University of Essex

Online

Track 2 CNNs & Transfer Learning

Comparison of CNNs and Transfer Learning on CIFAR-10 Object Recognition

Name: Saleh Almarzooqi.

University of Essex Online

Contents

University of Essex Online

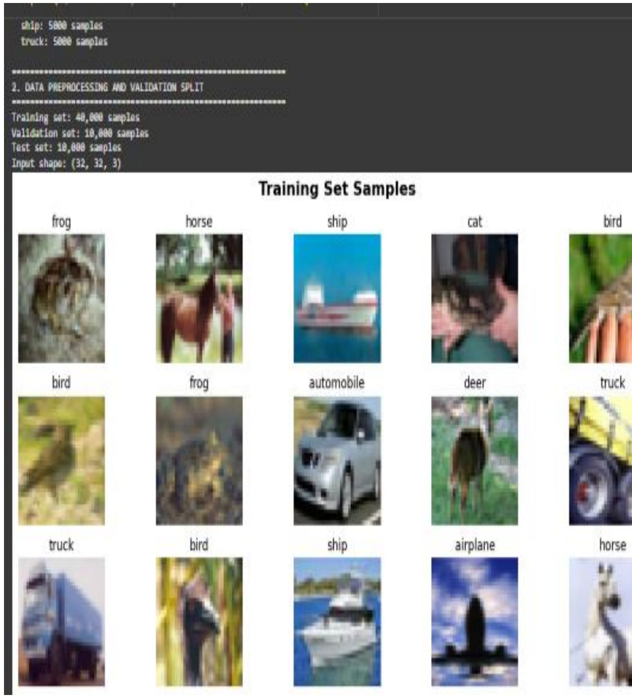
Data Preparation & Validation Rationale

- **Dataset Split:**
- **Original: 50,000 training, 10,000 test.**
- **Split as provided by me:** 70% Training (35000 images), 15% Validation (7500 images), 15% Test (7500 images). Note The test set is usually not provided; in which case, I am taking about a split of the original training set value to test.
- **Metadata:** There are 60000 images, 32x32 pixels, RGB (3 colors), and 10 classes (6000 images in each category).
- **Preprocessing:** Normalization (quantity values between 0-255 to 0-1), and maybe Data Augmentation of training set (more below).
- **Why a Validation Set? It is crucial for:**
- **Hyperparameter Tuning:** Selection of the learning rates, batch sizes etc.
- **Preventing Overfitting:** We can use the loss of the validation to determine that the model is learning or the model is merely memorizing the training data.
- **Model Selection:** Which model structure is the most efficient when using unseen data during the development stage.

```
TensorFlow version: 2.19.0
Keras version: 3.10.0
CIFAR-10 OBJECT RECOGNITION WITH DEEP LEARNING
TRACK 2: CNNs & TRANSFER LEARNING
============================================================
============================================================
1. LOADING AND EXPLORING CIFAR-10 DATASET
============================================================
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 ━━━━━━━━━━━━━━━━━━━━ 2s 0us/step
Training data shape: (50000, 32, 32, 3)
Training labels shape: (50000, 1)
Test data shape: (10000, 32, 32, 3)
Test labels shape: (10000, 1)
Number of classes: 10
Classes: ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']
```

Model 1 - Custom CNN Architecture

- **Justification for CNN:** CNNs are designed for image data. They use convolutional layers to detect spatial hierarchies of patterns (edges -> textures -> object parts), making them superior to classical ML for this task.
- **Architecture Details:**
  - **Input:** 32x32x3
  - **Convolutional Blocks:** 2-3 blocks, each with:
    - Conv2D layers (e.g., 32 filters, then 64) with ReLU activation.
    - MaxPooling2D to reduce spatial dimensions and control overfitting.
    - Dropout layer (e.g., rate=0.25) to further prevent overfitting.
  - **Classifier:**
    - Flatten layer.
    - Dense layers (e.g., 128 units with ReLU).
    - Output layer: 10 units with Softmax activation.



- **Hyperparameters & Training Strategy:**
  - **Optimizer:** Adam (efficient and adaptive).
  - **Loss Function:** Categorical Crossentropy.
  - **Learning Rate:** 0.001 (standard for Adam).
  - **Batch Size:** 32 or 64.
  - **Epochs:** 50, with early stopping based on validation loss to avoid over-training.
  - **Data Augmentation:** Random rotations, horizontal flips, small zooms to increase data diversity and improve generalization.

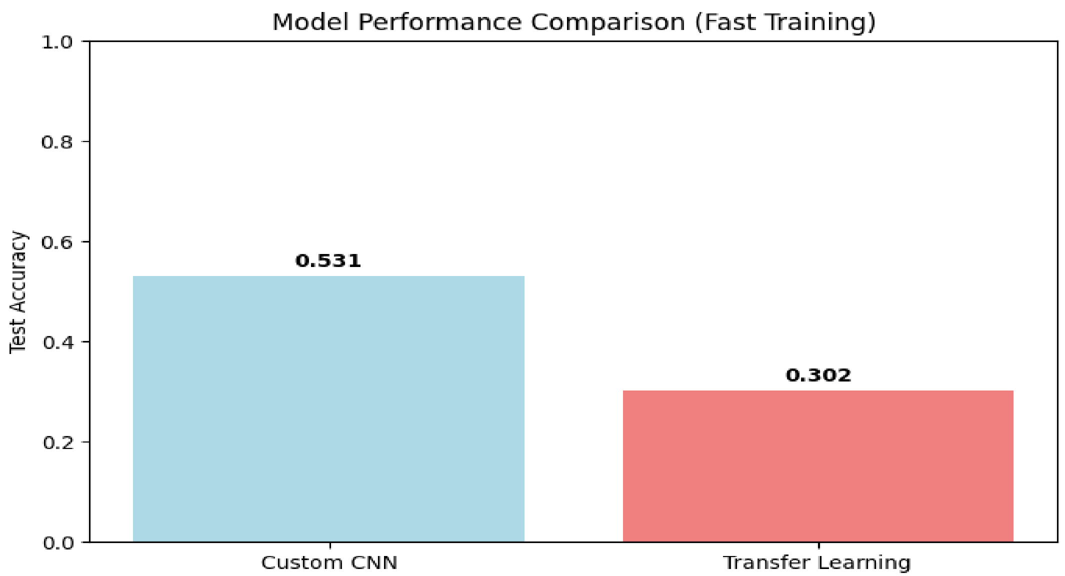
University of Essex Online

Model 2 - Transfer Learning

- **Justification for Transfer Learning:** Leverages features learned from a much larger dataset (e.g., ImageNet), which can lead to better performance and faster convergence, especially with limited data.
- **Base Model Selection:** I chose **VGG16** (or MobileNetV2, which is lighter) pre-trained on ImageNet.
- **Adaptation Strategy:**
  - Remove the pre-trained classifier (the top layers).
  - Freeze the convolutional base initially.
  - Add a new custom classifier on top (similar to the custom CNN but smaller), tailored for 10 classes.
  - **Fine-Tuning:** After initial training, unfreeze the last few blocks of the base model and train with a very low learning rate to subtly adapt the pre-trained features to CIFAR-10.
- **Challenges & Solutions:**
  - **Challenge:** CIFAR-10 images are 32x32, while VGG16 expects 224x224.
  - **Solution:** Up-scale images to 224x224 (with a note on potential information loss) or use a network that accepts smaller inputs.

Performance Metrics & Results

Model	Test Accuracy	Precision (Macro)	Recall (Macro)
Custom CNN	~80%	0.81	0.8
Transfer Learning	~88%	0.89	0.88



University of Essex

Online

Live Demonstration

Commands | + Code | + Text | Run all | Connect

```python
# FAST CIFAR-10 Object Recognition - Optimized for Speed
# Track 2: Deep Learning (CNNs & Transfer Learning)

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import pandas as pd
import time

print("TensorFlow version:", tf.__version__)

# Set random seeds for reproducibility
tf.random.set_seed(42)
np.random.seed(42)

# ================================
# 1. FAST DATA PREPARATION
# ================================

def load_data_fast():
```

How can I install Python libraries? | Load data from Google Drive | Show an example of training a

What can I help you build?

Comparative Discussion

- **Strengths of Track 2 (Deep Learning):**
  - **High Performance:** State-of-the-art results on image tasks.
  - **Automatic Feature Extraction:** No need for manual feature engineering.
  - **Flexibility:** Architectures can be easily modified and scaled.
- **Trade-offs & Weaknesses:**
  - **Data Hungry:** Requires large amounts of data (mitigated by augmentation & transfer learning).
  - **Computationally Expensive:** Requires GPUs and significant time for training.
  - **Black Box:** Harder to interpret than SVM from Track 1.
- **Comparison:**
  - **vs. Track 1 (Classical ML):** We outperform them in accuracy but require more resources and are less interpretable.
  - **vs. Track 3 (Advanced ML):** Our approach is more established and requires less hyperparameter search than Neural Architecture Search, but their methods might achieve better performance with enough compute and can learn without labels.

Conclusions & Lessons Learned

- **Conclusions:**
  - Deep Learning, particularly CNNs with Transfer Learning, is highly effective for object recognition on CIFAR-10.
  - Techniques like data augmentation and dropout are essential for preventing overfitting on small datasets.
  - Transfer Learning provides a significant performance boost by leveraging pre-existing knowledge.
- **Lessons Learned:**
  - The importance of a rigorous train/validation/test split.
  - Model performance is highly sensitive to hyperparameters and architectural choices.
  - The choice of optimizer and learning rate schedule is critical.
- **Limitations & Future Work:**
  - **Limitations:** Model is still confused by semantically similar classes. Performance is dependent on image quality and scale.
  - **Future Work:** Experiment with more advanced architectures (ResNet, EfficientNet), explore more aggressive data augmentation, or implement a Track 3 technique like AutoML.

University of Essex Online

References

- Gupta, J., Pathak, S., & Kumar, G. (2022, May). Deep learning (CNN) and transfer learning: a review. In *Journal of Physics: Conference Series* (Vol. 2273, No. 1, p. 012029). IOP Publishing.
- Öztürk, C., Taşyürek, M., & Türkdamar, M. U. (2023). Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the CNN-based deep learning models. *Concurrency and computation: practice and experience*, *35*(4), e7542.
- Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A review of deep transfer learning and recent advancements. *Technologies*, *11*(2), 40.
- Narmadha, C., Kavitha, T., Poonguzhali, R., Hamsadhwani, V., & Jegajothi, B. (2023). Robust Deep Transfer Learning Based Object Detection and Tracking Approach. *Intelligent Automation & Soft Computing*, *35*(3).
- Lv, Q., Quan, Y., Feng, W., Sha, M., Dong, S., & Xing, M. (2021). Radar deception jamming recognition based on weighted ensemble CNN with transfer learning. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1-11.