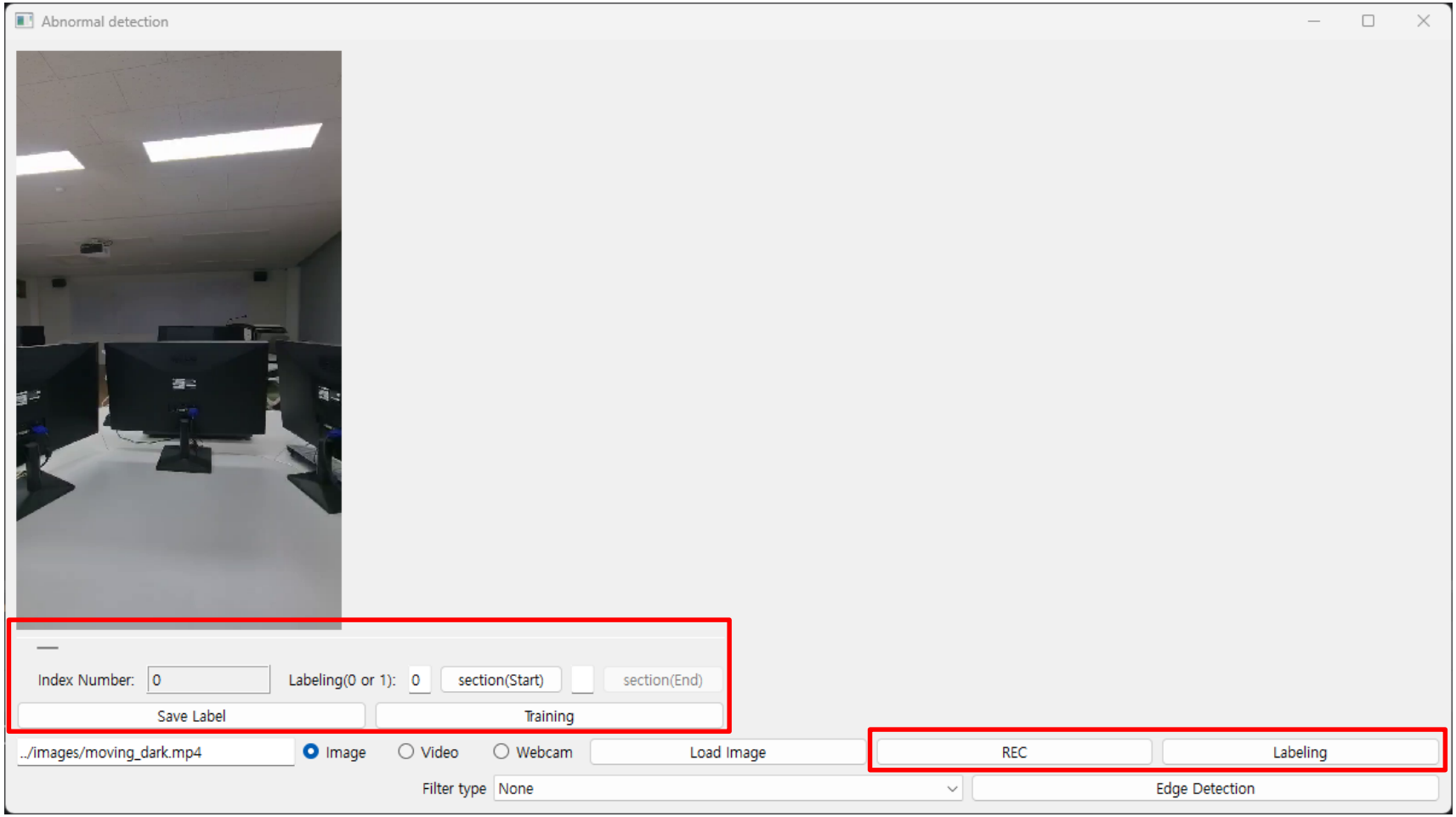
이상 탐지를 위한 임계값 학습

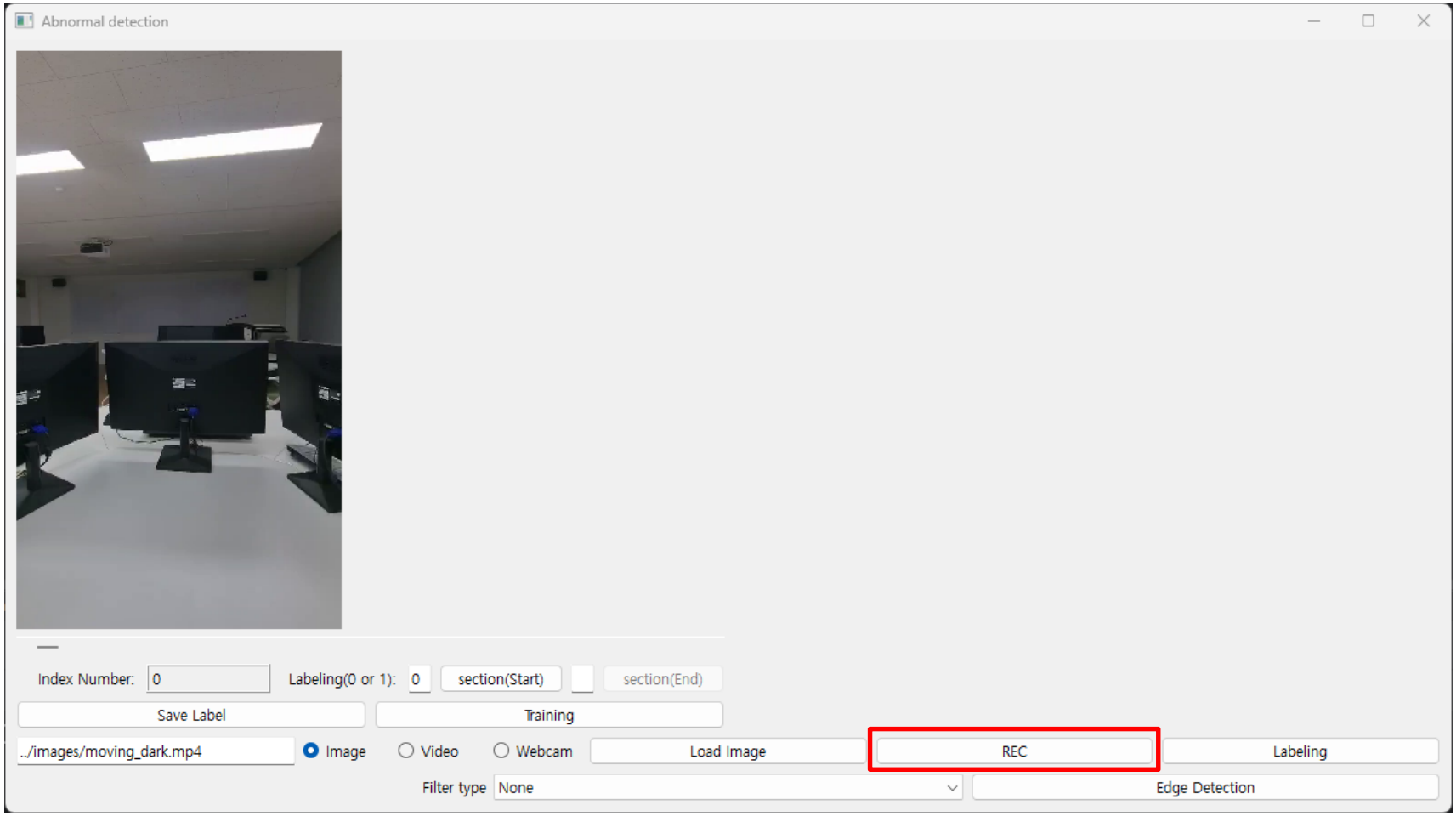
5.8 기본 배치

- Labeling 및 학습을 위한 기본 버튼 app에 배치하기
  - REC, Labeling, Save Label, Training 버튼 추가



5.10 REC 버튼

- REC 버튼



5.9 REC 버튼

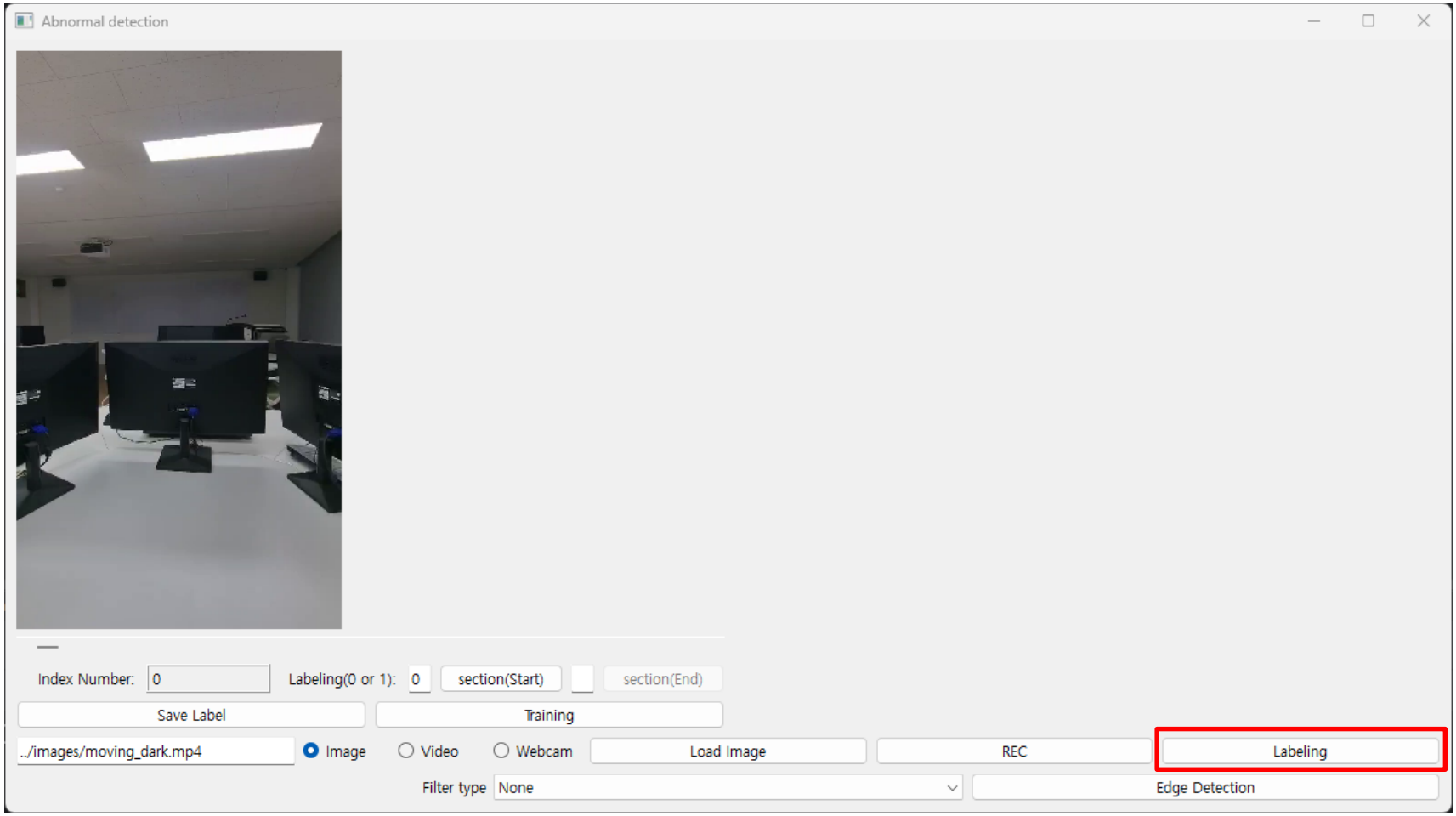
- REC 버튼
  - REC 버튼은 학습용 영상 비디오를 생성하기 위한 버튼
  - 웹캠의 장치 접근 권한 획득
  - 비디오 생성을 위한 cv2.VideoWriter_fourcc (압축방식 정의) 및 cv2.VideoWriter (비디오 생성) 클래스 객체 생성
  - frame을 읽어오고 cv2.VideoWriter 객체에 frame 쓰기

5.9 REC 버튼


```
def recording(self):
    cap = cv2.VideoCapture(0)
    if not cap.isOpened():
        print("Camera open failed!")
        sys.exit()
    w = round(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    h = round(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = cap.get(cv2.CAP_PROP_FPS)

    fourcc = cv2.VideoWriter_fourcc(*'DIVX')  # *'DIVX' == 'D', 'I', 'V', 'X'
    delay = round(1000 / fps)

    out = cv2.VideoWriter('output.avi', fourcc, fps, (w, h))
    if not out.isOpened():
        print('File open failed!')
        cap.release()
        sys.exit()
    while True:
        ret, frame = cap.read()
        if not ret:
            break
        out.write(frame)
        cv2.imshow('frame', frame)
        if cv2.waitKey(delay) == 27:
            break

    cap.release()
    out.release()
```

5.10 Labeling 버튼

- Labeling 버튼



5.10 Labeling 버튼

Labeling 버튼

- Labeling 버튼은 동영상 파일을 불러오고 기본적인 스크롤바 셋팅등 Labeling을 위한 준비 상태를 만드는 버튼
- QLineEdit의 파일경로에 접근하여 동영상 파일을 불러오기
- 동영상 Frame들을 frame_list에 저장
- 스크롤바의 최대값 설정 및 활성화
- 스크롤바의 현재 위치 값을 QLabel에 출력
- 동영상 파일 접근권한 반환
- QLineEdit의 Labeling 값 수정 발생 시 값을 Ground Truth List에 저장
- 주요 버튼의 이벤트 및 Slot 함수 연결

5.10 Labeling 버튼

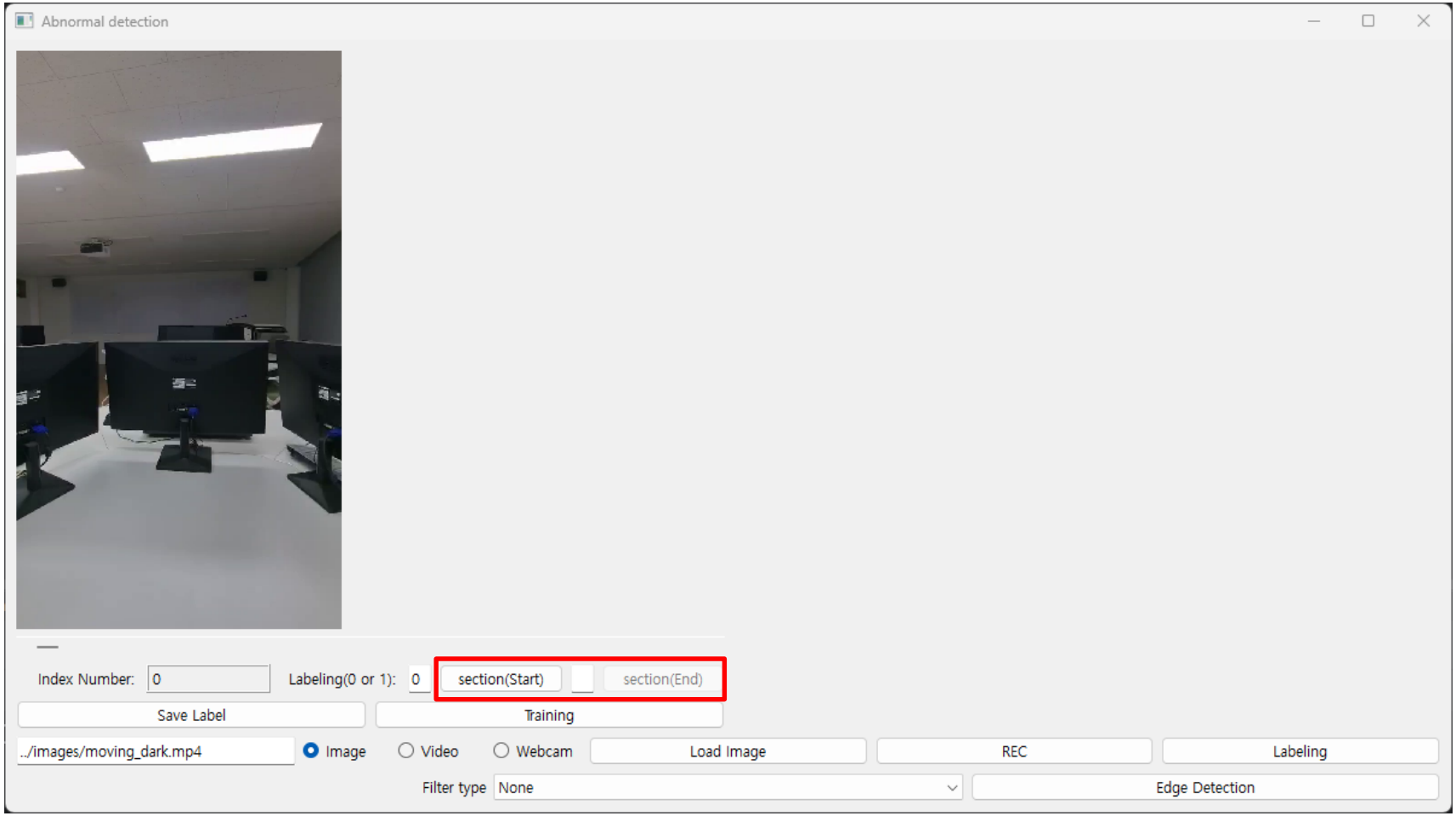

```
def labeling(self):
    print("labeling")
    self.kill_thread()
    self.labeling_capture = cv2.VideoCapture(self.edit.text())

    ret = True
    cnt = 0
    self.frame_list = []
    while ret:
        ret, frame = self.labeling_capture.read()

        if not ret:
            break
        cnt += 1
        self.frame_list.append(frame)
        self.labeling_Ground_truth.append(0)

    print("Loading video is complete")

    self.button_save_label.clicked.connect(self.save_label)
    self.button_training.clicked.connect(self.training_perceptron)
```

5.10 Labeling 버튼


```
        cnt += 1
        self.frame_list.append(frame)
        self.labeling_Ground_truth.append(0)

    print("Loading video is complete")
    print(f"The number of frame: {len(self.frame_list)}  Ground-truth len: {len(self.labeling_Ground_truth)}" )
    self.scroll_bar.setMaximum(len(self.frame_list)-1)  # 최대값 설정
    self.scroll_bar.setVisible(True)

    # scroll의 index number QLabel과 Labeling의 QLineEdit을 초기값 설정
    self.label_idx_scroll.setText("0")
    self.edit_text_labeling.setText(f"{self.labeling_Ground_truth[0]}")

    self.labeling_capture.release()
    self.update_image(self.frame_list[0])

    self.edit_text_labeling.editingFinished.connect(self.set_label)
    self.button_start_labeling.clicked.connect(self.set_section_start)
    self.button_end_labeling.clicked.connect(self.set_section_end)
    self.button_save_label.clicked.connect(self.save_label)
    self.button_training.clicked.connect(self.training_perceptron)
```

5.11 section 버튼

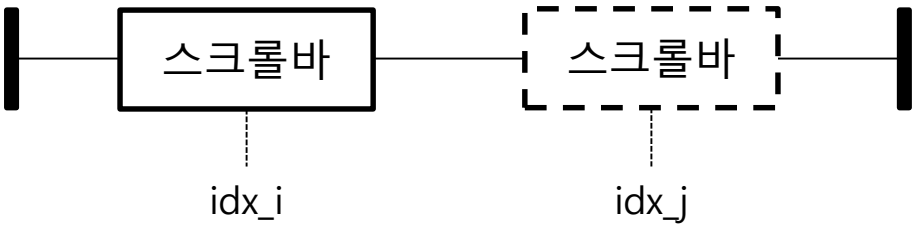
- section(start)와 section(end) 버튼



5.11 section 버튼

section(start)와 section(end) 버튼

- section(start)과 section(end) 버튼은 start 버튼을 눌은 시점의 스크롤바 위치의 frame부터 end 버튼을 눌은 시점의 스크롤바 위치의 frame 까지 지정한 label 번호로 labeling을 돕는 버튼
- section(start) 버튼
  - 현재 스크롤바의 위치를 저장
  - section(end) 버튼 활성화
  - section(start) 버튼 비활성화
- section(end) 버튼
  - end 버튼 클릭 시점의 스크롤바 위치 (idx_j) 저장
  - start 버튼 시점의 스크롤바 위치 (idx_i)와 end 시점 스크롤바 위치 (idx_j) 구간의 Ground-truth list를 수정



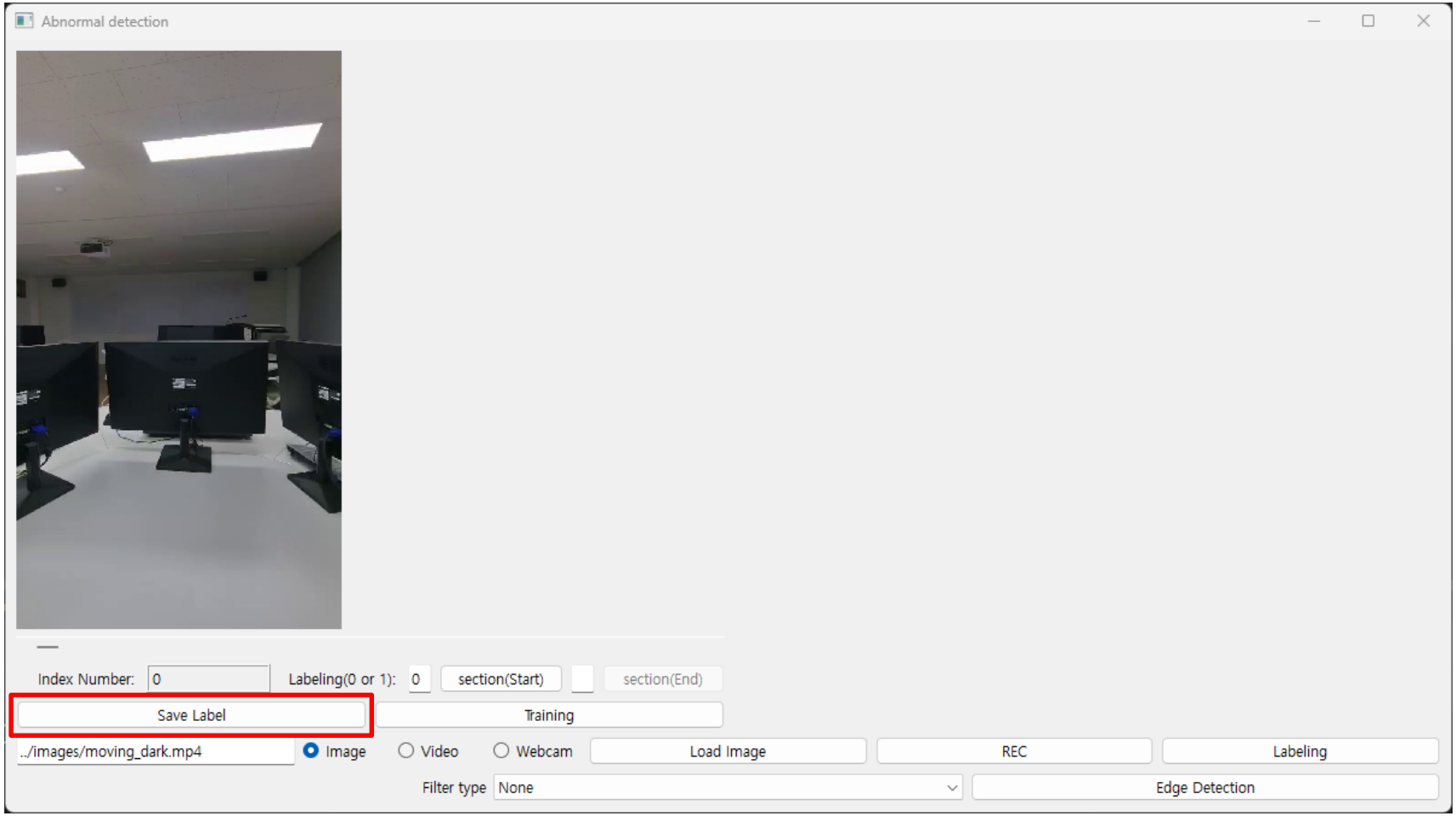
5.11 section 버튼


```
def set_section_start(self):
    print("section start")
    self.start_idx = self.scroll_bar.sliderPosition()
    self.button_end_labeling.setEnabled(True)
    self.button_start_labeling.setEnabled(False)
```

```
def set_section_end(self):
    print("section end")
    idx_i = self.start_idx
    idx_j = self.scroll_bar.sliderPosition()

    if idx_j > idx_i:
        label_value = [int(self.edit_section_label.text()) for i in range(idx_j-idx_i+1)]
        self.labeling_Ground_truth[idx_i:idx_j+1] = label_value
    elif idx_j < idx_i:
        label_value = [int(self.edit_section_label.text()) for i in range(idx_i - idx_j + 1)]
        self.labeling_Ground_truth[idx_j:idx_i+1] = label_value

    self.button_end_labeling.setEnabled(False)
    self.button_start_labeling.setEnabled(True)
```

5.12 Save Label 버튼

- Save Label 버튼



5.12 Save Label 버튼

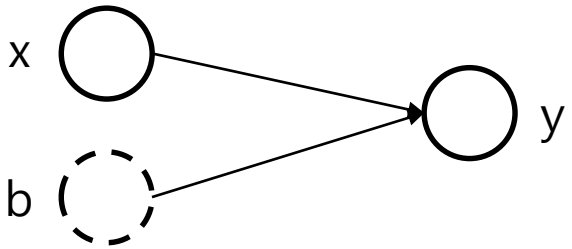
- Save Label 버튼
  - Save Label 버튼은 frame마다 기록된 정답(Ground-truth) list를 넘파이(.npy) 파일로 저장하는 버튼
  - Ground_truth list를 numpy array로 변환
  - 불러온 파일경로의 파일 이름을 추출
  - .npy 파일로 저장

5.12 Save Label 버튼


```
def save_label(self):
    print("save label")
    save_Ground_truth = np.array(self.labeling_Ground_truth)

    path_split = self.edit.text().split('/')
    self.file_name = path_split[-1].split('.')

    np.save(f"./{self.file_name[0]}.npy", save_Ground_truth)
```

5.13 Training 버튼

- Training 버튼
  - Training 버튼은 정상과 비정상 상태 구분을 위한 모델 학습을 위해 준비된 학습 영상과 Ground-truth 값을 이용하여 퍼셉트론 모델을 학습 시키는 버튼
  - 모델 정의: 입력층, 하나의 은닉층, 출력층이 있는 모델 정의
  - 입력 데이터 준비: frame list로부터 edge를 찾고 각 프레임의 총 edge 화소 수를 계산
  - 정답 데이터 준비: Ground-truth 값이 저장된 .npy 파일로부터 numpy array 읽어오기
  - 모델 학습 시작



5.14 임계값 학습을 통한 화면 전환 구현

구현 및 실습

- (필수) Thread 구현 Version 2를 기본으로 다음 요소들 추가하기
  - REC, Labeling, Save Label, Training, section(start), section(end) 버튼 추가
  - Labeling 버튼 클릭 후 스크롤바로 frame 이동과 Labeling 삽입 기능 추가
- (필수) REC 버튼을 이용하여 웹캠으로부터 영상을 저장 (정상 상황과 비정상 상황 포함)
- (필수) Labeling 버튼을 이용하여 기록된 영상을 불러오고 Labeling 작업 수행
- (필수) 저장된 모델을 이용하여 frame의 엣지 수에 따른 화면 전환 구현
- (선택) 모든 frame에 대한 sum_edge 값을 기록하고 최대, 최소값을 이용하여 frame들의 엣지 수를 0부터 1 사이 값으로 정규화 **후 모델을 학습**. 그리고 **학습된 모델을** 이용하여 화면 전환 구현하기 (※ 완성 시 가산점)

```
# 정규화 Hint
# scaler = MinMaxScaler()
# scaler.fit(train_x)
# train_x = scaler.transform(train_x)
```