

CS4442/9542 Assignment 3 :

CNN for Image Classification

Due: April. 9st, 2019 at 9:00 pm

Weight: 20%

Goal:

Follow the [hand-on ML chapter 13](#)'s example to learn how to build a CNN to solve a new classification problems. Note you can use other similar or equivalent platforms or tools to accomplish the tasks in this assignment.

- Learn to prepare and process your image dataset
- Understand a CNN's architecture with multiple layers
- Hand-on practice on building a CNN model and training the image dataset

Dataset:

Select one image dataset from the options below. All the dataset options are from the [kaggle](#) where you can find more details on certain dataset.

(Note: only the training set with label information is needed for this assignment, you do not need to report the performance on the test set from kaggle for this assignment.)

1. [Fashion MINIST](#) : it consists of a training set of 60,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.



2. **Face Images for Sentiment Classification**: this is a private kaggle dataset, I uploaded to [google drive](#) where you can download. This dataset consists of about 28000 images. Each example is 48x48 grayscale image, associated with a label from 6 classes.



0-angry



1-disgust



2-fear



3-happy



4-sad

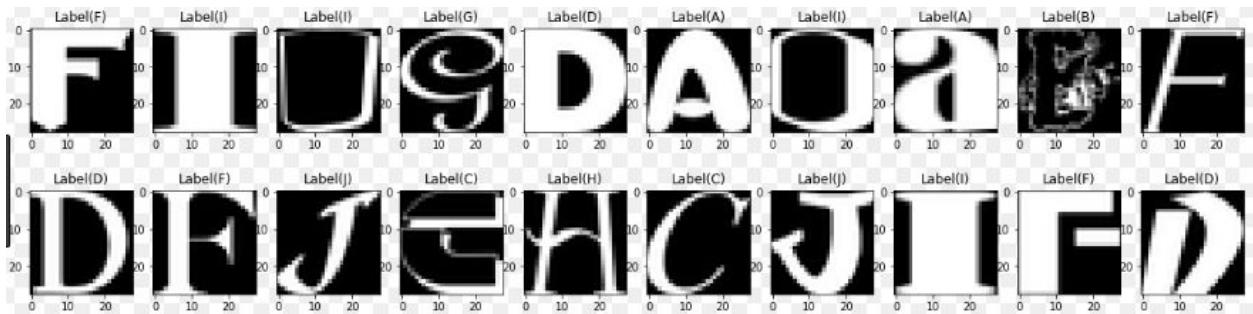


5-surprise



6-neutral

3. [notMINST_small](#) : 18726 images, 28 x 28 grayscale images of letters A - J, sorted into directories by letter.



Specifications:

1. Download/load the dataset from the raw csv or folders, convert the labels to one-hot label if necessary
2. Write a preprocessing step that will **resize** and crop the image to $N \times N$, with some randomness for data **augmentation**. In your code work, show some typical sample images from the raw dataset and the processed dataset.
3. Split your dataset into a training set and a test set.
4. Create a CNN model with **at least three** hidden layers(*conv+pooling will be considered as one layer*), report your detailed design.
i.e: the structure of your model, each layer's kernel size, the total parameters your model will learn etc.
5. Train the model on the training set, evaluate it on the test set and report your model's performance. Describe your training progress, like hyperparameters setting, what the loss function is selected.

What You Will Submit:

1. Your source code file with necessary organized comments for each step. The jupyter notebook file (.ipynb) is recommended, then you can attach a single file with the name format UWOuserid_Assign2.ipynb;
if you choose other methods, like matlab, to implement this project, please attach all your code files in to one zip file with the same name format.
(The TAs are not responsible to debug your submitted code, If your code file and comments are not adequate to specify all your work, please attach a **written report** to describe your work on each specification of this assignment, as well as the necessary outputs for each step)

Appendix:

A good Keras example to build CNN: [minist_cnn.py](#)

Hands on ML chapter 13: [CNN](#) ipynb example(implemented by Tensorflow)

[Image Augmentation](#) for DL with Keras