**Image Recognition Application**

Boyi Song(TODO), Jingyu Su(TODO), Yi You(TODO), Xinkung Yin (U44255956)

***GitHub link:*** <https://github.com/troyyxk/image_recognition_application>

***Video demo:*** <TODO>

***Project on GINI and any public link:***

**1. Introduction / Problem Statement**

For this project, our group is creating an online image recognition application. The application is full stack from the UI and frontend to bankend service and computational nodes. Users upload image from a webpage we made with React.JS and the image is sent to backend with Axios.JS client. A Django backend running on a GENI node will handle the requests from frontend. Celery task queue running on the backend node will oversee distributing the image to computational nodes. Google net that are running on the computational nodes will take in images and return expected result. User on the frontend will only need to refresh the page to get the result.

During the implementation of the experiment, our team study and practiced frontend webpage programming, socket programming with JS libraries, python backend framework, task queue implementation, distributed system and neural network deployment.

**2. Experimental Methodology**

2.1 Architecture Diagrams

<TODO GENI node screenshot>

<TODO Overview architecture layout diagram>

<TODO description of each part>

2.2 Experimental Methodology

<TODO assumptions>

**3. Results**

3.1 Usage Instruction

Enter the URL: <TODO> and you should be able to see this page:

Text

Description automatically generated

Go to the “Submit” tab, choose an image to upload and submit the image:

Graphical user interface, application

Description automatically generated

Now to go the “Review” tab and refresh using the “Refresh” button, you should see the result.

Graphical user interface, text, application, website

Description automatically generated

3.2 Analysis

**4. Conclusion**

**5. Division of Labor**