2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)

Video-Based Windshield Rain Detection and Wiper Control Using Holistic-View Deep Learning

Chi-Cheng Lai and Chih-Hung G. Li*, *Member, IEEE*

3256 IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 70, NO. 4, APRIL 2021

Real-Time Rain Detection and Wiper Control Employing Embedded Deep Learning

Chih-Hung G. Li, *Member, IEEE*, Kuei-Wen Chen, Chi-Cheng Lai, and Yu-Tang Hwang

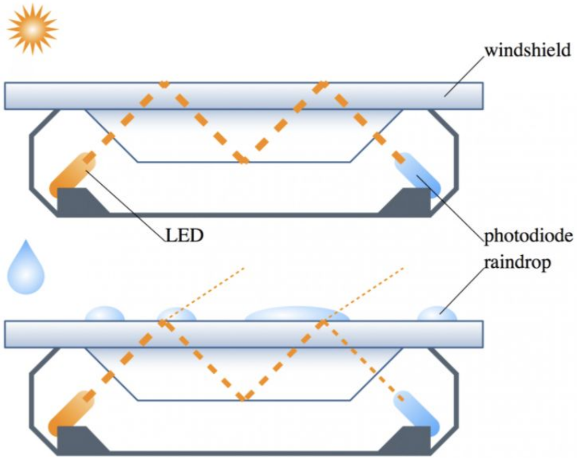
Tian Liu
Sep 13, 2021

Outline

- Review of Windshield Rain Detection
- Deep Learning Framework
- Embedded Systems
  - Personal Computer
  - Raspberry Pi 4
  - Jetson Nano
  - Neural Compute Stick 2
- Speed and Accuracy Comparison
- Conclusions

Windshield Rain Detection

- Current rain sensors
  - Infrared emitters and receivers,
  - Capacity change, rain intensity by pressure, electrical conductivity
  - Only sample a small region of windshield
  - Detect humidity instead of visibility



Windshield Rain Detection

- Computer vision methods
  - Focus on detecting and counting raindrops
  - Cannot capture other water forms, like streak and downpour
- Proposed Deep Learning Approach
  - Deep ConvNet
  - Holistic-view
  - Robust to various background, illumination, water forms
  - State-of-art computing speed and detection accuracy
  - Embedded system achieves good balance between accuracy and speed, operable for mobile applications

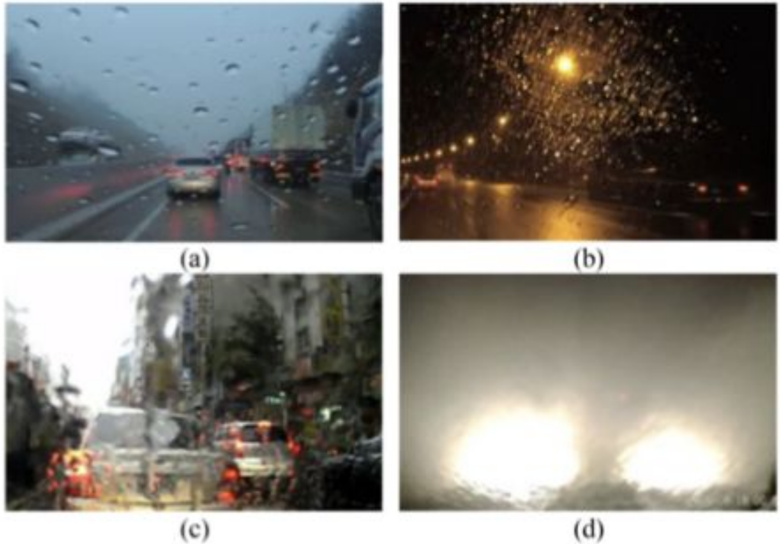


Fig. 1. Dashcam images showing the rain effects on the windshield: (a) day raindrops, (b) night raindrops, (c) water streaks; (d) downpours.

Deep Learning Framework

- Pictures input from dashcam
- ConvNet output recommendations rather than rain conditions
- Temporal filtering to suppress outliers
- Command sent to wiper and repeat

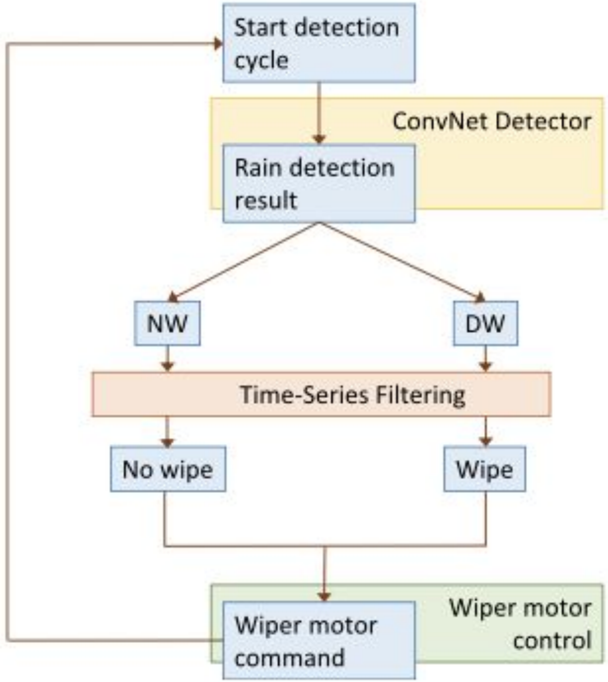


Fig. 2. Proposed windshield wiper control scheme. NW and DW are ConvNet output scores recommending "not to wipe" and "detected to wipe", respectively.

Data and Model

- 44 hrs videos from website
- Sliced into images at 1 FPS
- 160k images, fair vs. rain equal
- Day vs. night 9:7
- 3 experienced drivers to manually annotate

TABLE I
CONTENTS OF THE TRAINING AND TESTING IMAGE DATA

		No-rain	Downpour	Streak	Raindrop
Train	Day	42,815	2,160	630	40,025
	Night	35,472	1,580	460	33,432

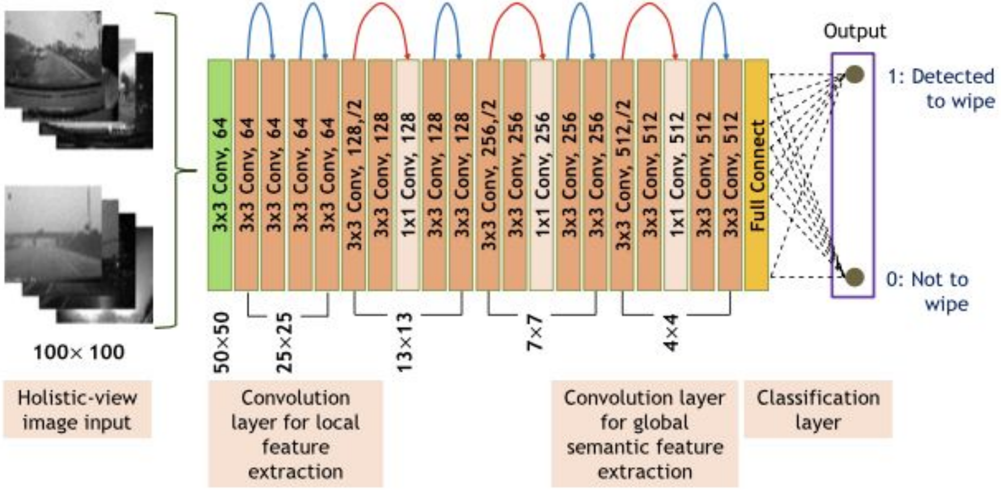


Fig. 3. Architecture of the 18-layer ResNet for visual rain detection.

Objective function and Time-series Filtering

- Minimize difference between recommendation and drivers' will
- Activate wiper when s exceeds threshold
  - n = 4
  - t = 2 or 3

$$DW^{*} = \underset{dw}{\arg\min} \| p(DW|X) - p(DW'|V) \|_2 .$$

$$\mathrm{S}\ (n) = \sum_{i-(n-1)}^{i} D(j) \geq \tau\ ,$$

$$\mathrm{D}\ (j) = 1 \text{ for DW; D } (j) = 0 \text{ for NW},$$

Embedded Systems

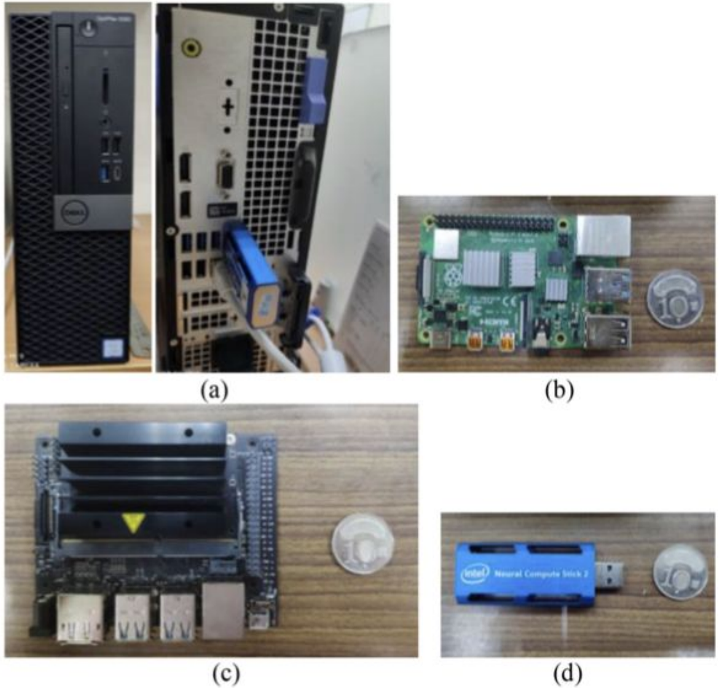


Fig. 4. Photos of the computing systems adopted in this study: (a) personal computer, (b) Raspberry Pi 4, (c) Jetson Nano; (d) Neural Compute Stick 2.

TABLE IV
SPECIFICATIONS OF THE EMBEDDED SYSTEMS AND THE PC IN THE STUDY

Item	Price (USD)	Dimensions (cm×cm×cm)	Mass (g)	Power (W)
PC	830	29×9×29	5260	260
PC+NCS2	912	35×9×29	5310	260
Pi4	54	4.5×8.5×2.5	46	2.1~6.4
Pi4+NCS2	136	4.5 × 14 ×2.5	96	~6.4
Jetson Nano	126	10×8×3	130	5~10
Cluster(4 Pi4+NCS2)	584	19×18×9	1450	~28

Fig. 8. Photos of the embedded computing cluster constructed with 6 sets o Pi4+NCS2.

Neural Compute Sticks

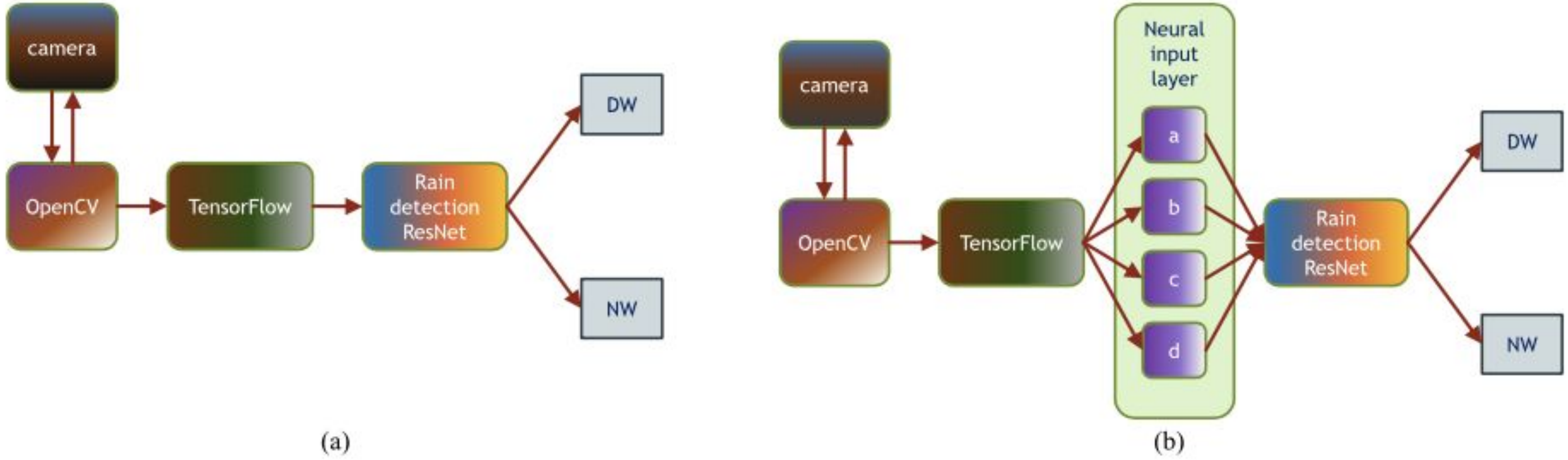


Fig. 5. Frameworks of the rain detection ResNet with and without NCS2: (a) the framework without NCS2; (b) the framework with NCS2.

Cluster Structure

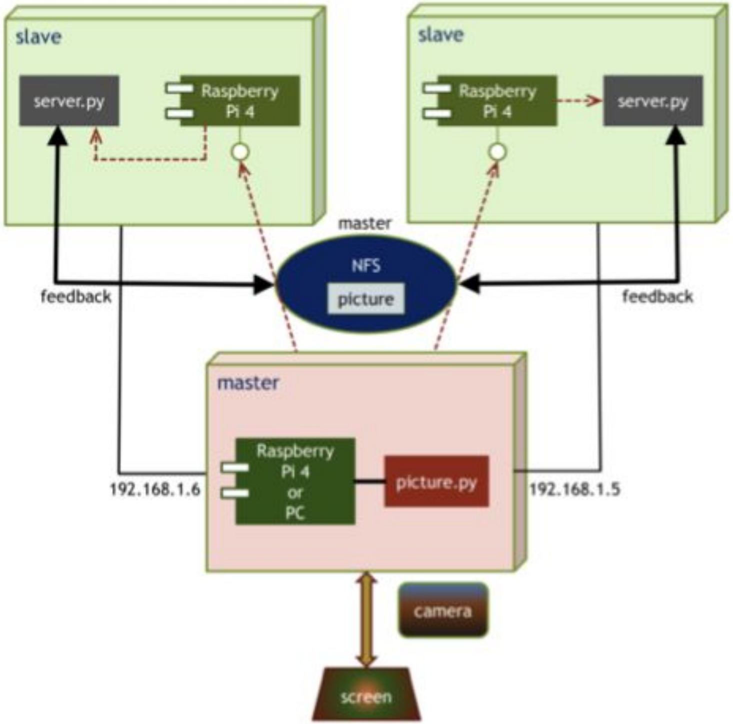


Fig. 6. Illustration of the structure of the computing cluster.

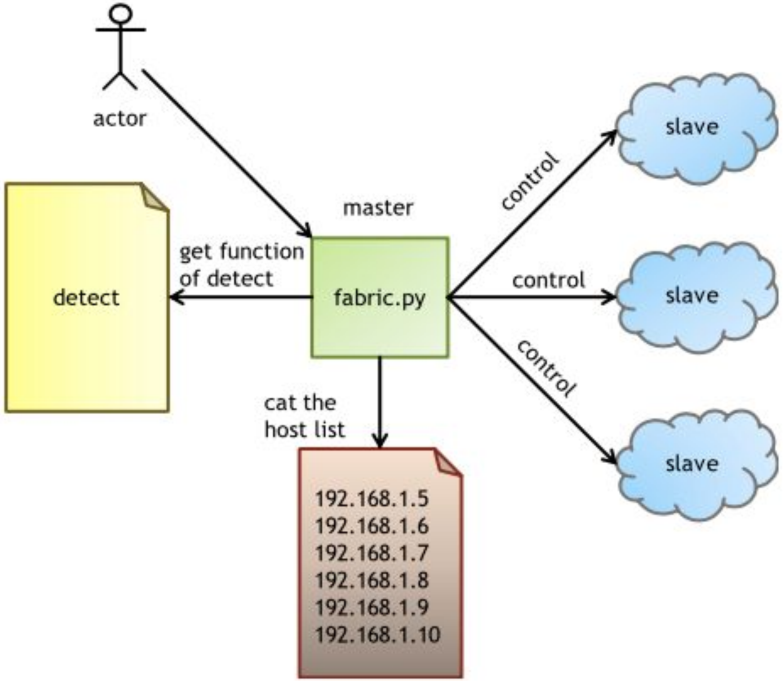


Fig. 7. Illustration of the Fabric structure for managing the data distribution of the nodes.

Real-time Rain Detection Test

Fig. 9. Experiment setup of real-time visual rain detection.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn},$$

$$\text{Precision} = \frac{tp}{tp + fp},$$

$$\text{Recall} = \frac{tp}{tp + fn},$$

Processing Speed

Static scenes

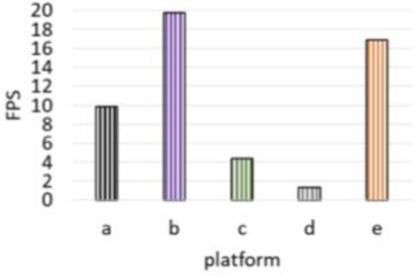
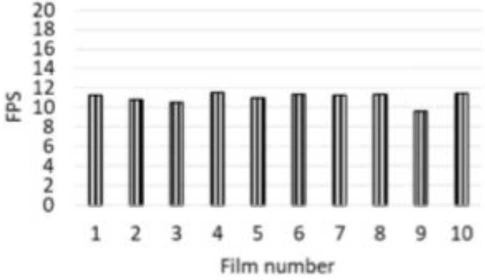
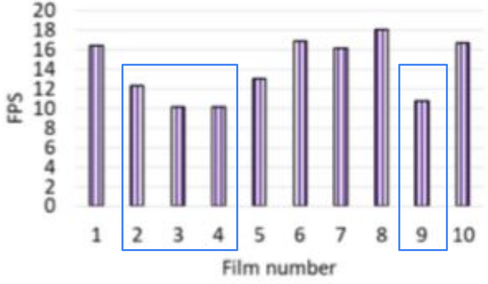


Fig. 10. Average FPS results of the five computation platforms on steadfast images: a. VM, b. VM+NCS2, c. Jetson Nano, d. Pi4; e. Pi4+NCS2.

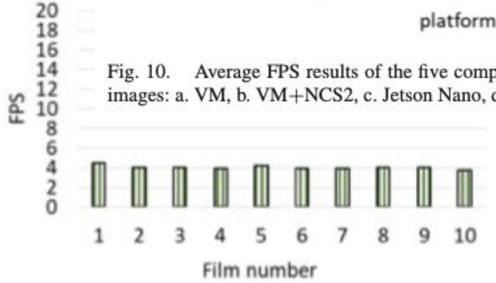
Darker images



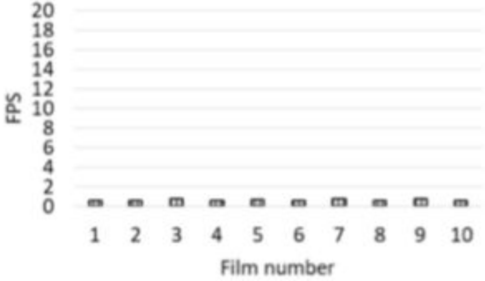
(a)



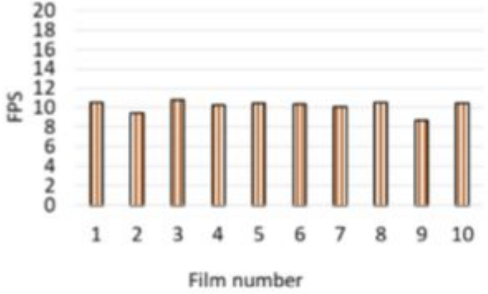
(b)



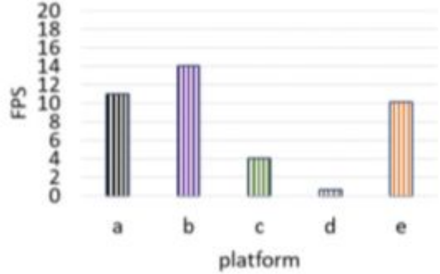
(c)



(d)



(e)



(f)

Dynamic scenes

Fig. 11. FPS results of ten dashcam videos by five different computation platforms: (a) VM, (b) VM+NCS2, (c) Jetson Nano, (d) Pi4, (e) Pi4+NCS2; (f) averages.

Processing Speed

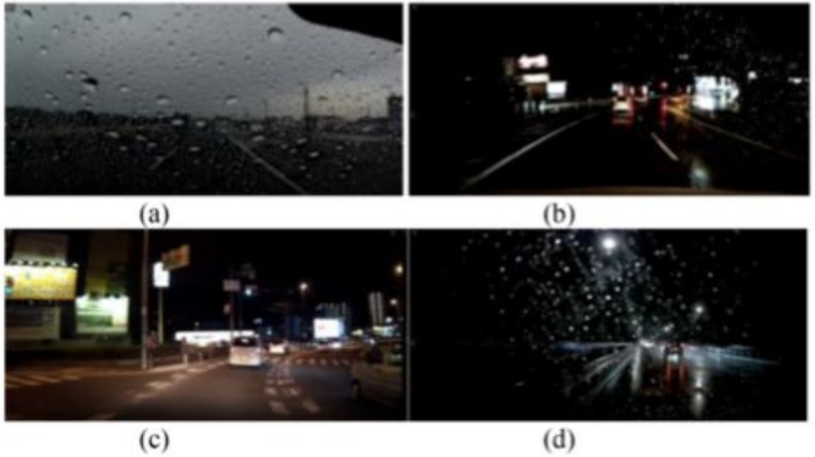


Fig. 12. Sample images of the test cases. Case 3, 4, and 9 are nighttime videos; Case 2 is daytime but very dark. All of them resulted in a relatively low FPS on VM+NCS2. (a) Case 2. (b) Case 3. (c) Case 4. (d) Case 9.

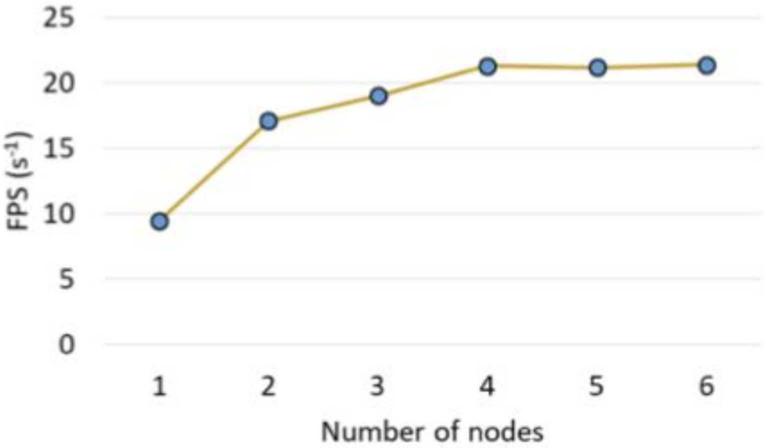


Fig. 13. FPS results of the computing clusters constructed with one to six sets of Pi4+NCS2.

Cluster surpassed PC

Accuracy

Variations reflects the difficulty with each file

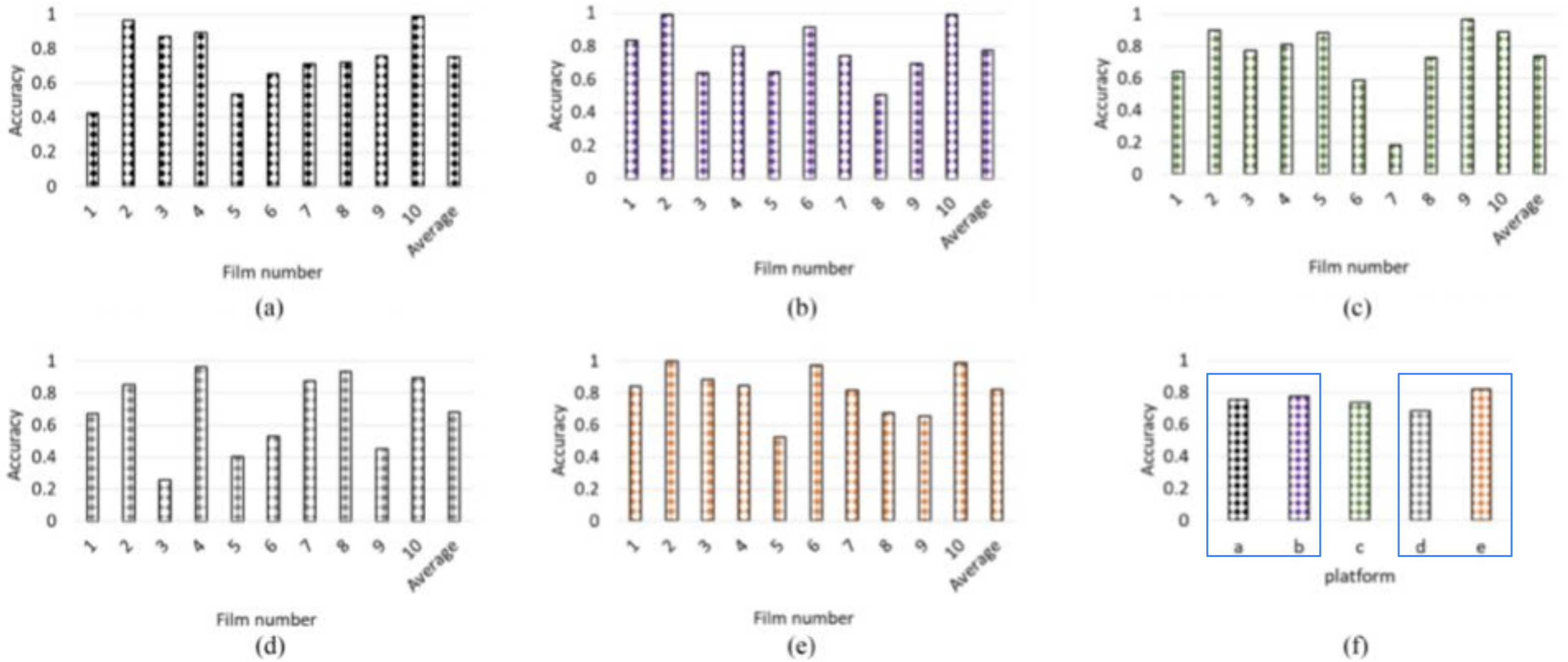


Fig. 14. Accuracy results of ten dashcam videos by five different computation platforms: (a) VM, (b) VM+NCS2, (c) Jetson Nano, (d) Pi4, (e) Pi4+NCS2; (f) average of each platform.

NSC not only increases speed, but also accuracy

Precision and Recall

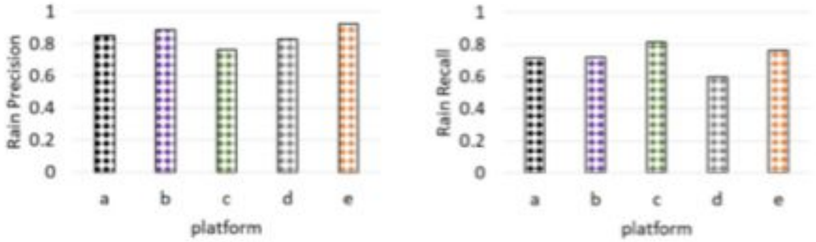


Fig. 15. Average precision and recall of the tested computation platforms: a. VM, b. VM+NCS2, c. Jetson Nano, d. Pi4; e. Pi4+NCS2.

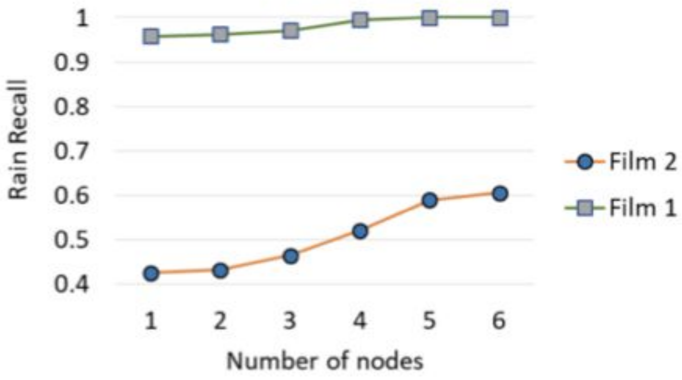


Fig. 16. Recall results of the computing cluster made of various numbers of Pi4+NCS2: film 1 contains high-recall rain scenes; film 2 contains low-recall rain scenes.

# of nodes affects both speed and accuracy

TABLE III
COMPARISON OF THE PRECISION AND RECALL RATES BETWEEN THE STATE-OF-THE-ARTS AND OUR RESULTS

	Kurihata1 [10] (sky only)	Kurihata2 [10] (all background)	Kurihata3 [10] (time-series)	Roser & Geiger [11] (raindrops)	Lai & Li [4] PC	Our results VM+NCS2	Our results Pi4+NCS2 (single set)
Precision	0.97	0.54	0.97	0.80	0.87	0.89	0.92
Recall	0.59	0.59	0.51	0.67	0.82	0.72	0.76

Normalized FPS and Accuracy

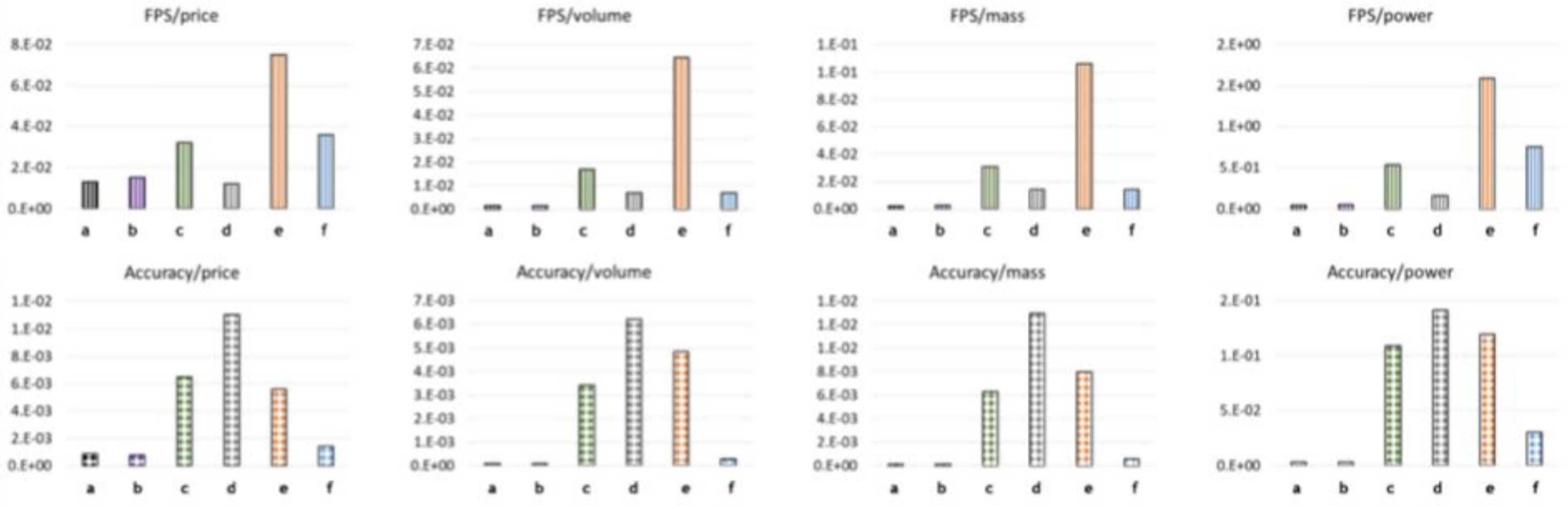


Fig. 17. Normalized FPS and accuracy where each pillar represents the result of a system denoted by a letter of the alphabet: a. VM, b. VM+NCS2, c. Jetson Nano, d. Pi4, e. Pi4+NCS2, and f. computing cluster with four sets of Pi4+NCS2.

FPS of Pi 4 is too low

Sample of Detection Results

Fig. 18. Samples of the detection results. The proposed visual detector is capable of detecting various rainy and fair conditions. (a) DW samples. (b) NW sample.

Conclusions

- Proposed a global-feature visual approach for windshield rain detection
- Deep learning framework detects various types of rain conditions and output recommendations
- Achieved state-of-art accuracy
- Tested on various embedded platforms
  - NSC2 improves both speed and accuracy
  - Pi 4 + NSC2 achieves 10 FPS, comparable to PC of 11 FPS, also has highest normalized FPS
  - Single Pi 4 has highest normalized accuracy but FPS is only 1
  - Cluster of 6-set Pi 4 + NSC2 achieved highest FPS of 20, but has worse normalized FPS and accuracy