



Lee seon jae

인공지능 모델 개발자로서의
저의 커리어는 '인공지능 개발자
양성과정'을 통해
시작되었습니다.

6개월 간 2개의 프로젝트에서
함께 한 노력 과정을 소개합니다.

[학력]

- 전남과학대학교 특수장비학과 졸업(21년)

[자격]

- 문서실무사 2급 (17년)
- e-test 워드 2급 (19년)
- e-test 파워포인트 2급 (20년)

PORTFOLIO

[프로젝트 요약 / 인덱스]

Project #1. 리버싱엔지니어링을 통한 반려동물 SNS 서비스

p. 1 ~ 1-3

Why	반려동물을 키우는 반려인들에게 유익한 정보와 소통공간 제공
What	① 게시물 검색 및 조회, ② 게시물 업로드, ③ 댓글, 좋아요, 공유 등
How	VS code, Node.js, HTML, CSS, JAVASCRIPT, MySQL, Flask 등



Project #2. Yolov5 와 Open CV 활용 얼굴 및 행동인식기반 아동학대 방지 서비스

p. 2 ~2-3

Why	아동학대 문제 방지 및 즉각적인 대응
What	① CCTV 활용 아동학대 감지, ② 보호자 및 어린이집 조회, ③ 전문가 및 관할경찰 연계
How	Open CV, yolov5, Python, HTML, CSS, JAVASCRIPT, MySQL, Flask 등



Python

파이썬을 활용하여 외부의 데이터를 수집하고, 가공하며,
시각화 작업을 수행할 수 있습니다. 특히, 조건문 / 반복문 /
함수 및 예외처리의 상황들을 경험하며, 구현방식을
학습하였습니다.



HTML, JS, CSS

HTML, JS, CSS를 활용하여 프론트엔드 웹 디자인
설계하고 개발하며, 시각화 작업을 수행할 수 있습니다.
특히, CSS / JS
디자인 및 클릭 이벤트를 경험하며, 구현방식을
학습하였습니다.

Dev.
역량 요약

YOLOv5

YOLOv5

YOLOv5를 활용하여 모델을 설계 및 수정, 보완하고
데이터를 학습시키며 커스텀 모델을 생성할 수 있습니다.
특히, 커스텀 데이터 제작 및 라벨링을 경험하며,
구현방식을 학습하였습니다.



Java & JDBC

자바와 JDBC를 활용하여 시그마와 오라클 DB 를 연결하여
입력한 데이터를 DB에 저장 및 DB 데이터 불러오기
작업을 수행할 수 있습니다. 특히, sql문 작성 및 예외처리를
경험하며, 구현방식을 학습하였습니다.

1. 요구사항 정의

- 서비스 목표 기능
 - 게시물을 통한 반려동물 정보 공유
 - Pinterest(SNS) Clone
 - 인터페이스 디자인 구현
 - 팔로우, 좋아요, 댓글 등 회원들 간의 소통
- 개발 데이터 범주 설정

회원정보	<u>회원번호</u> , 이메일, 비밀번호,
게시물	<u>게시물번호</u> , 제목, 내용,
댓글	<u>댓글순번</u> , 댓글 내용, 일시,

2. UI/UX

- HTML을 활용한 페이지 구현
 - 예시 : 로그인 페이지, 메인 페이지 등
- CSS를 활용한 애니메이션 효과 구현
 - 예시: 로그인 페이지, 메인 페이지 등
- JavaScript를 활용한 동적페이지 구현
 - 로그인, 댓글 페이지의 숨겨진 창 설계

MY ROLE

3. 기능 구현

- Node.js - Express 모듈을 활용한 페이지에 정보 전달
 - 예시 : 팔로우, 좋아요 및 댓글 등
- Node.js - FileSystem모듈을 사용한 이미지 업로드
 - 예시 : 게시물 작성 및 업로드 등
- DB, 서버
 - Node.js로 서버 연결
 - mysql을 사용한 DB생성

MY ROLE

주요 개발 환경



본 프로젝트를 통해 프론트,백 / 웹 디자인 구현에 필요한 다양한 자료 탐색 및 적용의 과정을 경험할 수 있었습니다.

개발 역량은 다양한 소스에 대한 탐색과 적용 노력을 통해 얻을 수 있다고 생각합니다.

이러한 노력을 웹서비스의 기존 코드 리뷰 및 외부 소스에 대한 적용으로 활용하고자 합니다.

Node.js 를 통한 사용자 프로필 사진 DB 저장 및 출력 서비스 개발을 통해 프론트,백 엔드 개발 역량을 갖추고자 노력하였습니다.

Router 프로필 이미지 변경 코드

```
router.post("/profile", upload.single("myimg_url"), (req, res) => {
  try {
    sharp(req.file.path) // 압축할 이미지 경로
      .resize({ width: 30 }) // 비율을 유지하며 가로 크기 줄이기
      .withMetadata() // 메타데이터의 exif데이터 유지
      .toBuffer((err, buffer) => {
        if (err) throw err;
        // 압축된 파일 새로 저장(덮어쓰기)
        fs.writeFile(req.file.path, buffer, (err) => {
          if (err) throw err;
        });
        let user_id = req.session.user.user_id;
        let img_url = req.file.filename;
        let sql = "update user set myimg_url = ? where user_id = ?";

        conn.query(sql, [img_url, user_id], function (err, rows) {
          //
        });
      });
  } catch (err) {
    console.log(err);
  }
  res.redirect("http://127.0.0.1:3000/Main")
});
```

- 프로필 수정 페이지에서 본인의 프로필 사진을 선택하여 직접 변경할 수 있도록 기능구현

마이 프로필 페이지

공개 프로필

회원님의 프로필을 방문하는 사용자에게 다음 정보가 표시됩니다.

사진



파일 선택 선택된 파일 없음

변경

이름

성

소개

회원님의 이야기를 들려주세요

- 이미지 업로드 기능 코드를 이해
- 이미지 파일 선택 후 변경 버튼 클릭시 DB에 정보 저장은 하나
웹페이지에 실시간으로 반영될 수 있도록 코드 구현

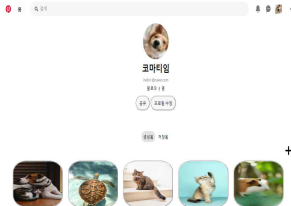
Node.js 를 통한 팔로우 서비스 개발을 통해 백 엔드 개발 역량을 갖추고자 노력하였습니다.

Router 팔로우 코드

```
router.get("/follow", function (request, response) {
  let pin_id = request.session.pin_all.pin_id[p].pin_id;
  let sql = "select * from comment where pin_id=?";
  let follow_id = null;
  let follower_id = request.session.user.user_id;
  let user_id = request.session.pin_all.writer[0].user_id;
  conn.query(sql, [request.session.pin_all.pin_id[p].pin_id], function (err, comment) {
    if (comment) {
      conn.query("select * from user", function (err, user_data) {
        response.render("pin", {
          pin_all: request.session.pin_all,
          comment_id: comment_id,
          user: request.session.user,
          user_data: user_data
        });
        conn.query("select * from pin where pin_id = " + pin_id, function (err, row) {
          if (row) {
            let chksql = "select * from follow where follower_id=? and user_id=?";
            conn.query(chksql, [follower_id, user_id], function (err, chk) {
              if (chk.length > 0) {
                console.log("팔로우 되어있음 - " + chk);
                console.log("이미 팔로우 되어있음");
              } else {
                console.log("팔로우 안되어있음 - " + err);
                let followsql = "insert into follow values(?, ?, ?)";
                conn.query(followsql, [follower_id, request.session.pin_all.writer_id, pin_id], function (err, row) {
                  if (err) {
                    // 실패시
                    console.log(err);
                  }
                });
              }
            });
          }
        });
      });
    }
  });
});
```

- 다른 인원들의 프로필 팔로우 버튼 클릭시 본인의 팔로우 목록에 추가

마이 프로필 페이지



- 본인이 팔로우한 인원 및 팔로잉 인원 확인 및 팔로우한 사람의 게시물이 보여짐

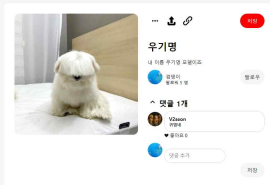
Node.js 를 통한 게시물 상세 페이지 서비스 개발을 통해 백 엔드 개발 역량을 갖추고자 노력하였습니다.

Router 게시물 상세 페이지

```
// 게시물 상세 페이지
router.post("/pin", function (request, response) {
  let pin_id = request.body.pin_id;
  let sql = "select * from pin where pin_id=?";
  con.query("select * from pin where pin_id =?" + pin_id, function (err, rows) {
    if (rows) {
      //DB에 저장된 게시물 찾기
      con.query("select * from user where user_id =?" + rows[0].user_id, function (err, writer) {
        con.query("select * from follow where user_id =?" + rows[0].user_id, function (err, follower_cnt) {
          if (writer) {
            con.query("select * from like", function (err, like_C) {
              con.query("select * from comment where pin_id =?" + pin_id, function (err, comment) {
                if (comment) {
                  let likeList = [];
                  for (let i = 0; i < comment.length; i++) {
                    sum = 0;
                    k = 0;
                    while (k < like_C.length) {
                      if (comment[i].comment_id == like_C[k].comment_id) {
                        sum++;
                      }
                      k++;
                    }
                    likeList.push(sum);
                  }
                  request.session.pin_all = {
                    "pin_id": rows,
                    "user_id": request.session.user,
                    "writer": writer,
                    "likeList": likeList,
                    "follower_cnt": follower_cnt,
                    // "comment": comment
                  };
                  response.redirect("/pin");
                }
              }
            }
          }
        }
      }
    }
  }
});
```

- 메인 페이지에서 게시물 클릭시 게시물 업로드한 인원의 아이디, 제목, 내용이 보여짐
- 좋아요, 댓글 기능을 사용할 수 있음

게시물 상세 페이지



- 이미지, 텍스트 업로드 기능 코드를 이해
- 게시물 클릭시 DB에 저장된 이미지경로, 제목, 내용, 댓글, 좋아요 갯수, 팔로워 등을 보여줌

요구사항 정의

- 서비스 목표 기능
 - 어린이집 CCTV영상 아동학대 판별
 - 전문가 연계 서비스 및 영상 열람 기능
 - 판별된 영상에 대해 기록/저장하여 증거훼손 방지
- 세부 기능 명세 구성
 - YOLOv5 이미지 특징분석을 활용한 아동학대 판별
 - Html, css, js를 활용한 웹 개발
 - Node.js를 활용한 서버 구축

주요 개발 도구



데이터 분석

데이터 확보

- 정상 행동과 폭력 행동으로 나뉘는 영상(3~5초)

데이터 촬영 : 1,300여 개

데이터 전처리

- 객체 탐지가 힘들 것으로 판단되는 영상이나 저 화질 영상 제거
- 프레임 단위로 나뉜 이미지 파일로 변환
- 'labelimg' 사용하여 각 이미지에 라벨링(바운딩 박스)

객체 탐지

- 라벨링한 커스텀 데이터를 YOLOv5 모델에 학습
- 영상 입력 시 성인, 아동, 폭행을 탐지 후 폭행 탐지 시 해당 부분 녹화

MY ROLE

웹 서비스 구현

UI / UX 설계 및 디자인

- Adobe Photoshop을 활용한 로고 및 일러스트 디자인
- Adobe XD를 활용한 레이아웃 설계

웹페이지 구현 및 서버 구축

- Node.js를 이용한 직관적이고 효율적인 서버 구축
- HTML, CSS 마크업을 활용한 웹 화면구현
- Javascript 및 JQuery 프레임워크를 활용한 웹 기능 구현

데이터베이스 구축

- MySQL을 이용한 데이터베이스 설계

본 프로젝트를 통해 YOLOv5의 구현 방식 및 절차를 학습할 수 있었습니다.
이러한 노력을 딥러닝 모델 개발 환경 개선에 활용하고자 합니다.

YOLOv5 와 Open CV 활용

얼굴 및 행동인식기반 아동학대 방지서비스

인공지능 융합서비스개발자과정/22년(약 4주)/4인 팀 프로젝트

Q. 프로젝트에서 어떤 결과가 있었고, 문제를 해결하고자 한 노력은 어떤 것이 있었나요?

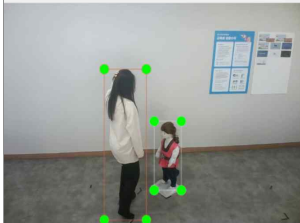


개발 과정에서의 주요 이슈	이슈 개선을 위한 노력
성인과 아동간의 폭행 데이터셋이 존재하지 않음	CCTV와 유사한 환경 조성 후 직접촬영
아동 마네킹을 활용한 영상 촬영 후 YOLOv5 테스트 결과 탐지 정확도 30~40% 현저히 낮음	마네킹을 사람과 유사하게 조성, 카메라 화질조정 후 재촬영 탐지 정확도 70~80%로 향상

YOLOv5 를 통한 아동학대 감지 서비스 개발을 통해 딥러닝 모델 개발 역량을 갖추고자 노력하였습니다.

데이터 전처리

ox#PCW\Desktop#0611 폭행#train#train02345.jpg [3739 / 6971]



YOLOv5 커스텀 모델 생성

```
Analyzing anchors... anchors/target = 3.92. Best P
Image sizes 416 train, 416 test
Using 2 dataloader workers
Logging results to runs/train/gun_yolov5s_results
Starting training for 50 epochs...
```

Epoch	gpu_mem	box	obj	cls
0/49	1.85G	0.09176	0.04846	0
	Class	Images	Targets	
	all	595	680	

Epoch	gpu_mem	box	obj	cls
1/49	1.85G	0.06467	0.04664	0

- 이미지 데이터 전처리와 YOLOv5 구동방식에 대해 이해
- 직접 생성한 데이터를 전처리
- 전처리한 데이터를 기반으로 높은 정확도와 성능을 보여줄 수 있도록 모델 수정 및 보완하여 학습

YOLOv5 를 통한 아동학대 감지 서비스 개발을 통해 딥러닝 모델 개발 역량을 갖추고자 노력하였습니다.

폭행감지 및 저장

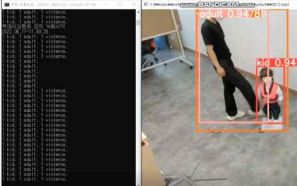
```

cv2.imshow(str(p), img)
key = cv2.waitKey(1)
img = annotator.result()
if vim_log:
    cv2.imshow(str(p), img)
    cv2.waitKey(1) # 1 millisecond
if record == False:
    if count2 == 0:
        record = True
        LOGGGR.info("3초 이상행동 감지 복화소자")
        count1 = 0
        timestr = time.strftime("%Y.%m.%d-%H.%M.%S")
        timestr2 = time.strftime("%Y.%m.%d-%H.%M.%S")
    if record == True:
        if count1 == 7:
            LOGGGR.info("녹화종료")
            count2 = 0
            timestr1 = time.strftime("%Y.%m.%d-%H.%M.%S")
            vid_writer[1].write(img)
            record = False
        LOGGGR.info(img)
        if 'violence' not in abc:
            count1 += 1
        if 'violence' in abc:
            count2 += 1

save_path = '스마트유치원/폭행감지/' + timestr
no_save_path = 'C:/Users/taehd/Desktop/yolov5/스마트유치원/폭행감지/' + timestr2
# Save results (image with detections)
if save_img:
    if dataset_mode == 'image':
        cv2.imwrite(save_path, img)
    else: # video: opencv
        if vid_path[1] != save_path: # new video
            vid_path[1] = save_path
            # release previous video writer
            if vid_cap: # video
                fps = vid_cap.get(cv2.CAP_PROP_FPS)
                w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
                h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
            else: # stream
                fps, w, h = 30, img.shape[1], img.shape[0]
            save_path = str(Path(save_path).with_suffix('.mp4')) # force *.mp4 suffix on results videos
            vid_writer[1] = cv2.VideoWriter(save_path, cv2.VideoWriter_fourcc('mp4'), fps, (w, h))
            vid_writer[1].write(img)

```

게시물 상세 페이지



- YOLOv5 구동방식에 대해 이해
- 기존 YOLOv5 구동 코드를 제작 프로젝트 방향성에 맞게 변경하여 객체가 탐지 되었을 때 일정 구간 녹화 및 저장

개발자로서의 경쟁력을 강화 시키기 위해

Java, Python 을 활용한 코딩/머신러닝,딥러닝 역량을 강화하고자 노력하였습니다.

Java, Python등 언어학습

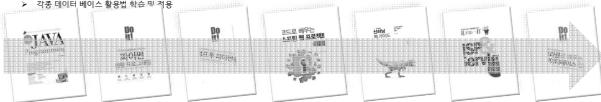
- Java, Python 의 언어 구조 및 학습
 - 구글링을 통한 이미지 데이터 크롤링
- 자료구조 및 알고리즘 독서
 - 케라스 창시자에게 배우는 딥러닝
 - 모두의 딥러닝, 파이썬 생활 프로그래밍
- 업무에 관련된 기술 학습 및 적용
 - 이클립스, Vs code, 주피터를 활용한 학습
 - 각종 라이브러리 문법, 연산자 학습 및 적용
 - 각종 데이터 베이스 활용법 학습 및 적용

머신러닝,딥러닝 학습

- CNN등 각종 모델 학습
- 데이터 전처리 방법 등 기초학습
- 캐글 문제 분석 및 실습
- 머신러닝과 딥러닝 기초학습
- YOLOv5 모델 수정 및 보안 학습
- YOLOv5 커스텀 모델 생성 학습

자료 관리 및 실무 현장 이해

- 깃허브를 활용한 자료 및 히스토리 관리
 - 깃허브 URL : <https://github.com/V2seon>
- '기획자 데이먼' 유튜브 채널을 통한 개발 환경 이해
 - IT 제품 기획을 위한 주요 직무 3
 - 웹/앱 설계 기본 문서작성 방법



이론적으로만 이루어진 기술학습에 그치지 않고
학습한 기술들을 실제 서비스를 구현하는데 기여하고자 합니다.

End of Document

빠르게 변화하는 트렌드와 사회 이슈에 적합한 더 나은 인공지능개발을 위해 여러 가지 모델들을 분석해 보고 기존의 인공지능보다 한층 더 성능이 향상된 인공지능 서비스를 고객들에게 제공할 수 있는 직무역량을 갖춘 전문가가 되고 싶습니다.