MNIST extended dataset을 이용한 CNN 모델 최적화 및 분석

최종발표(5조)

20192650 한수호
20180543 강구현
20192572 김민규
20192651 홍준기

2024.06.12

프로젝트 개요

MNIST extended dataset을 이용한 CNN 모델 최적화 및 분석

프로젝트 목표

자체 설계한 모델과 기존 CNN 기반 모델의 정확도를 비교 분석하여 **최적의 모델을 선정**하고,
EMNIST 데이터셋에 학습시켜 이미지 분류(Image Classification) 작업을 수행

프로젝트 수행

- CNN 기반의 기존 모델들과 자체적으로 설계한 모델을 비교
- 선정한 모델의 **옵티마이저(Optimizer), 활성화 함수, 잔차 블록(Residual Block)의 수, 학습률**을 조정하는 **Ablation** 형태의 실험을 계획하여 진행
- **EMNIST byClass**, byMerge, Balanced 데이터셋에 각각 학습시킨 모델을 세 가지 데이터셋에 학습하여 결과를 분석

프로젝트 개요

프로젝트 역할 분담

- 한수호 : Ours 모델 설계/학습 후 분석/개선, 결과 분석, 발표
- 강구현 : Baseline 모델 학습 후 결과 분석, Ours 모델 설계/학습
- 김민규 : Baseline 모델 분석/개선, Ours 모델 개선, 보고서 작성
- 홍준기 : 데이터셋 분석, Ours 모델 결과 분석, 보고서/발표 준비

프로젝트 수행 과정

데이터셋 분석

자체 모델 설계

하이퍼파라미터 튜닝

결과 분석

EMNIST byClass

가장 클래스 개수가 많고 클래스 불균형이 존재

이미지 특성

28*28 pixel, gray-scale

픽셀값(0~255)

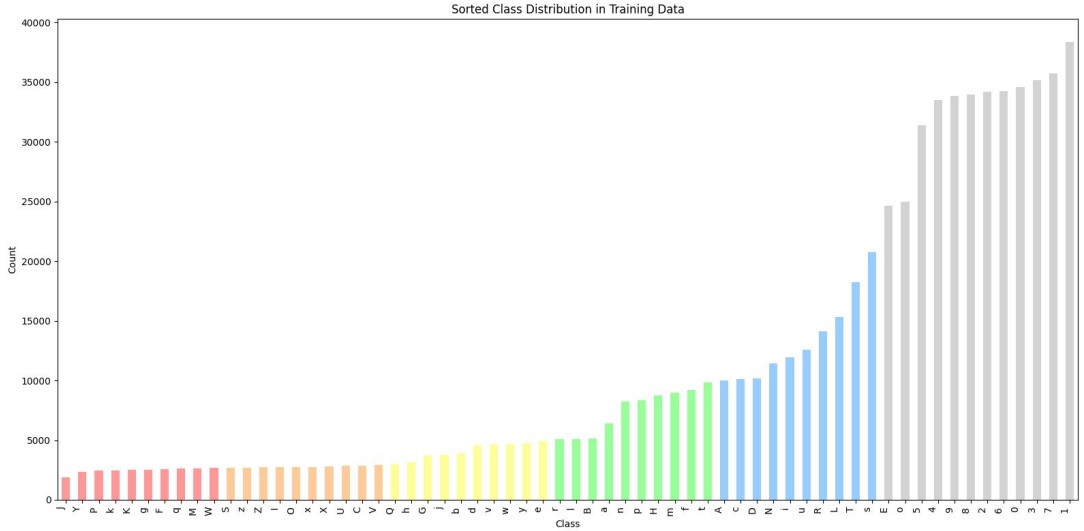
클래스 구성

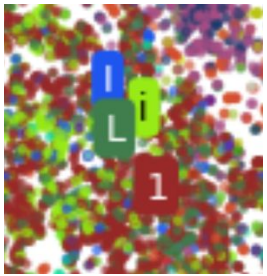
0~9, a-z, A-Z (62개 클래스)

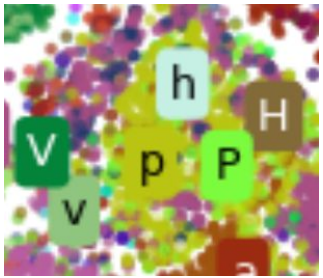
클래스 불균형

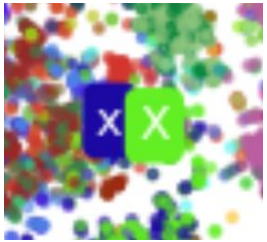
J: 1,896, Y: 2,365

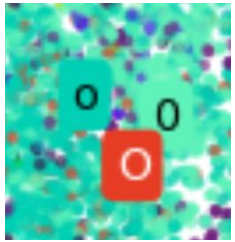
1: 38,374, 7: 35,754

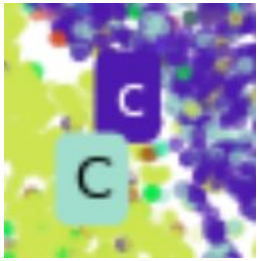


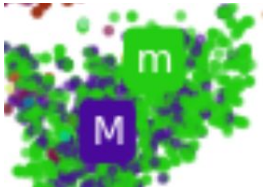


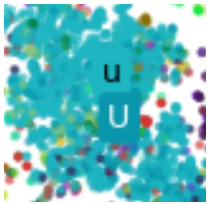


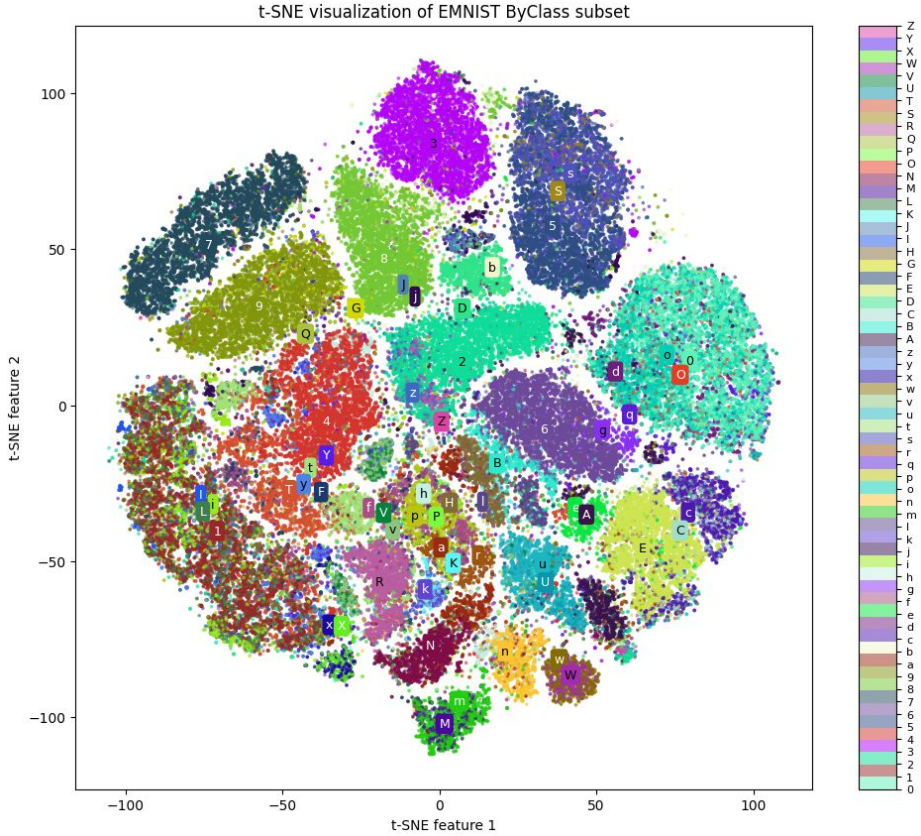











t-SNE visualization of EMNIST ByClass subset
t-SNE feature 1
t-SNE feature 2
100
50
0
−50
−100
−100
−50
0
50
100

Our 모델 비교

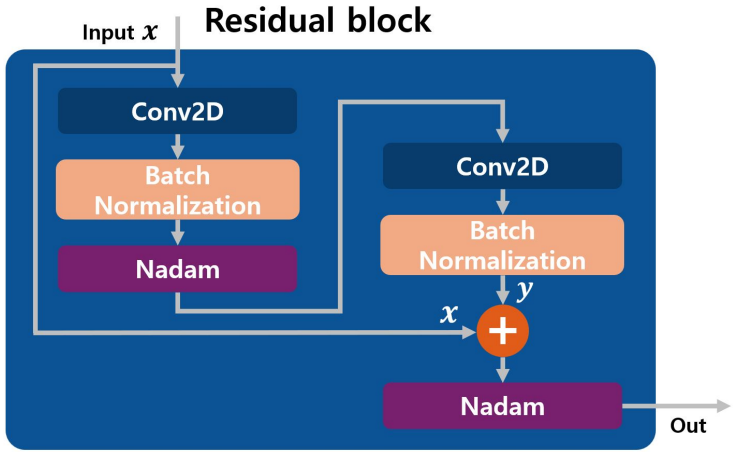
Model / Result	loss	accuracy(%)	Training Time(s)	params
WaveMix-Lite-128/7	-	**88.43**	-	9.6M
LeNet-5	0.3691	86.58	**272.90**	**64k**
ResNet-50	**0.3282**	87.76	3086.12	25.73M
Our Model(CNN)	0.3734	86.25	324.04	0.27M

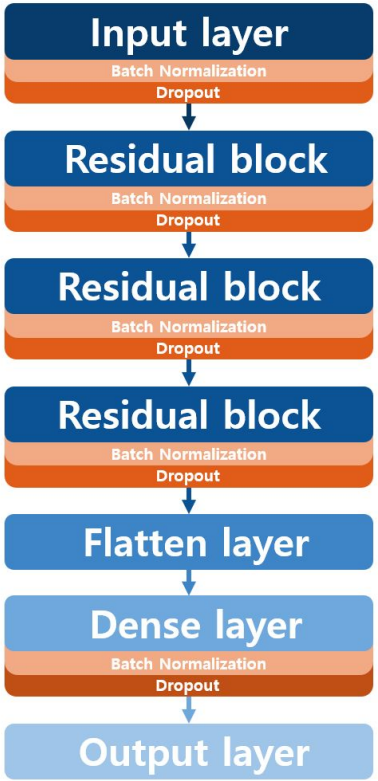
Our 모델 설계

Baseline 모델(LeNet5, ResNet-50), Our 모델: 합성곱 층만 사용한 모델

-> **새로운 모델** 설계: ResNet의 잔차 연결(Residual Connection)을 차용

Loss ftn: Cross Entropy Loss





callbacks

- **EarlyStopping**

  validation loss가 일정 epoch동안 감소하지 않을 때 학습을 중단

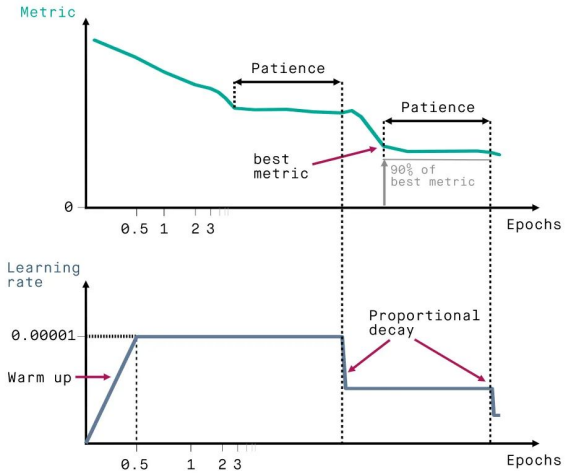
- **ReduceLROnPlateau**

  validation loss가 일정 epoch동안 감소하지 않을 때

  학습률을 절반으로 감소

- **ModelCheckpoint**

  모델의 중간점을 저장하는 콜백



하이퍼파라미터 변경

목적

EMNIST: 다양한 필기체와 복잡한 패턴, 비선형성, 노이즈 및 클래스 불균형
-> 하이퍼파라미터를 튜닝하여 모델의 성능을 향상

변경한 하이퍼파라미터

옵티마이저(Optimizer), 활성화 함수(Activation Function), 잔차 블록의 개수, 학습률

학습 조건

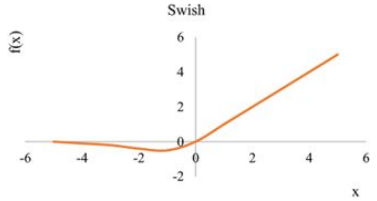
Epoch:50, batch_size: 500

옵티마이저(Optimizer)

Optimizer	Test Loss	Test Accuracy(%)	Training Time(s)
SGD	0.3396	87.17	1277.04
SGD(Nestrov)	0.3338	87.38	1286.94
Adagrad	0.3779	86.21	1289.94
RMSprop	0.3140	88.26	503.07
Adam	0.3086	88.35	605.06
Adamax	**0.3062**	88.43	775.57
Nadam	0.3082	**88.44**	784.48

Nesterov Accelerated Gradient

활성화 함수(Activation Function)

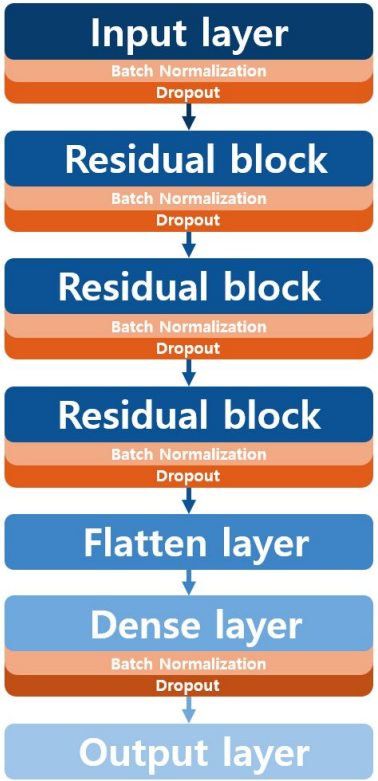


Activation Function	Test Loss	Test Accuracy(%)	Training Time(s)
ReLU	0.3328	87.20	242.82
Leaky ReLU	0.3321	87.40	244.90
eLU	0.3289	87.47	246.59
SeLU	0.3384	87.24	245.21
Sigmoid	0.4001	85.20	242.76

Activation Function	Test Loss	Test Accuracy(%)	Training Time(s)
tanh	0.3395	87.20	244.21
softmax	0.5600	83.41	273.51
softsign	0.3476	87.02	243.27
swish	**0.3163**	**87.91**	286.79
GeLU	0.3163	87.85	321.97

잔차 블록의 개수

# of Residual Blocks	Test Loss	Test Accuracy(%)	Training Time(s)
1 block	0.3190	88.24	1108.55
2 blocks	**0.3136**	88.37	1413.16
3 blocks	0.3264	**88.37**	1672.99
4 blocks	0.3303	88.30	1863.36



학습률(1차)

Learning Rate	Test Loss	Test Accuracy(%)	Training Time(s)
1e-2	0.3223	88.02	652.85
1e-3	0.3120	**88.33**	758.32
1e-4	**0.3071**	88.32	1382.78
1e-5	0.3328	87.38	1618.95
1e-6	0.4499	84.48	1634.89

학습률(2차)

Learning Rate	Test Loss	Test Accuracy(%)	Training Time(s)
1e-4	0.3071	88.32	1382.78
2e-4	**0.3067**	**88.41**	1005.71
3e-4	0.3092	88.37	932.36
4e-4	0.3090	88.26	803.46
5e-4	0.3091	88.28	709.32

Learning Rate	Test Loss	Test Accuracy(%)	Training Time(s)
6e-4	0.3109	88.37	808.81
7e-4	0.3098	88.24	673.69
8e-4	0.3106	88.26	704.26
9e-4	0.3107	88.22	669.50
1e-3	0.3100	88.22	671.94

결과 분석

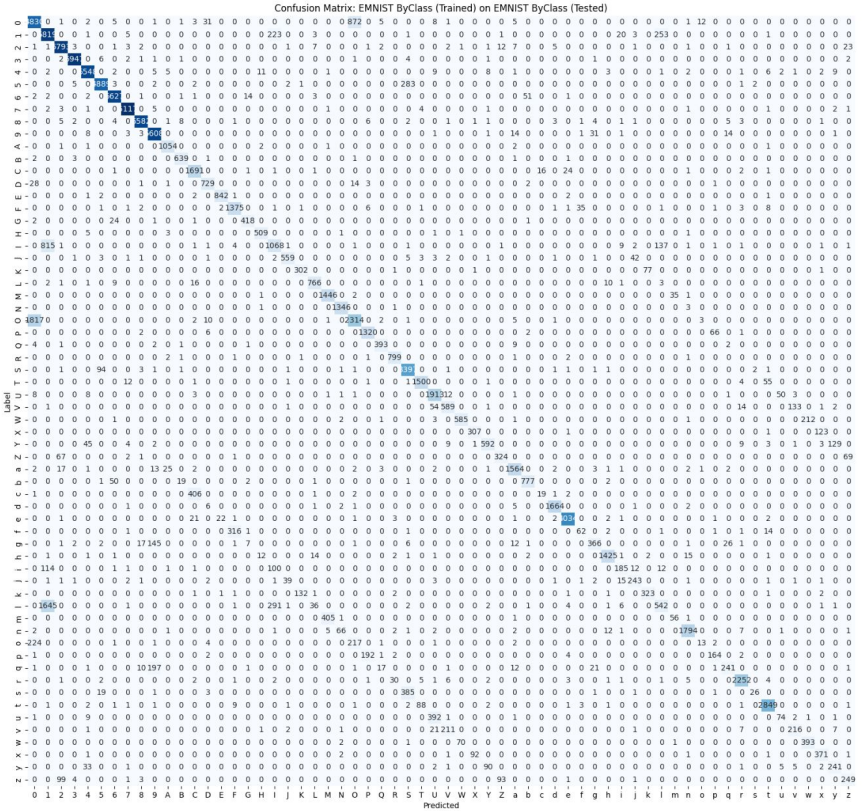
Model / Result	loss	accuracy(%)	Training Time(s)	params
WaveMix-Lite-128/7	-	**88.43**	-	9.6M
LeNet-5	0.3691	86.58	**272.90**	**64k**
ResNet-50	0.3282	87.76	3086.12	25.73M
Our Model(CNN)	0.3734	86.25	324.04	0.27M
Our Model(Res) (fine-tuned)	**0.3076**	88.27	811.98	1.77M

결과 분석 - **Confusion Matrix**

{0, O}, {1, i, l}, {9, g, q} 등

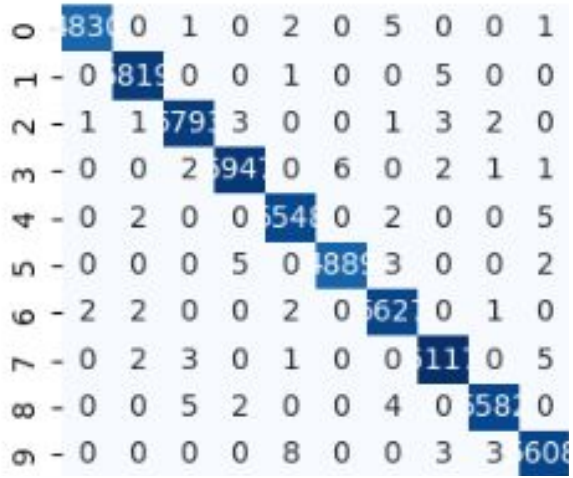
비슷한 형태의 숫자와 알파벳들에

대한 예측이 어려움

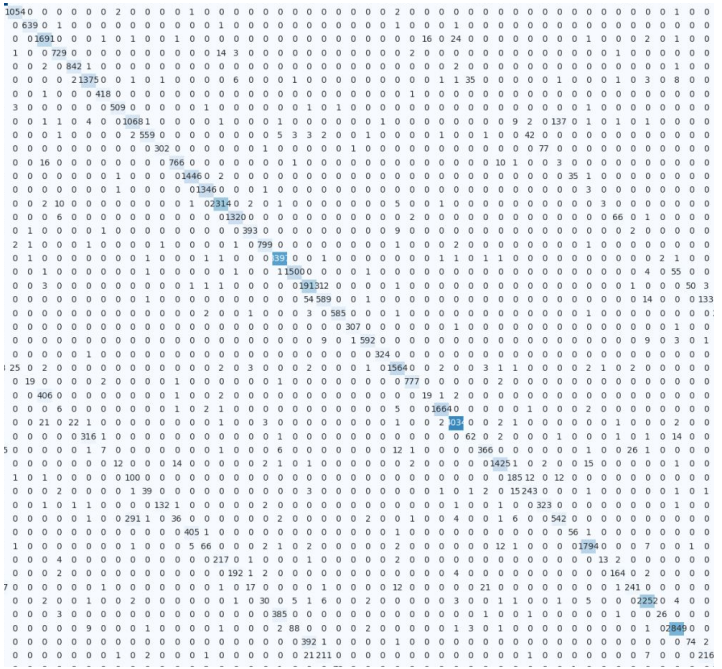


결과 분석 - Confusion Matrix

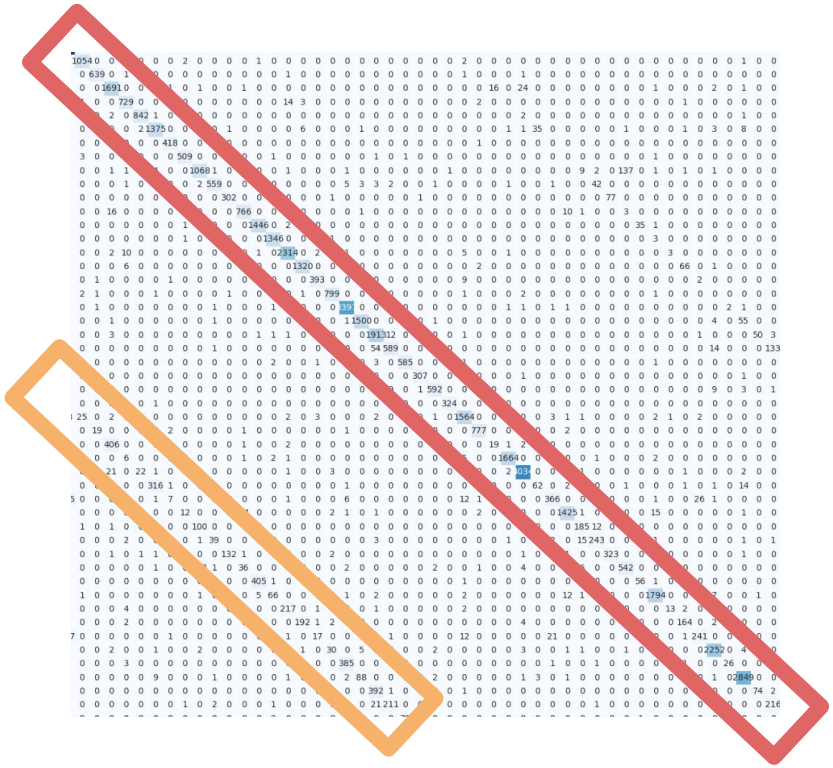
숫자(0~9)



알파벳(a-z, A-Z)



결과 분석 - Confusion Matrix

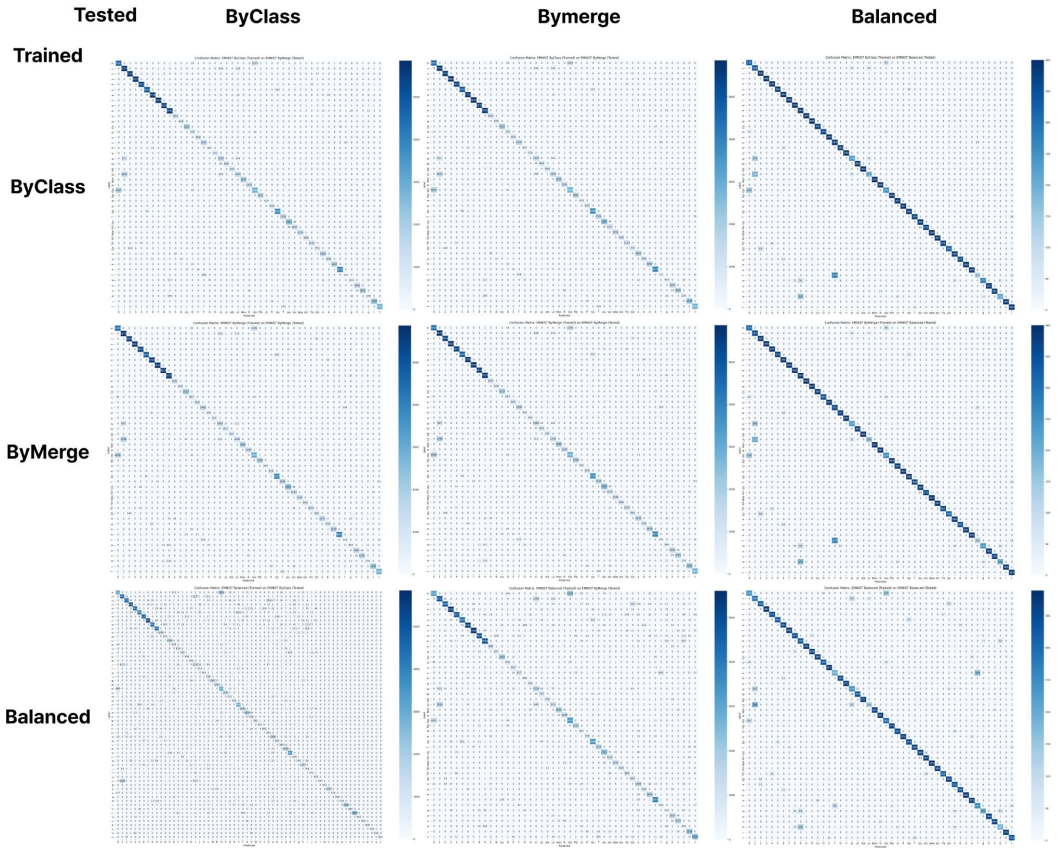


대문자와 소문자의 구별이 어려운 것을 확인

결과 분석 - 학습 데이터셋, 테스트 데이터셋 변경

Accuracy(%) / F1 Score / Inference Time(s)		Tested Dataset		
		byClass	byMerge	Balanced
Trained Dataset	byClass	88.27 / 0.8756 / 28.20	92.08 / 0.9167 / 22.81	90.53 / 0.9005 / 2.63
	byMerge	92.03 / 0.9150 / 22.75	91.42 / 0.9103 / 24.05	90.30 / 0.8992 / 3.46
	Balanced	86.53 / 0.8643 / 23.13	86.44 / 0.8655 / 20.53	86.99 / 0.8679 / 3.62

결과 분석 - Confusion Matrix



Lesson Learned

문제점 분석 - class_weight

TensorFlow의 class_weight를 도입
-> 각 클래스에 다른 가중치를 부여함으로써 소수 클래스의 학습을 강화

가중치 부여가 과도하게 이루어져 **소수 클래스에 대한 과적합 발생**

$$L = \frac{1}{n} \sum_{i=1}^{n} w_i L_i$$

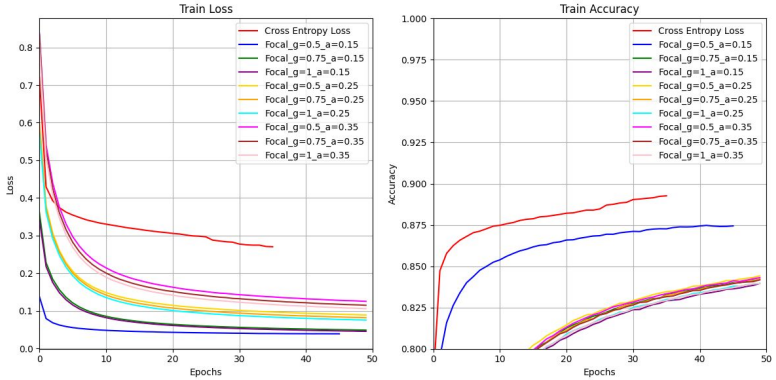
Lesson Learned

문제점 분석 - focal loss

$$\text{Focal Loss}(p_t) = -\alpha_t(1-p_t)^\gamma log(p_t)$$

클래스 불균형을 해결하고 어려운 예제에 더 많은 가중치를 부여하는 focal loss

-> 모델이나 하이퍼파라미터 설정이 focal loss의 장점을 충분히 활용하기에 적합하지 않았을 가능성



Loss Function	accuracy(%)	Training Time(s)
Cross Entropy Loss	**88.34**	1149.58
Focal Loss beta=1	87.88	1629.38
Focal Loss beta=2	86.97	1642.57
Focal Loss beta=3	86.80	1614.12

Lesson Learned

프로젝트의 의의

- 인공신경망 수업에서 **배운 내용**에 대한 **깊은 이해**
- **문제를 해결하는 데에 적합한 방법을 찾는 것**에 대한 방안
    - 데이터 전처리 단계에서의 샘플링 기법이나 추가적인 정규화 기법을 도입
- 협력과 논의를 통한 효과적인 학습

References

MNIST: Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017, May). EMNIST: Extending MNIST to handwritten letters. In 2017 international joint conference on neural networks (IJCNN) (pp. 2921-2926). IEEE.

LeNet: LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

ResNet: He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

ResNet50 : Zhang, L., Bian, Y., Jiang, P., & Zhang, F. (2023). A transfer residual neural network based on ResNet-50 for detection of steel surface defects. Applied Sciences, 13(9), 5260.

Convolution Layer: O'shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

Wavemix: Jeevan, P., & Sethi, A. (2022). Wavemix: resource-efficient token mixing for images. arXiv preprint arXiv:2203.03689.

Swish: Fatima, A., & Pethe, A. (2022). Periodic analysis of resistive random access memory (RRAM)-based swish activation function. SN Computer Science, 3(3), 202.