

# model: **Artificial Intelligence-GT19**

## **AI (Artificial Intelligence) Training System**

### **Features**

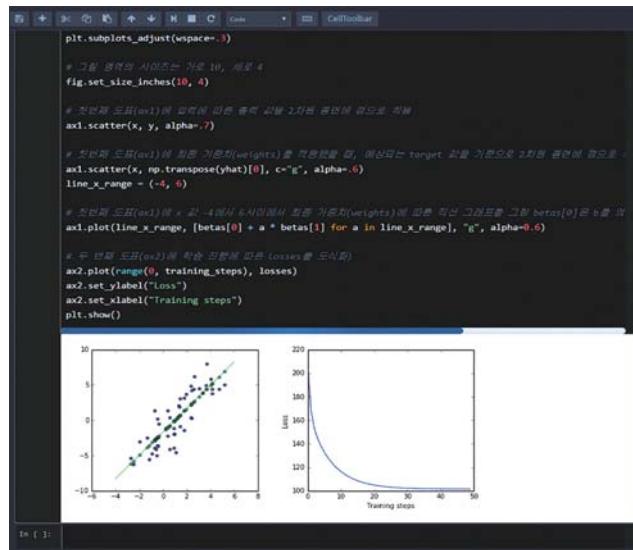
- 1) Artificial Intelligence (AI) is a field that focuses on solving cognitive problems, primarily related to human intelligence, as learning, problem solving, and pattern recognition.
- 2) Artificial Intelligence-GT19 use RaspberryPi and Python to learn the use of Google Assistant voice recognition and image processing (visual recognition) by using camera.
- 3) Through voice recognition, user can do the practice of application system using various devices from sensor and actuator control.
- 4) Through visual recognition, the user realizes object detection and recognition from various images and learns how to control the device.
- 5) The user also learn how to detect and recognize images using OpenCV, which is widely used for visual recognition.
- 6) In addition, the Jupiter Lab in Windows environment will be used to learn the basics of AI machine learning such as Linear Regression, Logistic Regression, Softmax classification, and CNN,, etc using TensorFlow and Python.
- 7) User also learn how to improve the recognition rate by applying various algorithms of machine learning to MINIST number recognition.
- 8) From our products to our open-source platforms, we ensure that the benefits of AI are accessible to everyone.
- 9) Artificial Intelligence-GT19 is the training system that can meet the needs for a myriad of national & regional initiatives underway seeking to establish organizational structures of AI.

### **System configuration**

- 1) Hardware photo



2) Software screen-shop



## Components

Hardware platform	1 set
Program CD	1 ea
USB Camera	1 ea
Speaker & USB Microphone	1 ea
Micro USB Cable	1 ea
Jupmer cable	1 ea
DC 12V/3A Adaptor	1 ea
Micri SD Card & Reader	1 ea
HDMI Cable	1 ea
USB Keyboard & Mouse	each 1 ea
7 Inch LCD	1 ea
Textbook	2 book

## Training contents

< book-1, AI Voice recognition and Machine learning >

- Chap 1, Overview of Artificial Intelligence (Theory)
  - 1.1 Introduction of AI (Artificial Intelligence)
  - 1.2 History and types of AI
- Chap 2, Artificial Intelligence Classification / Product Configuration and Features (Theory)
  - 2.1 Learning and reasoning
  - 2.2 Understanding languages
  - 2.3 Visual recognition
  - 2.4 Situation awareness
  - 2.5 AI Product Configuration and Features
- Chap 3, RaspberryPi development environment (Practice)
  - 3.1 About RaspberryPi
  - 3.2 Hardware Assemble
  - 3.3 Operating system installation
  - 3.4 Network settings
  - 3.5 SAMBA setting
  - 3.6 GPIO introduction
  - 3.7 GPIO library installation
  - 3.8 System development method
- Chap 4, Voice recognition and Google assistant (Practice)
  - Introduction of Google Assistant
  - Build Google assistant development environment
- Chap 5, Voice recognition device (Practice)
  - Device control method through speech recognition
  - LED control practice
  - Switch (KEYPAD) control practice
- Chap 6, Voice recognition device (Practice)
  - FAN control practice
  - TEXTLCD control practice
  - Buzzer control practice
- Chap 7, Interlock with voice recognition sensor -1 (Practice)
  - Interlock with illuminance sensor
  - Interlock with GAS sensor

- Chap 8, Interlock with voice recognition sensor -2 (Practice)
  - Interlock with Temperature and Humidity Sensor (DHT11)
  - Interlock with Motion sensor (PIR)
- Chap 9, Design of voice recognition system (Theory)
  - Fire monitoring system
  - Cooling control system
  - Lighting control system
  - Intrusion detection system
- Chap 10, Overview of Machine learning (Practice)
  - Introduction of Machine learning
  - Basics of Machine learning
- Chap 11, Machine learning development environment (Theory & Practice)
  - Introduction of TensorFlow
  - Construction of TensorFlow development environment
  - TensorFlow API
- Chap 12, Linear regression (Theory & Practice)
  - Theory of Linear regression
  - Minimize the losses
  - Multivariable Linear Regression
- Chap 13, Machine Learning Practice (Theory & Practice)
  - Generalization, Over Integration, and Learning Rate Theory
  - Verification exercise
- Chap 14, Machine learning and characteristics (Theory & Practice)
  - Character extraction and data refining theory
  - Cross-Cultural Practice
- Chap, 15 Logistic regression (Theory & Practice)
  - Theory of Logistic regression
  - Minimize the losses
- Chap 16, Overview of Internet of Things (IoT)
- Chap 17, Design of Internet of Things (IoT)

< book-2, AI Vision and Machine learning >

- Chap 1, Overview of Artificial Intelligence (Theory)
  - 1.1 Introduction of AI (Artificial Intelligence)
  - 1.2 History and types of AI
- Chap 2, Artificial Intelligence Classification / Product Configuration and Features (Theory)
  - 2.1 Learning and reasoning
  - 2.2 Understanding languages
  - 2.3 Visual recognition
  - 2.4 Situation awareness
  - 2.5 AI Product Configuration and Features
- Chap 3, RaspberryPi development environment (Practice)
  - 3.1 About RaspberryPi
  - 3.2 Accessories of RaspberryPi
  - 3.3 Hardware Assemble
  - 3.4 Operating system installation
  - 3.5 Network settings
  - 3.6 SAMBA setting
  - 3.7 GPIO introduction
  - 3.8 GPIO library installation
  - 3.9 AI development environment setup
- Chap 4, Visual recognition and Google assistant (Practice)
  - Introduction of Google Assistant
  - Build Google assistant development environment
  - Visual recognition devices
- Chap 5, Visual Recognition-1 using Google Vision API
  - Label recognition
  - Text recognition
- Chap 6, Visual Recognition-2 using Google Vision API
  - Human face recognition
  - Facial expression recognition
  - Landmark recognition
- Chap 7, Visual Recognition-3 using Google Vision API
  - Image properties
  - Safe search properties
- Chap 8, Visual Recognition-4 using Google Vision API
  - Hints for cut
  - Web recognition
- Chap 9, Visual Recognition-5 using Google Vision API
  - Document text recognition
  - Logo recognition
- Chap 10, TensorFlow development environment
  - Introduction of TensorFlow
  - Development environment setting for TensorFlow
  - use of TensorFlow library
  - Implement TensorFlow algorithm / programming
- Chap 11, Image Recognition-1 using Machine Learning
  - Softmax classification
  - MNIST number recognition
- Chap 12, Image Recognition-2 using Machine Learning
  - CNN (Convolution Neural Network)
  - MNIST number recognition using CNN
- Chap 13, Image Recognition-3 using Machine Learning
  - Introduction of objects recognition
  - Image recognition
- Chap 14, OpenCV development environment
  - Introduction of OpenCV
  - OpenCV development environment setting
- Chap 15, Visual recognition-1 using OpenCV
  - use the camera
  - Human face recognition

## Spec

### 1) Hardware spec

#### (1) Raspberrypi4 Board

Broadcom BCM2711, Quad Core Cortex-A72 (ARMv8) 64-bit SoC @ 1.8GHz  
 2.4GHz and 5.0 GHz IEEE 802.11ac wireless LAN  
 Bluetooth 5.0, BLE  
 Gigabit Ethernet  
 1GB,2GB,4GB or 8GB LPDDR4-3200 SDRAM (depending on model)  
 Extended 40-pin GPIO header  
 2 x micro-HDMI ports (up to 4kp60 supported)  
 2 USB 3.0 ports, 2 USB 2.0 ports  
 CSI camera port for connecting a Raspberry Pi camera  
 DSI display port for connecting a Raspberry Pi touchscreen display  
 4-pole stereo output and composite video port  
 H265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)  
 OpenGL Es 3.1, Vulkan 1.0  
 Micro SD port for loading your operating system and storing data  
 5V DC via USB-C connector (minimum 3A\*)  
 5V DC via GPIO header (minimum 3A\*)  
 Power-over-Ethernet (PoE) support (requires separate PoE HAT)  
 Operation temperature: 0 - 50 degrees C ambient

#### (2) Base Interface

Raspberry Pi 4 Adapter Board : LEVEL Buffer I/O Connector  
 Raspberry Pi 4 40-pin GPIO header  
 Raspberry Pi 4 8-pin Analog Input Port  
 Base board power supply switch  
 MUX Switch (SW1, SW2)  
 Sensor and Actuator Control Connector(9)  
 LED BAR display, Microphone input, Speaker output  
 DC 12V Power Supply Connector

#### (2) Base devices

PIR Motion Sensor	Block time : 2.5S (default)
	Delay time : 5 S (default)
	Sentry Angle : < 110 degree
	Sentry Distance : 3m (default) - max 7m
	Lens Size : Diameter : 23mm (Default)
ILUM Sensor	CDS 10mm Light Sensor
TEMP/HUMI Sensor (DHT11)	Temperature : -40~80° ... Range, Digital input I/F
	Humidity : 0~100% Range, Digital input I/F
Smoke Sensor (Gas)	Interface type : Analog, Wide detecting scope Wide detecting scope
DC BLOWER FAN	51 X 52mm DC BLOWER FAN
LED	10mm Round type LED(4EA)
KEYPAD	3-COLUMN, 4-ROW push Button Switch
BUZZER	Interface type : Analog
	wide detecting scope
TEXT LCD	16 X 2 Text lcd

#### (4) External Device

Display	7 inch IPS Screen LCD
	1024 x 600 hardware resolution
	Capacitive Touch Control
WebCam	HD Web Cam C310
	Max resolution : 720p / 30fps
	Angle : 60 degree
Microphone	CMP-G7 USB Microphone
	Direction : Omnidirectional
	Sensitivity : 20 ± 4dB

## 2) Software spec

- (1) Linux Kernel Version : Linux raspberrypi 6.1.24-v8+
- (2) Rasbian OS Version : Rasbian GNU/Linux 10 (buster)
- (3) Google Assistant : Google Assistant library 1.0.1, Google Assistant grpc 0.2.1
- (4) OpenCV : opencv-python 3.4.4, opencv-contrib-python 3.4.3
- (5) TensorFlow : TensorFlow 1.14.0
- (6) ALL Device Demo Program
- (7) Provides video recognition demo program based on Google Vision and OpenCV
  - Google Vision based image analysis and prediction
    - \* GV Face Detection : Recognizing human faces in images
    - \* GV Facial Expression Detection : Recognition of emotional states on human faces in images
    - \* GV Image Properties Detection : Detects the parts color of the object shown in the image
    - \* GV Label Detection : Recognize each element in the image
    - \* GV Landmark Detection : Recognize the name and location of the building in the image
    - \* GV Safe-Search Detction : Recognize the safety status of the person in the image
    - \* GV Text Detection : Text detection in images
  - Face recognition through OpenCV-based image processing and face-recognition library
  - \* OpenCV Control Device by Face Recognition : LED blinks when recognizing an image learned through the camera
  - \* OpenCV Face Detection : Detecting human faces through the camera
  - \* OpenCV Face detection in picture : Recognition of a given person in the image
  - \* OpenCV Face Recognition : Recognition of the person determined through the camera
  - \* OpenCV Face Recognition from WebCam : Recognition of a number of learned characters through the camera
  - OpenCV-based image processing and numeric recognition through Tensorflow
  - \* OpenCV Number Recognition : Learn Mnist numeric data through Tensorflow, recognize the numbers in the handwritten image file,  
and output the numbers to the speaker.
- (8) Voice Recognition Device Control based on Google Assistant
  - Device Test Application Source based on voice recognition : Individual actuator control and sensing through voice recognition
    - \* led.py, keypad.py, fan.py, textlcd.py, buzzer.py, cds.py, gas.py, dht11.py, motion.py,
  - Application Programmig Source based on voice recognition : Various actuator control and sensing through voice recognition
    - \* fire\_detect.py, air\_condition.py, light\_control.py, security.py
- (9) IoT based Application Programmig Source : LED control and sensor value sensing through Android App
  - adc.py, \_led.py, deviceserver.py
- (10) Tensorflow machine learning theory and practice program is provided
  - Machine learning basics
  - Linear regression
  - Logistic regression
  - Softmax
  - CNN
- (11) Tensorflow based Machine learning Programming Source
  - : A practical program on the theory of linear regression and logistic regression, methods of implementing them, and algorithms
    - linear\_regression.py, cost\_graph.py, gradient\_descent.py, gradient\_descent\_tf\_optimizer.py,
    - multi\_variable\_linear\_regression.py, multi\_variable\_matmul\_linear\_regression.py, input\_file\_data.py,
    - logistic\_regression.py
- (12) Tensorflow based Deep learning Programming Source
  - : Deep learning theory using Mnist data, a practical program on how to implement it, and algorithms
    - softmax.py, mnist\_softmax.py, mnist\_nn.py, mnist\_nn\_xavier.py, mnist\_nn\_dropout.py, mnist\_cnn.py
- (13) Tensorflow based image recognition deep learning Programming Source
  - : fRecognize flowers through lower image data deep learning
    - label\_image.py, retrain.py