

**University of Wollongong**  
**School of Computing and Information Technology**

**CSCI446/946**

**Big Data Analytics**

**Spring 2019**

**Assignment 3**

**(Due: 30 October 2018, Wednesday)**

**20 marks**

### **Aim**

This assignment is intended to provide basic experience in conducting image analytics experiments with R (or any other languages preferred by students). After having completed this assignment you should know how to perform image classification by training a deep convolutional neural network or using a pre-trained deep CNN network.

### **Preliminaries**

Read through the lecture notes and recommended readings on image analysis. Complete the tutorial and study all example programs therein so that you fully understand these techniques and know how to perform them with R (or any other languages you prefer). Important references include:

1. <https://mxnet.incubator.apache.org/api/r>
2. <https://github.com/apache/incubator-mxnet/tree/master/R-package>
3. <https://www.r-bloggers.com/image-recognition-tutorial-in-r-using-deep-convolutional-neural-networks-mxnet-package/>

### **Task 1 – Understanding CNNs for image analysis (8 marks)**

Deep convolutional neural networks (CNNs) have recently achieved the state-of-the-art performance on image recognition. It is important to understand the basic working principles of deep CNNs in order to appropriately utilize them to resolve image analysis tasks. Study the lecture notes, recommended readings, and other learning resources in the Internet to **answer the following questions**.

1. Describe deep convolutional neural networks and discuss why they suit image analysis tasks; **(2 marks)**
2. Describe the functions of “convolution layer” and “pooling layer” in deep CNNs; **(2 marks)**
3. Explain the following concepts in training deep CNNs: a) activation function; b) epoch number; c) batch size; d) learning rate; and e) momentum. **(4 marks)**

### **Task 2 – Handwritten Digits Classification with CNNs (6 marks)**

MNIST is a benchmark dataset for handwritten digits classification. Each sample in this dataset is a small image of the size 28 by 28. Each image belongs to one of the 10 categories corresponding to digits “0” to “9”. Handwritten Digits Classification is to design image recognition algorithms that can best classify each image into the right category. This dataset has been pre-partitioned into training and test sets. Information on this dataset can be obtained from the webpage [https://mxnet.incubator.apache.org/api/r/docs/tutorials/mnist\\_competition](https://mxnet.incubator.apache.org/api/r/docs/tutorials/mnist_competition). Carefully read this webpage and obtain the training and test data sets. **Complete the following tasks:**

1. Reshape each image into a long vector of 784 (i.e., 28 x 28) dimensions. Train a logistic regression classifier (LRC) or a multi-class linear support vector machine (SVM) with the training data set and test it on the testing data set;
2. Train the convolutional neural network “LeNet” with the training data set and test it on the testing data set;
3. Evaluate and compare the classification performance obtained in the two ways above.

In your report, you need to

1. Describe this MNIST data set and its training and test subsets.
2. Describe how you reshape each image into a long vector and how you train the LRC or SVM.
3. Describe the convolutional neural network “LeNet” and how you train it. Try other settings of the network parameters and/or training parameters, and describe the changes on classification accuracy and training time.
4. Report the best classification accuracy and the corresponding confusion matrices obtained by the above two classification methods (i.e., LRC (or SVM) versus LeNet). Evaluate and compare their classification performance.
5. Attach your code at the end of the report.

### Task 3 – Image Classification with a Pre-trained CNN Model (6 marks)

Carefully read the following webpage

[https://mxnet.apache.org/api/r/docs/tutorials/classify\\_real\\_image\\_with\\_pretrained\\_model](https://mxnet.apache.org/api/r/docs/tutorials/classify_real_image_with_pretrained_model) and use the provided code to classify some object images. You are free to use any object images that are similar to those in the ImageNet dataset (<http://www.image-net.org/challenges/LSVRC/>).

After that, download (a reduced version of) Caltech256 training and test sets via the provided links.

<https://drive.google.com/drive/folders/0Bwnyd83DcfEdX0ppY0x4S293X00>

Unzip the file and you will see 257 folders. Each folder corresponds to one class. Use the pre-trained “Inception-BatchNorm” network provided in the first webpage to extract feature representation of each image in the Caltech256 training and test sets. In addition to the above webpages, you may find the following webpage to be helpful:

<https://github.com/apache/incubator-mxnet/blob/master/R-package/vignettes/classifyRealImageWithPretrainedModel.Rmd>.

When feature representations are extracted for all images, train a logistic regression classifier (LRC) or a multi-class linear support vector machine (SVM) with the training data set and test it on the testing data set.

In your report, you need to

1. Describe how you use the pre-trained Inception-BatchNorm network to classify some object images.
2. Describe the provided Caltech256 training and test sets.
3. Describe how you use the pre-trained Inception-BatchNorm network to extract feature representation for each image in the Caltech256 dataset.
4. Report the best classification accuracy obtained by the LRC or SVM classifier. List the top 10 pairs of classes that are confused most in the confusion matrix.
5. Attach your code at the end of the report.

### Submit:

#### Important:

1. **The report must be in PDF format.**
2. **The report shall contain sufficient and detailed description, explanation, justification and discussion. Marks will be deducted for a BRIEF report.**
3. **Sufficient annotation shall be provided in your code to make it easy to understand.**

Neatly print your report and code (i.e. first the report then the code) on A4 pages with an appropriate cover sheet and hand it in during the lecture on the 30<sup>th</sup> of October 2019. Make sure your report and code are correctly formatted and titled. (Marks will be deducted for untidy or incorrectly formatted work.) Also, submit your report and the source code in a Zipped file named A3.zip via the submit link provided in the Moodle site.

Note: Failure of your code to run may attract zero marks. Code or reports considered to be unreasonably same due to copying will attract zero marks. You may be requested to demonstrate and explain your program when necessary. Marks will be awarded for correct design, implementation and style. Any request for an extension of the submission deadline or demonstration time limit must be made to the Subject Coordinator before the submission deadline. Supporting documentation must accompany the request for any extension. Late assignment submissions without granted extension will be marked but the mark awarded will be reduced by 25% of the assignment mark for each day (including weekends) late.

--- END ---