ICCSPA20 1570619393

Real-time Littering Detection for Smart City using Deep Learning Algorithm

Basma Korchani
InnoV'COM Lab, SUPCOM
University of Carthage, Tunisia

Kaouthar Sethom
InnoV'COM Lab, SUPCOM
University of Carthage, Tunisia
kaouther.sethom@enicar.u-carthage.tn

***Abstract*— Recently, there is a great interest in spreading awareness about the environment protection and the risks of pollution to human health, what made us work hard to fulfill smart and healthy cities. In this article, we will introduce a new platform that would help the ecologic police to catch and identify people who cast down litter in public places.**

Keywords—Smart city, Deep Learning, IoT, Kafka

I. Introduction

Throughout the ages, countries are striving hard to improve their citizen's life. Thus, the development of advanced informative communication technologies enabled the appearance of a new generation in the world of services crowned by the apparition of smart cities technologies.

Smart cities aim, not only to develop mechanisms and applications that facilitates the practical life of people, but also innovating smart platform by stimulating psychological and cultural concern in the life of the citizen. Thereby, smart cities will reach their ambitious goals such as, reducing the incidence of disease, reducing crime rate and eliminating the different forms of pollution…by analyzing human behavior using data mining, artificial intelligence and deep learning tools.

Many efforts tries to combine machine learning with computer vision to simulate high advanced technologies used in IOT systems [1-2] such as criminal detection and recognition systems, using micro-unmanned aerial vehicle capable of real-time detection using video surveillance footage through an ensemble-based machine learning model performing three basic CNN (Convolutional Neural Network) algorithms to suggest the optimum way to detect and recognize garbage in term of recognition performance.

Municipality and ecologic police face many difficulties to punish people who cast down litter in public places, in this article, we will introduce a new paradigm that would help the ecologic police to catch and identify those violators. The goal of this work is to present a cloud-based intelligent littering detection system for use in public area and streets. Our proposal is described along with details of its design, implementation, and operation. Several software solutions, including Kafka clusters, distributed NoSQL database, and rule engine are proposed to provide best littering tracking service experience to users.

II. Background and Related Work

Technology has brought an unprecedented explosion in unstructured data. Sources like mobile devices, websites, social media, scientific apparatus, satellites, IoT devices, and surveillance cameras are generating a vast number of images and videos every second. Managing and efficiently analysing this data is a challenge. Computer vision and digital image processing are currently being widely applied in face recognition, biometric validations, the Internet of Things (IoT), criminal investigation, etc. All these applications use image and real-time video processing so that the live capture of multimedia impressions can be made for detailed analysis and predictions.

Computer Vision using deep learning can be used for :

1) Image Classification

Image Classification is the task for assigning a label to an image. This is useful when there is a single class in the image and is distinctly visible in the image. Canonical example is classification of cat vs dog. Image classification models have come a long way in the last 2–3 years.

2) Object Detection

The difference between classification and detection is that in detection there could be multiple instances of the same class or different classes in the same image. While classification just outputs what the likely class of the image is, detection goes one step further and draws a bounding box on the image.

3) Object Tracking

Once an object is detected it can be tracked across a video. So in some sense object detection can then be extended to object tracking.

When it comes to deep learning-based object detection, there are three primary object detectors:

- R-CNN and their variants, including the original R-CNN, Fast R- CNN, and Faster R-CNN

- Single Shot Detector (SSDs)
- YOLO

R-CNNs are one of the first deep learning-based object detectors and are an example of a two-stage detector. In the first R-CNN publication [3], Girshick et al. proposed an object detector that required an algorithm such as Selective Search (or equivalent) to propose candidate bounding boxes that could contain objects. These regions were then passed into a CNN for classification, ultimately leading to one of the first deep learning-based object detectors.
The problem with the standard R-CNN method was that it was painfully slow and not a complete end-to-end object detector. Girshick et al. published a second paper in 2015, entitled *Fast R- CNN* [4]. The Fast R-CNN algorithm made considerable improvements to the original R-CNN, namely increasing accuracy and reducing the time it took to perform a forward pass; however, the model still relied on an external region proposal algorithm.
While R-CNNs tend to very accurate, the biggest problem with the R-CNN family of networks is their speed — they were incredibly slow, obtaining only 5 FPS on a GPU.
To help increase the speed of deep learning-based object detectors, both Single Shot Detectors (SSDs) and YOLO [4] use a one-stage detector strategy. These algorithms treat object detection as a regression problem, taking a given input image and simultaneously learning bounding box coordinates and corresponding class label probabilities.
In general, single-stage detectors tend to be less accurate than two-stage detectors but are significantly faster. The result is a YOLO model, called YOLO9000, that can predict detections for object classes that don't have labeled detection data. We'll be using YOLOv3 in this paper, in particular, YOLO trained on the COCO dataset.

III. Our Prototype

The growth of cities is directly convoluted with the rising of pollution rates which causes big dangers on human's health. Recently, there is a great interest in spreading awareness about the environment protection and the risks of pollution, to human health what made us work hard to fulfill smart and healthy cities.
We hereafter present a cloud-based intelligent dumping garbage detection framework. It is based on real-world surveillance camera system deployed in a smart city. This framework will help the ecologic police to catch and identify those violators.

A. Real-time data acquisition layer

The first step to realize is the detection and the recognition of littering actions in the street surveillance system. Our program will consume the real-time video stream emitted from multiple clusters of IP cameras installed in different city buildings, and receive as result an alarm if there was a person who throws garbage, the location and the date and the image of the person's face will instantly sent to our MongoDB database; so the system must first of all detect if there is human body then, we compare his acts with the littering and the alarm will be sent automatically to Kafka broker when detecting database setting.

Kafka is known as a publish subscriber system composed of the producers which create the key/value messages and sent it to the Kafka cluster where these messages will be stored into partition within different topics, the distributed feature of Kafka confirm the safety transition of a massive real time stream.

Here are some of the key aspects of why we are using Kafka:

- ✓ Store permanently a huge amount of data that can be persistent, controlled and replicated for fault tolerance.
- ✓ Process continuous flow of data (data streams) in real time across systems.
- ✓ Allow applications to publish data or data streams independently to the application.

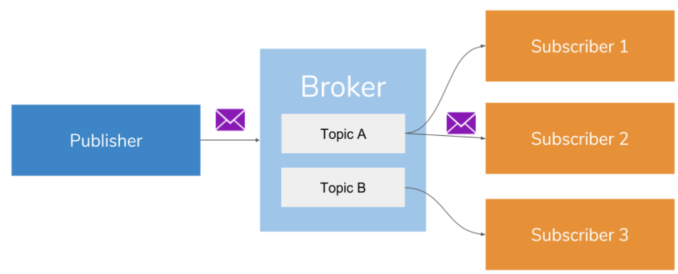

Figure 1: Kafka architecture

We can propose two Kafka topics as in figure 1:

1. Violator identification (photo, personal information …).
2. The dumped garbage location.

The first client (the ecologic police) will subscribe to the first topic and the cleaning agents of the municipalities can subscribe to the second topic.
For the first topic, we need a face recognition procedure; this procedure will use images and information of citizens stored in the database and the cloud.

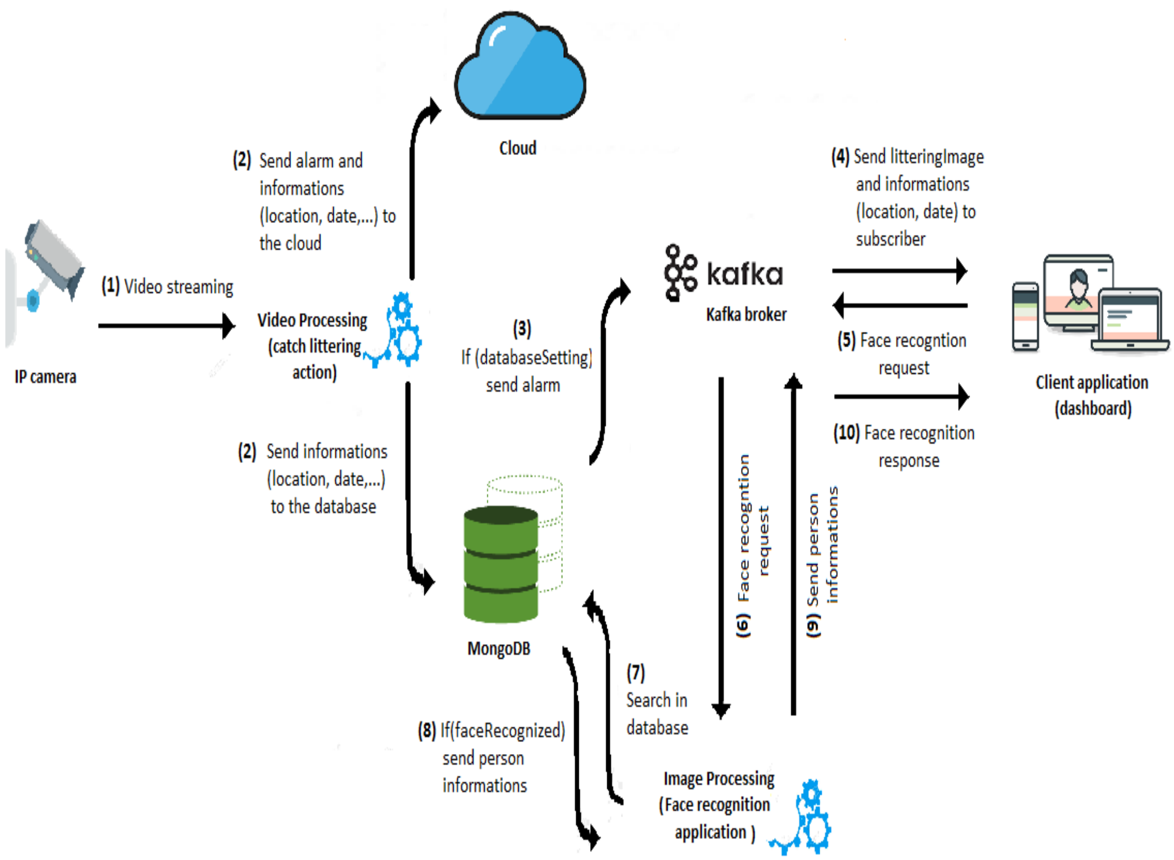

Figure 2: Littering detection framework architecture

B. Data storage and display layer

From a storage point of view, the data in IoMT systems is divided into two main categories; structured data (mostly sent from sensors) and unstructured data (mostly multimedia events) which impose us for using NoSQL database: MongoDB Atlas [5] MongoDB Atlas does an excellent job in infrastructure provisioning and setup. The whole process uses a dynamic web interface that walks you through various deployment options. It is easy, intuitive and doesn't require specialized knowledge.

However, MongoDB documents have a size limit of 16MB and are stored in JSON format. And because in our application, we need to store large files (high-resolution images and videos) we used an additional tool: GridFS. GridFS divides each large file into multiple small chunks and stores each chunk as a separate document in MongoDB. Once the chunk is stored as a document in the database, all operations that can be performed on regular MongoDB documents can be performed on these chunks.

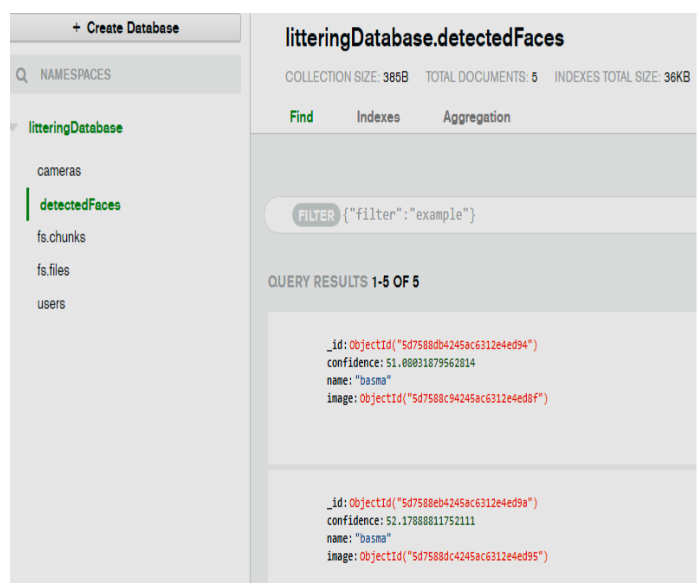

Figure 3: Detected Faces collection in MongoDB Atlas

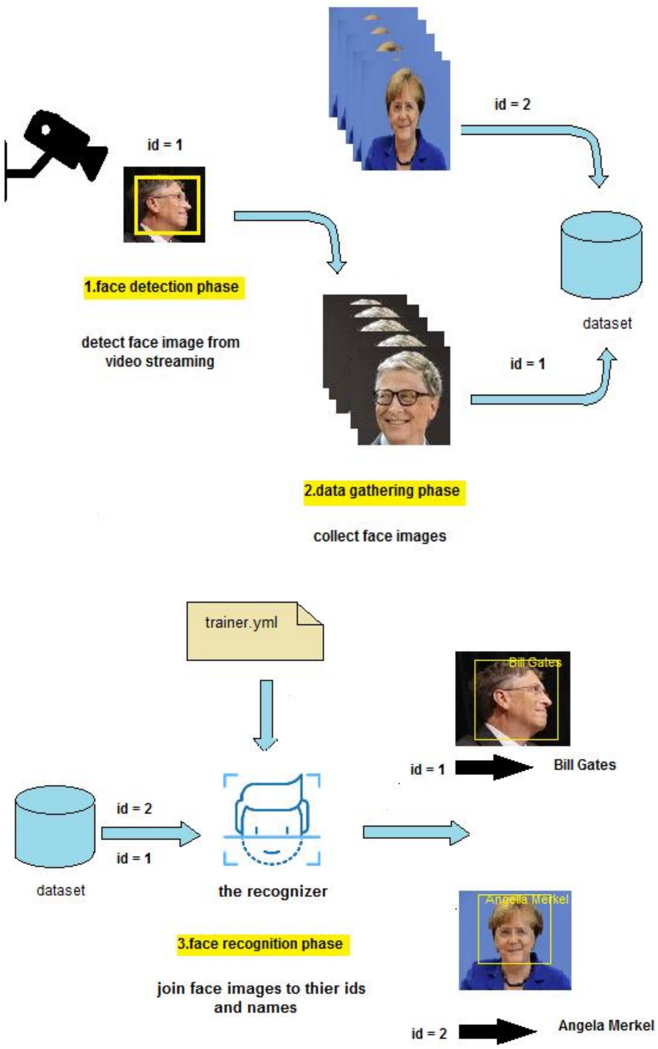

Figure 4: Face recognition procedure

C. The analysis layer

The face recognition procedure base on three main phases:

1) Face detection phase

Initially, the algorithm demands a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we want to extract features from it. In this case, OpenCV contain a trainer as well as a detector.

2) Recognizer training phase

After detecting the face in the video streaming, we will simply create a dataset, where we will store for each id a group of photos in gray with the section that was used for face detecting (Figure 6).

We take 30 capture samples from each id. On this second phase, we have to take all user data stored on the dataset and train the OpenCV Recognizer. OpenCV offer a specific function for this task. We will obtain as a result a .yml file that will be saved on a trainer directory.

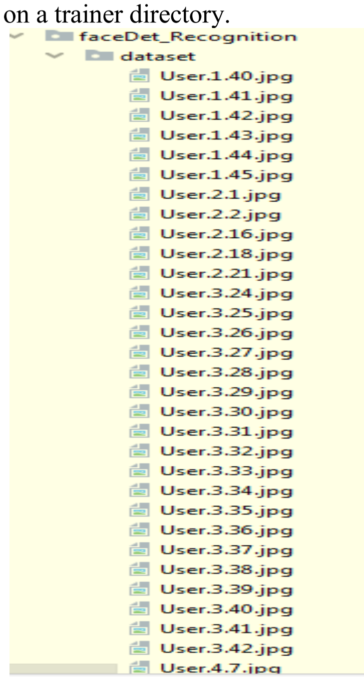

Figure 5: the dataset directory of our project

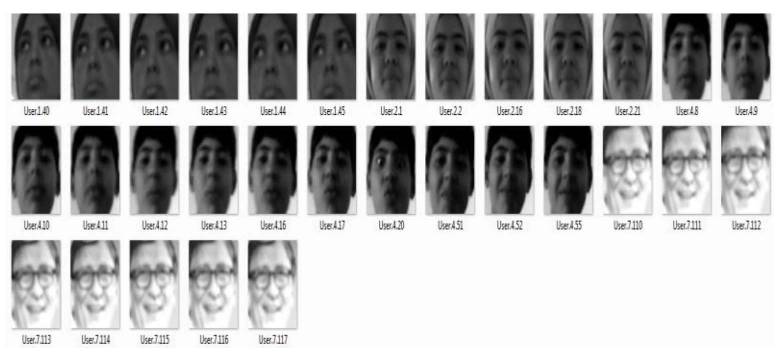

Figure 6: users ids stored on the dataset

3) Face recognition phase

Finally, we reached the face recognition phase of our project. Here, while the face is captured from our camera and if this person had his face captured and trained before, the recognizer will make a prediction giving back its id and a score shown how confident the recognizer is with this match.

The prediction function of the recognizer will take as a parameter a captured section of the face to be analyzed and will return its prospective owner, indicating its id and how much confidence the recognizer is in relevance with this match.

And at last, if the recognizer could predict a face (Figure 7), we put a rectangle border over the image with the expecting id and how much is the match probability in % is correct (mach probability= 100 — confidence index). If not, an unknown label is put on the face.

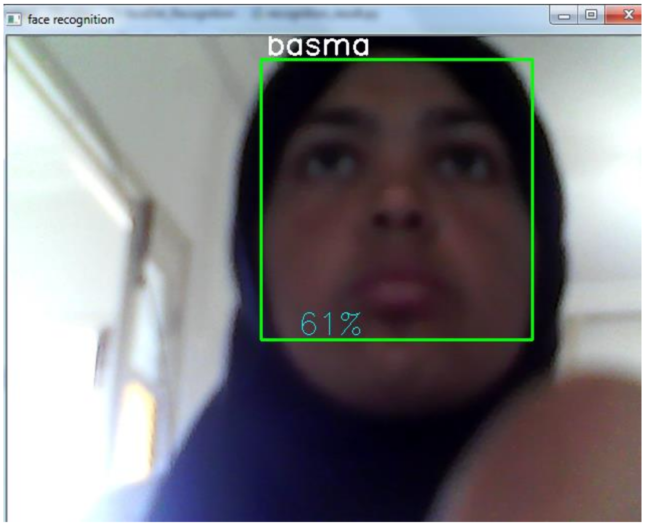

Figure 7: face recognition result

4) *Dumping detection*

Many technologies used for human action recognition; in our project we need a speed algorithm to recognize the action; also we must keep in mind the use of outdoor cameras. Because the dumping actions in the real-world take a variety of forms, we detect the littering action if there is a change in relation between a person and the object being dump by him. The first step in the littering detection procedure is tracking the person with an indeterminate shape, so we used the background subtraction algorithm and estimate the human joint (Figure 8). The noteworthy part in the detection of litter procedure is tracking the garbage and verify if it fell to the ground or it throw in the trash; so we must verify first of all, the presence of a garbage container.

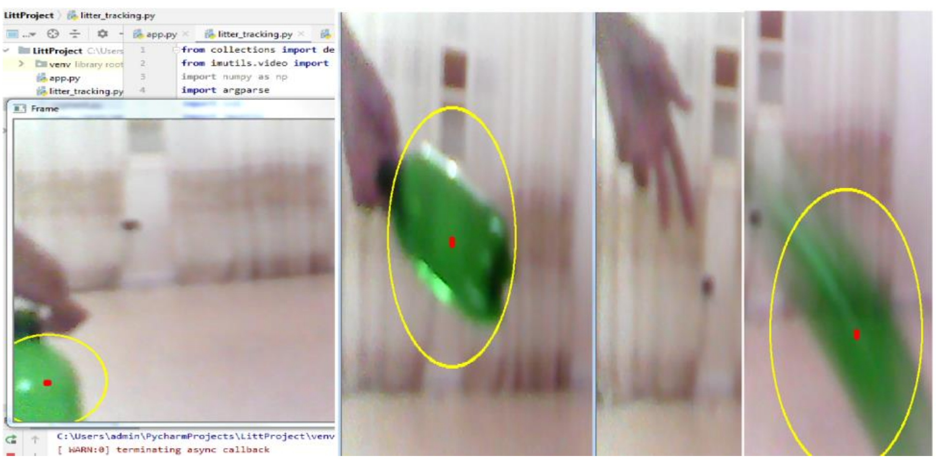

Figure 8: Littering detection capture

IV. Conclusion

We have proposed a framework to help ecologic policy to catch violator, it includes scalable tools to manage and treat a massive multimedia data. It also provides municipalities to automatically pickup garbage and its location. Our framework will give solutions to many ecologic problems. Instead, smart cities have great potential to be designed to tackle social problems and change public behavior.

References

[1] Gregory, T., & Andrew, K. (2017). Stretching “smart”: advancing health and wellbeing through the smart city agenda. 19.

[2] Kimin, Y., Yongjin, K., Sungchan, O., Jinyoung, M., & Jongyoul, P. (2019). Vision-based garbage dumping action detection for real-world. *WILEY ETRI Journal* , 12.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, (2013)

[4] Ross Girshick “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal networks ”, The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448

[5] MongoDB Atlas, https://www.mongodb.com/cloud/atlas