24년도 졸업논문

인공지능을 활용한
유해 콘텐츠 차단 애플리케이션

AI YOLO 객체 감지 기술을 활용한
유해 콘텐츠 사전 차단 및 보호자 보고 채팅 애플리케이션

한국공학대학교

전자공학부

임재혁 장민혁 조수연 진우열

인공지능을 활용한
유해 콘텐츠 차단 애플리케이션

AI YOLO 객체 감지 기술을 활용한
유해 콘텐츠 사전 차단 및 보호자 보고 채팅 애플리케이션

Harmful content blocking Application using AI

2024年 12月 1日

한국공학대학교

전자공학부

임재혁 장민혁 조수연 진우열

인공지능을 활용한
유해 콘텐츠 차단 애플리케이션

指導教授 장 승 관 (인)

이 論文을 졸업논문으로 提出함

2024年 12月 1日

한국공학대학교

전자공학부

임재혁 장민혁 조수연 진우열

2024년도

졸업논문

유해 콘텐츠 차단 애플리케이션

인공지능을 활용한

임재혁
장민혁
조수연
진우열

목차

그림 목차

요약

본 논문에서는 최근 어린 아이들의 유해 매체 노출 빈도가 증가에 따라 이를 예방하기 위한 애플리케이션 시스템을 제안한다. 본 시스템은 단말기 간 채팅 과정에서 유해한 콘텐츠(이미지, 영상 등)가 포함된 매체를 전송하려는 경우, 이를 사전에 감지하여 상대방에게 전송되지 않도록 차단하고, 해당 사실과 이미지를 지정된 보호자에게 실시간으로 보고하는 기능을 제공한다.

시스템 설계를 위해 하드웨어로는 Z Flip3를 사용하였으며, 소프트웨어 환경으로는 Android Studio와 PyCharm을 활용하였다. 사용 언어로는 Python과 Java를 적용하여 구현하였다.

특히, 유해 콘텐츠 감지를 위해 YOLOv5 엔진 기반의 AI 모델을 사용하였으며, 권총, 담배, 칼의 3가지 클래스를 효과적으로 감지할 수 있도록 모델을 학습시켰다.

결과적으로 본 논문에서는 이러한 시스템의 설계 및 구현 과정을 상세히 설명하며, 이를 통해 어린이의 유해 매체 접근을 효과적으로 차단할 방안을 제시하고자 한다.

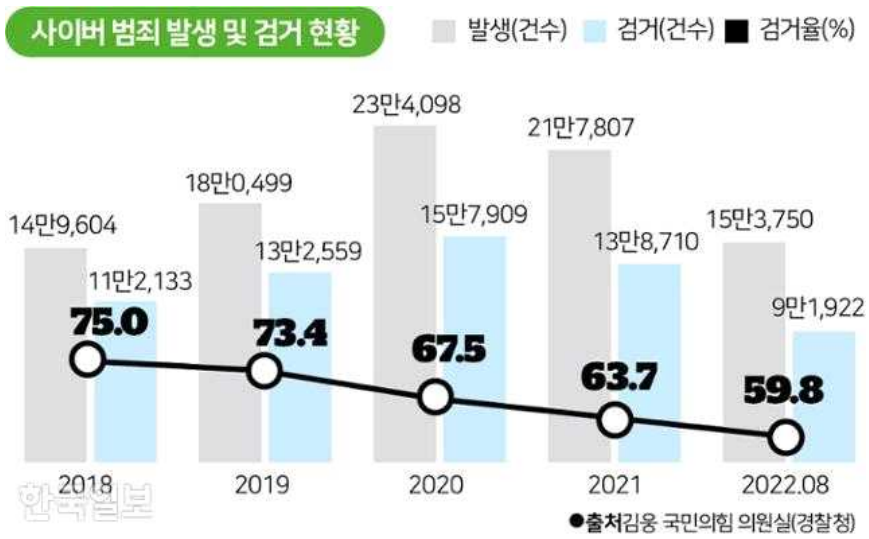
제 1 장 서론

제 1 절 연구 배경

4차 산업혁명이 불러온 정보 통신 기술의 발달은 우리의 삶을 편리하고 풍족하게 만들었다. 정보의 바다라고 불리는, 인터넷상에서 사회의 구성원은 SNS와 같은 연결망으로 긴밀히 연결되고 다양하고 많은 양의 정보를 손쉽게 주고받는다. 그러나 이 점을 악용하는 사례 또한 나날이 늘어가고 있다.

현재 누군가가 불특정 다수에게 선정적이고 폭력적인 이미지를 보낸다고 하더라도, 이를 사전에 차단하는 것은 불가능한 실정이다. 이러한 상황에 나이가 어린 아이와 청소년들은 더욱이 위험에 취약하다. 몸과 마음이 성장하는 시기의 아이들이 선정적이고 폭력적인 환경에 노출된다면 형성될 자아와 가치관에 심각한 영향을 끼칠 수 있다.

이러한 문제점을 해결하기 위하여, AI를 이용한 유해 콘텐츠 차단 시스템을 구상하였다. 이 시스템은 기존의 메신저 앱 등의 부가적인 기능으로 동작하며, 채팅방 내의 해로운 콘텐츠를 자동으로 탐지하여 차단한다. 해당 기능을 사용한다면 사용자는 이미지나 영상 내의 유해 콘텐츠에 노출되지 않고도 사전에 알아차리는 것이 가능할 것이다.



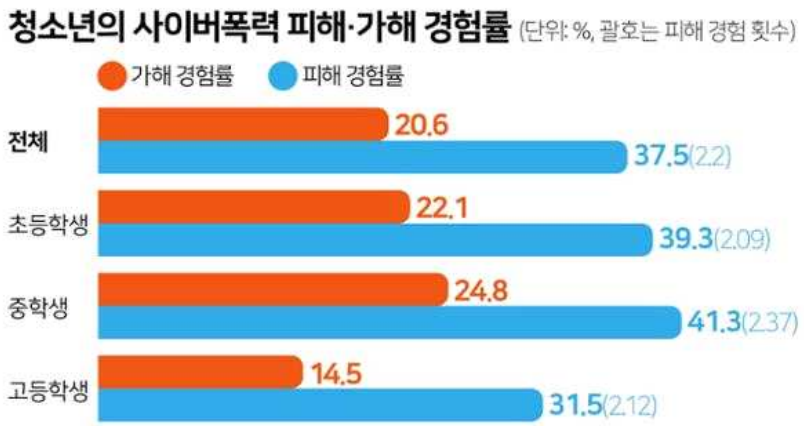
<그림 1> 사이버 범죄 발생 및 검거 현황

제 2 절 선행 연구

최근 한국에서 메신저 앱을 통한 사이버 범죄, 그중에서도 아동을 대상으로 한 성 착취 범죄와 딥페이크 합성물 범죄가 조명되었다. 2019년부터 2020년까지 이어진 'N번방 성 착취물 제작 및 유포 사건' 은 커다란 사회적 물의를 빚었으며 사이버 범죄에 대한 대비책 마련 요구가 이어졌다.

메신저 앱의 특성상 서버를 옮기며 추적이 어렵고 검열, 수사 요청에 비협조적인 경우가 많다. 가해자의 정보를 알아내기 힘들며 해외 이용자의 경우 실질적인 처벌은 거의 불가하다. 표현의 자유라는 명목 아래 여러 규제를 피해 가며 성 착취, 딥페이크, 마약 거래 등 다양한 범죄에 악용되고 있다.

여성가족부가 발표한 '아동·청소년 대상 성범죄 판결 분석 보고서' 에 따르면 온라인에서 이루어진 아동 성 착취 범죄 가해자의 약 60%가 '인터넷 채팅 등을 통해 아는 사람'이었다. 그리고 방송통신위원회의 2023년 사이버폭력 실태조사에서 아동들은 사이버 공간에서 이루어진 범죄에 대하여 '가해자 차단, 나의 정보 변경'으로 대응한 경우가 42%로 가장 많았다. 하지만 단순한 차단이나 정보 변경은 가해자가 피해자의 정보를 탐색하여 다시 보복하기가 쉬우며 신체 사진이나 영상을 촬영하도록 협박하는 경우도 적지 않다. 이처럼 아이들은 온라인상에서 채팅앱을 통해 쉽게 범죄에 노출되며 간단한 방어 수단조차 전무한 상황이다.



<그림 2> 청소년 사이버폭력 경험률

제 3 절 기대효과

선행된 연구에서 알아본, 아동을 대상으로 하는 사이버 범죄를 근절하기 위해 AI를 활용한 유해물 차단 기능을 메신저 앱에 추가하는 시스템을 제작했다. 우선 개인 간의 연락 내에서 기존의 검열 방식을 사용한다면 개인 정보, 사생활에 대한 침해가 우려된다. 하지만 본 논문에서 구상한 시스템을 이용한다면 개인 정보 제공 없이, 단지 유해물에 대한 탐지만이 이루어지기에 신뢰할 수 있는 방어 수단이 될 것이다. 또한 이 기능은 기존의 메신저 앱에 추가되는 부가 기능으로 구현되는 점을 염두 하여 구상하였기에 다양한 앱에 추가되어 범용성이 높다.

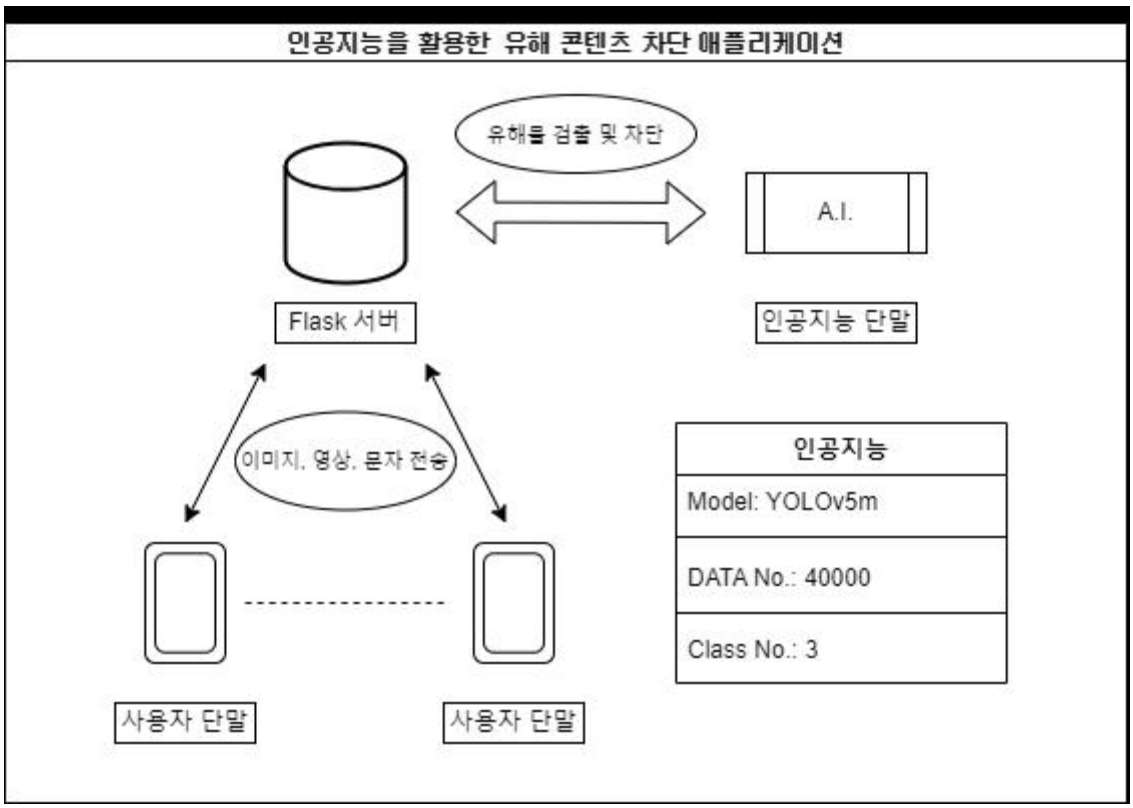
실제 사용자는 자신에게 오는 메시지 내 유해 콘텐츠가 제외된 사진이나 영상을 받게 된다. 만약 해당 기능을 자녀의 핸드폰에 설정하는 경우, 보호자는 자녀가 유해물에 노출되는 상황을 피할 수 있으며 누가 어떤 사진을 보내왔는지를 알 수 있으므로 실제 피해로 이어지기 전 예방할 수 있다.

그리고 이 기능을 메신저 앱에만 국한되는 것이 아니라 디스플레이 전체에 대한 검열을 등록한다면 화면에 표시되는 불쾌한 광고나 사이트에 대한 노출을 방지할 수 있다. 더 나아가 이 기능을 단지 유해물이 아닌, 개인 정보 제공을 동의받아 자신의 얼굴을 등록한다면 해당 메신저 내에서 자신의 얼굴이 도용되는 것 또한, 미리 방지할 수 있도록 하는 등 다양한 활용 방안이 기대된다.

제 2 장 본론

제 1 절 시스템 구상도

본 시스템에서는 단말기 간 이미지, 영상, 문자 등의 전송 과정에서 Flask 서버를 중개로 활용하여, 전달되는 콘텐츠 내부에 유해 요소가 포함되어 있는지 YOLOv5 AI 모델로 검사한다. 만약 유해 요소가 발견되면, 전송을 즉시 중단하고, 사전에 설정된 보호자에게 실시간으로 해당 내용을 보고한다.

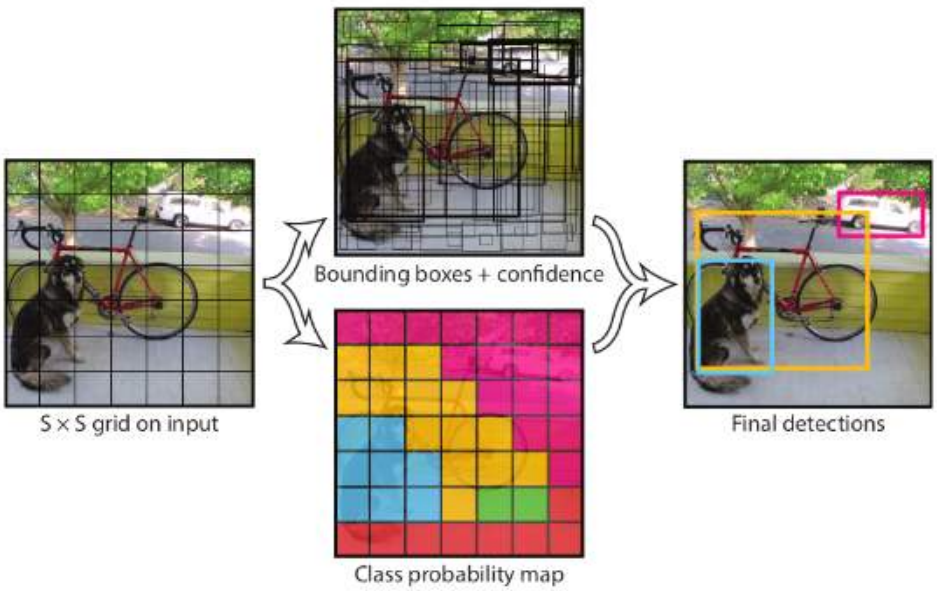


<그림 3> 시스템 구상도

제 2 절 YOLO 학습 모델

(1) 선정 이유

해로운 객체를 검출하기 위해, 본 시스템에서는 YOLOv5를 사용하였다. YOLO는 You Only Look Once의 약자로 Object Detection One-Step 분야의 대표적인 모델이다. 이미지를 한 개의 모델을 사용하여 빠른 속도와 우수한 성능을 보장하여, 본 시스템에서 구현하고자 하는 영상 전송 중 서버에서 파일을 분석하여, 그 영상의 유해물이 포함되어 있는지 실시간으로 탐지 후 그에 대한 응답을 구현하기에 가장 적합하다고 판단되어 YOLO 모델 중 YOLOv5를 사용하였다.

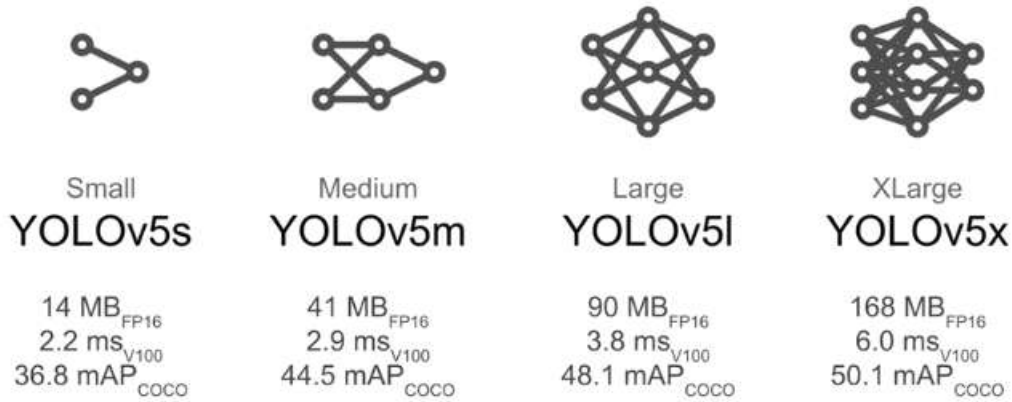


<그림 4> YOLO 모델 구조 예시

Image Detection 대상 유해물은 총(권총), 날붙이(칼), 담배로 지정하였다. 선정 이유는 국내 영화와 같은 많은 영상매체에서 총기나 날붙이를 사용하는 장면들이 많고, 담배 또한 마찬가지로 본 시스템에서 대상으로 하는 아동 혹은 청소년에게 안 좋은 영향을 미칠 것으로 생각하여 위의 3가지 요소를 선정하였다.

(2) 데이터 셋

YOLO에서는 Pretrained Checkpoints가 존재하는데, 640픽셀의 이미지에서 모델의 크기가 커질수록 모델의 성능이 올라가, 정확도는 높아지지만, 시간이 더 오래 걸리기 때문에 여러 모델을 테스트하였다.

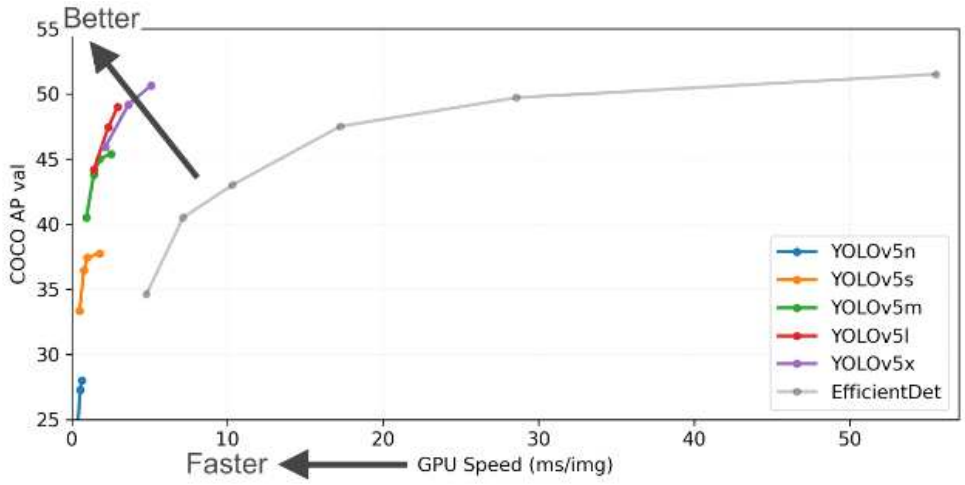


<그림 5> YOLO 모델의 다양한 크기

초기모델을 만들기 위해 처음에는 인터넷에서 사진을 긁어 라벨링을 하며 진행하였고, 더 많은 양을 확보하기 위해 Roboflow의 데이터 셋을 포함해 약 26000장의 사진으로 small, nano 사이즈 모델 위주로 다양한 batch와 100번의 epochs를 수행하였다.

하지만 원하는 성능만큼 모델이 학습되지 않아 사진 수를 약 4만 장으로 늘리고, custom 데이터 셋의 품질을 높이며, 학습에 무리가 되지 않는 batch와 overfitting이 되지 않을 정도의 학습 수를 고려하여 진행하였다.

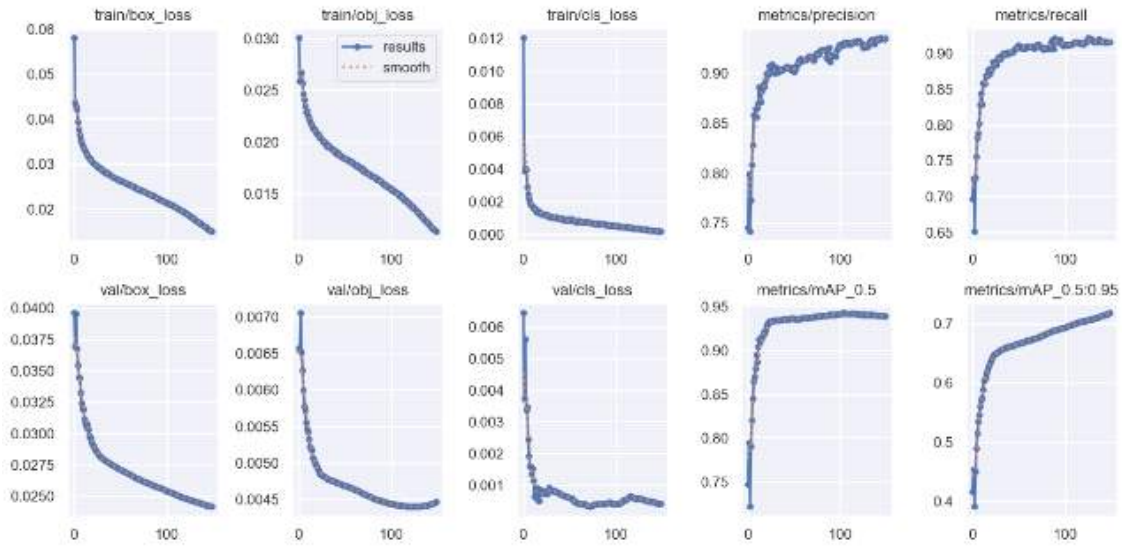
또한, 성능 향상을 위해 유해물의 탐지에 문제가 없으며 탐지 시간이 크게 차이가 나지 않는 YOLOv5m weight를 사용하였다.



<그림 6> YOLOv5 640 image

(3) 성능 지표

완성된 데이터 셋과 변수들을 바꿔가며, 최종적으로 각 요소의 predicted가 0.9에 근접하고 종합적으로 mAP (0.93935 at IoU 0.5) 의 수치를 가지는 모델을 만들었다.



Class	Images	Instances	P	R	mAP50	mAP50-95:
all	4424	5449	0.934	0.916	0.939	0.718
pistol	4424	2704	0.965	0.959	0.971	0.807
knife	4424	1150	0.887	0.856	0.887	0.572
ciga	4424	1595	0.951	0.932	0.96	0.774

<그림 7> YOLOv5 성능 지표 및 요약

제 3 절 애플리케이션 개발

(1) 소프트웨어 개발 환경

SNS는 각종 유해 요소에 노출되기 쉬운 환경이다. 본 시스템은 이러한 환경에서 유해 요소 탐지 기능의 효능을 효과적으로 시연하기 위해 채팅 애플리케이션을 개발하였다. 개발 도구로는 안드로이드 전용 애플리케이션 제작을 위한 공식 개발 환경인 안드로이드 스튜디오를 사용하였으며 본 시스템의 구현에는 기본적으로 지원되는 언어인 Java를 사용하였다.

<그림 8> 애플리케이션 개발 환경

```
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import android.Manifest;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.database.*;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.TimeUnit;
import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.OkHttpClient;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
import retrofit2.Callback;
import retrofit2.Response;
```

<그림 9> 이용한 Main Activity 라이브러리

<그림 10> Firebase 데이터 읽기

addChildEventListener() 함수를 사용해 실시간으로 새 메시지를 수신하였다. 또한, onChildAdded() 함수는 새 메시지가 추가될 때마다 호출되며, 새로운 메시지를 ChatMessage 객체로 변환한다. 그 후에 chatAdapter.notifyDataSetChanged() 함수에서, 수신된 메시지를 리스트에 추가하고 RecyclerView를 업데이트하여 채팅방의 스크롤을 조정한다.

제 4 절 서버 통신 방식

(1) Flask

Flask는 Python 기반으로 작성된 마이크로 웹 Framework 중 하나로, 간단한 서버를 만드는데 특화된 framework이다. Flask는 경량성과 확장성이 용이하여 필요한 기능만 사용하여 간단하게 서버를 구현하기 좋고, YOLOv5의 구동과 실시간으로 수정하기 좋다는 장점 때문에 본 논문에서는 YOLOv5를 통해 객체 탐지 결과를 서버에서 처리하고, 클라이언트로 응답하기 위해 Flask를 사용하여 Python 기반의 서버를 구현하였다.

클라이언트에서 영상을 전송하면 서버는 전송된 영상을 YOLOv5로 처리한 후, 탐지 결과에 따라 아래와 같은 기능을 수행한다. 만약 해로운 콘텐츠가 탐지되지 않는다면, 해당 영상이 정상적으로 전송되게 JSON 응답을 전송한다. 반대로, 일정 횟수 이상 발견된다면 발견된 유해 매체의 사진을 Twilio Python api를 이용하여, 해당 유해 매체 프레임 사진에 접근할 수 있는 경로를 가진 URL을 문자로 해당 보호자에게 전송하며, 어떠한 물체가 감지되었는지 알리고, 해당 영상이 전송되지 않게 클라이언트에 JSON 응답을 보낸다.

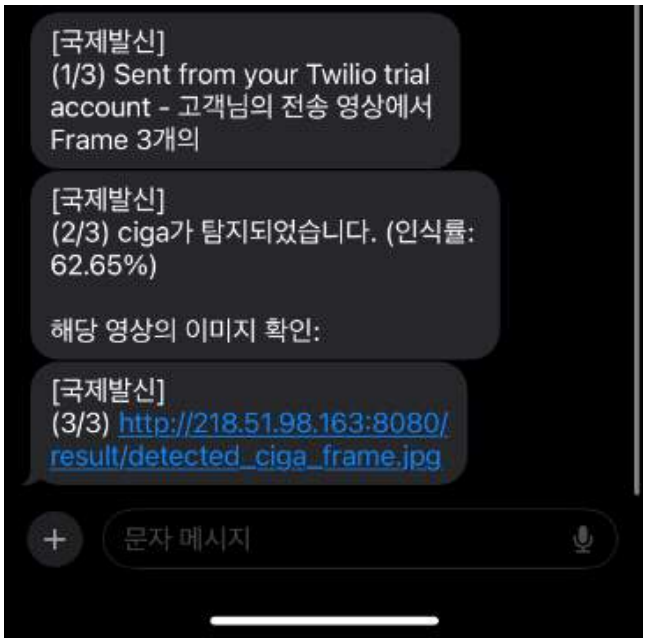
<그림 11> 서버 개발 환경

(2) 통신 세부 과정

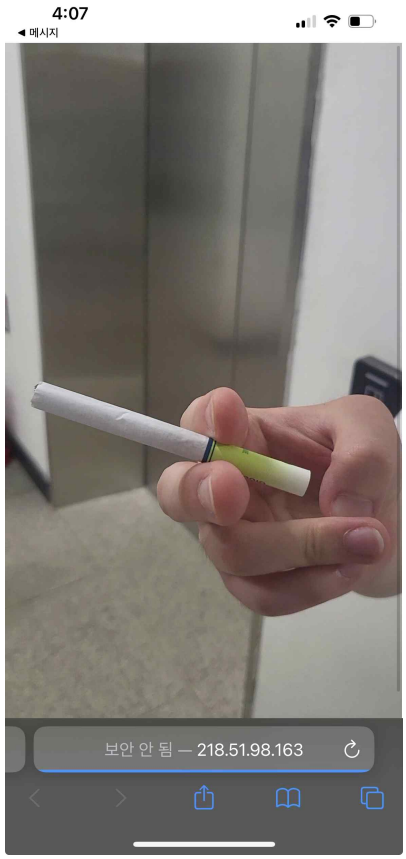
> **동영상 업로드**: 사용자가 전송한 동영상 파일은 Flask 서버에 저장된다. 서버는 OpenCV를 이용하여 동영상을 프레임 단위로 분리하고, YOLOv5 모델을 통해 각 프레임을 분석하여 유해 요소를 탐지한다.

> **유해 요소 탐지**: YOLOv5 모델은 클래스 레이블과 신뢰도를 반환하며 사전 정의된 임곗값 이상인 경우, 탐지된 것으로 인식하여 해당 프레임과 레이블, 신뢰도를 저장한다.

> **탐지 결과 처리 및 보고서 전송**: 유해 요소가 3프레임 이상 탐지된 경우, 해당 동영상 전송을 차단하며 탐지된 프레임과 레이블, 신뢰도를 포함한 보고서를 Twilio Python api를 통해 보호자 등 설정된 사용자에게 전송한다. 탐지가 이루어지지 않은 경우 동영상 URL을 반환하여 채팅에 참여한 사용자에게 전송된다.



<그림 12> 서버 보고서



<그림 13> 보고서 링크 접속 결과

If문을 활용하여 허용된 동영상 확장자 파일이 아니라면 에러를 반환하였다. 허용한 확장자의 동영상이라면 secure_filename() 함수로, 생성된 고유의 폴더에 저장되도록 하였으며, 저장된 동영상은 process_video() 함수로 영상 분석 및 객체 탐지를 수행하게 된다. 그 후에, 동영상 처리가 완료되었다면 try:문을 활용하여 저장된 비디오 파일을 삭제한다.

```python
@app.route('/upload', methods=['POST'])
def upload_video():
    if 'video' not in request.files:
        return jsonify({'error': 'No file part'}), 400
    file = request.files['video']
    if file.filename == '' or not allowed_file(file.filename):
        return jsonify({'error': 'Invalid file'}), 400

    filename = secure_filename(file.filename)
    file_path = os.path.join(UPLOAD_FOLDER, filename)
    file.save(file_path)

    results = process_video(file_path)

    # 처리 후 영상 파일 삭제
    try:
        os.remove(file_path)
    except Exception as e:
        print(f"Error deleting file: {e}")

    return jsonify(results)
```

<그림 14> 동영상 업로드 및 처리

업로드된 비디오 파일은 OpenCV로 로드하여, 프레임 단위로 읽는다. 현재 시간을 기준으로 고유한 비디오의 id를 생성한 후, 비디오의 프레임을 저장할 폴더를 생성하며, 모델은 class_names에 포함된 “담배”, “권총”, “칼”을 객체 탐지한다. 유해 요소가 탐지될 때마다 detection_count가 증가하며 3이 되면 동영상 전송을 취소한다.

```python
def process_video(video_path):
    cap = cv2.VideoCapture(video_path)
    detection_count = 0
    detected_frames = []
    detected_class_names = set()
    frame_count = 0
    video_id = int(time.time())  # 비디오 id
    video_folder = f"detected_video{video_id}"
    os.makedirs(video_folder, exist_ok=True)

    class_names = {0: "담배", 1: "권총", 2: "칼"}
```

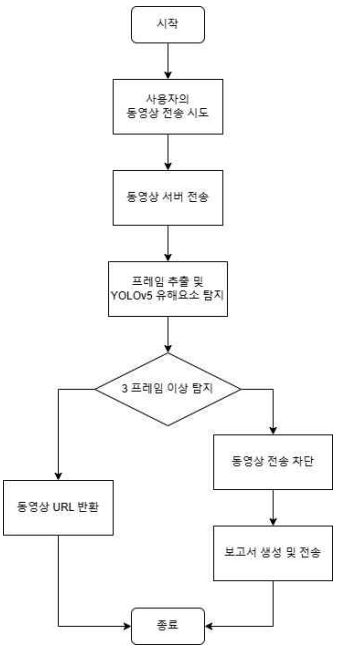
<그림 15> 유해 요소 탐지

유해 요소가 탐지되었다면, 해당 객체의 레이블이 포함된 파일명을 생성하고 프레임을 JPEG로 저장한다. save_detection_info() 함수는 탐지된 객체에 대한 정보를 텍스트 파일로 저장한다. 포함된 정보는 순서대로 텍스트를 저장할 폴더 경로, 탐지 순번, 탐지된 객체의 이름, 객체 탐지의 신뢰도이다.

```
1 usage
def save_detected_frame(frame, folder_path, class_name):
    output_filename = f"detected_{class_name}_frame.jpg"
    output_path = os.path.join(folder_path, output_filename)
    cv2.imwrite(output_path, frame)
    print(f"저장된 프레임: {output_path}")
    return output_path


1 usage
def save_detection_info(video_folder, detection_number, class_name, confidence):
    info_path = os.path.join(video_folder, f"detected_video{detection_number}_txt")
    with open(info_path, 'w', encoding='utf-8') as f:
        f.write(f"Frame {detection_number}: {class_name}가 탐지되었습니다. (인식률: {confidence:.2%})\n")
```

<그림 16> 탐지 결과 처리



<그림 17> 서버 통신 구조

제 3 장 결론

제 1 절 개발 결과

최근 스마트폰 보급이 증가하면서, 나이가 어린 아이들이 해로운 매체에 노출될 가능성이 높아지고 있다. 이러한 매체는 정신적으로 아직 미성숙한 아이들에게 부정적인 영향을 미칠 수 있어 심각한 문제로 대두된다. 이러한 문제를 예방하고자 유해 이미지를 사전에 차단하고 보호자에게 이를 보고하는 시스템을 개발하게 되었다.

본 논문에서는 AI YOLO Engine 모델을 활용하여 유해 이미지 및 영상매체를 차단하고 보호자 보고 기능을 갖춘 채팅 애플리케이션의 개발 과정과 결과를 다룬다. 프로젝트에서는 권총, 칼, 담배의 세 가지를 유해 이미지로 정의하고, 이에 대한 데이터 셋을 구축한 뒤, 해당 클래스들을 효과적으로 감지할 수 있는 YOLO 기반 객체 감지 모델을 개발하였다.

초기 개발 목표는 AI 객체 감지 모델뿐만 아니라 언어 감지 모델인 WHISPER를 활용하여, 영상 속 유해 이미지는 모자이크 처리하고, 유해 음성은 묵음 처리하며, 분석된 내용을 자막으로 삽입해 영상을 재구성하는 것이었다. 특히, 묵음 처리된 구간은 * 표시를 통해 시각적으로 표현되도록 설계하였다.

그러나 초기 설계안이 차별성이 부족하다는 피드백을 바탕으로 프로젝트 방향을 수정하였다. WHISPER 모델 기반 기능 대신, 채팅앱 환경을 새롭게 구성하여, 해로운 이미지나 영상이 전송될 경우 이를 사전에 감지하고 차단하며, 해당 사실을 보호자에게 즉시 보고하는 방식으로 개발하였다.

또한, 객체 감지 모델의 특성상 낮은 성능의 하드웨어(예: 라즈베리파이)에서 실행 효율이 저하된다는 점을 고려하여, 스마트폰 기반 애플리케이션 환경에서 동작하도록 설계하였다. 이를 더 보완하기 위해 모델 감지 과정은 서버를 이용해 데스크톱 환경에서 수행되도록 구현하였다. 이로써 30초 분량의 영상 처리 시간을 기존 약 180초에서 10초로 단축하여 시스템 전체의 효율성을 극대화할 수 있었다.

이 과정에서 서버와 애플리케이션 간 통신을 위해 FIREBASE와 FLASK를 활용하여 안정적이고 빠른 데이터 전송 및 처리가 가능하도록 설계하였다.

제 2 절 활용 분야

현재 감지 가능한 클래스는 권총, 칼, 담배의 3가지로 제한되어 있지만, 향후 데이터 셋을 확장하여 더 많은 객체를 감지할 수 있는 모델로 업그레이드하면, 다양한 환경에서 해당 채팅 애플리케이션의 활용성을 더욱 넓힐 수 있을 것이다.

> 가정 내 어린이 보호

스마트폰 사용이 보편화된 가정 환경에서, 어린이가 유해한 이미지나 영상을 접하는 것을 예방하는 데 효과적으로 활용될 수 있다. 특히, 보호자에게 실시간으로 유해 매체 전송 시도를 알림으로써 부모와의 신뢰를 강화하고, 아이의 안전한 디지털 환경을 제공할 수 있다.

> 학교 및 교육기관

학교나 학원에서 제공하는 디지털 학습 도구나 플랫폼에서도 해당 시스템을 적용하면, 학생들이 유해 콘텐츠에 노출되는 것을 방지할 수 있다. 이를 통해 학습 환경의 질을 높이고, 교육기관이 아이들의 디지털 안전을 더욱 적극적으로 관리할 수 있다.

> 기업 및 조직의 내부 채팅 관리

기업이나 조직 내부의 채팅 플랫폼에 시스템을 적용하면, 부적절한 이미지나 영상의 전송을 방지하여 업무 생산성을 저해할 수 있는 요소를 사전에 차단할 수 있다. 특히, 민감한 자료가 포함된 비즈니스 환경에서 데이터 유출이나 부적절한 콘텐츠 유통을 막는 데 유용하다.

> 공공기관 및 사회적 보호 프로그램

아동 보호 관련 공공기관이나 NGO에서도 이 시스템을 활용하여 디지털 환경에서 아동을 보호하는 데 기여할 수 있다. 특히, 온라인 환경에서의 유해 콘텐츠 유통을 막아 아동 대상 범죄 예방에 활용할 수 있다.

> 커뮤니티 플랫폼

SNS나 메신저와 같은 커뮤니티 기반 플랫폼에서도 적용할 수 있다. 유해 콘텐츠의 사전 차단 기능을 통해 사용자 간 신뢰를 높이고, 플랫폼 내에서 건강한 소통 환경을 조성할 수 있다.

제 3 절 문제점 및 보완점

(1) 문제점

> 서버 통신 과정에서의 성능 문제

본 시스템은 서버를 활용한 모델 감지 구조를 기반으로 설계되었기 때문에, 통신 과정에서 네트워크 오류가 발생하면 감지 기능이 원활하게 동작하지 않을 수 있다.

> 모델의 정확도 제한

객체 감지 모델의 정확도가 99.9%에 도달하지 못한 점이 문제로 지적될 수 있다. 특히 '칼' 클래스의 경우 정확도가 떨어지기에, 유사한 형태를 가진 객체(예: 금속 막대, 특정 도구 등)와 혼동되어 잘못된 판단을 내릴 가능성이 있다.

> 억압성 문제

채팅창에서 유해 이미지나 영상 전송을 차단하는 기능은 본래의 취지에서는 긍정적일 수 있으나, 사용자의 관점에서 채팅의 자유로운 소통이 제한된다고 느끼게 할 가능성이 있다. 이는 앱의 사용자 경험(UX)을 저하시킬 수 있다.

(2) 보완점

> 데이터셋 확장을 통한 정확도 향상

모델의 정확도를 높이기 위해 다양한 환경에서 촬영된 데이터를 포함한 데이터 셋을 구축하여 '칼'과 같은 유사 객체를 더욱 정확히 구분할 수 있도록 학습시켜야 한다.

> 서버 이중화 및 안정성 강화

서버를 통한 통신 오류를 최소화하기 위해 이중화(redundancy) 구조를 도입하여, 한 서버가 다운되더라도 다른 서버가 백업 역할을 수행할 수 있도록 설계해야 한다. 이를 통해 시스템의 안정성과 사용자 신뢰도를 동시에 향상시킬 수 있다.

> 유저 친화적인 차단 시스템 설계

차단 기능의 억압성을 완화하기 위해 사용자에게 차단된 이미지나 영상의 내용을 수정하거나 재검토할 수 있는 선택권을 제공하는 기능을 추가할 수 있다. 예를 들어, 보호자나 관리자의 허가를 요청할 수 있는 "차단 해제 요청" 버튼을 제공하는 방식이 있다.

> 모바일 환경 최적화

스마트폰의 하드웨어 성능 한계를 극복하기 위해 모델을 경량화하거나, 서버와의 통신이 끊겼을 때도 최소한의 객체 감지 기능을 스마트폰에서 수행할 수 있도록 경량화된 백업 모델을 추가하는 방안을 고려할 수 있다.

참고 문헌

[1] 개인정보 철통보안 - 은밀한 범죄 온상… 두 얼굴의 ‘비밀 메신저’[10문10답]

▶ https://www.munhwa.com/news/view.html?no=2024090301032107080001

[2] 아동 성착취 범죄 방지 위한 온라인 안전 장치 만들자

▶ https://www.ibabynews.com/news/articleView.html?idxno=122660

[3] 아동·청소년 성범죄 60%가 '아는 사람'…'채팅앱' 접촉 많다

▶ https://www.news1.kr/society/women-family/5396016

[4] YOLOv5 RESEARCH

▶ https://www.researchgate.net/figure/YOLOv5-different-model-sizes-where-FP16-stands-for-the-half-floating-point-precision_fig3_354846944

[5] Ultralytics YOLO GUIDE

▶ https://docs.ultralytics.com/ko/guides/

[6] Github of YOLOv5

▶ https://github.com/ultralytics/yolov5

[7] You Only Look Once. YOLO

▶ https://blog.nerdfactory.ai/2021/05/06/You-Only-Look-Once.-YOLO.html

[8] FIREBASE.GOOGLE

▶ https://firebase.google.com/?hl=ko

[9] Github of Flask

▶ https://github.com/pallets/flask

[10] Github of Twilio-Python

▶ https://github.com/twilio/twilio-python