## Assignment 2 (20%) – Computer vision with CNNs

**California State University San Bernardino, School of Computer Science and Engineering (CSE)**

**Date of Issue**: April 24, 2024, **Date of submission**: **May 10, 2024 – 11:59 pm (PST)**

**Module**: CSE 5120 Introduction to Artificial Intelligence

**Assessment brief:** We have covered the power of deep learning in real-world object detection and recognition in our course that included the use of Artificial Neural Networks (ANNs) and their successors Convolutional Neural Networks (CNNs). Object recognition in image/video data is an increasingly famous example of harnessing the power of deep learning in solving real-world problems. This assignment requires designing the most basic example of object recognition for machine and deep learning which will be based on handwritten digit recognition using CNNs. The dataset to be used for this assignment comprises of 70,000 images of preprocessed images of digits scanned from a large number of documents by the National Institute of Standards and Technology (NIST). We will use this modified NIST (MNIST) dataset and implement a CNN model that should achieve near state-of-the-art performance in Python using Keras deep learning library.

After completing this assignment, you will have expert skills on:

* Loading the MNIST dataset in Keras.
* Developing near state-of-the-art deep learning model for MNIST using CNNs.

Your assignment is based on two parts as given below:

1. Code implemented for handwritten digit recognition (MNIST) via CNNs in given digitRecognizer.py and evaluation.py files in **Files\_to\_edit\_and\_sample\_images.zip** folder.
2. A brief report on what you did for your digit recognition model (i.e., how you implemented with screenshots)

|  |  |
| --- | --- |
| **File Name** | **Description** |
| digitRecognizer.py | Where all of your handwritten digit recognition code via CNNs for MNIST dataset will reside. In this file, you will need to implement your CNN model from scratch using Keras deep learning library. At the end of this file, you should save your model and upload your saved model (.h5 file). This will be used to load your trained model, recognize digits of test cases, and print your model score in the output window. You should also print model accuracy and your model predictions through your code in this file and copy them in your assignment report (wherever needed in the report) |
| evaluation.py | The evaluation file where you will implement your code to load your model, run it on test cases for MNIST dataset, and print your model score. |
| digitRecognizer.h5 | This will be the trained model (CNN) on ***MNIST*** dataset that you will produce in your digitRecognizer.py file using Keras built-in functions *model.save(‘digitRecognizer.h5’)* and *load\_model(digitRecognizer.h5’)* |
| Assignment Report.doc | Report explaining your work for all the above work with description and screenshots. See below for more details |

You can use Spyder (installed through Anaconda from week 1 Wednesday’s lecture) or other IDE.

**Files to Edit and Submit:** You will need to edit and submit (digitRecognizer.py) file which will consist of your CNN model for MNIST dataset. You can copy and paste all the necessary pieces of code that we discuss here in the Assignment details section. Once you have completed the implementation of your model in this file, you should save your model with the name “digitRecognizer.h5” and upload it in Zipped folder in your assignment. To load your model, you can do the import “from keras.models import load\_model” library from Keras. Once completed your digitRecognizer.py, you will need to implement the code in evaluation.py file to load your model (i.e., digitRecognizer.h5) via load\_model, **predict the classes for the test cases via the function already implemented in** evaluation.py and print your predictions. For the development of your CNN model, please refer to the flowchart given in the Assignment details section and the code we covered in Week 13 for CNNs implemented for cat/dog classification. You will need to test your model with evaluation.py to copy screenshots in your report.

**Academic Dishonesty:** Your code will be checked against other submissions in the class for logical redundancy. If you copy someone else’s code and submit it with minor changes, they will be detected, so please do not try that, and submit your own work only. In case of cheating, the University’s academic policies on cheating and dishonesty will strictly apply which may result from the deduction in your grade to expulsion.

**Getting Help:** If you are having difficulty in implementing your CNN model, contact course staff for help and refer to the content recorded and uploaded in Week 13 part 2 (Wednesday) on Computer Vision using CNNs). Office hours and Slack are there for your support. If you are not able to attend office hours, then please inform your instructor to arrange for additional time. The intent is to make this assignment rewarding and instructional, not frustrating and demoralizing. You can either complete this assignment on your own or discuss the problem and collaborate with another member of the class (or different section). Please clearly acknowledge and mention your group member in your assignment report submission who you will collaborate with in this assignment. Your report and program (digitRecognizer.py, evaluation.py files and digitRecognizer.h5 model) will be separately submitted by yourself on Blackboard irrespective of your collaboration with your group member. Group discussions are encouraged but copying of programs is NOT allowed. Programming based on your own skills is encouraged.

**Assignment Details**

1. **Modified National Institute of Standards and Technology (MNIST) dataset.**

The MNIST dataset was compiled and processed by Yann LeCun, Corinna Cortes and Christopher Burges to develop and evaluate new machine learning models on handwritten digit classification problems. The dataset was constructed from scanned document dataset available from the NIST, therefore naming it to Modified NIST (MNIST). As discussed in the class, the images of digits were taken from a variety of scanned documents, and then normalized in size and centered for model development. Image data preprocessing already makes the dataset in nearly ready form requiring little to no cleaning/augmentation task unlike we experienced in cat/dog classification task in Week 13. Each image in the dataset is a 28 × 28 pixel square, with a total of 784 pixels. When you load the dataset via Keras library in your code, you will realize that the dataset is already divided into train and test splits. There is a total of 70,000 images where 60,000 images are used to train a model and a separate set of 10,000 images are used to test it.

To date, various models including the CNNs have been developed with near state-of-the-art accuracy of above 99%. You have also seen a live demonstration of handwritten digit recognition via the web in the class. Your task will be to develop your own similar CNN for handwritten digit recognition via Keras library in Python. With a fairly larger CNN (see section 2 for implementation), it is expected that you will achieve an accuracy of nearly or even above 97% with <3% classification error. To learn how to load and assign train/test portions of the data, please visit [this web page:](https://nextjournal.com/gkoehler/digit-recognition-with-keras)

1. **Convolutional Neural Network (CNN) for MNIST dataset – 20%**

Please see below flowchart to implement your CNN model. For more details, please see the empty digitRecognizer.py file with comments to implement your code, and review code example covered in Week 13 on Computer Vision using Convolutional Neural Networks (CNN) with the example of cat/dog classification. **Please note that you might need to change your CNN settings (i.e., number of Conv filters in each layer, number of units in each Dense layer or even adding/removing Conv/Dense layers) to achieve the accuracy of 97% or above.**



**Note:** For model evaluation, you can use built-in Keras model evaluation function by writing code like model.evaluate(X\_test, y\_test, verbose=0). For this work, please write the description of your work in the report (2-page maximum) that may include your modifications in the CNN structure, or the description of your CNN even if you did not make any modification in the structure but still were able to get the accuracy of >97%. Use screenshots wherever needed.

**Evaluation**: You will copy the code in your evaluation.py file to only load your saved model from digitRecognizer.py file, use your saved model and make prediction on a few test images given in the assignment folder.

1. **Assignment Report – 20%**

Your report will be based on your explanation on MNIST dataset which will include your explanation of your CNN model implementation. You can even propose your own test images of custom drawn images (from 0 to 9) in a software of your own choice (e.g., digits drawn in MS paint with black background and white digit pixels). Please use screenshots (or manually edited versions) of your model training process to support your explanation. Your model training process should print 10 epochs with model metrics.

**Assignment 2** (20%)

CSE 5120 (Section 01) – Introduction to Artificial Intelligence – Spring 2024

*Submitted to*

Department of Computer Science and Engineering  
California State University, San Bernardino, California

*by*

Student name (CSUSB ID)

(Your collaborator in this homework (if any))

Date: *Month Day, Year*

*Email:*

* *Your email*
* *Your collaborator’s email (if you collaborated with any)*

**Assignment Report**

Brief description of your work here acknowledging your collaboration with your class fellow (or a friend from other CSE 5120 section), and the capacity at which he/she collaborated with you, followed by the algorithms you implemented.

1. **digitRecognizer.py for MNIST dataset**

Your brief explanation of the dataset, your code solution, and any documentation with screenshots of your code Evaluation (results from digitRecognizer.py)

1. **Evaluation (evaluation.py) for your model performance evaluation**

You can also provide brief description of your code written in evaluation.py to load your saved model that can be readily used on test dataset for the staff.