## Description

This application is designed to train a convolutional neural network to identify whether an image is a dog or cat in distributed mode. It contains a master server and 2 child servers. An interactive web app is designed in Streamlit to upload images and for visualisation. The images need to be uploaded in JPEG or PNG formats. The name of image file should contain either cat or dog in it for proper training The servers are implemented using Flask framework. The master server on initialisation sends a global copy of a CNN model to the child servers. When an image is uploaded to either one of the servers the child server trains the model on this image. It updates the weights. The updated weights are sent to the master server. The weights are then aggregated to get a new global model. This distributed training allows for privacy of images. The images are not shared to a common server and are stored locally. Only the weights are sent to the master server for processing.

## Installation

This project requires Python and a host of Python libraries installed

If you have Python installed but not the libraries, you can easily install them using

```
pip install -r requirements.txt
```

## Usage

To use the application please execute the following commands in the order given below

- python masterserver.py
- python childserver1.py
- python childserver2.py
- streamlit run ./webapp.py

## Code Contributors

- Sreerag Chandran(2022ac05086)
- M V Deepak (2022aa05071)
- Masilamani D (2022ac05355)
- Nishi Chandra (2022ac05351)
- Trilok Sachin Chittala(2022ac05072)

## Choice of Platform and Parameters

Flask – Create a backend server app
Streamlit – Create a frontend UI
CNN Model- 10 Layers Model for binary classification of cats and dogs
Optimiser – Adam
Loss Function - binary_crossentropy
Metrics- Accuracy

Communication of Servers – REST API POST and GET Methods

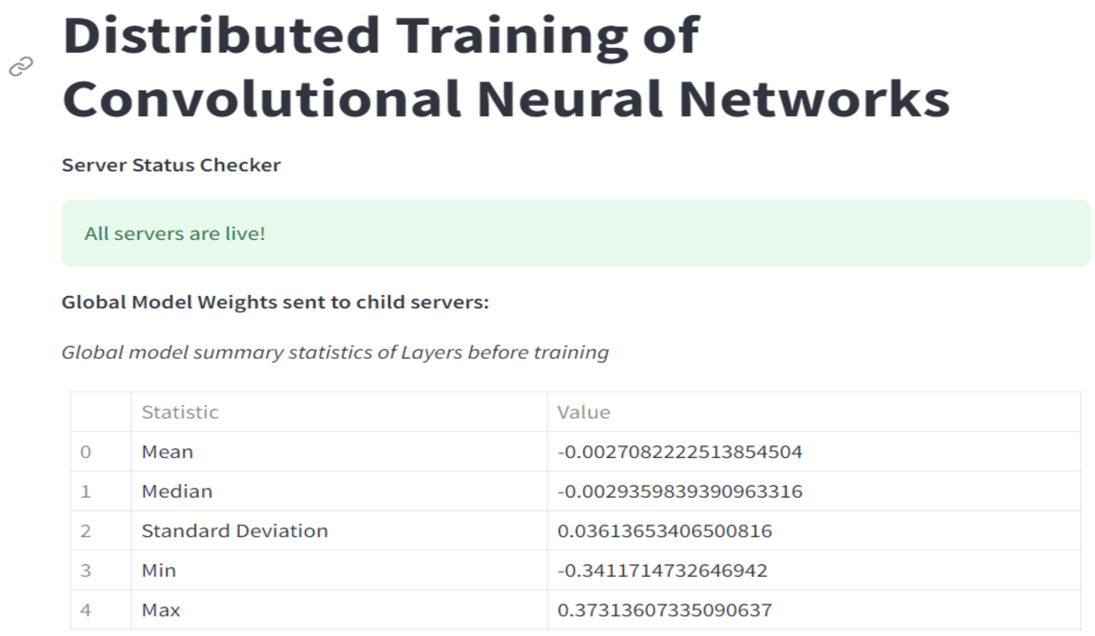
Running Time – Single Iteration of Training takes 1 min

Performance – Dependent on REST API connectivity

Child Server 1: Accuracy: 100% Loss : 0.009770618751645088
Child Server 2: Accuracy: 100% Loss : 4.732050001621246e-05

High accuracy due to low number of training samples

Screenshots



Distributed Training of Convolutional Neural Networks

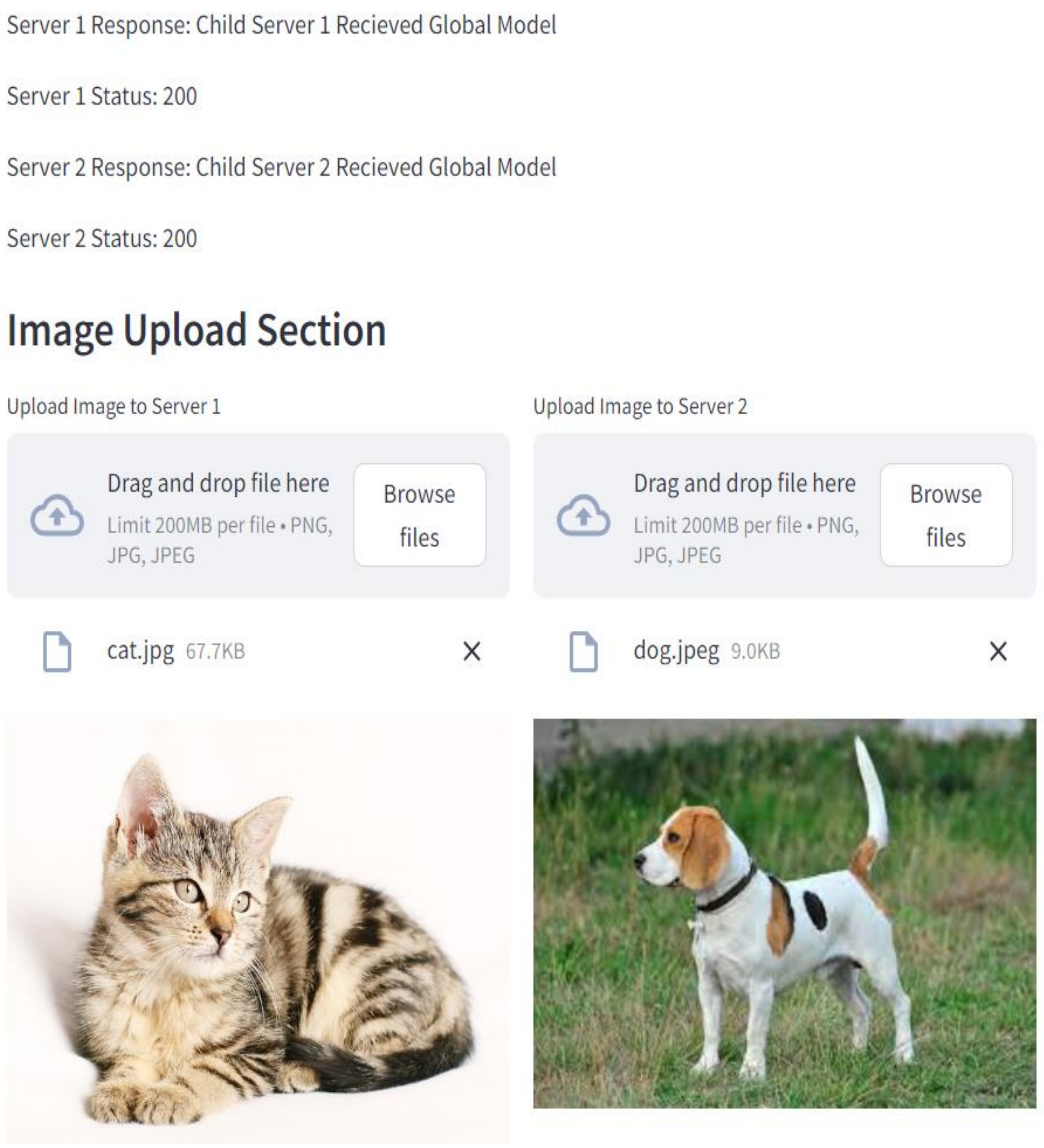
Server Status Checker

All servers are live!

Global Model Weights sent to child servers:

Global model summary statistics of Layers before training

	Statistic	Value
0	Mean	-0.0027082222513854504
1	Median	-0.0029359839390963316
2	Standard Deviation	0.03613653406500816
3	Min	-0.3411714732646942
4	Max	0.37313607335090637



Server 1 Response: Child Server 1 Recieved Global Model

Server 1 Status: 200

Server 2 Response: Child Server 2 Recieved Global Model

Server 2 Status: 200

Image Upload Section

Upload Image to Server 1

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

cat.jpg 67.7KB

Upload Image to Server 2

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

dog.jpeg 9.0KB

Process

Logs

```
Response from server 1: {"message":"Image received"} Input image size : (100, 100, 3)

Response from server 2: {"message":"Image received"} Input image size : (100, 100, 3)

Finished training in Server 1 Number of epoch 3 , Accuracy: [1.0], Loss :[0.003424697322770953]

Finished training in Server 2 Number of epoch 3 , Accuracy: [1.0], Loss :[0.3559414744377136]

Logs From Master Server:

Global model summary statistics of Layers after training
```

	Statistic	Value
0	Mean	-0.0027056022081524134
1	Median	-0.0029354162979871035
2	Standard Deviation	0.036137524992227554
3	Min	-0.3412221670150757
4	Max	0.37313607335090637