수학적 통찰을 통한 딥러닝

다양한 주제의 통합적 연구

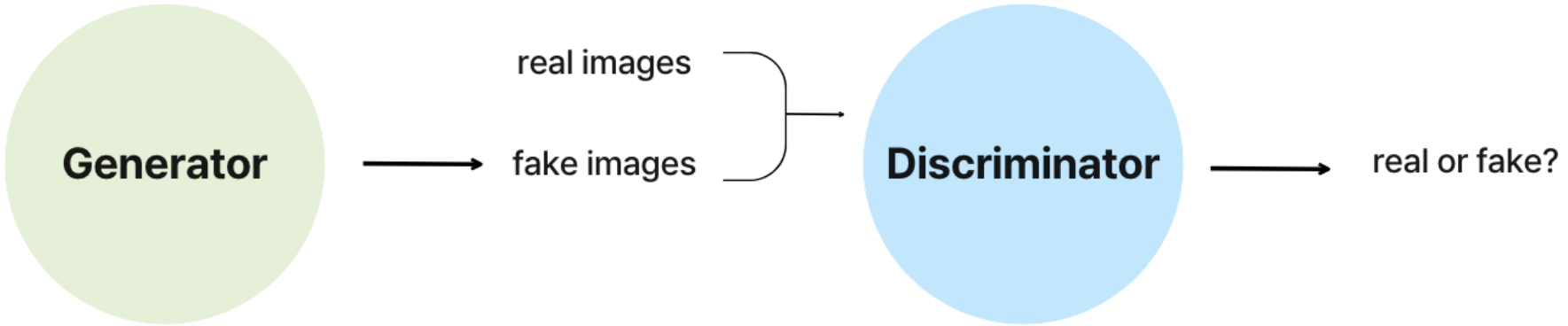
서울시립대학교 수리계산연구실
수학과 18 강경헌
수학과 18 강태욱
수학과 19 소 신
수학과 19 황태연

목차

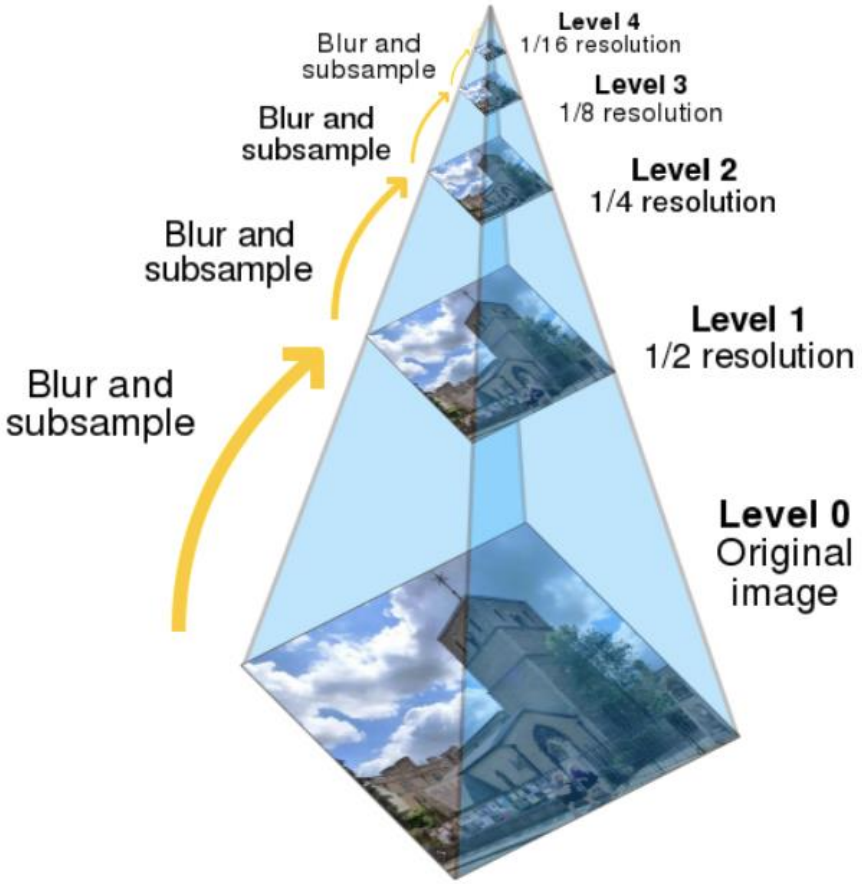
1. Laplacian pyramid를 활용한 GAN 모델

발표자: 강경헌

1. What is GAN?


Generator
real images
fake images
Discriminator
real or fake?

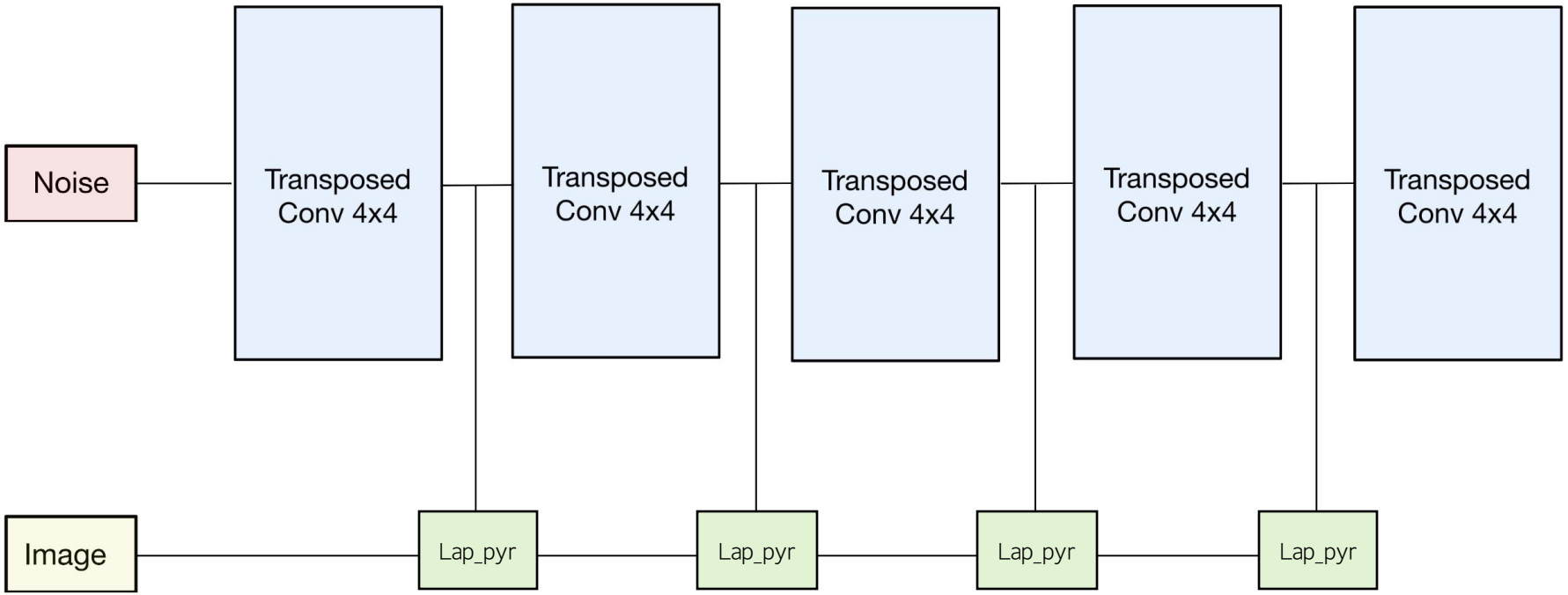
2. Laplacian pyramid



3. Our model

- Generator

DCGAN에서의 Generator 구조에 훈련 이미지의 laplacian pyramid를 concat해주어 훈련 이미지의 laplacian pyramid를 conditioning한 구조라 생각할 수 있다.



4. Experiments

- **데이터셋**은 CelebA와 CIFAR-10 벤치마크 데이터셋을 증강 없이 사용하였다.

- Optimizer는 AdamW를 사용하였고 learning rate는 0.0002, betas는 (0.1, 0.999), weight_decay는 0.00002를 사용했다.

- **기본 구조인** DCGAN과 비교하기 위해 DCGAN은 위 실험과 동일하게 진행했다.

- **평가 지표로는** FID(Fréchet Inception Distance) score와 LPIPS(Learned perceptual image patch similarity)로 평가한다.

5. Results

- 본 연구의 제안 모델이 생성한 이미지

Real Images

Fake Images

5. Results

- DCGAN 모델이 생성한 이미지

5. Results

- FID Score, LPIPS score 비교

	Our	DCGAN
FID score	159.12	112.07
LPIPS score	0.4803	0.5063

6. Future works

- 고해상도에서의 이미지 생성

- 좋은 퀄리티의 이미지 생성

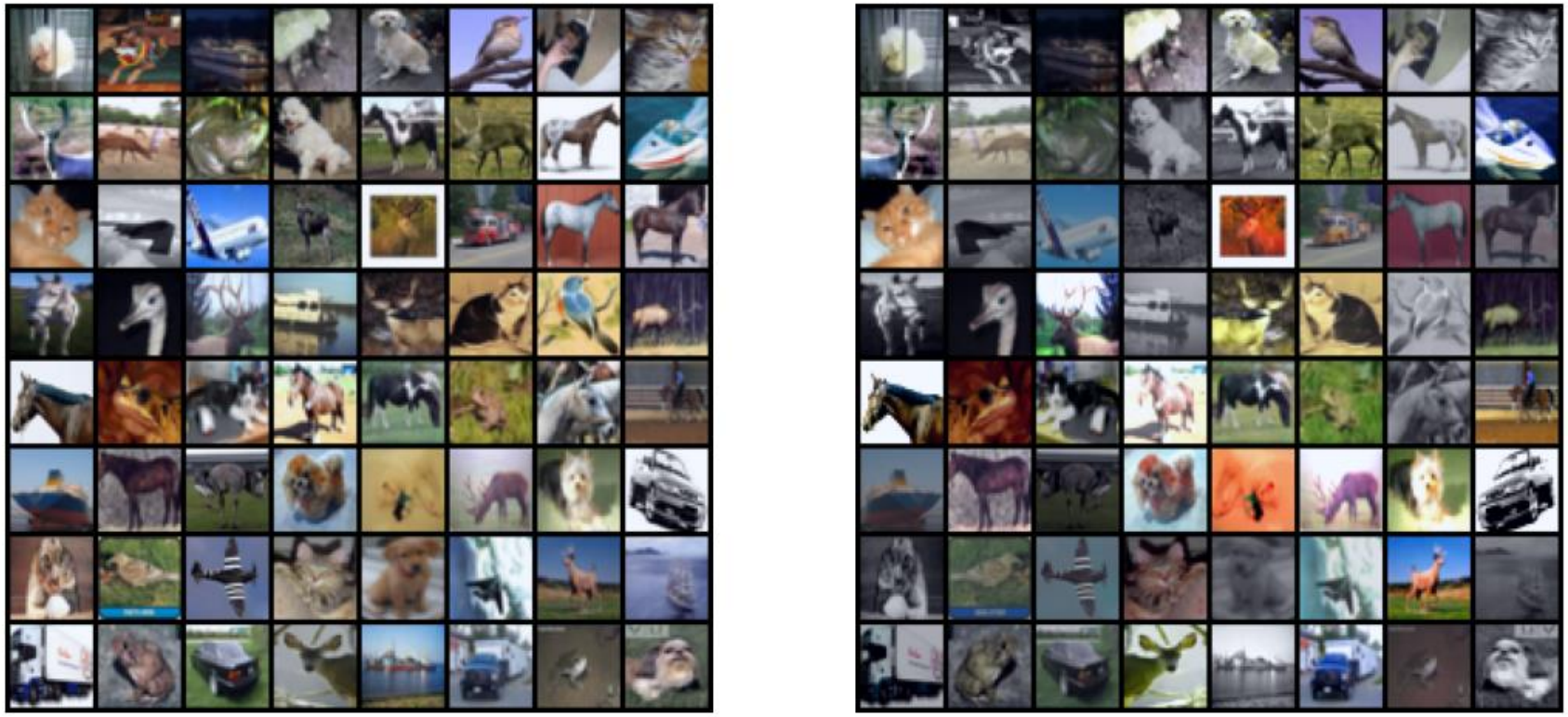
- 데이터 증강에서의 Laplacian pyramid

- GAN의 안정적인 학습을 위한 목적 함수 설정

2. Categorical Similarity Learning

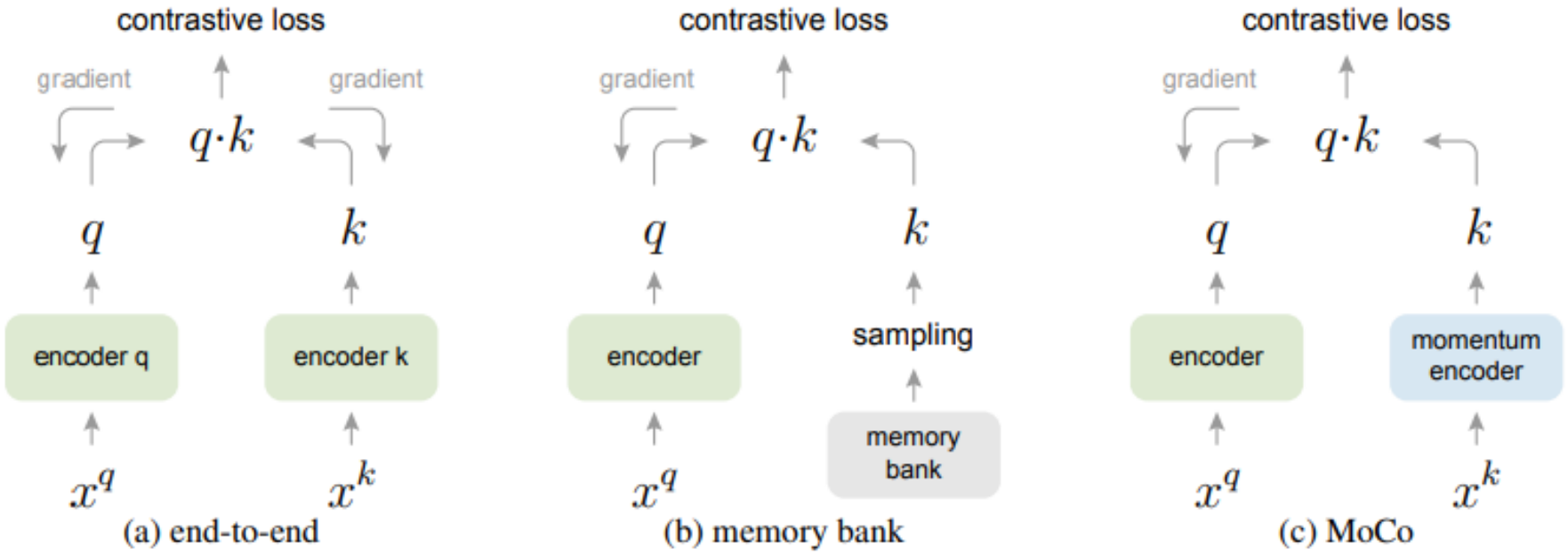
발표자: 강태욱

1. Similarity Learning



▲ [그림 1] Similarity Learning

2. Related Work



▲ [그림 2] Momentum Contrast for Unsupervised Visual Representation Learning

3. Categorical Similarity

x_i : $[0, 2, 1, 1, 0, \ldots, 4]$

x_i' : $[0, 2, 1, 1, 3, \ldots, 2]$

$\mathrm{x_j}$: $[0, 3, 4, 4, 3, \ldots, 4]$

▲ [그림 3] Categorical Similarity

4. Method

Input,
(64x64x3)

Encoder

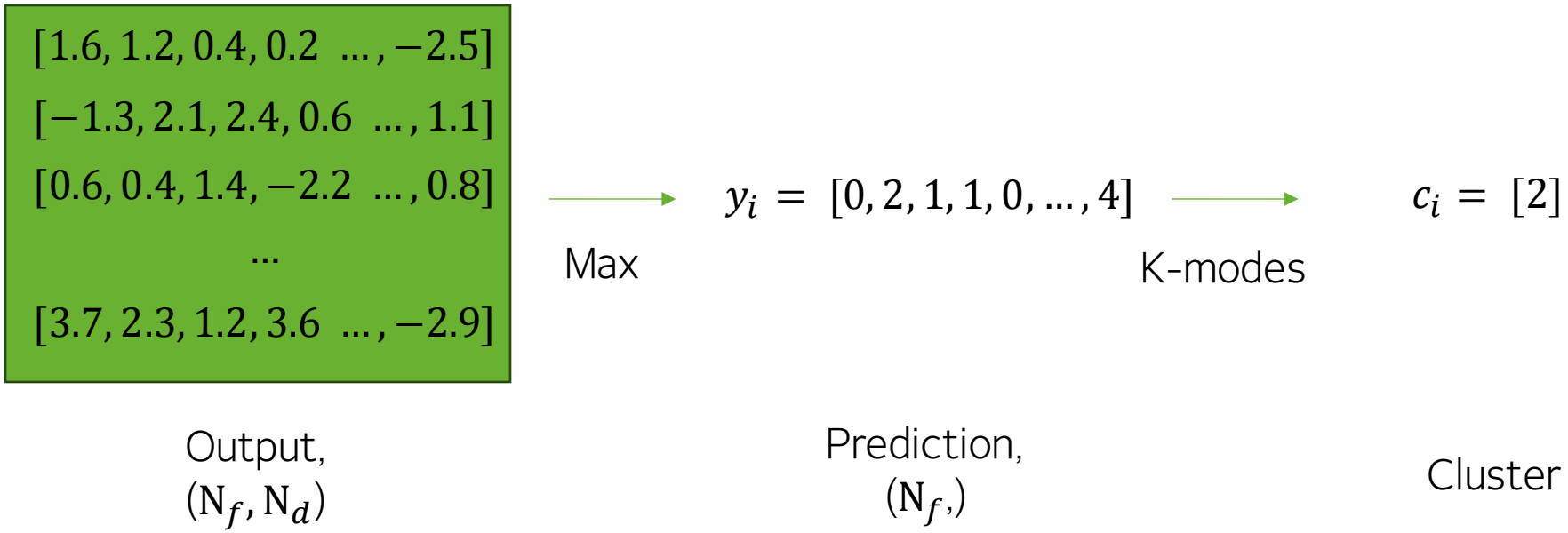
Feature maps,
(4x4x512)

AvgPool

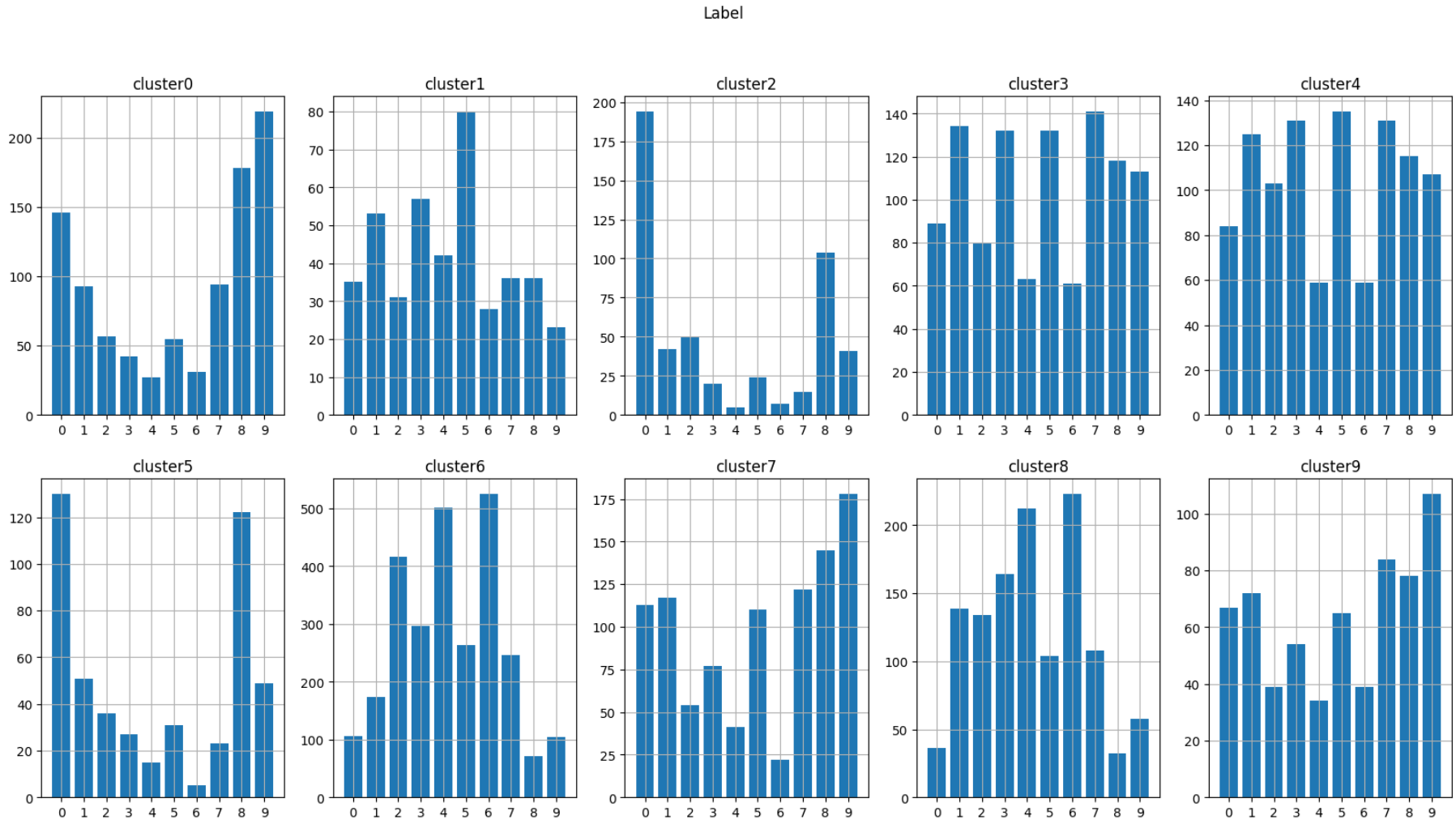
$$[1.6, 1.2, 0.4, 0.2 \ \ldots, -2.5]$$
$$[-1.3, 2.1, 2.4, 0.6 \ \ldots, 1.1]$$
$$[0.6, 0.4, 1.4, -2.2 \ \ldots, 0.8]$$
$$\ldots$$
$$[3.7, 2.3, 1.2, 3.6 \ \ldots, -2.9]$$

Output,
$(\mathrm{N}_f, \mathrm{N}_d)$

4. Method

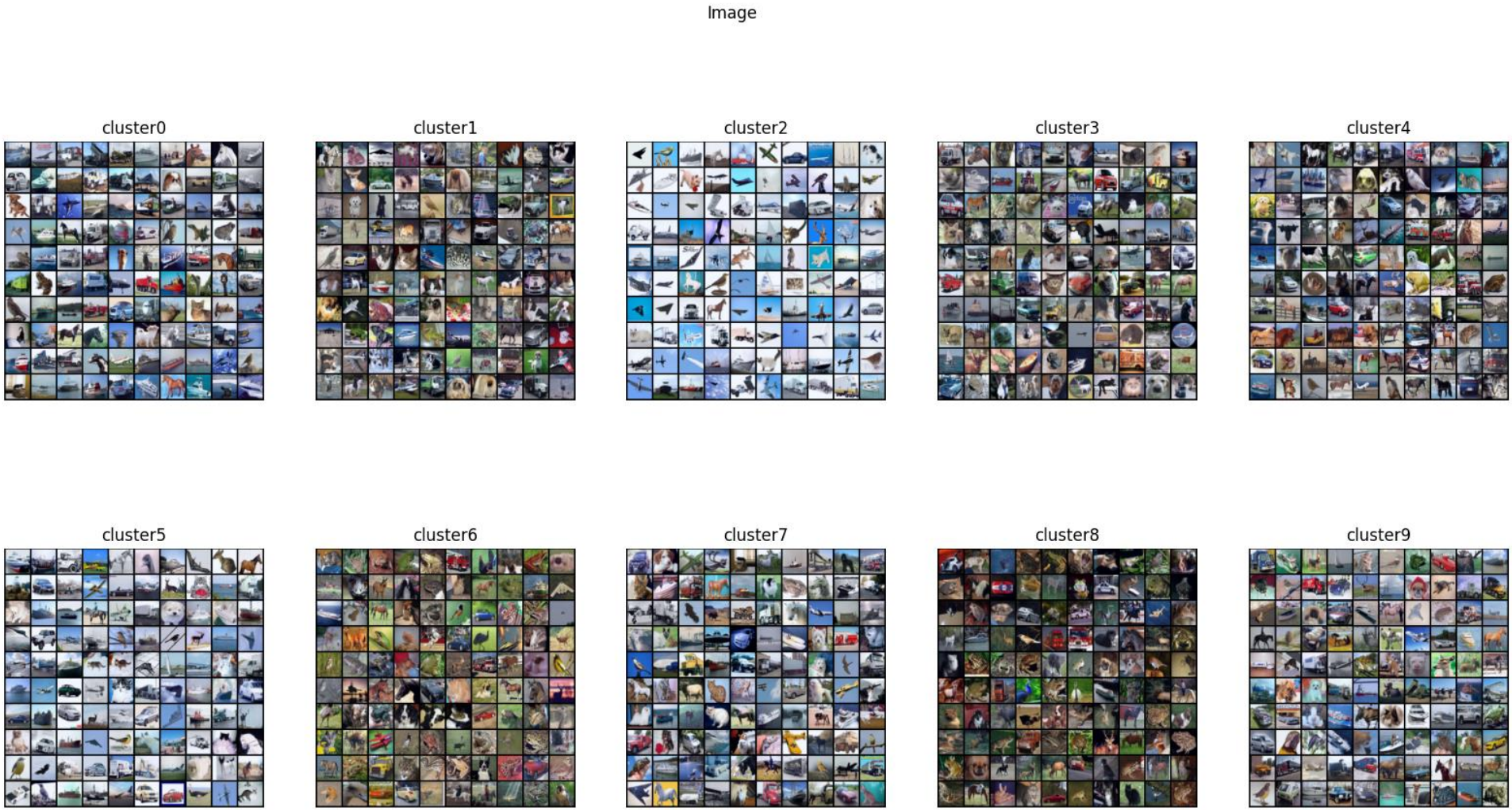

[1.6, 1.2, 0.4, 0.2 … , −2.5]
[−1.3, 2.1, 2.4, 0.6 … , 1.1]
[0.6, 0.4, 1.4, −2.2 … , 0.8]
…
[3.7, 2.3, 1.2, 3.6 … , −2.9]
Max
$y_i = [0, 2, 1, 1, 0, \dots, 4]$
K-modes
$c_i = [2]$
Output, $(\mathrm{N}_f, \mathrm{N}_d)$
Prediction, $(\mathrm{N}_f,)$
Cluster

5. Experiment



▲ [그림 4] Clustering Results (1)

5. Experiment



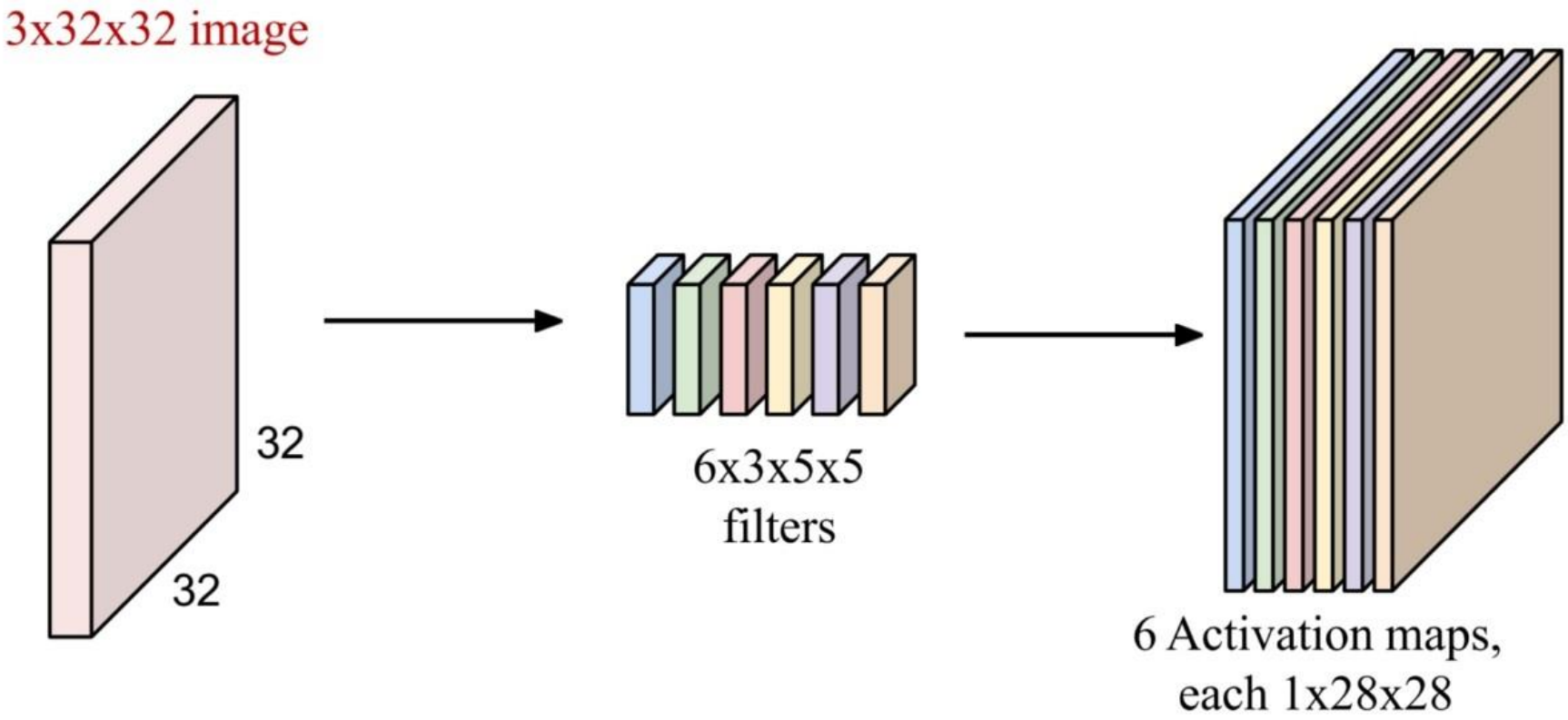
▲ [그림 5] Clustering Results (2)

3. Corregularization

: Filter간 Gram matrix를 이용한 Loss function design

발표자: 소 신

1. CNN Filter

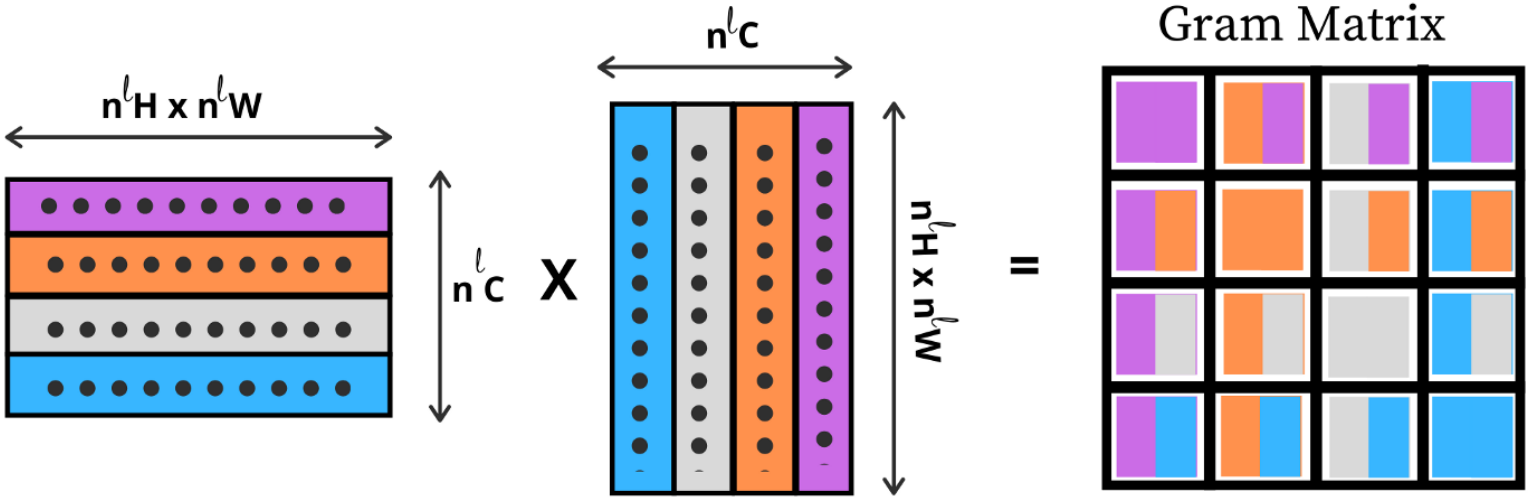


▲ [그림 1] CNN Filter (1)

1. CNN Filter

▲ [그림 2] CNN Filter (2)

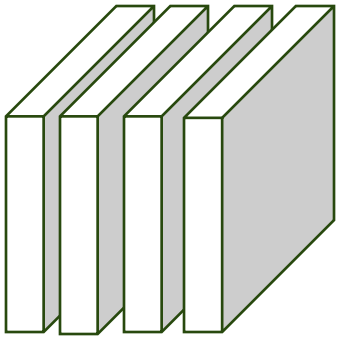
2. Gram Matrix



$$G(x_1, \ldots, x_n) = \begin{vmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle & \ldots & \langle x_1, x_n \rangle \\ \langle x_2, x_1 \rangle & \langle x_2, x_2 \rangle & \ldots & \langle x_2, x_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n, x_1 \rangle & \langle x_n, x_2 \rangle & \ldots & \langle x_n, x_n \rangle \end{vmatrix}.$$

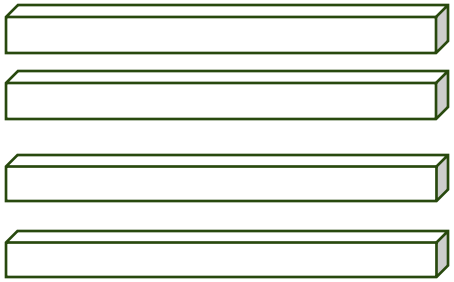
▲ [그림 3] Gram Matrix의 정의

3. Method



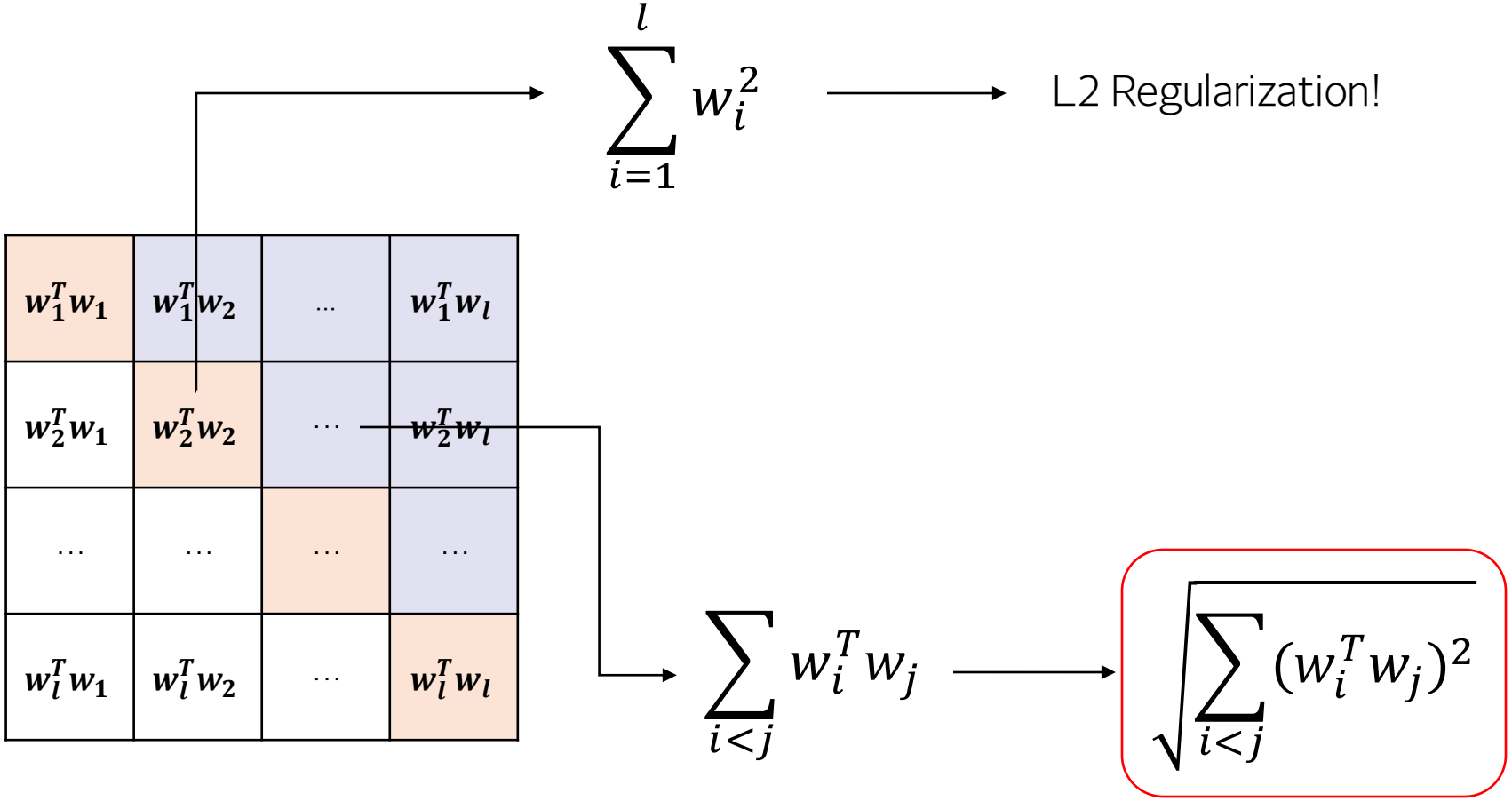
w_1, w_2, w_3, w_4

vectorization



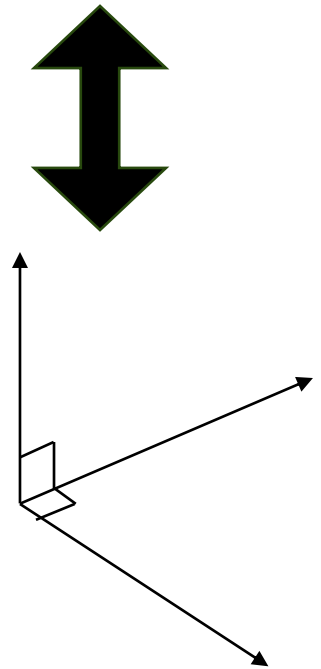
$w_1^T w_1$	$w_1^T w_2$		$w_1^T w_l$
$w_2^T w_1$	$w_2^T w_2$		$w_2^T w_l$
$w_2^T w_1$			
$w_l^T w_1$	$w_l^T w_2$		$w_l^T w_l$

3. Method


$\sum_{i=1}^{l} w_i^2$
L2 Regularization!
$w_1^T w_1$ $w_1^T w_2$ … $w_1^T w_l$
$w_2^T w_1$ $w_2^T w_2$ … $w_2^T w_l$
… … … …
$w_l^T w_1$ $w_l^T w_2$ … $w_l^T w_l$
$\sum_{i<j} w_i^T w_j$
$\sqrt{\sum_{i<j} (w_i^T w_j)^2}$

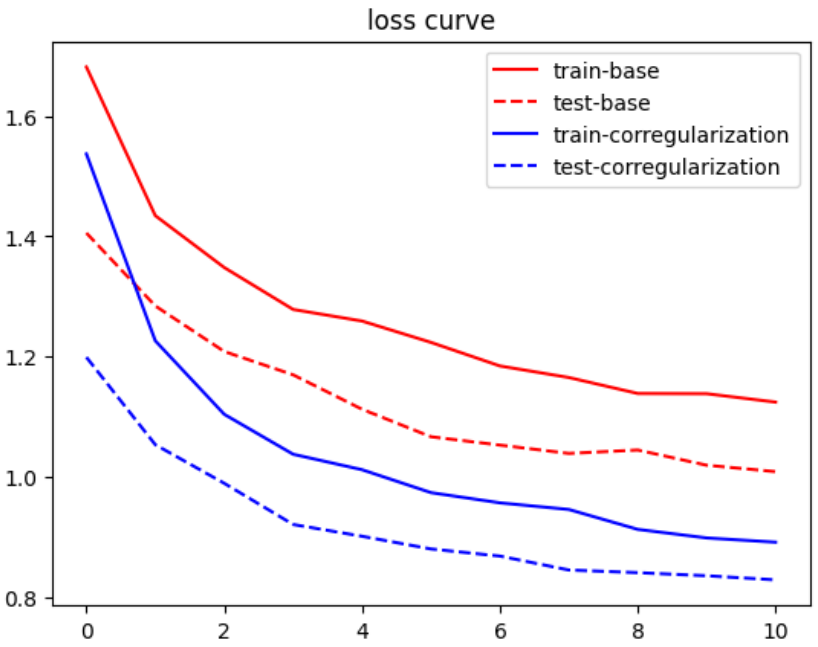
3. Method: loss term의 의미

$$\sqrt{\sum_{i<j}(w_i^T w_j)^2} \longrightarrow 0$$

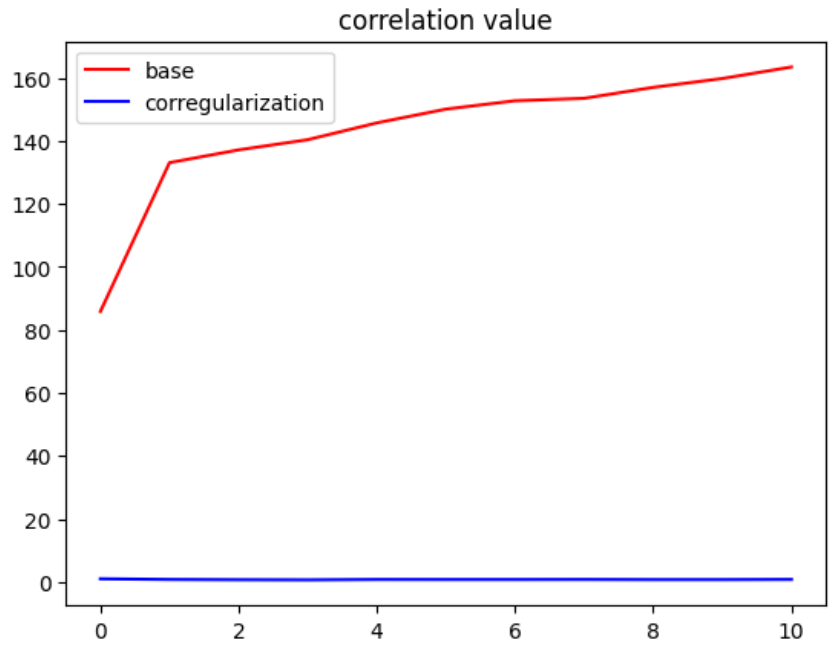


3. Method: First Experiment

Data set : CIFAR-10
Model : Vanilla CNN

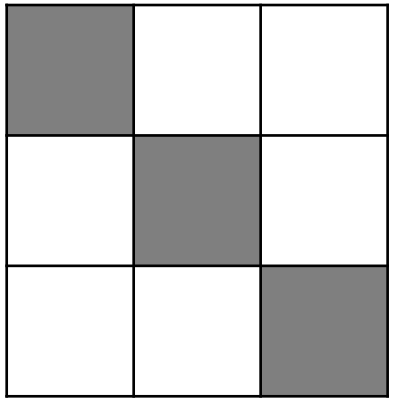


▲ [그림 4] Loss curve (1)

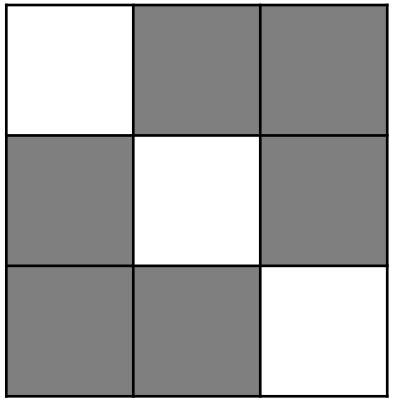


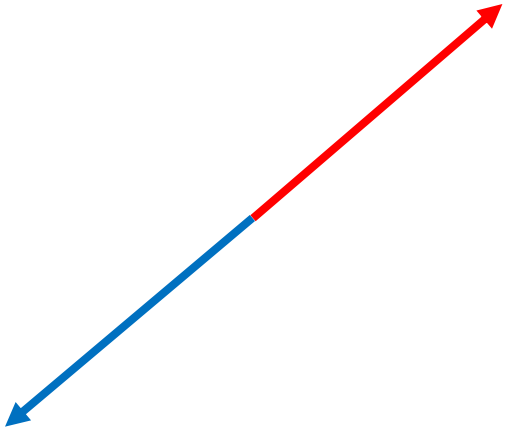
▲ [그림 5] Correlation value

3. Method

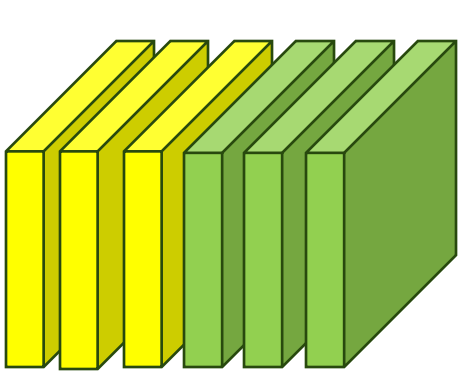


VS

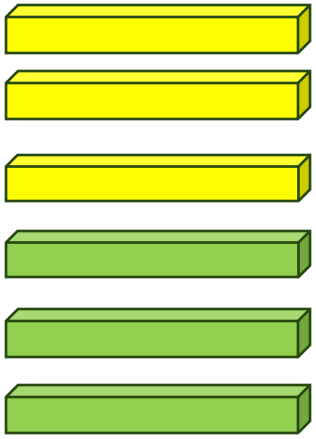




3. Method: Grouping



vectorization



$w_1, w_2, w_3, w'_1, w'_2, w'_3$

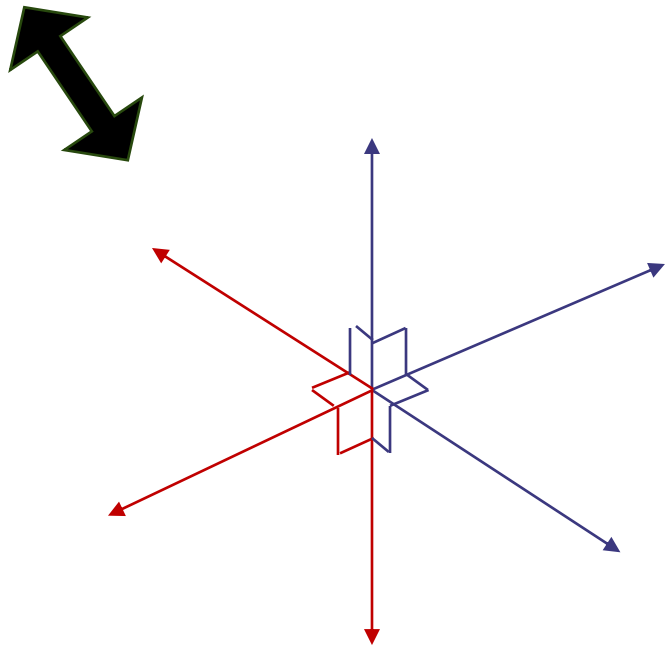
$w_1^T w_1$	$w_1^T w_2$	$w_1^T w_3$
$w_2^T w_1$	$w_2^T w_2$	$w_2^T w_3$
$w_3^T w_1$	$w_3^T w_2$	$w_3^T w_3$

$w'^T_1 w'_1$	$w'^T_1 w'_2$	$w'^T_1 w'_3$
$w'^T_2 w'_1$	$w'^T_2 w'_2$	$w'^T_2 w'_3$
$w'^T_3 w'_1$	$w'^T_3 w'_2$	$w'^T_3 w'_3$

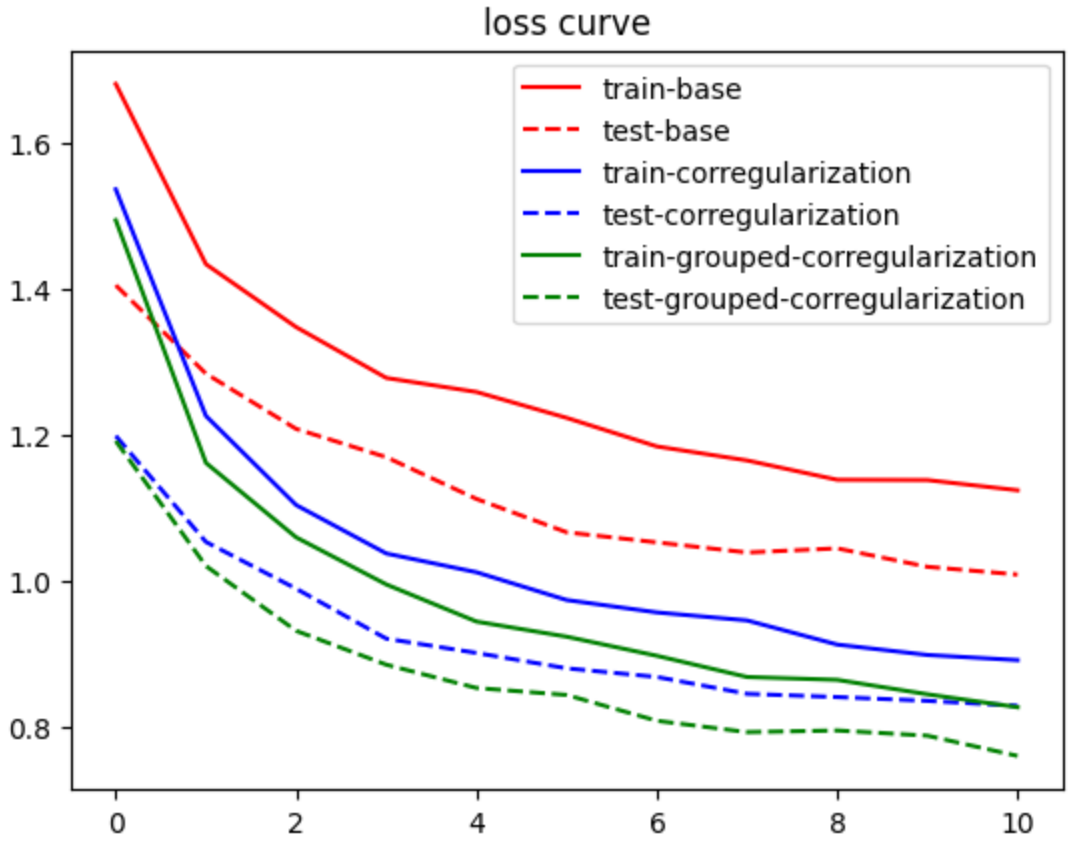
3. Method: Grouping

$$\sqrt{\sum_{i<j}(w_i^T w_j)^2} + \sqrt{\sum_{i<j}(w'^T_i w'_j)^2} + \left\|\sum_{i=1}^{l} w_i\right\|^2 \longrightarrow 0$$

Group1 Group2 Group polarization



3. Method: Second Experiment



▲ [그림 6] Loss curve (2)

4. Conclusion

- 각 Filter간 중복을 줄이는 것은 중요하다.

- 이를 위해 Gram Matrix를 통해 유사도를 낮출 수 있다.

- Group을 나눠 더 많은 벡터 공간을 표현해볼 수 있다.

5. Future Works

- Feature Vector의 차원과 개수의 관계

- 벡터의 내적을 세분화하여 벡터의 크기와 각도 관찰

- 현재 나와있는 다른 모델에 적용

- Group의 수를 더 늘려보기

- 생성 모델에 적용하여 Disentanglement 관점으로 관찰

4. Whisper를 활용한 위급상황 음성 인식 모델

발표자: 황태연

1. 서론

- **주제**: 가정 내 위급상황 음성 인식 모델 개발

- **기존 연구의 한계점**

1. In-Home Emergency Detection Using an Ambient Ultra-Wideband Radar Sensor and Deep Learning
2. 딥러닝을 이용한 청각장애인 위험 방지 및 STT 서비스의 통합 플렛폼 구현
3. 청각장애인을 위한 위험소리 분류 AI 헤드폰 설계

→ CNN, RNN과 같은 **단순한 딥러닝 기법**만을 활용함.

4. Monitoring In-Home Emergency Situation and Preserve Privacy using Multi-modal Sensing and Deep Learning
5. 비접촉 조작 및 위급 상황 탐지를 위한 엘리베이터 음성 인식 시스템
6. 스마트 홈 사용자를 위한 라이다 영상 오디오 센서를 이용한 인공지능 이상 징후 탐지 알고리즘

→ **데이터가 현저히 부족**하거나 **이진 분류(위험/안전)**만을 활용함.

1. 서론

- **주제**: 가정 내 위급상황 음성 인식 모델 개발

- **기존 연구의 한계점 극복 방법**

  1) CNN, RNN과 같은 **단순한 딥러닝 기법**만을 활용함.

  → Whisper와 같은 STT(Speech-to-Text) 모델을 함께 사용!

  - **Transformer 기반**의 모델이므로 성능이 더 좋을 것으로 예상
  - **음성 신호**(예: “살려주세요”)로 위험 감지 가능
  - **비음성 신호**(예: 화재 소리)로 위급 상황 판별 가능

  2) **데이터가 현저히 부족**하거나 **이진 분류(위험/안전)**만을 활용함.

  → **다양한 분류**가 존재하는 AI Hub 데이터 활용

1. 서론

- **주제**: 가정 내 위급상황 음성 인식 모델 개발

- **시나리오**

: 1인 가구, 독거 노인 등 위급 상황에 취약한 계층을 대상으로 함.

1) 가정 내에서 발생한 음성 신호가 **위급상황**인지 모델이 판단

2) 만약 위급상황이면 **가족(또는 119)에게** 음성 파일과 함께 **위급 상황 종류**와 위치 정보를 전달

2. 데이터셋

- AI Hub 위급상황 음성/음향 데이터

- 치안안전: 1. 강제추행, 2. 강도범죄, 3. 절도범죄, 4. 폭력범죄
- 소방안전: 5. 갇힘, 6. 전기사고, 7. 가스사고, 8. 화재, 9. 응급의료
- 자연재해: 10. 태풍/강풍, 11. 지진
- 사고 발생: 12. 낙상, 13. 붕괴사고
- 일반(위급): 14. 도움요청
- 일반(정상): 15. 실내, 16. 실외

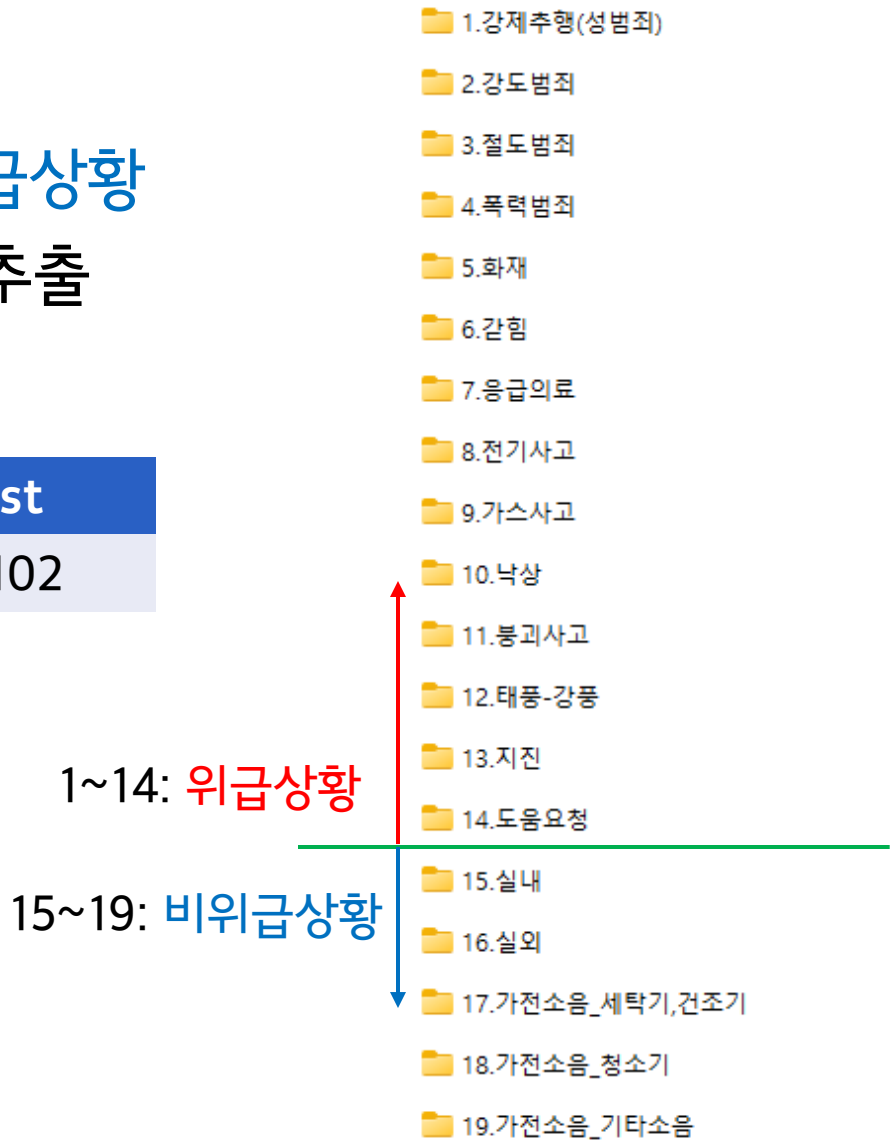
- 대조군: 소음 환경 음성인식 데이터

17. 가전소음_세탁기, 건조기
18. 가전소음_청소기
19. 가전소음_기타소음

2. 데이터셋

- **총 78,616개의 음원 (77.2GB)**
- 14가지의 위급상황 / 5가지의 비위급상황
- 각 상황마다 3,000개 이상의 음원 추출

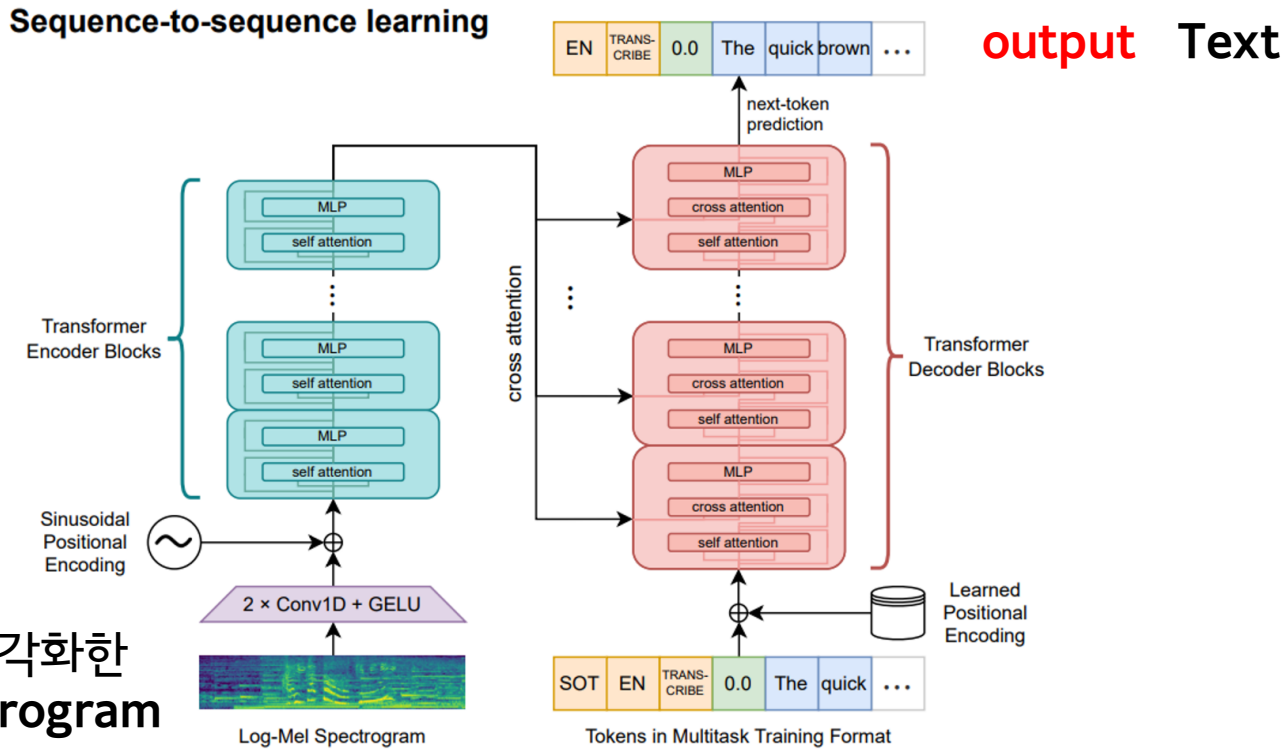
Training	Validation	Test
65,848 (99%)	666 (1%)	12,102



3. 제안 모델

- Background: **Whisper Model**

: 2022년 OpenAI에서 발표한 STT(Speech-to-Text)모델



▲ [그림 1] Whisper Model Architecture

3. 제안 모델

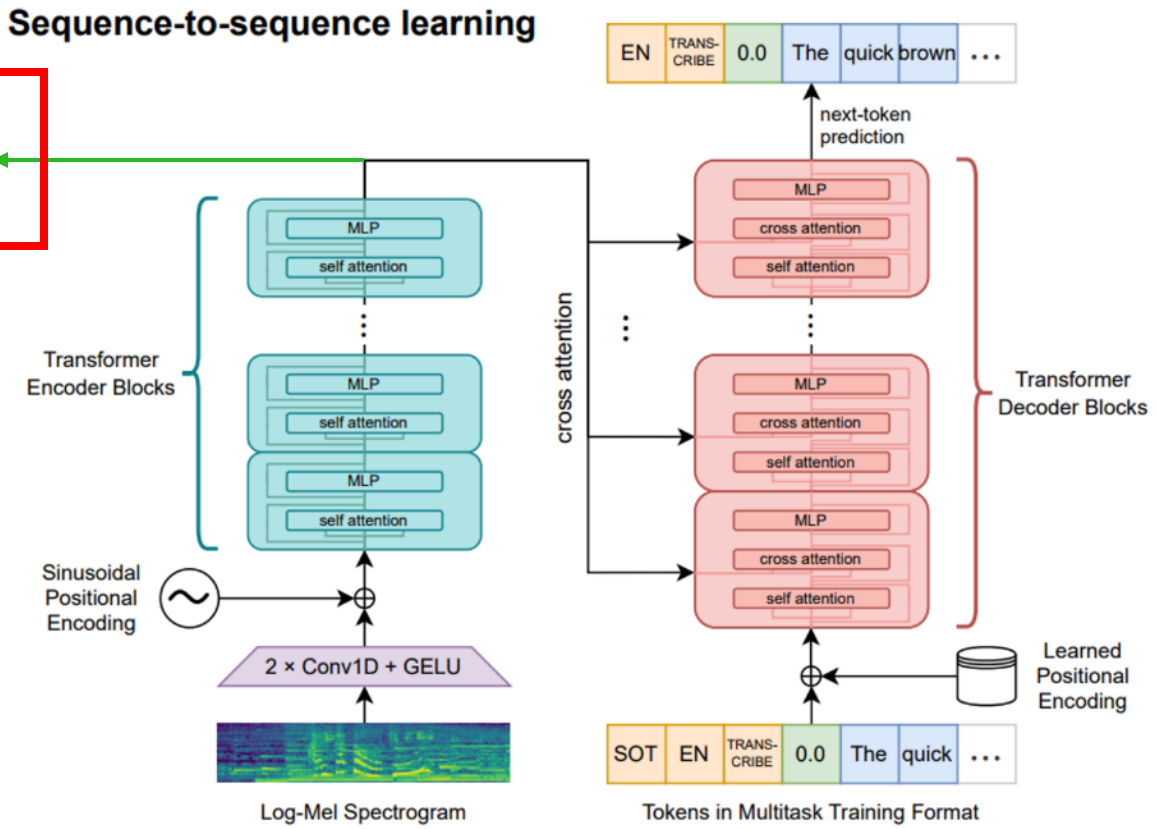
1. Whisper의 Encoder에 **MLP Head**를 추가

→ Encoder에서 **위급 상황 분류**

위급 상황 분류 ← **MLP Head**

- Encoder 뒤에 분류기를 추가해서 Fine-tuning 시도

- BERT, ViT 논문으로부터 아이디어를 얻음.

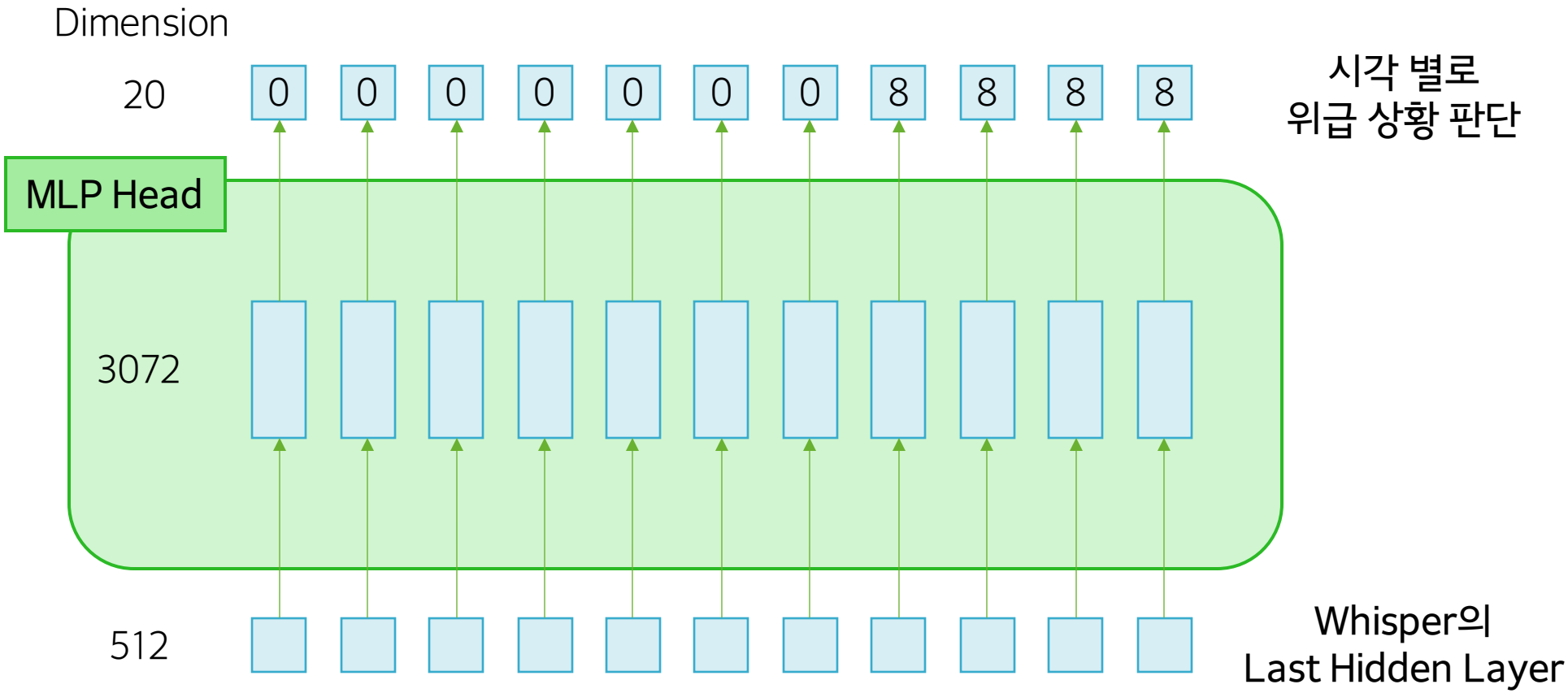


▲ [그림 2] Whisper Model의 변형

3. 제안 모델

1. Whisper의 Encoder에 **MLP Head**를 추가

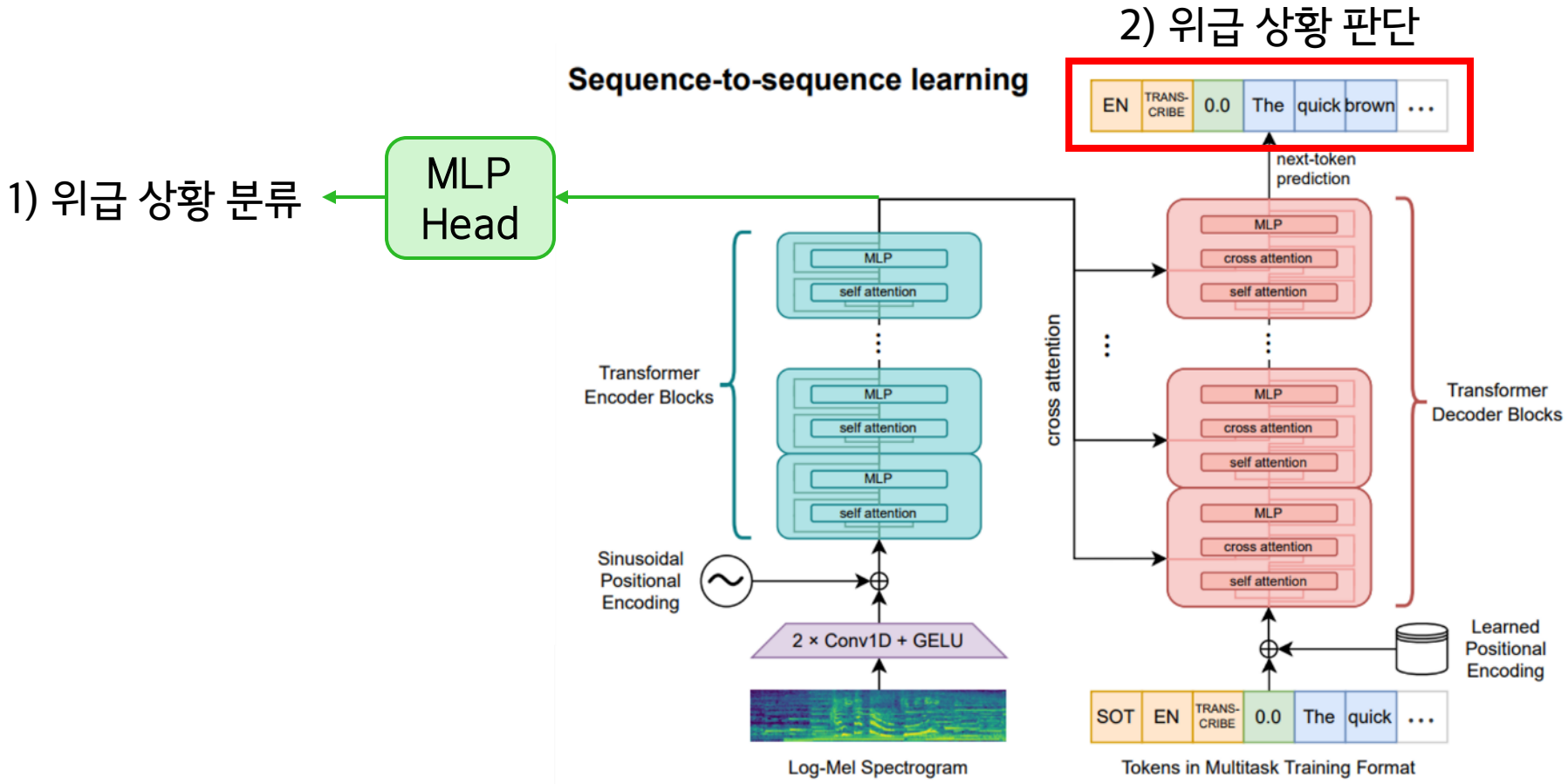
→ Encoder에서 **위급 상황 분류**



▲ [그림 3] MLP Head Architecture

3. 제안 모델

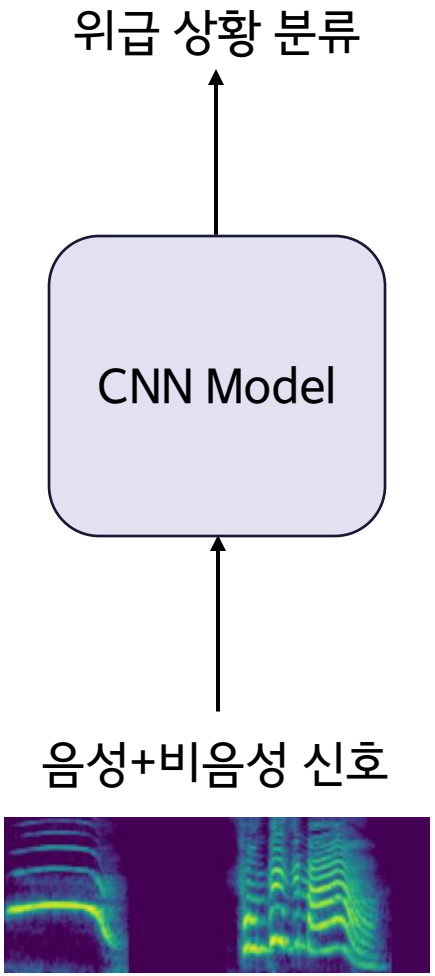
2. Whisper의 Decoder가 생성한 Text 정보로 **위급 상황 판단**



▲ [그림 2] Whisper Model의 변형

3. 제안 모델

3. 기존에 사용되던 간단한 CNN 모델로 **위급 상황 분류**



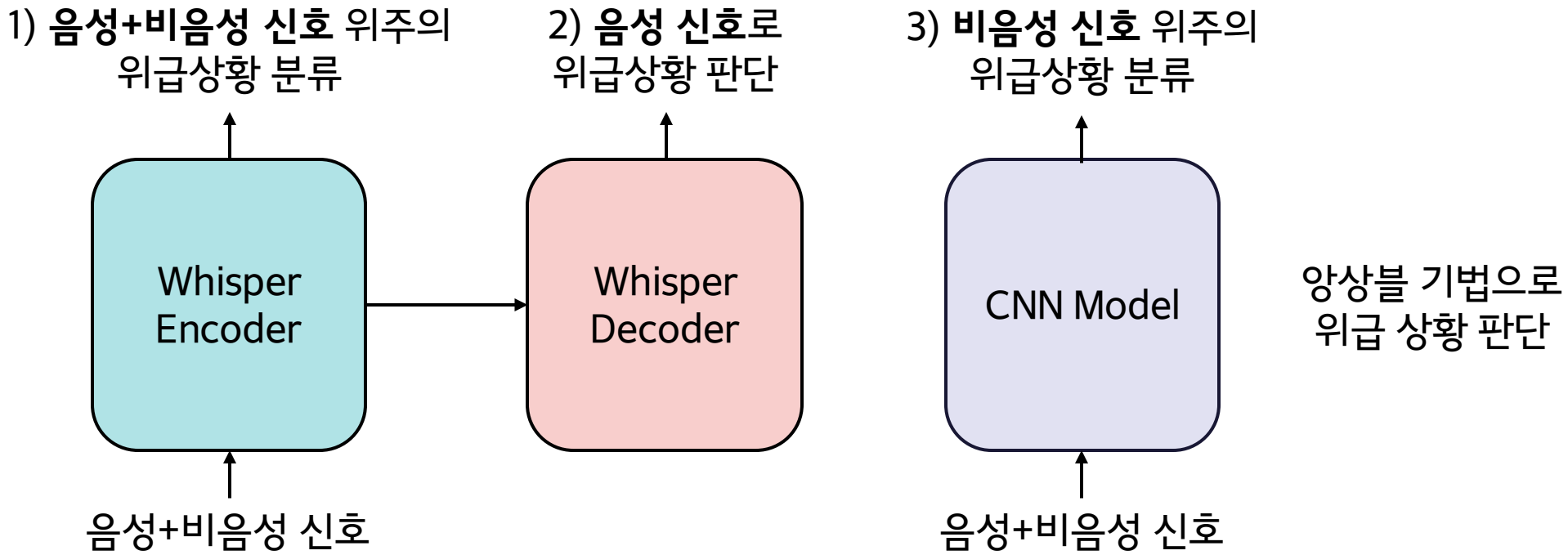
Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 8, 80, 1500]	80
ReLU-2	[-1, 8, 80, 1500]	0
MaxPool2d-3	[-1, 8, 40, 750]	0
Conv2d-4	[-1, 16, 40, 376]	1,168
ReLU-5	[-1, 16, 40, 376]	0
MaxPool2d-6	[-1, 16, 20, 188]	0
Conv2d-7	[-1, 32, 20, 94]	4,640
ReLU-8	[-1, 32, 20, 94]	0
MaxPool2d-9	[-1, 32, 10, 47]	0
Conv2d-10	[-1, 64, 10, 25]	18,496
ReLU-11	[-1, 64, 10, 25]	0
MaxPool2d-12	[-1, 64, 5, 12]	0
Conv2d-13	[-1, 128, 5, 6]	73,856
ReLU-14	[-1, 128, 5, 6]	0
MaxPool2d-15	[-1, 128, 5, 3]	0
Linear-16	[-1, 128]	245,888
Linear-17	[-1, 128]	245,888
ReLU-18	[-1, 128]	0
Dropout-19	[-1, 128]	0
Linear-20	[-1, 64]	8,256
Linear-21	[-1, 64]	8,256
ReLU-22	[-1, 64]	0
Dropout-23	[-1, 64]	0
Linear-24	[-1, 19]	1,235

Total params: 607,763
Trainable params: 607,763
Non-trainable params: 0

▲ [그림 4] CNN Model Architecture

3. 제안 모델

- **최종 모델**: **앙상블 기법** 이용
- 3개의 모델 중 **2개의 모델**이 위급상황이라고 판단 → **위급상황**으로 분류

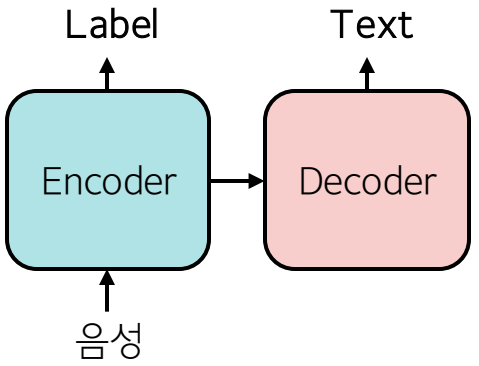


▲ [그림 5] 최종 모델 Architecture

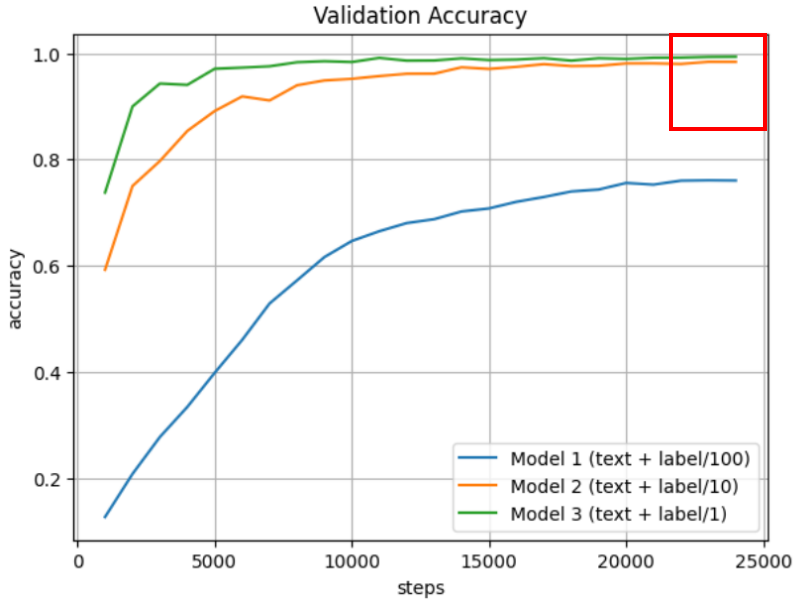
4. 실험 결과

1. Whisper Model (Encoder, Decoder)

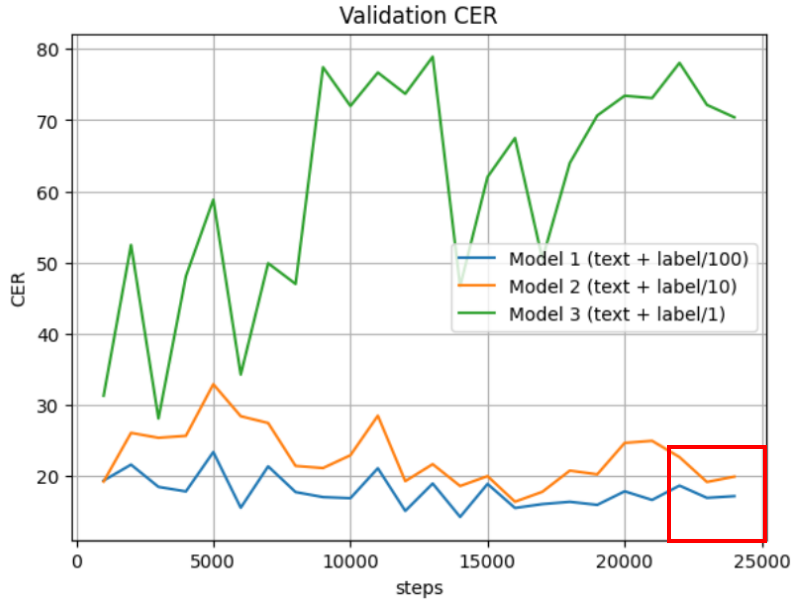
- 실험 방법: Text Loss와 Label Loss 비율 조절
  가장 적합한 모델로 선정
- 실험 결과



Accuracy는 높을수록 **Label을 잘 분류함**



CER은 낮을수록 **Text에 오류가 없음**



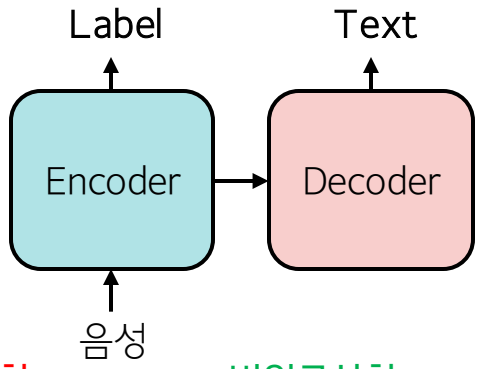
▲ [그림 6] Whisper Model Accuracy/CER

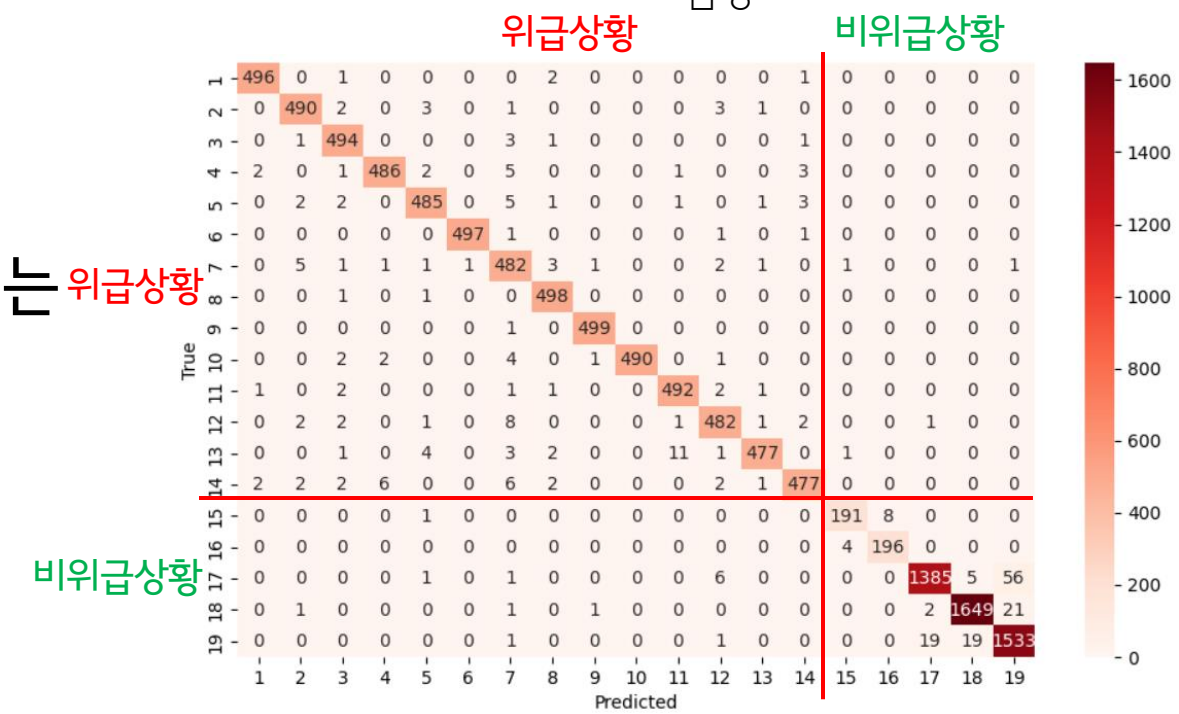
→ Model 2 선정

4. 실험 결과

1. Whisper Model (Encoder, Decoder)

- Model 2의 성능
  (1) Encoder
    - 19가지 위급상황을 **97.4%**의 정확도로 분류
    - 위급/비위급상황 이진 분류는 **99.85%**의 정확도로 분류

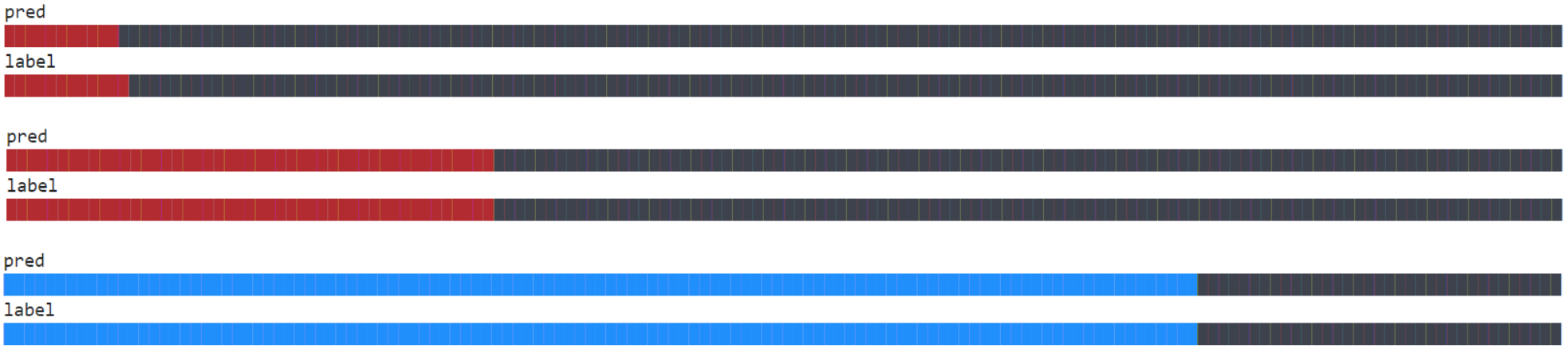




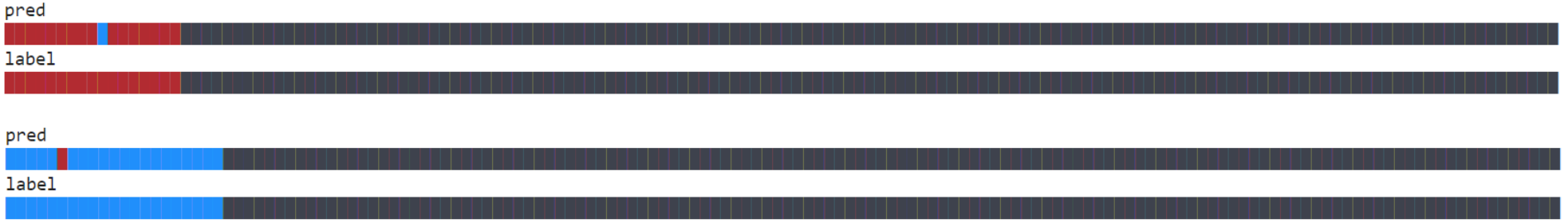
▲ [그림 7] Model 2의 Confusion Matrix

4. 실험 결과

- 시간대별 위급상황 분류 (**MLP Head, 정확도: 93.38%**)
- **빨간색**: 위급상황 / **파란색**: 비위급상황 / **검은색**: 무음 구간



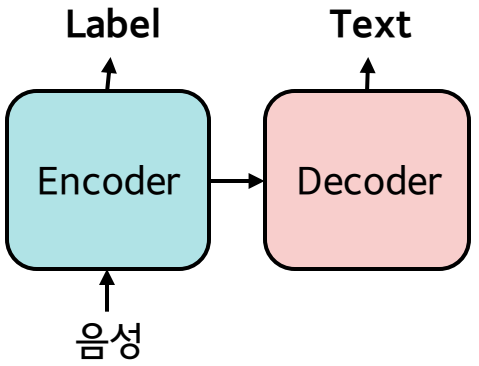
▲ [그림 8] 시간대별 위급상황 분류 (성공)



▲ [그림 9] 시간대별 위급상황 분류 (일부 실패)

4. 실험 결과

1. Whisper Model (Encoder, Decoder)



- Model 2의 성능

  (2) Decoder

  - CER을 12.63까지 낮춤
  - 텍스트를 통한 위급상황 이진 분류는 **96.21%**의 정확도로 분류

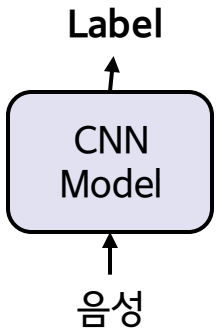
```
원본: 내 몸에 손대지 마    | 예측: 내 몸에 손대지 마
원본: 만지지 마            | 예측: 만지지 마만만
원본: 내 몸에 손대지 마    | 예측: 내 몸에 손대지 마
원본: 어딜 손대            | 예측: 어딜 손대
원본: 만지지 마            | 예측: 만지지 마
원본: 손대지 마            | 예측: 손대지 마
원본: 내 몸 만지지 마      | 예측: 내 몸 만지지 마
```

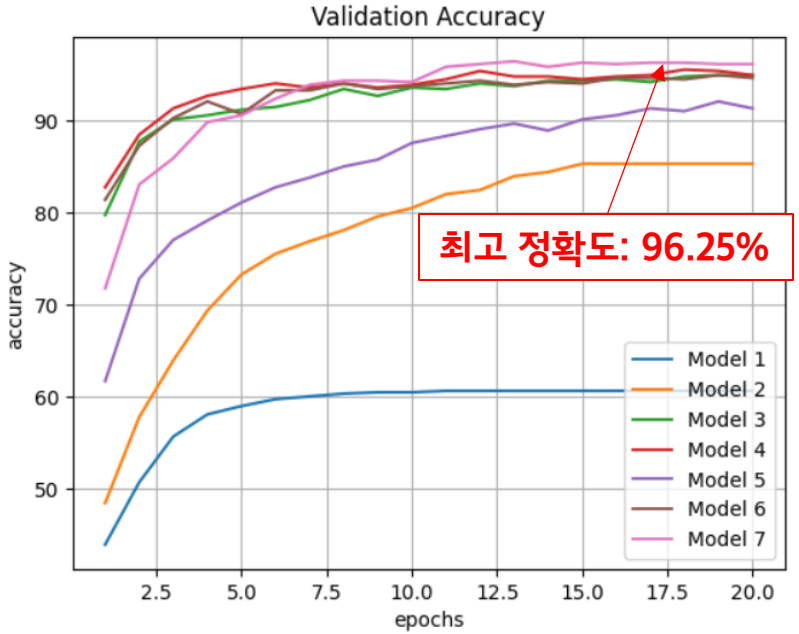
▲ [그림 10] Model 2의 Text 출력 결과

4. 실험 결과

2. CNN Model

- 실험 방법: 모델의 구조와 하이퍼파라미터 조정
  가장 정확도가 높은 모델 선정





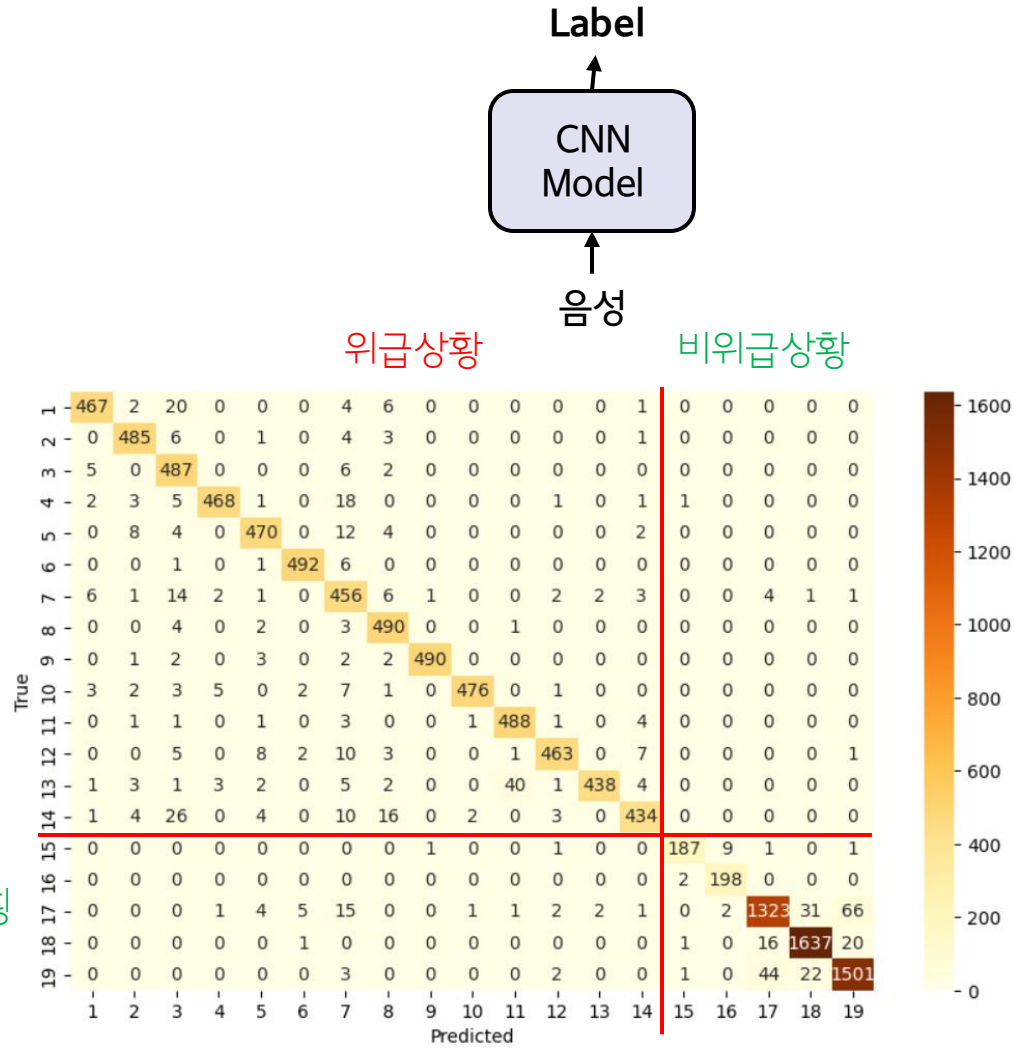
→ Model 7 선정

▲ [그림 11] CNN Model Validation Accuracy

4. 실험 결과

2. CNN Model

- Model 7의 성능
  - 19가지 위급상황을 **94.61%**의 정확도로 분류
  - 위급/비위급상황 이진 분류는 **99.60%**의 정확도로 분류



▲ [그림 12] Model 7의 Confusion Matrix

4. 실험 결과

2. CNN Model

- CNN Model이 실제로 비음성 신호 위주로 분석하는가?
- LRP(Layer-wise Relevance Propagation)를 이용한 분석

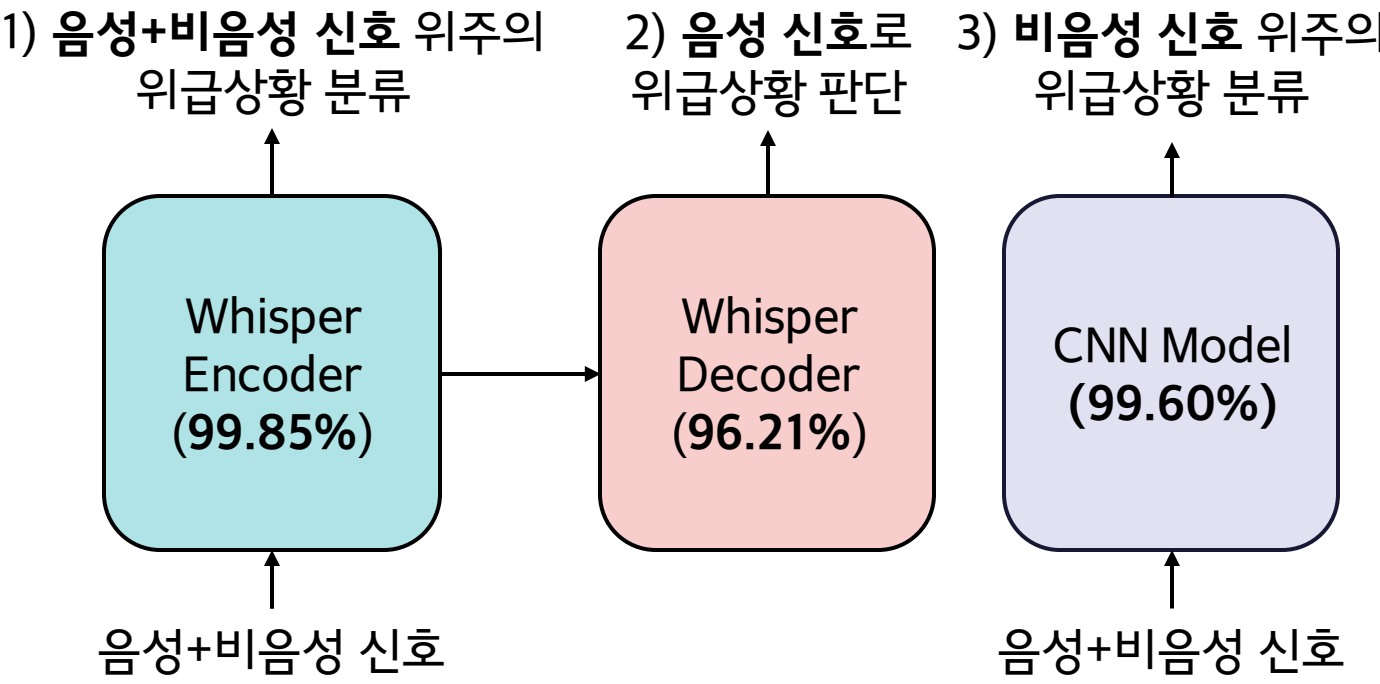
▲ [그림 13] Mel-spectrogram과 LRP output

4. 실험 결과

3. Ensemble Model

- 1) Whisper Encoder, 2) Whisper Decoder, 3) CNN Model
  3개의 모델을 결합한 **앙상블 모델**

→ 최종적으로
위급/비위급상황
이진분류에서
99.93% 정확도



▲ [그림 5] 최종 모델 Architecture

5. 향후 연구 방향

1. 시간대별 위급상황 판단을 위한 데이터 보강
   - 현재는 모든 위급/비위급상황이 0초부터 시작
   - 혼합된 데이터를 추가하는 방식으로 보완

2. Transformer XAI를 통한 모델의 설명력 증가
   - Whisper Encoder에서 음성 신호 위주로 추론하는지를 확인

논문 투고 일정

- 모든 연구는 2월 전에 마무리하여 논문을 작성 완료할 예정

1. Laplacian pyramid를 활용한 GAN 모델(강경헌)
2. Categorical Similarity Learning(강태욱)
   → IEEE Access에 3월 투고 예정

3. Corregularization: Filter간 Gram matrix를 이용한 Loss function design (소신)
   → NeurIPS에 5월 투고 예정

4. Whisper를 활용한 위급상황 음성 인식 모델
   → 한국음성학회에 4월 투고 예정

감사합니다

서울시립대학교 수리계산연구실
수학과 18 강경헌
수학과 18 강태욱
수학과 19 소 신
수학과 19 황태연