APPORTION HEAVY WORKS TO THE OTHERS

Trial and Error, Efforts, and the Next Step

Mingyu Zhao

February 2 2018

Smart Mobile Networking and Computing Lab Workshop

Contents

I. Background

Why do I need this framework?

A Simple Scenario
- A group of people go to travel.
- One of them want to perform
  a Deep Learning task.

Assumptions (based on experiences)
- There always exists idle devices.
- There is always no Wi-Fi outside.

Fig. 1.1 Applications of CNNs in Our Life

I. Background

Problems of the cloud

- User's privacy
- Network Latency
- Traffic Acquisition Costs (without free Wi-Fi)

Benefits of our framework

- Make the use of idle devices
- Low latency
- No need of a public Wi-Fi

Fig. 1.2 Cloud Computing

II. Algorithm

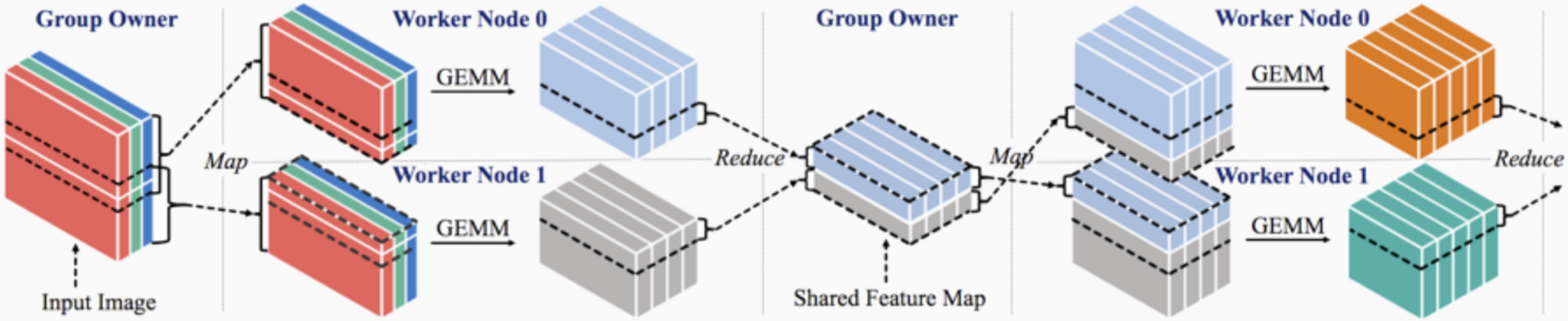

Fig. 2.1 The Map-Reduce Algorithm Mentioned in the paper of Jiachen Mao, et.al. Local Distributed Mobile Computing System For Deep Neural Networks

II. Algorithm

Group Owner

1. Read the input tensor.
2. Divide the input into several parts.
3. Send divided data to workers.
4. Collect the result from all workers.
5. If there is no more layers
   - Calculate the final result.

   Else
   - Send related data to workers.
   - Return to step 4.

Worker

1. Read CNN layer models.
2. Wait for divided input data from the GO.
3. Perform convolution operation.
4. If there is no more layers
   - Return all result to the GO.

   Else
   - Return padding lines of data to the GO.
   - Wait for updated data from the GO.
   - Update related lines.
   - Return to step 3.

III. Challenges

Version 1: Android Implement (CNNdroid)

- Debug Problem
- Socket:
  - Many threads
    - UI thread
    - SocketServerListener thread
    - SocketServer thread
    - ...
  - Complicated Thread Management
- RenderScript

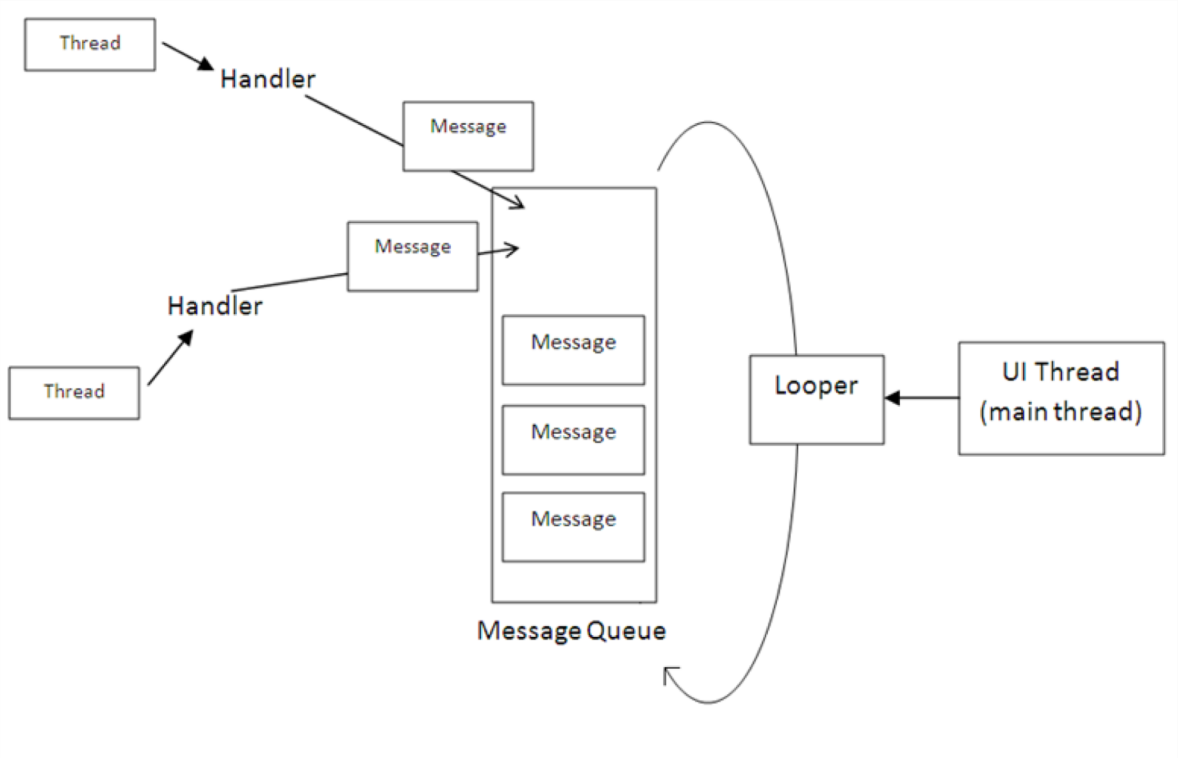

Fig. 3.1 Android's Thread Management

III. Challenges

Version 2: Web Implement

- Data Transmit Problem
- Performance of JavaScript
- Limitation of JavaScript
  - e.g Read pixels from a image

Benefits of Version 2

- Heterogeneous Devices Support
- GPU Usage (WebGL)
- Advanced Debuging Tools
- Easy to Management Groups

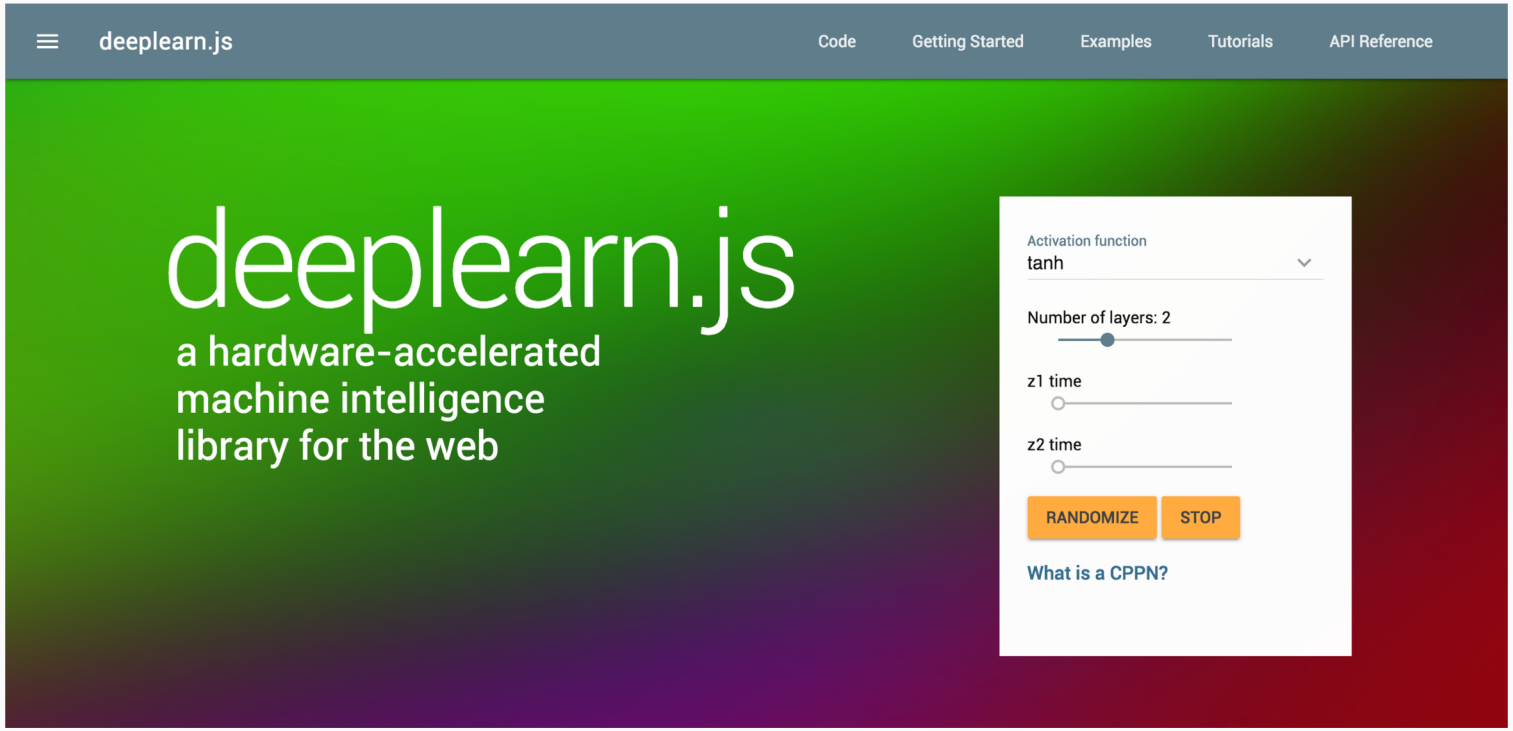

Fig. 3.2 DeepLearn.js Website

IV. Framework Design

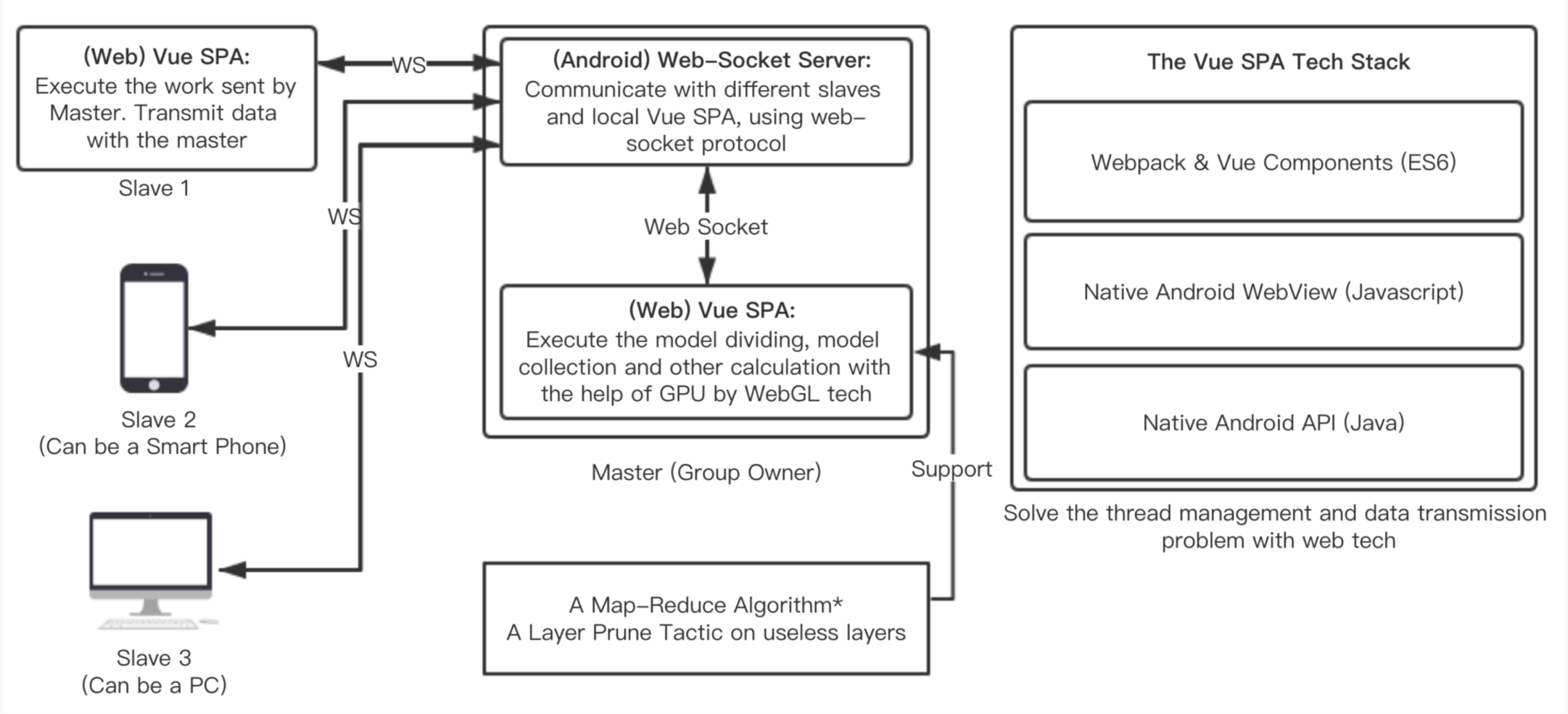

Fig. 4.1 Overview

IV. Framework Design

DCNN Master Vue

- /: local inference
- /multi: co-inference
- /slave: the page for worker nodes

HTTP Server

- Run in an Android service
- Provide HTML, JS,CSS, etc
- Provide CNN model
- Provide test input images

WebSocket Server

- Run in an Android service
- Forward WebSocket messages
- Maintain the WebSocket Connection

Why WebSocket?

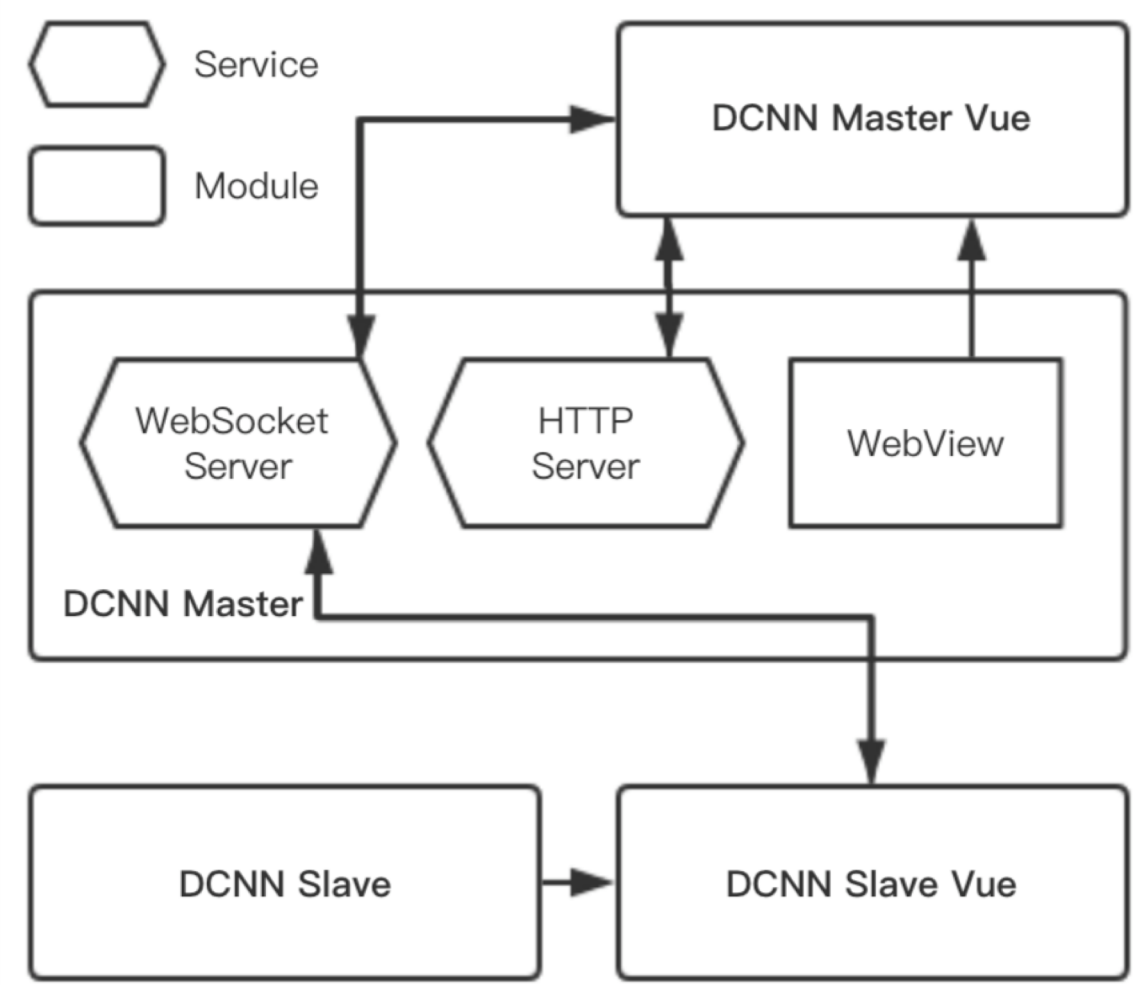

Fig. 4.2 Project Structure

Data Transmit Format

```json
{
  "func": "function name",
  "data": {}
}
```

Fig. 4.3 Basic Format

```json
{
  "func": "send2node",
  "data": {
    targetId: nodeId,
    data: {}
  }
}
```

Fig. 4.4 Send to a certain node

```json
{
  "func": "reduce",
  "data": {
    "sourceId": nodeID,
    "result": tensor1D,
    "layer": "Conv1"
  }
}
```

Fig. 4.5 Workers Send Back Data

V. Cifar-10 Demo

Cifar-10 CNN Model

- Contain 2 conv-layers.
- Remove norm layers.
- Use TensorFlow to train.
- Use Deeplearn.js to infer.
- 90% accuracy

Layer Name	Description
conv1	convolution and rectified linear activation.
pool1	max pooling.
norm1	local response normalization.
conv2	convolution and rectified linear activation.
norm2	local response normalization.
pool2	max pooling.
local3	fully connected layer with rectified linear activation.
local4	fully connected layer with rectified linear activation.
softmax_linear	linear transformation to produce logits.

Table 5.1 CNN structure

V. Cifar-10 Demo

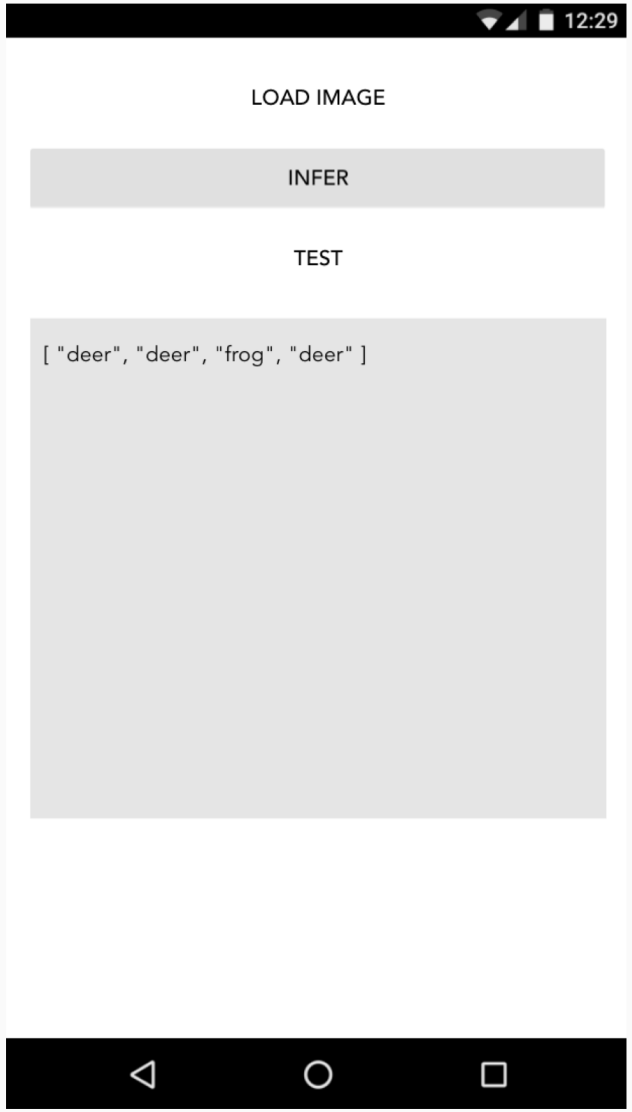

Fig. 5.1 localhost:8080/master

Fig. 5.2 localhost:8080/multi

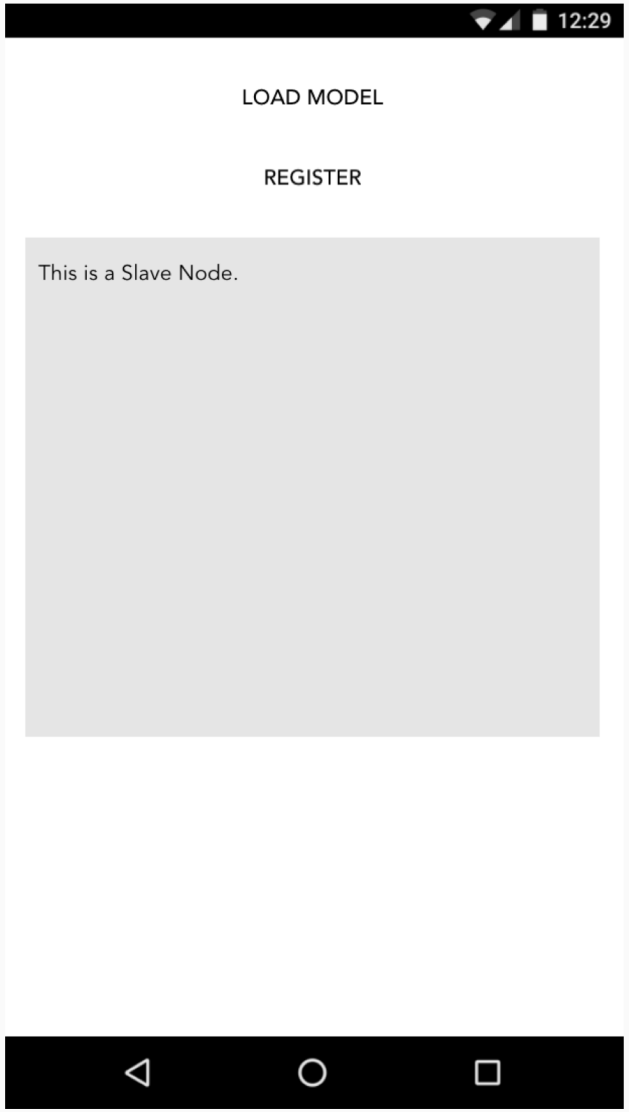

Fig. 5.3 localhost:8080/slave

V. Cifar-10 Demo

Operation Statistic

- More Experiments Required

	MBP Chrome	Nexus5 Chrome	iPhone7 Safari	2 * Nexus5
Loading Images Time	0.059	0.103	0.082	0.105
Inference Time	0.636	1.230	0.809	1.032

Table 5.2 Operation Time(s)

VI. The Next Step

- Perform more Experiments.
- Analyze more statistic data.
- Implement the AlexNet Demo —— a more complicated CNN model.
- Design the Regression Model.
- Improve Performance.

Thank You