

---

## YOLO를 활용한 유실 및 유기 반려동물 조회 앱 개발

- Android Studio & YOLOv5 -

---

- ▷ 이름: 손 수 지
- ▷ 학번: 202011933
- ▷ 학과: 전자정보통신공학부 전자공학 전공
- ▷ 과목(분반): 임베디드시스템설계 (101)
- ▷ 담당교수: 박 한 훈 교수님
- ▷ 제출일: 2023.06.21.(수)

- 목 차 -

|                                    |           |
|------------------------------------|-----------|
| <b>I. 서론</b> .....                 | <b>3</b>  |
| 1. 과제 수행 배경(동기)과 목적 .....          | 3         |
| 2. 기존 사례 분석 및 수행한 과제 내용의 차별성 ..... | 4         |
| 3. 과제 수행 방향 및 방법 .....             | 4         |
| <br><b>II. 본론</b> .....            | <b>5</b>  |
| 1. 앱의 구조와 동작 및 디자인 .....           | 5         |
| 2. 이미지 처리 .....                    | 11        |
| <br><b>III. 결과 및 분석</b> .....      | <b>11</b> |
| 1. 실험(시뮬레이션) 환경 .....              | 11        |
| 2. 실험 결과 및 분석 .....                | 11        |
| 3. 한계 및 개선 방향 .....                | 12        |
| <br><b>IV. 결론</b> .....            | <b>12</b> |
| 1. 과제 수행 결과 요약 및 느낀 점 .....        | 12        |
| <br><b>V. 참고문헌</b> .....           | <b>14</b> |

# I. 서론

## 1. 과제 동기 및 목적

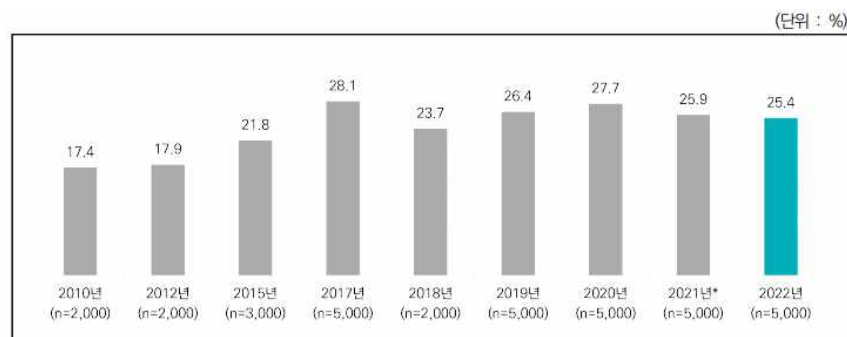
### 1) 과제 수행 배경

#### ① 배경

현대사회는 1인 가구의 증가와 저출산·고령화 등으로 인해 반려동물과 함께하는 인구가 증가하고 있으며, 관련 산업들은 지속적으로 성장하는 추세를 보이고 있습니다.

농림수산물식품교육문화정보원에서 공개한 ‘동물보호에 대한 국민 의식조사’에서 2010년부터 2022년까지의 반려동물 양육 여부 결과[그림 1]를 공개했습니다. 결과 [그림 1]을 확인해보면, 전체 응답자(5,000명) 중 25.4%가 현재 거주지에서 직접 반려동물을 기르고 있습니다. 또한, 현재 반려동물을 미양육하는 응답자(3,627명) 중 52.7%가 과거에 반려동물 양육 경험이 있다고 합니다. 즉, 과거 반려동물 양육 경험을 포함하면 전체 응답자의 65.7%가 반려동물을 기른 경험이 있다고 합니다.

신한카드 빅데이터연구소는 지난해(2022년) 자사 고객이 동물병원, 애견 호텔 등에서 이용한 금액이 1인당 연평균 35만3000원 수준으로 나타났다고 밝혔습니다. 반려동물 관련 지출액은 2019년 26만2000원이었지만 2020년 28만3000원, 2021년 31만3000원 등으로 매년 증가하는 추세입니다.



[그림 1]

반려동물 양육 가구와 반려동물 관련 지출액이 늘어난다는 데이터를 통해 국내 반려동물 시장에서는 엄청난 양적 성장과 함께 질적인 성장도 이뤄지고 있다는 것을 알 수 있습니다. 또한 「동물보호법」도 점점 강화되고 있어 동물을 학대할 경우 3년 이하의 징역이나 3천만 원 이하의 벌금에 처해진다는 기준이 마련되어 있습니다. 현대사회에서는 동물을 하나의 물건으로 대하던 시대에서 개체로서 최소한의 권리를 인정해 줘야 한다는 방향으로 인식이 변화하고 있습니다.

반려동물에 대해서는 이미 하나의 가족처럼 생각하는 문화가 만들어지고 있는 현재 사회에서 반려동물 유실 및 유기는 큰 사회적 문제로 다가오고 있습니다. 이러한 사회적

문제에 도움이 되고자 나온 한가지 방안은 “유실 및 유기 반려동물 앱”입니다.

“유실 및 유기 반려동물 앱”이란 주로 반려동물을 유실하거나, 유실 혹은 유기된 동물을 발견한 사람이 앱의 게시판에 동물의 특징 및 사진을 업로드해 주인을 찾아주거나, 유기견 임시보호 및 입양을 진행할 수 있도록 하는 시스템입니다.

저는 이러한 사회적 문제를 배경으로 하여 “유실 및 유기 반려동물 앱”을 개발하고자 하였습니다.

## 2. 기존 사례 분석 및 수행한 과제 내용의 차별성

### 1) 기존 사례 분석

#### 포인핸드 - 유기 동물 입양 & 실종 동물 찾기

다양한 “유실 및 유기 반려동물 앱”이 존재했지만, 가장 널리 알려져있고 시스템이 체계적이라고 생각하게 된 ‘포인핸드’라는 앱이 존재하였습니다. 해당 앱은 실종/보호 게시물을 앱에 작성할 때, 날짜, 지역, 장소, 연락처, 품종, 성별 등등과 함께 이미지를 첨부하면 글 등록이 진행되는 방식이었습니다. 어떤 방향으로 앱을 제작 및 보완할지 고찰하기 위해 제가 직접 앱을 사용해보고 사용자들의 불만사항을 찾아보았습니다. 그 결과로 저는 게시글의 이미지 등록에 대해 보완을 하면 좋지 않을까 생각하게 되었습니다.

기존의 사용자들 중 일부는 “유기동물 입양 시 무조건 직접 가서 사진에 있는 유기동물이 맞는지 확인해라”고 합니다. 그 이유는 거짓된 사진 때문입니다. 유기동물 입양 게시판을 통해 입양을 진행한 한 사용자는 게시물의 사진과는 다른 유기동물을 받게 되었다는 사례입니다. 뿐만 아니라, 제가 직접 앱을 사용해본 결과 동물의 사진이 아닌 전혀 연관없는 아무사진이나 넣어도 게시글이 작성되는 것을 확인하였습니다. 이런 부정확하고 악의적인 게시글을 통해 사용자들은 앱 이용에 불편함을 느끼게 됩니다.

그래서 저는 이러한 문제점을 보완하고자 이미지에는 동물이 감지된 이미지만 등록할 수 있도록 수정한 앱을 개발하고자 하였습니다.

## 3. 과제 수행 방향 및 방법

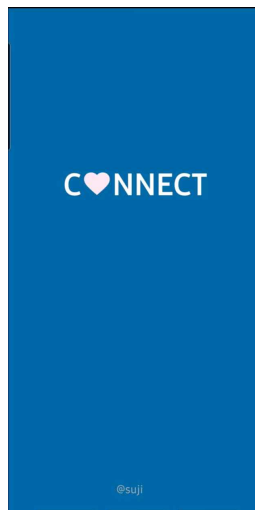
앞에서 제기한 문제점을 보완한 앱 개발을 위해, 영상처리와 컴퓨터 비전 분야에서 중요한 응용 프로그램 중 하나인 객체 탐지기법 중 하나인 YOLO(You Only Look Once)를 사용하여 등록할 사진을 객체 탐지를 통해 동물을 감지하고, 동물이 존재한다고 판단되는 사진만 등록 가능하도록 개발을 하고자 합니다. 모든 동물들 중에서도 가장 높은 양육 비율을 차지하는 강아지와 고양이에 대해서만 등록할 수 있도록 진행하고자 합니다.

## II. 본론

### 1) 앱의 구조와 동작 및 디자인

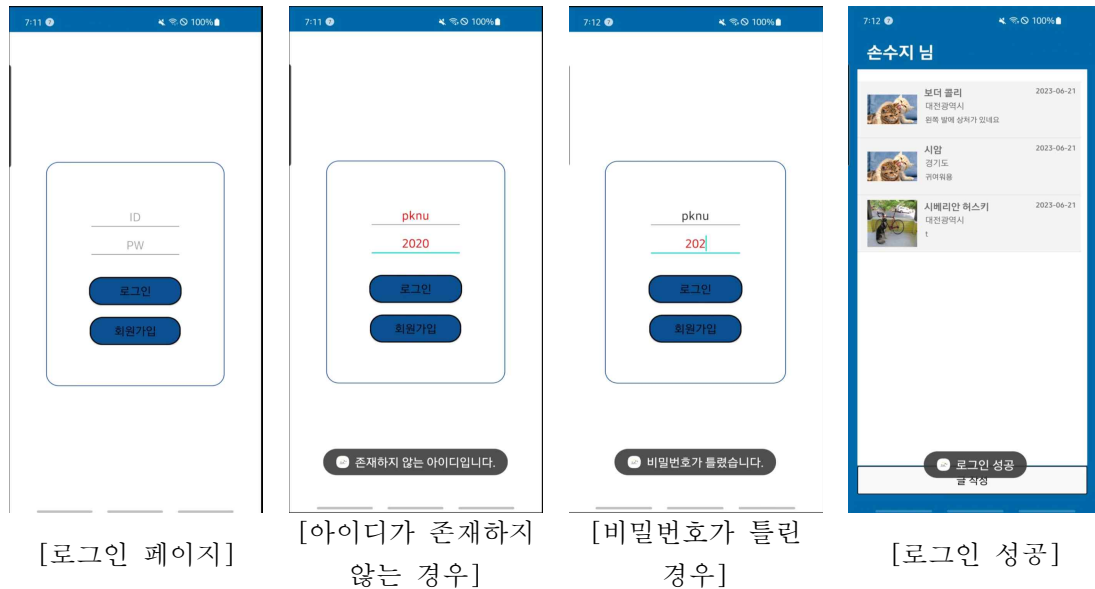
앱의 구조 및 동작은 아래의 [그림 3], 데이터베이스의 구조 및 데이터는 [그림 4]와 같이 설계하였습니다.

먼저, 앱을 실행하게 되면 앱의 이름과 제 이름이 로고로 들어간 로딩페이지가 보이게 하는 LoadingActivity가 실행 됩니다. 로딩페이지는 3초동안 실행되고, 이후 로그인 및 회원가입 페이지인 LoginSigninActivity가 실행됩니다.

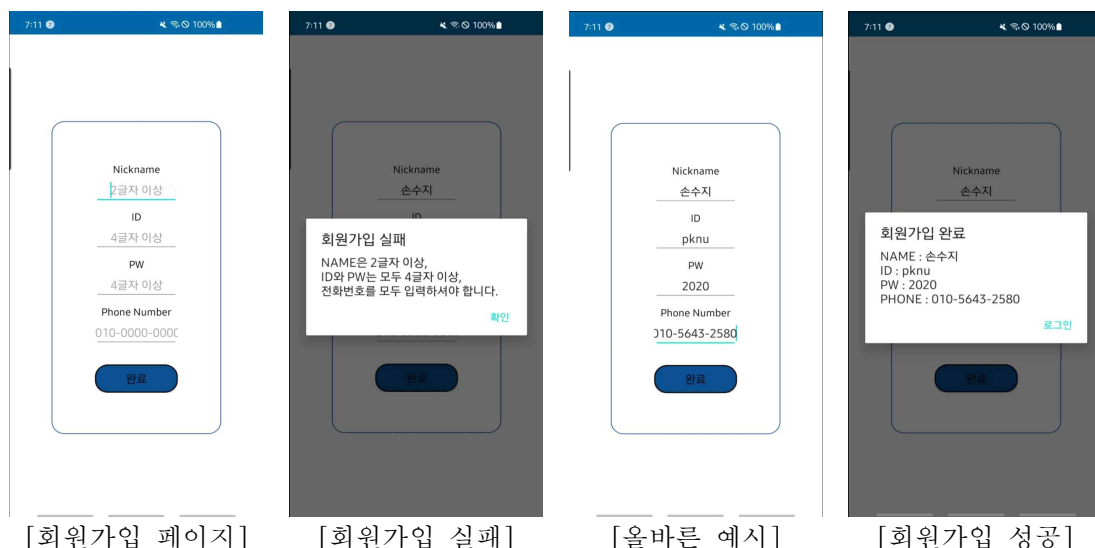


[로딩 페이지]

로그인 및 회원가입 페이지는 아이디와 비밀번호를 입력할 수 있는 각각의 EditText칸과 로그인을 시도할 수 있는 로그인 버튼, 회원가입페이지로 이동할 수 있는 회원가입 버튼이 존재합니다. 아이디와 비밀번호를 입력하는 칸에 사용자가 아이디 및 비밀번호를 입력하고 로그인 버튼을 누르게 되면 UserDatabase의 user 테이블의 모든 데이터 중 입력된 아이디가 존재하는지, 아이디가 존재한다면 해당 아이디의 비밀번호와 입력한 비밀번호가 같은지를 확인한 후 처리됩니다. 만약 아이디가 존재하지 않는다면 입력된 아이디와 비밀번호의 텍스트를 빨간색으로 설정하고, “존재하지 않는 아이디입니다.”라는 토스트메세지를 띄웁니다. 만약 아이디가 존재하지만, 비밀번호가 틀린 경우에는 비밀번호의 텍스트만 빨간색으로 설정하고 “비밀번호가 틀렸습니다.”라는 토스트메세지를 띄웁니다. 만약 아이디가 존재하고, 해당 아이디에 대한 비밀번호가 일치한다면 “로그인 성공”이라는 토스트메세지를 띄우고, 게시판이 존재하는 ConnectActivity로 이동합니다.



로그인 및 회원가입 페이지에서 회원가입 버튼을 누른 경우, 회원가입을 진행할 수 있는 페이지로 이동하는 SignInActivity가 실행됩니다. 회원가입의 경우 사용자의 이름(닉네임), 아이디, 비밀번호, 전화번호를 입력받습니다. 이름은 두글자 이상, 아이디 및 비밀번호는 4글자 이상, 전화번호는 11글자가 모두 입력되어야만 완료 버튼을 눌러 회원가입이 정상적으로 진행되도록 설계하였습니다. 만약, 4개의 EditText칸의 조건을 모두 만족시키지 못했다면 회원가입 실패를 알리는 AlertDialog창을 띄우고, 확인 버튼을 누르면 회원가입을 이어서 진행할 수 있도록 설계하였습니다. 모든 조건을 만족하도록 데이터를 입력한 후에 완료 버튼을 누르면 회원가입 완료를 알리는 AlertDialog창에 사용자가 입력한 이름, 아이디, 비밀번호, 전화번호를 확인할 수 있도록 보여준 뒤에, 로그인 버튼을 누르면 로그인 페이지로 이동할 수 있도록 LoginSignInActivity가 실행됩니다.



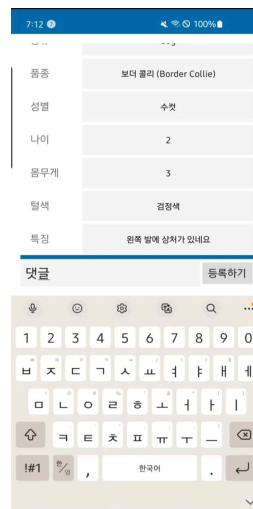
로그인을 완료한 뒤 게시판이 존재하는 ConnectActivity가 실행되면 PostDatabase의 post 테이블에서 모든 게시글을 불러와 리스트뷰 어댑터를 통해 반려동물의 사진, 종류, 지역, 특징, 작성 날짜가 들어가도록 하였습니다. 사용자는 간략한 정보가 있는 게시판의 글들을 편리하게 읽을 수 있고, 더 자세한 정보가 알고 싶거나 댓글을 남기기 위해서 게시글을 클릭하면 게시글의 전체 정보와 댓글 창 및, 댓글을 남길 수 있는 EditText와 등록 버튼이 있는 게시글 페이지로 이동할 수 있습니다. 게시글 페이지로 이동하는 PostDetailActivity에서는 ConnectActivity에서 클릭된 아이템의 텍스트들을 받아와 PostDatabase의 post 테이블에서 해당 게시글의 모든 정보를 불러와 PostDetailActivity로 넘겨주는 과정을 통해 게시글의 모든 정보를 받아오고, 사용자가 간편하게 확인할 수 있도록 나타냅니다. 동시에 PostDatabase의 comment 테이블에서 해당 게시글의 ‘게시글 번호’와 같은 데이터들을 모두 불러와 댓글창에 나타내어줌으로써 게시글 및 댓글을 확인할 수 있게 됩니다. 댓글을 입력하는 EditText창에 텍스트를 입력하고 등록하기 버튼을 누르면 PostDetailActivity는 작성자 아이디, 작성 날짜, 텍스트, 게시글 번호를 PostDatabase의 comment 테이블에 저장함과 동시에 댓글이 나타나는 리스트뷰에 데이터를 추가함으로써, 실시간으로 댓글을 등록하고 확인할 수 있도록 구현하였습니다. 동시에 토스트메세지를 통해 “댓글 등록 완료”라는 문구를 확인할 수 있도록 하였습니다. 페이지의 상단 좌측에는 게시판 페이지로 돌아갈 수 있도록 이전 버튼을 설정해줬습니다. 이 버튼을 누르게 되면 현재 액티비티가 종료되고 ConnectActivity로 돌아가게 됩니다.



[게시판 페이지]



[게시글 페이지]



[댓글 작성 예시]



[댓글 등록 성공]

ConnectActivity의 하단에 있는 글 작성 버튼을 누르게 되면 게시글을 작성할 수 있는 MainActivity로 이동하게 됩니다. 거짓정보 입력 및 사용자의 편리함을 위해 작성자 아이디, 작성 날짜, 연락처 칸은 ConnectActivity에서 인텐트로 전달한 UserID를 사용하여 UserDatabase의 user 테이블에서 해당 아이디에 대한 전화번호

호가 설정되고, calendar 클래스를 사용해 사용자가 게시글 작성을 위해 들어온 순간의 날짜를 날짜칸에 setText() 해주었습니다.

‘지역’버튼을 누르게 되면 assets에 들어있는 region.txt의 데이터를 줄별로 읽어와 AlertDialog에 리스트 형태로 띄우고, 선택한 지역이 버튼의 text로 나타나도록 구현하였습니다. region.txt는 1특별시, 6광역시, 1특별자치시, 7도, 2특별자치도를 줄별로 나눠 저장해둔 데이터입니다.

‘종류’ 컬럼에서는 고양이와 강아지 두 개의 버튼이 있습니다. 작성하고자하는 게시글의 대상이 고양이인지 강아지인지 선택하는 항목입니다. 강아지버튼을 누르고 품종 버튼을 누르게 되면 assets 폴더의 dog.txt파일에 있는 텍스트를 줄별로 읽어와 AlertDialog에 리스트 형태로 띄우고, 선택한 품종이 버튼의 text로 나타나도록 구현하였습니다. 고양이는 cat.txt의 파일을 불러오게되며 똑같이 동작하게됩니다. 이때, 고양이와 강아지버튼이 동시에 눌러지지 않도록 조건문을 설정해 두 버튼 중 무조건 하나의 버튼만 눌러질 수 있도록 설계하였습니다. 강아지 버튼을 누르고 품종을 강아지의 품종 중 하나로 설정한 후, 고양이버튼을 누르게 되면 품종에 나타나도록 했던 텍스트가 제거될 수 있도록 수정하였습니다.

‘성별’ 컬럼에서는 수컷, 암컷 및 모름이라는 세 개의 버튼을 통해 설정할 수 있습니다. 이 버튼들 또한 세 개중 한 개만 선택될 수 있도록 조건문을 통해 구현하였습니다.

‘나이’와 ‘몸무게’컬럼에는 숫자만 입력될 수 있도록 InputType을 숫자만 입력되도록 (TYPE\_CLASS\_NUMBER)로 설정하였습니다.

‘털색’ 및 ‘특징’ 컬럼에는 텍스트가 입력될 수 있도록 EditText로 구현했습니다.

사진 등록을 위해 ‘사진등록’버튼을 누르게 되면 ‘카메라’, ‘갤러리’, ‘테스트 이미지 1~3’, ‘취소’ 항목을 AlertDialog에 리스트 형태로 띄도록 구현하였습니다. ‘카메라’항목을 누르는 경우 카메라로 연결되고, ‘갤러리’항목을 누르는 경우 갤러리로, 각각의 ‘테스트’항목을 누르는 경우에는 assets에 저장해둔 테스트이미지를 로드하도록, ‘취소’항목은 AlertDialog가 꺼지도록 구현했습니다. 각각의 방식으로 이미지를 로드하면, 해당 이미지에 강아지 또는 고양이가 존재하는지 객체 검출을 위한 동작이 실행되도록 ‘사진 검출’ 버튼이 뜨게 됩니다. 만약 사진에서 강아지나 고양이가 검출되지 않는다면 “이미지 재설정 필요”라는 문구가 AlertDialog를 통해 나타납니다. 이 경우 ‘완료’버튼을 누르고 ‘이미지 재설정’버튼을 눌러 다시 이미지를 로드하고 객체 검출을 실행해야 합니다. 사진 검출에 통과한 경우 하단에 어떠한 버튼도 존재하지 않게 됩니다.

게시글 작성 페이지의 하나의 항목이라도 비어있게 되면 글 등록하기 버튼을 누른 경우 “모든 항목을 채워주세요”라고 토스트메세지가 뜨며, 글 등록에 실패하게 됩니다.



게시글 작성 페이지의 모든 항목이 채워지고, 사진 검출에 성공한 경우 글 등록하기 버튼을 누르면 “글 등록 완료”라는 토스트 메시지가 뜨며, 게시판 페이지로 이동하게 됩니다. 해당 페이지에서 작성한 게시글이 뜨는 것을 확인할 수 있습니다. 동시에 모든 항목의 데이터를 PostDatabase의 post 테이블에 추가하게 되며, 이미지는 앱의 내부에 저장되고 이미지의 경로가 post 테이블에 추가됩니다. 이후 이미지를 불러오는 과정들에서는 해당 경로를 사용해 이미지를 로드하는 방법으로 진행됩니다.

[게시글 작성 페이지]

[지역 선택]

[종류 선택]

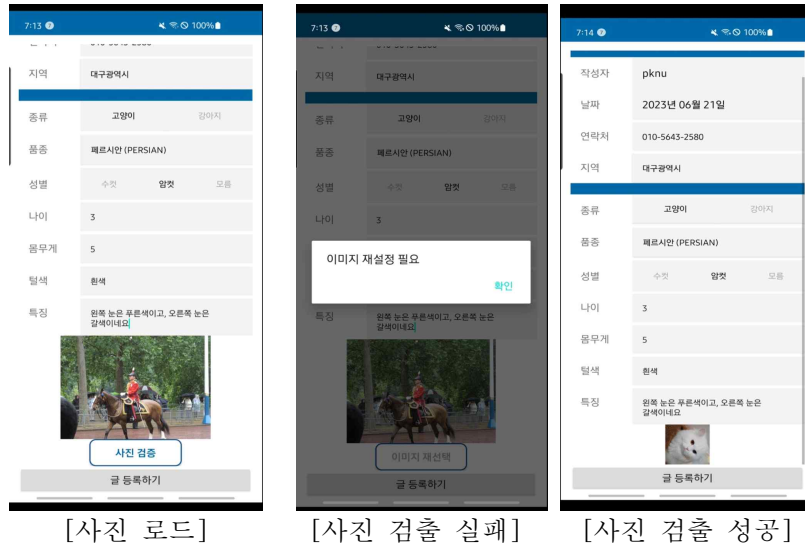
[품종 선택]

[성별 선택]

[나이/몸무게 작성]

[글 등록 실패]

[글 등록 성공]



[사진 로드]

[사진 검출 실패]

[사진 검출 성공]



[ 그림 3 ]



[ 그림 4 ]

## 2) 이미지 처리

게시글 작성 페이지에서 등록된 이미지의 객체 검출을 하는 과정에서 이미지는 학습된 yolo를 통해 객체 검출을 진행하게 됩니다. yolo는 assets 폴더에 추가해둔 체크포인트 'yolov5s.torchscript.ptl'를 로드하여 이미지의 객체 검출을 진행하고, 결과 고양이(15), 강아지(16)에 해당하는 경우 정상적으로 진행되도록 구현하였습니다. 로그캣에서 라벨 및 정확도, 바운딩박스 좌표를 확인할 수 있도록 코드를 작성해 뒀습니다. 고양이 및 강아지가 존재하는 경우 아래의 이미지와 같이 'Class Index', 'Score', 'Rect'값이 출력되지만, 검출되지 않은 경우에는 Class Index가 false로 뜨는 것을 확인할 수 있습니다.

```
I/System.out: Class Index: false
```

[ 강아지 혹은 고양이가 감지되지 않은 경우 로그캣 출력 결과 ]

```
I/System.out: Class Index: 16  
I/System.out: Score: 0.39588395  
I/System.out: Rect: Rect(70, 164 - 169, 391)
```

[ 강아지 혹은 고양이가 감지된 경우 로그캣 출력 결과 ]

## III. 결과 및 분석

### 1. 실험(시뮬레이션) 환경

- Android Studio (2021.3.1.)
- Galaxy S10
- Window 10
- CUDA 11.4
- cuDNN 8.2.4
- PyTorch 1.9

### 2. 실험 결과 및 분석

앱 제작 결과는 영상을 통해 첨부하였습니다.

앱을 완성한 뒤 직접 구현한 결과, 데이터 처리, 액티비티 및 xml 간의 처리가 정상적으로 동작하고, yolo모델도 잘 동작함을 확인할 수

있었습니다.

### 3. 한계 및 개선 방향

먼저, 이미지 처리 부분에서는 본론-2)의 [강아지 혹은 고양이가 감지된 경우 로그아웃 출력 결과]의 Score와 실험 결과 영상에서 고양이 사진 업로드 실패하는 부분 영상을 보시면 사용한 YOLOv5의 결과가 기대한 만큼 매우 좋지는 않은 것을 확인할 수 있습니다. 이 부분은 더 많은 데이터셋을 사용해 YOLO를 전이학습을 진행하는 방향으로 성능을 향상시킬 수 있지 않을까 생각했습니다.

다음으로는 앱의 기능 부분입니다. 게시글 등록 이미지에 고양이나 강아지가 포함되어있지만 하면 게시글 등록이 성공적으로 처리되도록 하였습니다. 앱을 제작하는 과정에서 똑같은 유기견을 입양 및 판매를 위해 게시글을 도배하는 부분도 큰 문제점이라고 생각하게 되었습니다. 일부 앱 및 서비스 사용자들도 이부분이 문제점이라고 생각하는 블로그 및 카페 글을 보기도 했습니다. 이 부분도 보완하기 위해 해당 이미지가 강아지이거나 고양이라는 것만 감지하는 것이 아닌, 이전에 등록된 게시글의 이미지들 중 똑같은 동물을 나타내는 게시글이 있으면 게시글 등록을 중지해 게시판 도배 방지 효과도 보일 수 있도록 개선하면 더욱 완성도 높은 앱이 될 수 있을 것 같다고 생각했습니다. 이후 객체 검출을 통한 동일한 객체 감지 혹은 OpenCV를 활용한 이미지 유사도 측정을 통해 이러한 부분을 개선할 수 있지 않을까 하고 생각하게 되었습니다.

## IV. 결론

### 1. 과제 수행 결과 요약 및 느낀 점

UserDatabase를 사용하여 사용자가 로그인 및 회원가입을 진행할 수 있도록 데이터를 처리하고, PostDatabase를 사용하여 사용자가 게시판의 게시글을 확인하고, 댓글을 추가하며, 게시글을 작성할 수 있도록 앱을 구현하였습니다. 기존에 있는 반려동물 앱과의 차별성은 사용자가 입력한 데이터를 게시글 작성 과정에서 수정할 수 없도록 하여 거짓으로 조작한 게시글 작성을 방지하고, 객체 탐지 모델인 yolo를 사용하여 게시글 작성 시 등록하는 이미지에 동물(강아지와 고양이)이 감지되어야만 사진을 등록할 수 있도록

하여 거짓된 사진 및 사용자에게 불편함을 주는 게시글 작성을 방지하는 효과를 주고자 하였습니다.

과제 수행 결과 데이터베이스와 액티비티, xml 및 어댑터들을 적절히 사용해 계획했던 대로 동작하도록 앱을 완성하였습니다.

자바 코드로 안드로이드스튜디오를 통해 아이디어부터 제작까지 직접 설계 및 개발하는 과정에서 많은 어려움이 있었지만, 해결하지 못할 줄 알았던 오류 및 에러들을 하나하나 수정해나가고 스스로 학습해 나가는 과정을 통해 안드로이드스튜디오에 대한 정확한 이해뿐만 아니라 자바 문법 및 자바 코드에 대해 많이 배울 수 있었습니다. 또한, 수업 시간에 조금씩 조금씩 해본 실습은 과제를 수행하는 것에 있어 도움이 많이 되었습니다. 이전에 다뤘던 내용이기에는 쉽다고 생각하고 넘기는 일이 많았는데 프로젝트를 진행하면서 아직 많이 부족하고 이론에 머물러있다는 것을 느끼게 되었습니다. 그래서 과제를 진행함에 있어 기존의 계획보다 다양한 기능을 추가해보면서 더 많은 것들을 실습해보자는 것에 의의를 둔 것 같습니다. 또한 파이썬 코드를 통해 모델을 로드하고 적용하는 것에 익숙해서 자바 또한 쉽게 처리할 수 있지 않을까 생각하고 시작한 프로젝트였는데 이 또한 자바의 문법 및 처리 과정이 파이썬과 달라 이 부분에 관해서도 공부하고 구현해보기까지의 과정을 통해 정말 많은 것을 배우고 실행할 수 있는 좋은 경험이 되었습니다.

## V. 참고문헌

- [1] J. Edmond, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015. 4, 5
- [2] 이지스퍼블리싱, Do it! 안드로이드 앱 프로그래밍