

EMNIST Classifier: From Research to Real-World Application

이요원 박서영 박주은 최서영 허성현

1 프로젝트 개요 및 목표

본 프로젝트에서는 MNIST Extended 데이터셋을 활용하여 직접 설계한 CNN 모델과 사전 학습된 CNN 모델을 학습시켜 성능을 평가한다. 특히, LeNet-5와 ResNet-50을 baseline model¹로 사용하여 학습 결과를 비교하였다. Hyperparameter tuning을 통해 최적의 학습 결과를 도출하는 것을 목표로 한다.

MNIST Extended 데이터셋을 분석한 후, LeNet-5와 ResNet-50 모델을 사용하여 학습을 진행하고, 두 모델의 성능을 비교 평가한다. 이 과정에서 정확도, 학습 시간, 추론 시간의 조합을 고려하여 모델의 효율성을 분석하였다.

다양한 CNN 모델 또는 직접 설계한 CNN 모델을 선정하고, 이를 학습시켜 결과를 분석한다. CNN 모델 선정 과정과 그 결과를 실험 데이터와 분석 결과를 바탕으로 상세히 제시한다. 최종적으로, 정확도와 추론 시간의 균형을 고려한 최적의 CNN 모델을 도출하는 것을 목표로 한다.

2 수행 계획 및 역할 분담

2.1 수행 계획

Table 1에 명시해둔 계획에 따라 프로젝트를 수행했다.

2.2 역할 분담

Table 2에 명시해둔 역할에 따라 프로젝트를 수행했다.

¹연구에서 새로운 모델이나 방법의 성능을 객관적으로 평가하기 위한 기준으로 사용되는 모델이나 알고리즘을 뜻함.

일시	프로젝트 수행 계획
04.27	프로젝트 수행 계획 수립 및 역할 분담
04.30	[데이터 준비 및 전처리] MNIST extended dataset 분석 및 dataset 분류
05.08	[baseline model 모델 활용 및 학습/평가] baseline model 학습 및 결과 분석 - LeNet-5와 ResNet-50 모델
05.12	[baseline model 최적화] baseline model hyperparameter tuning
05.15	[직접 설계 cnn 모델 개발 및 학습/평가] 다수의 후보 모델 학습 및 결과 분석, 발표 자료제작, 중간 발표 준비
05.22	중간 발표
05.23	[최종 모델 선정 및 결과분석] baseline model-직접 설계 CNN model 후보군 도출, 최적의 모델 선택 [demo-web 개발] 최종 모델 이용한 demo web 개발
05.26	보고서 작성, 발표 자료제작, 최종 발표 준비

Table 1: 프로젝트 수행 계획

성명	분담 내용	기여도 (%)
이요원	baseline model 구현, 직접 설계 CNN model 개발, EMNIST-recognition demo-web 개발	45
박서영	직접 설계 CNN model 개발, 다수의 후보 모델 학습 및 결과 분석	25
박주은	MNIST extended dataset 분석, baseline model optimization	10
최서영	MNIST extended dataset 분석, baseline model optimization	10
허성현	baseline model optimization, 최종 후보 모델 비교 및 분석	10

Table 2: 역할 분담

3 수행 과정

3.1 EMNIST dataset 분석

EMNIST 데이터셋 [1]은 NIST 특수 database 19에서 파생된 손으로 쓴 문자와 숫자 집합으로 MNIST 데이터셋과 일치하는 28x28 픽셀 이미지 형식이다. byclass, bymerge, balanced, letters, digits, mnist 총 여섯 가지 카테고리 구성되어 있으며, 각 카테고리의 특징은 Table3과 같다. 이 중 숫자와 영문 대소문자가 혼합된 byclass, bymerge, balanced 데이터셋 세 가지를 비교해 학습에 사용할 데이터셋을 선정하고자 했다. Table 4와 같이 각 데이터셋은 서로 다른 클래스 수와 데이터 분포를 가지고 있어, 모델의 학습과 성능 평가에 중요한 영향을 미친다. class 수는 적지만 데이터의 양이 많은 bymerge

dataset이 가장 training에 적합할 것이라고 예측했다. 3.2에서 baseline model을 이용해 학습한 performance metrics를 가지고 최종적으로 모델 학습에 유리한 데이터셋을 bymerge dataset으로 선정했다.

category	class	training	testing	total
Byclass	62	697932	116323	814255
Bymerge	47	697932	116323	814255
Balanced	47	112800	18800	131600
Letters	26	88800	14800	103600
digits	10	240000	40000	280000
mnist	10	60000	10000	70000

Table 3: Dataset categories and their corresponding number of classes, training samples, testing samples, and total samples.

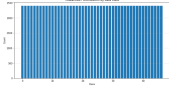

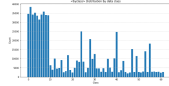
	balanced	bymerge	byclass
class 개수	47	47	62
전체 data 개수	131,600	814,255	814,255
특징	모든 class 균등하게 분포 형태가 비슷한 대소문자 동일 class로 통합	형태가 비슷한 대소문자 동일 class로 통합 byclass에 비해 data가 상대적으로 균등하게 분포	숫자, 대소문자 모두 구분된 class 각 class의 불균등
figure			

Table 4: Balanced, Bymerge, Byclass dataset 특징

3.2 baseline model

3.2.1 LeNet-5

LeNet-5[2]는 Yann LeCun과 그의 동료들이 개발한 초기의 합성곱 신경망(CNN) 구조 중 하나로, 손으로 쓴 숫자를 인식하는데 사용되었다. 이 architecture는 1998년에 발표되었으며, 당시에는 숫자 이미지 처리 분야에서 혁신적인 성과를 이뤘다. Figure1에 표현된 LeNet-5의 주요 특징은 합성곱 계층과 풀링 계층의 번갈아가는 구조를 가지고 있다는 점이다. 이 구조는 입력 이미지의 공간적인 구조를 유지하면서도 계층별로 특징을 추출하고 감소시키는 방식으로 작동한다. 합성곱 계층은 입력 이미지에 필터를 적용하여 특징 맵을 생성하고, 풀링 계층은 특징 맵의 크기를 줄여 계산량을 감소시킨다. 이러한 계층들은 이미지의 중요한 특징을 추출하여 숫자를 인식하는데 사용된다.

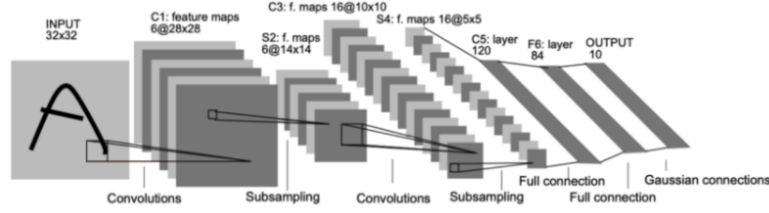


Figure 1: LeNet-5 structure

Performance on different Datasets LeNet-5 모델에서 optimizer는 Adam, epoch는 70, learning rate는 0.001, batch size는 256으로 동일하게 통제하고 balanced, bymerge, byclass 세 가지 데이터셋을 학습했다. 학습 결과는 Table 5와 같다. 데이터 개수가 가장 적은 balanced에서는 예상대로 빠른 학습 속도를 보였으며, 클래스 개수는 byclass보다 적지만 balanced보다 데이터 양이 많은 bymerge가 가장 우수한 validation accuracy를 기록했다. 그러나 learning curve(Figure 2)을 분석한 결과, overfitting 문제가 발생하고 있음을 확인하였다. 이는 데이터에 비해 모델의 복잡도가 높기 때문으로 판단된다. 따라서 CNN 모델 설계 시 이러한 문제를 해결하기 위해 모델의 복잡도를 데이터 양에 맞추어 조정할 필요성을 확인했다.

Table 5: Performance metrics for balanced, bymerge, and byclass datasets using LeNet-5.

	Balanced	Bymerge	Byclass
time	4s 6ms/step	14s 3ms/step	14s 3ms/step
train acc	0.9662	0.9183	0.8864
val acc	0.8481	0.8866	0.8489

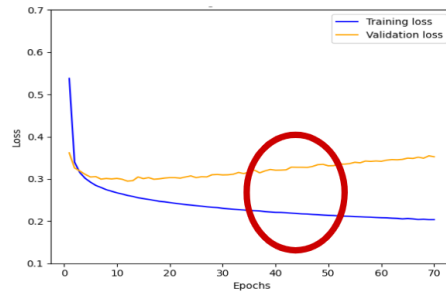


Figure 2: Learning Curve of LeNet-5 with Bymerge dataset

Optimization LeNet-5 모델의 최적화를 위해 하이퍼파라미터 튜닝을 수행하였다. 이때 가장 우수한 성능을 보인 Bymerge dataset을 사용해 진행했다. 먼저 최적의 학습률을 찾고, 초기화 방법 및 최적화 알고리즘의 조합을 통해 최적의 하이퍼파라미터를 도출하고자 하였다. 그 결과, Table 6에 나타난 바와 같이, 학습률이 0.01 일 때 random 초기화 방법과 SGD(momentum=0.9) 최적화 알고리즘의 조합이 가장 우수한 성능을 보였다.

	Adam	Nadam	SGD)
random	0.894	0.893	0.897
He	0.894	0.894	0.896
LeCun	0.895	0.894	0.895

Table 6: Finding best combination of initializer and optimizer on LeNet-5

3.2.2 ResNet50

ResNet-50[3]은 'Residual Network'의 50개 레이어를 갖는 딥러닝 구조이다. 이 구조는 마이크로소프트 연구원이 개발한 것으로, 2015년 이미지 인식 대회인 ILSVRC에서 우승한 모델 중 하나이다. ResNet은 네트워크의 깊이를 증가시키면서도 그래디언트 소실 문제를 효과적으로 해결하기 위해 residual connection이라는 개념을 도입했다. residual connection(Figure 3)은 입력과 출력을 더하는 것으로, 이를 통해 네트워크가 층을 건너뛰어 정보를 효과적으로 전달할 수 있다. 이는 그래디언트 소실 문제를 완화하고, 더 깊은 네트워크를 구축할 수 있게 해준다.

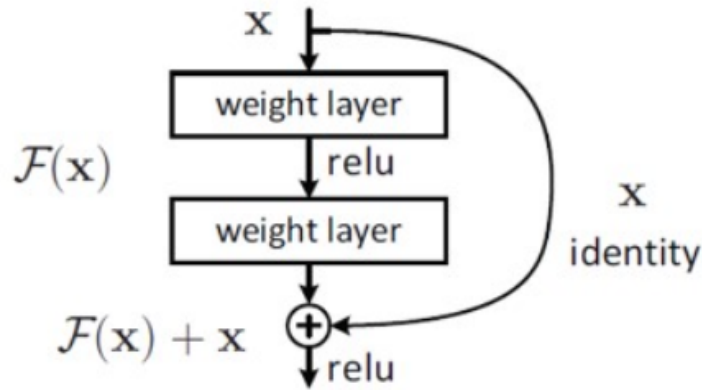


Figure 3: residual connection

Performance on different Datasets ResNet50 모델에서 optimizer는 adam, epoch는 70, learning rate는 0.001, batch size는 256으로 동일하게 통제하고 balanced, bymerge, byclass 세 가지 데이터셋을 학습했다. 학습 결과는 Table 7와 같다. 데이터 개수가 가장 적은 balanced에서 가장 빠른 학습 속도를 보였으며, 클래스 개수는 byclass보다 적지만 balanced보다 데이터 양이 많은 bymerge가 예상과 같이 가장 우수한 validation accuracy를 기록했다. 그러나 learning curve(Figure 4)을 분석한 결과, overfitting 문제가 발생하고 있음을 확인하였다. 이는 데이터에 비해 모델의 복잡도가 높기 때문으로 판단된다. 따라서 CNN 모델 설계 시 이러한 문제를 해결하기 위해 모델의 복잡도를 데이터 양에 맞추어 조정할 필요성을 확인했다.

Table 7: Performance metrics for balanced, bymerge, and byclass datasets using ResNet50.

	Balanced	Bymerge	Byclass
time	22s 31ms/step	132s 31ms/step	132s 30ms/step
train acc	0.9763	0.9567	0.9564
val acc	0.8665	0.8946	0.8503

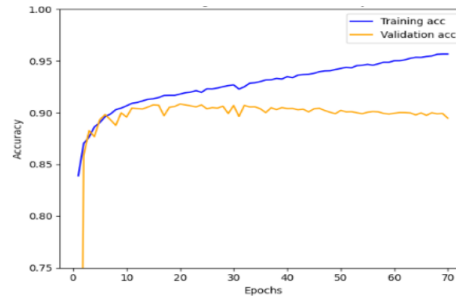


Figure 4: Learning Curve of ResNet50 with Bymerge dataset

Optimization ResNet50 모델의 최적화를 위해 하이퍼파라미터 튜닝을 수행하였다. 이때 가장 우수한 성능을 보인 Bymerge dataset을 사용해 진행했다. 먼저 최적의 학습률을 찾고, 초기화 방법 및 최적화 알고리즘의 조합을 통해 최적의 하이퍼파라미터를 도출하고자 하였다. 그 결과, Table 8에 나타난 바와 같이, 학습률이 0.01일 때 LeCun 초기화 방법과 Adam 최적화 알고리즘의 조합이 가장 우수한 성능을 보였다.

initializer/optimizer	Adam	Nadam	SGD
random	0.907	0.893	0.904
He	0.907	0.894	0.897
LeCun	0.909	0.894	0.903

Table 8: Finding best combination of initializer and optimizer on ResNet50

3.2.3 LeNet-5 & ResNet50 비교

두 개의 baseline 모델인 LeNet-5와 ResNet-50의 성능 비교는 Table 9에 명시되어 있다. LeNet-5는 상대적으로 얇고 넓은 구조를 갖고 있으며, ResNet-50은 깊고 복잡한 구조를 특징으로 한다. 이러한 구조적 차이로 인해 LeNet-5는 학습 속도가 빠르지만, ResNet-50은 더 긴 학습 시간이 소요됨에도 불구하고 성능 면에서 더 우수한 결과를 보인다. 이는 깊이가 넓이보다 성능 향상에 기여한다는 것을 시사하지만, 학습 시간이 오래 걸리는 단점이 있다. 이러한 점을 고려하여, 깊이와 학습 속도 간의 균형을 맞춘 모델을 설계하고자 했다.

Table 9: Comparison LeNet-5 and ResNet50

	LeNet-5	ResNet50
val acc	0.897	0.909
time	14s 3ms/step	132s 31ms/step
trainable param	64,851	23,630,895

3.3 CNN model 설계

3.3.1 ResNet 기반

28*28*1 input	zero padding 3*3	conv 1 [5*5, 32]	zero padding 1*1	conv 2 [3*3, 64] [3*3, 64] *3	conv 3 [3*3, 64] (stride=2) [3*3, 64] *2	GAP	dense layer
------------------	------------------------	---------------------	------------------------	--	--	-----	----------------

Figure 5: custom-designed CNN model based on ResNet

baseline 모델인 ResNet50은 EMNIST 데이터셋에 비해 지나치게 복잡하므로 이를 간소화하여 사용자 정의 모델로 재구성하였다(Figure 5). ResNet50은 원래 RGB 3차원 입력을 받도록 설계되었으나, 이를 1차원 입력으로 수정해도 정확도에 영향을 주지 않음을 확인하였다. 이를 통해 메모리 효율성을 높일 수 있었다. 또한, residual block의 사용을 줄이는 대신, 넓이 보다는 깊이에 더 중점을 두어 설계하였다. 그 결과, ResNet50 모델과

비교하여 best model까지의 학습 시간이 3240초에서 385초로 88.14% 단축되었으며, validation accuracy는 0.47% 향상되었다.

3.3.2 VGG 기반

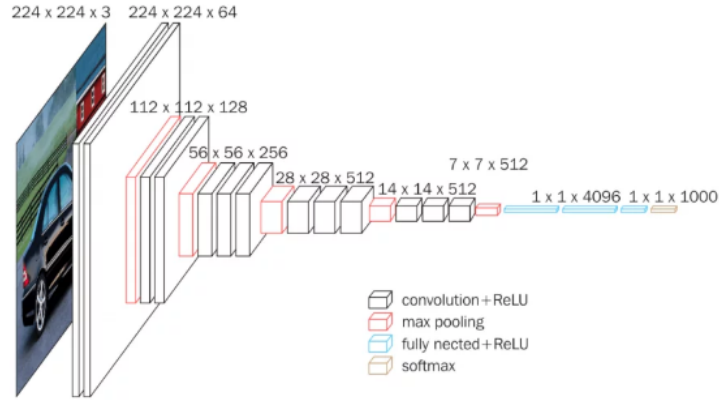


Figure 6: VGG-16 Architecture

VGG [4] 네트워크는 2014년 Simonyan과 Zisserman이 제안한 합성곱 신경망 구조 (Figure 6)로, 깊이와 성능 간의 상관관계를 조사하기 위해 설계되었다. 3x3 크기의 필터를 사용하는 convolution layer와 2x2 크기의 max pooling layer로 구성된다. 이러한 단순하고 일관된 구조는 네트워크의 깊이를 쉽게 증가시킬 수 있게 한다. VGG 모델의 단점 중 하나는 깊이가 깊어짐에 따라 계산 비용이 증가하고, 메모리 사용량이 많아진다는 점이다. 그러나 여전히 VGG는 그 단순성과 강력한 성능 덕분에 많은 연구와 응용에서 사용되고 있다. 본 연구에서도 VGG 구조의 장점을 활용하여 EMNIST dataset 인식 문제를 해결하고자 한다.



Figure 7: custom-designed CNN model based on vgg16

VGG16 모델은 EMNIST 데이터셋에 비해 지나치게 복잡하므로 이를 간소화하여 사용자 정의 모델로 재구성하였다 (Figure 7). 이미지 크기를 고려하여 convolution kernel 개수를 조정하고 Fully connected layer도 간소화했다. 그러나, 모델의 깊이는 동일하게 구성해 성능을 유지한 채로 학습 시간을 단축시켰다.

3.4 최종 모델 선정

Table 10: 최종 모델 후보군

	LeNet-5	ResNet50	custom-ResNet	custom-VGG
val acc	0.897	0.909	0.909	0.913
time	14s 3ms/step	132s 31ms/step	35s 8ms/step	44s 10ms/step

네 개의 모델, 즉 두 개의 베이스라인 모델 (LeNet-5와 ResNet50) 과 두 개의 직접 설계한 모델 (custom-ResNet, custom-VGG) 을 정확도와 학습 시간을 기준으로 비교하였다. 비교 결과는 Table 10과 같다. 정확도는 custom-VGG가 가장 우수했으며, 학습 시간은 LeNet-5가 가장 짧았다. 그러나 LeNet-5는 정확도에서 낮은 성능을 보였기에, 학습 시간 면에서도 비교적 우수한 성능을 보인 custom-VGG를 최종 모델로 선정하였다. 이와 같이, 정확도와 학습 시간을 적절히 고려하여 최종 모델을 선정하였다.

4 A further study

EMNIST 데이터셋으로부터 최적의 모델을 도출하는 것에서 프로젝트가 그치는 것이 아쉬웠다. 따라서 프로젝트를 확장하여 해당 모델의 실제 활용 가능성을 고려해보았다. EMNIST 데이터셋은 모든 영문자를 포함하고 있기 때문에, 타이핑 대신 필기로 입력된 문자를 인식하는 웹 또는 앱을 개발하고자 했다. 이를 위해 오픈소스 데모 웹을 수정하여 필기 인식 데모 웹²을 구현했다. 최종 선정한 모델인 custom-VGG가 사용자의 손 글씨 이미지를 inference하여 결과를 출력한다. 사용자들은 이를 통해 편리하게 손으로 쓴 글씨를 입력으로 사용할 수 있다. Figure 8은 실제 사용자 인터페이스 화면을 보여준다. 이를 통해 CNN 모델에 대한 연구로부터 실제 활용 가능한 제품으로의 전환을 이루어내어 보다 의미 있는 성과를 이끌어냈다.

5 Discussion

이번 프로젝트를 진행하며 MNIST Extended 데이터셋을 활용하여 다양한 CNN 모델을 학습시키고 성능을 비교하여 최적의 모델을 선정하는 과정에서 많은 것을 배울 수 있었다. 그 중 몇 가지 주요한 배움을 공유하고, 앞으로의 연구 방향에 대해 논의하고자 한다.

데이터셋에 맞는 모델 복잡도 고려 첫 번째로 중요한 점은 데이터셋에 맞는 모델 복잡도를 고려하여 모델을 설계해야 한다는 것이다. 본 프로젝트에서는 LeNet-5와 ResNet-50

²Available at <https://github.com/2oil/HandScriptAI.git>



Figure 8: demo-web: user interface

을 baseline 모델로 사용하여 성능을 비교하였고, 그 결과 ResNet-50이 더 나은 성능을 보였지만 높은 복잡도로 인해 학습 시간이 길어지는 단점이 있었다. 이로 인해 CNN 모델을 설계할 때는 데이터셋의 크기와 복잡도에 맞추어 모델의 복잡도를 조정하는 것이 중요하다는 점을 깨달았다. 데이터가 적을 경우, 모델의 복잡도가 너무 높으면 오히려 과적합(overfitting) 문제가 발생할 수 있으므로 주의가 필요하다.

CNN 모델의 깊이와 넓이 두 번째로, CNN 모델에서 깊이와 넓이의 균형이 중요한 요소임을 알게 되었다. ResNet-50과 같은 깊은 모델은 일반적으로 넓은 모델보다 더 나은 성능을 보이는 경향이 있었다. 하지만 깊이가 깊어질수록 학습 시간이 길어지며, 메모리 사용량도 증가하기 때문에 이를 고려한 최적의 설계가 필요하다. 이번 프로젝트에서 custom-ResNet과 custom-VGG 모델을 설계하면서, 깊이를 유지하면서도 불필요한 복잡도를 줄이는 방향으로 모델을 간소화하여 효율성을 높일 수 있었다.

모델 최적화와 하이퍼파라미터 튜닝 세 번째로, 모델 최적화를 위한 하이퍼파라미터 튜닝의 중요성을 다시 한 번 확인할 수 있었다. LeNet-5와 ResNet-50 모두에서 학습률, 초기화 방법, 최적화 알고리즘 등을 조정하여 성능을 향상시킬 수 있었다. 이는 모델 성능을 극대화하기 위해서는 다양한 하이퍼파라미터 조합을 시도해보는 것이 필수적임을 보여준다.

Demo 웹 개발 또한, 본 프로젝트의 일환으로 개발한 데모 웹은 연구 결과를 실제 활용 가능한 형태로 구현한 좋은 예시가 되었다. 필기 인식 데모 웹을 통해 사용자들이 손으로 쓴 글씨를 편리하게 입력할 수 있도록 함으로써, 연구 성과를 실질적인 응용 프로그램으로 발전시킬 수 있었다. 이는 연구 결과를 실생활에 적용하는 데 있어 중요한 단계이며, 향후 이러한 접근 방식을 더욱 확대할 필요가 있다고 느꼈다.

앞으로의 연구 방향 향후 연구에서는 다음과 같은 방향으로 확장할 수 있을 것이다.

- **다양한 데이터셋 활용:** EMNIST 외에도 다양한 손글씨 데이터셋을 활용하여 모델의 일반화 성능을 평가하고, 다양한 글씨체와 언어에 대해 모델을 확장할 수 있다.
- **모델 경량화:** 모바일 기기나 임베디드 시스템에서 실시간으로 사용할 수 있도록 모델을 경량화하는 연구가 필요하다.

이번 프로젝트를 통해 데이터셋에 맞춘 모델 설계의 중요성과 CNN 모델의 깊이와 넓이 조정, 하이퍼파라미터 튜닝의 중요성을 다시 한 번 확인할 수 있었다. 또한, 연구 결과를 실질적인 응용 프로그램으로 발전시키는 과정을 경험하며 많은 성과를 얻을 수 있었다. 앞으로의 연구에서도 이러한 경험을 바탕으로 더욱 발전된 결과를 도출할 수 있기를 기대한다.

References

- [1] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.
- [2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.