[Project #2]

Deep Learning for Image Classification with EM Algorithms

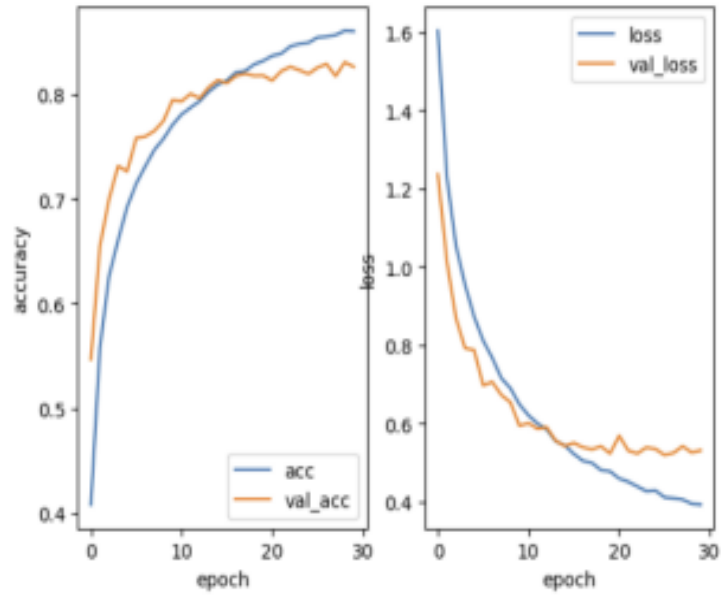
[오늘도(05)]

박상은 20211522
이찬회 20192923
민경준 20192899
함정규 20211556

Contents

Comparison of Hyper parameters

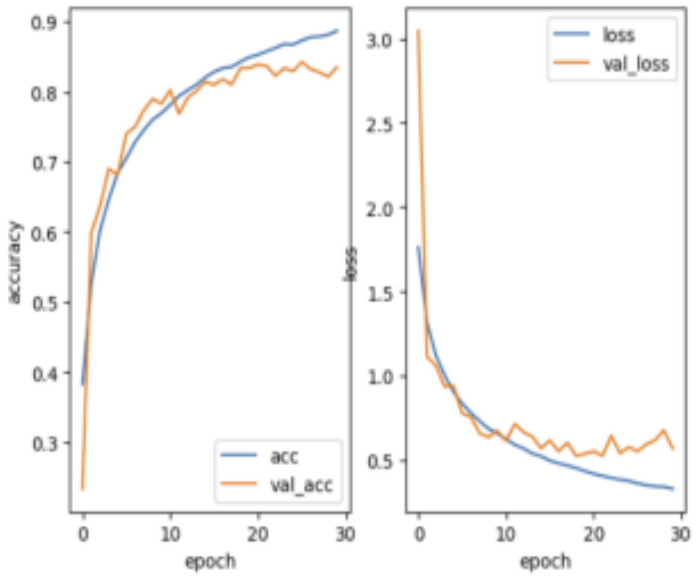
a. 기본 keras 의 sequential model



313/313 - 15s - loss: 0.5523 - acc: 0.8177 - 15s/epoch - 47ms/step

Loss: 0.5523 | acc: 0.8177

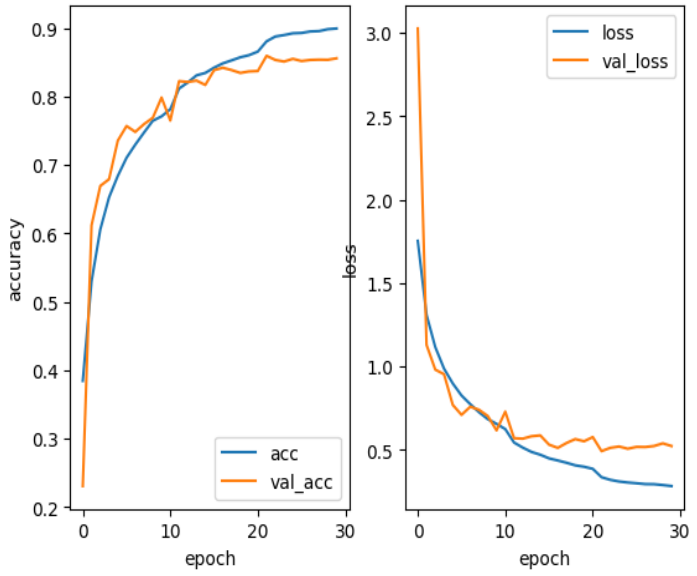
b. Batch normalization layer 추가



313/313 - 17s - loss: 0.5942 - acc: 0.8261 - 17s/epoch - 55ms/step

Loss: 0.5942 | acc: 0.8261

c. b + epoch 에 따른 learning rate 변형

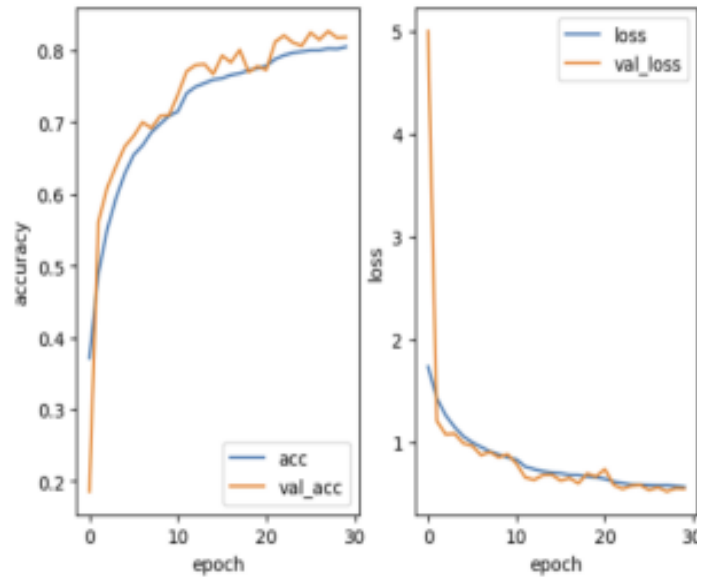


313/313 - 1s - loss: 0.5474 - acc: 0.8457 - 1s/epoch - 4ms/step

Loss: 0.5474 | acc: 0.8457

Comparison of Hyper parameters

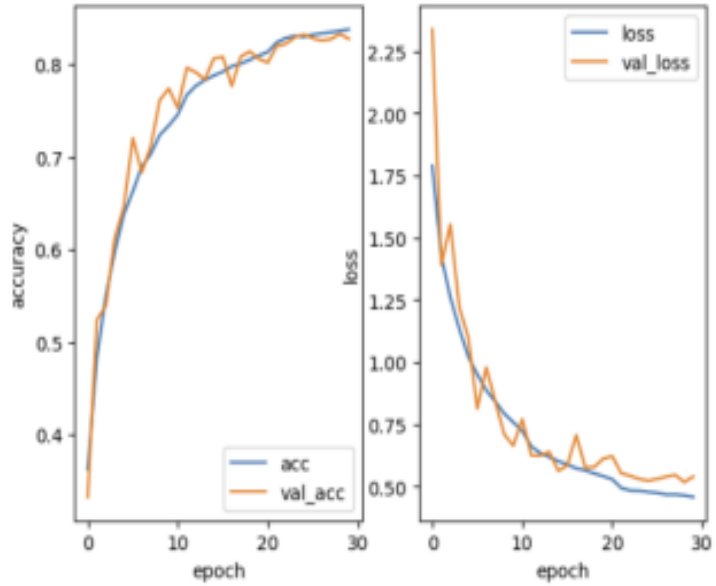
d. b + c + training image 의 rotation & flip



313/313 - 1s - loss: 0.5840 - acc: 0.8121 - 1s/epoch - 4ms/step

Loss: 0.5840 | acc: 0.8121

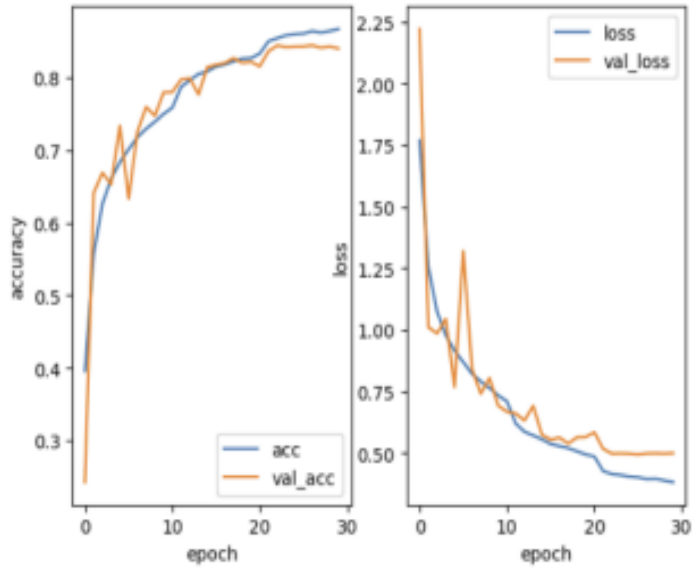
e. b + c + SGD optimize



313/313 - 1s - loss: 0.5686 - acc: 0.8180 - 909ms/epoch - 3ms/step

Loss: 0.5686 | acc: 0.8180

f. b + c + RMSprop optimize 의 모델



313/313 - 1s - loss: 0.5215 - acc: 0.8347 - 1s/epoch - 4ms/step

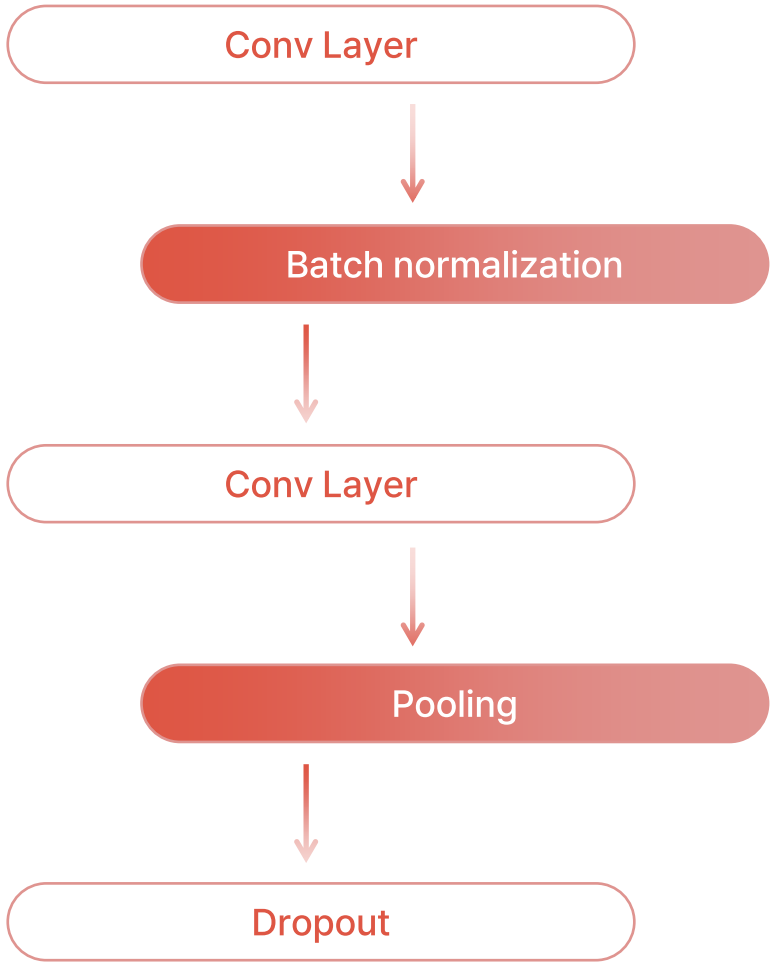
Loss: 0.5215 | acc: 0.8347

CNN layer

CNN layer

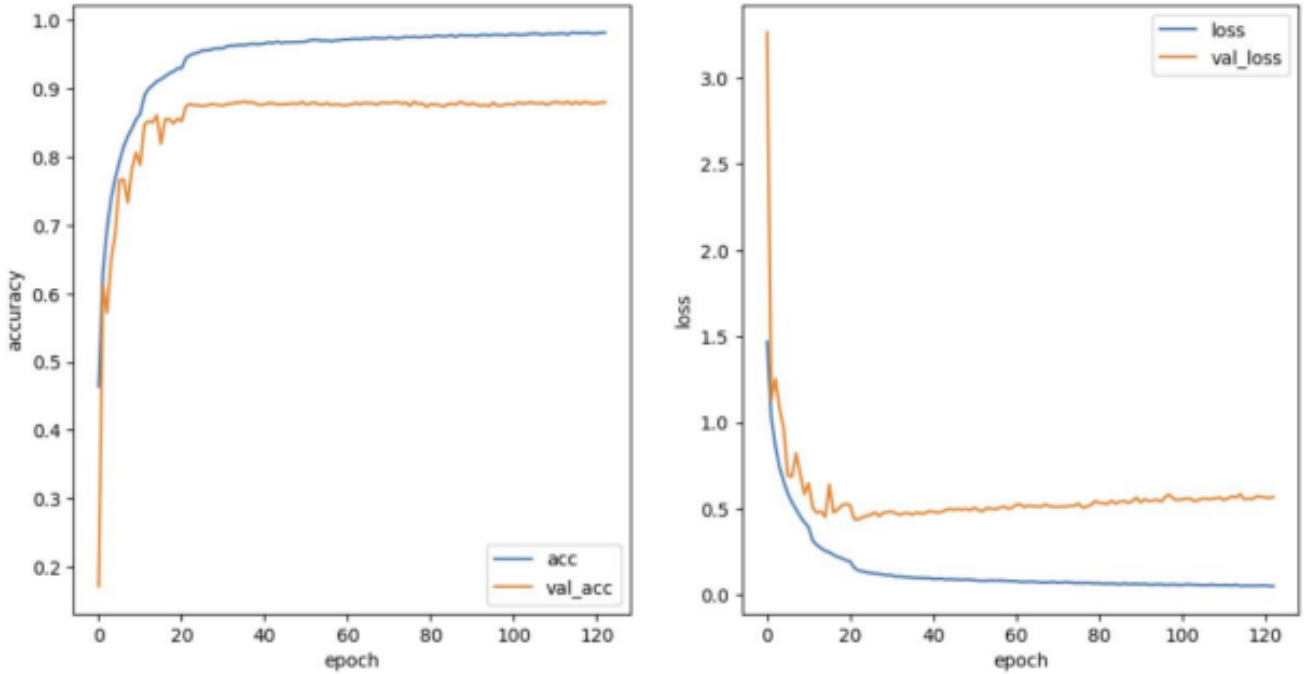
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 64)        1792

 batch_normalization (BatchN  (None, 32, 32, 64)       256
 ormalization)

 conv2d_1 (Conv2D)           (None, 32, 32, 64)        36928

 max_pooling2d (MaxPooling2D  (None, 16, 16, 64)       0
 )

 dropout (Dropout)           (None, 16, 16, 64)        0

 conv2d_2 (Conv2D)           (None, 16, 16, 128)       73856

 batch_normalization_1 (Batc  (None, 16, 16, 128)      512
 hNormalization)

 conv2d_3 (Conv2D)           (None, 16, 16, 128)       147584

 max_pooling2d_1 (MaxPooling  (None, 8, 8, 128)        0
 2D)

 dropout_1 (Dropout)         (None, 8, 8, 128)         0

 conv2d_4 (Conv2D)           (None, 8, 8, 256)         295168

 batch_normalization_2 (Batc  (None, 8, 8, 256)        1024
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 256)        0
 2D)

 dropout_2 (Dropout)         (None, 4, 4, 256)         0

 global_average_pooling2d (G  (None, 256)              0
 lobalAveragePooling2D)

 dense (Dense)               (None, 512)               131584

 dropout_3 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 693,834
Trainable params: 692,938
Non-trainable params: 896
_________________________________________________________________
```

Keras - sequential model



Accuracy /loss graph

accuracy/loss graph

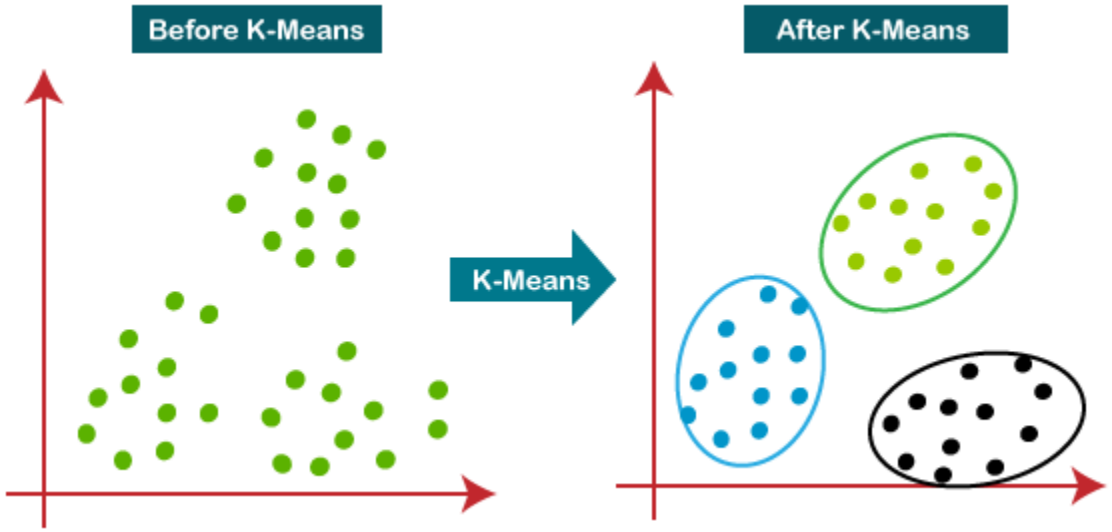



```
313/313 - 3s - loss: 0.4914 - acc: 0.8644 - 3s/epoch - 10ms/step
```

EM ALGORITHMS
FOR K-MEANS

loss function layer

K-MEANS



- K-means clustering 의 사용으로 class 의 중심을 효과적으로 탐색, loss 계산에 사용

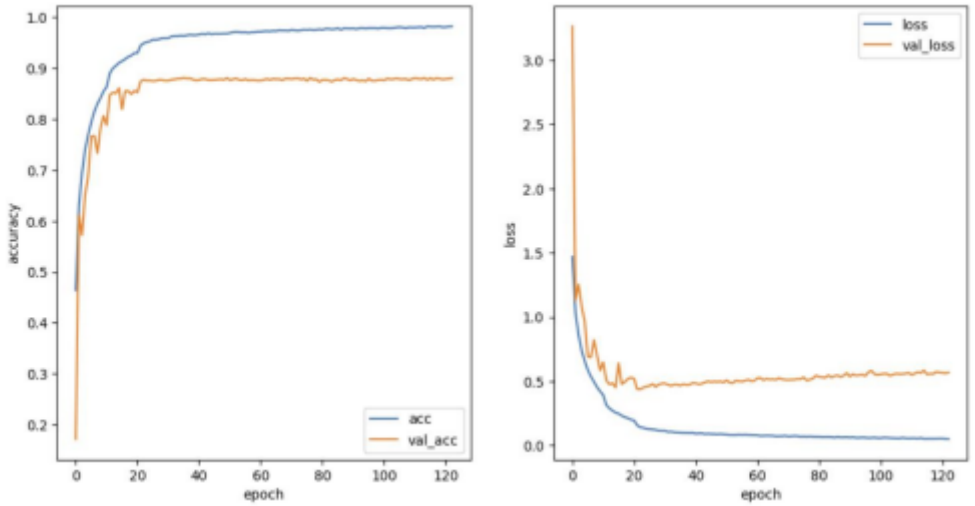
새로운 loss function 도입 →

- class 내부의 표본들 간 거리 최소화, 내부 분산 감소
- class 간의 거리는 최대화, 정확도 상승

Accuracy /loss graph

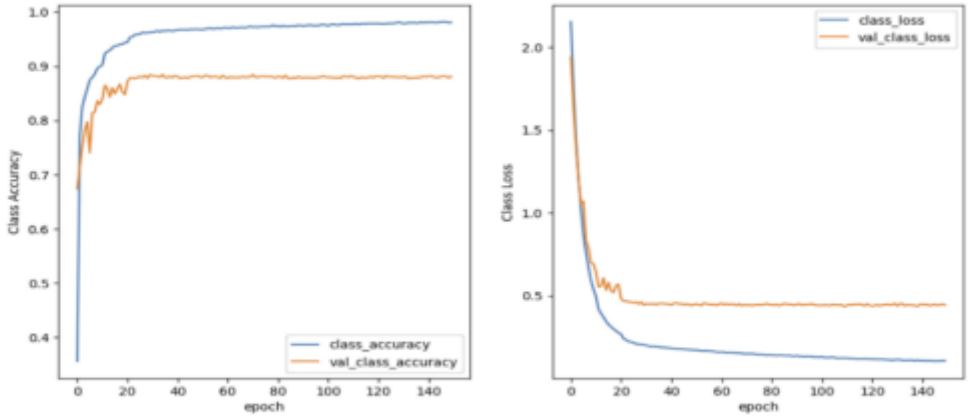
accuracy/loss graph comparison

acc: 0.864



<그림 7-1. 기존 CNN 모델의 accuracy/loss graph>

acc: 0.878



<그림 7-2. 새로운 loss function 적용한 CNN 모델의 accuracy/loss graph>

grad-CAM
Heating map

CAM visualization

Grad-CAM

GlobalAveragePooling2D

gradient

→ Heatmap

<그림 6. 각 class 별 heatmap>

t-SNE Visualization

t-SNE Process

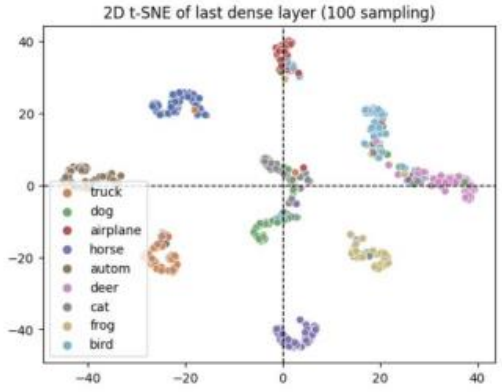
중간 layer output → 차원 축소를 위한 데이터 형태 변경

↓

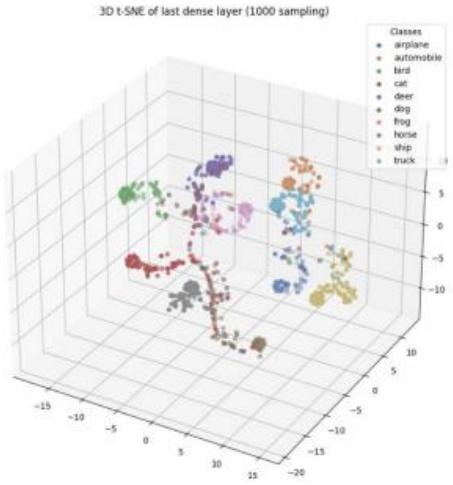
PCA 를 사용한 차원 축소 → sampling 과 t-SNE

↓

PCA 결과 sampling → t-SNE 출력



<그림 5-1. 2D t-SNE of last dense>

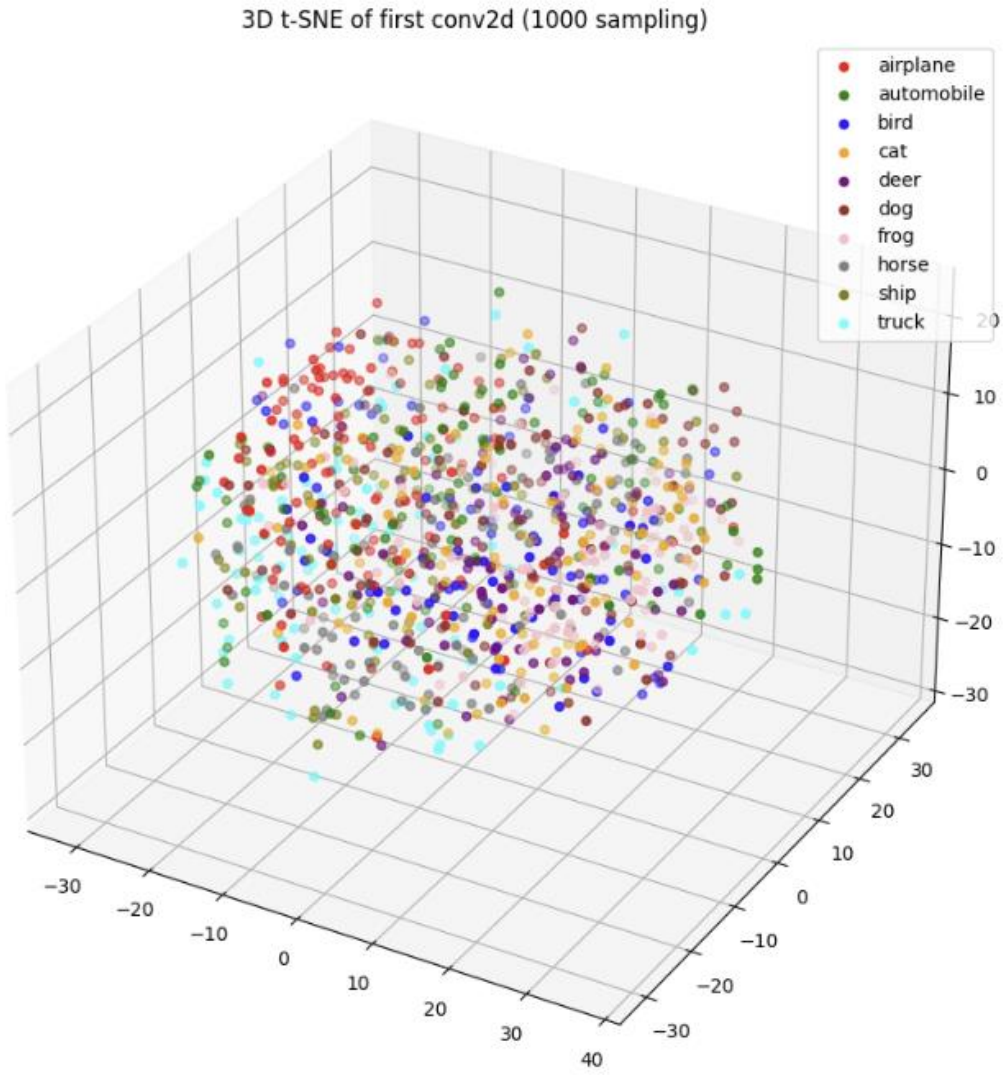


<그림 5-2. 3D t-SNE of last dense>

t-SNE Visualization

3D Surface Visualization (First conv2d layer) 2 / 19 layer

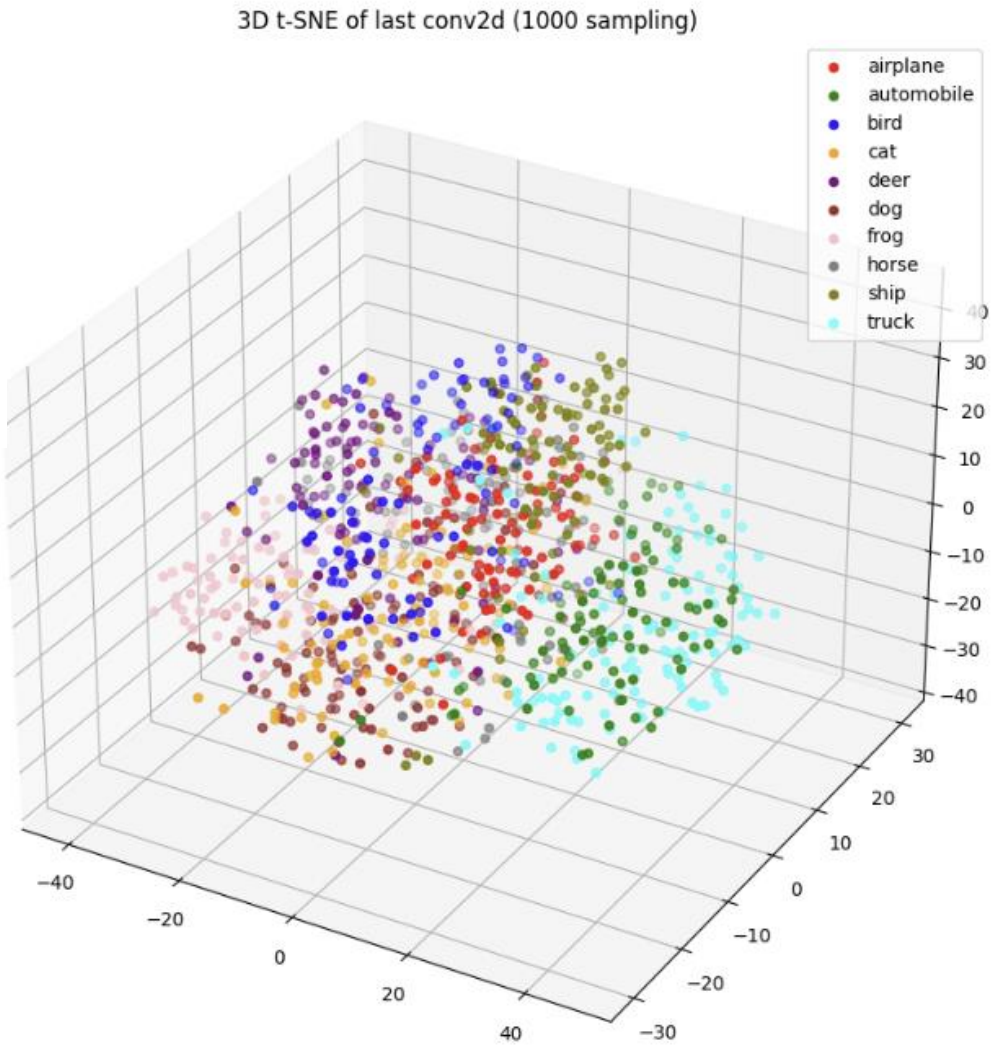
conv2d	input:	(None, 32, 32, 3)
Conv2D	output:	(None, 32, 32, 64)



t-SNE Visualization

3D Surface Visualization (last conv2d layer) 12 / 19 layer

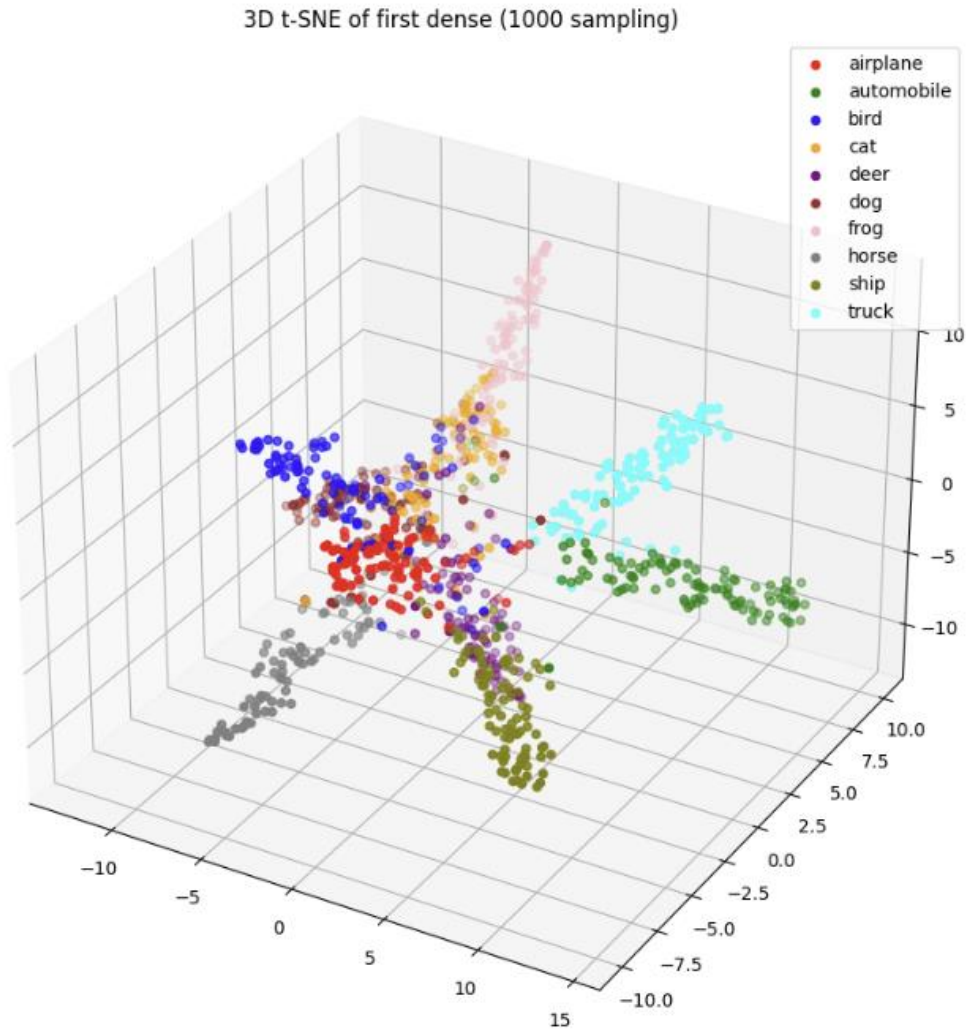
conv2d_4	input:	(None, 8, 8, 128)
Conv2D	output:	(None, 8, 8, 256)



t-SNE Visualization

3D Surface Visualization (First dense layer) 17 / 19 layer

dense	input:	(None, 256)
Dense	output:	(None, 512)

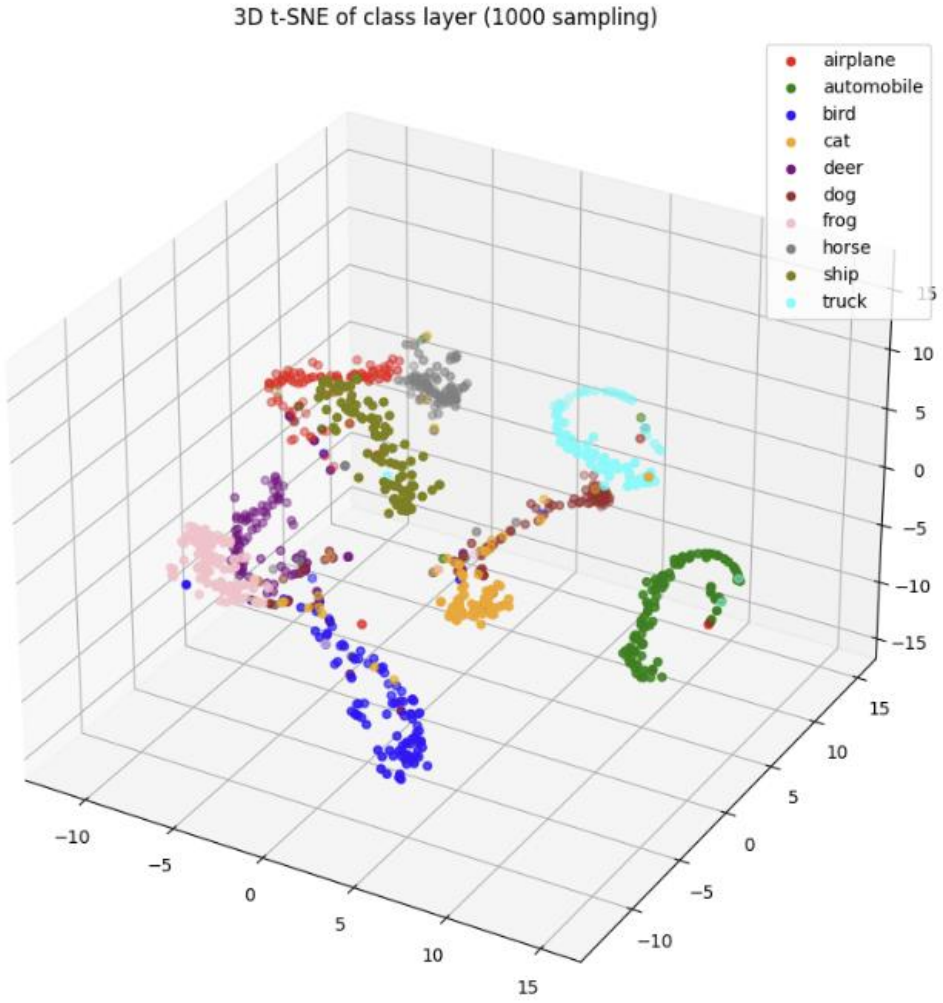


t-SNE Visualization

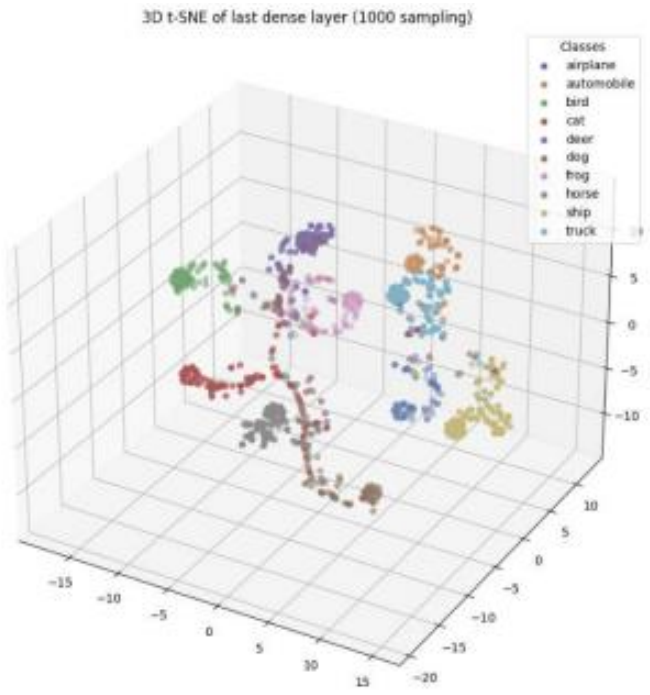
3D Surface Visualization (Last dense layer)

19 / 19 layer

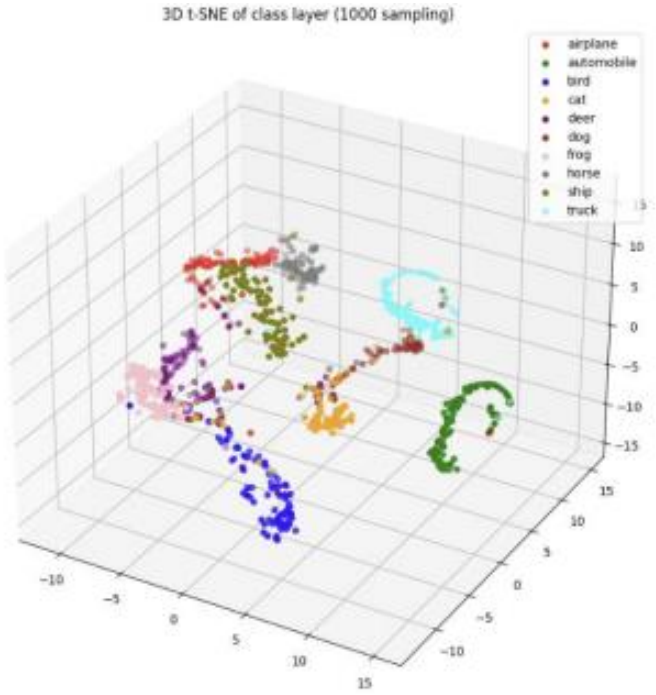
dense_1	input:	(None, 512)
Dense	output:	(None, 10)



t-SNE comparison



<그림 9-1. 기존 CNN 모델 3D t-SNE>



<그림 9-2. 새로운 CNN 모델 3D t-SNE>

Conclusion

- 최적화함수 변경, 학습률 조정, train 이미지 변환 등을 통해 최적의 모델 선택CNN 모델을 학습시킨 후 t-SNE 가시화 및 Grad-CAM 을 이용한 heat-map 가시화를 함으로써 클래스 간의 분류를 시각적으로 확인
- K-means clustering 후 새로운 loss function 을 설계, 적용한 모델 학습 후, 3D t-SNE 가시화를 4 개의 layer output 으로 진행하고 기존 CNN 모델과의 비교로 성능 향상을 확인