

Deep Learning based violent protest detection system

Yeon-su Lee*, Hyun-chul Kim*

Abstract

In this paper, we propose a real-time drone-based violent protest detection system. Our proposed system uses drones to detect scenes of violent protest in real-time. The important problem is that the victims and violent actions have to be manually searched in videos when the evidence has been collected. Firstly, we focused to solve the limitations of existing collecting evidence devices by using drone to collect evidence live and upload in AWS(Amazon Web Service)[1]. Secondly, we built a Deep Learning based violence detection model from the videos using Yolov3 Feature Pyramid Network for human activity recognition, in order to detect three types of violent action. The built model classifies people with possession of gun, swinging pipe, and violent activity with the accuracy of 92, 91 and 80.5% respectively. This system is expected to significantly save time and human resource of the existing collecting evidence.

▶ Keyword: Collecting evidence, drone, deep learning, detection, AWS

I . Introduction

시위는 다수의 사람이 공동의 목적을 추구하는 과정에서 나라 또는 조직에 대해 영향력을 행사하고 일반 시민에게 알리기 위해 하는 공개적이고 집합적인 의사 표현 행위이다. 시위는 크게 폭력 시위와 평화 시위로 구분할 수 있다.[2] 조선일보 사회 일반 뉴스에 따르면 최근 광화문에서의 집회는 1년 새 43%가 급증하였다.[3] 집회가 급증함으로써 그에 따르는 폭력 시위도 같이 증가하였다. 경찰청 예규 제495호에 의하면 각종 집회나 시위 현장에서 불법이 우려되는 상황을 촬영 또는 녹음하여 증거자료를 확보해야 한다. 폭력시위는 한국뿐만 아니라 프랑스, 미국, 중국 등 다른 여러 나라에서도 빈번히 일어나는 것을 볼 수 있다. 폭력 시위가 일어날 때는 크게 2가지의 경우가 발생한다. 첫 번째로는 시위하는 사람들과의 경찰들의 싸움과 난폭한 행위들이 있다. 그리고 경찰들의 일방적인 폭력 행사가 있다. 이러한 행위들을 포착하여 폭력을 행사한 피의자들을 쉽게 찾아낼 수 있게 하기 위해 증거를 수집하는 활동을 한다. 그리고 수집된 증거를 기반으로 폭력을 행사한 시민들과 경찰들을 찾아낸다. 하지만 기존 채증 방식은 데이터수집을 스마트폰이나

비디오카메라를 사용하여 증거 수집을 하고, 채증된 동영상상을 재생시켜 수작업으로 찾아내야 하는 구조이다. 이러한 방법은 소요 시간이 길고, 노동 인력이 많이 필요하다는 한계점이 있다. 이러한 한계점을 보완하기 위해서 정부에서는 재난용 드론 활용 연구 및 스마트 드론을 이용한 무인탐지 기술 도입을 위한 연구[4]가 진행되고 있다. 본 연구 또한 위의 한계점을 극복하기 위해서 드론을 활용한 딥러닝 기반 폭력 시위 탐지 시스템을 개발하였다. 드론을 이용하여 동영상 증거를 확보한 후 세 가지 클래스인 총을 소지하고 있는 행위, 파이프를 휘두르는 행위, 폭력 행위로 탐지된다.

본 논문 2장에서는 본 연구에서 제안한 시스템과 유사한 시스템에 관해서 설명하며, 3장에서는 본 논문의 시스템 구조에 관하여, 4장에서는 데이터를 수집하는 방법 및 데이터의 종류들과 데이터 학습을 설명하고, 5장에서는 폭력 시위 탐지시스템의 탐지 정확도 및 결과를 설명한다. 마지막으로 6장에서는 결론 및 향후 계획에 관해서 기술한다.

• First Author: Yeon-su Lee, Corresponding Author: Hyun-chul Kim
*Yeon-su Lee (yeonsulee530@gmail.com), Dept. of Software, Sangmyung University
*Hyun-chul Kim (hyunchulk@gmail.com), Dept. of Software, Sangmyung University
• Received: 2019. 02. 12, Revised: 2019. 03. 13, Accepted: 2019. 03. 13.
• This work was supported by a 2017 Research Grant from Sangmyung University.

II. Related Work

본 장에서는 제안한 시스템인 폭력 시위 탐지 시스템과 관련된 시스템 총 3가지를 소개한다. 이 중 2가지 시스템은 본 시스템과 같이 물체를 학습하여 탐지하는 시스템과 행동을 학습하여 탐지할 수 있는 시스템이며 CCTV를 사용한다는 공통점이 있다. 다른 한 가지 시스템은 드론과 서버 컴퓨터 간의 연결에 참고할 수 있는 시스템이다. 첫 번째 시스템으로는 NVIDIA사에서 개발한 Cat chaser[5]이다. 위의 그림 1에서 보이는 것처럼 Cat chaser는 물체를 인식한다는 점에서 본 연구와 관련이 있다. 이 시스템은 고정된 위치에서 정원을 망치는 고양이를 탐지하여 내쫓아주는 시스템이다. 또 다른 하나의 시스템은 Eye in the Sky: DSS[6]로 캠브리지 대학교에서 연구된 시스템이다. DSS는 본 연구와 같이 폭력적인 행동을 학습시켜 실시간으로 폭력을 탐지한다. 그러나 차이점은 행동을 탐지하는 방법이다. DSS는 사람의 관절 마디마디를 기반으로 행동을 학습시킨 후 폭력적인 행위를 탐지하도록 구현되어있다. 마지막으로 소개할 시스템은 RTMP 서버를 사용하여 드론으로 촬영된 동영상 상을 실시간으로 서버 PC로 전송 후 딥러닝 기반으로 학습된 모델로 탐지가 된 동영상을 스트리밍해 주는 프로그램[7]이다. 테이블 1을 통해 3가지 시스템에서 사용한 학습 알고리즘과 탐지 방법의 차이점을 볼 수 있다. 본 논문에서 제안하는 시스템은 드론으로 사방면을 촬영하여 CCTV나 사람이 직접 촬영했을 경우 사각지대가 촬영되지 않는 것을 예방할 수 있다. 또한 사람이나 고양이 등의 물체뿐만 아니라 행동을 탐지할 수 있다는 장점이 있다. 클래스를 탐지하기 위한 알고리즘으로 YOLO v3[8] 알고리즘을 사용하였다. YOLO v3 알고리즘의 53-layer를 사용하여 residual skip connection 및 업샘플링(upsampling) 과정을 거쳐 각 클래스들을 정확하게 분류할 수 있다. 이 알고리즘은 빠르고 확장성이 좋기 때문에 다른 문제점을 찾아 시스템을 확장할 수 있는 가능성이 열려 있다. 이 과정을 거쳐 클래스 별로 분류된 동영상들은 실시간으로 AWS

storage에 저장되기 때문에 빠르게 증거를 채집할 수 있다. AWS storage를 사용하면 파일 개수에 제한이 없기 때문에 충분히 데이터를 넣을 수 있고, 데이터의 손실이 발생했을 때 자동으로 복원해주는 기능이 있기 때문에 안전성이 보장되어 사용하게 되었다. 또한 정보의 중요도에 따라 보호 수준을 차등할 수 있기 때문에 비용 절감의 효과도 있다. 가장 중요한 장점은 접근성으로 언제 어디서든 스토리지에 접근하여 데이터를 볼 수 있다는 장점이 있다. 본 시스템은 탐지하는 클래스가 3 가지이기 때문에 다른 시스템과의 차별화가 생기고 이 3가지 클래스를 확장시켜 확장 및 개선 된 시스템을 만들 수 있는 접근성 및 확장성이 있다.

III. System Architecture

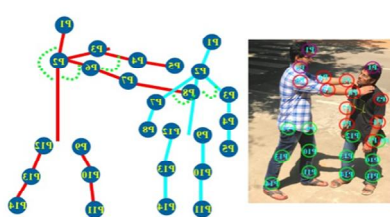
본 논문의 시스템 구조는 그림 2와 같다. 드론으로 시위 영상을 촬영한 뒤 촬영된 동영상은 AWS(Amazon Web service)를 통해 서버 PC로 전송된다. 동영상 촬영을 위한 드론은 DJI사의 Spark를 사용하였다. 이 드론은 고성능 카메라가 장착되어 있어 흔들림 없는 영상을 찍을 수 있을 뿐만 아니라 조종 난이도가 낮아 초보자들이 쉽게 조종할 수 있다. HD Wi-Fi를 이용하여 최대 2km 내에서 영상을 전송할 수 있고, 강한 바람 속에서도 최대 50km/h의 안정된 비행을 할 수 있다는 장점이 있다. 위의 여러 가지 측면에서 DJI사의 Spark모델은 본 시스템과 적합하다고 판단하였다. 이 드론으로 촬영된 동영상 파일은 Youtube로 실시간 스트리밍 된 후 AWS CLI를 통해 AWS 버킷으로 복사되어 저장된다. 이 과정에서는 AWS CLI(Command Line Interface)를 사용하여 Simple Storage Service(S3)[9]에 버킷을 생성하고 동영상을 해당 버킷으로 복사하게 된다. AWS는 데이터 가용성을 제공하는 객체 스토리지 서비스로 사용하기 위해서는 명령줄 인터페이스를 설치해야 한다. AWS CLI를 사용하여 S3에 버킷을 생성하고 파일을 해당 버킷에 복사하

Table 1. Related Work

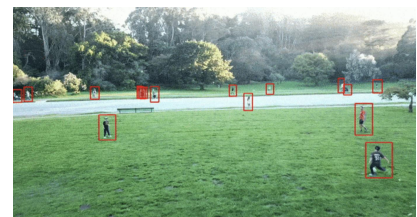
	Cat Chaser	Eye in the Sky:DSS	Real-time Drone Detection System
Camera	CCTV	CCTV	Drone
Method	Object Detection	Joint Detection	Object Detection
Object	Cat	Fighting people	People
Training algorithm	FCN	FPN	R-FCN



(a) Cat Chaser



(b) Eye in the sky:DSS



(c) Real-time Drone Detection System

Fig. 1. Related Work: Screenshots

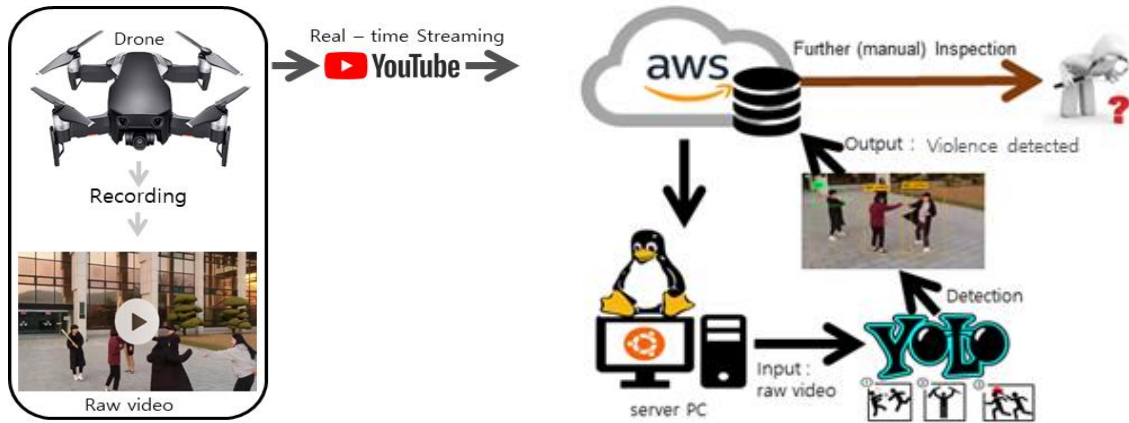


Fig. 2. Architecture of the proposed system

Table 2. Parameters and results

	batch	subdivision	learning_rate	max_batches	accuracy
yolov3.cfg	64	16	0.001	500200	74%
yeonsu.cfg	64	2	0.001	500200	76%
yolo_oct25.cfg	64	32	0.001	75000	80%
yolo_nov5_test.cfg	64	16	0.004	60000	73.2%
yolo_nov5.cfg	64	16	0.004	50000	78%
yolo_nov6_2.cfg	32	8	0.001	45000	82%
yolo_nov6.cfg	128	32	0.001	60000	79.6%
yolo_nov7.cfg	64	16	0.00025	160000	87.8%

면 동영상 파일은 Server PC로 전송되어 저장된다. 전송된 영상은 YOLO(You Only Look Once)[10] 알고리즘으로 만들어진 폭력 탐지 모델을 통해 탐지된다. 폭력 탐지 모델을 만드는 과정에서의 데이터 수집 및 데이터 학습에 관해서는 4장과 5장에서 기술 한다. 서버 PC에서 처리된 동영상은 다시 AWS의 S3인 데이터베이스에 AWS CLI를 통해 해당 버킷에 저장된다. 마지막으로 데이터베이스에 전송된 동영상은 폭력적인 행위가 탐지되어 채증 목적의 증거 동영상이 된다.

IV. Data Training

데이터 학습을 위한 이미지 수집은 3가지 클래스로 분류하여 수집하였다. 이미지 데이터는 그림 4 데이터 라벨링 부분에서 표현된 것과 같이 각각 총을 소지하고 있는 행위, 파이프를 휘두르는 행위 그리고 폭력적인 행동이다. 라벨은 각각 Gun, Pipe, Fight_motion이다. 본 연구는 데이터를 수집 하기 위해 구글 크롤링, Youtube 동영상 및 영화를 프레임 단위로 자르는 방법, 데이터 늘리기를 통해 수집하였다. 위의 3가지 방법을 이용하여 총 8561장의 이미지를 수집했고, 각각 Gun 4901장, Pipe 1789장, Fight_motion 1871장이다.

4.1 Data collection

구글 크롤링을 하는 방법으로 여러 가지 방법이 존재한다. Google API[11]를 사용하여 크롤링하는 방법과 파이썬과 웹

테스팅 도구인 Selenium[12]을 사용하여 크롤링하는 방법이 있다. 기존 Google API를 사용하면 이미지 데이터를 100개씩 또는 500개씩 크롤링해야 하지만 Selenium을 사용하면 웹페이지 자동화 시스템을 통해 자동 스크롤링 및 크롤링을 할 수 있다는 장점이 있어 본 시스템은 Selenium을 사용하여 구글 크롤링을 하였다. 다음 방법으로 이미지를 수집하기 위해 Youtube 동영상 및 영화를 FPS(Frame Per Second)로 잘라내어 이미지 데이터로 변환하였다. 이 과정에서 동영상을 자르기 위해 FFmpeg를 사용하였다. FFmpeg는 디지털 음성 스트림과 영상 스트림에 대해서 다양한 종류의 형태로 기록하고 변환하는 프로그램[13]이다. 명령어를 직접 입력하는 방식으로 동작하며 이미지 데이터를 수집하기 위해 명령어를 사용하여 동영상 파일을 이미지 파일로 변환하였다. 그리고 Thumos Challenge 2015[14]에서 제공된 동영상을 사용하였다. Thumos Challenge 2015는 action detection을 위한 competition으로 13320개의 action 동영상을 제공해 준다. 이 중 Fight_motion에 해당하는 동영상 175개의 동영상을 사용해 위와 같은 방법으로 이미지 데이터를 수집하였다.

4.2 Data augmentation

부족한 이미지 데이터를 증가시키기 위해서 딥러닝 라이브러리인 Keras를 사용하여 데이터 늘리기를 하였다. Keras는 파이썬으로 구현 오픈소스 신경망 라이브러리로 쉽고 간결하게 데이터를 늘릴 수 있다. 그림 3과 같이 하나의 기존 이미지를 이미지 회전, 좌우 반전, 상하 반전, 찌그러뜨리기, 흑백 처리를 수행하였다. 이미지를 늘이기 위한 회전, 찌그러뜨리기 등

은 파라미터 값을 조정하여 어느 정도로 변형을 할 것인지 설정할 수 있다. 이 과정을 통해 총 3000장의 데이터를 8561장으로 늘릴 수 있었다.



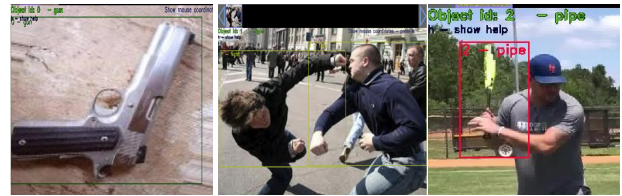
Fig. 3. Data augmentation

4.3 Data labelling

YOLO 알고리즘은 Google의 ImageNet Large-Scale Visual Recognition Challenge(ILSVRC 2014)[15]에서 사용했던 GoogLeNet[16]을 변형한 구조이며 object의 bounding box를 실시간으로 검출할 수 있는 네트워크이다. Object detection을 regression 문제로 접근하며 별도의 region proposal을 위한 구조 없이 YOLO_MARK[17]를 통해 bounding box로 한 번에 전체 이미지로부터 어떤 object들이 어디에 위치하고 있는지 예측할 수 있다. YOLO는 3가지 버전인 YOLO, YOLOv2[18], Yolov3가 있다. 본 연구에서는 가장 빠르고 효율적으로 object detection을 할 수 있는 YOLOv3 알고리즘을 사용하였다.

YOLO 알고리즘을 통해 이미지를 학습시키기 위해서는 YOLO_MARK의 과정이 필요하다. 그림 4에서 보이는 것과 같이 YOLO_MARK를 통해 직접 이미지 파일들에 Bounding Box를 그려줌으로써 Box의 좌표를 지정할 수 있다. 각각의 클래스들의 이미지들을 그림 4와 같이 라벨링을 진행하게 되면 Bounding Box의 각 지점의 좌표들은 클래스 번호와 함께 이미지 파일과 동일한 이름의 텍스트 파일로 저장된다. Bounding Box의 과정에서는 학습하고 싶은 객체가 무엇인지, 이미지에

서 어디에 위치하고 있는지와 클래스 정보를 상세화하는 작업이기 때문에 학습하기 전에 해야 하는 중요한 작업이다. 라벨링 과정을 통해 저장된 텍스트 파일과 이미지 파일은 데이터를 학습 시킬 때 함께 사용된다.



(a) Gun (b) Fight motion (c) Pipe

Fig. 4. Data labeling

4.4 Training

데이터 학습의 클래스는 국내와 해외의 폭력 시위에서 많이 일어나는 3가지 클래스로 분류하였다. 이 3가지는 각각 Gun, Pipe, Fight_motion으로 분류하였다. 총 8561장의 데이터는 표 3에서 나타난 것과 같이 Training, Validation, Test set 각각 80%, 15%, 5%로 구성하였다.

Table 3. Data set

Training		Test
Training set (80%)	Validation set (15%)	Test set (5%)

Batch size는 64로 설정하였고, minibatch는 16으로 설정하였다. 학습률은 0.001로 총 160000번 학습시켰다. 학습은 Yolo v3 알고리즘을 사용하여 학습시켰다. Yolo v3는 Feature pyramid network[19]와 유사한 방식으로 다양한 스케일에서 feature들을 추출한다. 또한 Yolo v3에서는 각각의 box에 대해 multi-label classification을 하는 데, Yolo v3에서는

```

Region 82 Avg IOU: 0.225930, Class: 0.735593, Obj: 0.496596, No Obj: 0.426691, .5R: 0.000000, .75R: 0.000000, count: 4
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.461168, .5R: -nan, .75R: -nan, count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.465269, .5R: -nan, .75R: -nan, count: 0
Region 82 Avg IOU: 0.476592, Class: 0.588316, Obj: 0.385756, No Obj: 0.425859, .5R: 0.000000, .75R: 0.000000, count: 1
Region 94 Avg IOU: 0.182723, Class: 0.410249, Obj: 0.299029, No Obj: 0.459049, .5R: 0.000000, .75R: 0.000000, count: 3
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.465854, .5R: -nan, .75R: -nan, count: 0
Region 82 Avg IOU: 0.243108, Class: 0.473977, Obj: 0.343702, No Obj: 0.425217, .5R: 0.000000, .75R: 0.000000, count: 5
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.460188, .5R: -nan, .75R: -nan, count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.464975, .5R: -nan, .75R: -nan, count: 0
Region 82 Avg IOU: 0.230492, Class: 0.371913, Obj: 0.187147, No Obj: 0.425824, .5R: 0.000000, .75R: 0.000000, count: 2
Region 94 Avg IOU: 0.084983, Class: 0.651520, Obj: 0.397844, No Obj: 0.461040, .5R: 0.000000, .75R: 0.000000, count: 2
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.464876, .5R: -nan, .75R: -nan, count: 0
Region 82 Avg IOU: 0.129963, Class: 0.584898, Obj: 0.288476, No Obj: 0.425910, .5R: 0.000000, .75R: 0.000000, count: 5
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.462007, .5R: -nan, .75R: -nan, count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.464356, .5R: -nan, .75R: -nan, count: 0
Region 82 Avg IOU: 0.246425, Class: 0.633835, Obj: 0.339378, No Obj: 0.426022, .5R: 0.200000, .75R: 0.000000, count: 5
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.460761, .5R: -nan, .75R: -nan, count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.464814, .5R: -nan, .75R: -nan, count: 0
Region 82 Avg IOU: 0.147733, Class: 0.428399, Obj: 0.423302, No Obj: 0.426641, .5R: 0.000000, .75R: 0.000000, count: 6
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.460465, .5R: -nan, .75R: -nan, count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.463517, .5R: -nan, .75R: -nan, count: 0
Region 82 Avg IOU: 0.103746, Class: 0.628727, Obj: 0.516948, No Obj: 0.423904, .5R: 0.000000, .75R: 0.000000, count: 5
Region 94 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.461036, .5R: -nan, .75R: -nan, count: 0
Region 106 Avg IOU: -nan, Class: -nan, Obj: -nan, No Obj: 0.464645, .5R: -nan, .75R: -nan, count: 0
Region 82 Avg IOU: 0.116090, Class: 0.212108, Obj: 0.526030, No Obj: 0.425695, .5R: 0.000000, .75R: 0.000000, count: 4
    
```

Fig. 5. Training

logistic classifier를 사용하였다. 손실함수로는 logistic classifier에서 평균오차제곱 손실함수보다 효율적으로 작용하는 Binary cross entropy 함수를 사용하였다.

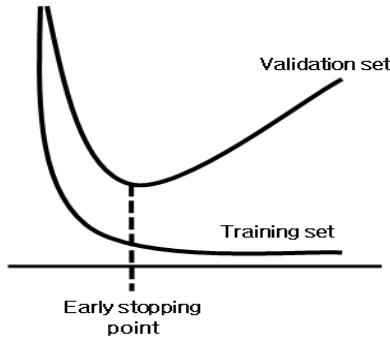


Fig. 6. Early stopping point

데이터 학습이 시작되면 그림 5와 같은 화면을 학습과정에서 볼 수 있다. Yolo v3 는 객체가 있을 것 같은 region 들을 뽑아 어떤 클래스가 있는지 확인해보라고 제안하는 방식으로 학습이 진행된다. 그림 5에서 region 82, 94, 106이 위에서 설명한 region 들이다. 각각 역할이 다른 3개의 네트워크가 하나로 연결되어 전체를 한 번에 학습 시키는 방식으로 학습이 된다. Avg IOU(Intersection Of Union)는 실제 객체가 있는 bounding box의 값과 예측된 값들의 정확도를 평가한 값이다. 학습이 다 된 이후 테스트를 할 때 IOU threshold 값을 이용하여 결과를 예측한다. 예를 들어, threshold값을 0.7로 주었을 때, 예측 값이 70% 이상이면 예측 결과를 출력하고, 그 이하이면 예측 결과를 출력하지 않는 방식으로 사용된다. 학습을 할 때 또 다른 중요한 것은 validation set을 두어 더 이상 avg loss가 줄어들지 않을 때 그림 6에서 볼 수 있는 Early stopping point에서 조기 종영을 해줘야 한다. Yolo v3 네트워크를 사용하여 학습시킬 때, validation의 경로를 지정해주어 조기종영을 하면 학습하는 과정에서 일어날 수 있는 오버 피팅을 방지할 수 있다. Loss 값이 줄어들지 않아 학습이 중단되면 여러 개의 weight 파일을 얻을 수 있는데, weight 파일 중 정확도가 가장 높게 나오는 파일을 골라 사용하면 된다.

데이터 학습을 위한 운영체제는 Ubuntu(16.04) 버전을 사용하였고, Nvidia 사의 내부 박스를 사용하였다. 내부 박스에는

GPU GTX 1080Ti 4개와 Intel (R) Xeon(R) E5-1650 v4 CPU @ 3.60GHz 그리고 RAM 64GB로 구성되어있다. 본 연구에서 학습은 위의 환경에서 50시간 동안 진행하였다.

V. Result

본 논문의 시스템에서는 딥러닝 기반으로 학습된 폭력 탐지 모델을 이용하여 드론으로 촬영된 동영상상을 폭력 탐지하여 아마존 클라우드에 저장된다. 아래의 그림 8과 같이 실제 폭력시위와 드론으로 직접 촬영한 영상으로 테스트를 하였다.



Fig. 7. Upload screenshot

각 클래스의 정확도는 Gun 92%, Pipe 91%, Fight_motion 80.5%이다. 이 시스템의 정확도를 측정하기 위해 실제 시위 영상과 드론으로 5m 높이에서 촬영한 영상들을 사용하였다. 이후 폭력 탐지된 동영상은 AWS CLI(Command Line Interface)를 사용하여 그림 7과 같이 성공적으로 업로드된다.

본 시스템에서 사용한 딥러닝 모델은 위의 테이블 3과 같이 8가지의 파라미터 사용하여 학습시켰다. 배치 사이즈와 subdivision, 학습률, max_batches를 조정하여 학습시켰다. 이 파라미터들 중 subdivision은 일반적으로 말하는 미니 배치를 의미하고 max_batches는 epoch를 의미한다. Batch size는 각각 32, 64, 128로 테스트하였고, subdivision은 2, 8, 16, 32를 사용하였다. 위의 표 2와 같이 8가지 파라미터들을 가지고 학습시키고 테스트를 해보았다. Batch size가 64이고 Subdivision이 16일 때 가장 좋은 성능을 보였다. 학습률은 0.001로 160000번 학습시켰을 때 정확도가 가장 높게 측정되었다. 정확도 테스트는 YOLO에서 제공하는 accuracy 함수를 사용하여 정확도를 측정하였다. 그 결과로 yolo_nov7.cfg의 파라미터들을 사용하여 학습시켰을 때, 87.8%로 가장 높은 결과가 나왔다. 학습 과정에 대한 자세한 정보와 구현 방법 및 참고 자료들은 개인 블로그 페이지[20]에 기술되어 있다.



Fig. 8. Detection Result

VI. Conclusion

본 논문에서는 드론으로 폭력 시위를 촬영하고 폭력 탐지를 하기 위해 폭력 탐지 모델을 만들었다. 이 모델을 만들기 위해 3가지 방법으로 데이터 수집을 하였다. 이미지 데이터는 구글 이미지, Youtube 동영상, 영화들을 데이터 늘이기를 통해 수집하였다. 수집된 동영상은 Yolov3 알고리즘을 사용하여 50시간에 걸쳐 학습하였고 Gun, Pipe, Fight_motion의 정확도는 각각 92%, 91%, 80.5%으로 평균 87.8%이다. 이 학습 모델을 가지고 탐지된 동영상은 AWS의 Simple Storage Service인 S3 버킷을 통해 데이터베이스에 저장된다. AWS의 S3 버킷을 사용하면 단순한 웹 서비스 인터페이스를 사용하여 언제 어디서나 원하는 양의 데이터를 저장 및 검색할 수 있기 때문에 본 시스템에 적합한 인터넷용 스토리지 서비스이다. 본 시스템에서는 드론과 AWS를 함께 활용하여 채증된 동영상을 업로드하는 데 효율적인 시스템을 만들 수 있었다. 또한, Yolo v3 를 기반으로 만든 모델로 3가지 클래스의 폭력을 탐지하고 폭력 시위 영상을 채증하여 기존 채증 활동에 사용되는 인력이 감축되고 시간이 단축되는 효과를 기대할 수 있다.

이 시스템의 향후 계획으로는 Fight_motion 클래스에서의 정확도를 높이는 것이다. Fight_motion의 경우 본 연구에서는 이미지 데이터로만 학습시켰지만, 동영상으로 학습을 시켰다면 더 높은 정확도를 얻을 수 있을 것으로 보인다. 동영상 데이터를 수집하고 동영상 데이터를 기반으로 행동 탐지 모델을 만들어 정확도를 높일 것이다. 동영상을 사용하여 행동 탐지 모델을 만들 때는 동영상을 타임 스텝으로 나누어 CNN과 LSTM을 함께 사용하여 모델을 설계하여 Human action recognition을 할 수 있는 모델을 만들어 본 시스템에 적용 해볼 것이다. 또한, 실시간으로 동영상을 스트리밍 하는 부분에서 2장 관련 시스템의 마지막에서 설명한 시스템[7]을 참고하여 실시간 스트리밍을 구현할 것이다. 현재는 Youtube에서 실시간 스트리밍을 하여 스트리밍 된 동영상 데이터가 AWS storage에 전송되는 방식으로 구현이 되어있다. 하지만 2장의 관련 시스템인 RTMP 서버 [21]를 통해 드론이 동영상을 촬영하면 서버로 직접 스트리밍 되고 저장될 수 있도록 구현할 것이다. 그렇게 하여 드론과 서버 간의 스트리밍을 완벽한 실시간으로 구현하는 것이 목표이다. 마지막으로 폭력 시위 탐지가 된 동영상에서 폭력이 탐지되는 부분의 캡처를 자동화할 것이다. 채증된 동영상에서 탐지된 폭력적인 부분들을 프레임 단위로 자동으로 캡처하여 저장된다면 효율성을 증대시킬 수 있을 것으로 보인다. 위의 3가지의 목표들이 본 시스템의 향후 계획이다.

REFERENCES

- [1] AWS(AmazonWebService)“https://aws.amazon.com/?nc2=h_lg”
- [2] Naver about protest “<https://terms.naver.com/entry.nhn?docId=1820712&cid=46634&categoryId=46634>”
- [3] Chosunllbo, ratio of demonstration “http://news.chosun.com/site/data/html_dir/2018/11/03/2018110300250.html”
- [4] Navernews research about drone“<http://www.sportsseoul.com/news/read/617479>”
- [5] CatChaser(2016)ByRobertBond Nvidia “<https://devblogs.nvidia.com/ai-cat-chaser-jetson-tx1-caffe>”
- [6] SINGH, Amarjot; PATIL, Devendra; OMKAR, S. N. Eye in the Sky: Real-time Drone Surveillance System (DSS) for Violent Individuals Identification using ScatterNet Hybrid Deep Learning Network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, p. 1629–1637, June 2018
- [7] How To Do Real Time Object Detection OnDroneVideo Streams“https://blog.nanonets.com/real-time-object-detection-for-drones/?utm_source=facebook&utm_medium=social&utm_campaign=drone&utm_content=group_DeepLearnng&fbclid=IwAR34ULQditi4eM3Gvm-Ic84IM6LDA9EIQR6bmFalsqRRJLhgauCgMTa-v8”
- [8] REDMON, Joseph; FARHADI, Ali. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, April 2018.
- [9] AWS Amazon Simple Storage Service “https://aws.amazon.com/s3/?nc2=type_a”
- [10] ZHANG, Jianming, et al. A real-time chinese traffic sign detection algorithm based on modified YOLOv2. Algorithms, Vol. 10 Issue 4, p127. 13p, Dec2017.
- [11] GoogleAPI “<https://support.google.com/webmasters/answer/1061943?hl=ko>”
- [12] Google crawling using selenium “<https://www.slideshare.net/winglessagl/seleniu-61525457>”
- [13] Image capture FFmpeg “<https://www.ffmpeg.org/>”
- [14] Thumos Challenge2015 “<http://www.thumos.info/download.html>”
- [15] Large Scale Visual Recognition Challenge 2014“<http://www.image-net.org/challenges/LSVRC/2014/>”
- [16] SZEGEDY, Christian, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. CVPR , p. 1–9, June 2015.
- [17] Labeling Yolo_mark “https://github.com/AlexeyAB/Yolo_mark”
- [18] REDMON, Joseph; FARHADI, Ali. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. CVPR, p.

7263-7271, July 2017.

- [19] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." CVPR. Vol. 1. No. 2. July 2017.
- [20] Tistory Blog yeonsu "<http://yeonsuuu-uuu.tistory.com/>"
- [21] Live streaming RTMP Server "<http://blog.manycam.com/rtmp-live-stream/#.XATy5hMzbcM>"

Authors



Yeon-su Lee received the B.S. degree in Computer Science from Sangmyung University, Korea, in 2019. She is interested in data mining and machine learning.



Hyun-Chul Kim received the B.S., M.S. and Ph.D. degrees in Computer Science from KAIST, Korea, in 1995, 1997, and 2005, respectively. Professor Kim joined the faculty of the Department of Software at Sangmyung University,

Cheonan, Korea, in 2012. He is currently an Associate Professor in the Department of Software, Sangmyung University. He is interested in Internet Data Science.