**Food recognition**

CS583 Deep Learning

Jiaqi Tu jtu3@stevens.edu

### • Abstract:

The Food-101 dataset is a large collection of images specifically designed for benchmarking machine learning algorithms in the task of food recognition. The project's primary objective is to develop a neural network model that can effectively classify images from the dataset.

This is my github link: <https://github.com/JiaqiTu/Food-recognition-project>

### • Introduction:

To refine the approach and manage computational resources more effectively, this project firstly concentrates on three distinct food categories: bibimbap, cheesecake, and donuts. For each of the three chosen categories, I extracted 750 images for training and 250 images for testing, resulting in a total of 3,000 images (2250 for training and 750 for testing).

I then expanded the scope to apply the model to the full Food-101 dataset. This comprehensive dataset includes 101,000 images across 101 food categories, with each category comprising 750 training images and 250 test images. This expansion necessitated scaling the computational resources and optimizing the data processing pipeline to handle the increased volume and diversity of data.

In transitioning to the full dataset, careful attention was paid to maintaining a balanced representation of each category to prevent class imbalance from skewing the model's learning process. Data augmentation strategies that were effective in the initial phase, such as random rotations and flips, were retained and applied to the broader dataset to enhance generalization further.

### • Data:

The Food-101 is a challenging data set of 101 food categories with 101,000 images. For each class, 250 manually reviewed test images are provided as well as 750 training images. On purpose, the training images were not cleaned, and thus still contain some amount of noise. This comes mostly in the form of intense colors and sometimes wrong labels. All images were rescaled to have a maximum side length of 512 pixels.

Link to the data: <https://pytorch.org/vision/main/generated/torchvision.datasets.Food101.html>

### • Method:

I first used a small VGG Model for the project validation. A custom variation of the VGG neural network architecture was employed, tailored to handle the classification task for the Food-101 dataset.

The VGG model is well-regarded in the field of deep learning for its simplicity and effectiveness in image classification tasks, that’s why I choose a small VGG model to validate.

The VGG model's deep convolutional layers are adept at capturing intricate details and features from food images, essential for accurate classification in such a diverse dataset. By adjusting and fine-tuning the VGG network, we ensured that the model could effectively differentiate between a wide array of food items, from simple fruits to complex dishes.

In addition to the VGG model, the ResNet architecture was also incorporated into our methodology. ResNet, renowned for its residual learning framework, significantly aids in training deeper networks. This is particularly beneficial for our project as it addresses the vanishing gradient problem, allowing us to construct deeper neural networks without compromising on training efficiency.

### • Tools & Technologies:

For this project, I utilized Google Colab and Pytorch for training and testing.

### • Experiments:

My model training was conducted using a custom VGG-like architecture, designed to classify images into three categories: bibimbap, cheesecake, and donuts. The training process was executed on Google Colab.

For my validation, The first convolutional block, Conv Block 1, comprises two convolutional layers. Each of these layers utilizes a 3x3 kernel with a stride of 1 and padding of 1, employing 10 filters to extract a variety of features from the input images. Following each convolutional layer, a Rectified Linear Unit (ReLU) activation function is applied. The use of ReLU introduces necessary non-linearity into the model, enabling it to learn more complex patterns from the data. This block concludes with a MaxPooling layer that has a 2x2 kernel and a stride of 2, effectively reducing the spatial dimensions of the resulting feature maps and thereby diminishing the computational load on subsequent layers.

The second convolutional block, Conv Block 2, mirrors the structure of the first. It consists of two convolutional layers with identical specifications to those in Conv Block 1, followed by ReLU activation functions and a MaxPooling layer. This design consistency ensures a thorough and intricate feature extraction from the images.

At the classifier stage of the network, a Flatten layer is employed to convert the 2D feature maps into a 1D vector.

VGG(

(conv\_block\_1): Sequential(

(0): Conv2d(3, 10, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(1): ReLU()

(2): Conv2d(10, 10, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(3): ReLU()

(4): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(conv\_block\_2): Sequential(

(0): Conv2d(10, 10, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(1): ReLU()

(2): Conv2d(10, 10, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(3): ReLU()

(4): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(classifier): Sequential(

(0): Flatten(start\_dim=1, end\_dim=-1)

(1): Linear(in\_features=2560, out\_features=3, bias=True)

)

)

The training involved several steps:

1. Loading and transforming the dataset with resizing to 64x64 pixels and random horizontal flipping.
2. Implementing a DataLoader with a batch size of 32 and using the maximum available CPU cores for parallel processing.
3. Training was performed in batches, with loss and accuracy metrics calculated for each epoch.

VGG-16 Experiments:

The VGG-16 model, known for its deep architecture with successive convolutional layers, was employed first. The model's layers were fine-tuned to adjust to the specifics of the Food-101 dataset. Data augmentation techniques such as random rotations and flips were applied to reduce overfitting. The training process was monitored over 100 epochs, and the model's performance was evaluated based on the training and validation loss, as well as accuracy. Despite VGG-16's tendency for high computational load, it demonstrated a notable decrease in training loss and a significant, albeit fluctuating, increase in validation accuracy.

ResNet-18 Experiments:

Following the VGG-16 experiments, the ResNet-18 architecture was utilized. ResNet-18 introduces residual learning to alleviate the vanishing gradient problem, allowing for the training of deeper networks. The same dataset and data augmentation strategies were applied to ensure consistency in comparison. The ResNet-18 model was observed for the same number of epochs as the VGG-16. The results indicated a smoother convergence in both training and validation metrics, suggesting more stable learning and better generalization capabilities than the VGG-16 model.

### Evaluation Method:

The model was evaluated on a separate test dataset to assess its generalization capabilities.

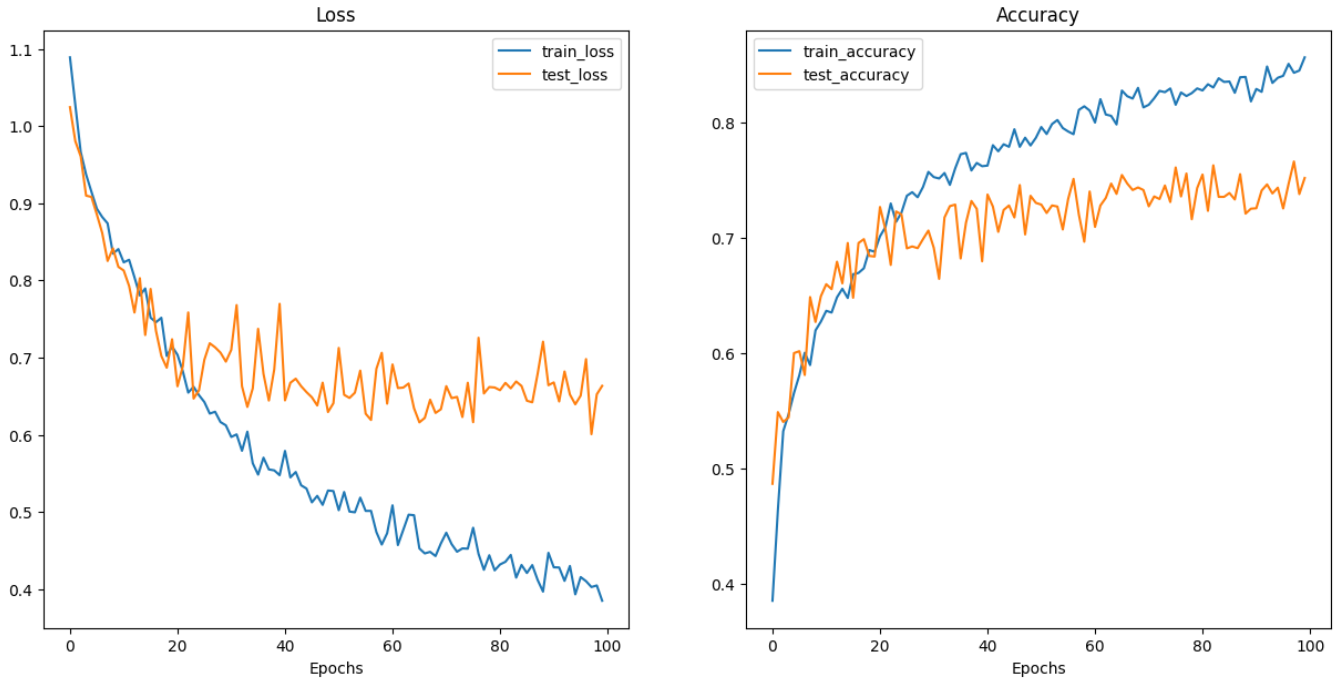
**Accuracy Metrics**: Model performance was measured in terms of accuracy, calculated as the proportion of correctly classified images in the test set.

**Loss Evaluation**: The loss on the test set was also recorded, providing insight into the model's predictive confidence.

### • Results:

The primary results of our experiments with the VGG model on the Food-101 subset (focusing on bibimbap, cheesecake, and donuts) are summarized below. I also plan to explore another model to further enhance the accuracy.

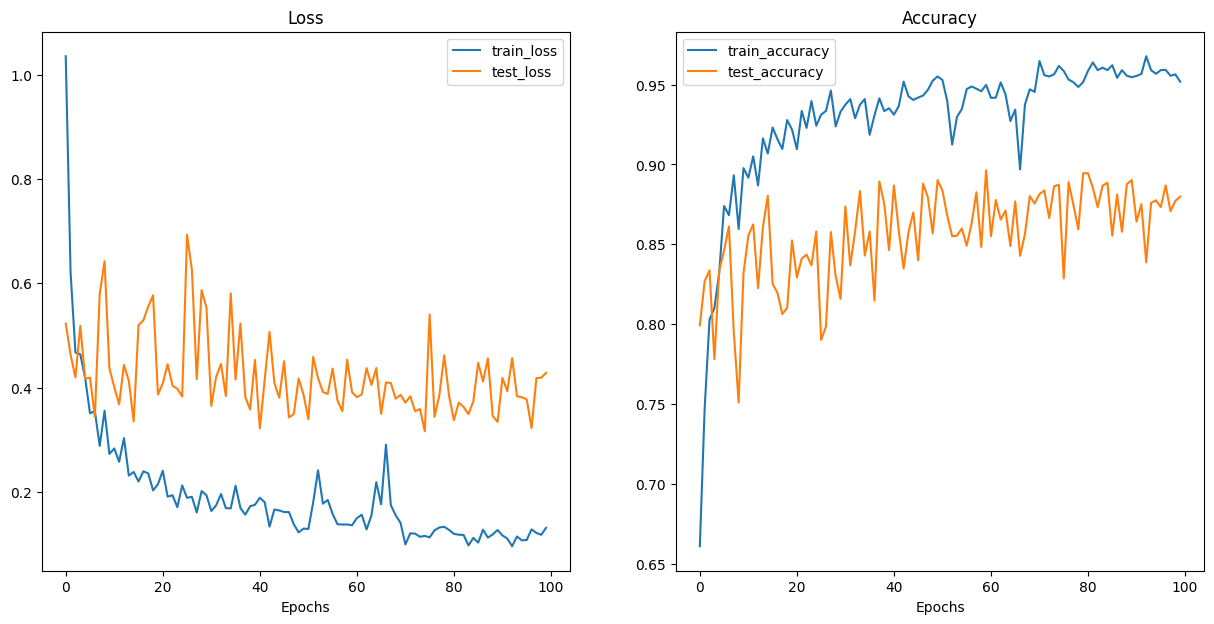
This result is for a custom VGG-like architecture:



There is a clear trend of improvement in both training and testing accuracy over the epochs, indicating that the model is learning effectively from the data.

In my approach to classifying images from the Food-101 dataset, I employed data augmentation techniques to enhance the robustness and generalization capability of our custom VGG-like architecture. Data augmentation included random rotations, width and height shifts, shearing, zooming, and horizontal flipping, which helped in simulating a variety of perspectives and scales, thereby enriching our model's exposure to the dataset's diversity.

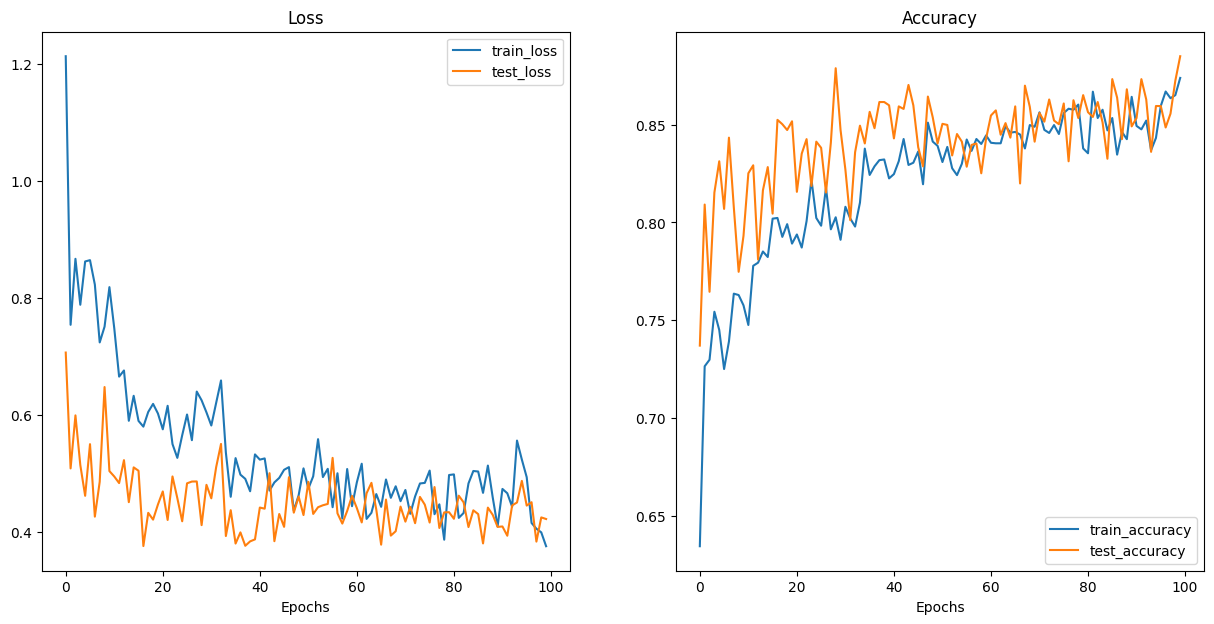
The ResNet-18 model results showed below:



My experimentation with the ResNet-18 architecture, pre-trained on ImageNet and fine-tuned on the Food-101 dataset, yielded insightful results. I adapted the ResNet-18 model by modifying its final fully connected layer to output 101 classes corresponding to the Food-101 dataset.

The ResNet-18 model showed promising results, with high training accuracy and low training loss, evidencing strong feature learning. However, the gap between training and test metrics and the fluctuating test accuracy highlight the challenges in model generalization. The overfitting observed may be addressed by techniques such as data augmentation, additional regularization, or hyperparameter tuning to improve model performance on unseen data.

The VGG-16 network result is shown below:



The VGG-16 network's performance on the Food-101 dataset illustrates its powerful feature extraction capabilities.

The application of the VGG-16 network to the Food-101 dataset has yielded promising results, with the network demonstrating strong learning capabilities. The results affirm the potential of VGG-16 in handling complex image classification tasks and provide a foundation for further refinement to bolster its performance on diverse and unseen datasets.

• Problems/Issues:

One of the main challenges I faced during this project was the extraction of three specific classes - bibimbap, cheesecake, and donuts - from the extensive Food-101 dataset.

Secondly is Overfitting: Despite the use of a robust architecture like VGG-16, the model may have overfitted to the training data. This is often indicated by the training loss continuing to decrease while the validation loss starts to plateau or even increase, which can also be accompanied by large fluctuations in validation accuracy. I used data augmentation to decrease it.

• Conclusion:

The fine-tuned ResNet-18 model achieved notable success on the Food-101 dataset. The consistent learning pattern in the loss graphs and the high training accuracy are testament to the model's capacity to learn complex features. Moving forward, further improvements can be made to enhance the model's generalization to new data, ensuring that the accuracy remains high when faced with diverse and unseen food images.

The VGG model appeared to reach a plateau in accuracy faster than the ResNet model. This could be due to the simpler, but deeper, architecture of VGG.