Animal Recognition and Identification with Deep Convolutional Neural Networks for Farm Monitoring

Ryunosuke Murakami, Yuichi Okuyama, Abderazek Ben Abdallah
Graduate School of Computer Science and Engineering,
Adaptive Systems Laboratory
The University of Aizu, Japan

Outline

- Background
- Motivation
- Research goal
- Approach
- Evaluation
- Conclusion

Background

- Due to the increasing demand in the agricultural industry, the need to effectively grow and protect a plant and increase its yield is necessary
- It is important to monitor the plant during its growth period and protect it from animals (pig, wolf, etc.) at the time of harvest

Fig 1. Farm and wild animals

Fig 2. Olive and Date Tree Diseases

Motivation

- Monitoring the plants from plantation to harvesting is necessary for better productivity
- Smart farming needs right decision and monitoring tools for better productivity, quality and profit
- Artificial neural network concept is efficient for image processing

Convolutional Neural Network

- Efficient to object recognition
  - Process data while keeping the shape of image
- Behavior is similar to *visual cortex*
  - Performance is close to human-level
- Learn *feature vector* automatically from data

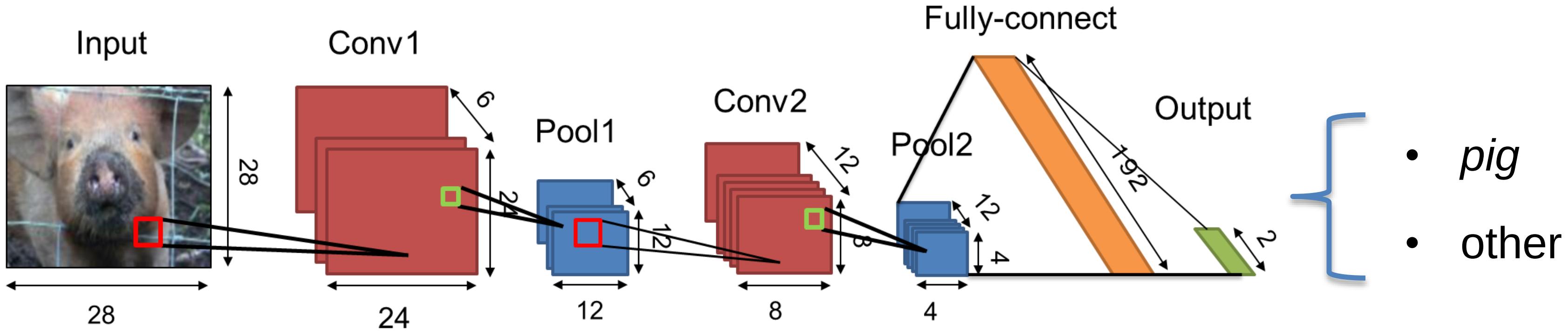


Fig 3. CNN example

Research goal

- Develop an efficient FARM monitoring system for better productivity, quality and profit based on artificial neural network concept:
  - Hardware implementation of Deep Convolutional Neural Network on FPGA
  - Evaluation of real hardware complexity (power and area) and performance (recognition accuracy, time)
- The purpose is to monitor strange animals (pig, etc) and diseases on the stem/leaf/fruits of the crop

System overview

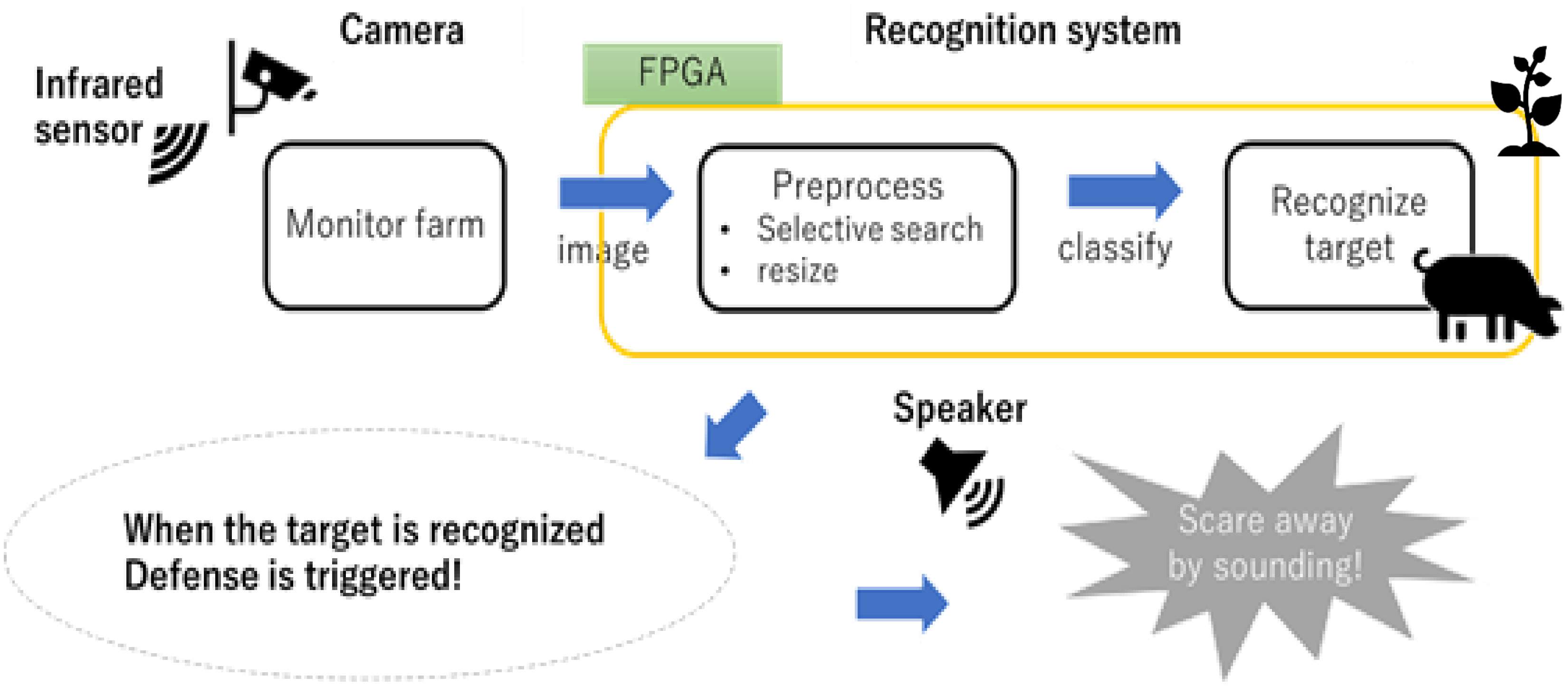


Fig 4. System overview: OASIS FMS-1

Flow of recognition system

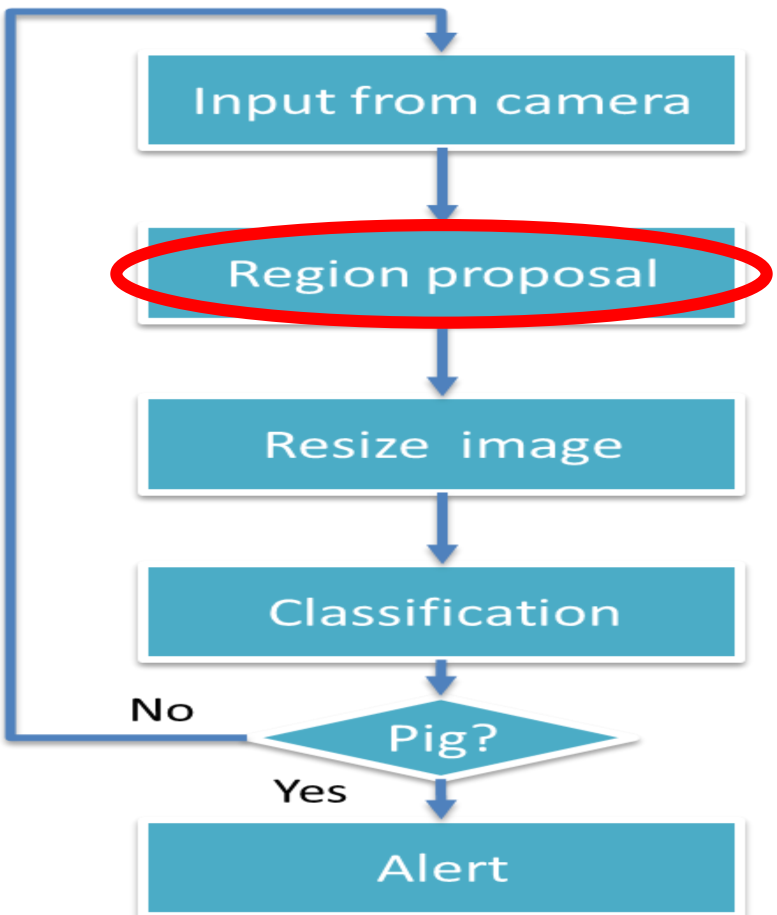
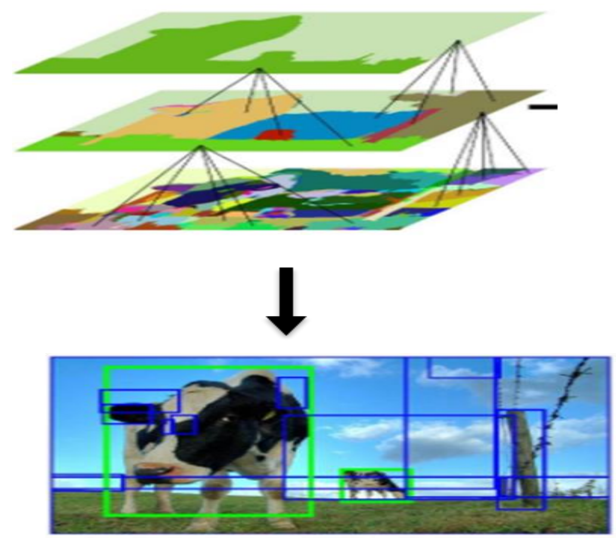


Fig 5. Flow chart of recognizer

- Region proposal
  1. Create initial region with pixel
  2. Group the similar regions
  3. Continue “2” until the whole image becomes a single region



出典:「Rich feature hierarchies for accurate object detection and semantic segmentation」

Approach

1. Software implementation by *Python*
   – Design D-CNN using *Chainer* framework
2. Implement selective-search and integrate
   – Region proposal for object position
3. Hardware implementation on *FPGA* by *HDL*
   – Install parameters already learned

My network structure

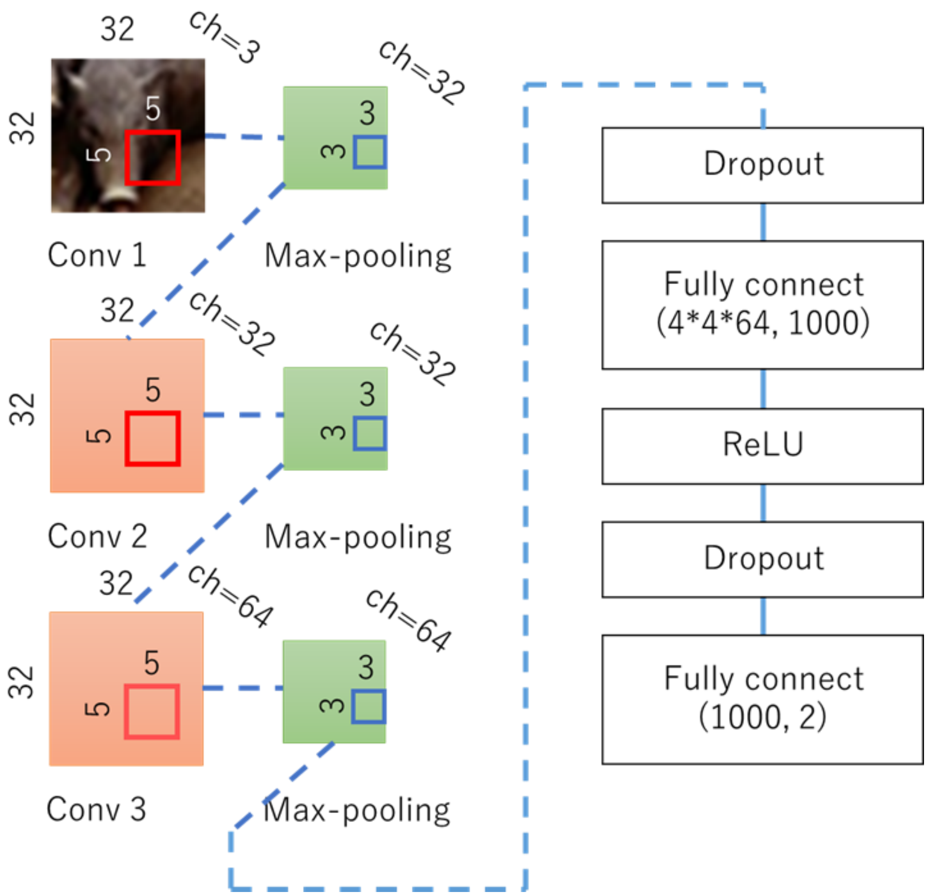


Fig 5. network structure

- Input image
  - Image size: 32x32
  - Channel depth=3 (RGB)
- Hidden size
  - 3 series of conv-layer (conv + Pooling)
- Output layer
  - *ReLU*
  - Dropout
  - Fully connect

Dataset

- Collected from *ImageNet*
- Image: 32x32 pixel, channel = 3(RGB)
- Distribution of original data
  - Pig images: 685
  - Other images: 800

Table 1. Data distribution

Class	Train	Valid	Test
Pig	549	68	68
Other animals	664	68	68

Data augmentation

- Augmented training data twice
  - Original data(Training): 1,213
  - Augmented data: 2,426

[applied image conversion]

1. Slide the pixels randomly within range (-4 ~ 4)
2. Fill in “0” with empty space
3. Flip horizontal randomly

Learning methods

- In the following experiments, 3 learning methods were compared

Table 2. Learning methods

Learning method	Contents
Momentum SGD	Optimized by gradient of loss function and learning rate is decreased with gradient
AdaGrad	Learning rate decreases with scale of weights
Adam	Combined with Momentum and AdaGrad

Evaluation configurations

- Learning parameters
  - Batch size: 128
  - Iterate num: 100
  - Learning decay: 0.1 times in each 20 epoch
- Experiment environment

Table 3. Machine spec

OS	Ubuntu 16.04.3 LTS
CPU	Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
GPU	GeForce GTX 1060
Language	Python (ver: 2.7.13)
Library	Chainer (ver: 2.0.2)

Evaluation result - Best Model

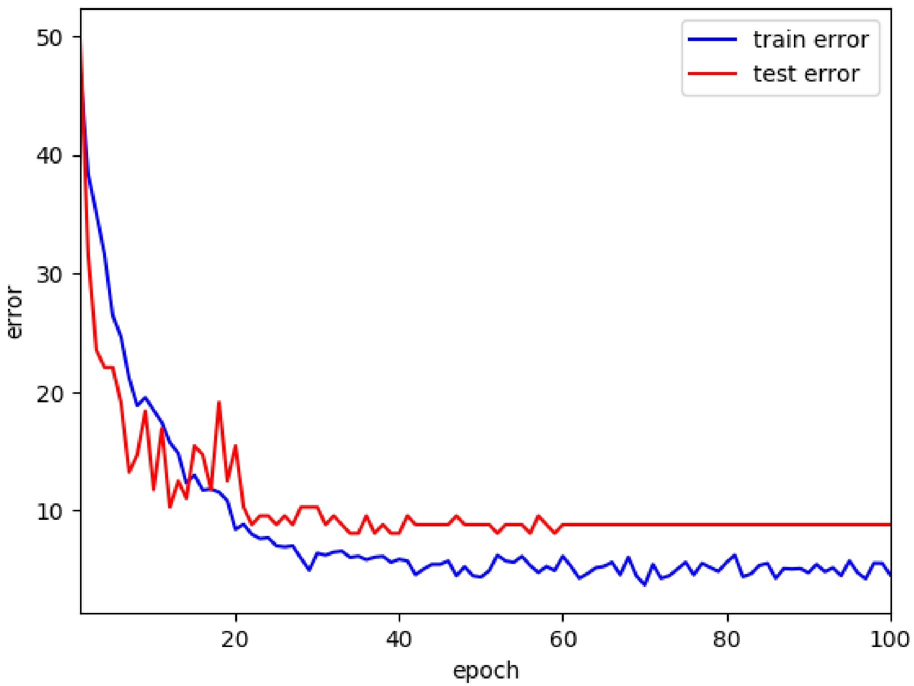


Fig 6. Learning result

[Distribution rate]
Data(1221) = {pig: 549, oth: 664}

[Batch size]
mini_batch(128) = {pig: 64, oth: 64}

[Network structure]
Conv1(filter=16, stride=1, pad=2)
Conv2(filter=32, stride=1, pad=2)
Conv3(filter=32, stride=1, pad=2)
l1=L.Linear(4 * 4 * 32, 1000),
l2=L.Linear(1000, 2)

[Learning parameters]
--optimizer: Adam
--iter 100
--lr_decay_iter 20
-- Actication: ReLU

Best test accuracy: 92.647

Accuracy evaluation with different filter numbers

Table 4. Evaluation with network

conv1	conv2	conv3	accuracy	elapsed time
			error	elapsed time
64	64	64	7.353	0.184
32	64	64	9.559	0.136
16	64	64	10.294	0.117
64	32	64	9.559	0.154
32	32	64	8.824	0.114
16	32	64	11.765	0.104
64	64	32	8.088	0.178
32	64	32	8.088	0.128
16	64	32	9.559	0.109
64	32	32	9.559	0.148
32	32	32	10.294	0.11
16	32	32	7.353	0.1

Base line

Best 2

Best 1

- Conv1,2,3 : each convolutional layer
- Error = 100 * (1 – accuracy)
- Accuracy: ratio of concordance with correct label
- Elapsed time(μs) is calculated to batch data(128 images) classification

Error rate evaluation

Table 5. Error rate evaluation with different learning method

(Time: μs)

Network (filter number)	momentum-SGD(error/time)	Adam (error/time)	adaGrad (error/time)
(16, 32, 32)	7.353/0.1	9.559/0.097	12.5/0.097
(32, 32, 64)	8.824/0.114	9.559/0.115	19.118/0.114
(32, 64, 32)	8.088/0.128	8.823/0.125	13.235/0.128

Table 6. Error rate evaluation with different Activation function

(Time: μs)

network	ReLu	sigmoid	tanh
(16,32,32)	7.353/0.1	11.029/0.099	8.088/0.101
(32,32,64)	8.824/0.114	11.765/0.114	8.824/0.112
(32,64,32)	8.088/0.128	8.824/0.127	9.559/0.127

Conclusions and future work

- This paper presented an Animal Recognition and Identification with Deep Convolutional Neural Networks for Farm Monitoring
- As a first step, the system was designed and evaluated in software
- Evaluation results shows that the system achieves 92.647 accuracy and 0.1µs

Conclusions and future work

- As a future work, we intend to design the system in hardware (Verilog and FPGA) and evaluate its real performance, complexity, and power consumption

Thank you for your kind attention.