Deep learning project

CNN vs Transfer learning

Hayley Hyejeong Lee

Workflow

Dataset: CIFAR-10

Data preprocessing

CNNs

- Baseline
- Optimization

Transfer learning

- MobileNetV2
- EfficientNetB0
- ResNet?

Model Comparison

CNNs

vs

Transfer learning

CIFAR-10 Dataset

Purpose	• A common benchmark for developing and evaluating algorithms for tasks like image classification and object recognition
Dataset structure	• In total 60000 colour images in 10 classes, with 6000 images per class • 50000 training images and 10000 test images • 32 x 32 x 3 (three color channels RGB)

CNN baseline

Simple but limited validation accuracy 69%

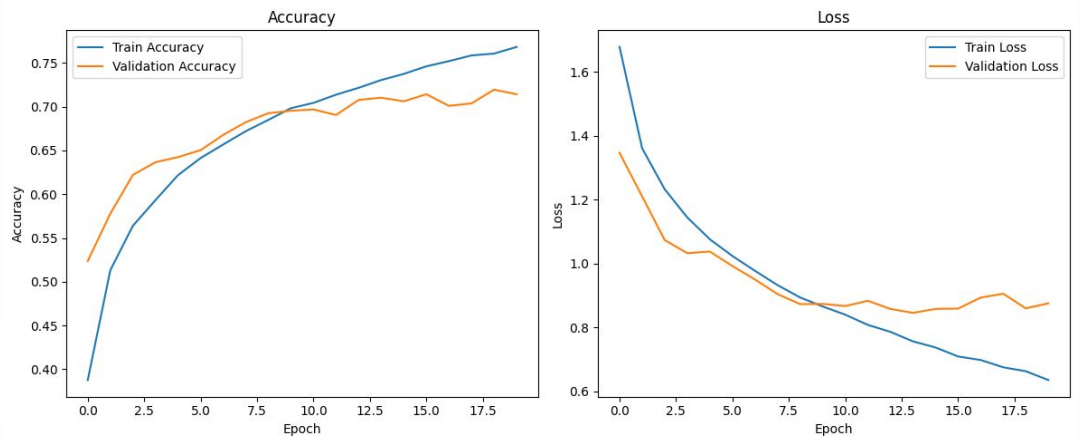
```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu',
           input_shape=(32, 32, 3)),
    MaxPooling2D(2, 2),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),

    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])
```

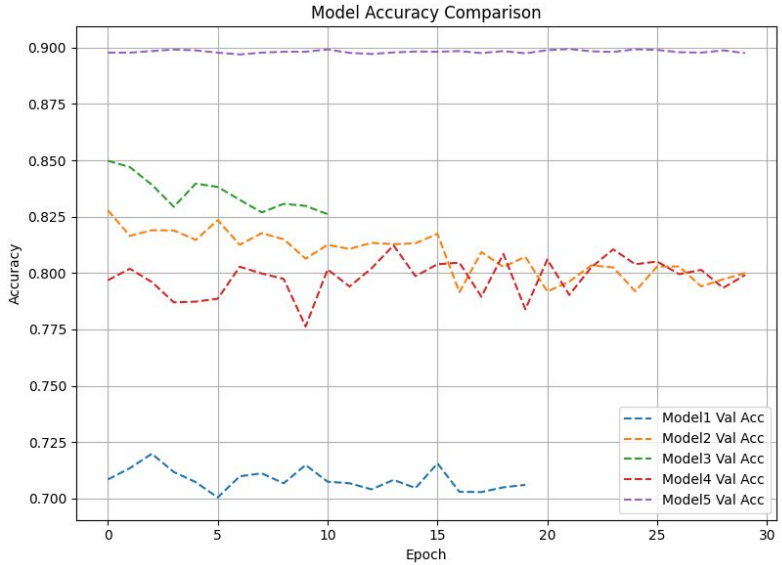
```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

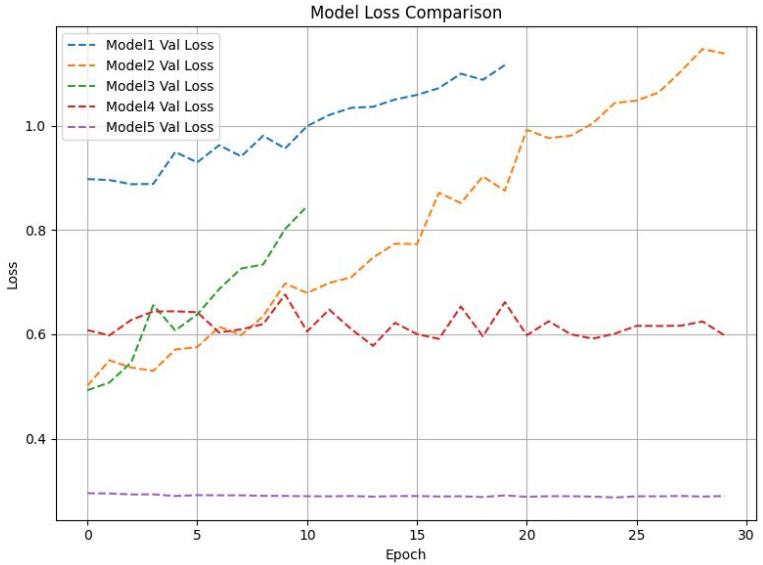
```
history = model.fit(
    X_train, y_train,
    epochs=20,
    batch_size=64,
    validation_split=0.2)
```



CNN improvement

Model 1: Baseline model **(69%)**
Model 2: Adding one layer + increased number of epochs (20 → 30) **(77%)**
Model 3: Model2 + early stopping (monitor: val_accuracy, patient: 10, epochs 60) **(80%)**
Model 4: Model2 + Data augmentation **(80%)**
Model 5: Model2 + Data augmentation/Batch Normalization/Learning rate reduction) **(86%)**

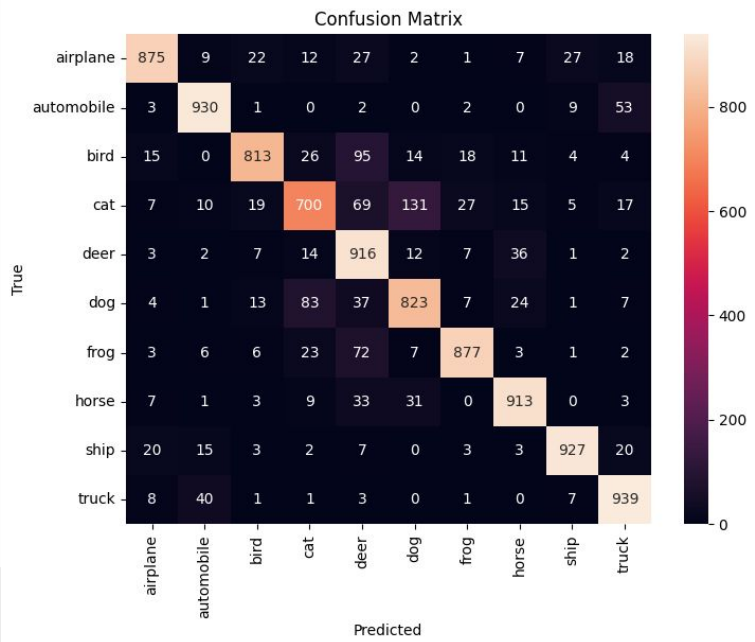


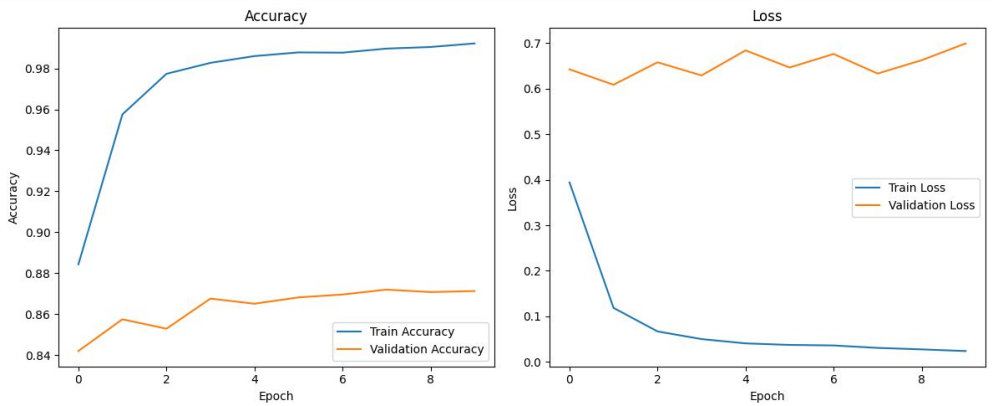


Transfer learning

MobileNetV2

- Frozen base without image resizing (32x32) (32%)
- Improvement (87%) after resize (96 x 96), unfreezing the last 20 layers, lower learning rate, but with the overfitting problem

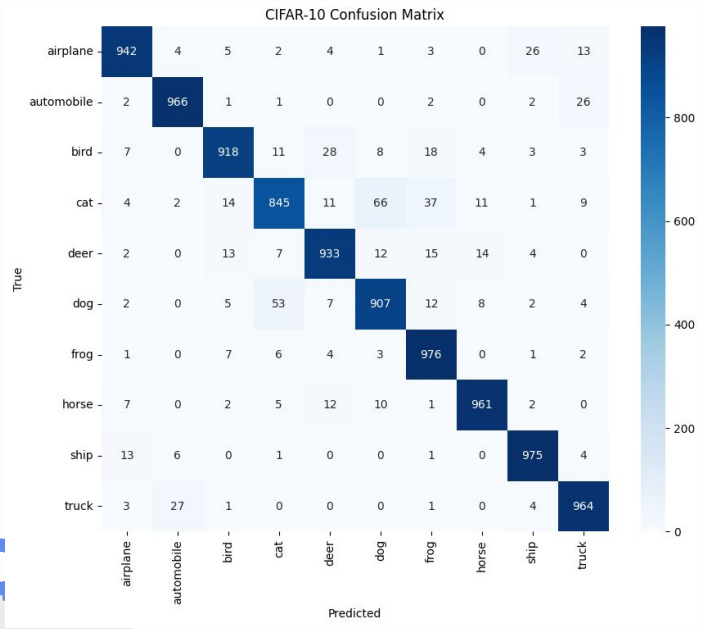


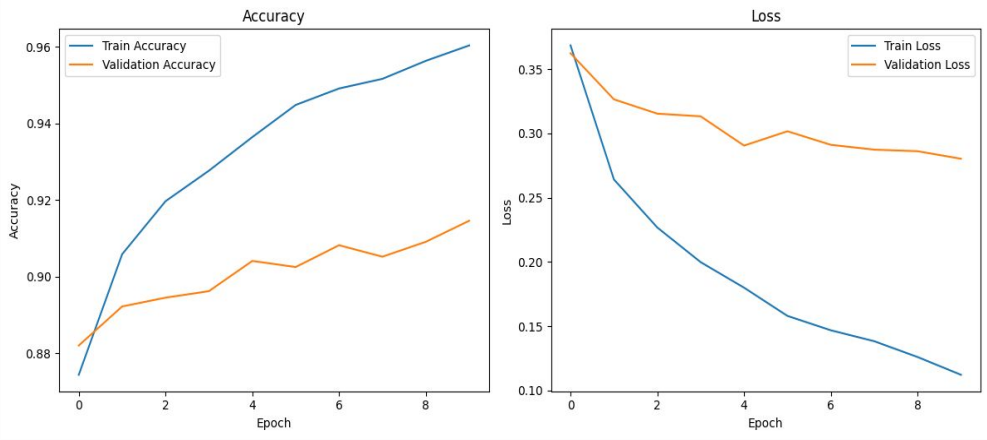


Transfer learning

EfficientNetB0

- Frozen base (90%)
- Improvement (93.8%) after data augmentation, unfreezing the last 20 layers, lower learning rate, early stopping but with the overfitting problem

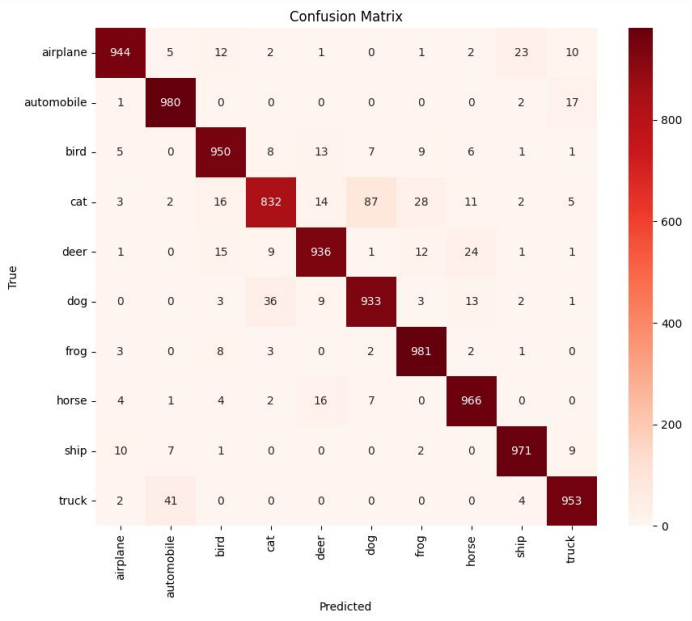


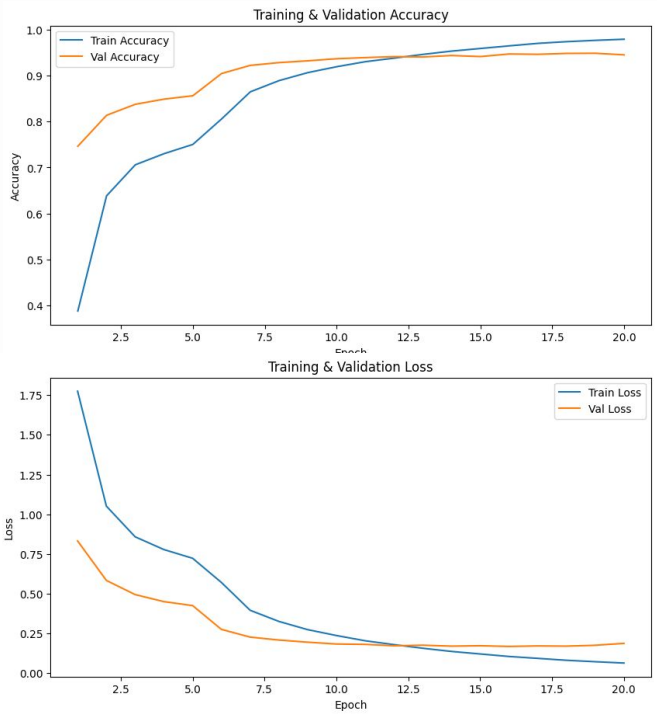


Transfer learning

ResNet50

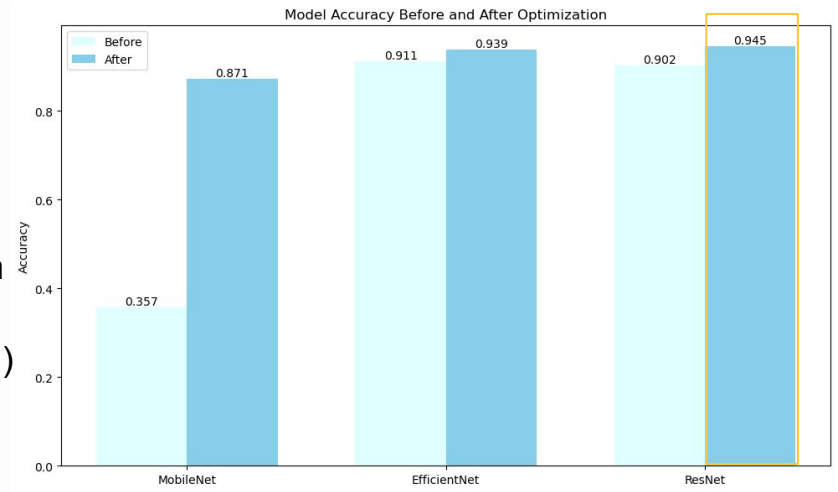
- Frozen base (90%) after resizing (224x224)
- Improvement (94.4%) after data augmentation, unfreezing the last 30 layers, lower learning rate





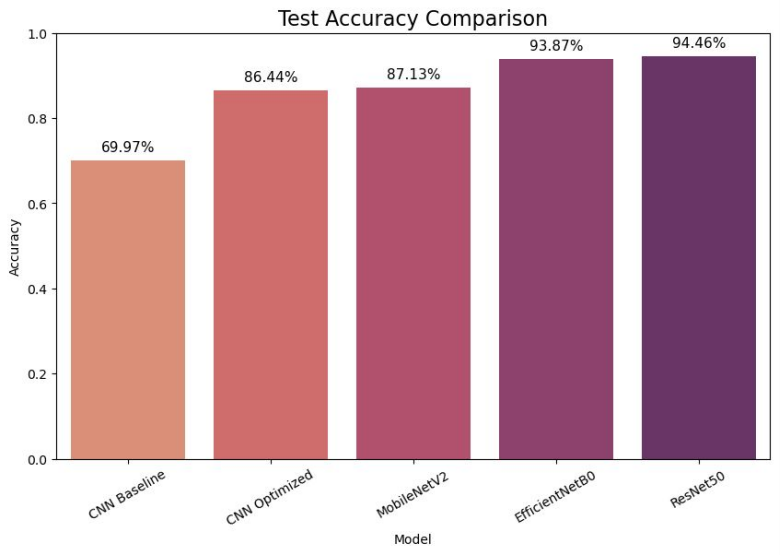
TL model comparison

- MobileNetV2 (optimized): **~87%**
- EfficientNetB0 (optimized): **~93.9%**
- ResNet50 (optimized): **~94.5%**
- Fine-tuned models consistently outperform frozen versions
- **Best overall model:** ResNet50 (optimized)



Criteria	Train acc(%)	Validation acc(%)	Train loss (%)	Validation loss (%)
MobileNetV2	99.18%	87.13%	2.47%	70.2%
EfficientNetB0	96.32%	93.87%	10.57%	19.67%
ResNet50	97.75%	94.46%	6.5%	18.76%

CNN vs Transfer learning



- Transfer learning models **outperformed all** scratch-built CNNs
- Best TL model exceeds best CNN by **8–9%**
- Lightweight TL models (MobileNetV2) already beat most CNN versions
- Fine-tuning is critical for unlocking pretrained model performance

Summary

	Advantages	Disadvantages
CNN	• Full architectural control • Lightweight • fast to train • Simpler to interpret	• Lower peak accuracy • Requires heavy hyperparameter tuning • More prone to overfitting
TL	• Higher accuracies (93–95%) • pretrained ImageNet features • Faster convergence, less manual tuning	• Higher computational cost • Less architectural flexibility • Fine-tuning can overfit if not managed

Custom CNN

- lightweight, flexible, educational,
- suitable for small-model

Transfer learning

- better performance
- used for accuracy-critical tasks

Thanks!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**