Deep Learning HW1

曾以諾
Institute of Data Science
National Cheng Kung University
Tainan, R.O.C.
re6111032@gs.ncku.edu.tw

***Abstract*—This report is an description document for HW1. And the GitHub link is as follows**

GitHub Link:
https://github.com/butterfly2012010/DeepLearning_HW1

I. Introduction

In homework 1, the general description is to build an "Image Classification Pipeline"

- Reading images to an array
- Feature extraction (transform the image into a fixed-length feature vector)
- Apply any classifier to verify the performance.

II. Background

A. Feature Extraction

I mainly use two methods of feature extraction, which are "color histogram", and "discrete wavelet transform".

1) Color Histogram [1]: In image processing and photography, a color histogram is a representation of the distribution of colors in an image. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges, that span the image's color space, the set of all possible colors.
   The color histogram can be built for any kind of color space, although the term is more often used for three-dimensional spaces like RGB or HSV. For monochromatic images, the term intensity histogram may be used instead. For multi-spectral images, where each pixel is represented by an arbitrary number of measurements (for example, beyond the three measurements in RGB), the color histogram is N-dimensional, with N being the number of measurements taken. Each measurement has its own wavelength range of the light spectrum, some of which may be outside the visible spectrum.
   If the set of possible color values is sufficiently small, each of those colors may be placed on a range by itself; then the histogram is merely the count of pixels that have each possible color. Most often, the space is divided into an appropriate number of ranges, often arranged as a regular grid, each containing many similar color values. The color histogram may also be represented and displayed as a smooth function defined over the color space that approximates the pixel counts.
   Like other kinds of histograms, the color histogram is a statistic that can be viewed as an approximation of an underlying continuous distribution of color values.
2) Discrete Wavelet Transform [2]: In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency and location information (location in time).

III. Models and Methods

I use 3 different models to train and evaluate their performance.

1) Perceptron: I use the scikit-learn package to realize the Perceptron. First, I use the color histogram to extract the feature of images, and then flatten the features into the shape, $(N_{samples}, N_{features})$. Second, use these features to train the Perceptron model.

   The parameter I used in Perceptron model are the followings,
   - n_job=8
   - random_state=42
2) XGBoost: I use the xgb package to realize the XGBoost classifier. The method of feature extraction is the same as the previous mentioned in "Perceptron". After feature extraction, I use the xgboost on training.

   The parameter I used in XGBoost model are the followings,
   - 'objective': 'multi:softmax'
   - 'num_class': 50
   - 'eta': 0.1
   - 'max_depth': 3
   - 'min_child_weight': 1,
   - 'gamma': 0,

- 'subsample': 0.8,
- 'colsample_bytree': 0.8,
- 'n_estimators': 100

3) CNN: I use Pytorch to build a CNN model. First, use opencv to read the images into BGR color mode. Second, use these images on CNN.

   The followings are the network architecture, the input size is (batch_size, 3, 64, 64)

   a) convolution layer 1
      - in_channels=3
      - out_channels=16
      - kernel_size=3
      - stride=1
      - padding=1
   b) relu layer 1
   c) maxpooling layer 1
      - kernel_size=2
      - stride=2
   d) batch normalization layer 1
   e) convolution layer 2
      - in_channels=16
      - out_channels=32
      - kernel_size=3
      - stride=1
      - padding=1
   f) relu layer 2
   g) maxpooling layer 2
      - kernel_size=2
      - stride=2
   h) batch normalization layer 2
   i) convolution layer 3
      - in_channels=32
      - out_channels=64
      - kernel_size=3
      - stride=1
      - padding=1
   j) relu layer 3
   k) maxpooling layer 3
      - kernel_size=2
      - stride=2
   l) batch normalization layer 3
   m) fully connected layer 1
   n) relu 4
   o) fully connected layer 2

IV. Result

Please check these files,

- "CH_Perceptron.ipynb"
- "CH_XGBoost.ipynb"
- "CNN.ipynb"
- "DWT_CNN.ipynb"

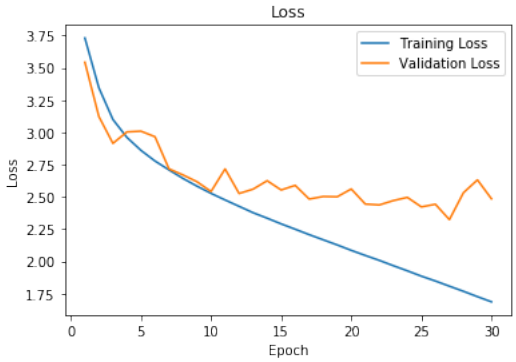
"CH" means color histogram, and "DWT" means discrete wavelet transform. Actually, I had done other methods, such as using DWT on feature extraction, and choose Perceptron and XGBoost as models. These attempts can be found in the file, "HW1_RE6111032_script.ipynb".

In the homework 1, the score metric are top-1 accuracy and top-5 accuracy. Here are the results of models.

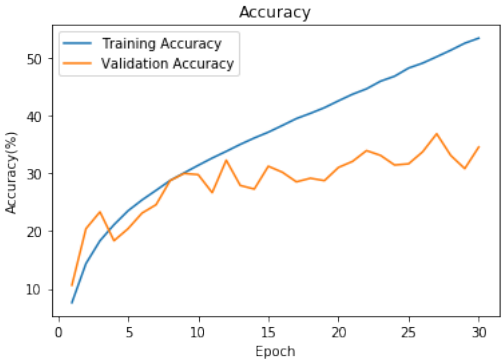
TABLE I
Performance

Methods	train		validation	
	top-1 acc	top-5 acc	top-1 acc	top-5 acc
CH_Perceptron	0.0377	0.1583	0.0422	0.1289
CH_XGBoost	0.2326	0.5278	0.1133	0.3289
CNN	0.5345	0.8442	0.3458	0.6729
DWT_CNN	0.4938	0.8001	0.2562	0.6000

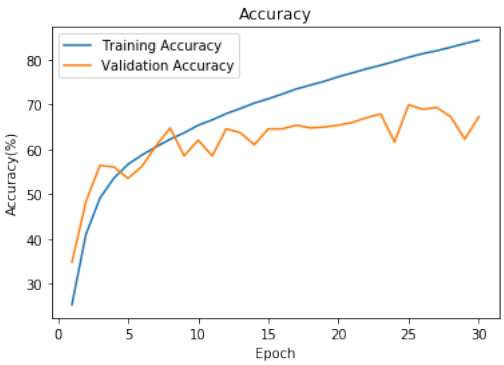
For CNN, the graph of epochs v.s. loss is



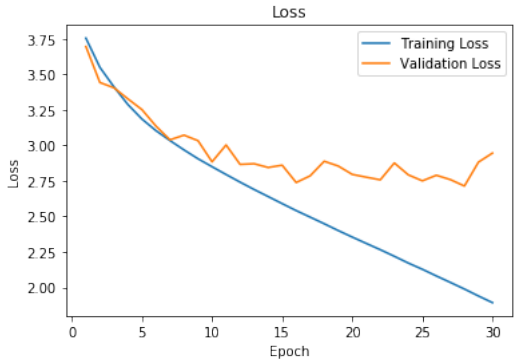
The graph of top-1 accuracy is



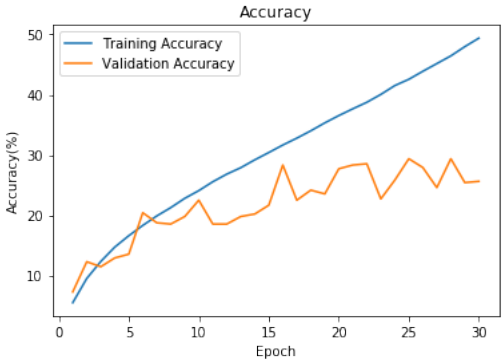
The graph of top-5 accuracy is



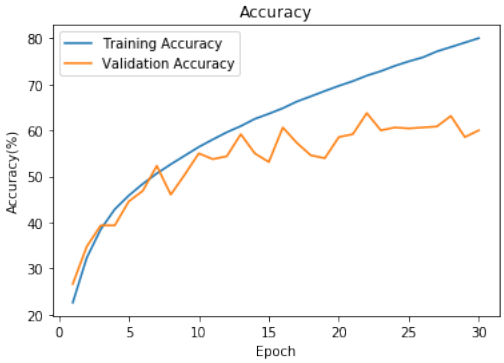
For DWT_CNN, the graph of epoch v.s. loss is



The graph of top-1 accuracy is



The graph of top-5 accuracy is



V. Performance Comparison

From the TABLE I, we may find that the performance of CNN model is the best in train and validation no matter what metric we choose.

VI. References

References

[1] color histogram,https://en.wikipedia.org/wiki/Color_histogram

[2] wavelet transform,https://en.wikipedia.org/wiki/Wavelet_transform

[3] ChatGPT