



Application for Summer Visiting Student

- Applicant: Zhong Guan
- Lab: [Terradynamic Lab](#) @ JHU
- Supervisor: Prof. Chen Li

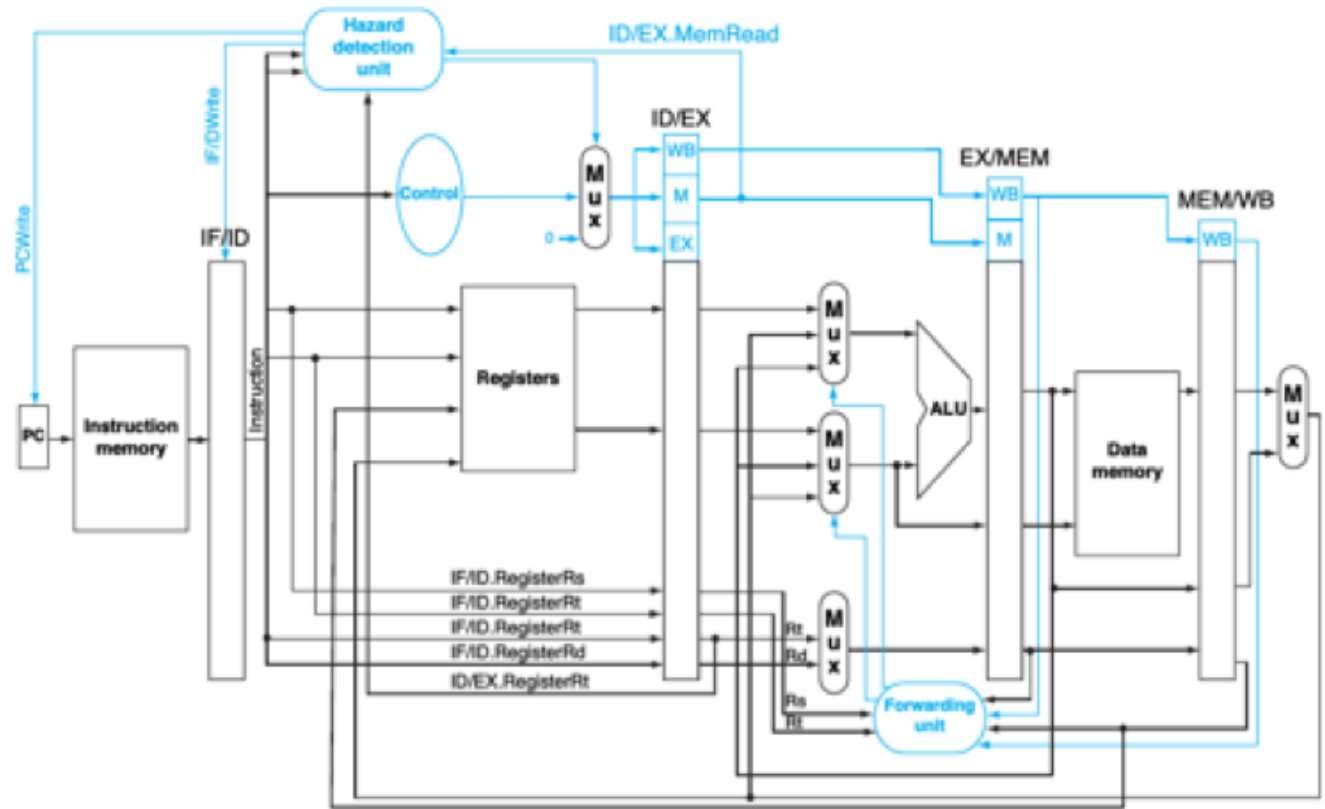


- Pipelined Processor
- Android-Based Mecanum Wheel Robot Car
- Shopping Cart Database Project

Project Experience

Project Experience

Pipelined Processor



Why Pipelined Processor?

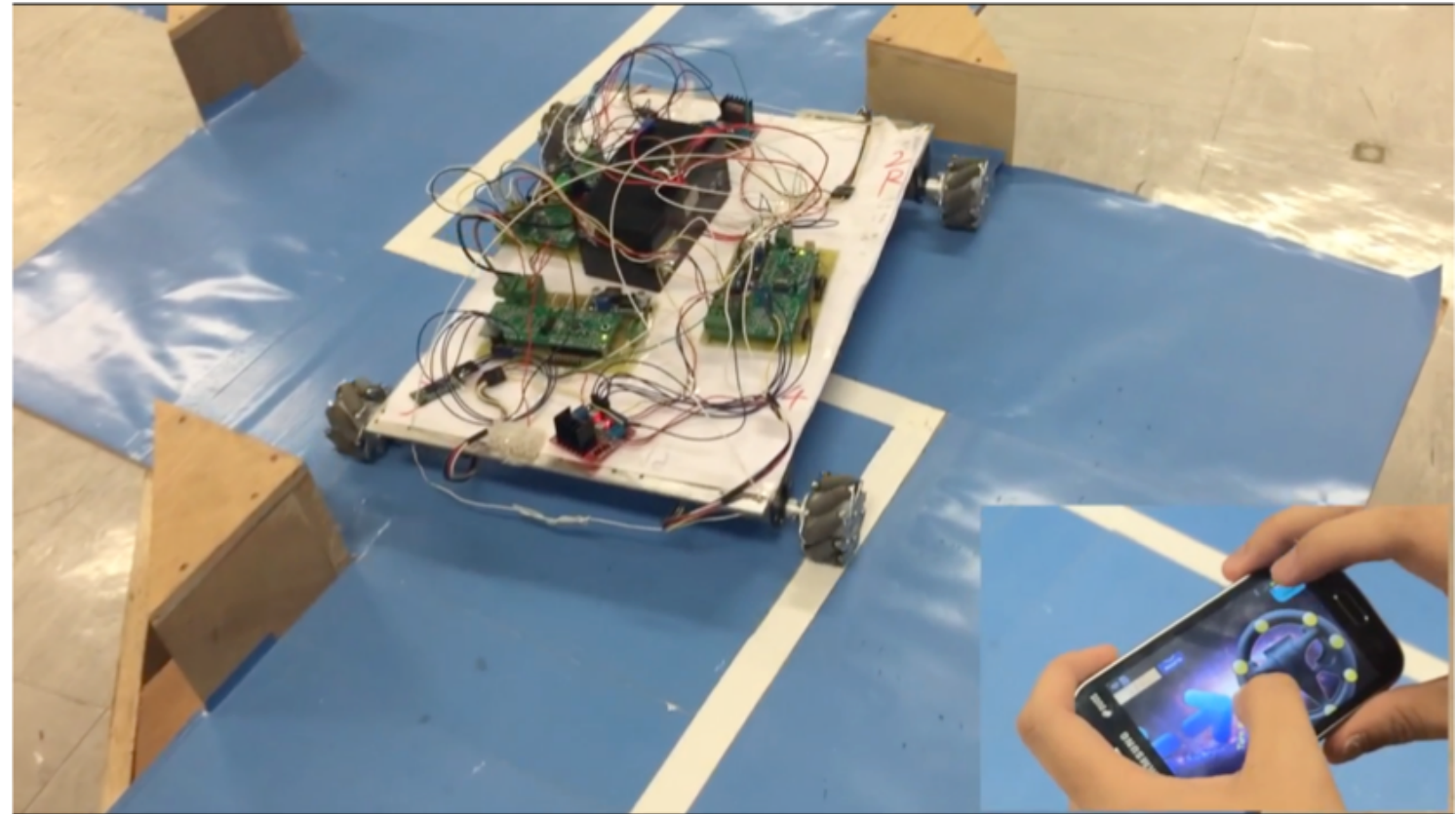
- Improve performance by performing many tasks simultaneously.
- Hardware level: Multiple memory banks.
- Instruction level: Pipelining and instruction level parallelism exploitation.
- Thread and data level: Multiprocessors.
- Pipelined processors were designed to increase the overall CPU throughput by using instruction level parallelism.

Optimizing Processor

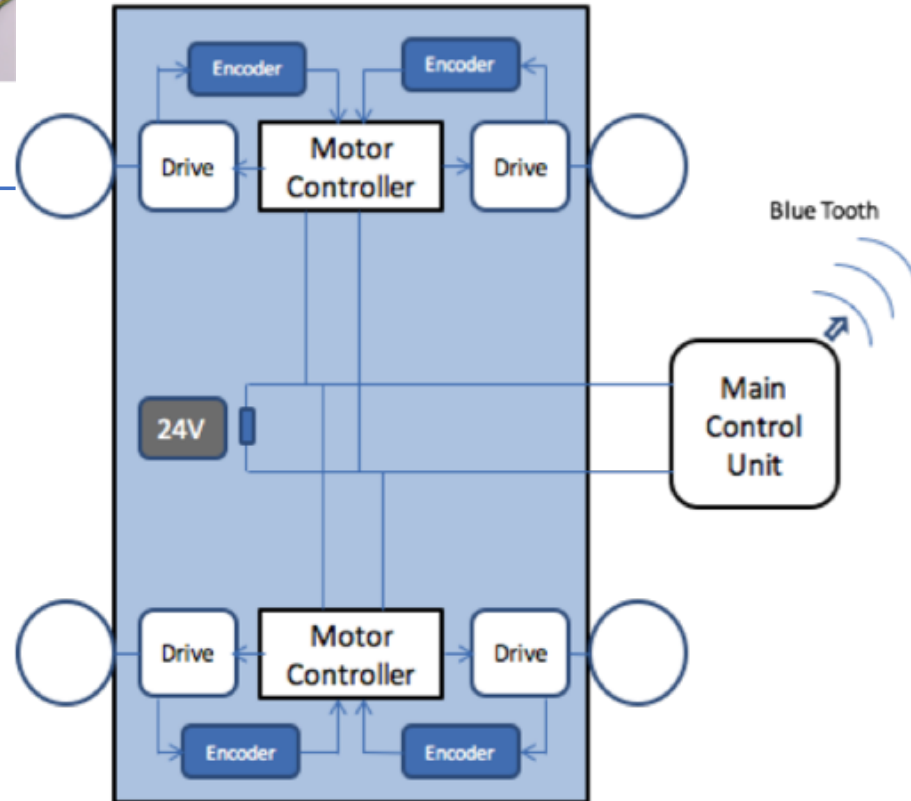
- Different types of hazards limit the improvement in performance gained with pipelining.
- Instruction Cache: speed up executable instruction fetch.
- Data Cache: speed up both data fetch and data store.
- Forwarding Unit & Hazard detection.
- Using VHDL and simulated in ModelSim.

Project Experience

Android-based Mecanum
Wheel Robot Car

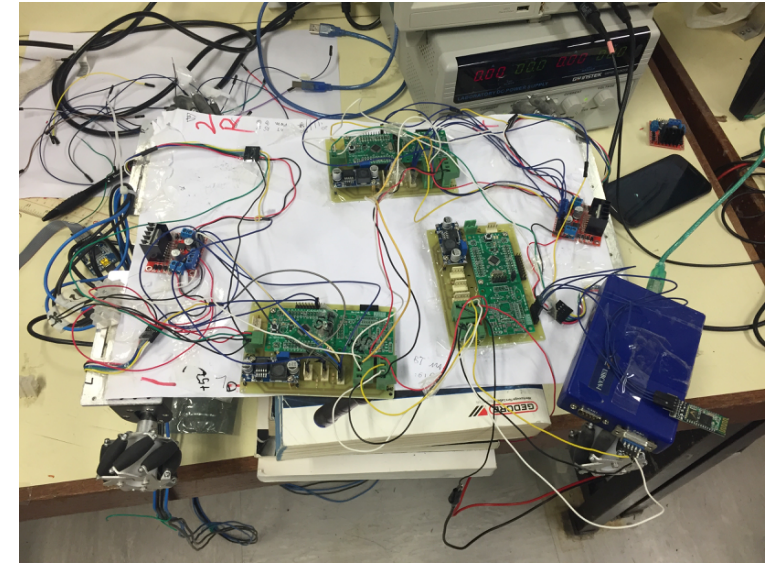
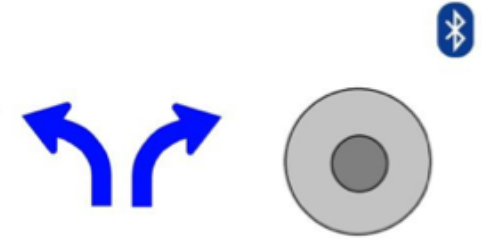


YouTube Link for demonstration: <https://www.youtube.com/watch?v=ALZCfhjqnJk>



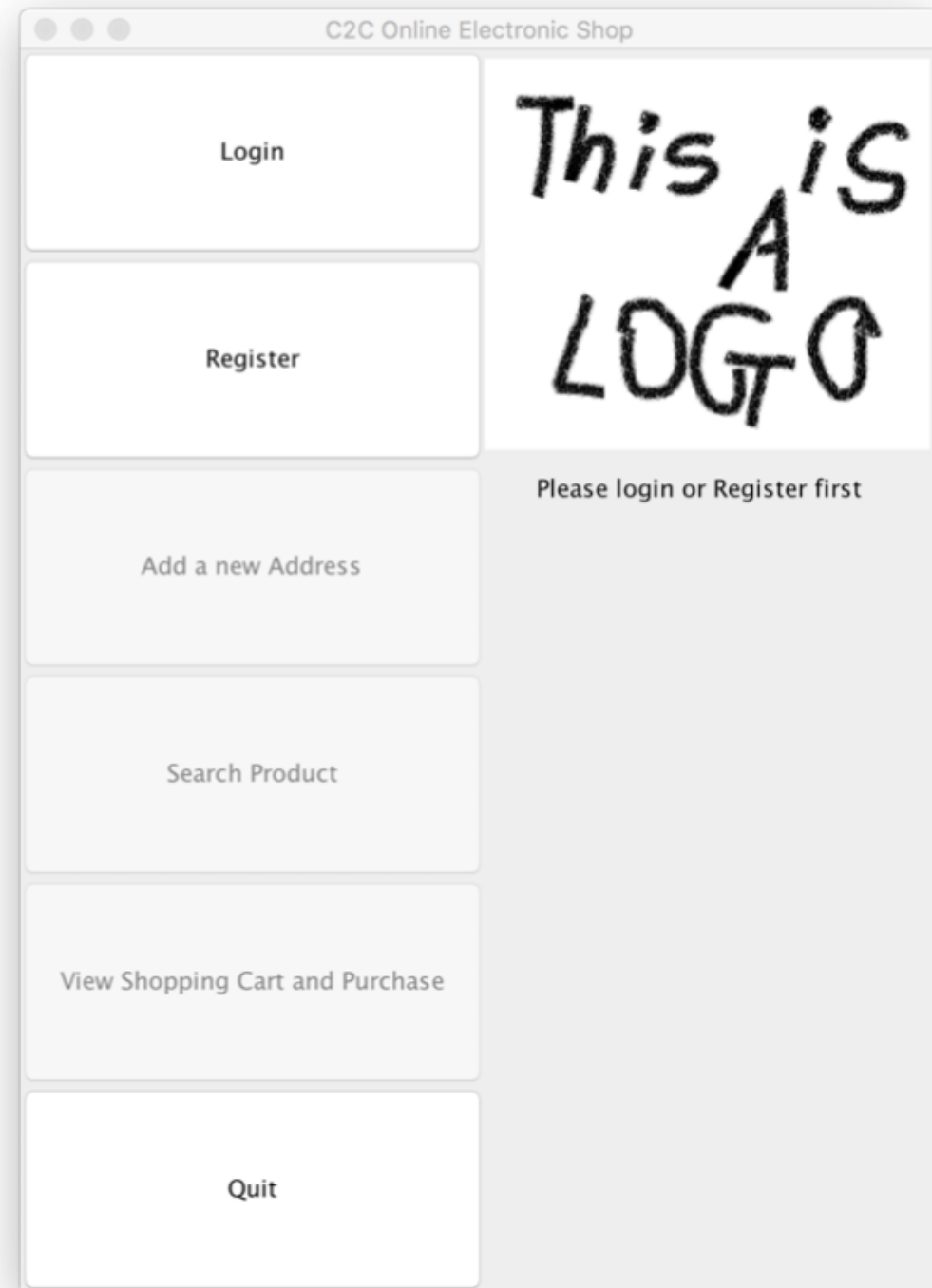
Android-Based Mecanum Wheel Robot Car

- Android Application: HTML5, CSS, JS, Node.js
- Bluetooth Communication
- Embedded System Develop: C
- Closed loop feedback speed control: PID control
- Circuit Design and Fabrication



Project Experience

Shopping Cart Database
Project



Shopping Cart Database Project

- Design the entities and relationships
- Create Database: PostgreSQL/DB2
- Create GUI: JDBC, JAVA
- Website: Java Servlet, JSP

http://localhost:8080/Lab2/check-out.jsp

Your total purchase is: 41.0

To purchase the item in your shopping cart, please provide us the following information:

Name:

Credit Card Number



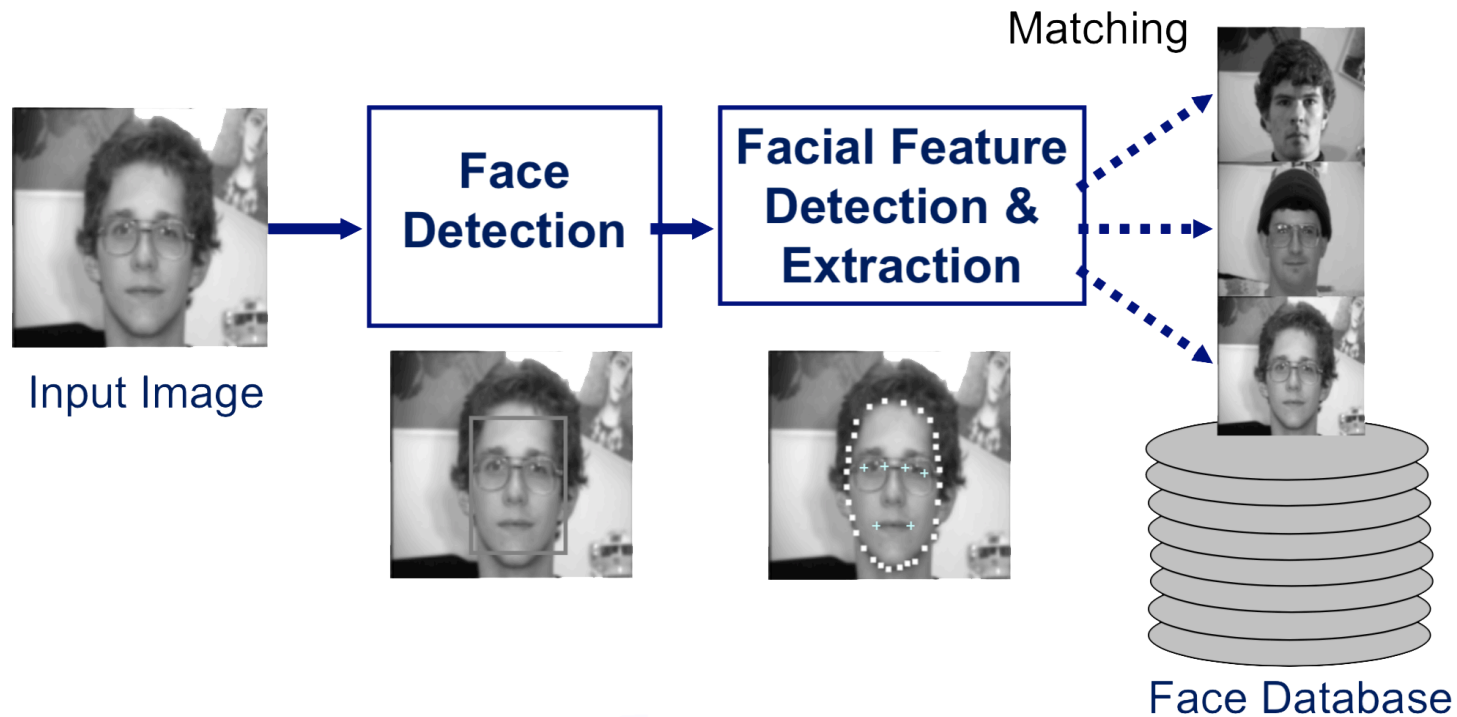
Image	ISBN	Title	Author	Edition Number	Publisher	Copyright
	0131490913	XML: How to Program	Paul Dael	2	McGraw-Hill	2000
	0131525239	Beginning with Objective-C	Paul Dael	3	McGraw-Hill	2000
	0132404188	iPhone SDK 3	Andrew Gindberg	2	McGraw-Hill	2000

- Machine Learning and Face Recognition
- Natural Language Processing (NLP)

Research Experience

Machine Learning and Face Recognition

- Currently working on the topic of face recognition in the wild.
- For face recognition, image features are first extracted and then matched to those features in a gallery set.

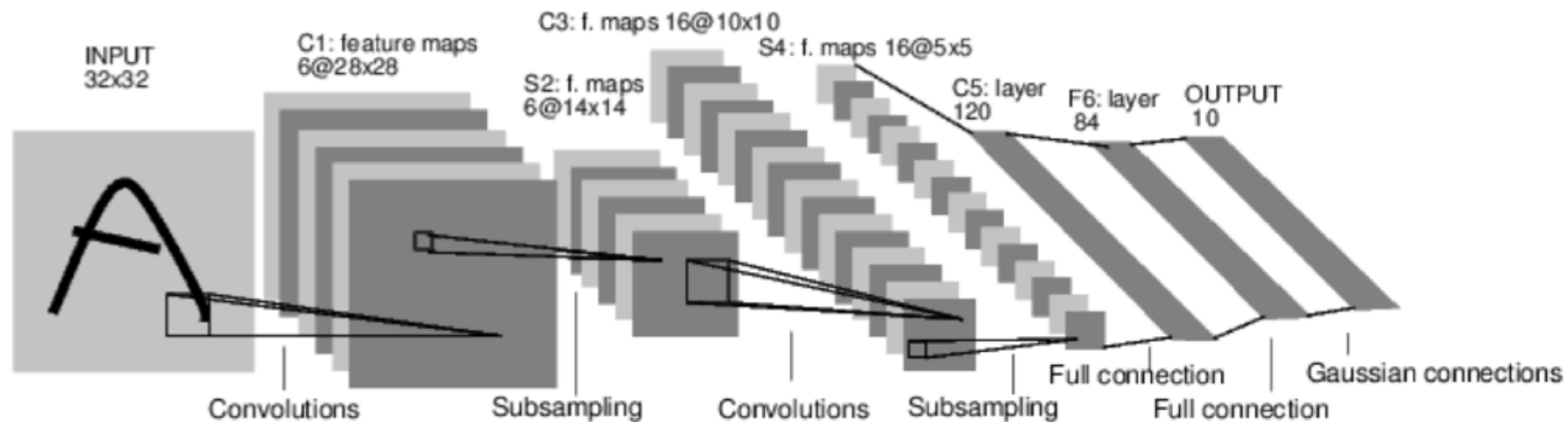


Machine Learning and Face Recognition

- Complicated environment and Multiresolution make it hard to detect face and extract features.
- The recognition performance depends on the quantity and quality of the features extracted from face images.
- Super-resolution techniques are needed for low-resolution face recognition to get high quality features.
- Various techniques/Algorithms can be used for face feature extraction, such as LBP, HOG, etc.
- Based on the extracted feature, we can perform face matching.

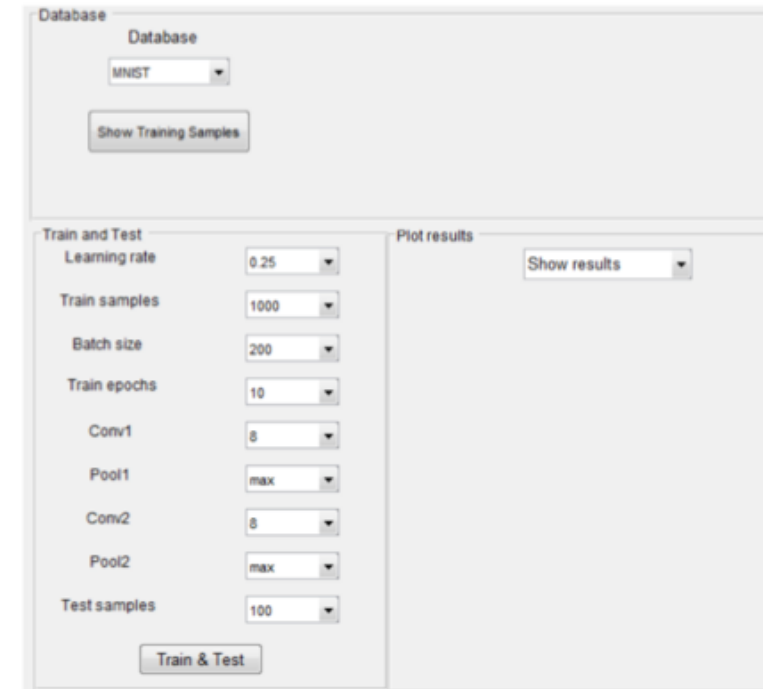
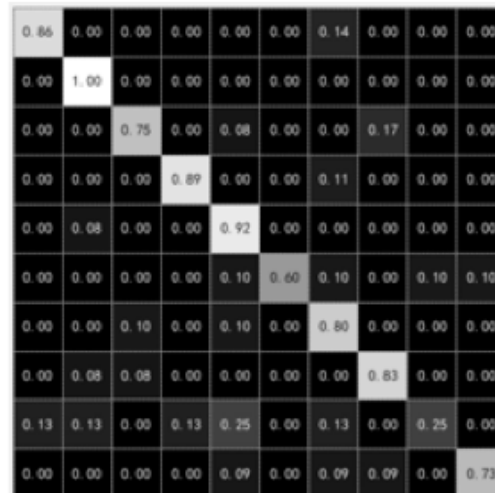
Machine Learning and Face Recognition

- Convolutional Neural Network (CNN) is the current state-of-the-art model architecture for image matching or detection.
- Apply filters to the raw pixel data of an image to extract and learn features, which the model can then use for classification.
- GPU can be used for parallel processing (pipelined process).
- Calculation and training is faster.



A CNN for MNIST Dataset

- MNIST: A handwritten digits 0–9 database, formatted as 28x28-pixel monochrome images.
- Objective: Create a CNN to classify the MNIST dataset into 10 classes (0-9).
- Created a GUI in Python for training and testing, which can allow user to adjust parameters and show the training and testing results.



Database

Database

MNIST

Show Training Samples

Train and Test

Learning rate: 0.25

Train samples: 1000

Batch size: 200

Train epochs: 10

Conv1: 8

Pool1: max

Conv2: 8

Pool2: max

Test samples: 100

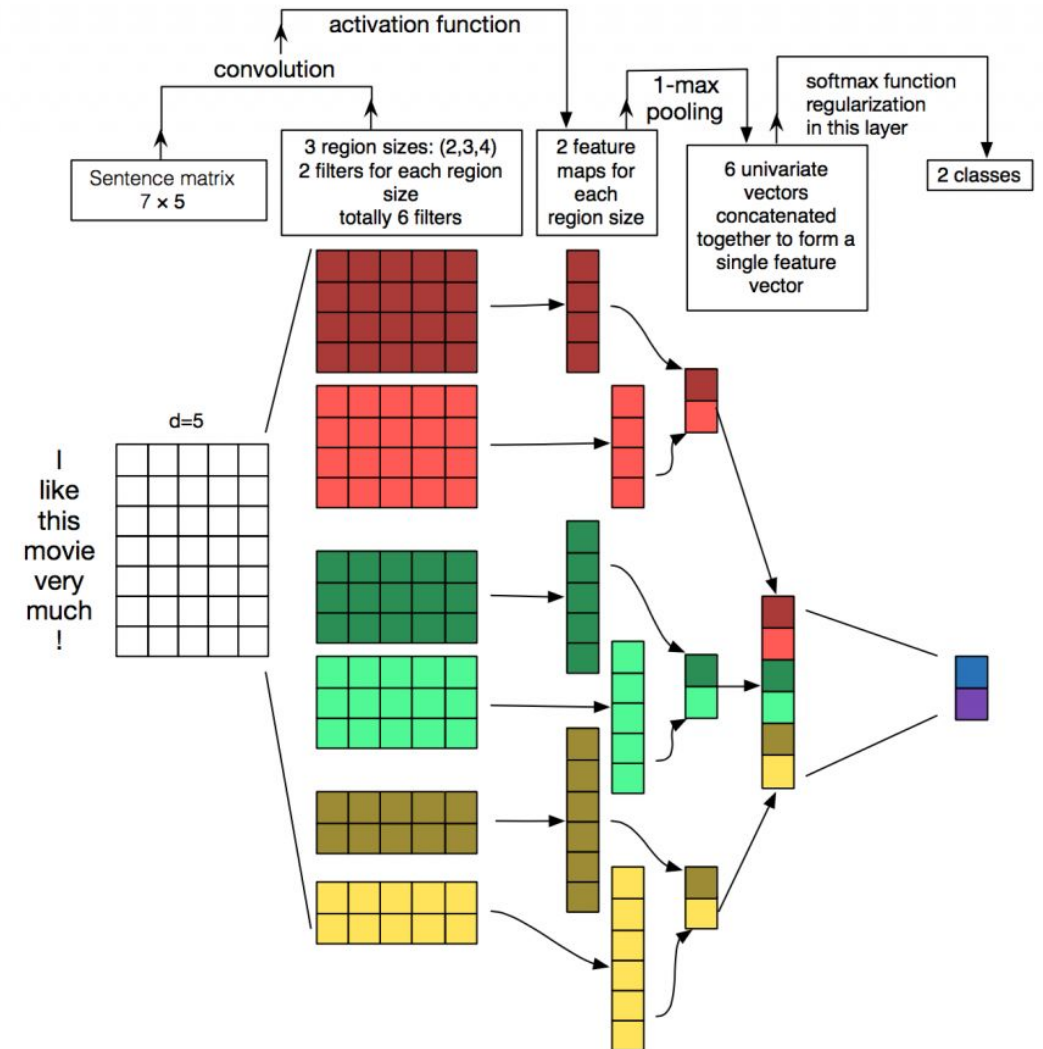
Train & Test

Plot results

Show results

Natural Language Processing (NLP)

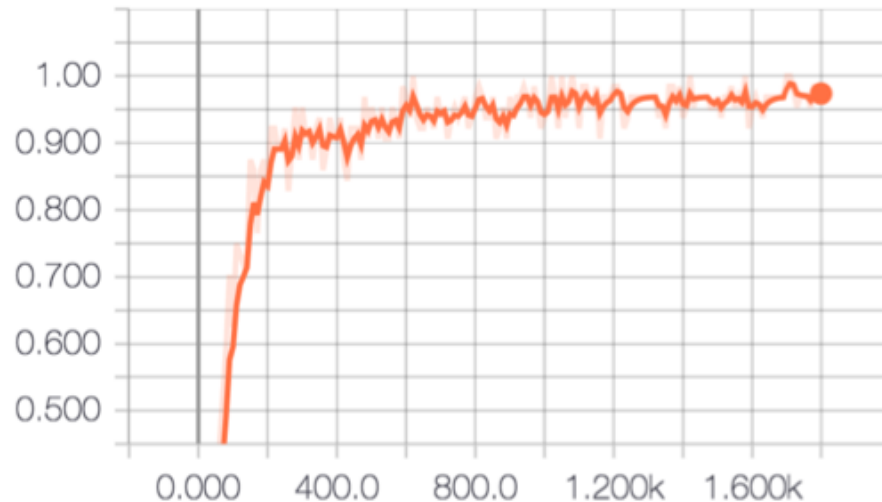
- Mainly focus on implementing a CNN for the Chinese Language text classification based in TensorFlow.
- Data and pre-processing: Remove noise of data.
- Word-embedding: The input to most NLP tasks are words represented as a matrix, instead of image pixel.
- Model Selection: CNN/RNN.
- Training & Testing in the TensorFlow.



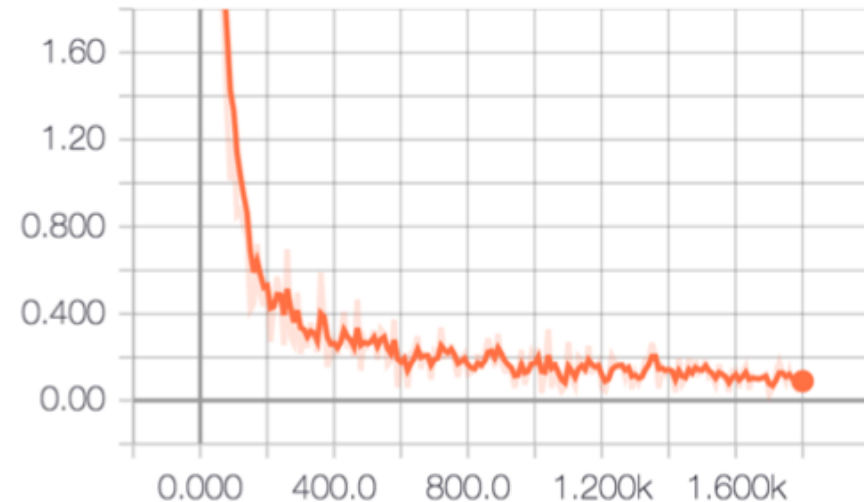
Natural Language Processing (NLP)

- Used a subset dataset provided by Jinri Toutiao and it contains 10 classes.
- Training dataset: 5000*10 and Verification dataset: 500*10.
- 1800 times iteration.

accuracy_1



loss



Natural Language Processing (NLP)

- Testing dataset: 1000*10
- Overall loss: 0.14 and Testing Accuracy: 96.04%.
- The confusion Matrix also shows a good classification result.
- Next step: Implement a RNN system to do text classification.

Configuring CNN model...

Loading test data...

Testing...

Test Loss: 0.14, Test Acc: 96.04%

Precision, Recall and F1-Score...

	precision	recall	f1-score	support
体育	0.99	0.99	0.99	1000
财经	0.96	0.99	0.97	1000
房产	1.00	1.00	1.00	1000
家居	0.95	0.91	0.93	1000
教育	0.95	0.89	0.92	1000
科技	0.94	0.97	0.95	1000
时尚	0.95	0.97	0.96	1000
时政	0.94	0.94	0.94	1000
游戏	0.97	0.96	0.97	1000
娱乐	0.95	0.98	0.97	1000
avg / total	0.96	0.96	0.96	10000

Confusion Matrix...

```
[[991  0  0  0  2  1  0  4  1  1]
 [  0 992  0  0  2  1  0  5  0  0]
 [  0  1 996  0  1  1  0  0  0  1]
 [  0 14  0 912  7 15  9 29  3 11]
 [  2  9  0 12 892 22 18 21 10 14]
 [  0  0  0 10  1 968  4  3 12  2]
 [  1  0  0  9  4  4 971  0  2  9]
 [  1 16  0  4 18 12  1 941  1  6]
 [  2  4  1  5  4  5 10  1 962  6]
 [  1  0  1  6  4  3  5  0  1 979]]
```