목 차

1. 과제 배경 및 목표

1.1 과제 배경

간접광고란 제품이나 기업을 직접적인 방법 대신 간접적으로 명시하는 광고를 총칭한다. 현재 한국의 방송법 시행령 제59조의3 간접광고에서는 방송에서의 간접광고의 지침을 제공하고 있다. 또한 최근 방송에서의 간접광고의 빈도는 계속해서 높아지고 있다.

대부분의 간접광고는 광고주와 방송사 사이에 대행사가 간접광고를 중재하는 형태이다. 광고주는 대행사 사이에서는 의뢰를 통해 간접광고를 기획하며, 대행사와 방송사는 직접적인 촬영을 진행한다. 그리고 최종 광고의 결과물에 대해 광고주는 검증을 거쳐 계약에 명시된 조건으로 촬영이 진행되었는지 검사할 필요가 있다.

1.2 과제 목표

우리는 증가하는 간접 광고의 수에 따라 원활한 검증을 위해 자동화된 검사 시스템을 개발하기로 했다. 광고가 촬영된 방송과, 광고에 사용된 제품이나 로고 이미지 입력으로 넣어 광고가 실제 행해진 시간을 출력으로 받을 수 있게 구현할 계획이다. 이를 개발한다면 광고주와 대행사 사이에서의 빠른 검증이 가능해지므로 시간에 대한 효율성을 확보할 수 있다.

2. 기술 조사

2.1 YOLO

YOLO는 You Only Looks One의 줄임말로 사진에서 하나의 신경망과 한 번의 계산으로 객체를 찾아낼 수 있는 객체 검출 모델이다. 사진에서 n X n 크기의 grid로 나눈 후, 각 grid에 대해 찾을 객체가 있을 확률을 각각 구하는 regression 방식이다. 현재 YOLO는 버전 5까지 개발되었고, 원래의 개발자 Joseph Redmon은 버전 3까지 담당하였다. 버전 4까지는 리눅스 환경의 Darknet에서 가능하며, 버전 5부터는 딥러닝 라이브러리인 Pytorch로 구현되었다.

2.2 SSD

Single Shot Multibox Detector는 기존의 객체 검출 모델의 문제점이 단 하나의 feature map으로 학습하기 때문이라고 생각해 이를 해결한 모델이다. 전의 모델들은 하나의 feature map에서 학습을 했기 때문에 객체의 크기에 따라 정확도가 달랐다. SSD는 이를 위해 해상도 별로 feature map을 만들어 학습을 진행했고, 이에 따라 좋은 정확도를 가질 수 있었다.

2.3 RetinaNet

RetinaNet은 기존의 1-stage detection 모델이 가지고 있던 객체와 배경 클래스의 불균형을 해결하기 위해 전의 loss 함수를 대신하여 Focal Loss라는 새로운 함수를 사용했다. Focal Loss는 기존의 Loss function에서 scale factor를 곱해 예측하기 쉬운 example에는 더욱 낮은 loss 값을, 반대로 예측하기 어려운 example에는 더욱 높은 Loss 값을 가진다. 이렇게 함으로써 Focal Loss는 전체적인 Loss를 낮추어 더 좋은 성능을 보이게 한다.

2.4 성능 비교

각 객체 검출 모델의 성능을 비교하기 위해 모델의 각 논문에서 기술된 평가 목록의 내용을 토대로 비교했다. 논문의 평가에서 사용된 Dataset 은 객체 검출 성능 평가에 주로 사용되는 COCO Dataset 을 사용했다. 평가 지표로는 mAP와 AP50이 존재한다. mAP는 AP의 mean 값이며, AP는 기존의 precision-recall 그래프를 알기 쉽게 숫자로 만든 지표이고, AP에 붙는 숫자는 threshold 값을 의미한다. AP와 mAP 값이 클수록 그 알고리즘은 객체 검출에서 우수한 성능을 보인다.

Table 9: Comparison of the speed and accuracy of different object detectors on the MS COCO dataset (test-dev 2017). (Real-time detectors with FPS 30 or higher are highlighted here. We compare the results with batch=1 without using tensorRT.)

Method	Backbone	Size	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv4: Optimal Speed and Accuracy of Object Detection									
YOLOv4	CSPDarknet-53	416	54 (P)	41.2%	62.8%	44.3%	20.4%	44.4%	56.0%
YOLOv4	CSPDarknet-53	512	43 (P)	43.0%	64.9%	46.5%	24.3%	46.1%	55.2%
YOLOv4	CSPDarknet-53	608	33 (P)	**43.5%**	**65.7%**	**47.3%**	**26.7%**	**46.7%**	53.3%

그림 1 - COCO dataset에서의 YOLOv4의 평가 지표 [1]

Method	VOC2007 test		VOC2012 test		COCO test-dev2015		
	07+12	07+12+COCO	07++12	07++12+COCO	trainval35k		
	0.5	0.5	0.5	0.5	0.5:0.95	0.5	0.75
SSD300	74.3	79.6	72.4	77.5	23.2	41.2	23.4
SSD512	76.8	81.6	74.9	80.0	26.8	46.5	27.8
SSD300*	77.2	81.2	75.8	79.3	25.1	43.1	25.8
SSD512*	**79.8**	**83.2**	**78.5**	**82.2**	**28.8**	**48.5**	**30.3**

Table 6: **Results on multiple datasets when we add the image expansion data augmentation trick.** SSD300* and SSD512* are the models that are trained with the new data augmentation.

그림 2 - COCO dataset에서의 SSD 평가 지표 [2]

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

Table 2. **Object detection** *single-model* results (bounding box AP), *vs.* state-of-the-art on COCO `test-dev`. We show results for our RetinaNet-101-800 model, trained with scale jitter and for 1.5× longer than the same model from Table 1e. Our model achieves top results, outperforming both one-stage and two-stage models. For a detailed breakdown of speed versus accuracy see Table 1e and Figure 2.

그림 3 - COCO dataset에서의 RetinaNet 평가지표 [3]

위의 표 이미지는 각각 YOLOv4, SSD, RetinaNet에 대한 결과이다. YOLOv5는 현재 논문이 없지만 대신 공식 사이트에서 YOLOv4와 평가 지표를 비교한 기록이 있다.[4] 버전 4와 비교했을 경우 큰 정확도의 차이는 없지만, 영상 분석에 있어 높은 속도를 내고 있음을 말하고 있다. 이를 유의해 mAP와 AP50의 점수를 비교한 결과 일반적으로 SSD < RetinaNet < YOLOv5의 순서로 성능을 가짐을 알 수 있었다. 이에 따라 프로젝트에서는 성능이 좋은 YOLOv5의 학습부터 진행할 계획이다.

3. 세부 과제 내용

3.1 순서도

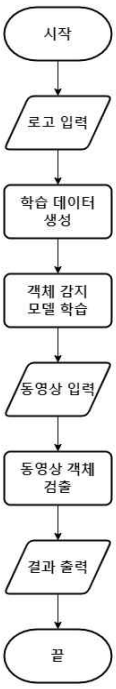

그림 4 - 개발의 순서도

개발할 서비스의 전체적인 순서도는 위의 이미지와 같다. 입력으로 받은 로고 이미지를 이용해 무작위로 생성된 훈련 데이터를 생성한다. 그다음 이 데이터들을 이용해 이미지 객체 검출 모델을 학습하여 로고의 전용 모델을 생성한다. 그 후 생성된 모델과 검증하고 싶은 영상을 통해 간접 광고에 사용된 시간을 타임스탬프의 기간으로 제공한다.

3.2 학습 데이터 생성

사용자에게 임의의 이미지 data를 검출할 수 있어야 하므로, 객체 검출 모델 학습 전 입력으로 받은 로고 이미지를 전처리하여 일정 이상의 train data를 제작해야 한다. 여러 상황과 방향에서 로고가 보일 것임을 고려해, 사진의 전처리는 3X3 크기의 Homography matrix를 이용해 처리할 계획이다.

Homography matrix의 8개의 변수를 조정하면, 크기, 회전, 기울기, 관점의 조절이 가능하다. 또한 각 사진의 pixel에 같은 세기를 더해주어 밝기의 조절이 가능하다. 전처리한 로고의 이미지와, 이를 붙일 임의의 배경 이미지를 합성해 학습 데이터를 자동 제작할 계획이다.

3.3 객체 검출 학습

학습할 Dataset이 만들어진 후 이를 학습해야 한다. 또한 학습 모델 개발에서는 하나의 모델만이 아닌 여러 모델을 고른 후 각각 성능을 비교할 예정이다. 사용할 모델의 후보는 YOLOv5, SSD, RetinaNet이 있다. 각각 정확도와 속도를 비교 후 가장 목적에 부합하는 성능을 가진 모델을 최종 선별할 계획이다.

딥러닝 라이브러리로는 Pytorch를 사용할 계획이다. 페이스북 AI 연구소가 개발한 Pytorch는 컴퓨터 비전, 딥러닝, 자연어 처리 등에 사용되고 있으며, 현재 다른 딥러닝 라이브러리 Tensorflow와 함께 높은 사용률을 보인다. 사용할 모델 중 YOLOv5가 Pytorch로 구현되었기도 하고, 또한 Tensorflow에 비해 배우기가 쉬우므로 선택했다.

3.4 웹 프레임워크

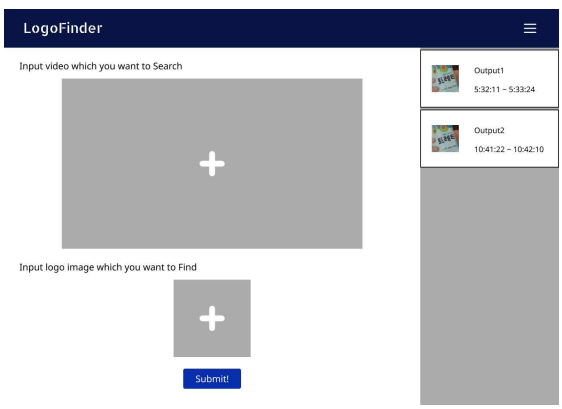

그림 5 – 웹 사이트의 청사진

서비스의 원활한 제공을 위해 위의 서비스를 웹 페이지를 제작할 계획이다. 다른 작업에 비해 상대적으로 부담이나 부하가 적기 때문에 Django를 이용해 프론트엔드, 백엔드를 동시에 구현할 계획이다.

Django는 Python으로 구현된 풀스택 개발 프레임워크이다. 최근에는 전용 패키지인 Rest Framework로 인해 API 개발이 수월하므로 백엔드 프레임워크로서의 활용도가 많이 늘어났다. 상대적으로는 프론트는 백에 비해 부실하지만 많은 기능이 필요가 없다고 판단했기 때문에 채택했다.

웹사이트에서 검사할 영상과 로고 이미지를 입력할 수 있다. 입력한 결과는 오른쪽의 결과에서 확인할 수 있다. 또한 각 결과는 버튼으로 구현되어 이를 누를 때 적힌 시간대로 영상으로 넘어갈 수 있게 구현한다.

3.5 배포 계획

웹으로의 실행을 위해 담당 연구실의 서버 vblab research website에 런칭한다. 또한 원활한 서버의 런칭을 위해 Docker 컨테이너를 이용할 계획이다. Docker는 컨테이너 기반의 오픈소스 가상화 플랫폼으로, 하나의 컨테이너로 관리함으로써 서버 관리 및 런칭이 가능해진다. 이를 적용함으로써 프로젝트의 견고함과 유연성을 더할 수 있으며, 다른 서버에도 배포가 쉬워진다.

4. 제약 사항 및 대책

학습 데이터를 제작 후 학습까지 해야 하므로 입력에 관한 결과를 얻기까지 많은 시간이 걸릴 것으로 예상된다. 그러므로 먼저 제작될 dataset의 수를 적절히 조절하며 동시에 각 data에 blur 와 resize를 하여 부담을 덜어 주는 방식을 통해 최대한 빨리 학습할 수 있게 할 계획이다.

5. 개발 분담 및 일정

5.1 역할 분담

이름	역할
전승윤	웹 프레임워크 개발 Github 버전 관리 Docker 컨테이너 개발 및 배포 관리
유동운	모델 학습 및 검증 리펙토링 및 테스트 담당
강태환	학습 데이터 생성 및 전처리 평가 지표 계산 및 평가
공통	필요한 개념 학습 보고서 작성 및 발표 준비

5.2 개발 일정

<table>
<tr><th colspan="3">5월</th><th colspan="5">6월</th><th colspan="5">7월</th><th colspan="5">8월</th><th colspan="4">9월</th></tr>
<tr><th>16</th><th>23</th><th>30</th><th>6</th><th>13</th><th>20</th><th>27</th><th>4</th><th>11</th><th>18</th><th>25</th><th>1</th><th>8</th><th>15</th><th>22</th><th>29</th><th>5</th><th>12</th><th>19</th><th>26</th></tr>
<tr><td colspan="3">웹 개발</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">중간
보고서
작성</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="3">학습 데이터
무작위 생성</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">평가 및
모델
확정</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="3">첫 번째 모델 학습</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">웹 연동 및 배포
관리</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="4">두 번째 모델 학습</td><td></td><td></td><td></td><td colspan="4">최종 테스트</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="4">세 번째 모델 학습</td><td></td><td></td><td></td><td colspan="2">최종 보고서
작성 및 발표
준비</td></tr>
</table>

6. 참고 문헌

[1] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao - YOLOv4: Optimal Speed and Accuracy of Object Detection

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg - SSD: Single Shot MultiBox Detector

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár - Focal Loss for Dense Object Detection

[4] Jacob Solawetz - YOLOv5 New Version - Improvements And Evaluation (https://blog.roboflow.com/yolov5-improvements-and-evaluation/)