BBM 418 - Computer Vision Laboratory

HACETTEPE UNIVERSITY

Department of Computer Engineering

Programming Assignment-2
Spring 2024-2025
Dr. Nazli Ikizler Cinbis

Due Date: 23:59 on Thursday, May 3rd, 2025

Image Classification with Convolutional Neural Networks

In this assignment, you will get familiar with image classification by training Convolutional Neural Networks (CNN).The goals of this assignment are as follows;

1. Understand CNN architectures and build a model from scratch and train on data.
2. Understand and implement transfer learning from a pre-trained CNN.
3. Analyze your results.
4. Experience with a deep learning frame, PyTorch.

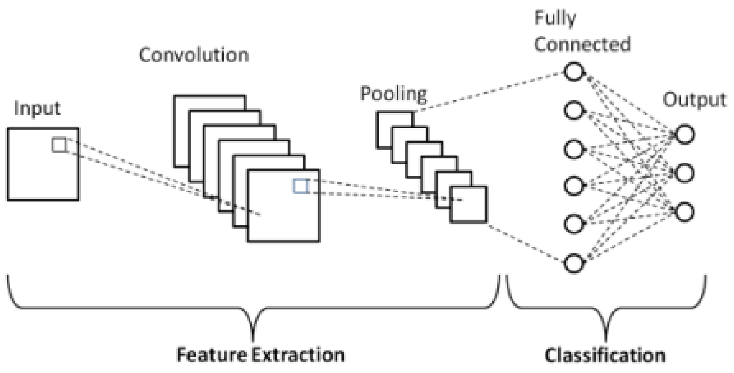

Figure 1: A simple CNN architecture [1].

Dataset

The Food-11 dataset contains about 2750 medium-quality food images belonging to 11 categories: apple pie, cheesecake, chicken curry, french fries, fried rice, hamburger, hot dog, ice cream, omelette, pizza, and sushi. The main directory is divided into train, validation, and test directories.

Download Link: Food-11

Figure 2: Examples for dataset [2].

PART 1 - Modeling and Training a CNN classifier from Scratch [60 pts]

In this part, you are expected to construct two CNN classification models and train them from scratch using the dataset provided. You should first define the components of model design. Your model should have 5 convolution layers and 2 fully connected layers. Also, you need to add a max pooling and a batch normalization layer on your model. One of your models have residual connection(s) and the other does not. Your model with residual connection(s) must have one or more residual connections.

- Give parametric details with relevant PyTorch code snippets; number of in channels, out channels, stride, etc. Specify your architecture in detail. Write your choice of activation functions, loss functions and optimization algorithms with relevant code snippets. [2.5 pts x 2 = 5 pts]
- Explain how you have implemented Convolution, Fully-Connected Layers, and residual connection(s) and give relevant code snippets. [5 pts]
- You may apply augmentation on images to obtain better accuracy.

For both of your models; you will set epoch size to 50 and evaluate your two model with three different learning rates and two different batch sizes. Explain how you calculate accuracy with relevant code snippet. Moreover for each of the points below add relevant code snippet to your document.

1. Draw a graph of loss and accuracy change for three different learning rates and two batch sizes. [5 pts x 2 = 10 pts]
2. Select your best model with respect to validation accuracy and give test accuracy result. [2.5 pts x 2 = 5 pts]
3. Integrate dropout to your best models (best model should be selected with respect to best validation accuracy. You need to integrate to both of your best models). In which part of the network you add dropout and why? Explore two different dropout values and give new validation and test accuracies. [5 pts x 2 = 10 pts]
4. Plot a confusion matrix for your best model's predictions. [2.5 pts x 2 = 5 pts]
5. Explain and analyze your findings and results. [10 pts x 2 = 20 pts]

PART 2 - Transfer Learning with CNNs [40 pts]

You will fine-tune the pre-trained network which is available at PyTorch. For this part, you can choose one of the pre-trained networks from MobileNetV2 [3], MobileNetV3 [4], or ShuffleNetV2 [5]. These networks are trained on ImageNet dataset so you are not initializing the weights randomly, instead, you are using the pre-trained weights from these networks. Freeze all the layers before training the network, except the FC layer. Consequently, the gradients will not be calculated for the layers except the ones that you are going to update (FC layer) over the pre-trained weights. However, since the number of classes is different for our dataset, you should modify the last layer of the network, which the probabilities will be calculated on. Therefore, the weights will be randomly initialized for this layer.

1. What is fine-tuning? Why should we do this? Why do we freeze the rest and train only FC layers? Give your explanation in detail with relevant code snippet. [5 pts]
2. Explore training with two different cases; train only FC layer and freeze rest, train last two convolutional layers and FC layer and freeze rest. Tune your parameters accordingly and give accuracy on validation set and test set. Compare and analyze your results. Give relevent code snippet. [20 pts]
3. Plot confusion matrix for your best model and analyze results. [5 pts]
4. Compare and analyze your results in Part 1 and Part-2. Please state your observations clearly and precisely. [10 pts]

What to Hand In

Your submission format will be:

- b<studentNumber>.ipynb (iPython notebook)
- b<studentNumber>.py (Convert your ipynb file to py file)
- report.pdf

Archieve this folder as **b<studentNumber>.zip** and submit to `https://submit.cs.hacettepe.edu.tr`.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out abstractly. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697

[2] https://drive.google.com/file/d/1a0uuiylWnyGAr0JZVfhw2EPZ6XzcPtyb/view?usp=sharing

[3] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[4] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1314-1324).

[5] Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European conference on computer vision (ECCV) (pp. 116-131).