

# 시각장애인을 위한 딥러닝 이미지 분류 기반 편의점 식품 구분 시스템

Convenience store food classification system based on  
deep learning image classification for the blinder

---

발표자 : 순천향대학교 컴퓨터공학과 20174068 김경환

지도 교수 : 홍 인 식 교수님

# 목 차

챕터 1. 주제 선정

챕터 2. 관련 연구

챕터 3. 시스템 설계 및 제작

챕터 4. 실험 및 결과







# 챕터 1

## 주제 선정

# 주제 선정 이유

매년 최저임금 인상안이 결정될 때마다 매장 점주들의 반발이 큰 가운데 무인 편의점 수가 매년 빠르게 늘고 있는 것으로 나타났다. 일반인의 경우 무인 매장을 이용하는데 큰 어려움이 없겠지만 시각장애를 앓고 있는 사람의 경우, 직원도 없어 혼자서 물품을 구매하지 못할 것 같다는 생각이 들어, 사진을 촬영하여 물품을 구분하고 해당 물품을 음성으로 알려주는 시스템이 있으면 좋겠다는 생각에 주제를 선정하였다.

편의점 4사 무인 점포 수 현황

		2019년	2020년	2021년	2022년*
	CU	90개	200개	300개	400개
	GS25	16개	140개	565개	723개
	세븐일레븐	17개	46개	210개	330개
	이마트24	85개	113개	1050개	1330개
	총합	208개	499개	2125개	2783개

\*6월 말 기준



# 챕터 2

## 관련 연구

# 시각장애인 편의점 이용시 문제점

먼저 시각장애인이 편의점을 이용할 때 어떠한 문제점이 있는지 분석하였다.

## 문제점

- 1 맥주나 주스 같은 음료는 같은 모양의 캔인 경우 '음료'나 '탄산' 이라고만 점자로 표기되어 있어서 어떤 음료인지 구분이 되지 않는다.
- 2 통조림이나 과자 같은 경우 점자 표기가 되어 있지 않다.
- 3 점자가 있는 식품일 경우 점자 간격이 너무 크거나 작아서 읽기 힘든 경우가 있다.

# 관련 연구

---

본 연구와 비슷한 연구가 있는 조사 하였다. 현재 스마트폰으로 사진을 촬영하여 식품의 글자를 읽어주는 어플이 있다. 하지만 시각장애인 스마트폰으로 어플을 정확히 실행시킬 수 있을지 의문이 들었고, 사진을 촬영하여도 글자가 정확히 나오지 않으면 제대로된 식품 구분이 안될 수도 있다고 생각하여, 하드웨어를 사용하여 편리하게 버튼만 눌러 촬영하면 식품을 구분하여 음성으로 알려주는 기기를 개발한다.



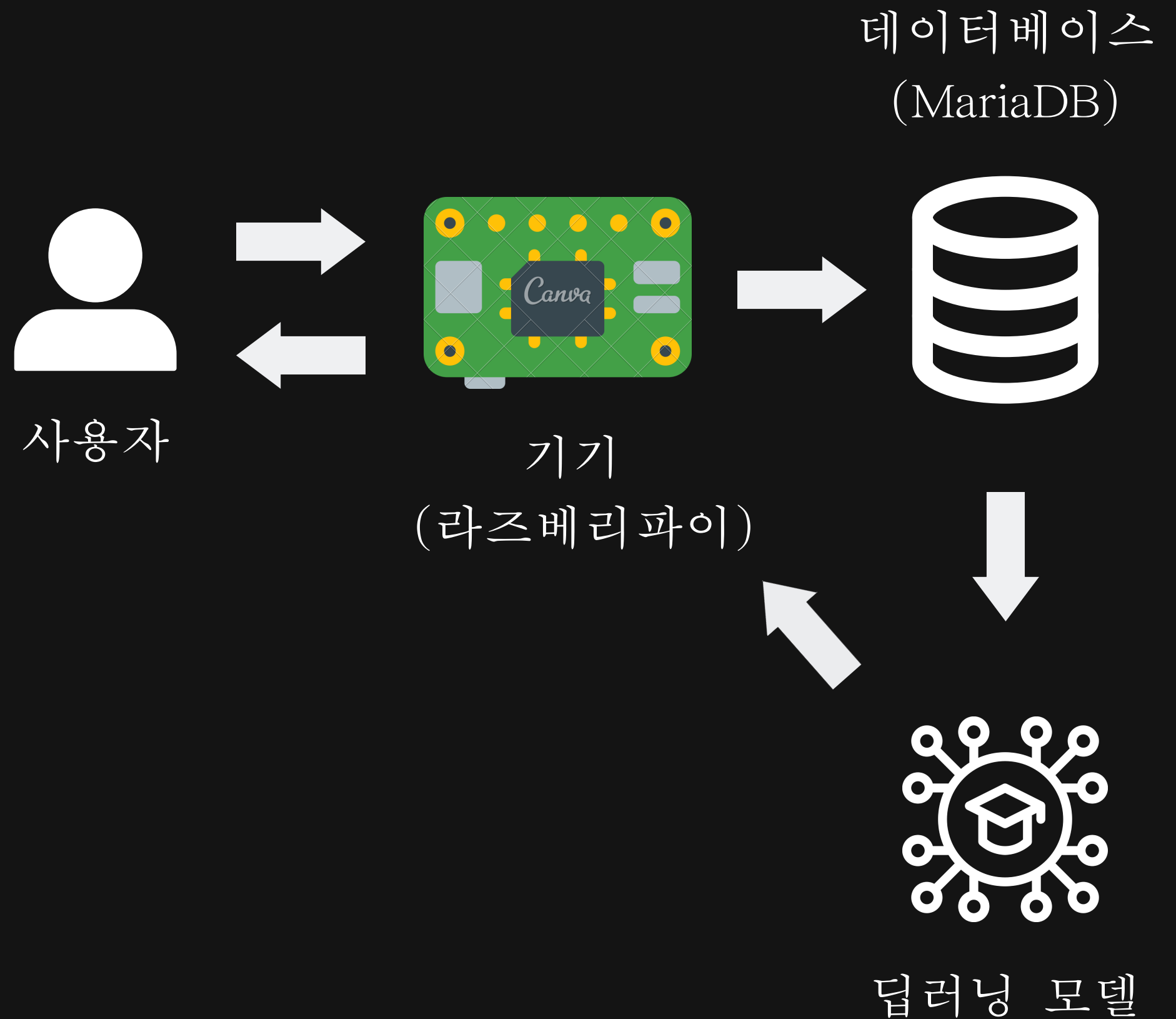
# 챕터 3

## 시스템 설계 및 제작

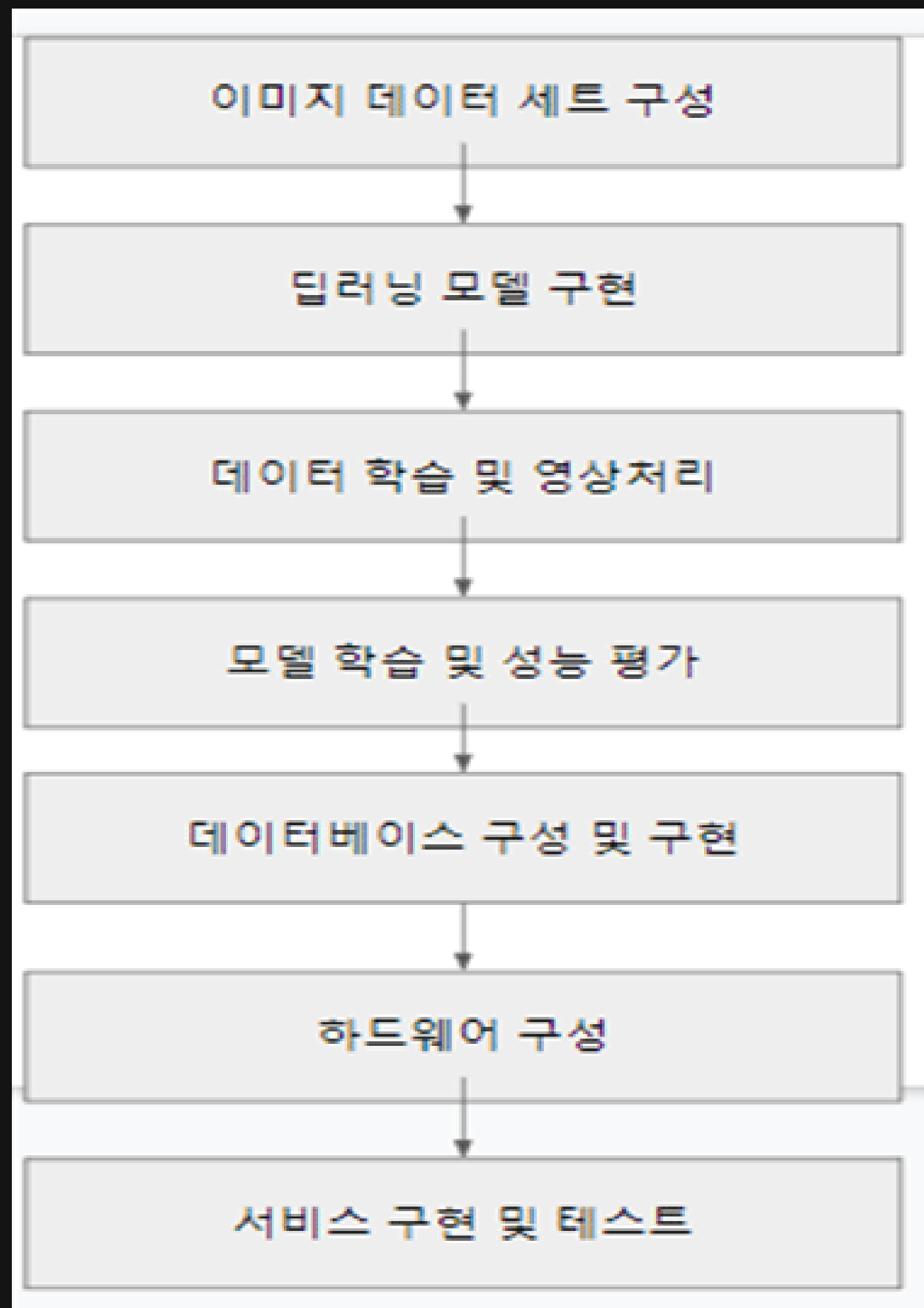


# 서비스 구성 과정

사용자가 기기를 사용하여 제품을 촬영하였을 경우, 해당 이미지 데이터를 DB에 저장 후 딥러닝 모델을 사용하여 식품을 분류하고 분류된 결과를 기기에 다시 전송하여 음성을 출력한다.



# 시스템 개발 과정



# 이미지 데이터 세트 구성

본 연구에서 사용한 데이터 세트는 편의점에서 판매하고 있는 제품을 임의로 4개 선정하고, 휴대폰을 사용해 해당 제품을 직접 촬영하여 구성하였다.

구분	식품 이름	이미지 데이터 개수
1	포카칩	1116개
2	새우깡	1080개
3	콜라	1116개
4	스프라이트	1116개

# 딥러닝 모델 구현

본 연구에서 딥러닝 모델 구현은 CNN 방법을 사용하여 구현하였고, 모델은 convolution layer 4개, pooling layer 4개, flatten layer 1개, fully connected layer 2개로 구성되어 있다.

Model: "sequential\_6"

Layer (type)	Output Shape	Param #
conv2d_24 (Conv2D)	(None, 150, 150, 32)	896
max_pooling2d_24 (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_25 (Conv2D)	(None, 75, 75, 32)	9248
max_pooling2d_25 (MaxPooling2D)	(None, 37, 37, 32)	0
conv2d_26 (Conv2D)	(None, 35, 35, 64)	18496
max_pooling2d_26 (MaxPooling2D)	(None, 17, 17, 64)	0
conv2d_27 (Conv2D)	(None, 15, 15, 64)	36928
max_pooling2d_27 (MaxPooling2D)	(None, 7, 7, 64)	0
flatten_5 (Flatten)	(None, 3136)	0
dense_10 (Dense)	(None, 512)	1606144
dense_11 (Dense)	(None, 4)	2052

Total params: 1,673,764  
Trainable params: 1,673,764  
Non-trainable params: 0

# 딥러닝 모델 컴파일

```
# Compiling the CNN
model.compile(optimizer = 'adam',
              loss = 'categorical_crossentropy',
              metrics = ['accuracy'])
```

본 연구에서 딥러닝 모델 컴파일은 4개의 카테고리를 분류하기에 categorical\_crossentropy 손실 함수를 사용하였고, 경사 하강법으로 adam 경사 하강법을 사용했다.

# 딥러닝 모델 학습 진행

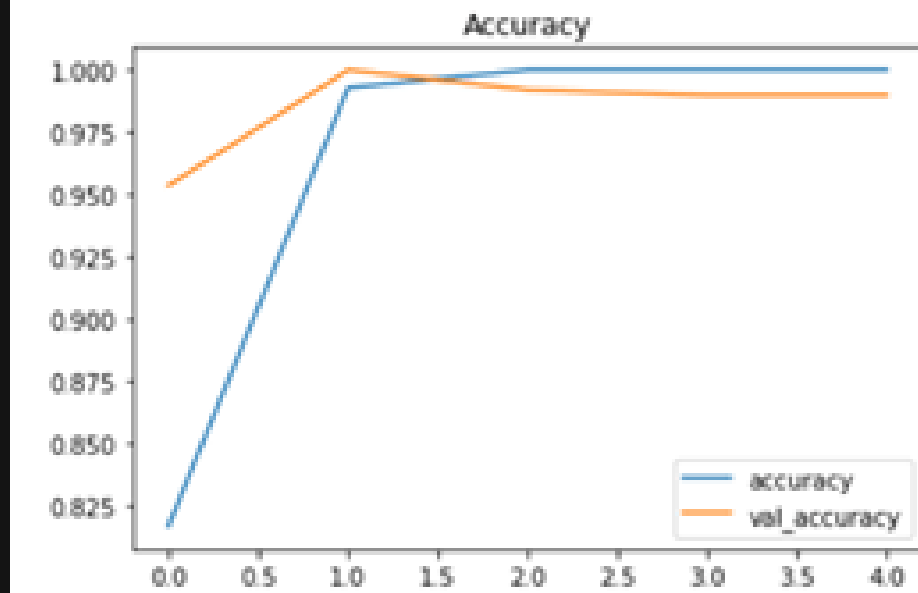
```
Epoch 1/5  
111/111 [=====] - 116s 1s/step - loss: 0.4269 - accuracy: 0.8171 - val_loss: 0.2081 - val_accuracy: 0.9537  
Epoch 2/5  
111/111 [=====] - 113s 1s/step - loss: 0.0299 - accuracy: 0.9929 - val_loss: 0.0025 - val_accuracy: 1.0000  
Epoch 3/5  
111/111 [=====] - 115s 1s/step - loss: 5.3262e-04 - accuracy: 1.0000 - val_loss: 0.0114 - val_accuracy: 0.9921  
Epoch 4/5  
111/111 [=====] - 115s 1s/step - loss: 1.1684e-04 - accuracy: 1.0000 - val_loss: 0.0127 - val_accuracy: 0.9898  
Epoch 5/5  
111/111 [=====] - 117s 1s/step - loss: 5.4978e-05 - accuracy: 1.0000 - val_loss: 0.0122 - val_accuracy: 0.9898
```

본 연구에서 딥러닝 모델 학습은 총 데이터 4428개를 학습  
데이터 3543개와 테스트 데이터 885개로 나누어 5번 학습했  
다.

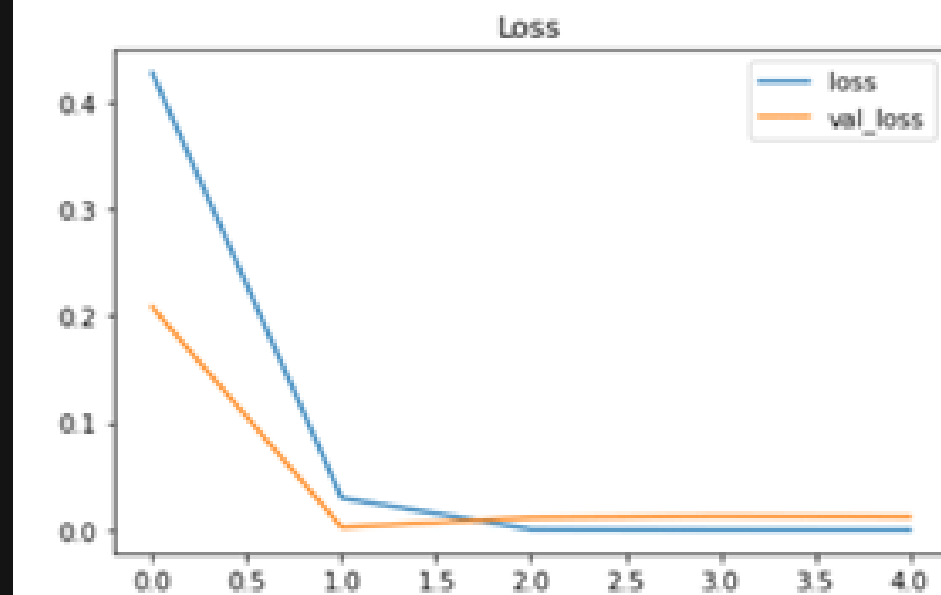
# 모델 학습 정확도 및 손실도 시각화

matplotlib 함수를 사용하여 모델의 학습 진행 상황을 시각화 하였을 때 학습이 진행 될수록 정확도가 상승하고, 손실도는 감소하는 것을 확인할 수 있다.

```
pd.DataFrame(history.history)[['accuracy', 'val_accuracy']].plot()  
plt.title("Accuracy")  
plt.show()
```



```
pd.DataFrame(history.history)[['loss', 'val_loss']].plot()  
plt.title("Loss")  
plt.show()
```



# 딥러닝 모델 성능 평가

식품 이름	테스트 데이터 개수	판별 성공	판별 실패	총 데이터 개수
포카칩	30	30	0	120
새우깡	30	30	0	
콜라	30	26	4	
스트라이트	30	30	0	

모델을 사용하여 이미지 데이터 120개를 촬영하여 출력 하였을 때 총 데이터 120개 중 116개 정확히 식품 이름으로 출력했다.



# 데이터 베이스 구현

```
MariaDB [image_upload]> show tables;
+-----+
| Tables_in_image_upload |
+-----+
| images                  |
+-----+
1 row in set (0.001 sec)

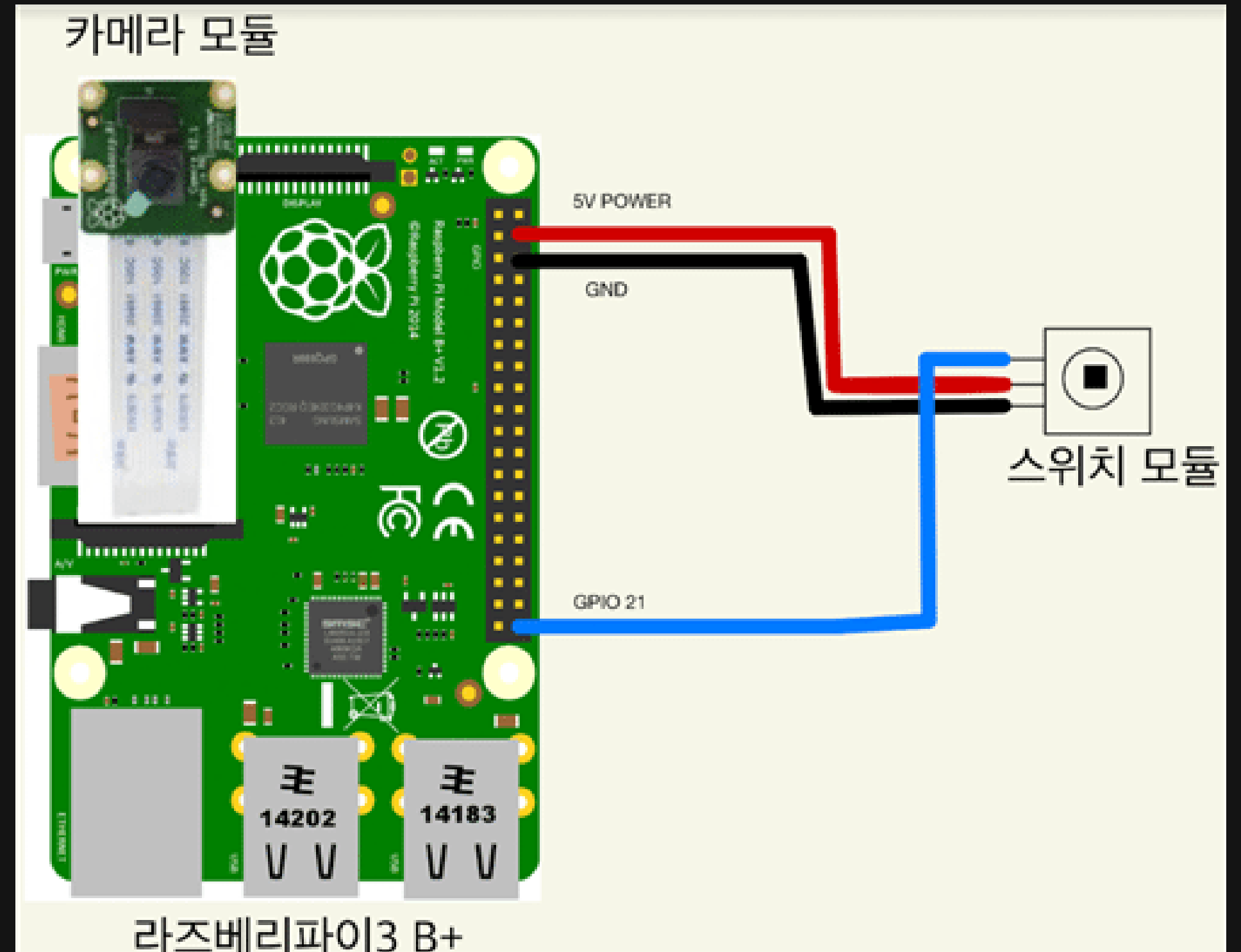
MariaDB [image_upload]> _
```

데이터베이스는 MariaDB를 사용하여 구현하였다. 데이터베이스 이름을 image\_upload로 지정하여 생성하였고, 이 데이터베이스에 images라는 테이블을 생성했다. 기기로 제품을 촬영 하였을 경우 그 이미지 데이터를 저장하기 위해 데이터 유형으로 BLOB을 사용했다.

저장된 이미지 데이터는 추후에 모델의 정확도를 더 높이거나, 이와 관련된 연구를 진행할 때 사용된다.

# 하드웨어 구성

구분	이름
1	라즈베리파이3 B+
2	카메라 모듈
3	3핀 스위치 모듈



하드웨어 설계도

# 하드웨어 구성

구분	이름
1	라즈베리파이3 B+
2	카메라 모듈
3	3핀 스위치 모듈



하드웨어 설계도  
(실제 모습)



# 챕터 4

## 실험 및 결과

# 실험 방식

---

본 연구의 실험은 20대 남성에게 안대를 착용하여 시각을 차단한 뒤 기기를 들고 제품의 사진을 촬영하여 실험했다.



# 실험 결과

실험자가 기기를 들고 포카칩, 콜라, 스트라이트 순서로 촬영하였고, 해당 사진의 식품이름을 정확하게 출력하고 있는 것을 확인할 수 있다.



2img



3img



4img

```
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\dkslf\deep-learning\project_clear\DB-OUT-Deep.py ===
1/1 [=====] - ETA: 0s
식품 이름 = 포카칩 (봉지 과자)
1/1 [=====] - ETA: 0s
식품 이름 = 콜라 (캔 음료)
1/1 [=====] - ETA: 0s
식품 이름 = 스트라이트 (캔 음료)
|
```

# 실험 결과

또한, 이미지를 촬영하여 그 이미지 데이터를 MariaDB에 저장되는 것도 확인할 수 있다.

```
MySQL Client (MariaDB 10.6 (x64)) - "C:\Program Files\MariaDB 10.6\bin\mysql.exe" "--  
MariaDB [project]> use image_upload  
Database changed  
MariaDB [image_upload]> delete from images;  
Query OK, 4 rows affected (0.016 sec)  
  
MariaDB [image_upload]> select count(image_data) from images;  
+-----+  
| count(image_data) |  
+-----+  
| 0 |  
+-----+  
1 row in set (0.000 sec)  
  
MariaDB [image_upload]> select count(image_data) from images;  
+-----+  
| count(image_data) |  
+-----+  
| 1 |  
+-----+  
1 row in set (0.001 sec)  
  
MariaDB [image_upload]> select count(image_data) from images;  
+-----+  
| count(image_data) |  
+-----+  
| 4 |  
+-----+  
1 row in set (0.004 sec)  
  
MariaDB [image_upload]>
```

Thank  
you

