San Jose State University

Department of Electrical Engineering

EE104, Fall 2021, Pham

Laboratory Assignment #8

# Objectives

This lab allows you to use CNN (Convolutional Neural Network) to recognize color images and use different methods to improve the accuracy.

# Grading

Refer to the section **Python Programming** for grading criteria.

# Bibliography

I would like to acknowledge the Python open-source community and respective suppliers for making the material available.

References:

Keras: <https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529>

CIFAR-10 dataset: <https://www.cs.toronto.edu/~kriz/cifar.html>

Baseline code: <https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/images/cnn.ipynb#scrollTo=WRzW5xSDDbNF>

# Download, Installation, and Licensing

## Install necessary Python packages

**From Google Colab: (It is the requirement for this lab to run from Google Colab)**

You will need to PIP INSTALL the followings:

!pip install tensorflow

You may need to install the followings. Check it out yourself!

!pip install keras

!pip install h5py

!pip install Matplotlib

!pip install numpy

# Graphic Files

1. CIFAR-10 Data Set: <https://www.cs.toronto.edu/~kriz/cifar.html>

The CIFAR-10 dataset is an image collection dataset created by the Canadian Institute for Advanced Research. This dataset contains 60,000 32x32 colored images in 10 different classes: airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

1. Given set of images for testing purposes.

# Tutorial Python Programs

1. You will perform the steps outlined from this tutorial: [**https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/images/cnn.ipynb#scrollTo=WRzW5xSDDbNF**](https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/images/cnn.ipynb#scrollTo=WRzW5xSDDbNF) **.** This baseline code achieved a test accuracy of over 70%.
2. You will modify the base code from the tutorial in step 1, and add your own code as needed using different techniques that you learned from the lecture notes to improve the accuracy. You will earn points as detailed in the Your Deliveries section.
3. You will add your own code to test with given images.

# Your Deliverables

Now that you are familiar with the basic concept of CNN, you can run more advanced models to achieve higher recognition accuracy.

Run the code from three sections to achieve 15% credit each:

|  |  |
| --- | --- |
| **Baseline + Increasing Dropout** | To achieve > 84% accuracy |
| **Baseline + Dropout + Data Augmentation** | To achieve > 86% accuracy |
| **Baseline + Increasing Dropout + Data Augmentation + Batch Normalization** | To achieve > 88% accuracy |

Based on what you learned from the provided code, develop your own method to achieve > 90% accuracy for another 15% credit:

|  |  |
| --- | --- |
| Your own method | To achieve > 90% accuracy |

Next you will provide 5 images for each of the 10 criteria and validate with your revised code that the code can recognize these images. Successful recognition of each image will earn you 5% (for maximum 25%) credit.

# Python Programming

#### Lab Submission

Once you learn the process and the code associate with each step in the process, you will be able to customize the program to do the followings.

|  |  |  |
| --- | --- | --- |
| **Program or Requirement** | **Use Case** | **Earned Score / Max Score** |
| README file & Developer Documentation | README is a brief user guide and developer documentation so that the user can install the proper python packages and knows how to execute your program.  The documentation section can contain sample screenshots with explanation. | \_\_\_\_\_ / 10 |
| Check in your README file, documentation, and codes to Github | You will put all your work on your Github account and submit the same code to Canvas. | \_\_\_\_\_ / 5 |
| **Baseline + Increasing Dropout** | To achieve > 84% accuracy and recognize successfully 84% of the given test images.  See video recording note. | \_\_\_\_\_ / 20 |
| **Baseline + Dropout + Data Augmentation** | To achieve > 86% accuracy and recognize successfully 86% of the given test images.  See video recording note. | \_\_\_\_\_ / 20 |
| **Baseline + Increasing Dropout + Data Augmentation + Batch Normalization** | To achieve > 88% accuracy and recognize successfully 88% of the given test images.  See video recording note. | \_\_\_\_\_ / 20 |
| Your own method | To achieve > 90% accuracy and recognize successfully 90% of the given test images.  See video recording note. | \_\_\_\_\_ / 20 |
| Challenge test | Recognize all unrecognizable images in the given test set | \_\_\_\_\_ / 5 |
|  | **TOTAL** | **100%** |

Video recording note: Because it will take a long time to run this test, you must submit a recording showing the result. The recording can be done by Zoom or any screen video capturing software.

That’s all for this lab. Hopefully you found it useful and increase your interest in the Python world! See you in the next lab.

# TEST IMAGES

Below are suggestions that you can use to test the accuracy of your code.

## 'airplane',

* <https://cbsnews2.cbsistatic.com/hub/i/r/2017/03/27/73bd41ff-5703-48ca-995c-131d1b3572b4/thumbnail/640x335/10f4b442d725b8fa79d3e2dbf286ba76/air-force-one-two-planes.jpg>
* <https://upload.wikimedia.org/wikipedia/commons/thumb/f/f0/Another_Airplane%21_%284676723312%29.jpg/1024px-Another_Airplane%21_%284676723312%29.jpg>
* <https://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/Voyager_aircraft.jpg/1200px-Voyager_aircraft.jpg>
* <https://static.wikia.nocookie.net/lostpedia/images/f/f7/Kateandplane.jpg/revision/latest/top-crop/width/360/height/450?cb=20061109012445> (not recognizable)
* <https://www.zdnet.com/a/img/resize/071727877ee9884b60edd728253d2baadcb3985f/2021/02/23/19631992-64df-4af9-a288-a0cb4112e682/bombardier-globaleye-jet.jpg?width=1200&height=900&fit=crop&auto=webp> (not recognizable)

## 'automobile',

* <https://sniteartmuseum.nd.edu/assets/166204/original/ferrari.jpg>
* <https://www.kbb.com/articles/wp-content/uploads/2020/04/00-2020-bmw-8-series-gran-coupe.jpg>
* <https://images.all-free-download.com/images/graphiclarge/classic_jaguar_210354.jpg> (not recognizable)
* <https://hips.hearstapps.com/hmg-prod.s3.amazonaws.com/images/devel-motors-sixteen-1540564064.jpg> (not recognizable)
* <https://amsc-prod-cd.azureedge.net/-/media/aston-martin/images/default-source/models/valkyrie/new/valkyrie-spider_f02-169v2.jpg?mw=1980&rev=-1&hash=92E23C911BDE23D418D37F9187844B7C> (not recognizable)

## 'bird',

* <https://ichef.bbci.co.uk/news/976/cpsprodpb/67CF/production/_108857562_mediaitem108857561.jpg> (not recognizable)
* <https://upload.wikimedia.org/wikipedia/commons/5/53/Weaver_bird.jpg> (not recognizable)
* <https://images.all-free-download.com/images/graphiclarge/flying_bird_201952.jpg> (not recognizable)
* <https://www.allaboutbirds.org/news/wp-content/uploads/2009/04/WKingbird-James.jpg>
* <https://www.pestworld.org/media/560900/istock_000027713740_large.jpg?preset=pestFeature360>

## 'cat',

* <https://static.toiimg.com/thumb/msid-67586673,width-1070,height-580,overlay-toi_sw,pt-32,y_pad-40,resizemode-75,imgsize-3918697/67586673.jpg> (not recognizable)
* <https://wagznwhiskerz.com/wp-content/uploads/2017/10/home-cat.jpg> (not recognizable)
* <https://res.cloudinary.com/dnkxl7hbd/images/f_auto,q_auto/w_400,h_664/v1610516304/Lanai-Cats-we-need-you/Lanai-Cats-we-need-you.png> (not recognizable)
* You can search for two of your own pictures that are recognizable

## 'deer',

* Search for 5 images that are recognizable with your own code.

## 'dog',

* Search for 5 images that are recognizable with your own code.

## 'frog',

* Search for 5 images that are recognizable with your own code.

## 'horse',

* <https://upload.wikimedia.org/wikipedia/commons/0/03/American_quarter_horse.jpg>
* <https://upload.wikimedia.org/wikipedia/commons/f/f0/White_horse.jpg>
* Search for 3 images that are recognizable with your own code.

## 'ship',

* Search for 5 images that are recognizable with your own code.

## 'truck'

* Search for 5 images that are recognizable with your own code.

# Laboratory Hand-In Requirements

Once you have completed a working design, prepare for the submission process. You are required to demonstrate a working design. You are also required to submit an archive of your project in the form of a ZIP file. Use 7-Zip option to create the ZIP file. Name the archive lab#\_yourlastname\_yourfirstname.zip. Refer to Lab 1 for detail instructions.

You will submit your zip file to the instructor through Canvas by the due date and time. If the class will be on campus, then you will expect to demonstrate in the classroom. If we ever have to go back to an online mode, turn in your archive to Canvas along with a narrated video capturing the screen of your computer running your program demonstration. If your program is not completely functional by the due date, you should demonstrate and turn in what you have accomplished to receive partial credit. See the syllabus for the late penalty guideline