EE 104-01 Project 4 Devin Ary
Fall 2020 CIFAR-10 Image Recognition Ahmad Ghanizadah
Professor Pham Alternative Option 4 Super Project (25%) Alon Kopelman

For our project, we pursued option 4, the CIFAR-10 image recognition project, using the alternative CIFAR-10 tutorial as a starting base, completing all 5 steps for the 25% option.

Background:

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.

4.1. For the first portion of this project we successfully completed the tutorial in order to learn more about the CIFAR-10 dataset, completing the development of the final model, and using a testing sample image of a deer in order to verify the final model.

The following .py files are available for running in the '4.1' folder:

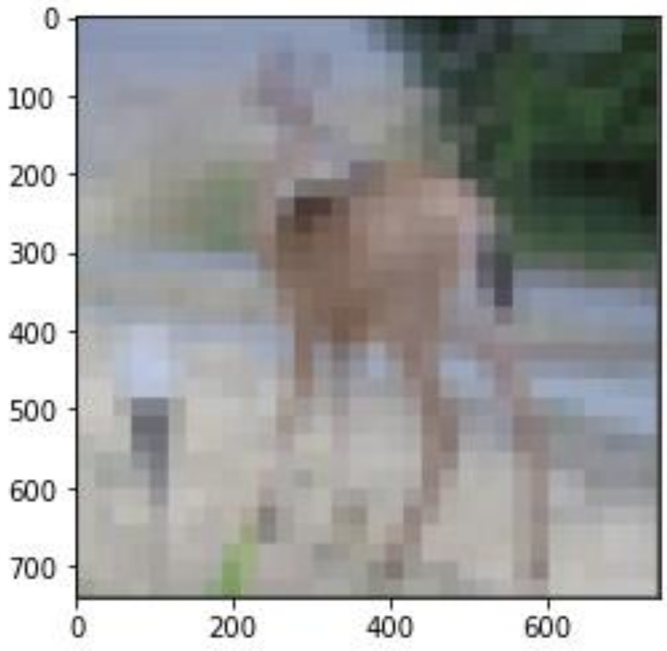
- VGG 1 .py file and console run in .HTML (with inline plots enabled)
- VGG 2 .py file and console run in .HTML (with inline plots enabled)
- VGG 3 .py file and console run in .HTML (with inline plots enabled)
- VGG 3 with Dropout Update .py file and console run in .HTML (with inline plots enabled)
- VGG 3 with Weight Decay Update .py file and console run in .HTML (with inline plots enabled)
- VGG 3 with Data Augmentation Update .py file and console run in .HTML (with inline plots enabled)
- VGG 3 with Dropout Regularization Update .py file and console run in .HTML (with inline plots enabled)
- VGG 3 with Dropout Regularization and Data Augmentation Update .py file and console run in .HTML (with inline plots enabled)
- VGG 3 with Dropout Regularization and Data Augmentation and Batch Linearization Update .py file and console run in .HTML (with inline plots enabled)
- Final model (with 100 epochs and 64 batch if you plan to do numbers 2-5 in option 4, else used the suggested sizes in the demo video) .py file and console run in .HTML
- Final model evaluation .py file and console run in .HTML
- Final model deer sample prediction .py file and console run in .HTML (with inline plots enabled)

However, it is suggested to just run 'Option_4_1_Prediction.py' because most of the programs require some time to run. To compensate, the console runs for each program are available in the 'Console_Runs' folder under the '4.1' folder, along with plots of each run available in the 'Run_Plots' folder under the '4.1' folder.

‘Option_4_1_Prediction.py’ console output taken from ‘4.1’ folder → ’Console_Runs’:

```
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Ahmad/Dropbox/EE104/Option_4/Option_4_1_Prediction.py',
wdir='C:/Users/Ahmad/Dropbox/EE104/Option_4')

2020-11-17 18:44:13.801593: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll

2020-11-17 18:44:13.801593: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll
2020-11-17 18:44:18.109913: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library nvcuda.dll
2020-11-17 18:44:18.163692: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1716] Found device
0 with properties:
pciBusID: 0000:01:00.0 name: GeForce GTX 1060 3GB computeCapability: 6.1
coreClock: 1.7085GHz coreCount: 9 deviceMemorySize: 3.00GiB deviceMemoryBandwidth: 178.99GiB/s
2020-11-17 18:44:18.163739: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll

2020-11-17 18:44:13.801593: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll
2020-11-17 18:44:18.109913: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library nvcuda.dll
2020-11-17 18:44:18.163692: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1716] Found device
0 with properties:
pciBusID: 0000:01:00.0 name: GeForce GTX 1060 3GB computeCapability: 6.1
coreClock: 1.7085GHz coreCount: 9 deviceMemorySize: 3.00GiB deviceMemoryBandwidth: 178.99GiB/s
2020-11-17 18:44:18.163739: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll
2020-11-17 18:44:18.707748: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cublas64_10.dll
2020-11-17 18:44:18.998941: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cufft64_10.dll
2020-11-17 18:44:19.029353: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library curand64_10.dll
2020-11-17 18:44:19.328222: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cusolver64_10.dll
WARNING:tensorflow:From C:\Users\Ahmad\Dropbox\EE104\Option_4\Option_4_1_Prediction.py:31:
Sequential.predict_classes (from tensorflow.python.keras.engine.sequential) is deprecated and will
be removed after 2021-01-01.
Instructions for updating:
Please use instead:* `np.argmax(model.predict(x), axis=-1)`, if your model does multi-class
classification (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) >
0.5).astype("int32")`, if your model does binary classification (e.g. if it uses a `sigmoid` last-
layer activation).

2020-11-17 18:44:13.801593: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll
2020-11-17 18:44:18.109913: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library nvcuda.dll
2020-11-17 18:44:18.163692: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1716] Found device
0 with properties:
pciBusID: 0000:01:00.0 name: GeForce GTX 1060 3GB computeCapability: 6.1
coreClock: 1.7085GHz coreCount: 9 deviceMemorySize: 3.00GiB deviceMemoryBandwidth: 178.99GiB/s
```

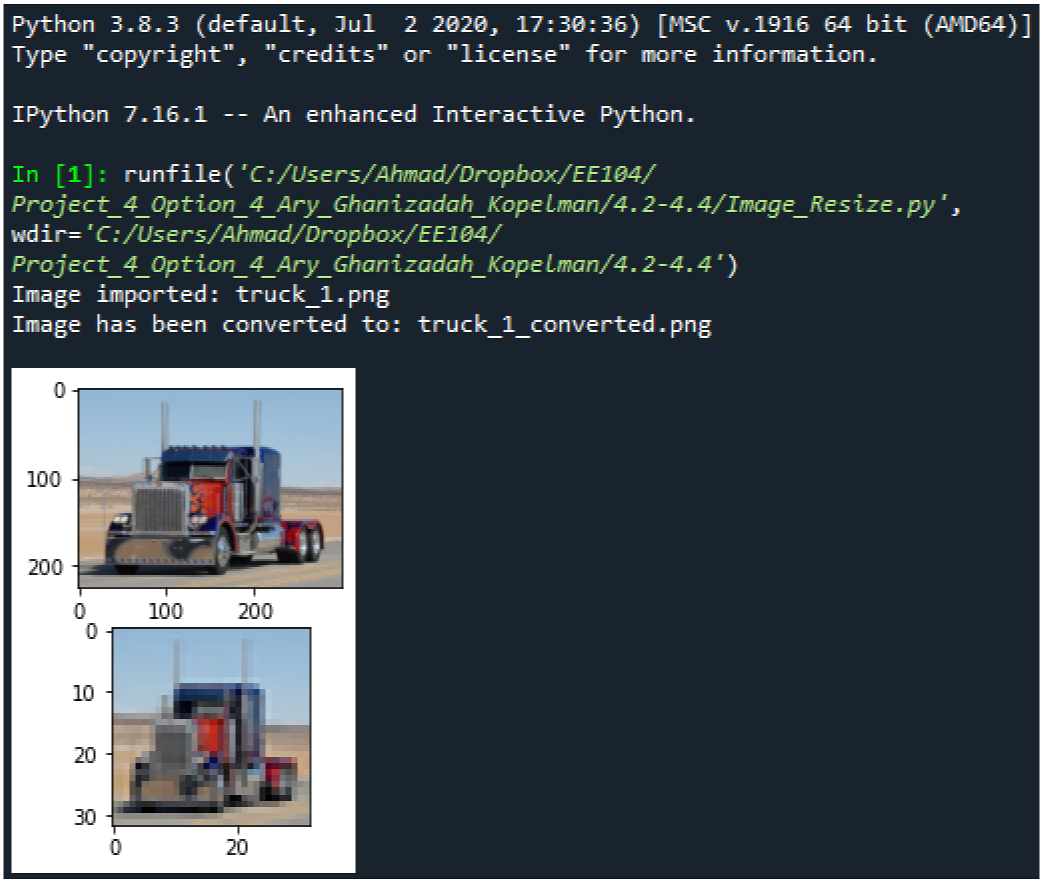
```
2020-11-17 18:44:18.163739: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll
2020-11-17 18:44:18.707748: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cublas64_10.dll
2020-11-17 18:44:18.998941: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cufft64_10.dll
2020-11-17 18:44:19.029353: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library curand64_10.dll
2020-11-17 18:44:19.328222: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cusolver64_10.dll
2020-11-17 18:44:19.605889: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cusparse64_10.dll
2020-11-17 18:44:19.606758: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could
not load dynamic library 'cudnn64_7.dll'; dlerror: cudnn64_7.dll not found
2020-11-17 18:44:19.606776: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1753] Cannot dlopen
some GPU libraries. Please make sure the missing libraries mentioned above are installed properly
if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how
to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2020-11-17 18:44:19.608871: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)to use the following CPU
instructions in performance-critical operations: AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2020-11-17 18:44:19.630483: I tensorflow/compiler/xla/service/service.cc:168] XLA service
0x1c45f90f570 initialized for platform Host (this does not guarantee that XLA will be used).
Devices:
2020-11-17 18:44:19.630510: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor
device (0): Host, Default Version
2020-11-17 18:44:19.631249: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1257] Device
interconnect StreamExecutor with strength 1 edge matrix:
2020-11-17 18:44:19.631265: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1263]
Image Classes from 0 to 9: ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',
'horse', 'ship', 'truck']
image falls under class:
4
--- 7.1301493644714355 seconds ---
```



In [2]:

4.2. We were then tasked with creating our own dataset by collecting 20 images of: airplanes, automobile, birds, cats, deer, dogs, frogs, horses, ships, trucks. We found 20 images of each from Google images and saved them to a folder called ‘Pictures_Database’ in the ‘4.2-4.4’ folder. We were then tasked with converting our images into a 32x32x3 pixel format where the ‘x3’ means the RGB color vector, so a 32x32 image, while maintaining color. We wrote a python program called ‘Image_Resize.py’ located in the ‘4.2-4.4’ folder. Which converts an imported PNG image into 32x32x3 format using the PIL library. The converted images were then placed in the ‘Pictures_Converted’ folder, however the original images and converted images are also sitting in the ‘4.2-4.4’ folder in order to provide usability to users who download the file, without needing to change and directories in the programs.

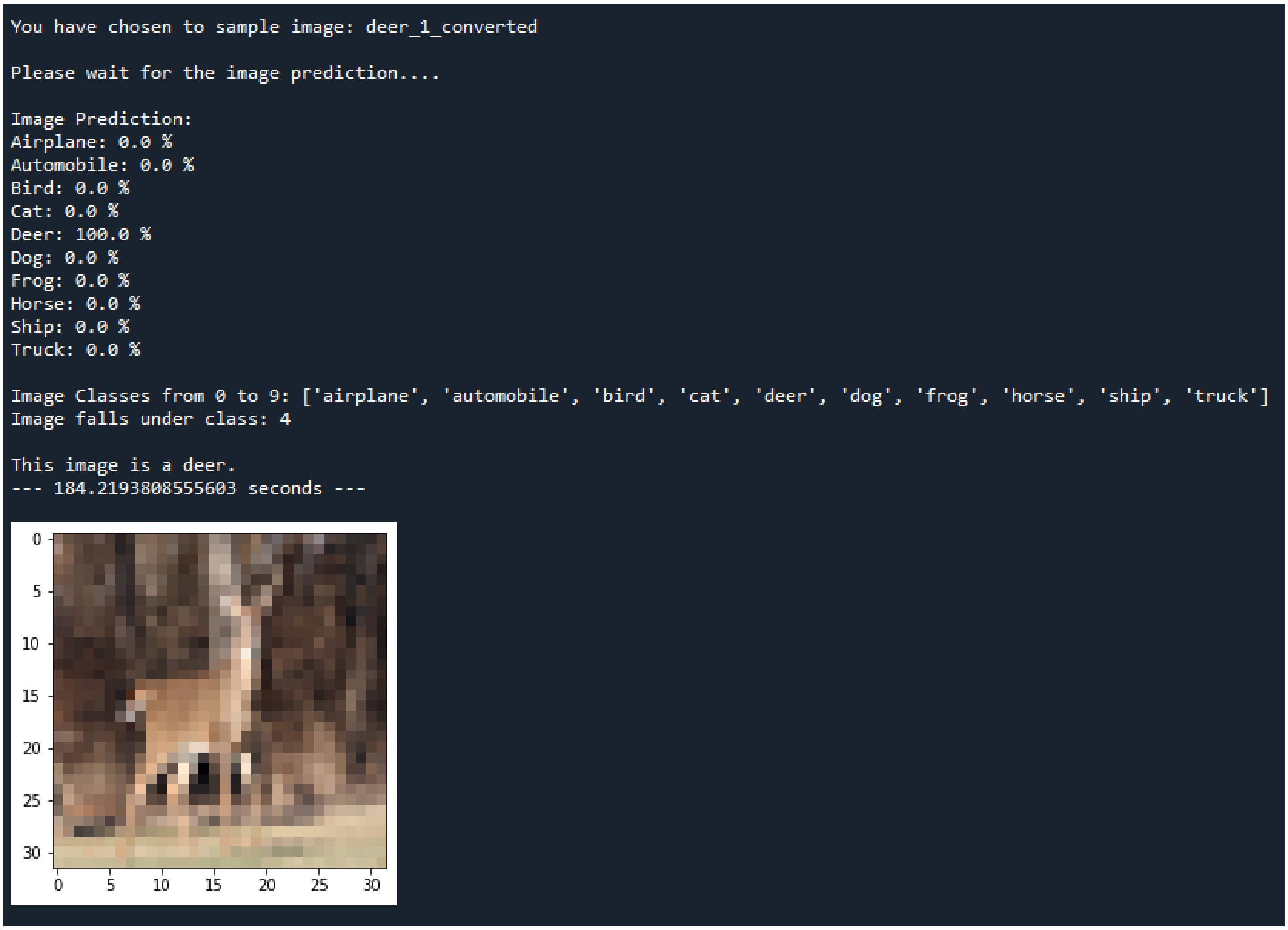
‘Image_Resize.py’ (located in ‘4.2-4.4’ folder) Spyder console output:



```
Python 3.8.3 (default, Jul  2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Ahmad/Dropbox/EE104/
Project_4_Option_4_Ary_Ghanizadah_Kopelman/4.2-4.4/Image_Resize.py',
wdir='C:/Users/Ahmad/Dropbox/EE104/
Project_4_Option_4_Ary_Ghanizadah_Kopelman/4.2-4.4')
Image imported: truck_1.png
Image has been converted to: truck_1_converted.png
```

4.3-4.4. Since the CIFAR-10 was a closed dataset, we used our 20 converted images in 32x32x3 format, and created a program called ‘Database_Prediction.py’ under the ‘4.2-4.4’ folder. This program imports all 20 of the images and presents the user with an option to run any of the possible images through a selection number from 1-20 (with built-in fail-safes to make sure a selection number is given). Upon the selection number, the program imports the ‘final_model.h5’ machine learning model developed in ‘4.1’ folder to make a prediction of the image class, while also showing the prediction probability. Since the model was trained at 100 epochs and 64 batches, we saw that it was very accurate, usually with 100% prediction scores on most classes except cat, dog, and deer due to the common features. It is encouraged to run this program as it imports a fully developed machine learning model, ‘final_model.h5’, reducing the runtime for each prediction to only a few seconds.

‘Database_Prediction.py’ Spyder console output taken from ‘4.2-4.4’ folder:

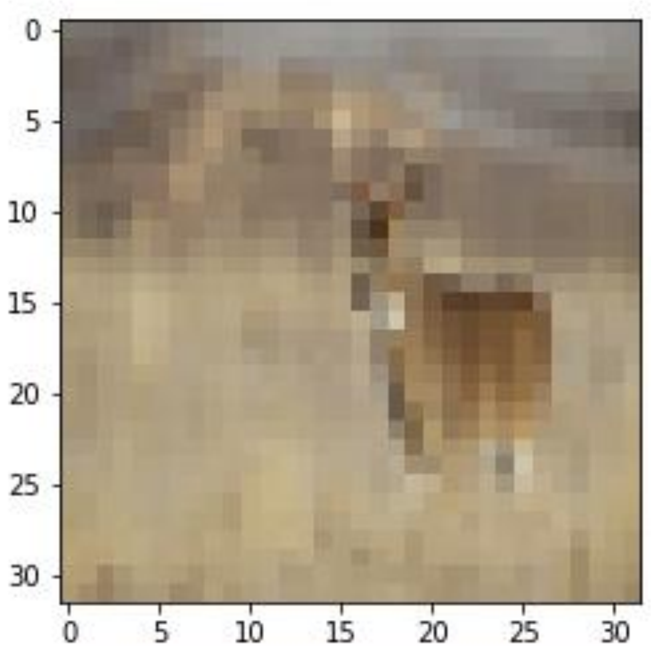


```
You have chosen to sample image: deer_1_converted

Please wait for the image prediction....

Image Prediction:
Airplane: 0.0 %
Automobile: 0.0 %
Bird: 0.0 %
Cat: 0.0 %
Deer: 100.0 %
Dog: 0.0 %
Frog: 0.0 %
Horse: 0.0 %
Ship: 0.0 %
Truck: 0.0 %

Image Classes from 0 to 9: ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']
Image falls under class: 4

This image is a deer.
--- 184.2193808555603 seconds ---
```

‘Database_Prediction’ console output taken from ‘4.2-4.4’ folder → ’Console_Runs’:

```
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Ahmad/Dropbox/EE104/Option_4/4.2-4.4/Database_Prediction.py',
wdir='C:/Users/Ahmad/Dropbox/EE104/Option_4/4.2-4.4')

2020-11-20 16:59:07.168651: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll


There are 20 sample image files.
*Note: Images are generic images saved from Google Images*


Which would you like to view?


Choices are:
1-2: Airplane
3-4: Automobile
5-6: Bird
7-8: Cat
9-10: Deer
11-12: Dog
13-14: Frog
15-16: Horse
17-18: Ship
19-20: Truck


Type '1' through '20' for the files:10
WARNING:tensorflow:From C:\Users\Ahmad\Dropbox\EE104\Option_4\4.2-4.4\Database_Prediction.py:107:
Sequential.predict_classes (from tensorflow.python.keras.engine.sequential) is deprecated and will
be removed after 2021-01-01.
Instructions for updating:
Please use instead:* `np.argmax(model.predict(x), axis=-1)`, if your model does multi-class
classification (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) >
0.5).astype("int32")`, if your model does binary classification (e.g. if it uses a `sigmoid` last-
layer activation).
WARNING:tensorflow:From C:\Users\Ahmad\Dropbox\EE104\Option_4\4.2-4.4\Database_Prediction.py:111:
Sequential.predict_proba (from tensorflow.python.keras.engine.sequential) is deprecated and will
be removed after 2021-01-01.
Instructions for updating:
Please use `model.predict()` instead.

2020-11-20 16:59:07.168651: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll
2020-11-20 16:59:15.133286: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library nvcuda.dll
2020-11-20 16:59:15.158267: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1716] Found device
0 with properties:
pciBusID: 0000:01:00.0 name: GeForce GTX 1060 3GB computeCapability: 6.1
coreClock: 1.7085GHz coreCount: 9 deviceMemorySize: 3.00GiB deviceMemoryBandwidth: 178.99GiB/s
2020-11-20 16:59:15.158312: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cudart64_101.dll
```

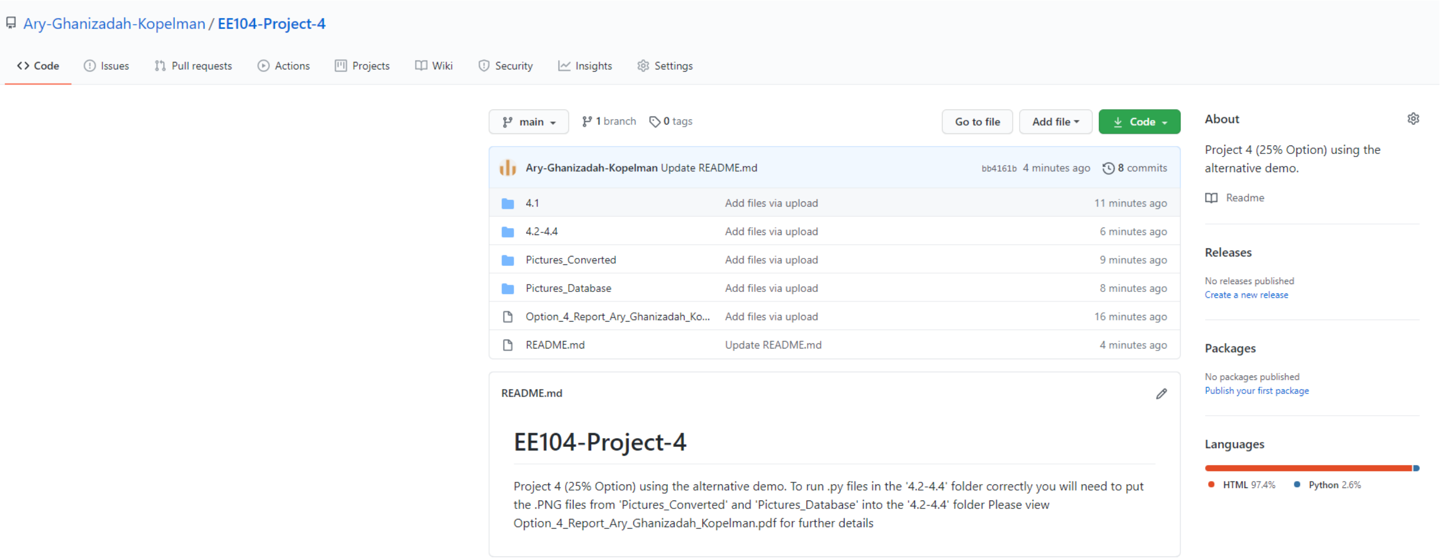
```
2020-11-20 16:59:15.162546: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cublas64_10.dll
2020-11-20 16:59:15.165512: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cufft64_10.dll
2020-11-20 16:59:15.166637: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library curand64_10.dll
2020-11-20 16:59:15.171057: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cusolver64_10.dll
2020-11-20 16:59:15.174216: I tensorflow/stream_executor/platform/default/dso_loader.cc:48]
Successfully opened dynamic library cusparse64_10.dll
2020-11-20 16:59:15.175038: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could
not load dynamic library 'cudnn64_7.dll'; dlerror: cudnn64_7.dll not found
2020-11-20 16:59:15.175054: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1753] Cannot dlopen
some GPU libraries. Please make sure the missing libraries mentioned above are installed properly
if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how
to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2020-11-20 16:59:15.175498: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)to use the following CPU
instructions in performance-critical operations: AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2020-11-20 16:59:15.182826: I tensorflow/compiler/xla/service/service.cc:168] XLA service
0x18df169a890 initialized for platform Host (this does not guarantee that XLA will be used).
Devices:
2020-11-20 16:59:15.182860: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor
device (0): Host, Default Version
2020-11-20 16:59:15.182982: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1257] Device
interconnect StreamExecutor with strength 1 edge matrix:
2020-11-20 16:59:15.182994: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1263]



You have chosen to sample image: deer_2_converted

Please wait for the image prediction....

Image Prediction:
Airplane: 0.0 %
Automobile: 0.0 %
Bird: 8.21 %
Cat: 8.67 %
Deer: 73.45 %
Dog: 0.03 %
Frog: 9.65 %
Horse: 0.0 %
Ship: 0.0 %
Truck: 0.0 %

Image Classes from 0 to 9: ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',
'horse', 'ship', 'truck']
Image falls under class: 4

This image is a deer.
--- 9.430943250656128 seconds ---
```



In [2]:

4.5. This project and all mentioned files will be posted on:

https://github.com/Ary-Ghanizadah-Kopelman/EE104-Project-4



Ary-Ghanizadah-Kopelman Update README.md bb4161b 4 minutes ago 8 commits

4.1	Add files via upload	11 minutes ago
4.2-4.4	Add files via upload	6 minutes ago
Pictures_Converted	Add files via upload	9 minutes ago
Pictures_Database	Add files via upload	8 minutes ago
Option_4_Report_Ary_Ghanizadah_Ko...	Add files via upload	16 minutes ago
README.md	Update README.md	4 minutes ago

README.md

EE104-Project-4

Project 4 (25% Option) using the alternative demo. To run .py files in the '4.2-4.4' folder correctly you will need to put the .PNG files from 'Pictures_Converted' and 'Pictures_Database' into the '4.2-4.4' folder Please view Option_4_Report_Ary_Ghanizadah_Kopelman.pdf for further details

About

Project 4 (25% Option) using the alternative demo.

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● HTML 97.4% ● Python 2.6%

- Github Note:
  - To properly run all of the .py files ‘4.2-4.4’ folder, the PNG files from the ‘Pictures_Converted’ and ‘Pictures_Database’ folder will need to be placed into the ‘4.2-4.4’ folder. This issue is because Github and a folder size limit