

CS655 Geni Mini Project Proposal - Image Recognition Application

Introduction

In this project, we form a group of members of 4, including Wei Jiang, Kaihong Wang, Ruikang Wang, Nianyi Zhang. We choose the “Image Recognition Application” as our topic.

In this project, we need to implement an image-recognition program that runs on two nodes separately. One of the nodes should take care of the image submitted from a client and query the other node which provides an interface that could be used to recognize an image and return the result.

Problem Definition

Concretely, the requirements mentioned above could be tackled by breaking down the tasks into three parts: frontend image receiving and querying, communication between frontend and backend and finally image recognition at the backend.

As a consequence, the whole process of the system consists of: A frontend system receives an image from a client and sends the image to the backend system. The backend will then receive the image, apply the image recognition system and finally return the result back to the frontend. Finally the frontend system will display the result to the client.

Detailed Requirements

- **Frontend**

The task of the frontend is to receive the image uploaded by the customer and transmit it to the backend. We will regard a node in Geni as our frontend and use CGI to complete the functions of the

webpage section. Then we will transmit the image using socket programming to the backend. Furthermore, the frontend also receive the results of image recognition from backend, and shows it on the webpage.

- Backend

Backend will achieve comprehension with frontend using socket programming or REST API in the first place. Once retrieving target image or image array from frontend, we will utilize image recognition algorithm interface to process and categorize corresponding images. In addition, final labelled results will be transferred to the client node and displayed on the web interface.

- Image recognition subroutine

The image recognition subroutine should be called when the backend system receives the image. After the image is passed in, the program should preprocess the image and input it into a SqueezeNet model pretrained on ImageNet, which is implemented using pyTorch. Finally, the recognition result vector will be output at the end of the model, converted into the corresponding class name and sent to the program that called this subroutine.