National University of Sciences and Technology
School of Electrical Engineering and Computer Science
Department of Computing

CS-405 Deep Learning

Fall 2023

# Lab 05

## Convolutional Neural Networks for Multiclass Classification

Date: 12th Oct 2023

Instructor: Bostan Khan

# Deep Learning

# Lab 05

The objective of this assignment is to give you hands-on experience in implementing and training convolutional neural networks. You will be demonstrating the versatility of such networks by implementing a single model that can perform classification tasks on a large dataset. You need to design the network, train it for a classification task, and then finally assess its performance.

**Provided Material:** This document as well as a starter **Jupyter Notebook** has been provided on LMS. You need to work on the Jupyter Notebook on Colab and prepare a report on your working as mentioned in part 4 of this assignment.

**Deliverables:** You need to submit a zipped file on LMS containing the finalized Jupyter Notebook along with a **PDF** for the prepared report.

## Dataset:

The given dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender, and ethnicity. **Unlike the last lab 04 when we had only used 7000 samples, this time we will be using all of the 20,000 images**. The dataset in this assignment is going to be used for the following task:

**Given the pixels of an image, perform ethnicity classification for the image using a convolutional neural network.**

**Number of Input features in an image:** 2304 (48x48 pixel values)

**Number of Labels for each input image:** 3

**Labels:**

1. Age (Regression)
2. Ethnicity (Multiclass classification)
3. Gender (Binary Classification)

# Instructions:

The dataset for this assignment is already cleaned and provided in a CSV format for you with images included as pixel values.

The 4 columns contain the following information:

1. **[age]** is an integer from 0 to 116, indicating the age. You can convert it to float for regression predictions
2. **[gender]** is either 0 (male) or 1 (female)
3. **[race]** is an integer from 0 to 4, denoting White, Black, Asian, Indian, and Others (like Hispanic, Latino, Middle Eastern).
4. **[pixels]** The 4th column contains the pixels of the images flattened to a single dimension. Each of the original image is of size 48x48 and the flattened image vector consists of 2304 pixel values.

**You would be only using the third label [race] for this lab.** An initial processing has already been done using pandas in the starter code Jupyter notebook. Hence you have a data frame ready for your assignment. Remember to use all the samples in the dataset.

## Part 1: Dataset Creation and Data Splitting (10 points):

Split the dataset into training (70%), validation (15%), and testing (15%) subsets. Try to standardize your input features for faster training times and possible better performance.

Cleary show how you manage datasets and dataloaders with clear comments.

Note: The code should be modular. Documentation, and comments in the code have their own marks.

## Part 2: Building a Unified Neural Network (60 points)

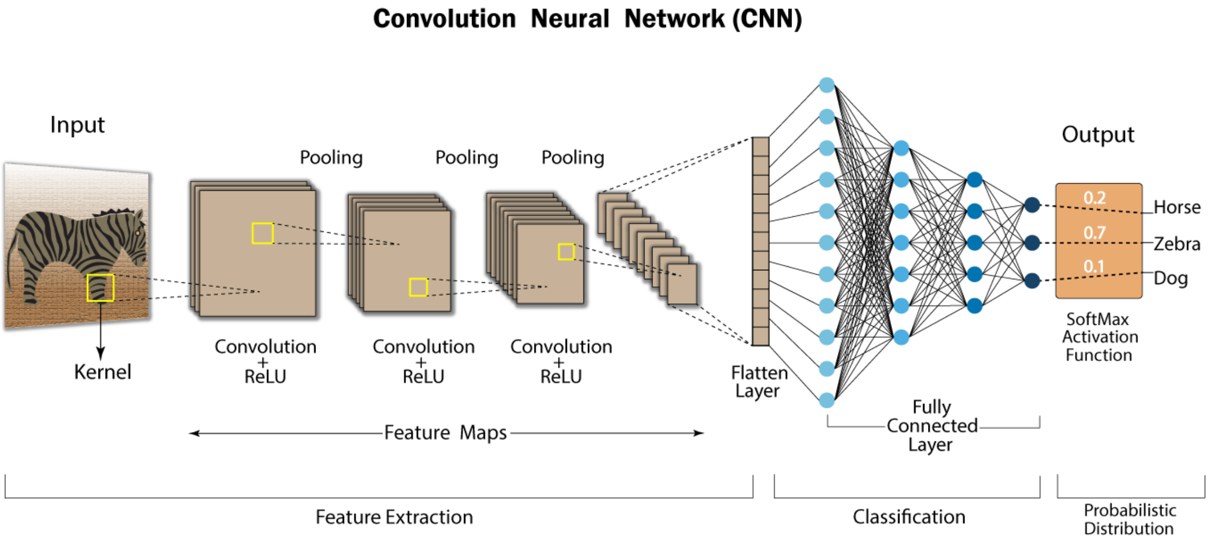
### (i): Convolutional Neural Network Architecture (30 points)

Design a convolutional neural network that can handle ethnicity classification. In addition to convolutional and ReLU activation layers, you should also try Dropout and Batch Normalization layers in your network and see how they affect your performance. Try to change the size of your model by increasing and decreasing the number of layers that you are using. An example CNN is provided to you in Figure 1 which may or may not provide you the desired performance for the current task at hand so you will have to try out different models and hyperparameters.

*Figure 1. An Example CNN Source: https://www.analyticsvidhya.com/blog/2022/03/basics-of-cnn-in-deep-learning/*

## (ii): Model Training (20 Points)

Train the network on the training subset of the data created in Part 1 while validating its performance on the validation set after each epoch.



## (iii): Performance Monitoring (10 Points)

Keep track of the training and validation losses and accuracy during the training for each task. Visualize the training and validation curves (e.g., loss vs. epochs, accuracy vs epochs) for the three tasks.

Note: The code should be modular. Documentation, and comments in the code have their own marks.

## Part 3: Evaluation (10 points)

### (i) Classification Evaluation

Evaluate the model's classification performance on the testing data. Calculate **accuracy** and **losses** for the classification task on the test sets. Finally, visualize the confusion matrix.

Note: The code should be modular. Documentation, and comments in the code have their own marks. Model performance in this case will be noted. Drastically low performances values on the test set would result in deduction of marks.

## Part 4: Conclusion and Discussion (20 points)

### (i) Report

Write a report summarizing the assignment, including data generation, network architecture, training process, and evaluation results. Other things to include are as follows:

- Discuss the advantages of convolutional neural networks that you might notice during this exercise over fully connected networks for such image processing tasks.
- Which settings for your model creation and the training process performed the best for you? Like how many convolutional layers in the network gave you good performance on the test set? What learning rate did you choose for your training. In short, what type of settings worked better for you and what is the reason for it?

## Submission:

Students should submit their code (**Jupyter Notebooks**) and the report as a **PDF** document in a single zipped file on LMS. Once you have submitted your zipped file, be sure to check whether the file has been uploaded correctly or not by redownloading your submission and opening the jupyter notebook as well as the PDF report.

## Grading:

The grading rubrics are mentioned in each section, however, you should note that the rubrics in general involve code correctness, modularity, documentation and clear comments.

Hopefully, this lab helps you learn and create convolutional neural networks for different classification tasks, encouraging you to think critically about the advantages of this approach over fully connected networks.

**End**