|  |
| --- |
| **Thesis for the Degree of Doctor** |

|  |
| --- |
| **Machine Learning Framework for  In-Disaster People Detection**  **재난 상황의  인명탐색 머신러닝 프레임워크** |

|  |
| --- |
| **December 2018** |

|  |
| --- |
| **Department of Computer Science and Engineering** |
| **Graduate School of Soongsil University** | |

|  |
| --- |
| **Woo Young Moon** |

|  |
| --- |
| **Thesis for the Degree of Doctor** |

|  |
| --- |
| **Machine Learning Framework for  In-Disaster People Detection**  **재난 상황의  인명탐색 머신러닝 프레임워크** |

|  |
| --- |
| **December 2018** |

|  |
| --- |
| **Department of Computer Science and Engineering** |
| **Graduate School of Soongsil University** | |

|  |
| --- |
| **Woo Young Moon** |

|  |
| --- |
| **Thesis for the Degree of Doctor** |

|  |  |
| --- | --- |
| |  | | --- | | **Machine Learning Framework for In-Disaster People Detection**  A thesis supervisor : Soo Dong Kim |   **Thesis submitted in partial fulfillment of the requirements for the Degree of Doctor** |

|  |
| --- |
| **December 2018** |

|  |
| --- |
| **Department of Computer Science and Engineering** |

|  |
| --- |
| **To approve the submitted thesis for the**  **Degree of Doctor by Woo Young Moon** |

**Thesis Committee**

|  |  |  |
| --- | --- | --- |
| **Chair** | 최 형 일 | (signature) |
| **Member** | 이 정 진 | (signature) |
| **Member** | 이 재 유 | (signature) |
| **Member** | 최 광 선 | (signature) |
| **Member** | 김 수 동 | (signature) |

|  |
| --- |
| **December 2018** |

|  |
| --- |
| **Graduate School of Soongsil University** |

**Acknowledgement**

(Times New Roman, 11p, Line interval 2.0)

**Table of Contents**

abstract in English ⅵ

abstract in Korean ⅶ

CHAPTER 1 Introduction 1

1.1 Motivation 1

1.2 Organization 3

CHAPTER 2 Related Works 4

2.1 Works on Disaster Detection 4

2.2 Works on People Detection 5

CHAPTER 3 Functionality of Framework 6

3.1 Functional Group of Capturing and Processing Event 7

3.1.1 Capturing Events 7

3.1.2 Processing Events 8

3.2 Functional Group of Disaster Detection 8

3.2.1 Disaster Profile Manager 9

3.2.2 Disaster Event Processor 9

3.2.3 Disaster Detector 10

3.3 Functional Group of *People Detector* 10

3.4 Functional Group of Image processing 11

3.5 Use Cases 11

*3.6* Mapping between use cases and Derived Component 12

CHAPTER 4 Architecture Design 13

*4.1* Architecture styles to apply 13

*4.2* Architecture for Client Tier 13

*4.3* Architecture for Server Tier 14

CHAPTER 5 Detailed Design 15

*5.1* Design for Disaster Detection 15

5.1.1 Capture Image 16

5.1.2 Initial Proposal 16

5.1.3 Distinction Threshold 17

5.1.4 Profile Embedding 18

5.1.5 Bidirectional LSTM 19

5.1.6 Detecting Disaster 20

*5.2* Design for People Detection 20

5.2.1 Create Initial Proposed Image 21

5.2.2 Crops the images 22

5.2.3 Distinct people on cropped images 22

5.2.4 Refine People Images 23

5.2.5 Detect In-Disaster People 24

CHAPTER 6 Design for Quality Requirements 25

*6.1* Design for High Accuracy 25

*6.2* Design for Time Efficiency 26

CHAPTER 7 PoC Implementation and Experiment 27

*7.1* PoC Implementation of the Framework 27

*7.2* Experiments 28

CHAPTER 8 Assessments 29

CHAPTER 9 Conclusion 30

REFERENCES 31

APPENDICES 33

REFERENCES 31

APPENDICES 33

**List of Tables**

[Table 2-1] (Times New Roman, 11p) 1

[Table 2-2] 5

**List of Figures**

[[Figure 1‑1] Data Quality of Disaster Periods 2](#_Toc529131455)

[[Figure 3‑1] Basic flow of the Detection schemes 6](#_Toc529131456)

[[Figure 3‑2] Concept of the In-Disaster People Detection Framework 7](#_Toc529131457)

[[Figure 3‑3] Use Cases for In-Disaster People Detection Framework 11](#_Toc529131458)

[[Figure 3‑4] Mapping between 16 use cases and 6 components 12](#_Toc529131459)

[[Figure 4‑1] Architecture Styles 13](#_Toc529131460)

[[Figure 4‑2] Architecture for Client Tier 14](#_Toc529131461)

[[Figure 4‑3] Architecture for Server Tier 14](#_Toc529131462)

[[Figure 5‑1] Process for Detecting Disaster 15](#_Toc529131463)

[[Figure 5‑2] Step 1. Process for Capture Image 16](#_Toc529131464)

[[Figure 5‑3] Initial Proposal with Disaster Image Denoiser 17](#_Toc529131465)

[[Figure 5‑4] Algorithm for distinction threshold 18](#_Toc529131466)

[[Figure 5‑5] Represent disaster-specific ‘Symptom’ 18](#_Toc529131467)

[[Figure 5‑6] Bidirectional LSTM for Disaster Profiling 19](#_Toc529131468)

[[Figure 5‑7] Probability of Disaster Detection 20](#_Toc529131469)

[[Figure 5‑8] Process for In-Disaster People Detection 21](#_Toc529131470)

[[Figure 5‑9] Create initial proposed image 21](#_Toc529131471)

[[Figure 5‑10] Generate initial Bounding Box proposal 22](#_Toc529131472)

[[Figure 5‑11] Algorithm of People Distinction 23](#_Toc529131473)

[[Figure 5‑12] Refining people images 24](#_Toc529131474)

[[Figure 5‑12] Detecting In-Disaster people 24](#_Toc529131475)

[[Figure 6‑1] Create initial proposed image 26](#_Toc529131476)

**ABSTRACT**

**Machine Learning Framework for In-Disaster People Detection**

MOON, WOO-YOUNG

Department of Computer Science and Engineering

Graduate School of Soongsil University

In-Disaster situation,

**국문초록**

**재난 상황의 인명탐색 머신러닝 프레임워크**

문우영

컴퓨터학과

숭실대학교 대학원

재난상황에서는

# Introduction

## Motivation

Disaster Occurrence is a severe problem in our society. Disasters creates fatal threats for the safety of life. Hence, effective handling disaster is essential in keeping our life safe. Disaster can effectively be managed by utilizing advanced IT such as Software Framework and Machine Learning Algorithm. Software Framework can cover detection of Disaster and People. And, machine learning can support high accuracy of detection. In disaster situation, the quality of collected camera images and sensor values is considerably low. This is due to heat, fire, flame, smoke, vibration and utility disconnection, etc. Under the environment of low-quality measurement, detecting disaster and people is very hard. After a disaster occurred, it is hard to recognizing the situation because the quality of collected data is poor due to poor environmental quality such as disconnection, unclear data as shown in [Figure 1‑1]. For the people presence detection, the collected data’s quality is very poor. The problems are classified into two categories; *Low quality* problem occurred after disaster occurring periods, and Accuracy problem occurred during all service periods. Hence, there is a strong demand on resolving quality and accuracy problems respectively.



[Figure ‑] Data Quality of Disaster Periods

In-Disaster people detection is mainly concerned with two objectives: The first object is to increase the quality of detection. For achieve this objective, generate a clear high-resolution event and people from a blurry small one. and the second objective is to increase the accuracy for reducing the damage and shortened response time in an emergency and to assist those who have been adversely affected. In this research, a comprehensive framework for people and disaster detection is proposed in terms of from the theoretical aspect to practical realizability aspect.

## Organization

In this section, overall structure of this dissertation is described. Section 2 is the survey of representative works on detection in terms of Disaster and People. Section 3 presents the functionality of the framework of In-Disaster People Detection. Section 4 presents the architecture of the Machine Learning Framework. Section 5 presents detailed design specifications of disaster detection and people detection. Section 6 presents detailed design for qualification requirements. Section 7 presents the environments of implementation on the proposed framework. The proposed detector is implemented with Java and Python. In section 8, an assessment on the proposed detection framework is conducted in terms of machine learning, lessons learned, and contribution of the work. By using learning algorithms presented here, framework with high accuracy and efficiency can be more effectively developed

# Related Works

In this section, representative works are summarized in two folds; works on disaster detection and works on people detection.

## Works on Disaster Detection

Fuentes‘s work distinguishes two generic emergency groups [1]: natural emergencies and man-made emergencies. Natural emergencies are divided into three emergency types: fire, flood and drought. Emergencies caused by humans are divided into two subtypes depending on the scope: emergencies that cause dangers to multiple persons and emergencies that cause dangers to a single person. Schnizler’s work proposed Method for Disaster Detection [2]. Each incident is characterized by a location, the area affected by the incident. a time interval (potentially of length zero) when the event occurred. Each incident is part of a situation, accident or more complex disaster situation Stange’s work proposed Feature for Disaster Detection [3]. Fire consists of Flame height, angle, width, smoke, color of fire. Water has color, ripple pattern of water. Zhu’s work propose algorithm learns to translate an image from one collection to the other [4]. Given two image collections does not require correspondence between images.

The second works proposed detection methods. However, it is not quite clear when and what the disaster could catch using their services. And the third work is only applied to fire and water disaster.

## Works on People Detection

Bai‘s work proposes methods for generating a clear high-resolution face from a blurry small one by adopting a generative adversarial network [4]. Tiny faces are often lacking detailed information and blurring. AlDahoul’s work propose highly abstract and discriminative features, which is produced automatically without the need of expert knowledge [5]. Automatic feature learning methods which combine optical flow and three different deep models. Lots of works also propose effective structures of neural networks for people detection. These works focus on increasing accuracy of people detection at building a model

# Functionality of Framework

In this section, we present specify key features, interface API and Service flow of between the various device and the In-Disaster People detection framework as an effort to find people in disaster situations. The proposed machine learning framework is applicable to online/offline disaster system. The basic detection flow of the detection scheme is shown in [Figure 3‑1]. There are four conceptual requirements in the basic detection schemes. Activities 1) is to gather data to send events. Activities 2) and 3) are to detect the disaster and people based on activities 4).



[Figure ‑] Basic flow of the Detection schemes

For meet these requirement, the People Detection Framework must provide four major key features: Capturing and Processing Event, Detecting Disaster, Detecting People and fix blurry and cropped image. [Figure 3‑2] displays the concept of the In-Disaster people detection framework. The *Device #1~#n* is a various device which has an agent for capturing and sending the events. And the *In-Disaster People Detection Framework* could transform the event with mediators, detect the symptoms of the disaster, and find the people in emergencies.



[Figure ‑] Concept of the In-Disaster People Detection Framework

## Functional Group of Capturing and Processing Event

### Capturing Events

It provides the ability to capture the event from various devices without restarting or any interruption by operator. It is capable of flexible adaptation to a variety of devices. And this feature should support the multiple transactions capturing complex pattern or ambiguous information. For detecting the transactions correctly and timely, data must be reliable and accurate

### Processing Events

It provides the functions corresponding based on the results received from the capturing events. For analyzing the events efficiently and correctly, it is required to use current disaster profile model and the latest disaster information. And this feature should support the multiple transactions capturing complex pattern or ambiguous information

The disaster profile data is formatted data which are in the form of patterns, symptoms and additional information. If any disaster symptoms are found corresponding to the disaster profile, depending on the severity (critical, suspicious, careful, etc.) are otherwise applicable. Process event uses background server-based processes for processing event and formatting disaster profile. Capture an event placed in different location on the network. Capture an event and information for any changes of contents. Convert formatted message

## Functional Group of Disaster Detection

It provides the ability to detect the Disaster early. For detecting the Disaster early and correctly, the *Framework* can work with smaller sample sets for the learning algorithms of classification and prediction. The *Framework* should catch symptoms that are described in the disaster profile model or learned from previous disaster data in the blurry images or complex pattern from various devices.

### Disaster Profile Manager

During disaster detection processing, the disaster detector must get the symptoms accurately from tiny blurry image or lack due to a disaster situation. To increase the accuracy, the detector uses learning algorithm and disaster profile, which is describes symptoms of target disaster. *The Disaster Profile Manager* provides the feature that upload disaster profile and mange disaster profile.

1. Manage Profile
2. Create Extra Profile Fields (Heading, Input, Hidden Input, Selects, Time zone etc.)
3. Upload Profile for finding disaster
4. Search and Update Profile

### Disaster Event Processor

The Event Processor provides the function to manage agents and transform information which was received from agents. The event processor formatting the several types of data from event Agents. The transformation format is changed by the pattern and association of the disaster.

1. Manage Agent
2. Deploy Agent to Target servers
3. Manage Event Pattern & Hierarchies
4. Event Aggregation and transformation

### Disaster Detector

It provides the ability to detecting of disaster. The collected information was transformed to decide the symptoms of the disaster through convert low-resolution to Super-Resolution, denoise and deblur image. It provides ability the Identify disasters by classifying and analyzing the types of disasters from various texts and images.

1. Manage Disaster Model
2. Detect Disaster (Identification of Events, Evaluation of Events)
3. Predict Disaster

## Functional Group of *People Detector*

It provides ability judged a person in a disaster from the collected information. After the conversion to clear and high resolution, the information should not be confused. Due to disaster's disconnection of the networks, converted sources will be used previously gathered data. an event placed in different location on the network. Capture an event and information for any changes of contents. Convert formatted message

1. Manage People Detection Model
2. Detect People

## Functional Group of Image processing

It provides the ability convert image process. First and foremost, we really need to understand what type of disaster data we are dealing with and what eventually we want to get out of it. The Image may not be clear and missing the key information due to disaster. So, this feature should consider characteristics of the low-resolution and blurry. It converts low-resolution to Super-Resolution, denoise and deblur image.

## Use Cases

The framework is basically considered to avoid the problem of deblurring and low-resolution. As shown in [Figure 3‑3], the framework is developed based on 16 use cases for detecting disaster and in-disaster people.



[Figure ‑] Use Cases for In-Disaster People Detection Framework

## Mapping between use cases and Derived Component

The Use Case Map aims to link behavior and structure in an explicit and visual way. It describes casual relationships between use cases and derived components. These paths represents scenarios that intend to bridge the gap between requirements (use cases) and detailed design.



[Figure ‑] Mapping between 16 use cases and 6 components

# Architecture Design

In this section, we present the architecture and design-level specification forimplementing detection framework

## Architecture styles to apply

We will use Client-Server Architecture Style and Layered Architecture Style. On the layered architecture style, there are four layers: control Layer, Interface Layer, Service Layer, Model Layer



[Figure ‑] Architecture Styles

## Architecture for Client Tier

To gather an event and send information for detecting disaster and People. The Client Tier consists of the following components. Input Adapter, Policy Downloader, Data Converters, Event Modeler, Event Processor, Output Adapter



[Figure ‑] Architecture for Client Tier

## Architecture for Server Tier

It represents the architecture for server tier. To training disaster and people detection, it can choose algorithm based on whether we are trying to do prediction, classification of Disaster and In-Disaster People Detection. We can also choose algorithm between RNN and CNN with LSTM. In addition, it tunes framework’s algorithm based on the gathered data we already have. After the results, it provides the archive area to People and Disaster detection



[Figure ‑] Architecture for Server Tier

# Detailed Design

In this section, disaster detection and people detection process is presented to develop each component for In-Disaster People detection framework developers.

## Design for Disaster Detection

This disaster detector is to initial proposal and determine the disaster. As shown in [Figure 5‑1], this is to Detecting disaster based on Generator Adversarial Network with Disaster Profile.



[Figure ‑] Process for Detecting Disaster

To detect disaster, our process consists of 6 steps:

1. Step ① Capture Image Data for Detecting Disaster
2. Step ② Initial Proposal (with Super Resolution, deblurring & denoising)
3. Step ③ Distinction threshold
4. Step ④ Profile Embedding
5. Step ⑤ Bidirectional LSTM
6. Step ⑥ Detecting Disaster

### Capture Image

This ***Step 1*** is Gather Image Data from agents for Disaster Detection.

1. Sampling Image data from multiple agents
2. Preprocessing the event data: transforming raw data into a pre-defined format.
3. Data Purification: detecting and correcting (or removing) corrupt or inaccurate record



[Figure ‑] Step 1. Process for Capture Image

### Initial Proposal

As shown in [Figure 5‑4], this ***Step 2*** Initial proposal with Generative Adversarial Network [7] is initial proposal for suggest disaster type.

1. Given unpaired data, noise blocks extracted from images are exploited to train using GAN(Generative Adversarial Network) [8]
2. Both extracted and generated noise blocks are combined with clean images to obtain paired training data which is used to train a deep Convolutional Neural Network (CNN) for denoising the input noisy images.
3. Suggest Disaster Type and provide Statistical Features from denoising image



[Figure ‑] Initial Proposal with Disaster Image Denoiser

As shown in [Figure 5‑4] the images are fed into the network. Disaster Image denoiser is our baseline and create initial proposed data. It extracts noise block extraction. Using the GAN and CNN, it creates disaster type, denoising image, statistical features.

### Distinction Threshold

This ***Step 3 is*** compute Similarity and distinction threshold. If Iteration is over 2 Times, throw away the captured image and go to ***Step 1***. At the first time, an operator will set the threshold by manually



[Figure ‑] Algorithm for distinction threshold

### Profile Embedding

This ***Step 4 is*** the disaster profile embedding. The Disaster Type and Symptoms are initialized using Glove Vector [9], which we fine-tune with our metadata. For the metadata we use a disaster embedding to represent the textual information in a continuous space and feed it to a bidirectional LSTM [10]. It uses ratios of co-occurrence probabilities, rather than the co-occurrence probabilities themselves.



[Figure ‑] Represent disaster-specific ‘Symptom’

As shown in [Figure 5‑5], our paper focus on the disaster of fire, flood, building demolishing. Each disaster has specific symptoms such as temperature change, color change, object shifting, noise, water level smog, etc. The disaster profile describes the common and specific symptoms of each disaster type. The profile is a tuple of three elements: Disaster type, Set of symptoms, annotation. A symptom is defined as a tuple of three elements; Symptom attribute, value range, constraints.

### Bidirectional LSTM

This ***Step 5 is uses*** bidirectional LSTM [10] for disaster text recognition on the profile and event. We model the sequence in both directions (forward and backward) LSTM(Long Short term Memory). As shown in [Figure 5‑6], it process the data in both directions with two separate hidden layers, which are then feed forwards to the same output layer.



[Figure ‑] Bidirectional LSTM for Disaster Profiling

### Detecting Disaster

This ***Step 6*** shows that the process of concatenation the image and text features followed by a fully connected layer and Softmax function. This step is to give a final probability and Image of the sample containing relevant information about a disaster



[Figure ‑] Probability of Disaster Detection

## Design for People Detection

In this chapter, we present the process and design for In-disaster people detection. To detect In-Disaster people based on Generator Adversarial Network, our process consists of 5 steps in [Figure 5‑8]

1. Step ① Initial Proposed Image
2. Step ② Initial Proposal (with Super Resolution, deblurring & denoising)
3. Step ③ Distinct People on cropped image
4. Step ④ Refine People Image
5. Step ⑤ Detect People



[Figure ‑] Process for In-Disaster People Detection

### Create Initial Proposed Image

This ***Step 1*** is Gather Image Data from denoised image. This step is consisting of three activities. As shown in [Figure 5‑9], the first activity is to get an image data from denoised imaged. The next activity is data purification for detecting and correcting (or removing) corrupt or inaccurate record. If denoised image is not so good for use as a people detection process, recapturing original image for later step.



[Figure ‑] Create initial proposed image

### Crops the images

This ***Step 2*** is crops the images based on multi-branch fully convolutional network [11]. The images are fed into the *FCN-People detector*. *FCN-People detector* is our initial bounding box proposal. *The detector* is trained to reconstruct a clear super-resolution image from the low-resolution input images, which includes the down-sample sub-network and the up-sample sub-network. After layers of convolution, a series of convolution feature maps at several scales are generated. And the People Detector is to distinguish the people image and non-people images and then, generate the initial bounding box proposal.



[Figure ‑] Generate initial Bounding Box proposal

### Distinct people on cropped images

This ***Step 3 is*** distinct people on cropped images. As shown in [Figure 5‑11], at start time, the initial bounding box proposal is the feature representation of its corresponding layer. A finite sequence of bounding box transformation actions is then executed based on a recurrent neural network which is trained by reinforcement learning.



[Figure ‑] Algorithm of People Distinction

### Refine People Images

This ***Step 4*** is refining people image model. Our generator network for refinement includes two components as shown in [Figure 5‑12]. One is Up-sample sub-network, which is takes the low-resolution images as the inputs and the outputs are the super-resolution images [12]. The generator network is trained to reconstruct a clear super resolution image (4 x upscaling) from the low-resolution input image. The other is accuracy sub-network, which is to refine the clear super-resolution images from the blurring images.



[Figure ‑] Refining people images

### Detect In-Disaster People

This ***Step 5*** is detecting In-Disaster People in an image. As shown in [Figure 5‑13], a small region of the image is extracted (called detection window) from refinement images. The people detector uses the google protocol buffer framework to determine if the detection windows contains a people. If the probability is bigger than 1 or analyze shape has the feature of people, the people detector detects people.



[Figure ‑] Detecting In-Disaster people

# Design for Quality Requirements

In this section, we design the quality of *Accuracy* and *Efficiency* on In-Disaster People detection framework. As previous chapter described, it is hard to determine disaster just blur or single noise image. Our approach is based on probabilistic framework and these tactics.

1. To convert blur or noise image we apply the GAN, using the pre-trained similarity and last decoding model of the network.
2. The Disaster Profile contains image and/or metadata, we use a word embedding to represent the textual information in a continuous space and feed it to a bidirectional LSTM.
3. We concatenate the image and text features followed by a fully connected layer to give a final probability of the Disaster images

## Design for High Accuracy

High Accuracy of detection is on about ratio of the number of disaster symptoms compared to all disaster symptoms within a specific period. Based on the captured symptoms, detection services recognize disaster’s situations and the provide the appropriate disaster’s detection to related persons. Missing symptoms, that is, can cause malfunction of Detection services. Therefore, accuracy is a key quality factor of Disaster Detection. For improving high accuracy, convert blur or noise image to clear and super resolution image. In addition, it should use not only image but also disaster profile textual context.

1. Up/Down-Sample resolution: Use the super-resolution network to generate clear and fine people with high resolution (4× up-scaling)
2. Refinement Network: promote the generator network to reconstruct sharper images and provide the high-resolution images to the people detector.

## Design for Time Efficiency

Time Efficiency of detection is timeliness in disaster analytics. As shown in [Figure 6‑1], the efficiency is minimizing the total time Tsum.



[Figure ‑] Create initial proposed image

# PoC Implementation and Experiment

In this section, we implement disaster and people detection as a proof-of-concept (POC) and perform experiments on a specific domain to show effectiveness and performance overheads of the framework

## PoC Implementation of the Framework

To evaluate accuracy and efficiency, we describe the mechanism our framework provides for assess the effectiveness of techniques for selecting the most accurate detection algorithms in the process described in chapter 4. We have implemented a proof-of-concept (PoC) version of the limited framework, and the Framework provides experiment. The implementation environment is consisting of open source software, machine learning detection framework such as Java and Python library, which is capable of high accuracy and used it for providing fast adaptation to emerging. [Table 1] shows the key environments of each item.

[Table ] PoC Environments

|  |  |  |
| --- | --- | --- |
| **Category** | **Item** | **Description** |
| **Language** | Python | Development Language and algorithm Library |
| **Database** | SQLite3, Hive | Event Transaction Store and learning Result store |
| **Data Processor** | Apache Spark | Large scale data processing |
| **Web Server** | Tornado | Web Socket server |

To experiment for detecting of disaster and people framework, we develop four key components; Image conventer component, Detection Model Building component, Event learning component, detection model component.

## Experiments

Experiment in terms of efficiency and accuracy improvement in people and Disaster detection. To compare accuracy of detecting People and Disaster experiments disaster detection and people detection. To compare efficiency of detection, Compare Average/Max Learning Time between Proposed Services and well-known algorithm for Detecting

# Assessments

Not all proposed function can be implementable in practice

|  |  |  |  |
| --- | --- | --- | --- |
| **Cases** | **Avg. Learning Time (msec)** | **Result of DisasterDetection** | **Result of People Detection** |
| **Without**  **Framework** | SVM |  |  |
| RFR |  |  |
| Bayesian |  |  |
| Decision Tree |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **With**  **Framework** | SVM |  |  |
| RFR |  |  |
| Bayesian |  |  |
| Decision Tree |  |  |
| Proposed Disaster Detection Service |  |  |
| Proposed People Detection Service |  |  |

# Conclusion

Developing

REFERENCES

1. Laura Lopez-Fuentes et al., Review on Computer Vision Techniques in Emergency Situation, *2017 Multimedia Tools and Applications* (pp. 1-39)
2. Francois Schnizler et al., Heterogeneous Stream Processing for Disaster Detection and Alarming, *2014 IEEE International Conference on Big Data* (pp. 914-923)
3. Hendrik Stange et al., Insight-driven Crisis Information Preparing for the Unexpected using Big Data, *In 2015 ISCRAM(Information Systems for Crisis Response and Management)*
4. Yancheng Bai et al., Finding Tiny Faces in the Wild with Generative Adversarial Network, *2018 Conference on Computer Vision and Pattern Recognition,* IEEE
5. Nouar AlDahoul et al., Real-Time Human Detection for Aerial Captured Video Sequences via Deep Models, *2018 Computational Intelligence and Neuroscience*
6. Zhu, Jun-Yan, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint* (2017)
7. Chen, Jingwen, et al. "Image Blind Denoising With Generative Adversarial Network Based Noise Modeling." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
8. Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.
9. Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
10. Yu, Zhou, et al. "Using bidirectional LSTM recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech." Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on. IEEE, 2015.
11. Bai, Yancheng, and Bernard Ghanem. "Multi-branch fully convolutional network for face detection." arXiv preprint arXiv:1707.06330 (2017)
12. Ledig, Christian, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network." In CVPR, vol. 2, no. 3, p. 4. 2017.

APPENDICES **(Times New Roman, 16p, Bold)**