

머신 러닝

2023 Fall

강미선

• 사람의 지능 행위와 지능 에이전트

- 사람은 주위를 센싱 → 인식 → 적절한 행위를 통해 환경과 상호작용
- 인공지능 제품도 비슷한 과정을 수행해야 쓸모가 있음([그림 7-1] 사례)



• 이 장에서는,

- 간단한 지능 에이전트 구축(사용자 인터페이스를 붙이는 실습)
- 지식 베이스에 대한 간단한 소개

7.1 지능 에이전트 시나리오

- 일상생활에서 특정한 일을 대행해주는 사람을 에이전트라 부름
 - 예, 여행 에이전트는 여행 일정을 짜고 교통편과 숙소를 예약해주는 사람
- 컴퓨터에서 사람 일을 대신하는 소프트웨어를 에이전트라 부름
 - 이 절에서는 사람의 지능적인 일을 대신해주는 지능 에이전트에 대해 다룸

7.1.1 지능 에이전트란

- 웹스터 영어 사전의 정의(인공지능에서는 ②의 정의가 적절)

에이전트(agent)

① one who is authorized to act for or in the place of another

다른 사람 대신 또는 다른 사람을 위하여 행위를 할 자격이 있는 사람

② something that produces or is capable of producing an effect

어떤 효과를 내거나 낼 능력이 있는 것

- ②의 정의에 따르면,

- 돌멩이는 에이전트가 아니고, 바이메탈은 에이전트임
- [그림 7-1]의 음식 인식 앱은 에이전트

7.1.1 지능 에이전트란

• 지능 에이전트([그림 7-2])

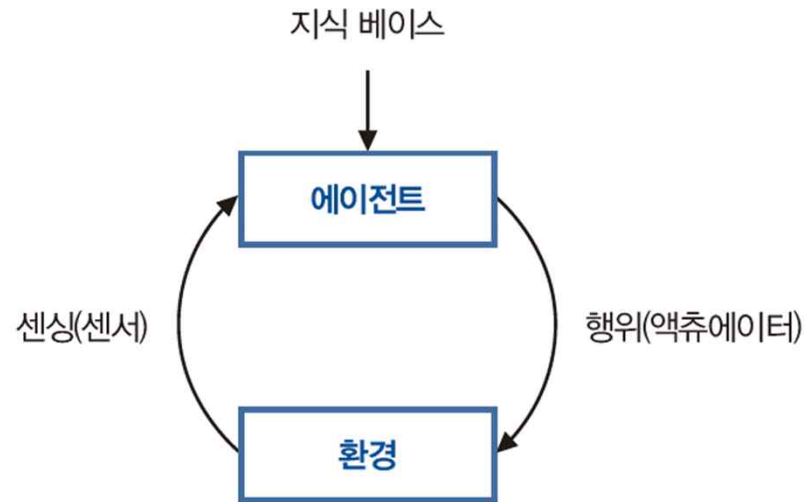


그림 7-2 지능 에이전트

- 로봇은 에이전트, 환경, 지식 베이스가 명확히 구분(로봇 몸과 소프트웨어가 에이전트, 카메라 등이 센서, 로봇 팔이 액추에이터)
- 음식 인식 앱은 구분이 조금 모호(스마트폰과 앱이 에이전트, 카메라는 센서, 화면과 스피커가 액추에이터)

• 지식 베이스

- 지식 표현을 위한 프레임, 의미망, 온톨로지, 규칙 기반 방법으로 구축(7.4절에서 간략히 소개)

7.1.2 라이프 사이클

• 지능 에이전트의 라이프 사이클

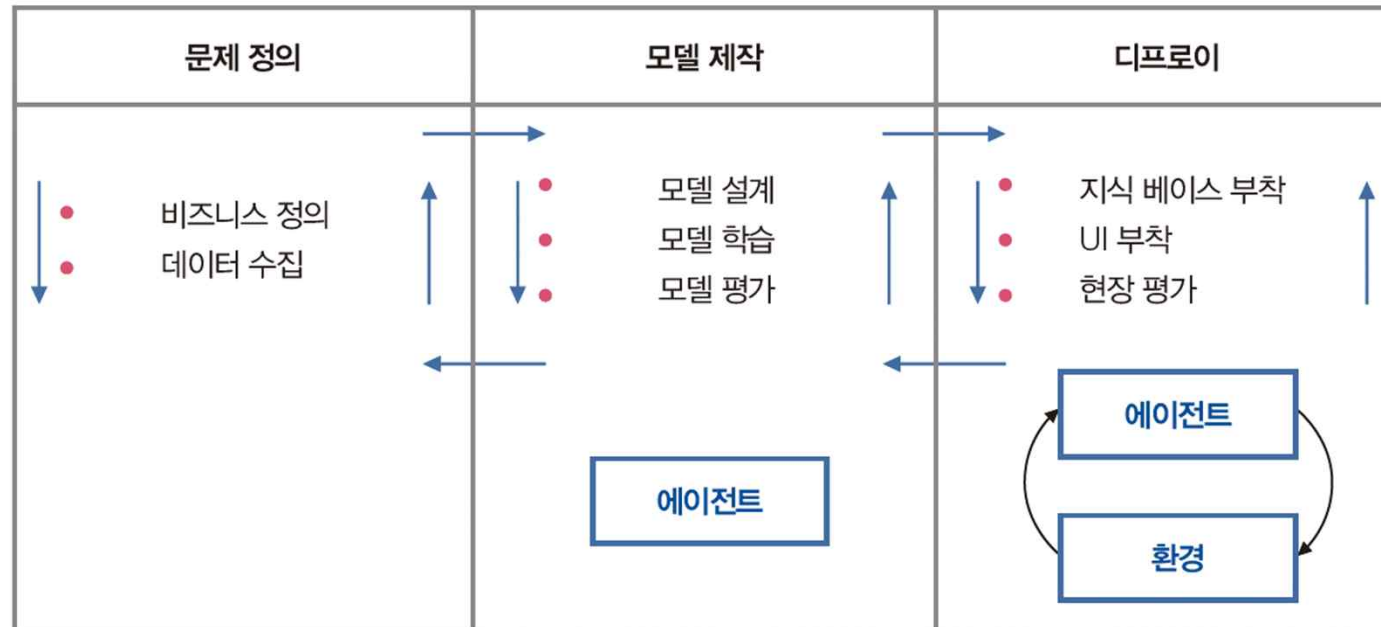


그림 7-3 인공지능 제품의 라이프 사이클

- 단계 1: 문제 정의

- 비즈니스 정의(음식 인식 앱의 경우, 음식의 범위, 음식의 계층 구조 등을 결정)
- 데이터 수집(현장을 대변하는 다양한 음식 영상 수집, 데이터가 가장 중요한 산출물)

7.1.2 라이프 사이클

- **지능 에이전트의 라이프 사이클(...앞에서 계속)**
 - **단계 2: 모델 제작**
 - 컨볼루션 신경망, 순환 신경망, 강화 학습 등에서 어떤 것을 사용할 지 결정
 - 모델의 구조 설계, 모델 학습, 성능 평가를 수행(교차 검증을 사용하여 성능 평가 신뢰도를 확보)
 - **단계 3: 디프로이**
 - 자체 서버 사용 또는 클라우드를 구매해 사용
 - 현대에는 주로 웹 또는 앱 인터페이스를 사용함
- **세 단계는 순환**
 - 운영 환경이나 사용자의 요구사항이 바뀌므로 꾸준히 업그레이드 필요
 - 예, 한식으로 국한했는데 세계 음식으로 업그레이드
- **예제는 공개 데이터를 사용하므로 모델 제작에 집중**

7.2 지능 에이전트 만들기

- 이 절에서는,
 - 6장에서 학습한 모델에 간단한 사용자 인터페이스를 붙여 지능 에이전트로 확장하는 실습

7.2.1 영상 바꾸니 담기

• 영상 바꾸니 담기 시나리오

- 영상을 분류하고 분류 결과에 따라 해당 폴더에 영상을 저장하는 서비스
- 구글 포토와 비슷

• 모델을 학습하고 저장하기

디프로이할 때 충분한 계산 자원을 사용하여 학습하는데 이 과정을 모방하기 위해 epochs를 늘려 학습

- CIFAR-10으로 컨볼루션 신경망을 학습

프로그램 7-1

CIFAR-10으로 컨볼루션 신경망을 학습하고 모델을 파일에 저장하기

```
01  
...  
31 } [프로그램 6-4(a)]의 1~31행  
32 hist=cnn.fit(x_train,y_train,batch_size=128,epochs=100,validation_data=(x_  
test,y_test),verbose=2)  
33  
34 } [프로그램 6-4(a)]의 34~58행  
...  
58  
59 cnn.save("my_cnn_for_deploy.h5")
```

디프로이에 쓰기 위해 학습된 모델을 저장

7.2.1 영상 바꾸니 담기

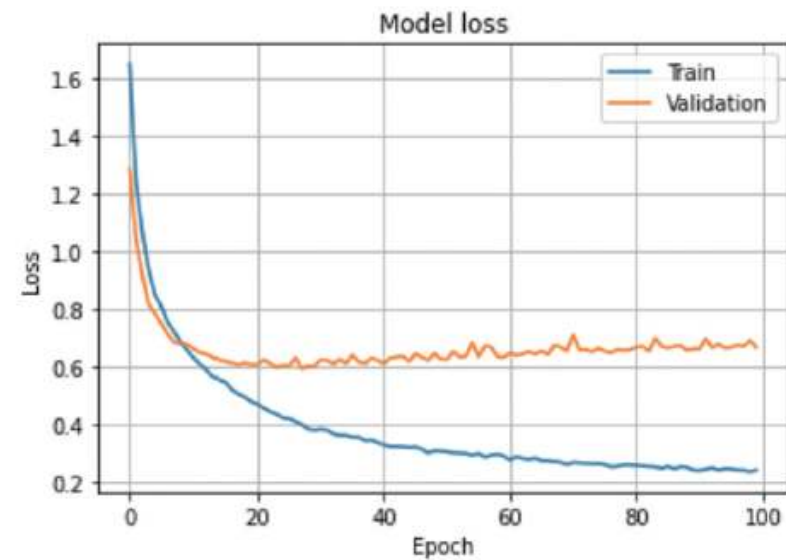
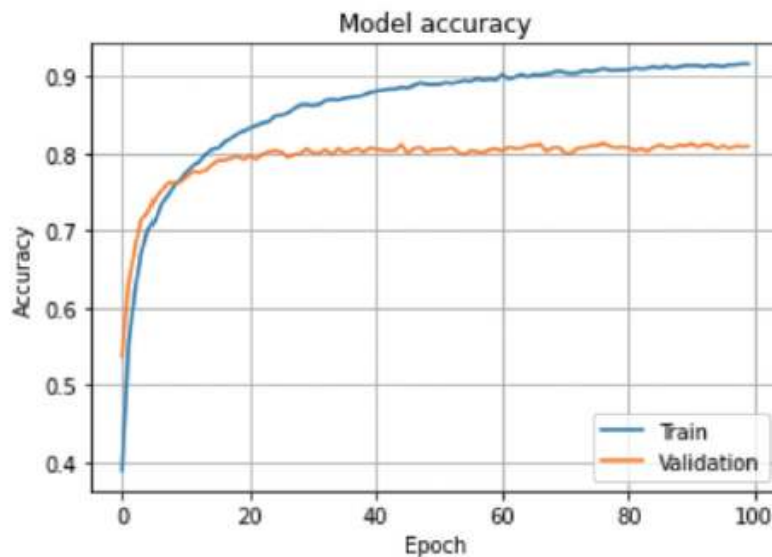
Train on 50000 samples, validate on 10000 samples

Epoch 1/30

50000/50000 - 126s - loss: 1.6821 - accuracy: 0.3850 - val_loss: 1.3316 - val_accuracy: 0.5239

83.21% 정확률의 모델 확보

정확률은 83.21%



7.2.1 영상 바꾸니 담기

- 영상을 인식하고 부류 확률을 막대로 그림
 - 테스트 영상에 대해 인식한 결과를 보여주는 인터페이스 제작
 - 여기서는 새로 획득한 영상을 폴더에 담아두고 인식하는 원시적인 인터페이스 제작
- 작업 폴더 아래에 test_images라는 폴더를 만들고 수집한 영상 저장([그림 7-4])

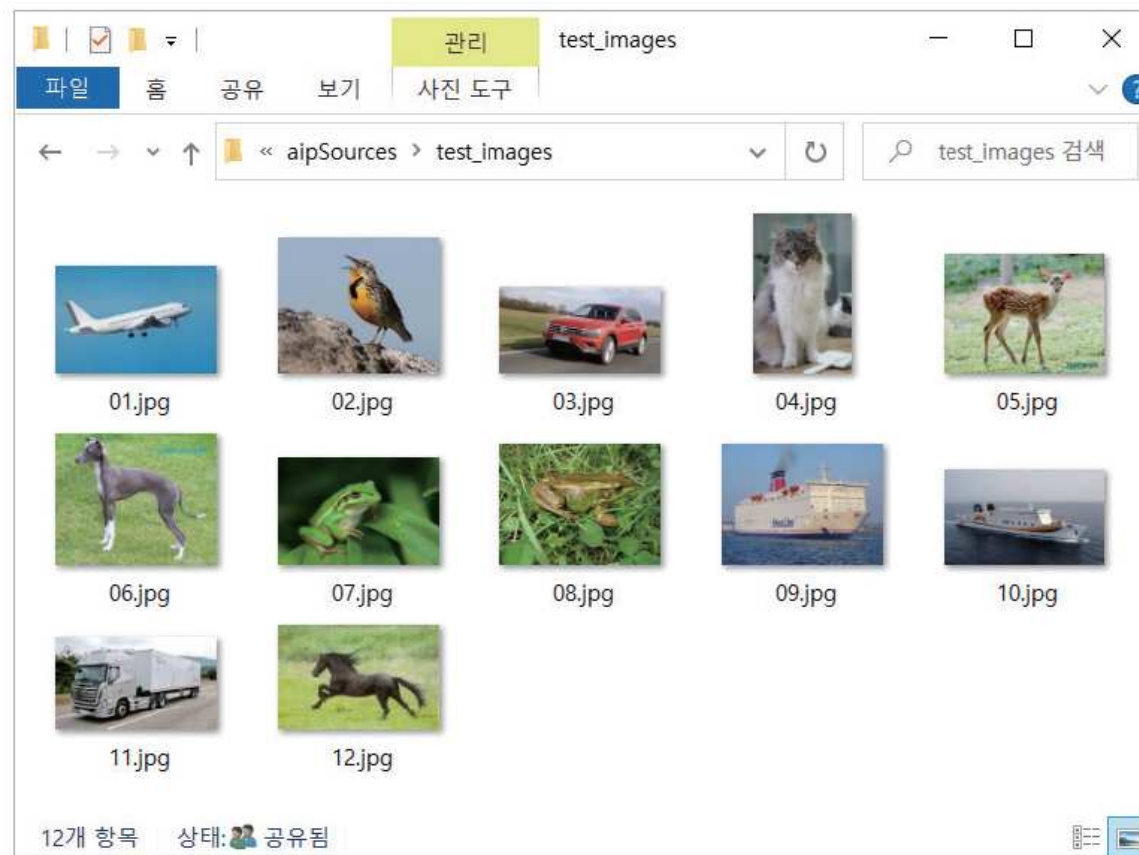


그림 7-4 웹에서 수집한 영상

7.2.1 영상 바꾸니 담기


• [프로그램 7-2]

프로그램 7-2

폴더에 담겨 있는 영상을 인식

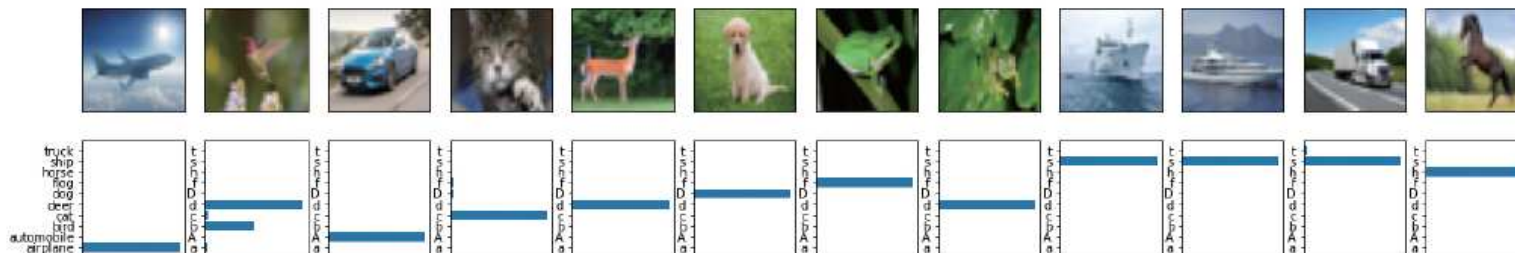
```
01 import numpy as np
02 import tensorflow as tf
03 from PIL import Image
04 import os
05
06 cnn=tf.keras.models.load_model("my_cnn_for_deploy.h5") # 학습된 모델 불러오기
07 class_names=['airplane','automobile','bird','cat','deer','dog','flog','horse','s
hip','truck'] # CIFAR-10의 부류 이름
08
09 x_test=[]
10 for filename in os.listdir("./test_images"): # 폴더에서 테스트 영상 읽기
11     if 'jpg' not in filename:
12         continue
13     img=Image.open("./test_images/"+filename)
14     x=np.asarray(img.resize([32,32]))/255.0
15     x_test.append(x)
16 x_test=np.asarray(x_test)
17
18 pred=cnn.predict(x_test) # 예측
```

[프로그램 7-1]에서 저장해둔 모델을 읽기



7.2.1 영상 바꾸니 담기

```
19
20 import matplotlib.pyplot as plt
21
22 n=len(x_test)
23 plt.figure(figsize=(18,4))
24
25 for i in range(n):
26     plt.subplot(2,n,i+1)
27     plt.imshow(x_test[i])
28     plt.xticks([]);plt.yticks([])
29     plt.subplot(2,n,n+i+1)
30     if i==0:
31         plt.barh(class_names,pred[i])
32     else:
33         plt.barh(['a','A','b','c','d','D','f','h','s','t'],pred[i])
34     plt.xticks([])
```



9개는 맞게 분류, 3개는 틀리게 분류

7.2.1 영상 바꾸니 담기

• 영상 바꾸니 담기 [프로그램 7-3]

프로그램 7-3

영상 바꾸니 담기 - 부류에 따라 영상을 폴더에 담기

```
01 import numpy as np
02 import tensorflow as tf
03 from PIL import Image
04 import os
05
06 cnn=tf.keras.models.load_model('my_cnn_for_deploy.h5') # 학습된 모델 불러오기
07 class_names=['airplane','automobile','bird','cat','deer','dog','flog','horse','s
hip','truck'] # CIFAR-10의 부류 이름
08
09 x_test=[]
10 img_orig=[]
11 fname=[]
12 for filename in os.listdir('./test_images'): # 폴더에서 테스트 영상 읽기
13     if 'jpg' not in filename:
14         continue
15     img=Image.open('./test_images/'+filename)
16     img_orig.append(img)
17     fname.append(filename)
18     x=np.asarray(img.resize([32,32]))/255.0
19     x_test.append(x)
20 x_test=np.asarray(x_test)
21
22 pred=cnn.predict(x_test) # 예측
23
```

[프로그램 7-1]에서 저장해둔 모델을 읽기

7.2.1 영상 바꾸니 담기

24~30행: test_images 밑에 class_buckets 폴더를 만들고
그 밑에 10 개 부류에 해당하는 폴더 만듦

```
24 os.chdir('./test_images')
25 if not os.path.isdir('class_buckets'):
26     os.mkdir('class_buckets')
27 os.chdir('class_buckets')
28 for i in range(len(class_names)):
29     if not os.path.isdir(class_names[i]):
30         os.mkdir(class_names[i])
31
32 for i in range(len(x_test)):
33     folder_name=class_names[np.argmax(pred[i])]
34     os.chdir(folder_name)
35     img_orig[i].save(fname[i])
36     os.chdir('..')
```

부류별로 폴더 만들기

인식 결과에 따라 폴더에 저장

32~36행: 인식 결과를 저장한 pred에 따라
해당 폴더에 원본 영상 저장

airplane	2020-11-13 오전 11:59	파일 폴더
automobile	2020-11-13 오전 11:59	파일 폴더
bird	2020-11-13 오전 11:59	파일 폴더
cat	2020-11-13 오전 11:59	파일 폴더
deer	2020-11-13 오전 11:59	파일 폴더
dog	2020-11-13 오전 11:59	파일 폴더
flog	2020-11-13 오전 11:59	파일 폴더
horse	2020-11-13 오전 11:59	파일 폴더
ship	2020-11-13 오전 11:59	파일 폴더
truck	2020-11-13 오전 11:59	파일 폴더

프로그램 실행 결과로
만들어진 10개 폴더

- 기능 확장하기

- [프로그램 7-3]은 구글 포토의 원시 버전(구글 포토는 촬영 장소와 사물에 따라 영상을 구분 등의 기능. 콜라주 또는 애니메이션 기능 등 제공)
- [프로그램 7-3]의 확장
 - CIFAR-10보다 큰 ImageNet 데이터셋으로 학습
 - 장소나 사물에 따라 분류하는 기능 추가
 - 웹 또는 앱 인터페이스 부착
 - 반려견과 함께 있는 사진만 골라주는 기능 등을 추가하여 경쟁력을 갖추

• [프로그램 7-4]

- tkinter 라이브러리를 활용해 사용자 인터페이스 부착
- gtts 라이브러리를 활용해 단어 발음을 들려줌

프로그램 7-4

다국어 단어 공부: tkinter를 이용한 대화형 인터페이스

```
01 import numpy as np
02 import tensorflow as tf
03 import tkinter as tk
04 from tkinter import filedialog
05 from PIL import Image, ImageTk
06 import winsound
07 from gtts import gTTS
08 import playsound
09 import os
10
11 cnn=tf.keras.models.load_model("my_cnn_for_deploy.h5") # 학습된 모델 불러오기
12
13 class_names_en=['airplane','automobile','bird','cat','deer','dog','frog','horse','ship','truck'] # CIFAR-10의 부류 이름(영어)
14 class_names_fr=['avion','voiture','oiseau','chatte','biche','chienne','grenouille','jument','navire','un camion'] # CIFAR-10의 부류 이름(프랑스어)
15 class_names_de=['Flugzeug','Automobil','Vogel','Katze','Hirsch','Hund','Frosch','Pferd','Schiff','LKW'] # CIFAR-10의 부류 이름(독일어)
16
```

[프로그램 7-1]에서 저장해둔 모델을 읽기



7.2.2 다국어 단어 공부 1 - tkinter를 이용한 대화형 인터페이스

```
17 class_id=0
18 tk_img=''
19
20 def process_image(): # 사용자가 선택한 영상을 인식하고 결과를 저장
21     global class_id, tk_img
22
23     fname=filedialog.askopenfilename() # 사용자가 폴더에서 영상을 선택하게 함
24     img=Image.open(fname)
25     tk_img=img.resize([128,128])
26     tk_img=ImageTk.PhotoImage(tk_img)
27     canvas.create_image((canvas.winfo_width()/2,canvas.winfo_height()/2),
28                         image=tk_img,anchor='center')
29
30     x_test=[]
31     x=np.asarray(img.resize([32,32]))/255.0
32     x_test.append(x)
33     x_test=np.asarray(x_test)
34     res=cnn.predict(x_test) # 영상 인식
35     class_id=np.argmax(res)
36     label_en['text']='영어: '+class_names_en[class_id] # 레이블 위젯용 텍스트 만들기
37     label_fr['text']='프랑스어: '+class_names_fr[class_id]
38     label_de['text']='독일어: '+class_names_de[class_id]
39     winsound.Beep(frequency=500,duration=250)
```

<영상 선택> 버튼을 클릭했을 때 실행되는 콜백 함수

7.2.2 다국어 단어 공부 1 - tkinter를 이용한 대화형 인터페이스

```
40 def tts_english():                # 영어로 들려주기
41     tts=gTTS(text=class_names_en[class_id],lang='en')
42     if os.path.isfile('word.mp3'): os.remove('word.mp3')
43     tts.save('word.mp3')
44     playsound.playsound('word.mp3',True)
45
46 def tts_french():                  # 프랑스어로 들려주기
47     tts=gTTS(text=class_names_fr[class_id],lang='fr')
48     if os.path.isfile('word.mp3'): os.remove('word.mp3')
49     tts.save('word.mp3')
50     playsound.playsound('word.mp3',True)
51
52 def tts_deutsch():                # 독일어로 들려주기
53     tts=gTTS(text=class_names_de[class_id],lang='de')
54     if os.path.isfile('word.mp3'): os.remove('word.mp3')
55     tts.save('word.mp3')
56     playsound.playsound('word.mp3',True)
57
58 def quit_program():               # 프로그램 종료
59     win.destroy()
60
```

tts_english, tts_french,
tts_deutsch는 <듣기>
버튼을 클릭했을 때
실행되는 콜백 함수

← <끝내기> 버튼을 클릭했을 때 실행되는 콜백 함수

7.2.2 다국어 단어 공부 1 - tkinter를 이용한 대화형 인터페이스

```
61 win=tk.Tk()
62 win.title('다국어 단어 공부')
63 win.geometry('512x500')
64
65 process_button=tk.Button(win,text='영상 선택',command=process_image) # 위젯 생성
66 quit_button=tk.Button(win,text='끝내기',command=quit_program)
67 canvas=tk.Canvas(win,width=256,height=256,bg='cyan',bd=4)
68 label_en=tk.Label(win,width=16,height=1,bg='yellow',bd=4,text='영어',anchor='w')
69 label_fr=tk.Label(win,width=16,height=1,bg='yellow',bd=4,text='프랑스어',anchor='w')
70 label_de=tk.Label(win,width=16,height=1,bg='yellow',bd=4,text='독일어',anchor='w')
71 tts_en=tk.Button(win,text='듣기',command=tts_english)
72 tts_fr=tk.Button(win,text='듣기',command=tts_french)
73 tts_de=tk.Button(win,text='듣기',command=tts_deutsch)
74
75 process_button.grid(row=0,column=0) # 위젯 배치
76 quit_button.grid(row=1,column=0)
77 canvas.grid(row=0,column=1)
78 label_en.grid(row=1,column=1,sticky='e')
79 label_fr.grid(row=2,column=1,sticky='e')
80 label_de.grid(row=3,column=1,sticky='e')
81 tts_en.grid(row=1,column=2,sticky='w')
82 tts_fr.grid(row=2,column=2,sticky='w')
83 tts_de.grid(row=3,column=2,sticky='w')
84
85 win.mainloop()
```

버튼을 클릭하면 호출되는 콜백 함수

위젯 배치

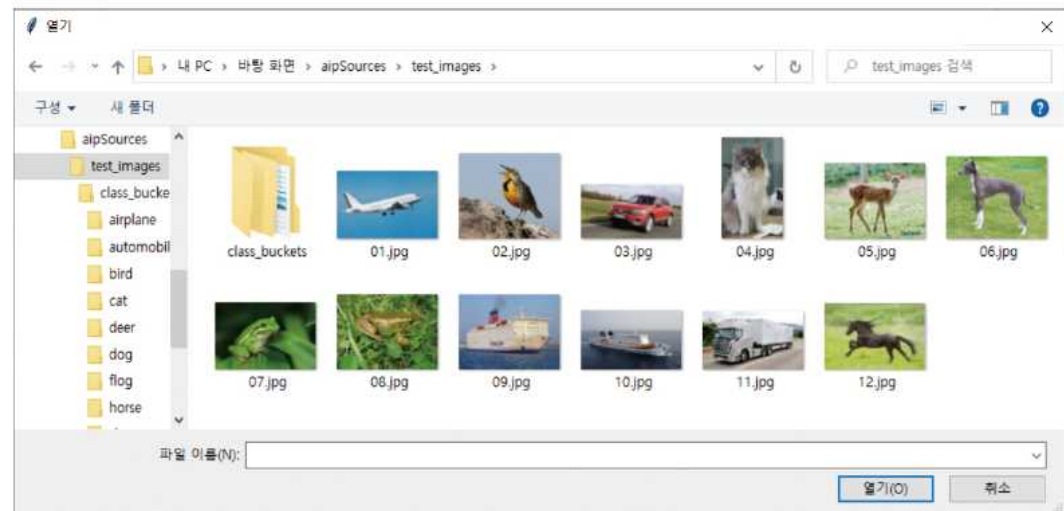
버튼을 클릭하면 호출되는 콜백 함수



7.2.2 다국어 단어 공부 1 - tkinter를 이용한 대화형 인터페이스



(a) 사용자 인터페이스



(b) 폴더에서 영상 파일 선택하기

그림 7-5 tkinter를 이용한 다국어 단어 공부 프로그램

• [프로그램 7-5]

- 웹캠에서 들어오는 영상을 인식하는 방식으로 인터페이스 개선(웹캠은 opencv로 제어)
- [프로그램 7-4]와 비슷한데 달라진 곳은 음영으로 표시

프로그램 7-5 opencv와 웹캠을 이용한 영상 인식

```
01 import numpy as np
02 import tensorflow as tf
03 import tkinter as tk
04 from tkinter import filedialog
05 from PIL import Image, ImageTk
06 import winsound
07 from gtts import gTTS
08 import playsound
09 import os
10 import cv2
11
12 cnn=tf.keras.models.load_model("my_cnn_for_deploy.h5")
13
14 class_names_en=['airplane','automobile','bird','cat','deer','dog','frog','horse',
15                'ship','truck']
16 class_names_fr=['avion','voiture','oiseau','chatte','biche','chienne',
17                'grenouille','jument','navire','un camion']
18 class_names_de=['Flugzeug','Automobil','Vogel','Katze','Hirsch','Hund','Frosch',
19                'Pferd','Schiff','LKW']
20
21 class_id=0
22 tk_img=''
```

7.2.3 다국어 단어 공부 2 – opencv와 웹캠을 이용한 대화형 인터페이스

```
20
21 def process_video():
22     global class_id, tk_img
23
24     video=cv2.VideoCapture(0)
25     while video.isOpened():
26         success,frame=video.read()
27         if success:
28             cv2.imshow('Camera',frame)
29             key=cv2.waitKey(1) & 0xFF
30             if key==27: ← 사용자가 ESC 키(아스키 27)를
31                 break                               누르는 순간의 영상 획득)
32
33     video.release()
34     cv2.destroyAllWindows()
35
36     frame=cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
37     img=Image.fromarray(frame)
38
39     tk_img=img.resize([128,128])
40     tk_img=ImageTk.PhotoImage(tk_img)
41     canvas.create_image((canvas.winfo_width()/2,canvas.winfo_height()/2),
42                        image=tk_img,anchor='center')
43
44     x_test=[]
45     x=np.asarray(img.resize([32,32]))/255.0
46     x_test.append(x)
47     x_test=np.asarray(x_test)
48     res=cnn.predict(x_test)
49     class_id=np.argmax(res)
50     label_en['text']='영어: '+class_names_en[class_id]
51     label_fr['text']='프랑스어: '+class_names_fr[class_id]
52     label_de['text']='독일어: '+class_names_de[class_id]
53     winsound.Beep(frequency=500,duration=250)
```

7.2.3 다국어 단어 공부 2 – opencv와 웹캠을 이용한 대화형 인터페이스

```
53
54 def tts_english():
55     tts=gTTS(text=class_names_en[class_id],lang='en')
56     if os.path.isfile('word.mp3'): os.remove('word.mp3')
57     tts.save('word.mp3')
58     playsound.playsound('word.mp3',True)
59
60 def tts_french():
61     tts=gTTS(text=class_names_fr[class_id],lang='fr')
62     if os.path.isfile('word.mp3'): os.remove('word.mp3')
63     tts.save('word.mp3')
64     playsound.playsound('word.mp3',True)
65
66 def tts_deutsch():
67     tts=gTTS(text=class_names_de[class_id],lang='de')
68     if os.path.isfile('word.mp3'): os.remove('word.mp3')
69     tts.save('word.mp3')
70     playsound.playsound('word.mp3',True)
71
72 def quit_program():
73     win.destroy()
74
```


7.2.3 다국어 단어 공부 2 – opencv와 웹캠을 이용한 대화형 인터페이스

```
75 win=tk.Tk()
76 win.title('다국어 단어 공부')
77 win.geometry('512x500')
78
79 process_button=tk.Button(win,text='비디오 선택',command=process_video)
80 quit_button=tk.Button(win,text='끝내기',command=quit_program)
81 canvas=tk.Canvas(win,width=256,height=256,bg='cyan',bd=4)
82 label_en=tk.Label(win,width=16,height=1,bg='yellow',bd=4,text='영어',anchor='w')
83 label_fr=tk.Label(win,width=16,height=1,bg='yellow',bd=4,text='프랑스어',
84 anchor='w')
85 label_de=tk.Label(win,width=16,height=1,bg='yellow',bd=4,text='독일어',anchor='w')
86 tts_en=tk.Button(win,text='듣기',command=tts_english)
87 tts_fr=tk.Button(win,text='듣기',command=tts_french)
88 tts_de=tk.Button(win,text='듣기',command=tts_deutsch)
89
90 process_button.grid(row=0,column=0)
91 quit_button.grid(row=1,column=0)
92 canvas.grid(row=0,column=1)
93 label_en.grid(row=1,column=1,sticky='e')
94 label_fr.grid(row=2,column=1,sticky='e')
95 label_de.grid(row=3,column=1,sticky='e')
96 tts_en.grid(row=1,column=2,sticky='w')
97 tts_fr.grid(row=2,column=2,sticky='w')
98 tts_de.grid(row=3,column=2,sticky='w')
99
100 win.mainloop()
```



7.2.3 다국어 단어 공부 2 – opencv와 웹캠을 이용한 대화형 인터페이스

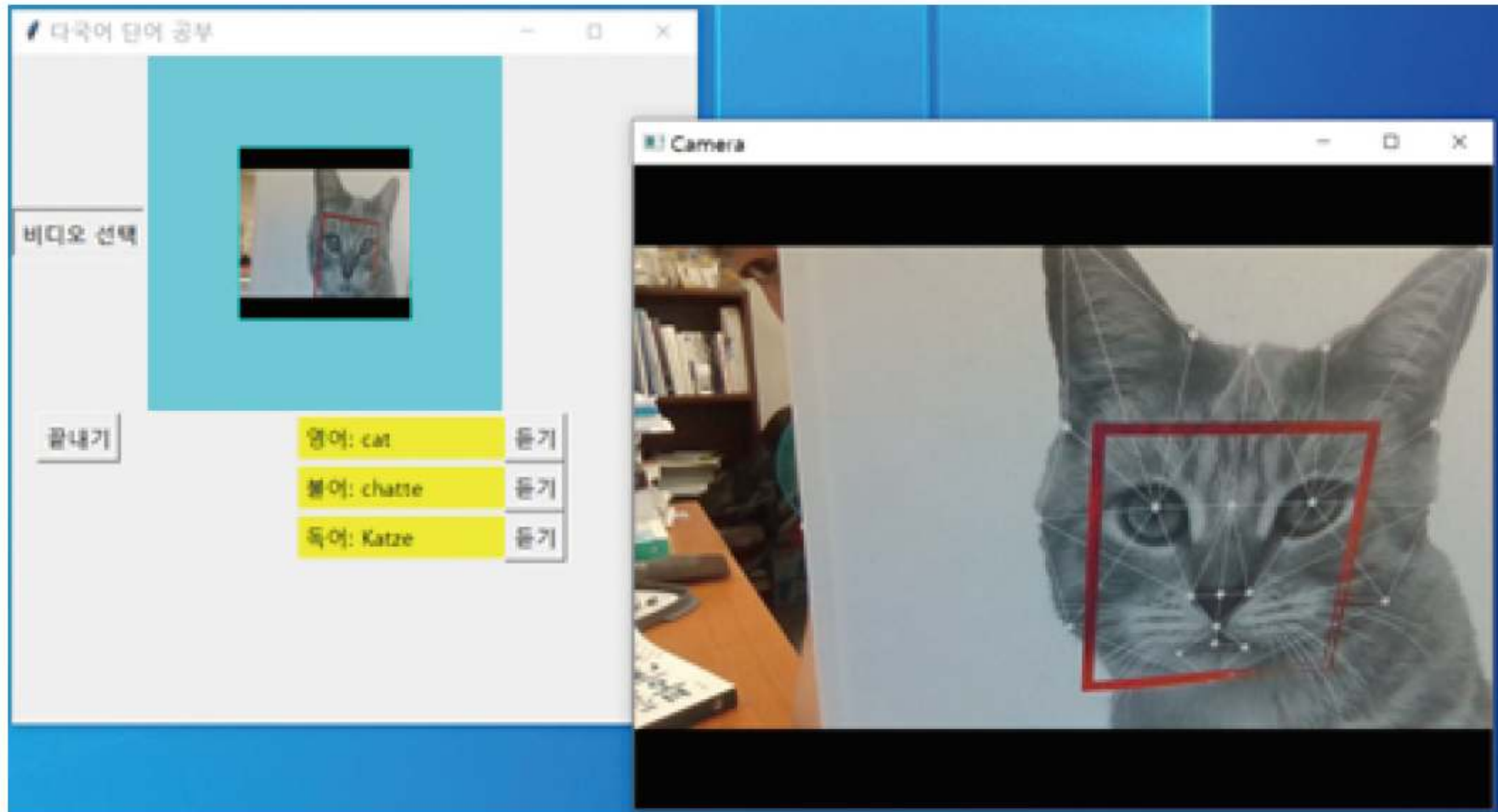


그림 7-6 opencv와 웹캠을 이용한 다국어 단어 공부 프로그램

• [프로그램 7-6]

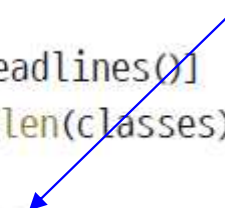
- 6.8.2절의 율로를 이용한 물체 검출을 확장하여 빈 집에 사람이 나타나면 경고음을 울려주는 프로그램
- [프로그램 7-5]와 [프로그램 6-12]를 결합하여 만듦

프로그램 7-6

웹캠을 이용해 침입자 검출하기

```
01 import numpy as np
02 import cv2
03 import winsound
04
05 classes = []
06 f=open('coco.names.txt','r')
07 classes=[line.strip() for line in f.readlines()]
08 colors=np.random.uniform(0,255,size=(len(classes),3))
09
10 yolo_model=cv2.dnn.readNet('yolov3.weights','yolov3.cfg') # 율로 읽어오기
11 layer_names=yolo_model.getLayerNames()
12 out_layers=[layer_names[i[0]-1] for i in yolo_model.getUnconnectedOutLayers()]
13
```

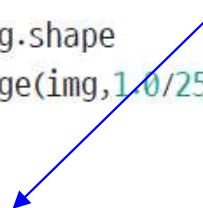
율로 모델을 읽어오기 ([프로그램 6-12] 참조)



7.2.4 집 지킴이 – YOLOv3를 이용한 침입자 검출

```
14 def process_video():                                     # 비디오에서 침입자 검출해 알리기
15     video=cv2.VideoCapture(0)
16     while video.isOpened():
17         success,img=video.read()
18         if success:
19             height,width,channels=img.shape
20             blob=cv2.dnn.blobFromImage(img,1.0/255,(448,448),(0,0,0),swapRB=True,crop=False)
21
22             yolo_model.setInput(blob)
23             output3=yolo_model.forward(out_layers)
24
25             class_ids,confidences,boxes=[],[],[]
26             for output in output3:
27                 for vec85 in output:
28                     scores=vec85[5:]
29                     class_id=np.argmax(scores)
30                     confidence=scores[class_id]
31                     if confidence>0.5:                     # 신뢰도가 50% 이상인 경우만 취함
32                         centerx,centery=int(vec85[0]*width),int(vec85[1]*height)
33                         # [0,1] 표현을 영상 크기로 변환
34                         w,h=int(vec85[2]*width),int(vec85[3]*height)
35                         x,y=int(centerx-w/2),int(centery-h/2)
36                         boxes.append([x,y,w,h])
37                         confidences.append(float(confidence))
38                         class_ids.append(class_id)
```

동영상에서 추출한 영상을 옰로에 입력하고 처리



7.2.4 집 지킴이 – YOLOv3를 이용한 침입자 검출

```
39         indexes=cv2.dnn.NMSBoxes(boxes,confidences,0.5,0.4)
40
41         for i in range(len(boxes)):
42             if i in indexes:
43                 x,y,w,h=boxes[i]
44                 text=str(classes[class_ids[i]]+'%.3f'%confidences[i]
45                 cv2.rectangle(img,(x,y),(x+w,y+h),colors[class_ids[i]],2)
46                 cv2.putText(img,text,(x,y+30),cv2.FONT_HERSHEY_PLAIN,2,
47                     colors[class_ids[i]],2)
48
49             cv2.imshow('Object detection',img)
50
51             if 0 in class_ids:      # 사람이 검출됨(0='person')
52                 print('사람이 나타났다!!!')
53                 winsound.Beep(frequency=2000,duration=500)
54
55             key=cv2.waitKey(1) & 0xFF
56             if key==27: break
57
58         video.release()
59         cv2.destroyAllWindows()
60     process_video()
```

7.2.4 집 지킴이 – YOLOv3를 이용한 침입자 검출



사람이 나타났다!!!

사람이 나타났다!!!

사람이 나타났다!!!

사람이 나타났다!!!

사람이 나타났다!!!

...

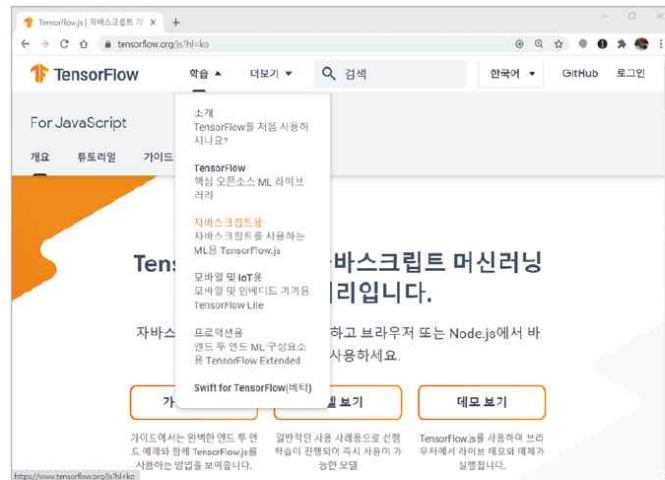


- 인터페이스 방식

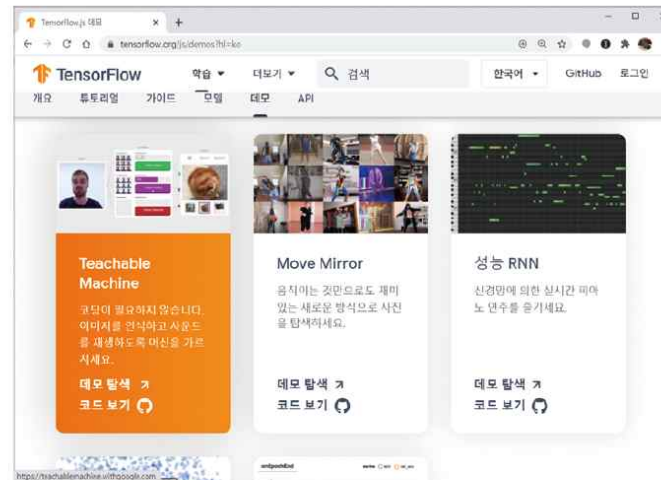
- 앞 절에서는 패키지 소프트웨어 방식의 인터페이스 사용
- 이 절에서는
 - 현대적인 웹 방식 또는 앱 방식에 대한 기초 지식을 간략히 소개
 - 로봇에 대한 간략한 소개

• 웹 프로그래밍

- HTML과 자바스크립트라는 웹 언어에 대한 지식 필요
 - HTML은 웹 브라우저에 표시되는 문서를 표현하는 언어
 - 자바스크립트는 쌍방향 상호작용을 해주는 언어
- 웹 서비스하려면 서버 필요
 - 자신의 컴퓨터를 웹 서버로 설정 또는 glitch.com과 같은 무료 서비스 활용
- 텐서플로 프로그램을 웹과 연결하려면 tensorflow.js를 사용하면 됨([그림 7-7])



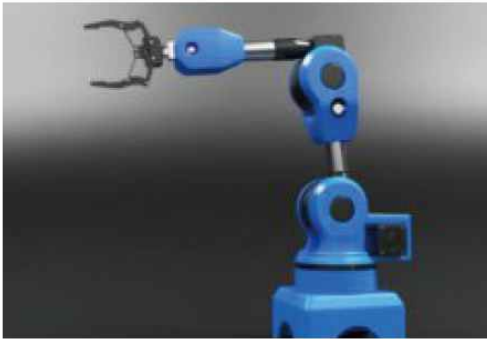
(a) tensorflow.js와 tensorflow.lite



(b) tensorflow.js의 데모 예제

그림 7-7 텐서플로 사이트

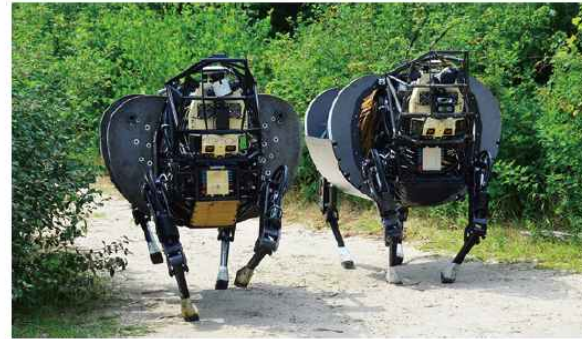
- 로봇은 가장 활동적인 지능 에이전트



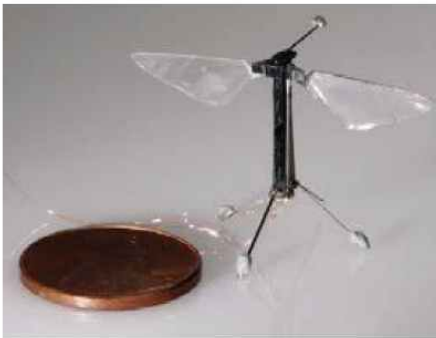
(a) 매니플레이터



(b) 청소용 로봇



(c) 빅독



(d) 곤충 로봇



(e) 휴머노이드



(f) 드론



(g) 자율주행차

그림 7-8 다양한 형태로 다양한 기능을 갖춘 로봇

- **지능 로봇의 수준(아래로 가면서 불확실성이 큰 과업 처리)**
 - 조립용 로봇: 이동 궤적을 정해주면 충실하게 단순 반복 작업 수행
 - 자세가 제멋대로인 부품을 집는 로봇: 컴퓨터 비전을 통한 인식 필요
 - 청소용 로봇: 카메라, 거리 센서, 적외선 센서 등이 필요하고 검출과 인식 알고리즘 필요. 경로 계획 알고리즘 필요
 - 모르는 건물에 진입해 작전 수행하는 빅독: 위치 파악과 지도 제작을 동시에 수행하는 SLAM_{simultaneous localization and mapping} 필요
 - 휴머노이드: 혼잡한 공간 이동, 물체 인식과 음성 인식. 상대의 표정을 인식
- **로봇 프로그래밍 교육**
 - 주로 단일 보드 컴퓨터(라즈베리파이, 젯슨, 오드로이드 등)를 탑재한 바퀴 달린 로봇 활용
 - 표준 미들웨어인 ROS_{robot operating system} 상에서 프로그래밍



그림 7-9 단일 보드 컴퓨터를 탑재한 바퀴 달린 로봇

- **지능 에이전트는 지식 베이스가 필요**

- 예) 다국어 단어 공부 제품([프로그램 7-5])은 동의어와 예문 등의 지식 베이스를 갖추면 제 모습이 됨

- **지식 표현과 추론**

신호등이 빨간 불이면 멈추고 파란 불이면 건넌다. 새는 난다. 공급이 부족하면 가격이 오른다. 바이러스에 감염되면 열이 난다. 홍길동의 직업은 교수이고 나이는 45세이며 취미는 목공이다. 고양이는 포유동물이다. 이런 지식을 어떻게 표현할 수 있을까?

- 인간 지능의 핵심이지만 인공지능에서는 미숙한 분야. 인간 지능과 인공지능의 격차가 가장 큰 연구 주제
- 이 절에서는 대표적인 방법을 간략히 소개

• 규칙 기반

- if-then 구절로 지식을 표현
- 1980대에 전문가 시스템을 만드는 중요 기술(Dendral, Mycin 등의 전문가 시스템)

• 프레임

- 슬롯-값 쌍으로 지식 표현(프로그래밍 언어의 구조체 또는 데이터베이스의 관계와 유사)
- KL-One 패키지: 개념으로 지식을 표현하고 연역적 분류기로 추론 수행

• 의미망

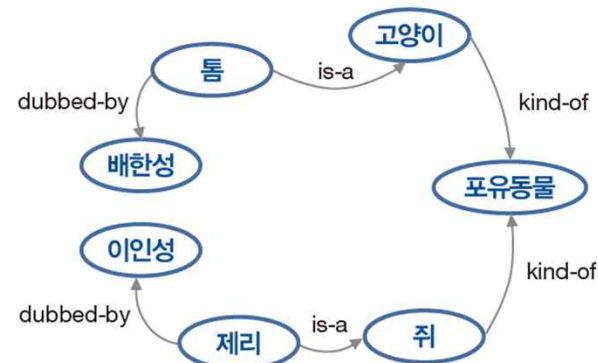
- 그래프로 지식을 표현(is-a와 kind-of 관계를 주로 사용)

If light is red, then stop.
If light is green, then walk.
If virus in body, then the body has fever.
If supply is short, then price rises.

(a) 규칙 기반

슬롯	값
이름	홍길동
직업	교수
나이	45
취미	목공

(b) 프레임



(c) 의미 망

• 온톨로지

- RDF(resource description framework)은 방대한 수의 트리플을 표현, 저장, 관리하는 기술
 - 주어, 술어, 목적어로 구성된 트리플 사용, 예)
- 온톨로지의 계층 구조
 - SPARQL은 RDF에서 동작하는 표준 질의 언어
 - OWL은 RDF에서 동작하는 대표적 온톨로지 언어
 - OWL을 구현한 소프트웨어로 Protégé와 지식 그래프
- 온톨로지로 구현한 지식 베이스
 - Cyc(최대 규모 지식 베이스 구축 프로젝트였으나 사실상 실패)
 - WordNet(인간 어휘 전반을 상위어, 하위어, 유사어로 연결한 지식 베이스)

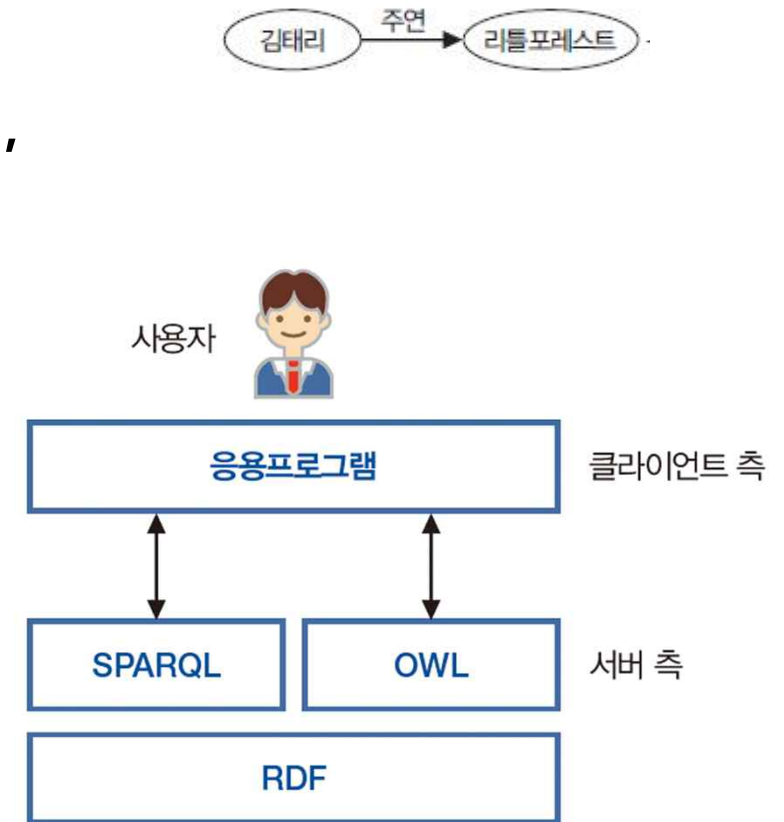


그림 7-11 온톨로지 계층 구조

7.4.2 지식 그래프

- 현재 가장 널리 쓰이는 지식 표현 방법

- 예) 구글의 지식 그래프는 2016년 기준으로 700억개의 사실을 포함한 방대한 지식 베이스

- 지식 그래프와 추론

- 예) 트리플로 표현한 김태리에 대한 지식 베이스인 [그림 7-12]
- 지식 그래프 구현 방법: 큐레이션(Cyc, WordNet), 협업(Wikidata, Freebase), 자연어 처리(Yago, DBpedia, Knowledge vault)

김태리	주연	리틀포레스트
김태리	역할	혜원
혜원	극중인물	리틀포레스트
임순례	감독	리틀포레스트
리틀포레스트	사례	영화
김태리	주연	미스터션사인
김태리	첫_주연작	미스터션사인
2018/7/7	개봉	미스터션사인
미스터션사인	사례	영화
김태리	조연	1987
1987	사례	영화
김태리	데뷔작	아가씨
2016/6/1	개봉	아가씨
무명배우_김태리	주연	문영
2017/1/12	개봉	문영
문영	사례	단편영화

그림 7-12 트리플로 표현한 지식 베이스

7.5 틈새 없는 협동과 능동성을 갖춘 지능 에이전트

- 틈새 없는 협동 seamless coordination
 - 컴퓨터 비전과 로봇 손이 협동하는 시각 서보잉 visual servoing (예, 수건 개는 로봇) – 기술 한계
 - 인간은 완벽하게 틈새 없는 협동을 함
- 능동성 proactive
 - 사람은 과업에서 과업으로 틈새 없이 전환(자신이 다음에 벌일 일을 주도적으로 구상)
 - 로봇은 매우 제한적으로 능동성 발휘(어떻게 빨래 개는 과업에서 밥 먹는 과업으로 자연스럽게 전환할 수 있을까?)



그림 7-14 현재 로봇 기술로는 처리하기 어려운 빨래의 다양성



(a) 틈새 없는 협동을 제한적으로 갖춘 로봇

그림 7-13 수건 개는 로봇과 사람



(b) 틈새 없는 협동과 능동성을 갖춘 사람