

종합설계 2

최종 진행상황 정리 (11/17)

컴퓨터공학과

2019305065

전영민

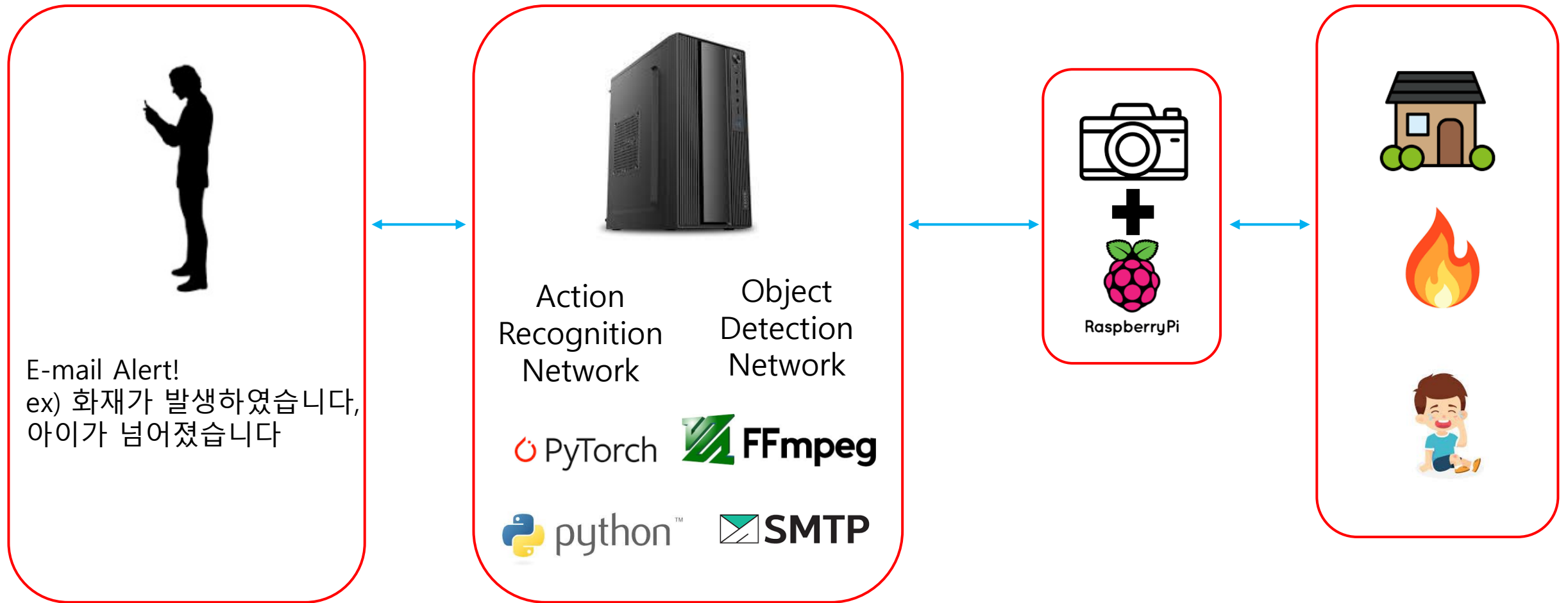
목차

- 졸업작품 개요
- System Architecture
- 지난 보고 후 진행상황
- 개발 진행상황

졸업작품 개요

- 작품명: 지능형 안전관리 시스템
- 작품 설명: In-door 환경에 설치할 수 있는 CCTV 혹은 홈캠으로 수집한 영상으로부터 물체 인식, 사람 동작 추정을 통해 화재, 넘어짐 등의 위험 상황을 감지하여 사용자에게 위험상황이 발생했음을 알리는 지능형 안전관리 시스템
- 세부 요소 설명: 영상 수집 장치(라즈베리파이 + 카메라), 인공지능 추론 환경(메인 프레임 컴퓨터)

System Architecture




지난 보고 후 진행상황

- 지난번 코멘트: 11월 8일 3/3회차 제출에는 완성 버전으로 제출할 것
- 현재 개발 진행상황: 전체 기능 구현 완료 및 기능 모듈 병합 완료
- 세부 개발 사항:
 1. Human Pose Estimation Network 설계, 학습 및 추론 모듈 구현
 2. 임베디드 카메라 통신 모듈, 화재감지 모듈, 낙상감지 모듈, 알림전송 모듈 병합

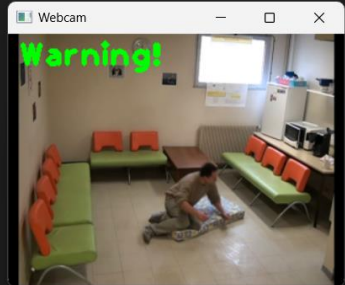
지난 보고 후 진행상황

```
fall_detection.py M X
GitHub > 2024-Graduation-Project > Sources > 3_fall_detection > fall_detection.py > ...
1 import cv2
2 import mediapipe as mp
3 import numpy as np
4 import torch.nn as nn
5 import torch
6 from time import time
7 device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
8
9
10 class GRU(nn.Module):
11     def __init__(self, input_size, hidden_size, sequence_length, num_layers, device):
12         super(GRU, self).__init__()
13         self.device = device
14         self.hidden_size = hidden_size
15         self.num_layers = num_layers
16         self.gru = nn.GRU(input_size, hidden_size, num_layers, batch_first=True)
17         self.fc = nn.Linear(hidden_size*sequence_length, 1)
18
19     def forward(self, x):
20         h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(self.device)
21         out, _ = self.gru(x, h0)
22         out = out.reshape(out.shape[0], -1)
23         out = self.fc(out)
24         return out
25
26
27 sequence_length = 30
28 input_size = 132
29 num_layers = 2
30 hidden_size = 50
31 model = GRU(input_size = input_size, hidden_size = hidden_size, sequence_length = sequence_length, num_layers = num_layers, device = device).to(device)
32 model.load_state_dict(torch.load(f=r'C:/Users/mkjsy/Desktop/YM/Source Code/VSCode/GitHub/2024-Graduation-Project/Sources/Data/Weights/weight.pth'))
33
34 mp_pose = mp.solutions.pose
35 pose = mp_pose.Pose(static_image_mode = True, min_detection_confidence = 0.1, model_complexity = 2)
36 mp_drawing = mp.solutions.drawing_utils
37 mp_drawing_styles = mp.solutions.drawing_styles
38 # cap = cv2.VideoCapture(0)
39 cap = cv2.VideoCapture(r'C:/Users/mkjsy/Desktop/YM/Source Code/VSCode/GitHub/2024-Graduation-Project/Sources/Data/video (3).avi')
40
41 queue_size = 30
42 queue = []
43 flag = 0
44
45 tensor([[ -15.1415]], device='cuda:0', grad_fn=<AddmmBackward0>) 0
46 tensor([[ -13.8067]], device='cuda:0', grad_fn=<AddmmBackward0>) 0
47 tensor([[ -18.4423]], device='cuda:0', grad_fn=<AddmmBackward0>) 0
48 tensor([[ -15.4869]], device='cuda:0', grad_fn=<AddmmBackward0>) 0
49 tensor([[ -18.1671]], device='cuda:0', grad_fn=<AddmmBackward0>) 0
50 tensor([[ -15.7284]], device='cuda:0', grad_fn=<AddmmBackward0>) 0
51 tensor([[ -14.6930]], device='cuda:0', grad_fn=<AddmmBackward0>) 0
52 []
```



지난 보고 후 진행상황

```
fall_detection.py M X
GitHub > 2024-Graduation-Project > Sources > 3_fall_detection > fall_detection.py > ...
1 import cv2
2 import mediapipe as mp
3 import numpy as np
4 import torch.nn as nn
5 import torch
6 from time import time
7 device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
8
9
10 class GRU(nn.Module):
11     def __init__(self, input_size, hidden_size, sequence_length, num_layers, device):
12         super(GRU, self).__init__()
13         self.device = device
14         self.hidden_size = hidden_size
15         self.num_layers = num_layers
16         self.gru = nn.GRU(input_size, hidden_size, num_layers, batch_first=True)
17         self.fc = nn.Linear(hidden_size*sequence_length, 1)
18
19     def forward(self, x):
20         h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(self.device)
21         out, _ = self.gru(x, h0)
22         out = out.reshape(out.shape[0], -1)
23         out = self.fc(out)
24         return out
25
26
27 sequence_length = 30
28 input_size = 132
29 num_layers = 2
30 hidden_size = 50
31 model = GRU(input_size = input_size, hidden_size = hidden_size, sequence_length = sequence_length, num_layers = num_layers, device = device).to(device)
32 model.load_state_dict(torch.load(f=r'C:/Users/mkjsy/Desktop/YM/Source Code/VSCode/GitHub/2024-Graduation-Project/Sources/Data/Weights/weight.pth'))
33
34 mp_pose = mp.solutions.pose
35 pose = mp_pose.Pose(static_image_mode = True, min_detection_confidence = 0.1, model_complexity = 2)
36 mp_drawing = mp.solutions.pose.drawing_utils
37 mp_drawing_styles = mp.solutions.pose.drawing_styles
38 # cap = cv2.VideoCapture(0)
39 cap = cv2.VideoCapture(r'C:/Users/mkjsy/Desktop/YM/Source Code/VSCode/GitHub/2024-Graduation-Project/Sources/Data/video (3).avi')
40
41 queue_size = 30
42 queue = []
43 flag = 0
44
45 문제 출력 디버그 콘솔 터미널 포트
46
47 tensor([[12.8631]], device='cuda:0', grad_fn=<AddmmBackward0>) 1
48 tensor([[1.6656]], device='cuda:0', grad_fn=<AddmmBackward0>) 0
49 tensor([[6.1074]], device='cuda:0', grad_fn=<AddmmBackward0>) 1
50 tensor([[9.8867]], device='cuda:0', grad_fn=<AddmmBackward0>) 1
51 tensor([[8.7831]], device='cuda:0', grad_fn=<AddmmBackward0>) 1
52 tensor([[9.6705]], device='cuda:0', grad_fn=<AddmmBackward0>) 1
53 tensor([[17.2386]], device='cuda:0', grad_fn=<AddmmBackward0>) 1
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```



개발 진행상황

- 라즈베리파이-카메라 모듈 연동: 라즈베리파이 4 보드와 usb 웹캠 간의 연동 확인(100%)
- 라즈베리파이-메인 프레임 컴퓨터 통신 구축: 구현(100%)
- Human Action Recognition: GRU 모델을 사용하여 사람의 넘어짐 감지 기능 구현(100%)
- Object Detection: YOLOv5 모델을 사용하여 Fire Detection 기능 구현(100%)
- 위험상황 알림: smtp python 라이브러리인 smtplib을 사용한 이메일 전송 기능 구현(100%)
- 각 기능별 모듈 병합(100%)