DELL Technologies Envision the Future Competition 2020-21 (Interim Progress Report Structure)

Project Title: Intelligent Surveillance Systems for Smart Cities

The concept of smart cities is emerging rapidly with the objectives of using the technology to enhance the efficiency and quality of services provided to the population in such cities. In recent times, there has been a lot of research done showing that the surveillance systems are of vital importance for such smart cities. This is justified because the surveillance systems can be considered vision organs of the smart city that can provide continuous and uninterrupted urban space monitoring. Thus, the design of the surveillance systems for a smart city must itself be smart. In this context, it is expected that a huge amount of data (Big Data) will be generated in smart cities. Therefore, to ensure the safety of its citizens, it is important to provide an efficient and real-time analysis of these data to get real-time responses, when catastrophic events occur. Accordingly, transmitting this massive data to the cloud, to be processed, is relatively slow. Therefore, the purpose of this project is to implement an edge/fog computing-based surveillance system to offer real-time data processing in smart cities. The primary objectives of this short report are to track the progress in the design and implementation of the project and to highlight the challenges and lessons learned.

**Section 1: Refined Project Description**

* 1. The problem addressed specifically in this project and its importance.

One of the challenges that face surveillance systems in smart cities is Big data, as the smart city involves huge amounts of cameras in 24/7 mode to cover the whole city. Each camera records a massive amount of video and audio data to be processed. To handle such massive data, it is essential to provide a way to get real-time data processing analysis efficiently. Cloud computing has been utilized to perform data processing analysis, leading to slower performance. Therefore, edge/fog computing is essential for real-time data processing analysis with high performance and low latency.

* 1. Project scope and expected outcome

Our main objective is to build a fog computing based framework for surveillance data in order to detect abnormal human behaviors in the streets (fighting, kidnapping, and robbery). The system is also developed to work on enhancing the latency by reducing the amount of data. Moreover, we will work on developing the privacy protection scheme for data security. The expected outcomes of the project can be summarized as follows:

1. Fog/Edge computing-based surveillance system to monitor the whole city 24/7.

2. Privacy assurance by blurring people’s faces to secure their exposed identity in the video.

3. Scalability system to handle such massive data generated from surveillance cameras and sensors.

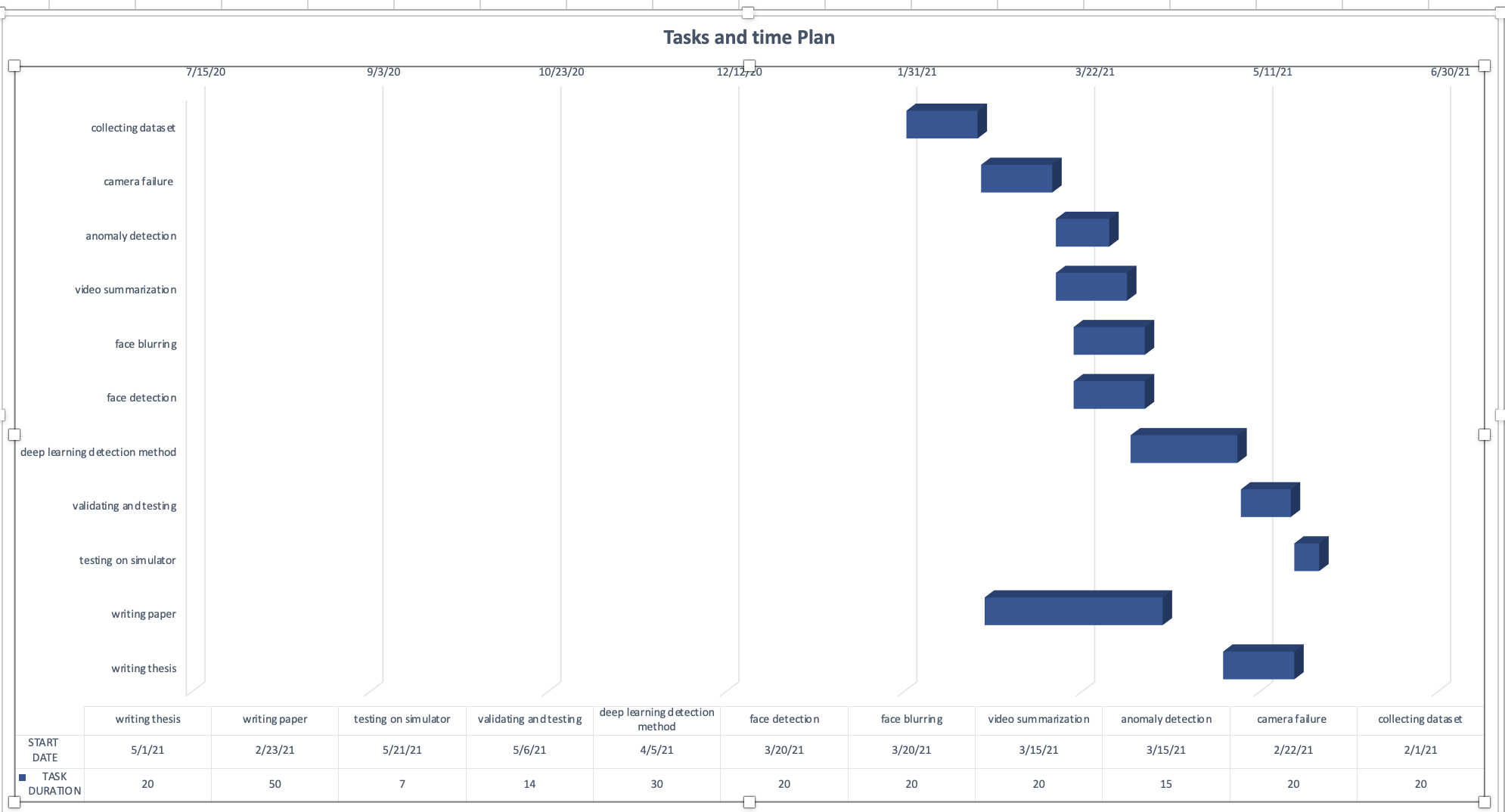
5 Efficient image processing techniques for certain failures in the edge cameras that impact the usefulness of the video streams.

5. Dispatch crime details (i.e., video and location of the event, criminal’s face) and giving alert to the nearest competent authority.

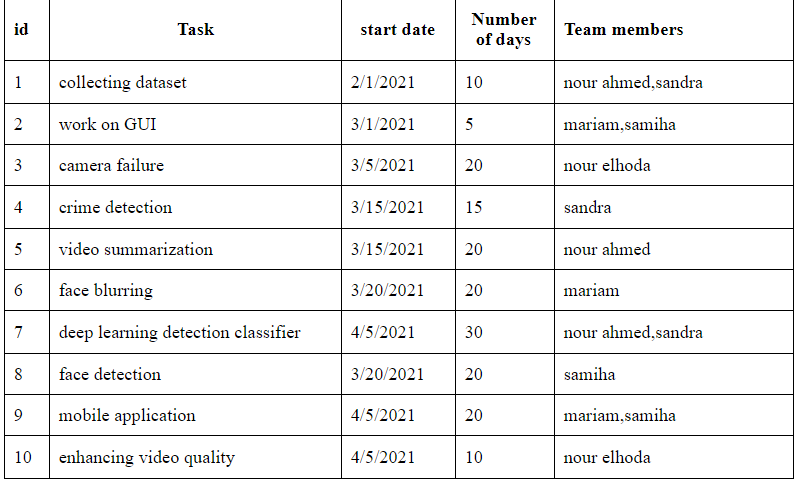
* 1. List and explanation of any changes/adjustments made relative to the abstract report.
* **Crime case specification:** we specified the crime cases that we’re going to train our models on, which are: kidnapping, robbery, and fighting.
* **Changes in the functionalities of the computing layers:** we divided the fog layer into two tiers in order to decrease the load, computation power and time. Moreover, we changed some of the algorithms in the layers with advanced and more specified ones to assist us with our goal.

**Section 2: Refined Project Plan**

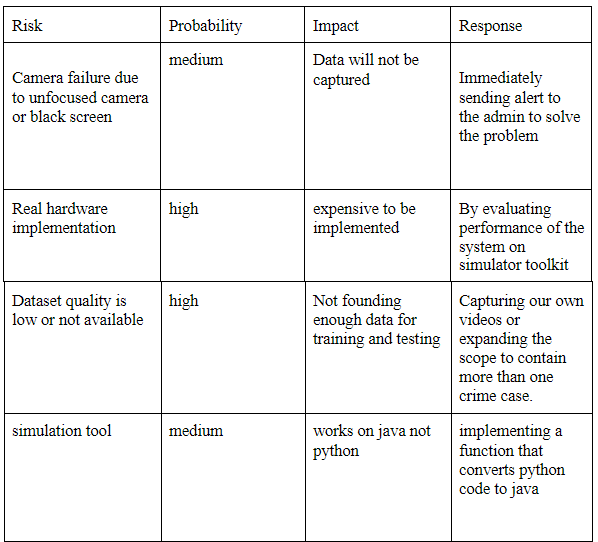
* 1. Detailed schedule and milestones (clearly show current status, progress, and issues)



* 1. Team structure and detailed roles/responsibilities of each member



* 1. Contingency and risk mitigation plan



**Section 3: System Requirements**

The proposed system is crucial, as its fundamental goal is to ensure the citizens’ safety. For our users, the system should provide a rapid response to take appropriate action in time. When an anomaly gets captured on a surveillance camera, the system should detect it and determine if it’s a criminal case. Moreover, the system should track the criminal, identify his face, and trigger an alarm for the nearest police unit.

* 1. Requirements Elicitation Process:
     1. With the expansion of IoT devices in smart cities, there is a massive amount of data generated from the cameras. Transmitting such massive data to the cloud, to be processed, is relatively slow. So, a fog computing-based surveillance system is implemented to offer real-time data processing efficiently. As one of the system's priorities is scalability. Fog computing provides the system low latency and high bandwidth for immediate response.
     2. System stakeholders, users and clients:

-Stakeholders: Nour Ahmed, Mariam Hesham, Sandra Fares, Nour Elhoda Hesham, Samiha Hesham, Eng. Lobna Shahen, Dr. Islam Tharwat.

-Users: Police officers, Admin.

-Clients: Police, Security and Government companies, ELSEWEDY technology company.

* + 1. Challenges encountered and lessons learned

One of the challenges that faced the project during the requirements gathering is to collect or get the kidnapping dataset. In addition to the challenges that face the project during the implementation, it's needed for a massive amount of cameras to cover the whole city. Moreover, adding cameras and sensor devices in public places required permission from several competent authorities to be placed. One of the lessons learned that the system should deal with a multi-tier architecture framework to reduce the computations on each tier for an immediate response and low latency.

* 1. System Requirements List:
     1. Functional requirements:
* The system must be able to detect the accident accurately.
* The system must be able to predict the event before occurring in order to prevent it.
* The system must be able to track the criminal from the location of the incident.
* The system should apply face blurring for normal citizens in order to enhance privacy in surveillance systems.
* The system should send an alert if any failure is detected in the cameras occurred in order to help the maintenance team to solve the problem.
* The system must send an alert message when anomaly behaviour is detected using the fog nodes in the architecture system.
* The system shall provide face recognition features for the criminal.
* The system should filter the incident using video splitting technique in order to be saved and sent to the cloud to save storage.
  + 1. Non-functional requirements
* Scalability: it must be constructed to be a real-time scalable up and down system.
* Performance: fog and edge models guaranteeing good performance for the real-time system.
* Usability: the system ensures usability to achieve the demand efficiently with low latency and high bandwidth.
* Reliability: with large numbers of edge computing devices and edge data centers associated with the network, the system must ensure that no failure or shut down service occurs.
* Security and privacy: the proposed system ensures security and provides data privacy to prevent malicious users from using the transmitting data.

**Section 4: System Design**

* 1. High level system architecture, data flows, etc

As shown in Figure 1, the proposed system is composed of three layers Edge, Fog, and cloud layers; each of them has responsibilities, and they communicate together through a stable network to provide the user at the end the needed result.

• Edge Layer: The first layer in our system is the surveillance cameras fixed within the smart street lights. Those cameras are responsible for capturing the pedestrians throughout the day and using a crime detection algorithm whenever any abnormal actions are recognized the video stream is transferred to our second layer; and if everything is normal, the videos are being stored on the camera’s local storage and then sent to the cloud when the network is free. Also, we consider any failure in cameras using a failure detection algorithm to avoid any blurry recording, green or black videos being recorded. Finally, there is a comparing algorithm that helps tracking wanted criminals whenever detected in any street the authorities are informed.

• Fog Layer: The second layer in our system, this layer is divided into two sub-layers; the first depends on connected vehicles near the incident and the second is smart traffic light which works as gateways. The vehicles layer: It is responsible for extracting more features from the video stream to be surer about the incident and summarize it into an image using a special algorithm for further processing in the second fog sub-layer. The smart traffic light: It is responsible for the final processing of the incident using a deep learning algorithm; to ensure the incident, classify it and send it to the cloud layer, the final layer. Also, an image processing deburring algorithm is applied for face detection. Then, the criminal face image is sent back to the edge nodes.

• Cloud Layer: The final layer is responsible for triggering the alarms and sending notifications for our end users, and backup all the incident data sent. Also, it is used for archiving all the videos captured throughout the day. All the layers should communicate together using a stable wireless network provided in the smart city.

|  |  |
| --- | --- |
|  |  |
| Figure 1. Architectural Design Diagram | Figure 2. Workflow diagram of the framework |

* 1. User Interface

As shown in Figure 3, the proposed system’s application focuses on two main ends. User end and admin end. The application allows the admin to login and creates accounts for the police officer or security guards. Also, the admin can update and delete the user’s account. The user will log in to the application with an email and password set by the admin. Also, the user can update his information and change his password. Moreover, the user will view the alerts sent by the system with images, location, and time. Admin: Admin will log in through the application login page. Admin can add accounts by filling the fields in the add account page with proper information. User: The user will log in through the application login page. He will be able to edit the profile by updating the user’s info through the profile page. Also, the user will view the alerts sent by the system with their details on the alert page.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 3. User Interface | | |

* 1. Algorithmic components

• Sensors placed on street lights are introduced in this system as our edge nodes. The edge nodes are responsible for checking for camera failures. They make sure that the camera capturing is not interrupted quality-wise by any external sources. Moreover, edge nodes apply a face comparison algorithm to compare faces detected in the fog layer with the faces stored in its local storage of previous captures, to trigger an alarm if the criminal showed up again in the incident’s location. Also, a crime detection algorithm using YOLO is applied to acknowledge if a crime occurred. When a crime is detected, the nodes determine the current location of the incident and send it, with the video streams, to the next layer.

• Vehicles are proposed as the system’s nodes for the second layer, the first tier fog layer. The fog nodes in this layer are responsible for applying a video summarization approach which is the key frame extraction technique. The goal of this approach is to decrease the computational power by eliminating repeated frames and choosing frames that have relevant information as key frames. Moreover, a face blurring algorithm is applied to protect the privacy of the citizens.

• Summarized video frames are then passed to the second-tier fog layer, where the nodes are the traffic lights. Deep learning techniques are applied to determine if the contact between people was unproblematic or if an incident occurred. If a crime occurred, the algorithm determines the crime’s type. Moreover, deblurring and face detection get applied to expose the criminal's face.

• The cloud layer, which is the last layer, collects all the data passed to it by the layers and triggers an alarm to police units. Moreover, it backs up the data received by the layers and contains an archive to store video streams from the edge layer.

* 1. Innovative aspects of the design

Main design concerns are focused on robust and efficient communication between each computing layer. We put into consideration the computational power of devices used in each layer to be compatible with the functions deployed to them. Performance of the functions, liable deep learning operation results, and avoiding false-negative detection also were of the main concerns to avoid false alarms. Finally, the system should follow ethics and protect the privacy of the citizens. So, we built our architecture that comprises all three layers to get maximum advantages from each layer; to easily monitor the whole smart city. Each layer performs specific functions either simple image processing or deep learning algorithms, therefore most of the false alarms were reduced; those functions were chosen to be deployed according to each layer’s components’ computational power and to efficiently serve the proper flow of the system. Then each layer communicates with others through a stable network to forward and process the results in case of detecting an incident only; thus the network traffic was reduced. And finally notifying our end users of those incidents efficiently.

**Section 5: System Implementation**

* 1. Hardware and software platforms

5.1.1 Hardware platforms

Even though we aim to evaluate the proposed systems via simulation environment, the targeted hardware can be listed as follows:

* Smart street lights (<https://www.tattile.com/cameras-on-top-of-traffic-light/> )
* Smart Vehicles (<https://www.aptiv.com/en/solutions/smart-vehicle-architecture> )
* Smart traffic light (<https://www.ledtrafficlight.cn/solution_view-4.html> )
* Cloud Environment (IaaS) (<https://www.delltechnologies.com/en-us/learn/cloud/iaas.htm> )

5.1.2 Software platforms

* iFogSim
* Java platform
* Flutter software development kit
* free subscription of Firebase - Google
* Overleaf LaTeX editor , free subscription of Lucid charts and creately
  1. Hardware and software development tools, languages, etc
* Hardware Development: N/A
* Software:
* GitHub
* PyCharm
* Visual Studio code
* Netbeans
* Jupyter
* Languages
* Python
* Java
* Dart
  1. Modules/components acquired from external sources (e.g., open source, licensed commercial/trial products, university/departmental resource libraries, etc):
* YOLOv4 Object Detection Model
* Viola–Jones Object Detection Framework
  1. Innovative aspects of the implementation

Although technologies and solutions enabling connectivity and data delivery are growing rapidly, not enough attention has been given to real-time analytics and decision making as one of the major objectives of smart cities. There are a number of benefits associated with adopting the Edge/Fog computing paradigm over the use-the-cloud approach. The first benefit is the reduction of traffic sent to the backbone network—as uncontrolled increase in network traffic may lead to congestion and result in increased latency. Edge/Fog computing provides a platform for filtering and analysis of the data generated by sensors by using resources of edge devices. This drastically reduces the traffic being sent to the cloud by allowing the placement of filtering operators close to the source of data. Considerable reduction in propagation latency is the next important advantage of using Edge/Fog computing paradigm especially for mission critical applications that require real-time data processing.

**Section 6: Other Relevant Issues and Challenges**

* 1. Technical

Some of the challenges we encountered during our development include the unavailability of surveillance kidnapping datasets, as kidnapping or abduction is one of the use cases we use to test our work. Moreover, most crime datasets tend to have low-quality videos which may be challenging when applying algorithms such as face detection and may affect the accuracy. As we're working with videos, multiple frames need to get processed by deep learning methods, which require defined dependencies, including high-performance GPUs, to be efficient.

* 1. Other

As we are working on multiple crime use cases, it’s challenging to make sure that they become equally trained. By means, our used models need to be unbiased. Moreover, as criminal cases are relatively similar, the accuracy might get affected.