Precipitation Recognition Using CCTV Video

Taeye Kwack Dohyeon Kim Hwisong Kim Jisoo Song

Abstract

High-resolution rainfall information is very important weather information because it can minimize meteorological disasters such as localized heavy rains. In this paper, we will propose a model that can recognize rainfall depth through CCTV videos. As rainfall in a single RGB image is hard to observed, it is essential to use optical flow to gain temporal raindrop movement. We use S3D, spatial and temporal separable 3D convolutions for our task because S3D is best-performing model among models that use optical flow and also computationally efficient. AWS station which measures precipitation and CCTV were installed on the roof of the same building (Building 49, SNU) and we selected the data from 2020-06–2020-09 and 2021-06–2021-09, where the precipitation is heavy. When we sample 8 frames from 5-minute video and train with batch size of 32, the accuracy is 64.32%. It is good result considering that precise rainfall classification in units of 0.1mm is given as a task in our research and we are short of computing power so that we only use 8, 16, 24 frames among 64 frames that we sample and we could not increase the batch size. We think that this could be improved through more diverse studies in the future; change of labels for classification, division of CCTV videos into day and night, training with more GPU, and so on.

1. Introduction

Rainfall is a very important factor in the water circulation process within the Earth and is closely related to global climate change and natural disasters. High-resolution rainfall information is very important weather information because it can minimize meteorological disasters such as localized heavy rains [1]. In addition, high-resolution rainfall information may be used to increase the accuracy of the model as an initial value of the numerical forecast model.

Currently, the Korea Meteorological Administration (KMA) installs the Automatic Weather System (AWS) nationwide to collect weather information. However, AWS stations are located 30 in Seoul and 980 nationwide, making it difficult to collect high-resolution rainfall information. Therefore, in this paper, we intend to estimate the rainfall depth through CCTV video data. If the rainfall depth can be estimated through CCTV images, high-resolution rainfall information can be obtained using CCTV data nationwide.

In this paper, we will estimate the rainfall from CCTV images using Separable 3D CNN (S3D) [2], which show excellent performance in recognizing the behavior of images. The composition of this paper is as follows. Chapter 2 shows related works, and Chapter 3 explains S3D's advantages and architecture. Chapter 4 discusses experimental setup, Chapter 5 shows the result of experiments, and Chapter 6 presents conclusions.

2. Related Works

Rainfall recognition research can be thought of as an image classification/regression task in the field of computer vision. Therefore, it is possible to estimate the rainfall depth by understanding the rainfall space pattern that changes in real time in the image through the model used in action recognition.

Ko et al. proposed a rainfall recognition method based on CNN-LSTM in images [5]. In this study, features were extracted through a pre-learned CNN model (Inception V3 [9]) for each frame of the image, and this was entered into the LSTM in the form of sequential data. The model was learned by dividing the rainfall in 10 minutes into 0.2, 0.5, and 1 mm, and the accuracy was about 80%.

Li et al. predicted rainfall through Temporal Segment Networks (TSN) [6]. In this study, after preprocessing of the observed rainfall, the rainfall was cumulated in units of 5 minutes and labeled with 0.1, 0.2, and 0.3mm to learn, and the accuracy was 70.8%.

Both models of the papers showed good results in estimating rainfall, but they have several limitations. First, in both papers, the distances between AWS stations and CCTV are far. The distance between the CCTV and the AWS station may lead to a mismatch between the image and the actual rainfall. In this paper, AWS and CCTV were installed on the roof of the same building (Building 49, SNU) to remove this mismatch. In addition, in this paper, the amount of data used for learning is much larger than that of the pre-

vious two papers. In the case of CNN-LSTM model, nine 10-minute-long videos were divided into 3-second videos and then used for training, and in the case of TSN model, 177 5-minute-long videos were used. In contrast, this paper intends to create a more accurate rainfall recognition model by training videos for 70 days of precipitation over two years which are a much larger amount of data compared to previous study.

3. Models

3.1. Why we chose S3D?

In this paper, Separable 3D CNN (S3D) [12], combined model of two-stream and one-stream approaches for video recognition, is used for our research. Conventionally, there are two approach for video action recognition, two-stream fusion and 3D convolution. Two-stream approach for video action recognition is composed of the spatial stream and temporal stream, and the results of each streams are fused to show video-level recognition score. For spatial stream, a sampled single RGB image is fed to learn appearance, and stacked optical flow fields are fed to learn motion. [7] One-stream approach learned the spatial and temporal information at once using 3D CNNs. One-stream approach is computationally expensive and the performance of using optical flow for temporal information is outperforming. Therefore, there are models that combined one-stream approach and two-stream approach like I3D, S3D. These models inflated the 3D CNNs into 2D CNNs and the optical flow fields are fed into temporal stream. Among those models, S3D is selected for recognizing rainfall depth.

The reason for selecting S3D is that it uses optical flow fields. Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image [4]. As rainfall in a single RGB image is hard to observed unless heavy downpour occur [6], temporal information of raindrop movement is important for rainfall depth recognition. By using S3D, temporal information can be fed by optical flow fileds. Also the light issue, usually a big issue for the outdoor image processing, can be treated by using optical flow. Our dataset is outdoor CCTV videos and it can be very sensitive to light environment such as cloudy and clean sky, or day and night.

I3D also uses optical flow fields, but S3D model is lighter than I3D. Considering the dataset we have and the computing power, we decide to use S3D for our research.

3.2. S3D

S3D is an upgrade version of I3D [2] for video recognition. By retaining 3D temporal convolutions at the top layer, and using 2D temporal convolutions for the lower layer (closer to the pixels), S3D model became faster and more accurate than I3D. Moreover, replacing 3D convolutions with spatial and temporal separable 3D convolution (k_t x k x k by 1 x k x k and k_t x 1 x 1) lead to less parameters and more computationally efficient than standard 3D convolution [12].

The architecture of S3D is similar to I3D, and I3D is similar to GoogLeNet [8]. The I3D model is 3D version of GoogLeNet and the S3D changed 3D convolutions to spatial and temporal separable 3D convolution like Inception-v2,v3 [9]

Our model is simplified S3D model(Fig. 1). Only two Sep-Inc block is used. One stream is for RGB frames(spatio) and the other stream is for optical flow frames(temporal). To concatenate two streams, we compared score average late fusion and late feature fusion by concatenating after flattening. Between them, score average fusion is widely used, and we adopted it.

4. Experiment Setup

4.1. Datasets and Observation Devices

Precipitation data was acquired from Korean government AWS record[1], on 509 post which is located in Seoul national university. CCTV videos were acquired courtesy of the Graduate School of Environmental Studies. The camera is installed nearby the AWS device. CCTV footage is 30FPS, 1920 pixels width, 1080 pixels height. You can see the CCTV specification from Table. 1 We selected the data from 2020-06 – 2020-09 and 2021-06 – 2021-09, where the precipitation is heavy. The dataset overview is provided in Table. 2

Table 1. CCTV specification

Item	Value
Frame size	1920*1080
Video length	5 minutes
Frames per second (fps)	30
Numbers of frames	9000

Table 2. Dataset Overview

Label / Year	5-minute rainfall depth(mm)					
	0.1	0.2	0.3	0.4	0.5	Total
2020	569	925	548	366	338	2746
2021	413	230	198	179	123	1143
Sum	982	1161	747	548	461	3889

[1] https://data.kma.go.kr/data/grnd/selectAwsRltmList.do

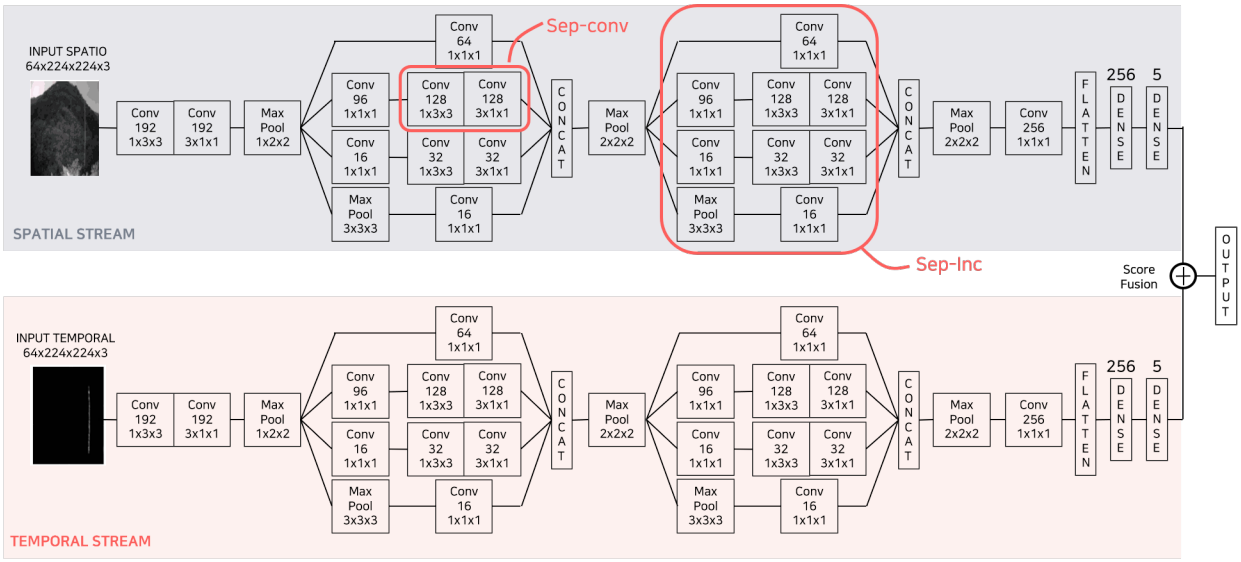

Figure 1. Structure of S3D model that we use

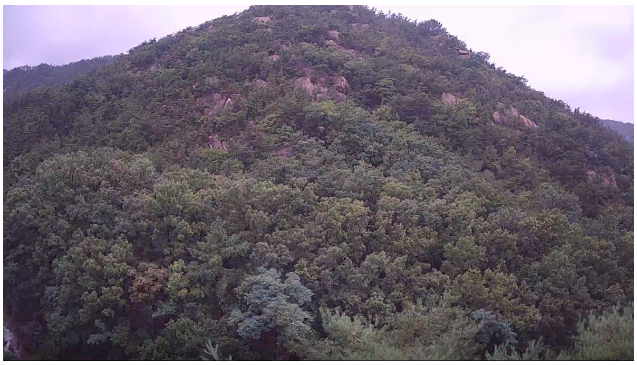

Figure 2. Capture of CCTV video

4.2. Preprocessing

4.2.1 Precipitation Labelling

Rainfall recognition is performed on frames and total precipitation in a fixed duration of time. Li et al. choosed 5 minutes rainfall value to train the model [6], and we use same setting for this study.

AWS periodically measures the precipitation. For example, Korean Meteorological Administration provides the rainfall data measured every minute. However the precipitation value is sporadic because of the mechanism of AWS device. It collects the rain in a container, and when water level reaches threshold value the device records it and empties the container. Threshold depth of KMA AWS is 0.5 mm, therefore if the precipitation is 0.1 mm/min the device will report 0 mm rainfall for 4 minutes and 0.5 mm at the fifth minute. To prevent this zero-rainfall-but-raining data from being confused with non-rainfall data, additional sensor provides the information whether it is raining or not.

There are four types of data by the rainfall value and whether it is raining.

1. Non-rainfall : zero rainfall, not raining
2. Zero-rainfall : zero rainfall, raining
3. Nonzero-rainfall : nonzero rainfall, raining
4. False-rainfall : nonzero rainfall, not raining

Date & Time	1-Minute Rainfall Depth (mm)	Is Raining or Not (Yes: 10, No: 0)	1-Minute Rainfall Depth (mm)	5-Minute Rainfall Depth (mm)
2021-08-31 21:56	0	10	0	
2021-08-31 21:57	0	10	0	
2021-08-31 21:58	0	10	0	
2021-08-31 21:59	0	0	0	
2021-08-31 22:00	1	0	0	
2021-08-31 22:16	0	10	0.25	
2021-08-31 22:17	0.5	10	0.25	
2021-08-31 22:18	0	10	0.125	
2021-08-31 22:19	0	10	0.125	
2021-08-31 22:20	0	10	0.125	0.875
2021-08-31 22:21	0.5	10	0.125	0.75
2021-08-31 22:22	0	10	0.25	0.75
2021-08-31 22:23	0.5	10	0.25	0.875
2021-08-31 22:24	0.5	10	0.5	1.25
2021-08-31 22:25	0	10	0.25	1.375
2021-08-31 22:26	0.5	10	0.25	1.5
2021-08-31 22:27	0.5	10	0.5	1.75
2021-08-31 22:28	0.5	10	0.5	2
2021-08-31 22:29	1	10	1	2.5

Non-rainfall | False-rainfall | Zero-rainfall | Nonzero-rainfall

Figure 3. Estimate 5-minute rainfall depth.

What we want to pay attention to is zero-rainfalls followed by nonzero-rainfall. Here, the data needs to be averaged into continuous rainfall and then accumulated before

being converted to label. We apply the following rules for averaging.

- When zero-rainfalls are followed by nonzero-rainfall, continuous zero-rainfalls and one nonzero-rainfall which terminates them are grouped and averaged. In the Fig. 3, precipitation between 22:18 and 22:21 is averaged into 0.125 (which is dividing 0.5 by 4)
- If zero-rainfalls are not followed by nonzero-rainfall, ignore these zero-rainfalls. In the Fig. 3, precipitation 21:56 to 21:58 is ignored.
- Non-rainfalls and Nonzero-rainfalls not following by zero-rainfalls are left intact.
- False-rainfalls are ignored.

The second rule requires further justification. The cause of this is either one of these:

- Precipitation is so small that the rain ceases before reaching the threshold value.
- Rain detection sensor malfunctioned.

We cannot tell the actual rainfall here, so we set it NaN and ignore this data. The drawback of this approach is that if the container is partially filled without being measured and rain starts again shortly after, previous rainfall might be added up. However the benefit, which is that we can avoid the error from sensor malfunction, exceeds the drawback therefore we adopt this strategy.

After averaging out, the data area accumulated by each five neighbouring data to calculate 5-min rainfall.

4.2.2 Temporal Sampling

Full 5-min video, which is 9,000 frames with 30 FPS, is to large to be directly fed to the module. Therefore we apply temporal sampling and spatial sampling.

Temporal sampling is done be randomly selecting 72 frames from 9,000 total frames. To apply optical flow, the frames must be continuous. However setting all 72 frames to be continuous cannot capture the change of precipitation during 5-minutes. Therefore we segment the frames into eight equal-length intervals and sample 9 continuous frames from each interval. This ensures the random frames to be continuous and evenly picked from the entire 5 minute frames.

To train S3D model, we need RGB frames and optical flow frames. From 8 groups of 9 frames, 8 optical flows are retrieved. This makes up to total 64 optical flow images. To make the number of RGB frames same with the number of optical flow images, we abandon the first frame of each group and save only the rest. As a result, we have 8 groups

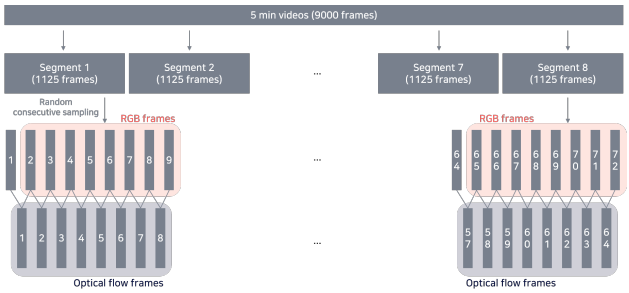

Figure 4. Temporal Sampling method

of 8 RGB frames saved, which make up to total 64 RGB images.

However, later it turned out that 64 RGB images and 64 optical flow images were too much for our computer. Due to the lack of machine performance, we had to use only a small number of images from 64 set. We tried 8, 16, and 24 images from the start in 64 images for training because of this reason.

4.2.3 Spatial Sampling

We use three ways of spatial sampling method. The S3D model basically uses an image of 224×224 size as input. Since our original image has a resolution of 1920×1080, we need to convert it to 224×224 size image. [12] resized input frames to 256×256 and then took random (for training) or center (for evaluation) crop of size 224×224. If we follow the way they used, the movement of the raindrops can be erased by resizing it to 256×256 size because the size of the raindrop is in pixel units which is so small. Therefore, we apply several sampling ways to preserve the movement of raindrops as follows. Fig. 5 delineates this.

1. Take 224×224 center crop from the original image.
2. Take 1080×1080 center crop from the original image, and then resize it to 224×224.
3. Take 1080×1080 center crop from the original image, resized it to 540×540, and then take 224×224 center crop.

Keeping the similarity to original frame and enlarging the raindrop are in trade-off relation. Method 1 only consider the center of original image. Method 2 almost preserves the original image and method 3 is the most balanced approach. We will test each method and compare the accuracy.

4.2.4 Optical flow

As mentioned earlier, it is important to use optical flow to recognize precipitation well through CCTV or video. To ef-

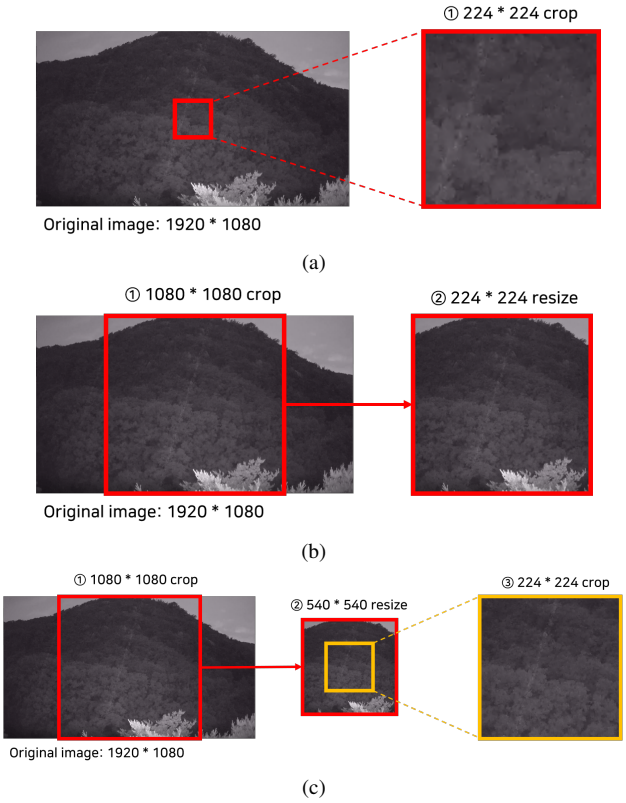

Figure 5. (a) Method 1, (b) Method 2, (c) Method 3

fectively detect raindrop in videos with fixed filming locations such as CCTV, separation from the background can have effective results [6]. Images mainly used in temporal streams use images that emphasize the movement of subjects separated from the background to effectively show frame changes over time. Therefore, I would like to compare the accuracy of the following four versions. Fig. 6 shows our preprocessed image sample.

1. Grey difference

   In addition to optical flow, RGB difference and warped optical flow were additionally used in the TSN as input for temporal stream [10]. RGB difference calculates difference between frames over R, G, and B bands respectively. It only uses simple differences, which has a limitation that it can be used only when the color or contrast change of the background and object is small in a fixed position camera. Precipitation images can be applied because the location is always fixed, and there is no significant difference in color or contrast in units of frames. When we filter grey scale for the frames and calculate grey difference, it catches temporal difference of raindrop well compared to RGB difference. For this reason, we use grey difference as a preprocess method.

2. Dense Optical flow : TVL1, Farnebacks

   The optical flow is divided into a sparse optical flow and a dense optical flow according to the calculation method. Since sparse optical flow tracks using feature points such as corners, the computation amount is small. However, compared to the sparse optical flow, dense optical flow takes longer time but is more accurate because it calculates all the changes for each pixel. We conduct a test on two methods of dense optical flow. This is because the size of raindrops is small, requiring pixel-by-pixel calculation and high accuracy. Among them, the most widely used Farneback and TVL1 were tested. The Farneback optical flow proposed by Gunnar Farneback is an algorithm that suggests a dense optical flow. The intensity and direction of optical flow are presented as the result of the 2d vector [3]. TVL1 (Total Variation L1) is a kind of dense optical flow suggested by wedel, pock, and zach. Through TVL1, discontinuities in optical flow field are preserved ny using total variation(TV) regularization and also singular value is calibrated by using robust L1 norm [11].

3. Warped optical flow

   Warped optical flow removes the movement of the background and focuses more on the movement of the subject. It is mainly used to remove the movement of the camera. Since CCTV is in a fixed state, the movement of the camera is not large, but it was used to correct the case of shaking due to strong winds.

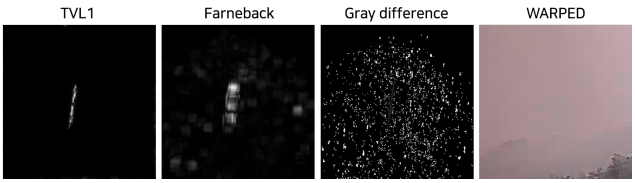

Figure 6. Image preprocessed by several optical flow method

5. Experiments

5.1. Input modalities

Considering our limited time and computing power, it is judged difficult to test all the cases presented above. Therefore, we will examine the performance of pre-processed images in the model through the Grey differentiation and normal optical flow options (TVL1, Farneback, Warped optical flow) described above, and use only the top two of them. For the rapid progress of the experiment, CCTV videos only in the August 2020 were used which is divided into 396 for train, 90 for valid and 50 for test. Batch size is 4 and the

number of frames is 16. Accuracy is defined as the number of data that predict correct label out of the total number of tests when the number of epochs is 20. (We proceeded to 20 epochs because even if the epoch number increased over 20, the account did not improve significantly and also we are short of time and computing power.)

Table 3. Experiment result on different input modalities

Input	Accuracy (%)
Grey difference	45.83
TVL1	62.50
Farneback	47.92
Warped optical flow	58.33

The result of the experiment is shown in Table. 3. The performance of TVL1 and Warped optical flow is more than 10% better than that of the other two methods. Therefore, the pre-processing method is determined with these two.

5.2. Results of Experiments

As mentioned in section 4.2.2, putting all of the extracted 64 frames into input is difficult considering the current computing power. When all 64 frames were used, batch size was possible up to 2, when the accuracy was as low as 30Since batch size is the number used to update the weight of the convolution, when the batch size is too small, the weight is not updated properly, so it seems that the accuracy is low. Considering the current computing power, it is difficult to increase the number of batch sizes and frames at the same time. Therefore, we reduced the number of frames and instead increased the batch size. When the number of frames was 8, 16, and 24, the maximum batch size available was 32, 16, and 8. These are our Temporal Sampling options. The three methods covered in section 4.2.3 were designated as the Special Sampling options. In addition, the experiment was conducted using TVL1 and Warped optical flow, which have good Accuracy in section 5.1, as inputs with RGB frame. Therefore, the experiment is conducted with a total of 12 settings using 3 Temporal Sampling, 3 Special Sampling, and 2 Optical flows, respectively. The experimental results for each are shown in Table. 4

Table 4. Experimental results for various conditions.

Spatial \ Temporal	Method 1 (%)		Method 2 (%)		Method 3 (%)	
	RGB+TVL1	RGB+Warped	RGB+TVL1	RGB+Warped	RGB+TVL1	RGB+Warped
8 frame, 32 batch	55.73	56.77	**64.32**	60.42	57.81	49.74
16 frame, 16 batch	61.2	54.69	60.16	60.68	55.21	54.43
24 frame, 8 batch	54.95	56.25	58.59	53.65	58.8	57.81

According to the results, the the accuracy is the best at 64.32% when the number of frames is 8, the batch size is 32, TVL1 is used for optical flow, and Method 2 is used for spatial Sampling. In case of temporal sampling, the accuracy of the case of frames with 8, batch size with 32 and the accuracy of the case of frames with 16, batch size with 16 is quite similar and slightly better than the other cases. Actually, we test the case of 64 frames with 2 batch size which accuracy is around 30%. Through this result, it can be seen that batch size and frame number plays a significant role in improving outcomes.

In case of optical flow, TVL1 outperform Warped optical flow. We think that it's because TVL1 is a kind of dense optical flow which is accurate because it calculates all the changes for each pixel. Warped optical flow can remove the movement of the background, but our CCTV instrument is fixed in position so it has little effect on the optical flow. In case of spatial sampling, the averaged accuracy with method2 (59.65%) is better than other methods(method 1, 2 : 56.6%, 55.63% respectively). Method 2 can represent much more pixels than method 1, and is more simple than method 3. Because of this property, the accuracy is seem to best in method 2.

6. Discussion

The results of our experiment are somewhat close to the accuracy of 70% of existing papers using TSN. However, since the paper was classified as 0.1, 0.2, and 0.3mm, there is a possibility that the accuracy will be higher than that of our models classified as 0.1, 0.2, 0.3, 0.4, and 0.5mm. The papers that showed 80% accuracy using CNN-LSTM were also classified into 0.2, 0.5, and 1 mm. Therefore, considering the task of our model, which requires precise precipitation classification at 0.1 mm intervals, the accuracy of the model is not bad.

In addition, our study have the advantage of collecting data in the same place by the two instruments, unlike previous papers, which were far from AWS station and CCTV. Also, we have a advantage in that we have much more training data than previous studies. Unfortunately, despite of these advantages, our research has not improved much in accuracy. Due to the lack of computing power, we could not use all of the 64 frames we extracted and could not increase batch size. It can be found in our experiments that the number of frames and batch size contributed greatly to accuracy. Definitely, it will produce better results in a better equipment environment.

In addition to computing power, we think that the accuracy could be improved through more diverse studies in the future. For example, if we classify rainfall into a wider range as in the previous paper, it will show better accuracy. Alternatively, it may be solved as a regression problem rather than a classification problem. If Loss function is set to MSE, the model may be able to recognize rainfall more accurately.

Another way to improve the accuracy of the model is to divide the data into day and night. As it can be see in the Fig. 7, raindrops reflect the light at night, making it look

(a) (b)

Figure 7. (a) CCTV capture during the day (b) CCTV capture during the night. Both videos are labeled 0.5mm.

brighter even if they are labeled the same label, 0.5mm. Since these differences may have hindered the learning of the model, the accuracy can be improved if the data of day and night are divided and trained respectively.

7. Conclusions

In this study, the authors trained the S3D model to recognize rainfall depth from CCTV video. Dataset were labelled and collected from raw data, and temporal sampling was applied. On temprally sampled frames,three different types of spatial samplings were applied and compared. From sampled videos, optical flow frames and RGB frames were extracted to train the model. Various optical flow algorithms were applied and compared. Because of the limit of computational resources, author couldn't use the entire sampled frames to train the model. Nevertheless, the performance with 64.32 % was achieved. Compared to previous studies, this work has advantages of better dataset and finer labels. With more computational resources, it is expected to give much better performance.

References

[1] Peter Berg, Lars Norin, and Jonas Olsson. Creation of a high resolution precipitation data set by merging gridded gauge data and radar observations for sweden. *Journal of Hydrology*, 541:6–13, 2016. 1

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 2

[3] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003. 5

[4] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 2

[5] B Ko, Z Li, and H Choi. Determination of precipitation from road cctv video by using cnn-lstm. In *Proceedings of Korea Software Congress*, pages 820–822, 2017. 1

[6] Zhun Li, Jonghwan Hyeon, and Ho-Jin Choi. Rainfall recognition from road surveillance videos using tsn. *Journal of Korean Society for Atmospheric Environment*, 34(5):735–747, 2018. 1, 2, 3, 5

[7] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014. 2

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1, 2

[10] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 5

[11] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l 1 optical flow. In *Statistical and geometrical approaches to visual motion analysis*, pages 23–45. Springer, 2009. 5

[12] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 2, 4