

# Image Recognition Application

Haoxuan Sun

Github Link: [https://github.com/luciusluo/cs655\\_mini\\_project](https://github.com/luciusluo/cs655_mini_project)

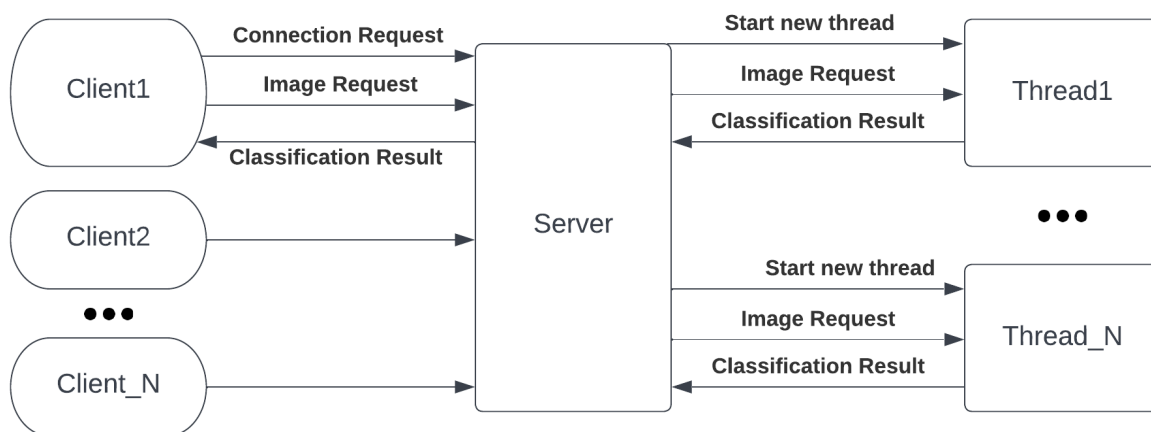
GENI slice: YuxiaoMiniProject

## 1. Introduction / Problem Statement

The project is to implement an image-recognition service. A user is able to submit an image query and the service uses AlexNet, a image recognition technique to classify the images, and returns the classification outcome to the user. The client and the recognition systems (server) are on separate nodes, and they are connected using TCP and socket programming.

## 2. Experimental Methodology

User (Client) first sends a connection request and sets up a connection with the server and then submit an image to the server. After that, the data of the image is received at server. Then, a new thread is created to start image classification work using the image data just received. At last, the classification result is returned to the client. The architecture diagram is shown below.



The project is implemented in Python. Python v3.9 is used in server and client uses Python v3.6. Up to 32 requests can be handled in parallel.

# Environment Setup on GENI NODE

On server node:

'''

Used python version == 3.9.16

PIL version == 9.3.0

'''

sudo apt-get install python3.9

sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.9 2

git clone https://github.com/luciusluo/cs655\_mini\_project

sudo apt update

sudo apt install python3-pip

sudo apt install python3.9-distutils

pip3 install --upgrade setuptools

pip3 install --upgrade pip

sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev

libreadline-dev libffi-dev curl software-properties-common

python3 -m pip install --upgrade Pillow

pip3 install torch torchvision torchaudio --extra-index-url <https://download.pytorch.org/whl/cpu>

On client node:

'''

Used python version == 3.6.9

PIL version == 8.4.0

'''

git clone https://github.com/luciusluo/cs655\_mini\_project

sudo apt update

sudo apt install python3-pip

python3 -m pip install --upgrade pip

pip3 install --upgrade setuptools

sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev

libreadline-dev libffi-dev curl software-properties-common

python3 -m pip install --upgrade Pillow

pip3 install torch torchvision torchaudio --extra-index-url <https://download.pytorch.org/whl/cpu>

## 3. Results

### Usage Instructions

1. Create a slice in geni and use our given xml to retrieve *rspec file*.
2. Connect to an instaGeni and login into the server and client nodes.
3. Follow the environment setup above
4. Make sure correct version of python and package are installed
5. Cd into the cs655\_mini\_project folder

6. In server node, input `python3 server.py <port number>`, which will print out server name
7. In client node, input `python3 client.py <server name> <port number>`
8. Follow the command line to upload file and check prediction

### Example client

```
[yxlue@client5:~/cs655_mini_project$ python3 client.py server1.yuxiaominiproject.]
ch-geni-net.instageni.sox.net 5000
----- Starting Client Socket -----

----- Available images to send are: -----
['hen.jpeg', 'doberman.jpeg', 'dog.jpeg', 'ladybug.jpeg', 'malamute.jpeg', 'jellyfish.jpeg', 'hamster.jpeg', 'chihuahua.jpeg', 'kite.jpeg', 'germansheperd.jpeg', 'quail.jpeg', 'pomeranian.jpeg', 'tiger.jpeg', 'cat.jpeg', 'otter.jpeg', 'frog.jpeg', 'koala.jpeg', 'coyote.jpeg']
Please choose an image to send from the above options:
germansheperd.jpeg
RTT in ms: 1673
File Size is : 9463 bytes
Throughput is: 45250.4482964734 bps
Received image! The best prediction is: German shepherd
----- Available images to send are: -----
['hen.jpeg', 'doberman.jpeg', 'dog.jpeg', 'ladybug.jpeg', 'malamute.jpeg', 'jellyfish.jpeg', 'hamster.jpeg', 'chihuahua.jpeg', 'kite.jpeg', 'germansheperd.jpeg', 'quail.jpeg', 'pomeranian.jpeg', 'tiger.jpeg', 'cat.jpeg', 'otter.jpeg', 'frog.jpeg', 'koala.jpeg', 'coyote.jpeg']
Please choose an image to send from the above options:
```

### example server

```
[yxlue@server1:~/cs655_mini_project$ python3 server.py 5000]
/users/yxlue/.local/lib/python3.9/site-packages/torchvision/models/_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.
  warnings.warn(
/users/yxlue/.local/lib/python3.9/site-packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a weight enum or 'None' for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing 'weights=AlexNet_Weights.IMAGENET1K_V1'. You can also use 'weights=AlexNet_Weights.DEFAULT' to get the most up-to-date weights.
  warnings.warn(msg)
server1.yuxiaominiproject.ch-geni-net.instageni.sox.net
Server listening on port 5000.

Server connected to ('172.17.4.2', 36908)
Receive an image from Client ('172.17.4.2', 36908)
[('German shepherd', 85.24147033691406), ('malinois', 14.47010612487793), ('kelpie', 0.19932177662849426), ('Great Dane', 0.024694712832570076), ('Ibizan hound', 0.019159331917762756)]
```

## Analysis

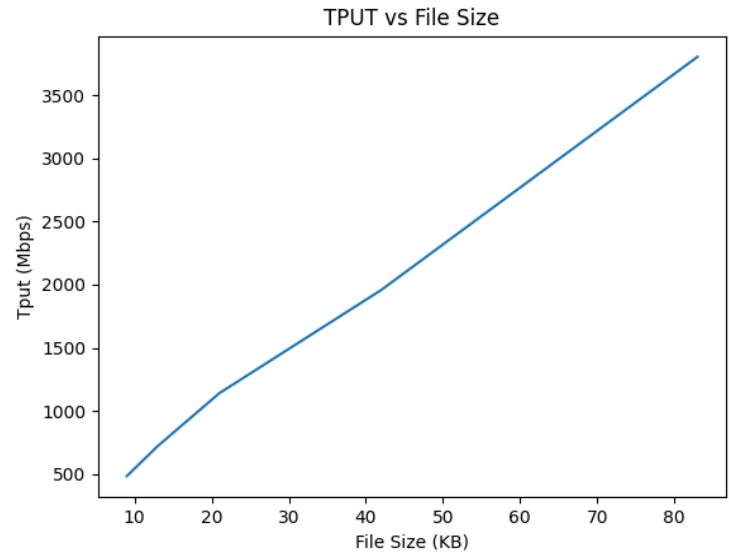
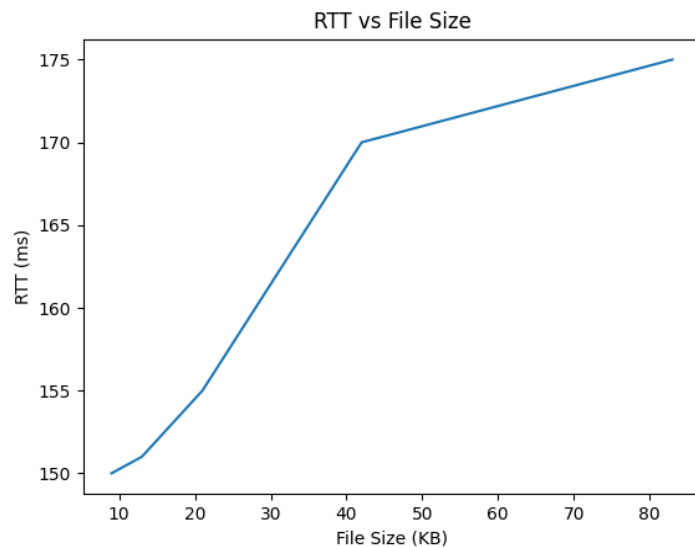
RTT and TPUT on client when there's 1 client connected to the server

Image name	German	hen	tiger	cat	frog
------------	--------	-----	-------	-----	------

	sheperd				
Image Size	9kB	13kB	21kB	42kB	83kB
RTT	150ms	151ms	155ms	170ms	175ms
TPut	485.282051 Kbps	722.416107 Kbps	1.14170667 Mbps	1.95816471 Mbps	3.80225287 Mbps

Figure of RTT v.s. File Size

Figure of Throughput v.s. File Size



## 4. Conclusion

Demo video (Please use BU email to access):

[https://drive.google.com/file/d/1MSCvKy5vyAGbS1kUPLNp9o0maNfqV\\_CI/view?usp=share\\_link](https://drive.google.com/file/d/1MSCvKy5vyAGbS1kUPLNp9o0maNfqV_CI/view?usp=share_link)

Based on our experiments, we found that the RTT and throughput between client and server increases proportionally to the size of images transmitted.

Possible Extensions: Our program could be further optimized with a convenient web interface which enables convenient and faster image file upload and prediction review. Prediction accuracy could be improved by using more complicated deep learning models such as VGGNet or ResNet.