Applying Machine Learning and Data Fusion to the "Missing Person" Problem

K.M.A. Solaiman, Purdue University
Tao Sun, Massachusetts Institute of Technology
Alina Nesen and Bharat Bhargava, Purdue University
Michael Stonebraker, Massachusetts Institute of Technology

We present a system for integrating multiple sources of data for finding missing persons. It can help authorities find children, developmentally challenged individuals who have wandered off, and persons of interest in investigations.

There are many circumstances in which the "missing person" problem arises. They include missing children alerts, family reunification during natural disasters, prison escapes, and people who are unaccounted for. Missing person search works similarly for prison escapees, adults with cognitive problems, and children. The police have the same problem when they search for a person of interest involved in a crime, whether as a suspect or a victim. For each situation listed here, the authorities have a physical description of a person (for example, a white male with a medium build, wearing a blue shirt and black jeans).[1] Physical attributes are used as soft markers for person search.[2] Additional information about missing persons comes from families, Twitter posts, and phone calls from the public. Vehicle information can be corelated with Department of Motor Vehicles (DMV) records.

Digital Object Identifier 10.1109/MC.2022.3145507
Date of current version: 3 June 2022

 0018-9162/22©2022IEEE

Irrespective of the source, the information will have identifying features of the missing person, based on which the search is conducted. According to related works from *Policing and Society*, one of the first steps in dealing with missing person incidents is to search surveillance camera video footage from the vicinity. For example, West Lafayette, Indiana, has cameras in all city buses, on many intersections in the downtown area, in the majority of the local businesses, and in all police cars. Moreover, police officers are equipped with body cameras when on duty. Police in West Lafayette spend hours manually searching videos for missing persons.[1] Data fusion from these disparate sources would be a valuable addition for automatic information retrieval and querying.

In this article, we report on a system we built—Find-Them—to perform video capture, tweets and tips collection, feature identification, and information fusion among data sources. In Find-Them, we do not attempt to perform facial recognition, as video is low resolution, taken from afar, and the lighting conditions are usually poor (due to snow, rain, and darkness). Persons of interest may be in the background or facing away from a camera,[2] which makes relying on face recognition infeasible for our task. Instead, we focus on other features, such as gender, clothing (for example, baseball hats and shirts), and markings (such as tattoos).

In the absence of facial recognition, the system is not suited for identifying and tracking particular citizens but, instead, helps with localizing a group of people with similar attributes. While in general the system is not optimized to be used as a "digital spy" for the benefit of society, the government should restrict the use of this technology, confining it to the law enforcement tasks of searching for specific persons of interest. Furthermore, the objective of Find-Them is different from the task of entity matching. *Entity matching* refers to identifying data instances that refer to one real-world entity across data sources. Successful systems, such as Magellan, AutoEM, and CloudMatcher, focus on entity matching by names, whereas Find-Them aims to find entities by their physical features.

To begin, identifying a missing person is a data capture problem. As specified earlier, information about a missing person comes from multiple sources, such as surveillance cameras, tweets, family members, and previous occurrences. Storing these multimodal data is a problem at scale. Since information comes from several modalities and in large amounts, a proposed storage system should normalize different modalities of data at a large scale. Finding relevant information about a missing person from multimodal data requires system-specific property identification in each modality and context-based data integration for a composite query. As discussed by Stonebraker et al.,[1] training data for property identification are expensive to acquire, and input data from real-world applications often have noise. For the missing person problem, traditional deep learning methods are costly at scale since they require an enormous amount of specific training data. Thus, traditional machine learning methods may fail for the extraction of specific features for on-demand missing person identification. Finally, in real-world applications, there are terabytes of information. Therefore, any data fusion has to be done at a large scale while accommodating multiple sources.

Find-Them implements a streaming data capture and downsampling method to tackle the problem of multimodal data capture and storage. To achieve scalability, it loads raw data and acquired properties into a PostgreSQL database. Raw data and properties are separately stored between cold storage and an online property server to achieve speed and scalability. We propose modality-specific feature identifiers for video feeds, unstructured text, and tweets. In this work, we explain the feature extractors required for the missing person problem. For data fusion, Find-Them implements entity–attribute–relationship (EAR) schemas compatible with the application domain. Using features specified by the user, we built Structured Query Language (SQL) queries using the data description language. By performing these queries (for example, JOIN) across the standard schemas, Find-Them delivers multimodal results relevant to user interest. The fusion methodology in Find-Them is expandable to other modalities and different feature identifiers for the discussed modalities.

RELATED WORK

Missing person search is a significant real-world problem that draws on work in several areas of social and computational aspects.

Missing person search applications

Applications such as People Locator (PL),[3] Myosotis,[4] the National Missing and Unidentified Persons System (NamUs) (https://namus.nij.ojp.gov/), and Google Person Finder enable different levels of missing person search

and comparison. PL,[3] a search application for family reunification after disasters, combines multiple modalities for searching and reporting missing people, including structured web forms, app-based community reports (ReUnite), unstructured text from email, image-based hospital reports (TriagePic), and other applications with the People Finder Interchange Format data standard. Similar to Find-Them, for relevance matching, PL employs SQL query-based database search and Apache Searching on Lucene With Replication-based indexing and search string matching. However, it lacks face matching and multimodal searching. NamUs facilitates 1) searching to match the demographics, descriptors, and distinctive characteristics of a missing person; 2) automatically comparing cases based on geography, dates, and physical features and helping to find connections and investigative leads; and 3) generating customized case maps. It has comparison and search functionality similar to Find-Them. Google Person Finder is a disaster time registry to post and search missing person statuses. Myosotis[4] aggregates data from heterogeneous missing people databases, enables visualization via interactive maps, and infers an estimation of the probability of a new occurrence. NamUs, Google Person Finder, and Myosotis do not support multimodal and on-demand search from streaming heterogeneous data.

Person reidentification

Person reidentification refers to searching for a person in video feeds through a textual or an image query. Existing person reidentification methods use supervised and unsupervised learning[1,5] techniques. Identity-aware annotations[6,7] and zero-shot learning have increased the matching performance among image and text descriptions for person reidentification by using text attribute queries. Attribute recognition in the preceding models requires a substantial number of training samples. Multimodal search differs from person reidentification in the query response formats. Cross-modal search enables using different data modalities as queries and responses.

Khan and Jalal[8] augmented person reidentification with facial sketches by fusing facial attributes and semantic color information in attributes via a fuzzy rule-based layered classifier. Find-Them does not perform facial recognition; rather, it reidentifies a person via various semantic attributes, including color information. Methods[6,7] for text attribute extraction consider noun phrases as potential attribute values. Aggarwal et al.[6] filter candidate phrases by using associated images. Wang et al.[7] categorize noun phrases to specific attribute phrases, such as *upper body*, following a dictionary clustering approach. These approaches do not consider noise in streaming documents and the performance bottleneck of parts-of-speech (POS) taggers. They also do not differentiate among attribute names and values extraction.

Cross-modal matching and correlation learning

Most of the previous works[9–12] in multimodal matching have followed the idea of projecting features from different modalities into a shared embedding space by using modality-specific transformations. Rupnik and Shawe-Taylor[9] focus on correlation learning to glean linear projections by using pairwise information. In contrast, Zhang et al.[10] use pairwise and semantic information, for example, class labels, to learn the common subspace. Wang et al.[12] extend deep canonical correlation analysis with an autoencoder regularization term for nonlinear representations of multimodal data objects. Peng et al.[11] better encode intra- and intermodality correlation with hierarchical networks.

Some recent methods learn richer semantic representations for different modalities by using attention mechanisms,[13] graph representations,[14] and generative models[15] to build encoding networks. Deep relational similarity learning[16] avoids explicitly learning a common space by integrating relation learning, capturing the implicit nonlinear distance metric. While these learning methods exhibit good performance, mainly on bimodal data sets, they require a large amount of training data and do not scale well. Data representations lack generalization across multiple modalities and sources. Besides, many application domains already have prederived domain-specific features with fine-tuned feature learning methods, but the preceding models cannot integrate these sources. Moreover, current metric learning methods can integrate only user-specified data relevancy with training samples and class labels. The data fusion methodology in Find-Them focuses on solutions for the problems of scalability, a lack of annotations, and the use of preidentified features for data fusion.

Data fusion among multiple modalities has been employed in many application domains, such as sentiment analysis,[17] image–text matching,[14] face retrieval,[8] and visual question answering for a better understanding of context. These approaches have performed

well for their respective application domains, but they lack generalization capabilities. Similar to Find-Them, Palacios et al.[18] built a multimodal relational knowledge base by continuously querying for detected objects from videos and matching objects in text. However, their approach does not perform attribute-specific search and cannot be generalized for multimodal person search.

SYSTEM OVERVIEW

Figure 1 illustrates the architecture of Find-Them. It is divided into four modules: data ingestion, feature identification, relevance modeling, and data retrieval. Data ingestion deals with the problem of data capture and storage. The system captures streaming data and loads them into PostgreSQL at the server end after preprocessing. Feature extraction is done during load time by using type-appropriate models for each data source. Extracted properties are inserted into PostgreSQL, following the schema determined by the EAR model. The defined schema is used to create data integration among multiple sources during the relevance modeling phase. Users issue one-shot and standing queries to the system in the data retrieval phase. The ingestion and retrieval systems can operate in parallel. A user preference model is built from the query history and used in conjunction with the relevance model for data retrieval.

Data ingestion

Data capture. In Find-Them, we employ a streaming data capture system for video, unstructured text, and tweets. While capturing tweets, we filter them with hashtags (such as #wetip and #FultonMissing) and user profiles (for instance, @CambMA and @WLPD). We utilize the Twitter search application programming interface (API) to find tweets with a specific hashtag or user ID from historical tweets. The streaming API captures streaming tweets matching the search tag. Finally, we deploy Kafka to ingest the tweets into the PostgreSQL database to keep missing person cases separated by using each case as a topic, as seen in the data capture module. Kafka consumers read from the topics and store the JavaScript Object Notation (JSON) output from the API to PostgreSQL. The tweet preprocessing module also uses the JSON output as input. Using Kafka to read from each case ensures the parallel processing of multiple missing person cases.

For each modality, we adapt a different preprocessing system with high-level property identification. The extracted properties are chosen based on the requirements of the application domain. This additional feature identification step is done at load time to reduce response time during a complex query. Subsequent feature identification stages use the output from the preprocessing steps as inputs. Granular features are more complex and often involve computational overhead. Hence, we extract them on demand. For example, for missing persons, authorities are looking for human attributes, so people are identified during data ingestion for video feeds. In later stages of feature identification, we extract different properties of a person, such as, gender, race, and clothes colors.

Preprocessing of video feeds. Find-Them follows ingress steps similar to those of SurvQ[1] for video feeds. When videos arrive at the server in real time or as a bulk manual upload, they are converted to MP-4 from their current format and downsampled to one frame per second for further processing. You Only Look Once (YOLO)[19] is applied to each frame to identify objects described in the Pascal Visual Object Classes (VOC) data set (http://host.robots.ox.ac.uk/pascal/VOC/). For high-level object detection, Find-Them uses YOLO because of its runtime efficiency and the availability of pretrained models with a large number of object classes. The Pascal VOC data set includes 20 class labels, including *person*, and seven types of vehicles, making it a good candidate for the pretrained model in the missing person problem. Each YOLO-detected object is further examined in the feature extraction stage to identify finer-granularity object properties.

Preprocessing for unstructured text and tweets. Documents are converted to plain text from their incoming formats. The preprocessing module standardizes text in the documents by removing jargon, articles, abbreviations, and short forms of regular English words, depending on the source of data collection. The remaining text is converted to lowercase. The result from the Twitter API comes with a lot of metadata, which is helpful during data fusion. Raw JSON object outputs from the API are parsed to separate metadata and original text. Text in tweets is similar to unstructured text but includes jargon, hashtags, user tags, and abbreviations. So, before processing the tweets as documents, text is cleaned after removing or replacing jargon with the closest English words. As the next step, hashtags and user tags are removed. The feature extraction module designed for documents takes the cleaned and parsed texts as inputs.

Find-Them has an extendable library of feature extractors for video and text.

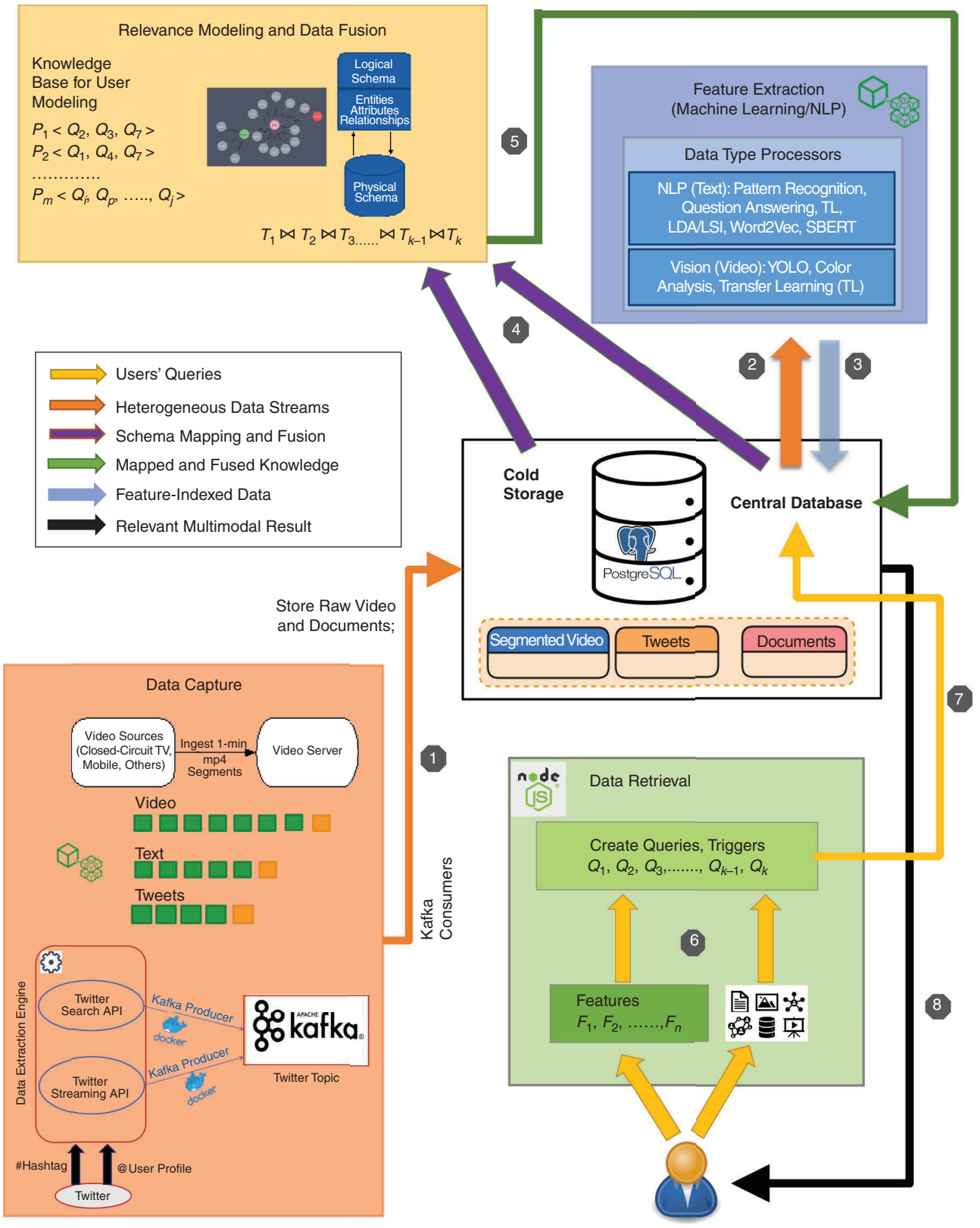

FIGURE 1. The Find-Them system architecture. LDA: latent Dirichlet allocation; YOLO: You Only Look Once; API: application programming interface; LSI: latent semantic indexing; SBERT: Sentence-BERT; NLP: natural language processing.

We explain the extractors needed for the missing person problem in detail in the "Feature Extraction" section along with the experimental results used for validation on data sets from real-world applications. However, Find-Them is extendable to other modalities and feature extractors. Feature extractors for other modalities can be added and used in a plug-and-play mode. It is also possible to use different feature extractors than the ones in this article, given that they have the same output features.

Data storage. To achieve scalability and a faster response, we store the outputs of the feature extractors in separate PostgreSQL tables for each modality, with pointers to archived raw videos and texts. Tweet metadata and user metadata are stored in different tables. This solution facilitates finding relevant data objects with SQL queries in real time.

Relevance modeling and data fusion

EAR model with schema mapping. For real-time data fusion, we propose to construct an EAR model for each application domain and then map to a relational database with schema S, as described in Figure 2. Each source needs to follow this schema. Adding a new data source to the system would require extending the EAR model and schema. For example, Figure 3(a) and (b) shows the individual schemas of incident reports and videos for the problem of person identification for the West Lafayette Police Department (WLPD). In Figure 3(c), we show the proposed combined schema for cross-modal retrieval for mining relevant data objects describing a person of interest. We translate all extracted features from video and text to the schema during data storage.

Data fusion with SQL JOIN. We propose to use the EAR model with SQL querying (EARS) for data fusion. Since data from each source have the same schema after mapping, matching among data objects of different modalities translates into JOIN queries among the tables. The results can be presented as an exact as well as an approximate match, depending on the conditions imposed on the JOIN query.

We implement a nested loop join on relations from each modality and the incident relation. Each queried missing person incident is converted into relation R with features $F_1, F_2, \ldots, F_m$. Features from modalities are translated into relations $T_1, T_2, \ldots, T_n$ where n is the number of modalities in the system. We perform a join between R and each T_k ($k \leq n$), using the join predicate JP on all queried features:

$$JP(T_k, R) := \sum_{1 \leq i \leq m} (T_k.F_i == R.F_i). \quad (1)$$

For example, in Figure 2, features from the video feed are translated into relation T_1, and features extracted from the incident report are translated into relation T_2 after schema mapping. If the user is interested in a person with features $F_2, F_6, \ldots, F_i$, we create a JOIN query across all the translated relations and the incident relation on features $F_2, F_6, \ldots, F_i$.

User preference modeling. Find-Them employs simplified user preference modeling to keep track of changes in requirements. We keep a record of the historical queries made by the user. For now, we issue notifications during streaming data delivery only for the current user query. For future improvements, we are building a predictive model using the history of user queries. This model will ensure better on-demand data delivery and the creation of notifications based on both the context and the current user's query.

Data retrieval

During data retrieval, Find-Them expects a user to either create a missing person incident or upload an example video/image/document/flyer (Figure 4) that describes the missing person. As seen in Figure 5, for incident creation, the user will upload the gender, race, upper body color, lower body color, and head/hair color as a description of the missing person. Users will also mention the date range and area they are interested in searching.

In the former case, the example is parsed using the modality-specific feature extractor, and the extracted features are used as user inputs. As evident in Figure 1, features mentioned by the user are considered predicates to SQL queries and defined as triggers to the PostgreSQL database management system. Using one-shot and standing queries enables us to find the desired result from both historical and streaming data. One-shot queries are immediately translated into SQL for schema S and executed. Standing queries are handled by triggers, which are automatically invoked when any matching data arrive. When queries involve information from one modality, the retrieval is straightforward. If similar data arrive in the future from other modalities, the trigger associated with the fusion model will link them and deliver the streaming data objects as standing query results.

FEATURE EXTRACTION

Our primary use case was person identification for the WLPD. The department searches for missing persons and suspects in a similar way. Persons of interest are described with different physical attributes, such as gender, race, physical build, height, hair color, color, and clothes, as well as other visible body features. These descriptions are circulated through press releases and missing person flyers. Whenever there is a related 9-1-1 call, the authorities generate an incident report describing the events. After investigation, officers write a report. Both of these reports include person descriptions, as mentioned previously. We analyzed the text in incident and investigation reports shared by the WLPD after the anonymization of identifying information. The top frequencies of different attributes for person profiling in the documents are as follows: almost all

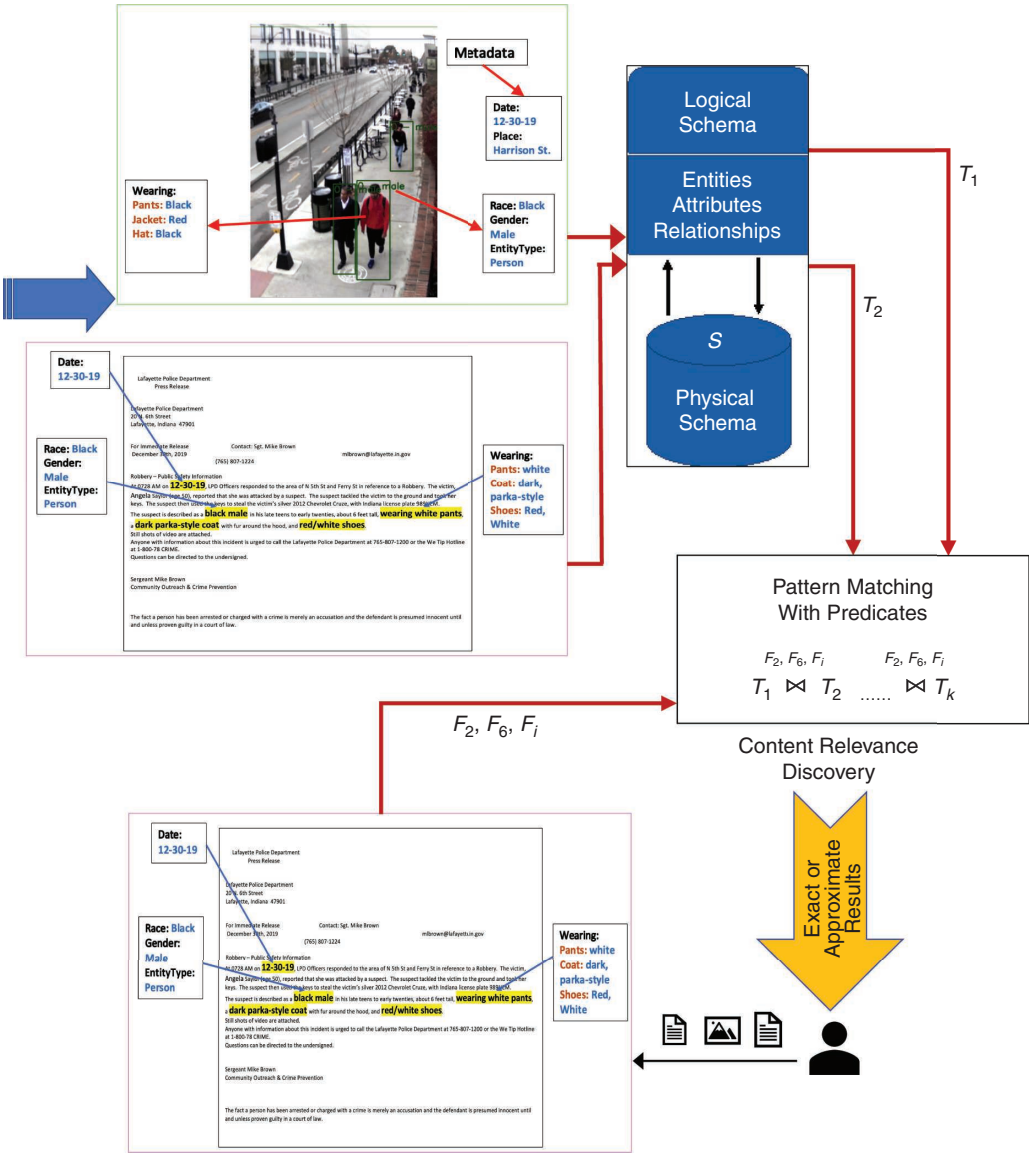

FIGURE 2. The data fusion for relevant information recommendation.

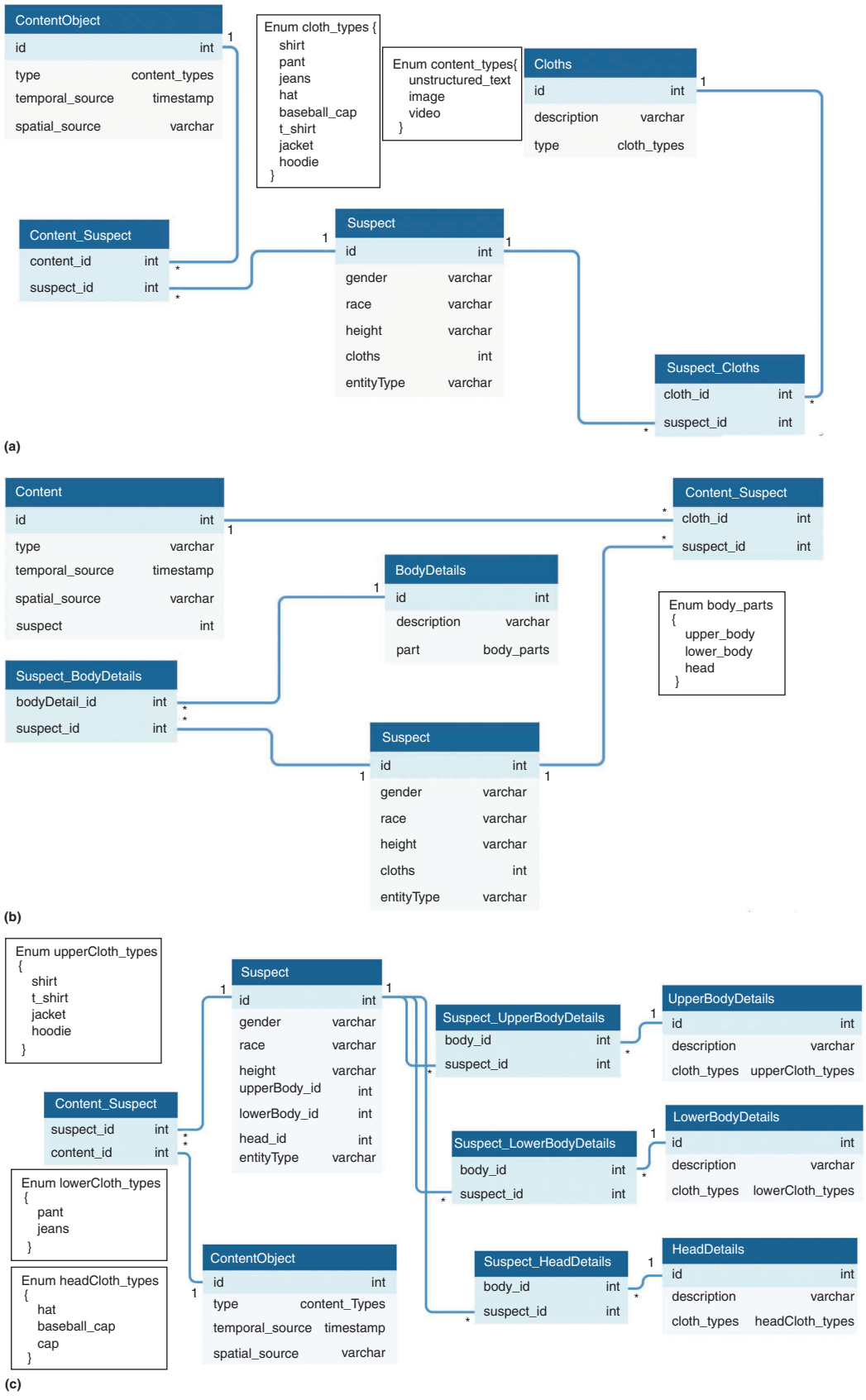

FIGURE 3. The data storage models. (a) The schema for the incident reports. (b) The schema for the video feeds. (c) The combined schema for fusion among multiple modalities.

documents use gender and race, 78% of the reports include clothes (such as shirts, jeans, pants, and jackets), and around 57% contain height. Therefore, in this work, we describe only the feature identifiers that were used to extract gender, race, and clothes colors in videos and text, as follows:

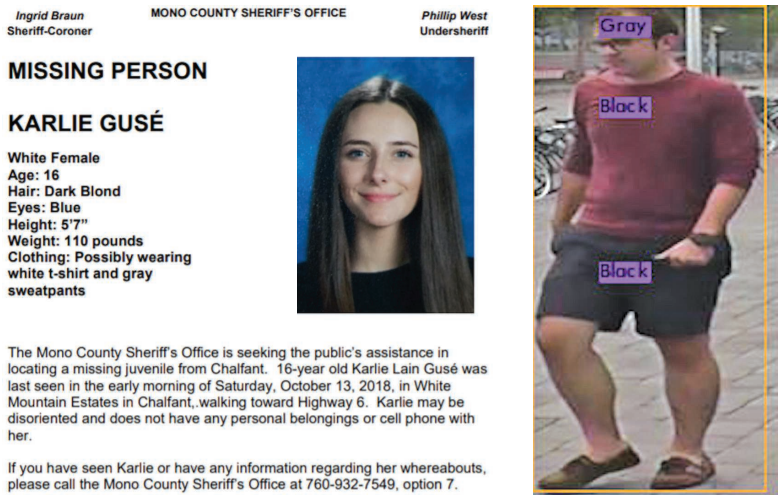

FIGURE 4. An example of data objects: (a) flyers and (b) screenshots.

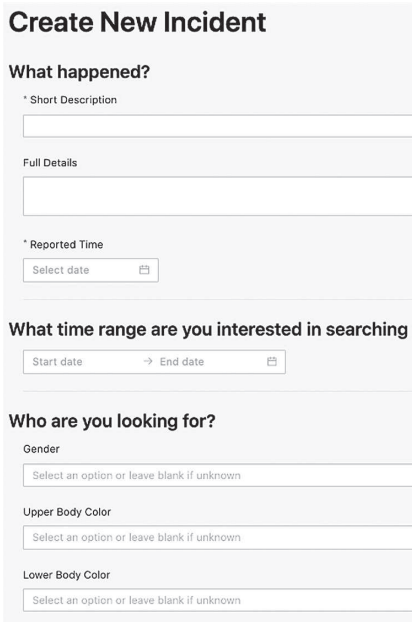

FIGURE 5. The incident creation page.

- For identifying clothes colors and tracking a person in the video feeds, we used a heuristic-based color sampling method[1] with YOLO as the background object identifier. This enabled us to identify and track a person based only on external identifiers without violating privacy.
- For gender and clothes detection in videos, we relied on the traditional deep learning object detection method and retrained YOLO with newer class labels.
- For gender, race, and clothes details detection in unstructured text, we used the Human Attribute Recognition from Unstructured Text (HART) model based on regular expression search, Word2Vec embedding, and pattern recognition. We also employed a topic-based similarity search technique for finding tweets and texts describing objects in the videos. Harnessing text embedding enabled us to identify ambiguities in different people's writing style when describing colors.

Feature identification in visual modalities is significantly different than it is in textual modalities. Since text modalities describe the color of clothes in words, there can be ambiguities. On the other hand, in videos, colors can have high variance ranging from light to dark. Extracted features are stored in PostgreSQL following the common EAR model, which enables us to perform uniform SQL queries across different modalities. We benchmarked these models on real-world data sets and used the extractor results during data fusion.

Color analysis for body details

For color sampling,[1] we use the bounding box of persons from the YOLO detection. The bounding box is segmented into three body parts: the head, upper section, and lower section. We segment the body parts by estimating the ratio of each to the bounding box according to human body proportions in anatomy. First, red–green–blue (RGB) values are extracted from each pixel in a segmented region. Colors for each segment are assigned by calculating the smallest distance between the extracted RGB values and standard RGB values. Integer RGB values make it easier to compare the extracted colors to baseline colors. In the case

of multiple colors in a region, majority voting is applied to determine the color of the area.

WLPD video data set. We collected and labeled more than 20 h of video from different cameras and locations in West Lafayette. Six custom classes with more than 12,200 images were manually labeled for retraining and testing the YOLO network to detect gender, clothes, and color. Each 1-min chunk of video consists of around 20 frames sampled at 3-s intervals. In the test set from the WLPD video data set, clothing colors were recognized with high precision, while the color of the sampled head area was more prone to be affected by that of the background, as shown in Figure 6. Based on color information, we can trace the movements of pedestrians across continuous frames. Figure 7 presents the routes of two pedestrians walking toward each other. The dotted line after each indicates their direction.

In cities, multiple cameras are installed at traffic cross sections to observe pedestrians from different angles, with each view providing additional information. We wanted to trace one person across multiple cameras installed at various locations for the missing person search. Figure 8 gives two examples of tracking a person through three areas. In Figure 8(a), we track a cyclist wearing a red shirt, passing from locations 1 to 3. It takes only 39 s because he is riding a bicycle. In Figure 8(b), we follow a pedestrian wearing a red shirt, passing from location 3 to 1 in the opposite direction. It takes him about 6 min. So, we can map the walking trajectory of a person as long as there is no change of clothes.

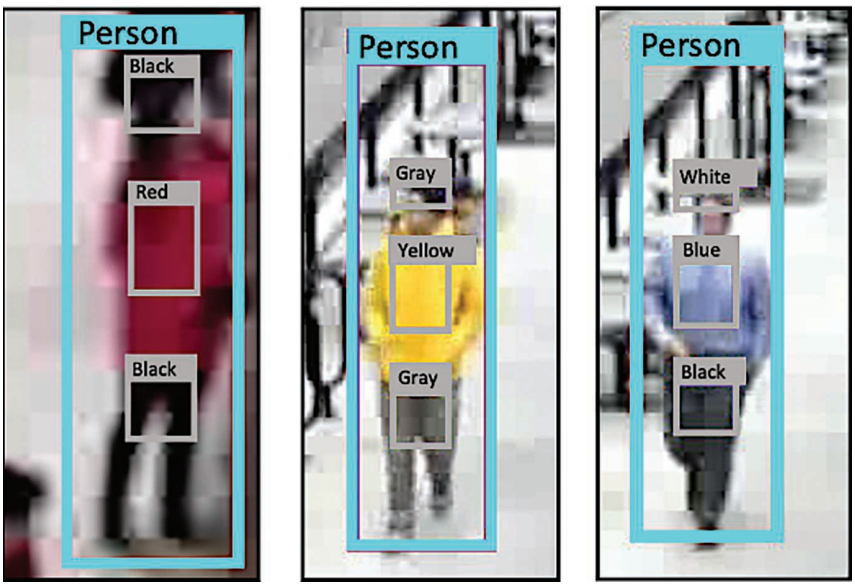

FIGURE 6. The color recognition for the WLPD video data set.

(a) (b)

FIGURE 7. The tracking from a single camera of a pedestrian crossing the street at (a) 2:01:02 p.m. and (b) 2:01:07 p.m.

Retraining YOLO

For gender and clothes detection in video feeds, we retrained YOLO.[19] The hue, saturation, and brightness (HSB) of each frame were analyzed to improve object detection and recognition under night and changing weather conditions. The range of HSB values are tracked for each color as time passes, and the updated values are used for more accurate object detection and recognition. We are building fine-tuned YOLO models for future improvement. We report results for both gender and cloth detection with YOLO v3 and YOLO v4 in Table 1. For gender and clothes detection, we achieved 68 and 67% mean average precision, respectively, when YOLO was retrained without pretrained features. Achieving higher performance with real-life, low-resolution raw video under different light and weather conditions is a difficult task that requires future work.

FIGURE 8. The tracking of a person at multiple scenes with multiple cameras. (a) A cyclist. (b) A pedestrian.

Human attributes from unstructured text

Using the stacked [regular expression (RE) + Word2Vec] variant of the HART model,[20] we identified candidate sentences (C_s) from the texts of cleaned documents and tweets. We searched for *clothes* with regular expressions in the sentences for finding C_s. If this returned no result, the problem was formulated as a similarity search among all tokens in a sentence, where *clothes* is used as the search token. We used the pretrained Word2Vec embedding for each token as features. If the cosine similarity between any token in a sentence and the search phrase reaches an empirical threshold, we consider it C_s. For the attribute value detection from C_s, specific patterns were searched for recognizing gender and race. For clothes identification, we followed the clothes name and value identification algorithm from Solaiman and Bhargava,[20] which uses POS tags of tokens to identify the description.

Feature centric multi-modal information retrieval (FemmIR) text data set. For benchmark results for text features, we used part of the text data from Solaiman and Bhargava,[20] consisting of incident reports, press

TABLE 1. The mean average precision of YOLO for gender and clothes detection in the WLPD video data set.

Object	YOLO v3	YOLO v4
Gender	0.59	0.68
Clothes	0.56	0.67

releases, and officer narratives from historical cases. It contains 13 press releases, 40 officer narratives, and five incident reports. Due to privacy reasons, the WLPD publicly released only a subset of redacted reports. For unstructured text, as seen in Table 2, the HART model performs adequately for an on-demand detection model. Results for clothes are reported for the RE + Word2Vec + POS model through two evaluation metrics: attribute only and attribute value.

Semantic similarity search by topic

We employed topic-based similarity search to extract documents describing objects and attributes found in videos. We also used it as an additional method for finding candidate sentences. Assuming that each sentence in a document is a mixture of topics, if any of those topics explains the search phrases, we posit that the sentence is a candidate one. We used latent Dirichlet allocation (LDA) to identify hidden topics in sentences in the documents and query phrases (for example, clothes, car, person, and male). LDA is a generative topic modeling technique in which documents are represented as random mixtures across unseen subjects, which are derived by calculating distributions across all the words in a document. In this case, we represented each sentence in a document and the query phrase as an individual mixture of topics. For distribution measurement, term frequency–inverse document frequency vectors of all tokens in a sentence were used as unigram features. The cosine similarity of the query phrase topic against the subjects of the corpus of sentences measures the closest sentence matching a query. We collected 249,857 tweets from 77,943 users, describing topics

TABLE 2. The evaluation of human attribute extraction on the FemmIR text data set (results reported from Solaiman and Bhargava[20]).

Attribute	Gender	Race	Clothes (attribute only)	Clothes (attribute value)
Precision	0.94	0.94	0.93	0.92
Recall	0.73	0.73	0.65	0.87
F1 Score	0.82	0.82	0.77	0.9

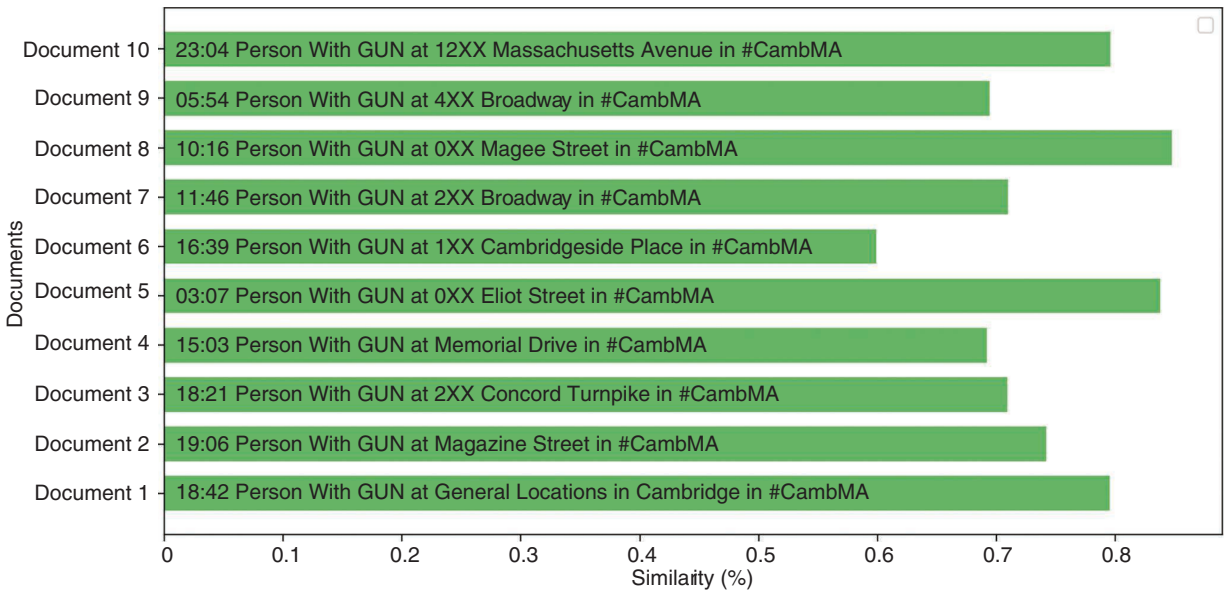

FIGURE 9. The relevant tweets with LDA in the CPAT data set, describing a person with a gun in the Cambridge area.

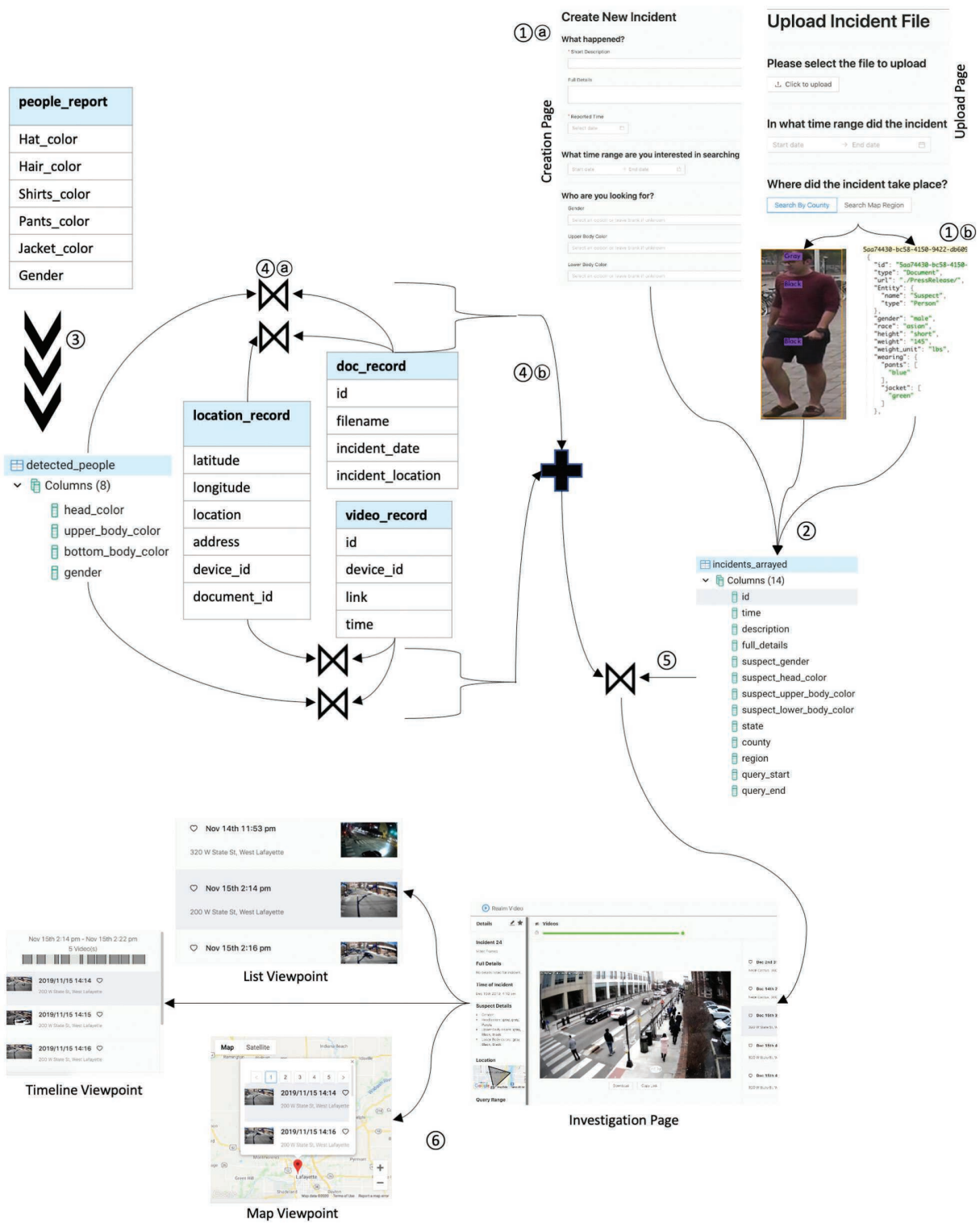

FIGURE 10. The Find-Them demonstration.

related to Cambridge, Massachusetts in the Cambridge Public Authority Tweets (CPAT) data set (see Figure 9).

DEMONSTRATION

Finally, we demonstrate Find-Them on the incident reports, press releases, and video feeds from the WLPD. We are working on adding DMV records and public tips as additional data sources in the future. We show how Find-Them can accurately detect and track a missing person based on noninvasive physical properties and minimize investigation efforts. We describe the user process through six steps, from the point of view of a WLPD officer. We annotate each of the following steps with a circle in Figure 10:

- *Step ① (create a missing person incident or upload an example)*: First, the user uploads an incident report, flyer, or tweet with a physical description of the missing person, with the search area and search timeline in step 1(b). He or she can also upload a video clip or snapshot of the missing person. In this case, we apply appropriate feature extractors to the examples based on their modality. Then, the predicates for the search query are created with the extracted features. When the user does not have examples, he or she can create a missing person incident by filling out the person's details, search area, and timeline, as in step 1(a).
- *Step ② (create predicates)*: To search for a person, the user specified the identifying properties in step 1. Using those inputs, we create an incident schema that becomes the search criteria for current and future streaming data in step 2. Triggers in PostgreSQL await streaming data with features similar to the incident, and they notify the user of matching video feeds and tweets. The user can always revisit incidents from the search history.
- *Step ③ (EAR mapping)*: As seen in Figure 3(a), incident reports have a feature extractor that outputs clothes as individual entities and then extracts their details, whereas in Figure 3(b), we observe that details are extracted in terms of body parts. Both of these modalities need to map to the common EAR model in Figure 3(c). The system maps the incoming document features to the common EAR model as follows: (shirts and jackets) → upper body, (pants and jeans) → lower body, and (hats and caps) → head.
- *Steps ④ and ⑤ (JOIN among data sources)*: Before this step, data from each modality are stored in PostgreSQL tables in an atomic manner. The data storage schema was built considering different categories of features necessary for missing person problems. For example, physical details about a missing person are saved in one table, whereas incident locations are stored in another. The separation of storage enables us to answer simple queries requiring only one type of information to be quickly available. When a query involves multiple types of information, we create SQL queries to perform a JOIN among tables representing features from different modalities. The first JOIN in step 4(a) separately creates the primary results from each modality. In step 4(b), we perform a union of all modalities. Finally, in step 5, we perform a JOIN between the accumulated results and previously created incident table to extract the subset of data objects that match the search criteria and show the multimodal result on the investigation page.
- *Step ⑥ (different viewpoints)*: Similar to Stonebraker et al.,[1] there are three possible viewpoints the user can choose from to see the results: list, map, and timeline. The timeline view was generated to mimic the investigation process, whereas the map view enables us to pinpoint a location. The user can also choose his or her favorite results and see them at a later time.

THE LINKING PROCESS FOR EARS CAN SCALE TO A LARGE NUMBER OF PROPERTIES FROM DATA OBJECTS, AND EARS DOES NOT REQUIRE TRAINING.

ABOUT THE AUTHORS

K.M.A. SOLAIMAN is a graduate research assistant in computer science assistant and a Ph.D. candidate at Purdue University, West Lafayette, Indiana, 47907, USA. His research interests include multimodal information retrieval, machine learning, and heterogeneous data mining. Solaiman received a B.Sc. in computer science and engineering from Bangladesh University of Engineering and Technology. Contact him at ksolaima@purdue.edu.

TAO SUN is a system design and management fellow at the Computer Science and Artificial Intelligence Laboratory and the Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139, USA. His research interests include smart sensors, artificial intelligence, computer vision, and mobile robotics. Sun received a Ph.D. in electronic engineering from the University of Southampton. Contact him at taosun@mit.edu.

ALINA NESEN is a Ph.D. candidate in computer science at Purdue University, West Lafayette, Indiana, 47907, USA. Her research interests include multimodal and multitask machine learning and video object detection and recognition. Contact her at anesen@purdue.edu.

BHARAT BHARGAVA is a professor of computer science at Purdue University, West Lafayette, Indiana, 47907, USA. His research interests focus on intelligent autonomous systems, data analytics, and machine learning, including cognitive autonomy, reflexivity, deep learning, knowledge discovery, fairness, trust, and explainable artificial intelligence. Bhargava received a Ph.D from Purdue University. He is a Fellow of IEEE. Contact him at bbshail@purdue.edu.

MICHAEL STONEBRAKER is a professor of computer science at the Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139, USA. His research interests include novel data structures for database management system (DBMS) implementations and new uses for DBMS technology, especially in the operating system stack. Contact him at stonebraker@csail.mit.edu.

SCALABILITY, UNIVERSALITY, AND MULTIPLE USERS

Find-Them establishes a common information model, the relational schema, across multiple data sources and eliminates the need for separate information representation and linking methods. These models are universal for all modalities, without additional overhead since converting features into relational tables is a linear process. The linking process for EARS can scale to a large number of properties from data objects, and EARS does not require training. The system demonstration shows that we could query historical data (in thousands of records) and streaming data in real time during inference. For the space constraint, we do not include the time comparisons here. Find-Them is capable of extension to multiple users, each with his or her preferences in the form of queries and data objects. Since all users have a mapping to the retrieval set with their queries, their queries are kept separate.

This article introduced Find-Them, a feature-based multimodal data fusion system for analyzing video feeds with other data modalities for finding missing persons. We described a database back end along with a schema and relational query-based fusion method that can scale to a considerably large amount of data, with a fast response time. Our experimental results showed satisfactory performance for the feature identifiers for commonly used missing person features. Find-Them can also discover connections among historical and incoming missing cases, giving law enforcement an edge in investigations.

In the future, we will expand the video and text data sets by including mobile camera videos, city maintenance files, and DMV records. We also have goals to include more data modalities and evaluate the effects that humans in the loop have on improving performance. We further benchmark the EARS algorithm for searching for a person with certain features in an incremental work. In future efforts, we will test Find-Them and its viewpoint capability during rush hours by employing data collected on days when there is heavy traffic and manually annotated map–timeline

ground truth. Finally, we will extend the framework to include feature extraction as part of the relevance modeling in an end-to-end neural network architecture, and user interests will be modeled based on historical queries.

ACKNOWLEDGMENTS

This work was supported by Northrop Grumman Mission Systems' University Research Program. We are grateful to Prof. Michael Cafarella (University of Michigan), Shivani Desai, Jason Kobes, Detective Gerry Palmer, and Sergeant Troy Greene for their helpful feedback on the work. We would like to thank Pelin Angin, Kevin Kotchpatcharin, MyeongSu Kim, Harshit Singh, Tomas Hrdlovics, Aaron Sipser, and Zachary Collins for their help with the system implementation and data annotation.

REFERENCES

1. M. Stonebraker *et al.*, "Surveillance video querying with a human-in-the-loop," in *Proc. Workshop on Human-in-the-Loop Data Anal. (HILDA 20)*, Portland, OR, USA, Jun. 14–19, 2020, doi: 10.1145/3398730.3399192.
2. J. Wang, X. Zhu, S. Gong, and W. Li, "Attribute recognition by joint recurrent learning of context and correlation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 531–540, doi: 10.1109/ICCV.2017.65.
3. G. Pearson, M. Gill, S. Antani, L. Neve, and G. Thoma, "People locator: A system for family reunification," *IT Professional*, vol. 14, no. 3, pp. 13–21, May 2012, doi: 10.1109/MITP.2012.25.
4. R. S. Ferreira, C. G. de Oliveira, and A. A. Lima, "Myosotis: An information system applied to missing people problem," in *Proc. XIV Brazilian Symp. Inf. Syst.*, 2018, pp. 1–7.
5. H.-X. Yu, W.-S. Zheng, A. Wu, X. Guo, S. Gong, and J.-H. Lai, "Unsupervised person re-identification by soft multilabel learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2148–2157, doi: 10.1109/CVPR.2019.00225.
6. S. Aggarwal, V. B. Radhakrishnan, and A. Chakraborty, "Text-based person search via attribute-aided matching," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2617–2625, doi: 10.1109/WACV45572.2020.9093640.
7. Z. Wang, Z. Fang, J. Wang, and Y. Yang, "ViTAA: Visual-textual attributes alignment in person search by natural language," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 402–420.
8. M. Khan and A. Jalal, "A fuzzy rule based multimodal framework for face sketch-to-photo retrieval," *Expert Syst. Appl.*, vol. 134, pp. 138–152, Nov. 2019, doi: 10.1016/j.eswa.2019.05.040.
9. J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," in *Proc. Conf. Data Mining Data Warehouses (SiKDD 2010)*, 2010, pp. 1–4.
10. L. Zhang, B. Ma, G. Li, Q. Huang, and Q. Tian, "Generalized semi-supervised and structured subspace learning for cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 128–141, 2017, doi: 10.1109/TMM.2017.2723841.
11. Y. Peng, J. Qi, X. Huang, and Y. Yuan, "CCL: Cross-modal correlation learning with multigrained fusion by hierarchical network," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 405–420, 2017, doi: 10.1109/TMM.2017.2742704.
12. W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.
13. S. Sah, S. Gopalakrishnan, and R. Ptucha, "Aligned attention for common multimodal embeddings," *J. Electron. Imaging*, vol. 29, no. 02, p. 23013, 2020, doi: 10.1117/1.JEI.29.2.023013.
14. K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4654–4662.
15. F. Wu *et al.*, "Modality-specific and shared generative adversarial network for cross-modal retrieval," *Pattern Recognit.*, vol. 104, p. 107335, Aug. 2020, doi: 10.1016/j.patcog.2020.107335.
16. X. Wang, P. Hu, L. Zhen, and D. Peng, "DRSL: Deep relational similarity learning for cross-modal retrieval," *Inf. Sci.*, vol. 546, pp. 298–311, Feb. 2021, doi: 10.1016/j.ins.2020.08.009.
17. S. Poria, E. Cambria, N. Howard, G.-B. Huang, and A. Hussain, "Fusing audio, visual and textual clues for sentiment analysis from multimodal content," *Neurocomputing*, vol. 174, pp. 50–59, Jan. 2016, doi: 10.1016/j.neucom.2015.01.095.
18. S. Palacios *et al.*, "WIP–SKOD: A framework for situational knowledge on demand," in *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*. Cham, Switzerland: Springer Int., 2019, pp. 154–166.
19. J. Redmon and A. Farhadi, "YOLOV3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.
20. K. Solaiman and B. Bhargava, "Feature centric multi-modal information retrieval in open world environment (FEMMIR)," to be published. [Online]. Available: https://tinyurl.com/2ust5xbm