Internal Development: AI CCTV Rainfall Camera

Handover

June 2019

Li Zhi

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Client** | | |  | | | | | |
| **Title** | | | Internal Development  Handover  June 2019 | | | | | |
| **Abstract** | | | | | | | | |
| This document describes the development of internal project “AI CCTV rainfall camera”. | | | | | | | | |
| **References** | | |  | | | | | |
| Ver. | Author |  | Date | Remarks | Reviewed |  | Approved |  |
| 1 | Li Zhi |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Project number | | - | | | | | | |
| Keywords | | AI CCTV rainfall camera | | | | | | |
| Number of pages | | 7 | | | | | | |
| Classification | | - | | | | | | |
| Status | | Final | | | | | | |
|  | | | | | | | | |

Table of Contents

[1. Introduction 5](#_Toc11850698)

[2. Methodology 5](#_Toc11850699)

[2.1 Classification 5](#_Toc11850700)

[2.2 Background subtraction and Rainfall intensity estimation 6](#_Toc11850701)

[2.2.1 Normal rainfall subtraction 6](#_Toc11850702)

[2.2.2 Heavy rainfall condition 7](#_Toc11850703)

[2.2.3 Night condition 8](#_Toc11850704)

[3. Data Validation 8](#_Toc11850705)

[3.1 2018-4-1 9](#_Toc11850706)

[3.2 2018-12-11 10](#_Toc11850707)

[3.3 2018-12-12 11](#_Toc11850708)

[4. Real-time Deployment 11](#_Toc11850709)

# Introduction

“CCTV rAInfall camera” is under the internal development project which seeks to estimate rainfall intensity by captured frames by surveillance cameras. By leveraging the multi-sourced cameras all over the island, it is possible to cover the street-level rainfall intensity along with cloud-level captured by radars. To develop such a system, it is of great importance to divide the main target into small stages, described in the following paragraphs. In the methodology, It provides different approaches to solve for the objective. For data validation, it compares camera estimated rainfall with other sources e.g. radar, rain gauge.

# Methodology

In this chapter, we will describe some methodologies in estimating rainfall under different circumstances, normal condition where the rain streaks are observable, heavy rainfall condition, night condition etc. The figure below illustrates the overview of workflow defined in this project. We will express each approach with each method we proposed.

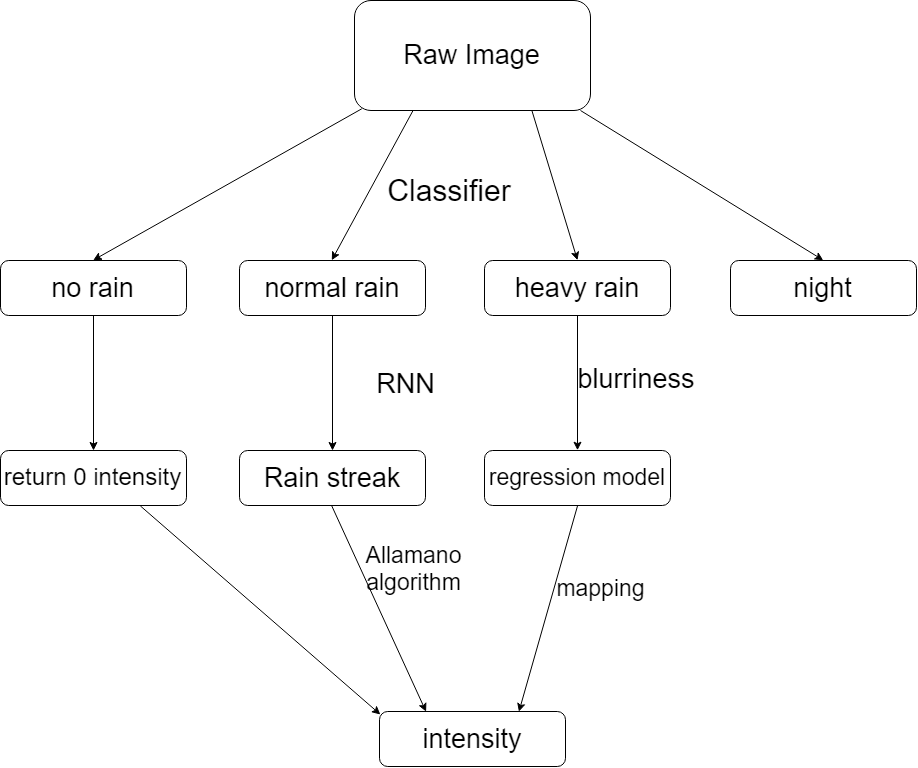


Figure Schematic overview of the workflow

## 2.1 Classification

The classification before injecting into each category is efficient because firstly, it wastes computational time and energy if it is rain-free image. Besides, we found it difficult to utilize one general approach to meet the requirement. In some conditions, the rain streak is not observable which means we have to target the problem within different context.

To make a robust algorithm which can be applied when the background is changing, we consider some generic machine learning algorithms only, e.g. PCA, SVM. These decomposition methods process the structured data and produce the information that is needed to make judgements. It is thus important to dig out some generic features during rainfall events. Here we explored the contrast, saturation, hue, sharpness, brightness, 5 information in total. For each image, we use small 10 by 10 patches to divide one frame. So for one 400 by 400 image, we finally get 1600 values for each information. Eventually, we produce a matrix with (5,1600) to feed into SVM, and output the label which is one of four categories “normal rain”, “heavy rain”, “no rain” and “night”.

## 2.2 Background subtraction and Rainfall intensity estimation

There are lots of algorithms to separate out background and foreground, which in general are categorised by video-based and image-based. Video-based rainfall removal which is the complementary problem in our case mainly uses the numerical methods to solve for optimization methods. It leverages the temporal feature in the video sequence. But they are computationally expensive as the video is injected, the optimization algorithm needs to run again. While image-based rain removal normally uses deep-learning methods e.g. CNN, RNN, GAN by learning the paired rainy images and rain-free images. It loses temporal information but in a way faster than video-based methods in operation.

Considering our product will be deployed into real-time running in cloud, we need an algorithm that can finish producing rainfall intensity within several seconds per frame. Thus, we decided to choose image-based method. However, we find out that both video-based and image-based algorithm cannot be well performed under heavy rainfall and night scenario because rainfall streak cannot be detected and the foreground is blurred by the atmospheric light. We have to seek for other methods to solve this.

### 2.2.1 Normal rainfall subtraction

As determined to use image-based method to subtract the background, we finally found a network (PReNet: Progressive Image Deraining Network) behaves in satisfaction. The figure below depicts the overview of this framework. It subtracts rain streak by multiple stages after considering the different rainfall conditions. Hence, a RNN based framework is developed to recurrently clean the image.

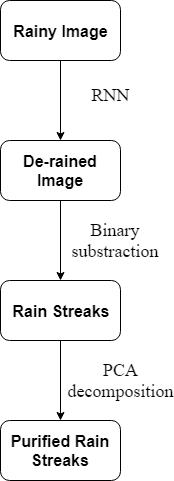


Figure Overview of the pipeline under normal condition



Figure Schematic overview of PReNet framework

After getting the cleaned image, we further get the rain streak image back by subtracting the original image and cleaned image.

The rain streak obtained by the binary subtraction sometimes is noisy with other objects which share the same pattern as rain streak. In order to purify this noisy rain streak, we add PCA decomposition to clarify it. The general approach is to project the rain streak into the rainfall orientation by PCA, and calculate its eigenvalues which stand for its width and length. Afterwards, by some common sense, such as the width of rain streak should not be too large, and the orientation should be within a range.

Thereafter, we can calculate the purified rain streak image with the help of Allamano algorithm which considers a control volume approach to calculate rainfall intensity given the camera parameters. The following graph shows the marked rain streak with streak length and velocity for each streak.

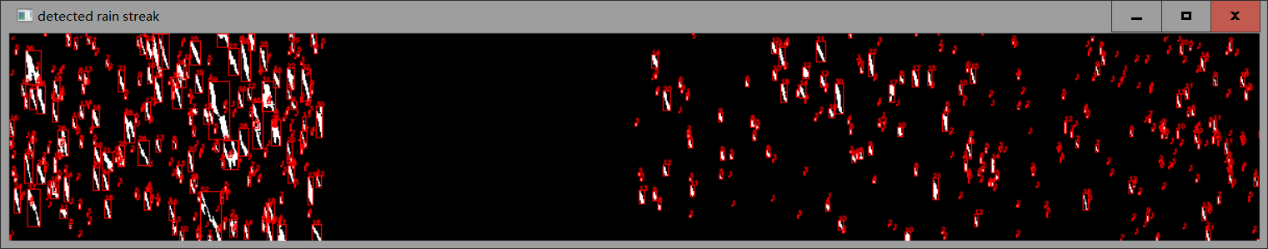


Figure an example of marked rainfall streak: red rectangule box identifies the rain streak

### 2.2.2 Heavy rainfall condition

Under heavy rainfall such as Figure 5, the image will be blurred out, thus any rain streak detection methods will not be valid anymore. This phenomenon is quite common in some tropical cities like Singapore where intensive rainfall could reach 100 mm/h or above. Alternatively, a blurriness mapping method should be developed to get the rainfall intensity. For the sake of data scarcity, we are not able to construct the regression model even though gained the idea. Further analysis will be followed once more data are gained. The main idea is to build a relationship between blurriness index and rain intensity.



Figure An example showing the image under heavy rain events.

### 2.2.3 Night condition

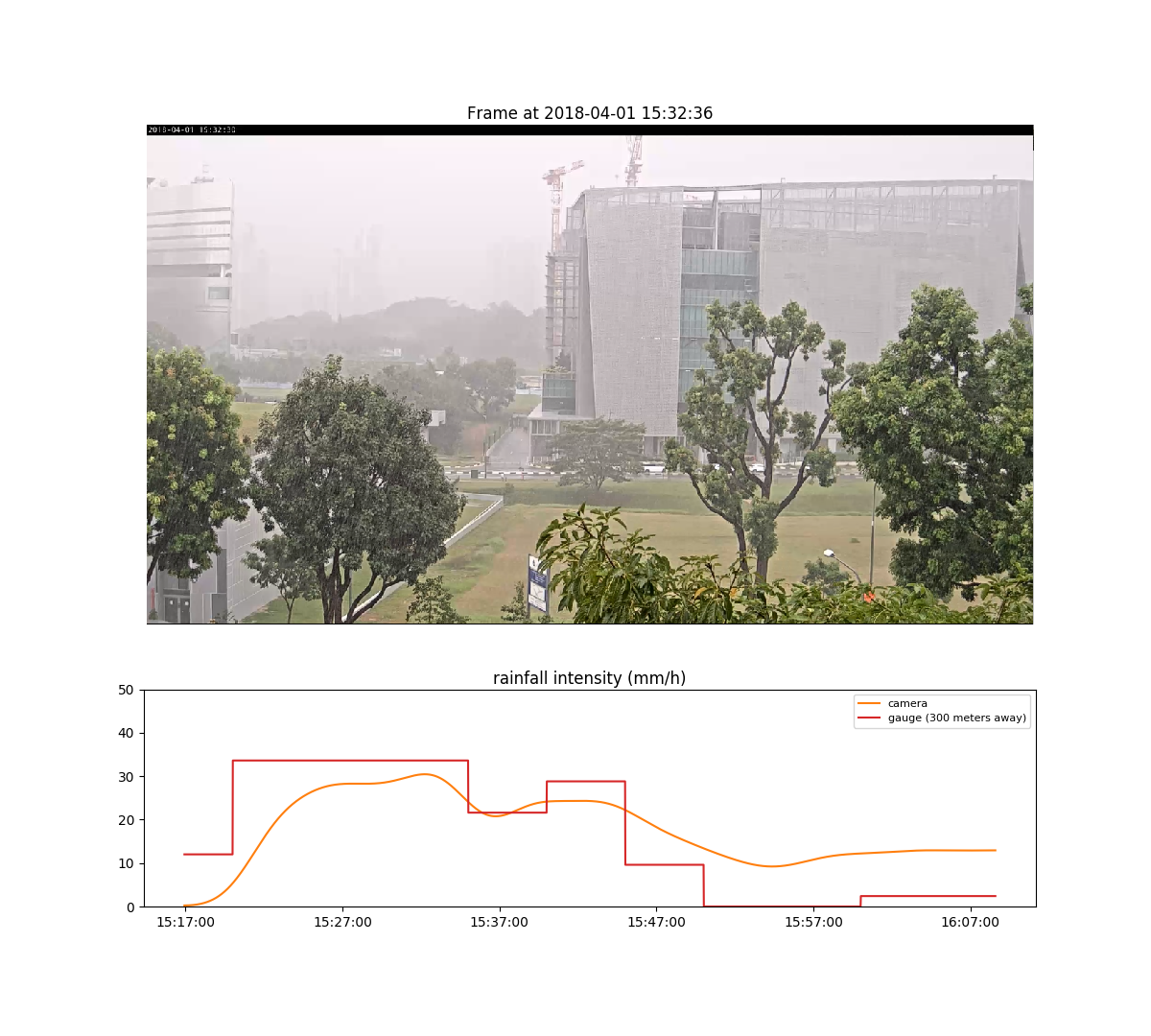
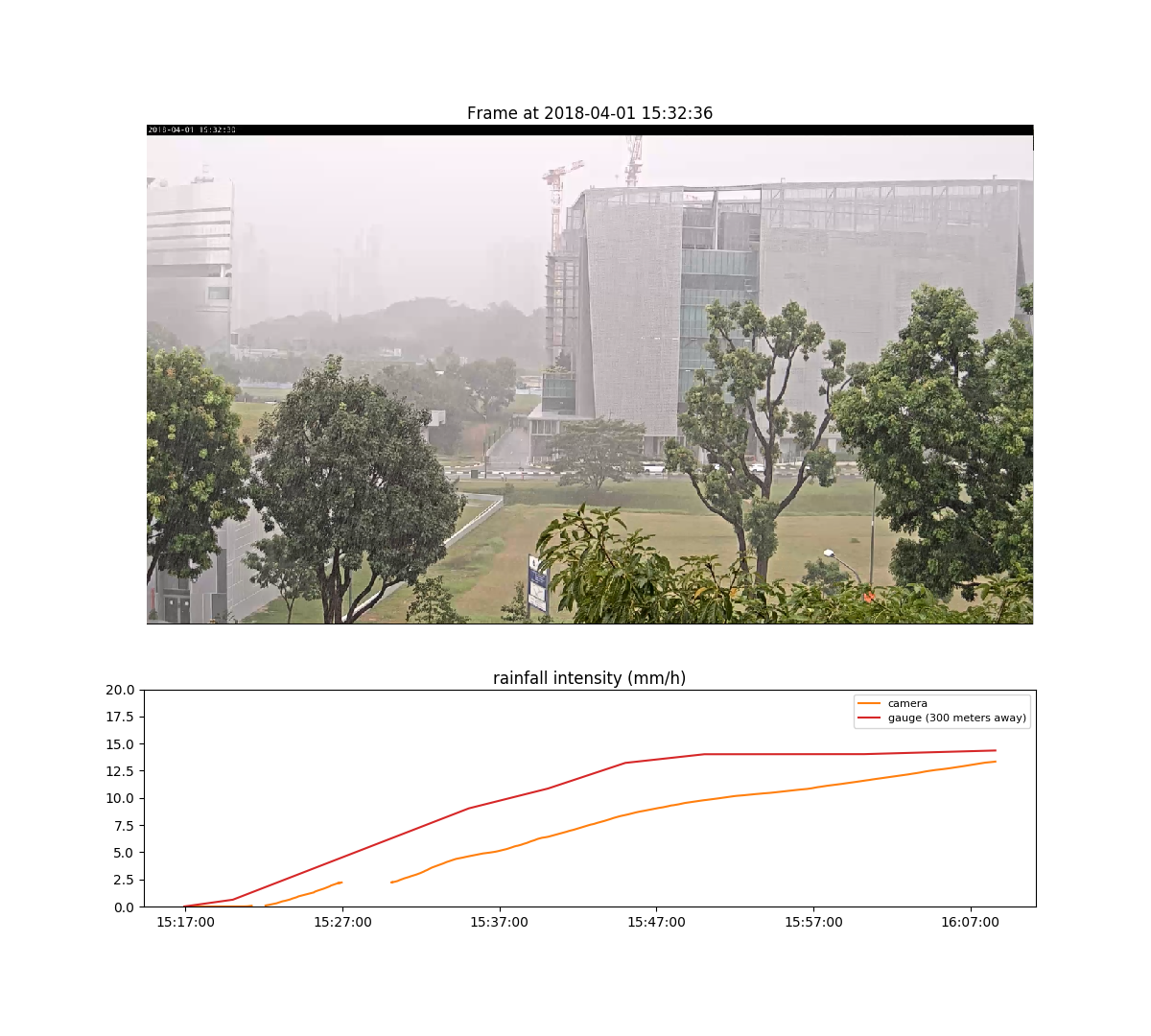
Till now, we don’t have any ideas about how to deal with the situation at night because of lack of atmospheric light. We’ve tried focus on the artificial light bulb on streets but the results are not satisfactory.

# Data Validation

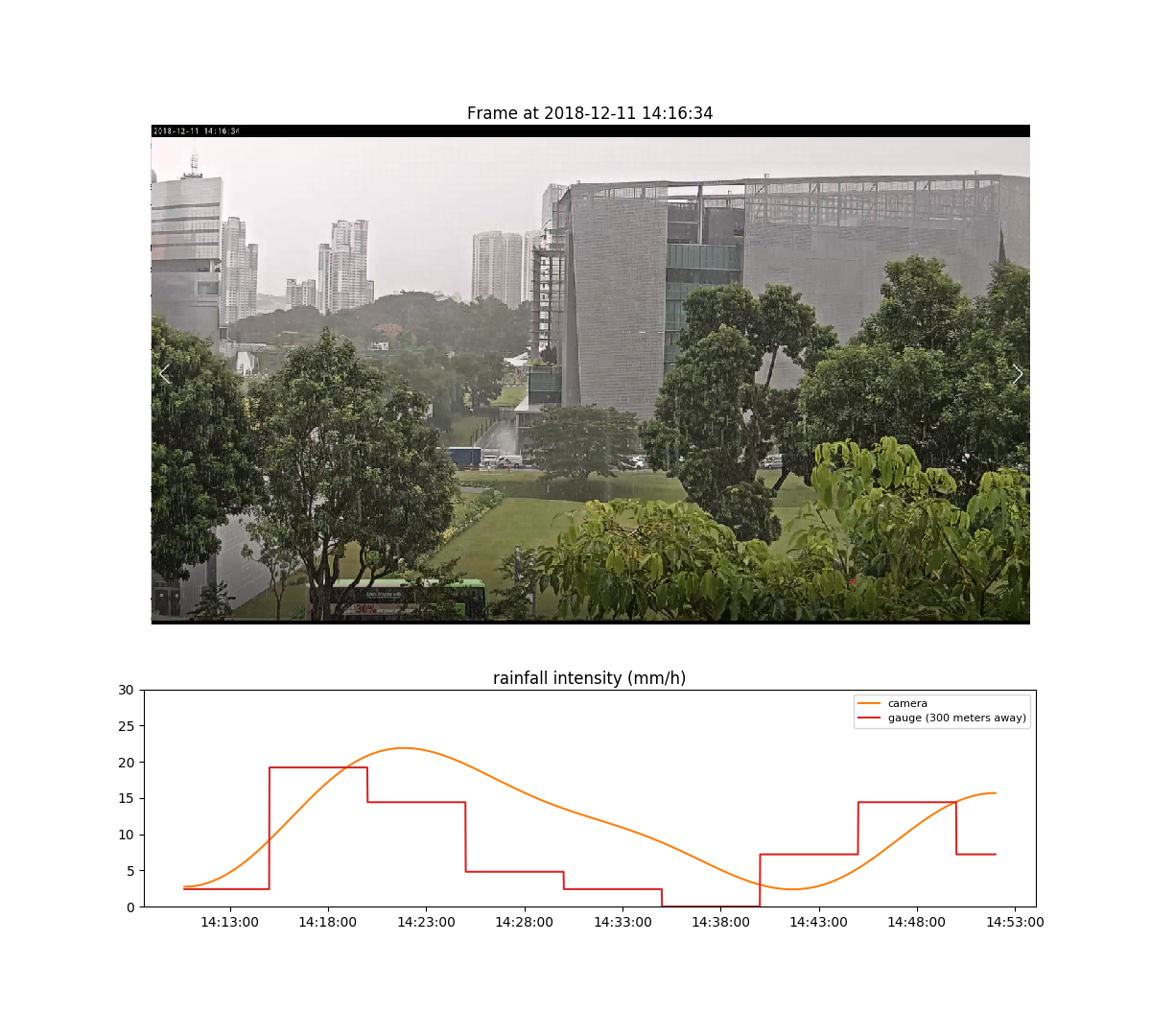
Table List of events that camera captured

|  |  |  |
| --- | --- | --- |
| Event | Total rainfall mm (estimated by radar) | duration |
| 2018-4-1 | 26 | 3 hours |
| 2018-12-8 | 10.2 | 1.5 hours |
| 2018-12-11 | 17.2 | 2 hours |
| 2018-12-12 | 23.4 | 3 hours |

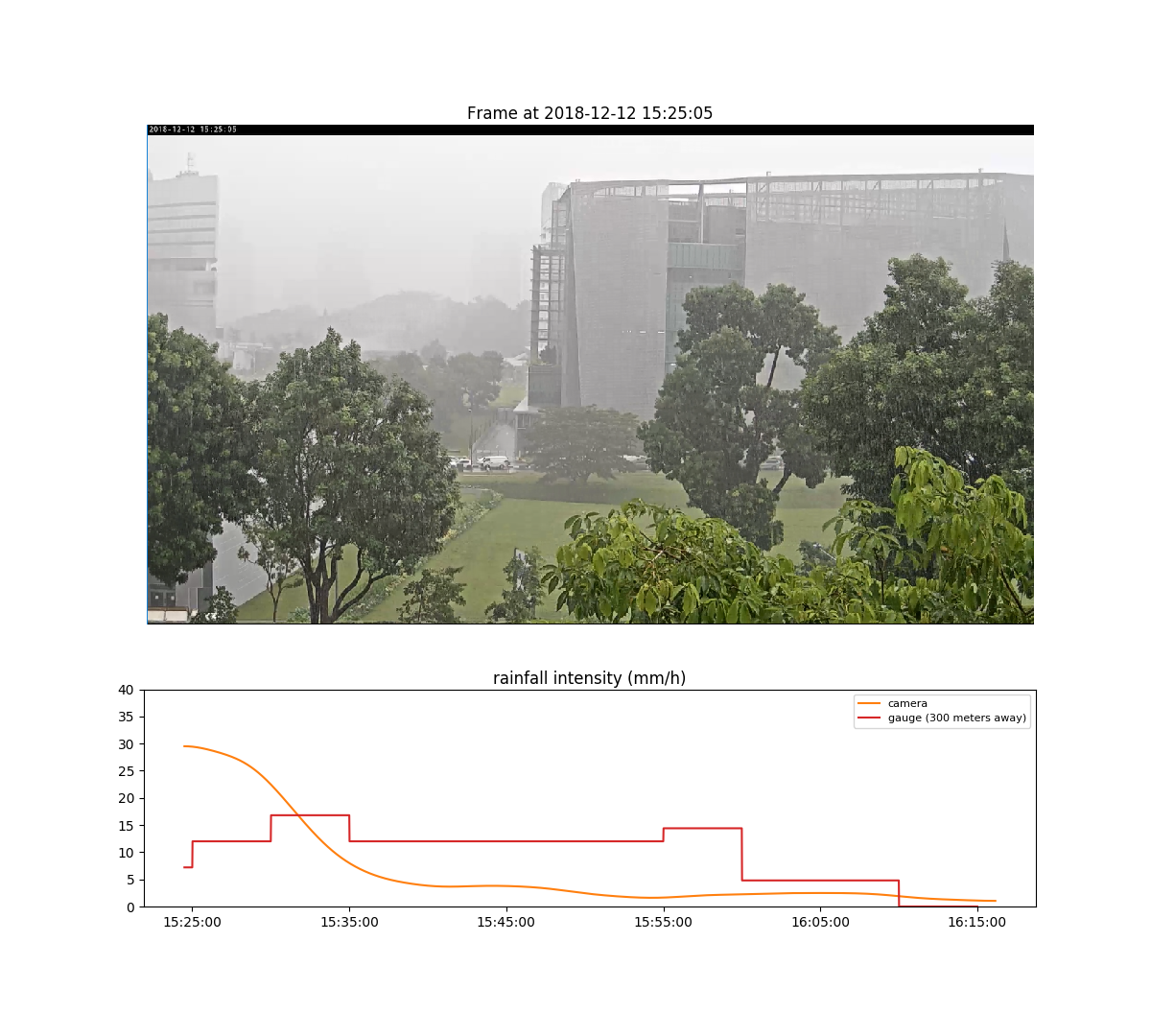
### 3.1 2018-4-1



### 3.2 2018-12-11



### 3.3 2018-12-12





# Real-time Deployment