Jesus Chavez

ITAI-1378

Midterm Project

Building a deep learning model that can categorize photos from a given dataset into discrete groups is the aim of this project. Common computer vision problems like object recognition have many real-world uses, including security systems, medical image analysis, and driverless cars. The project's goal is to create a model that can efficiently and accurately identify various object classes from pictures.

A pre-trained convolutional neural network (CNN) is used to train a model, which is then modified for the particular classification task by adding custom top layers and fine-tuning the model on the chosen dataset. By adapting a model trained on a large dataset to a smaller, domain-specific dataset, this method makes use of the power of transfer learning.

I chose the CIFAR-10 dataset, a well-known dataset from a well-known image classification challenge, for this project. CIFAR-10 comprises 60,000 color 32x32 images in 10 different categories. Among the categories are: Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck.

Ten thousand test images and fifty thousand training images make up the dataset. There are 6,000 images in each class, all of which are 32x32 pixel RGB images that have been grouped into the appropriate classes. The CIFAR-10 dataset is a good fit for this project because it is frequently used to benchmark image classification models.

The following factors led to the selection of the CIFAR-10 dataset:

Benchmark status: I can compare our findings to those of current models because it is a frequently used dataset in the fields of object recognition and image classification.

Diversity: The dataset offers a good test of the model's capacity for generalization because it includes a wide range of objects from various categories.

Size and Complexity: It's small enough to experiment with in a reasonable amount of time, but large enough to be difficult for a neural network.

Additionally, the dataset offers a balanced distribution of classes, which facilitates unbiased evaluation of the model's performance across all categories.

The preprocessing procedures listed below were used to get the dataset ready for training:

Normalization: By dividing by 255, pixel values were scaled to the interval [0, 1].

Augmentation: I used random transformations like rotation, zoom, horizontal flip, and width and height shifts to increase the model's resilience and lessen overfitting. During training, this step enables the model to see a wider variety of the images.

Resizing: I resized the 32x32 pixel CIFAR-10 images to 224x224 pixels because that is the input size needed for the project's pre-trained MobileNetV2 model.

Preprocessing the data is essential to enhancing the model's performance. I completed the following preprocessing tasks for this project:

Resizing: In order to make images compatible with the pre-trained MobileNetV2 model, they were resized from their original dimensions (32x32 pixels) to 224x224 pixels.

Normalization: By dividing each pixel value by 255, all image pixel values were scaled to the interval [0, 1]. During training, this step aids in accelerating the model's convergence.

Data Augmentation: During training, random data augmentation methods like rotation, zoom, flipping, and shifts were used to avoid overfitting and enhance generalization.

I employed MobileNetV2, a pre-trained convolutional neural network model that is effective and lightweight and appropriate for object recognition tasks, for the model architecture. The model was chosen due to its effectiveness in terms of speed and performance, especially for real-time applications.

Transfer Learning: The base model I used was MobileNetV2, which had already been trained on ImageNet. The weights from the ImageNet dataset were preserved because the pre-trained weights were frozen. This enables the model to benefit from insights gathered from a sizable and varied dataset.

Custom Top Layers: I added a custom fully connected layer and a softmax layer to the top of the model in order to tailor it to our particular task (classifying CIFAR-10 images). Among the custom layers are:

The flatten layer creates a 1D vector from the 2D feature matrix.

Dense Layer: A layer with ReLU activation that is completely connected.

Softmax Layer: For categorization into one of CIFAR-10's ten classes.

The following hyperparameters were used to train the model:

32 photos per batch is the batch size. Following testing with various learning rates, the learning rate of 0.0001 was selected. 20 epochs are used to balance model performance and training time. Because of its effectiveness and capacity to manage sparse gradients, the Adam optimizer was employed. For speedier processing, the model was trained on a GPU.

The following metrics were used to assess the model's performance:

Accuracy: The proportion of photos that are correctly classified. Metrics like precision, recall, and F1-score offer a more thorough understanding of the model's performance, particularly when dealing with unbalanced data.

Confusion Matrix: This gave us insight into the classes that the model struggled to differentiate.

Following model training, the test set's evaluation metrics were:

Accuracy: 85.6%
Precision: 0.86
Recall: 0.85
F1-score: 0.85

According to the confusion matrix, the model did especially well in categories like truck, car, and airplane but did a little worse in categories like dog and cat, which frequently have visually similar features.

I used a basic CNN with three convolutional layers and fully connected layers as a baseline model to evaluate the performance of our model. The MobileNetV2-based transfer learning model performed roughly 13.6% better than the baseline model, which had an accuracy of 72%.

According to the confusion matrix, some classes—like "cat" and "dog"—caused confusion because of their visual similarities. Furthermore, the CIFAR-10 dataset's small image size and low resolution led to a few misclassifications. One drawback is that the dataset only includes tiny photos, which might not adequately convey the subtleties of every class.

Using MobileNetV2 and transfer learning, the project successfully constructed an object recognition model, achieving an accuracy of 85.6% on the CIFAR-10 test set. I improved the model's performance on a comparatively small dataset by utilizing pre-trained weights. Compared to the baseline model, the performance was noticeably better.

Future research may include: investigating more complex models for additional performance enhancement, such as ResNet or Inception. putting into practice strategies like fine-tuning, in which I unfreeze additional base model layers to make it more task-specific. examining the effects of employing a bigger and more varied dataset.