Solving the Defect in Application of Compact Abating Probability to Convolutional Neural Network Based Open Set Recognition

2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)

Yunseon Byun
May 04, 2020

Contents

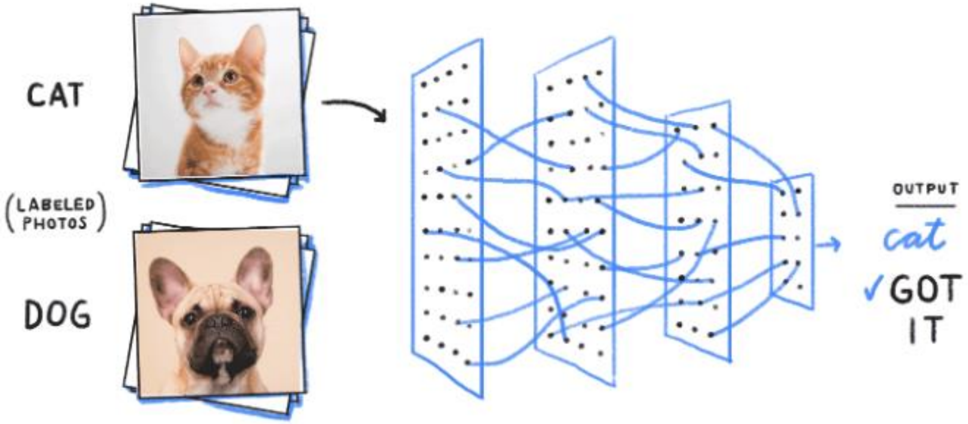
출처:https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8

1. Introduction

What is open set recognition(OSR)?

Classification

✓ 하나의 입력에 대해서 특정 범주(class, label)를 구분하는 것



Closed-set recognition

✓ 이미 알고 있는 객체에 범주(class)를 할당하는 것

✓ 새로운 객체가 생기면, 해당 객체에 대한 재 학습이 필요함

Open-set recognition

✓ 'Closed-set' + 'unknown/unseen'

✓ 새로운 객체가 생겼을 때, 새로운 범주(class)를 할당하는 것

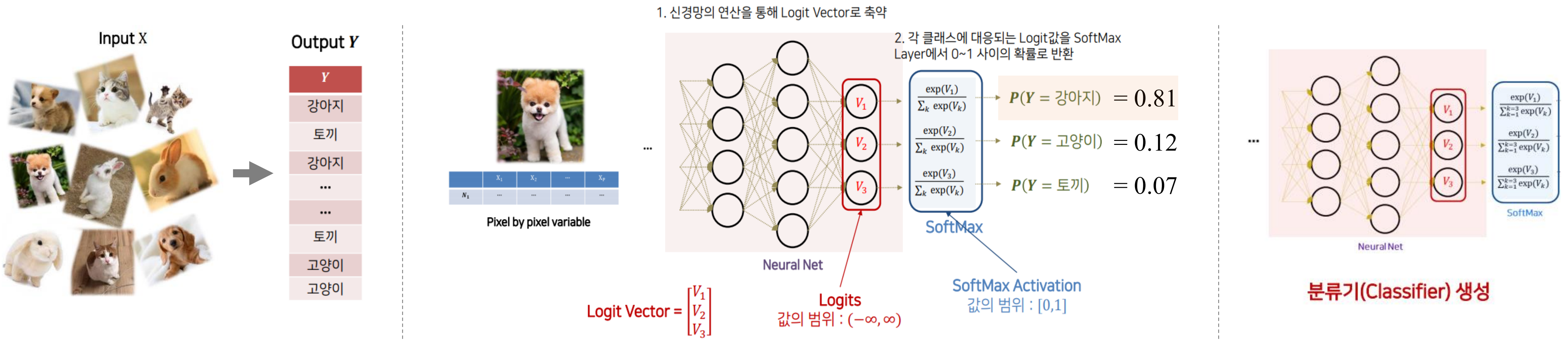
✓ 새로운 객체에 대한 재 학습이 필요하지 않음

출처http://dmqm.korea.ac.kr/activity/seminar/281

1. Introduction

Example

Animal Image Classification (ex. Neural networks)



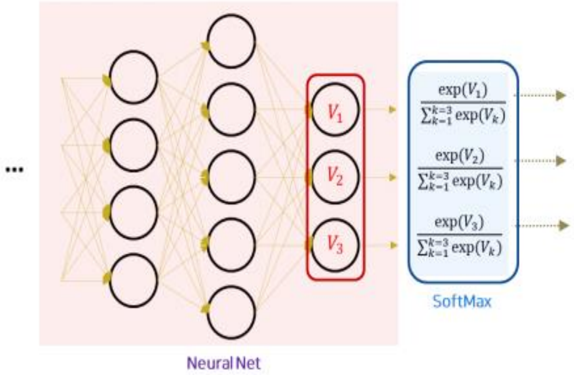
출처http://dmqm.korea.ac.kr/activity/seminar/281

1. Introduction

Example

Animal Image Classification (ex. Neural networks)

너구리..?

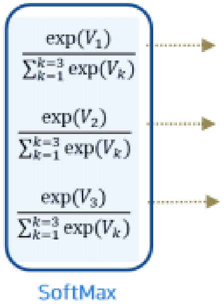


강아지/고양이/토끼
Classifier

기존 분류모델의 한계점

→ 강아지

오분류!



$\boldsymbol{P(Y = \text{강아지}) = 0.93}$

$\boldsymbol{P(Y = \text{고양이}) = 0.02}$

$\boldsymbol{P(Y = \text{토끼}) = 0.05}$

출처:Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 427-436).

1. Introduction

Example

Deep neural networks are easily fooled: High confidence predictions for unrecognizable images

(Nguyen, A., Yosinski, J., & Clune, J. (2015). CVPR)

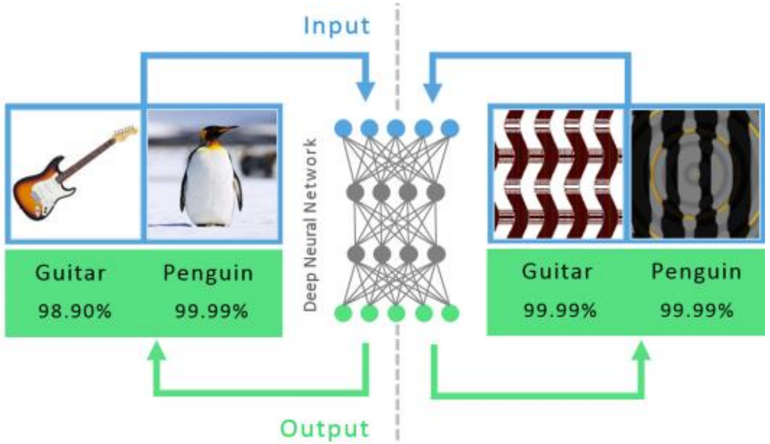


Figure 2. The recognition of state-of-the-art DNNs

- ✓ State-of-the art DNNs can recognize real images with high confidence
- ✓ DNNs are easily fooled: images can be produced that are unrecognizable to humans, but DNNs believe with 99.99% certainty are natural objects

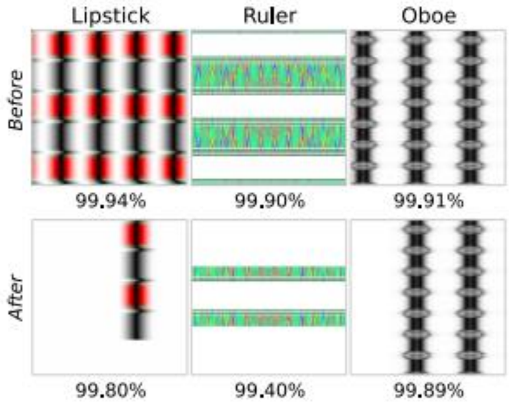


Figure 10. The results after manually removing repeated elements

- ✓ After manually removing repeated elements, the confidence score drops

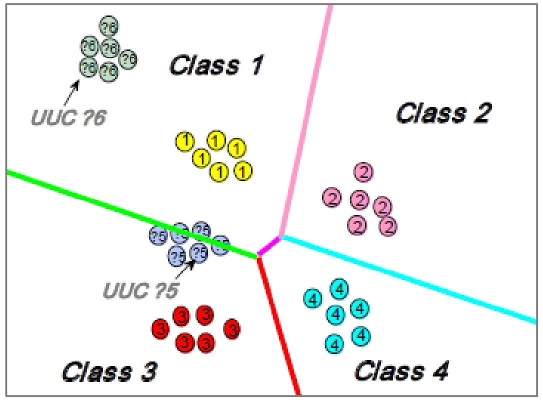
출처:Geng, C., Huang, S. J., & Chen, S. (2020). Recent advances in open set recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

1. Introduction

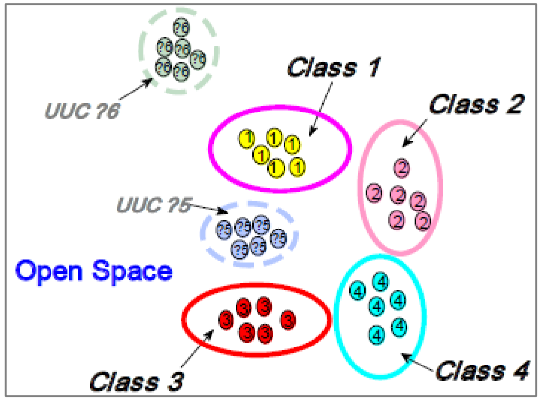
Example

The necessity of open set classification

새로운 범주(class)가 생겼다면, 새로운 범주라고 제대로 분류하자



Traditional recognition/classification



Open set recognition/classification

출처:Geng, C., Huang, S. J., & Chen, S. (2020). Recent advances in open set recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

1. Introduction

Recent advances in OSR

KKCs (known, known)

- ✓ The classes with distinctly labeled positive training samples
- ✓ 알고 있는 data, label

KUCs (known, unknown)

- ✓ Labeled negative samples, not necessarily grouped into meaningful classes
- ✓ 알고 있는 data, 새로운 label

UKCs (unknown, known)

- ✓ Classes with no available samples in training, but available side information of them during training
- ✓ 새로운 data, 알고 있는 label

UUCs (unknown, unknown)

- ✓ Classes without any information regarding them during training, not only unseen but also having not side information during training
- ✓ 새로운 data, label

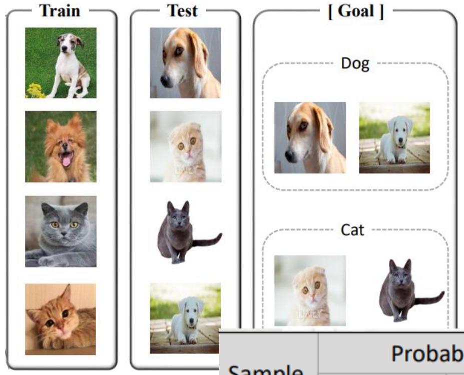
Table 1. Differences between open set recognition and its related tasks

TASK \ SETTING	TRAINING	TESTING	GOAL
Traditional Classification	Known known classes	Known known classes	Classifying known known classes
Classification with Reject Option	Known known classes	Known known classes	Classifying known known classes & rejecting samples of low confidence
One-class Classification (Anomaly Detection)	Known known classes & few or none outliers from KUCs	Known known classes & few or none outliers	Detecting outliers
One/Few-shot Learning	Known known classes & a limited number of UKCs' samples	Unknown known classes	Identifying unknown known classes
Generalized Few-shot Learning	Known known classes & a limited number of UKCs' samples	Known known classes & unknown known classes	Identifying known known classes & unknown known classes
Zero-shot Learning	Known known classes & side-information[1]	Unknown known classes	Identifying unknown known classes
Generalized Zero-shot Learning	Known known classes & side-information[1]	Known known classes & unknown known classes	Identifying known known classes & unknown known classes
Open Set Recognition	Known known classes	Known known classes & unknown unknown classes	Identifying known known classes & rejecting unknown unknown classes

1. Introduction

Recent advances in OSR

Traditional classification

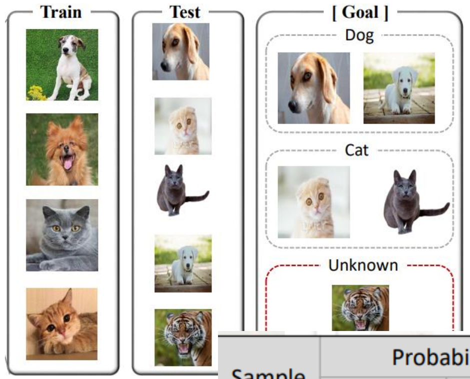


Sample	Probability		Class
	Dog	Cat	
1	0.97	0.03	Dog
2	0.01	0.99	Cat
3	0.23	0.77	Cat
4	0.37	0.63	Cat

$$p_i = \frac{e^{zj}}{\sum_{j=1}^{k} e^{zj}}$$

$$label\ index = argmax[p_1, p_2, \ldots, p_k]$$

Classification with reject option

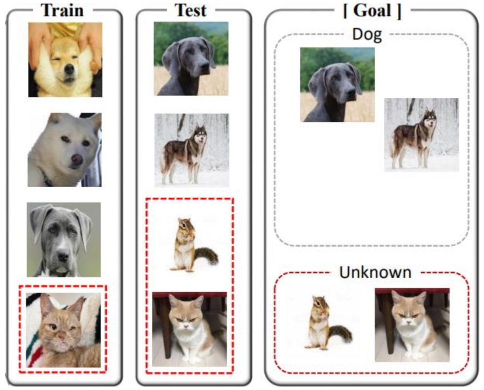


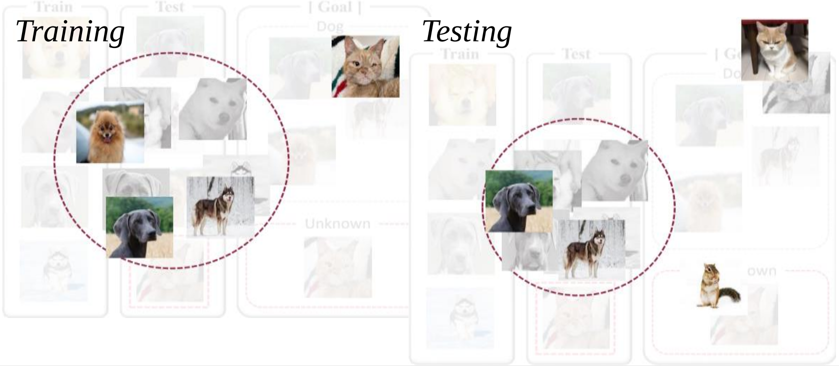
Sample	Probability		Class
	Dog	Cat	
1	0.97	0.03	Dog
2	0.01	0.99	Cat
3	0.23	0.77	Cat
4	0.37	0.63	Unknown

$$Threshold:\ \alpha$$

$$Class = \begin{cases} reject, if\ softmax < \alpha \\ argmax, if\ softmax \geq \alpha \end{cases}$$

One-class classification (Anomaly detection)





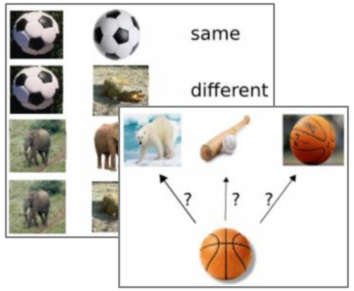
1. Introduction

Recent advances in OSR

One/few-shot learning

One shot learning

- ✓ (Training) 각 image별 same/different 학습
- ✓ (Test) 유사도 계산을 통해 분류함

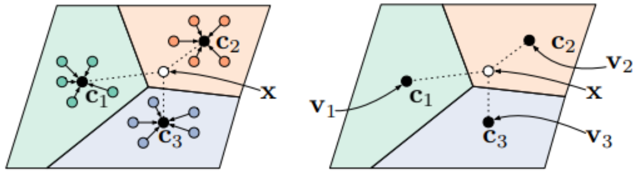


2-way 1-shot classification (N-way, k-shot learning)

- ✓ Classes N개 / examples k개
- ✓ Training data는 $n \times k$ 개의 point로 이루어짐
- ✓ 적은 데이터로도 충분한 학습이 가능함
- ✓ Robust feature 학습이 목적임

Zero-shot learning

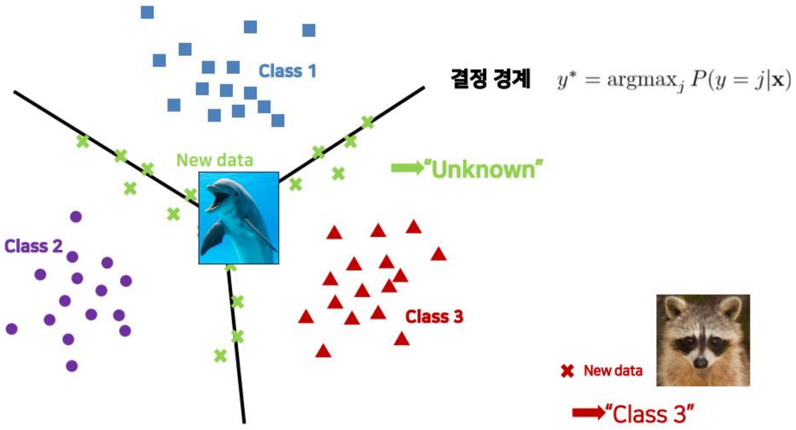
Figure 1



Few-shot   Zero-shot

- ✓ 학습을 한 번도 하지 않고 신뢰성 있는 예측을 하고자 함
- ✓ 새로운 범주(class)가 지속적으로 생성됨

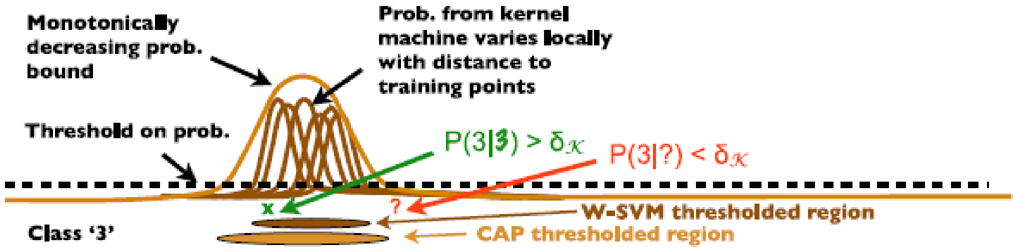
Open set recognition

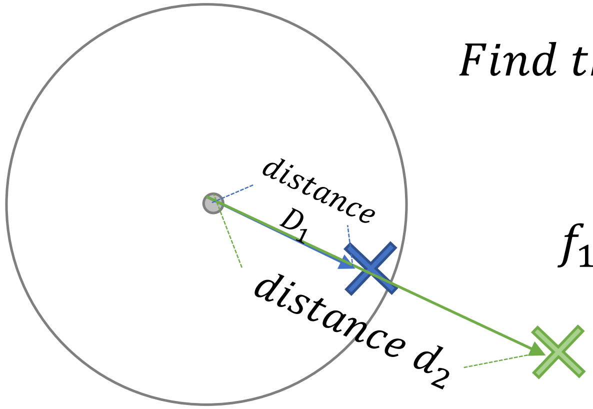


- ✓ 학습에 사용되지 않은 데이터의 범주를 분류
- ✓ 평균으로부터 떨어진 거리, 다른 클래스와의 마진 등을 활용해 결정경계 생성
- ✓ GAN 등의 생성모델을 통해 각 클래스와 비슷한 다른 이미지를 생성하고, 새로운 범주로 추가한 후 학습

2. Related works

Compact Abating Probability(CAP)





$$Find\ the\ function\ f(x)\ !$$

$$D_1 < D_2$$

$$f_1(x) > f_2(x)$$

- ✓ Recognition 능력을 감소시키는 것이 목적임

$$f(x) > 0\ when\ the\ class\ y\ recognized$$
$$f(x) = 0\ when\ not\ recognized$$

- ✓ Threshold 설정을 통해 known space K와 open space O를 구분함

$$if \min_{x_i \in K} \|x - x_i\| > \tau, x \in O$$

- ✓ Open space:

$$O = S_0 - \underset{i \in N}{union} B_r(x_i)$$

출처:Bendale, A., & Boult, T. E. (2016). Towards open set deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1563-1572)., http://dmqm.korea.ac.kr/activity/seminar/281

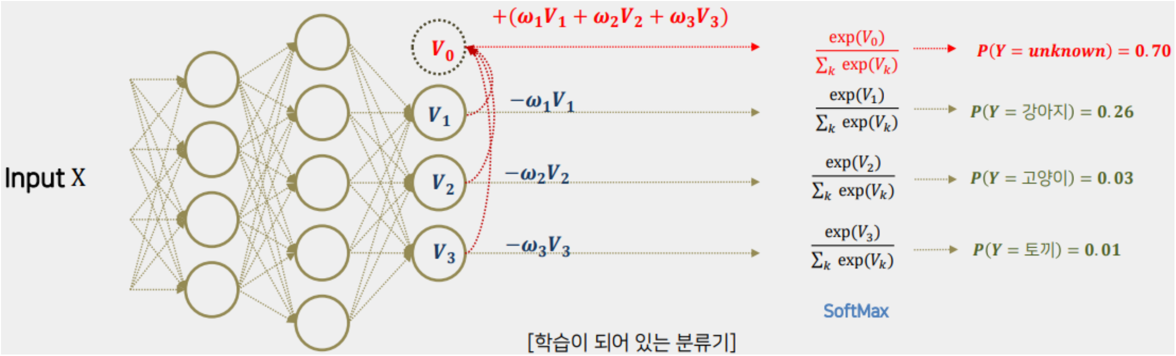
2. Related works

OpenMax

a. Known data로 모델 학습

Unknown class 의 logit 값

b. 모델이 k class로 잘못 분류했을 확률에 대응하는 가중치 정의



출처:Bendale, A., & Boult, T. E. (2016). Towards open set deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1563-1572)., http://dmqm.korea.ac.kr/activity/seminar/281

2. Related works

OpenMax

c. Extreme value theorem을 통한 가중치 정의

✓ Class 별 평균 logit vector로부터의 거리에 대한 극단 값(이상치)의 분포 확인

Obs.	V_1	V_2	V_3
N_{11}	9.87	-2.13	-6.23
N_{12}	18.5	3.18	4.98
…	…	…	…
N_{1a}	4.89	-3.91	1.01

Obs.	V_1	V_2	V_3
N_{21}	2.17	8.29	-1.32
N_{22}	1.89	9.13	2.11
…	…	…	…
N_{2b}	-3.14	5.42	1.35

Obs.	V_1	V_2	V_3
N_{31}	-3.23	-1.23	6.98
N_{32}	1.52	1.52	9.78
…	…	…	…
N_{3c}	5.72	3.12	11.2

Class 별 logit vector matrix

$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix} \quad \mu_{cat} = \begin{bmatrix} 0.12 \\ 11.27 \\ -3.53 \end{bmatrix} \quad \mu_{rabit} = \begin{bmatrix} 1.32 \\ -2.35 \\ 9.32 \end{bmatrix}$$

평균 logit vector

$$\left\| \begin{bmatrix} 9.87 \\ -2.13 \\ -6.23 \end{bmatrix} - \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix} \right\|_2 = 40.3225$$

$$\left\| \begin{bmatrix} 18.5 \\ 3.18 \\ 4.98 \end{bmatrix} - \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix} \right\|_2 = 221.134$$

$$\vdots$$

$$\left\| \begin{bmatrix} 4.89 \\ -3.91 \\ 1.01 \end{bmatrix} - \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix} \right\|_2 = 8.6291$$

평균과의 거리 계산

거리 계산 값 내림차순 정렬

Obs.	평균 Logit Vector와의 거리
N_{11}	221.134
N_{12}	217.532
…	…
N_{1a}	0.014

평균 Logit Vector와의 거리 중 가장 큰 η(=20)개 Sample 추출

$$\underbrace{[221.134, 217.532, \cdots, 197.423]}_{\eta(=20)\text{개}}$$

거리 계산 값 중 가장 큰 N개의 sample을 class 별로 추출

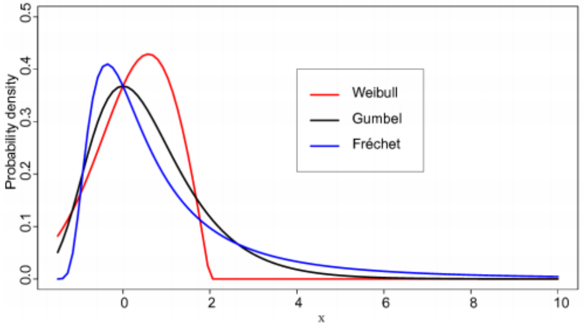
출처:Bendale, A., & Boult, T. E. (2016). Towards open set deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1563-1572)., http://dmqm.korea.ac.kr/activity/seminar/281

2. Related works

OpenMax

c. Extreme value theorem을 통한 가중치 정의

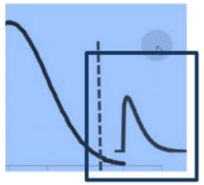
- ✓ 극단 값의 분포, 이상치의 분포를 추정
- ✓ 동일분포에서 독립적으로 추출한 sample 중 가장 큰 값을 추출하였을 때, 가장 큰 값보다 클 확률은 Weibull 분포 / Frechet 분포, Gumbel 분포의 형태를 따를 수 있음



Obs.	평균 Logit Vector와의 거리
N_{11}	221.134
N_{12}	217.532
…	…
N_{1a}	0.014

[221.134, 217.532, ⋯ , 197.423]

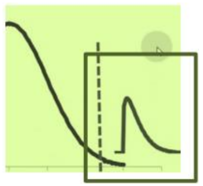
η(=20)개



Obs.	평균 Logit Vector와의 거리
N_{21}	195.312
N_{22}	194.904
…	…
N_{2b}	0.082

[195.312 , 194.904 , ⋯ , 177.197]

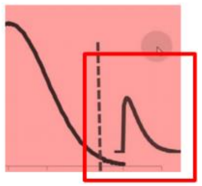
η(=20)개



Obs.	평균 Logit Vector와의 거리
N_{31}	139.641
N_{32}	121.896
…	…
N_{3c}	0.167

[139.641 , 121.896 , ⋯ , 116.493]

η(=20)개



평균 logit vector와의 거리 계산 값의 극단 분포 추정

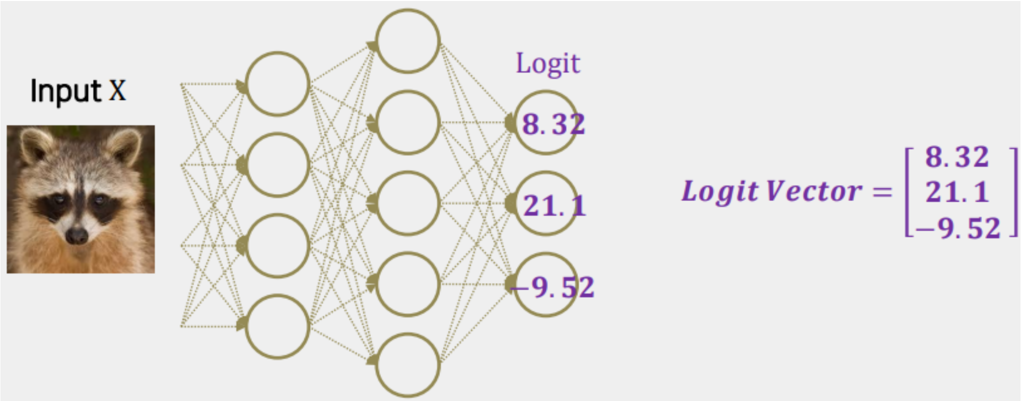
출처:Bendale, A., & Boult, T. E. (2016). Towards open set deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1563-1572)., http://dmqm.korea.ac.kr/activity/seminar/281

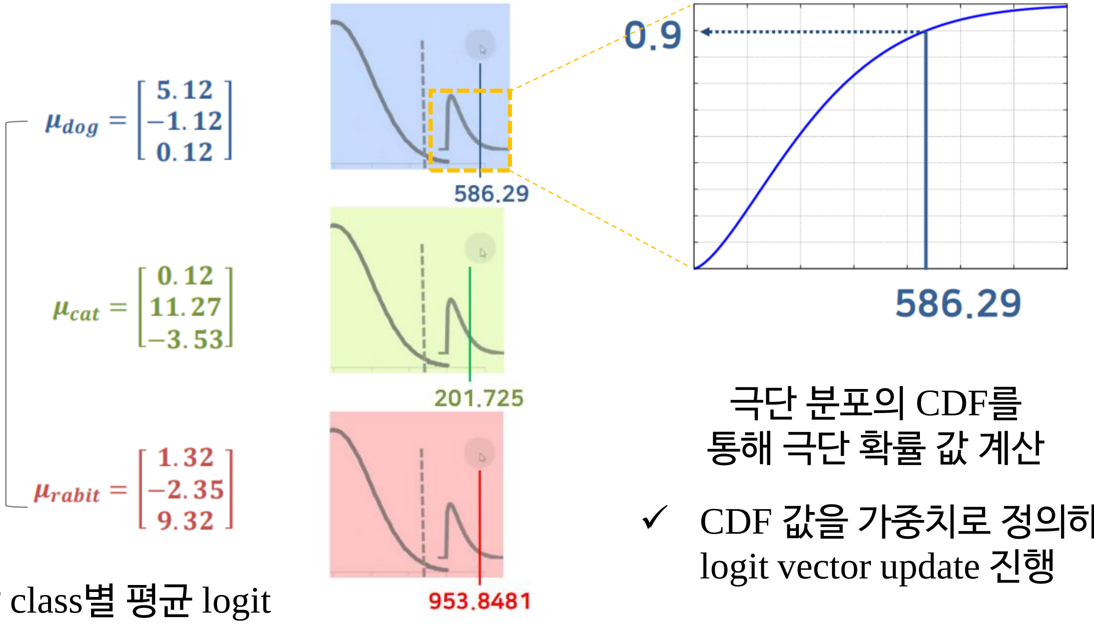
2. Related works

OpenMax

c. Extreme value theorem을 통한 가중치 정의

✓ 학습된 모델 실행





각 class별 평균 logit vector와의 거리 계산

극단 분포의 CDF를 통해 극단 확률 값 계산

✓ CDF 값을 가중치로 정의하여, logit vector update 진행

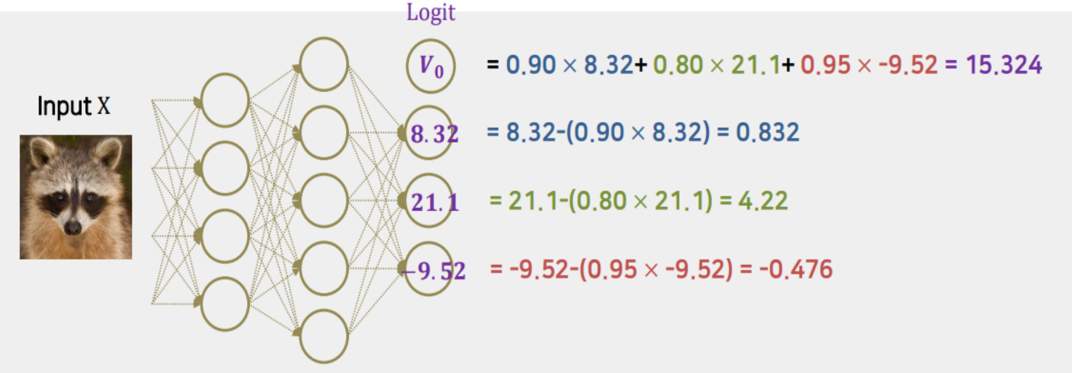
Bendale, A., & Boult, T. E. (2016). Towards open set deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1563-1572).,
http://dmqm.korea.ac.kr/activity/seminar/281

2. Related works

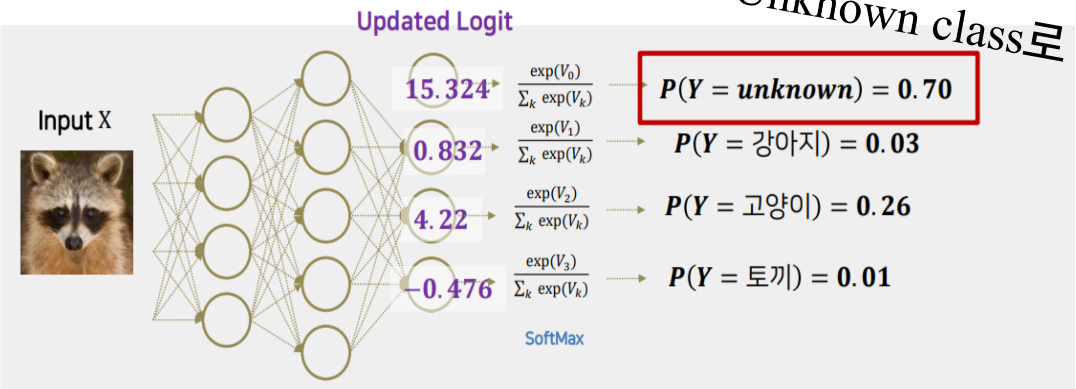
OpenMax

c. Extreme value theorem을 통한 가중치 정의

✓ 가중치 정의 후, 학습된 모델 업데이트



Unknown class로 분류



출처:Sun, X., Li, X., Ren, K., & Song, J. (2019, November). Solving the Defect in Application of Compact Abating Probability to Convolutional Neural Network Based Open Set Recognition. In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 856-863). IEEE.

3. Methodology

What is the problem?

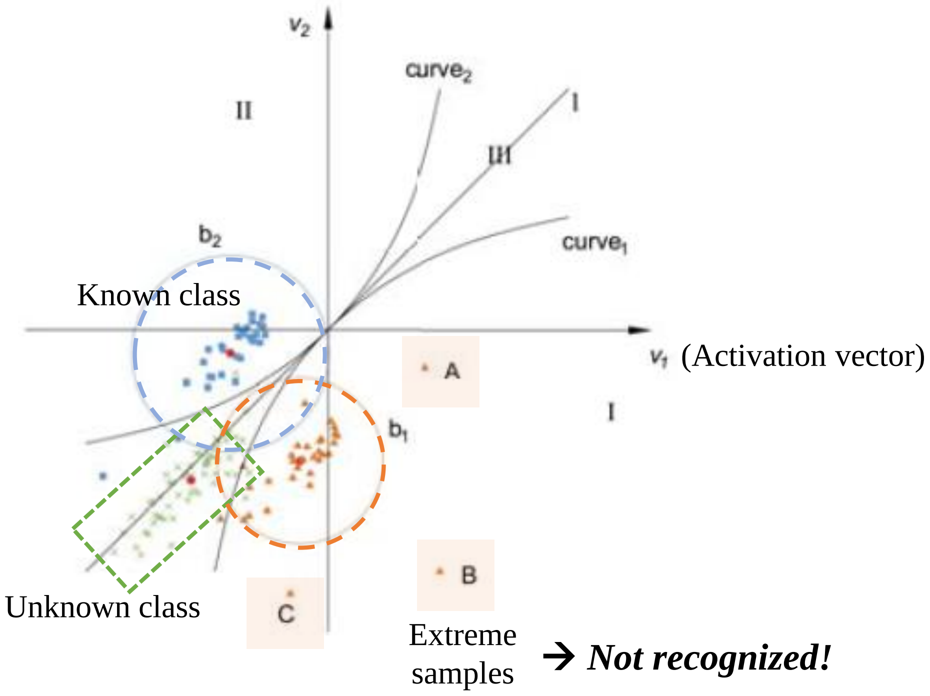


Figure 1. AV distribution of 2-dimension samples

- ✓ Extreme samples에 대해서는 인식하지 못함
- ✓ Curve 형태로 boundary가 설정되는 것이 이상적임
- ✓ Logit layer 공간을 활용하는 것은 부적합함

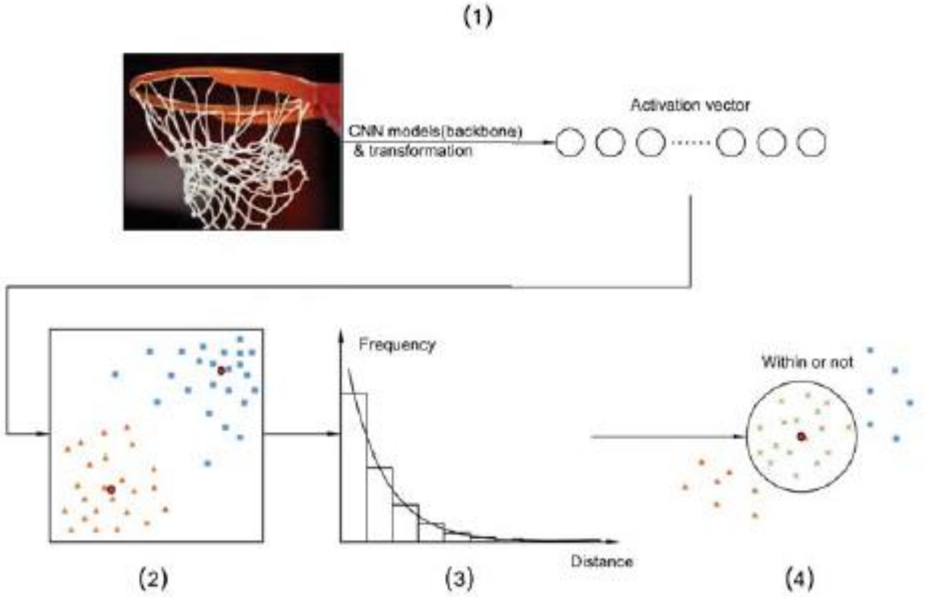
출처:Sun, X., Li, X., Ren, K., & Song, J. (2019, November). Solving the Defect in Application of Compact Abating Probability to Convolutional Neural Network Based Open Set Recognition. In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 856-863). IEEE.

3. Methodology

Overview of the methodology

Logit layer 공간의 activation vector에 softmax transformation을 적용하여 boundary를 생성하는 ***OpenSoftMax***

Figure 2. The overview of the proposed method



1) Deep feature extraction
2) Center calculation
3) Threshold setting
4) Determination

출처:Sun, X., Li, X., Ren, K., & Song, J. (2019, November). Solving the Defect in Application of Compact Abating Probability to Convolutional Neural Network Based Open Set Recognition. In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 856-863). IEEE.

3. Methodology

OpenSoftMax

a. Deep feature extraction

- ✓ Deep features 추출을 위해 CNN 모델 적용
- ✓ AlexNet 사용

Table 1. The structure of AlexNet

layer name	output size	parameters
cov1	55 × 55	11 × 11, 96, stride 4
pool1	27 × 27	3 × 3, stride 2
conv2	27 × 27	5 × 5, 256, stride 1
pool2	13 × 13	3 × 3, stride 2
conv3	13 × 13	3 × 3, 384, stride 1
conv4	13 × 13	3 × 3, 384, stride 1
conv5	13 × 13	3 × 3, 256, stride 1
pool5	6 × 6	3 × 3, stride 2
fc6	4096	6 × 6 × 256
fc7	4096	4096
fc8	num_of_class	4096
SoftMax	num_of_class	num_of_class

- ✓ Penultimate layer가 아닌 SoftMax layer를 사용
- ✓ Logits layer에서 추출된 features에 대해 normalized exponential transformation 적용

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

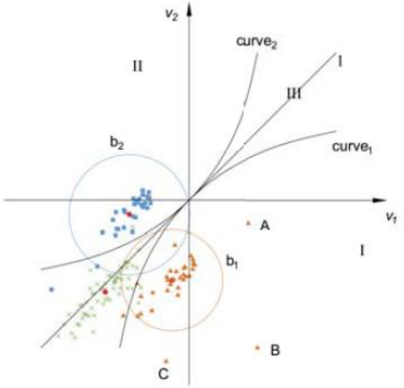


Figure 1. Original space

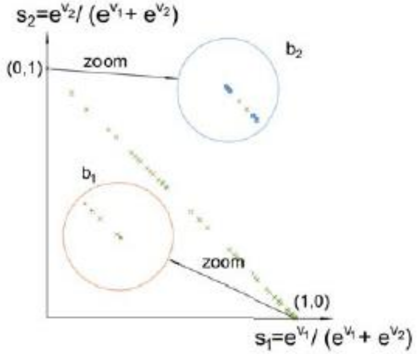


Figure 3. Transformed space

출처:Sun, X., Li, X., Ren, K., & Song, J. (2019, November). Solving the Defect in Application of Compact Abating Probability to Convolutional Neural Network Based Open Set Recognition. In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 856-863). IEEE.

3. Methodology

OpenSoftMax

b. Center calculation

- ✓ 각 class별 평균 activation vectors(AVs) 계산
- ✓ 평균 activation vector를 center point로 정의

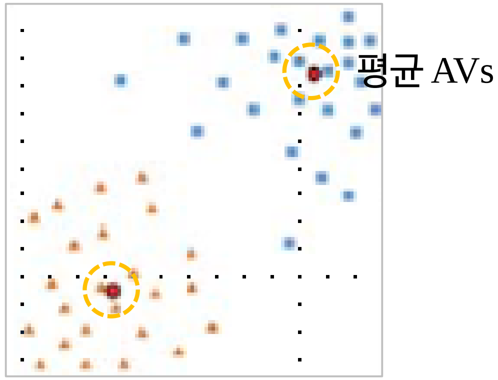


Figure 2-(2). Average AVs

c. Threshold setting & Determination

- ✓ Intra-class points와 center 간 Euclidean 거리 계산

$$\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\dots+(p_n-q_n)^2}=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}$$

- ✓ 각 class별 거리 분포를 활용해 threshold τ 설정
  - Threshold는 거리 분포의 평균 $\bar{d}$와 표준편차 σ 사이에 위치 $([\bar{d}-\sigma,\ \bar{d}+\sigma])$
  - 실험 반복을 통해 threshold 값 결정
- ✓ Threshold 값 기준으로 pen set에 속하는지 여부를 결정

$$if\ \min_{i\in[1,M]}\|AV_x - MAV_i\| > \tau, x \in O$$

4. Experimental results

The experimental design

Data: Caltech256 dataset

- ✓ 257 visual categories
- ✓ 30,607 images with at least 80 images for each categories
- ✓ Randomly choosing parts of classes as unknown data

Table 2. Detailed division of Caltech256

Number of Known Classes	Number of Known Images	Number of Unknown Classes	Number of Unknown Images	Percentage of Open Data
237	28153	20	2032	6.73%
217	26031	40	4154	13.76%
197	23739	60	6446	21.36%
177	21345	80	8840	29.29%
157	18443	100	11742	38.90%
137	16027	120	14158	46.90%
117	13712	140	16473	54.57%
97	11661	160	18524	61.37%
77	8031	180	22154	73.39%
57	5933	200	24252	80.34%
37	3786	220	26399	87.46%

Measurements of model evaluation

- ✓ Accuracy
- ✓ F1 – score
- ✓ Error rate

Model parameters

Table 4. Parameters in training process

Name	GoogleNet_v3
Trainable layers	Logits, AuxLogits
Optimizer	RMSProp
Learning Rate	0.0001
Batch Size	32
Maximum Steps	6000

4. Experimental results

Results

OpenMax와 OpenSoftMax 비교

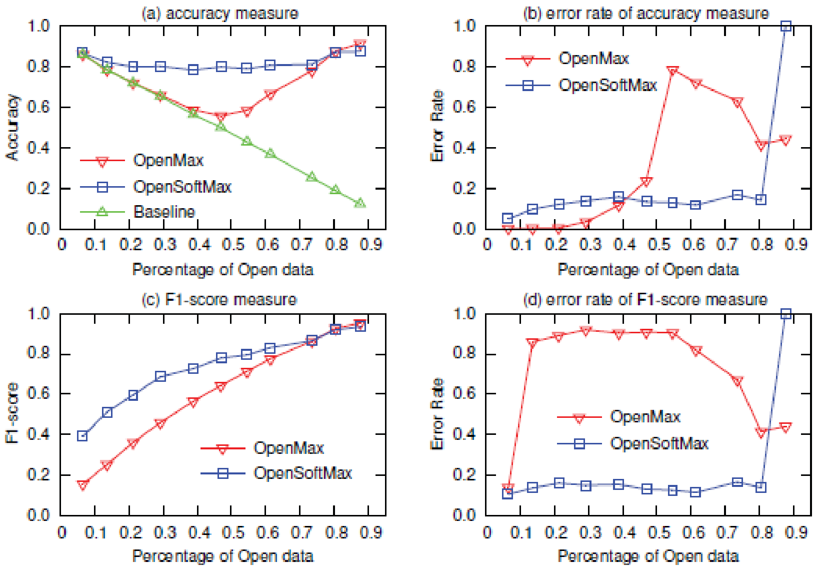


Figure 4. Overall results of comparison

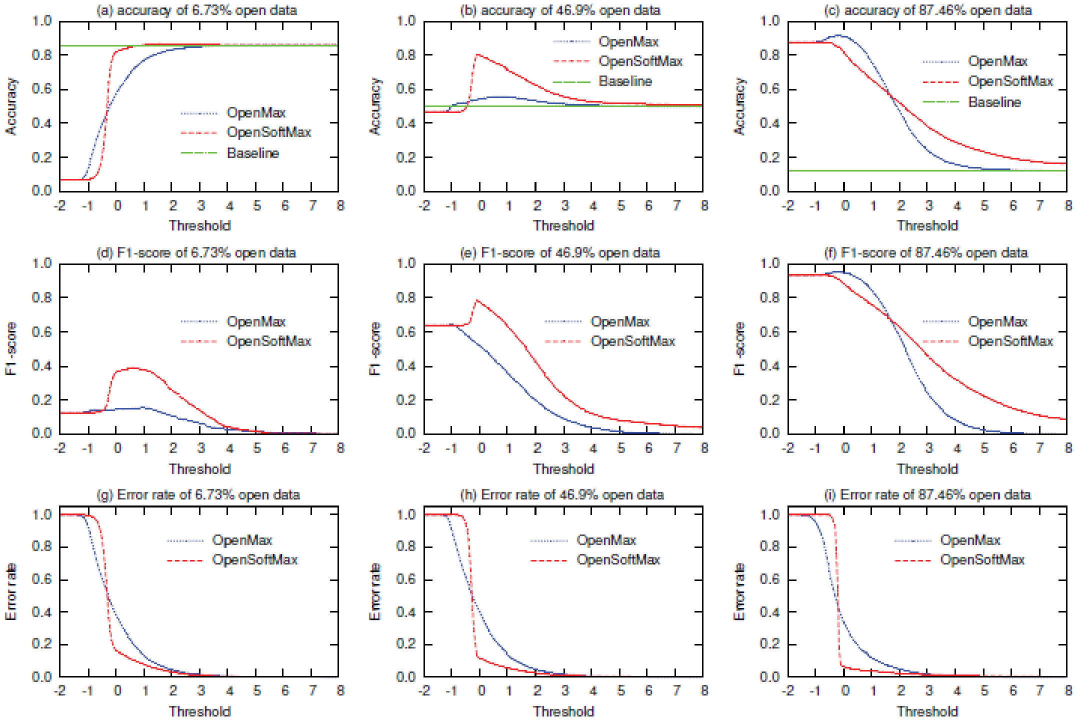


Figure 5. Results of the detailed comparisons

5. Conclusion

Summary

- CNN-based OSR 방법인 OpenMax에 대한 CAP 모델 적용이 부적합함을 보임
- Activation vector에 SoftMax transformation을 적용해 OpenMax의 단점을 보완하였음
- OpenSoftMax는 OpenMax에 비해 좋은 성능을 보였음

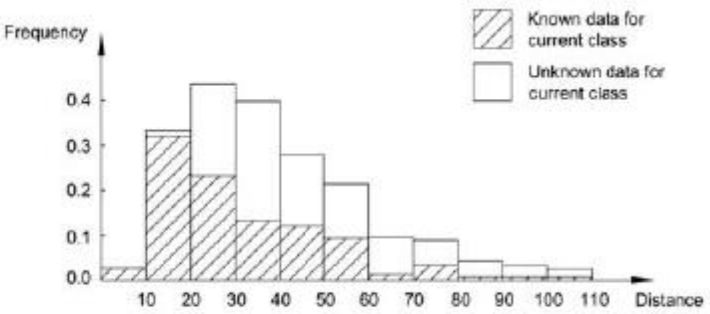


Figure 6. OpenMax distance distribution

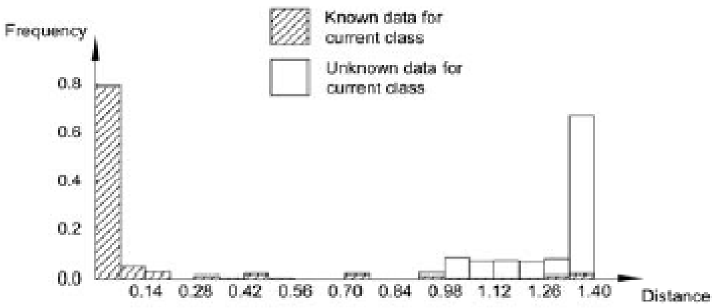


Figure 7. OpenSoftMax distance distribution

Thank you for your listening ☺

Yunseon Byun
May 04, 2020