

---

# CNN 기반 이미지 분류 실험 보고서

---

(Baseline, Data Augmentation, Transfer Learning 비교 분석)

2025 DECEMBER 30

민찬영

# Contents

1. Problem Definition & Object.....	3
1.1 Problem Definition .....	3
1.2 Practical Context .....	3
1.3 Project Context .....	3
1.4 Objectives of this Report.....	4
2. Experimental Setup & Evaluation Criteria.....	5
2.1 Dataset Description .....	5
2.2 Dataset Split Strategy .....	5
2.3 Experimental Enviroment.....	5
2.4 Model Configuration.....	6
2.5 Model Evaluation .....	6
3. Baseline Limitations and Data Augmentation Effects .....	7
3.1 비교 실험 설계 개요 .....	7
3.2 Baseline vs Data Augmentation 학습 특성 비교.....	7
3.3 Data Augmentation 효과 에 대한 해석 .....	8
4. Verification of Internal Representations .....	9
4.1 내부 표현 분석의 필요성.....	9
4.2 Conv Layer 별 Feature Map 시각화 결과 .....	9
4.3 계층적 특징 학습에 대한 해석 .....	10
5. Model Decision Verification with XAI .....	11
5.1 Grad-CAM 분석의 목적.....	11
5.2 Grad-Cam 시각화 결과.....	11
5.3 모델 판단의 신뢰성에 대한 해석.....	12
6. Transfer Learning as an Extension Option .....	13
6.1 Transfer Learning 도입 배경 .....	13

6.2 사전 학습 모델 및 적용 전략 .....	13
6.3 Transfer Learning 적용 결과에 따른 해석 .....	14
7. Final Comparison Practical Decision .....	16
7.1 실험 조합별 성능 종합 비교 .....	16
7.2 데이터 전략과 모델 전략의 역할 분리 .....	17
7.3 실무 관점에서의 최종 선택 .....	18
8. Conclusion.....	18
<b>Appendix.....</b>	<b>19</b>
A. Experimental Environment & Reproducibility.....	19
B. Data Preprocessing .....	20
C. Data Augmentation Settings .....	20
D. Model Architecture Tables .....	21

# 1. Problem Definition & Object

## 1.1 Problem Definition

Convolutional Neural Network(CNN)는 이미지 분류 문제에서 강력한 성능을 보이지만, 학습 데이터의 양이 제한된 환경(small-data scenario)\*\*에서는 일반화 성능이 급격히 저하되는 문제가 자주 발생한다.

특히 데이터 수가 충분하지 않을 경우, CNN은 객체의 본질적인 시각적 특징을 학습하기보다, 훈련 데이터에 존재하는 특정 위치, 특정 방향, 특정 색 분포와 같은 우연적 패턴을 암기(overfitting)하는 경향을 보인다.

이러한 현상은 실제 서비스 환경에서 새로운 데이터에 대한 예측 성능 저하로 이어진다.

본 프로젝트는 다음의 질문에서 출발한다.

**“ 데이터가 적은 상황에서 CNN은 왜 일반화에 실패하는가,**

**그리고 이 문제를 모델 복잡도 증가 없이 해결할 수 있는 방법은 무엇인가?”**

## 1.2 Practical Context

현업 환경에서는 다음과 같은 제약 조건이 빈번하게 존재한다.

- 대규모 이미지 데이터 수집 및 라벨링의 비용 문제
- 도메인 특화 데이터로 인한 데이터 확장 한계
- 경량 모델 또는 제한된 학습 자원 요구

이러한 상황에서 단순히 더 깊은 네트워크를 사용하거나 더 많은 파라미터를 추가하는 방식은 현실적인 해결책이 되기 어렵다.

따라서 본 프로젝트는 모델중심(Model-Centric) 접근이 아닌, 데이터와 학습과정 자체를 개선하는 Data-Centric 접근이 실무적으로 얼마나 효과적인 대안이 될 수 있는지를 검증하는데 초점을 둔다.

## 1.3 Project Context

본 프로젝트는 CIFAR-10 데이터셋을 기반으로 하여, 소량 데이터 환경에서 CNN의 일반화 성능을 단계적으로 개선하는 과정을 다룬다.

접근 방향은 다음과 같이 정의하였다.

1. Baseline CNN의 한계 명확화
  - 성능 수치뿐 아니라 학습 특성 관점에서 문제를 정의한다.
2. Data Augmentation을 통한 데이터 중심 개선
  - 입력 다양성 증가와 학습 동역학에 미치는 영향을 분석한다.
3. CNN 내부 표현 및 판단 근거 검증
  - Feature Map과 Grad-Cam을 통해 모델이 무엇을 학습하고, 어디를 보고 판단하는지를 확인한다.
4. Transfer Learning을 실무적 확장 옵션으로 평가
  - 성능 개선의 상한선과 현실적인 선택지를 도출한다.

#### 1.4 Objectives of this Report

본 보고서는 단순한 성능 비교 보고서가 아닌, 실무적 의사결정을 지원하는 기술 검증 문서를 목표로 한다.

구체적인 목표는 다음과 같다.

- 소량 데이터 환경에서 CNN 일반화 실패 원인을 구조적으로 설명한다.
- Data Augmentation 이 왜 효과적인지 정량적 · 정성적으로 검증한다.
- 모델의 내부 동작과 판단 근거를 시각적으로 검증하여 신뢰성을 평가한다.
- 여러 접근 방식 중 실무 환경에서 가장 합리적인 선택을 도출한다.

## 2. Experimental Setup & Evaluation Criteria

### 2.1 Dataset Description

본 프로젝트에서는 CIFAR-10 데이터셋을 사용하였다.

CIFAR-10은 10개의 클래스와 각 클래스당 6000장의 이미지로 구성된 데이터셋으로, 이미지 해상도는  $32 * 32$  RGB 이다.

이 데이터 셋은 다음과 같은 특성을 가진다.

- 상대적으로 작은 이미지 해상도
- 클래스별 데이터 수의 제한
- 객체의 세부 정보가 충분히 드러나지 않는 구조

이러한 특성은 CNN의 일반화 성능을 평가하기에 적절한 조건을 제공하며, 특히 소량 데이터 환경에서 발생하는 과적합 문제를 관찰하기에 적합하다.

### 2.2 Dataset Split Strategy

모델의 일반화 성능을 객관적으로 평가하기 위해 데이터는 다음과 같이 분할하였다.

- Training Set : 모델 학습에 사용
- Validation Set : 학습 중 성능 모니터링 및 비교 기준
- Test Set : 최종 성능 평가

Validation 데이터는 학습 데이터의 일부를 분리하여 구성하였으며, 모든 실험에서 동일한 분할 기준을 유지하여 모델 간 성능 비교의 공정성을 확보하였다.

### 2.3 Experimental Environment

실험은 다음과 같은 환경에서 수행되었다.

- Framework : TensorFlow / Keras
- Programming Language : Python
- Random Seed 고정을 통해 실험 재현성 확보

모든 실험은 동일한 환경과 설정에서 반복 수행되었으며, 이를 통해 실험 결과의 신뢰성과 재현

가능성을 보장하고자 하였다.

## 2.4 Model Configuration

Baseline CNN 및 Transfer Learning 모델 모두 다음과 같은 공통 학습 조건을 적용하였다.

- Optimizer : Adam
- Loss Function : Categorical Cross-Entropy
- Batch Size 및 Epoch 수 고정
- 동일한 학습 스케줄 적용

이를 통해 모델 구조나 데이터 처리 방식 외의 요인이 결과에 영향을 미치지 않도록 통제하였다.

## 2.5 Model Evaluation

본 프로젝트에서는 단일 수치 기반 평가의 한계를 보완하기 위해 시각화를 함께 활용하여 모델 성능을 종합적으로 평가하였다. Epoch에 따른 Validation Accuracy 그래프를 주요 평가 지표로 사용하였다.

### 3. Baseline Limitations and Data Augmentation Effects

#### 3.1 비교 실험 설계 개요

본 장에서는 소량 데이터 환경에서 CNN의 일반화 성능을 개선하기 위한 전략으로 데이터 증강(Data Augmentation)을 도입하고, 그 효과를 검증하기 위해 비교 실험을 수행한다. 제한된 데이터로 학습되는 CNN은 훈련 데이터에 포함된 특정 패턴을 빠르게 암기하는 경향이 있으며, 이로 인해 학습 데이터에 대해서는 높은 성능을 보이지만 새로운 데이터에 대해서는 성능이 저하되는 문제가 발생할 수 있다. 본 프로젝트에서는 이러한 과적합 문제를 모델 구조의 복잡도 증가가 아닌, 입력 데이터의 다양성을 확장하는 방식으로 해결할 수 있는지를 확인하고자 한다.

이를 위해 동일한 CNN 구조와 동일한 학습 설정을 유지한 상태에서 입력 데이터 처리 방식을 달리한 두 가지 학습 전략을 비교하였다. 첫 번째는 원본 이미지를 그대로 사용하여 학습하는 Baseline 접근 방식이며, 두 번째는 회전, 이동, 반전과 같은 데이터 증강 기법을 적용한 방식이다. 이 비교를 통해 모델의 성능 변화가 네트워크 구조나 학습 파라미터의 차이가 아닌, 데이터 증강 자체의 효과로부터 발생한 것임을 명확히 구분하고자 한다. 본 비교 실험의 목적은 데이터 증강이 학습 양상과 일반화 성능에 어떤 영향을 미치는지를 확인하고, 소량 데이터 환경에서 실무적으로 적용 가능한 개선 전략인지 평가하는 데 있다.

#### 3.2 Baseline vs Data Augmentation 학습 특성 비교

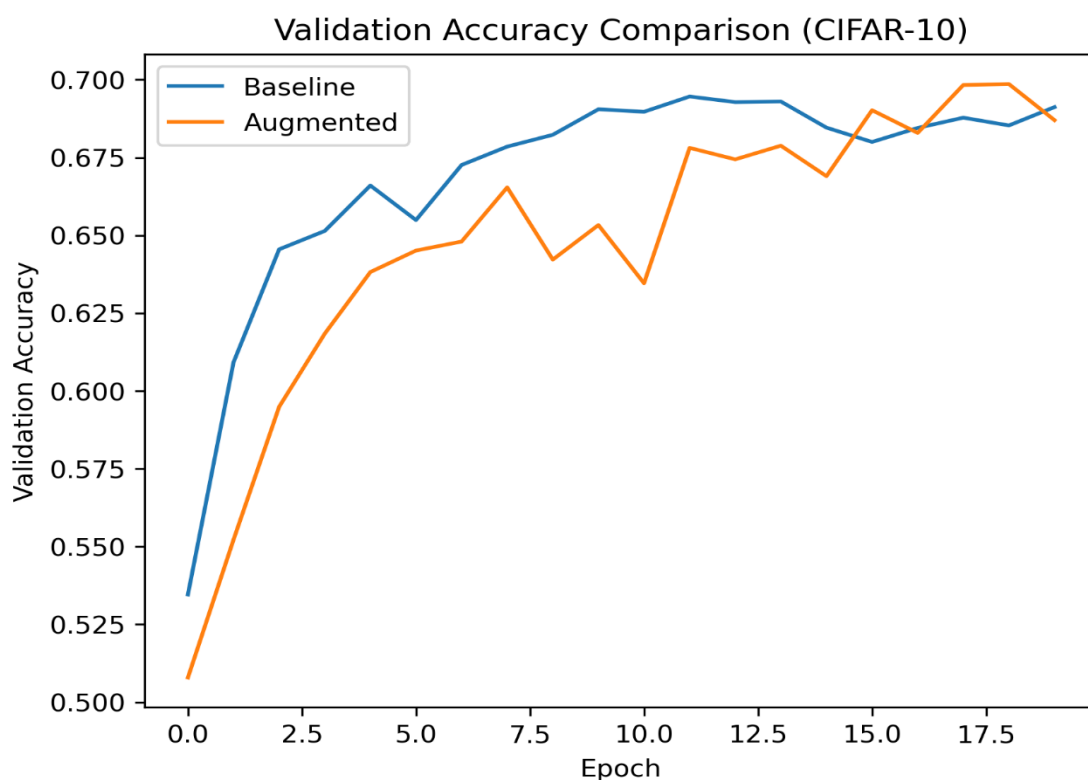


Figure 1. Validation Accuracy Comparison (Baseline vs Data Augmentation)



Baseline CNN의 학습 과정에서는 Validation Accuracy가 초기 Epoch에서 빠르게 상승한 후, 일정 수준에서 정체되는 경향을 보였다. 이는 모델이 비교적 이른 단계에서 학습 데이터에 존재하는 구분 가능한 패턴을 학습했음을 의미하지만, Epoch가 증가함에 따라 새로운 데이터에 대한 성능 개선이 제한적임을 보여준다.

반면 동일한 CNN 구조에 Data Augmentation을 적용한 경우, Validation Accuracy의 상승 속도는 상대적으로 완만하게 나타났으나 학습이 진행될수록 지속적인 성능 개선이 관찰되었다. 특히 중 후반 Epoch 구간에서는 Augmentation을 적용한 모델이 Baseline 모델과 유사하거나 더 높은 Validation Accuracy를 기록하였다. 이러한 학습 양상은 데이터 증강이 모델의 초기 수렴 속도는 낮추는 대신, 일반화 성능을 점진적으로 향상시키는 역할을 수행했음을 시사한다.

### 3.3 Data Augmentation 효과 에 대한 해석

Validation Accuracy의 학습 곡선 비교 결과, 데이터 증강(Data Augmentation)은 학습 초반의 수렴 속도보다는 학습이 진행될수록 일반화 성능을 점진적으로 개선하는 데 기여한 것으로 나타났다. 입력 이미지가 지속적으로 변형되는 환경에서 모델은 특정 위치나 방향에 의존한 패턴이 아닌, 클래스의 의미를 유지하는 일반적인 시각적 특징을 학습하게 된다. 이로 인해 Baseline 모델에서 관찰된 성능 정체 현상이 완화되었으며, 데이터 증강이 소량 데이터 환경에서 과적합을 줄이는 효과적인 전략임을 확인할 수 있다.

## 4. Verification of Internal Representations

### 4.1 내부 표현 분석의 필요성

앞선 장에서는 Validation Accuracy를 기준으로 모델의 일반화 성능을 평가하였다. 그러나 성능 지표만으로는 CNN이 실제로 어떤 특징을 학습했는지, 그리고 그 특징이 어떤 수준의 정보인지를 직관적으로 이해하기 어렵다. 이에 본 장에서는 합성곱 계층에서 생성되는 Feature Map을 시각화하여, CNN이 계층적으로 어떤 시각적 특징을 학습하는지를 확인한다.

본 분석의 목적은 모델 간 성능 비교가 아니라, CNN이 저수준 특징에서 고수준 특징으로 점진적으로 정보를 추상화하는 과정을 내부 표현 관점에서 이해하는 데 있다. 이는 이후 Grad-CAM을 통한 모델 판단 근거 분석을 해석하기 위한 기초 자료로 활용된다.

### 4.2 Conv Layer별 Feature Map 시각화 결과

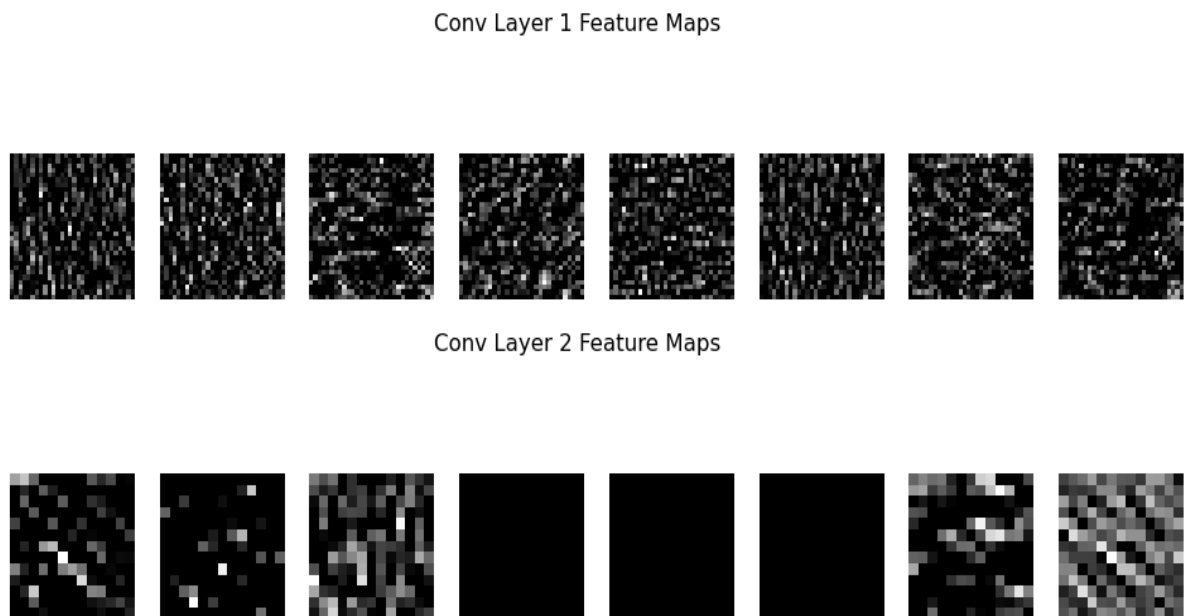


Figure 4-1. Feature map visualizations from the first and second convolutional layers

첫 번째 합성곱 계층(Conv Layer 1)의 Feature Map을 시각화한 결과, 입력 이미지의 엣지, 방향성, 밝기 변화와 같은 저수준 시각적 특징이 주로 강조되어 나타났다. 이는 CNN의 초기 계층이 픽셀 단위의 국소적 변화에 반응하도록 학습된다는 기존의 이론적 설명과 일치한다.

반면 두 번째 합성곱 계층(Conv Layer 2)의 Feature Map에서는 단순한 엣지 반응을 넘어, 여러 엣지 조합을 통해 형성된 텍스처, 반복 패턴, 국소적인 형태 정보가 상대적으로 두드러지게 나타났

다. 일부 Feature Map에서는 특정 영역에 대한 반응이 약화되거나 제거되는 현상도 관찰되었는데, 이는 계층이 깊어질수록 불필요한 세부 정보가 점차 제거되고, 보다 추상적인 특징 표현이 형성됨을 의미한다.

#### 4.3 계층적 특징 학습에 대한 해석

Feature Map 시각화 결과는 CNN이 계층적으로 특징을 학습한다는 일반적인 가설을 내부 표현 수준에서 확인해준다. 초기 계층에서는 입력 이미지의 국소적인 시각 정보가 강조되며, 이후 계층으로 갈수록 이러한 저수준 특징들이 결합되어 보다 의미 있는 구조적 표현으로 발전한다.

이러한 계층적 표현 구조는 CNN이 단순한 픽셀 분류기가 아니라, 점진적으로 시각 정보를 추상화하는 모델임을 보여준다. 본 장의 분석은 이후 장에서 수행될 Grad-CAM 분석을 통해 “모델이 어디를 보고 판단했는가”를 해석하는 데 중요한 기반을 제공한다.

## 5. Model Decision Verification with XAI

### 5.1 Grad-CAM 분석의 목적

앞선 장에서는 Feature Map 시각화를 통해 CNN이 계층적으로 시각적 특징을 학습한다는 점을 확인하였다. 그러나 Feature Map은 각 계층이 어떤 종류의 특징에 반응하는지를 보여줄 뿐, 모델이 **최종 예측을 수행할 때 실제로 어떤 이미지 영역을 중요하게 활용했는지**를 직접적으로 설명하지는 못한다. 이에 본 장에서는 Grad-CAM(Gradient-weighted Class Activation Mapping)을 활용하여, 모델의 예측 과정에서 중요하게 작용한 이미지 영역을 시각적으로 확인하고자 한다.

Grad-CAM 분석의 목적은 단순히 모델의 예측 결과를 확인하는 데 있지 않다. 본 분석은 모델이 특정 클래스를 예측할 때 주목한 영역이 사람이 인식하는 객체의 의미 있는 부분과 일치하는지를 확인함으로써, 모델의 판단이 합리적인 근거에 기반하고 있는지를 검증하는 데 있다.

### 5.2 Grad-Cam 시각화 결과

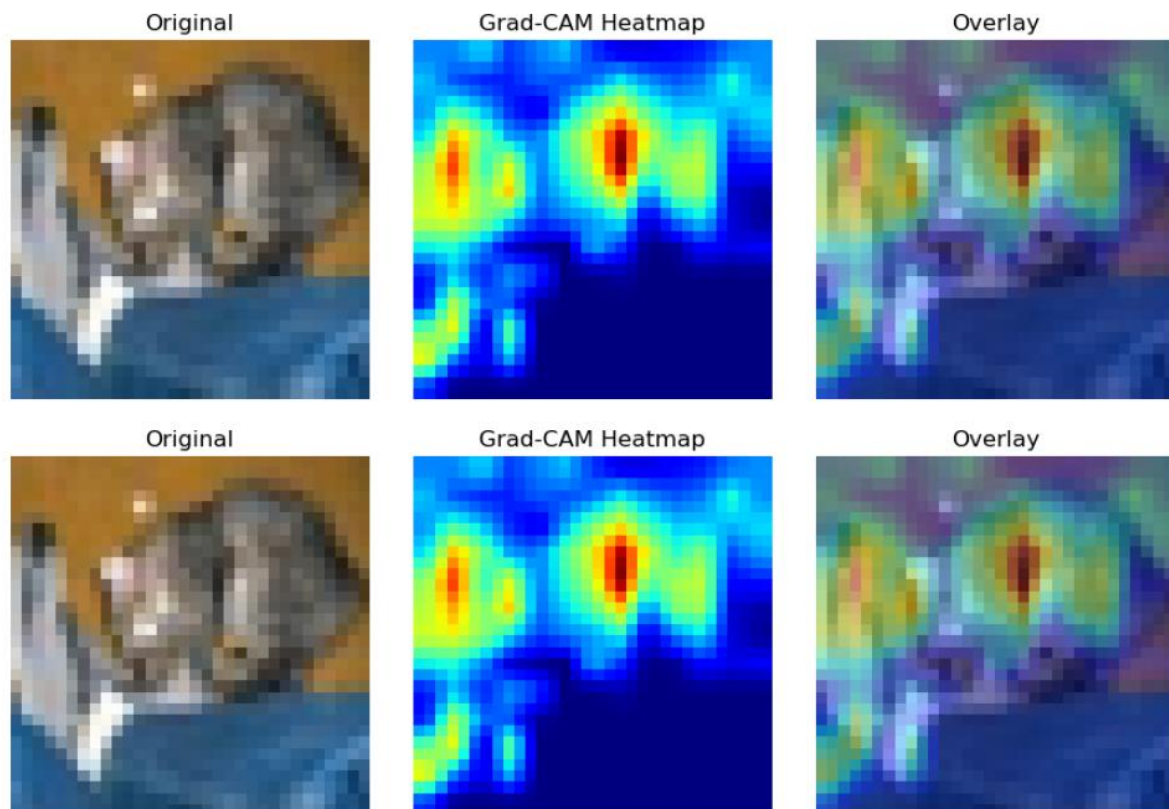


Figure 5-1. Grad-CAM visualization of model attention regions for correct predictions.

Grad-CAM 시각화 결과, 동일 클래스에 대해 모델은 입력 이미지가 달라지더라도 유사한 영역에

일관된 반응을 보였다. 특히 객체의 중심부나 윤곽이 포함된 영역에 상대적으로 강한 활성화가 관찰되었으며, 배경 전체에 무작위로 반응하는 양상은 나타나지 않았다. 이는 모델이 입력마다 임의의 영역을 참조하는 것이 아니라, 분류에 중요한 시각적 단서를 비교적 일관되게 활용하고 있음을 시사한다.

### 5.3 모델 판단의 신뢰성에 대한 해석

Grad-CAM 결과가 서로 유사하게 나타났다는 점은, 모델의 판단 기준이 입력에 따라 불안정하게 변하지 않고 일정한 패턴을 유지하고 있음을 의미한다. CIFAR-10과 같이 해상도가 낮은 데이터셋에서는 Grad-CAM의 공간적 분해능이 제한적이지만, 그럼에도 불구하고 모델이 객체와 무관한 영역에 집중하지 않는다는 점은 중요한 확인 사항이다. 본 분석을 통해 모델의 예측이 무작위적이지 않으며, 최소한의 시각적 근거에 기반하고 있음을 확인할 수 있다.

## 6. Transfer Learning as an Extension Option

### 6.1 Transfer Learning 도입 배경

앞선 장에서는 데이터 증강(Data Augmentation)을 통해 소량 데이터 환경에서도 CNN의 일반화 성능을 일정 수준까지 개선할 수 있음을 확인하였다. 그러나 데이터 증강은 기존 데이터의 변형을 활용하는 전략이기 때문에, 데이터의 절대적인 정보량이 매우 제한된 상황에서는 성능 향상에 한계가 존재할 수 있다. 이러한 조건에서 실무적으로 자주 활용되는 대안이 바로 Transfer Learning이다.

Transfer Learning은 대규모 데이터셋에서 사전 학습된 모델이 이미 학습한 일반적인 시각적 표현을 재사용함으로써, 적은 데이터만으로도 안정적인 성능을 확보하는 접근 방식이다. 본 장에서는 기존 CNN 기반 접근 방식과 비교하여 Transfer Learning이 어느 정도의 성능 개선을 제공하는지, 그리고 그 적용 방식에 따라 성능 양상이 어떻게 달라지는지를 검토한다.

### 6.2 사전 학습 모델 및 적용 전략

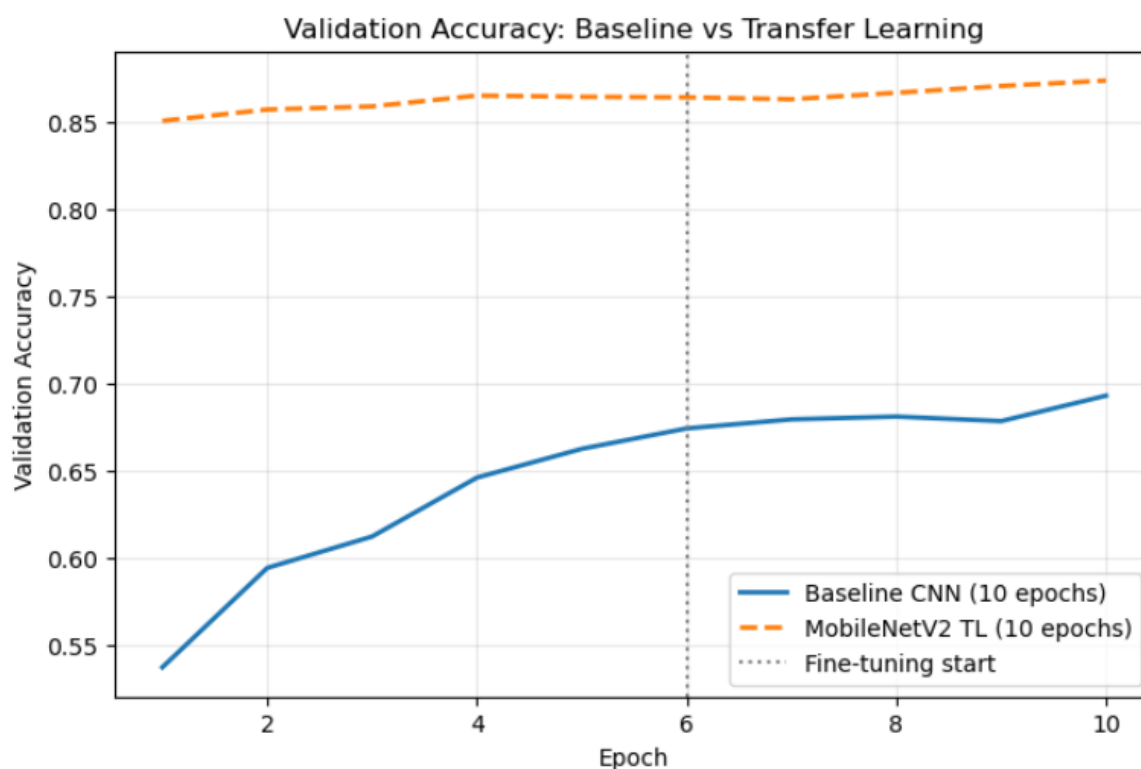


Figure 6-1. Validation Accuracy (Baseline vs Transfer Learning)

먼저 Baseline CNN과 Transfer Learning 모델의 Validation Accuracy를 비교하였다. 동일한 학습 Epoch 조건에서 비교한 결과, Transfer Learning을 적용한 MobileNetV2 기반 모델은 학습 초기부

터 Baseline CNN보다 현저히 높은 Validation Accuracy를 기록하였다. 이는 사전 학습된 모델이 이미 엣지, 텍스처, 형태와 같은 일반적인 시각적 특징을 충분히 학습하고 있기 때문으로 해석할 수 있다.

Baseline CNN의 경우 Epoch가 증가함에 따라 Validation Accuracy가 점진적으로 상승하였으나, 동일한 학습 구간에서 Transfer Learning 모델과의 성능 격차는 유지되었다. 이러한 결과는 소량 데이터 환경에서 Transfer Learning이 단순한 CNN 학습보다 훨씬 효율적으로 일반화 성능을 확보할 수 있음을 보여준다. 즉, Transfer Learning은 학습 초반부터 안정적인 성능을 제공하는 강력한 대안임을 확인할 수 있다.

### 6.3 Transfer Learning 적용 결과에 따른 해석

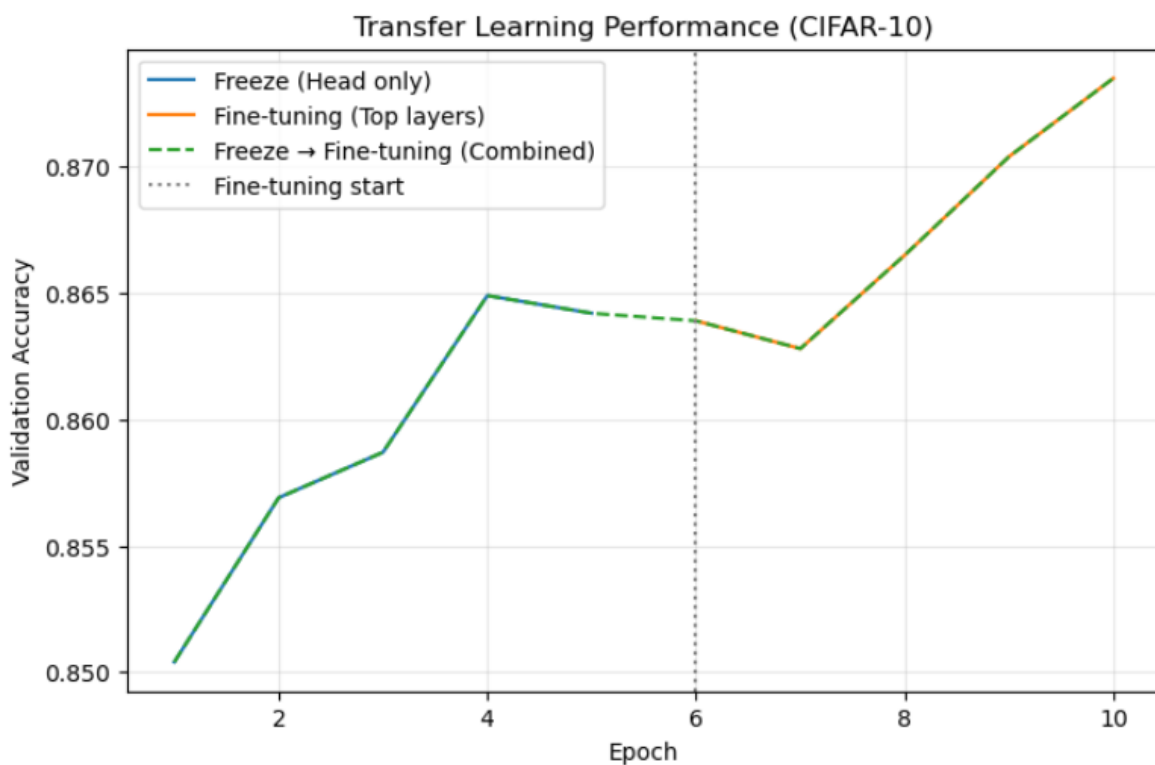


Figure 6-1. Transfer Learning Performance

Transfer Learning 적용 시에는 사전 학습된 모델의 어느 범위까지를 재학습할 것인지가 중요한 설계 요소가 된다. 이를 분석하기 위해, 본 실험에서는 특징 추출기 부분을 고정(freeze)한 상태에서 분류기만 학습하는 방식과, 상위 계층에 대해 미세 조정(fine-tuning)을 수행하는 방식을 비교하였다.

실험 결과, 초기 단계에서는 특징 추출기 부분을 고정한 상태에서도 비교적 높은 Validation 성능을 확보할 수 있었으며, 이는 사전 학습된 표현의 재사용만으로도 충분한 효과가 있음을 보여

준다. 이후 일부 상위 계층에 대해 fine-tuning을 적용한 경우, Validation Accuracy가 추가적으로 개선되는 경향이 관찰되었다. 이는 CIFAR-10 데이터의 특성이 사전 학습 데이터(ImageNet)와 완전히 동일하지 않기 때문에, 제한적인 범위의 미세 조정이 성능 향상에 기여했음을 의미한다.

다만 fine-tuning을 적용하는 시점과 범위에 따라 성능 변동이 발생할 수 있으며, 과도한 fine-tuning은 오히려 과적합으로 이어질 가능성도 존재한다. 따라서 Transfer Learning은 단순히 적용 여부뿐만 아니라, freeze와 fine-tuning의 균형을 고려하여 설계해야 하는 전략임을 확인할 수 있다.



## 7. Final Comparison Practical Decision

### 7.1 실험 조합별 성능 종합 비교

본 장에서는 앞선 장에서 단계적으로 검증한 모든 접근 방식을 하나의 기준으로 종합 비교한다. 비교 대상은 Baseline CNN, Data Augmentation을 적용한 CNN, Transfer Learning 기반 모델, 그리고 Transfer Learning과 Data Augmentation을 함께 적용한 모델로 구성된다. 이는 데이터 전략과 모델 전략이 각각, 그리고 동시에 적용되었을 때의 효과를 명확히 비교하기 위한 설계이다.

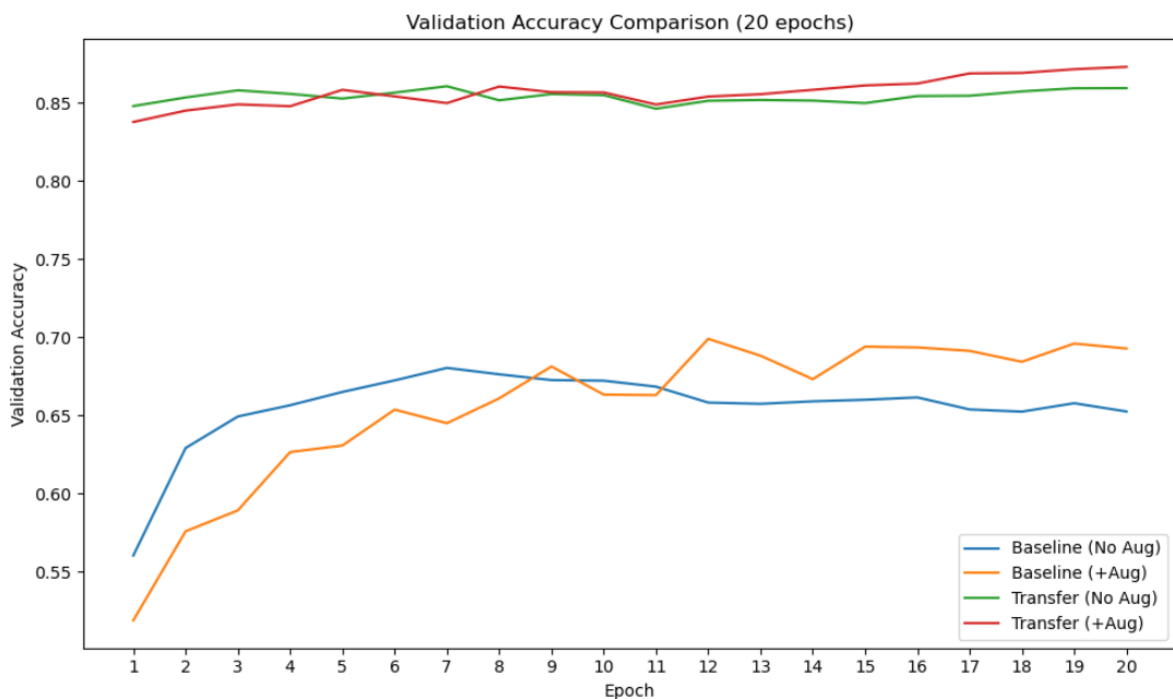


Figure 7-1. Validation Accuracy Comparison

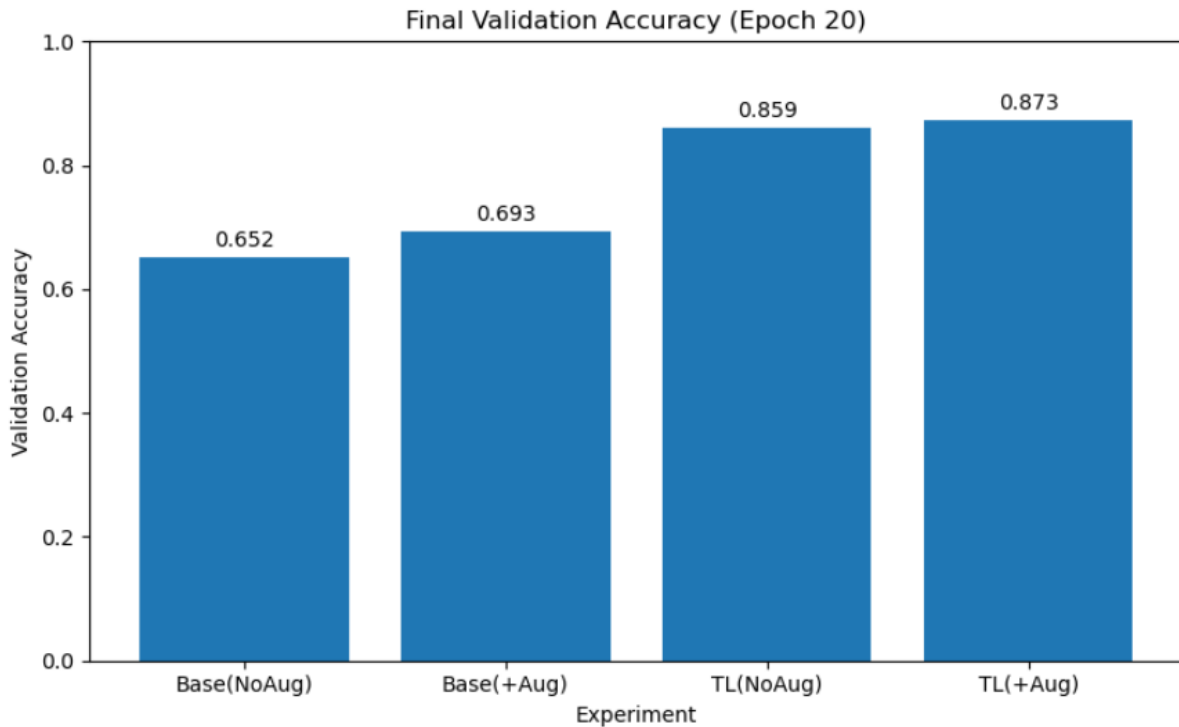


Figure 7-2. Final Validation Accuracy

Figure 7-1은 CIFAR-10 데이터셋에 대해 20 Epoch 동안 학습한 각 실험 조합의 Validation Accuracy 변화를 나타낸다. Baseline CNN은 학습 초반 빠른 성능 상승을 보였으나, 일정 Epoch 이후 성능이 정체되는 경향이 확인되었다. 이는 소량 데이터 환경에서 모델이 입력 데이터의 제한된 패턴을 빠르게 학습한 뒤, 추가적인 일반화로 이어지지 못했음을 의미한다.

Baseline CNN에 Data Augmentation을 적용한 경우, 학습 초반 성능은 상대적으로 낮게 시작하지만 Epoch가 증가함에 따라 Validation Accuracy가 점진적으로 개선되는 양상이 관찰되었다. 이는 데이터 증강이 모델의 학습 속도는 다소 늦추지만, 입력 다양성을 통해 일반화 성능을 개선하는 데 기여했음을 보여준다.

Transfer Learning을 적용한 모델은 Baseline CNN과 비교했을 때 학습 초기부터 현저히 높은 Validation Accuracy를 기록하였다. 이는 사전 학습된 모델이 이미 일반적인 시각적 특징을 충분히 내재하고 있기 때문에, 제한된 데이터 환경에서도 안정적인 성능을 확보할 수 있음을 의미한다. Figure 7-2는 Data Augmentation을 추가로 적용한 경우가 가장 높은 Validation Accuracy를 얻을 수 있음을 보여준다.

## 7.2 데이터 전략과 모델 전략의 역할 분리

종합 비교 결과는 데이터 전략과 모델 전략이 서로 다른 방식으로 성능에 기여함을 보여준다. Data Augmentation은 기존 모델 구조를 변경하지 않고도 일반화 성능을 개선하는 데 효과적이며,

특히 학습 후반부에서 성능 정체를 완화하는 역할을 수행한다. 이는 모델이 특정 입력 조건에 과도하게 의존하는 것을 방지하고, 보다 일반적인 특징을 학습하도록 유도하기 때문이다.

반면 Transfer Learning은 데이터 양이 제한된 환경에서 성능의 상한선을 빠르게 끌어올리는 역할을 한다. 사전 학습된 모델을 활용함으로써, 모델은 학습 초기부터 안정적인 성능을 확보할 수 있으며, 이는 실무 환경에서 빠른 검증과 배포에 큰 장점으로 작용한다. 즉, 데이터 전략은 “어떻게 학습할 것인가”에 영향을 미치고, 모델 전략은 “어디서부터 시작할 것인가”를 결정한다고 해석할 수 있다.

### 7.3 실무 관점에서의 최종 선택

본 프로젝트의 실험 결과를 실무 관점에서 해석하면, 소량 데이터 환경에서 가장 합리적인 선택은 **Transfer Learning을 기본 전략으로 채택하고, Data Augmentation을 보조적으로 적용하는 방식**이다. 이 조합은 높은 성능과 학습 안정성을 동시에 확보할 수 있으며, 데이터 수집이나 모델 재설계에 드는 비용을 최소화할 수 있다.

## 8. Conclusion

본 프로젝트는 소량 데이터 환경에서 CNN 기반 이미지 분류 성능을 개선하기 위한 실무적 접근을 단계적으로 검증하였다. 단순히 모델 구조를 복잡하게 만드는 대신, 데이터 증강과 Transfer Learning이라는 데이터 및 모델 전략을 중심으로 일반화 성능 개선 과정을 분석하였다.

실험 결과, Data Augmentation은 모델이 입력의 우연적 패턴에 과도하게 의존하는 것을 완화하고, 학습 후반부에서 일반화 성능을 안정적으로 향상시키는 데 효과적인 전략임을 확인하였다. 또한 Feature Map과 Grad-CAM 분석을 통해, 성능 향상이 단순한 수치적 결과가 아니라 모델 내부 표현과 판단 근거의 변화로 이어졌음을 검증하였다.

Transfer Learning은 소량 데이터 환경에서 가장 강력한 성능 향상 수단으로 작용하였으며, 특히 Data Augmentation과 결합했을 때 가장 높은 성능과 안정적인 학습 양상을 보였다. 이는 실무 환경에서 빠른 성능 확보와 재현 가능한 모델 설계를 가능하게 하는 현실적인 선택지로 평가할 수 있다.

종합하면, 본 프로젝트는 “모델을 키우기 전에 데이터와 학습 전략을 먼저 설계해야 한다\*”는 데이터 중심(Data-centric) 접근의 중요성을 실험적으로 보여준다. 본 결과는 소량 데이터 기반 이미지 분류 문제에서 실무적인 의사결정을 위한 유의미한 기준을 제공한다.

# Appendix

## A. Experimental Environment & Reproducibility

### Experimental Environment

- **Dataset:** CIFAR-10
  - 32×32 RGB 이미지, 10개 클래스
  - Train: 50,000 images (클래스당 5,000)
  - Test: 10,000 images (클래스당 1,000)
- **Framework:** TensorFlow / Keras
- **Training Epochs:** 20
- **Batch Size:** 64
- **Loss Function:** Sparse Categorical Cross-Entropy
- **Evaluation Metric:** Accuracy

### Reproducibility Settings

- **Random Seed:** 42
  - TensorFlow 및 NumPy seed 고정
- **Train / Validation Split:**
  - Training 데이터의 20%를 Validation으로 분리
  - stratify 옵션을 사용하여 클래스 비율 유지
- **Optimizer:**
  - Baseline CNN / Freeze stage: Adam (default learning rate)
  - Fine-tuning stage: Adam

## B. Data Preprocessing

### B-1. Baseline CNN Preprocessing

- **Input Size:**  $32 \times 32 \times 3$
- **Normalization:**
  - 픽셀 값을 0~255 범위에서 **0~1** 범위로 스케일링
  - $x = x / 255.0$

### B-2. Transfer Learning Preprocessing (MobileNet V2)

- **Input Size:**  $96 \times 96 \times 3$
- **Resize:**
  - CIFAR-10 이미지( $32 \times 32$ ) 를 ( $96 \times 96$ )으로 리사이즈
  - $x = x / 255.0$
- **Preprocessing Function:**
  - Preprocess\_input() (MobileNetV2 전용 정규화)
  - 입력 값을  $[-1,1]$  범위로 변환

## C. Data Augmentation Settings

Table C-1. Data Augmentation Settings (ImageDataGenerator)

Technique	Parameter	Description
Rotation	20°	이미지 회전 범위
Width Shift	0.1	가로 방향 이동(이미지 너비 대비 10%)
Height Shift	0.1	세로 방향 이동(이미지 너비 대비 10%)
Horizontal Flip	TRUE	좌우 반전 적용
Applied To	Training only	학습 데이터에만 적용(Validation / Test 제외)

## D. Model Architecture Tables

Table D-1. Baseline CNN Architecture Summary

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 64)	147,520
dense_1 (Dense)	(None, 10)	650

**Total params:** 167,562 (654.54 KB)

**Trainable params:** 167,562 (654.54 KB)

**Non-trainable params:** 0 (0.00 B)

Table D-2. CNN Architecture for Feature Map Visualization

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 32, 32, 3)	0
conv1 (Conv2D)	(None, 30, 30, 16)	448
max_pooling2d (MaxPooling2D)	(None, 15, 15, 16)	0
conv2 (Conv2D)	(None, 13, 13, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 32)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 10)	11,530

**Total params:** 16,618 (64.91 KB)

**Trainable params:** 16,618 (64.91 KB)

**Non-trainable params:** 0 (0.00 B)

Table D-3. CNN Architecture for Grad-CAM Analysis

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 32, 32, 3)	0
conv1 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
last_conv (Conv2D)	(None, 13, 13, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 64)	147,520
dense_1 (Dense)	(None, 10)	650

**Total params:** 167,562 (654.54 KB)

**Trainable params:** 167,562 (654.54 KB)

**Non-trainable params:** 0 (0.00 B)

Table D-4. Classification Head Architecture of the Transfer Learning Model

Layer (type)	Output Shape	Param #
keras_tensor_153CLONE (InputLayer)	(None, 3, 3, 1280)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dense (Dense)	(None, 64)	81,984
dense_1 (Dense)	(None, 10)	650

**Total params:** 82,634 (322.79 KB)

**Trainable params:** 82,634 (322.79 KB)

**Non-trainable params:** 0 (0.00 B)