Student Name: Gowtham Kumar Kamuni
Student ID: A20549435
LAB EXPERIMENT 2

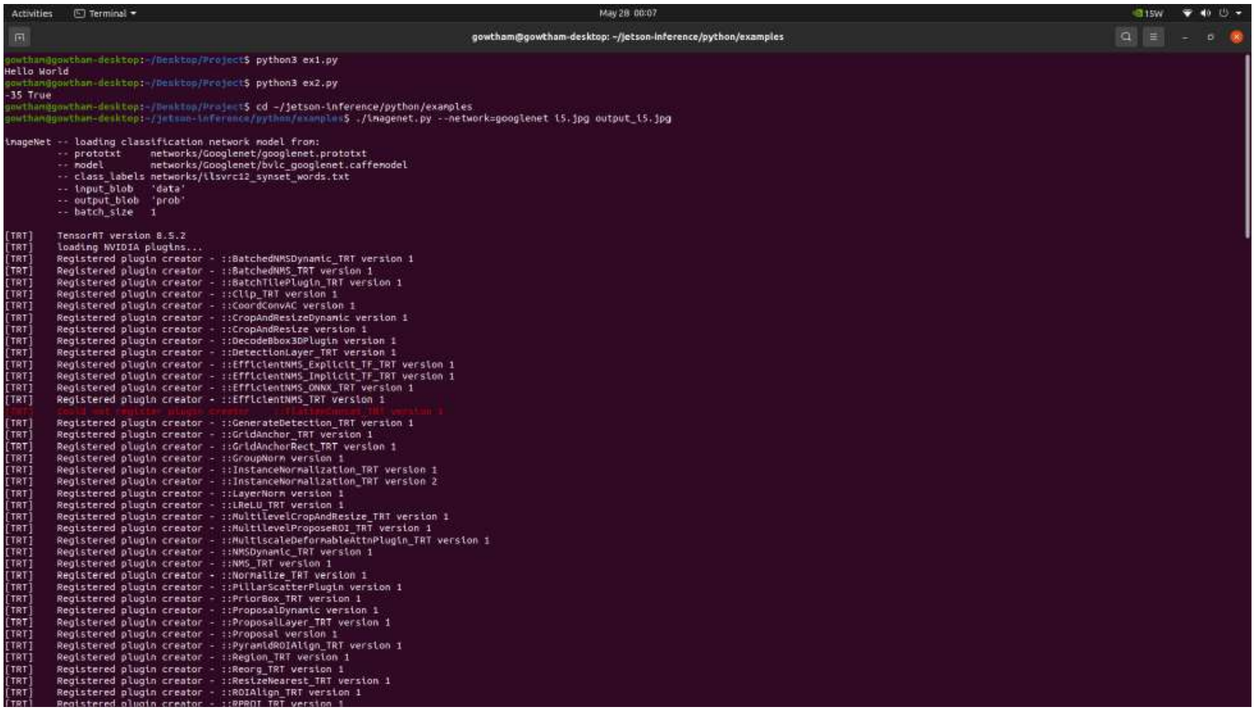
OUTPUTS:

Jetson Orin Nano – Image Classification Output Summary

This document presents the results from image classification tests performed using the GoogLeNet model on the Jetson Orin Nano. Each test image was processed through the imagenet.py script provided in NVIDIA’s jetson-inference library.

For each case, the original input image is shown alongside the corresponding output with a predicted label and confidence score overlayed. The intent of this output file is to visually demonstrate the performance, accuracy, and behavior of the classification model across a variety of real-world test cases captured during the lab. All outputs were generated directly on the Jetson device as part of the Lab Experiment 2 demonstration.

1. Preliminary Work:



```
gowtham@gowtham-desktop:~/Desktop/Project$ python3 ex1.py
Hello World
gowtham@gowtham-desktop:~/Desktop/Project$ python3 ex2.py
-35 True
gowtham@gowtham-desktop:~/Desktop/Project$ cd ~/jetson-inference/python/examples
gowtham@gowtham-desktop:~/jetson-inference/python/examples$ ./imagenet.py --network=googlenet i5.jpg output_i5.jpg

imageNet -- loading classification network model from:
         -- prototxt     networks/Googlenet/googlenet.prototxt
         -- model        networks/Googlenet/bvlc_googlenet.caffemodel
         -- class_labels networks/ilsvrc12_synset_words.txt
         -- input_blob   'data'
         -- output_blob  'prob'
         -- batch_size   1

[TRT]    TensorRT version 8.5.2
[TRT]    loading NVIDIA plugins...
[TRT]    Registered plugin creator - ::BatchedNMSDynamic_TRT version 1
[TRT]    Registered plugin creator - ::BatchedNMS_TRT version 1
[TRT]    Registered plugin creator - ::BatchTilePlugin_TRT version 1
[TRT]    Registered plugin creator - ::Clip_TRT version 1
[TRT]    Registered plugin creator - ::CoordConvAC version 1
[TRT]    Registered plugin creator - ::CropAndResizeDynamic version 1
[TRT]    Registered plugin creator - ::CropAndResize version 1
[TRT]    Registered plugin creator - ::DecodeBbox3DPlugin version 1
[TRT]    Registered plugin creator - ::DetectionLayer_TRT version 1
[TRT]    Registered plugin creator - ::EfficientNMS_Explicit_TF_TRT version 1
[TRT]    Registered plugin creator - ::EfficientNMS_Implicit_TF_TRT version 1
[TRT]    Registered plugin creator - ::EfficientNMS_ONNX_TRT version 1
[TRT]    Registered plugin creator - ::EfficientNMS_TRT version 1
[TRT]    Could not register plugin creator -  ::FlattenConcat_TRT version 1
[TRT]    Registered plugin creator - ::GenerateDetection_TRT version 1
[TRT]    Registered plugin creator - ::GridAnchor_TRT version 1
[TRT]    Registered plugin creator - ::GridAnchorRect_TRT version 1
[TRT]    Registered plugin creator - ::GroupNorm version 1
[TRT]    Registered plugin creator - ::InstanceNormalization_TRT version 1
[TRT]    Registered plugin creator - ::InstanceNormalization_TRT version 2
[TRT]    Registered plugin creator - ::LayerNorm version 1
[TRT]    Registered plugin creator - ::LReLU_TRT version 1
[TRT]    Registered plugin creator - ::MultilevelCropAndResize_TRT version 1
[TRT]    Registered plugin creator - ::MultilevelProposeROI_TRT version 1
[TRT]    Registered plugin creator - ::MultiscaleDeformableAttnPlugin_TRT version 1
[TRT]    Registered plugin creator - ::NMSDynamic_TRT version 1
[TRT]    Registered plugin creator - ::NMS_TRT version 1
[TRT]    Registered plugin creator - ::Normalize_TRT version 1
[TRT]    Registered plugin creator - ::PillarScatterPlugin version 1
[TRT]    Registered plugin creator - ::PriorBox_TRT version 1
[TRT]    Registered plugin creator - ::ProposalDynamic version 1
[TRT]    Registered plugin creator - ::ProposalLayer_TRT version 1
[TRT]    Registered plugin creator - ::Proposal version 1
[TRT]    Registered plugin creator - ::PyramidROIAlign_TRT version 1
[TRT]    Registered plugin creator - ::Region_TRT version 1
[TRT]    Registered plugin creator - ::Reorg_TRT version 1
[TRT]    Registered plugin creator - ::ResizeNearest_TRT version 1
[TRT]    Registered plugin creator - ::ROIAlign_TRT version 1
[TRT]    Registered plugin creator - ::RPROI_TRT version 1
```

Fig: Preliminary Work – Python Execution on Jetson Orin Nano

The screenshot above shows successful execution of two basic Python programs on Jetson.

- ex1.py prints **"Hello World"**, confirming Python is running correctly
- ex2.py implements a simple **BankAccount class**, showcasing object-oriented programming with a withdrawal operation and overdraw check.
- The terminal also shows navigation to the jetson-inference directory, ready for image classification.
  This validated the Python environment and ensured Jetson was prepared for the main classification tasks.

<u>Test Case 1:</u>

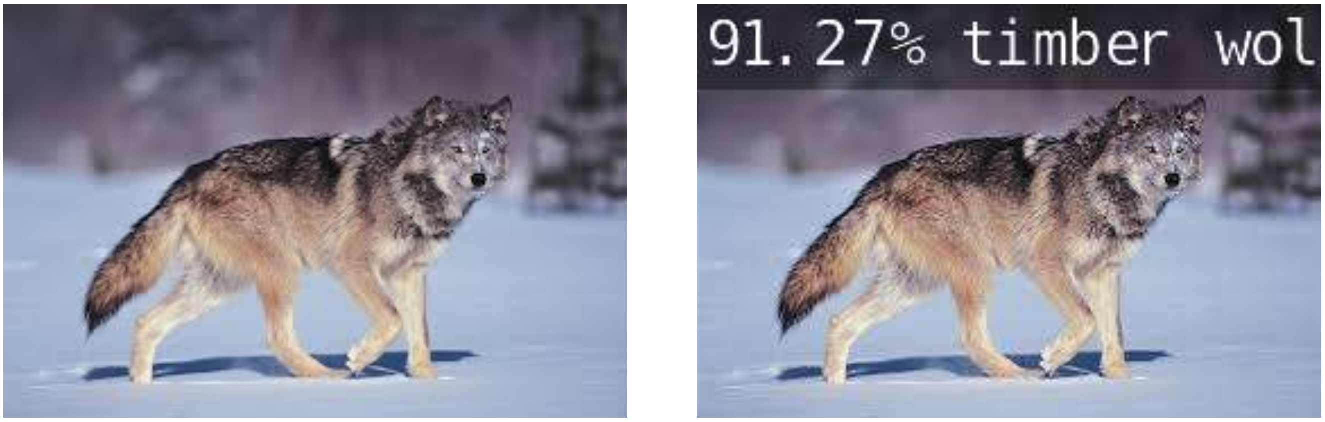


Fig: Timber Wolf

Input Image: A high-resolution image of a timber wolf walking in the snow.

Output Description: The image was successfully recognized with **91.27% confidence** as a "timber wolf". This confirms that the model performs very well on clear, centered images that align with its training dataset.

<u>Test Case 2:</u>

Fig: Backpack

Input: A black “Mountain Adventure” backpack placed on a couch.
Output: Classified as **“blue jean, denim”** with **62.57% confidence**.

Comment: This was a misclassification. The model seemed to rely on color/ texture, likely confusing the lighting on the backpack with denim. Shows limitations in object recognition beyond color similarity.

<u>Test Case 3:</u>

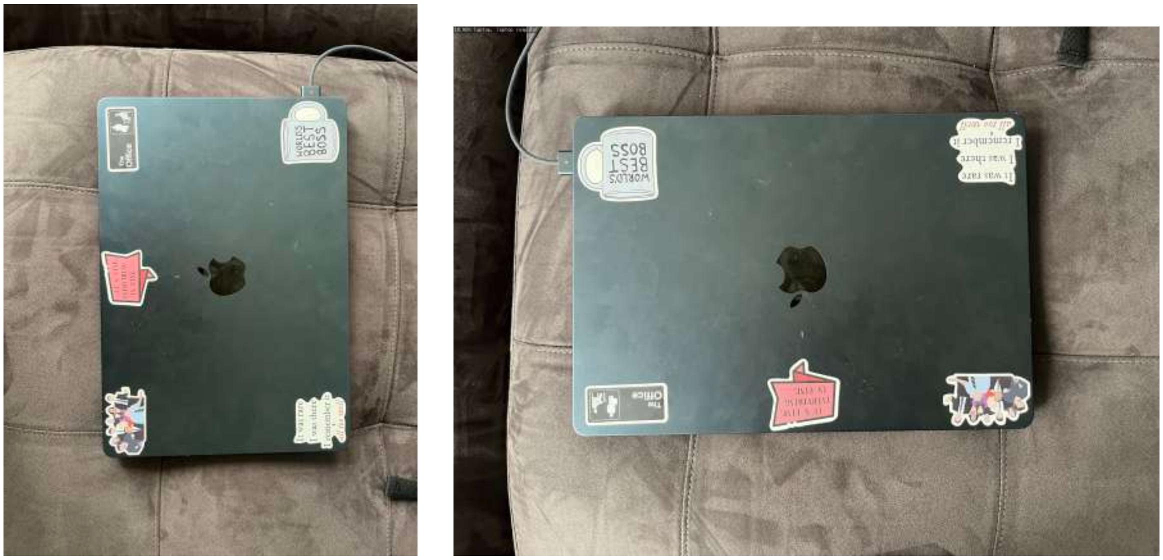


Fig: Laptop

Input: Closed MacBook Air with custom stickers on a brown sofa.
Output: Classified as “laptop, laptop computer” with **18.60%** confidence.
Comment: While the model did identify the object as a laptop, the confidence was quite low. This shows that external visual elements like stickers may impact how clearly the object is recognized by the model.

Test Case 4:

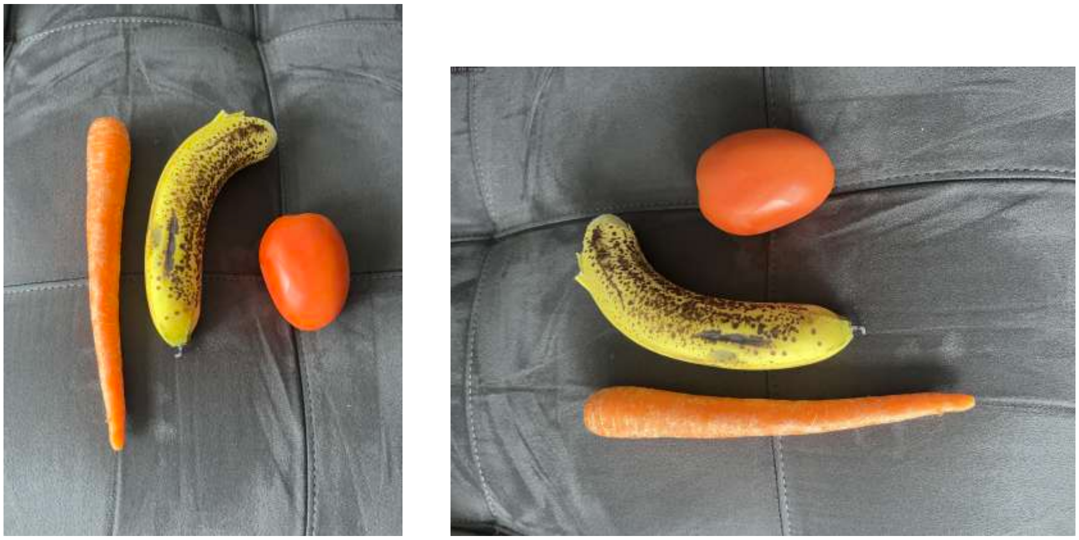


Fig: Fruits and Vegetables

Input: A carrot, a spotted banana, and a tomato placed on a couch.
Output: Classified as **“orange”** with **23.63% confidence**.
Comment: Partial misclassification. While none of the items were oranges, the dominant color may have influenced the prediction. Demonstrates the model's reliance on color features more than object structure in some cases.

Test Case 5:

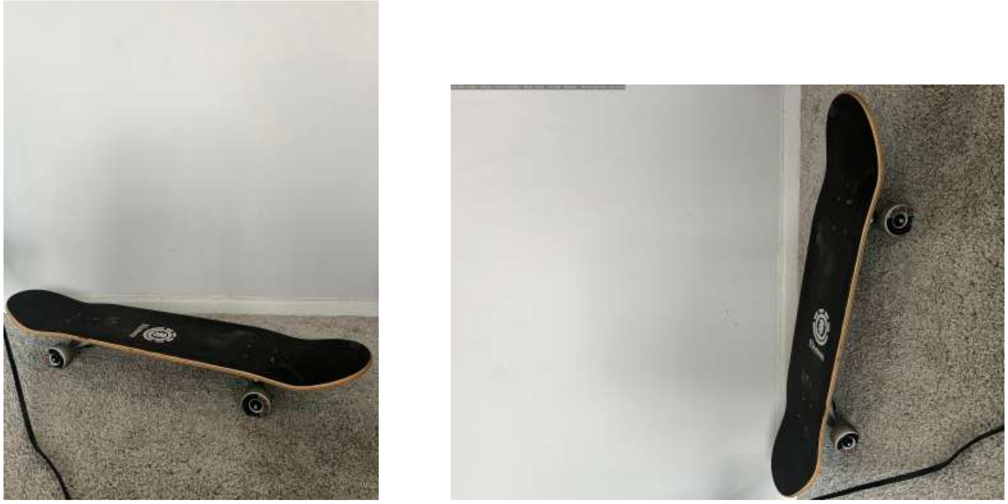


Fig: Skateboard

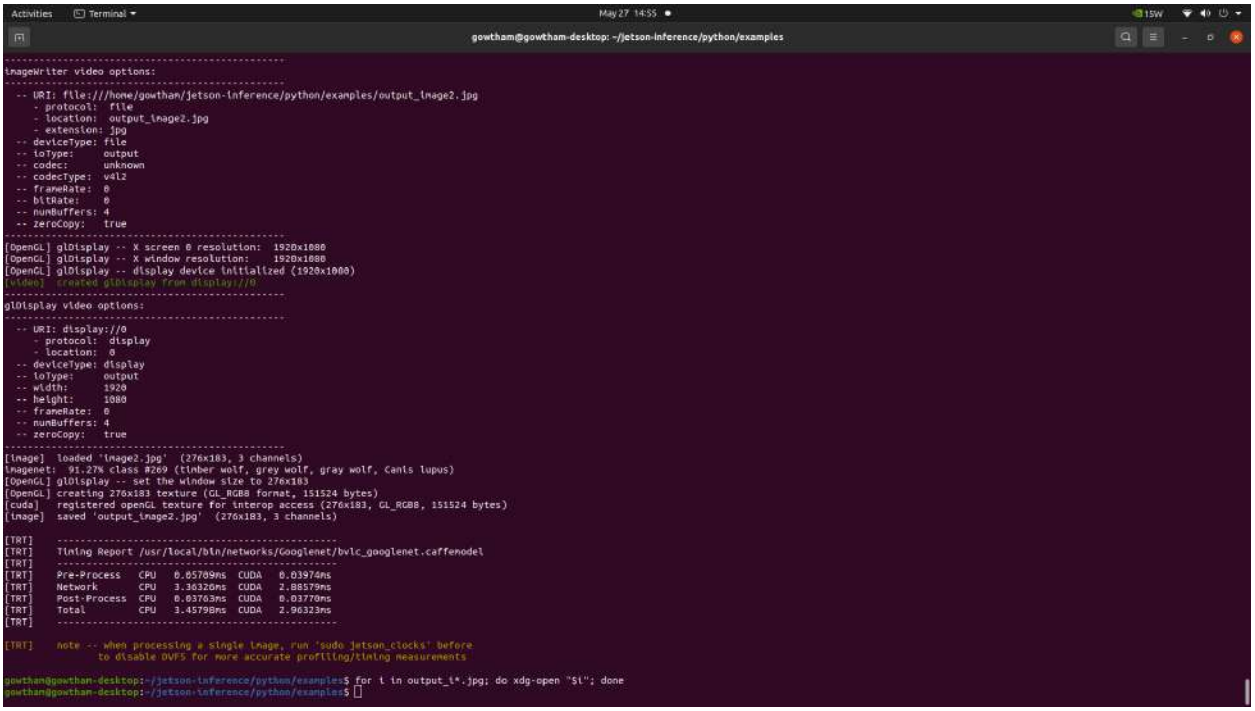
Input: A skateboard on a carpeted floor next to a wall.
Output: Classified as **“coho salmon, silver salmon”** with **14.53% confidence**.
Comment: Incorrect result. The skateboard was completely misclassified, likely due to shape and texture mismatch in the model’s training set. This highlights GoogLeNet’s limitations with unfamiliar objects.

Test Case 6:

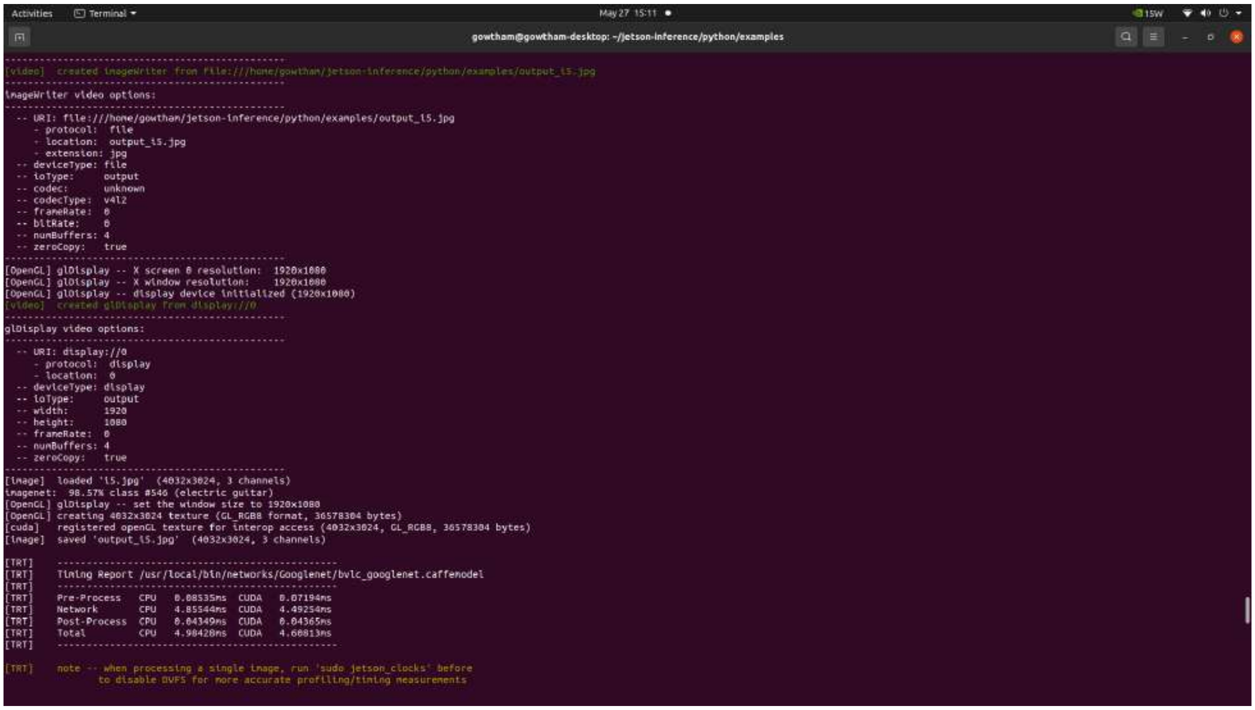
Fig: Friend

Input Image: A portrait photo of a person (Rajesh) sitting at a desk in a casual indoor setup.
Output Description: The model classified the image with **40.56% confidence** as "miniskirt, mini", which is a completely inaccurate result. This highlights a limitation of the model, it misclassifies images that contain people in natural poses, especially when their clothing doesn't match clear class labels in the training set. This test shows the importance of dataset diversity and the limits of generalization in pre-trained models.

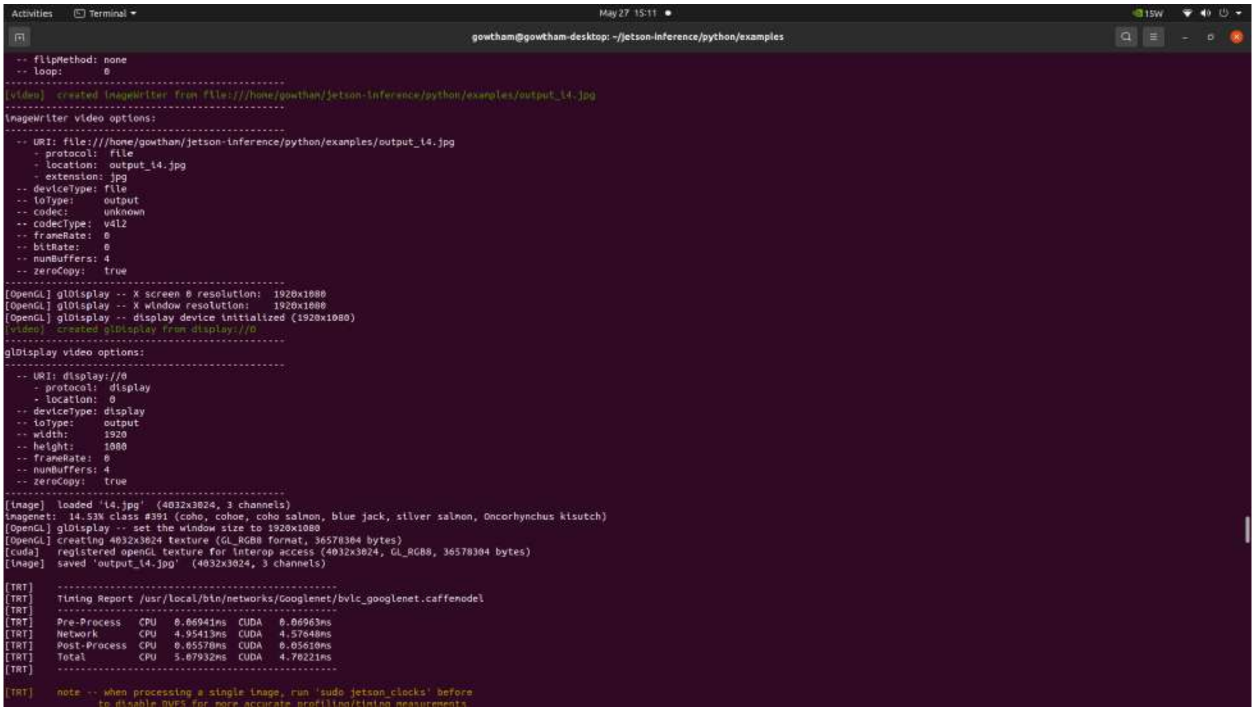
The following screenshots display the terminal output generated during image classification using the Jetson Orin Nano and the imagenet.py script. Each test case (from i1.jpg to i5.jpg) was processed using the GoogLeNet model. The outputs show the classification labels, confidence scores, image processing pipeline initialization, and detailed inference timing reports from both the CPU and CUDA. These logs confirm successful execution and demonstrate the internal workings of the model during inference.



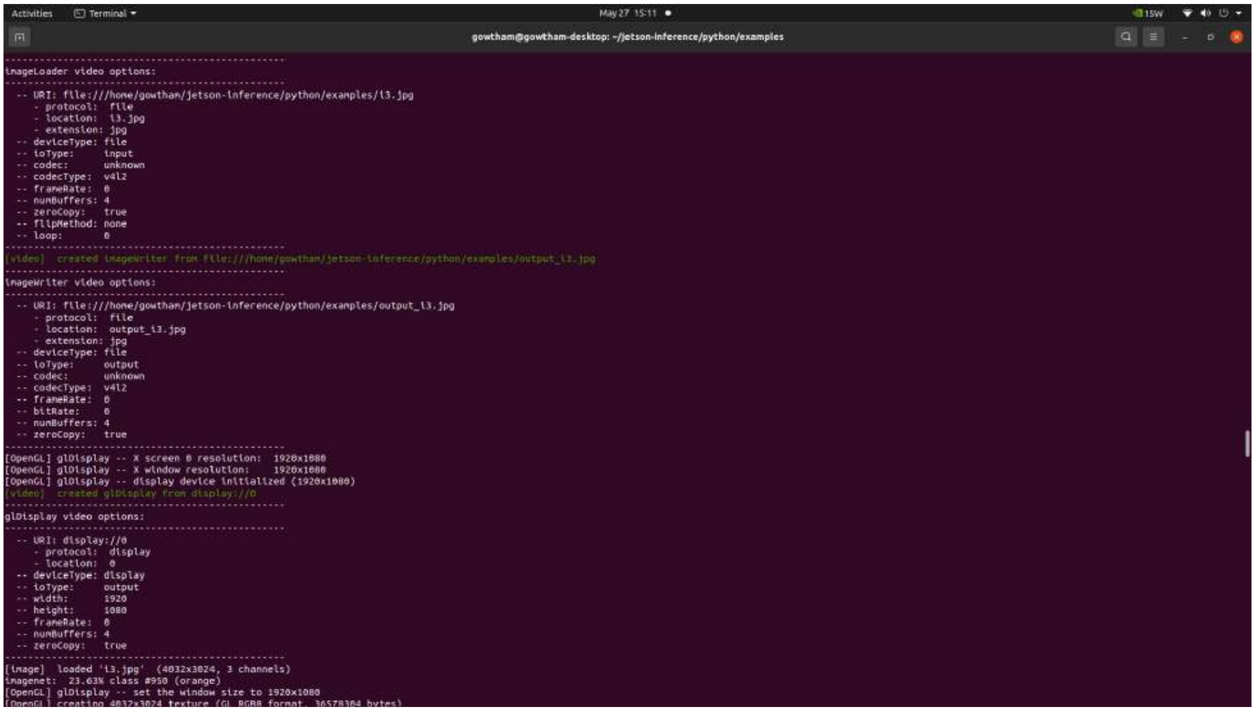
```
------------------------------------------------
imageWriter video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/output_image2.jpg
     - protocol:  file
     - location:  output_image2.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     output
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- bitRate:    0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[OpenGL] glDisplay -- X screen 0 resolution:  1920x1080
[OpenGL] glDisplay -- X window resolution:    1920x1080
[OpenGL] glDisplay -- display device initialized (1920x1080)
[video]  created glDisplay from display://0
------------------------------------------------
glDisplay video options:
------------------------------------------------
  -- URI: display://0
     - protocol:  display
     - location:  0
  -- deviceType: display
  -- ioType:     output
  -- width:      1920
  -- height:     1080
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[image]  loaded 'image2.jpg'  (276x183, 3 channels)
imagenet:  91.27% class #269 (timber wolf, grey wolf, gray wolf, Canis lupus)
[OpenGL] glDisplay -- set the window size to 276x183
[OpenGL] creating 276x183 texture (GL_RGB8 format, 151524 bytes)
[cuda]   registered openGL texture for interop access (276x183, GL_RGB8, 151524 bytes)
[image]  saved 'output_image2.jpg'  (276x183, 3 channels)

[TRT]    ------------------------------------------------
[TRT]    Timing Report /usr/local/bin/networks/Googlenet/bvlc_googlenet.caffemodel
[TRT]    ------------------------------------------------
[TRT]    Pre-Process   CPU   0.05709ms  CUDA   0.03974ms
[TRT]    Network       CPU   3.36326ms  CUDA   2.88579ms
[TRT]    Post-Process  CPU   0.03763ms  CUDA   0.03770ms
[TRT]    Total         CPU   3.45798ms  CUDA   2.96323ms
[TRT]    ------------------------------------------------

[TRT]    note -- when processing a single image, run 'sudo jetson_clocks' before
                to disable DVFS for more accurate profiling/timing measurements

gowtham@gowtham-desktop:~/jetson-inference/python/examples$ for i in output_i*.jpg; do xdg-open "$i"; done
gowtham@gowtham-desktop:~/jetson-inference/python/examples$ 
```



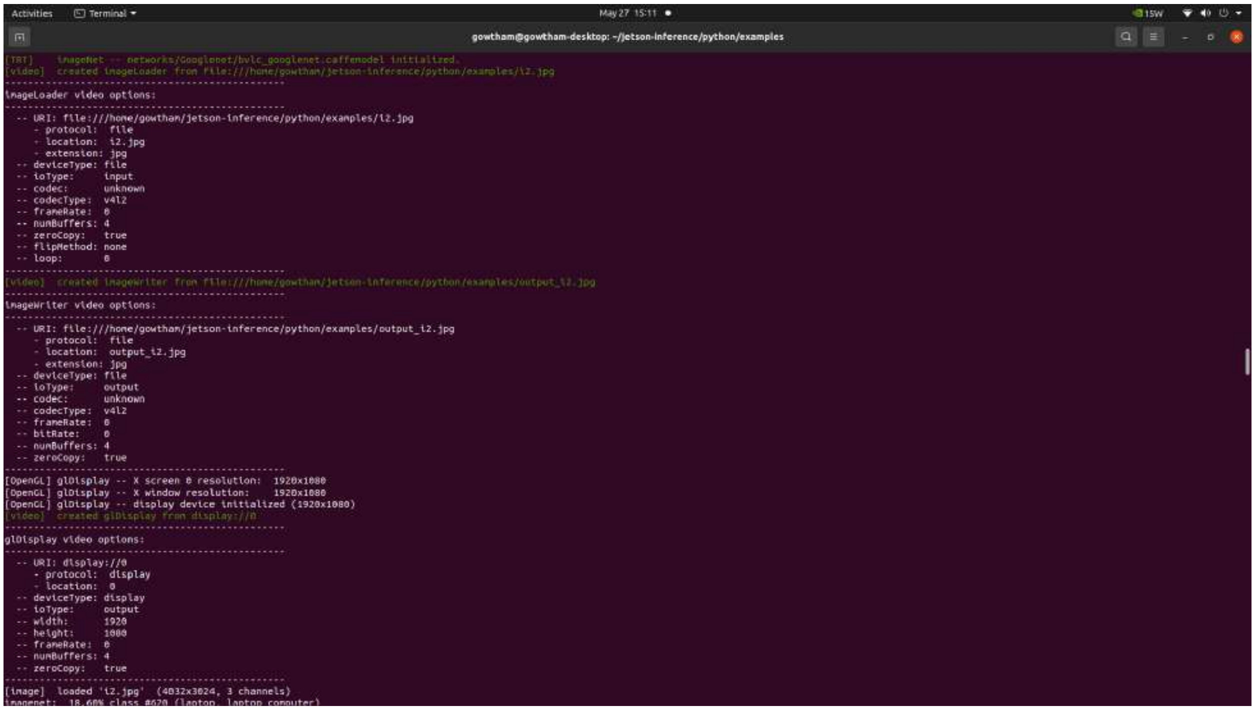
```
------------------------------------------------
[video]  created imageWriter from file:///home/gowtham/jetson-inference/python/examples/output_i5.jpg
------------------------------------------------
imageWriter video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/output_i5.jpg
     - protocol:  file
     - location:  output_i5.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     output
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- bitRate:    0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[OpenGL] glDisplay -- X screen 0 resolution:  1920x1080
[OpenGL] glDisplay -- X window resolution:    1920x1080
[OpenGL] glDisplay -- display device initialized (1920x1080)
[video]  created glDisplay from display://0
------------------------------------------------
glDisplay video options:
------------------------------------------------
  -- URI: display://0
     - protocol:  display
     - location:  0
  -- deviceType: display
  -- ioType:     output
  -- width:      1920
  -- height:     1080
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[image]  loaded 'i5.jpg'  (4032x3024, 3 channels)
imagenet:  98.57% class #546 (electric guitar)
[OpenGL] glDisplay -- set the window size to 1920x1080
[OpenGL] creating 4032x3024 texture (GL_RGB8 format, 36578304 bytes)
[cuda]   registered openGL texture for interop access (4032x3024, GL_RGB8, 36578304 bytes)
[image]  saved 'output_i5.jpg'  (4032x3024, 3 channels)

[TRT]    ------------------------------------------------
[TRT]    Timing Report /usr/local/bin/networks/Googlenet/bvlc_googlenet.caffemodel
[TRT]    ------------------------------------------------
[TRT]    Pre-Process   CPU   0.08535ms  CUDA   0.07194ms
[TRT]    Network       CPU   4.85544ms  CUDA   4.49254ms
[TRT]    Post-Process  CPU   0.04349ms  CUDA   0.04365ms
[TRT]    Total         CPU   4.98428ms  CUDA   4.60813ms
[TRT]    ------------------------------------------------

[TRT]    note -- when processing a single image, run 'sudo jetson_clocks' before
                to disable DVFS for more accurate profiling/timing measurements
```



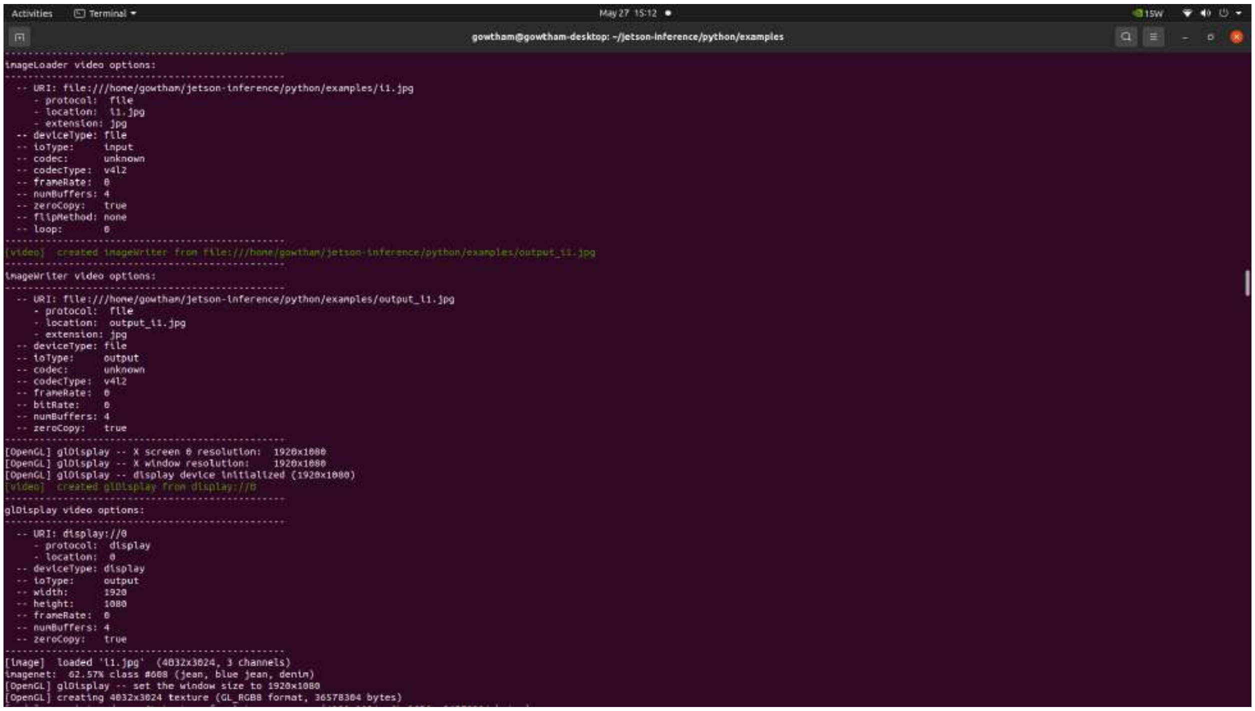
```
  -- flipMethod: none
  -- loop:       0
------------------------------------------------
[video]  created imageWriter from file:///home/gowtham/jetson-inference/python/examples/output_i4.jpg
------------------------------------------------
imageWriter video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/output_i4.jpg
     - protocol:  file
     - location:  output_i4.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     output
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- bitRate:    0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[OpenGL] glDisplay -- X screen 0 resolution:  1920x1080
[OpenGL] glDisplay -- X window resolution:    1920x1080
[OpenGL] glDisplay -- display device initialized (1920x1080)
[video]  created glDisplay from display://0
------------------------------------------------
glDisplay video options:
------------------------------------------------
  -- URI: display://0
     - protocol:  display
     - location:  0
  -- deviceType: display
  -- ioType:     output
  -- width:      1920
  -- height:     1080
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[image]  loaded 'i4.jpg'  (4032x3024, 3 channels)
imagenet:  14.53% class #391 (coho, cohoe, coho salmon, blue jack, silver salmon, Oncorhynchus kisutch)
[OpenGL] glDisplay -- set the window size to 1920x1080
[OpenGL] creating 4032x3024 texture (GL_RGB8 format, 36578304 bytes)
[cuda]   registered openGL texture for interop access (4032x3024, GL_RGB8, 36578304 bytes)
[image]  saved 'output_i4.jpg'  (4032x3024, 3 channels)

[TRT]    ------------------------------------------------
[TRT]    Timing Report /usr/local/bin/networks/Googlenet/bvlc_googlenet.caffemodel
[TRT]    ------------------------------------------------
[TRT]    Pre-Process   CPU   0.06941ms  CUDA   0.06963ms
[TRT]    Network       CPU   4.95413ms  CUDA   4.57648ms
[TRT]    Post-Process  CPU   0.05578ms  CUDA   0.05610ms
[TRT]    Total         CPU   5.07932ms  CUDA   4.70221ms
[TRT]    ------------------------------------------------

[TRT]    note -- when processing a single image, run 'sudo jetson_clocks' before
                 to disable DVFS for more accurate profiling/timing measurements
```



```
------------------------------------------------
imageLoader video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/i3.jpg
     - protocol:  file
     - location:  i3.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     input
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
  -- flipMethod: none
  -- loop:       0
------------------------------------------------
[video]  created imageWriter from file:///home/gowtham/jetson-inference/python/examples/output_i3.jpg
------------------------------------------------
imageWriter video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/output_i3.jpg
     - protocol:  file
     - location:  output_i3.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     output
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- bitRate:    0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[OpenGL] glDisplay -- X screen 0 resolution:  1920x1080
[OpenGL] glDisplay -- X window resolution:    1920x1080
[OpenGL] glDisplay -- display device initialized (1920x1080)
[video]  created glDisplay from display://0
------------------------------------------------
glDisplay video options:
------------------------------------------------
  -- URI: display://0
     - protocol:  display
     - location:  0
  -- deviceType: display
  -- ioType:     output
  -- width:      1920
  -- height:     1080
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[image]  loaded 'i3.jpg'  (4032x3024, 3 channels)
imagenet:  23.63% class #950 (orange)
[OpenGL] glDisplay -- set the window size to 1920x1080
[OpenGL] creating 4032x3024 texture (GL_RGB8 format, 36578304 bytes)
```



```
[TRT]    imageNet -- networks/Googlenet/bvlc_googlenet.caffemodel initialized.
[video]  created imageLoader from file:///home/gowtham/jetson-inference/python/examples/i2.jpg
------------------------------------------------
imageLoader video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/i2.jpg
     - protocol:  file
     - location:  i2.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     input
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
  -- flipMethod: none
  -- loop:       0
------------------------------------------------
[video]  created imageWriter from file:///home/gowtham/jetson-inference/python/examples/output_i2.jpg
------------------------------------------------
imageWriter video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/output_i2.jpg
     - protocol:  file
     - location:  output_i2.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     output
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- bitRate:    0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[OpenGL] glDisplay -- X screen 0 resolution:  1920x1080
[OpenGL] glDisplay -- X window resolution:    1920x1080
[OpenGL] glDisplay -- display device initialized (1920x1080)
[video]  created glDisplay from display://0
------------------------------------------------
glDisplay video options:
------------------------------------------------
  -- URI: display://0
     - protocol:  display
     - location:  0
  -- deviceType: display
  -- ioType:     output
  -- width:      1920
  -- height:     1080
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[image]  loaded 'i2.jpg'  (4032x3024, 3 channels)
imagenet:  18.60% class #620 (laptop, laptop computer)
```



```
------------------------------------------------
imageLoader video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/i1.jpg
     - protocol:  file
     - location:  i1.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     input
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
  -- flipMethod: none
  -- loop:       0
------------------------------------------------
[video]  created imageWriter from file:///home/gowtham/jetson-inference/python/examples/output_i1.jpg
------------------------------------------------
imageWriter video options:
------------------------------------------------
  -- URI: file:///home/gowtham/jetson-inference/python/examples/output_i1.jpg
     - protocol:  file
     - location:  output_i1.jpg
     - extension: jpg
  -- deviceType: file
  -- ioType:     output
  -- codec:      unknown
  -- codecType:  v4l2
  -- frameRate:  0
  -- bitRate:    0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[OpenGL] glDisplay -- X screen 0 resolution:  1920x1080
[OpenGL] glDisplay -- X window resolution:    1920x1080
[OpenGL] glDisplay -- display device initialized (1920x1080)
[video]  created glDisplay from display://0
------------------------------------------------
glDisplay video options:
------------------------------------------------
  -- URI: display://0
     - protocol:  display
     - location:  0
  -- deviceType: display
  -- ioType:     output
  -- width:      1920
  -- height:     1080
  -- frameRate:  0
  -- numBuffers: 4
  -- zeroCopy:   true
------------------------------------------------
[image]  loaded 'i1.jpg'  (4032x3024, 3 channels)
imagenet:  62.57% class #608 (jean, blue jean, denim)
[OpenGL] glDisplay -- set the window size to 1920x1080
[OpenGL] creating 4032x3024 texture (GL_RGB8 format, 36578304 bytes)
```