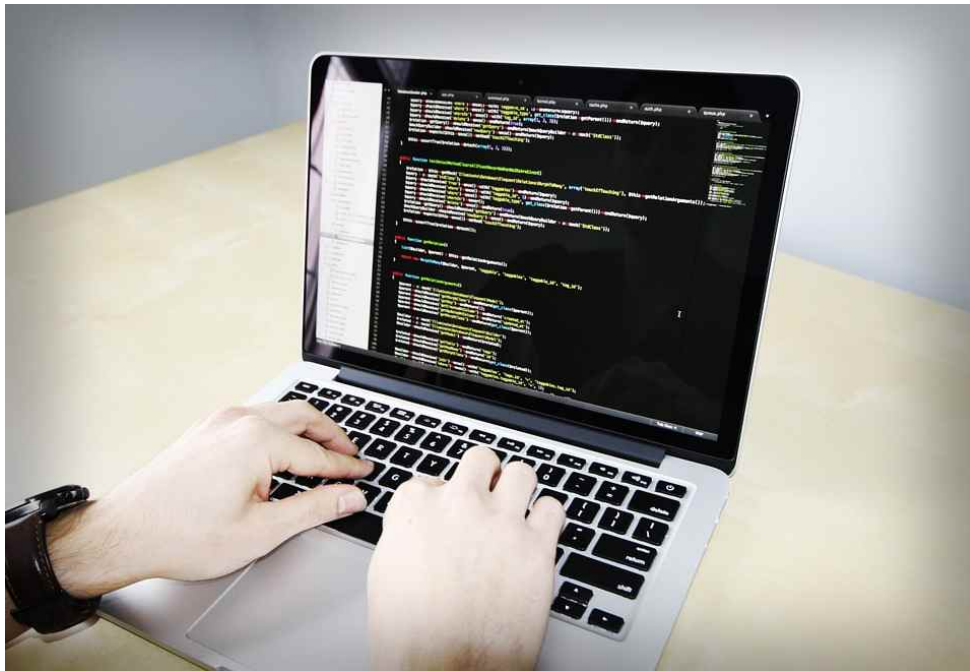

영상인식과 머신러닝을 이용한 실용 로봇제어 기초

영상인식과 머신러닝을 이용한 로봇제어 기초 목차

1. 프로그래밍 기초.....	1
1.1. 파이썬 개발 환경 설정 하기.....	2
1.2. 파이썬 기초.....	10
2. 개발환경.....	18
2.1. 리눅스 설치하기.....	19
2.2. 리눅스 사용하기.....	33
3. 머신 러닝 기초.....	39
3.1. 머신러닝 알고리즘.....	40
3.2. weka를 이용한 알고리즘 실습.....	62
3.3. 파이썬을 이용한 기초 알고리즘 실습.....	75
4. 로봇 제어실습.....	87
4.1. 손가락으로 제어하는 모바일 로봇.....	88
4.2. 화염학습을 통한 소방 로봇.....	103
4.3. 영상 인식을 통한 자율 주행 로봇.....	129
5. 부록	
5.1. 프로그램소스.....	147

프로그래밍 기초

1. 파이썬 개발환경 설정하기
2. 파이썬 기초



<대단원 개요>

이 단원에서는 머신러닝에 유용하게 사용하는 파이썬 프로그래밍을 위한 통합 개발 환경(파이참)의 설치와 기초적인 파이썬 문법에 대하여 공부한다.

01 파이썬 개발 환경 설정하기

학습목표

- 파이썬 프로그램 개발환경을 설치할 수 있다.
- 아나콘다를 이용한 파이썬 머신러닝 개발 환경을 설정할 수 있다.

1. 머신러닝을 위한 프로그램 개발환경

파이썬을 이용하여 머신러닝 프로그램을 개발하거나 학습을 할 때는 다양한 패키지들을 설치해 주어야 한다. 이런 불편함을 해결하기 위해 머신러닝용 패키지들이 개발되어 있는데 그 중 하나가 아나콘다이다. 이 단원에서는 아나콘다를 이용하여 머신러닝 프로그램 개발 환경을 설정하는 기법을 학습한다.

2. 아나콘다의 설치

2-1. 아나콘다 받기

<https://www.continuum.io/downloads>로 들어간다.

자신의 운영체제에 맞는 아나콘다 버전을 다운로드 한다.

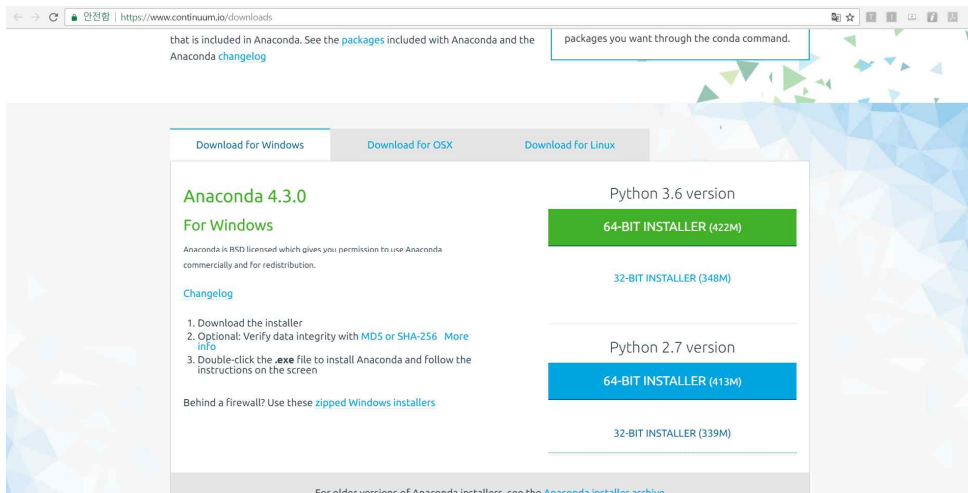


그림 2 아나콘다 다운로드

2-2. 자신의 컴퓨터가 64bit면 64bit installer를, 32bit면 32bit installer를 설치한다. (Python 버전은 3.6 버전 이상을 다운받는다.)

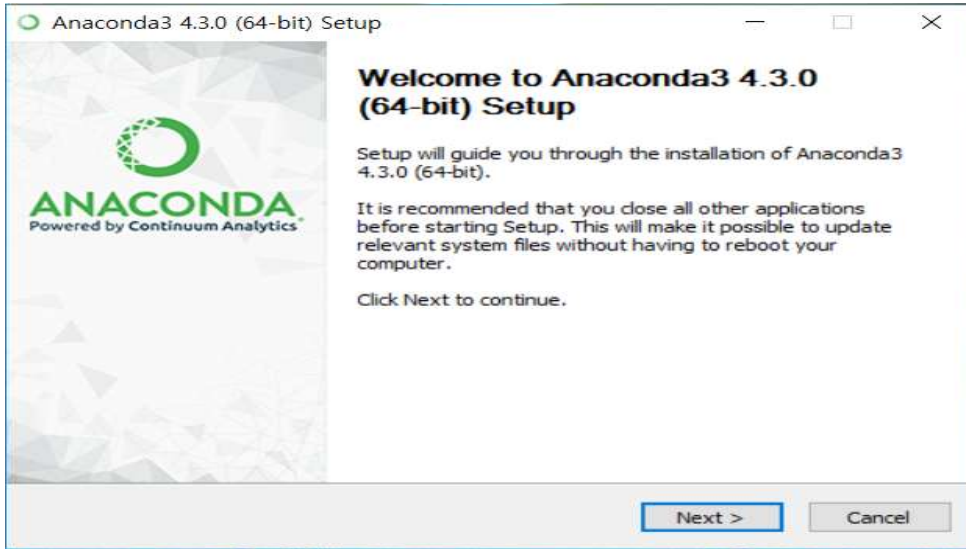


그림 3 아나콘다 설치

3. 파이참 설치(pycharm)

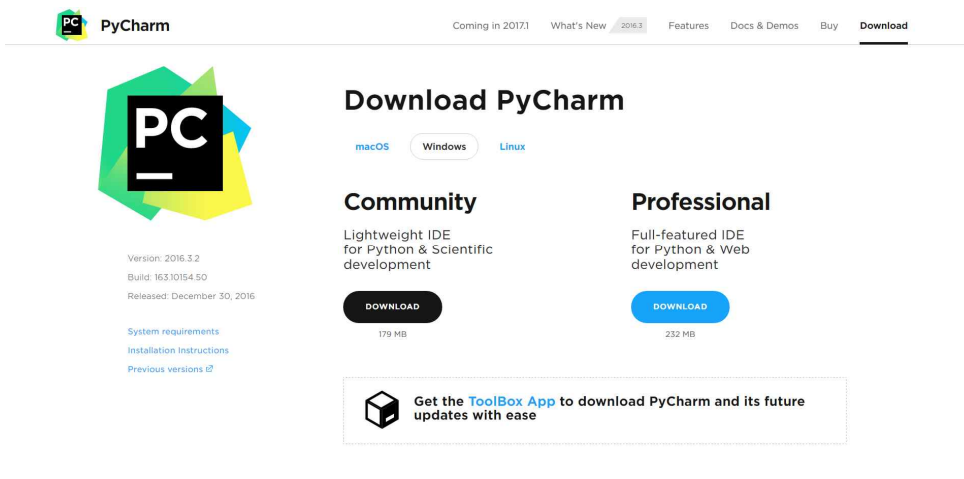


그림 4 파이참 설치

파이참은 파이썬 통합개발환경이다. 파이썬은 다양한 통합개발환경이 있는데 어떤 것은 유료이며 강력한 성능의 무료 통합개발환경 프로그램도 존재한다. 파이참은 강력한 기능을 가지고 있는 통합개발환경으로 다양한 기능을 모두 사용할 수 있는 유료 버전과 일부 기능을 제안한 무료 버전이 있다.

<https://www.jetbrains.com/pycharm/download/>로 들어간다.

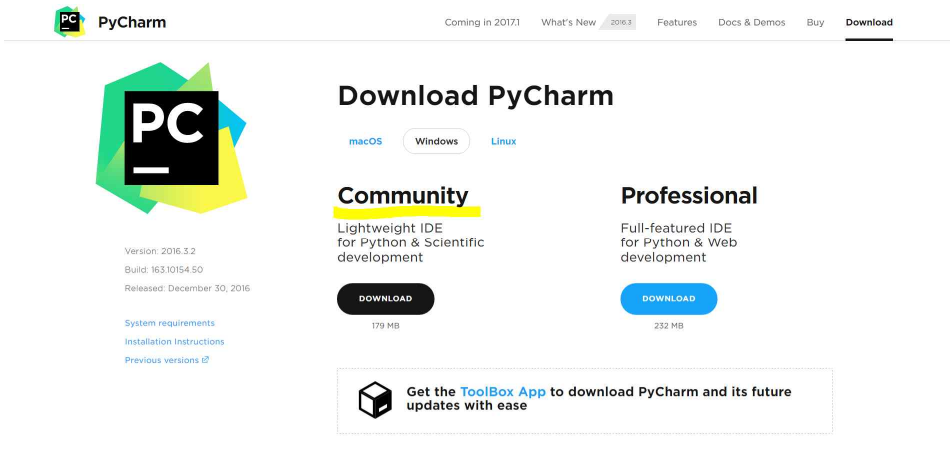


그림 5 파이참 다운로드

community 버전을 다운받은 후 파이참 설치파일을 열면 아래와 같은 창이 나타난다.

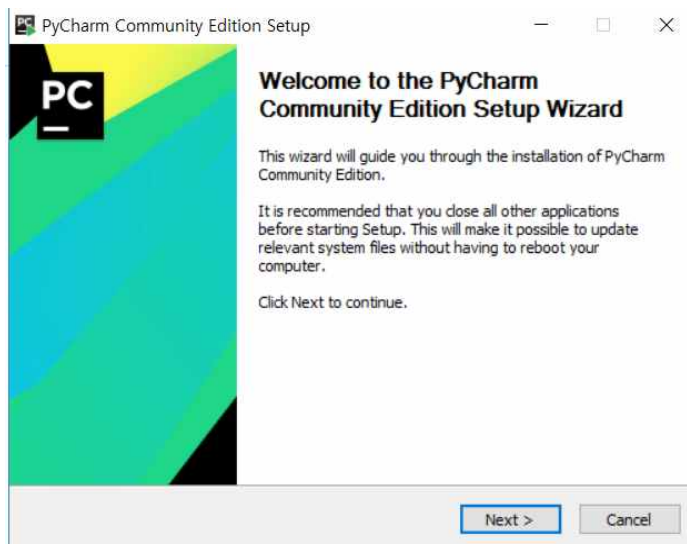
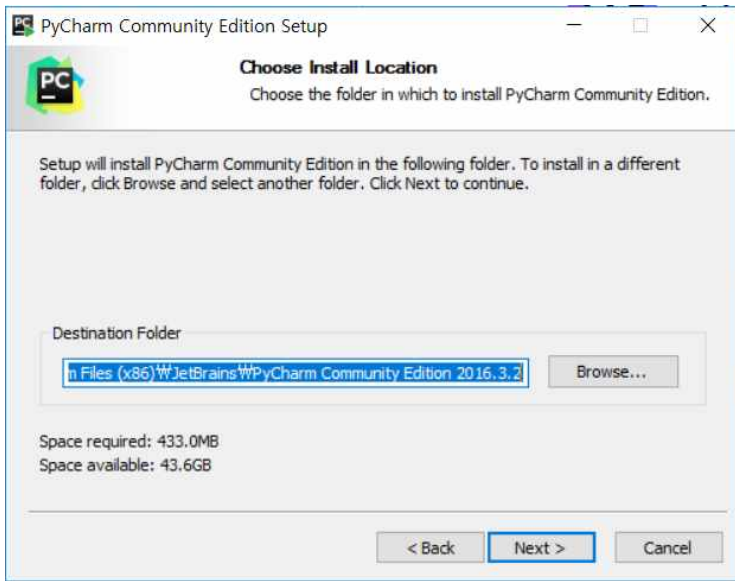


그림 6 커뮤니티 버전

설치될 경로를 설정하고 입력한다.



파이참 상업용 버전
파이참을 이용하여 상업용 프로그래밍을 개발 할 때는 상업용 버전을 구입하여 사용하여야 한다.

그림 7 파이참 설치 경로 입력

정상적으로 설치한 후 파이참을 실행시키면 다음과 같은 창을 볼 수 있다.



그림 8 파이참 설치 완료

여기서 Create New Project를 클릭하면 아래와 같은 창이 나타나는데, Interpreter 오른쪽의 설정 버튼을 클릭한 후 Add Local을 클릭한다.

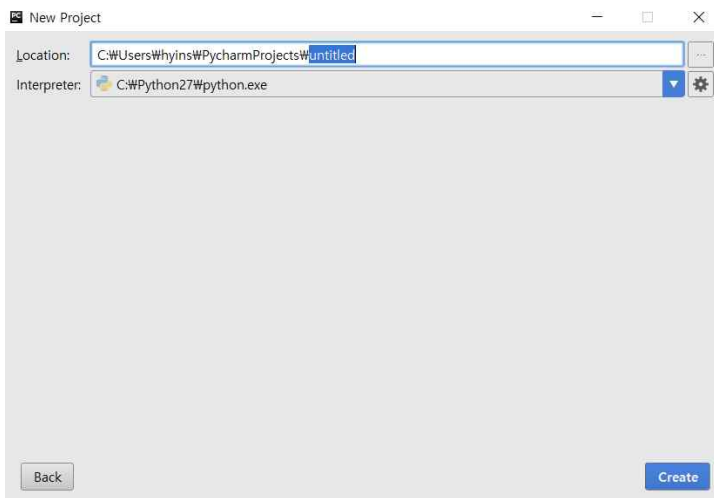


그림 9 파이참 프로젝트의 생성

이곳에서 아나콘다에 있는 Python.exe를 연결한다. 설치만 클릭했다면 C:\Users\<사용자이름>\Anaconda3로 들어가서 Python.exe를 클릭하여 연결한다. 한 대의 컴퓨터에 여러 버전의 파이썬 인터프리터를 가지고 있는 경우 여기서 적절한 인터프리터를 연결해주어야 한다.

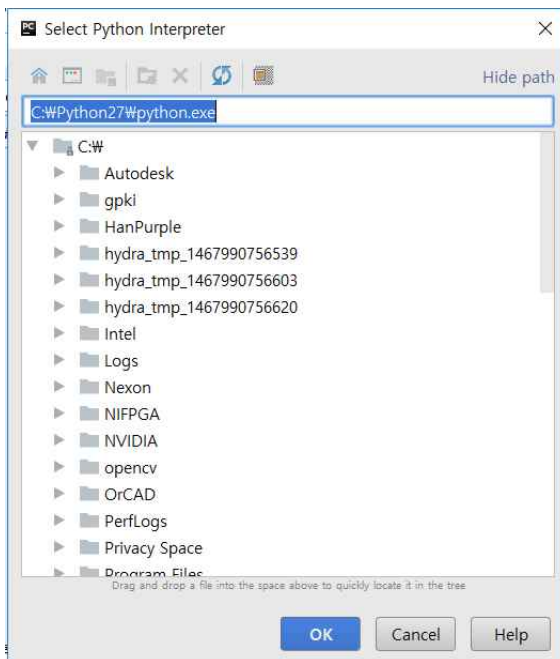


그림 10 파이참 프로젝트 경로

2. 파이참 사용

2.1 프로젝트의 설정

대부분의 통합 개발환경이 그렇듯이 파이참도 처음으로 프로그램을 작성하기 위해서는 프로젝트를 설정해야 한다.

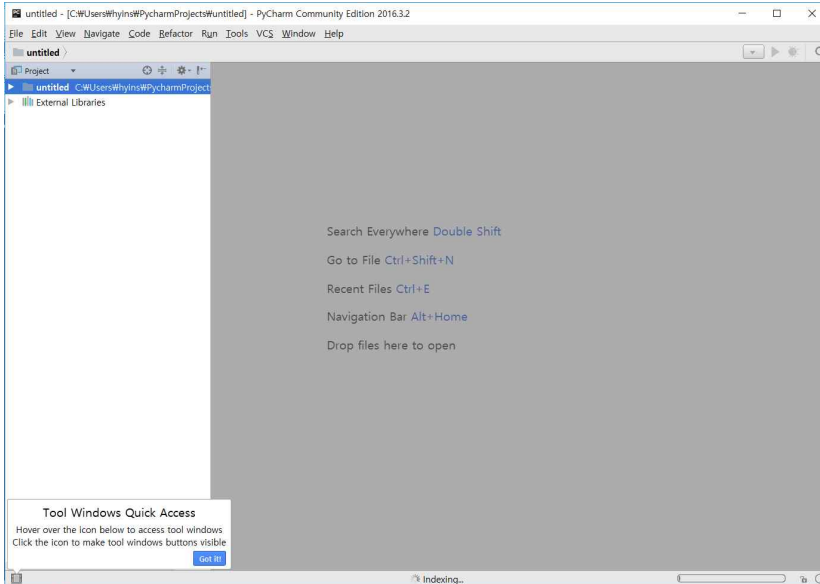


그림 11 파이참 프로젝트 생성

File->New Project 로 들어간다.

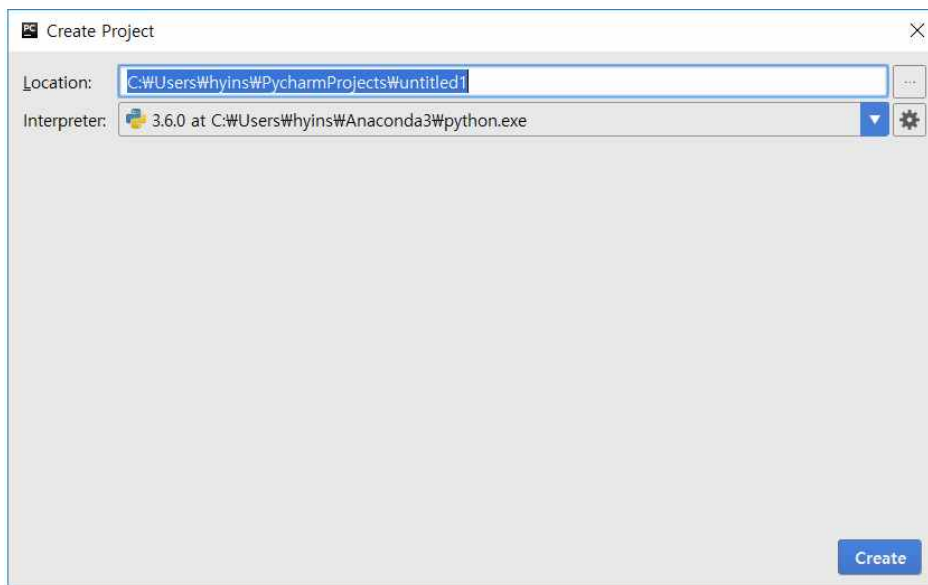
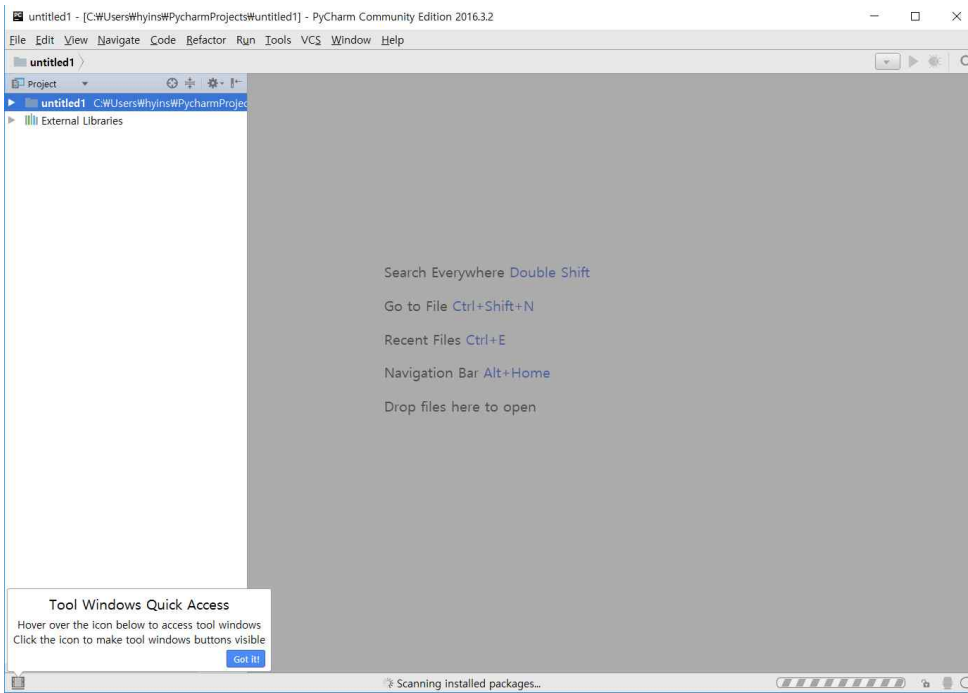


그림 12 프로젝트 경로 설정

Location에서 untitled를 변경하여 프로젝트 파일 이름을 만들 수 있다.
그 다음 Create를 클릭하면 기본적인 프로젝트를 설정할 수 있다.

2-2 .py파일 생성

2-1에서 생성한 프로젝트를 클릭 후 오른쪽 마우스 클릭 -> New -> Python File 클릭한다.



.py
파이썬의 확장자는 .py 이다.
파일을 생성하는 창에서 파일
이름만 입력하면 자동으로 확
장자가 입력된다.

그림 13 파이참 소스 파일 생성

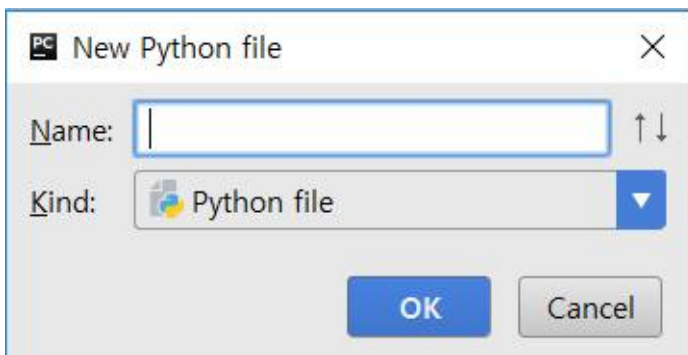


그림 14 파이참 소스 파일 생성 2

Name에 파일 이름을 쓴 후 OK 클릭하면 .py파일을 만들 수 있다.

2-3. py파일 실행하기

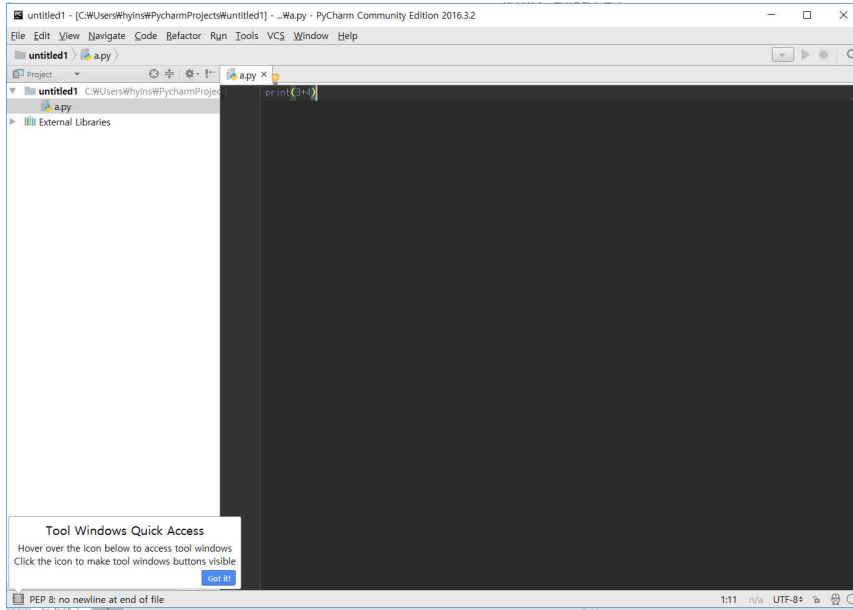


그림 15 프로그램 실행

프로그램을 작성하고 왼쪽 파일을 클릭한 후 오른쪽 마우스 클릭 -> Run “파일이름” 으로 파이썬을 실행한다.

더 알아보기: 프로그래밍 언어 파이썬(python)



귀도 반 로섬

파이썬(Python)은 1991년 프로그래머인 귀도 반 로섬(Guido van Rossum) 발표한 고급 프로그래밍 언어로, 플랫폼 독립적이며 인터프리터식, 객체지향적, 동적 타이핑(dynamically typed) 대화형 언어이다. 파이썬이라는 이름은 귀도가 좋아하는 코미디 <Monty Python's Flying Circus> 에서 따온 것이다.

파이썬은 비영리의 파이썬 소프트웨어 재단이 관리하는 개방형, 공동체 기반 개발 모델을 가지고 있다. C언어로 구현된 C파이썬 구현이 사실상의 표준이다. 파이썬은 무료이며 누구나 다운받아 사용 가능 하다. 파이썬은 확장 모듈이 다양하여 머신러닝에 자주 활용된다.

02 파이썬 기초

학습목표

- 기초적인 파이썬 문법을 사용할 수 있다.
- 통합 개발 환경을 이용하여 파이썬 프로그램을 작성할 수 있다.

1. 파이썬 기본 문법 알아보기

파이썬은 다른 언어들과 다르게 문장을 구분할 때 '{}'를 사용하지 않는다. 그 대신 ':'와 들여쓰기를 통해 '{}'와 같이 사용한다. 그리고 파이썬은 세미콜론(;)이 없기 때문에 매우 간결한 문장으로 모든 걸 표현할 수 있다. 파이썬 문법을 익히는 것은 많은 내용을 필요로 한다. 본 단원에서는 가장 기본적인 문법에 대해서만 살펴보도록 한다.

1-1. 파이썬에서의 사칙연산

파이썬 기본 문법을 학습할 때는 통합개발환경을 사용하기보다 파이썬에서 기본으로 제공하는 idle을 사용하면 편리하다. idle은 윈도 명령창에서 idle이라고 입력하고 실행하면 아래와 같은 프로그램이 실행된다.

인터프리터와 컴파일러 언어

C나 java와 같은 언어는 컴파일러형 언어이다. 이들은 프로그램을 작성하고 컴파일러로 번역을 하고 실행 파일을 만들어 실행한다. 그러나 파이썬은 인터프리터 언어이다. 따라서 파이썬은 한 줄 입력하여 바로 실행할 수 있다.

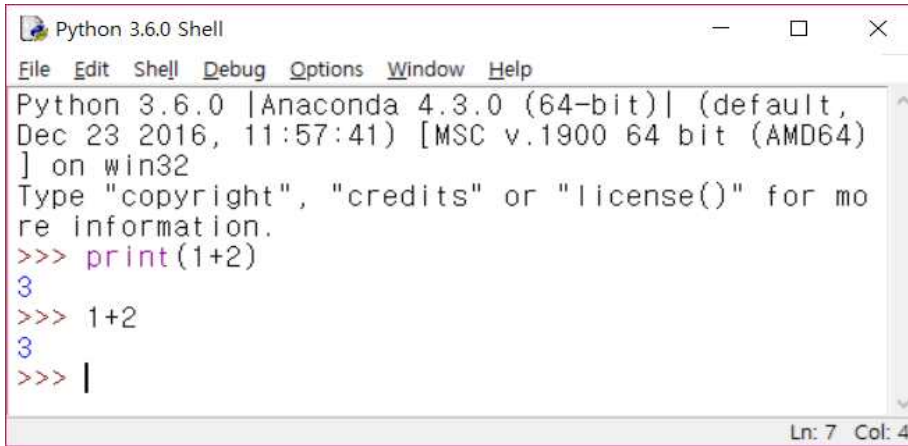


그림 16 python 기본 idle

여기서 기본적인 사칙연산을 연습해보자. print() 함수는 화면에 출력하기 위한 함수이다.

예제) print(1+2)

결과) 3



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 |Anaconda 4.3.0 (64-bit)| (default,
Dec 23 2016, 11:57:41) [MSC v.1900 64 bit (AMD64)
] on win32
Type "copyright", "credits" or "license()" for mo
re information.
>>> print(1+2)
3
>>> 1+2
3
>>> |
```

idle에서는 print()함수 없이도 출력을 확인할 수 있다.

그림 17 덧셈 연산

파이썬은 기호만 넣어주면 간결하게 결과 값을 보여준다. 나눗셈과 곱셈도 결과 값을 잘 보여준다.

예제) print(3*4)
print(4/2)

결과)12
2

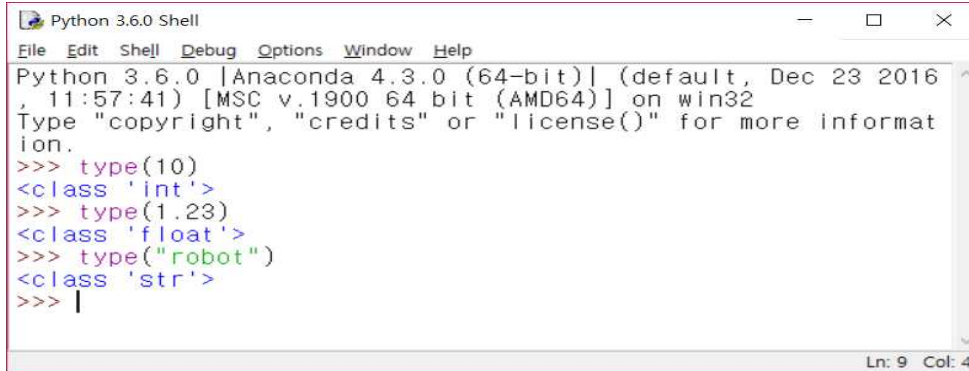
변수에 문자열을 넣으면 그 값을 출력한다.

예제) a="Hello"
print(a)

결과) Hello

1-2 자료형

파이썬도 다른 언어와 같이 자료형이 있다. 그러나 C나 자바와는 다르게 자료형의 사용이 매우 유연하다.



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 [Anaconda 4.3.0 (64-bit)] (default, Dec 23 2016
, 11:57:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more informat
ion.
>>> type(10)
<class 'int'>
>>> type(1.23)
<class 'float'>
>>> type("robot")
<class 'str'>
>>> |
```

그림 18 자료형의 확인

위 그림을 보면 10은 정수 1.23은 실수, robot은 문자열인 것을 알 수 있다.

항목	사용 예
정수	1,2,-4
실수	123.45, -124.5
복소수	1+2j, -3j
8진수	0o34,0o25
16진수	0x2A,0xFF

표 2 파이썬 자료형의 예

1-2-1 리스트

리스트는 자료형을 하나로 묶을 수 있게 도와준다. 아래 예제는 1,3,5,7,9를 하나로 묶을 수 있다. 리스트는 대괄호([])를 이용하여 표시한다.

예제) odd=[1,3,5,7,9]

1-2-2 튜플

튜플은 리스트와 거의 같지만 튜플은 한번 만들면 다시 바꿀 수 없다. 이 예제는 소괄호(())로 이용한다.

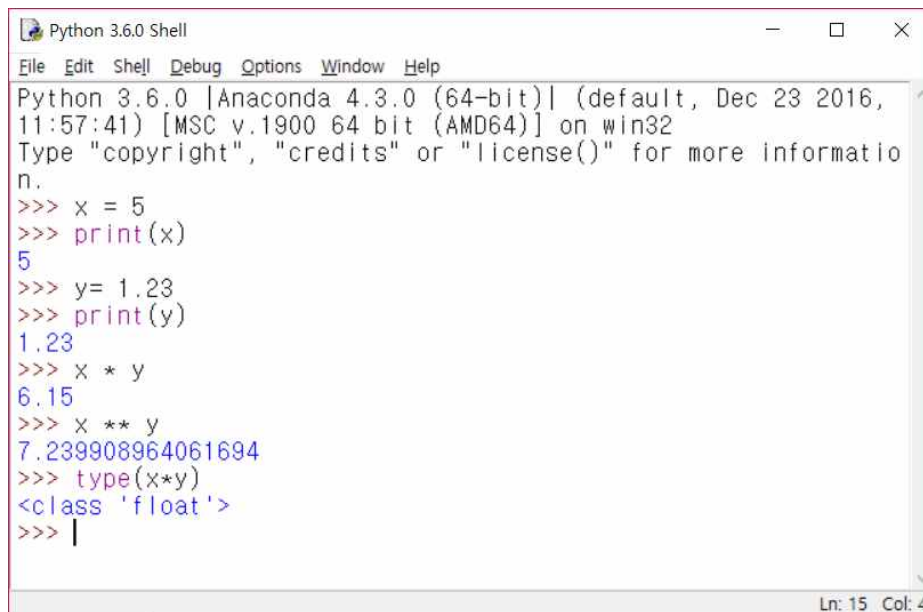
예제) tu=(3,5)

1-2-3. 딕셔너리

딕셔너리는 대응관계를 이용한 자료형이다. 중괄호({})를 이용하여 표시한다

예제) a={1:"hi",2:"hello",3:"help"}

1-3 변수



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 [Anaconda 4.3.0 (64-bit)] (default, Dec 23 2016, 11:57:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> x = 5
>>> print(x)
5
>>> y= 1.23
>>> print(y)
1.23
>>> x * y
6.15
>>> x ** y
7.239908964061694
>>> type(x*y)
<class 'float'>
>>> |
```

그림 19 파이썬 변수 사용 예

모든 프로그래밍 언어는 변수를 사용할 수 있다. 변수는 값을 대입하여 계산 등에 사용할 수 있다.

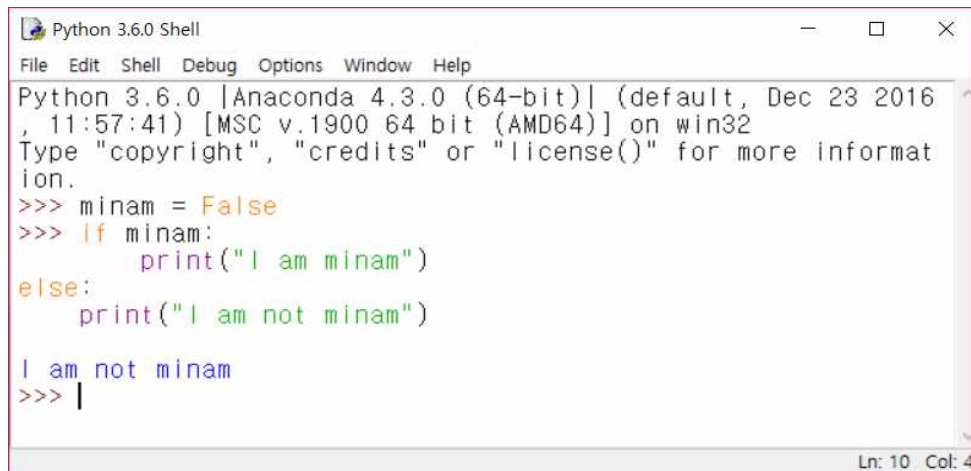
파이썬은 동적언어이다. 동적언어는 변수의 자료형이 계산의 결과나 입력 값에 따라 자동으로 변환하는 것이다. 따라서 파이썬은 변수를 사용할 때 C언어처럼 먼저 변수형을 선언해 줄 필요가 없다.

1-4 조건문 if

조건문은 다른 언어와 매우 비슷하다. 파이썬도 C나 자바의 문법 영향을 받았으므로 그 형태가 유사한 것이 많다.

```
예제) a=3
      if a>1:
          print("a is greater than 1")
```

결과) a is greater than 1



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 [Anaconda 4.3.0 (64-bit)] (default, Dec 23 2016
, 11:57:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more informat
ion.
>>> minam = False
>>> if minam:
>>>     print("I am minam")
else:
>>>     print("I am not minam")

I am not minam
>>> |
```

그림 20 조건문의 사용

1-5 반복문 for

다음은 for를 이용해서 [1,2,3]안의 값을 하나씩 출력하는 것을 보여주는 예이다.

```
예제) for a in [1,2,3]:
      print(a)
```

결과) 1
2
3

여기서는 [1,2,3] 이라는 리스트 안의 원소를 하나씩 모두 출력한다. 이때 변수 i는 임시로 리스트의 값을 저장하는 장소가 된다.

1-6 라이브러리 추가

예제) import sys

사용을 하면 sys.py라는 파일에서 라이브러리나 함수등을 불러올 수 있다.

1-7. 사용자 정의 함수

파이썬에서 함수를 정의하는 방법은 def라는 함수를 이용해서 할 수 있다.

이 예제는 sum이라는 함수를 만들고 출력하는 예제이다.

예제) def sum(a,b):

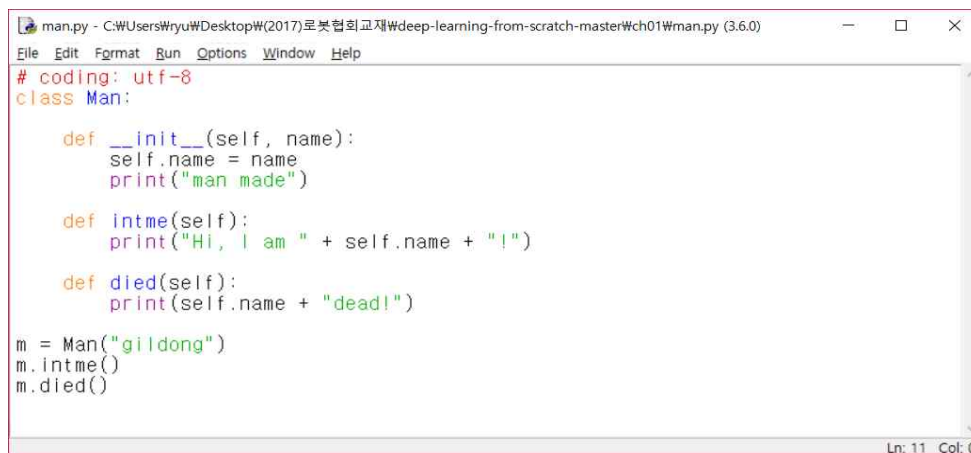
```
    return a+b
```

```
print((sum(3,4))
```

결과) 7

1-8. 클래스

파이썬은 기본적으로 객체지향 프로그래밍 언어로서 클래스를 지원한다. 실제 파이썬에는 모든 자료형이 일종의 클래스이다. 파이썬에서는 클래스라는 키워드를 이용하여 클래스를 정의한다. 다음은 Man 클래스를 정의하고 객체를 생성하여 사용하는 모습을 보여 준다.



```
man.py - C:\Users\Wryu\Desktop\2017로봇협회교재\deep-learning-from-scratch-master\ch01\man.py (3.6.0)
File Edit Format Run Options Window Help
# coding: utf-8
class Man:

    def __init__(self, name):
        self.name = name
        print("man made")

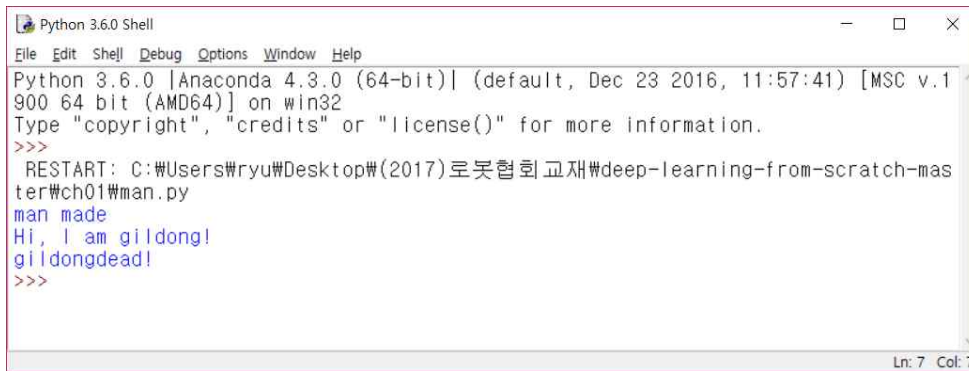
    def intme(self):
        print("Hi, I am " + self.name + "!")

    def died(self):
        print(self.name + "dead!")

m = Man("gildong")
m.intme()
m.died()
```

그림 21 클래스 생성

위 프로그램을 실행한 결과는 다음과 같다.

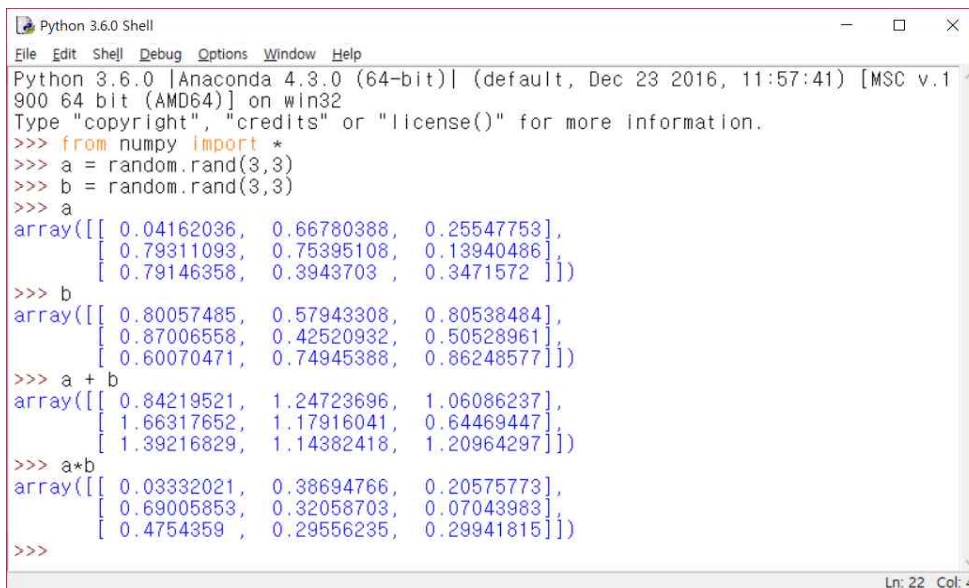


```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 [Anaconda 4.3.0 (64-bit)] (default, Dec 23 2016, 11:57:41) [MSC v.1
900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users#ryu\Desktop#\2017)로봇협회교재#deep-learning-from-scratch-mas
ter#ch01#man.py
man made
Hi, I am gildong!
gildongdead!
>>>
```

그림 22 클래스 실행

1.9 numpy(넘파이)

머신러닝을 파이썬을 이용해서 구현할 때는 행렬 계산을 많이 한다. 이때 넘파이를 사용하면 행렬계산을 쉽게 할 수 있다.



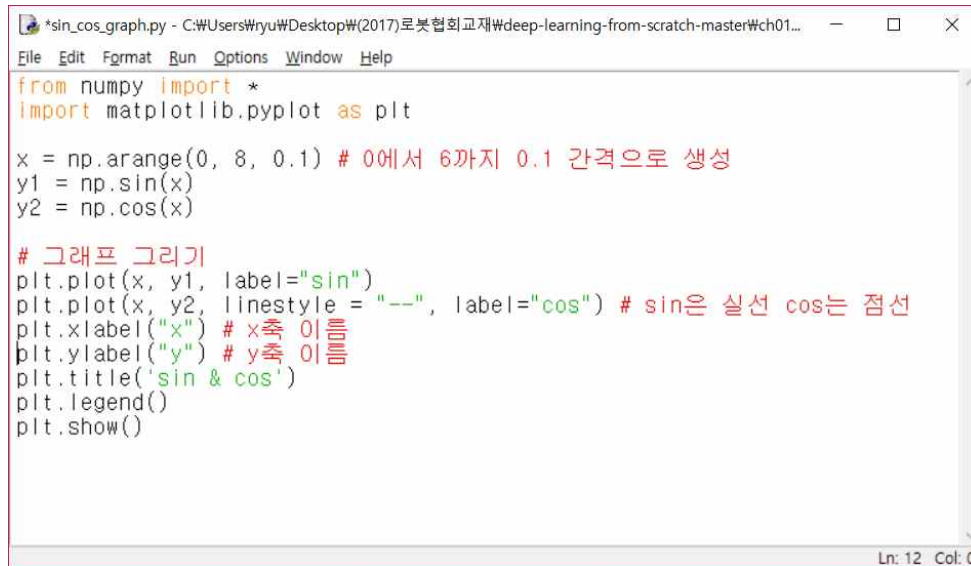
```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 [Anaconda 4.3.0 (64-bit)] (default, Dec 23 2016, 11:57:41) [MSC v.1
900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from numpy import *
>>> a = random.rand(3,3)
>>> b = random.rand(3,3)
>>> a
array([[ 0.04162036,  0.66780388,  0.25547753],
       [ 0.79311093,  0.75395108,  0.13940486],
       [ 0.79146358,  0.3943703 ,  0.3471572 ]])
>>> b
array([[ 0.80057485,  0.57943308,  0.80538484],
       [ 0.87006558,  0.42520932,  0.50528961],
       [ 0.60070471,  0.74945388,  0.86248577]])
>>> a + b
array([[ 0.84219521,  1.24723696,  1.06086237],
       [ 1.66317652,  1.17916041,  0.64469447],
       [ 1.39216829,  1.14382418,  1.20964297]])
>>> a*b
array([[ 0.03332021,  0.38694766,  0.20575773],
       [ 0.69005853,  0.32058703,  0.07043983],
       [ 0.4754359 ,  0.29556235,  0.29941815]])
>>>
```

그림 23 넘파이를 이용한 행렬 계산

위 프로그램은 랜덤 함수를 이용하여 3,3 행렬 두 개를 생성하고 두 행렬에 대하여 덧셈 연산과 곱셈연산을 실행한 모습을 보여 준다.

1.10 matplotlib

머신러닝 실험에서는 그래프 그리기와 자료의 시각화도 매우 중요하다. matplotlib는 그래프를 그려주는 외부 라이브러리 이다.



```
*sin_cos_graph.py - C:\Users\Wryu\Desktop\#(2017)로봇협회교재\#deep-learning-from-scratch-master\#ch01...
File Edit Format Run Options Window Help
from numpy import *
import matplotlib.pyplot as plt

x = np.arange(0, 8, 0.1) # 0에서 6까지 0.1 간격으로 생성
y1 = np.sin(x)
y2 = np.cos(x)

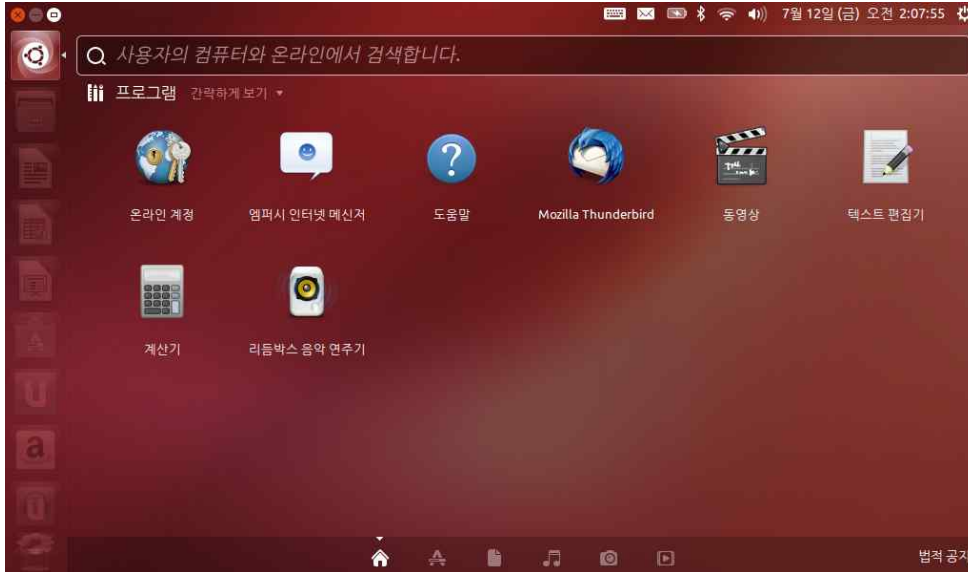
# 그래프 그리기
plt.plot(x, y1, label="sin")
plt.plot(x, y2, linestyle = "--", label="cos") # sin은 실선 cos는 점선
plt.xlabel("x") # x축 이름
plt.ylabel("y") # y축 이름
plt.title('sin & cos')
plt.legend()
plt.show()

Ln: 12 Col: 0
```

그림 24 사인과 코사인 그래프 시각화

개발환경 설정하기

1. 리눅스 사용하기



<대단원 개요>

이 단원에서는 머신러닝 학습에 필요한 기본 환경인 리눅스 시스템에 대하여 학습합니다. 리눅스에서만 머신러닝을 학습하고 개발할 수 있는 것은 아니지만 라이브러리가 풍부하고 실제 현장에서도 머신러닝 시스템을 개발하는데 많이 활용 하고 있다.

01 리눅스 설치하기

학습목표

- 머신러닝 프로그램 개발을 위해 기본적인 리눅스 명령을 사용할 수 있다.
- 머신러닝 프로그램 개발을 위해 리눅스 시스템을 구축할 수 있다.

1. Ubuntu 리눅스



그림 25 Ubuntu 리눅스 로고

1.1 Ubuntu 사용 이유

Ubuntu는 개인 서버를 구축하기 위한 운영체제이다. 리눅스는 다양한 배포판을 가지고 있는데 그중 근래에는 Ubuntu와 CentOS등을 많이 사용한다. CentOS보다 Ubuntu를 사용하는 이유는 Unity 계열의 인터페이스가 더 좋기 때문이다.

Ubuntu의 개발은 캐노니컬의 CEO인 마크 셔틀워스가 시작했으며 현재 Ubuntu의 개발 주체는 Ubuntu재단이다. Ubuntu의 구성 요소는 여러 개발자들이 팀을 이루어 만들고 있으며 Ubuntu 재단이 이를 합쳐 사용하기 쉽게 만드는 역할을 하고 있다.

Ubuntu는 배포판에 대한 수정, 편집, 재배포가 합법적인 자유 소프트웨어로, 지금까지 수많은 변형 배포판이나 공식 지원하지 않는 창 관리자를 데스크톱으로 하는 배포판들이 나왔다. Ubuntu의 기본적인 철학, 즉 전 세계의 사람 누구나 어렵지 않게 리눅스를 사용하자는 표어에서 알 수 있듯이, 기본적으로 세계의 다양한 언어를 지원하고 그다지 높은 사양의 컴퓨터가 필요하지 않다.

Ubuntu는 사용자가 손쉽게 운영 체제를 설치하고 사용할 수 있도록 설

유니티(unity)

유니티는 게임 엔진 기술이자 통합개발환경(Integrated Development Environment, IDE)이다. 이 기술을 처음 만든 이는 데이비드 헬가손(David Helgason), 니콜라스 프랜시스(Nicholas Francis), 요하킴 안테(Joachim Ante)이다. 최근 가장 큰 화제를 모은 게임을 꼽자면 단연 '포켓몬 고'를 빼놓을 수 없다. '포켓몬 고'를 실행하면 게임 첫 화면에 'UNITY 3D'라는 로고를 볼 수 있다.

계되었다. 이를테면 7.10의 경우, CD를 넣고 시동한 다음, 일곱 단계만 거치면 바로 사용할 수 있도록 설치된다. Dapper(6.06) 버전부터 데스크톱 CD에는 Ubiquity가 들어 있어 컴퓨터의 재시동 없이 데스크톱 CD를 구동하는 동안 Ubuntu를 시스템에 설치할 수 있는 기능이 있다. 또한, 아직까지 그래픽 카드에 따라 데스크톱 관리자가 시동이 되지 않는 경우가 많기 때문에, 텍스트 기반에서 설치할 수 있는 얼터너티브 CD를 따로 배포한다. 버전 11.04부터는 유니티가 기본 데스크탑으로 탑재된다.

또한 데스크톱 CD 자체에서 처음 실행할 때 'Ubuntu 체험하기' 메뉴를 통해 라이브 CD 상태로 사용할 수 있다. (6.06 이전 버전에서는 설치와 라이브 CD가 분리되어 있었다.) 라이브 CD를 통해서 컴퓨터의 하드디스크를 건드리지 않고, CD를 통해 부팅하여 Ubuntu를 어느 정도 이용할 수 있다. (이를테면 파이어폭스를 통한 웹 브라우징이나 같이 포함되는 여러가지 게임 등을 실행할 수 있으며, 데이터는 저장되지 않는다) 이 상태에서 설치 아이콘을 클릭하면 간단히 설치할 수 있다. USB 플래시 드라이브를 이용하여 라이브USB를 만들 수도 있으며, 라이브CD와 동일하게 USB만으로 Ubuntu를 사용할 수 있으며, 라이브USB 생성 시 설정에 따라 사용 데이터를 USB에 저장할 수 있다.

또한 시스템 관리 작업에서 sudo 도구를 사용한다는 점이 있다. sudo는 사용자가 자신의 암호를 이용해 시스템 관리 권한을 얻어 작업을 진행할 수 있도록 인증한다. 따라서 관리 작업을 하기 위해 root 사용자의 암호를 따로 만들지 않아도 되고, 여러 사용자가 관리를 위해 root 암호를 공유함으로써 생길 수 있는 잠재적인 보안문제를 예방할 수 있다. Ubuntu에서는 1회 sudo 인증이 완료되면 기본적으로 5분 동안 다시 인증하지 않고도 시스템 관리를 계속할 수 있다. 또한, '제한된 장치 관리자'를 사용할 수 있는데, 이것을 통해 사용자는 드라이버를 번거롭게 직접 설치하는 과정을 거칠 필요 없이, 버튼을 몇 번 누르는 것으로 장치를 설정할 수 있다. 7.10 이후부터는 자동으로 제한된 장치 드라이버를 검색해 적용해준다. 그래픽 카드 드라이버의 경우 설치하면 기능이 비정상적으로 작동할 가능성이 있기 때문에 사용자에게 드라이버를 설치할 것인지를 물어본다.

Ubuntu는 기본 시스템 도구 뿐 만 아니라 LibreOffice(구 버전의 오픈 오피스를 대체), 파이어폭스 웹브라우저, GIMP (10.04부터는 기본으로는

미설치, 공식 저장소로서 다운로드 가능) 그림 편집기, 에볼루션(메일/일정 편집기), 엠펜시(메신저), Gwibber(마이크로블로그 관리)도 함께 기본으로 제공한다. 장애인을 위한 도우미 기능과 국제화(다국어, 유니코드 지원) 관련 기능도 빼놓지 않았다. 7.04 버전부터는 저작권이 있는 파일 포맷에 대해 사용자가 동의하면 자동으로 설치해 주는 기능이 추가되었다.

여기서는 Ubuntu를 사용하기 위해서 Window OS위에서 Ubuntu를 설치하여 Linux를 사용할 것이기 때문에 먼저 VMWare를 설치한다.

1.2 VMWare 설치

<http://www.VMWare.com/kr.html>

위에 링크에 접속하면 시험용으로 사용할 수 있는 VMWare를 다운로드 받을 수 있다.



그림 26 VMWare 사이트

위에 사진 왼쪽 상단 위에 있는 다운로드 아이콘을 클릭하면 무료 제품 평가판을 내려 받을 수 있다.



그림 27 VMWare 내려 받기

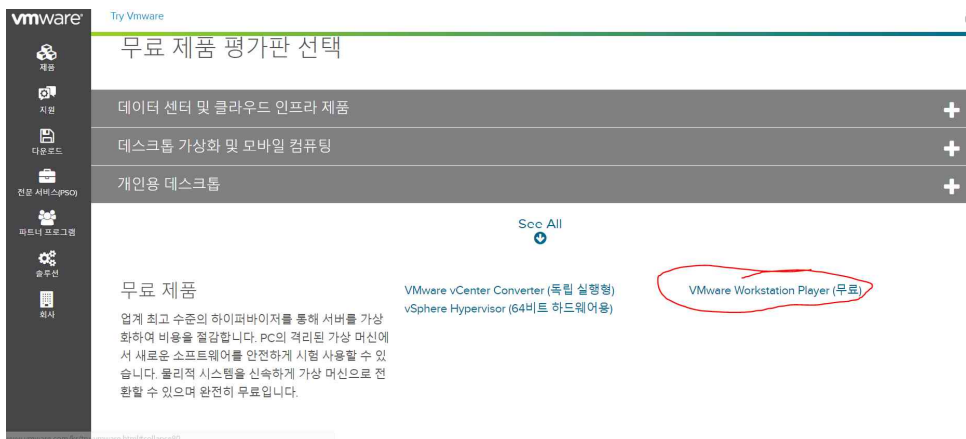


그림 28 VMWare 내려 받기 2

VMWare workstation PLayer(무료)를 선택하면 아래 사진과 같이 Window용 하고 Linux용이 나오는데 여기서 Window를 기반으로 사용할 것 때문에 빨간색 원이 있는 window64bit를 선택한다.

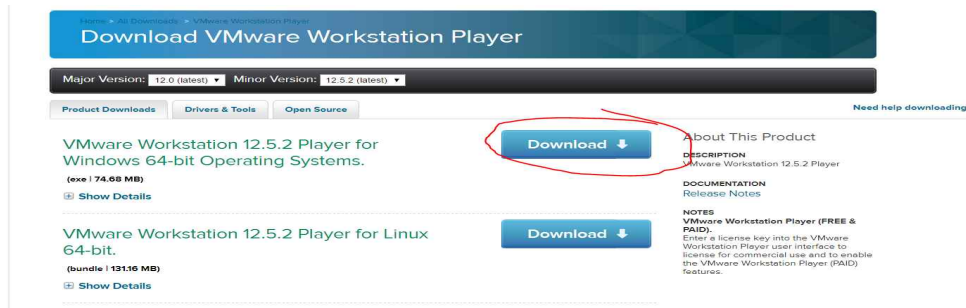


그림 29 VMWare 버전 선택



그림 30 VMWare 설치 과정

VMWare를 올바르게 설치하면 아래 사진처럼 인터페이스가 나온다.

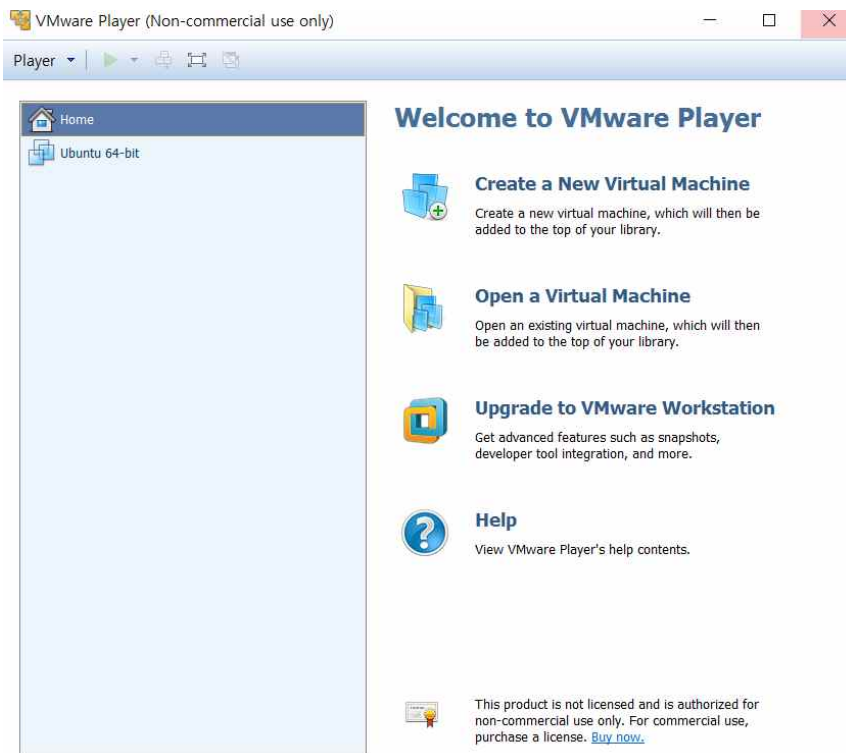


그림 31 VMWare 설치 완료

여기까지 나오면 1단계는 성공한 것이다. 다음 단계로 진행한다.

1.3 Ubuntu 리눅스의 설치

https://www.ubuntu.com/ 링크에 들어가면 Ubuntu 홈페이지에 접속할 수 있다. 접속한뒤 우측 상단 위 download를 선택한다.

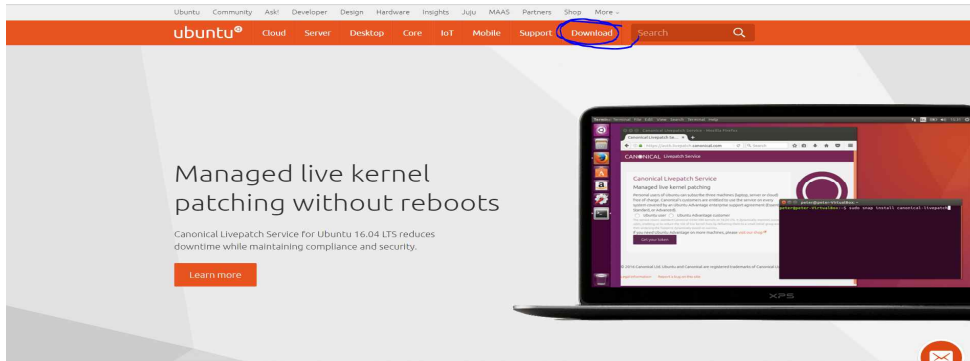


그림 32 Ubuntu 리눅스 홈페이지

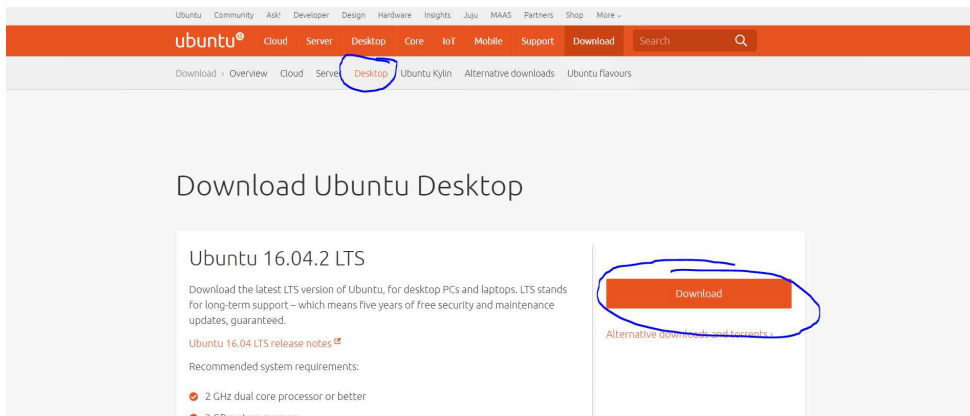


그림 33 Ubuntu 리눅스 다운로드

Ubuntu는 비용지불을 하지 않는다. 다만 기부는 받는데 여기서는 무료로 받기로 한다. 무료로 받기 위해선 바를 0원으로 하고 ISO파일을 받는다.

다운 받은 파일을 설치하는 것은 어렵지 않다. 먼저 VMWare를 실행하고 VMWare에서 새로운 가상머신을 만들고 다운로드 받은 ISO파일을 연결해준다.

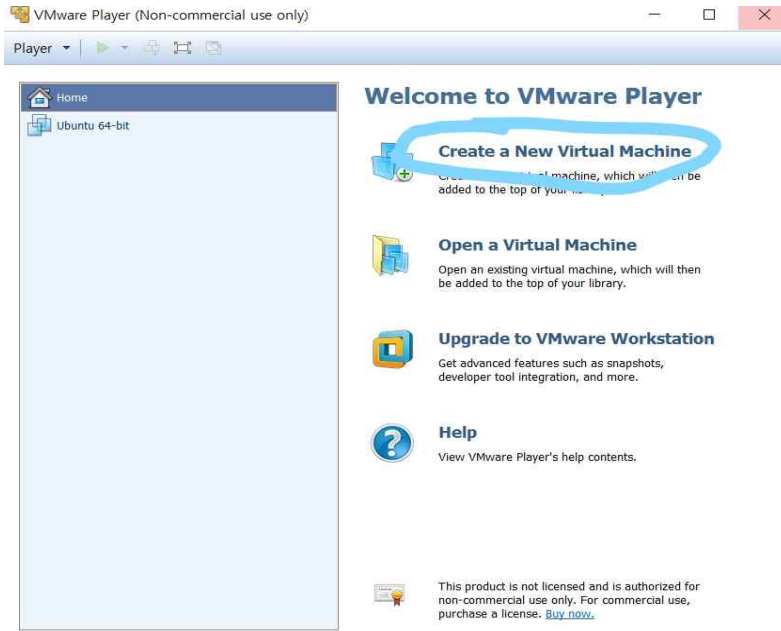


그림 34 VMWare에 Ubuntu 리눅스 설치

먼저 1단계로 VMWare 인터페이스 우측에 Create a New Virtual Machine 바를 선택한다.

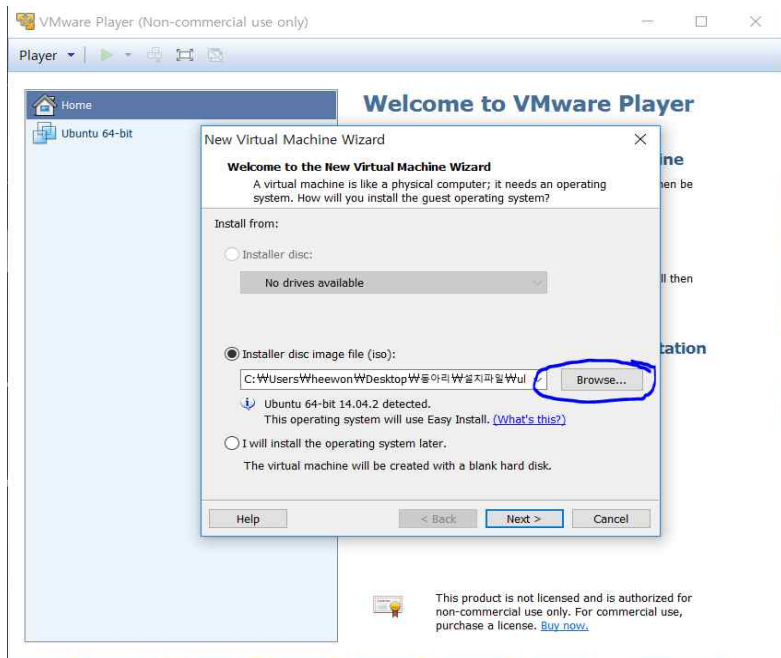


그림 35 VMWare를 이용한 리눅스 설치 2

선택하면 아래 사진과 같은 인터페이스가 나오는데 Browse를 선택하여 다운받은 Ubuntu iso파일을 선택하고 Next를 클릭한다.

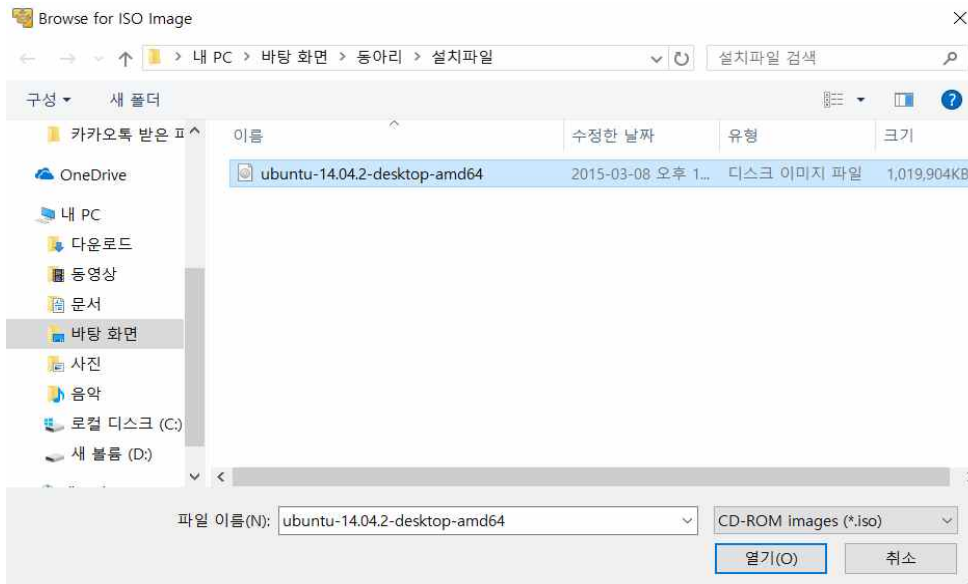


그림 36 VMWare를 이용한 리눅스 설치 3

위에 사진과 같이 Ubuntu에 사용할 아이디와 패스워드를 기입한다.

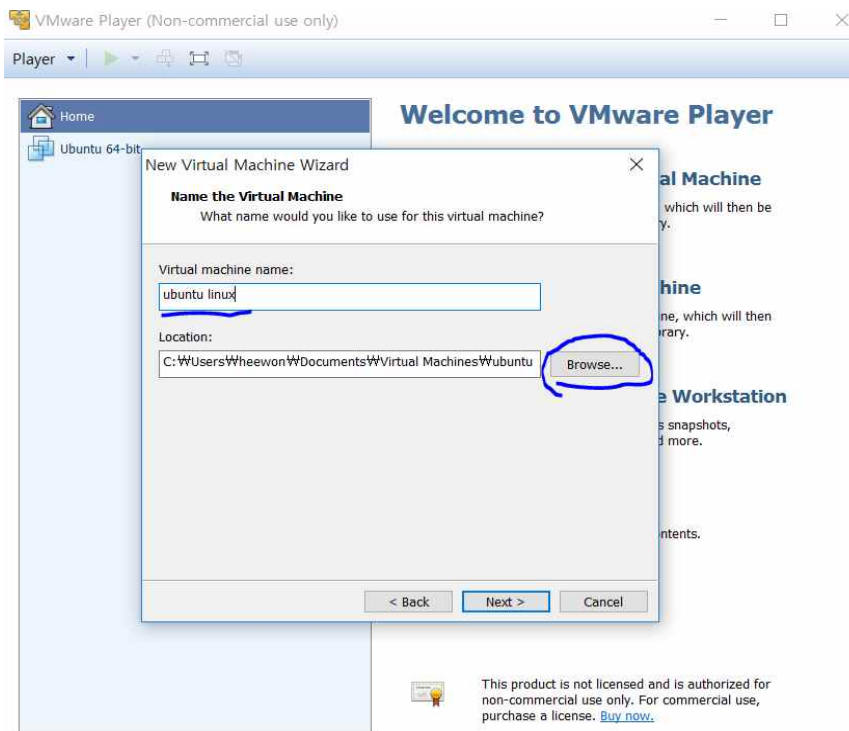


그림 37 VMWare를 이용한 리눅스 설치 4

기입한 후 Next를 클릭하면 사진과 같이 Virtual Machine Name과 Location의 Browse가 있는 인터페이스가 나오는데 Name은 사용할 이름을 작성하고 Location의 Browse를 선택하여 저장할 공간을 설정해주면 된다.

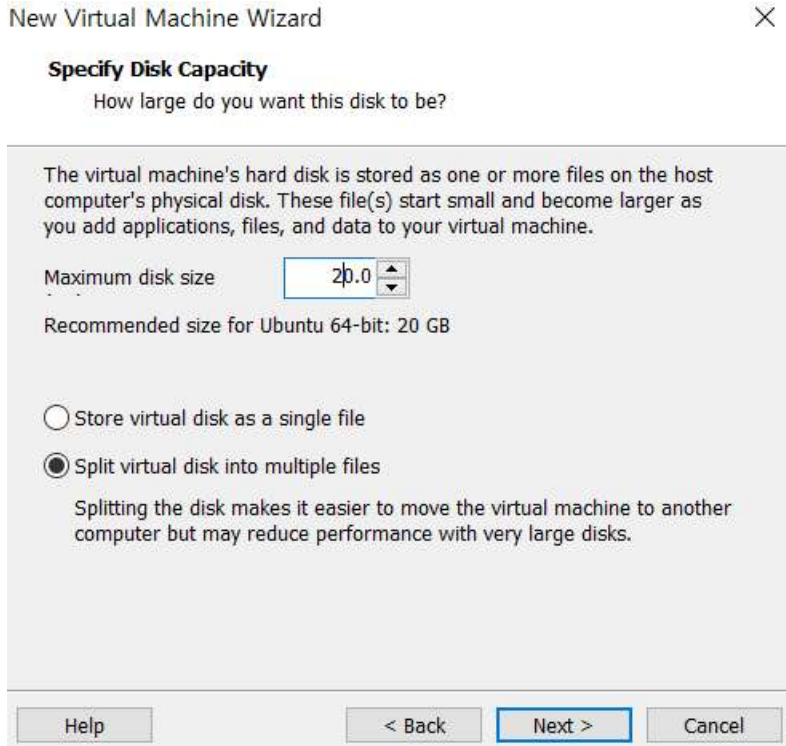


그림 38 가상 머신의 사용 용량 설정

다음 화면에서는 Virtual Machine의 용량을 설정할 수 있는 인터페이스가 나온다. 통상 머신러닝 실험을 위해서는 20GB정도면 충분하니 20GB로 설정한다.

설정해주고 Next를 클릭해주면 아래 사진과 같은 인터페이스가 나오는데 Customize Hardware를 클릭해 준다.

그러면 가상 머신의 세부 설정을 할 수 있는 창이 나온다.

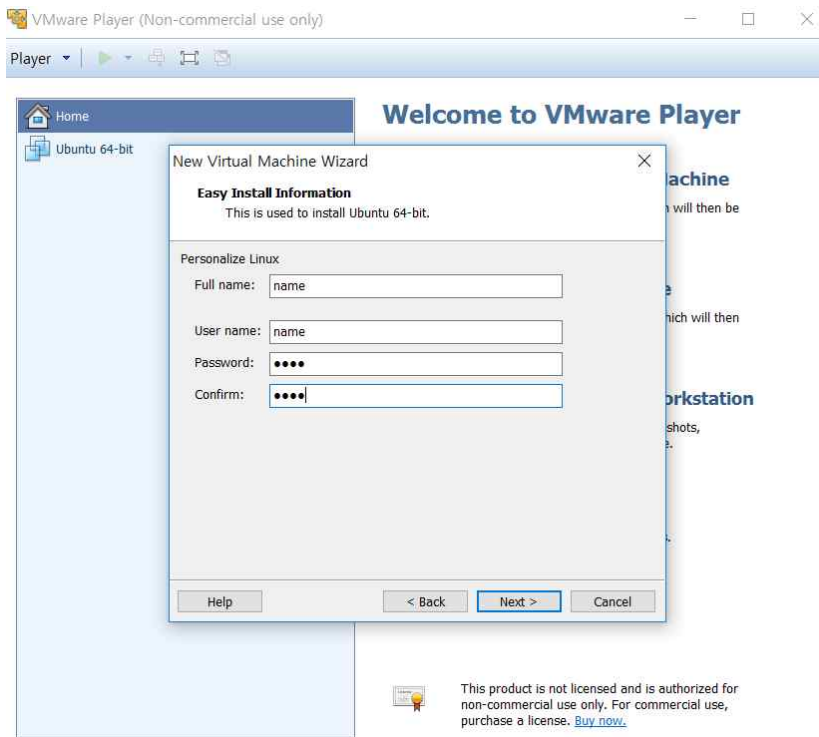


그림 39 가상 머신의 세부 설정

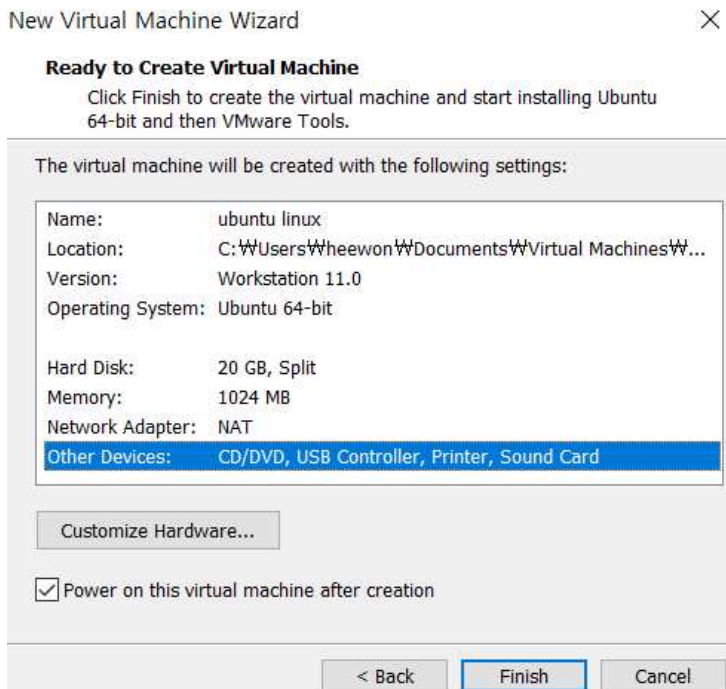


그림 40 가상 머신의 주변 장치 설정

사진에 있는 인터페이스에 Memory는 Ram의 할당을 설정해 주는 것이다.

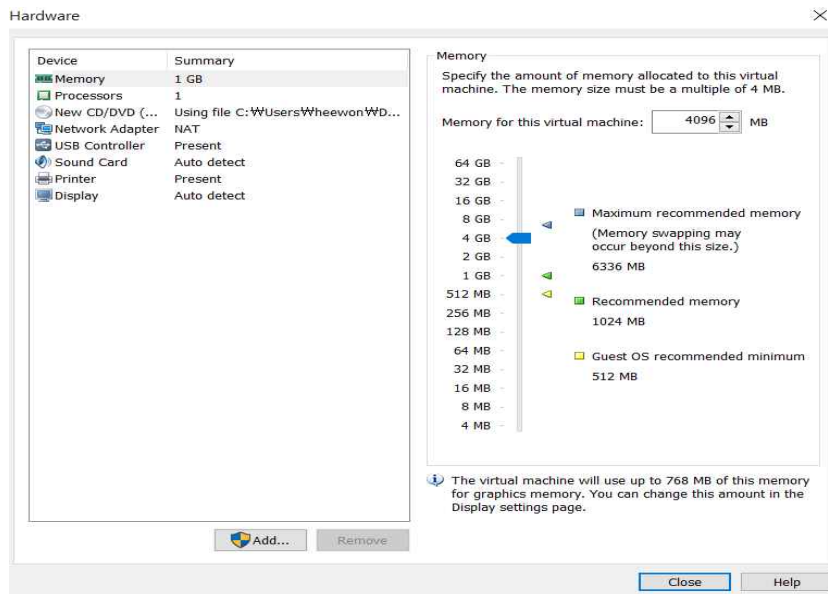


그림 41 가상 머신의 메모리 설정

Ram은 컴퓨터에 따라 적당히 잡아주는 것이 좋은데 Ram이 충분히 있다면 8GB이상 잡아주는 것이 좋고 충분하지 않다면 설치된 메모리의 반을 잡아준다.

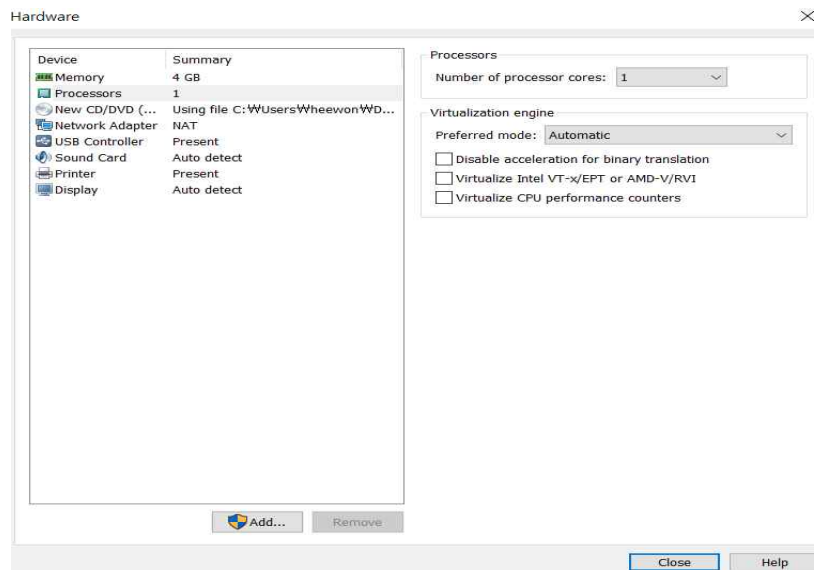


그림 42 프로세서 설정

다음은 Process의 개수를 지정해 주는 건데 이것 또한 사용자 컴퓨터에 맞게 지정해 준다.

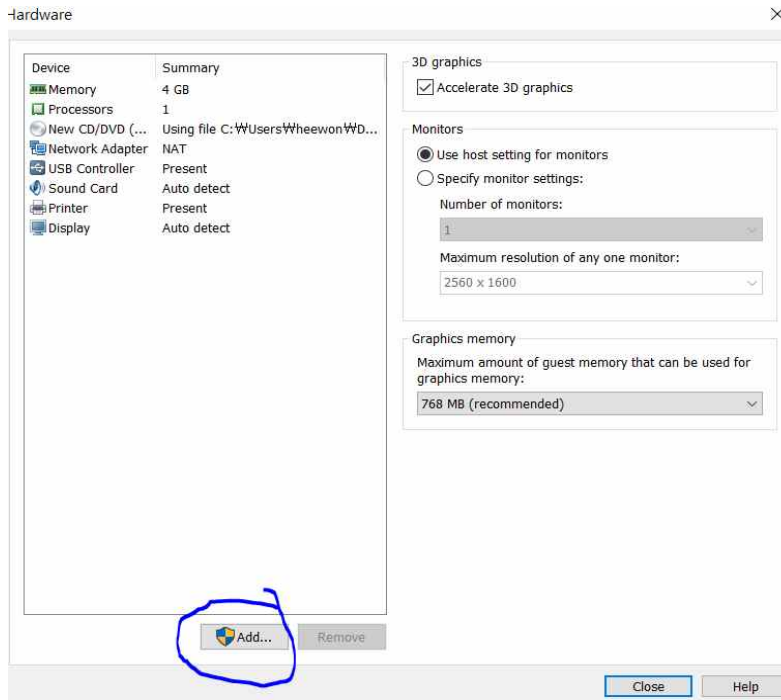


그림 43 가상 머신을 위한 주변장치 설정

그 다음은 필요에 따라 주변장치를 설정해주면 된다.

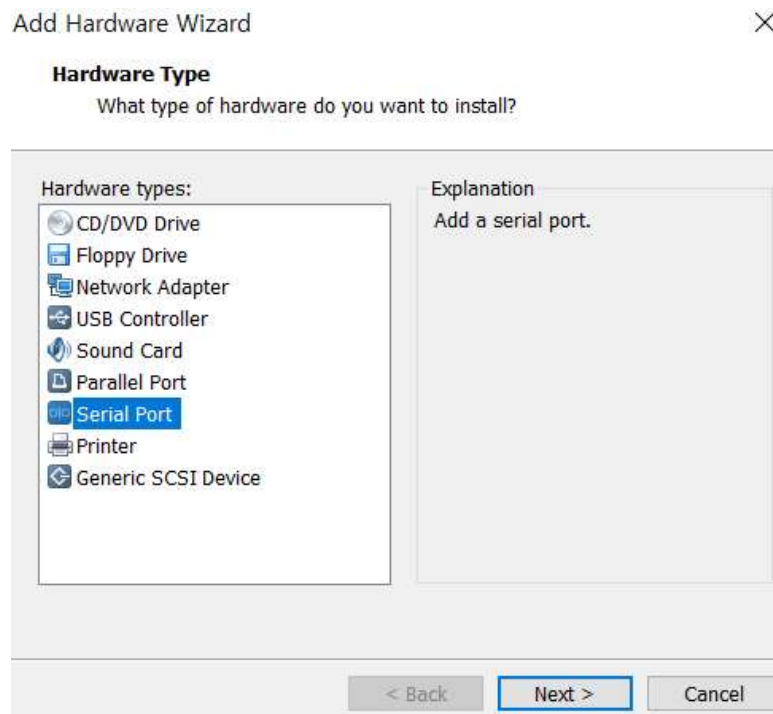


그림 44 시리얼 통신 설정

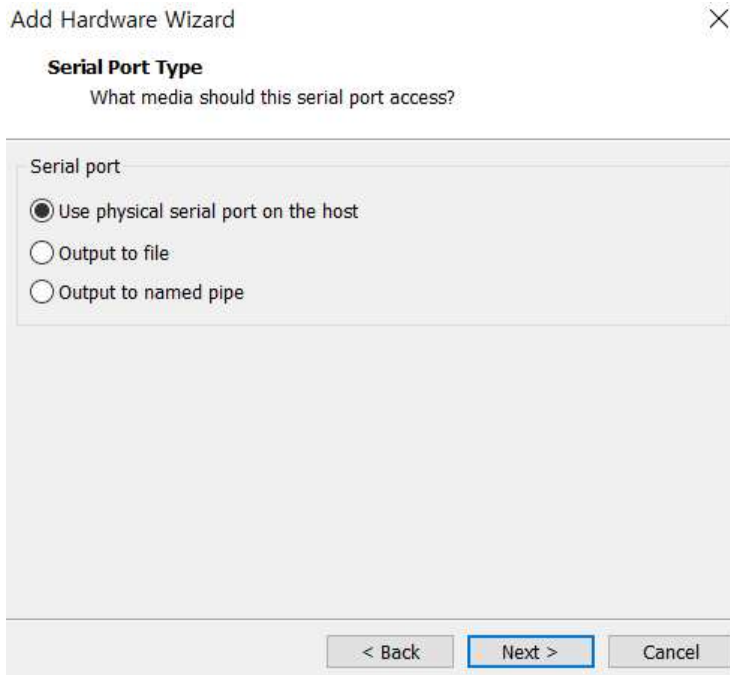


그림 45 시리얼 통신 설정 2

여기서는 시리얼 포트를 설정해보자. Serialport를 추가로 설정해주고 close, finish를 눌러주면 설치는 종료된다.

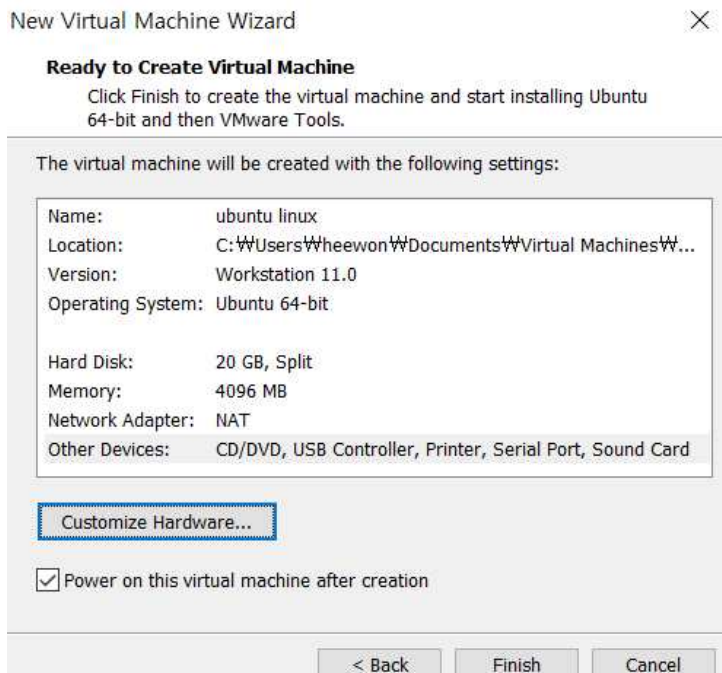


그림 46 하드웨어 설정 종료

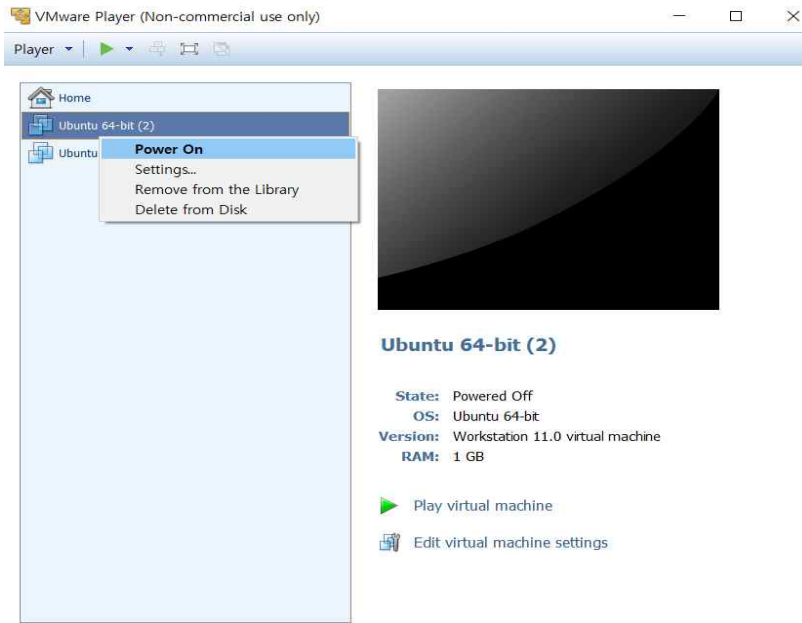


그림 47 가상 머신의 제거

Virtual Machine을 삭제하는 방법은 우측 클릭한 후 Delete From Disk를 클릭해주면 된다. 설치 후 첫 인터페이스는 아래와 같은 화면을 볼 수 있다.

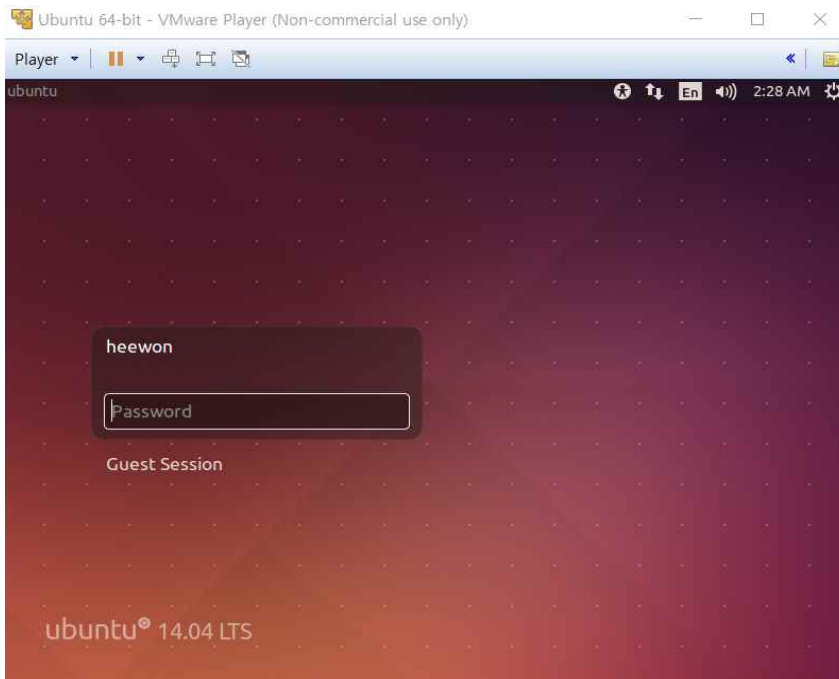


그림 48 VMWare를 통한 로그인

02 리눅스 사용하기

학습목표

- 간단한 리눅스 명령어를 사용할 수 있다.
- 기본적인 리눅스 명령어를 이용하여 머신러닝 응용 프로그래밍을 작성하는데 활용할 수 있다.

1. Linux기초 명령어

기본적인 리눅스이 사용을 위해 반드시 알아두어야 할 기본 명령어는 다음과 같다.

ls	<p>파일 리스트 보기(dir)</p> <p>F : 파일 유형을 나타내는 기호를 파일명 끝에 표시 (디렉토리는 '/', 실행파일은 '*', 심볼릭 링크는 '@'가 나타남). l :파일에 관한 상세 정보를 나타냅니다. a:dot파일(.access 등)을 포함한 모든 파일 표시. t :파일이 생성된 시간별로 표시 C:도스의 dir/w명령과 같 이 한줄에 여러개의 정보를 표시 R:도스의 dir/s 명령과 같이 서브디렉토리 내용까지.</p> <p>(예) #ls-al #ls-aC #ls-R</p>
cd	<p>디렉토리를 변경</p> <p># cd cgi-bin : 하부 디렉토리인 cgi-bin으로 들어감. #cd .. : 상위 디렉토리로 이동 #cd 또는 cd ~ : 어느 곳에서든지 자기 홈디렉토리로 바로 이동 #cd/webker : 현재 작업중인 디렉토리의 하위나 상위 디렉토리가 아닌 다른 디렉토리(webker)로 이동하려면 /로 시작해서 경로이름을 입력하면 된다.</p>

cp	<p>파일 복사(copy)</p> <pre># cp index.html index.old : index.html 파일을 index.old 란 이름으로 복사.</pre> <pre># cp/home/test/*.* . : test디렉토리내의 모든 파일을 현재 디렉토리로 복사.</pre>
mv	<p>파일이름(reName) / 위치(move)변경</p> <pre># mv index.htm index.html : index.htm 파일을 index.html 로 이름 변경</pre> <pre>\$ mv file ../main/new_file : 파일의 위치변경</pre>
mkdir	<p>디렉토리 생성</p> <pre># mkdir download : download 디렉토리 생성</pre>
rm	<p>파일삭제</p> <pre># rm test.html : test.html 파일 삭제</pre> <pre># rm-r<디렉토리> : 디렉토리 전체를 삭제</pre> <pre># rm-ia.* : a로 시작하는 모든 파일을 일일이 삭제할 것인지 확인하면서 삭제</pre>
rmdir	<p>디렉토리 삭제</p> <pre># rmdir cgi-bin : cgi-bin 디렉토리 삭제</pre>

<p>chmod</p>	<p>파일 permission 변경</p> <p>리눅스에서는 각 파일과 디렉토리에 사용권한을 부여. 예) -rwxr-xr-x guestbook.html rwx :처음 3개 문자 = 사용자 자신의 사용 권한 r-x :그다음 3개 문자 = 그룹 사용자의 사용 권한 r-x :마지막 3개 문자 = 전체 사용자의 사용 권한 읽기(read)----- 파일 읽기 권한 쓰기(write)----- 파일 쓰기 권한 실행(execution)----- 파일 실행 권한 없음(-)----- 사용권한 없음 명령어 사용법 chmod[변경모드] [파일] # chmod 666 guestbook.html : test.html 파일을 자신에게만 r,w,x 권한을 줌 # chmod 766 guestbook.html : 자신은 모든 권한을 그룹 사용자와, 전체사용자에게는 읽기와 쓰기 권한만 줌</p>
<p>cat</p>	<p>파일의 내용을 화면에 출력하거나 파일을 만드는 명령(도스의 TYPE명령)</p> <p># cat fileName</p>
<p>vi, touch, cat</p>	<p>새로운 파일을 만드는 방법</p> <p># vi newfile : vi 편집기 상태로 들어감 # touchnewfile :빈 파일만 생성됨 # cat>newfile : vi편집기 상태로 들어감, 문서 작성후 Ctrl+D로 빠져나옴</p>

표 3 기본 리눅스 명령어

더 알아보기: 유닉스(UNIX)



켄 톰슨과 데니스 리치

유닉스(Unix)는 교육 및 연구 기관에서 즐겨 사용되는 범용 다중 사용자 방식의 시분할 운영 체제이다.[1] 1970년대 초반 벨 연구소 직원인 켄 톰슨, 데니스 리치, 더글러스 매클로리 등이 처음 개발하였다. 오늘날의 유닉스 시스템은 AT&T를 비롯한 여러 회사들과 버클리대학(UC Berkeley) 등 비영리 단체들이 개발한 다양한 버전들이 있다. 유닉스는 처음부터 다양한 시스템 사이에서 서로 이식할 수 있고, 멀티 태스킹과 다중 사용자를 지원하도록 설계되었다.

서울시 공공 데이터 수집

머신러닝은 데이터를 기반으로하기 때문에 머신 러닝을 하기 위해서는 당연히 머신러닝을 위한 데이터를 준비해야 한다. 데이터 수입은 데이터 준비 절차의 첫 단계로 어떤 형식의 데이터를 수입해야 할지 어떤 방법으로 데이터를 수입해야 할지를 생각 해야 한다. 수집된 데이터의 품질은 데이터 분석 결과에 큰 영향을 미치며 데이터가 얼마나 잘 정리되어 있는지에 따라서도 전체 데이터 분석에 들어가는 시간과 노력이 좌우되기 때문에 데이터 수입 과정은 데이터 마이닝에서 매우 중요한 단계이다. 여기서는 로봇을 위한 머신러닝의 실전적 학습을 위해 정부, 공공기관에서 수집되어 무료로 제공하는 공공 데이터를 수집하는 방법을 연습해 보기로 한다.



□ 실습 목표

- 정부 또는 공공 기관에서 데이터를 제공하는 사이트를 파악한다.
- 어떤 정보를 어떻게 제공하는지 살펴본다.

□ 실습 과제

- 서울시 공공 데이터 제공 사이트인 서울 열린 데이터 광장에서 제공하는 여러 가지 정보중 서울시 고등학교 정보자료(.xlsx)를 내려 받아 파이썬 코드를 이용하여 서울 용산구에 있는 학교를 모두 출력해 보자.

The screenshot shows a web browser window at the URL `data.seoul.go.kr/openinf/fileview.jsp?infid=OA-12061`. The page displays metadata for a file named '서울교육통계시스템_학교일람표_고등학교현황.xlsx'. The metadata table includes fields such as '제공기관' (Seoul Metropolitan Government), '제3차권자' (None), '담당자/연락처' (Lee Mal-sook / 02-2133-3947), '원본형태' (File), '데이터 공개일자' (2014.12.19), '갱신주기' (Quarterly), '최종수정일' (2014.12.19), '저작권지연락처' (02-2133-3947), '태그' (High school statistics, High school list, etc.), and '제공부서' (Seoul Metropolitan Government Education Planning and Research Institute). Below the metadata, there is a table with one row listing the file name, size (62 KB), and dates. At the bottom, there are social media sharing buttons and a search bar.

No	파일명	파일크기(KB)	마지막수정일	최초공개일	다운로드
1	서울교육통계시스템_학교일람표_고등학교현황.xlsx	62	2014.12.19	2014.12.19	Down

□ 해결 과정

- Step 1: 먼저 위 사이트에서 고등학교 정보 자료를 내려 받는다.
- Step 2: 받은 자료는 엑셀 파일로 되어 있는데 파이썬에서 그대로 처리 가능하지만 여기서는 간단히 csv 파일로 변환하여 저장 한다.
- 저장할 때 주의 할 점은 파일 인코딩을 utf-8로 해주어야 한다.
- 파이썬 코드를 작성하여 용산구에 있는 고등학교 명단을 출력한다.

□ 참고 코드

```
import csv
```

```
input_file_Name = "hsData.txt"
```

```
line_count = 0
```

```
with open(input_file_Name, "r", encoding="utf-8-sig") as input_file:
```

```
    csvReader = csv.DictReader(input_file)
```

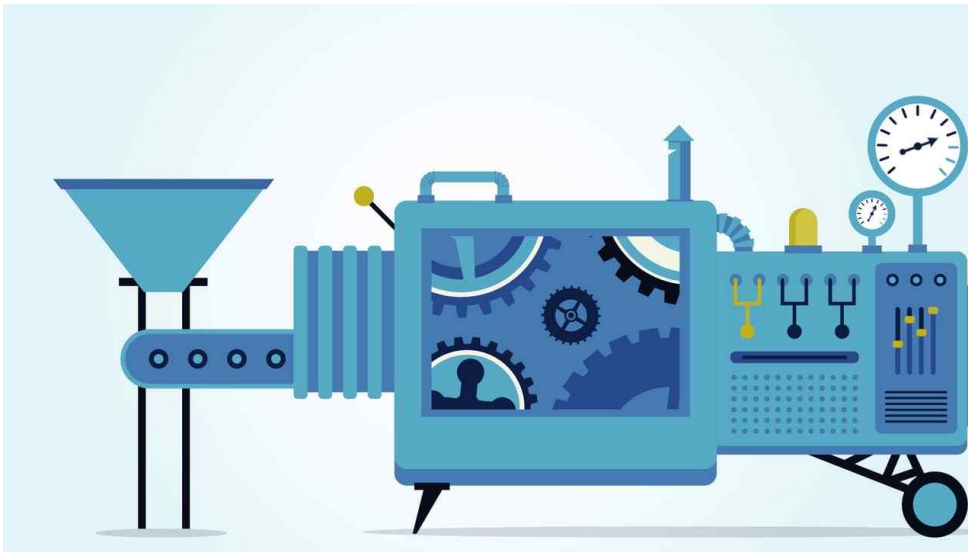
```
    for line in csvReader:
```

```
        if line['소재구'] == '용산구':
```

```
            print(line['소재구'], line['학교명'])
```

머신러닝 기초

1. 머신러닝 알고리즘
2. Weka를 이용한 알고리즘 실습
3. 파이썬을 이용한 ANN 실습



<대단원 개요>

이 단원에서는 기초적인 머신러닝 알고리즘의 이론을 학습하고 기존에 개발된 머신러닝 라이브러리를 이용하여 몇몇의 알고리즘을 실험하여 본다. 나아가 자작 로봇에 사용하기 위하여 파이썬을 이용한 인공신경망을 통해 숫자를 인식하는 실습을 수행 한다.

01 머신러닝 알고리즘

학습목표

- 머신러닝의 기본 알고리즘을 설명할 수 있다.
- 목적에 맞는 머신러닝 알고리즘을 선택할 수 있다.

1. Decision Tree

머신 러닝 알고리즘 중에 제일 처음으로 Decision Tree를 공부해보자. 실제 실험과 구현은 다음 장에서 weka와 파이썬으로 하기로 하고 여기서는 이론을 살펴보기로 한다.

decision tree를 공부하기 위해서 우선 알아야 할 개념은 entropy이다. 우선 이미지로 알아보자. entropy가 높은 경우와 낮은 경우이다.



그림 52 엔트로피의 비교

왼쪽이 entropy가 높은 경우, 오른쪽이 entropy가 낮은 경우이다. entropy가 높다는 말은 그만큼 Data가 지저분하다는 뜻이다. 반대로 entropy가 낮다는 말은 Data가 간단하다는 것이다.

왼쪽에 entropy가 높은 messy Data에는 1부터 3까지의 숫자가 골고루 있다. 반면에 오른쪽에 entropy가 낮은 neat Data에는 1만 있다. 그렇다면 entropy는 어떻게 구하는지 공식을 통해 알아보자. 아래의 식과 같다.

Messy data	Neat data
1	1
1	1
1	1
1	1
1	1
1	1
1	1
1	1
2	1
2	1
2	1
2	1
2	1
2	1
2	1
2	1

$$Entropy(p) = \sum p_i \log \frac{1}{p_i}$$

위 식은 확률을 계산하는 것인데 각 값이 나타날 확률과 그 확률의 로그 값을 곱해서 모두 합하여 -1을 곱하는 식이다. 위의 공식으로 messy Data 와 neat Data의 entropy를 구해보면 아래와 같다.

$$\begin{aligned}
 &Entropy(messy) \\
 &= \frac{4}{14} \log \frac{1}{\frac{4}{14}} + \frac{5}{14} \log \frac{1}{\frac{5}{14}} + \frac{5}{14} \log \frac{1}{\frac{5}{14}} \\
 &= 1.5774062828523452
 \end{aligned}$$

$$\begin{aligned}
 &Entropy(neat) \\
 &= \frac{14}{14} \log \frac{1}{\frac{14}{14}} = 0
 \end{aligned}$$

위 식의 결과는 다소 이상하게 보일 수 있다. 엔트로피 연산을 결과는 0과 1사이의 값을 가진다. 그러나 위식의 결과는 1 이상의 값이다. 일반적으로 possible value가 2개일 경우 즉 엔트로피 공식이 아래와 같이 simplify되는 경우에는 0과 1 사이이다. 그러나 possible value가 3개 이상일 경우에는 entropy가 1 이상일 수 있다.

$$Entropy(p) = p \log \frac{1}{p} + (1 - p) \log \frac{1}{1-p}$$

위에 messy Data와 neat Data의 entropy값을 보면, messy Data는 1.5774062828523452가 된다. 이 Data를 간단하게 나누어 entropy를 낮추는 과정을 진행하면 다음과 같다.

Messy data	Splitted data
1	1
1	1
1	1
1	1
2	2
2	2
2	2
2	2
2	2
3	3
3	3
3	3
3	3
3	3

- 1로 쪼개진 그룹의 엔트로피는 당연히 0
- 2로 쪼개진 그룹의 엔트로피도 당연히 0
- 3로 쪼개진 그룹의 엔트로피도 당연히 0 이다.

1, 2, 3은 각각 4/14, 5/14, 5/14비율로 쪼개졌으니까

$$WeightedAverageofEntropy = \frac{4}{14} \times 0 + \frac{5}{14} \times 0 + \frac{5}{14} \times 0 = 0$$

세 entropy의 Weighted Average을 구하면 0이 된다.

결과적으로 1.5774062828523452라는 entropy가 어떤 기준에 따라 1, 2, 3으로 나누면 entropy(Weighted Average entropy)가 0으로 떨어진다. 그러면 우리는 "아 그 기준에 따라서 우리가 Data를 나누면 뭔가 정보를 많이 얻겠구나?"라고 이해할 수 있다. 이것을 우리가 전문용어로 Information Gain이라고 한다.

$$\text{InformationGain} = \text{EntropyofParent} - \text{EntropyofChild}$$

$$= 1.5774062828523452 - 0 = 0$$

nominal value들에 대해서는 위와 같이 entropy를 구해서 Information Gain을 구할 수 있다. Numeric Data(수치형 자료)들에 대해서는 유사한 방법으로 다음과 같이 구할 수 있다.

x2	class	entropy for x2	Information gain
65	c1	0.8925768472	0.0477091114
70	c1	0.9253298887	0.0149560699
70	c1	0.9253298887	0.0149560699
70	c2	0.9253298887	0.0149560699
75	c1	0.8949517866	0.045334172
78	c1	0.8500096093	0.0902763494
80	c1	0.8380423951	0.1022435636
80	c1	0.8380423951	0.1022435636
80	c2	0.8380423951	0.1022435636
85	c2	0.9152077852	0.0250781735
90	c1	0.9299678578	0.0103181009
90	c2	0.9299678578	0.0103181009
95	c1	0.9402859587	0
95	c2	0.9402859587	0
total entropy			
0.940285959			

entropy for x2를 구한 방법은 아래와 같다. 예를 들어 값이 x2값이 65 일 때는 $x2 \leq 65$ 인 경우의 확률과 $x2 > 65$ 인 경우의 확률을 구해서 entropy를 구하는 것이다.

	c1	c2
≤ 65	1/1	0/1
> 65	8/13	5/13

Weight Average of Entropy

$$= \frac{1}{14} (1 \log \frac{1}{1} + 0 \log \frac{0}{0}) + \frac{13}{14} (\frac{8}{13} \log \frac{1}{13} + \frac{5}{13} \log \frac{1}{13})$$

참고로 x2전체에 대한 entropy는 0.9402859587이므로 이 값에서 각 x2에 대한 entropy를 빼면 아래와 같은 결과가 나온다.

x2	class	entropy for x2	Information gain
65	c1	0.8925768472	0.0477091114
70	c1	0.9253298887	0.0149560699
70	c1	0.9253298887	0.0149560699
70	c2	0.9253298887	0.0149560699
75	c1	0.8949517866	0.045334172
78	c1	0.8500096093	0.0902763494
80	c1	0.8380423951	0.1022435636
80	c1	0.8380423951	0.1022435636
80	c2	0.8380423951	0.1022435636
85	c2	0.9152077852	0.0250781735
90	c1	0.9299678578	0.0103181009
90	c2	0.9299678578	0.0103181009
95	c1	0.9402859587	0
95	c2	0.9402859587	0
total entropy			
0.940285959			

x2가 80일 경우 가장 Information Gain이 높다. 이 원리를 이용해서 다음과 같이 Decision Tree를 만들 수 있다.

우선 머신러닝 Decision Tree 부분에서 가장 많이 사용되는 weather Data를 이용해보자.

outlook	windy	temperature numeric	humidity numeric	play
rainy	TRUE	65	70	no
rainy	TRUE	71	91	no
sunny	FALSE	85	85	no
sunny	TRUE	80	90	no
sunny	FALSE	72	95	no
overcast	FALSE	83	86	yes
overcast	TRUE	64	65	yes
overcast	TRUE	72	90	yes
overcast	FALSE	81	75	yes
rainy	FALSE	70	96	yes
rainy	FALSE	68	80	yes
rainy	FALSE	75	80	yes
sunny	FALSE	69	70	yes
sunny	TRUE	75	70	yes

이것을 가지고 우리는 아래와 같은 Tree를 만들려고 한다.

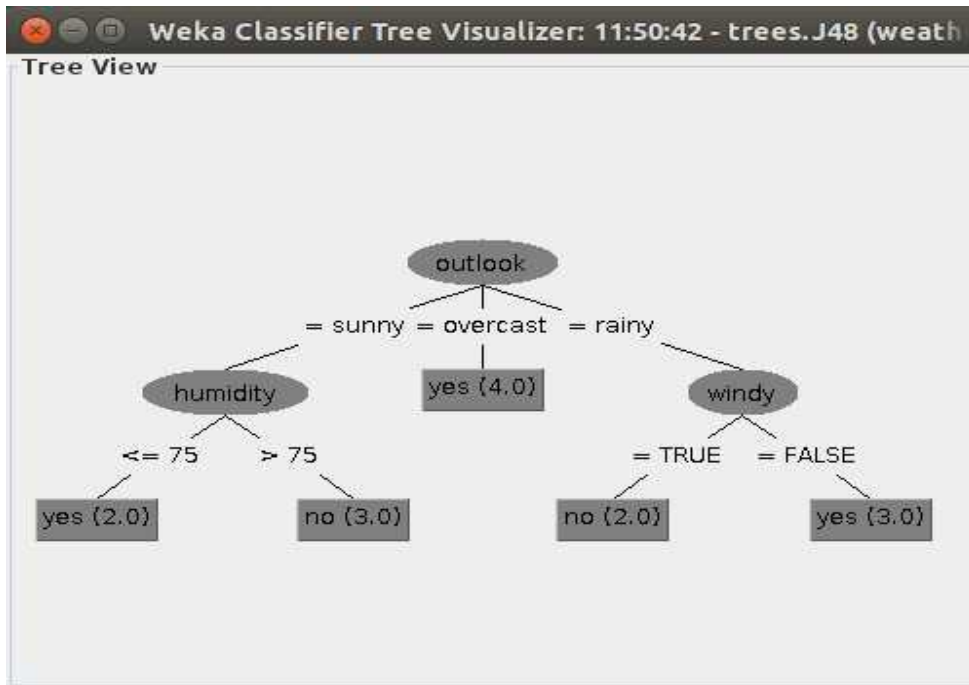


그림 53 날씨 자료를 이용한 의사결정트리

$$\text{WeightAverageof Entropy(Outlook)}$$

$$= \frac{4}{14} \times \text{Entropy}(1) + \frac{5}{14} \times \text{Entropy}\left(\frac{3}{5}\right) + \frac{5}{15} \times \text{Entropy}\left(\frac{2}{5}\right)$$

$$= 0.4807$$

$$\text{WeightAverageof Entropy(Humidity)}$$

$$= \frac{8}{14} \times \text{Entropy}\left(\frac{6}{8}\right) + \frac{4}{14} \times \text{Entropy}\left(\frac{3}{6}\right)$$

$$= 0.6184$$

$$\text{MinimumEntropy(Temperature)} = 0.6186871251$$

$$\text{MinimumEntropy(Humidity)} = 0.5808867233$$

제일 위에 보면 Outlook으로 Data가 나뉜다. 이 때 Information Gain을 구하면 그림과 같이 Outlook이 최상위 root가 됨을 알 수 있다.

지금 우리는 Outlook, Temperature, Humidity, Windy 네 개의 Attribute을 가지고 있다. 이것들의 Information Gain을 구해서 가장 높은 값을 구하는 것이다. 이유는 바로 Information Gain이 높은 값으로 쪼개야 Data가 가장 less messy해지기 때문이다. 위와 같이 구하면 된다.

$$\text{(단, Entropy}(p) = p \cdot \log(1/p) + (1-p) \cdot \log(1/(1-p)))$$

참고로 위의 Minimum Entropy (Temperature)과 Minimum Entropy (Humidity)는 아래와 같이 구했다.

Temperature	Entropy	humidity	Entropy
64	0.6186871251	65	0.6186871251
65	0.6446045986	70	0.6413898035
68	0.6514172870	75	0.6203333076
69	0.6413898035	80	0.5465122115
70	0.6203333076	85	0.6183974457
71	0.6508279225	86	0.5808867233
72	0.6508279225	90	0.5967872362
75	0.6343736959	91	0.6446045986
80	0.6514172870	95	0.6186871251
81	0.6446045986	96	0.6517565612
83	0.5731530719		
85	0.6517565612		

Outlook, Humidity, Temperature, Windy의 Entropy를 구해서 Parent Entropy에서 빼서 가장 큰 Attribute을 택한다.

	Outlook	Temperature	Humidity	Windy
	0.4807	0.61868	0.5808	0.6184
Parent Entropy	0.6518	0.6518	0.6518	0.6518
Parent Entropy - Entropy(each)	0.1711	0.03312	0.071	0.0334

Outlook이 가장 크다. 따라서 Data를 Outlook으로 쪼갠 때 가장 Data가 Certain해진다는 것이다.

그 다음에 해야 할 일은 반복적이다. 각각의 나누어진 것에 대해서 또 똑같이 Information Gain을 구한다. Outlook은 overcast, sunny, rainy 값을 가질 수 있기 때문에 각 값으로 쪼갠 후 각각의 Entropy를 다시 구하면 된다.

Outlook이 Overcast인 경우에는 이미 "Yes" 값 밖에 없다. entropy는 0일 것이고, 충분히 Certain한 값이기 때문에 더 이상 Tree를 쪼개지 않는다.

Outlook이 Sunny일 경우에는 75로 쪼개진다.

Outlook이 Rainy일 경우에는 Windy로 쪼개진다.

방법은 같기 때문에 생략한다.

마지막 Leaf Node들을 보면 모든 값이 No나 Yes를 갖게 된다. 이제 Tree를 쪼갤지 말지 결정하는 기준으로는 아래의 세 가지 기준이 있다.

1. 모든 Data가 같은 Class를 가질 경우
2. 모든 Data의 Attribute가 같을 경우
3. Data가 하나 밖에 없을 경우

그런데 문제가 있다. Weather Data의 경우에는 크기가 작기 때문에 위의 Tree와 같이 깔끔하게 나누어지는데, Data Set이 커지면 트리도 같이 커질 확률이 높다. 왜냐하면 Decision Tree는 위의 세 가지 조건이 만족할 때까지 Tree를 계속 쪼갤 것이기 때문이다. 이렇게 해서 만들어진 Tree는 너무 Training Data에 잘 맞아 떨어져서 Test Data에는 잘 맞지 않게 된다. 즉, 너무 자세하게 만들어진 Tree라는 것이다. 이러한 현상을 Overfitting이라고 한다. Train Set에 너무 Overfit(과적합)하여 Test Set에는 좋지 않다는 것이다.

2. Naive Bayes Classification

이번 장에서는 Naive Bayes Classification을 살펴본다. 우선 Bayes는 Bayes Rule의 Bayes이다. Bayes Rule은 우리가 수학시간에 배우는 조건부 확률(Conditional Probability)와 관련이 있는 원리이다. 우선 Conditional Probability에 대해서 먼저 공부해보도록 하자. 여기서도 일단 날씨 데이터를 활용하기로 한다.

특별한 경우를 가정한(조건부) 확률을 구하는 것이 Conditional Probability이다. 다른 말로 말하면, 두 개 이상의 Case에 대하여 어떤 Case가 정해졌을 때, 나머지들의 확률을 물어보는 것이다.

outlook	temperatur	humidity	windy	play
rainy	cool	normal	TRUE	no
rainy	mild	high	TRUE	no
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
overcast	hot	high	FALSE	yes
overcast	cool	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	mild	normal	TRUE	yes

위의 Data에서 Play가 yes일 때, Outlook이 sunny일 확률은 얼마일까?

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(\text{outlook} = \text{sunny} | \text{play} = \text{yes})$$

$$= \frac{P(\text{outlook}=\text{sunny} \cap \text{play}=\text{yes})}{P(\text{play}=\text{yes})}$$

$$= \frac{\frac{2}{14}}{\frac{9}{14}} = \frac{2}{9}$$

그냥 세서 나누면 된다. 우선 Play=yes인 경우가 총 9개 있고, 그 중 Outlook=sunny인 경우는 2개니까, 2/9가 답이다. 아주 쉽다. 그런데 이것을 conditional probability를 이용하면 아래와 같은 공식으로도 풀수 있다.

이제 Bayes Rule을 살펴보자. Bayes Rule은 아래와 같다.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

위 식을 풀이하면 B일 때, A가 일어날 확률은 P(B|A)와 P(A)와 P(B)를 구하면 알 수 있다는 뜻이다. 위의 날씨 Data를 보자.

Play가 우리가 예측을 해야 하는 Class Attribute이라고 가정하자. 우리는 위에서 P(Outlook=sunny|Play=yes)는 쉽게 구할 수 있었다. 그러면 P(Play=yes|Outlook=sunny)는 어떤 의미를 가지는 것인지를 알아보자. Bayes Rule을 이용하면 아래와 같이 구할 수 있다.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(\text{play} = \text{yes} | \text{outlook} = \text{sunny})$$

$$= \frac{P(\text{outlook} = \text{sunny} | \text{play} = \text{yes})P(\text{play} = \text{yes})}{P(\text{outlook} = \text{sunny})}$$

$$= \frac{\frac{2}{9} \times \frac{9}{14}}{\frac{14}{5}} = \frac{9 \times 2}{9 \times 5} = \frac{2}{5}$$

우리는 위의 날씨 Data에서 Outlook, Temperature, Humidity, Windy와 같은 Attribute을 가지고 있다. 모두다 nominal value들이므로 아래의 확률들을 쉽게 구할 수 있다.

P(Outlook=sunny|Play=yes)

P(Temperature=hot|Play=yes)

P(Humidity=high|Play=yes)

P(Windy=true|Play=yes)

위의 값을 구했다면 아래의 값도 구할 수 있다.

P(Play=yes|Outlook=sunny)

P(Play=yes|Temperature=hot)

P(Play=yes|Humidity=high)

P(Play=yes|Windy=true)

이 말은 날씨 Data가 아래와 같이 주어졌을 때, Class Attribute이 yes일 확률과 no일 확률을 구하여 비교한 후, 더 큰 확률로 예측하겠다는 것이다. 아래의 확률표를 보고 계산을 해보자.

	outlook		temperature		humidity		windy				
	yes	no	yes	no	yes	no	yes	no			
overcast	4/9	0/5	no	2/9	2/5	high	3/9	4/5	TRUE	3/9	2/5
rainy	3/9	2/5	cool	3/9	1/5	normal	6/9	1/5	FALSE	6/9	2/5
sunny	2/9	3/5	hot	4/9	2/5						

Class Attribute=yes 일 때,

$$\begin{aligned}
 &P(\text{yes}|\text{sunny, hot, high, true}) \\
 &= \frac{P(\text{sunny, hot, high, true}|\text{yes})P(\text{yes})}{P(\text{sunny, hot, high, true})} \\
 &= \frac{(\frac{2}{9} \times \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9}) \frac{9}{14}}{\frac{5}{14} \times \frac{4}{14} \times \frac{7}{14} \times \frac{6}{14}}
 \end{aligned}$$

Class Attribute=no 일 때,

p(Play=yes|sunny and hot and high and true) < p(Play=no|sunny and hot and high and true) 라는 것을 알 수 있다. 그런데 위의 식에서 하나 이상한 것을 볼 수 있다.

$$P(\text{sunny, hot, high, true}|\text{no}) = \frac{3}{5} \times \frac{2}{5} \times \frac{4}{5} \times \frac{3}{5}$$

위 식은 확률 이론에 기반해서 보면 사실이 아니다. 각각의 사건, 즉 Outlook, Temperature, Humidity, Windy가 각각 독립이라면, (서로 영향을 안 주는 사건) 위의 경우가 맞다. 그러나 사실은 독립사건이 아니다. 일반적으로 Temperature가 높을수록 Humidity도 높다. 그래서 Naive라는 단어를 붙인 것이다. 그렇지만 나이브 베이즈가 위와같이 계산 값을 구해도 실제로는 굉장히 좋은 결과를 내놓기도 한다.

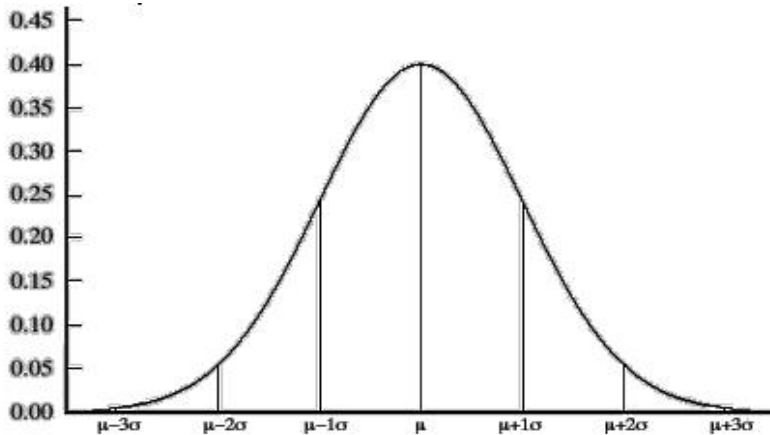


그림 54 정규분포곡선

만일 입력값이 수치형 자료이면 앞선 Decision Tree와 같은 형태로 정규 분포를 이용해서 확률을 구한다. 정규분포 식은 아래와 같다.

$$\mu = \text{mean}$$

$$\sigma = \text{standard deviation}$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

위의 식만 있으면 Numeric Data에 대해서도 확률을 구해낼 수 가 있다. 아래와 같이 날씨 Data를 추가해보자.

outlook	temperature	humidity	windy	temperature numeric	humidity numeric	play
rainy	cool	normal	TRUE	65	70	no
rainy	mild	high	TRUE	71	91	no
sunny	hot	high	FALSE	85	85	no
sunny	hot	high	TRUE	80	90	no
sunny	mild	high	FALSE	72	95	no
overcast	hot	high	FALSE	83	86	yes
overcast	cool	normal	TRUE	64	65	yes
overcast	mild	high	TRUE	72	90	yes
overcast	hot	normal	FALSE	81	75	yes
rainy	mild	high	FALSE	70	96	yes
rainy	cool	normal	FALSE	68	80	yes
rainy	mild	normal	FALSE	75	80	yes
sunny	cool	normal	FALSE	69	70	yes
sunny	mild	normal	TRUE	75	70	yes

Temperature=70일 경우에는 어떻게 확률이 계산되는지 확인해보자. Play=yes일 경우에는 Temperature Numeric의 Average와 Standard Deviation은 79.11111111, 10.21572861이므로, 공식에 넣으면 0.0262370186193가 나온다. 같은 방법으로 Play=no일 경우에는 Temperature Numeric의 Average와 Standard Deviation은 86.2와 9.731392501이므로, 공식에 집어넣으면 0.0102555691137가 나온다. Play=yes일 확률이 더 높게 나온다.

3. Artificial Neural Network(ANN)

기본적으로 ANN은 입력자료들을 연산 처리하여 그 Data에 대한 Predicted Value를 만들어낸다. 예를 들어, 간단하게 XOR 함수를 놓고 생각해보자. x1과 x2가 1이면 그 값은 0이 된다. 입력 x1과 x2가 각각 1이니까, 이 값들을 처리 하여 output을 0으로 혹은 1으로 추측해야한다고 가정해보자. 결론적으로 ANN의 추측 방법은 각각의 입력에 Weight을 곱해서 결과를 추측한다. 즉, ANN이 학습한다는 것은 각각의 Weight를 구하는 것이다. ANN의 기본 개념은 퍼셉트론이다. 퍼셉트론의 기본 형태는 매우 간단하다.

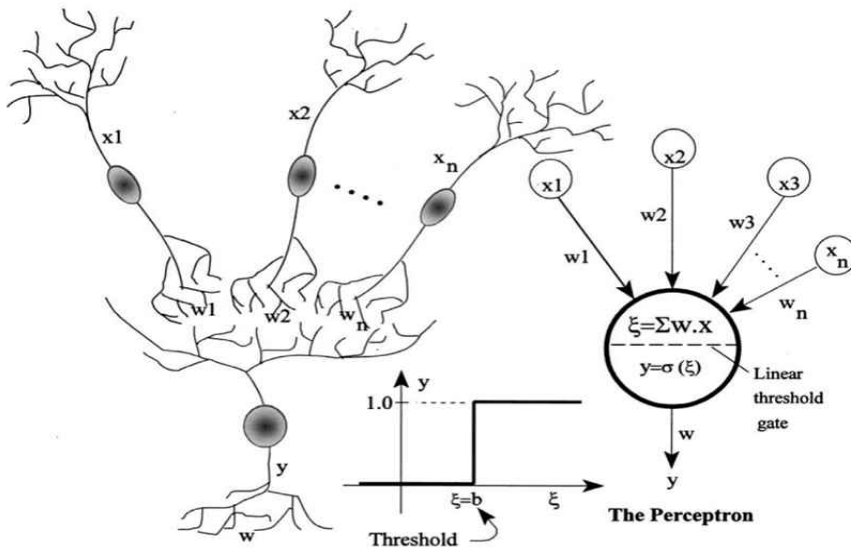


그림 55 신경망과 퍼셉트론

퍼셉트론은 인간의 신경구조를 흉내 낸 것인데 정보가 들어오는 각각의 선에 가중치(중요도)를 곱하여 모든 정보 값을 더한 후 일정한 값을 넘으면 의미가 있는 것으로 간주하는 것이다.

아래 그림에서 x1, x2는 1이고 w1, w2는 각각 x1과 x2에 대한 Weight이다. (x1, x2)와 (w1, w2)의 dot product를 구하면 o1이 나온다. Artificial Neural Network에서 Neural이 위의 그림 저것이다. Neural Network는 저 것들이 여러개 모였다는 것이다. 즉, 우리의 XOR 예를 가지고 생각해 보자면, XOR를 예측하기 위해 여러개의 Neural Network를 만들었다고 보면 된다. Neural Network는 아래와 같은 형태를 가진다.

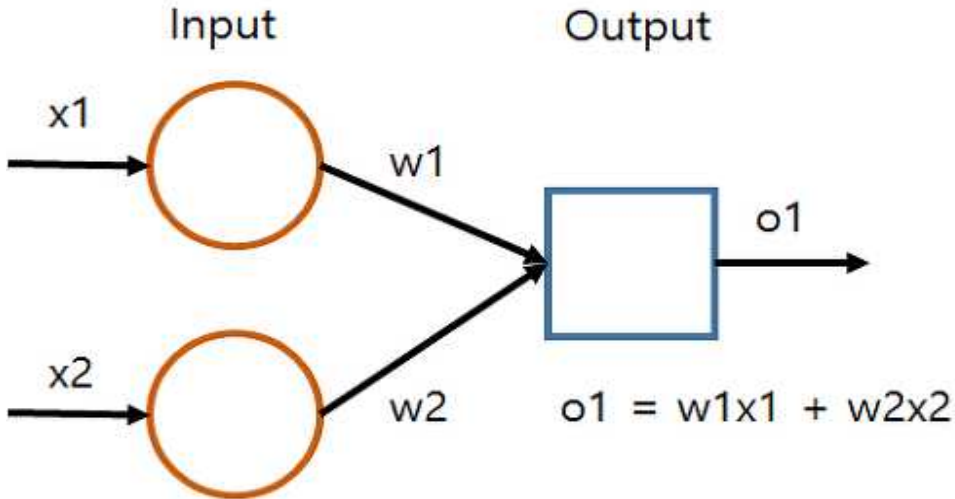


그림 56 단일 퍼셉트론을 이용한 신경망

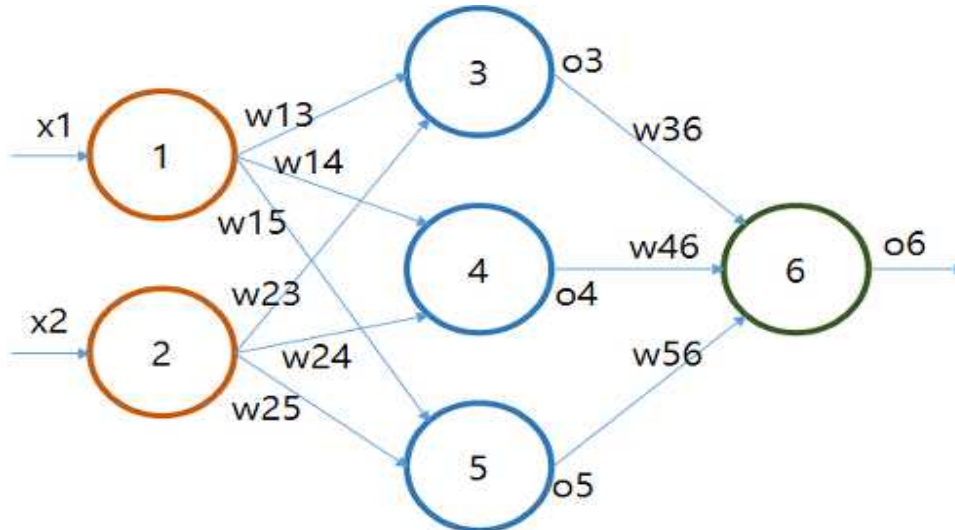


그림 57 다중 다층 퍼셉트론을 이용한 신경망

위의 그림에서 Notation부터 정리를 해보자. 우선 각각 원 i 에서 원 j 으로의 Weight를 w_{ij} 로 표시했다. 그리고 원은 1부터 6까지 있다. 그리고 원 i 에서 나오는 output을 o_i 로 표시했다. 자 위의 그림에서라면 우리는 x_1 과 x_2 의 XOR을 예측하기 위해 아래와 같은 일을 해야 된다.

여기서 중요한 용어를 한번 정리해보자. 바로 Input Layer, Hidden Layer, Output Layer이다. (사실 위의 식은 완벽하지 않다. 중요한 과정 하나가 누락되었는데 뒤에서 설명한다.)

$$o_3 = w_{13}x_1 + w_{23}x_2$$

$$o_4 = w_{14}x_1 + w_{24}x_2$$

$$o_5 = w_{15}x_1 + w_{25}x_2$$

$$o_6 = w_{36}o_3 + w_{46}o_4 + w_{56}o_5$$

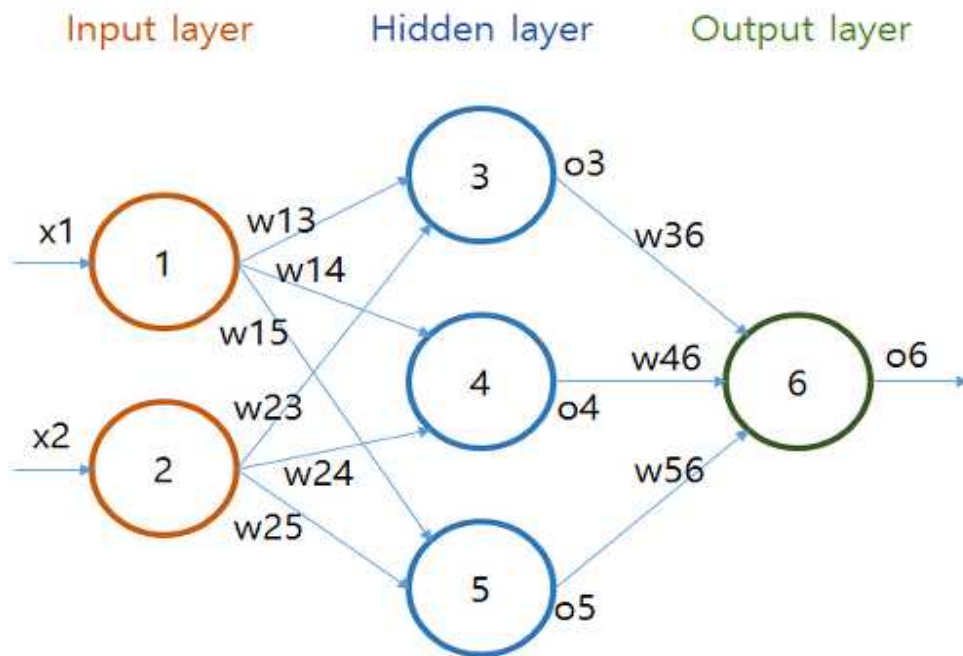


그림 58 다중 인공신경망의 가중치 계산

여기서 Hidden Layer는 문자 그대로 감추어진 레이어이다. Hidden Layer에서는 무슨 일이 일어나는지 알 수 없다. 여기서 Hidden Layer는 하나 밖에 없지만, 근래 알파고의 영향으로 많이 사용되는 딥러닝에서는 더욱 깊은(Deep) 여러 층의 hidden Layer가 있는데, 여기에서의 Deep이 Deep Learning의 Deep이다.

먼저, w_{13} 부터 w_{56} 까지 특정 값을 가지고 있다고 했을 때, x_1 과 x_2 만 있으면 o_6 를 구할 수 있다. o_6 를 구하기 위해서 우리는 Dot Product를 Input Layer에서 Hidden Layer로 3번하고 Hidden Layer에서 Output Layer로 1번, 총 4번을 하는데, 이 과정에서 Dot Product의 결과가 0이나

1로 정확히 떨어지지 않는다. 결과가 0과 1로 떨어지지 않으면 퍼셉트론을 이용한 결과를 사용하기 어렵다. 이 문제를 해결하기 위해 아래 그림과 같이 활성화 함수를 사용한다.

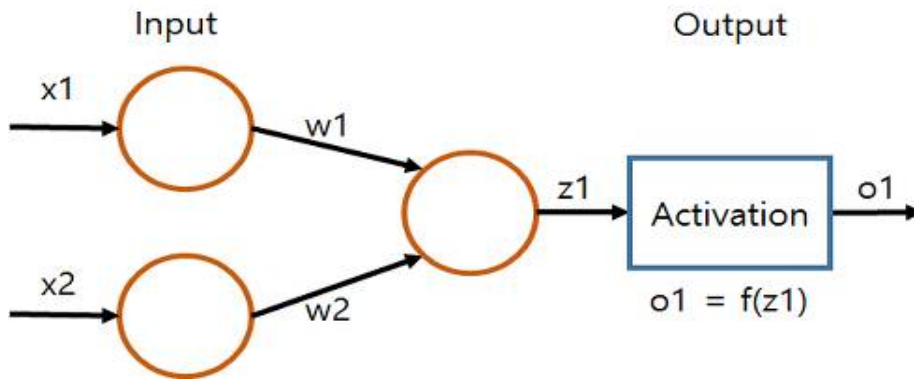


그림 59 인공신경망 활성화 함수

Activation Function

Sigmoid Function 외에 Step Function도 사용한다. 다만 step 함수는 특성상 미분 불가능하다. step 함수는 모양이 계단처럼 생긴데서 유래한 이름이다.

Dot Product 결과 z_1 을 Activation Function(활성화 함수)에 입력한다. 이 Activation Function은 어떤 함수를 택하느냐에 따라 o_1 의 값이 바뀔 수 있는데, 가장 일반적으로 Sigmoid Function(s자형 함수)을 택한다.

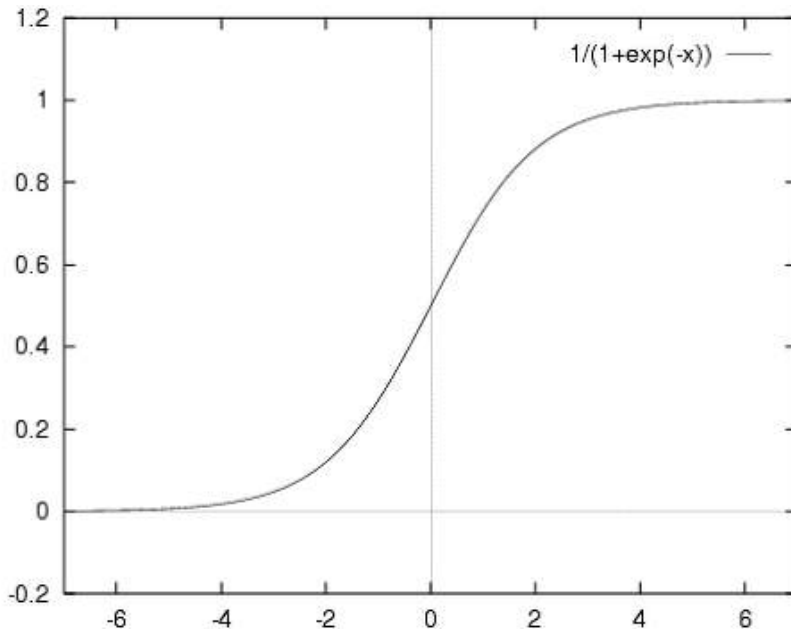


그림 61 시그모이드(sigmoid) 함수 그래프

Sigmoid Function은 그 값이 0과 1 사이(Probability 구할 때 유용)라는 특징이 있는 "미분 가능한" 함수이다.

$$z3 = w13x1 + w23x2$$

$$z4 = w14x1 + w24x2$$

$$z5 = w15x1 + w25x2$$

$$o3 = \text{sigmoid}(z3)$$

$$o4 = \text{sigmoid}(z4)$$

$$o5 = \text{sigmoid}(z5)$$

$$z6 = w36o3 + w46o4 + w56o5$$

$$o6 = \text{sigmoid}(z6)$$

활성함수를 통과하면 결과는 0과 1사이의 값을 가진다. 위 식을 따라 연산하여 결과가 0.8이 나왔다면 그 값은 1이 될 확률이 큰것이고, 0.2가 나왔다면 그 값은 0일 확률이 크다는 의미이다.

이와 같이 연산하는 과정을 ANN에서 forwarding이라고 한다. 즉 주어진 X의 값으로 그 ANN의 최종 Output을 만들어내는 과정이다. ANN을 실제 적용할 때 앞선 연산 과정을 한 번만 실행한다면 당연히 쓸만한 결과를 도출하지는 못한다. Weight가 어떤 값이냐에 실제 데이터를 적용했을 때 결과는 아주 많이 달라진다. 따라서 여러 번의 학습 과정을 통해 가장 좋은 Weight 값을 찾아야 한다. 그 과정은 아래 Cost Function(J)을 통하여 구한다.

$$J = \sum \frac{1}{2} (y - \hat{y})^2$$

여기서 \hat{y} 은 계산에 의한 Predicted Value이다. 즉 Predicted y와 실제 y의 차이를 구해서 제곱을 한 후 그것을 반으로 나누어 준다. 반으로 나누어 주는 이유는 다음 장에서 설명할 Back Propagation에서 위의 식을 미분하게 되는데, 그 때 제곱의 2가 밑으로 떨어지면서 1로 상쇄되면 계산이 간편해지기 때문이다.

결과적으로 신경망이 올바르게 동작하기 위해서는 Cost Function의 값이 가장 작도록 하는 Weight을 찾아야 한다. 구하는 과정에서 만약 1부터 1000까지 하나씩 Weight을 설정하여 나온 값의 최소값을 구할 수도 있으나 그 경우 시간이 아주 많이 소요된다. 따라서 최적의 속도로 Weight를 구하기 위해 미분을 사용한다. Cost Function을 J 로 놓고 dJ/dw 를 구해서 그 값이 0이 되는 방향으로 계속 Weight을 Update해 준다. 이렇게 최소값을 찾는 과정을 Gradient Decent(경사 하강법)라고 한다.

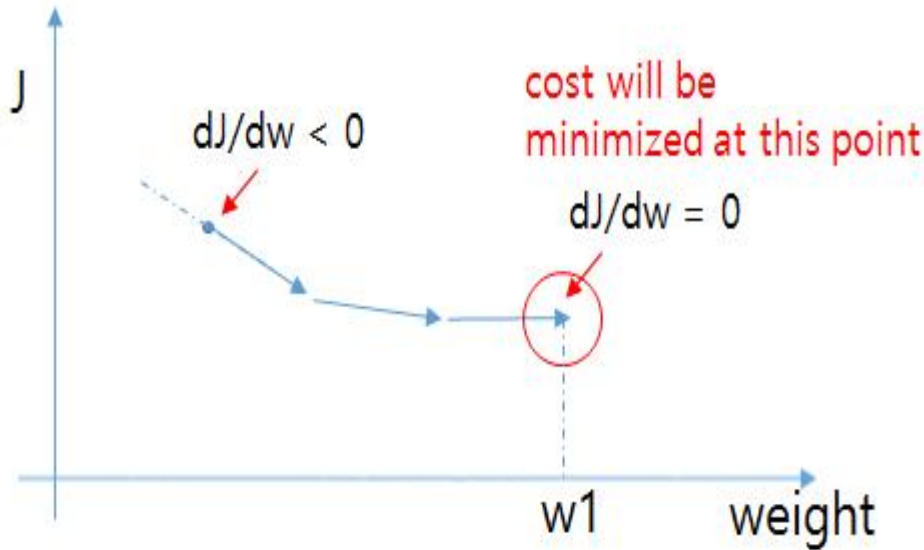


그림 62 Gradient Decent

마치 게임할 때 적절한 능력치를 올려줘서 balance를 맞춰주는 것처럼, 혹은 오디오에서 여러가지 음향 효과 버튼을 돌려서 최적의 소리를 세팅하는 것처럼, 각각의 Weight(w_{13} 부터 w_{56} 까지)의 Weight라는 "버튼"을 조금씩 돌려서 Cost Function이 최소화되는 그 순간을 찾는 것이다.

4. k-Nearest Neighbors

k-NN 알고리즘은 기존의 데이터와 비교해서 입력이 들어왔을 때, 이 입력 값이 기존의 데이터와 얼마만큼 연관 관계가 있느냐에 착안해서 만들어진 알고리즘이다. 분류군의 수를 정해줘야 하기 때문에 k라는 분류군의 수가 이름에 붙어 있다.

입력 값이 들어왔을 경우 입력 값과 기존 데이터들과의 거리를 구하고, 가장 거리가 가까운 곳을 찾아서 비슷함을 찾아내는 방법이다.

다음 파이썬 코드를 살펴보자.

처음 코드를 보면 createDataSet이란 함수를 통해 사전 데이터 집합을 생성하였다. 사전 데이터는 [1.0, 1.1], [1.0, 1.0], [0, 0], [0, 0.1]의 값을 가지는 데이터이며 각각 'A','A','B','B'의 Label을 가진다.

이렇게 데이터 셋을 생성한 후에, Classify 함수를 호출하는데, Classify 함수는 입력 데이터가 사전 데이터 셋과 비교하여 어떤 Label을 가지게 될지를 분류해주는 기능을 한다.

Classify 함수에 대해 조금 더 자세히 살펴보자.

우선 사전 데이터 set 크기를 가져온 후에, 사전 데이터 set과 입력 값과의 차이를 구한다. 예를 들어 사전 데이터가 [1.0, 1.1] 이고 입력 값이 [1.0, 1.0]이라면, 결과 값은 [0, 0.1]이 되는 것이다.

tile 함수를 쓴 이유는 DataSet과 동일한 크기의 array 행렬을 만들기 위해 사용되었으며, 위와 같은 경우에는 4x1의 행렬이 만들어진다.

이렇게 만들어진 행렬과 데이터 set의 차이 값을 구하고(diffMat), 구해진 diffMat를 2제곱을 한다.

2제곱 한 결과 행렬에 대해 각각을 더한다. (sqDiffMat.sum(axis = 1)) 그 후에 다시 0.5제곱을 곱하면 거리 값이 나타나게 된다.

말이 어려울 수 있는데 결국은 각 입력요소간의 거리를 구하는 것이다.

$$PH = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$


```

from numpy import *
import operator

def createDataSet():
    group = array([[1.0, 1.1], [1.0, 1.0], [0, 0], [0, 0.1]])
    labels = ['A', 'A', 'B', 'B']
    return group, labels

def Classify(inX, DataSet, labels, k):
    DataSetSize = DataSet.shape[0]
    diffMat = tile(inX, (DataSetSize, 1)) - DataSet
    sqDiffMat = diffMat ** 2
    sqDistances = sqDiffMat.sum(axis = 1)
    distances = sqDistances ** 0.5
    sortedDistIndicies = distances.argsort()
    ClassCount = {}

    for i in range(k):
        voteLabel = labels[sortedDistIndicies[i]]
        ClassCount[voteLabel] = ClassCount.get(voteLabel, 0) + 1
    sortedClassCount = sorted(ClassCount.iteritems(), key=operator.itemgetter(1),
reverse=True)
    return sortedClassCount[0][0]

if __Name__ == "__main__":
    group, labels = createDataSet()
    print Classify([0, 0], group, labels, 3)

```

표 5 k-NN 샘플코드

이제 구한 거리 값에 대하여 작은 순서대로 Sorting을 해준다.
가장 가까운 거리부터 어느 정도까지 루프를 돌려서 가장 많이 나온 Label을 찾아야 한다.
그에 해당하는 부분이 for문과 그 아래의 코드 부분이다.
이렇게 해서 나온 Label 결과가 입력 값에 대해 가장 가까운 Label이다.

입력 값이 어떤 값이 들어왔는지는 확실히 알 수가 없지만 대략 가장 가까운 거리에 있는 Label과 비슷한 값이라고 추측을 해볼 수가 있다.

이것이 가장 단순한 형태의 k-NN알고리즘이다.

실제 시스템에 적용하기 위해서는 저렇게 실데이터 값을 집어넣기전에 정규화 과정 등을 거치는 것이 더 효율적이다.

두 속성간의 비교 연산을 통한 결과 값을 구하는 것이기 때문에 속성간의 값 차이가 크게 나버리면 정확도가 떨어지기 때문이다.

가령 두 속성 값이 (100000, 0.1), (40, 0.9) 를 나타낸다고 가정할 시에 0.1과 0.9의 차이가 아무리 크게 나더라도 100000과 40 사이에 끼치는 영향의 효과가 더 크기 때문에 0.1과 0.9 차이는 묻혀져 버리게 된다.

이를 피하기 위해 거리를 구하기 전에 값의 정규화 작업은 필수적으로 이루어져야 한다.

정규화작업은 대부분 아래와 같은 방식으로 이루어 진다.

$$(value - min) / (max - min)$$

여기서 min, max는 각 속성이 가지고 있는 값 내에서의 최소값, 최대값을 나타낸다.

예를 들어 사람1에 해당하는 속성 값이 (1, 100, 0.1)이고 사람2에 해당하는 속성 값이 (5, 300, 0.5), 사람3이 (10, 200, 0.3)일 경우, 첫 번째 속성에 대한 min max는 1, 10, 두 번째 속성에 대한 min max는 100, 300, 세 번째 속성에 대한 min max는 0.1, 0.5가 되는 것이다.

02 Weka를 이용한 머신러닝 실습

학습목표

- weka를 이용하여 머신러닝의 기본 알고리즘을 이해하고 사용할 수 있다.

I. Weka 설치

1. Weka란?

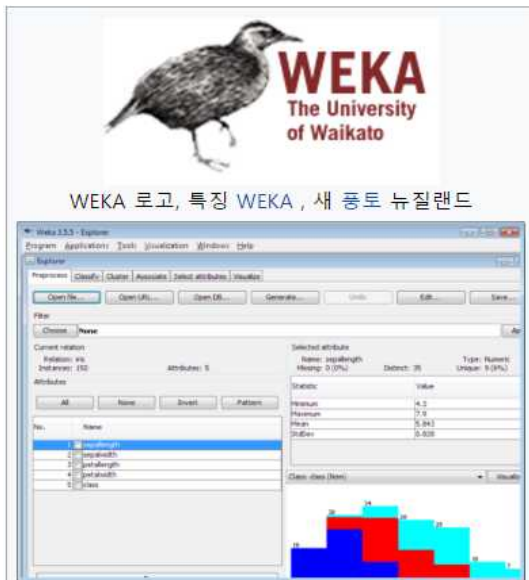


그림 63 Weka 로고

1.1 정의

“Weka” 는 데이터 분석 및 예측 모델링을 위해 시각화 도구 및 알고리즘 모음과 이러한 기능을 쉽게 액세스할 수 있는 GUI 이다. Weka는 GNU 일반 공중 라이선스 에 따라 무료로 사용가능하다.

Java 프로그래밍 언어로 완벽하게 구현되어 거의 모든 최신 컴퓨팅 플랫폼에서 실행되고 그래픽 사용자 인터페이스로 인해 사용하기 쉽다.

1993년, 와이 카토 대학에서 개발을 시작했다. 1997년, 모델 알고리즘의 구현을 포함하여 Java를 이용하여 Weka를 재개발하였다.

2. WEKA 설치



그림 64 weka 사이트

공식사이트에 접속한다. 주소는 다음과 같다.

(<http://www.cs.waikato.ac.nz/ml/weka/>)

- **Windows**
 - Click [here](#) to download a self-extracting executable for 64-bit Windows that includes Oracle's 64-bit Java VM 1.8 (weka-3-8-1-jre-x64.exe; 112.0 MB)
 - Click [here](#) to download a self-extracting executable for 64-bit Windows without a Java VM (weka-3-8-1-x64.exe; 50.6 MB)
 - Click [here](#) to download a self-extracting executable for 32-bit Windows that includes Oracle's 32-bit Java VM 1.8 (weka-3-8-1-jre.exe; 106.4 MB)
 - Click [here](#) to download a self-extracting executable for 32-bit Windows without a Java VM (weka-3-8-1.exe; 50.6 MB)
 - These executables will install Weka in your Program Menu. Download the version without the Java VM if you already have Java 1.7 (or later) on your system.
 - **Mac OS X**
 - Click [here](#) to download a disk image for OS X that contains a Mac application including Oracle's Java 1.8 JVM (weka-3-8-1-oracle-jvm.dmg; 123.8 MB)
 - **Other platforms (Linux, etc.)**
 - Click [here](#) to download a zip archive containing Weka (weka-3-8-1.zip; 50.9 MB)
- First unzip the zip file. This will create a new directory called weka-3-8-1. To run Weka, change into that directory and type

그림 65 weka 다운로드

위의 그림과 같이 다운받을 수 있는 플랫폼이 적혀 있는데, 컴퓨터에 맞게 설치하면 된다.

다운로드가 완료되면 설치를 시작한다.



그림 66 weka 설치

클릭하면 다음과 같이 설치 마법사 화면이 나온다.

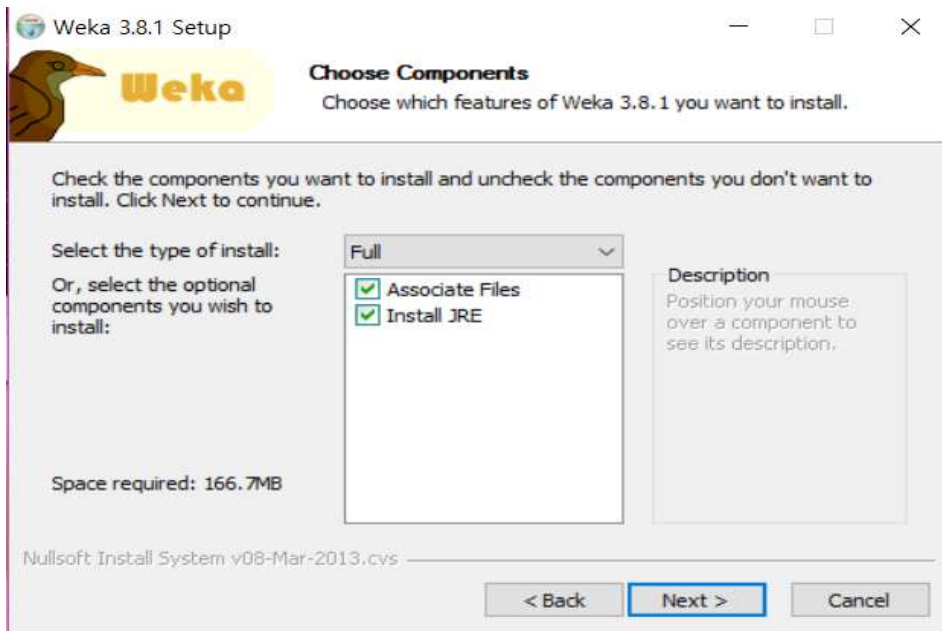


그림 67 weka 설치 옵션설정

전부 체크를 확인 되어있는 것을 확인하고 "Next >" 클릭해 넘어간다.

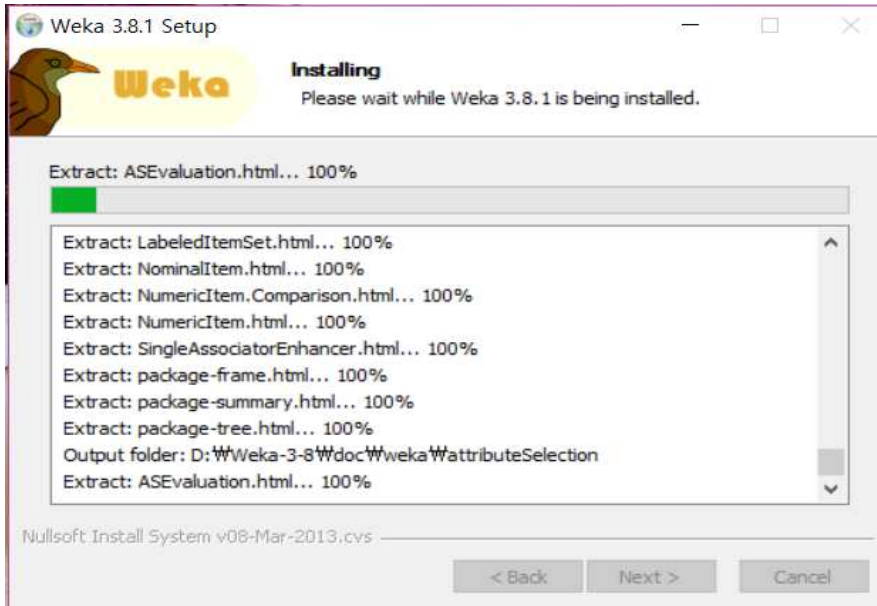


그림 68 weka 설치

넘어가게 되면 이와 같이 설치가 시작된다.



그림 69 java 설치

Weka 설치하는 도중 자바를 설치하라고 팝업창이 뜨면 “설치” 버튼을 누르면 된다.

II. Weka를 이용한 머신러닝 알고리즘 실습 Decision Tree



그림 70 weka 시작

“Explorer”를 클릭해 작업할 수 있는 환경을 만들어준다.

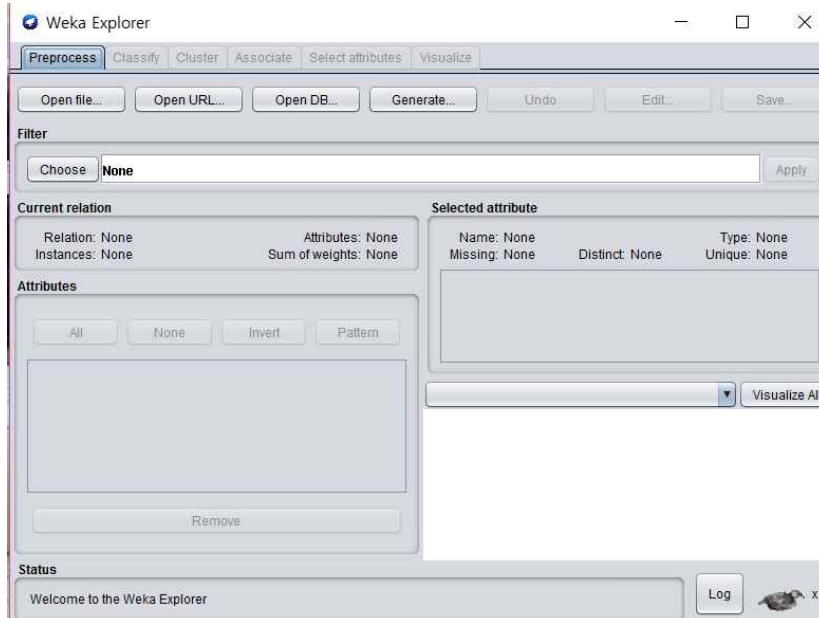


그림 71 weka 메인 화면

다음과 같은 화면이 뜨는데 여기서 open file을 눌러준다.

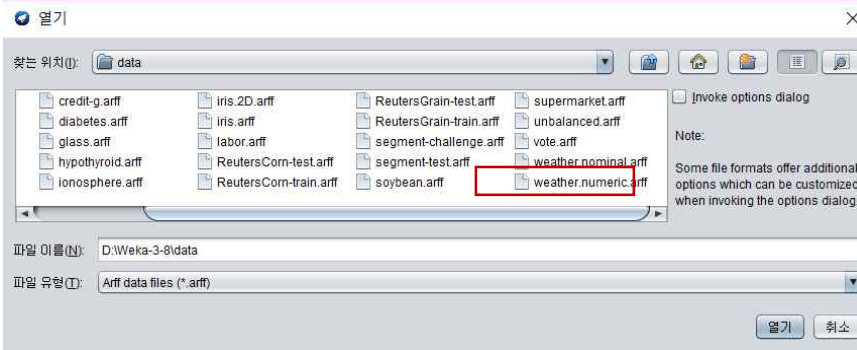


그림 72 실습을 위한 데이터 선택

간단한 예제로 Weka-3-8/Data/weather,numeric.arff를 열어보자.
 의사결정트리에 사용되는 날씨 자료는 머신러닝을 설명할 때 공통적으로
 사용한다.

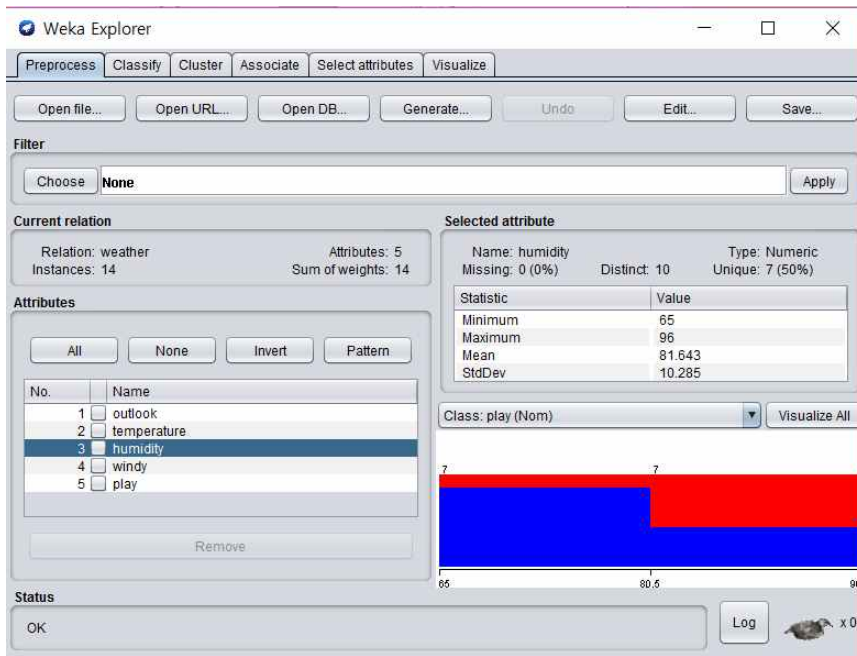


그림 73 훈련 자료의 내용

다음과 같은 GUI를 볼 수 있는데, 위에서 Outlook, Temperature, Humidity, Windy 변수가 있는 걸 확인할 수 있다.

Supervised인 의사결정트리를 이용해 결과를 예측할 수 있게 머신러닝을 시키자.

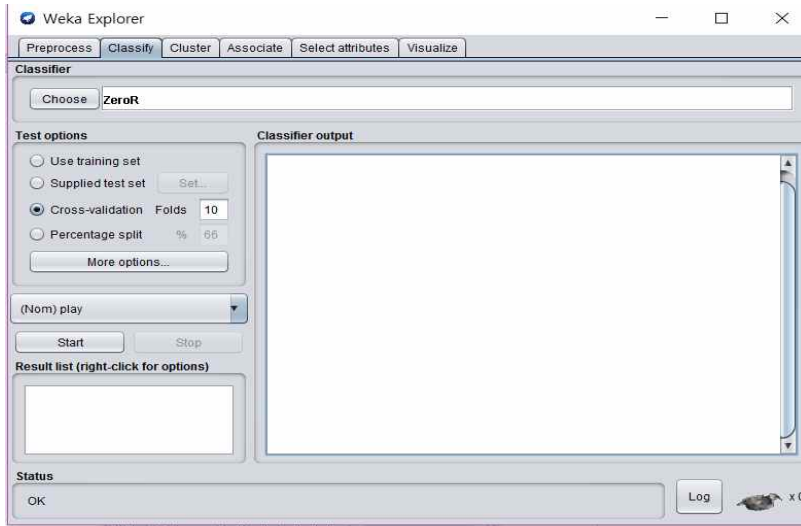


그림 74 캡션 알고리즘의 선택

“Choose” 클릭하여 알고리즘을 선택하자.



Tree 디렉토리를 들어가 j48을 누른다. 그런 다음 start 버튼을 누르면 의사결정트리가 만들어진다.

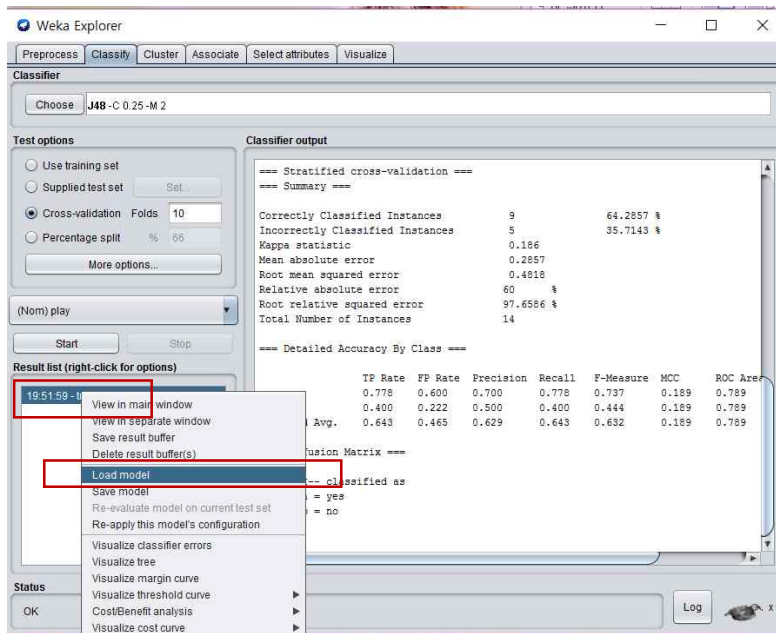


그림 75 훈련 및 결과 확인

“START”을 누르고-> “Visualize Tree” 클릭해 Model을 확인해보자.

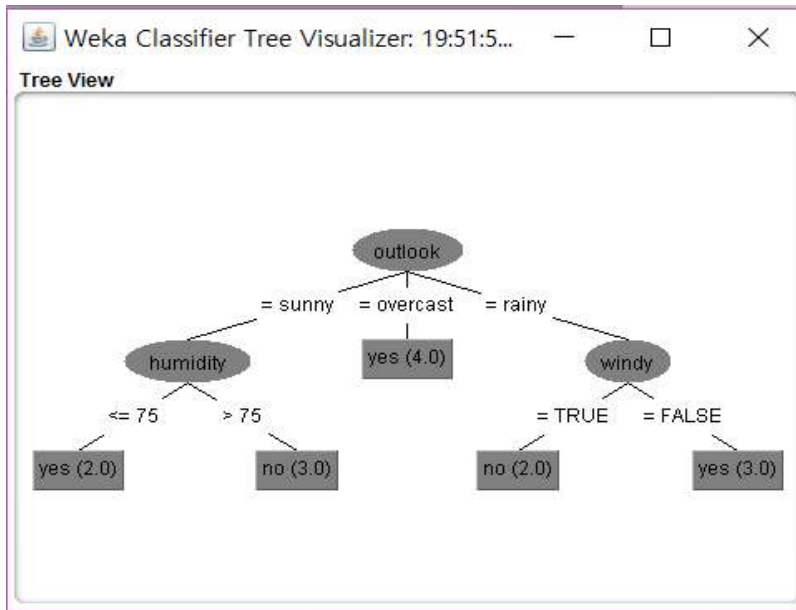


그림 76 생성된 의사 결정 트리

다음과 같은 의사결정트리 Model 화면 이 뜨게 되는 것을 확인할 수 있다. 지금 이 의사결정트리 모델을 통해 간단히 설명해보자.

Outlook의 상태가 Overcast인 경우 모든 사람들은 밖에 나가 놀았다. 그러나 sunny거나 rainy일 경우 나간 사람과 안 나간 사람들로 나누어졌다.

Sunny일 경우, Humidity가 75 이상일 경우에만 밖에 나가 놀았으며, 75 이하 일 경우 밖에 나가지 놀지 않았음을 알 수 있다.

2. ANN을 이용한 당뇨병 예측하기

ANN과 자료를 이용하여 당뇨병에 걸릴 가능성을 예측하는 모델을 만들어 보자.

먼저 weka를 설치하면 기본적으로 제공되는 데이터 중에서 하나를 선택 하자.

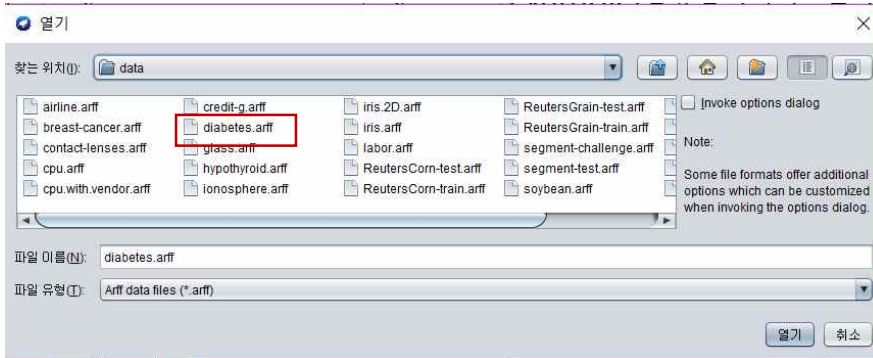


그림 77 당뇨 관련 데이터 선택

당뇨라는 뜻을 가진 diabetes.arff를 열어보자.

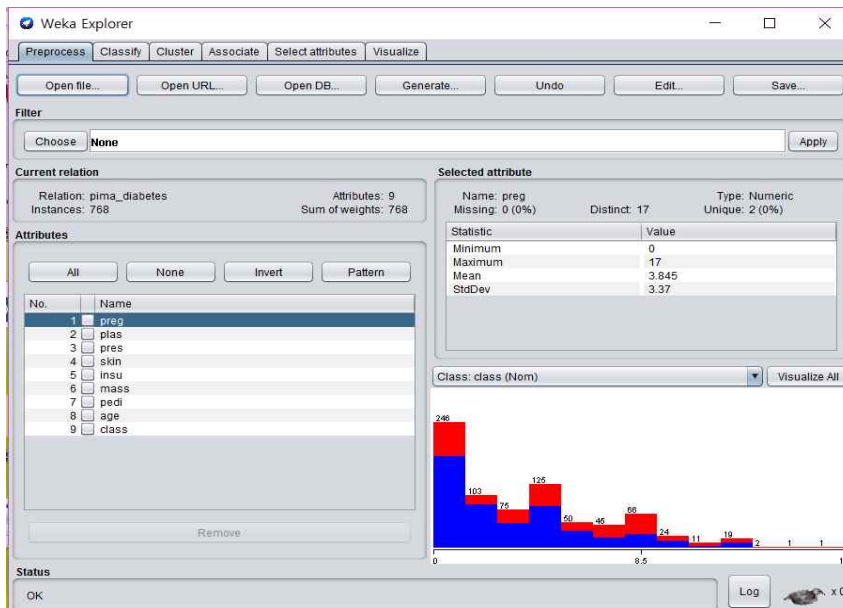


그림 78 데이터 로딩 결과

다음과 같은 내용을 weka에서 확인할 수 있다.

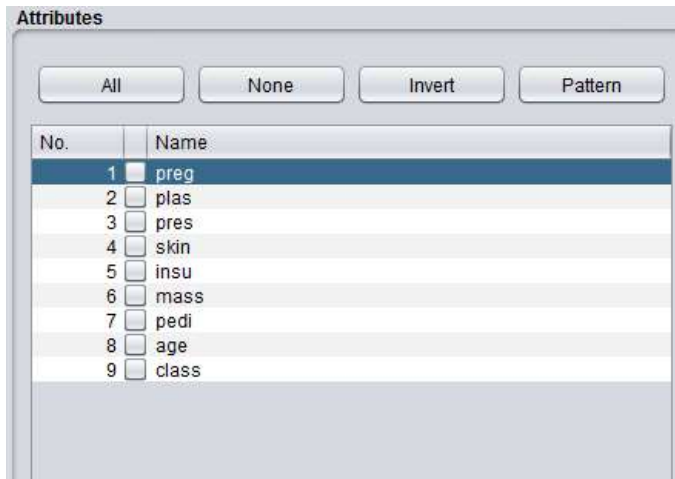


그림 79 데이터 속성

위와 같은 변수를 볼 수 있는데, 각 내용은 당뇨병증과 관계된 변수 이다.

test

- % 3. Diastolic blood pressure (mm Hg)
- % 4. Triceps skin fold thickness (mm)
- % 5. 2-Hour serum insulin (mu U/ml)
- % 6. Body mass index (weight in kg/(height in m)^2)
- % 7. Diabetes pedigree function
- % 8. Age (years)
- % 9. Class variable (0 or 1)

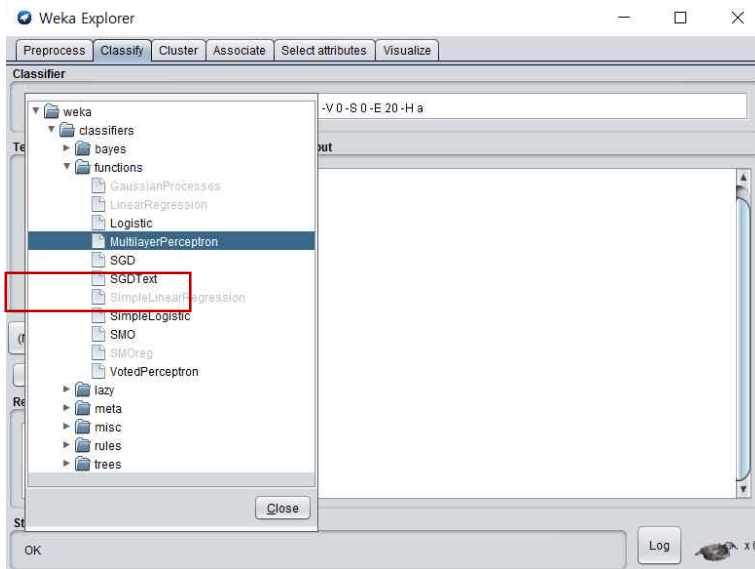


그림 80 훈련 알고리즘의 선택

알고리즘 선택에서 다층 ANN을 선택한다.
Function/multiLayerperceptron 을 사용하자.

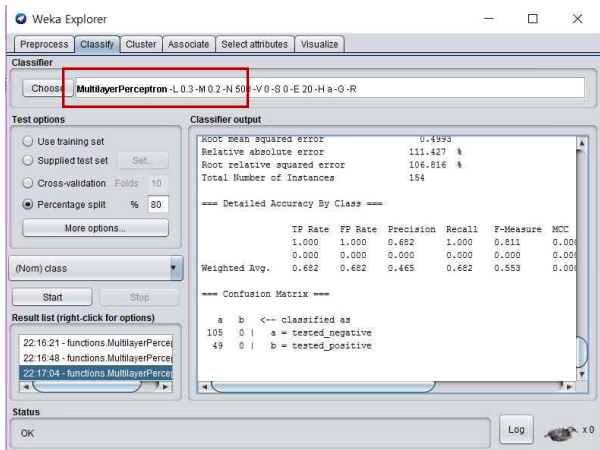


그림 81 알고리즘 세부 속성 설정

이름부분을 클릭하게 되면 속성창을 열 수 있다.

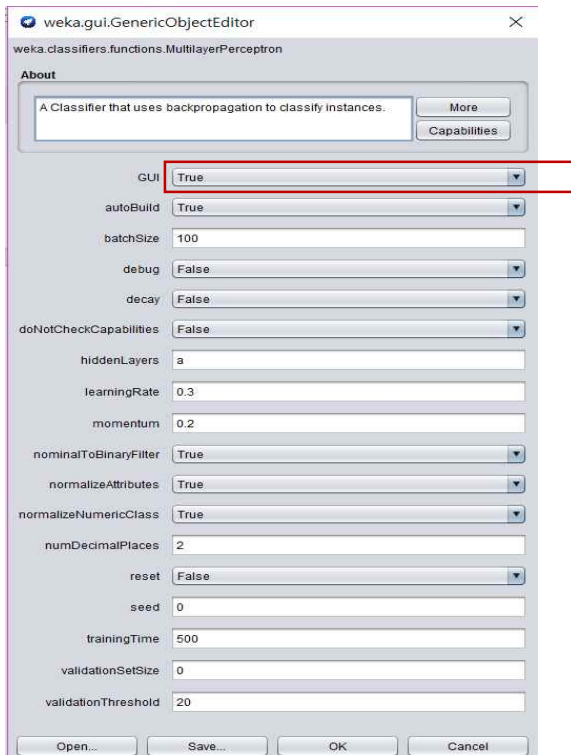


그림 82 GUI 옵션 활성화

알고리즘 옵션에서 그래픽 부분을 활성화 해주면 다음과 같은 모델을 그림으로 확인할 수 있다. GUI를 TRUE로 두고 START를 해보자!

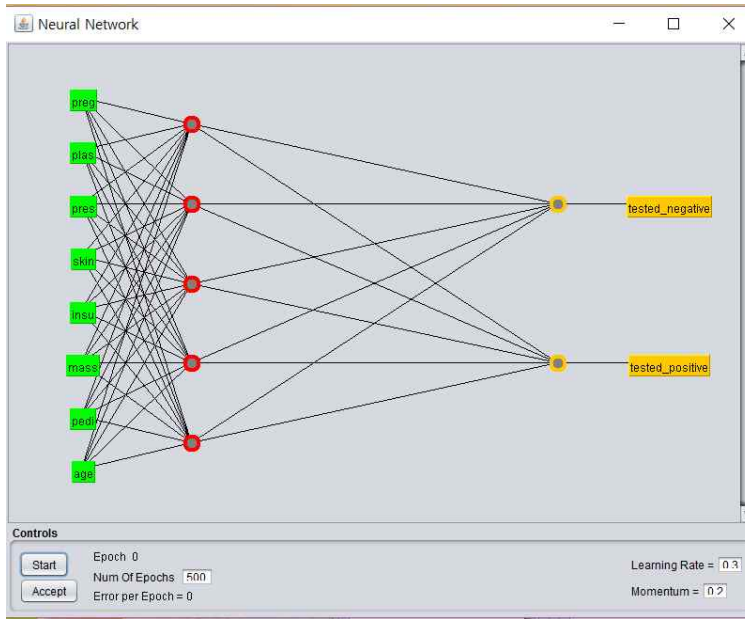


그림 83 훈련에 의해 생성된 신경망

이와 같은 모델이 나오게 되는 걸 볼 수 있다.

붉은색 원이 퍼셉트론이다. 퍼셉트론의 역할은 함수와 같은 역할이 되어 필요한 퍼셉트론을 이용해 계산을 하고 그 이후에 결과에 따라 컴퓨터가 알아서 가중치를 적용한다. 여기서 "START" 버튼을 누르면 학습을 시작하고 그 후에 창을 닫고 경고창이 뜨면 "YES"를 누르고 결과를 확인 할 수 있다.

```

Attrib plas -6.062276016682769
Attrib pres -1.3896840458164514
Attrib skin 0.34997084802061346
Attrib insu -2.211940814726504
Attrib mass -0.9589656235930499
Attrib pedi 6.090003751353257
Attrib age -8.833262465394691
Class tested_negative
  Input
  Node 0
Class tested_positive
  Input
  Node 1

Time taken to build model: 4.27 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      105          68.1818 %
Incorrectly Classified Instances    49           31.8182 %
Kappa statistic                     0
Mean absolute error                 0.4993
Root mean squared error             0.4993
Relative absolute error             111.427 %
Root relative squared error         106.816 %

```

그림 84 훈련결과창

이와 같이 정답률이 68%라는 것을 볼 수 있는데, 이것은 학습의 횟수와 관련되어 있다. 학습 횟수를 높이면 과연 정답률이 커질지 궁금해지니 한번 높여보자.

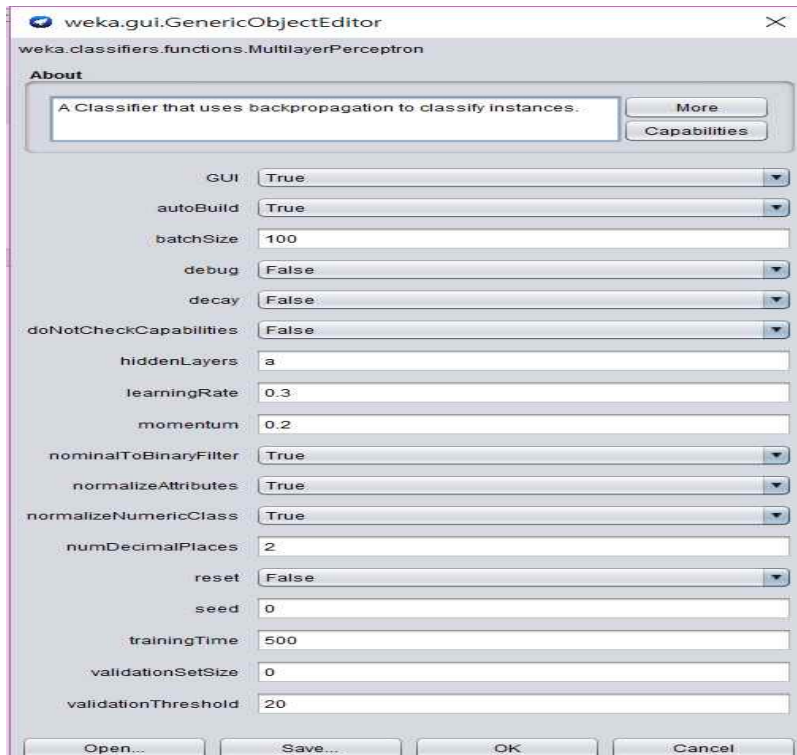


그림 85 훈련 횟수의 변경

사진 순서대로 5000->8000->15000 교육하였다.

Correctly Classified Instances	113	73.3766 %
Incorrectly Classified Instances	41	26.6234 %
Correctly Classified Instances	114	74.026 %
Incorrectly Classified Instances	40	25.974 %
Correctly Classified Instances	113	73.3766 %
Incorrectly Classified Instances	41	26.6234 %

순차적으로 교육 횟수를 늘렸는데 점차 정확도는 올라가나 횟수를 크게 늘려주니 유지되는 것이 아니라 오히려 떨어졌다.

이런 현상을 오버피팅이라 하는데 횟수가 너무 높아 값이 튀는 경우라 한다.

앞으로 머신러닝을 할때 교육 횟수를 잘 생각해서 주지 않으면 이로 인해 정확도가 오히려 떨어지는 경우를 볼 수 있다.

03 파이썬을 이용한 ANN 실습

학습목표

- 파이썬 프로그램을 이용하여 ANN을 구현할 수 있다.

1. 간단한 연습

Python 이란 쉽고 간단한 언어로 머신러닝을 예제를 구현해 볼 것이다. 그러기 위해서는 넘파이, 사이파이 등 계산 능력과 통계 능력이 탁월한 라이브러리를 이용하자.

1.1 간단한 그래프 그려 보기

넘파이로 간단한 sin 그래프를 그려보자.

```
import numpy as np
import matplotlib.pyplot as plt

# 데이터 준비
x = np.arange(0, 6, 0.1) # 0에서 6까지 0.1 간격으로 생성
y = np.sin(x)

# 그래프 그리기
plt.plot(x, y)
plt.show()
```

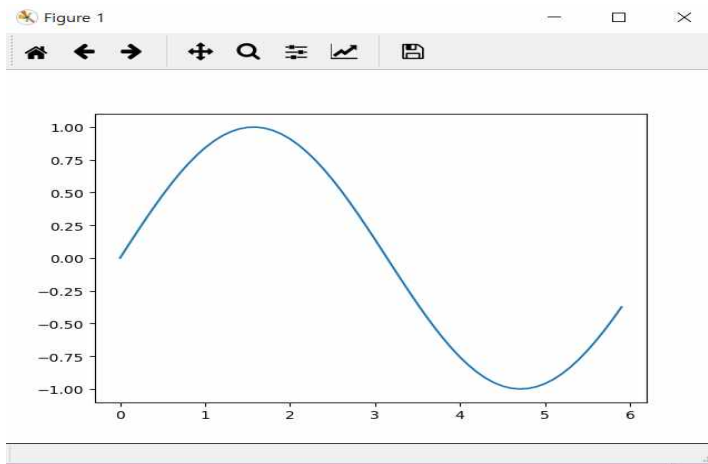



그림 86 파이썬으로 작성한 사인 그래프

Arrange 함수로 데이터를 생성하여 plt.show()를 호출해 화면에 출력하여 위와 같은 그래프가 그려진다.

다음으로 pyplot의 기능을 이용해서 cos함수를 그려 넣어보자

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
# 데이터 준비
```

```
x = np.arange(0, 6, 0.1) # 0에서 6까지 0.1 간격으로 생성
```

```
y1 = np.sin(x)
```

```
y2 = np.cos(x)
```

```
# 그래프 그리기
```

```
plt.plot(x, y1, label="sin")
```

```
plt.plot(x, y2, linestyle = "--", label="cos") # cos 함수는 점선으로 그리기
```

```
plt.xlabel("x") # x축 이름
```

```
plt.ylabel("y") # y축 이름
```

```
plt.title('sin & cos')
```

```
plt.legend()
```

```
plt.show()
```

(예제코드)

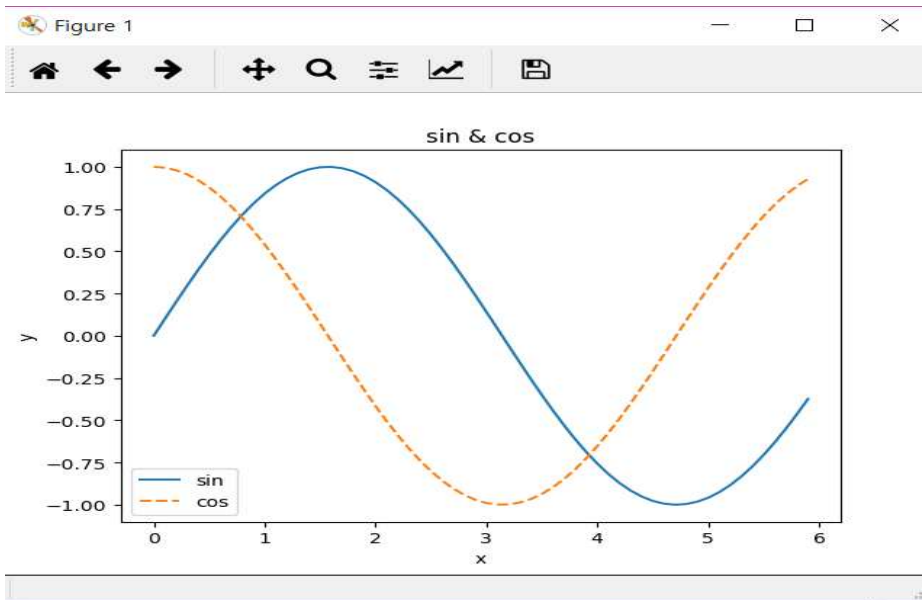


그림 87 사인과 코사인 그래프

2. ANN의 구현

2.1 퍼셉트론의 구현

퍼셉트론이란 다수의 신호를 받아 하나의 신호를 출력한다.

퍼셉트론이 동작하는 방식은 다음과 같다. 각 노드의 가중치와 입력치를 곱한 것을 모두 합한 값이 활성화함수에 의해 판단되는데, 그 값이 임계치 (보통 0)보다 크면 뉴런이 활성화되고 결과값으로 1을 출력한다. 뉴런이 활성화되지 않으면 결과값으로 -1을 출력한다.

퍼셉트론이 알려진 후 마빈 민스키와 시모어 페퍼트는 저서 "퍼셉트론"에서 단층 퍼셉트론은 XOR 연산이 불가능하지만, 다층 퍼셉트론으로는 XOR 연산이 가능함을 보였다.

논리회로 AND 게이트의 퍼셉트론을 구현하면 다음과 같다

```

import numpy as np

def AND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.7
    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else:
        return 1

if __Name__ == '__main__':
    for xs in [(0, 0), (1, 0), (0, 1), (1, 1)]:
        y = AND(xs[0], xs[1])
        print(str(xs) + " -> " + str(y))

```

D:\anaconda\python.exe "D:/untitled"

```

(0, 0) -> 0
(1, 0) -> 0
(0, 1) -> 0
(1, 1) -> 1

```

Process finished with exit code 0

논리회로 NAND 게이트의 퍼셉트론을 구현하면 다음과 같다

```

NAND

import numpy as np

def NAND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([-0.5, -0.5])
    b = 0.7
    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else:
        return 1

if __Name__ == '__main__':
    for xs in [(0, 0), (1, 0), (0, 1), (1, 1)]:
        y = NAND(xs[0], xs[1])
        print(str(xs) + " -> " + str(y))

```

```
D: #anaconda#py1
(0, 0) -> 1
(1, 0) -> 1
(0, 1) -> 1
(1, 1) -> 0
```

퍼셉트론의 가장 큰 문제는 선형으로 구분 될 수 없는 것은 처리할 수 없는 것이었다. 그러나 이 문제는 다층 퍼셉트론으로 처리할 수 있다.

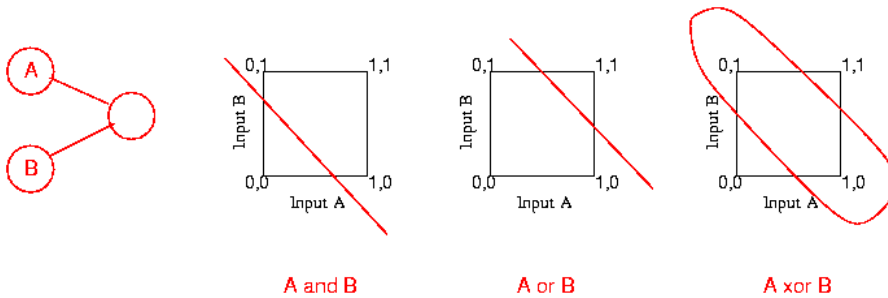


그림 88 XOR의 문제

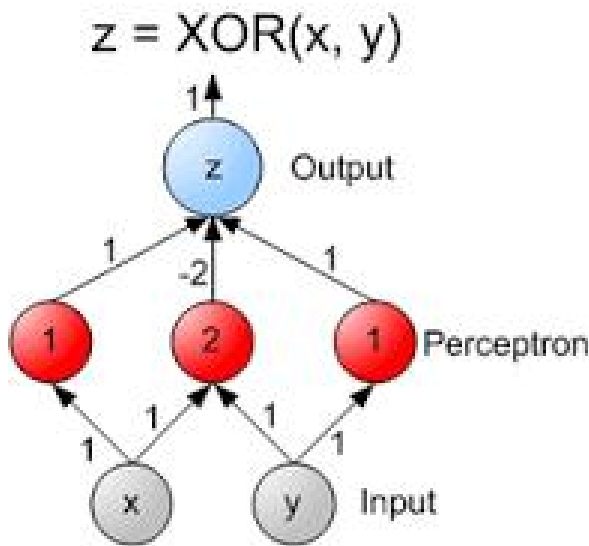


그림 89 다중 퍼셉트론으로 XOR 해결

위 그림처럼 XOR는 직선하나로는 그 값을 구분할 수 없다. 그러나 층을 여러 개 만들면 이에 대한 처리가 가능하다.

XOR게이트의 파이썬 구현은 다음과 같다.

```
from and_gate import AND
from or_gate import OR
from nand_gate import NAND

def XOR(x1, x2):
    s1 = NAND(x1, x2)
    s2 = OR(x1, x2)
    y = AND(s1, s2)
    return y

if __Name__ == '__main__':
    for xs in [(0, 0), (1, 0), (0, 1), (1, 1)]:
        y = XOR(xs[0], xs[1])
        print(str(xs) + " -> " + str(y))
```

U: #anaconda#pyt

(0, 0) -> 0

(1, 0) -> 1

(0, 1) -> 1

(1, 1) -> 0

3. ANN을 이용한 손글씨 인식

MNIST 데이터를 사용하면 다양한 손글씨 자료를 얻을 수 있다 이를 이용하여 ANN이 손글씨 자료를 인식하는 프로그램을 작성하여 보자.

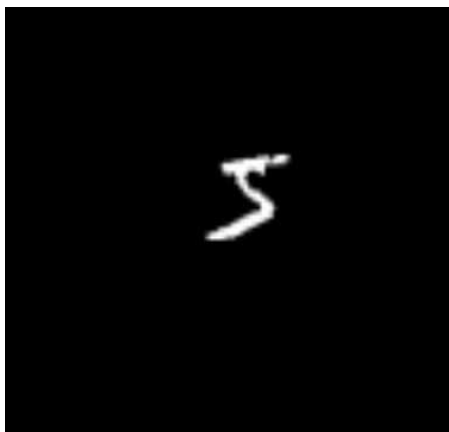


그림 90 MNIST 손글씨 예

위 그림은 MNIST에서 얻은 손글씨 자료의 일부이다.

다음 프로그램은 넘파이와 MNIST를 이용하여 ANN을 구현하고 이를 이용하여 손글씨를 인식한 메인 프로그램이다. 최종 그림에서 보듯이 그 결과가 90%이상의 적중률을 보인다.

```
import sys, os
sys.path.append(os.pardir) # 부모 디렉터리의 파일을 가져올 수 있도록
설정
import numpy as np
import pickle
from Dataset.mnist import load_mnist
from common.Functions import sigmoid, softmax

def get_Data():
    (x_train, t_train), (x_test, t_test) = load_mnist(normalize=True,
    flatten=True, one_hot_label=False)
    return x_test, t_test

def init_network():
    with open("sample_Weight.pkl", 'rb') as f:
        network = pickle.load(f)
    return network

def predict(network, x):
    W1, W2, W3 = network['W1'], network['W2'], network['W3']
    b1, b2, b3 = network['b1'], network['b2'], network['b3']

    a1 = np.dot(x, W1) + b1
    z1 = sigmoid(a1)
    a2 = np.dot(z1, W2) + b2
```

```
z2 = sigmoid(a2)
a3 = np.dot(z2, W3) + b3
y = softmax(a3)
```

```
return y
```

```
x, t = get_Data()
network = init_network()
accuracy_cnt = 0
for i in range(len(x)):
    y = predict(network, x[i])
    p= np.argmax(y) # 확률이 가장 높은 원소의 인덱스를 얻는다.
    if p == t[i]:
        accuracy_cnt += 1
```

```
print("Accuracy:" + str(float(accuracy_cnt) / len(x)))
```

```
D:\anaconda\python.exe "D:/untitled/ee/dee
Accuracy:0.9352
```

```
Process finished with exit code 0
```

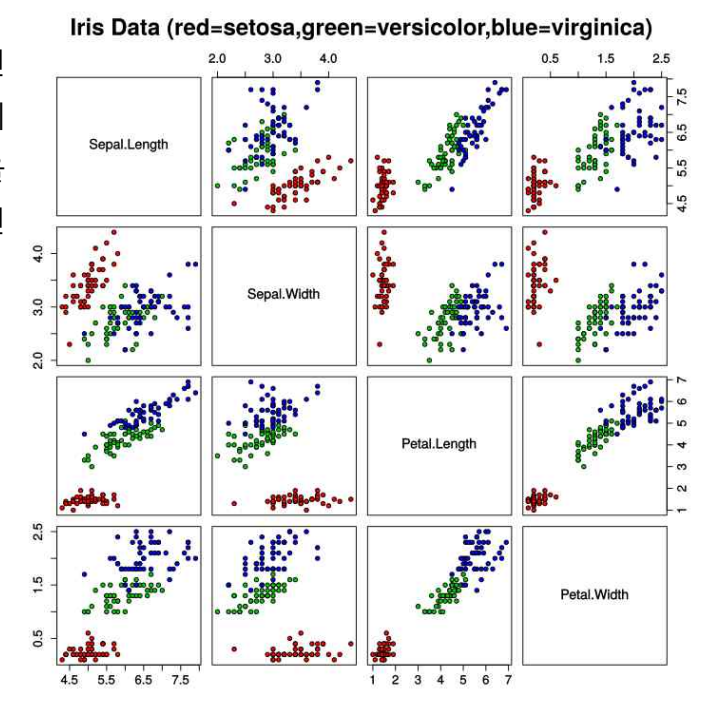


Iris Data Set

아이리스 데이터 셋은 영국 학자 로널드 피셔가 1936년 그의 논문에서 선형 판별 분석을 사용한 데이터 이다. 이 자료에는 4가지 속성(꽃받침의 길이, 넓이, 꽃잎의 길이와 넓이)을 가진 3종류의 꽃 자료가 50샘플이 있다. 이 자료를 최근 머신러닝 각 분야에 알고리즘을 측정하기 위한 기본 자료로 다양하게 활용하고 있다.



오른쪽 그림을 보면 세 가지 종류의 꽃에 따라 다른 특성을 가지고 있음을 확인할 수 있다.



□ 실습 과제

- 파이썬을 이용하여 아이리스 데이터에 대한 Decision Tree를 만들어 보자

□ 참고코드

```
print(__doc__)
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from sklearn.Datasets import load_iris
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
# Parameters
```

```
n_Classes = 3
```

```
plot_colors = "bry"
```

```
plot_step = 0.02
```

```
# Load Data
```

```
iris = load_iris()
```

```
for pairidx, pair in enumerate([[0, 1], [0, 2], [0, 3],
```

```
                                [1, 2], [1, 3], [2, 3]]):
```

```
    # We only take the two corresponding features
```

```
    X = iris.data[:, pair]
```

```
    y = iris.target
```

```
# Train
```

```
clf = DecisionTreeClassifier().fit(X, y)
```

```
# Plot the decision boundary
```

```

plt.subplot(2, 3, pairidx + 1)

x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, plot_step),
                    np.arange(y_min, y_max, plot_step))

Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
cs = plt.contourf(xx, yy, Z, cmap=plt.cm.Paired)

plt.xlabel(iris.feature_Names[pair[0]])
plt.ylabel(iris.feature_Names[pair[1]])
plt.axis("tight")

# Plot the training points
for i, color in zip(range(n_Classes), plot_colors):
    idx = np.where(y == i)
    plt.scatter(X[idx, 0], X[idx, 1], c=color,
label=iris.target_Names[i],
                cmap=plt.cm.Paired)

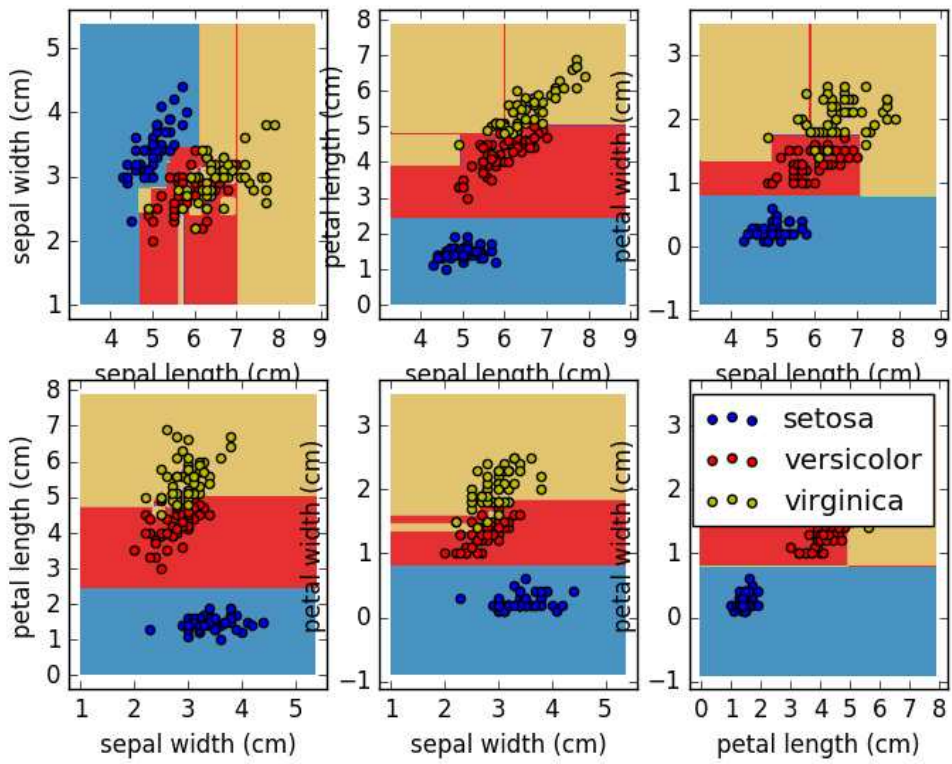
plt.axis("tight")

plt.suptitle("Decision surface of a decision tree using paired features")
plt.legend()
plt.show()

```

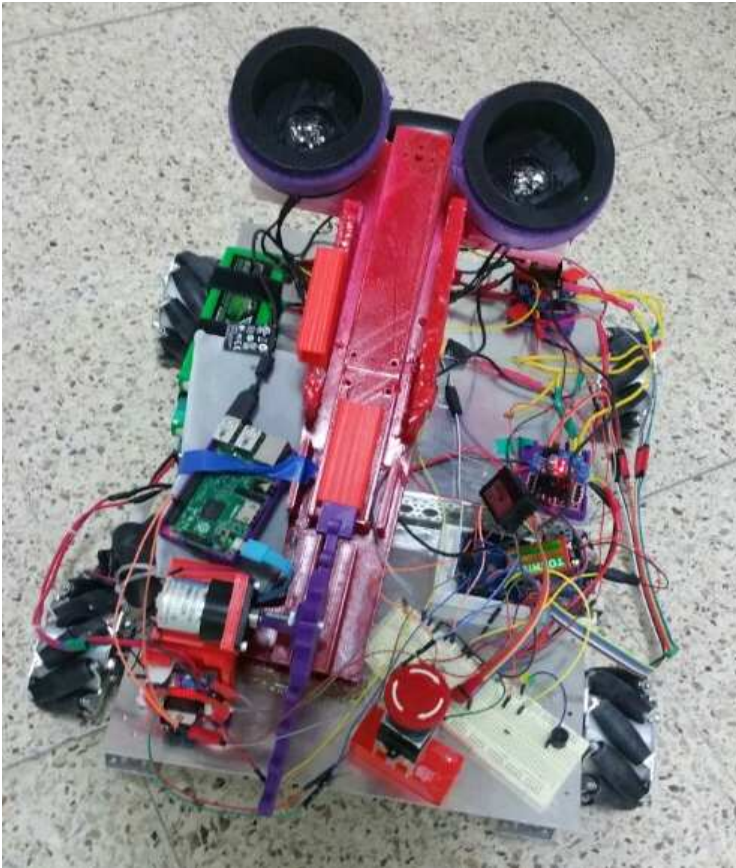
□ 참고 코드 실행 결과

Decision surface of a decision tree using paired features



로봇제어 실습

1. 손가락 인식을 통한 움직임 제어 로봇
2. 화염 학습을 통한 소방 로봇
3. 자율주행 로봇 만들기



<대단원 소개>

이번 단원에서는 영상처리 기술과 기본 적인 머신 러닝 기술을 이용하여 자율 주행 및 활동하는 로봇을 직접 제작해 본다.

01 손가락 인식을 통한 움직임 제어로봇

학습목표

- 기본적인 영상인식을 통한 로봇제어 프로그램을 작성할 수 있다.

1. 로봇 제작을 위한 구성

이번 장에서는 영상처리 기술을 이용하여 손가락의 움직임으로 동작하는 로봇을 만들어 본다.

	<p>라즈베리파이 X 1 -영상처리를 위한 모듈로 활용</p>
	<p>HC-06 X 2 - 컴퓨터와 블루투스 통신을 위해 사용</p>
	<p>아두이노 X 2 -로봇 모션 제어를 위해 사용</p>
	<p>이산솔루션 교육용 모바일 로봇</p>

표 10 로봇 제작 주요 재료

여기서는 이산솔루션의 학습용 모바일 로봇을 플랫폼과 라즈베리파이, 무블르투스 통신 모듈, 아두이노 그리고 통신을 위한 탭PC를 이용하여 손가락을 인식하여 전후, 좌우를 움직이는 로봇을 제작한다.

본 로봇의 기본 운용 방식은 간단하다. 모바일 로봇에 라즈베리 파이와 카메라를 장착하고 openCV를 이용하여 영상 즉 손가락의 개수를 인식한다. 본 실습에서는 단순하게 openCV를 사용하는 수준이나 추후에는 좀 더 다양한 모션을 이용하기 위해 머신 러닝 기법을 적용하기로 한다. 다만 여기에서는 다음장에서 실습할 내용을 위하여 영상처리 방법을 익히는데 중점을 두기로 한다. 다음 그림은 본 장에서 실습해 볼 로봇의 기본 아이디어를 설명한 것이다.

openCV
 OpenCV(Open Computer Vision)은 오픈 소스 컴퓨터 비전 라이브러리이다. 원래는 인텔이 개발하였다. 윈도, 리눅스 등의 여러 플랫폼에서 사용할 수 있다. 실시간 이미지 프로세싱에 중점을 둔 라이브러리이다. 인텔 CPU에서 사용되는 경우 속도의 향상을 볼 수 있는 Intel Performance Primitives (IPP)를 지원한다.

아이디어



그림 96 손가락으로 제어 하는 로봇의 아이디어

위 그림과 같이 기본적인 원리는 이해하기 어렵지 않다. 손가락의 수와 모양을 카메라로 인식하고 이를 openCV 라이브러리를 활용하여 프로그램으로 처리하여 각 손모양에 따라 로봇의 움직임을 달리 한다.

기술분야

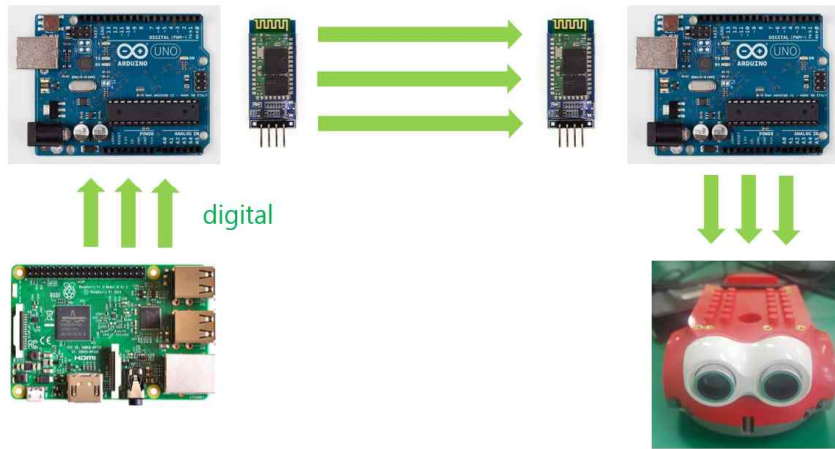


그림 97 로봇에 활용된 장치와 장치간의 데이터 전송 기술

실험에 사용된 로봇은 이산솔루션의 교육용 로봇 플랫폼을 이용하였다. 여기에 영상을 촬영할 카메라와 처리할 라즈베리파이 보드를 모두 설치할 수 없기 때문에 라즈베리파이와 1개의 아두이노 보드를 이용하여 영상을 처리하고 그결과를 로봇에 블루투스로 전송하는 방식을 이용하였다.

2. 기초 실습

2.1 프레임 캡처

이 예제는 간단히 프레임을 캡처하고 그것을 보여주는것이다. 라즈베리 파이보드에서 카메라를 설치하고 리눅스시스템 위에서 openCV와 파이썬을 활용한 기초적인 예제이다.

```
import cv2
import numpy as np          #importing libraries
cap = cv2.VideoCapture(0)   #creating camera object
while( cap.isOpened()):
    ret,img = cap.read():   #reading the frames
    cv2.imshow('input',img) #disPlaying the frames
    k=cv2.waitKey(10)
    if k == 27:
        break
```

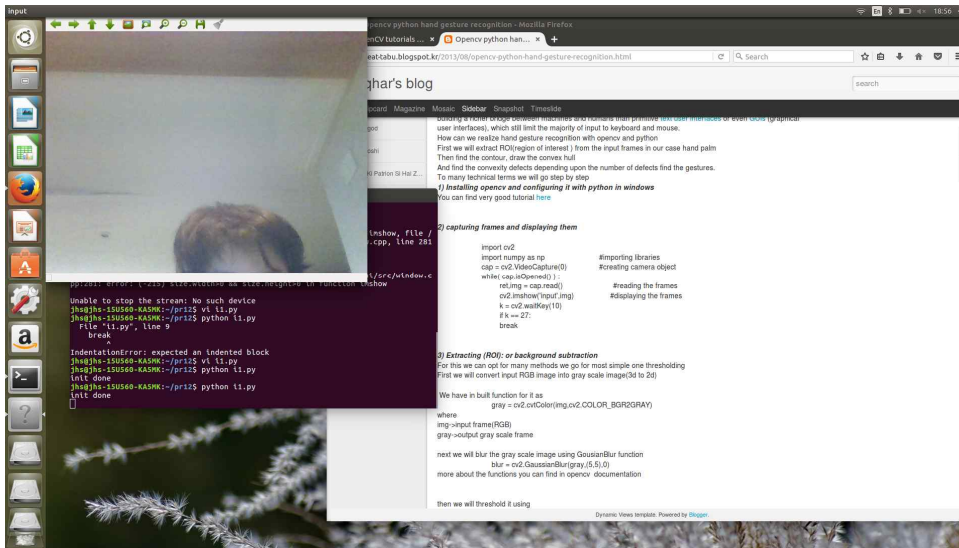


그림 98 리눅스를 탑재한 라즈베리파이에서 openCV를 이용한 영상 캡처

2.2 영상의 이진화 작업

프레임을 캡처한 후에는 이제는 찍어낸 사진을 아래 사진처럼 손만 남기고 구별을 해야 내야한다. 그러기 위해서는 색깔을 통해 구별을 해야 된다. 보통 이때 사용하는 방법은 두가지의 방법 있는데 살색을 통해서 구별 하는 것과 회색을 중심 색깔을 잡아 그보다 어둡게 되면 검은색, 밝으면 흰색으로 잡는다.

첫 번째 방법은 살색을 통해서 잡는데 실제로 실험 해보면 인식은 잘하나 주변 환경의 영향을 많이 받기에 주변 환경의 영향을 덜 받는 후자의 방법을 선택해야 한다.



그림 99 손영상의 이진화


```

import cv2
import numpy as np          #importing libraries
cap = cv2.VideoCapture(0)  #creating camera object
while( cap.isOpened()):
    ret,img = cap.read(): #reading the frames
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray,(5,5),0)
    ret,thresh1 = cv2.threshold(blur,70,255,cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
    cv2.imshow('input',img) #displaying the frames
    k=cv2.waitKey(10)
    if k == 27:
        break

```

2.3 윤곽선 따기

다음으로 해야 될 작업은 윤곽선 따기이다.

손가락의 윤곽선을 잡게되면 손가락을 내밀었는지의 판단이 가능해지기 때문에 손가락의 감지를 위해 해야 되는 과정이다 손가락의 윤곽선을 따되 아래의 사진처럼 따야될 것이다.



그림 100 손(가락)윤곽선 따기

이와 같이 우리는 다섯 손가락을 윤곽선을 따는데 손가락의 끝점을 이용해 따야 된다. 아래의 파이썬 코드를 추가 하게 되면 위의 사진처럼 추출할 수 있다.

```

for i in range(len(contours)):
    cnt=contours[i]
    area = cv2.contourArea(cnt)
    if(area>max_area):
        max_area=area
        ci=i
        cnt=contours[ci]
        hull = cv2.convexHull(cnt)

drawing = np.zeros(img.shape,np.uint8)
cv2.drawContours(drawing,[cnt],0,(0,255,0),2)
cv2.drawContours(drawing,[hull],0,(0,0,255),2)

```

다음 코드는 첫 번째로 가장 큰 윤곽선을 추출한 뒤 손가락의 꼭짓점 부분을 잡은 뒤 그것을 윤곽선을 나타내는 소스이다.

```

import cv2
import numpy as np          #importing libraries
cap = cv2.VideoCapture(0)   #creating camera object
while( cap.isOpened()):
    ret,img = cap.read():    #reading the frames
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray,(5,5),0)
    ret,thresh1 = cv2.threshold(blur,70,255,cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
    cv2.imshow('input',img) #disPlaying the frames
    k=cv2.waitKey(10)
    if k == 27:
        break

```

이제 아래 사진과 같이 손가락 사이의 중앙부분을 잡아 점을 표현한다. 이 작업은 손가락사이의 각도를 나타내 손가락의 개수를 판단해내는데 쓰인다. 여기서 실험하는 과제가 손가락의 개수는 움직임을 결정하는 명령이므로 손가락 개수는 정확하게 파악되어야 한다.

```

hull = cv2.convexHull(cnt,returnPoints = False)
defects = cv2.convexityDefects(cnt,hull)

mind=0
maxd=0
i=0
for i in range(defects.shape[0]):
    s,e,f,d = defects[i,0]
    start = tuple(cnt[s][0])
    end = tuple(cnt[e][0])
    far = tuple(cnt[f][0])
    dist = cv2.pointPolygonTest(cnt,centr,True)
    cv2.line(img,start,end,[0,255,0],2)
    cv2.circle(img,far,5,[0,0,255],-1)
    print(i)

```

4. 로봇과 연결하기

이산솔루션의 로봇은 아두이노로 제어된다. 따라서 라즈베리로 처리한 영상을 이용하여 아두이노에 적절한 모션 신호를 넣어주어야 한다.

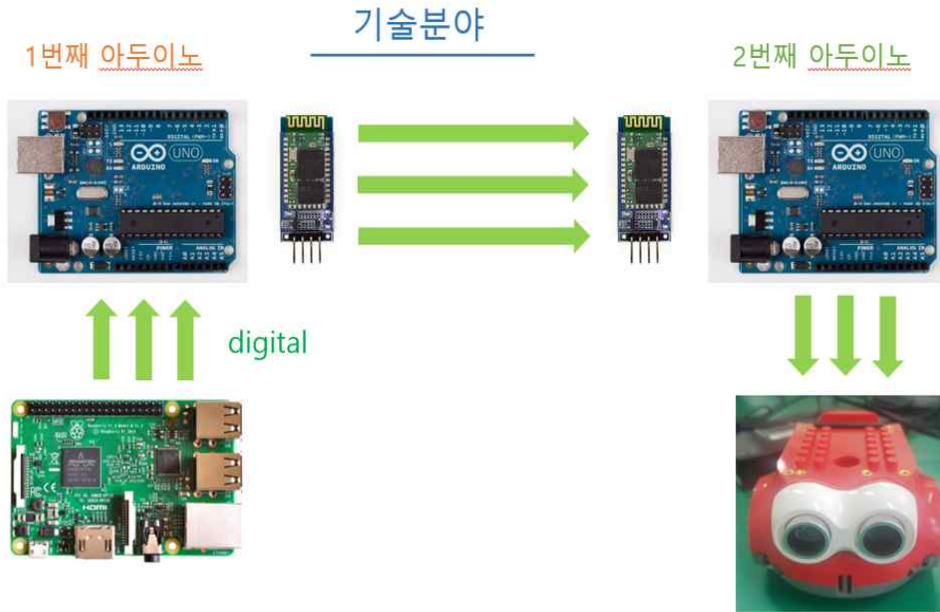


그림 101 라즈베리파이 보드와 로봇과 통신

다음은 인식한 영상을 이용하여 움직임을 제어하기 위해 구현한 스케치 파일이다.

```
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(2,3);
char a='x';

void setup() {
  pinMode(8,INPUT);
  pinMode(9,INPUT);
  pinMode(10,INPUT);
  pinMode(11,INPUT);
}
```

```
pinMode(12,INPUT);
Serial.begin(9600);
bluetooth.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(digitalRead(8)==1)
  {
    Serial.println('8');
    bluetooth.write('x');
    //0
  }
  else if(digitalRead(9)==1)
  {
    Serial.println('9');
    bluetooth.write('b');
    //3
  }
  else if(digitalRead(10)==1)
  {
    Serial.println('a');
    bluetooth.write('d');
    //5
  }
  else if(digitalRead(11)==1)
  {
    Serial.println('b');
    bluetooth.write('c');
    //4
  }
}
```

else if(digitalRead(12)==1)
{
Serial.println('c');
bluetooth.write('a');
//2
}
else
delay(10);
}

다음은 로봇 수신 코드이다. 이 코드는 송신속의 신호를 받아 로봇을 전후 좌우로 움직이게 한다.

#include <SoftwareSerial.h>
SoftwareSerial bluetooth(2,3);
char a='x';
void Go();
void Left();
void Right();
void Back();
void STP();
void setup() {
pinMode(8,OUTPUT); // 방향 제어하는 핀
pinMode(9,OUTPUT); // 방향 제어하는 핀
pinMode(10,OUTPUT); // 속도 제어하는 핀
pinMode(11,OUTPUT); // 속도 제어하는 핀
Serial.begin(9600);
bluetooth.begin(9600);
}

```
void loop() {  
  // put your main code here, to run repeatedly:  
  if (bluetooth.available())  
    char a=bluetooth.read();  
  
  if(a=='b')  
    Go();  
  else if(a=='c')  
    Left();  
  else if(a=='a')  
    Right();  
  else if(a=='d')  
    Back();  
  else  
    STP();  
  
  delay(10);  
}  
  
void Go()  
{  
  digitalWrite(8,1);  
  digitalWrite(9,1);  
  analogWrite(10,125);  
  analogWrite(11,125);  
}  
  
void Left();  
{  
  digitalWrite(8,1);
```

```
digitalWrite(9,0);
analogWrite(10,125);
analogWrite(11,125);
}
void Right();
{
digitalWrite(8,0);
digitalWrite(9,1);
analogWrite(10,125);
analogWrite(11,125);
}
void Back();
{
digitalWrite(8,0);
digitalWrite(9,0);
analogWrite(10,125);
analogWrite(11,125);
}
void STP();
{
digitalWrite(8,0);
digitalWrite(9,0);
analogWrite(10,0);
analogWrite(11,0);
}

```

5. 동작

5.1 동작 확인

동작은 영상에서 손만 인식하여 움직이는 것을 보여준다. 주먹은 정지가위는 전진 그리고 손가락 개수에 따라 좌회전과 우회전을 한다.

5.1.1 Stop(묵 or 감지x)

오픈소스 라이브러리인 openCV의 기능이 우수하여 화면에서 손모양을 정확하게 잘 인지하여 준다.



그림 102 주먹 영상인식



그림 103 라즈베리파이로 처리된 명령을 로봇에 전달

5.1.2 전진(가위)

가위는 전진을 의미한다. 여기서는 두 손가락을 적당히 벌려 인지하는 것이 중요하다.

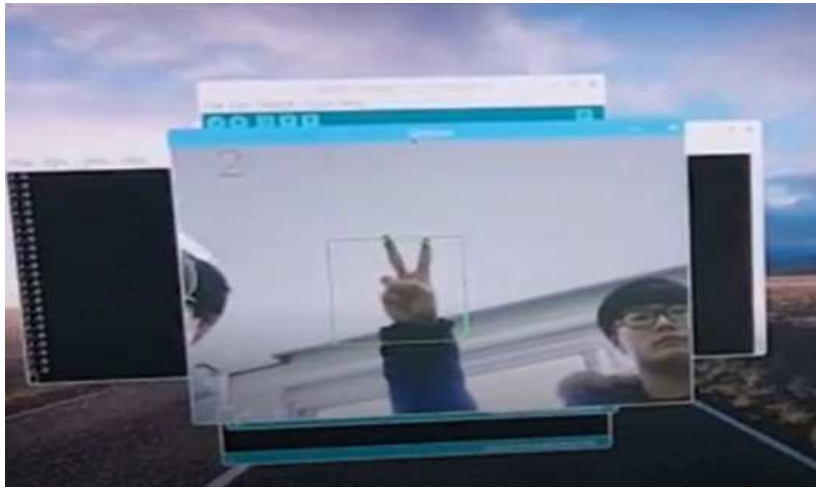


그림 104 전진 신호의 인식

손가락 인식

손가락의 개수 인식은 전통적으로 많이 활용되는 문제이기 때문에 openCV 라이브러리를 활용하면 잘 인식된다.

5.1.3 좌회전 (3개손가락)

손가락이 3개이면 좌회전을 하게 프로그래밍 하였다. 간혹 두 개와 세 개를 명확하게 인식 못할 때가 있다.



그림 105 손가락 3개 인식

5.1.4 좌회전(4개손가락)

손가락 4개는 좌회전을 의미한다. 손가락 4개는 움직이는 모양에 따라 5개와 혼동이 될 때가 많다.



그림 106 손가락 4개의 인식

5.1.5.후진(5손가락)

손을 완전히 펴서 5개의 손가락을 모두 보이면 후진을 한다. 5개는 예상과 달리 명확하게 인식되는 경우가 많다.



그림 107 손가락 5개 인식

02 화염 학습을 통한 소방 로봇

학습목표

- 기본 화염 영상을 학습하여 화염을 인지하고 소화하는 로봇을

1. 로봇 제작을 위한 구성

본 장에서는 화염 영상을 학습하여 스스로 화염을 인지하여 소화하는 로봇을 제작하여 본다. 전체적인 외형은 아래 사진과 같다.

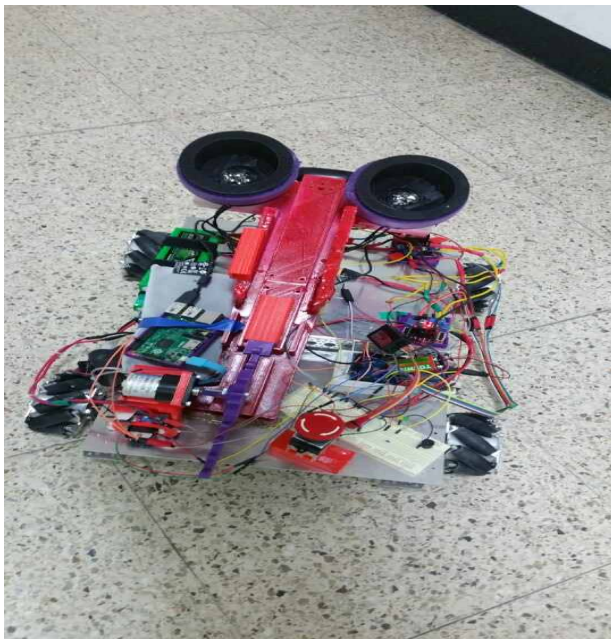


그림 108 제작한 소방로봇

소방로봇은 특징으로는 유압모터를 사용해 물을 펌프를 통해 물을 발사하는 것이 아닌 투척용 소화기를 이용하자는 목표로 bldc를 사용해 다른 소방로봇과 다른 차이점을 주었고 우리는 두 가지 모드로 동작할 수 있도록 한다.

첫 번째로는 사람이 조종하여 직접 발사하는 것과 버튼을 눌러 프로그램 모드를 사용해 영상인식을 통해 불을 감지하여 스스로 끄는 모드가 있다. 그리고 또한 바퀴에도 차별 점을 두어 메카넘 휠을 사용해 일반 바퀴로써 나올 수 없는 좌우 이동을 사용해 일반 로봇이 가려면 시간이 걸리는 부분

을 좀 더 쉽고 빠르고 이동할 수 있다. 아래 사진은 실제 사용하는 투척용 소화액과 메카넘 휠이다.



그림 109 투척용 소화액

실제 화재 시에는 당황하는 경우도 많고 인체에 해를 입는 경우도 많아 일반적인 분말 방사식 소화기 보다 투척용 소화기가 효과를 볼 때가 많다.



그림 110 전후좌우 움직임이 가능한 메카넘 휠

본 로봇을 제작하기 위한 물품은 다음과 같다.

	물품명	모델명
1	DC모터	12V DC 기어드모터 120RPM
2	요가매트(고무)	친환경요가매트16/10mm
3	dc모터드라이버	dc모터드라이버
4	카메라	(로지텍) HD WebCam C310 정품 화상카메라
5	BLDC모터	EMAX BLDC 모터
6	변속기	하비윙 XRotor 20A Wired Type (HobbyWing ESC)
7	스텝모터	일체형 스텝모터 SBC-NK245-03AT
8	DEVO7	Devo 7 조종기 + RX701수신기
9	적외선 온도센서	적외선 온도센서 (Infrared Thermometer - MLX90614)
10	투척용 소화기	투척용 소화기
11	배터리	Multistar High Capacity 3S 5200mAh Multi-Rotor Lipo
12	SD카드	라즈베리파이3 마이크로 SD 32GB 마이크로 SD 리더기
13	순간접착제	순간접착제
14	3M 양면테이프	3M 양면테이프

표 12 로봇 제작용 부품 리스트

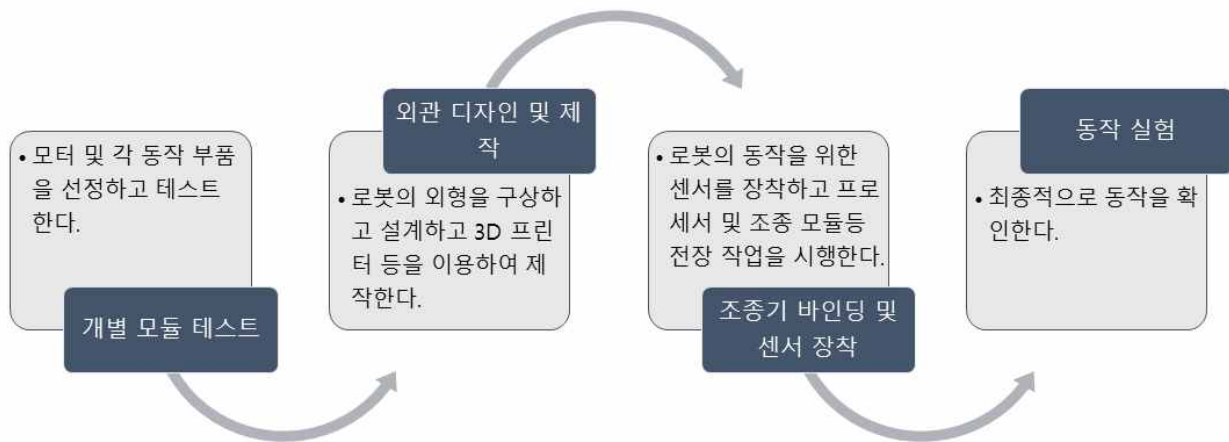


그림 111 소화로봇 제작과정

2. 개별 모듈 테스트

로봇제작을 위하여 제일 먼저 개별 부품을 선정하고 테스트를 수행하였다.

2.1 모터 테스트

```
int on_off = 8;
int cw_ccw = 7;
int step = 6;

void setup()
{
  pinMode(on_off, OUTPUT);
  pinMode(cw_ccw, OUTPUT);
  pinMode(step, OUTPUT);
}

void loop()
{
  digitalWrite(on_off, HIGH);
  digitalWrite(cw_ccw, LOW);
  delay(3000);
  digitalWrite(on_off, HIGH);
  digitalWrite(cw_ccw, HIGH);
  delay(3000);
  digitalWrite(on_off, LOW);
  delay(1000);
}
```

먼저 아두이노 보드와 모터드라이버를 이용하여 로봇제작에 활용할 스텝모터의 구동 실험을 하였다. 위 코드를 이용하여 3초 간격으로 구동, 정지를 반복하였다. 모터구동 드라이버는 아래와 같은 형태를 사용하였다.

그림 112 스텝모터 구동 코드

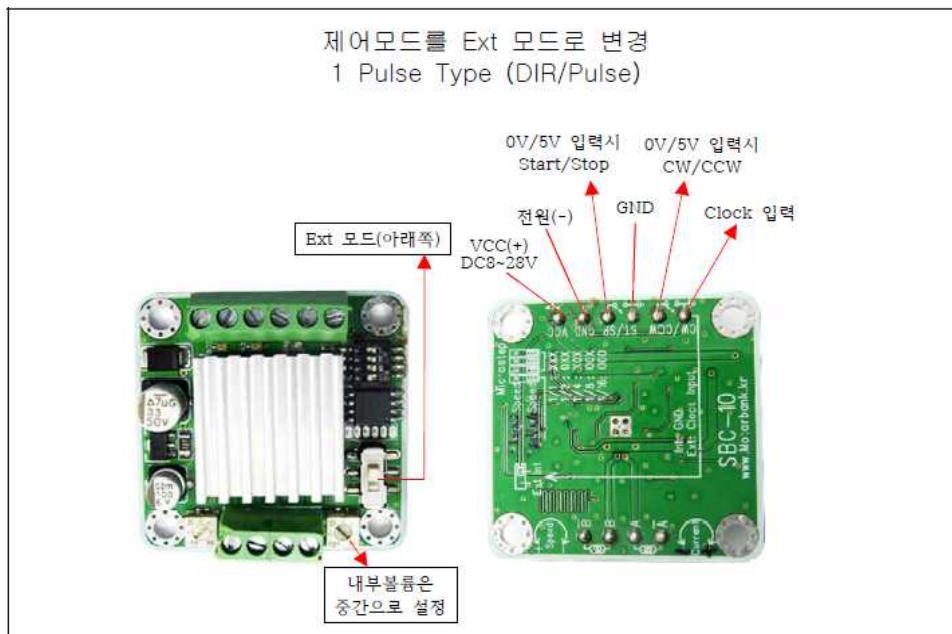


그림 113 모터 구동 드라이버



그림 114 실제 테스트 구성

2.2 센서 테스트(온도)

화염을 인지하여 소화하는 로봇이므로 화염의 온도를 인지하는 기능은 당연히 있어야 한다. 따라서 온도센서를 장착하였다. 그리고 온도센서의 실험회로를 만들어 실험하였다.

```
#include <Wire.h> #include <Adafruit_MLX90614.h> Adafruit_MLX90614 mlx = Adafruit_MLX90614();
void setup()
{ Serial.begin(9600); Serial.println("Adafruit MLX90614 test"); mlx.begin(); //mlx모듈을 읽어들이기 시작합니다.
}
void loop()
{
  Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempC()); // 주변온도를 읽습니다.
  Serial.print("C\n");
  Serial.print(mlx.readObjectTempC());
  Serial.println("C"); //객체 온도를 읽습니다.
  Serial.print("Ambient = ");
  Serial.print(mlx.readAmbientTempF());
  Serial.print("F\n");
  Serial.print(mlx.readObjectTempF());
  Serial.println("F"); //화씨로 읽어들이니다.
  Serial.println();
  delay(500);
}
|
```

그림 115 온도센서 실험 코드

그러나 실제 화염에 대하여 테스트 하지는 못하였다. 이유는 학교에서 화염을 가지고 실험하기에는 위험이 너무 다르기 때문이다. 따라서 화염 사진을 이용하였으며 실제 로봇의 구동에는 온도센서는 이용하지 않았다.

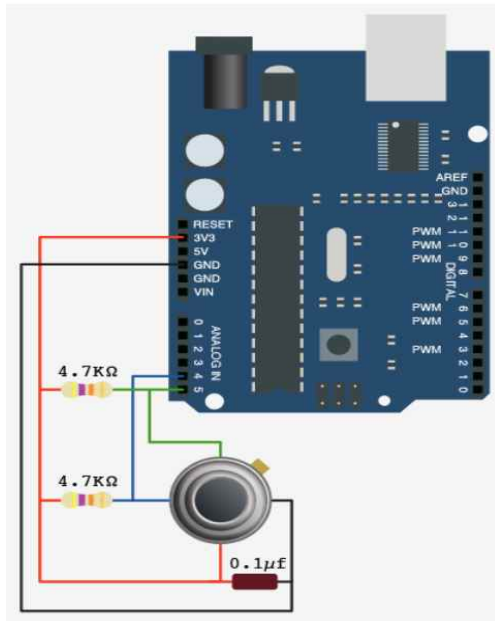


그림 116 온도센서와 아두이노보드

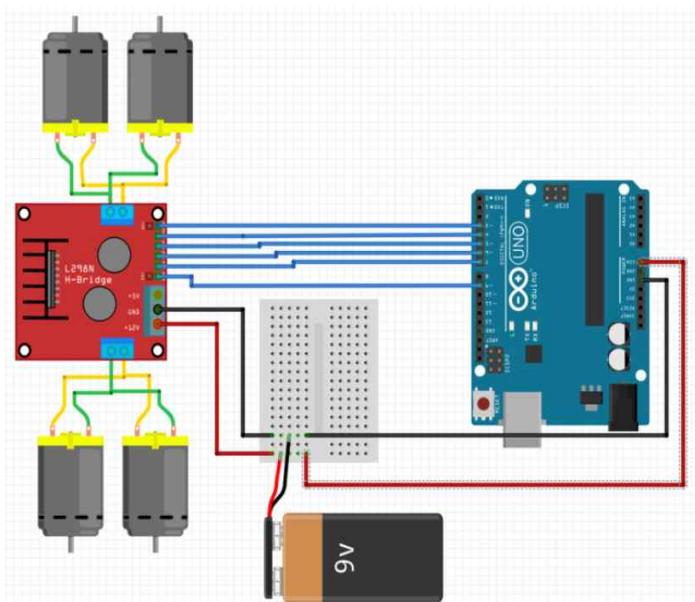


그림 117 모터구동테스트를 위한 회로 연결

2.3 모터 테스트

바퀴 구동을 위한 모터는 직류 모터를 이용하였다. 직류 모터의 정역회전을 위한 모터 드라이버는 쉽게 구할 수 있는 L298N을 이용하였다.

```
#define IN1 7
#define IN2 6

#define IN3 5
#define IN4 4

#define ENA 9
#define ENB 3

//모터 번호
#define CH1 0
#define CH2 1
#define ALL_CH 2

void setup() {
  // put your setup code here, to run once:
  //핀 설정
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
}

//앞으로
void forward(){
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
```

```
//오른쪽
void turnRight(){
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

//스톱
void stopMotor(){
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
}

void loop() {
  // put your main code here, to run repeatedly:
  //최대 속도로 동작
  setMotorSpeed(ALL_CH, 255);

  //앞으로 가기
  forward();
  delay(1000);
  //뒤로가기
  backward();
  delay(1000);
  //왼쪽으로 돌기
  turnLeft();
  delay(1000);
  //오른쪽으로 돌기
  turnRight();
  delay(1000);
}
```

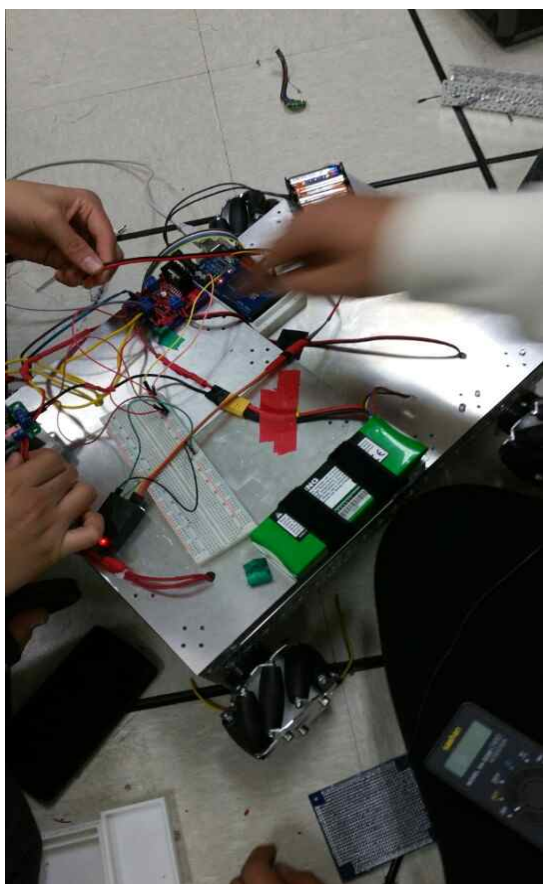
```
//앞으로
void forward(){
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

//뒤로
void backward(){
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

//왼쪽
void turnLeft(){
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
}

//오른쪽
void turnRight(){
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
```

그림 118 모터 구동 테스트를 위한 단순 코드



모터드라이브는 보이는 코드와 같이 2개의 신호선으로 1개의 모터를 컨트롤 한다. 이후 이 코드를 로봇에 적용하였다.

그림 119 테스트된 모듈의 메인 보드 장착

3. 외관디자인 및 제작

로봇제작을 위하여 제일 먼저 개별 부품을 선정하고 테스트를 수행하였다.

3.1 슈터 제작

화재진압 로봇에 특성상 투척용 소화기를 투척할 수 있는 슈터가 필요하다. 그러므로 소화기를 장착할 수 있는 공간을 만들어 주었고 빨간색 원 안에 있는 모형은 밑에 있는 슈터 같은 형태로 BLDC모터를 구동시켜서 소화기를 던질 수 있도록 제작하였다.

BLDC모터의 힘이 강력해서 투척용 소화기가 날아가는 방향이 정확하지 않을 수 있으므로 옆에 가이드를 세웠다.



그림 120 야구공 슈터

외형 제작
외관을 설계한후 대부분 3D 프린터로 출력하였다. 일부 판형의 외관은 평판 절삭기를 이용하였고 인벤터나 Cadian 같은 프로그램을 사용하여 제작하였으므로 로봇이나 기계를 전공하는 학생이면 쉽게 제작할 수 있다.

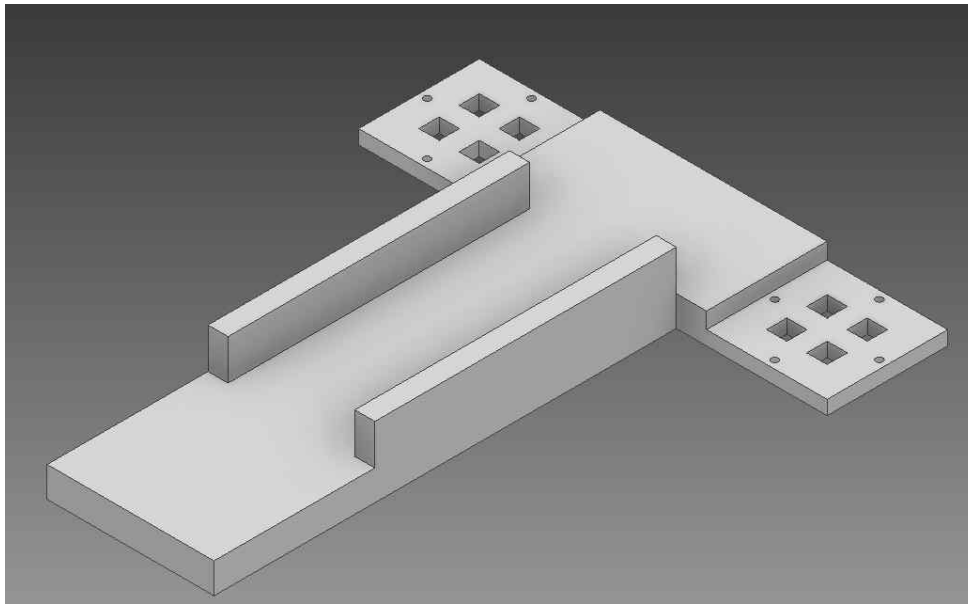


그림 121 소화액 투척을 위한 슈팅 가이드

3.2 카메라 가이드 제작

화재진압 로봇은 화염 영상을 인지하여 동작하므로 화염을 촬영할 카메라를 안정적으로 장착하여야 하므로 카메라 장착을 위한 가이드를 제작한다.

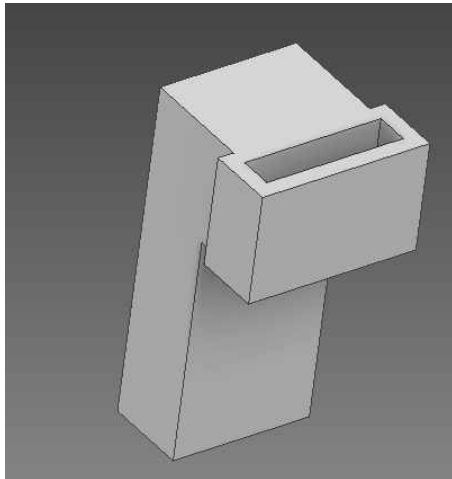


그림 122 카메라 가이드

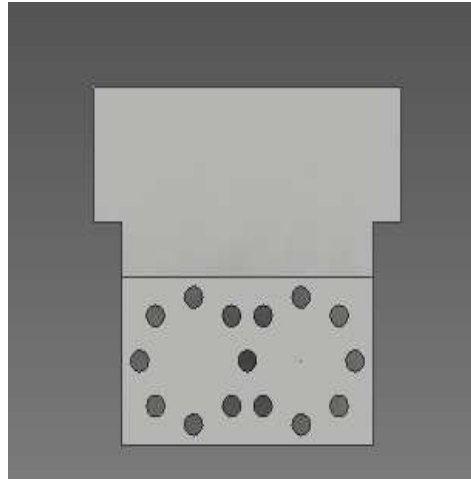


그림 123 카메라 가이드 2

카메라 고정대는 카메라에 치수를 측정한 후 그에 맞는 높이와 구멍을 만들어서 고정시켰다. 그림 117은 카메라 고정부품의 밑부분이다. 화재진압 로봇의 베이스는 나사홈이 2m이므로 그의 맞는 구멍과 규격의 맞게 설계하여 나사로 고정시켰다.

3.3 소화액 투척 발사대 기어부

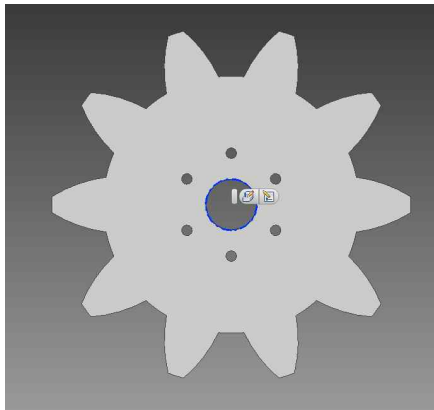


그림 124 발사대 동작기어(피니언)

몸체를 제작한 후 버튼을 누르면 투척용 소화기가 발사될 수 있도록 뒤에서 밀어주는 기계요소가 필요하다. 그래서 DC 모터의 기어를 장착해서 밀어주는 기어부를 제작한다.

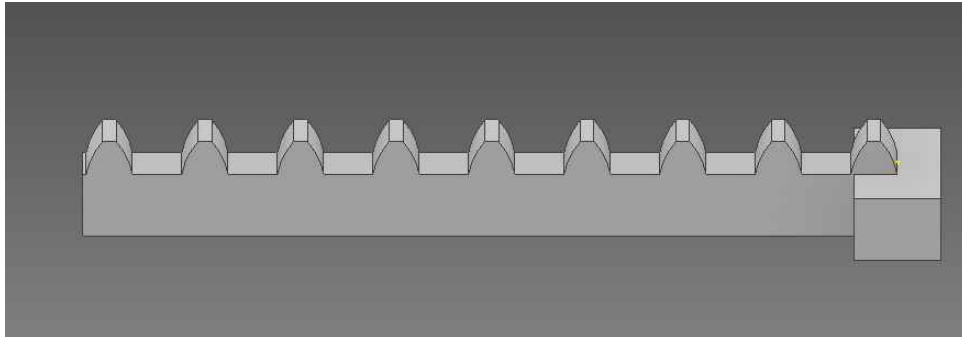


그림 125 발사대 동작부 랙

앞에 부품(기어)가 돌아가면 밀 수 있도록 기어의 이 규격대로 설계해 주었고 기어가 돌아가게 되면 이가 맞물려 앞으로 나아가게 된다.

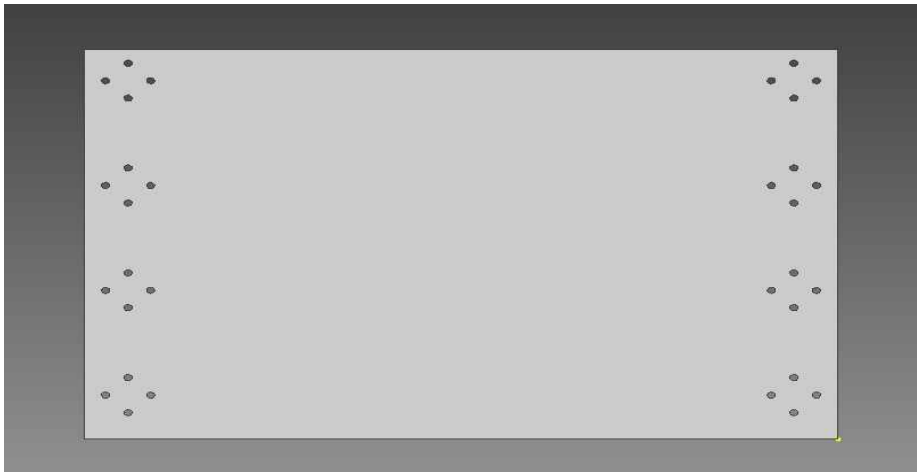


그림 126 로봇 기본 플레이트

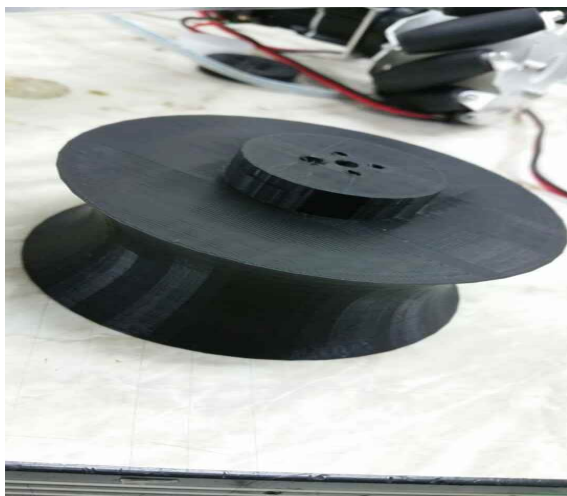


그림 127 3D로 제작한 슈터 일부

여기서는 플레이와 샤프시를 TETRIX의 부품을 사용하였으나 실제 적용을 하고자 할 때는 각종 샤프시 및 다양한 프레임을 이용할 수 있다. 가능하다면 평판 밀링 등을 이용하여 필요한 플레이트와 샤프시를 제작할 수도 있다. 고속회전을 하는 기어부나 강한 힘을 받아야 하는 플레이트를 제외하고는 3D프린터를 이용한다.

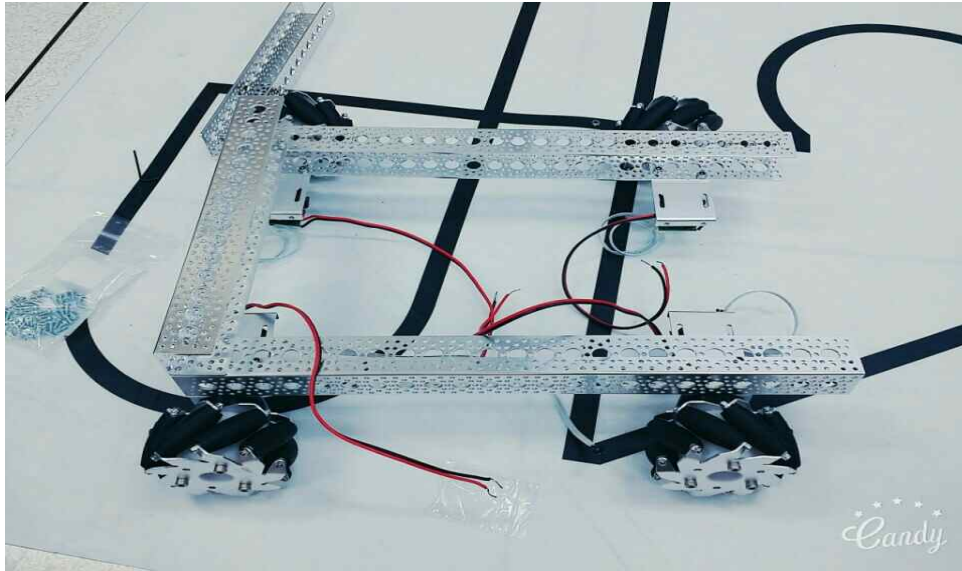


그림 128 TETRIX 샴시를 이용한 로봇 몸체 제작

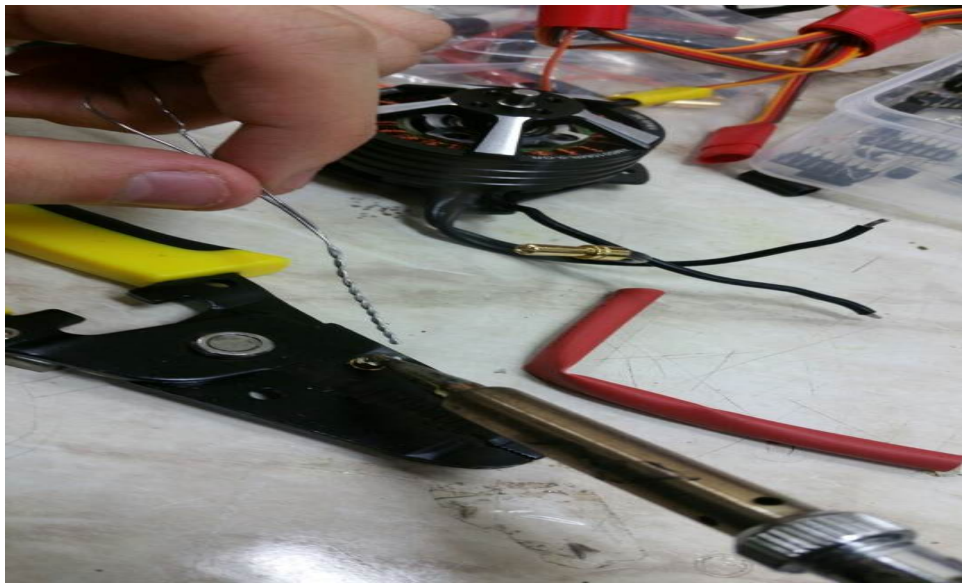


그림 129 모터연결을 위한 납땜 작업

기본적인 몸체 제작을 마치면 센서 장착과 프로세서와 센서들의 연결 모터의 연결을 위한 납땜 및 전장작업을 실시한다. 전장 작업을 할 때는 움직임에 흔들리지 않도록 잘 고정한다.

4. 프로그램 제작

로봇제작을 위하여 제일 먼저 개별 부품을 선정하고 테스트를 수행하였다.

4.1 영상인식을 위한 파이썬 코드

```
import cv2

        7
import RPi.GPIO as GPIO
import time
import numpy as np
=====①
GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(17,GPIO.OUT)

#setupvideocapture#####

cam=cv2.VideoCapture(0)
cam.set(cv2.CAP_PROP_FRAME_WIDTH,320)
cam.set(cv2.CAP_PROP_FRAME_HEIGHT,240)
=====②
font=cv2.FONT_HERSHEY_SIMPLEX
#setupvideocapture#####

#othersettingVariable#####

lower_color =np.array([0,145,145])#hsv color
upper_color =np.array([20,255,255])
=====③
kernel=np.ones((7,7),np.uint8)#remove noise Var
#othersettingVariable#####
```

```

#mainloop#####

if __Name__=='__main__':
    try:
        while(1):
            _, frame=cam.read()
            hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)#convert
hsv_color
            detect_color(hsv,frame)
            k=cv2.waitKey(5)&0xFF #=====④
            if k==27:
                break
        except KeyboardInterrupt:
            pwm1.stop()
            GPIO.cleanup()

            #mainloop#####

#detectedredcoloranddrawrectangle#####

def detect_color(img,imgor):
    mask=cv2.inRange(hsv,lower_color,upper_color) #find range
ofhsvcolor

    opening=cv2.morphologyEx(mask,cv2.MORPH_OPEN,kernel)#remove
noise Function
    x,y,w,h=cv2.boundingRect(opening)#draw rectangle

    wow="temp:"+str((x+w)/2)

```



```

imgor=cv2.rectangle(imgor,(x,y),(x+w,y+h),(0,255,0),2)

cv2.putText(imgor,wow,(10,10),font,0.5,(255,255,255),2,cv2.LINE_AA)
=====⑤
cv2.imshow('frame',imgor)
cv2.imshow('mask',mask)

#print((x+w)/2)
if(0<(x+w)/2<70):
    print("Left")
    GPIO.output(18,True)
    GPIO.output(17,False)
elif(160>(x+w)/2>100):
    print("Right")
    GPIO.output(18,False)
    GPIO.output(17,True)
elif(100>=(x+w)/2>=70):
    print("Centor")
    GPIO.output(18,True)
    GPIO.output(17,True)
else:
    print('not detect')
    GPIO.output(18,False)
    GPIO.output(17,False)

#detect red color and draw rectangle#####

```

표 13 영상인식을 위한 파이썬 코드

-코드 설명

- ①: 영상제어와 I/O제어에 필요한 헤더파일들을 먼저 정의시켰다. 그리고 GPIO.set이라는 함수를 이용해서 라즈베리파이의 17번 핀과 18번 핀을 출력 핀으로 설정시켰다.
- ②: 먼저 cam이라는 객체를 비디오를 위한 클래스인 cv2.VideoCapture(0)에서 만들고 영상을 불러오기 전 영상의 처리속도를 높이기 위해 cam.set이라는 함수를 사용하여 가로 세로를 각각 320x240으로 설정했다.
- ③: 영상처리를 위한 변수 등을 설정했다 (자세한 설명은 설명⑤로.)
- ④: main문에서 cam.read()함수를 읽어와 영상을 불러온 후, detect_color라는 사용자정의 함수를 이용하여 영상을 처리한다. 그리고 영상의 처리를 돕기 위해 cv2.cvtColor라는 함수를 사용하여 hsv색 영상으로 변환시킨다. 만약 키보드에서 esc버튼이 눌리면 영상프로그램을 종료하게 된다.
- ⑤: 일단 영상을 받으면 이런 영상을 가져오게 된다.



그림 130 기본 프로그램을 이용한 영상인식

cv2.cvtColor라는 함수를 사용하여 영상을 변환하는 도중 문제점이 발생했는데, 곳곳에 섞여있는 주황색과 빨간색이 아주 작은 점이 되어 잡음을 발생시켰다. 그래서 이 문제를 해결하기 위해 이진화된 영상을 보강해주는 cv2.morphologyEx라는 함수 중에서 opening라는 처리방법을 사용하였다. 이 처리는 점으로 찍혀있는 점들을 밑에 사진처럼 제거해준다.

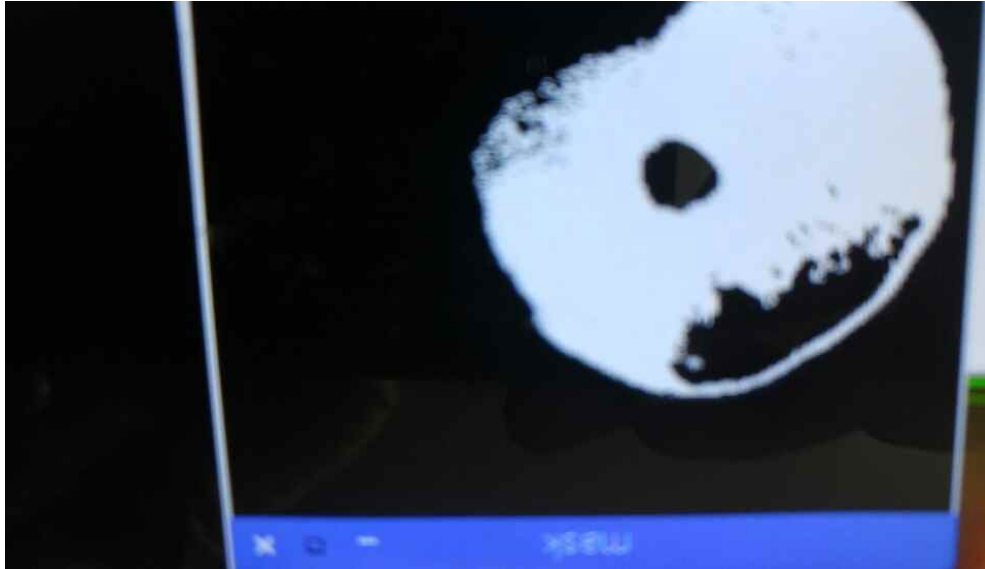


그림 131 잡음을 제거 처리



그림 132 잡음 제거 후와 제거전의 비교

이렇게 점들을 제거했지만 그중에 가장 큰문제가 남아 있다. 이 영상만으로는 불의 위치를 알 수 없었기 때문이다. 그래서 이 진화된 영상에서

흰 부분의 가장 끝점과 그다음에 반대에서 가장 끝부분을 리턴하는 cv2.boundingRect라는 함수를 이용하여 x,y,w,h라는 변수에 각각 값을 집어넣었다. 그 다음에 영상에 사각형을 그려주는 cv2.rectangle라는 함수를 이용해 기본 영상으로 받아온 imgor 영상에 사각형을 그려준다. 그리고 cv2.imshow를 이용해 영상을 띄워주면 다음과 같이 나오게 된다.

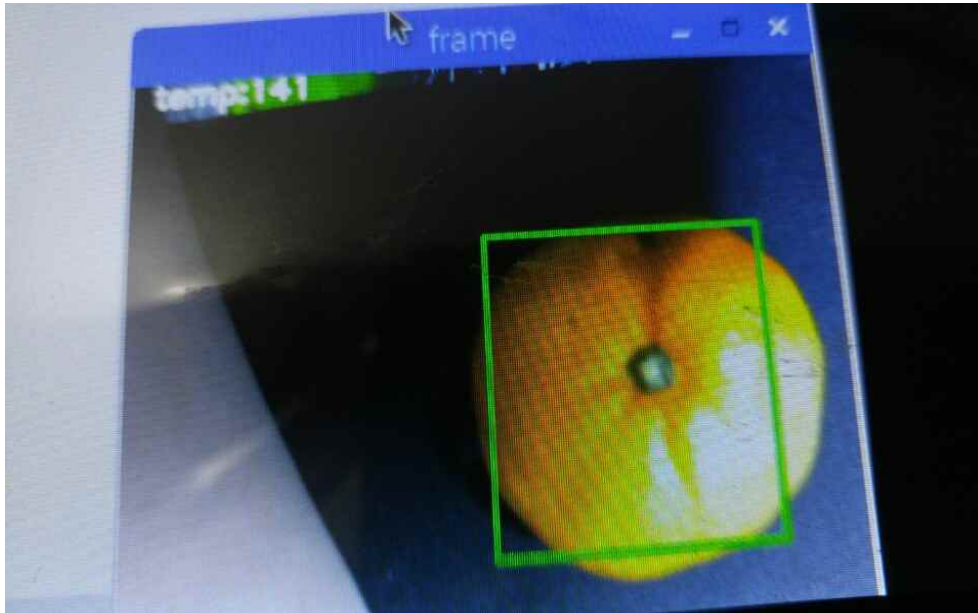


그림 133 영상에서 특정 색상 부분의 확인

이제는 어느 위치에 따라 핀에 어느 값을 넣어줄지 생각해야 한다. 그래서 2개의 핀을 이용하여 이렇게 4개의 경우의 수를 만들어 사용한다.

(X는 off, O는 on)

핀번호/동작	감지하지 못함	왼쪽에서 감지	오른쪽에서 감지	중앙에서 감지
17번 핀	X	O	X	O
18번 핀	X	X	O	O

표 14 핀과 감지점과의 연결

그리고 기준점을 잡기위하여 x값과 w값을 더한 후 나누기 2를 한 값을 기준으로 잡고 if~else문을 작성한 후 GPIO.output함수를 이용해 17번핀

과 18번핀에 적절한 값을 출력시킨다. 그리고 영상에서 바로 $(x+w)/2$ 한 값을 보기위해 cv2.putText를 이용하여 영상에 글자를 올린다. 그림 128은 지금까지 처리한 것을 보여 주는 영상이다.

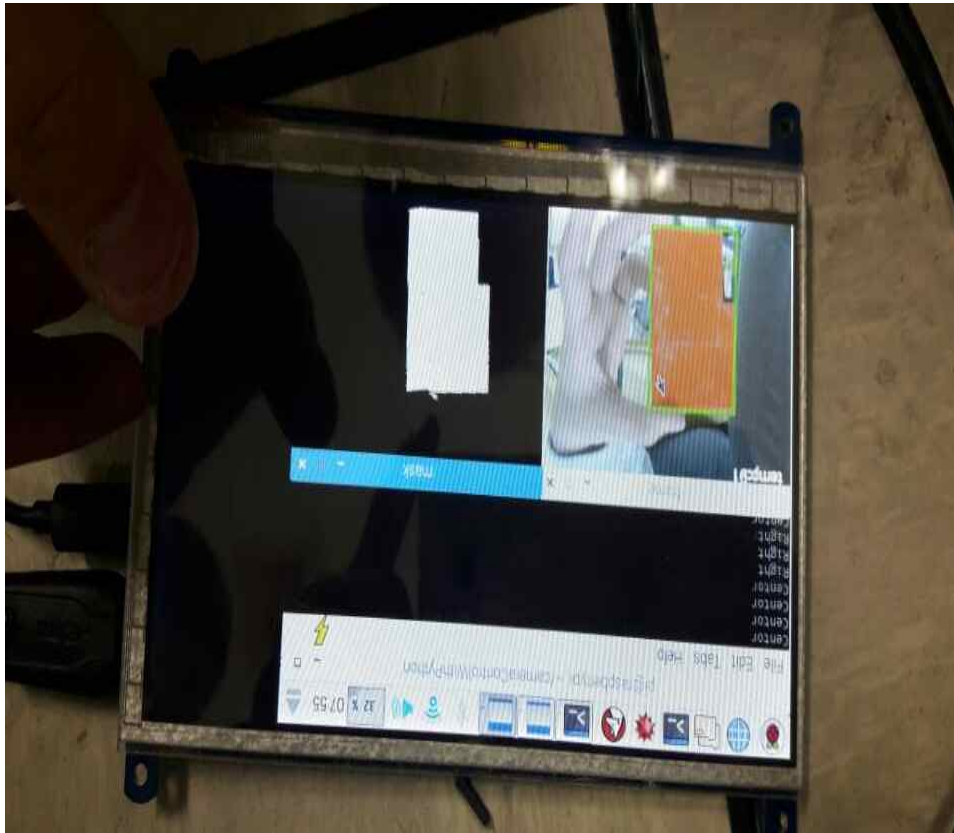


그림 134 정확한 화염과악을 위한 영상처리

4.2 화염의 정확한 인식을 위한 머신 러닝

기본적인 openCV 라이브러리만 이용하여도 화염을 인지할 수 는 있다. 그러나 화염의 모양은 다양하고 화염인지 그냥 램프인지 구별하는 것도 쉽지 않다. 따라서 여기에서는 화염의 정확한 인식을 위해 머신 러닝 기법을 이용해 보았다. 높은 수준의 머신러닝 알고리즘과 구현을 통해 많은 자료를 이용해 학습하면 좋겠지만 여기서는 머신러닝을 이용하면 좀 더 정확한 인식을 할 수 있다는 경험적인 측면을 강조한다.

알고리즘은 SVM(Support Vector Machine)을 이용하였다. Support vector Machine(SVM) 은 분류(Classification) 와 회귀(regression) 에 응용할 수 있는 지도학습 (supervised learning) 의 일종이다.

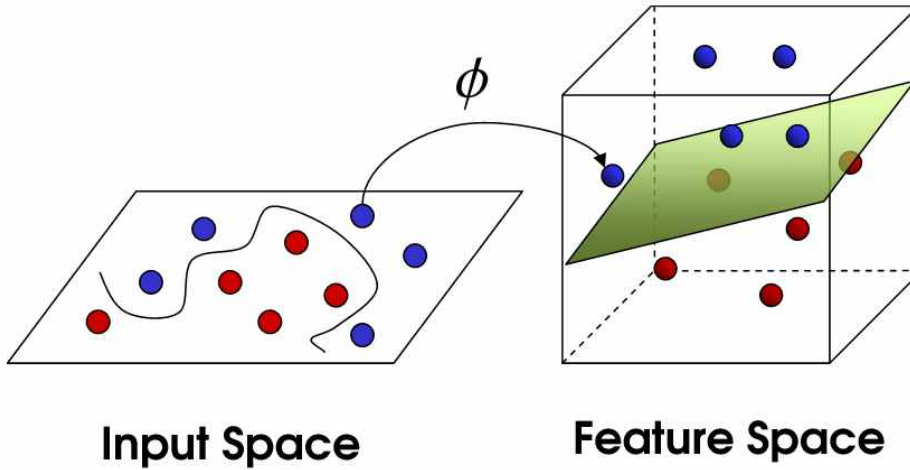
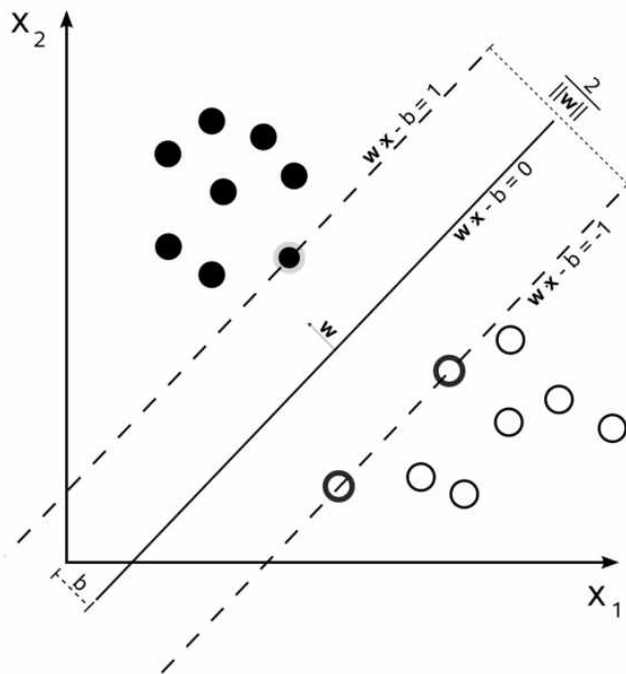


그림 135 SVM의 핵심원리 1



간단하게 설명하면, 그림 129와 같이 입력된 자료를 고려하면 붉은색과 파란색 공을 아무리 잘 나누려 해도 선형식으로 즉, 직선으로 나눌 수 없다. 그러나 각각의 요소들의 특징을 살리면서 차원을 변환하여 주면 선형의 평면으로 두 군을 구분할 수 있게 된다.

그림 136 Support Vector

이 때 중요한 것은 그림 130과 같이 두 군을 분리할 때 최대의 마진을 찾아주는 각 군의 요소를 찾는 것이다 그렇게 해야 분류의 효율이 매우 높아진다. 각 군을 분류하는 결정적인 역할을 하는 요소를 Support Vector라고 한다.

이 알고리즘을 화염 인식에 사용하기 위해 여기서는 기존에 잘 활용되는 번호판 인식용 프로그램을 활용하였다. 번호판 인식용 프로그램은 인터넷에서 쉽게 구할 수 있는데 여기서는 그 프로그램을 수정하여 그림 131과 같이 화염 사진을 이용하여 화염 인식 학습을 수행하였다.

화염 사진도 실제 것을 촬영하면 좋으나 그렇게 하지 못하고 인터넷에서 구한 것을 이용하여 실험하였다.



그림 137 학습용 화염 사진

실제학습 결과는 openCV 기본 라이브러리만 사용하였을 때 보다는 조금 향상 된다. 화염 사진과 화염에 맞는 코드로 좀 더 수정하면 효과적인 학습을 할 수 있을 것으로 생각된다. 다음은 학습에 사용한 파이썬 코드이다. 코드는 특성을 인식하는 부분 학습하는 부분 등으로 나뉘어져 있으나 여기서는 정제된 자료로부터 학습하는 부분만 수록한다.

```
import os
import sys
import argparse
import cPickle as pickle
import numpy as np
from sklearn.multiClass import OneVsOneClassifier
```

```

from sklearn.svm import LinearSVC
from sklearn import preprocessing

def build_arg_parser():
    parser = argparse.ArgumentParser(description='Trains the Classifier
models')
    parser.add_argument("--feature-map-file", dest="feature_map_file",
required=True,
help="Input pickle file containing the feature map")
    parser.add_argument("--svm-file", dest="svm_file", required=False,
help="Output file where the pickled SVM model will be stored")
    return parser

Class ClassifierTrainer(object):
def __init__(self, X, label_words):
self.le = preprocessing.LabelEncoder()
self.clf = OneVsOneClassifier(LinearSVC(random_state=0))

y = self._encodeLabels(label_words)
X = np.asarray(X)
self.clf.fit(X, y)
def _fit(self, X):
X = np.asarray(X)
return self.clf.predict(X)

def _encodeLabels(self, labels_words):
self.le.fit(labels_words)
return np.array(self.le.transform(labels_words), dtype=np.float32)

def Classify(self, X):
labels_nums = self._fit(X)

```



```

labels_words = self.le.inverse_transform([int(x) for x in labels_nums])
return labels_words

if __Name__=='__main__':
args = build_arg_parser().parse_args()
feature_map_file = args.feature_map_file
svm_file = args.svm_file

# Load the feature map
with open(feature_map_file, 'r') as f:
feature_map = pickle.load(f)

# Extract feature vectors and the labels
labels_words = [x['label'] for x in feature_map]
dim_size = feature_map[0]['feature_vector'].shape[1]
X = [np.reshape(x['feature_vector'], (dim_size,)) for x in
feature_map]

# Train the SVM
svm = ClassifierTrainer(X, labels_words)
if args.svm_file:
with open(args.svm_file, 'w') as f:
pickle.dump(svm, f)

```

표 15 SVM을 이용한 영상 학습용 샘플 코드

5. 로봇의 움직임 제어

영상 인식을 위한 프로그램은 앞서 작성하였다. 이번에는 인식된 영상에 따라 움직이는 로봇을 제작하기 위한 코드를 작성한다.

본 로봇은 사용자가 직접조종할 수 있는 모드 1과 스스로 움직이는 모드 2로 구분하여 제작한다. 라즈베리파이 보드에서는 영상처리 및 머신러닝을 담당하고 모션제어는 아두이노에서 하도록 하였다.

다음은 수동제어를 위한 코드의 일부이다.

```
#include "rc_rx.h"
#include <Servo.h>

int Fleft=5;
int Fright=7;
int Bleft=9;
int Bright=11;
int FLD=6; //Left motor
int FRD=8; //Left motor Direction
int BLD=10; //right motor
int BRD=12; //right motor Direction
int stepmotor=22,dir=23,On=24;//스텝모터 설정
int speedvalue=1000,s;
int blidcmotor1=30,blidcmotor2=32;//BLDC motor setting
Servo shoot1;
Servo shoot2;
```

변수를 설정해 준다.

```

void setup()
{
  Serial.begin(115200); // Initializing serial port before calling init_rc_rx() function
  init_rc_rx();
  pinMode(led, OUTPUT);
  for(i=5;i<13;i++)
    pinMode(i,OUTPUT);

  shoot1.attach(40);
  shoot1.writeMicroseconds(200); //BLDC motor setting
  shoot2.attach(41);
  shoot2.writeMicroseconds(200); //BLDC motor setting
  pinMode(22,OUTPUT);
  pinMode(23,OUTPUT);
  pinMode(24,OUTPUT);
}

```

조종기 셋팅과 BLDC모터 셋팅, 각각 필요한 핀 설정을 해준다.

```

if(CH2>40 && CH2<70 && CH1>40 && CH1<70 && CH4>40 && CH4<70)
{
  digitalWrite(Fleft,LOW);
  digitalWrite(FLD,LOW);
  digitalWrite(Fright,LOW);
  digitalWrite(FRD,LOW);
  digitalWrite(Bleft,LOW);
  digitalWrite(BLD,LOW);
  digitalWrite(Bright,LOW);
  digitalWrite(BRD,LOW);
  Serial.println("@@@@@@@@@@@@@@@@@@@@Check stop@@@@@@@@@@@@@@@@@@@@");
}
//HIGH LOW 정방향 LOW HIGH 역방향
else if(CH1>70 && CH2>40 && CH2<70 && CH4<70 && CH4>40)
,

```

모터를 제어하는 것은 생각보다는 어렵지 않다. 가장 간편하게 if문을 사용하여 모터를 조작할 수 있다.

아래코드는 if문을 이용하여 모터를 조작한 것이다. if문의 조건절을 보면 조건이 다소 복잡하다. 이는 핀의 신호를 반영하기 위한 것으 쓰레드(Thread)를 이용하면 간단히 만들 수 있지만 아두이노는 멀티태스킹이 지원

되지 않기 때문에 쓰레드를 사용할 수 없다.

Mode 2는 프로그램에 의해 움직이는 모드이다. 먼저 복잡한 연산이 가능한 라즈베리파이가 영상처리를 한 다음 영상처리 결과에 따라서 2개의 아두이노 디지털핀에 신호를 보내준다. 그 신호에 따라서 아두이노가 로봇의 동작을 제어한다.

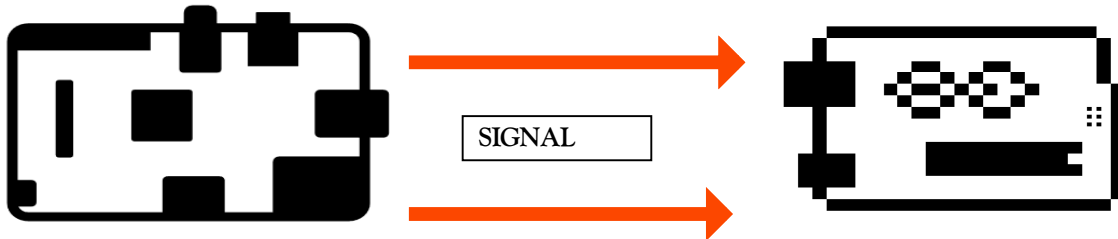


그림 138 화염 로봇의 제어 구성

```

//////////////////////////////////// MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2////////////////////////////////////
//////////////////////////////////// MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2////////////////////////////////////
//////////////////////////////////// MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2 MODE2////////////////////////////////////
while(change_status==2){
  cam_pin_1=digitalRead(ras_pin_1);//라즈베리파이 신호를 받는다
  cam_pin_2=digitalRead(ras_pin_1);//라즈베리파이 신호를 받는다

  digitalWrite(led1,LOW);
  digitalWrite(led2,HIGH);
  if(cam_pin_1==HIGH&&cam_pin_2==HIGH)//CENTER////////////////////////////////신호에 따라서 움직임////////////////////////////////
  {
    digitalWrite(Fleft,LOW);
    digitalWrite(FLD,LOW);
    digitalWrite(Fright,LOW);
    digitalWrite(FRD,LOW);
    digitalWrite(Bleft,LOW);
    digitalWrite(BLD,LOW);
    digitalWrite(Bright,LOW);
    digitalWrite(BRD,LOW);
    Serial.println("@@@@@@@@@@@@@@@@@@Check stop@@@@@@@@@@@@@@@@@@@@");
    for(speedvalue=800;speedvalue<=2000;speedvalue=speedvalue+100)
    {
      shoot1.writeMicroseconds(speedvalue);
      shunt?.writeMicroseconds(speedvalue);
    }
  }
}

```

03 영상인식을 통한 자율 주행 로봇

학습목표

- 기본적인 영상인식을 통한 로봇제어 프로그램을 작성할 수 있다.
- 기본적인 영상인식을 통한 로봇제어 프로그램을 작성할 수 있다.

여기서는 (주)이산솔루션의 교육용로봇 알티노(ALTINO)를 이용하여 자율 주행 로봇을 제작한다. 이 자율주행 모바일 로봇은 2016 로보월드 대회에 참여한 로봇을 기반으로 하였다.

1. 알티노 하드웨어 구성

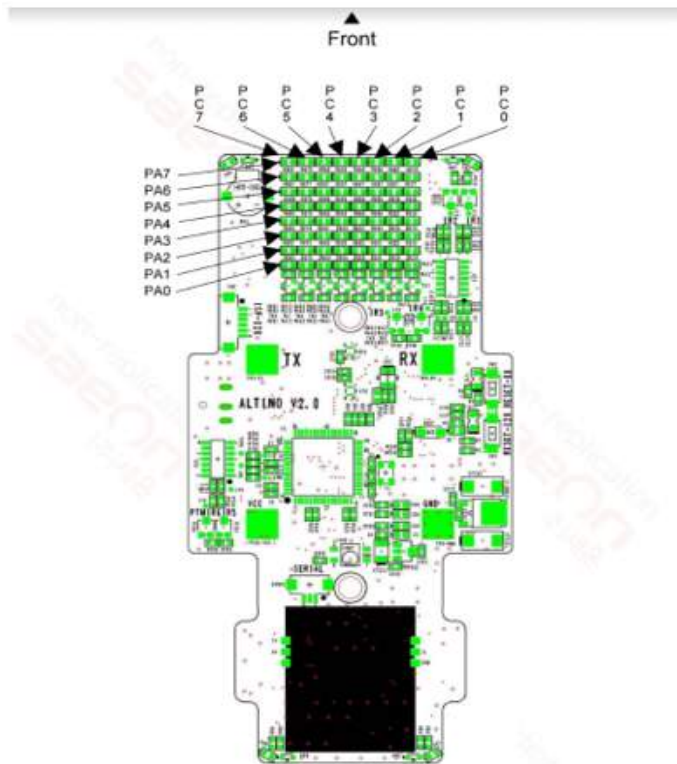


그림 139 알티노 내부 구성

알티노는 4바퀴 모바일 로봇으로 바닥을 인지하는 빛 센서 전방 측면을 인식하는 적외선 센서 등을 가지고 있으며 상부에 라즈베리파이와 카메라를 연결할 수 있는 구조를 가지고 있어 인식한 영상을 이용하여 자율 주행할

수 있도록 하였다. 그리고 전방 조향은 차륜 구동이 가능하도록 하여 실제 자동차와 같은 움직임을 구현할 수 있다.

전술한 바와 같이 여기서 구현한 로봇은 2016년 R-Biz 챌린지에 출전한 로봇을 기반으로 한다. 이 대회에서 미션은 4가지가 있다.

2. 알티노 움직임 구현

첫 번째로 그림 134를 보면 <경기장 도면>의 빨간색 원이 보일 것이다. 이 원이 있는 구역의 목적은 사진의 오른쪽 <미션규격>에서 빨간색 원안에 있는 화살표가 보이는 데 화살표는 왼쪽 오른쪽을 랜덤으로 가리키게 되어 있으며 왼쪽 오른쪽을 판단하고 그 방향에 맞게 움직이면 된다. 두 번째로 그림 134를 보면 초록색 원이 보인다. 초록색 원은 첫 번째 미션인 화살표의 방향을 인식한 후 초록색 원에 다가 알티노를 주차시킨다. 세 번째로 그림 134를 보면 파란색 원을 볼 수 있다. 파란색 원은 라즈베리 카메라로 영상을 받아온 후 횡단보도를 판단하여 5cm 이내에 멈추는 것이 목적이다. 마지막으로 그림 134를 보면 보라색 직사각형이 보이는데 직사각형 안에서는 자신이 구상한 자율 미션을 수행하면 된다.

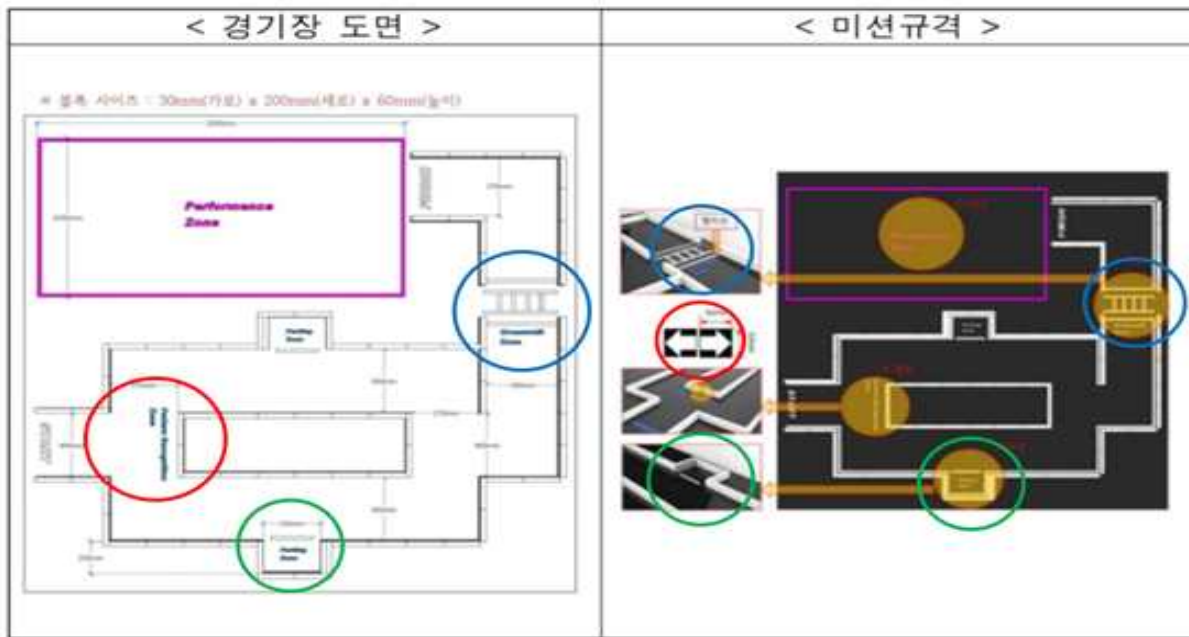


그림 140 R-Biz 챌린지 대회 미션도

첫 번째 미션을 수행하기 위한 과정은 다음과 같다. 아래 그림 135에서 빨간색 선이 보이는데 작성자는 빨간색을 기준으로 주차를 시도했다.

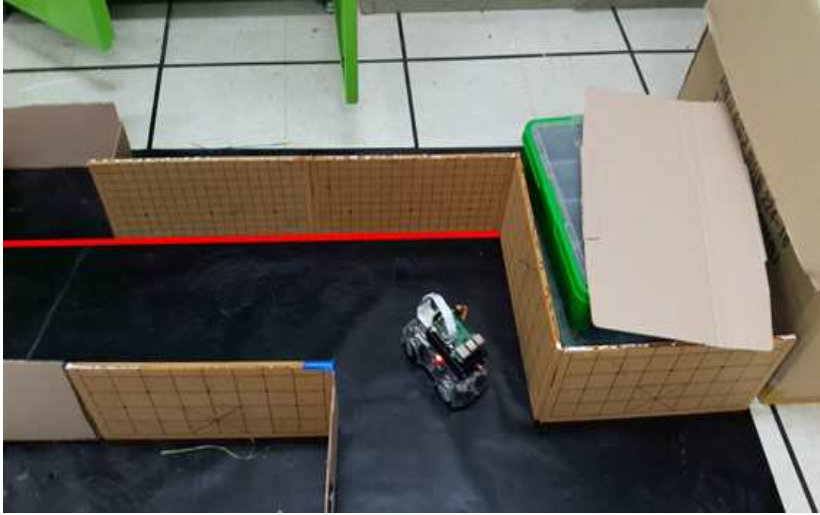


그림 141 알티노의 회전

우선 빨간색 선쪽으로 알티노를 움직인 후 사진처럼 바퀴를 우측으로 방향을 틀은 후 아주 잠깐 후진을 시도했다. 그리고 난 후 다시 바퀴를 정면으로 바꾼 후 왼쪽 바퀴가 오른쪽 바퀴보다 조금 더 빠르게 설정해준다.



그림 142 알티노의 주차 움직임 1

그러면 빨간색 선에 평행해 질 수 있지만 완전 평행이 아닌 알티노가 주차 공간에 들어가게 된다.

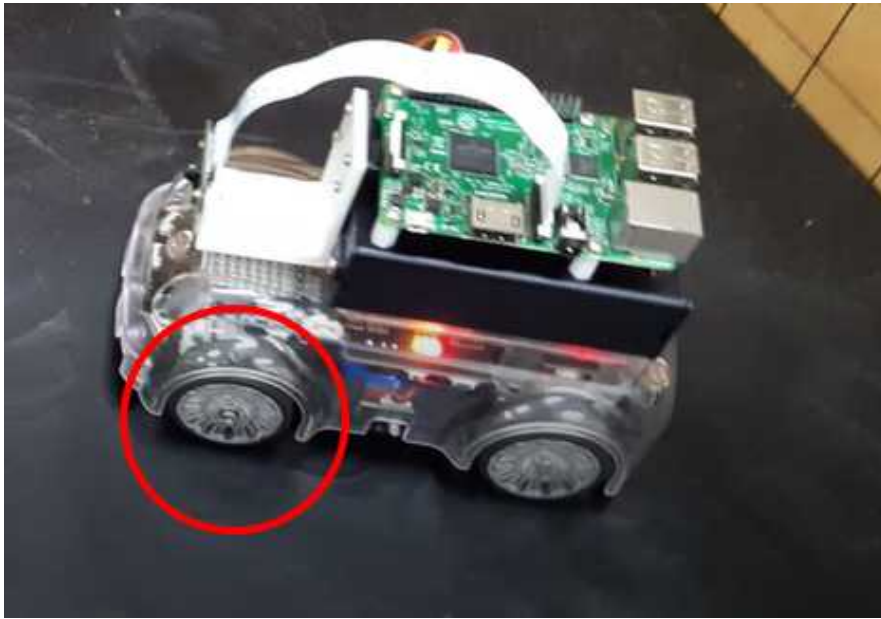


그림 143 알티노의 주차 움직임 2

그 다음에는 알티노를 벽에 단지 않게 쪽 직진하면 평행 주차는 아니지만 주차 공간에 주차를 하게 된다.

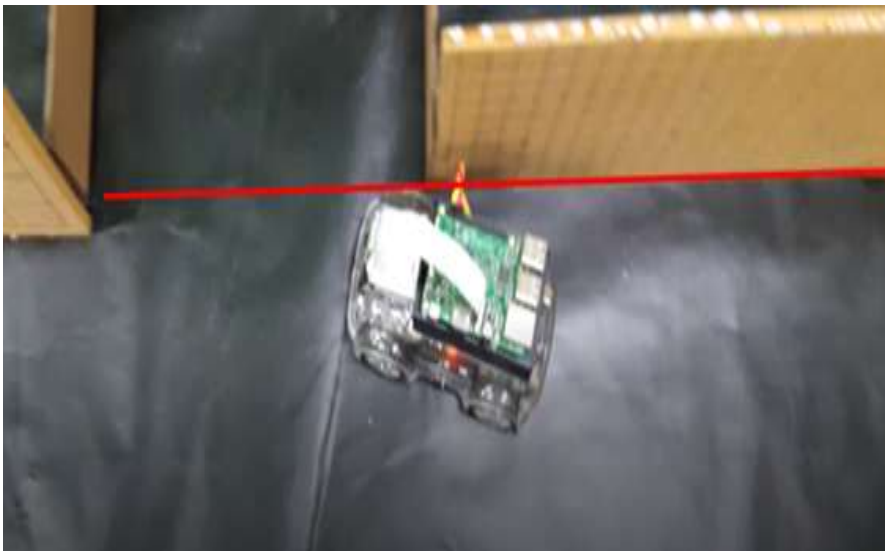


그림 144 알티노 주차를 위한 움직임

2.1 주차

2.1.1 주차를 위한 소스 코드

```
while(Robot.IRSensor[1]>10)
{
    Steering(3); //바퀴의 방향을 조절하는 함수
    Go(500,500); //앞으로 가는 함수파라미터는 왼쪽,오른쪽 바퀴
이다.
    Robot(0); //Robot형태의 구조체 센서 값을 받아옴
}
Go(0,0);

while(Robot.IRSensor[3]<5)// 후진후 직진을 IRSensor[3]이 인식될 때
까지 계속 반복해 준다.
{
    Steering(3);
    Go(-300,-300);
    delay(100);
    Steering(2);
    Go(300,300);
    Robot(0);
}
Steering(2);
```

```

while(Robot.IRSensor[0]>5 && Robot.IRSensor[1]>5 &&
Robot.IRSensor[2]>5) // 주차 공간 틈으로 들어간후 벽이 나올 때
까지 직진 해준다.

{
    Go(100,100);

    Robot(0);
}

// 주차 하기 싫게 후진을 해준다.

Go(0,0);

Steering(1);

Go(-100,-100);

delay(200);

Go(0,0);

Steering(3);

while(Robot.IRSensor[1]>5) //주차공간 안쪽까지 직진
{
    Go(100,100);

    Robot(0);
}

Go(0,0);

```

표 17 주차 미션을 위한 소스

2.1.2 함수 기반 소스 설명

-Open():

알티노와 컴퓨터를 연결한다.

-Go(int left,int right):

앞뒤로 이동한다.

파라미터:

left: 왼쪽 모터 속도(0~1000)

right: 오른쪽 모터 속도(0~1000)

-Steering(int steeringvalue):

앞쪽 바퀴의 각도를 설정한다.

파라미터:

steeringValue: 앞쪽바퀴의 각도(0~180)

-Steering2(int value1, int value2):

앞쪽 바퀴의 각도를 각각 설정한다.

파라미터:

value1: 왼쪽 앞쪽바퀴의 각도(0~180)

value2: 오른쪽 앞쪽바퀴의 각도(0~180)

-chekData():

알티노에서 들어오는 데이터를 확인한다.

-Sensor(int command):

여러 센서에서 들어오는 값을 받을 수 있다.

파라미터:

command: 센서번호

2.1.3 알티노에 사용된 opencv 함수

- cv2DRotationMatrix()
- cvCvtColor()
- cvRectangle()
- cvHoughLines2()
- cvInRange()
- cvMatchShapes()
- cvRead()
- cvWaitKey()
- cvContourArea()

2.1.4 함수 사용방법

-cv2DRotationMatrix(①,②,③,④):

영상을 회전시키거나 크기변환을 시킨다.

파라미터:

- ①: center: 회전하려고 하는 회전의 중심(좌표)
- ②: Angle: 회전각도(양수: 반시계방향 음수: 시계방향)
- ③: Scale: 확대 또는 축소할 값
- ④: mapMatrix: 2*3 matrix 포인터. 회전 연산의 결과가 저장됨. 이 map Matrix는 다음과 같은 꼴을 지닌다.

-cvCvtColor(①,②,③): 이미지의 색을 변환시킨다.

파라미터:

- ①: src: 원본 배열. 8-bit(8u), 16-bit(16u), single-precision floating point(32f)형태의 이미지.
- ②: dst: 변환된 배열을 저장할 배열.
- ③: code: 색 변환 작업에 관한 파라미터. 그 내용은 다음과 같다.

Conversion code	Meaning
CV_BGR2RGB CV_RGB2BGR CV_RGBA2BGRA CV_BGRA2RGBA	Convert between RGB and BGR color spaces (with or without alpha channel)
CV_RGB2RGBA CV_BGR2BGRA	Add alpha channel to RGB or BGR image
CV_RGBA2RGB CV_BGRA2BGR	Remove alpha channel from RGB or BGR image
CV_RGB2BGRA CV_RGBA2BGR CV_BGRA2RGB CV_BGR2RGBA	Convert RGB to BGR color spaces while adding or removing alpha channel
CV_RGB2GRAY CV_BGR2GRAY	Convert RGB or BGR color spaces to grayscale
CV_GRAY2RGB CV_GRAY2BGR CV_RGBA2GRAY CV_BGRA2GRAY	Convert grayscale to RGB or BGR color spaces (optionally removing alpha channel in the process)
CV_GRAY2RGBA CV_GRAY2BGRA	Convert grayscale to RGB or BGR color spaces and add alpha channel
CV_RGB2BGR565 CV_BGR2BGR565 CV_BGR5652RGB CV_BGR5652BGR CV_RGBA2BGR565 CV_BGRA2BGR565 CV_BGR5652RGBA CV_BGR5652BGRA	Convert from RGB or BGR color space to BGR565 color representation with optional addition or removal of alpha channel (16-bit images)
CV_GRAY2BGR565 CV_BGR5652GRAY	Convert grayscale to BGR565 color representation or vice versa (16-bit images)

-cvRectangle(①,②,③,④,int thickness = 1, int lineType = 8, int shift = 0)

: 채워진 또는 빈 사각형을 그린다.

파라미터:

- ①: img: 원본 이미지
- ②: pt1: 사각형의 꼭짓점 중 한 점(좌표)
- ③: pt2: 반대 꼭짓점
- ④: color: 사각형의 선의 색상

-thickness: 사각형을 구성하는 선의 두께. 만약 이 값이 음수이거나 CV_FILLED가 사용되면 사각형의 내부가 채워진다.

-lineType: 선의 종류

-shift

-cvHoughLines2(①,②,③,④,⑤,⑥,double param1 = 0, double param2 = 0)

: 이진화된 이미지에서 허브변환을 통해 선을 검출한다.

파라미터:

- ①: image: 허프 변환(Hough Transform)을 하고자 하는 원본 이미지.
- ②: storage: 검출된 선(Line)을 저장할 Storage.
- ③: method: 허프 변환의 방법을 의미.

- CV_HOUGH_STANDARD: 기본 허프 변환(Hough Transform).
- CV_HOUGH_PROBABILISTIC: 확률적 허프 변환. 전체 선이 아닌 선의 일부만 반환한다. 선의 일부는 시작점과 끝 점의 정보로 나타난다.
- CV_HOUGH_MULTI_SCALE: 기본 허프 변환에서 멀티 스케일 변형을 한 것.

④: rho: 픽셀 단위로 나타나는 거리 값.

⑤: theta: 라디안 단위로 나타나는 각도 값.

⑥: threshold: 임계값. 해당 축적 값이 이 값보다 크면 Line이 출력된다.

param1: 이 파라미터는 method에 따라 다르다.

- 기본 허프 변환에서는 이 값이 사용되지 않는다(=0).
- Probabilistic 허프 변환에서는 선 길이의 최소 값을 나타낸다.
- Multi Scale 허프 변환에서는 거리 값에 대한 나누는 수를 의미한다.

param2 : 이 파라미터도 method에 따라 다르다.

- 기본 허프 변환에서는 이 값이 사용되지 않는다(=0).
- Probabilistic 허프 변환에서는 선들 사이의 최대 gap을 의미
- Multi Scale 허프 변환에서는 각도에 대한 나누는 수를 의미한다.

-cvInRange(①,②,③,④):

원본 배열의 원소들이 두 개의 다른 배열들 사이에 포함되는지 확인한다.

파라미터:

①: src: 원본 배열.

②: lower: 가장 낮은 수준의 배열 또는 스칼라 값.

③: upper: 가장 높은 수준의 배열 또는 스칼라 값.

④: dst: 비교 결과가 저장되는 배열

-cvMatchShapes(①,②,③,④):

두 모양을 비교한다.

파라미터:

①: object1: 첫 번째 비교 대상. 경계선 또는 흑백의 이미지.

②: object2: 두 번째 비교 대상. 경계선 또는 흑백의 이미지.

- ③: method: 비교의 방법. 그 종류는 다음과 같다.
- ④: parameter: 사용되지 않는 파라미터임.

$$\text{CV_CONTOUR_MATCH_I1: } I_1(A, B) = \sum_{i=1, \dots, 7} \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right|$$

$$\text{CV_CONTOURS_MATCH_I2: } I_2(A, B) = \sum_{i=1, \dots, 7} |m_i^A - m_i^B|$$

$$\text{CV_CONTOURS_MATCH_I3: } I_3(A, B) = \sum_{i=1, \dots, 7} \frac{|m_i^A - m_i^B|}{|m_i^A|}$$

- cvRead(①,②,③):

개체를 해석하여 그 포인터 값을 반환한다.

파라미터:

- ①: fs: 파일 저장체(File Storage).
- ②: node: 개체의 Node.
- ③: Attributes: 사용되지 않는 파라미터.

- cvWaitKey(①):

키보드의 키(Key) 입력 신호를 기다린다.

파라미터:

- ①: delay: 키(Key)의 입력에 대해 기다리는 딜레이. 즉, 입력된 시간만큼 키 입력이 이루어 져야 키 입력을 받아들이는 것.
단위는 millisecond.

- cvContourArea(①,②,③):

관심 있는 부분 또는 전체 경계선이 포함하는 영역의 넓이를 계산한다.

관심 있는 부분의 넓이를 구하는 기준은 다음 그림과 같다.

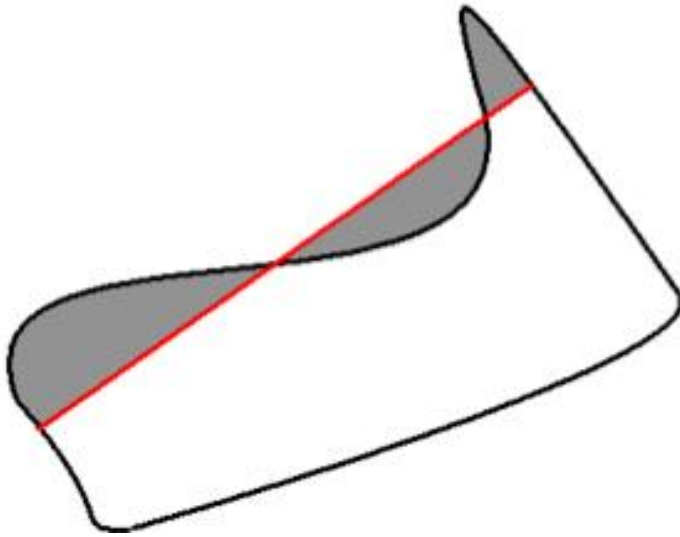


그림 145 경계선 영역 계산

파라미터:

- ①: contour: Sequence나 배열 형태의 경계선
- ②: slice: 관심 영역의 시작점과 끝 점. 이 값이 없으면 전체 경계선의 영역이 계산된다.

- cvLine(①,②,③,④, int thickness = 1, int lineType = 8, int shift = 0):
이미지에 선을 그린다.

파라미터:

- ①: img: 선을 그릴 원본 이미지.
- ②: pt1: 선의 양 끝 점 중 한쪽 점.
- ③: pt2: 선의 양 끝 점 중 나머지 한쪽 점.
- ④: color: 선의 색깔.

thickness: 선의 두께

lineType: 선의 종류

-8: 8-Connected Line

-4: 4-Connected Line

-CV_AA: Anti-Aliasing Line

Shift: Fractional Bits의 갯수.

2.2 화살표 감지

2.2.1 화살표 감지

주차와 연관된 미션 중에서 화살표를 인식해야 하는 과제가 있다. 다음은 화살표 인식을 위한 소스 코드이다.

```
#include "opencv2/objdetect.hpp"
#include "opencv2/highgui.hpp"
#include "opencv2/imgproc.hpp"
#include <iostream>
#include "Altino.h"

void main(){
    while(1){
        Mat trans;
        Mat frame,img;
        int left=0;
        int right=0;
        img = imread( inputName, 1 );
        Mat right=imread(/right.jpg,1);
        Mat left=imread(/left.jpg,1);
        Mat matrotation = getRotationMatrix2D( Point(160, 120),
180, 1);
        warpAffine(img, img, matrotation, img.size());
        for(int i=0;i<10;i++){
            cvInRange(img,Scalar(0,0,0),Scalar(61,61,61),trans);
```

```

float
right_acc=cvMatchShapes(right,img,CV_CONTOUR_MATCH_I2,0);

float
left_acc=cvMatchShapes(left,img,CV_CONTOUR_MATCH_I2,0);

                (right+=right_acc)/2;
                (left+=left_acc)/2;

}

if(right>left){
    cout <<"maybe this shape is right arrow";
        Steering(1);
}

else if(right<left){
    cout <<"maybe this shape is left arrow";
        Steering(3);
}

else if(right==left){
    cout <<"can't detect";
        Steering(2);
}

else{
    cout <<"can't detect";
        Steering(2);
}

}
}
}

```

표 18 화살표 검출을 위한 소스 코드

2.3 선검출

선검출은 횡단보도 주차를 위한 필수 요소이다. 여기서는 openCV를 사용하여 선 검출을 시행하였다. 그 소스는 다음과 같다.

2.3.1 선검출 소스

```
#include "opencv2/objdetect.hpp"
#include "opencv2/highgui.hpp"
#include "opencv2/imgproc.hpp"
#include <iostream>
#include "Altino.h"
void main(){
    while(1){
        Mat trans;
        Mat frame,img;
        Mat gray;
        img = imread( inputName, 1 );
        Mat matrotation = getRotationMatrix2D( Point(160,
120), 180, 1);
        warpAffine(img, img, matrotation, img.size());
        cvCvtColor(img,gray,CV_BGR2GRAY);
        for(int i=0;i<10;i++){
            cvHoughLines2(gray,line,CV_HOUGH_PROBABILISTIC,5,10,2,double
param1 = 0, double param2 = 0);
            Scalar xy=cvContourArea(line,CvSlice slice =
```

```
CV_WHOLE_SEQ);  
  
        cvLine(img,xy[0],xy[1],Scalar(255,255,255), int  
thickness = 1, int lineType = 8, int shift = 0);  
  
        }  
  
        cvShowImage("hello",img);  
  
        }  
  
}
```

표 19 선검출 소스

3. 퍼포먼스

퍼포먼스는 어려운 것을 하지 않고 도착하면 색을 인지하여 정지하는 정도로 수행한다.

```
void main(){
    while(1){
        Mat trans;
        Mat frame,img;
        Mat gray;
        img = imread( inputName, 1 );
        Mat matrotation = getRotationMatrix2D( Point(160,
120), 180, 1);
        warpAffine(img, img, matrotation, img.size());
        cvCvtColor(img,gray,CV_BGR2HSV);
        cvInRange(img,(0,100,100),(0,255,255),trans);
        x,y=cvBoundingRect(trans, int update = 0);
        cvRectangle(img,x,y,(0,255,255),int thickness = 1, int
lineType = 8, int shift = 0);
        cvShowImage("hello",img);
    }
}
```

표 20 마지막 동작 소스

부록

여기에 수록한 소스코드들은 메인 코드들만 수록하였으며 각종 헤더 파일과 부수 파일들은 지면 관계상 수록하지 않았다.

<부록> 소스코드

I. 소방로봇

1.1 메인 코드

```
#include "rc_rx.h"
```

```
#include <Servo.h>
```

```
int Fleft=5;
```

```
int Fright=7;
```

```
int Bleft=9;
```

```
int Bright=11;
```

```
int FLD=6;
```

```
int FRD=8;
```

```
int BLD=10;
```

```
int BRD=12;
```

```
int stepmotor=22,dir=23,On=24;///서보 ⑦'° ?스
```

```
int speedvalue=1000,s;
```

```
int bldcmotor1=30,bldcmotor2=32;///BLDC motor setting
```

```
Servo shoot1;
```

```
Servo shoot2;
```

```
int stat;
```

```
int ras1,ras2;
```

```
#define led 13
```

```
#define dt 20 // [ms] Task time
```

```
int Up;
```

```
int Down;
```

```
int left;
```

```
int right;
```

```

int Pright;//
int Pleft;
int i;
int k=0;
unsigned long t=0;
void setup()
{
Serial.begin(115200); // Initlizing serial port before calling init_rc_rx() Function
init_rc_rx();
pinMode(led, OUTPUT);
for(i=5;i<13;i++)
pinMode(i,OUTPUT);

shoot1.attach(40);
shoot1.writeMicroseconds(200);
shoot2.attach(41);
shoot2.writeMicroseconds(200);
pinMode(22,OUTPUT);
pinMode(23,OUTPUT);
pinMode(24,OUTPUT);
}

void loop()
{
while(stat==0)
////////////////////////////////////MODE1////////////////////////////////////
{
t = micros()/1000;

read_rc_rx();      // CH1, CH2, CH3 and CH4 float variables will be calculated with
this Function is called

analogWrite(led,2.55*CH1);

Up=map(CH1,48,98,0,240);
Down=map(CH1,48,0,0,240);
left=map(CH2,48,98,0,240);
right=map(CH2,48,0,0,240);
}
}

```



```

digitalWrite(dir,HIGH);
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
delay(3500);
digitalWrite(On,LOW);
}
}
k++;
}
speedvalue=800;
shoot1.writeMicroseconds(speedvalue);
shoot2.writeMicroseconds(speedvalue);
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
Serial.println("out out out out out out out out out out out out");
}
else if(CH3<80)
{
speedvalue=800;
shoot1.writeMicroseconds(speedvalue);

```

```

shoot2.writeMicroseconds(speedvalue);
digitalWrite(On,LOW);
k=0;
}

if(CH2>40 && CH2<70 && CH1>40 && CH1<70 && CH4>40 && CH4<70)
{
STOP();
Serial.println("@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@Check
stop@@@@@@@@@@@@@@@@@@@@@@@@@@@@");
}
//HIGH LOW ????LOW HIGH ??? else if(CH1>70 && CH2>40 && CH2<70
&& CH4<70 && CH4>40)
{
Goforward();
Serial.println("*****forwad*****");
}
else if(CH1<40 && CH2>40 && CH2<70 && CH4<70 && CH4>40)
{
Gobackward();
Serial.println("*****BACKWARD*****");
}
else if(CH4<40 && CH2>40 && CH2<70 && CH1>40 && CH1<70)
{
TTR();
Serial.println("*****To the right*****");
}
else if(CH4>70 && CH2>40 && CH2<70 && CH1<70 && CH1>40)
{
TTL();
Serial.println("*****TO the left*****");
}
else if(CH2<40 && CH1>40 && CH1<70 && CH4<70 && CH4>40)
{
turnright();
Serial.println("*****turn right*****");
}
else if(CH2>70 && CH1>40 && CH1<70 && CH4<70 && CH4>40)

```

```

{
turnleft();
Serial.println("*****turn left*****");
}

Serial.print(CH1);//梨'ë' 媛? Serial.print("\t");
Serial.print(CH2);
Serial.print("\t");
Serial.print(CH3);
Serial.print("\t");
Serial.print(CH4);
Serial.print("\t");
Serial.print(left);
Serial.print("\t");
Serial.print(right);
Serial.print("\t");
Serial.println(((micros()/1000)- (float)t)*100/dt);

```

```

while (dt > (micros()/1000)- t){
// do nothing
    }
}

```

```

while(stat==1)
////////////////////MODE2////////////////////////////////////
{
ras1=digitalRead(3);
ras2=digitalRead(4);

if(ras1==1 && ras2==0)
{
turnleft();
}
else if(ras1==0 && ras2==1)
{
turnright();
}
}

```

```
else if(ras1==1 && ras2==1)
{
shoot();
}
}

}
```

```
void turnleft()
{
digitalWrite(Fleft,HIGH);
digitalWrite(FLD,LOW);
digitalWrite(Fright,LOW);
digitalWrite(FRD,HIGH);
digitalWrite(Bleft,HIGH);
digitalWrite(BLD,LOW);
digitalWrite(Bright,LOW);
digitalWrite(BRD,HIGH);
}
```

```
void turnright()
{
digitalWrite(Fleft,LOW);
digitalWrite(FLD,HIGH);
digitalWrite(Fright,HIGH);
digitalWrite(FRD,LOW);
digitalWrite(Bleft,HIGH);
digitalWrite(BLD,LOW);
digitalWrite(Bright,LOW);
digitalWrite(BRD,HIGH);
}
```

```
void TTL()
{
digitalWrite(Fleft,LOW);
digitalWrite(FLD,HIGH);
digitalWrite(Fright,HIGH);
```

```
digitalWrite(FRD,LOW);  
digitalWrite(Bleft,LOW);  
digitalWrite(BLD,HIGH);  
digitalWrite(Bright,HIGH);  
digitalWrite(BRD,LOW);  
}
```

```
void TTR()  
{  
digitalWrite(Fleft,HIGH);  
digitalWrite(FLD,LOW);  
digitalWrite(Fright,LOW);  
digitalWrite(FRD,HIGH);  
digitalWrite(Bleft,LOW);  
digitalWrite(BLD,HIGH);  
digitalWrite(Bright,HIGH);  
digitalWrite(BRD,LOW);  
}
```

```
void Gobackward()  
{  
digitalWrite(Fleft,LOW);  
digitalWrite(FLD,HIGH);  
digitalWrite(Fright,LOW);  
digitalWrite(FRD,HIGH);  
digitalWrite(Bleft,LOW);  
digitalWrite(BLD,HIGH);  
digitalWrite(Bright,LOW);  
digitalWrite(BRD,HIGH);  
}
```

```
void Goforward()  
{  
digitalWrite(Fleft,HIGH);  
digitalWrite(FLD,LOW);  
digitalWrite(Fright,HIGH);  
digitalWrite(FRD,LOW);  
digitalWrite(Bleft,HIGH);  
}
```

```

digitalWrite(BLD,LOW);
digitalWrite(Bright,HIGH);
digitalWrite(BRD,LOW);
}

void STOP()
{
digitalWrite(Fleft,LOW);
digitalWrite(FLD,LOW);
digitalWrite(Fright,LOW);
digitalWrite(FRD,LOW);
digitalWrite(Bleft,LOW);
digitalWrite(BLD,LOW);
digitalWrite(Bright,LOW);
digitalWrite(BRD,LOW);
}

void shoot()
{
while(k<1)
{
for(speedvalue=800;speedvalue<=2000;speedvalue=speedvalue+100)
{
shoot1.writeMicroseconds(speedvalue);
shoot2.writeMicroseconds(speedvalue);
delay(100);
Serial.println("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^CHAEGING
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^");
if(speedvalue==2000)
{
digitalWrite(On,HIGH);
digitalWrite(dir,LOW);
Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
}
}
}
}

```

```

Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
Serial.println("PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH
PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH");
delay(3500);
digitalWrite(On,HIGH);
digitalWrite(dir,HIGH);
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
Serial.println("*****PULL PULL PULL PULL PULL PULL PULL PULL PULL
PULL PULL *****");
delay(3500);
digitalWrite(On,LOW);
}
}
k++;
}
speedvalue=800;
shoot1.writeMicroseconds(speedvalue);
shoot2.writeMicroseconds(speedvalue);

```



```

// Interrupt ISRs

void CH1_int_ISR()
{
if ((micros()-CH1_t) < valid_pulse_limit){
CH1_delta = micros()-CH1_t;
}
CH1_t = micros();
}

void CH2_int_ISR()
{
if ((micros()-CH2_t) < valid_pulse_limit){
CH2_delta = micros()-CH2_t;
}
CH2_t = micros();
}

void CH3_int_ISR()
{
if ((micros()-CH3_t) < valid_pulse_limit){
CH3_delta = micros()-CH3_t;
}
CH3_t = micros();
}

void CH4_int_ISR()
{
if ((micros()-CH4_t) < valid_pulse_limit){
CH4_delta = micros()-CH4_t;
}
CH4_t = micros();
}

// Call able Function

```

```

void init_rc_rx(){

Serial.print("Channel 1 connected to pin number....Wt");
Serial.println(CH1_pin);
Serial.print("Channel 2 connected to pin number....Wt");
Serial.println(CH2_pin);
Serial.print("Channel 3 connected to pin number....Wt");
Serial.println(CH3_pin);
Serial.print("Channel 4 connected to pin number....Wt");
Serial.println(CH4_pin);

pinMode(CH1_pin, INPUT);
pinMode(CH2_pin, INPUT);
pinMode(CH3_pin, INPUT);
pinMode(CH4_pin, INPUT);

attachInterrupt(CH1_int, CH1_int_ISR, CHANGE);
attachInterrupt(CH2_int, CH2_int_ISR, CHANGE);
attachInterrupt(CH3_int, CH3_int_ISR, CHANGE);
attachInterrupt(CH4_int, CH4_int_ISR, CHANGE);
}

void read_rc_rx(){
CH1 = ((float)CH1_delta-(float)min_high_time)*100/(max_high_time-min_high_time);
CH2 = ((float)CH2_delta-(float)min_high_time)*100/(max_high_time-min_high_time);
CH3 = ((float)CH3_delta-(float)min_high_time)*100/(max_high_time-min_high_time);
CH4 = ((float)CH4_delta-(float)min_high_time)*100/(max_high_time-min_high_time);
}

```

II. 손가락 제어 로봇

1. 메인 소스

```
import cv2
import numpy as np
import math
import serial
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup(17,GPIO.OUT)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(27,GPIO.OUT)
GPIO.setup(22,GPIO.OUT)
GPIO.setup(23,GPIO.OUT)

countin=0
cap = cv2.VideoCapture(0)
sum1=0.1
las=0

while(cap.isOpened()):
    sum1=0.1
    for i in range(1,5):
        ret, img = cap.read()
        cv2.rectangle(img,(350,350),(100,100),(0,255,0),0)
        crop_img = img[100:350, 100:350]
        grey = cv2.cvtColor(crop_img, cv2.COLOR_BGR2GRAY)
        value = (35, 35)
        blurred = cv2.GaussianBlur(grey, value, 0)
        _, thresh1 = cv2.threshold(blurred, 127, 255,
        cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
        #cv2.imshow('Thresholded', thresh1)

    (version, _, _) = cv2.__version__.split('.')

    if version is '3':
```

```

image, contours, hierarchy = cv2.findContours(thresh1.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
elif version is '2':
contours, hierarchy = cv2.findContours(thresh1.copy(),cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

cnt = max(contours, key = lambda x: cv2.contourArea(x))

x,y,w,h = cv2.boundingRect(cnt)
cv2.rectangle(crop_img,(x,y),(x+w,y+h),(0,0,255),0)
hull = cv2.convexHull(cnt)
drawing = np.zeros(crop_img.shape,np.uint8)
cv2.drawContours(drawing,[cnt],0,(0,255,0),0)
cv2.drawContours(drawing,[hull],0,(0,0,255),0)
hull = cv2.convexHull(cnt,returnPoints = False)
defects = cv2.convexityDefects(cnt,hull)
count_defects = 0
cv2.drawContours(thresh1, contours, -1, (0,255,0), 3)
for i in range(defects.shape[0]):
s,e,f,d = defects[i,0]
start = tuple(cnt[s][0])
end = tuple(cnt[e][0])
far = tuple(cnt[f][0])
a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2)
b = math.sqrt((far[0] - start[0])**2 + (far[1] - start[1])**2)
c = math.sqrt((end[0] - far[0])**2 + (end[1] - far[1])**2)
angle = math.acos((b**2 + c**2 - a**2)/(2*b*c)) * 57
if angle <= 90:
count_defects += 1
cv2.circle(crop_img,far,1,[0,0,255],-1)
#dist = cv2.pointPolygonTest(cnt,far,True)
cv2.line(crop_img,start,end,[0,255,0],2)
#cv2.circle(crop_img,far,5,[0,0,255],-1)

fir=count_defects
if i>1:
sum1=(fir+las)/2

```

```

if count_defects == 2:
cv2.putText(img,"2", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
elif count_defects == 3:
cv2.putText(img, "3", (5,50), cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
elif count_defects == 4:
cv2.putText(img,"4", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
elif count_defects == 5:
cv2.putText(img,"5", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
else:
cv2.putText(img,"none", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
las=sum1

```

```

#cv2.imshow('drawing', drawing)
#cv2.imshow('end', crop_img)
cv2.imshow('Gesture', img)
all_img = np.hstack((drawing, crop_img))
#cv2.imshow('Contours', all_img)
print(sum1)

```

```

if sum1>=1.5 and sum1<2.5:
GPIO.output(17,True)
GPIO.output(18,False)
GPIO.output(27,False)
GPIO.output(22,False)
GPIO.output(23,False)

```

```

elif sum1>=2.5 and sum1<3.5:
GPIO.output(17,False)
GPIO.output(18,True)
GPIO.output(27,False)
GPIO.output(22,False)
GPIO.output(23,False)

```

```

elif sum1>=3.5 and sum1<4.5:
GPIO.output(17,False)
GPIO.output(18,False)
GPIO.output(27,True)
GPIO.output(22,False)

```

```

GPIO.output(23,False)

elif sum1>=4.5:
    GPIO.output(17,False)
GPIO.output(18,False)
GPIO.output(27,False)
GPIO.output(22,True)
GPIO.output(23,False)
else:
GPIO.output(17,False)
GPIO.output(18,False)
GPIO.output(27,False)
GPIO.output(22,False)
GPIO.output(23,True)

k= cv2.waitKey(5) &0xFF
if k==27:
    GPIO.cleanup()
    break

```

2. 아두이노 부분 소스코드

```

#include <SoftwareSerial.h>
#define m1G 4
#define m1P 6
#define m2G 7
#define m2P 5
SoftwareSerial bluetooth(2, 3);
char a;
void Go();
void Left();
void Right();
void Back();
void Stop();

void setup()

```

```

{
  pinMode(m1G,1);
  pinMode(m1P,1);
  pinMode(m2G,1);
  pinMode(m2P,1);

  Serial.begin(9600);

  bluetooth.begin(9600);

}
void loop()
{
  if (bluetooth.available()) {
    Serial.println(" ");
    a=bluetooth.read();
  }
  if (Serial.available()) {
    bluetooth.write(Serial.read());
  }
  Serial.print("xx");
  Serial.print(a);

  if(a=='x')
  Stop();
  else if(a=='a')
  Go();
  else if(a=='b')
  Left();
  else if(a=='c')
  Right();
  else if(a=='d')
  Back();
  else{}
}

void Go()

```

```
{
  digitalWrite(m1G,1);
  analogWrite(m1P,125);
  digitalWrite(m2G,1);
  analogWrite(m2P,125);
}
```

```
void Left()
```

```
{
  digitalWrite(m1G,1);
  analogWrite(m1P,125);
  digitalWrite(m2G,0);
  analogWrite(m2P,125);
}
```

```
void Right()
```

```
{
  digitalWrite(m1G,0);
  analogWrite(m1P,125);
  digitalWrite(m2G,1);
  analogWrite(m2P,125);
}
```

```
void Back()
```

```
{
  digitalWrite(m1G,0);
  analogWrite(m1P,125);
  digitalWrite(m2G,0);
  analogWrite(m2P,125);
}
```

```
void Stop()
```

```
{
  digitalWrite(m1G,1);
  analogWrite(m1P,0);
  digitalWrite(m2G,1);
  analogWrite(m2P,0);
}
```


3. 아두이도 소스 2

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial bluetooth(2,3);
```

```
char a='x';
```

```
void setup() {  
  pinMode(8,INPUT);  
  pinMode(9,INPUT);  
  pinMode(10,INPUT);  
  pinMode(11,INPUT);  
  pinMode(12,INPUT);  
  Serial.begin(9600);  
  bluetooth.begin(9600);  
}
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  if(digitalRead(8)==1)  
  {  
    Serial.println('8');  
    bluetooth.write('x');  
    //0  
  }  
  else if(digitalRead(9)==1)  
  {  
    Serial.println('9');  
    bluetooth.write('b');  
    //3  
  }  
  else if(digitalRead(10)==1)  
  {  
    Serial.println('a');  
    bluetooth.write('d');  
    //5  
  }  
  else if(digitalRead(11)==1)  
  {
```

```
Serial.println('b');  
bluetooth.write('c');  
//4  
}  
else if(digitalRead(12)==1)  
{  
Serial.println('c');  
bluetooth.write('a');  
//2  
}  
else  
delay(10);  
}
```

Ⅲ. 알티노를 이용한 영상인식 주행 로봇

1. 메인소스코드

```
////////////////////////////////////  
// 알티노를 이용한 영상 인식 주행 로봇 메인 소스코드 //  
////////////////////////////////////  
// This source all right resered SAEON //  
////////////////////////////////////  
  
//Header area////////////////////////////////////  
#include "iostream"  
#include "fstream"  
#include "cstdlib"  
#include "raspicam_still_cv.h"  
#include "stdio.h"  
#include "pthread.h"  
#include "string.h"  
#include "stdlib.h"  
#include "vector"  
#include "map"  
#include "opencv2/core.hpp"  
#include "opencv2/imgproc.hpp"  
#include "opencv2/highgui.hpp"  
#include "opencv2/videoio.hpp"  
#include "opencv2/calib3d.hpp"  
#include "opencv2/imgcodecs.hpp"  
#include "opencv2/features2d.hpp"  
#include "opencv2/xfeatures2d.hpp"  
#include "opencv2/imgcodecs.hpp"  
#include "opencv2/opencv.hpp"  
#include "vector"  
#include "errno.h"  
#include "wiringPi.h"  
#include "wiringSerial.h"  
#include "raspicam_cv.h"
```

```

#include "ctime"
#include <sys/time.h>
#include <signal.h>
#include "Altino.h"
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <softPwm.h>
//Header area////////////////////////////////////

void mission1(void);
void mission2(void);
void mission2_2(void);
void mission3(void);
void mission3_2(void);
void mission4(void);
void mission5(void);
//Define area////////////////////////////////////
#define IR_INTEGRAL_VALUE 5
#define TIMER_VALUE 20000
//Define area////////////////////////////////////

int RobotV[7][2]={};
int Robot1=0;
int Robot2=0;
int Robot3=0;
int Robot4=0;
int Robot5=0;
int turn_C=0;
int c_right=0;
int c_left=0;
int missionb=0;
int line_trace_count=0;

//Namespace area////////////////////////////////////

```

```

using Namespace std;
using Namespace cv;
using Namespace cv::xfeatures2d;
//Namespace area////////////////////////////////////

//Create Object////////////////////////////////////
raspicam::RaspiCam_Still_Cv Camera;
SensorData AltinoSensor_id1;
SensorData AltinoSensor_id129;
//Create Object////////////////////////////////////

//Variable area////////////////////////////////////
Mat video_img; //original img
unsigned int u16_capture_counter = 0;
unsigned char u8_capture_flg = 0;
int lower_color[3]={3,3,125};
int higher_color[3]={40,40,255};
//Variable Altino control////////////////////////////////////
unsigned char u8_steering_value = 0;
unsigned char u8_steering_mode = 0;
unsigned char u8_steering_direction_value = 0;
int s16_left_motor_value = 0;
int s16_right_motor_value = 0;
unsigned char u8_sound = 0;
unsigned char u8_dot_command = 0;
unsigned char u8_dot_array[8] = {0, 0, 0, 0, 0, 0, 0, 0};
unsigned char u8_led;

unsigned int u16_ir_sensor[6] = {0, 0, 0, 0, 0, 0};
unsigned int u16_drive_torque[2] = {0, 0};
signed int s16_gsensor[3] = {0, 0, 0};
signed int s16_msensor[3] = {0, 0, 0};
unsigned int u16_steering_position = 0;
unsigned int u16_steering_torque = 0;

```

```

unsigned int u16_cds = 0;
unsigned int u16_temp = 0;
unsigned char u8_ir_remote = 0;
unsigned int u16_battery = 0;

signed int s16_ir_move_vector[3][6] = {(0,0,0,0,0,0), (0,0,0,0,0,0), (0,0,0,0,0,0)};
signed int s16_ir_integral[2][6] = {(0,0,0,0,0,0), (0,0,0,0,0,0)};
unsigned char u8_ir_integral_counter = 0;

unsigned char u8_steer_status_var = 0;
unsigned int u16_strategy_value[2] = {0, 0};
unsigned char u8_final_status_Machine_code = 0;
//Variable Altino control////////////////////////////////////

unsigned char u8_drive_conter = 0;
unsigned char u8_drive_status = 0;
unsigned char u8_wall_counter = 0;

//Variable Vision control////////////////////////////////////
Mat arrow_left;
Mat arrow_right;

double d_arrow_left_probability = 0;
double d_arrow_right_probability = 0;
unsigned char u8_arrow_direction_flg = 0; //0 Emty 1 Right 2 Left
//Variable Vision control////////////////////////////////////

//Variable area////////////////////////////////////

//thread Function////////////////////////////////////
pthread_t thread_m_1; //Video capture//
pthread_t thread_m_2; //
pthread_t thread_m_3; //
int thread_err_m_1 = 0;

```

```

int thread_err_m_2 = 0;
int thread_err_m_3 = 0;
//thread Function////////////////////////////////////
SensorData Robot;

//parameter read////////////////////////////////////
float getParamVal ( string id,int argc,char **argv,float defvalue ) {
    for ( int i=0; i<argc; i++ )
        if ( id== argv[i] )
            return atof ( argv[i+1] );
    return defvalue;
}
//parameter read////////////////////////////////////

//camera open////////////////////////////////////
void camera_setting(int argc, char *argv[])
{
    Camera.set ( CV_CAP_PROP_FRAME_WIDTH,    getParamVal (
"-w",argc,argv,320 ) );
    Camera.set ( CV_CAP_PROP_FRAME_HEIGHT,    getParamVal (
"-h",argc,argv,240 ) );
    Camera.set ( CV_CAP_PROP_BRIGHTNESS,getParamVal (
"-br",argc,argv,80 ) );
    Camera.set ( CV_CAP_PROP_CONTRAST ,getParamVal (
"-co",argc,argv,60 ) );
    Camera.set ( CV_CAP_PROP_SATURATION,    getParamVal (
"-sa",argc,argv,60 ) );
    Camera.set ( CV_CAP_PROP_GAIN, getParamVal ( "-g",argc,argv ,50 )
);
    Camera.set ( CV_CAP_PROP_EXPOSURE, getParamVal ( "-ss",argc,argv,
50 ) );
    Camera.open();
}
//camera open////////////////////////////////////

```

```

//Vision templat matching////////////////////////////////////
double templat_match(Mat video_mat, Mat target_mat)
{
    Mat img_display;
    Mat result;
    int match_method = 5;
    video_mat.copyTo(img_display);
    int result_cols = video_mat.cols - target_mat.cols + 1;
    int result_rows = video_mat.rows - target_mat.rows + 1;
    result.create(result_rows, result_cols, CV_32FC1);
    matchTemplate(video_mat, target_mat, result, match_method);
    double minVal; double maxVal; Point minLoc; Point maxLoc; Point
matchLoc;
    minMaxLoc(result, &minVal, &maxVal, NULL, &maxLoc);
    //matchLoc = minLoc;
    matchLoc = maxLoc;
    rectangle(img_display, maxLoc, Point(maxLoc.x+target_mat.cols,
maxLoc.y+target_mat.rows), Scalar::all(0), 2, 8, 0);
    //cout << "X=" << maxLoc.x << " Y=" << maxLoc.y << "
Maxvalue=" << maxVal << endl;
    imshow("templat match", img_display);
    waitKey(10);
    return maxVal;
}
//Vision templat matching////////////////////////////////////

```

```

//Perfomance setting////////////////////////////////////
void performance(Mat video_mat){
    Mat img_display;
    Mat hsv;
    Mat mask,mask1,mask2;
    Mat result;
    int morph_size=2;
    //Mat element=getStructuringElement

```



```

video_mat.copyTo(result);
result.convertTo(result,CV_16U);
cvtColor(video_mat,hsv,CV_BGR2HSV);
inRange(hsv,Scalar(0,50,50),Scalar(10,255,255),mask1);
inRange(hsv,Scalar(170,70,50),Scalar(180,255,255),mask2);
mask=mask1|mask2;
morphologyEx(mask,mask,MORPH_OPEN,2);
vector<Point> pts;
findNonZero(mask,pts);
Rect box=boundingRect(pts);
rectangle(video_mat,box,Scalar(0,0,255),3);
imshow("result2",mask);
imshow("Color",video_mat);
printf("x:%d,y:%d",box.x,box.y);
if(box.x>120){//rightturn
    Steering(3);
    Go(350,350);
    Go(250,250);
    Robot=Sensor(1);
    if(Robot.IRSensor[0]>202  &&  Robot.IRSensor[1]>260  &&
Robot.IRSensor[2]>150)
    {
        while(1)
        {
            Go(0,0);
            Display('D');
            delay(1000);
            Display('I');
            delay(1000);
            Display('C');
            delay(1000);
        }
    }
}
}

```

```

else if(box.x<120){//leftturn
    Steering(1);
    Go(350,350);
    Go(250,250);
    Robot=Sensor(1);
}

}

//Perfomance setting////////////////////////////////////

//Vision hough transform////////////////////////////////////
void hough_transform(Mat video_mat)
{

    Mat gray_video;
    Mat result;
    int failed_count=0;
    video_mat.copyTo(result);
    cvtColor(video_mat, gray_video, CV_RGB2GRAY);
    Canny(gray_video, gray_video, 100, 200, 3);
    //vector<Vec2f> lines;
    vector<Vec4i> lines;
    HoughLinesP(gray_video, lines, 1, CV_PI/180, 100, 0, 0);
    Vec4i params;
    int x1,y1,x2,y2;
    for(int k=0; k<lines.size();k++){
        params=lines[k];
        x1=params[0];
        y1=params[1];
        x2=params[2];
        y2=params[3];
    }
}

```

```

        Point pt1(x1,y1),pt2(x2,y2);

if(x1+2<=x2&&x2>=x1-2&&lines.size()>110&&lines.size()<180&&y1>170&&y2
>170){

                printf("lines[%2d]=P1(%4d,%4d)=P2(%4d,%4d) count:
%d lineSize: %d\n",k,x1,y1,x2,y2,line_trace_count,lines.size());
                line(result,pt1,pt2,Scalar(0,0,255),2);
                if(y1>310&&y2>310){line_trace_count++;}
        }
        //imshow("hough transform", result); waitKey(10);

}
/*for( size_t i = 0; i < lines.size(); i++ )
{

        float rho = lines[i][0], theta = lines[i][1];
        Point pt1, pt2;
        double a = cos(theta), b = sin(theta);
        double x0 = a*rho, y0 = b*rho;
        pt1.x = cvRound(x0 + 1000*(-b));
        pt1.y = cvRound(y0 + 1000*(a));
        pt2.x = cvRound(x0 - 1000*(-b));
        pt2.y = cvRound(y0 - 1000*(a));
        line(result, pt1, pt2, Scalar(0,0,255), 3, CV_AA);
}*/
//imshow("canny edge", gray_video);

}

//Vision hough transform////////////////////////////////////

//img capture////////////////////////////////////
void img_capture(void)
{

```

```

if ( !Camera.grab ( ) ){ printf("Camera ErrorWn"); }
else
{
    u16_capture_counter++;
    //cout<< u16_capture_counter << endl;
    u8_capture_flg = 1;
    Camera.retrieve (video_img);
    Mat matrotation = getRotationMatrix2D( Point(160, 120), 180, 1);
    warpAffine(video_img, video_img, matrotation, video_img.size());
    u8_capture_flg = 0;
    //imshow("Video image", video_img);
    d_arrow_right_probability =   templat_match(video_img,
arrow_right);
    d_arrow_left_probability =   templat_match(video_img, arrow_left);

    if((d_arrow_right_probability > 0.5) && (d_arrow_left_probability >
0.5))
    {
        //cout   <<   "Right   Probability   =   "   <<
d_arrow_right_probability
        //<< "Left Probability = " << d_arrow_left_probability;

        if(d_arrow_right_probability > d_arrow_left_probability)
        {
            u8_arrow_direction_flg = 1;
            c_right++;
            //cout << "   It will be right arrow" << endl;
        }
        else
        {
            c_left++;
            u8_arrow_direction_flg = 2;
            //cout << "   It will be left arrow" << endl;
        }
    }
}

```

```

    }
    else
    {
        cout << " Did not find any kind of arrow " << endl;
    }

}

//hough_transform(video_img);
//performance(video_img);
}
//img capture////////////////////////////////////

//altino control send////////////////////////////////////
void altino_send_Data(void)
{
    //steering control////////////////////////////////////
    Sendbuf[5] = u8_steering_value; //steering value
    Sendbuf[24] = u8_steering_mode; //steering mode
    Sendbuf[25] = u8_steering_direction_value; //steering direction value
    //steering control////////////////////////////////////

    //drive motor control////////////////////////////////////

    if(s16_right_motor_value > 1000){ s16_right_motor_value = 1000; }
    else if(s16_right_motor_value < -1000){ s16_right_motor_value =
-1000; }
    else{ }

    if(s16_left_motor_value > 1000){ s16_left_motor_value = 1000; }
    else if(s16_left_motor_value < -1000){ s16_left_motor_value = -1000;
}
    else{ }
}

```

```
    if(s16_right_motor_value < 0){ s16_right_motor_value = 32768 -  
s16_right_motor_value; }  
    if(s16_left_motor_value < 0){ s16_left_motor_value = 32768 -  
s16_left_motor_value; }
```

```
    Sendbuf[7] = (byte)(s16_right_motor_value / 256);  
    Sendbuf[8] = (byte)(s16_right_motor_value % 256);
```

```
    Sendbuf[10] = (byte)(s16_left_motor_value / 256);  
    Sendbuf[11] = (byte)(s16_left_motor_value % 256);  
    //drive motor control////////////////////////////////////
```

```
    //sound control////////////////////////////////////  
    Sendbuf[22] = u8_sound;  
    //sound control////////////////////////////////////
```

```
    //disPlay control////////////////////////////////////  
    Sendbuf[12] = u8_dot_command;  
    Sendbuf[13] = u8_dot_array[0];  
    Sendbuf[14] = u8_dot_array[1];  
    Sendbuf[15] = u8_dot_array[2];  
    Sendbuf[16] = u8_dot_array[3];  
    Sendbuf[17] = u8_dot_array[4];  
    Sendbuf[18] = u8_dot_array[5];  
    Sendbuf[19] = u8_dot_array[6];  
    Sendbuf[20] = u8_dot_array[7];  
    //disPlay control////////////////////////////////////
```

```
    //led control////////////////////////////////////  
    Sendbuf[23] = u8_led;  
    //led control////////////////////////////////////
```

```
    SendData(Sendbuf);
```

}

```

//altino control send////////////////////////////////////

//IR sensor vector control //////////////////////////////////
void ir_sensor_vector_calculator(void)
{
    s16_ir_move_vector[0][0] = u16_ir_sensor[0];
    s16_ir_move_vector[0][1] = u16_ir_sensor[1];
    s16_ir_move_vector[0][2] = u16_ir_sensor[2];
    s16_ir_move_vector[0][3] = u16_ir_sensor[3];
    s16_ir_move_vector[0][4] = u16_ir_sensor[4];
    s16_ir_move_vector[0][5] = u16_ir_sensor[5];

    s16_ir_move_vector[2][0]      =      s16_ir_move_vector[1][0]      -
s16_ir_move_vector[0][0];
    s16_ir_move_vector[2][1]      =      s16_ir_move_vector[1][1]      -
s16_ir_move_vector[0][1];
    s16_ir_move_vector[2][2]      =      s16_ir_move_vector[1][2]      -
s16_ir_move_vector[0][2];
    s16_ir_move_vector[2][3]      =      s16_ir_move_vector[1][3]      -
s16_ir_move_vector[0][3];
    s16_ir_move_vector[2][4]      =      s16_ir_move_vector[1][4]      -
s16_ir_move_vector[0][4];
    s16_ir_move_vector[2][5]      =      s16_ir_move_vector[1][5]      -
s16_ir_move_vector[0][5];

    s16_ir_move_vector[1][0] = s16_ir_move_vector[0][0];
    s16_ir_move_vector[1][1] = s16_ir_move_vector[0][1];
    s16_ir_move_vector[1][2] = s16_ir_move_vector[0][2];
    s16_ir_move_vector[1][3] = s16_ir_move_vector[0][3];
    s16_ir_move_vector[1][4] = s16_ir_move_vector[0][4];
    s16_ir_move_vector[1][5] = s16_ir_move_vector[0][5];

/*
    system("clear");
    printf("IR0= %d ", s16_ir_move_vector[2][0]);

```

```

    printf("IR1= %d ", s16_ir_move_vector[2][1]);
    printf("IR2= %d ", s16_ir_move_vector[2][2]);
    printf("IR3= %d ", s16_ir_move_vector[2][3]);
    printf("IR4= %d ", s16_ir_move_vector[2][4]);
    printf("IR5= %d\n", s16_ir_move_vector[2][5]);
*/

}

//IR sensor vector control //////////////////////////////////////

//IR sensor integral control //////////////////////////////////////
void ir_sensor_integral_control(void)
{
    if(u8_ir_integral_counter < IR_INTEGRAL_VALUE)
    {
        s16_ir_integral[0][0]      =      s16_ir_integral[0][0]      +
s16_ir_move_vector[0][0];
        s16_ir_integral[0][1]      =      s16_ir_integral[0][1]      +
s16_ir_move_vector[0][1];
        s16_ir_integral[0][2]      =      s16_ir_integral[0][2]      +
s16_ir_move_vector[0][2];
        s16_ir_integral[0][3]      =      s16_ir_integral[0][3]      +
s16_ir_move_vector[0][3];
        s16_ir_integral[0][4]      =      s16_ir_integral[0][4]      +
s16_ir_move_vector[0][4];
        s16_ir_integral[0][5]      =      s16_ir_integral[0][5]      +
s16_ir_move_vector[0][5];
        u8_ir_integral_counter++;
    }
    else
    {
        s16_ir_integral[1][0]      =      (s16_ir_integral[0][0]      +
s16_ir_move_vector[0][0]) / (IR_INTEGRAL_VALUE + 1);
        s16_ir_integral[1][1]      =      (s16_ir_integral[0][1]      +
s16_ir_move_vector[0][1]) / (IR_INTEGRAL_VALUE + 1);

```



```

        s16_ir_integral[1][2]      =      (s16_ir_integral[0][2]      +
s16_ir_move_vector[0][2]) / (IR_INTEGRAL_VALUE + 1);
        s16_ir_integral[1][3]      =      (s16_ir_integral[0][3]      +
s16_ir_move_vector[0][3]) / (IR_INTEGRAL_VALUE + 1);
        s16_ir_integral[1][4]      =      (s16_ir_integral[0][4]      +
s16_ir_move_vector[0][4]) / (IR_INTEGRAL_VALUE + 1);
        s16_ir_integral[1][5]      =      (s16_ir_integral[0][5]      +
s16_ir_move_vector[0][5]) / (IR_INTEGRAL_VALUE + 1);

```

```

        s16_ir_integral[0][0] = s16_ir_move_vector[0][0];
        s16_ir_integral[0][1] = s16_ir_move_vector[0][1];
        s16_ir_integral[0][2] = s16_ir_move_vector[0][2];
        s16_ir_integral[0][3] = s16_ir_move_vector[0][3];
        s16_ir_integral[0][4] = s16_ir_move_vector[0][4];
        s16_ir_integral[0][5] = s16_ir_move_vector[0][5];

```

```

        u8_ir_integral_counter = 0;

```

```

    }

```

```

    printf("IR0= %d ", s16_ir_integral[1][0] );
    printf("IR1= %d ", s16_ir_integral[1][1] );
    printf("IR2= %d ", s16_ir_integral[1][2] );
    printf("IR3= %d ", s16_ir_integral[1][3] );
    printf("IR4= %d ", s16_ir_integral[1][4] );
    printf("IR5= %d ", s16_ir_integral[1][5] );
    printf("arrow %d ", u8_arrow_direction_flg);
    printf("status %d\n", u8_drive_status);

```

```

}

```

```

//IR sensor integral control //////////////////////////////////////

```

```

//drive steering control //////////////////////////////////////

```

```

void drive_direction_maker(signed int t8_motor_right_value, signed int
t8_motor_left_value)

```

```

{

```

```

s16_right_motor_value = t8_motor_right_value;
s16_left_motor_value = t8_motor_left_value;
}

void drive_steer_maker(unsigned char t8_steer_value, unsigned char
t8_steer_mode_value)
{
    /*
    if(t8_steer_value < 128)

    {
        //if(s16_ir_integral[1][0] > 100){ u8_steering_mode = 1;
u8_steering_value = 100; }
        if(s16_ir_integral[1][0] > 50){ u8_steering_mode = 0;
u8_steering_value = 3;}
        else{ u8_steering_mode = 1; u8_steering_value =
s16_ir_integral[1][0] * 2; }
    }
    else
    {
        //if(s16_ir_integral[1][2] > 100){ u8_steering_mode = 1;
u8_steering_value = 100; }
        if(s16_ir_integral[1][2] > 50){ u8_steering_mode = 0;
u8_steering_value = 1;}
        else{ u8_steering_mode = 1; u8_steering_value =
s16_ir_integral[1][2] * 2 + 128; }
    }
    */
    //printf("%d", u8_steering_value);
    u8_steering_value = t8_steer_value;
    u8_steering_mode = t8_steer_mode_value;
}

```

```

void drive_control(void)
{
    //drive_direction_maker(0, 0);
    //drive_steer_maker(100,1);

    switch(u8_final_status_Machine_code)
    {
        case 0: // Error Status //
            drive_direction_maker(0, 0);
            drive_steer_maker(0,1);
            break;

        case 1: // Drive Front //
            drive_direction_maker(300, 300);
            drive_steer_maker(0,1);
            break;

        case 2: // Drive Rear //
            drive_direction_maker(-300, -300);
            drive_steer_maker(0,1);
            break;

        case 5: // Drive Forward Turn Right //
            drive_direction_maker(300, 300);
            drive_steer_maker(100,1);
            break;

        case 6: // Drive Rear Turn Right //
            drive_direction_maker(-300, -300);
            drive_steer_maker(100,1);
            break;

        case 9: // Drive Forward Turn Left //
            drive_direction_maker(300, 300);
            drive_steer_maker(100+128,1);
    }
}

```

```

        break;

        case 10: // Drive Rear Turn Left //
            drive_direction_maker(-300, -300);
            drive_steer_maker(100+128,1);
        break;
    }
}

//drive Function////////////////////////////////////
void drive_wall_dect(void)
{
    if((u8_steer_status_var & 0x07) == 0x07)
    {
        u8_wall_counter++;
        if(u8_wall_counter > 200){ u8_drive_status++;
u8_wall_counter = 0; }
        else{ u8_wall_counter++; }
    }
    else
    {
        u8_wall_counter = 0;
    }
}

void drive_wall_disappear(void)
{
    if((u8_steer_status_var & 0x07) == 0x00)
    {
        u8_wall_counter++;
        if(u8_wall_counter > 200){ u8_drive_status++;
u8_wall_counter = 0; }
        else{ u8_wall_counter++; }
    }
}

```

```

        else
        {
            u8_wall_counter = 0;
        }
    }

void drive_left_wall_dect(void)
{
    if((u8_steer_status_var & 0x04) == 0x04)
    {
        u8_wall_counter++;
        if(u8_wall_counter > 200){ u8_drive_status++;
u8_wall_counter = 0; }
        else{ u8_wall_counter++; }
    }
    else
    {
        u8_wall_counter = 0;
    }
}

```

```

void drive_right_wall_dect(void)
{
    if((u8_steer_status_var & 0x01) == 0x01)
    {
        u8_wall_counter++;
        if(u8_wall_counter > 200){ u8_drive_status++;
u8_wall_counter = 0; }
        else{ u8_wall_counter++; }
    }
    else
    {
        u8_wall_counter = 0;
    }
}

```

```

void drive_front(void)
{
    if(u8_wall_counter == 0){ drive_direction_maker(250, 250);
drive_steer_maker(0,1); }
    else{ drive_direction_maker(0, 0); }
    drive_wall_dect();
}

```

```

void drive_back(unsigned char t8_streering_buf)
{
    if(u8_wall_counter == 0){ drive_direction_maker(-400, -400);
drive_steer_maker(t8_streering_buf,1); }
    else{ drive_direction_maker(0, 0); }//
    drive_wall_disappear();
}

```

```

void turn_left(void)
{
    if(u8_wall_counter == 0){ drive_direction_maker(300, 300);
drive_steer_maker(110+128,1); }
    else{ drive_direction_maker(0, 0); }
    drive_left_wall_dect();
}

```

```

void turn_right(void)
{
    if(u8_wall_counter == 0){ drive_direction_maker(300, 300);
drive_steer_maker(110,1); }
    else{ drive_direction_maker(0, 0); }
    drive_right_wall_dect();
}

```

```

void mission1(void)
{
    Go(350,350);
    delay(400);
    Go(0,0);
    while(Robot.IRSensor[1]<2700)
    {
        Go(400,400);
        Go(230,230);
        Robot=Sensor(1);
        for(int countIR=0;countIR<6;countIR++)
            {
                printf("print  %d=
%d  ",countIR,Robot.IRSensor[countIR]);
                Robot=Sensor(1);
            }
        printf("\n");
    }

    Go(-350,-350);
    delay(400);
    Go(0,0);

    printf("aaaaaaaaadADADA");
    Go(0,0);
    delay(6000);
    Robot=Sensor(1);
    Go(-400,-400);
    delay(1200);
    Go(0,0);
    delay(300);
    if(c_right>c_left)
    {

```

```

        mission2();
        mission3();
        mission4();
        mission5();
    }
    else if(c_left>c_right)
    {
        mission2_20();
        mission3_20();
    }
    else
    {
        mission2();
        mission3();
        mission4();
        mission5();
    }
}

```

```
void mission2(void)
```

```

{
    int RightV=0;
    Robot=Sensor(1);
    if(c_right>c_left)
    {
        while(Robot.IRSensor[1]<275 && Robot.IRSensor[0]<200 &&
Robot.IRSensor[2]<143)
        {
            Robot=Sensor(1);
            Steering(3);
            Go(450,450);
            Go(270,270);
        }
    }
}

```



```

}

Go(0,0);
delay(1000);

while(RightV<12000)
{

    Steering(3);
    Go(-230,-230);
    Go(-400,-400);
    printf("aaaaaa\n");

    Robot=Sensor(1);
    while(Robot.IRSensor[2]<162)
    {

        printf("oooooooooooooooooooooooo\n");
        Steering(2);
        Go(450,400);
        Go(240,250);
        Robot=Sensor(1);
        printf("RightV= %d\n",RightV);

        if(Robot.IRSensor[3]>135)
        {
            RightV++;
        }
        else

        if(Robot.IRSensor[3]<108 && RightV>1)
        {
            Go(0,0);
            while(Robot.IRSensor[2]<180           &&

```

```
Robot.IRSensor[1]<350)
```

```
{  
    Go(450,450);  
    Go(250,250);  
    Robot=Sensor(1);  
    printf("ccccccccccccccccccccWn");  
}
```

```
Go(0,0);  
delay(2000);
```

```
while(RightV<9999)
```

```
{  
    Robot.IRSensor[3]=3;  
    while(Robot.IRSensor[3]<118)  
    {  
        Steering(1);  
        Go(-400,-400);  
        Go(-250,-250);  
        Robot=Sensor(1);  
    }  
    Go(0,0);  
    delay(300);  
    Steering(2);  
    Go(350,350);  
    delay(400);  
    Go(0,0);  
    delay(300);
```

```
while(Robot.IRSensor[2]<160)
```

```
{
```

```
printf("aaaaaaaaaaaaaaaaWn");
```

```
Steering(3);
```

```

        Go(400,400);
        Go(250,250);
        Go(200,200);
        Robot=Sensor(1);
    }

    Go(-330,-330);
    delay(700);
    Go(0,0);
    delay(1000);
    Steering(1);
    Go(350,350);
    delay(400);

    while(Robot.IRSensor[0] < 500    &&
Robot.IRSensor[2] < 500)
    {

    printf("bbbbbbbbbbbbbbWn");

        Steering(2);
        Go(400,400);
        Go(250,250);
        Robot=Sensor(1);
    for(int countIR=0;countIR<6;countIR++)
        {
            printf("print  %d=
%d  ",countIR,Robot.IRSensor[countIR]);

            Robot=Sensor(1);
        }
        printf("Wn");
    }
}

```

```

Steering(2);
Go(-350,350);
delay(400);

Go(0,0);
RightV=14000;
//mission5();
Display('1');
delay(1000);
Display('2');
delay(1000);
Display('3');
delay(1000);
Display(0);
printf("iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiWn");

```

```

}
```

```

}
```

```

}
```

```

}
```

```

}
```

```

void mission2_2(void)

```

```

{

```

```

    int RightV=0;

```

```

    Robot=Sensor(1);

```

```

    if(c_right>c_left)

```

```

    {

```

```

        while(Robot.IRSensor[1]<275 && Robot.IRSensor[0]<200 &&

```

```

Robot.IRSensor[2]<143)

```

```

        {

```

```

        Robot=Sensor(1);
        Steering(3);
        Go(450,450);
        Go(270,270);
    }
}

Go(0,0);
delay(1000);

while(RightV<12000)
{

    Steering(1);
    Go(-230,-230);
    Go(-400,-400);
    printf("aaaaaa\n");

    Robot=Sensor(1);
    while(Robot.IRSensor[0]<200)
    {
        printf("oooooooooooooooooooo\n");
        Steering(2);
        Go(450,400);
        Go(240,250);
        Robot=Sensor(1);
        printf("RightV= %d\n",RightV);

        if(Robot.IRSensor[4]>220)
        {
            RightV++;
        }
        else

```

```

if(Robot.IRSensor[4]<220 && RightV>1)
{
    Go(0,0);
    while(Robot.IRSensor[0]<230           &&
Robot.IRSensor[1]<350)
    {
        Go(450,450);
        Go(250,250);
        Robot=Sensor(1);
        printf("ccccccccccccccccccccWn");
    }

    Go(0,0);
    delay(2000);

    while(RightV<9999)
    {
        Robot.IRSensor[4]=3;
        while(Robot.IRSensor[4]<250)
        {
            Steering(1);
            Go(-400,-400);
            Go(-250,-250);
            Robot=Sensor(1);
        }
        Go(0,0);
        delay(300);
        Steering(2);
        Go(350,350);
        delay(400);
        Go(0,0);
        delay(300);
    }
}

```

```

while(Robot.IRSensor[0]<200)
{
printf("aaaaaaaaaaaaaaaaWn");

Steering(1);
Go(400,400);
Go(250,250);
Go(200,200);
Robot=Sensor(1);
}

Go(-330,-330);
delay(700);
Go(0,0);
delay(1000);
Steering(3);
Go(350,350);
delay(400);

while(Robot.IRSensor[0]<500 &&
Robot.IRSensor[2]< 500)
{
printf("bbbbbbbbbbbbWn");

Steering(2);
Go(400,400);
Go(250,250);
Robot=Sensor(1);
f o r ( i n t
countIR=0;countIR<6;countIR++)
{
printf("print %d=
%d ",countIR,Robot.IRSensor[countIR]);

Robot=Sensor(1);
}
}

```

```
    }  
    printf("\n");  
}
```

```
Steering(1);  
Go(-350,350);  
delay(400);
```

```
Go(0,0);  
RightV=14000;  
//mission5();  
DisPlay('1');  
delay(1000);  
DisPlay('2');  
delay(1000);  
DisPlay('3');  
delay(1000);  
DisPlay(0);  
printf("iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii\n");
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
void mission3(void)
```

```
{
```

```
    printf("mission3\n");
```



```

Steering(1);
Go(-350,-350);
delay(800);
Steering(2);
while(Robot.IRSensor[5]<200)
{
    Go(0,0);
    Go(-200,-200);
    Go(-450,-450);
    Robot=Sensor(1);
}
Go(0,0);
delay(1000);

Steering(1);
Go(350,350);
delay(1200);

while(Robot.IRSensor[1]<245 && Robot.IRSensor[2]<135)
{
    Steering(2);
    Go(270,270);
    Go(350,350);
    Robot=Sensor(1);
}

Go(0,0);
Steering(2);
}

void mission3_2(void)
{
    printf("mission3\n");
}

```

```

Steering3;
Go(-350,-350);
delay(800);
Steering(2);
while(Robot.IRSensor[5]<200)
{
    Go(0,0);
    Go(-200,-200);
    Go(-450,-450);
    Robot=Sensor(1);
}
Go(0,0);
delay(1000);

Steering(3);
Go(350,350);
delay(1200);

while(Robot.IRSensor[1]<245 && Robot.IRSensor[0]<180)
{
    Steering(2);
    Go(270,270);
    Go(350,350);
    Robot=Sensor(1);
}

Go(0,0);
Steering(2);
}

void mission4(void)
{
    line_trace_count=0;
    printf("mission4\n");
}

```

```

while(1)
{
    Steering(2);
    Go(400,400);
    Go(270,270);
    Robot=Sensor(1);

    while(Robot.IRSensor[0]>202 && Robot.IRSensor[1]>276 &&
Robot.IRSensor[2]>140)
    {
        Steering(3);
        Go(-300,-300);
        Steering(2);
        Go(-300,-300);
        Robot=Sensor(1);
    }
    while(Robot.IRSensor[0]>202 && Robot.IRSensor[0]<280)
    {

        Steering(3);
        Go(350,350);
        Robot=Sensor(1);
    }
    while(Robot.IRSensor[0]>280)
    {

        Steering(1);
        Go(-350,-300);
        Steering(2);
        Go(-350,-350);
        Robot=Sensor(1);
    }
}

```

```

while(Robot.IRSensor[2]>147 && Robot.IRSensor[2]<180)
{
    Steering(1);
    Go(350,300);
    Robot=Sensor(1);
}

```

```

while(Robot.IRSensor[2]>170)
{
    Steering(3);
    Go(-350,-300);
    Steering(2);
    Go(-350,-350);
    Robot=Sensor(1);
    if(missionb==1)
    {
        missionb=2;
    }
}

```

```

while(Robot.IRSensor[4]>260)
{
    Steering(3);
    Go(350,350);
    Steering(2);
    Go(270,270);
    Robot=Sensor(1);
    missionb=1;

    for(int countIR=0;countIR<6;countIR++)
    {
        printf("print %d= %d ",countIR,Robot.IRSensor[countIR]);
        Robot=Sensor(1);
    }
}

```

```

    }
    printf("\n");

    Go(0,0);

}

while(Robot.IRSensor[3]>105 && Robot.IRSensor[3]<700)
{

    Steering(1);
    Go(350,350);
    Steering(2);
    Go(270,270);
    Robot=Sensor(1);
}
if(missionb==2)
{

    Go(0,0);
    Steering(1);
    Go(400,400);
    delay(800);
    Go(0,0);
    delay(3000);
    for(int i=0;i<70000000;i++){
        missionb++;
    }
}

Go(0,0);
delay(1000);
}
//cross
void mission5(){
    line_trace_count=0;
    printf("mission4\n");
}

```

```

while(1)
{

    Steering(2);
    Go(400,400);
    Go(270,270);
    Robot=Sensor(1);
    if(Robot2>2)
        break;

    while(Robot.IRSensor[0]>202 && Robot.IRSensor[1]>276 &&
Robot.IRSensor[2]>140)
    {
        Steering(3);
        Go(-300,-300);
        Steering(2);
        Go(-300,-300);
        Robot=Sensor(1);
    }

    if(Robot2>2)
        break;

    while(Robot.IRSensor[0]>202 && Robot.IRSensor[0]<280)
    {

        Steering(3);
        Go(350,350);
        Robot=Sensor(1);
    }
    if(Robot2>2)
        break;

```

```

while(Robot.IRSensor[0]>280)
{

    Steering(1);
    Go(-350,-300);
    Steering(2);
    Go(-350,-350);
    Robot=Sensor(1);
}
if(Robot2>2)
    break;

while(Robot.IRSensor[2]>147 && Robot.IRSensor[2]<180)
{

    Steering(1);
    Go(350,300);
    Robot=Sensor(1);
}
if(Robot2>2)
break;

while(Robot.IRSensor[2]>170)
{

    Steering(3);
    Go(-350,-300);
    Steering(2);
    Go(-350,-350);
    Robot=Sensor(1);
    if(missionb==1)
    {
        missionb=2;
    }
}

```

```

if(Robot2>2)
    break;
while(Robot.IRSensor[4]>260)
{
    Steering(3);
    Go(350,350);
    Steering(2);
    Go(270,270);
    Robot=Sensor(1);

}
if(Robot2>2)
    break;

while(Robot.IRSensor[3]>105 && Robot.IRSensor[3]<700)
{

    Steering(1);
    Go(350,350);
    Steering(2);
    Go(270,270);
    Robot=Sensor(1);

}
if(Robot.IRSensor[0]<180 && Robot.IRSensor[1]<240 &&
Robot.IRSensor[2]<140)
{
    Robot4++;
    printf(" %d \n",Robot4);
}
}

```



```

        Go(0,0);
        delay(1000);
    }
    void turn_check(void){

        if(u8_arrow_direction_flg==1){
            mission1();
        }
        else if(u8_arrow_direction_flg==2){
            turn_left();
        }
        else{printf("cannot find Arrow!");}
        drive_direction_maker(300, 300);
        u8_drive_status=3;
    }

    void drive_rear(void)
    {
    }

    void drive_Function(void)
    {
        switch(u8_drive_status)
        {
            case 0:
                delay(1000);
                break;

            case 1:
                drive_back(50+128);
                break;

            case 2:

```

```
        turn_check();
    break;

    case 3:
        drive_back(70);
    break;

    case 4:
        drive_control();
    break;

/*
    case 5:
        drive_back(0);
    break;

    case 6:
        turn_right();
    break;

    case 7:
        drive_back(0);
    break;

    case 8:
        drive_front();
    break;

*/
    default:
    break;
}
}
```

```
//drive Function////////////////////////////////////
```

```
void strategy_maker(void)
```

```
{
```

```
    switch(u8_steer_status_var)
```

```
    {
```

```
        case 0: case 5: case 8: case 13: case 16: case 21: case 24:  
        case 29: case 32: case 37: case 40: case 45: case 48: case
```

```
53: case 56:
```

```
        case 61: case 64:
```

```
            u8_final_status_Machine_code = 1;
```

```
        break;
```

```
        case 2: case 7: case 10: case 15: case 18: case 23: case 26:
```

```
case 31:
```

```
        case 34: case 39: case 42: case 47: case 50: case 55: case
```

```
58: case 63:
```

```
            u8_final_status_Machine_code = 2;
```

```
        break;
```

```
        case 1:case 9:case 17:case 25:case 33:case 41:case 49:case
```

```
57:
```

```
            u8_final_status_Machine_code = 5;
```

```
        break;
```

```
        case 6:case 14:case 22:case 30:case 38:case 46:case 54:case
```

```
62:
```

```
            u8_final_status_Machine_code = 6;
```

```
        break;
```

```
        case 4:case 12:case 20:case 28:case 36:case 44:case 52:case
```

```
60:
```

```
            u8_final_status_Machine_code = 9;
```

```
        break;
```

```

        case 3:case 11:case 19:case 27:case 35:case 43:case 51: case
59:
            u8_final_status_Machine_code = 10;
        break;

        default:
            u8_final_status_Machine_code = 0;
        break;
    }
}

```

```

void drive_status_maker(void)
{
    //basic status maker//////////////////
    u8_steer_status_var = 0;
    if(s16_ir_integral[1][0] > 7){ u8_steer_status_var = 0x01;}
    else{}

    if(s16_ir_integral[1][1] > 7){ u8_steer_status_var = u8_steer_status_var
+ 0x02;}
    else{}

    if(s16_ir_integral[1][2] > 7){ u8_steer_status_var = u8_steer_status_var
+ 0x04;}
    else{}

    if(s16_ir_integral[1][3] > 0){ u8_steer_status_var =u8_steer_status_var
+ 0x08;}
    else{}

    if(s16_ir_integral[1][4] > 0){ u8_steer_status_var = u8_steer_status_var
+ 0x10;}
    else{}
}

```

```

        if(s16_ir_integral[1][5] > 0){ u8_steer_status_var = u8_steer_status_var
+ 0x20;}
        else{}

        //printf("%d", u8_steer_status_var);
        //basic status maker//////////

        //strategy maker//////////
        strategy_maker();
        //strategy maker//////////

        drive_Function();

        //strategy maker//////////
        drive_control();
        //strategy maker//////////
    }
    //drive steering control ////////////////////////////////////////

    //altino control read//////////////////////////////////////////
    void altino_sensor_disPlay(void)
    {
        system("clear");
        printf("IR0= %d ", u16_ir_sensor[0]);
        printf("IR1= %d ", u16_ir_sensor[1]);
        printf("IR2= %d ", u16_ir_sensor[2]);
        printf("IR3= %d ", u16_ir_sensor[3]);
        u16_ir_sensor[4] = AltinoSensor_id1.IRSensor[4];
        u16_ir_sensor[5] = AltinoSensor_id1.IRSensor[5];
        u16_drive_torque[0] = AltinoSensor_id1.TorqueSensor[0];
        u16_drive_torque[1] = AltinoSensor_id1.TorqueSensor[1];
        u16_cds = AltinoSensor_id1.CDSSensor;
        u16_temp = AltinoSensor_id1.TemperatureSensor;
    }

```

```

s16_gsensor[0] = AltinoSensor_id129.GSensor[0];
s16_gsensor[1] = AltinoSensor_id129.GSensor[1];
s16_gsensor[2] = AltinoSensor_id129.GSensor[2];

s16_msensor[0] = AltinoSensor_id129.MSensor[0];
s16_msensor[1] = AltinoSensor_id129.MSensor[1];
s16_msensor[2] = AltinoSensor_id129.MSensor[2];

u16_battery = AltinoSensor_id129.Battery;

//altino_sensor_disPlay();
}
//altino control read////////////////////////////////////

//Thread Function////////////////////////////////////
void *thread_m_1_Function(void *arg)
{
    while(1)
    {
        img_capture();
        //printf("Thread module 1\n");
    }
}
void *thread_m_2_Function(void *arg)
{
    while(1)
    {
        //img_capture();
        //printf("Thread module 2\n");
    }
}
void *thread_m_3_Function(void *arg)
{

```

```

while(1)
{
    //printf("Thread module 3\n");
}
}
//Thread Function////////////////////////////////////

```

```

void altino_read_Data(void)
{
    AltinoSensor_id1 = Sensor(1);
    AltinoSensor_id129 = Sensor(129);

    u16_ir_sensor[0] = AltinoSensor_id1.IRSensor[0];
    u16_ir_sensor[1] = AltinoSensor_id1.IRSensor[1];
    u16_ir_sensor[2] = AltinoSensor_id1.IRSensor[2];
    u16_ir_sensor[3] = AltinoSensor_id1.IRSensor[3];
    u16_ir_sensor[4] = AltinoSensor_id1.IRSensor[4];
    u16_ir_sensor[5] = AltinoSensor_id1.IRSensor[5];
    u16_drive_torque[0] = AltinoSensor_id1.TorqueSensor[0];
    u16_drive_torque[1] = AltinoSensor_id1.TorqueSensor[1];
    u16_cds = AltinoSensor_id1.CDSSensor;
    u16_temp = AltinoSensor_id1.TemperatureSensor;

    s16_gsensor[0] = AltinoSensor_id129.GSensor[0];
    s16_gsensor[1] = AltinoSensor_id129.GSensor[1];
    s16_gsensor[2] = AltinoSensor_id129.GSensor[2];

    s16_msensor[0] = AltinoSensor_id129.MSensor[0];
    s16_msensor[1] = AltinoSensor_id129.MSensor[1];
    s16_msensor[2] = AltinoSensor_id129.MSensor[2];

    u16_battery = AltinoSensor_id129.Battery;

    //altino_sensor_disPlay();
}

```

```
//Timer Function////////////////////////////////////
```

```
void timer_handler (int signum)
```

```
{  
    altino_send_Data();  
    altino_read_Data();  
    ir_sensor_vector_calculator();  
    ir_sensor_integral_control();  
    drive_status_maker();  
}
```

```
void timer_start(void)
```

```
{  
    struct sigaction basic_timer;  
    struct itimerval timer;  
    memset (&basic_timer, 0, sizeof (basic_timer));  
    basic_timer.sa_handler = &timer_handler;  
    sigaction (SIGVTALRM, &basic_timer, NULL);  
    timer.it_value.tv_sec = 0;  
    timer.it_value.tv_usec = TIMER_VALUE;  
    timer.it_interval.tv_sec = 0;  
    timer.it_interval.tv_usec = TIMER_VALUE;  
    setitimer (ITIMER_Virtual, &timer, NULL);  
}
```

```
//Timer Function////////////////////////////////////
```

```
//Main Function////////////////////////////////////
```

```
int main(int argc, char *argv[] )
```

```
{  
    arrow_left = imread("arrow_left.jpg", 1);  
    arrow_right = imread("arrow_right.jpg", 1);  
}
```



```

    camera_setting(argc, *(&argv));
    Open();
    thread_err_m_1 = pthread_create(&thread_m_1, NULL,
thread_m_1_Function, NULL);

    mission1();

    while(1)
    {/*
        int pos=180;
        int dir =1;
        if(wiringPiSetup () ==-1)
            break;

        pinMode(7,OUTPUT);
        softPwmCreate(7,0,254);

        while(1);
        {
            pos+=dir;
            if(pos<180 || pos>194)
                dir*=-1;
            softPwmWrite(7,pos);
            delay(50);
        }*/
    }
}
//Main Function////////////////////////////////////

```

Since 1999



한국로봇산업협회
Korea Association of Robot industry

로봇 생태계를 선도하는 기업과 사람을 위한
국내 최대 커넥티드 플랫폼, 한국로봇산업협회

발행일 : 2017년 2월

저자 : 前서울로봇마이스터고등학교 장병철 교사

발행 및 편집 : 한국로봇산업협회 전략기획팀

연구원 : 한국로봇산업협회 전략기획팀 이경준 팀장, 주재영 전임

- ▶ 본 교재에 게재된 내용은 한국로봇산업협회의 공식 견해와 다를 수 있습니다.
- ▶ 본 교재는 산업통상자원부의 산업별 인적자원개발협의체지원 사업을 통해 제작되었습니다. (비매품)
- ▶ 본 교재의 내용을 무단 전재할 수 없으며 인용할 경우 출처를 반드시 명시하시기 바랍니다.