Rainfall Estimation from Traffic Cameras

Remmy Zen[1(✉)], Dewa Made Sri Arsa[2], Ruixi Zhang[1], Ngurah Agus Sanjaya ER[2], and Stéphane Bressan[1]

[1] National University of Singapore, Singapore, Singapore
{remmy,zhangruixi}@u.nus.edu, steph@nus.edu.sg
[2] Universitas Udayana, Bali, Indonesia
{dewamsa,agus_sanjaya}@unud.ac.id

2019 International Conference on Database and Expert Systems Applications

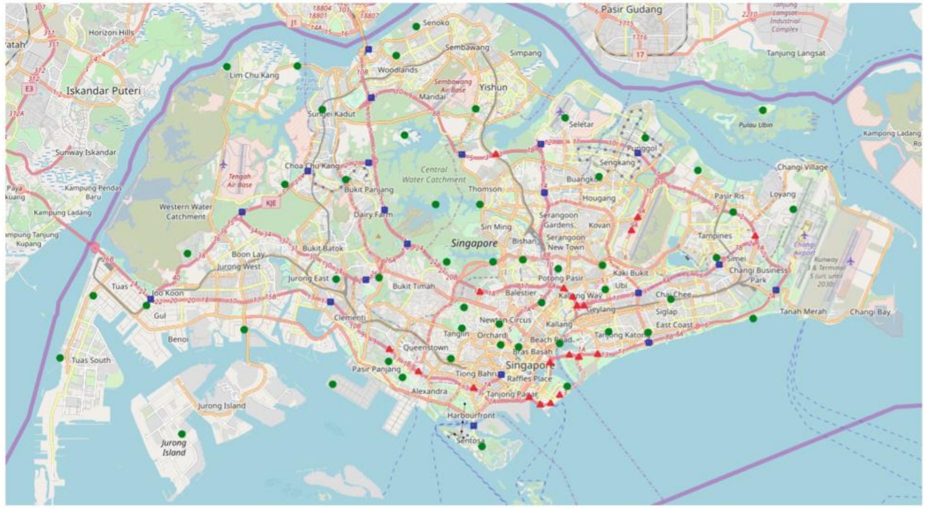
Tian Liu

Nov 23, 2020

Outline

- Background and Objectives
- Methodology
  - Local Rain Function
  - Global Rain Function
  - Data set
- Results and Discussion
- Conclusions

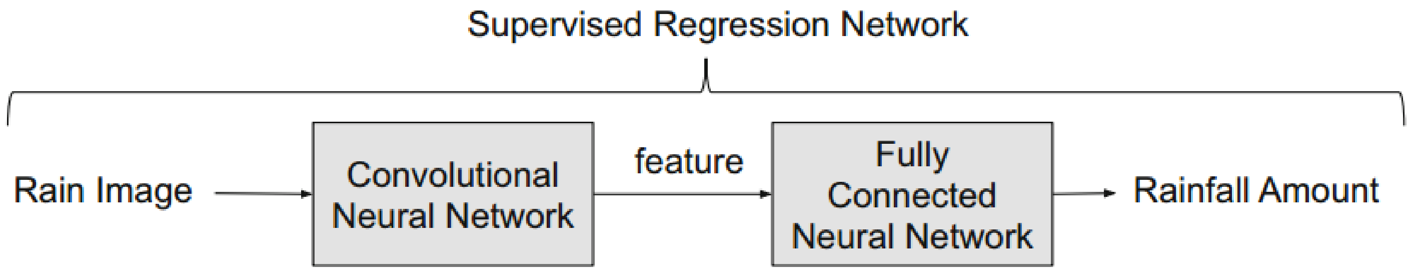
Background and Objectives



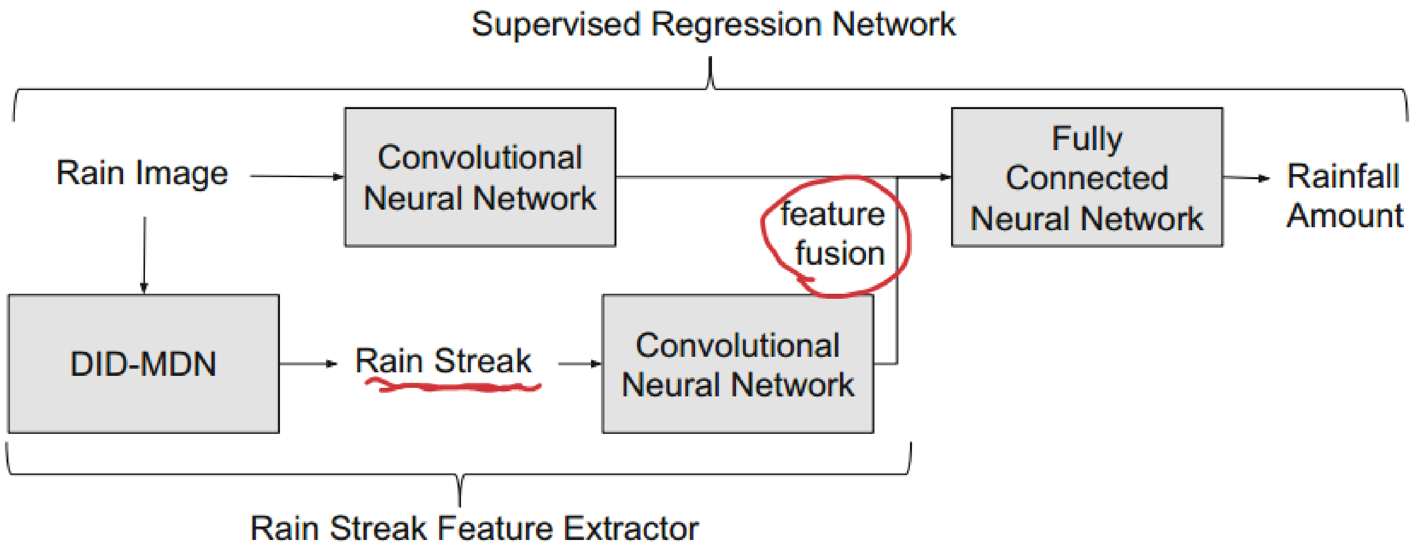
- Estimate rainfall value from traffic camera images
- Learn local rain function through Neural networks
    - Train a CNN for each camera (location-specific)
    - Standard model: VGG network with 19 layers
    - Hybrid model: VGG + Density-Aware Multi-stream Densely CNN (2018) as rain streak extractor

- Global rain function
    - Interpolate the estimated rainfall values from cameras
    - Compare 3 interpolation methods: Nearest Neighbors, Inverse Distance Weighting, Kriging
- Results show the hybrid model and Inverse Distance Weighting yield better results with best average accuracy nearly 1%

Local Rain Function by CNN: Standard vs. Hybrid



(a) Standard model.



(b) Hybrid model.

VGG: Very Deep CNN

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Multi-stream Densely-connected De-raining Network

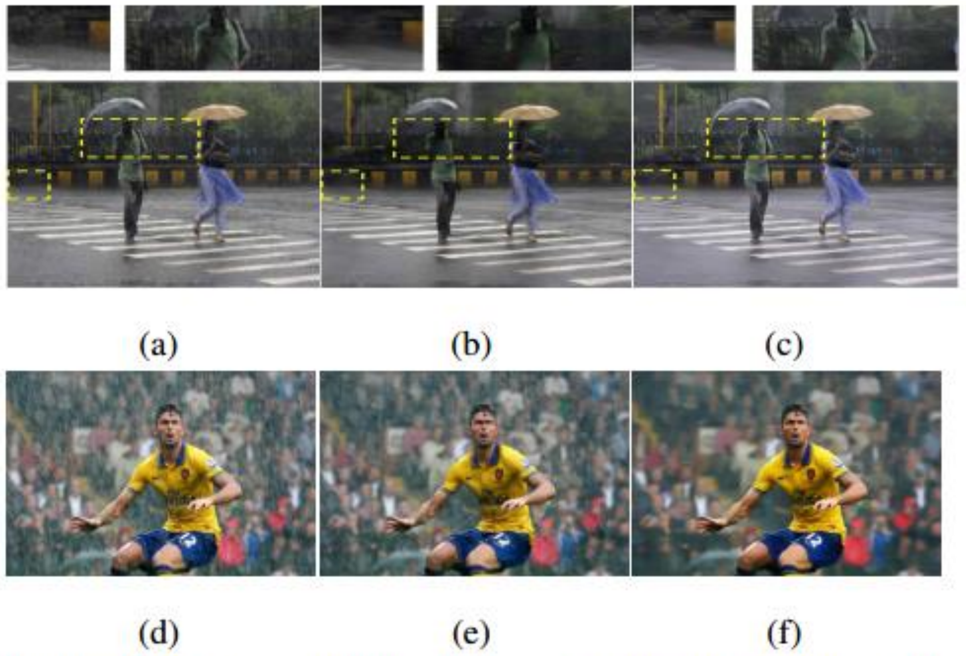


Figure 1: Image de-raining results. (a) Input rainy image. (b) Result from Fu *et al.* [6]. (c) DID-MDN. (d) Input rainy image. (e) Result from Li *et al.* [33]. (f) DID-MDN. Note that [6] tends to over de-rain the image while [33] tends to under de-rain the image.

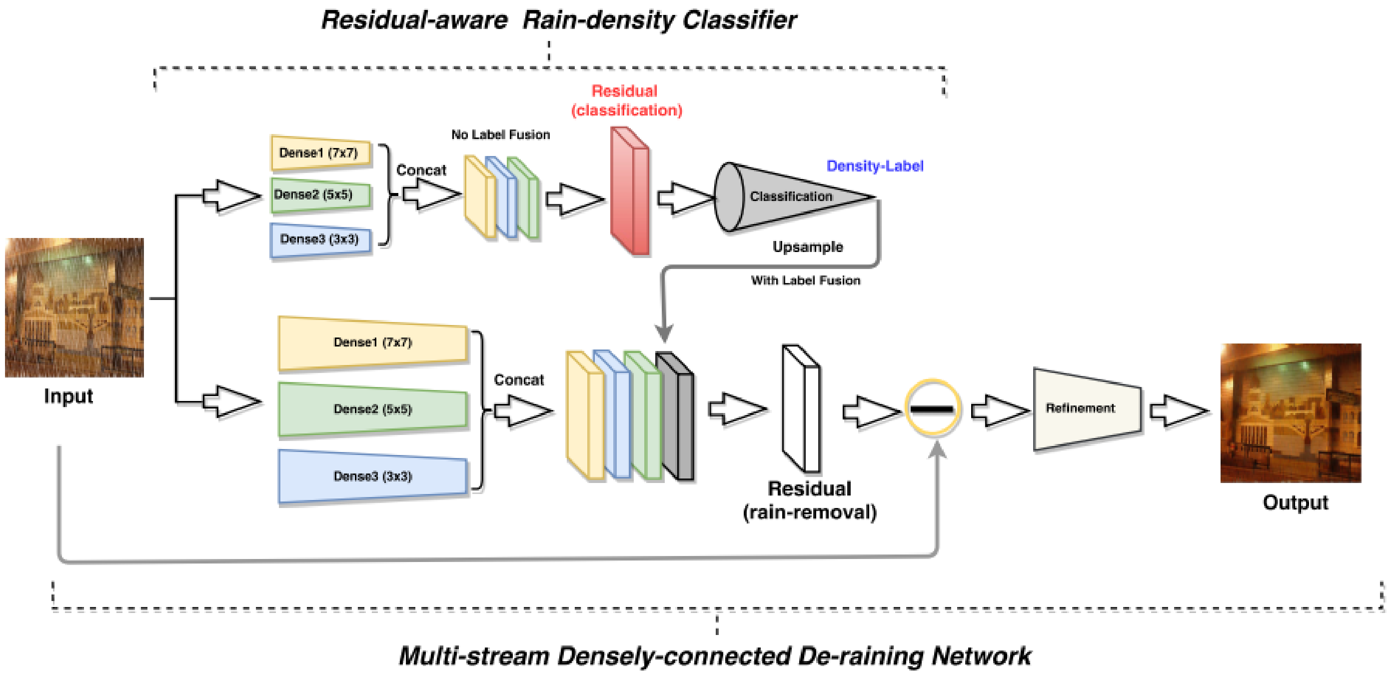
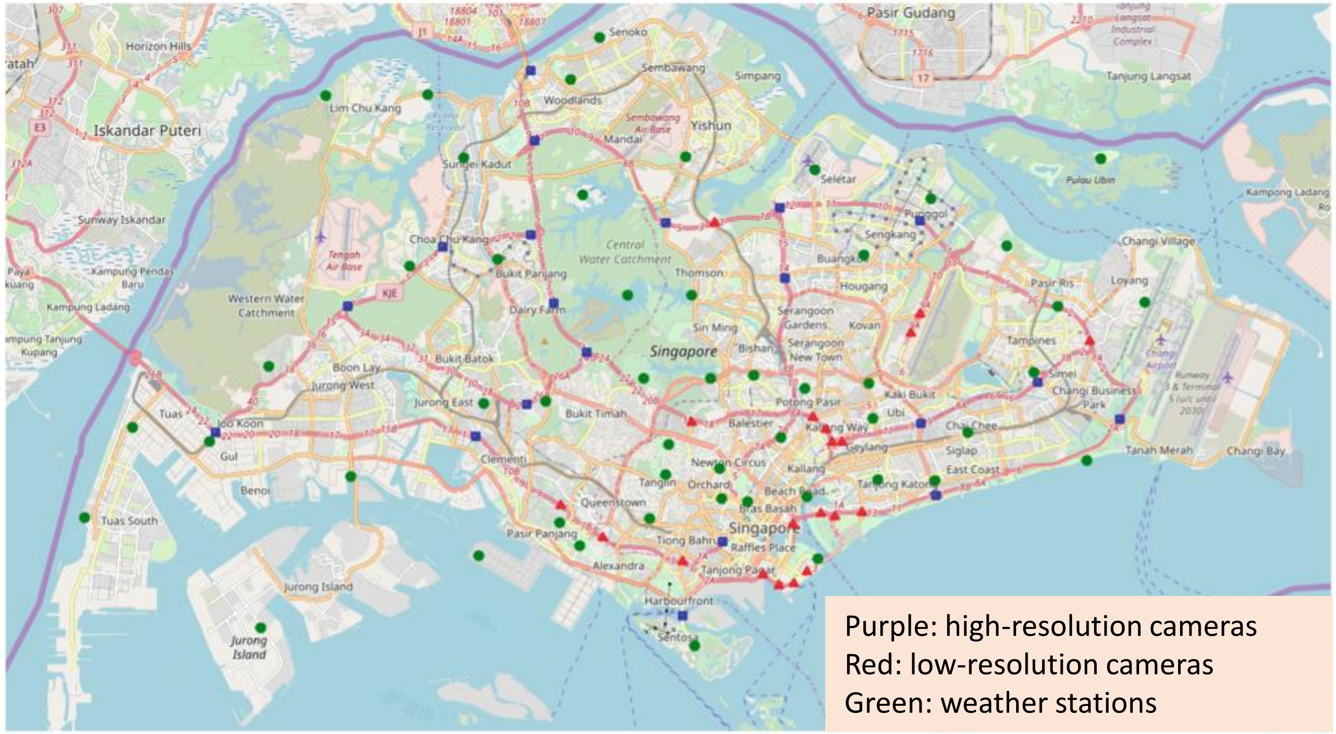


Figure 2: An overview of the proposed DID-MDN method. The proposed network contains two modules: (a) residual-aware rain-density classifier, and (b) multi-stream densely-connected de-raining network. The goal of the residual-aware rain-density classifier is to determine the rain-density level given a rainy image. On the other hand, the multi-stream densely-connected de-raining network is designed to efficiently remove the rain streaks from the rainy images guided by the estimated rain-density information.

Video Data

- Feb 2018 – Feb 2019, daytime (6am to 7pm) images for every 10 minutes
- Total 85 traffic cameras, select 20 high-resolution and 20 low-resolution for implementation
- Feb 2018- Oct 2018 as training and validation (70% for training, 30% for validation)
- Nov 2018 – Feb 2019 for testing



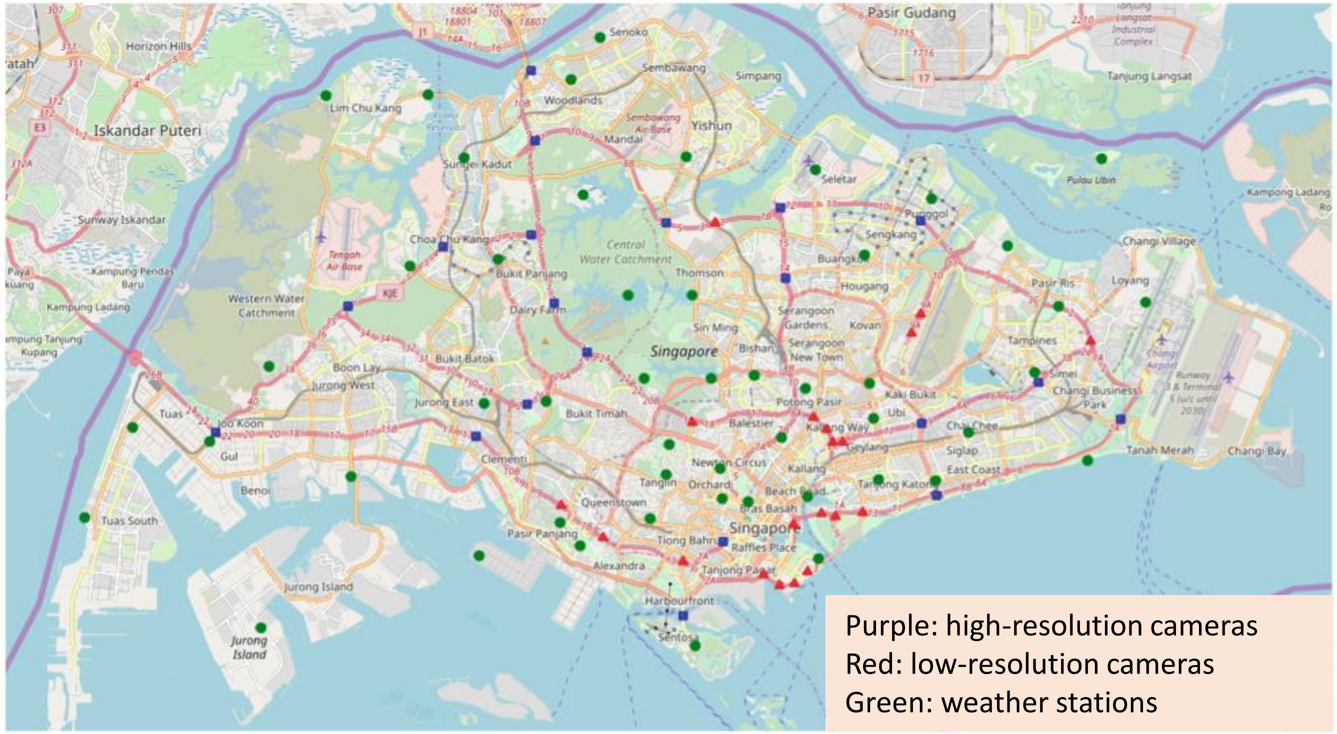
(a) Example image of high-resolution camera with 640 × 480 resolutions.

(b) Example image of low-resolution camera with 320 × 240 resolutions.

Fig. 2. Example images from high-resolution and low-resolution cameras.

Rain Data

- 5-minute rainfall values from 55 weather stations
- tipping bucket with 0.2-mm rainfall precision
- Use interpolated rainfall values at each camera location to train each camera (Nearest Neighbor, Inverse Distance Weighting, Kriging)
- Hope interpolated data sufficient for training, but may not be correct. Thus, evaluate the model in global rain function
- Overcome imbalanced data (3.31% rainy) set by random oversampling (with replacement)

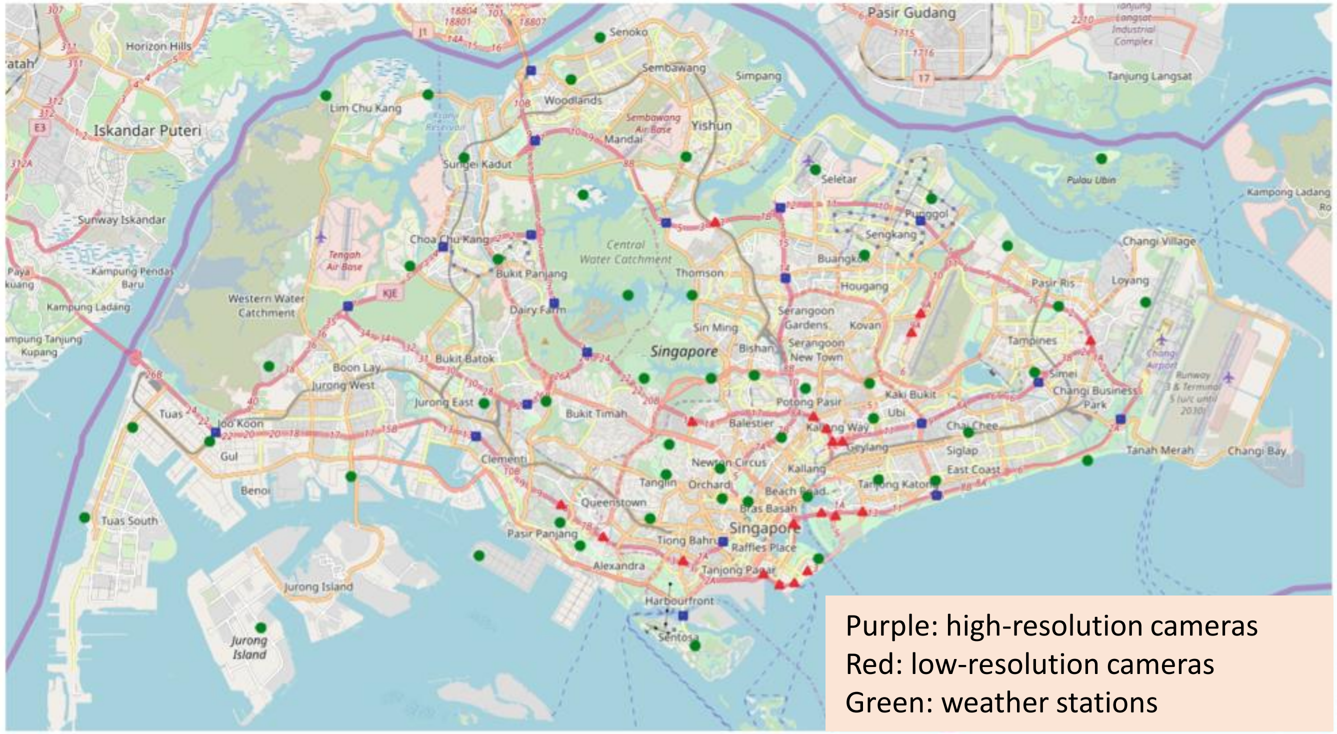


$$v = v_{\text{argmin}_x(\text{distance}(l, l_x))}$$

$$v = \frac{\Sigma_x(w(l, l_x) \times v_x)}{\Sigma_x w(l, l_x)}$$

Global Rain Function

- With trained models, estimate the rainfall amount at each camera location
- Interpolate from the estimated rainfall at camera locations for rainfall at each weather station
- Calculate the error based on ground truth
    - MSE
    - MAAPE (mean arctangent absolute percentage error): normalized to 0-100%



Purple: high-resolution cameras
Red: low-resolution cameras
Green: weather stations

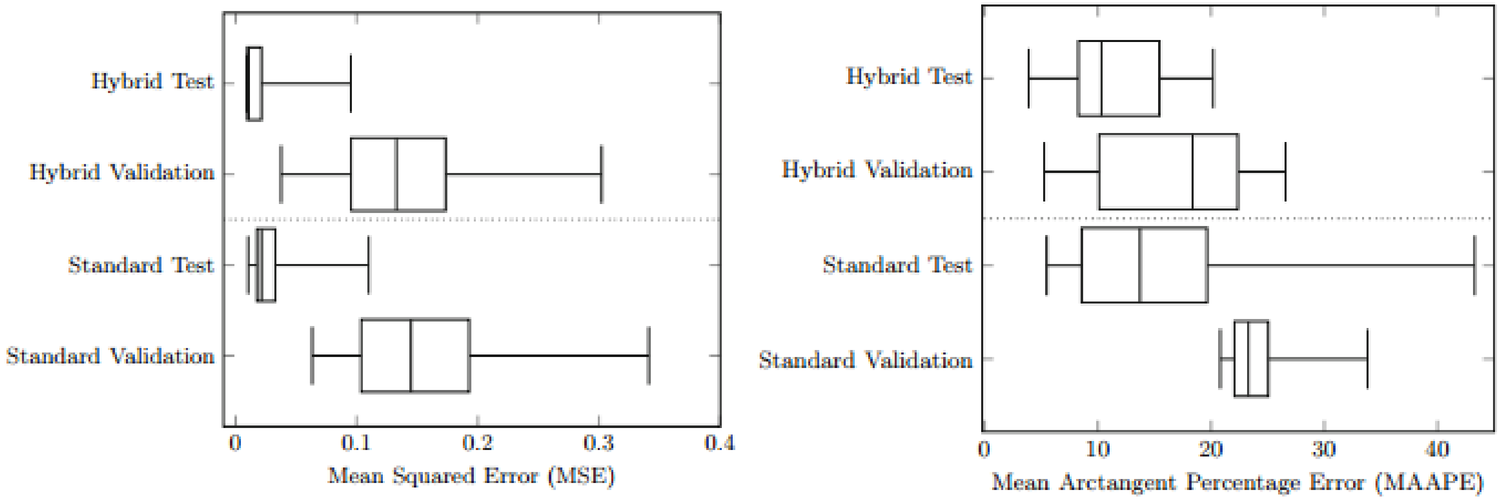
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$\text{MAAPE} = \frac{100}{N*0.5*\pi}\sum_{i=1}^{N}\left(\arctan\left(\left|\frac{y_i-\hat{y}_i}{\hat{y}_i}\right|\right)\right)$$

- Didn’t justify the advantages of combining camera inferred rainfall with measurements of actual rain gauges
- should compare camera + gauge > gauge only

Results: Standard vs. Hybrid

- Hybrid better than Standard in both Validation and Test by 3% - 4% (MAAPE)
- Test better than validation



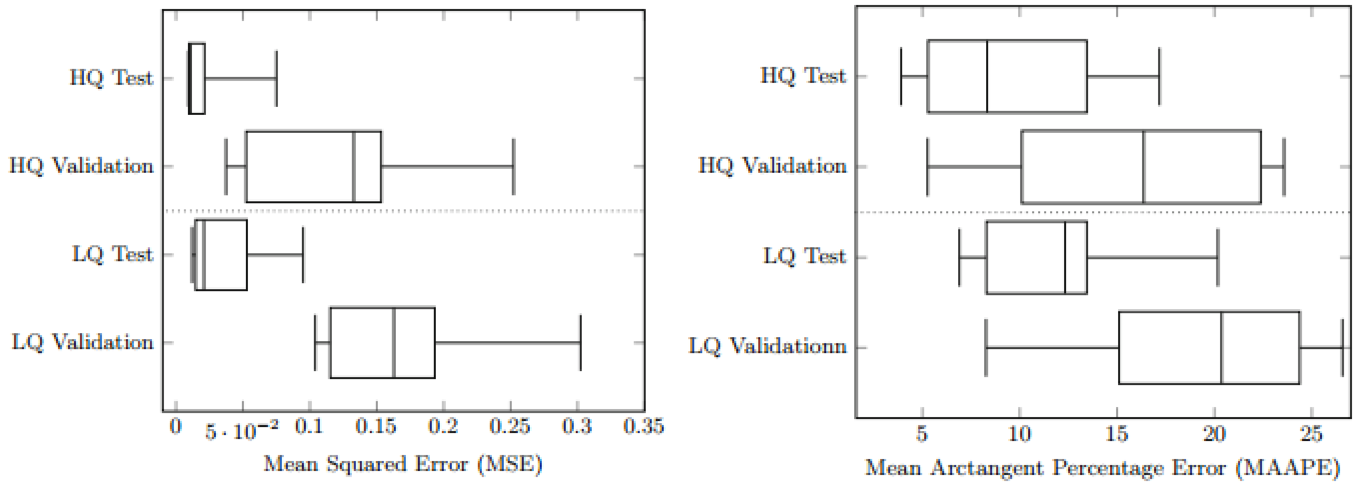
(a) Mean squared error of the validation and test phase.

(b) Mean arctangent percentage error of the validation and test phase.

Fig. 4. Local performance of the standard and hybrid models during the validation and test phase with inverse distance weighting as the ground truth rainfall amount value. For both metrics, the lower the value the better.

Results: HQ vs. LQ

- High-resolution better than low resolution (4% MAAPE)
- Test better than validation



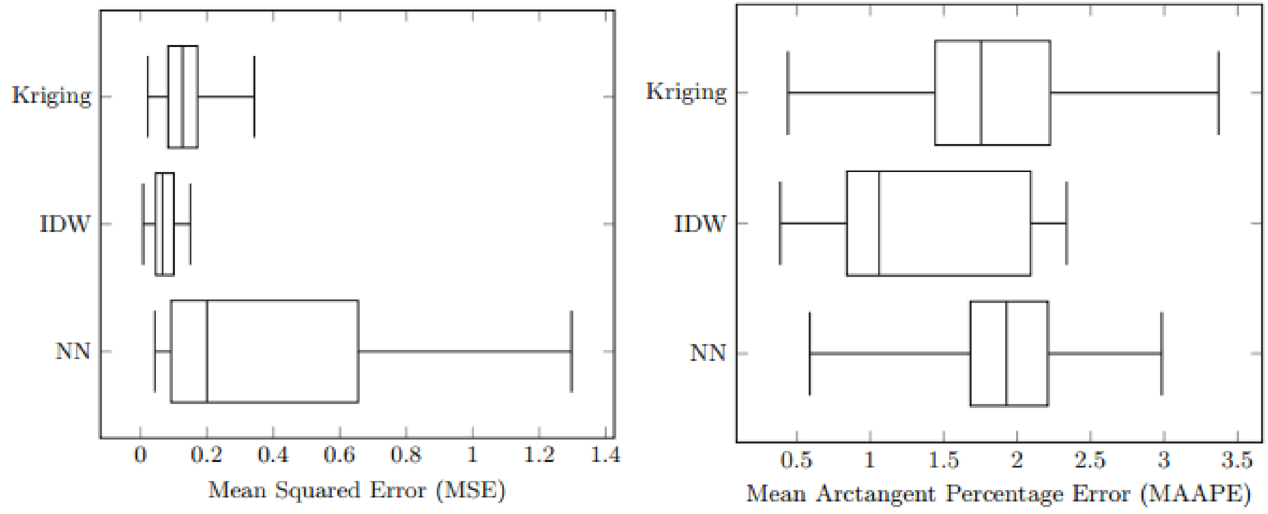
(a) Mean squared error of the validation and test phase.

(b) Mean arctangent percentage error of the validation and test phase.

Fig. 5. Local performance of the hybrid model for training the during the validation and test phase with high-resolution (HQ) and a low-resolution (LQ) camera, respectively. For both metrics, the lower the value the better.

Results: Testing of Global Rain Function

- IDW yields the better results than Kriging (unexpected, probably because specific climate and elevation)
- NN yields the worst results



(a) Mean squared error of testing with the rain gauges.

(b) Mean arctangent percentage error of testing with the rain gauges.

Fig. 6. Global performance of the hybrid model with the three different spatial interpolation methods nearest neighbour (NN), inverse distance weighting (IDW) and Kriging.

Conclusions

- Proposed a method for estimating the rainfall from closed-circuit TV camera images
- Local: Hybrid model outperforms Standard model by 3-4%
- Global: IDW > Kriging > NN
- Error as low as nearly 1%
  - 1% is averaged over 8 months (w/ only 3.31% rainy days), not representative
  - didn't show model performance on rain detection
- Noises in rain streak image, e.g. road marking, car lights still eminent.
  - rain-remover model trained on artificial data, lost the perception of depth
  - model could be improved with better rain streaks extraction algorithms
- Using video data rather than single images

Back Up

Conclusions

- Rain detection based on rain streaks
  - Analytical methods (intensity change) + filtering may not be accurate
  - rain-remover are based on single rain image
  - ➢ ML-assisted rain streaks extraction for video data
    - ➢ compare streak vs. whole pictures
    - ➢ compare analytical vs. ML assisted vs. rain-remover
  - ➢ try how rain-remover works on video data
- Rain estimation based on rain streaks
  - use CNN to learn the camera-specific characteristics, e.g. depth of field, angle, resolution, etc.