工業技術研究院
Industrial Technology
Research Institute

EDGE-COMPUTING CNN
WITH
HOMOGRAPHY-AUGMENTED DATA
FOR
FACIAL EMOTION RECOGNITION

Authors: Yan-Ying Li, Hsin-Jung Cheng, Yen Liu, and Chih-Tsung Shen
Reporter: Chih-Tsung Shen, Ph.D.
Date: 2019.9.25

工業技術研究院
Industrial Technology
Research Institute

The Need of Our Solution

- Edge Computing → **Tiny Model**
  - AI without cloud
  - Rapid response
    - Industrial Inspection
    - Sleep detection for bus drivers

- Facial Emotion Recognition → **High Accuracy**
  - To understand the customers in the shops
  - To tell the student's reaction in the classrooms
  - Business Intelligence, and so on…

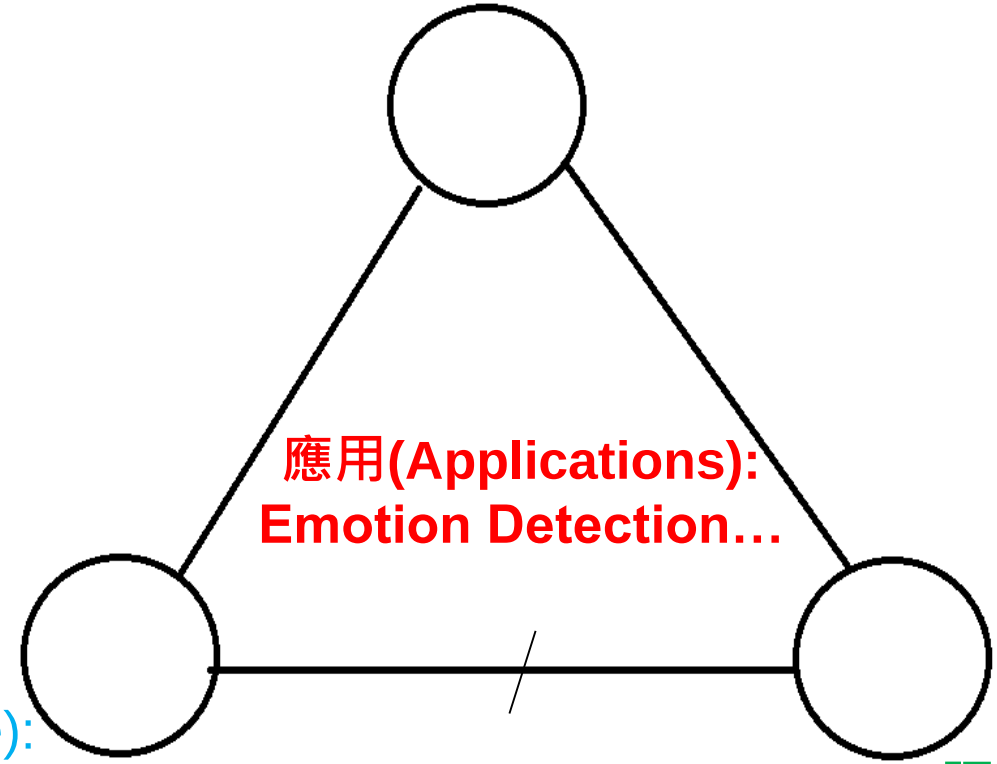
工業技術研究院
Industrial Technology
Research Institute

Our Solution using Intel VPU

工業技術研究院
Industrial Technology
Research Institute

Our Solution

資料(Data):
FER2013, JAFFE, …



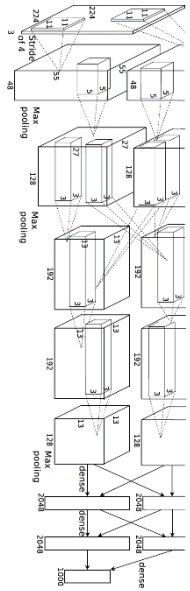
架構(Architecture):
AsicNet,
AlexNet,
GoogLeNet,
MobileNet, …

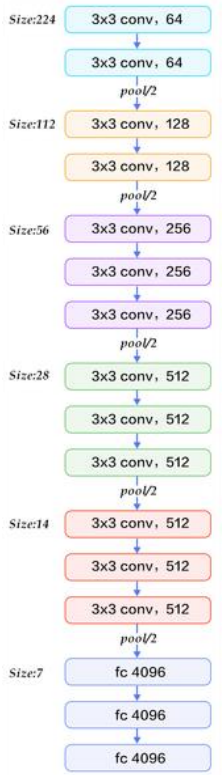
運算平台/開發套件
(Platform/Toolkit):
Google TensorFlow,
Intel OpenVINO, …

硬體(Hardware):
Nvidia GTX 1080Ti,
Intel CPU Apollo Lake,
Intel VPU, …

工業技術研究院
Industrial Technology
Research Institute

Our AsicNet





Type	Level
2×Conv	Low-Level Feature Extraction
MaxPool	
2×Conv	
MaxPool	
2×Conv	Middle-Level Feature Extraction
AvgPool	
2×Conv	High-Level Feature Extraction
AvgPool	
Flatten	Multi-Layer Perceptron
Dense	
Dense	Classifier

AlexNet GoogLeNet VGGNet **Our AsicNet**

Model sizes are too big for embedded IoT devices! Model size=30MB (→10MB)

Our AsicNet

- The CNN Architecture of our AsicNet
- Input Size = 64x64

Type	Shape	Output	Layer
2× Conv	3 × 3 × 32	64 × 64 × 32	Low-Level
MaxPool	2 × 2	32 × 32 × 32	Convolution
2× Conv	3 × 3 × 64	32 × 32 × 64	Low-Level
MaxPool	2 × 2	16 × 16 × 64	Convolution
2× Conv	3 × 3 × 128	16 × 16 × 128	Middle-Level
AvgPool	2 × 2	8 × 8 × 128	Convolution
2× Conv	3 × 3 × 128	8 × 8 × 128	High-Level
AvgPool	2 × 2	4 × 4 × 128	Convolution
Flatten	2048	2048	Fully-Connected
Dense	1024	1024	Fully-Connected
Dense	7	7	Classifier

Face Emotion Data in use

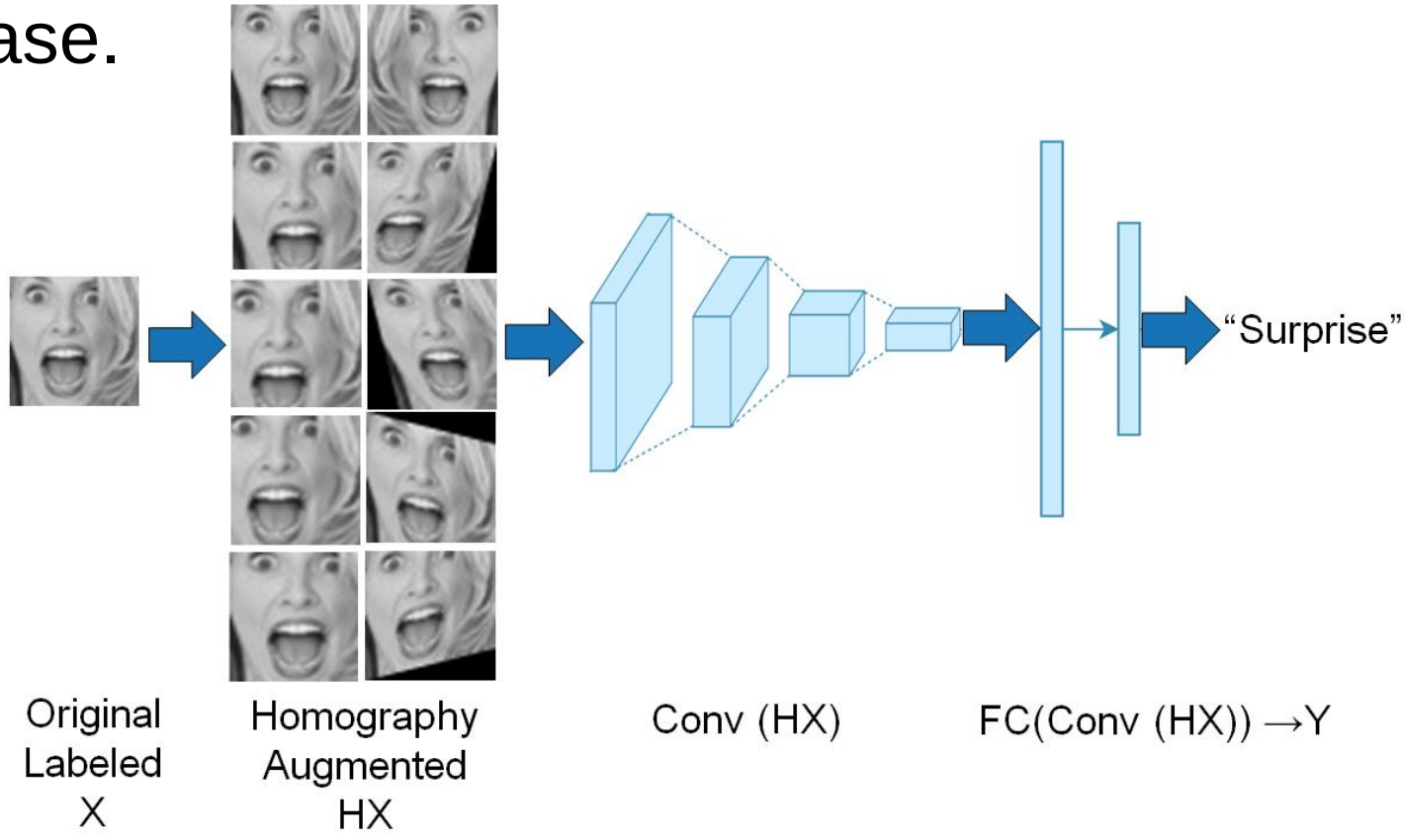
- 7 Classes: happy, sad, surprise, fear, anger, disgust, and neutral
  - FER2013 Dataset

  - JAFFE Dataset

- CK+ is too noisy. We don't use this dataset.

Data Augmentation

- The method to augment data:
  - Brightness, Histogram-Equalization, Blur, Crop, …
  - To absorb the camera distortion during the training phase.

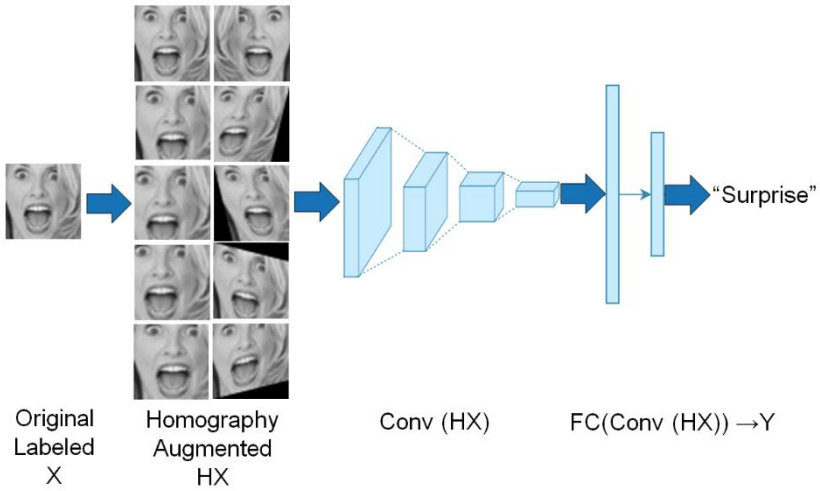


How to optimize a CNN?

- The Steps of our Training Phase
  - Add or Mix Data
  - Augment Data
  - Adjust the Numbers of Convolutional Layers [3x3]
  - Adjust the Numbers of Neurons of FC Layers
  - Validation
- The Flow:

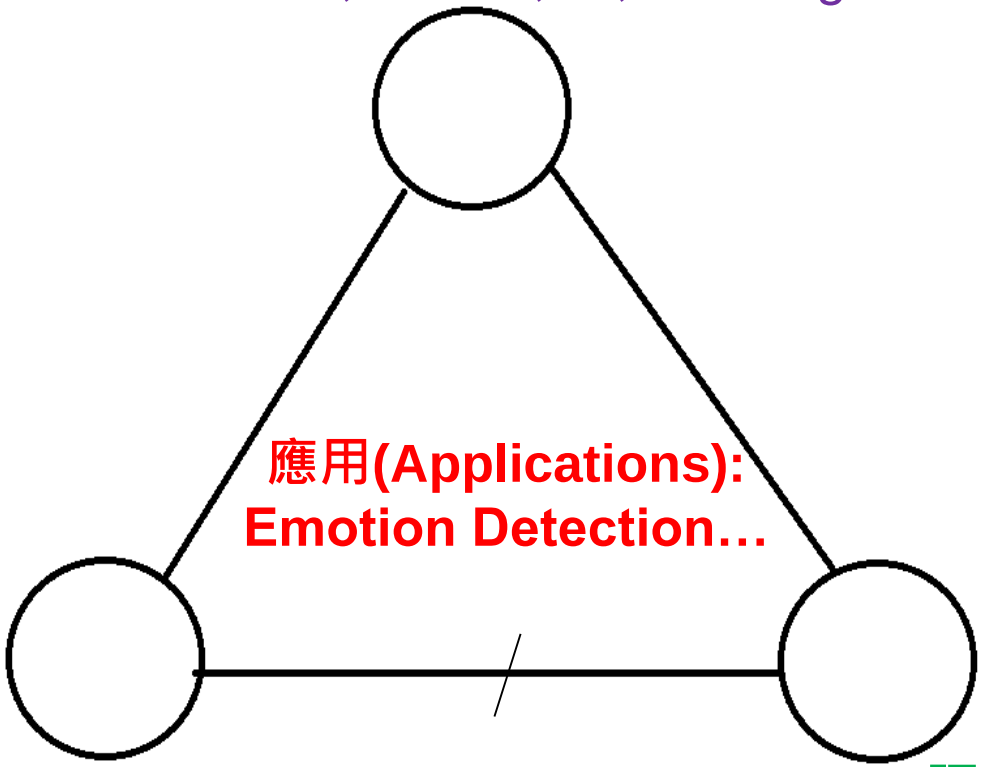
$$y \leftarrow FC_k(Conv_j(H_i x))$$

  - This version, j = 4, k = 2.
- For ASICs, no short-cut.



Our Solution

資料(Data):
FER2013, JAFFE, …, with Augmentation, ...



架構(Architecture):
AsicNet

運算平台/開發套件
(Platform/Toolkit):
Google TensorFlow,
Intel OpenVINO, …

硬體(Hardware):
Nvidia GTX 1080Ti,
Intel CPU Apollo Lake,
Intel VPU, …

AsicNet on (original) FER2013

- In Experiment 1, we compare our AsicNet with the state-of-the arts on FER2013.

Dataset	FER2013									
Method	AlexNet		GoogLeNet		MobileNet		Ours			
Data Augmentation							No	No	Yes	Yes
Batch Size	32	64	32	64	32	64	32	64	32	64
Model Size	529MB	529MB	77.9MB	77.9MB	26.5MB	26.5MB	30.9MB	30.9MB	30.9MB	30.9MB
Training Accuracy	25.13%	85.11%	99.37%	99.72%	99.72%	99.79%	98.7%	99.16%	96.43%	97.23%
Inference Accuracy	24.49%	53.44%	67.48%	67.09%	64.25%	63.66%	68.65%	68.82%	**71.72%**	71.16%
Inference Time on GPU	4.18ms	4.18ms	19.50ms	19.50ms	16.40ms	16.40ms	3.06ms	3.06ms	**3.06ms**	3.06ms
Inference Time on VPU							15.25ms	15.25ms	**15.25ms**	15.25ms

GPU scheme: Nvidia GTX 1080Ti + Google TensorFlow
VPU scheme: Intel Movidius Myriad X VPU + Intel OpenVINO

AsicNet on (original) JAFFE

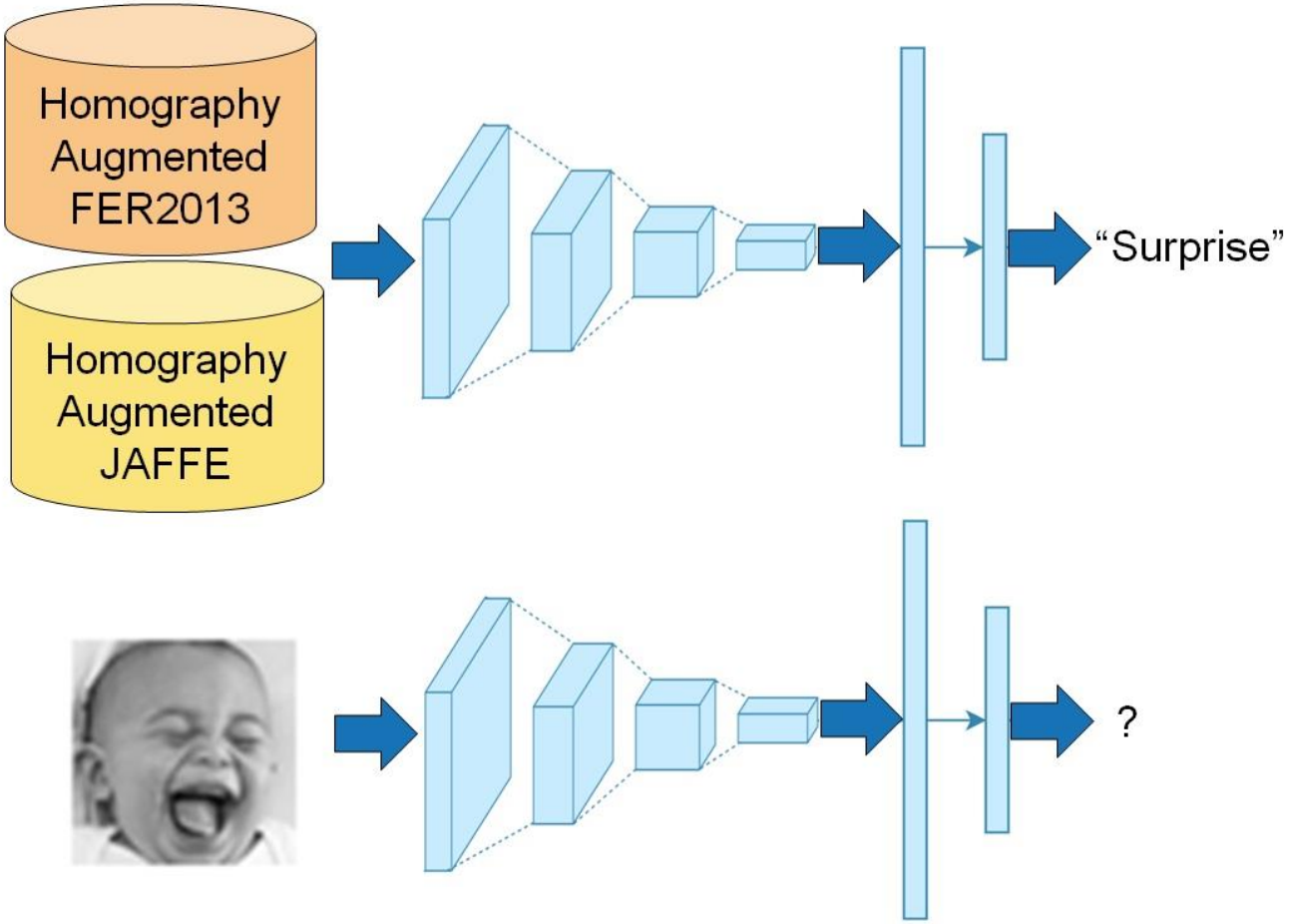
- In Experiment 2, we compare our AsicNet with the state-of-the arts on JAFFE.

Dataset	JAFFE						
Method	Liu[19]	Mlakar[20]	Mayya[21]	Ours			
Data Augmentation				No	No	**Yes**	Yes
Batch Size				32	64	**32**	64
Inference Accuracy	91.80%	87.82%	98.12%	92.97%	91.19%	**99.82%**	99.65%
Inference Time on GPU			140ms	3.06ms	3.06ms	**3.06ms**	3.06ms
Inference Time on CPU	210ms	65ms	800ms	11.98ms	11.98ms	**11.98ms**	11.98ms
Inference Time on Embedded CPU				41.22ms	41.22ms	**41.22ms**	41.22ms
Inference Time on VPU				15.25ms	15.25ms	**15.25ms**	15.25ms

GPU scheme: Nvidia GTX 1080Ti + Google TensorFlow
CPU scheme: Intel i7 + Google TensorFlow
Embedded CPU scheme: Intel Apollo Lake SOC + Intel OpenVINO
VPU scheme: Intel Movidius Myriad X VPU + Intel OpenVINO

Data Argumentation

- In Experiment 3, we mix the datasets.



AsicNet on FER2013/JAFFE

- In Experiment 3, we improve the accuracy.

Ours			
No	No	Yes	Yes
32	64	32	64
30.9MB	30.9MB	30.9MB	30.9MB
98.7%	99.16%	96.43%	97.23%
68.65%	68.82%	**71.72%**	71.16%
3.06ms	3.06ms	**3.06ms**	3.06ms
15.25ms	15.25ms	**15.25ms**	15.25ms

Trained by FER2013 dataset.
Testing on FER2013 dataset.

Dataset	FER2013			
Method	Ours			
Data Augmentation				
JAFFE	Yes	Yes	Yes	Yes
FER2013 (Affine)	No	No	Yes	Yes
Batch Size	32	64	**32**	64
Inference Accuracy	72.03%	71.24%	**72.42%**	71.72%

Trained by the FER2013/JAFFE mixed datasets.
Testing on FER2013 dataset.

工業技術研究院
Industrial Technology
Research Institute

The Conclusion

- We design our AsicNet to improve FPS:
  - 24.26FPS on Embedded CPU
  - 65.57FPS on VPU
- … and Accuracy:

72.42%

Trained by FER2013/JAFFE mixed dataset.
With Data Augmentation on both FER2013 and JAFFE.

72.03%

Trained by FER2013/JAFFE mixed dataset.
Without Data Augmentation on FER2013.
With Data Augmentation on JAFFE

71.72%

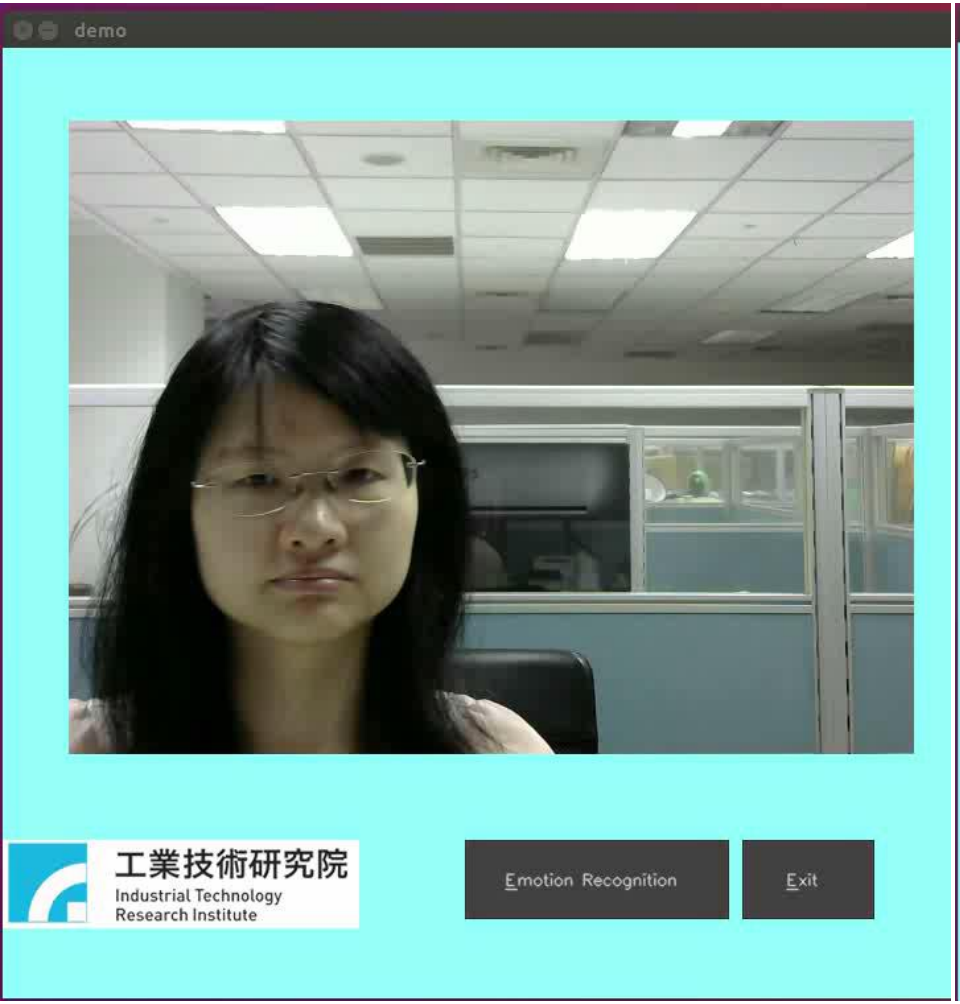
Trained by FER2013 dataset.
With Data Augmentation.
Testing on FER2013 dataset.

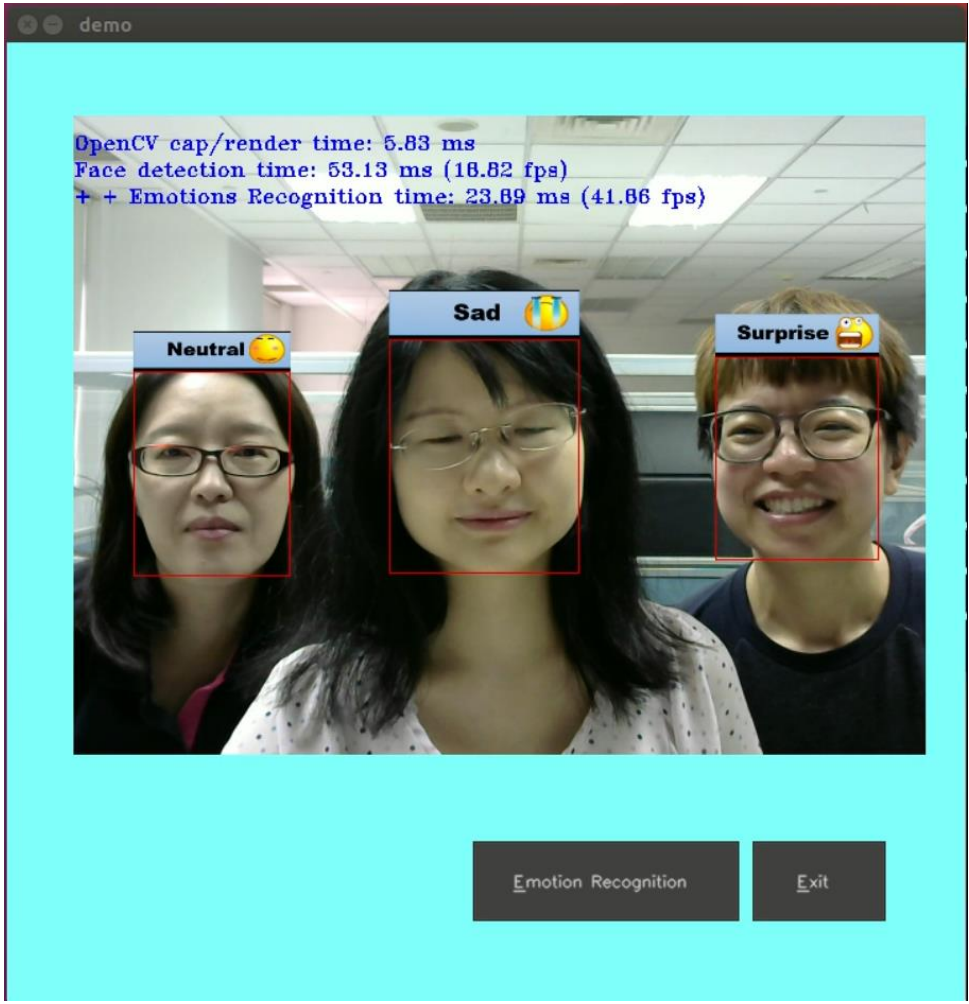
68.65%

Trained by FER2013 dataset.
Without Data Augmentation.
Testing on FER2013 dataset.

工業技術研究院
Industrial Technology
Research Institute

More Demo of Our Solution





工業技術研究院
Industrial Technology
Research Institute

Appendix: