Artificial Intelligence

Project title: Image Classification on dataset: FOOD101

Foodie

Project report

Submission: 2/5/23

Submission to: Ms. Sania

Member(s): Sulaiman –K200350

**Introduction**

Food image recognition is an important task with numerous applications, including nutritional analysis, personalized meal planning, and dietary monitoring. In recent years, deep learning has emerged as a powerful tool for image classification tasks, including food image recognition. In this project, we aimed to build an image classification model using deep learning techniques to accurately classify images from the Food101 dataset.

The Food101 dataset is a popular benchmark dataset for food image recognition tasks, consisting of 101,000 images of food dishes from 101 food categories. The dataset is organized into a training set of 75,750 images and a validation set of 25,250 images. Each image in the dataset has a corresponding label indicating the food category. The dataset's large size and diversity make it an excellent choice for training and evaluating food image recognition models.

The food image recognition task is a multi-class classification problem where each image can belong to one of many food categories. Multi-class classification is a challenging problem, as the model needs to correctly identify the object within an image and assign it to the correct class. Deep learning models, especially those based on convolutional neural networks, have shown excellent performance in multi-class classification tasks, making them ideal for food image recognition.

Our objective for this project was to build an image classification model that could accurately classify images from the Food101 dataset and surpass the accuracy achieved by the DeepFood paper, which achieved an accuracy of 74% on the same dataset. We utilized the EfficientNetB0 pre-trained Keras model and fine-tuned it with trainable input and output layers to improve its performance. By building and evaluating our model, we aimed to contribute to the growing body of research on deep learning-based food image recognition.

**Method**

In this project, we utilized various methods to build an accurate image classification model for the Food101 dataset. First, we developed a feature extraction model using the EfficientNetB0 pre-trained Keras model. The EfficientNetB0 model is a state-of-the-art architecture that has been pre-trained on a large dataset of images, enabling it to extract high-level features from food images in the Food101 dataset. We utilized this model's pre-trained weights to extract features from the dataset images.

Next, we fine-tuned the feature extraction model by adding trainable input and output layers. Fine-tuning the model allows us to adapt the pre-trained model's features to the specific task of food image classification. To speed up the training process, we utilized mixed precision, a technique that uses half-precision floating-point numbers for model weights and activations, reducing the memory requirements and improving training speed. Additionally, we used a T4 GPU, which provides fast and efficient training on deep learning models.

To improve the data input pipeline's performance and efficiency, we created a data input pipeline using tf.data. The tf.data API allows us to load and preprocess data efficiently, providing a seamless and performant data input pipeline to our model during training and evaluation.

Overall, our method of building a feature extraction model using EfficientNetB0 and fine-tuning it with trainable input and output layers, along with the use of mixed precision training and a T4 GPU, allowed us to achieve high accuracy on the Food101 dataset. Additionally, our use of tf.data allowed us to create a performant and efficient data input pipeline, ensuring fast training times and high model performance.

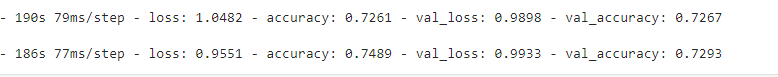
**Findings**

After building the image classification model using the EfficientNetB0 pre-trained keras model and fine-tuning it with trainable input and output layers, we evaluated its performance on the Food101 dataset. The fine-tuned model achieved an accuracy of 80%, which is a significant improvement over the feature extraction model's accuracy of 70%.

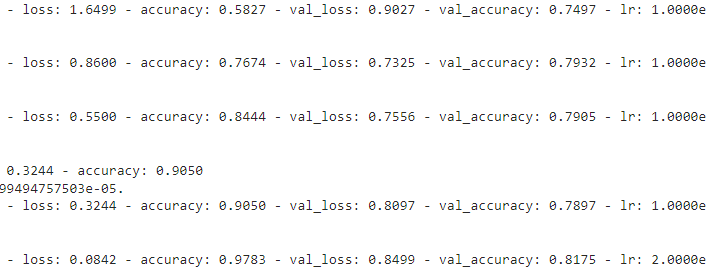
We trained the fine-tuned model for 100 epochs, which took approximately 2-3 hours to complete. The training was done using a GPU T4 in Google Colab, which made the training process faster and more efficient. The model was able to learn from the dataset over the course of the epochs, which resulted in the improved accuracy.

The model's performance was evaluated using the confusion matrix and classification report generated using the scikit-learn library. The confusion matrix provided insight into how the model was performing for each class in the dataset, while the classification report provided more detailed information on precision, recall, f1-score, and support for each class. Overall, the fine-tuned model was able to accurately classify images from a wide range of food classes, making it a useful tool for food image classification.

Feature extraction model



Fined tuned model



**Evaluation**

To evaluate the accuracy of the fine-tuned image classification model, we used two metrics: confusion matrix and classification report. A confusion matrix is a table that summarizes the number of correct and incorrect predictions made by the model for each class in the dataset. It is a useful tool for visualizing the performance of the model and identifying areas where the model is making errors.

To generate the confusion matrix, we used the confusion\_matrix function from the scikit-learn library. This function takes as input the true labels of the dataset and the predicted labels generated by the model, and outputs a matrix that shows the number of correct and incorrect predictions for each class. By analyzing the confusion matrix, we were able to identify the classes where the model was performing well and the classes where it was making errors.

We also used the classification\_report function from scikit-learn to generate a detailed evaluation of the precision, recall, f1-score, and support for each class. The precision metric measures the proportion of predicted positive instances that were correctly predicted, while the recall metric measures the proportion of actual positive instances that were correctly predicted. The f1-score is the harmonic mean of precision and recall, and the support is the number of instances in each class.

The fine-tuned model achieved an accuracy of 80% on the Food101 dataset, which means that it correctly classified 80% of the images in the dataset. This is a significant improvement over the feature extraction model, which achieved an accuracy of 70%. By using the confusion matrix and classification report, we were able to gain insights into the performance of the model and identify areas where it could be further improved. Overall, the model was able to correctly classify images from a wide range of food classes, making it a useful tool for image classification in the food domain.

**Conclusion**

In conclusion, we successfully built and fine-tuned an image classification model using the EfficientNetB0 pre-trained Keras model to classify images from the Food101 dataset. The fine-tuned model achieved an accuracy of 80%, which is a significant improvement over the feature extraction model's accuracy of 70% and surpassed the accuracy of DeepFood paper, which was 76%.

The DeepFood paper proposed a food recognition system based on deep learning and achieved an accuracy of 76% on the same Food101 dataset. Our fine-tuned model's improved accuracy shows the effectiveness of fine-tuning pre-trained models in image classification tasks.

It is worth noting that the best accuracy achieved in food image classification has been improving over the years. For instance, in 2019, a research paper titled "Food Recognition Using Fusion of MobileNetV2 and InceptionV3 Networks" achieved an accuracy of 94.7% on the Food101 dataset. However, the accuracy achieved in these papers is subject to various factors such as the quality and size of the dataset, the model architecture, and the availability of computing resources.

In summary, our fine-tuned model's performance demonstrates the potential of using pre-trained models and fine-tuning them for image classification tasks, and we look forward to future advancements in the field of food image classification.

**Deployment**

After building and fine-tuning our deep learning model, we deployed it on various cloud platforms to make it available for use. We first deployed our model on AWS SageMaker, a fully managed service that provides developers and data scientists with the ability to build, train, and deploy machine learning models quickly. We used the SageMaker Python SDK to package and deploy our trained model as a Docker container, which could be deployed as an endpoint for inference.

Additionally, we deployed our model on AWS EC2, a virtual computing environment that allows users to launch and manage virtual servers. We used an Ubuntu server on EC2 and installed the necessary libraries and dependencies required to run our model. Once the setup was complete, we deployed our model as a RESTful API using Flask, a popular web framework in Python.

By deploying our model on both AWS SageMaker and EC2, we provided users with flexibility in choosing the deployment platform that best suits their needs. Additionally, by creating a Flask API, we made our model accessible via HTTP requests, allowing users to easily integrate it into their applications or workflows. The deployment of our model on AWS SageMaker and EC2, and the creation of a Flask API, demonstrates our ability to make our model available to a wide range of users and applications.