—

# **Final Project Report: Development of a DCNN for Food Image Recognition**

—

**CMPS261**

**Date: April 25, 2024**

**Karim Daouk**

**Elie Estephan**

**Ahmad Kanounji**

**Chadi Yachoui**

## **1.** **Abstract**

This project develops a deep convolutional neural network (DCNN) for the recognition of food images, drawing inspiration from established research in the field, particularly focusing on enhancing accuracy and efficiency in food recognition systems as outlined in the referenced study [1]. The model, "K-foodNet," utilizes advanced image processing techniques and neural network architectures, implemented using Python and libraries such as TensorFlow and Keras, we have implemented a model using a similar approach tailored to the small dataset available to us and tweaked it until we reached a satisfactory result.

## **2. Introduction**

Advancements in image recognition have significant applications in culinary technology, such as nutritional tracking and automated dietary management. This project builds upon the methodologies and findings described in a key study [1], adapting its concepts to develop a specialized DCNN for recognizing various food items with high precision. Due to the small dataset, however, we have simplified the model both for efficiency and to avoid overfitting as much as possible.

The primary goal of this project is to implement a DCNN capable of accurately identifying food items from images. Objectives were set to:

- Develop a scalable model trained on a substantial dataset of labeled images.

- Achieve high accuracy in multi-class food item classification.

## **3. Dataset**

**Data Collection and Preprocessing**

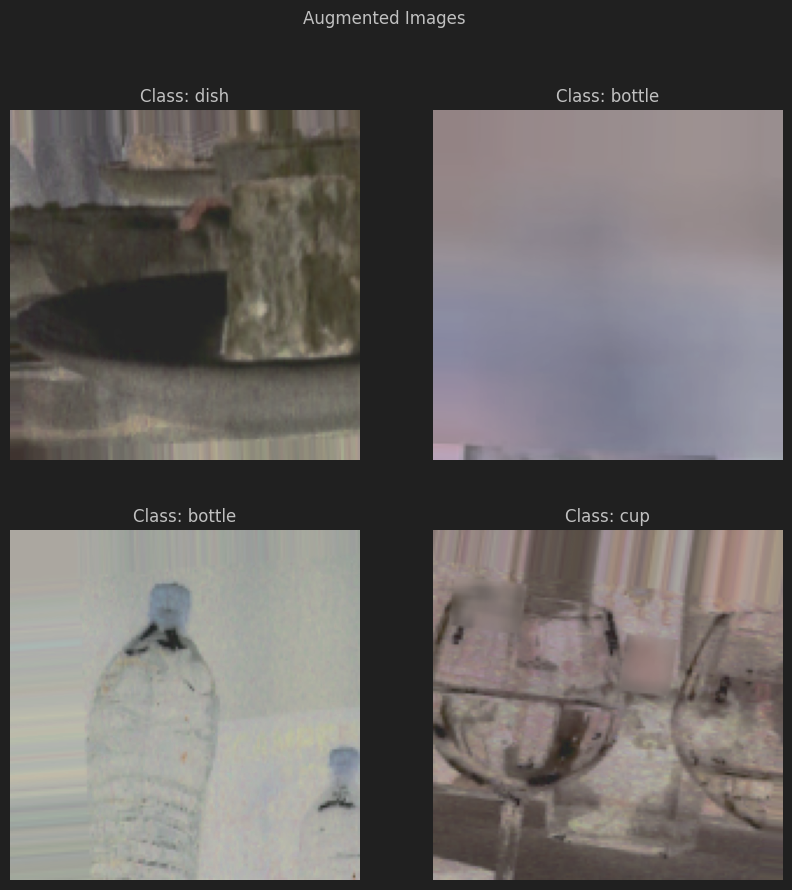
The dataset we had was around 5000~ images of a variety of food items plus utensils, like cups, mugs, jars… etc.

Following the preprocessing steps outlined in the referenced study [1], images were resized and normalized to prepare them for effective training, and then augmented for better quantity and training:

- **Image Resizing**: Consistent with study recommendations, images were resized to 150x150 pixels.

- **Normalization**: Pixel values were scaled to a range between 0 and 1, which aligns with the best practices for neural network training detailed in [1].

- **Data Augmentation**: We have used a data generator to augment the data so that our dataset increases by a couple of multitudes. Here are some of the photos after augmentation:



## **4 Models**

The inital model incorporated structural insights and architectural principles from the referenced paper [1]:

- **Inception Modules**: Utilizing mixed convolutional filters within inception modules to capture a wider range of image features, as suggested by the study.

- **Network Architecture**: Includes layers for depth and complexity, critical for capturing detailed image features essential for accurate classification.

- **Activation Functions**: ReLU is employed across the model for its effectiveness in non-linear feature learning.

- **Output Layer**: A softmax output layer categorizes the probabilities across various food types.

However, our final model became simpler:

-**Inception-like modules**: we implemented a similar layer to inception layers instead, with fewer filters. The rest of the model has Stayed the same

## **4.1 Model Training**

Model training protocols were adapted from [1], ensuring rigorous testing and validation:

- **Data Generators**: Used for augmenting the training process, enhancing the model's ability to generalize from the training data.

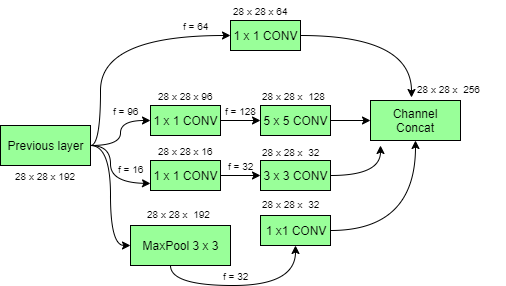
- **Optimizer and Loss Function**: Adam optimizer and sparse categorical cross-entropy, as recommended by the research for efficient backpropagation and convergence.

## **5. Results**

The model at first was not accurate, only achieving a validation accuracy of around 40%. This prompted us to use data augmentation such as the study[1] did.

However, the accuracy and loss even after augmentation were still not satisfactory  

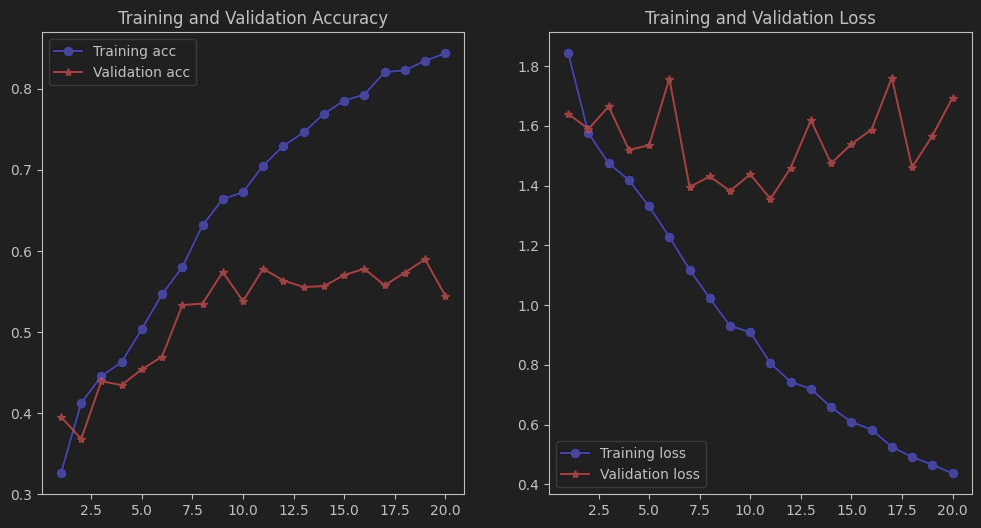

So, after a deeper research into the inceptionV1 model that the research paper based the model off of, it appears it uses a deeper inception layer [4]:



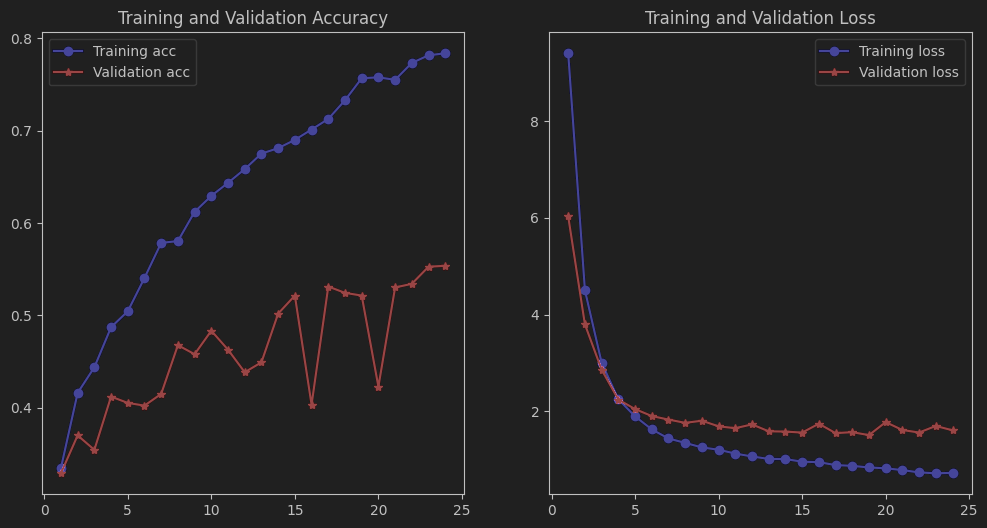
Implementing these changes was easy after all the experience gained from creating and tweaking the model.

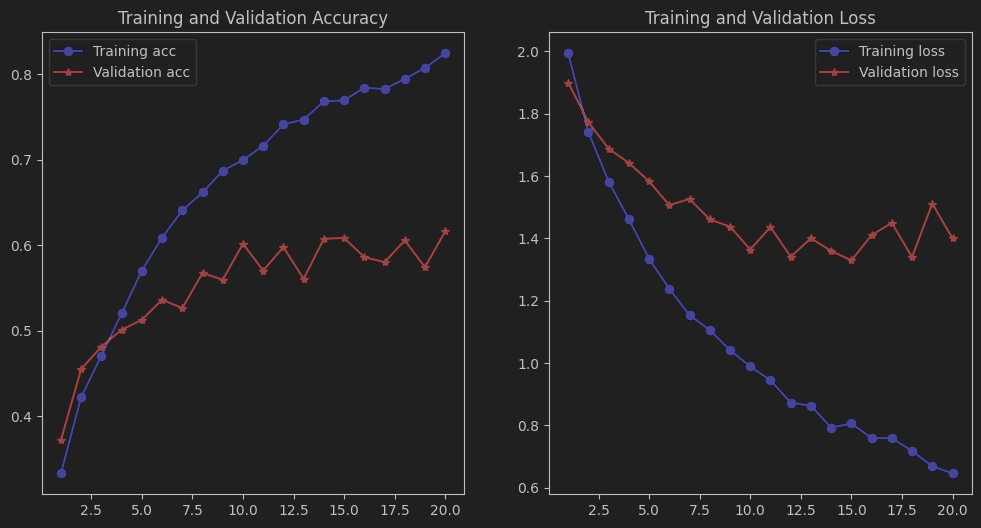
Also, we reduced the augmentation parameters as we thought they might be too aggressive and introduce too much noise and unrealistic photos that would be too difficult for the model to train on.

Finally, we were using 10 epochs to train the model, the paper, however, was using 20. We decided to go with 10 to avoid overfitting, but we then decided to match the number of epochs and see the results:



Overfitting is a persistent issue our model is facing. Another attempt to reduce this overfitting is L2 regularization[5], and batch normalization with pre-activation[6], as well as removing one inception layer, decreasing the regularization factor, adding dropout layers, increasing the dropout rate, and decreasing neuron density. All these tweakings helped with loss, but unfortunately not with accuracy.



One last effort in improving the model was to simplify it. After discussion and consideration, the number of images in the dataset was too low for the model to capture the correct features and predictions, so we went ahead and reduced the number of layers and filters, keeping a simple version of the inception layer:  


The model shows an improvement in accuracy and efficiency in recognizing food images, reflecting the enhancements and optimizations derived from the study [1].

## **6. Conclusion**

The implementation faced challenges similar to those discussed in [1], particularly around the optimization of layer dimensions and filter sizes[3]. Adjustments were made accordingly to align the model's performance with theoretical expectations. The concatenation of layers was also a new sight, hence having to research it using official documentation[2]

Reflecting on the objectives and the guidance from the literature, "K-foodNet" demonstrates the successful application of deep learning techniques in food image recognition. Thus, following in its steps has provided the proper technologies and approach to successfully detect food items.

## **7. References**

1. [**The development of food image detection and recognition model of Korean food for mobile dietary management**](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6883229/pdf/nrp-13-521.pdf)
2. [**Concatenate layer**](https://keras.io/api/layers/merging_layers/concatenate/)
3. [**Conv2D layer**](https://keras.io/2.15/api/layers/convolution_layers/convolution2d)
4. [**ML | Inception Network V1 - GeeksforGeeks**](https://www.geeksforgeeks.org/ml-inception-network-v1/)
5. [**How to Treat Overfitting in Convolutional Neural Networks**](https://www.analyticsvidhya.com/blog/2020/09/overfitting-in-cnn-show-to-treat-overfitting-in-convolutional-neural-networks/)
6. [**Introduction to Batch Normalization**](https://www.analyticsvidhya.com/blog/2021/03/introduction-to-batch-normalization/)

—