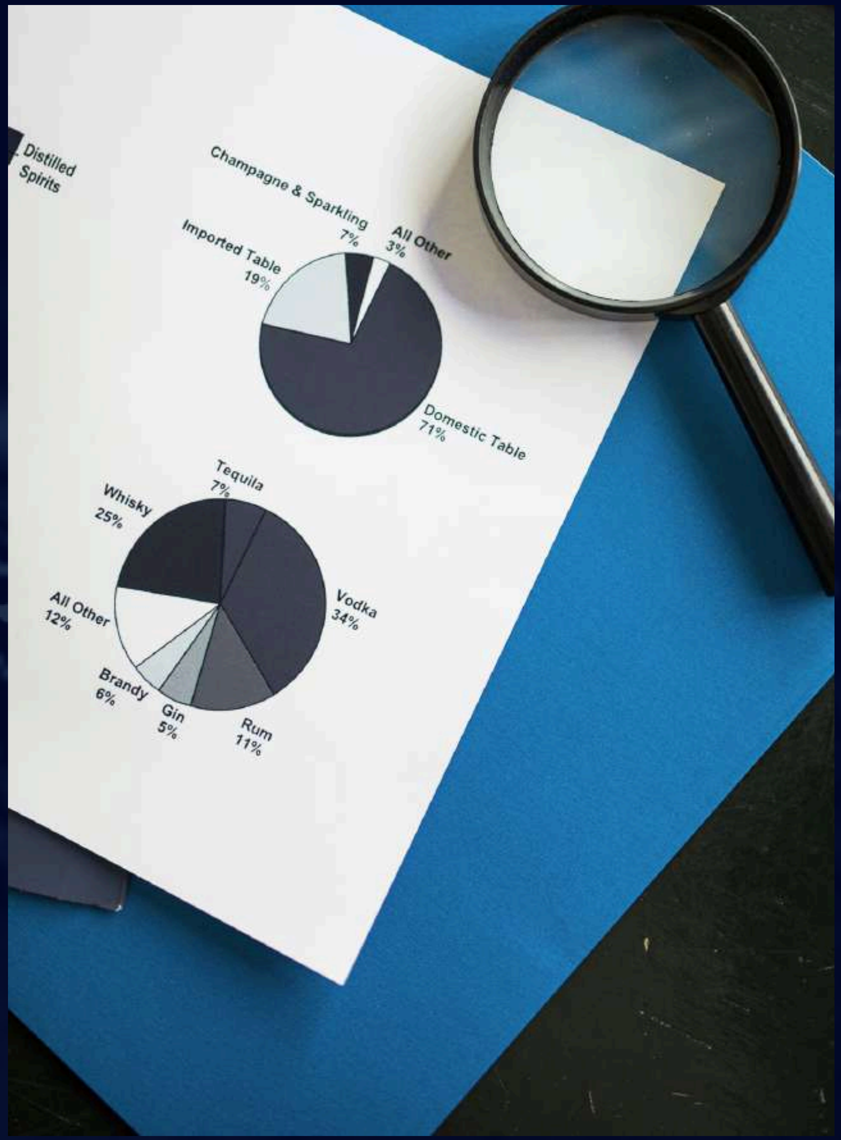
Deep Learning Approaches

CNN from Scratch vs Transfer Learning

-Jithin, Kaan, Vinicius

Project overview

A comparison between a CNN model created from scratch with
different transfer learning model to get insights on the model performances
with various evaluations and visualizations on Cifar 10 dataset.



CNN Built from the Ground Up

CNN:

Training a CNN model with Cifar10 dataset from scratch.

Approachs:

- Trained on minimal layers
- Two optimisers (Adam, SGD)
- Augumentation of data and Fine-tuning (Adding more layers)

Example Architecture:

Conv → ReLU → Pooling → Fully Connected → Softmax.

Training Setup

Dataset: CIFAR-10 (60,000 color images, 10 classes).

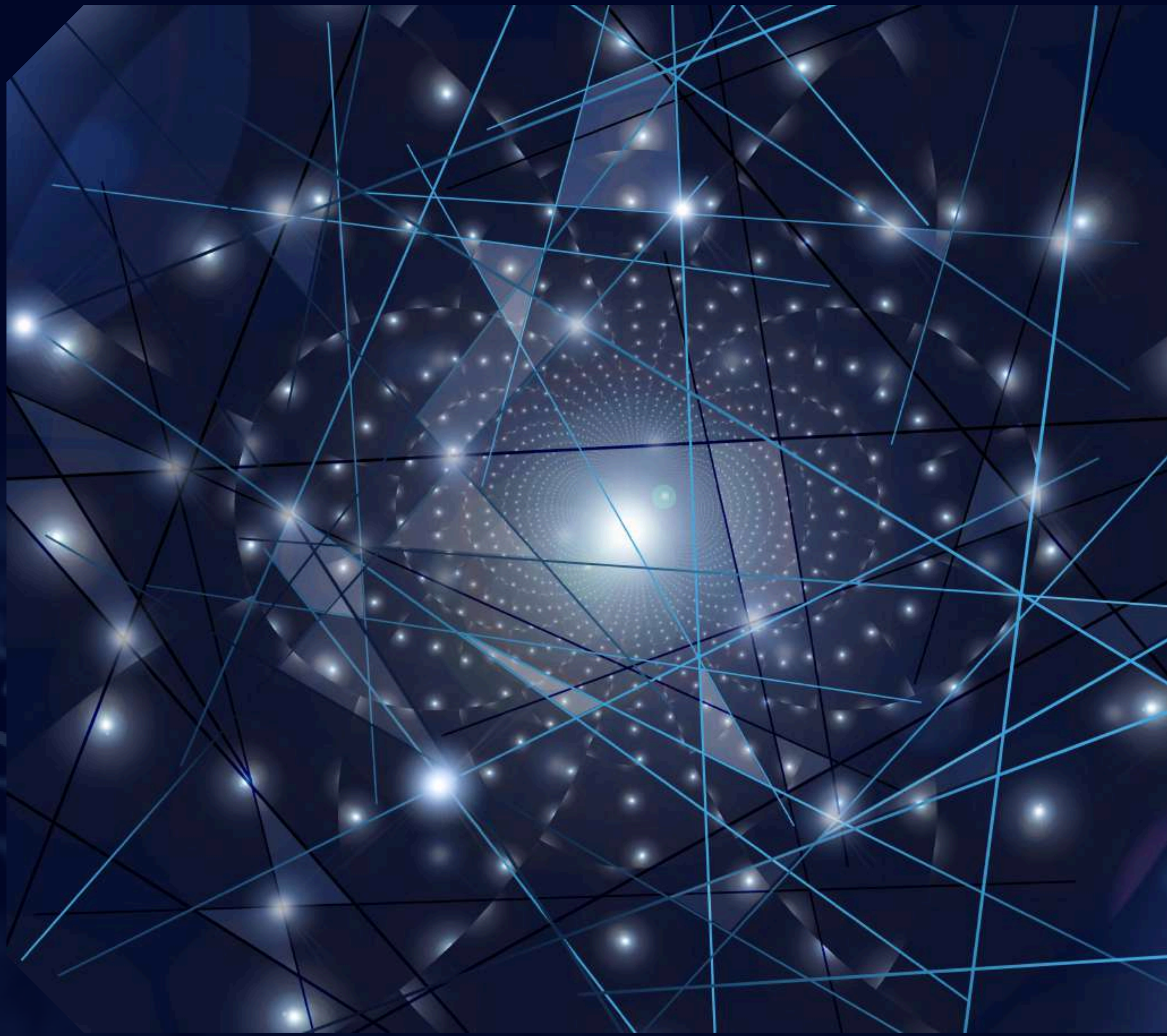
- Training: 50,000 | Testing: 10,000.

Preprocessing:

- Normalization
- One hot encoding for labels

Hyperparameters:

- Optimizer: Adam | SGD
- categorical_crossentropy



Models Used

CNN :

- Custom CNN with 3 Conv layers, pooling, dropout, dense layers.
- Simple but requires extensive training.
- Experimented with 2 optimizers
- Agumentation and more layers

Transfer Learning Models:

- **MobileNetV2:** Lightweight, optimized for speed and mobile devices.
- **DenseNet:** Deep network with dense connections that reuse features for efficient learning.
- **EfficientNetB4/B5:** Scaled CNN models that balance depth, width, and resolution for high accuracy with fewer parameters.

Evaluation Results

Variant	Optimizer	Train Accuracy-Loss	Test Accuracy-Loss
CNN (Base)	Adam	0.8013(0.5747)	0.7175(0.8246)
CNN (Augmented)	Adam	0.7390(0.7496)	0.7212(0.8082)
CNN (More layers)	Adam	0.8540(0.4230)	0.7192(0.8402)
CNN (Aug-more layers)	Adam	0.7674(0.6676)	0.7339(0.7787)
CNN (Base)	SGD	0.7718(0.6646)	0.6733(0.9616)
CNN (Augmented)	SGD	0.8142(0.5289)	0.7733(0.6475)
CNN (More layers)	SGD	0.7934(0.6030)	0.6734(0.9349)
CNN (Aug-more layers)	SGD	0.7687(0.6609)	0.7398(0.7635)

Insights from evaluations

We tested Adam vs SGD on four CNN variants. Adam consistently outperformed SGD on base models, reaching 71–72% test accuracy. However, SGD benefitted more from data augmentation, jumping from 67% to 77% test accuracy. Adding more layers improved training accuracy but caused overfitting, unless combined with augmentation. Overall, augmentation is critical for deeper CNNs, while Adam gives better baseline performance without heavy tuning.”

Transfer Learning

Using a model trained on a large benchmark dataset (e.g., ImageNet with 14M images) and adapting it to a new task.

Why it works:

- Pretrained CNNs already learn universal features (edges, shapes, textures).
- These features can be reused for similar tasks.

Methods:

- Feature Extraction: Freeze pretrained layers; train only the last classifier.
- Fine-tuning: Retrain some layers to adapt features for the new dataset.

Benefits:

- Works with smaller datasets.
- Faster convergence & better accuracy.
- Reduces need for heavy computation.

Mobilenetv2

- Image resized - 224 x 224 x 3
- Base model with two optimizers and fine tuning

Efficientnetb4/b5

- Image resized - 380 x 380(B4) , 456 x 456 (B5)
- only Adam

Densenet

- Image resized - 224 x 224 x 3
- Base model with two optimizers

Evaluation Results

Variant	Optimizer	Train Accuracy-Loss	Test Accuracy-Loss
MobilenetV2	Adam	0.8744(0.3587)	0.8445(0.4452)
MobilenetV2 Finetuned		0.8744(0.3587)	0.8604(4091)
Densenet		0.8398(0.4761)	0.8343(0.4979)
Efficientnet B4		0.9531(0.1383)	0.9376(0.1873)
Efficientnet B5		0.9534(0.1365)	0.9335(0.1933)
MobilenetV2	SGD	0.9063(0.2690)	0.8599(0.4067)
Mobilenet V2 Finetuned		0.9124(0.2567)	0.8619(0.3996)

Insights

- EfficientNet (B4 & B5) clearly outperforms all other models in both train and test accuracy, with very low loss.
- Fine-tuning MobileNetV2 improves test accuracy slightly compared to the base version, indicating transfer learning benefits.
- **SGD vs Adam on MobileNetV2:** SGD gives slightly higher train accuracy, but Adam performs similarly or better on test data.
- Densenet performs well but is slightly behind EfficientNet, showing that deeper architectures may not always outperform larger pretrained networks on CIFAR-10.
- Overall, EfficientNetB4 strikes the best balance between train and test performance, making it the most reliable model here.

Conclusion

- Observation from Custom CNN (Adam & SGD):
- Training accuracies range from ~0.72 to 0.85, with test accuracies generally lower (~0.67 to 0.84).
- Models with more layers or data augmentation slightly improve train/test performance.
- SGD sometimes gives better generalization (smaller gap between train and test accuracy), but Adam achieves higher training accuracy.
- Overall, even the best custom CNN struggles to consistently reach >0.85 test accuracy.
- Observation from Transfer Learning Models:
- EfficientNetB4/B5 achieved ~0.93–0.94 test accuracy, much higher than any CNN variant.
- MobileNetV2, even without fine-tuning, reaches ~0.84–0.86 test accuracy. Fine-tuning further improves it slightly (~0.86).

Conclusion

- Densenet achieves ~0.83 test accuracy, still competitive but lower than EfficientNet.
- Transfer learning models converge faster, achieve higher accuracy, and generalize better than custom CNNs.
- Conclusion / Key Insights:
- Transfer learning is more effective than training from scratch for CIFAR-10, especially with complex architectures like EfficientNet.
- EfficientNetB4/B5 clearly outperform all custom CNN variants in both train and test performance.
- Fine-tuning pre-trained models can further improve performance slightly, without requiring changes to the architecture.
- Custom CNNs are useful for learning basics and experimentation, but for high accuracy and fast convergence, pre-trained models are preferred.

Thank You

-Jithin, Kaan, Vinicius