6-25-2017

A Knowledge Graph Framework for Detecting Traffic Events Using Stationary Cameras

RoopTeja Muppalla
Wright State University - Main Campus, muppalla.4@wright.edu

Sarasi Lalithsena
Wright State University - Lake Campus, lalithsena.2@wright.edu

Tanvi Banerjee
Wright State University - Main Campus, tanvi.banerjee@wright.edu

Amit Sheth
Wright State University - Main Campus, amit@sc.edu

A Knowledge Graph Framework for Detecting Traffic Events Using Stationary Cameras

RoopTeja Muppalla
Kno.e.sis Center, Wright State University
Dayton, Ohio
roopteja@knoesis.org

Sarasi Lalithsena
Kno.e.sis Center, Wright State University
Dayton, Ohio
sarasi@knoesis.org

Tanvi Banerjee
Kno.e.sis Center, Wright State University
Dayton, Ohio
tanvi@knoesis.org

Amit Sheth
Kno.e.sis Center, Wright State University
Dayton, Ohio
amit@knoesis.org

ABSTRACT

With the rapid increase in urban development, it is critical to utilize dynamic sensor streams for traffic understanding, especially in larger cities where route planning or infrastructure planning is more critical. This creates a strong need to understand traffic patterns using ubiquitous sensors to allow city officials to be better informed when planning urban construction and to provide an understanding of the traffic dynamics in the city. In this study, we propose our framework ITSKG (Imagery-based Traffic Sensing Knowledge Graph) which utilizes the stationary traffic camera information as sensors to understand the traffic patterns. The proposed system extracts image-based features from traffic camera images, adds a semantic layer to the sensor data for traffic information, and then labels traffic imagery with semantic labels such as congestion. We share a prototype example to highlight the novelty of our system and provide an online demo to enable users to gain a better understanding of our system. This framework adds a new dimension to existing traffic modeling systems by incorporating dynamic image-based features as well as creating a knowledge graph to add a layer of abstraction to understand and interpret concepts like congestion to the traffic event detection system.

KEYWORDS

Traffic image feature extraction, Knowledge graphs, Traffic events

ACM Reference format:
RoopTeja Muppalla, Sarasi Lalithsena, Tanvi Banerjee, and Amit Sheth. 2017. A Knowledge Graph Framework for Detecting Traffic Events Using Stationary Cameras. In *Proceedings of Industrial KG workshop - ACM Web Science conference, Troy, NY USA, June 2017 (IndustrialKG'17),* 6 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Cities have played a key role in the economy of countries for centuries as a center for income and innovation[9], thus creating a huge influx of rural, foreign, and other migrants in search of job opportunities. With the increasing population of people in larger cities, the infrastructure of cities, such as the road networks, are overwhelmed with growing amounts of congestion as well as adverse events such as accidents. Understanding the relationship between different events within the city can enable city departments to realize the causes of traffic congestion in certain regions at certain times during the week and it can also provide critical information on current traffic conditions and public safety.

Systems such as 511 Traffic[1] provide real-time traffic data from 39 states that can be utilized for extracting time sensitive, actionable information such as information on amber alerts and route detours with providing real-time sensor data from stationary cameras located at intersections across different parts of the cities. While citizen-sensing has played an increasing role in providing real-time traffic information, there remains a need to extract meaningful information from sensor data for better understanding of the traffic conditions in a certain area, especially near city centers, which are more prone to traffic congestion. In particular, there remains a need to harness existing data sources such as traffic camera images for determining the current traffic conditions, both for route planning applications as well as for urban planning policies to decide dates and times for road construction that best minimize traffic delays.

In this work, we address the following research questions: 1) How do we extract features that represent the *dynamic* traffic conditions from the camera imagery?; 2) How can we represent and populate image features in a *knowledge graph* framework as a source of sensor information?; 3) Can we demonstrate the use of the imagery features extracted in 1) and the knowledge graph developed in 2) for *actionable information* to represent dynamic traffic conditions such as congestion? We present our preliminary investigation of traffic events in the greater New York area using traffic camera data from 511ny.org to check the feasibility of using the data to analyze traffic conditions. We also introduce Imagery-based Traffic Sensing Knowledge Graph (ITSKG), a knowledge graph framework to represent and populate the information we gather from the traffic cameras and then to query the knowledge graph to map the relationship between imagery features and traffic events as a means to derive actionable information from raw sensor data.

DOI: 10.475/123_4

[1]http://www.dmv.org/travel/511.php

2 RELATED WORK

There have been studies using image data for traffic event detection, specifically congestion detection. Nidhal et al. [12] used features extracted from traffic image data to detect backlight pairs as a means of detecting the number of vehicles in the scene. However, the performance of the system under different lighting conditions was not described, nor was the denoising process which is critical in this application as it uses more fine-grained image features. Anantharam et al. [1] utilized speed link data from the San Francisco bay area to build "normalcy" models of each link in the city using linear dynamical systems as a means to detect anomalies in the road links. However, missing link data, as well as erroneous sensor data were concerns as there was no means of collecting a gold standard or ground truth data to ensure the validity of the sensor data.

With respect to the knowledge graph applications involving sensor data for anomaly detection, Issa et al. [8] utilized an Android-based GPS tracker on a bus to extract meaning from the GPS sensor data. Specifically, the sensor attributes were described using the SSN ontology[2] to annotate accelerometer readings taken from the GPS tracker which were used to detect vehicular speed as an indicator of traffic conditions. We adapt this idea of utilizing knowledge graphs in corroboration with sensor readings to detect traffic events in the city through the use of imagery features.

There have been relatively fewer studies that have utilized an ontology modeling that relies on multimodal data for event understanding. STAR-CITY [10] presents a system which uses heterogeneous data sources as journey travel times, bus dynamics, social media feeds and event to support traffic analysis. However, STAR-CITY does not consider image-based features. In work using image-based features, Zhang et al. [18] used a probabilistic approach to automatically extract semantic concepts using ImageNet data which was then compared with the WordNet hierarchy. Both image-based and word-based features were extracted to learn the concept similarities in different topics within the ImageNet data. In another approach, Sun et al. [15] described a clustering-based method to extract concepts from a parallel corpus of text and images using ImageNet. In comparison with the ImageNet annotations, the clustering approach outperformed the ImageNet annotation results. In both these studies, the work focused on learning concepts automatically, while we can reuse any such ontologies with appropriate mappings.

In this study, we create an ontological system to maps sensor data to traffic events using a combination of a knowledge-driven and data-driven approach, which can lead to two major advantages: 1) it allows us to use a smaller amount of training data in a more confined search space since we define relationships between different sensor readings using a knowledge graph; 2) it allows for higher accuracy as the learning has some prior knowledge incorporated in it that reduces the errors in detecting unrelated concepts. We discuss the methodology and our use case results in the next sections.

3 METHODOLOGY

Our proposed framework mainly consists of two tasks: (1) extracting image features from dynamic traffic conditions from the camera imagery (section 3.1) and (2) annotating image features using a knowledge graph framework (Section 3.2). Figure 1 describes the main components of our proposed framework.

3.1 Image feature extraction from dynamic traffic conditions

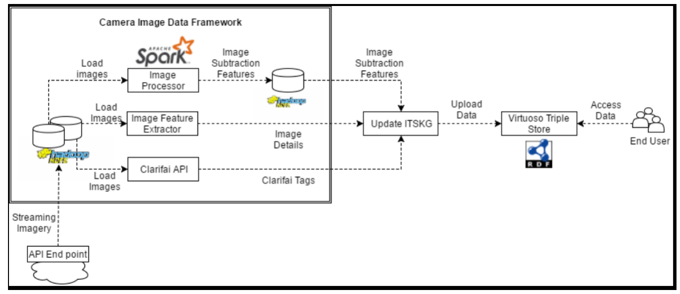

Figure 1: Architecture of our Imagery-based Traffic Sensing Knowledge Graph (ITSKG) framework.

3.1.1 Data Collection. We used the 511ny.org API[3] to obtain and access to around 1,100 camera URLs in and around New York city. In particular, we used the traffic images coming from cameras from the New York State Department of Transportation (NYSDOT). We selected 463 of these cameras based on better coverage, reliability and focus on the traffic. These cameras generate traffic images every 10 - 30 seconds. For this study, data was collected from the selected cameras once every minute for a period of 2 months between September and October in 2016.

3.1.2 Feature extraction. Even though many traffic sensing detectors such as loop detectors, infrared and radar sensors [16, 17] can sense traffic fairly well, these sensors are prone to device failure, obstruction due to rain and fog, and are tedious to install and maintain. Due to this, image processing and video processing have become increasingly popular because they can be controlled and maintained in a distributed environment to perform both surveillance and traffic analysis in real-time. With this motive, many studies [7, 14] have focused on tracking vehicles and detecting objects in traffic, which makes use of background subtraction (also known as foreground detection, where an image's foreground is extracted for processing). It is common to observe traffic spikes during rush hours, but it is crucial to detect traffic when it deviates from such normality. Background subtraction can be used to get pixel-level differences between the images that can reflect the temporal changes in the traffic. In this study, we make use of hourly and quarterly references along with background subtraction to observe relative traffic that deviates from what is normal.

To accomplish this, the streaming data of traffic camera images were collected from the NYSDOT endpoint and stored in an Apache Hadoop Distributed File System (HDFS) standalone cluster. We implemented image median and subtraction calculation as services in the Spark. Once the images were loaded, a Spark service was called to compute median images. A median image is defined as the pixel-wise median across the set of images. This service calculates an hourly median for the images collected within an hour and a quarterly median for the images collected within 6 hours. This

[2] https://www.w3.org/TR/vocab-ssn/

[3] https://511ny.org/developers/help

resulted in 24 hourly median images and 4 quarterly median images (00:00 - 6:00, 6:00 - 12:00, 12:00 - 18:00, 18:00 - 24:00) for every day. This Spark service also performs image subtraction (where the magnitude of the differences in the intensity values of each individual pixel is computed between two images) with the help of python packages such as NumPy and SciPy, for each image with their respective hourly and quarterly medians to get relative traffic. For example, Figure 2 shows a traffic image (Figure 2a) at time 12:46 pm, which was then subtracted from its 2 medians:
(i) hourly median (Figure 2b) calculated based on the images between 12:00 pm and 1:00 pm,
(ii) quarterly median (Figure 2c) calculated based on the images between the daily quarter that the current image belonged to i.e. between 12:00 pm and 6:00 pm (since the image was in the second quarter of the day).

Once the image subtraction had been computed (Figures 2d & e), the result was quantified by using Manhattan norm (the sum of the absolute values between each pixel) per pixel to measure how much the image had changed. In this case, the result for the hourly and quarterly change were (i) 35.35 (from the hourly median) and (ii) 36.36 (from the quarterly median) respectively. This information along with its image identifier was then stored into HDFS.

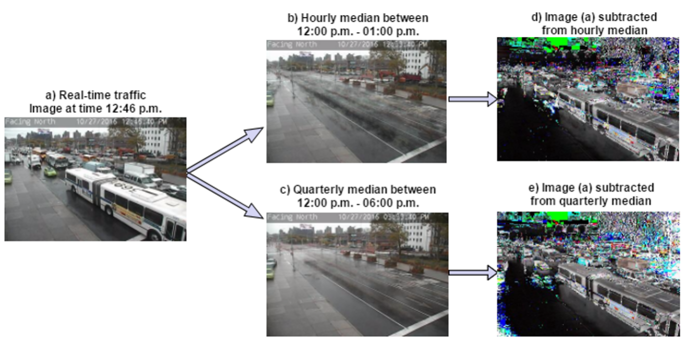

Figure 2: Image Subtraction (median images) from real-time traffic image captured at 12:46pm on 2nd Avenue and 125 St, Manhattan.

Features such as image dimensions and timestamp were extracted with the help of a feature extractor. These images were also passed on to the Clarifai API[4] which recognizes objects and tags each image with the help of advanced machine learning techniques as shown in Figure 3. All these features, along with the camera coordinates, were combined to add a semantic layer and generate the knowledge graph which is discussed in the next section.

3.2 Image feature annotation using knowledge graph framework ITSKG

Semantic Web techniques have played a key role as a platform for integrating, publishing, and sharing data in a number of domains including government, medicine, education, etc. The use of Semantic Web techniques enable us to define or describe a domain model for a given domain and hence improves the understanding of the domain. These domain models can then serve as the schema to populate the raw data. The generated data can be shared and later

[4]https://developer.clarifai.com/

Figure 3: Tags generated by clarifai API for real-time traffic image captured at 12:30pm on 2nd Avenue and 125 St, Manhattan.

on can be integrated with other similar types of data. Furthermore, this data can easily be used for analytical and intelligent applications. Motivated by these, we create a knowledge graph framework using Semantic Web techniques to annotate and publish image data collected by various means in the context of traffic events. The knowledge graph framework allows us to (1) model traffic-related features extracted using images, and (2) annotate the raw image features as semantic data using the model defined in task 1. We describe each of these steps below.

3.2.1 Semantic Modelling - Imagery-based Traffic Sensing Knowledge Graph (ITSKG). By adhering to the principle of reusing existing ontologies, we adopt and extend the W3C Semantic Sensor Network (SSN) ontology to describe traffic imagery information. The SSN ontology provides sensors, sensor observations, and the knowledge of the environment. It is being quickly adopted by many applications [2, 8, 13], including traffic-related studies [3, 4] and is considered as a standard ontology for Semantic Sensor Networks. We consider a traffic camera as a sensing device, a camera image as sensor output, and traffic as an observation. In particular, SSN serves as an upper level schema to the knowledge graph for Sensing Traffic Imagery.

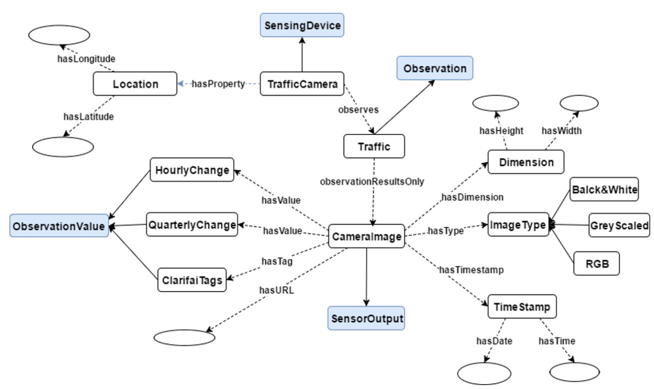

Figure 4: Semantic Model for the Imagery-based Traffic Sensing Knowledge Graph (ITSKG).

As shown in Figure 4, Imagery-based Traffic Sensing Semantic Model in the knowledge graph contains information about traffic cameras and camera images. In order to describe a sensor device, sensor output, and observations, we use existing concepts in SSN

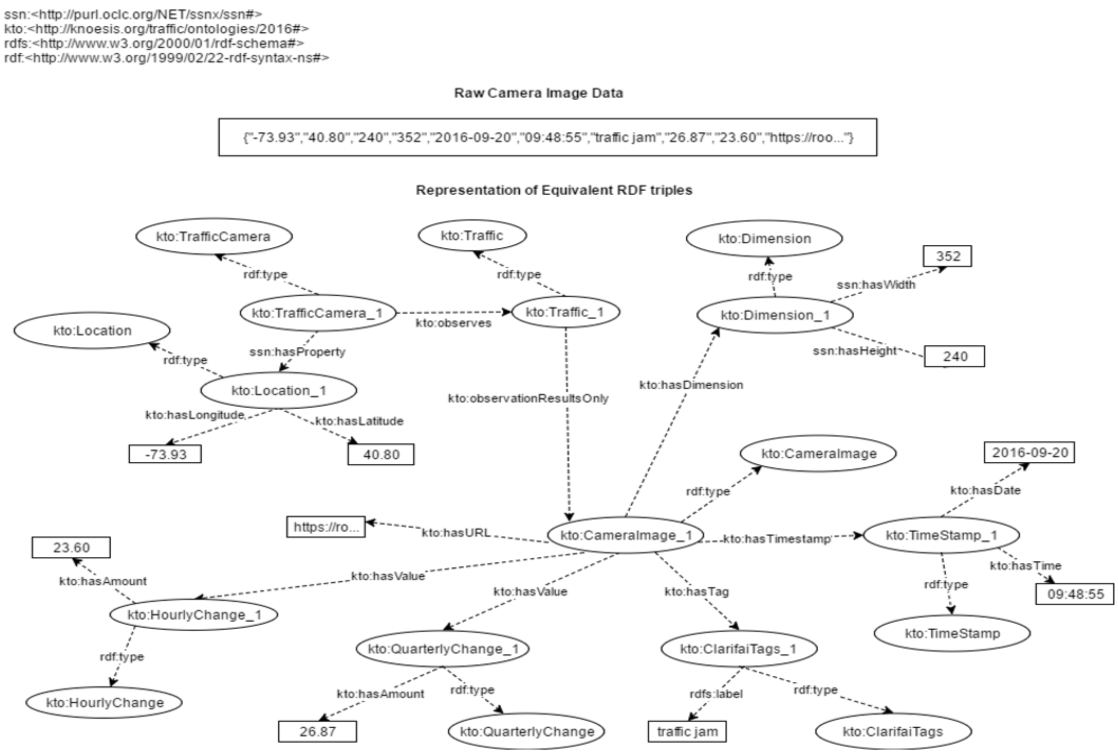

Figure 5: Example of raw imagery output converted to its equivalent RDF triples.

as given in Figure 4. The **TrafficCamera, Traffic, CameraImage** classes in ITSKG are subclasses of the SSN classes **SensingDevice, Observation**, and **SensorOutput** respectively. Table 1 shows the properties captured in **TrafficCamera** and **CameraImage** classes. We use the OWL [6] language to define the semantic model using Protege[5][5][6].

Table 1: Properties used to describe TrafficCamera and CameraImage.

	Properties
Traffic Camera	Latitude, Longitude
Camera Image	Time stamp, Hourly change, Quarterly change, Clarifai tags, URL, Image width, Image height

3.2.2 Annotate raw image features using ITSKG. We use the semantic model described above to populate raw data collected in the data collection step. The input data is stored in the form of an XML file and then we convert this input data into the facts in our knowledge graph using the semantic model described above. Converted data is then loaded into a RDF triple store. A sample of the raw data and its equivalent RDF triple can be seen in Figure 5. The populated data is stored in a Virtuso graph database to make it accessible to the interested parties[7]. End users can access this information using SPARQL queries through the Virtuoso SPARQL endpoint.

[5]http://protege.stanford.edu/
[6]The schema can be found at https://roopteja.bitbucket.io/trafficCams/
[7]The Virtuoso querying endpoint can be found at http://130.108.85.135:8891/sparql

4 EXAMPLE USE CASE

4.1 Data Source and Feature Extraction

Since this is a feasibility study, we focused our study on a few use cases (and describing one use case in detail). We initially collected the imagery data for the cameras which are located in and around the most accident-prone areas in New York, such as the intersections between Tillary St. and Flatbush Ave. Brooklyn, E. 138th St. and Alexander Ave. Bronx, 2nd Ave. and E. 59th St. Manhattan. These intersections are identified as the most dangerous intersections to drive in NYC[8,9]. However, this dataset also contained a large volume of imagery data, collecting images for every minute from about 50 cameras. It became a tedious task to wait for an incident to happen and manually verify the incident. So, we restricted our focus to the cameras near Yankee Stadium, which is home to many sport events, where a high frequency of accidents take place. In particular, we collected the imagery from the 10 cameras near Yankee Stadium within the bounding box of longitude and latitude of [-73.934877,40.800904] SW and [-73.919728,40.830720] NE.

In order to identify events of interest, we narrowed our search to specific dates and times of incidents identified by formal traffic event sources such as 511. Our use case in this study is focused on one particular traffic camera at 2nd Avenue and 125 St. Manhattan, where an event occurred on October 27, 2016 as reported by 511 New York on their official twitter page as *"Update: Closure on #WillisAvenueBridge from Manhattan Side to Bronx Side"* at 12:45 pm. Once the data was collected, we then performed image subtraction

[8]http://www.residentmar.io/2016/03/23/worst-places-to-drive.html
[9]http://gothamist.com/2016/04/01/most_dangerous_intersections_nyc.php

for each image with their respective hourly and quarterly medians. However, our approach is generic enough and scalable enough to deal with multiples cameras.

4.2 Query Formulation

Since our aim in this study was to analyze traffic patterns as a means to detect abnormal conditions, we need to extract the set of images which would deviate the most from their respective median. For this, we used a threshold that filtered out the images whose magnitude of hourly or quarterly change was approximately 2 standard deviations from their respective hourly or quarterly median.

4.2.1 Hourly Change. Hourly change is useful to sense the traffic within an hour and can help us detect recent changes in the traffic conditions. The query (as shown in Figure 6) was used to get the list of images between 12:00 pm and 1:00 pm which deviate the most within the hour. This query gives the URL for the image and the hourly change of that image with respect to the hourly median. Queries were thresholded using a margin of one standard deviation from the median value of the hourly or quarterly value. Figure 7 shows a couple of images given by the hourly query.

```
select (?cameraurl as ?CameraURL) (?amount as ?HourlyChange)
{
?camera <http://knoesis.org/traffic/ontologies/2016#observes> ?traffic .
?traffic <http://knoesis.org/traffic/ontologies/2016#observationResultsOnly> ?cameraimage .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasTimestamp> ?timestamp .
?timestamp <http://knoesis.org/traffic/ontologies/2016#hasTime> ?time .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasUrl> ?cameraurl .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasValue> ?hourlychange .
?hourlychange <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://knoesis.org/traffic/ontologies/2016#HourlyChange> .
?hourlychange <http://knoesis.org/traffic/ontologies/2016#hasAmount> ?amount .
filter (hours(?time) = 12 && ?amount > 25.5) .
}
```

Figure 6: SPARQL query to get the traffic images based on hourly change.

Figure 7: Sample images returned from hourly change query (query provided above).

4.2.2 Quarterly Change. Sometimes we cannot rely on hourly change to sense the traffic. For instance, if there was a traffic jam due to an incident for a considerable portion of the hour, using only the hourly information may not be sufficient to sense the traffic movement or congestion. In such scenarios, quarterly change can be useful to detect such anomalies since it considers a wide range of time. The query (as shown in Figure 8) was used to get the list of images between 12:00 pm and 1:00 pm which deviate the most within a quarter (6 hours). This result also contains the set of images from the hourly query along with another set of images as shown in Figure 9.

```
select (?cameraurl as ?CameraURL) (?amount as ?QuarterlyChange)
{
?camera <http://knoesis.org/traffic/ontologies/2016#observes> ?traffic .
?traffic <http://knoesis.org/traffic/ontologies/2016#observationResultsOnly> ?cameraimage .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasTimestamp> ?timestamp .
?timestamp <http://knoesis.org/traffic/ontologies/2016#hasTime> ?time .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasUrl> ?cameraurl .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasValue> ?quarterlychange .
?quarterlychange <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://knoesis.org/traffic/ontologies/2016#QuarterlyChange> .
?quarterlychange <http://knoesis.org/traffic/ontologies/2016#hasAmount> ?amount .
filter (hours(?time) = 12 &&   ?amount > 28.5) .
}
```

Figure 8: SPARQL query to get the traffic images based on quarterly change.

Figure 9: Sample images returned from quarterly change query (query provided above).

4.2.3 Use of the Clarifai API. This system also makes use of Clarifai API for traffic sensing through the use of class labels using object recognition. For example, the query (as shown in Figure 10) gave a set of 43 images between 12:00 pm and 1:00 pm which are labelled as *'traffic jam'* by the Clarifai API[10].

```
select (?cameraurl as ?CameraURL)
{
?camera <http://knoesis.org/traffic/ontologies/2016#observes> ?traffic .
?traffic <http://knoesis.org/traffic/ontologies/2016#observationResultsOnly> ?cameraimage .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasTimestamp> ?timestamp .
?timestamp <http://knoesis.org/traffic/ontologies/2016#hasTime> ?time .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasUrl> ?cameraurl .
?cameraimage <http://knoesis.org/traffic/ontologies/2016#hasTag> ?hasTag .
?hasTag rdfs:label ?label .
filter (hours(?time) = 12 && ?label = "traffic jam") .
}
```

Figure 10: SPARQL query to get the traffic images mentioned by Clarifai API.

We further explored the use of Clarifai API to test its reliability. For example, Figure 11a was one of the results generated by the query labelled as *'traffic jam'* by Clarifai API with a probability of 0.78, whereas Figure 11b was not labelled as *'traffic jam'* by the same Clarifai API. Whereas, the ITSKG system gave Figure 11b in the result with the mentioned threshold and not Figure 11a. This clearly shows that Clarifai API was not able to sense the traffic while the ITSKG system, though not entirely accurate, was good enough to sense the traffic in the city. The Clarifai API gave inconsistent results which fail to address the main purpose of this study.

4.3 Results and Discussion

Three annotators independently labeled 60 images from noon to 1:00 pm as either high congestion or low congestion. The inter-rater agreement was measured using Fleiss Kappa [11], achieving a performance of 0.71. From Table 2, we see that the performance using the Clarifai tags alone, i.e. without any of the dynamic image-based

[10]The query result is provided in https://roopteja.bitbucket.io/trafficCams/

a) Real-time image labelled as '***traffic jam***' by clarifai API but not by ITSKG system

b) Real-time image labelled as '***traffic jam***' by ITSKG system but not by clarifai API

Figure 11: Clarifai API vs. ITSKG system.

features from the ITSKG system is not very promising (precision <0.5). However, both the quarterly and hourly change metrics have fairly high values of precision and recall. This is not surprising since traffic conditions change vastly on an hour-to-hour basis. Moreover, the labeling of the individual images as high or low without having intermediate levels also adds subjectivity that affects the current evaluation framework (Figure 12). However, despite the subjective nature of the evaluation, the performance of our ITSKG system looks promising. Based on the aggregation of the number of image frames detected as having a high congestion, we were able to detect high congestion for the event from October 2016.

Table 2: Results of the different components in the ITSKG system for Use Case Example.

Query	Precision	Recall	F1 Score
Quarterly Change	**0.69**	**0.77**	**0.73**
Hourly Change	0.63	**0.77**	0.69
Clarifai	0.4	0.65	0.5

Figure 12 shows sample images from the query results for the quarterly and hourly change indicating that the ITSKG system labeled these images as depicting high congestion, although the annotators did not.

Figure 12: Sample images detected as high congestion by our ITSKG system, but marked as not high congestion by the annotators.

5 CONCLUSION AND FUTURE WORK

With cities growing rapidly, traffic management has become a high priority for improving the traffic flow and minimize wasted time. It is important to detect incidents such as accidents, closures etc., and clear them to reduce the impact on traffic. The problems with existing detectors have led to the usage of traffic cameras to represent real-time traffic. Our proposed ITSKG framework has the potential to identify dynamic traffic conditions from camera imagery as shown in our example use case. This framework is well integrated with the existing Semantic Sensor Network (SSN) and aids in analysing heterogeneous streams of sensor data to extract meaningful information. Such system can help transportation agencies provide real-time traffic analysis and help in managing traffic. For our future work, we will address the limitations of trafffic cameras such as weather, lighting, maintenance, and proper placement with respect to line-of-sight and incorporate advanced image processing algorithms. Social media features can also be incorporated into the current ITSKG framework, adding to the system by aggregating information across heterogeneous sensors.

ACKNOWLEDGMENTS

Research reported in this publication was supported in part by NIH of the National Institutes of Health under award number 1K01LM012439-01. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES

[1] Pramod Anantharam, Krishnaprasad Thirunarayan, Surendra Marupudi, Amit P Sheth, and Tanvi Banerjee. 2016. Understanding City Traffic Dynamics Utilizing Sensor and Textual Observations.. In *AAAI*. 3793–3799.

[2] Stefan Bischof, Athanasios Karapantelakis, Cosmin-Septimiu Nechifor, Amit P Sheth, Alessandra Mileo, and Payam Barnaghi. 2014. Semantic modelling of smart city data. (2014).

[3] Susel Fernandez, Rafik Hadfi, Takayuki Ito, Ivan Marsa-Maestre, and Juan R Velasco. 2016. Ontology-Based Architecture for Intelligent Transportation Systems Using a Traffic Sensor Network. *Sensors* 16, 8 (2016), 1287.

[4] Susel Fernandez and Takayuki Ito. 2016. Using SSN Ontology for Automatic Traffic Light Settings on Inteligent Transportation Systems. In *Agents (ICA), IEEE International Conference on*. IEEE, 106–107.

[5] John H Gennari, Mark A Musen, Ray W Fergerson, William E Grosso, Monica Crubézy, Henrik Eriksson, Natalya F Noy, and Samson W Tu. 2003. The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies* 58, 1 (2003), 89–123.

[6] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F Patel-Schneider, and Sebastian Rudolph. 2009. OWL 2 web ontology language primer. *W3C recommendation* 27, 1 (2009), 123.

[7] Bo-Jhen Huang, Jun-Wei Hsieh, and Chun-Ming Tsai. 2017. Vehicle Detection in Hsuehshan Tunnel Using Background Subtraction and Deep Belief Network. In *Asian Conference on Intelligent Information and Database Systems*. Springer, 217–226.

[8] Hassan Issa, Ludger van Elst, and Andreas Dengel. 2016. Using smartphones for prototyping semantic sensor analysis systems. In *Proceedings of the International Workshop on Semantic Big Data*. ACM, 7.

[9] Neil Kleiman, Adam Forman, Jae Ko, David Giles, and Jonathan Bowles. 2013. Innovation and the City. *Center for an Urban Future* (2013).

[10] Freddy Lécué, Robert Tucker, Simone Tallevi-Diotallevi, Rahul Nair, Yiannis Gkoufas, Giuseppe Liguori, Mauro Borioni, Alexandre Rademaker, and Luciano Barbosa. 2014. Semantic traffic diagnosis with star-city: Architecture and lessons learned from deployment in dublin, bologna, miami and rio. In *International Semantic Web Conference*. Springer, 292–307.

[11] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica* 22, 3 (2012), 276–282.

[12] Ahmed Nidhal, Umi Kalthum Ngah, and Widad Ismail. 2014. Real time traffic congestion detection system. In *Intelligent and Advanced Systems (ICIAS), 2014 5th International Conference on*. IEEE, 1–5.

[13] Riccardo Petrolo, Valeria Loscrì, and Nathalie Mitton. 2015. Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms. *Transactions on Emerging Telecommunications Technologies* (2015).

[14] S Cheung Sen-Ching and Chandrika Kamath. 2004. Robust techniques for background subtraction in urban traffic video. In *Electronic Imaging 2004*. International Society for Optics and Photonics, 881–892.

[15] Chen Sun, Chuang Gan, and Ram Nevatia. 2015. Automatic concept discovery from parallel text and visual corpora. In *Proceedings of the IEEE International Conference on Computer Vision*. 2596–2604.

[16] Carlos Sun, Stephen Ritchie, and Kevin Tsai. 1998. Algorithm development for derivation of section-related measures of traffic system performance using inductive loop detectors. *Transportation Research Record: Journal of the Transportation Research Board* 1643 (1998), 171–180.

[17] Scott D Thompson. 1996. A microwave radar for vehicular applications. In *Microwave and Millimeter-Wave Monolithic Circuits Symposium, 1996. Digest of Papers., IEEE 1996*. IEEE, 7–9.

[18] Hao Zhang, Zhiting Hu, Yuntian Deng, Mrinmaya Sachan, Zhicheng Yan, and Eric P Xing. 2016. Learning concept taxonomies from multi-modal data. *arXiv preprint arXiv:1606.09239* (2016).