# TVDP: Translational Visual Data Platform for Smart Cities

Seon Ho Kim, Abdullah Alfarrarjeh, George Constantinou, Cyrus Shahabi
*Integrated Media Systems Center, University of Southern California, Los Angeles, CA 90089, USA*
⟨ *seonkim, alfarrar, gconstan, shahabi* ⟩*@usc.edu*

***Abstract*—This paper proposes a platform, dubbed "*Translational Visual Data Platform* (*TVDP*)", to collect, manage, analyze urban visual data which enables participating community members connected not only to enhance their individual operations but also to smartly incorporate visual data acquisition, access, analysis methods and results among them. Specifically, we focus on geo-tagged visual data since location information is essential in many smart city applications and provides a fundamental connection in managing and sharing data among collaborators. Furthermore, our study targets for an image-based machine learning platform to prepare users for the upcoming era of machine learning (ML) and artificial intelligence (AI) applications. TVDP will be used to pilot, test, and apply various visual data-intensive applications in a collaborative way. New data, methods, and extracted knowledge from one application can be effectively translated into other applications, ultimately making visual data and analysis as a smart city infrastructure. The goal is to make value creation through visual data and their analysis as broadly available as possible, thus to make social and economic problem solving more distributed and collaborative among users. This paper reports the design and implementation of TVDP in progress and partial experimental results to demonstrate its feasibility.**

***Keywords*-geo-tagging, visual data management, machine learning, smart city, data as infrastructure**

## I. Introduction

Visual data such as images and videos are widely being captured and available, forming a new source of communication, information and knowledge for smart city applications (e.g., street cleanliness [1], traffic flow analysis [2], situation awareness of disasters [3], road damage detection [4], material recognition [5], and visual information generation [6] [7]). Especially, mobile cameras (smartphone, drone, vehicle blackbox, etc.) are recording every corner of urban streets for various purposes and such visual data by itself become an important asset carrying plentiful information about urban life. Furthermore, it will be even more critical as advanced technologies such as machine learning and artificial intelligence automatically analyze visual data and extract new useful information and knowledge hidden in raw data.

Visual data as infrastructure promises great potential in a smart city so that various applications are being developed by many organizations to enhance the functions of the city in a wide scope from a simple reporting of graffiti (e.g., MyLA311 App [8]) to a sophisticated intelligent surveillance system for terrorism. The recent rapid development of machine learning techniques such as deep learning provides us with a great opportunity to utilize such data in a truly democratic way [9]. Because machine learning and artificial intelligence have tremendous potential for value creation, people with intelligence to create great things can come from anywhere, and there is a need to make sure that these people can readily access the data, knowledge, and analysis tools they will need to realize their full potential. When we believe visual data as infrastructure, then democratizing the use of visual data is a rational strategy. However, there are still many challenges in managing visual data, especially combined with the automatic collection, management, and analysis.

One critical observation is that current applications and approaches so far have been fragmented and provided individual solutions for a single problem without systematic, synergistic ways and real practical infrastructures to be used and shared among community members with similar interests. For example, organizations will gain benefits from individual machine learning applications, even if they are not integrated into a larger whole. However, the benefits become much greater when these applications are on an integrated platform [10]. First, considering that visual data collection and analysis (such as deep learning) are very expensive, especially in big urban scale applications, it is critical for users to collectively work together for mutual benefits by sharing collected data and analysis tools in a platform. A benefit of having a unified platform is that participants can use it to create a single point of access to data needed for machine learning. The larger and richer the dataset, the more accurate the results. Furthermore, a well-designed platform can 1) maximize the availability and reusability of data, the utility of data management and analysis methods, and 2) minimize redundant data and analysis, effort and cost of their operations with an easy and effective working environment.

Thus, we propose a platform, dubbed "*Translational Visual Data Platform* (*TVDP*)", to collect, manage, analyze urban visual data, which enables participating users to share their data and tools not only to enhance their individual operations but also to mutually benefit by efficiently sharing visual data acquisition, access, analysis methods and results. Specifically, our proposal focuses on geo-tagged visual data since location information is essential in most smart city applications and provides a fundamental connection in managing and sharing data among collaborators. Furthermore,

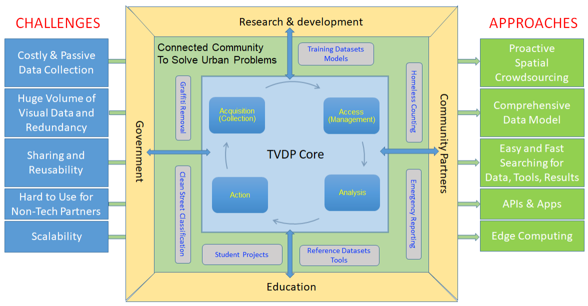


Figure 1: Overview of the TVDP Platform

our study targets for an image-based machine learning platform, not only a highly efficient visual data sharing platform, to prepare users for the upcoming era of machine learning (ML) and artificial intelligence (AI) applications. TVDP will be used to pilot, test, and apply various image-based smart city applications among participating users to improve functions and quality of urban life. TVDP does not provide only sharing data and acquisition methods, but also analysis methods and results so that new data, methods, and extracted knowledge from one application can be effectively translated into other applications, ultimately making visual data and analysis as urban infrastructure. The goal is to make value creation through visual data and their analysis as broadly available as possible, thus to make social and economic problem solving for a smart city more distributed and collaborative among community partners.

## II. Platform Design

In order to achieve the goals of a unified platform, TVDP provides four core services (Fig. 1):

- **Acquisition:** data collection methods including uploading and crowdsourcing,
- **Access:** data management including archiving, indexing, and searching,
- **Analysis:** data analysis including machine learning, and
- **Action:** An edge-computing component which dispatches analysis models to various devices based on their capabilities and re-uses the generated results on edges for enhancing future analysis models.

These four core services form a cycle of data pipeline which can be applied for most data-centric applications. Using all or a part of the core services, users can utilize TVDP for their own purposes or collaboration among them. It can be as simple as quickly searching for an image (i.e., using only access), or as complex as IoT-based city-wide monitoring application with smart camera sensors (i.e., continuously using the full cycle). In particular, these core services can be used for implementing a smart city application individually or collaboratively among different participants such as 1) *governments*: who provide an open dataset for a problem of interest, 2) *professional researchers and developers*: who provide algorithms and technical solutions, 3) *community partners*: who seamlessly act either by operating the technical solution or crowdsourcing new data, and 4) *academic partners (e.g., students)*: who can use the open datasets for building or extending analysis modules.

Our focus is on geo-tagged visual data for smart city applications; thus our primary target datasets consist of street images and videos with location metadata. An example scenario of TVDP in this study is as follows: 1) a government, Los Angeles Sanitation Department (LASAN), periodically collects and uploads videos of street scenes while their garbage collection trucks are being operated on the streets (**individual data acquisition for its own purpose**), 2) the collected videos are organized in a spatial-temporal way and being analyzed to automatically detect certain objects for street cleaning (i.e., a classification model built for recognizing objects on the streets such as illegal dumping, abandoned furniture, encampment of homeless people) by USC researchers (**shared data access and collaborative analysis**), 3) the classified result and location will be reported to LASAN for quick street cleaning (action) and stored back as augmented knowledge in the TVDP database, 4) one of the classification results, encampment, can be directly used by another user, Homeless Coordinator of City of Los Angeles, to trace the locations of homeless tents (**shared analysis results**) and perform its own study to solve a social problem, such as clustering of tents in Los Angeles, spatio-temporal movements of homeless people (**social impact**), 5) another department can analyze the same collected data and annotated results from previous analysis for a different purpose (e.g., graffiti detection, correlation study between the levels of both street cleanliness and safety) saving time and effort in learning (**translational data science**).

A general platform for visual data has the following challenges: 1) passive data collection may not provide enough quality datasets to achieve a certain learning, 2) visual data is huge in size and many times redundant, so it is very hard to manage and search efficiently, 3) there are few systematic ways to share data collection methods, storage, access methods, processing as reusable resources among collaborators, 4) it is not easy for non-technical partners to use the collected data and methods for their own purposes, and 5) scalable system is needed for large-scale city applications. To overcome these challenges, our proposed platform provides the following approaches (Fig. 1):

- Spatial crowdsourcing: to collect data proactively by enabling a participant to create a data collection campaign for certain types of visual data at specific locations.
- Comprehensive modeling of collected data: to represent different types of metadata including spatial, descriptive text, visual features to support the usability of the collected data for a target analysis and various types of future analysis (i.e., translational).

- Effective data management methods: to store, access, and query the visual data in a management system that uses the comprehensive model of data.
- APIs and sample apps: to support easy use of the platform and to quickly develop applications or prototypes in economic and democratic ways.
- Edge computing: to distribute computing intensive processing to edge devices for scalability which is essential in a large-scale platform.

The following sections describe our proposed approaches one by one in details.

## III. Data Acquisition

The availability of image datasets which cover streetscapes is a fundamental challenge for developing various smart city applications. There are some APIs (e.g., Google Street View, Flickr) and public datasets (e.g., GeoUGV [11]) that provide geo-tagged images for streets. However, such datasets are passively collected which might result in a shortage of the images required for developing an application of interest. Thereafter, when visual data is needed for a certain application at an urban scale, a well-planned proactive data collection is critical. In particular, proactive data collection can be performed using spatial crowdsourcing [12] [13] which refers to the practice of engaging a group of people for performing a task at a specific location. Spatial crowdsourcing has been shown to be a cost- and time-efficient way to acquire geo-tagged visual data (e.g., in disasters [14]) and then to analyze the collected data too [15]. There exist platforms for crowdsourcing mobile video collection along with spatial metadata at fine granularity (i.e., each frame of the collected video is tagged with spatial metadata), such as MediaQ [16].

Thereafter, the adequacy of the collected data should be evaluated by estimating its coverage by utilizing its associated spatial metadata. In particular, the platform uses the spatial measurement models that consider the spatial properties of the images (e.g., the spatial extent of a view and viewing direction) [17]. Consequently, iterative spatial crowdsourcing can be performed towards assuring the sufficiency of the available data.

## IV. Data Management

The data management component of the TVDP platform comprises three parts: data modeling, data storage, and data access.

### A. Data Modeling

TVDP represents visual data (i.e., images or videos) using a comprehensive model that comprises various properties. These properties include visual, spatial, annotation, textual, and temporal descriptors. Some of these descriptors require either human intervention (e.g., textual descriptor), sensors

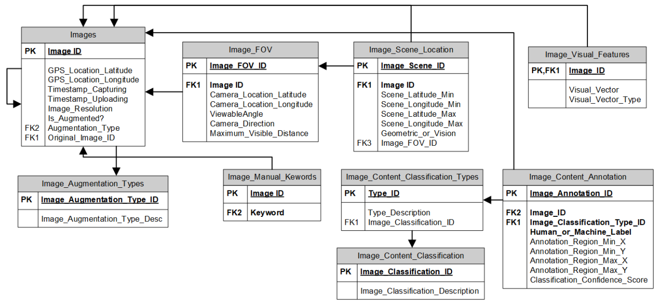


Figure 2: ER-Diagram of the TVDP Data Schema

(e.g., spatial descriptor), or engineering methods (e.g., visual descriptor), while other descriptors can be obtained using multiple methods (e.g., annotation descriptor using human or engineering method). In what follows, we discuss each of these descriptors.

**Visual Descriptors:** There are various ways to extract visual feature vectors to represent image content. Here are some of the widely used ones in image-based machine learning applications.

- *Color Histogram:* The color histogram [18] is generated by counting the image pixels belonging to each color range of the discretized color space.
- *SIFT-based Bag of Words (SIFT-BoW):* SIFT [19] is a vision algorithm for detecting interesting points which lie on the high-contrast regions of images. Subsequently, the SIFT-BoW is a vector representation of the occurrences of words consisted of SIFT features.
- *Convolutional Neural Network (CNN) based Features:* CNN showed successful results for various computer vision tasks (e.g., image classification [20], object detection [21]). Subsequently, it is also used to extract a rich feature vector for representing image content.

**Spatial Descriptors:** There are different ways to represent the spatial context of an image. Here is a summary of them.

- *GPS Location*: A straightforward spatial representation of an image is the point location of the camera at the image capturing time using GPS. The limitation of the camera location is that the scene depicted in an image can be located in any direction and distance from the camera.
- *Field of View (FOV) [22]*: Utilizing various built-in sensors (e.g., GPS, and digital compass) associated with smart cameras, various spatial metadata (i.e., camera location $L$, viewing direction $\theta$, viewable angle $\alpha$, and maximum visible distance $R$) can be captured to represent the spatial extent of an image. The FOV descriptor (see Fig. 3) provides a more accurate representation than the camera location descriptor because it specifies the viewing direction of the image.
- *Scene Location* [23]: Using the FOV descriptor, the scene location can be calculated which is the minimum bounding box surrounding the geographical region de-

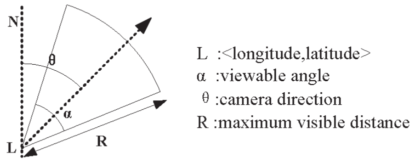


Figure 3: Image Field-of-View (FOV) Representation

picting the image scene. The image scene location most accurately represents the semantic spatial extent of the scene depicted in the image.

**Annotation Descriptors:** Understanding visual content correctly has turned out to be a very challenging task. Two main approaches have been utilized to annotate videos and images. First, manual annotations by human users have been the most practical and preferred way to identify the class (i.e., label) of the visual content. However, manual annotations are both error-prone and time-consuming. Alternatively, the visual data can be annotated using an available ML model (referred to as machine-based annotation), and each annotation is associated with a confidence score of the assigned label. Furthermore, an image can be tagged with multiple annotations in correspondence to multiple visual content classifications designed for different smart city applications. Consequently, the TVDP data model enables the usability of visual data to various ML models for multiple smart city applications. Examples of visual content classifications include street cleanliness [1], underlying material [24], and road damage [4]. Moreover, an annotation descriptor is a label describing either an entire image or a part of the image. The latter label is associated with a boundary surrounding a visual part of the image.

**Textual Descriptors:** A textual description can help to efficiently search visual data because it may include keywords that provide additional semantics of the visual content or the context of an image (e.g., event description). Therefore, the MediaQ app used in the data acquisition phase enables users to provide manual textual keywords while uploading an image or a video. Subsequently, the TVDP data model includes the textual descriptor of the collected visual data.

**Temporal Descriptor:** This descriptor includes two attributes depicting both capturing and uploading times.

*B. Data Storage*

Efficient storage of visual data is essential, especially when dealing with large-scale datasets. Given the rich and comprehensive data model of visual data, TVDP includes a corresponding database schema (see Fig. 2) which enables storing all properties of the visual data. In TVDP, a video is represented by a sequence of key frames; hence the video is stored as a set of images where each one is tagged with various descriptors[1]. In general, stored visual data are categorized into two types: *original* and *augmented* ones. The augmented images are synthesized using the visual content of an image by applying image processing techniques (e.g., cropping and rotating)[2]. The various properties of the TVDP data are modeled in the database schema as follows:

- Spatial descriptors are modeled using the GPS attributes of the $Images$ entity, and both $Image\_FOV$, $Image\_Scene\_Location$ entities.
- Visual descriptors are modeled in the $Image\_Visual\_Features$ entity.
- Annotation descriptors are modeled in the $Image\_Content\_Annotation$ entity. Each annotation refers to one of the labels defined in the $Image\_Content\_Classification\_Types$ entity that corresponds to an image classification stored in the $Image\_Content\_Classification$ entity.
- Textual descriptors are modeled in the $Image\_Manual\_keywords$ entity.
- Temporal descriptors are modeled using the two attributes ($Timestamp\_Capturing$, $Timestamp\_Uploading$) of the $Images$ entity.

*C. Data Access*

Due to the versatility of the properties associated with the visual data stored in TVDP, the platform is capable of exposing the visual data using various types of queries that can be beneficial for diverse users of different interests. Hence, such queries enable TVDP to achieve its main objective for supporting translational services. In particular, the TVDP platform provides the following types of queries:

- **Spatial Queries**: to search visual data using a referential spatial point or spatial range,
- **Visual Queries**: to search visual data using an example image (e.g., retrieve top-k similar images to the example image or all similar images using a similarity threshold),
- **Categorical Queries**: to search visual data corresponding to specific annotation labels,
- **Textual Queries**: to search visual data using a set of keywords, and
- **Temporal Queries**: to search visual data using temporal filters.

In addition, TVDP supports a set of hybrid queries consisting of a combination of different query types (e.g., spatial-visual query, and spatial-textual query). To enable efficient access using these types of queries, TVDP includes various types of indexes: single-modal indexes (e.g., Oriented R-tree [25] for spatial queries, LSH [26] for visual queries, and inverted indexes [27] for textual queries), and hybrid indexes for hybrid queries (e.g., Visual R*-tree [28] for spatial-visual queries).

[1]MediaQ app enables users to capture spatial metadata at a fine-granular level (i.e., at the level of frames).

[2]An example of an image augmentation library is the Python Augmentor library (http://augmentor.readthedocs.io).

## V. Data Analysis

A vital part of the proposed TVDP platform is to share data and provide interoperability of analysis processes among community partners (i.e., government, industry, a non-profit organization, and academia). The core of any machine-learning-enabled platform is application programming interfaces (APIs). APIs allow one developer to plug into other applications without having to know anything about the complex code at the heart of those applications. The latest trend is to provide commoditized ML as a service to analysts and developers. Thus, there are companies such as Google [29], IBM [30], Amazon [31], and Microsoft [32] helping analysis process by building commercial ML APIs so that organizations without deep programming skills can make the best use of the ML technology. However, they are mainly focusing on business applications, and few provide well-defined APIs for smart city problems. Moreover, such APIs were built and customized for pre-defined problems[3] (e.g., detecting vehicles) and hence their supported models might not be suitable for smart city applications of interest (e.g., detecting road damages and graffiti). In particular, the most time-consuming and important part of using ML APIs are identifying the problem that needs to be solved and building meaningful datasets. Regarding dataset, ML practitioners should gather as much data as they can around a given context, and manage as efficiently as possible to reduce processing time. This is easier to say than to actually perform since it requires lots of repetition of manual data selection processes. TVDP especially focuses on APIs to enhance data management for collaborative learning among users with diverse backgrounds.

TVDP exposes Restful API web services for the main components of the platform. Users can create API keys to use TVDP features. Given that the users of our APIs might not have a strong programming background (e.g., governments and non-profit organizations), the design of the TVDP's APIs is simple. API users without deep programming experience easily have access to APIs through web interfaces, while more programming experienced users can directly access APIs through cross-platform client libraries. Some examples of common APIs are: 1) Add new data: Define new visual data streams to be added in TVDP. This allows our platform to be up-to-date with new data sources including crowdsourced data and makes our ML components more robust. 2) Search datasets: Data can be efficiently searched by metadata such as location. 3) Download datasets: Searched data can be downloaded in their raw form or only metadata in predefined forms. 4) Get visual features: Upload images and retrieve visual features. 5) Use machine learning models: Expose various ML components (existing or newly developed) to generate analysis results. Collaborators can upload raw data (images) or processed data (extracted visual features) and use the trained models to extract results for data-driven predictions or decisions in their applications. 6) Download machine learning models: Edge devices with limited access to Internet and capabilities can benefit by running the ML model locally. Collaborators can download the trained ML models and run them directly on the edge devices. 7) Devise new ML models: Collaborators can build and share their ML models with others through our platform by defining its input and output specifications.

[3]Recently, some ML APIs (e.g., Google AutoML API [33]) have been developed to enable creating an ML model for a customized problem by providing a labeled training dataset. However, the main drawback of such APIs is that the analysis results are not integrated seamlessly in a unified platform such as TVDP.

## VI. Actions

A unified platform has the following scalability challenges in city-level applications: The size of visual data is huge so 1) storage system should be scalable, 2) bandwidth should be sufficient for the transmission of data from mobile cameras, and 3) computing power should be enough to perform multiple ML processes with a huge amount of visual data. Due to the limited computation and storage on end devices, the data streams are typically off-loaded to resource-rich hardware platforms, which lead to the inception of edge computing. Edge computing paradigm recently emerged to offload the computation closer to edge devices in order to minimize bandwidth consumption, processing time and latency. Advances in neural networks along with the processing power of edge devices have made it possible to run machine learning models locally on edge devices, which provides a great potential to utilize lots of mobile edge devices in public domain for smart city applications. TVDP mainly focuses on this.

Most of the existing frameworks for edge computing with machine learning rely on a single pre-trained static model, i.e., a model is trained on the server side and then distributed to edge devices. This approach works well in the case when the model owner is well aware of the processing and communication capabilities of the participating edge devices. However, in many cases, the model owner may not be the edge device owner, which is especially true for crowd-based and incentive-driven smart city applications; thus the processing power of edge devices is unknown. Having a single model for a diverse set of edge devices with different processing capabilities (i.e., communication bandwidth, memory, battery capacity and processing power) introduces new challenges because for example, a high-end device can run a more complex version of the model which potentially can provide more accurate results, or on the other side of the spectrum, a low-end device can run a simpler version of the model much faster but with less accurate results. In addition, retraining the existing model to enhance its quality, requires the model owner to perform the expensive and time-consuming process of collecting new labeled data, without taking advantage of the data generated on the edge devices themselves. Thus, these approaches are very limited in supporting broad and ever-evolving classes of edge devices and lack the mechanism to collect and retrain the existing models with flexibility automatically.

To overcome the aforementioned challenges, we designed and developed a crowd-based learning framework which integrates machine learning, edge computing and crowdsourcing [34]. As shown in Fig. 4, the framework trains models on the server with diverse complexities and dispatches the appropriate model according to the edge device capabilities, improves the machine learning model of interest by utilizing the crowdsourced data collected by edge devices and supports automatic and manual labeling from edge devices users. Moreover, to limit the bandwidth consumption, the framework deploys a distributed selection algorithm that prioritizes the crowdsourced data and transfers a selected subset of data. To reduce network traffic and processing cost on the server, the framework extracts the visual feature vectors of the selected subset locally on the edge device and transmits them to the TVDP server, instead of sending the raw high-quality image. Our experiments show that this approach can efficiently upgrade the learning model.

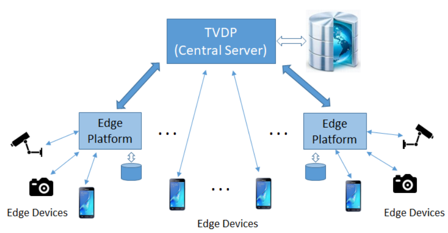


Figure 4: Edge computing to Distribute Platform Load

## VII. EXPERIMENTS WITH USE CASE

To evaluate the effectiveness of the proposed platform, we conducted experiments in a real use case using large-scale datasets. This section describes how a collaborative work among multiple interested parties was achieved through TVDP.

### A. Collaborative Visual Data Analysis and Sharing

We have studied an automatic classification of street scenes to identify their cleanliness level using a big real dataset obtained from the Los Angeles Sanitation (LASAN) Department. We first investigated the use of various image features and classifiers following the cleanliness labels defined by LASAN.

First, working with LASAN, we investigated image classifications based on LASAN's cleanliness levels using a dataset of 22K geo-tagged street images with correct labels (i.e., bulky item, illegal dumping, encampment, overgrown vegetation, and clean, (see Fig. 5)). All images were processed for visual feature extraction. For color feature extraction, images were processed in the HSV color space, and the color histogram was divided into 20, 20, and 10 bins in H, S, and V, respectively. For SIFT-BoW, to generate the dictionary of visual words, SIFT key points were extracted from 80% of the dataset and clustered into 1000 clusters (using kMeans). For CNN, the Caffe architecture was fine-tuned using 80% of the dataset. For the classifiers adopted in our approaches, we used the Python scikit-learn [35] library. All classifiers were trained on 80% of the dataset using 10-fold cross-validation.

After such a shared dataset was prepared as a one-time job, many machine learning algorithms were applied to explore various image features. Individual learning was able to independently select its own dataset on TVDP in a very flexible and fast way (i.e., by multiple people in parallel). Thus, different classifiers could be easily tried to identify the best features and classifier to label an image based on the level of street cleanliness. Fig. 6 summarizes these experiments by showing all different used features and classifiers and the F1 score of each combination of them in the experiments, assuming that we used the entire dataset and region as one single dataset. Our classification results show that the best F1 score was achieved when using the CNN image features due to the rich feature representation provided by CNN. Among the classifiers, SVM achieved the best F1 score with both SIFT-BoW and CNN obtaining scores of 0.64 and 0.83, respectively. Since the best F1 score was achieved with SVM using CNN, we further studied its F1 score among all image categories for street cleanliness. As shown in Fig. 7, SVM with CNN achieved an F1 score higher than 0.8 in all image categories obtaining the highest F1 score with the "Overgrown Vegetation" category and the lowest score with the "Encampment" category.

To evaluate the edge computing component, we generated image-based analysis models for street cleanliness using transfer learning on three pretrained models (i.e., MobileNetV1, MobileNetV2, and InceptionV3). Thereafter, we transferred the analysis models using three types of devices: a common desktop machine, a Raspberry PI 3 B+ (RPI) and a smartphone. Fig. 8 shows the average inference time in milliseconds (in base-10 logarithmic scale) using the three models running on the three edges. RPI has limited resources compared to desktop class devices. In most cases, it requires thousands of milliseconds inference time, and on average is *1.5x* order of magnitude slower compared to desktop class devices. The desktop class devices are capable of processing the visual features in a short amount of time (tens of milliseconds in most cases) for models with various complexities and image sizes. This shows the importance of the TVDP edge computing component that enables smartly dispatching the suitable model, based on the resource capacities of edge devices and their performance capabilities, which not only accounts for the resource consumption but also aims to achieve the best result possible within the available resource budgets.

(a) Bulky Item (b) Illegal Dumping (c) Encampment (d) Overgrown Vegetation (e) Clean

Figure 5: Image Examples for the Street Cleanliness Classes [1]

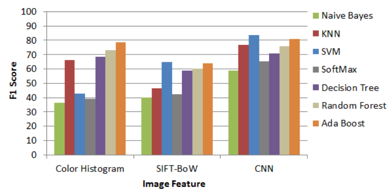


Figure 6: Various Classifiers and Image Features

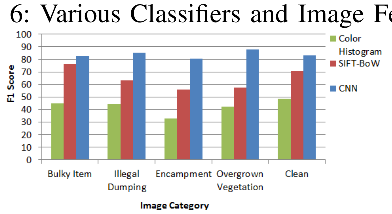


Figure 7: SVM and Various Features w.r.t. Cleanliness Categories

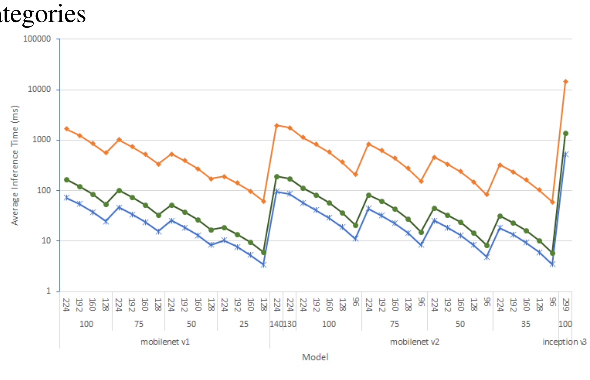


Figure 8: Inference Time vs. Models for Street Cleanliness Dataset on Desktop, Raspberry PI and Smartphones.

### B. Discussion

Once the classification of new unlabeled images is done, the results are annotated as an augmented knowledge of the original images in the database. Then, it can be shared and utilized for other independent analysis on TVDP by any interested parties, realizing the concept of translational data. As an example, the dataset collected by LASAN can be utilized for another city operation. We performed separate learning to identify graffiti using the same dataset and annotated the dataset with the results. In this way, various visual analysis can be performed, and their results are annotated and shared, evolving into a more comprehensive and translational visual information database. Newly learned information can also be translated into other analysis. In our experiments, we first directly utilized the street cleanliness classification of LASAN data for homeless counting since one of the resulting classes is encampment on the streets. This resulting knowledge from one analysis for street cleanliness was translated into useful information for another social study, i.e., homeless counting, without any extra learning processing. Next, we can further closely investigate the images with an encampment and figure out more extra knowledge about homeless people. Also, a comprehensive spatial crowdsourcing campaign can be performed to further collect visual data from specific regions for in-depth studies; to name a few, 1) Clustering of homeless tent locations in Los Angeles and their weekly changes, 2) Identifying the same tents and analyze their spatial movement over time, 3) Extra detailed visual analysis of street cleanliness around tents.

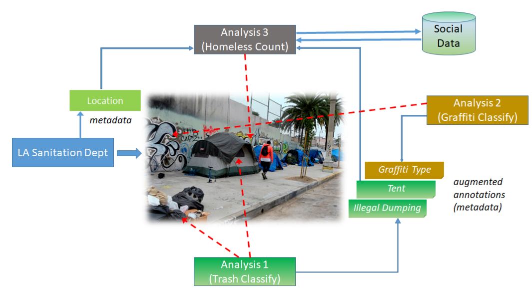


Figure 9: Translational Data Example

## VIII. CONCLUSION AND FUTURE DIRECTIONS

This paper describes a bold idea of translational visual data platform where users can collaborate in data collection, management, analysis, and sharing in an efficient and democratic way. A use case with real dataset demonstrates the feasibility of the proposed core concept even though there still remain lots of work to be done. We believe that the merit of this research lies in the evaluation of the hypothesis that properly designed visual data platform will enable visual data collection, management, and analysis, that are currently quite fragmented, to become structured, well-integrated and organized, facilitating the design of more efficient, sharable, economical and sustainable data-centric, democratic solutions for social problems, taking into account multiple stakeholders in large-scale smart cities. In TVDP, intelligent representation and management of features provide smart connections among fragmented techniques into a unified visual data platform where translational data science becomes practical and easy to scale to different applications.

As a future work, we will focus on making TVDP as a disaster data platform which requires a fast visual data acquisition and a high level of situation awareness, an automatic event detection, and an efficient translation of newly learned information. Especially, due to the advance in drone technology, Drones are widely being used in monitoring disaster situation awareness by governments and non-profit organizations. However, there are few systematic collection, management, and sharing of time-critical areal videos. Thus, we aim to collect and analyze drone videos for a wide area real-time monitoring in disasters (e.g., wildfire) using the proposed TVDP. We will also investigate image-based learning for damage evaluation when we successfully collect enough amount of visual data for a certain disaster.

## Acknowledgment

This work was supported in part by NSF grants IIS-1320149 and CNS-1461963, the USC Integrated Media Systems Center, and unrestricted cash gifts from Google, and Oracle.

## References

[1] A. Alfarrarjeh, S. H. Kim, S. Agrawal, M. Ashok, S. Y. Kim, and C. Shahabi, "Image classification to determine the level of street cleanliness: A case study," in *BigMM*. IEEE, 2018, pp. 1–5.

[2] S. H. Kim, J. Shi, A. Alfarrarjeh, D. Xu, Y. Tan, and C. Shahabi, "Real-time traffic video analysis using intel viewmont coprocessor," in *DNIS*. Springer, 2013, pp. 150–160.

[3] A. Alfarrarjeh, S. Agrawal, S. H. Kim, and C. Shahabi, "Geo-spatial multimedia sentiment analysis in disasters," in *DSAA*. IEEE, 2017, pp. 193–202.

[4] A. Alfarrarjeh, D. Trivedi, S. H. Kim, and C. Shahabi, "A deep learning approach for road damage detection from smartphone images," in *Big Data*. IEEE, 2018, pp. 5184–5187.

[5] A. Alfarrarjeh, D. Trivedi, S. H. Kim, H. Park, C. Huang, and C. Shahabi, "Recognizing material of a covered object: A case study with graffiti," 2019, manuscript submitted for publication.

[6] S. H. Kim, Y. Lu, J. Shi, A. Alfarrarjeh, C. Shahabi, G. Wang, and R. Zimmermann, "Key Frame Selection Algorithms for Automatic Generation of Panoramic Images from Crowd-sourced Geo-tagged Videos," in *W2GIS*. Springer, 2014, pp. 67–84.

[7] G. Wang, Y. Lu, L. Zhang, A. Alfarrarjeh, R. Zimmermann, S. H. Kim, and C. Shahabi, "Active Key Frame Selection for 3d Model Reconstruction from Crowdsourced Geo-tagged Videos," in *ICME*. IEEE, 2014, pp. 1–6.

[8] "MyLA311 App," https://www.lacity.org/myla311.

[9] Q. Contributor, "Why is it important to democratize machine learning?" 2016. [Online]. Available: https://www.forbes.com/sites/quora/2016/12/28/why-is-it-important-to-democratize-machine-learning/#39ca0f17582a

[10] D. Wellers, J. Woods, D. Jendroska, and C. Koch, "Why machine learning and why now?" 2017. [Online]. Available: http://www.digitalistmag.com/executive-research/why-machine-learning-and-why-now

[11] Y. Lu, H. To, A. Alfarrarjeh, S. H. Kim, Y. Yin, R. Zimmermann, and C. Shahabi, "GeoUGV: User-generated mobile video dataset with fine granularity spatial metadata," in *MMSys*. ACM, 2016, pp. 43–48.

[12] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *SIGSPATIAL GIS*. ACM, 2012, pp. 189–198.

[13] A. Alfarrarjeh, T. Emrich, and C. Shahabi, "Scalable Spatial Crowdsourcing: A Study of Distributed Algorithms," in *MDM*. IEEE, 2015, pp. 134–144.

[14] H. To, S. H. Kim, and C. Shahabi, "Effectively crowdsourcing the acquisition and analysis of visual data for disaster response," in *Big Data*. IEEE, 2015, pp. 697–706.

[15] S. Tavakkol, H. To, S. H. Kim, P. Lynett, and C. Shahabi, "An entropy-based framework for efficient post-disaster assessment based on crowdsourced data," in *EM-GIS*. ACM, 2016, p. 13.

[16] S. H. Kim, Y. Lu, G. Constantinou, C. Shahabi, G. Wang, and R. Zimmermann, "MediaQ: mobile multimedia management system," in *MMSys*. ACM, 2014, pp. 224–235.

[17] A. Alfarrarjeh, S. H. Kim, A. Deshmukh, S. Rajan, Y. Lu, and C. Shahabi, "Spatial coverage measurement of geo-tagged visual data: A database approach," in *BigMM*. IEEE, 2018.

[18] M. S. Nixon, *Feature Extraction & Image Processing for Computer Vision*. Academic Press, 2012.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *INT J COMPUT VISION*, vol. 60, no. 2, pp. 91–110, 2004.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[22] S. A. Ay, R. Zimmermann, and S. H. Kim, "Viewable scene modeling for geospatial video search," in *ACM MM*. ACM, 2008, pp. 309–318.

[23] A. Alfarrarjeh, S. H. Kim, S. Rajan, A. Deshmukh, and C. Shahabi, "A data-centric approach for image scene localization," in *Big Data*. IEEE, 2018, pp. 594–603.

[24] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," in *CVPR*, 2015, pp. 3479–3487.

[25] Y. Lu, C. Shahabi, and S. H. Kim, "Efficient Indexing and Retrieval of Large-scale Geo-tagged Video Databases," *GeoInformatica*, vol. 20, no. 4, pp. 829–857, 2016.

[26] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *SoCG*. ACM, 2004, pp. 253–262.

[27] J. Zobel and A. Moffat, "Inverted files for text search engines," *CSUR*, vol. 38, no. 2, p. 6, 2006.

[28] A. Alfarrarjeh, C. Shahabi, and S. H. Kim, "Hybrid indexes for spatial-visual search," in *ACM MM Thematic Workshops*. ACM, 2017, pp. 75–83.

[29] "Google Cloud Vision API," https://cloud.google.com/vision/.

[30] "IBM Visual Recognition API," https://www.ibm.com/watson/services/visual-recognition/.

[31] "Amazon Rekognitionn API," https://aws.amazon.com/rekognition/.

[32] "Microsoft Computer Vision API," https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/.

[33] "Google Cloud AutoML API," https://cloud.google.com/automl/.

[34] G. Constantinou, G. S. Ramachandran, A. Alfarrarjeh, S. H. Kim, B. Krishnamachari, and C. Shahabi, "A crowd-based learning framework using edge computing for smart cities," 2019, manuscript submitted for publication.

[35] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *JMLR*, vol. 12, no. Oct, pp. 2825–2830, 2011.