

Deep Residual Learning report

Junseo Park

School of Computer Science and Engineering, Pusan National University.

1 Introduction

합성곱 신경망은 깊이와 구조 설계의 발전으로 성능이 꾸준히 향상되었다. VGG-16(Type-D)은 3×3 stacked convolutions, ResNet-50(bottleneck)은 residual connections로 학습을 안정화한다. 본 보고서는 *mini-CIFAR-10* 서브셋(plane/car/bird, 32×32)에서 두 모델을 동일 설정으로 구현·비교하고, 데이터 증강 조합과 마지막 BatchNorm γ 초기화가 모델 성능에 미치는 영향을 정량적으로 분석하고, 소규모 데이터셋에서의 효율적인 학습 전략을 탐색한다.

2 Implementation

본 섹션은 구현한 VGG-16(Type-D)과 ResNet-50(bottleneck)의 구조적 설계와 공통 학습 설정을 간략히 정리한다.

2.1 VGG-16 (Type-D, CIFAR-10 맞춤)

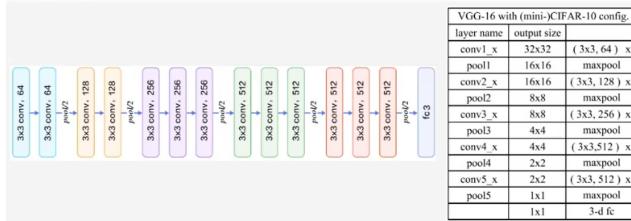


Figure 1: VGG-16 (Type-D) 아키텍처

3×3 stacked convolutions 뒤에 BN-ReLU를 두고, 스테이지마다 2×2 max-pooling(stride 2)으로 $32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 로 축소한다. 최종 1×1 특징맵(채널 512)을 펼쳐 FC(512 \rightarrow 3)로 분류한다.

- **Feature extractor:** $(3 \times 3, 64) \times 2 \rightarrow \text{MP} \rightarrow (3 \times 3, 128) \times 2 \rightarrow \text{MP} \rightarrow (3 \times 3, 256) \times 3 \rightarrow \text{MP} \rightarrow (3 \times 3, 512) \times 3 \rightarrow \text{MP} \rightarrow (3 \times 3, 512) \times 3 \rightarrow \text{MP}$.
 - **Head:** Flatten $\rightarrow \text{FC}(512 \rightarrow 3)$ (GAP 미사용).

2.2 ResNet-50 (Bottleneck, CIFAR-10 맞춤)

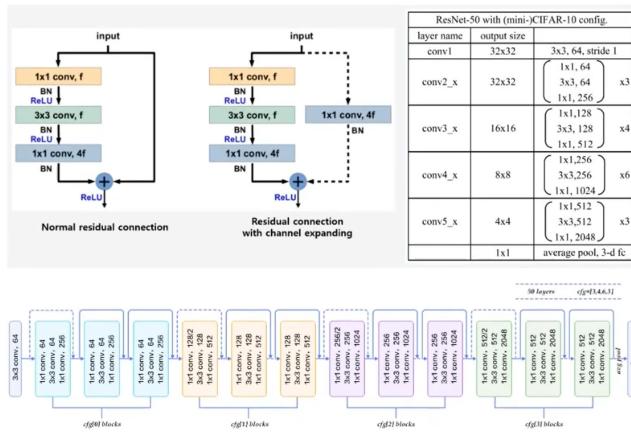


Figure 2: ResNet-50 (bottleneck) 아키텍처

기본 블록은 $[1 \times 1, f] \rightarrow [3 \times 3, f] \rightarrow [1 \times 1, 4f]$ 이며, 합성곱 뒤 BN-ReLU를 둔다. 초기 max-pool은 제거하고 각 스테이지 첫 블록에서만 stride 2로 다운샘플링하며, 채널/해상도가 다르면 1×1 projection shortcut(+BN)을 사용한다.

- **Stem:** $\text{Conv}(3 \times 3, 64, s=1) \rightarrow \text{BN} \rightarrow \text{ReLU}$.
 - **Stages:** $[64, 64, 256] \times 3 \rightarrow [128, 128, 512] \times 4 \rightarrow [256, 256, 1024] \times 6 \rightarrow [512, 512, 2048] \times 3$ (각 스테이지 첫 블록 stride 2).
 - **Head:** $\text{GAP} \rightarrow \text{FC}(2048 \rightarrow 3)$.

2.3 Training Setup

두 모델은 동일한 데이터 파이프라인과 핵심 하이퍼파라미터로 학습/평가하였다. 차이점은 에폭 수 뿐이다.

- **데이터/입력:** mini-CIFAR-10의 *plane/car/bird* 클래스, 해상도 32×32 .
 - **배치/로더:** train batch 128 (shuffle on), test batch 100.
 - **최적화:** SGD ($\text{lr} = 10^{-2}$, momentum = 0.9, weight decay = 5×10^{-4}).
 - **손실/지표:** nn.CrossEntropyLoss 사용, 테스트 정확도 보고.
 - **에폭:** VGG-16은 20 epochs, ResNet-50은 15 epochs.

3 Results & Analysis

이 섹션에서는 구현한 두 모델의 학습 결과와 일반화 성능을 중심으로 분석한다. 먼저, VGG-16(Type-D)과 ResNet-50(bottleneck)을 동일 조건에서 학습하여 정확도와 수렴 양상, 연산 비용(에폭당 시간)을 비교한다. 이어, 주어진 ResNet 네트워크 구성에서 데이터 증강(transform) 조합이 성능에 미치는 영향을 $No\ Aug \rightarrow Crop+Flip \rightarrow Crop+Flip+Cutout$ 순으로 정량 분석한다. 마지막으로, Identity mapping을 유도하기 위해 마지막 BatchNorm의 gamma 값을 0으로 설정했을 때와 설정하지 않았을 때를 비교하여 수렴 안정성 및 최종 성능 차이를 평가한다. 이하 결과는 각 학습별로 해석을 제시한다.

3.1 VGG-16 vs. ResNet-50

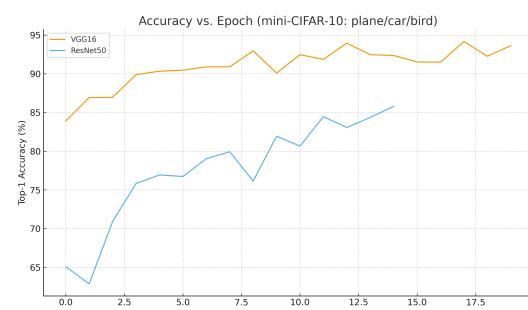


Figure 3: Top-1 Accuracy vs. Epoch. VGG-16은 더 높고 안정적인 정확도를 보이며, ResNet-50은 15 epochs 범위에서 완만히 상승하지만 더 낮은 수준에 머문다.

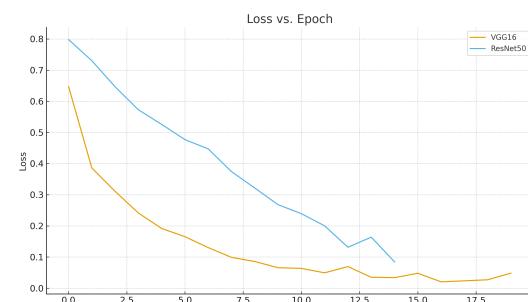


Figure 4: Loss vs. Epoch. 두 모델 모두 감소 추세이나, VGG-16이 더 빠르게 안정 그간에 진입했다

Summary 그림 3 기준으로 VGG-16은 Best 94.17%, Last 93.63%를 기록했고, ResNet-50은 Best=Last 85.80%였다. 즉 최종 기준으로 VGG-16이 +7.83 pp, 최고점 기준으로 +8.37 pp 우세다. 그림 4에서는 두 모델 모두 손실이 감소하지만, VGG-16이 더 이른 시점에 변동이 작아지며 안정 구간에 가까워진다. 에폭당 평균 시간은 ResNet-50 약 51.56초, VGG-16 약 6.28초로 약 8.2배 차이가 나서 시간 대비 성능 면에서도 VGG-16의 효율이 높다.

Interpretation 이번 설정(입력 32×32 , 3-class, 20/15 epochs, 고정 학습률)에서는 구조가 단순한 VGG-16이 유리하게 작동했다. 작은 해상도에서 bottleneck($1 \times 1 \rightarrow 3 \times 3 \rightarrow 1 \times 1$)은 채널 축을 초기에 압축했다가 다시 확장하는 과정에서 유용한 국소 패턴 형성이 더딜 수 있는 반면, 규칙적인 3×3 스택은 공간 특징을 빠르게 축적해 초기 상승 기울기와 최종 일반화가 높게 나온다. 또한 ResNet-50은 스테이지 전환마다 projection shortcut(1×1 conv+BN)이 포함되고 대부분의 conv 뒤에 BN이 붙어 연산/메모리 오버헤드가 크다. 동일 배치·학습률에서는 이러한 구조적 비용이 유효 배치 축소와 비슷한 불리함을 만들어 수렴 속도를 낮춘다. 최적화 측면에서도 깊은 잔차 네트워크는 보통 warmup과 cosine/step decay 전제를 두는데, 본 실험처럼 스케줄 없이 15 epochs 내에서는 ResNet-50이 완만한 상승만 보이며 충분히 수렴하지 못했다. 헤드 차이도 영향을 준다. 3-class · 저해상도 맥락에서는 GAP+FC(2048→3)보다 Flatten+FC(512→3)가 약한 위치 민감성을 유지해 결정 경계 형성에 유리했을 가능성이 있다. 종합하면 본 과제의 제약에서는 VGG-16이 더 빠르고 안정적으로 수렴했고, 같은 시간 대비 정확도에서도 우세했다.

3.2 Effect of Data Augmentation

주어진 ResNet-50(bottleneck) 구성에서, 데이터 증강(transform) 조합에 따른 test accuracy 변화를 비교하였다. 사용한 조합은 다음과 같다.

- No Augmentation: 기본 전처리만 적용.
- Crop + Flip: RandomCrop(padding 적용)과 RandomHorizontalFlip을 학습 시 매 배치에 적용.
- Crop + Flip + Cutout: 위 조합에 RandomErasing(Cutout)을 추가 적용.

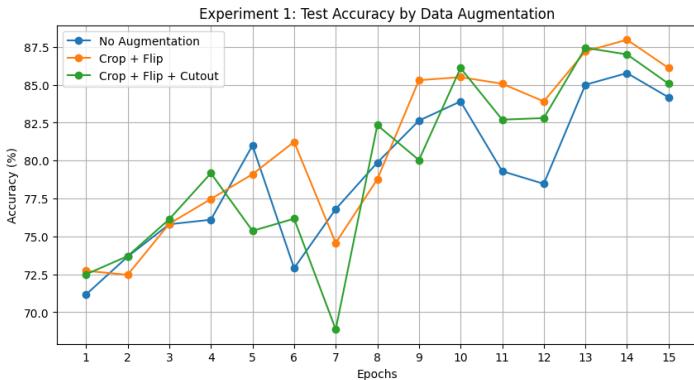


Figure 5: 증강 조합에 따른 test accuracy 추이(ResNet-50, 15 epochs).

Summary 로그 기준 최고 성능은 Crop + Flip 87.97%, Crop + Flip + Cutout 87.43%, No Aug 85.77% 순서였다. 마지막 epoch에서도 각각 86.10%, 85.07%, 84.17%로 동일한 순서를 보였다. 세 조건의 에폭당 시간은 모두 약 51.6초로 유사하므로, 차이는 주로 일반화 효과에서 비롯된다.

Interpretation RandomCrop과 RandomHorizontalFlip은 입력의 위치·반전 변형을 통해 모델이 translation/pose 변화에 덜 민감해지도록 만들고, 효과적으로 데이터 다양성을 늘려 과적합을 자연시킨다. 그 결과 학습 후반의 정확도가 상승하고 곡선이 안정적이다. RandomErasing(Cutout)은 일부 영역을 가려 occlusion에 대한 강건성을 학습시키지만, 초기에 정보가 제거되면서 정확도 변동이 커지거나(예: 7 epoch 부근의 일시 하락) 수렴 속도가 느려지는 trade-off가 나타난다. 에폭 예산이 15로 제한된 본 조건에서는 Crop + Flip이 가장 실용적인 선택으로 나타났고, 학습 에폭을 늘리거나 학습률 스케줄을 도입하면 Cutout의 정규화

효과가 누적되어 No Aug 대비 격차가 더 커지고 Crop + Flip과의 성능 차이도 줄어들 가능성이 있다.

3.3 Effect of Last BN γ Initialization

Identity mapping을 유도하기 위해 bottleneck의 마지막 BatchNorm의 scale 파라미터(γ)를 0으로 초기화한 경우와, 기본 초기화인 경우를 비교하였다.

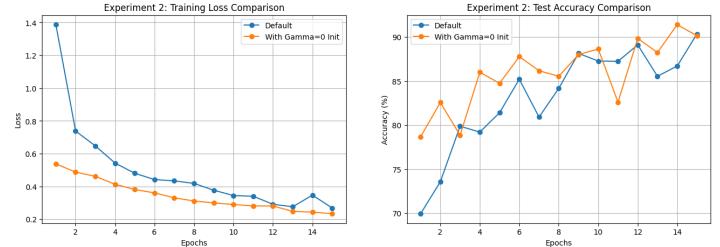


Figure 6: 마지막 BN γ 초기화 비교(ResNet-50, 15 epochs). 좌: training loss, 우: test accuracy.

Summary 로그 기준으로 기본 초기화(Default)는 best 90.33%, last 90.33%였고, $\gamma=0$ 초기화는 더 높은 최고점 91.40%를 달성했으나, 최종적으로는 90.13%로 마감했다. 초기 1~5 epoch 구간의 평균 정확도는 $\gamma=0$ 이 82.16%, 기본이 76.79%로 약 5.4 pp 높았고, 표준편차는 $\gamma=0$ 이 더 작아 초기 변동이 작았다. loss 곡선에서도 $\gamma=0$ 이 전 구간에서 더 낮게 시작해 빠르게 수렴한다. 두 설정 모두 에폭당 시간은 약 51.6초로 동일하다.

Interpretation $\gamma=0$ 으로 시작하면 각 bottleneck 블록의 마지막 BN이 출력을 거의 0으로 스케일하므로, 초기에 블록 출력은 $y \approx x + F(x)$ 에서 $F(x) \approx 0$ 이 되어 사실상 $y \approx x$ (입력이 거의 그대로 통과)로 동작한다. 이 때 역전파의 민감도는 $\delta y / \delta x \approx I$ 가 되어 gradient가 약해지거나 폭주하지 않고 skip 경로를 따라 안정적으로 전달된다. 이렇게 안정적인 흐름 위에서 학습이 진행되면, $F(x)$ 의 가중치는 0 근처에서 점차 커지며 필요한 변화(잔차)만 조금씩 추가로 학습하게 된다. 그 결과 초기 loss가 빠르게 내려가고, 초반 정확도와 곡선 안정성이 개선되는 양상이 관찰된다. 다만 본 과제처럼 학습 예산이 15 epoch로 짧고 학습률 스케줄이 없을 때는, 후반에 잔차가 충분히 커져 항등에서 멀어지는 데 시간이 부족하여 최종 정확도는 기본 초기화와 큰 차이가 나지 않거나 소폭 낮게 끝날 수 있다(여기서는 last 90.13% vs. 90.33%). 학습을 더 길게 가져가거나 warmup+cosine decay를 사용하면, $\gamma=0$ 의 초기 안정화 이점이 후반 수렴까지 이어져 최종 성능 이득이 더 크게 나타날 가능성이 높다.

4 Conclusion

본 보고서는 mini-CIFAR-10 환경에서 VGG-16과 ResNet-50의 성능을 비교하고, 데이터 증강과 마지막 BatchNorm의 γ 초기화가 학습에 미치는 영향을 분석하였다.

전체 결과는 딥러닝 모델과 기법의 성능이 절대적이지 않음을 보여준다. 효율성은 데이터 규모·복잡도, 입력 해상도, 연산 예산, 최적화 전략 등 주어진 문맥에 따라 달라진다. 복잡한 최신 아키텍처가 항상 최선은 아니며, 제한된 환경에서는 간결한 구조가 더 실용적일 수 있다. 또한 특정 최적화 기법의 효과는 충분한 학습 시간과 적절한 학습률 스케줄링이 뒷받침될 때 극대화될 수 있음을 시사한다. 본 보고서는 이론적 우수성만으로는 충분치 않으며, 문제의 문맥을 함께 고려하는 실용적 설계의 중요성을 실험적으로 확인하였다.