using news articles. The procedure is done by first searching for articles in which the person is mentioned and then apply named entity recognition (NER) using Stanfornd NLP [48] to detect all other people names mentioned in the same articles. For every person name existing in the article, a vertex is added to the person social graph. For all people mentioned in the same article, an edge is added

[1] Facebook graph API: https://developers.facebook.com/docs/graph-api/ Accessedon9/6/2018.

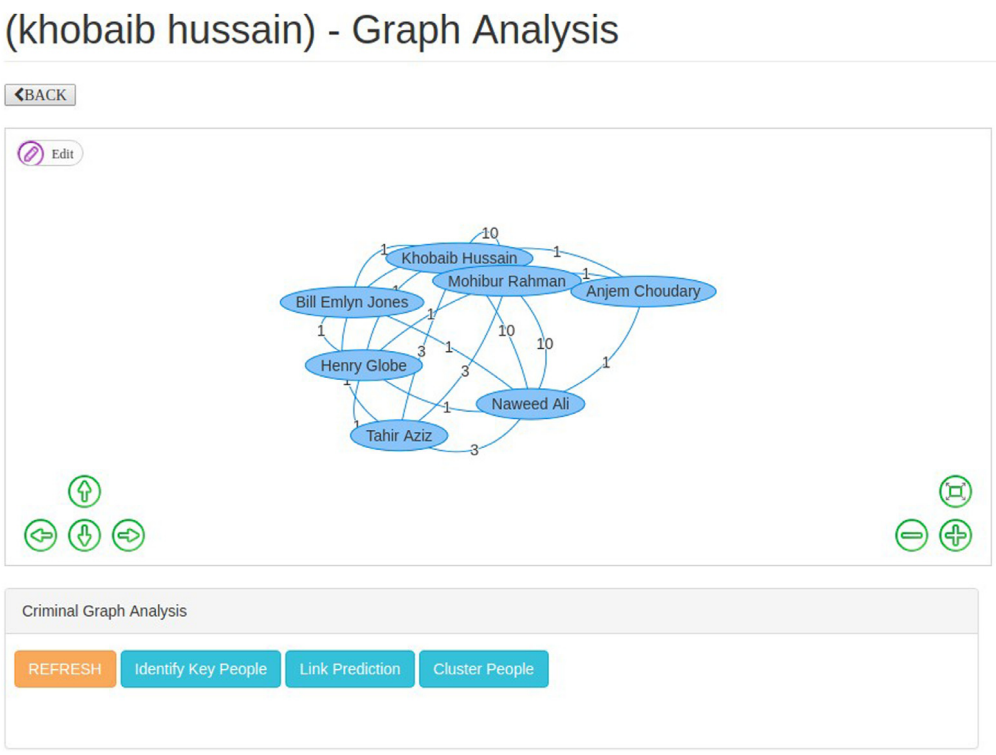

Fig. 5. Person Collected Information Details.

between them representing an interaction. If two people are mentioned in more than one article, then a weight is added to their edge to show the number of articles they were mentioned in. Modeling the social interactions and mentions in the text is an important mechanism for analysts as it allows network visuals to see a criminal network and different interactions in a clear way. This leads further investigation of other people in a network by applying network analysis techniques. In case a person has links with suspects then the analyst will cluster this person as a potential suspect and later appearances of this person in the surveillance cameras will raise an alert.

Fig. 5 shows the network graph of "Khobaib Hossain" as shown to the analysts. First, the names of the related people are extracted from the collected text where "Khobaib Hossain" is mentioned. We use Stanford NLP toolkit [48] to extract names from the text, the tool uses the method described in [49] to train a NER model using a combination of CRF sequence taggers trained on various text. The graph is then populated from these names and mentions in same articles.

We provide several graph analysis techniques for the analyst to apply further investigation on the graph. These are useful especially in large networks. We chose four different network analysis techniques in our system. These will provide enough information for analysts in their investigation.

- *Identify Key Nodes*: This process aims to identify what are the major and most influential criminals/nodes in the criminal graph. In graph theory, centrality indicators identify the most important vertices in the network. There are three main centrality measures that are calculated for every node in the graph. Degree Centrality which is defined as the number of links/edges a node has. A high degree centrality means that a criminal is mentioned and involved with many other criminals. Closeness Centrality is defined as the average length of the shortest path between a node and all other nodes in the graph. A high value of the closeness centrality refers to a criminal who is at the center of the network where he can easily reach all other criminals in the network. Betweenness Centrality is defined as the number of times a node acts as a bridge along the shortest path between two other nodes. Nodes with high betweenness value are the ones who have more control over information passing between other nodes. Removing these types of criminals will cutoff the linkage of the graph because other nodes rely on these criminals to reach other nodes. The formula of the betweenness value for a node v is given in Eq. (1). Where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of paths which pass through v. Eigenvector Centrality is a measure that identifies the most important and influential nodes in the network. The importance of the node comes if this node is linked to by other important nodes. To calculate such network measures, the analyst can click on the identify key people button to select his/her metric to calculate.

$$Betweenness(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{1}$$

	Algorithm	Description
Adamic/Adar	$\sum_{z\in\Gamma(u)\cap\Gamma(v)} \frac{1}{log\lvert\Gamma(z)\rvert}$	This index measures similarity with counting of common neighbors z between nodes u and v by weighing the less-connected or rare neighbors more.
Jaccard	$\frac{\lvert\Gamma(u)\cap\Gamma(v)\rvert}{\lvert\Gamma(u)\cup\Gamma(v)\rvert}$	Common neighbors are divided by total number of neighbors of u and v. It looks for uniqueness in the shared neighborhood.
Dice	$\frac{2\lvert\Gamma(u)\cap\Gamma(v)\rvert}{\lvert\Gamma(u)\rvert+\lvert\Gamma(v)\rvert}$	Common neighbors are divided by their arithmetic mean. It is a semimetric version of Jaccard.
Katz	$\sum_{\ell=1}^{\infty} \beta^{\ell} \cdot \left\lvert paths_{u,v}^{\ell} \right\rvert$	This index looks for path lengths and counts by weighting shorter paths between nodes more heavily. The parameter $\beta \in [0, 1]$ controls the contribution of paths. And ℓ represents the length between nodes. Smaller values for β will decrease the contribution of higher values for ℓ.

Fig. 6. Link Prediction techniques.

- *Adjust Network*: After identifying key nodes in the network using the previous process, it is essential to give an analyst the option to view the criminal network and modify the existence of some nodes depending on their importance to see what effect they have on the network. By removing criminals from the network, the analyst can look into how to disrupt the criminal network structure so that they can arrest these persons to possibly collapse the network. This can be done using the edit button on the graph to add/delete/update nodes and edges.
- *Cluster Nodes*: We provide hierarchical clustering which aims to show the criminal network as a set of communities. It is essential for an analyst to view the network as a set of communities because it generally infers what criminal groups reside within a network. We provide a hierarchical community view as a functionality for the analyst using our framework which provides the option to zoom in and out of the network using community detection algorithms and display each community as a node in a zoom out mode and display each node as a community in a zoom in mode. This is provided by the cluster nodes button.
- *Link Prediction*: Link prediction refers to the problem of mining what links between nodes in the criminal graph created may exist without our knowledge or which new interactions among its members are likely to occur in the near future. Finding hidden links in a criminal graph is very important because we can predict an interaction between two criminals by analyzing the graph structure using social network analysis techniques. Many link prediction methods have been proposed for this purpose, e.g., [50–52]. This analysis technique is provided using the link prediction button. Fig. 6 lists some of the link prediction algorithms implemented in our framework.

3. Dataset

To evaluate our system to identify and classify people into suspects, known, and unknown, we chose to use the Chokepoint dataset [53] for our experiments. The Chokepoint dataset is a video based dataset designed for experiments on identifying and verifying people's identity under real-world surveillance conditions. The dataset is collected using an array of 3 cameras above several portals to capture people walking through each portal in different face views (frontal/profile). The dataset consists of 25 subjects (19 male and 6 female) in portal 1 and 29 subjects (23 male and 6 female) in portal 2. In total, it consists of 48 video sequences and 64,204 face images. Each set has variations in terms of illumination conditions, pose, sharpness and has been taken in different times of the day to make the dataset more challenging. Sequence names are unique and correspond to the recording conditions, where P, S, and C stand for portal, sequence and camera, respectively. Further, E and L indicate subjects either entering or leaving the portal, respectively. The dataset environment of surveillance cameras is similar to those observed at airports [54] where individuals pass in a natural free-flow way in a narrow corridor.

A sample of the dataset is shown in Fig. 8. Fig. 7 shows gallery images of photos taken for every subject who passes through the portals. There are two images taken for every person, one of them with neutral face while the other one with a smile. These images are used as database images in our system to specify people as suspects or known from these images. Fig. 8 shows a sample of the video images collected from different ports and the three different camera angles for every sequence; it is used for validating the system.

4. Experiments & results

For our experiments, we used face images collected in the gallery settings from the Chokepoint dataset as our database images. We did two separate experiments, in the first one we used gallery images consisting of only neutral face images. In the another experiment, for every person there are two gallery images (neutral and smile images). For our experiments, we considered people (1, 2, 3, 4, 5, 6) as known while people (7, 9, 10, 11, 12, 13) as suspects. The rest of the people are deemed to be unknown and gallery images of these people were not used.

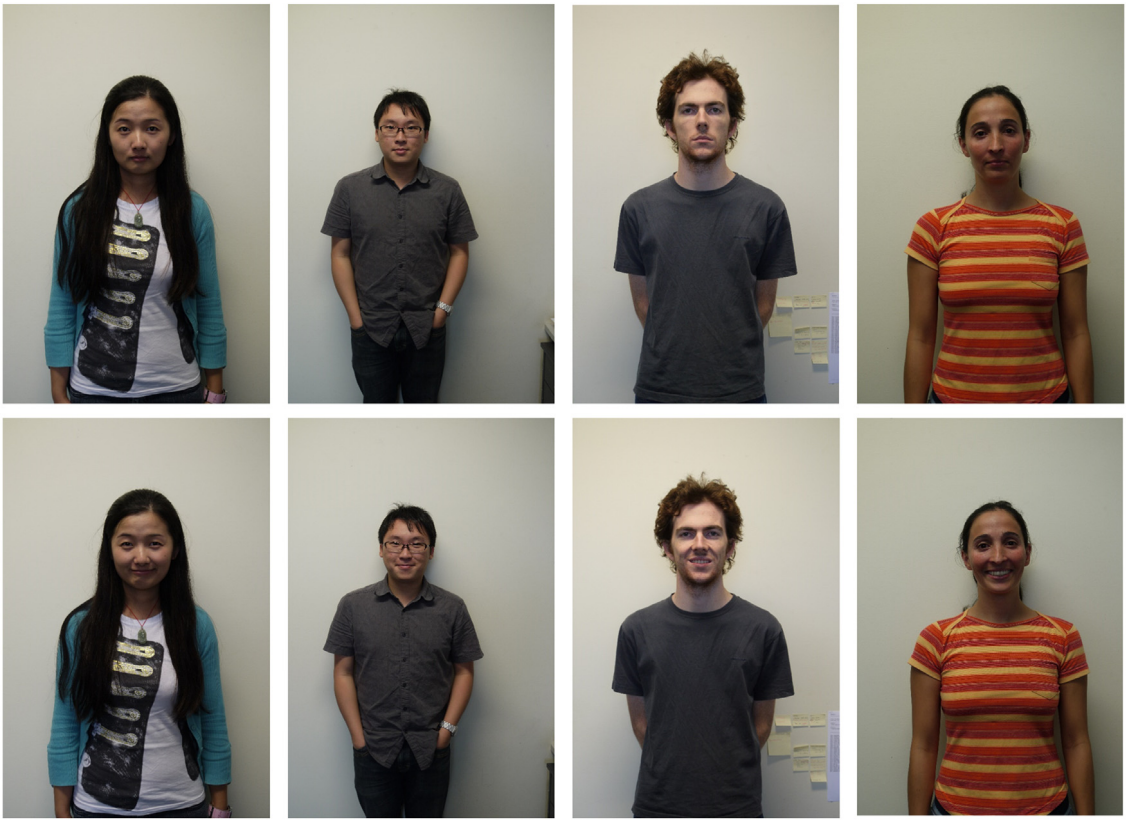

Fig. 7. Sample of gallery images (smile and neutral) of the Chokepoint Dataset.

Fig. 8. Sample of the video images collected from the Chokepoint dataset.

For video images in the dataset, we did not apply our face detection on images. The reason is that in the ground truth of the dataset, they do not include face images of far people. Thus, we could not evaluate our recognition accuracy with the ground truth, where Fig. 9 shows a sample of the face detection process of our model; it reports two faces detected

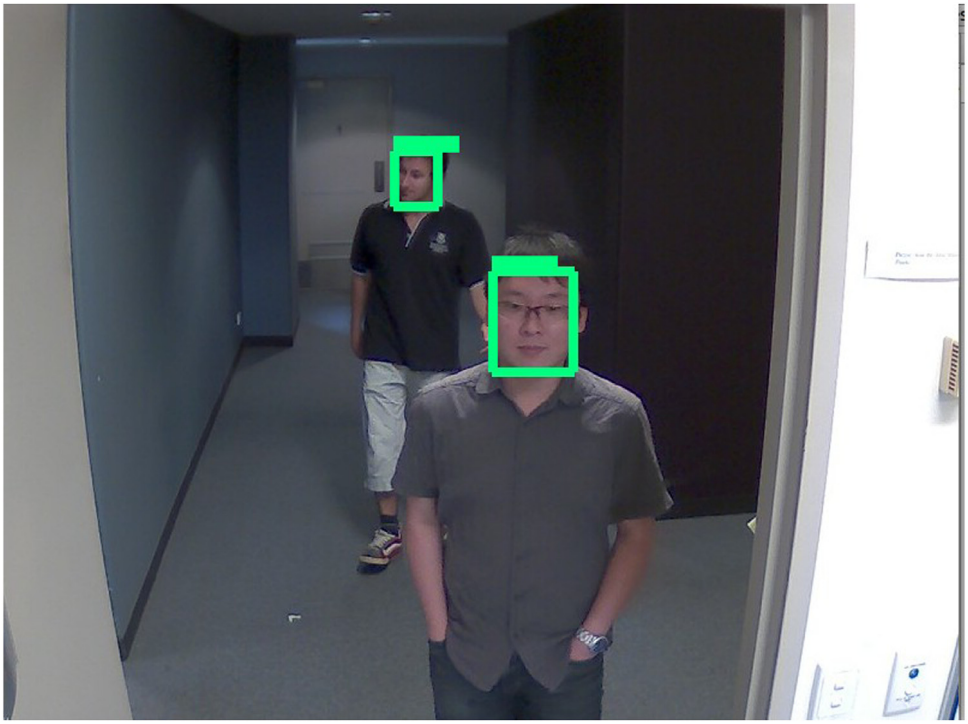

Fig. 9. Sample of the output of our face detector.

Table 1
Accuracy results of the known, suspects and unknown people using the P1L-S3-C1 camera sequence.

P1L_S3_C1	KnownAcc	SuspectAcc	UnkownAcc
openFaceNeutral	64.07	72.57	65.10
openFaceNeutralSmile	66.12	72.93	67.96
ourNeutral	81.82	80.70	78.11
ourNeutralSmile	84.85	82.67	80.47

Table 2
Accuracy results of the known, suspects and unknown people using the P1L-S3-C2 camera sequence.

P1L_S3_C2	KnownAcc	SuspectAcc	UnkownAcc
openFaceNeutral	64.53	66.30	66.32
openFaceNeutralSmileNew	69.19	67.00	71.28
ourNeutral	83.05	81.40	78.34
ourNeutralSmile	86.44	83.60	80.41

in the image. The ground truth of the dataset, however, defines only one face, namely the frontal one. Instead, for every image frame in the ground truth that shows a person, we applied our different feature extraction technique on the face.

First, we extracted feature vectors of all face images from camera feed and gallery images. We used OpenFace model and our trained model to collect two separate feature vectors for every image to compare which feature extraction model works better in a surveillance environment. Recall that we had specified people with IDs (1, 2, 3, 4, 5, 6) as known, people (7, 9, 10, 11, 12, 13) as suspects and the rest as unknown. We then compare the accuracy of recognition in two different settings, one with only one face of a person in the gallery (neutral face) compared to when we have two faces in the gallery for every person (neutral and smiling face). After setting up the database gallery, we ran our tests on each portal and sequence with the three different cameras available. Tables 1, 2, 3 show accuracy results of using the different recognition models to classify people passing in portal 1 sequence 3 into known, suspects and unknown people. On each row, we have the type of the model we used for accuracy; it is either OpenFace or our model with each having neutral face or neutral and smile face in the database gallery.

We only show accuracy results of camera sequence of P1L S3 because we got similar accuracy results for all other camera sequences. As shown in Tables 1, 2, 3 the accuracy results across different camera angles for the same sequence do not hugely affect the recognition rate of our trained neural network model, where the accuracy for known people

Table 3
Accuracy results of the known, suspects and unknown people using the P1L-S3-C3 camera sequence.

P1L_S3_C3	KnownAcc	SuspectAcc	UnkownAcc
openFaceNeutral	64.85	66.74	66.28
openFaceNeutralSmileNew	75.25	67.95	70.00
ourNeutral	81.01	80.30	78.06
ourNeutralSmile	82.28	81.20	80.59

using openfaceneutral model is around 65% for the three camera angles. Same findings are observed for the other models where the accuracy is almost the same through the different camera angles. Camera 2 shows slightly better results. Also we show that our approach works better than OpenFace feature extraction technique by almost 15%. Further, using two face images in the gallery slightly enhances recognition rate. This means that even using one face image in the gallery produces good result.

5. Conclusions

We present in this paper an early warning system that integrates face recognition, social media and text analysis for recognizing people in surveillance camera environments. Monitoring people in surveillance systems is being used for security purposes where security officers have to manually watch suspicious people or activities. Many scenarios happen when security officers cannot recognize well people in a surveillance environment as shown in many previous research efforts, e.g., [1–5]. Even a person who passed in front of a camera might be a potential suspect who the system does not know about. We propose a system that first takes as input image frames from surveillance cameras. These images are then used to locate and recognize people based on their faces. The system maintains a database of known and suspicious people to raise an alarm for security officials when a suspect is shown in a scene. When a person identity is unknown, his/her image is added to a queuing system. The same people passing a number of times will be then forwarded for further investigation to know who they possibly are and if they are dangerous. A person face image that has been forwarded by the queuing system will then have his/her face compared with social media profile images collected from Facebook. If the social media profile is found in the database, the name of the person is used to collect more information and text from news and other social media profiles. The result is used for text analysis which is applied to get important sentences and people mentioned with a given person. This leads to construct social graph of the person. Using our tool, the analysts can then use a variety of network analysis tools to identify important people in the network and check if this person is suspicious or not. We show by the conducted experiments that using our trained neural network provides good accuracy levels in recognizing people compared to other approaches in a surveillance camera environment. Finally, we are currently investigating how to develop the system further to reduce the runtime and improve the accuracy by trying techniques other than those already used in the current implementation.

References

[1] Vicki Bruce, Zoë Henderson, Karen Greenwood, Peter J.B. Hancock, A. Mike Burton, Paul Miller, Verification of face identities from images captured on video, J. Exp. Psychol.: Appl. 5 (4) (1999) 339.
[2] Zoe Henderson, Vicki Bruce, A. Mike Burton, Matching the faces of robbers captured on video, Appl. Cogn. Psychol. 15 (4) (2001) 445–464.
[3] Vicki Bruce, Zoë Henderson, Craig Newman, A. Mike Burton, Matching identities of familiar and unfamiliar faces caught on cctv images, J. Exp. Psychol.: Appl. 7 (3) (2001) 207.
[4] Ahmed M. Megreya, A. Mike Burton, Unfamiliar faces are not faces: Evidence from a matching task, Mem. Cogn. 34 (4) (2006) 865–876.
[5] A. Mike Burton, Stephen Wilson, Michelle Cowan, Vicki Bruce, Face recognition in poor-quality video: Evidence from security surveillance, Psychol. Sci. 10 (3) (1999) 243–248.
[6] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, et al., Deep face recognition, in: BMVC, Vol. 1, 2015, p. 6.
[7] Matthew A. Turk, Alex P. Pentland, Face recognition using eigenfaces, in: In Computer Vision and Pattern Recognition, 1991 Proceedings CVPR'91, IEEE Computer Society Conference on, IEEE, 1991, pp. 586–591.
[8] Timo Ahonen, Abdenour Hadid, Matti Pietikainen, Face description with local binary patterns: Application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[9] Yan Ke, Rahul Sukthankar, Pca-sift: A more distinctive representation for local image descriptors, in: Computer Vision and Pattern Recognition, 2004 CVPR 2004 Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 2, IEEE, 2004, p. II.
[10] Rama Chellappa, Charles L. Wilson, Saad Sirohey, Human and machine recognition of faces: A survey, Proc. IEEE 83 (5) (1995) 705–741.
[11] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, Hong-Jiang Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.
[12] Florian Schroff, Dmitry Kalenichenko, James Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[13] Yandong Wen, Kaipeng Zhang, Zhifeng Li, Yu Qiao, A discriminative feature learning approach for deep face recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 499–515.
[14] Yi Sun, Ding Liang, Xiaogang Wang, Xiaoou Tang, Deepid3: Face recognition with very deep neural networks, 2015, arXiv preprint arXiv: 1502.00873.
[15] Paul Viola, Michael Jones, Rapid object detection using a boosted cascade of simple features, in: Computer Vision and Pattern Recognition, 2001 CVPR 2001 Proceedings of the 2001 IEEE Computer Society Conference on, Vol. 1, IEEE, 2001, p. I.
[16] Navneet Dalal, Bill Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005 CVPR 2005 IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 886–893.

[17] Cha Zhang, Zhengyou Zhang, Improving multiview face detection with multi-task deep convolutional neural networks, in: Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on, IEEE, 2014, pp. 1036–1041.
[18] Sachin Sudhakar Farfade, Mohammad J. Saberian, Li-Jia Li, Multi-view face detection using deep convolutional neural networks, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, ACM, 2015, pp. 643–650.
[19] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[20] Vidit Jain, Erik G. Learned-Miller, Fddb: A Benchmark for Face Detection in Unconstrained Settings, UMass Amherst Technical Report, 2010.
[21] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[22] Ross Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[23] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[24] Jasper R.R. Uijlings, Koen E.A. Van De Sande, Theo Gevers, Arnold W.M. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[25] C. Lawrence Zitnick, Piotr Dollár, Edge boxes: Locating object proposals from edges, in: European Conference on Computer Vision, Springer, 2014, pp. 391–405.
[26] Huaizu Jiang, Erik Learned-Miller, Face detection with the faster r-cnn, 2016, arXiv preprint arXiv:1606.03473.
[27] Shuo Yang, Ping Luo, Chen-Change Loy, Xiaoou Tang, Wider face: A face detection benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5525–5533.
[28] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv preprint arXiv:1704.04861.
[29] David G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[30] Herbert Bay, Tinne Tuytelaars, Luc Van Gool, Surf: Speeded up robust features, in: Computer vision–ECCV 2006, 2006, pp. 404–417.
[31] Cong Geng, Xudong Jiang, Face recognition using sift features, in: Image Processing (ICIP), 2009 16th IEEE International Conference on, IEEE, 2009, pp. 3313–3316.
[32] Manuele Bicego, Andrea Lagorio, Enrico Grosso, Massimo Tistarelli, On the use of sift features for face authentication, in: Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on, IEEE, 2006, p. 35.
[33] Geng Du, Fei Su, Anni Cai, Face recognition using surf features, in: Sixth International Symposium on Multispectral Image Processing and Pattern Recognition, International Society for Optics and Photonics, 2009, 749628.
[34] Chi-Ho Chan, Josef Kittler, Kieron Messer, Multi-scale local binary pattern histograms for face recognition, in: Advances in Biometrics, 2007, pp. 809–818.
[35] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.
[36] Brandon Amos, Bartosz Ludwiczuk, Mahadev Satyanarayanan, Openface: A General-Purpose Face Recognition Library with Mobile Applications, Technical Report, CMU-CS-16-118, CMU School of Computer Science, 2016.
[37] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alexander A. Alemi, Inception-v 4, inception-resnet and the impact of residual connections on learning, in: AAAI, Vol. 4, 2017, p. 12.
[38] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, Jianfeng Gao, MS-celeb-1M: A dataset and benchmark for large scale face recognition, in: European Conference on Computer Vision, Springer, 2016.
[39] Song Jiang, Xiaodong Zhang, Lirs: An efficient low inter-reference recency set replacement policy to improve buffer cache performance, ACM SIGMETRICS Perform. Eval. Rev. 30 (1) (2002) 31–42.
[40] Donghee Lee, Jongmoo Choi, Jong-Hun Kim, Sam H. Noh, Sang Lyul Min, Yookun Cho, Chong Sang Kim, Lrfu: A spectrum of policies that subsumes the least recently used and least frequently used policies, IEEE Trans. Comput. 50 (12) (2001) 1352–1361.
[41] Donghee Lee, Jongmoo Choi, Jong-Hun Kim, Sam H. Noh, Sang Lyul Min, Yookun Cho, Chong Sang Kim, On the existence of a spectrum of policies that subsumes the least recently used (lru) and least frequently used (lfu) policies, in: ACM SIGMETRICS Performance Evaluation Review, Vol. 27, ACM, 1999, pp. 134–143.
[42] Kin-Yeung Wong, Web cache replacement policies: A pragmatic approach, IEEE Netw. 20 (1) (2006) 28–34.
[43] Jytte Klausen, Tweeting the jihad: Social media networks of western foreign fighters in syria and iraq, Stud. Confl. Terror. 38 (1) (2015) 1–22.
[44] Emily Goldberg Knox, The slippery slope of material support prosecutions: Social media support to terrorists, Hastings Law J. 66 (2014) 295.
[45] Rada Mihalcea, Paul Tarau, Textrank: Bringing order into text, in: EMNLP, Vol. 4, 2004, pp. 404–411.
[46] Tarique Anwar, Muhammad Abulaish, A social graph based text mining framework for chat log investigation, Digit. Invest. 11 (4) (2014) 349–362.
[47] Hsinchun Chen, Wingyan Chung, Jennifer Jie Xu, Gang Wang, Yi Qin, Michael Chau, Crime data mining: A general framework and some examples, Computer 37 (4) (2004) 50–56.
[48] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, David McClosky, The stanford corenlp natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60.
[49] Jenny Rose Finkel, Trond Grenager, Christopher Manning, Incorporating non-local information into information extraction systems by gibbs sampling, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2005, pp. 363–370.
[50] Mark E.J. Newman, Clustering and preferential attachment in growing networks, Phys. Rev. E 64 (2) (2001) 025102.
[51] Lada A. Adamic, Eytan Adar, Friends and neighbors on the web, Soc. Netw. 25 (3) (2003) 211–230.
[52] Salim Afra, Alper Aksaç, Tansel Özyer, Reda Alhajj, Link prediction by network analysis, in: Prediction and Inference from Social Networks and Social Media, Springer, 2017, pp. 97–114.
[53] Yongkang Wong, Shaokang Chen, Sandra Mau, Conrad Sanderson, Brian C. Lovell, Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on, IEEE, 2011, pp. 74–81.
[54] Eric Granger, D. Gorodnichy, Evaluation Methodology for Face Recognition Technology in Video Surveillance Applications, Defence R & D Canada, 2014.