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

KMA Solaiman
Purdue University

Tao Sun
Massachusetts Institute of Technology

Alina Nesen
Purdue University

Bharat Bhargava
Purdue University

Michael Stonebraker
Massachusetts Institute of Technology

***Abstract*—We present a system for integrating multiple sources of data for finding missing persons. This system can assist authorities in finding children during amber alerts, mentally challenged persons who have wandered off, or person-of-interests in an investigation. Authorities search for the person in question by reaching out to acquaintances, checking video feeds, or by looking into the previous histories relevant to the investigation. In the absence of any leads, authorities lean on public help from sources such as tweets or tiplines. A missing person investigation requires information from multiple modalities and heterogeneous data sources to be combined.**

Existing cross-modal fusion models use separate information models for each data modality and lack the compatibility to utilize pre-existing object properties in an application domain. A framework for multimodal information retrieval, called Find-Them is developed. It includes extracting features from different modalities and mapping them into a standard schema for context-based data fusion. Find-Them can integrate application domains with previously derived object properties and can deliver data relevant for the mission objective based on the context and needs of the user. Measurements on a novel open-world cross-media dataset show the efficacy of our model. The objective of this work is to assist authorities in finding uses of Find-Them in missing person investigation.

INTRODUCTION

There are many circumstances in which the "missing person" problem arises. They include amber alerts, family reunification during natural disasters, prison escape, or unaccounted people. Missing person search works similarly for prison escapees, adults with cognitive problems, or missing children. The police have the same problem when they search for a person of interest involved in a crime whether as a suspect or as a victim. For each situa-

tion listed above, the authorities have a physical description of the person (e.g., 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 on missing persons comes from their families, Twitter posts, and phone calls from the public. Available vehicle information can be co-related with Department of Motor Vehicles (DMV) records. Irrespective of the information source, it will have some identifying features of the missing person, based on which the search is conducted. According to related works on missing persons from Policing and Society journal, one of the first steps in dealing with missing person incidents is to search the surveillance camera video footage in the vicinity. For example, West Lafayette, IN, has cameras in all city buses, on many intersections in the downtown area, in the majority of the local business buildings, and in all police cars. Moreover, the policemen themselves are equipped with bodycams when on duty. Police in West Lafayette already spend hours manually searching videos for missing persons [1]. Data fusion from these disparate data sources would be a valuable addition for automatic information retrieval and querying.

In this paper, we report on a system we have built, called Find-Them, to perform video capture, tweets and tips collection, feature identification, and information fusion among these 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, or darkness). The persons of interest may be in the background or facing away from the camera [2] which makes relying of face recognition infeasible for our task. Instead, we focus on other features, such as gender, clothing (e.g., baseball hat, shirts), and markings (e.g., tattoos).

To begin with, identifying a missing person is a data capture problem. As it was specified before, information about a missing person can come from multiple sources such as surveillance cameras, tweets, family members, and previous occurrences. Storing these multimodal data is a storage problem at scale. Since data comes from multiple modalities and in large amounts, the proposed storage system should normalize different modalities of data at a large scale. Finding relevant information about a specific missing person from multimodal data requires system-specific property identification in each modality and a context-based data integration for a composite query through these modalities. As discussed in [1], training data for property identification is 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 the traditional machine learning methods may fail for 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 data sources.

Find-Them implements a streaming data capture and downsampling method to tackle the problem of multimodal data capture and storage. To achieve scalability, Find-Them loads both the raw data and the acquired properties into a Postgres database. Raw data and properties are stored separately 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 schemas compatible with the application domain. Using the features specified by the user, we built SQL queries using the data description language. By performing these queries over the standard schemas, Find-Them delivers the multimodal results relevant to the user interest. The fusion methodology in Find-Them is expandable to other modalities and different feature identifiers for the discussed modalities.

We explain the related works in Section 2, then discuss the Find-Them architecture in detail in Section 3. We describe the proposed feature identifiers in Section 4 with benchmark experimental results. In Section 5 and 6, we discuss a demonstration scenario for Find-Them and the generalization capabilities of the system. Finally, we include the future directions and conclusion in Section 7.

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], NamUs (https://nij.ojp.gov/topics/articles/solving-missing-indigenous-person-data-crisis-namus-20), Google Person Finder (https://en.wikipedia.org/wiki/Google_Person_Finder) allow different levels of missing person search and comparison capabilities. People-locator [3], a search application for family reunification post-disaster, combines multiple modalities for searching and reporting missing people such as structured web form, app-based community reports (ReUnite), unstructured text from email, image-based hospital reports (TriagePic), and other applications with PFIF (People Finder Interchange Format) data format. Similar to Find-Them, for relevance matching PL employed SQL query-based database search and Apache Solr (http://lucene.apache.org/solr) based indexing and search-string matching. However, it lacks the capability of face matching or multimodal searching. NamUs allows to 1) search for matching demographics, descriptors, and distinctive characteristics of a missing person; 2) automatically compare cases based on geography, dates, and physical features and helps to find connections and investigative leads; 3) generate customized case maps. NamUs has a similar comparison and search functionality as Find-Them. Google Person Finder is a disaster time registry to post and search for missing person status. Myosotis [4] aggregates data from heterogeneous missing people databases, allows visualization via interactive maps, and infers an estimation of the probability of a new occurrence. Neither NamUs, nor Google Finder, nor Myosotis allow for multimodal or on-demand search from streaming heterogeneous data.

Person Re-identification (RE-ID). Person re-id refers to searching for a person in video feeds through a textual or an image query. Existing person re-id methods use both supervised and unsupervised learning [1], [5] techniques. Identity-aware annotations [6], [7] and zero-shot learning have increased the matching performance between image and text descriptions for person re-id by using text attribute query. Attribute recognition in the above models requires a substantial amount of training samples. Multimodal search differs from person re-id in the query-response formats. Cross-modal search allows using different data modalities as a query and as a response as well.

[8] augmented person re-id with facial sketch by fusing the facial attributes and the semantic color information in attributes using a fuzzy rule-based layered classifier. Find-Them does not perform any facial recognition, rather re-ids a person via various semantic attributes, including the color information. Existing methods [6], [7] for text attribute extraction considers noun phrases as potential attribute values. [6] filters the candidate phrases using associated images. [7] categorizes the noun phrases to specific attribute phrases such as, *upper-body* following a dictionary clustering approach. These approaches do not consider the noise in streaming documents and the performance bottleneck of parts-of-speech taggers. They also do not differentiate between attribute names and values extraction.

Cross-modal Matching and Correlation Learning. Most of the previous works [9], [10], [11], [12] in multimodal matching have followed the idea of projecting the features from different modalities into a shared embedding space using modality-specific transformations. Canonical Correlation Analysis [9] focuses on correlation learning to learn linear projections using pairwise information. In contrast, [10] uses both pairwise and semantic information, such as class labels, to learn the common subspace. [12] extends deep canonical correlation analysis with an auto-encoder regularization term for nonlinear representations of multimodal data objects. Peng et al. [11] better encodes the intra-modality and inter-modality correlation with hierarchical networks.

Some recent methods learn richer semantic representations for different modalities by using encoder networks. Previously, attention mechanism [13], graph representations [14], and generative models [15] were used to build the encoding networks. Deep Relational Similarity Learning [16] avoids explicitly learning a common space by integrating relation learning, capturing the implicit nonlinear distance metric.

While the learning methods mentioned above exhibit good performance, mainly on bi-modal datasets, they require a large amount of training data and do not scale well. Data representations lack generalization capability across multiple modalities or data sources. Besides, many existing application domains already have pre-derived domain-specific features with established feature learning methods, but the above models cannot integrate these sources. Moreover, many mining strategies proposed on metric learning methods are for unimodal matching. These methods can only integrate user-specified data relevancy with training samples or with class labels. The data fusion methodology described in Find-Them focuses on solutions for the above problems.

Data fusion among multiple modalities has been used 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 respective application domains, but they lack generalization capabilities. Similar to Find-Them [18] built a multimodal relational knowledge base by continuously querying for detected objects from videos and matching objects in text. However, they do not perform any attribute-specific search and cannot be generalized for multimodal person search.

System Overview

Figure 1 illustrates the architecture of Find-Them. Find-Them is divided into four modules — *data ingestion*, *feature identification*, *relevance modeling*, and *data retrieval*. *Data ingestion* deals with the problem of data capture and data storage. The system captures the streaming data and loads it into Postgres at the server end after necessary pre-processing. *Feature extraction* is done during load time using type-appropriate models for each data source. Extracted properties are inserted into Postgres following the schema determined by the entity-attribute-relationship 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. *Ingestion* and *retrieval* systems can operate in parallel. A user preference model is built from the history of user queries and is 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 filtered the tweets with hashtags (#wetip, #FultonMissing) and user profiles (@CambMA, @WLPD). We utilize Twitter search API to find tweets with a specific hashtag or user-id from historical tweets. Streaming API captured the streaming tweets matching the search tag. Finally, we deploy Kafka to ingest them into the Postgres database to keep missing person cases separated by using each case as a topic, as seen on the *data capture* module. Kafka consumers read from the topics and store the JSON output from the API to Postgres. The tweet pre-processing module also uses the JSON output as input. Using Kafka to read from each case separately ensures parallel processing of multiple missing person cases.

For each modality, we adapt a different pre-processing system with a 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 pre-processing steps as inputs. The granular features are more complex and often involve computational overhead. Hence, we extract these features 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 the feature identification, we extract different properties of a person, such as, gender, race, cloth colors.

Pre-processing of Video Feeds. Find-Them follows similar ingress steps as SurvQ [1] for video feeds. When the videos arrive at the server in real-time or as a manual bulk upload, they are converted to MP-4 from their current format and are downsampled to one frame per second for further processing. YOLO [19] is applied to each of these frames to identify the *objects* described in the Pascal VOC dataset (http://host.robots.ox.ac.uk/pascal/VOC/).

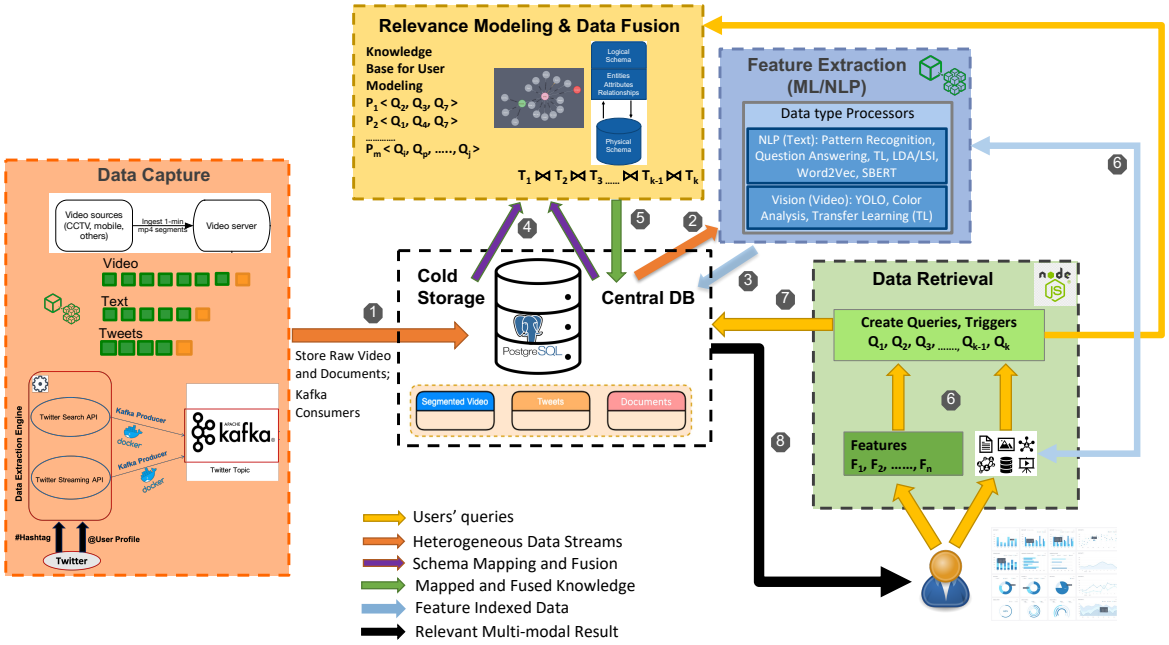


Figure 1: System Architecture for Find-Them.

For high-level object detection, Find-Them uses YOLO because of its run-time efficiency and availability of pre-trained models with a large number of object classes. The Pascal VOC dataset includes 20 class labels, including *person* and seven types of vehicles, making it a good candidate for the pre-trained model in the missing person problem. Each YOLO-detected object is then further examined in the feature extraction stage to identify finer granularity *object properties*.

Pre-processing for Unstructured Text and Tweets. Documents are converted to *plain text* from their incoming formats. Pre-processing module standardizes texts in the documents by removing jargon, articles, abbreviations, and short forms of regular English words, depending on the source of data collection. The remaining texts are converted to lower cases.

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 the metadata and original text. Texts in tweets are similar to unstructured text but include jargon, hashtag, user tags, and abbreviations. So, before processing the tweets as documents, text in the tweets is cleaned after removing or replacing the jargon with the closest English words. As the next step, hashtags and user tags are removed from the text. Feature extraction module designed for documents takes these cleaned and parsed texts as inputs.

Find-Them has an extendable library of feature extractors for video and text. 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 datasets 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 in Find-Them. It is also possible to use different feature extractors than the ones used in this paper, given that they have the same output features.

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

Relevance Modeling and Data Fusion

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

Data Fusion with SQL JOIN. We propose to use the **E**ntity-**A**ttribute-**R**elationship model with **S**QL querying (EARS) for data fusion. Since data from each source has the same schema after schema mapping, matching between data objects of different modalities translates into JOIN queries between the tables. The results can be presented as an exact or approximate match depending on the conditions imposed on the JOIN query. As a simplified example, in Figure 2, features from the video feed are translated into table T_1, and features extracted from the incident report are translated into table 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 over all the translated tables on $F_2, F_6, \ldots, F_i$.

User Preference Modeling. Find-Them employs a simplified user preference modeling to keep track of changes in user requirements. We keep a record of the historical queries made by the user. For now, we issue notifications during the streaming data delivery only for the current user query. For future improvements, we are building a predictive model using the history of the user queries. This model will ensure a better on-demand data delivery and 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 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 the search area they are interested in searching through.

In the former case, the example is parsed using the modality-specific feature extractor, and the extracted features are used as user inputs. As seen in Figure 1, features mentioned by the user are considered as predicates to SQL queries and are defined as triggers to the Postgres DBMS. Using one-shot and standing queries allows us to find the desired result from both historical and streaming data. One-shot queries are immediately translated into SQL for schema S and are executed. Standing queries are handled by triggers, which are invoked automatically when any matching data arrives. When the queries involve information from only one modality, the retrieval is straightforward. If similar data arrives 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 for the missing person was person identification for West Lafayette Police Department (WLPD). WLPD 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 description of their clothes, and other visible features in their body. These descriptions are circulated through press releases and missing person flyers. Whenever there is a 9-1-1 call, the authorities generate an incident report describing the series of events. After the due investigation, the involved officers write an investigation report on the incident. Both of these reports include person descriptions, as mentioned above. We analyzed the text in the incident and investigation reports shared by WLPD with us. WLPD shared these reports after proper

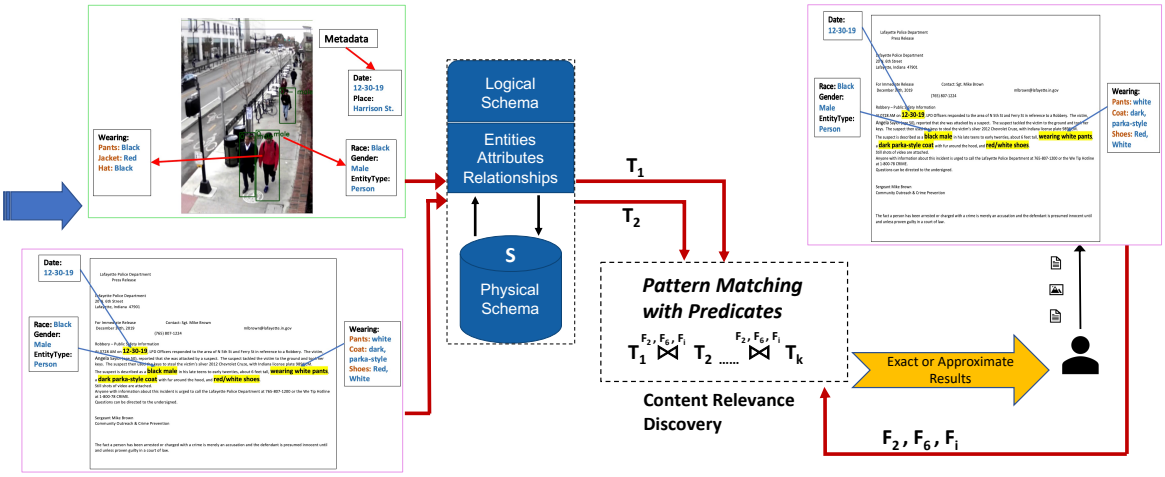
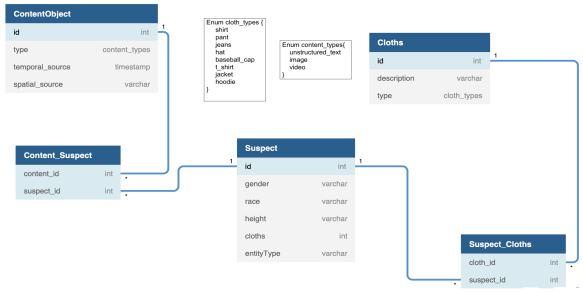
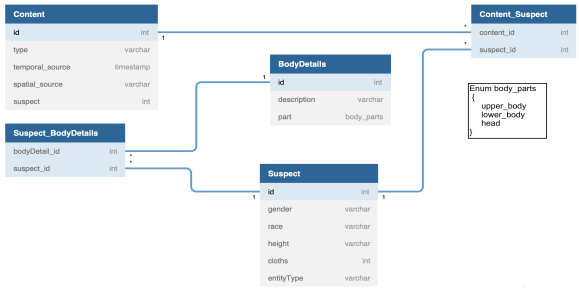


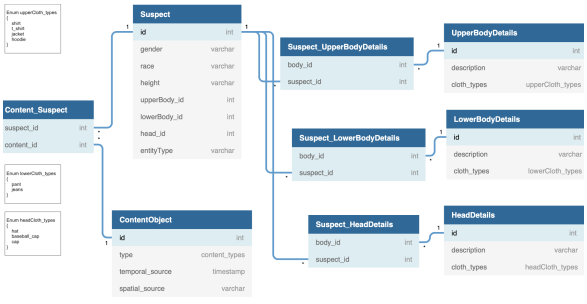
Figure 2: Data Fusion for Relevant Data Recommendation



(a) Schema for Incident Reports

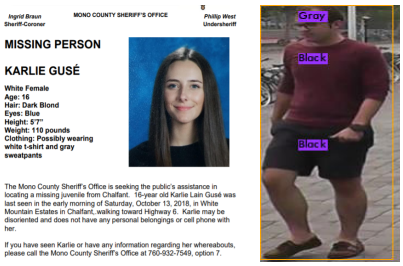


(b) Schema for Video Feeds



(c) Combined Schema for Fusion between Multiple Modalities

Figure 3: Schema Models for Data Storage



(a) Flyers (b) Screenshots

Figure 4: Upload Example of Data Objects

anonymization of identifying information. The top frequencies of different attributes for person profiling in the documents are as follows: almost all documents use gender and race, 78% of the reports use clothes (such as shirts, jeans, pants, jackets), and around 57% use height to describe a person. Therefore, in this work, we only describe the feature identifiers that were used to extract gender, race, clothes color in videos and text, as follows:

- 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 method allows us to identify and track a person only based on external identifiers without violating privacy.
- For gender and clothes detection in videos, we relied on the traditional deep learning object detection method and re-trained YOLO with newer class labels.

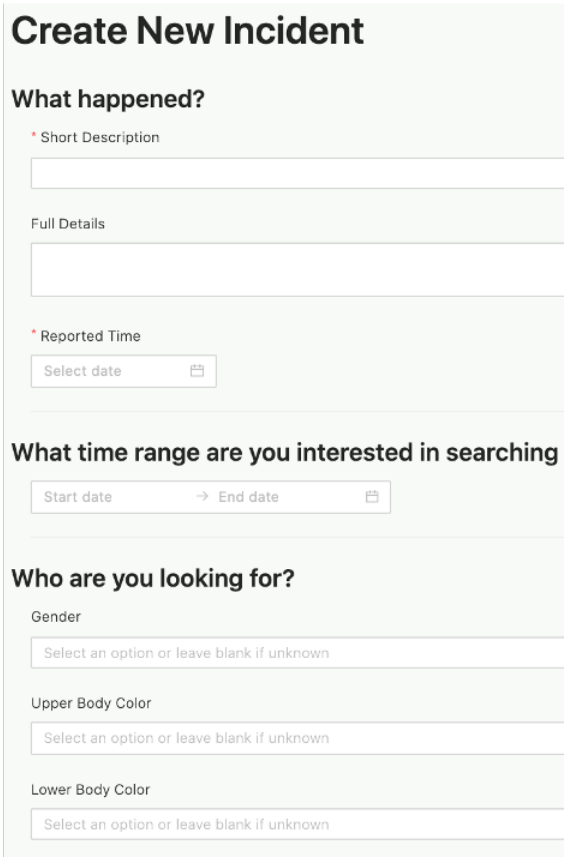


Figure 5: Incident Creation Page

- For gender, race, and clothes details detection in unstructured text, we used the *HART* model based on regular expression search, Word2Vec embedding, and pattern recognition. We also used a topic-based similarity search technique for finding tweets or texts describing the objects in the videos.

We have benchmarked these models on real-world datasets and used the extractor results during the 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 - head, upper section, and lower section. We segment the body parts by estimating the ratio of each part to the bounding box according to the human body proportions in anatomy. First, 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 the standard RGB values. In the case of multiple colors in a region, the area's color.

WLPD-Video-Dataset. We have collected and labeled over 20 hours of video from different cameras and locations in the city of West Lafayette. Six custom classes with over 12200 images were labeled manually for re-training and testing the YOLO network to detect gender, clothes, and color. Each one-minute chunk of the video consists of around 20 frames, sampled at 3-second intervals.

In the test set from WLPD-Video-Dataset, the clothing colors were recognized with high precision, while the color of the sampled head area were more prone to be affected by the color of the background, as shown in Figure 6.

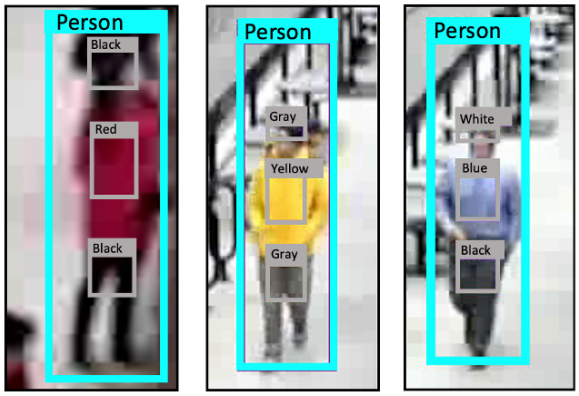


Figure 6: Color recognition on *WLPD-Video-Dataset*

Based on the color information, we can trace the movements of pedestrians across continuous frames. Figure 7 shows the moving routes of two pedestrians walking towards each other. The dotted line after each pedestrian indicates their moving direction.

In cities, multiple cameras are installed at the same traffic cross-section to observe pedestrians from different angles, with each view providing additional information. We wanted to trace the same person across multiple cameras installed at various locations for the missing person search. Figure 8 shows two examples of tracking the same pedestrian passing through three areas. In the left, we track a cycling person wearing a red shirt passing from locations 1 to 3. It takes only 39 seconds since he is cycling. In the right, we follow a walking person wearing a red shirt

Figure 7: Tracking of a pedestrian crossing the street.

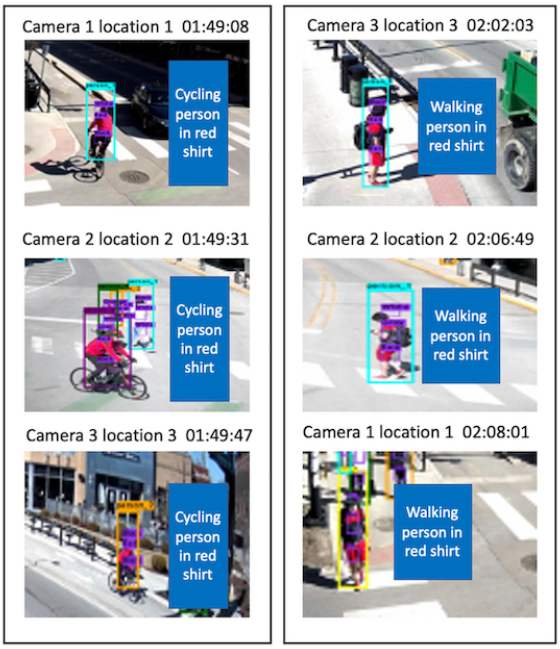


Figure 8: Pedestrian tracking at multiple scenes with multiple cameras.

passing from location 3 to 1 in the opposite direction. It takes him about 6 minutes. So we can map out the walking trajectory of a person as long as there is no change of clothes.

Re-training YOLO.
For gender and clothes detection in video feeds, we re-trained YOLO [19] to identify gender and clothes (shirts and jackets) in video feeds. Hue, saturation and brightness (HSB) for each frame has been analyzed to improve object detection and recognition under night and changing weather conditions. The range of the 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 YOLOv3 and YOLOv4 in Table 1. For gender and clothes detection, we achieved 68% mAP and 67% mAP, respectively, when YOLO is re-trained without pre-trained features.

Human Attributes from Unstructured Text.
Using the stacked (Regular Expression (RE) + Word2Vec) variant of the HART model [20],

Table 1: Mean Average Precision of YOLO for gender and clothes detection in WLPD-Video-Dataset

Object	YOLO v.3	YOLO v.4
Gender	0.59	0.68
Clothes	0.56	0.67

we identified Candidate Sentences (C_s) from the texts of cleaned documents and tweets. We searched for *clothes* with regular expressions on the sentences for finding C_s. If it returns no result, the problem is formulated as a similarity search among all tokens in a sentence, where *clothes* are used as the search token. We used the pre-trained 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 as 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 *Cloths Name and Value Identification* algorithm from [20] which uses Parts-of-Speech (POS) tags of tokens to identify the description.

FemmIR-text Dataset. For benchmark results on text features, we used part of the text data from

[20] consisting of incident reports, press releases, and officer narratives from historical cases. It contains 13 press releases, 40 officer narratives, and five incident reports. Due to privacy reasons, WLPD publicly released only a subset of redacted reports.

Table 2: Evaluation of Human Attribute Extraction on FemmIR-text (Results reported from [20])

Attributes	**Gender**	**Race**	**Clothes** Attr-only	**Clothes** Attr-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	0.90

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 (RE + Word2Vec + POS) model on two evaluation metrics, attribute-only and attribute-value.

Semantic Similarity Search by Topic. We employed a topic-based similarity search to extract documents describing the same objects and attributes found in videos. We also used topic-based similarity search as an additional method for finding candidate sentences. Assuming that each sentence in a document is a mixture of some topics, if any of those topics explain the search phrases, we posit that the sentence is a Candidate Sentence. We used Latent Dirichlet Allocation (LDA) to identify the hidden topics of the sentences in the documents and the query phrases (e.g., clothes, car, person, male). LDA is a generative topic modeling technique where documents are represented as random mixtures over unseen topics, and the topics are derived by calculating distributions over all the words in the document. In this case, we have represented each sentence in the document and the query phrase as individual mixture of topics. For distribution measurement, term frequency–inverse document frequency (Tf-idf) (https://en.wikipedia.org/wiki/Tf-idf) vectors of all tokens in each sentence were used as unigram features. The cosine similarity of the query phrase topic against the topics of the corpus of sentences measures the closest sentence matching the query. We have collected 249857 tweets from 77943 users describing topics related to Cambridge, MA in the *Cambridge-Public-Authority-Tweets (CPAT)* dataset. Figure 9 shows the relevant tweets in the CPAT dataset describing a *PERSON WITH GUN* in *Cambridge Area*.

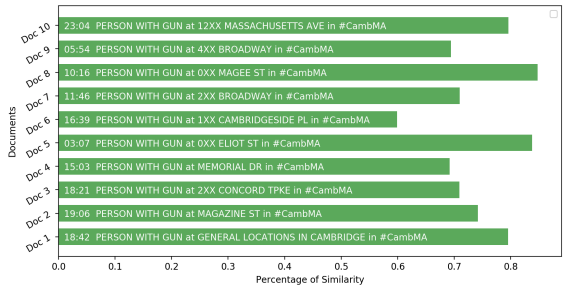


Figure 9: Relevant Tweets with Semantic Similarity Search by Topic.

Demonstration

Finally, we demonstrate Find-Them on the incident reports, press releases, and video feeds from West Lafayette Police Department. We show how it can accurately detect and track a missing person based on non-invasive physical properties and minimize the investigation effort to find a missing person. We describe the user's interaction through six steps, impersonating an officer in WLPD. We annotate each step with a circle in Figure 10.

Step ① (Create Missing Incident or Upload Example): First, the user uploads an incident report, a flyer, or a tweet with a physical description of the missing person with the search area and the search timeline in step (1b). They 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 any examples, they can create a missing person incident by filling out the person's details, the search area, and the timeline, as seen in step (1a).

Step ② (Creation of Predicates): For searching a person, the WLPD officer specified the identifying properties in step 1. Using those inputs, we created an *incident* schema which becomes the search criteria for current and future streaming data in step (2). Triggers in Postgres await for streaming data with similar features to the *incident*, and it notifies the police officer of

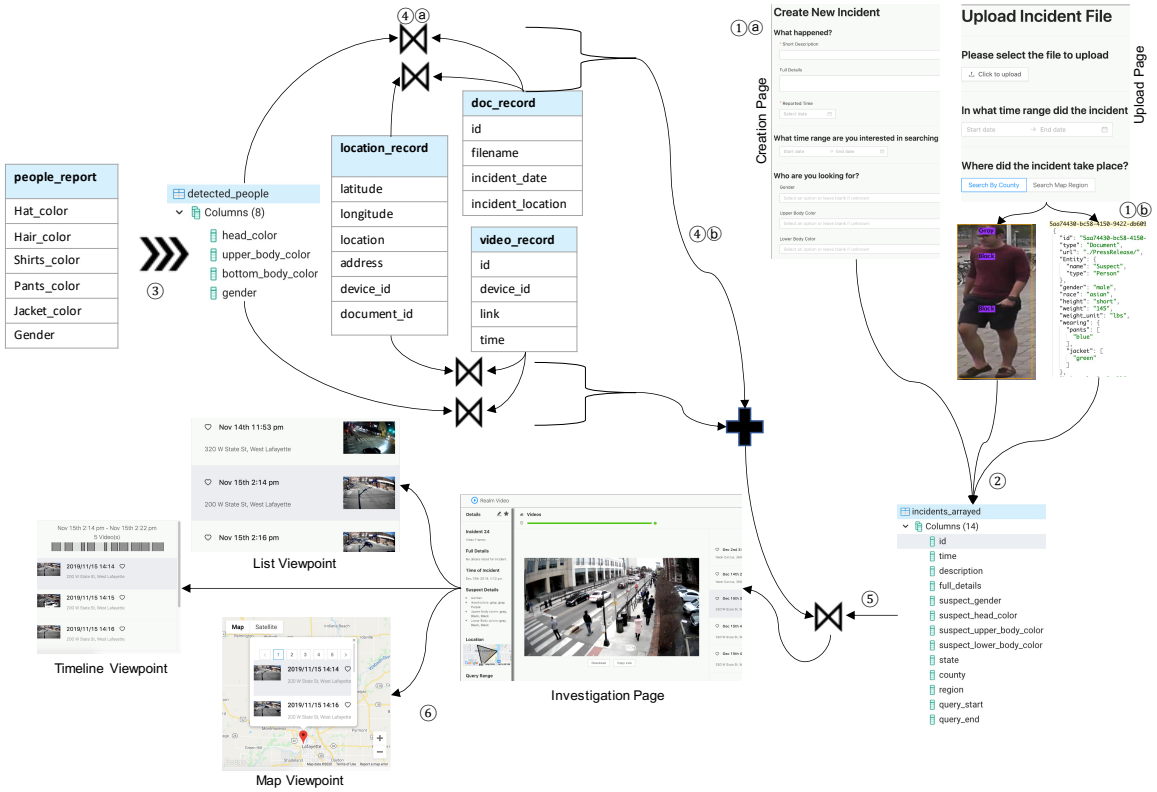


Figure 10: Find-Them Demo.

any matching video feed or tweets. The police officer can always revisit the *incidents* from their search history.

Step ③ (EAR Mapping). As seen in Figure 3a, incident reports have a feature extractor that outputs clothes as individual entities and then extracts their details, whereas, in Figure 3b, we can see the details are extracted in terms of body parts. Both of these modalities need to map to the common EAR model shown in Figure 3c. The system maps the incoming document features to the common EAR model as follows -
(shirts, jackets) → upper body,
(pants, jeans) → lower body, and
(hat, cap) → head.

Step ④, ⑤ (JOIN among Data Sources). Before this step, data from each modality is stored in Postgres 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 were saved in one table, whereas incident location was stored in another. Separation of storage allows us to answer simple queries requiring only one type of information quickly. When the query involves multiple types of information, we create SQL queries to perform JOIN between separate tables representing features from different modalities. The first JOIN in step (4a) separately creates the primary results from each modality. In step (4b), we performed a union of all modalities. Finally, in step (5), we perform a JOIN between the accumulated results and the previously created *incident* table to extract the subset of data objects which match the user search criteria and show the multimodal result on the investigation page.

Step ⑥ (Different Viewpoints). Similar to [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 timeline, whereas the map view allows us to pinpoint a location. The user can also choose their favorite results and can see the filtered result at a later time.

Scalability, Universality, and Multi-user.

Find-Them establishes a common information model, *relational schema* across multiple data sources, and eliminates the need for separate data 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 any training. The system demonstration shows that we could query historical data (in thousands of records) and streaming data in real-time during inference time. For the space constraint, we do not include the time comparisons here. Find-Them is capable of extension to multiple users, each with their own set of preferences in the form of queries and data objects. Since each user has a mapping to the retrieval set with their queries, their queries are kept separate.

CONCLUSION

This paper has introduced Find-Them, a feature-based multimodal data fusion system for analyzing video feeds with other data modalities for finding missing persons. We have described a database backend, along with a schema and a relational query-based fusion method that can scale to a considerably large amount of data volume, along 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 identify the connections between historical and incoming missing cases, giving the law enforcement officers an edge in their investigations. In the future, we plan to grow the video and text datasets by including mobile camera videos (both in-vehicle and body-mounted), city maintenance records, and Bureau of Motor Vehicles records. We also have future goals to include more data modalities and evaluate the effects of humans-in-the-loop on improving performance. Finally, our future work will involve extending the approach to include feature extraction as part of the relevance modeling in an end-to-end deep neural network architecture and modeling user interests based on their historical queries.

ACKNOWLEDGMENT

This research is supported by Northrop Grumman Mission Systems' University Research Program. We are grateful to Professor Michael Cafarella of the 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, B. Bhargava, M. Cafarella, Z. Collins, J. McClellan, A. Sipser, T. Sun, A. Nesen, K. Solaiman, G. Mani, K. Kochpatcharin, P. Angin, and J. MacDonald, "Surveillance video querying with a human-in-the-loop," in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics with SIGMOD*, 2020.
2. J. Wang, X. Zhu, S. Gong, and W. Li, "Attribute recognition by joint recurrent learning of context and correlation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 531–540.
3. G. Pearson, M. Gill, S. Antani, L. Neve, and G. Thoma, "People locator: A system for family reunification," *IT Professional*, vol. 14, no. 03, pp. 13–21, may 2012.
4. R. S. Ferreira, C. G. de Oliveira, and A. A. Lima, "Myosotis: An information system applied to missing people problem," in *Proceedings of the XIV Brazilian Symposium on Information Systems*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2148–2157.
6. S. Aggarwal, V. B. RADHAKRISHNAN, and A. Chakraborty, "Text-based person search via attribute-aided matching," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2617–2625.
7. Z. Wang, Z. Fang, J. Wang, and Y. Yang, "Vitaa: Visual-textual attributes alignment in person search by natural language," in *European Conference on Computer Vision*. Springer, 2020, pp. 402–420.
8. M. Khan and A. Jalal, "A fuzzy rule based multimodal framework for face sketch-to-photo retrieval," *Expert Systems with Applications*, vol. 134, 05 2019.
9. J. Rupnik and J. Shawe-Taylor, "Multi-view canonical

correlation analysis," in *Conference on Data Mining and 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 Transactions on Multimedia*, vol. 20, no. 1, pp. 128–141, 2017.
11. Y. Peng, J. Qi, X. Huang, and Y. Yuan, "Ccl: Cross-modal correlation learning with multigrained fusion by hierarchical network," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 405–420, 2017.
12. W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *International conference on machine learning*. PMLR, 2015, pp. 1083–1092.
13. S. Sah, S. Gopalakrishnan, and R. Ptucha, "Aligned attention for common multimodal embeddings," *Journal of Electronic Imaging*, vol. 29, pp. 023 013 – 023 013, 2020.
14. K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4654–4662.
15. F. Wu, X.-Y. Jing, Z. Wu, Y. mu Ji, X. Dong, X. Luo, Q. Huang, and R. Wang, "Modality-specific and shared generative adversarial network for cross-modal retrieval," *Pattern Recognit.*, vol. 104, p. 107335, 2020.
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, 2021.
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," *Neurocomput.*, vol. 174, no. PA, pp. 50–59, Jan. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2015.01.095
18. S. Palacios, K. Solaiman, P. Angin, A. Nesen, B. Bhargava, Z. Collins, A. Sipser, M. Stonebraker, and J. Macdonald, "Wip - skod: A framework for situational knowledge on demand," in *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*. Cham: Springer International Publishing, 2019, pp. 154–166.
19. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767
20. K. Solaiman and B. Bhargava, "Feature centric multimodal information retrieval in open world environment (femmir)," unpublished.

KMA Solaiman, is currently a Ph.D. Candidate in Computer Science at Purdue University. His research interest is in Multimodal Information Retrieval, Machine Learning, and Heterogeneous Data Mining. Contact him at ksolaima@purdue.edu.

Tao Sun, is currently a System Design and Management Fellow at the Computer Science and Artificial Intelligence Laboratory and Sloan School of Management at M.I.T. Contact him at taosun@mit.edu.

Alina Nesen, is currently a Ph.D. Candidate in Computer Science at Purdue University. Contact her at anesen@purdue.edu.

Bharat Bhargava, is a Professor of Computer Science at Purdue University. He is leading a Northrup Grumman sponsored consortium on Real Applications of Machine Learning (REALM) with MIT, CMU and Stanford. He is contributing to Department of Defense on The Science of Artificial Intelligence and Learning for Open-world Novelty (SAIL-ON) and another project with NGC on explainable AI and adversarial machine learning. He works with Sandia Corporation on science and technology for advancing resilience for contested space (STARCS) to maintain mission capabilities of the US Space Enterprise, Jet Propulsion Lab to predict attacks on space systems and with Ford Corporation on software defined networking for Vehicle to Vehicle communication. He is major thesis advisor of the very first African American woman to receive her Ph.D. in Computer Science department at Purdue in May 2019. He has worked extensively at research laboratories of Air Force and Navy. He has successfully completed several Darpa, Navy STTR and AFRL projects. He has won eight best paper awards in addition to the technical achievement award and golden core award from IEEE, and is a fellow of IEEE. Contact him at bbshail@purdue.edu.

Michael Stonebraker, is a member of the Computer Science and Artificial Intelligence Laboratory at M.I.T. and an Adjunct Professor of Computer Science. He is active in DBMS research, and received the 2014 Turing Award for his contributions in this area. He has also started 10 venture capital backed startups over the course of his career. Contact him at stonebraker@csail.mit.edu.