碩士學位 請求論文

指導敎授 黃 憲

YOLOv4와 Opticalflow를 활용한 보행자 행동검출

成均館大學校 一般大學院

바이오메카트로닉스 學科

金 湧 玹

碩士學位請求論文

YOLOv4와 OptIcalflow를 활용한 보행자 행동검출

2021

金湧玹

碩士學位 請求論文
指導敎授 黃 憲

YOLOv4와 Opticalflow를 활용한 보행자 행동검출

Pedestrian behavior detection using YOLOv4 and optical flow

成均館大學校 一般大學院
바이오메카트로닉스학과

金 湧 玹

碩士學位 請求論文
指導敎授 黃 憲

YOLOv4와 Opticalflow를 활용한 보행자 행동검출

Pedestrian behavior detection using YOLOv4 and optical flow

이 論文을 工學 碩士學位請求論文으로 提出합니다.

2020 年 10 月 日

成均館大學校 一般大學院

바이오메카트로닉스 學科

金 湧 玹

이 論文을 金 湧 玹의 工學

碩士學位 論文으로 認定함.

2020 年 12 月 日

審査委員長

審査委員

審査委員

목차

표목차

그림목차

논문요약

YOLOv4와 Opticalflow를 활용한 보행자 행동검출

최근 하드웨어의 성능과 통신서비스의 기술이 점차 발전하면서, 이미지가 아닌 영상으로 할 수 있는 많은 연구 들이 진행되고 있다. 그 중에서도 최근 이상 감지 연구가 활발히 진행되고 있다. 음성, 텍스트, 이미지 분야에서도 이상 감지를 하기 위해 기계학습과 딥러닝을 많이 쓰고 있다.

2012년부터 해마다 무연고 사망자는 증가추세를 보이며 2019년에는 2536명으로 만65세 이상 고령자가 45%를 차지한다. 이러한 문제점 해결을 위한 인력은 부족하고 마땅한 해결방안도 없다. 최근 들어 유아, 반려견, 노인을 케어하기 위한 많은 종류의 카메라를 활용한 서비스들이 나타나고 있다. 하지만 문제점은 사람을 인식은 하지만, 행동과 방향성을 판단 할 수는 없다는 점이다. 쓰러짐과 같은 이상행동은 이미지만으로는 판단이 정확하지 않아서 심장박동수 측정 웨어러블 장치나 다른 외부장치 없이 영상만으로 사람의 쓰러짐을 인식하기는 어렵다. 본 연구에서는 객체 인식시스템과 객체의 픽셀 움직임을 파악해서 걷기, 뛰기를 구분하고 방향성을 판단할 수 있으며, 픽셀의 움직임을 분석해서 외부장치 없이 쓰러짐의 유무를 판단할 수 있다.

본 논문에서는 영상에서 보행자의 행동을 걷기와 뛰기로 나누었을 때 기존 YOLOv4 인식률의 정확도를 올렸으며, 보행을 하다가 쓰러짐의 상황에서 쓰러짐을 정확하게 판단하는 연구를 진행하였다. 인식 시스템에 사용한 기술은 올해 4월에

발표된 영상에서 객체를 검출하는 YOLOv4 모델을 활용해서 행동 데이터를 걷기, 뛰기, 쓰러짐 3가지 class로 나누어 학습을 시켰다. 이 후 행동이 변할 때의 인식률이 떨어지는 것을 보완하기 위해, Opticalflow를 인식시스템에 접목해서 이상 행동 검출의 정확도를 높이는 방법을 제안하였다. 이를 통해 걷기에서 뛰기의 변화를 프레임 간의 픽셀 움직임 비교를 통해 감지할 수 있으며, 쓰러짐 같은 움직임이 없는 행동을 인식할 때도 기존보다 정확하게 인식을 할 수 있다.

본 연구에서는 YOLOv4의 Cocodataset을 학습해서 추출된 pretrain모델을 걷기, 뛰기, 쓰러짐 이미지 각 600장씩, 총 1800장의 annotation(bounding box)된 이미지 데이터에 전이학습 시켜서 추출된 custom_model 을 기반으로 실험 하였다.

주제어 : YOLOv4 , Opticalflow, 행동검출, 물체 방향성판단 , 딥러닝

제 1장 서 론

1.1 연구 배경 및 논문 구성

인공지능이 발전하면서 우리의 생활은 점점 체계적이고 정확하고 편리하게 되어 가고 있다. 기존에 사람이 하던 일을 컴퓨터가 대신 다양한 분야에서 빠르게 일 처리가 되고 있다. 이제는 영화에서나 보던 범인의 모습과 얼굴을 인식하고 카메라로 실시간 추적을 하고 범인을 잡는 상황이 완전히 불가능한 현실은 아니다. 이처럼 사람이 일일이 눈으로 찾고 감시할 필요가 없이 딥러닝의 활용으로 많은 상황에서 사람들을 관리 할 수 있게 되었다.

요새 가정에서도 홈케어카메라 같이 노인 또는 아이의 안전을 위한 가정용 cctv가 인기를 끌고 있다. 노약자를 지켜보기 위해 가정에 카메라는 설치했지만, 하루 종일 녹화영상만 볼 수는 없다. 고령화 시대에 접어들면서 노인들이 홀로 방치되어서 쓰러지는 사고가 빈번히 발생하고 점점 증가하는 추세를 보이지만 큰 대책은 없다. 본 논문에서는 영상정보를 받아서 컴퓨터가 스스로 보행자의 행동을 검출하고 판단 하는 객체인식 시스템연구 방법을 제안한다.

논문의 2장에서는 YOLO의 배경이론과 객체 인식시스템에 관련해서 one-stage 기법과 two-stage기법으로 나눠서 객체 인식시스템의 기존연구에 대하여 설명을 했으며, OpenCV와 그 안의 opticalflow 기법에 대한 기존 연구 이론에 대해서도 설

명하였다. 본 논문의 3장에서는 YOLOv4를 활용한 전이학습 방법에 대하여 설명한다. 전이학습에 의한 기존의 weight값을 반영하기 때문에 높은 정확도와 빠른 학습으로 특정 객체를 검출할 수 있을뿐더러 weight값을 연구 주제에 편향 시켰기 때문에 가벼운 모델을 추출할 수 있다. 뿐만 아니라 opticalflow로 영상의 픽셀흐름을 분석해서 객체인식 시스템에서의 한계를 보완하기 위해 사람의 움직임 방향성과 쓰러짐 판단을 보다 정확하게 할 수 있는 방법을 제시하였다. 4장에서는 전이학습을 통해 얻은 모델의 정확도를 평가한 내용에 대해 서술했으며, Opticalflow를 같이 활용해서 행동을 검출했을 때의 결과 값과 정확도에 대해서 서술했다. 마지막 5장의 결론에서는 최종 연구를 통해 얻은 결론으로 향후 진행되어야 할 연구의 방향성을 제시한다.

1.2 연구 목적 및 내용

일반적인 단일 렌즈 영상만으로 행동을 분석하기는 쉽지 않다. 기존 RNN계열의 LSTM[23]을 활용했을 경우 시계열 분석이 가능하지만, 객체별 계산량이 많아서 오래 걸린다는 한계가 있다. 기존 객체인식 시스템으로는 사람을 검출하는 것은 가능해도 행동과 방향성을 검출하기는 쉽지 않다. 특히 쓰러짐 같이 움직임이 없는 행동은 웨어러블 장치 없이는 판단이 힘들다.

기존 행동연구에서는 LSTM[23]을 활용해서 영상분석을 통해 행동을 예측하는 연구가 있었으며, IMU(Inertial Measurement Unit)센서를 부착한 사람의 낙상을 미리 판단할 수 있는 연구[22]가 있었다.

본 논문에서는 오래 걸리는 계산을 줄이기 위해 YOLOv4[3]와 Opticalflow[16]를 활용하였으며, 웨어러블 장치 없이도 쓰러졌을 때, 픽셀의 움직임이 없는 것을 판단해서 빠르게 감지할 수 있다. 이를 통해 실시간으로 환자나 노약자의 상태를 인식할 수 있으며, 지속적인 모니터링 없이 쓰러짐 같은 이상행동을 인식할 수 있다.

연구방법의 순서는 전이학습을 통해 얻은 모델을 통해서 사람의 행동을 인식한다. 사람을 검출한 바운딩박스 중심좌표와 높이와 넓이의 값을 통해 검출되지 않은 부분에는 블랙화면을 뿌려서 opticalflow가 배경을 움직임 포인터로 잡지 못하게 하였다. 라벨링된 각 객체별로 위의 방법을 실행한 후, 최종 출력 영상에 각 라벨링별 행동과 움직임을 판단할 수 있다. 적은 계산량으로 사람의 걷기와 뛰기를 구분하고 방향성을 파악할 수 있으며, 기존처럼 외부장치를 필요로 하지 않고 한번 인식된 객체의 픽셀 움직임이 없을 경우, 빠르게 쓰러짐을 판단할 수 있다.

제 2장 관련연구 배경이론

2.1. 객체인식 시스템 YOLO

객체인식은 이미지 또는 영상에서 물체를 식벽하는 Computer Vision 기술이다. 딥러닝과 머신러닝의 결과물이라고 할수 있으며, 인풋값으로 영상이나 사진을 넣으면 우리가 찾고자 하는 사람, 객체. 장면, 그밖에 시각적인 요소들을 특징으로 잡아서 찾을 수 있다. 목표는 사람이 보고 판단하는 것처럼 자연스럽게 빠르고 정확하게 판단하는 것을 목표로 한다.

객체 인식은 자율주행, 바이오이미지, 산업적 분야, 로봇비전 분야에서도 다양하게 쓰인다. 물체분류와 유사한 면이 있다. 하지만 이미지 식별뿐만 아니라 위치까지 찾는 물체 탐지는 물체인식의 한 부분이다.

객체인식 시스템에는 크게 두가지 방법이 있다. 물체를 어떠한 구조로 학습하고 인식하느냐에 따라 나뉘는데 우리는 이를 one-stage기법, two-stage기법 이라고 부른다. one-stage기법의 대표적인 모델이 YOLO[1], SSD, RetinaNet 계열등이 있고, two-stage기법에는 R-CNN계열의 Fast-RCNN[4], Faster-RCNN[5]이 가장 대표적이다. 두 방법의 큰 차이점은 목적이 다르다. one-stage기법은 속도에 연구중점을 맞추며, two-stage기법은 정확도에 연구 중점을 둔다.

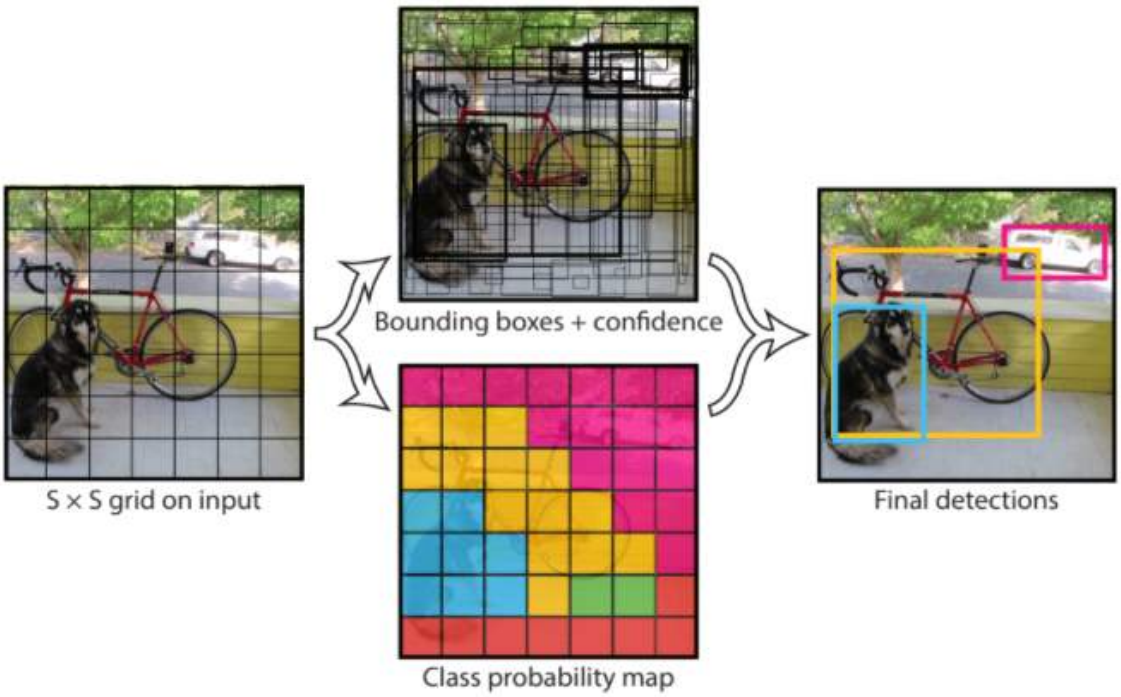

그림2-1. YOLO의 객체인식 시스템

그림2-1은 YOLO[1]의 이미지인식 기법의 구조를 보여준다. YOLO같은 onestage 기법은 기존 R-CNN계열의 region proposal[5]기법이 생략됨으로써, 객체인식 속도가 많이 향상 되었다. input이미지를 S X S 그리드 영역으로 나눠서 물체가 있을 것 같은 영역을 예측해서 바운딩박스를 칩니다. 그 후 바운딩박스의 중심좌표와 박스의 크기를 계산해서 박스의 정확도인 confidence를 계산한다. ground truth박스와

예측값의 겹치는 비율을 나타내는 Intersection over Union(IoU)과 곱하여서 Confidence값을 구한다.

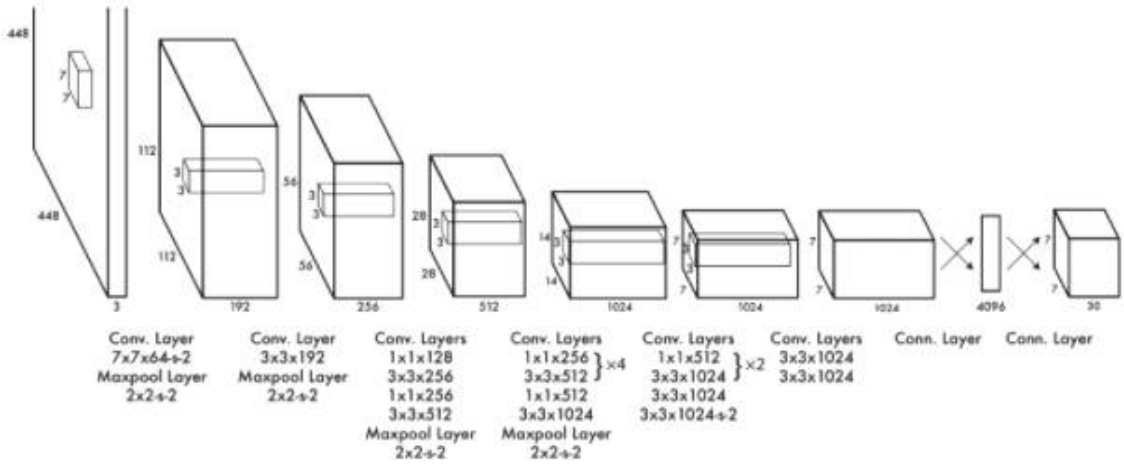

그림2-2. YOLO CNN Architecture

그림2-2는 YOLO의 컨볼루션네트워크 구조를 나타낸다. YOLO의 신경망 구조는 이미지 분류에 쓰이는 GoogLeNet[17]을 사용했다. YOLO는 총 24개의 컨볼루션 계층(convolutional layers)과 2개의 전결합 계층(fully connected layers)으로 구성되어 있습니다. GoogLeNet의 인셉션 구조 대신 YOLO는 1x1 축소 계층(reduction layer)과 3 x 3 컨볼루션 계층의 결합을 사용했습니다. 1x1 축소 계층(reduction layer)과 3 x 3 컨볼루션 계층의 결합이 인셉션 구조를 대신한다고 합니다. YOLO 모델의 전체 구조는 다음과 같습니다. 이 모델의 최종 아웃풋은 7 x 7 x 30의 예측 텐서(prediction tensors)입니다.

$$
\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
$$

그림2-3. loss계산식

그림3은 물체를 인식할 때의 loss 계산식이다. 물체가 존재하는 그리드i 의 바운딩 박스 j에 대해, x, y의 loss 계산을 하고 x, h에 대한 loss 값도 계산은 합니다. 그 후 큰 바운딩 박스에 대해서 분산 값을 반영하기 위해 제곱근을 하고, SSE 값을 구합니다. 물체가 존재하는 그리드 셀의 바운딩 박스 j에 대해서 Confidence score의 loss값을 계산하고, 물체가 존재하지 않는 그리드 셀에 대해서도 똑같이 Confidence score의 loss 값을 계산합니다.

이런 방식이 YOLO의 시작이며 여기에 Darknet-19모델을 통한 전이학습을 통해 속도와 정확성을 올린 것이 YOLOv2입니다. YOLOv3[2]에서도 멀티라벨 분류 기법과 모델을 Darknet-19에서 Darknet-53으로 좀 더 깊게 모델을 설정하면서 정확도와 속도를 올렸습니다. 본 연구에서는 YOLOv4[3]를 사용하였으며, BoF 와 같은 데이터 증강기법을 통해 인풋이미지에 광도변화 및 기하변형을 통한 가변성을 주어, 학습모델이 이미지의 변형이나 환경에 의한 픽셀 변화에도 견고하게 하였다. 또한 BoS와같이 기존 네트워크 모듈과 활성화 함수들을 변경하고, backbone 단계에 기존의FPN[13] 대신 PANet[8] 레이어를 추가해서 기존 YOLO에 비해 정확성과 속도를 모두 끌어올렸다.

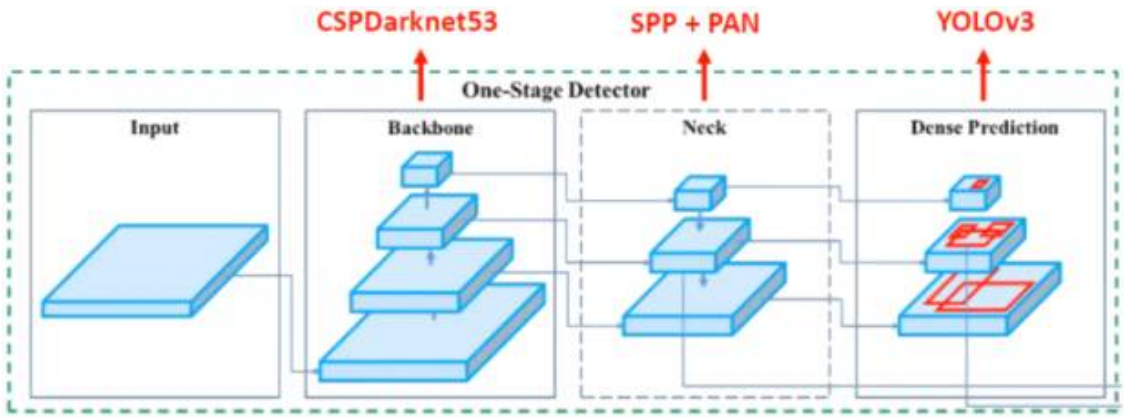

그림2-4. YOLOv4 Architecture

그림4는 YOLOv4[3]의 전체적인 구조를 나타낸다. 인풋 이미지가 들어왔을 때 backbone구조에서 CSP(Cross-Stage Partial connections)DarkNet-53을 사용하며, 중복 기울기를 없앰으로써, 속도를 올리고 성능을 높였다. 그 외에도 Neck부분에서

SPP를 적용했다는 점인데, SPP의 가장 큰 장점은 인풋이미지 사이즈를 기존 YOLOv3[2]처럼 224x224로 고정을 할 필요가 없다는 점이다. 그러므로 큰 이미지 사이즈를 넣어서 정확성을 올려 학습을 할 수가 있다. 마지막 꼬리 부분에서는 YOLOv3[2]의 architecture를 갖다 쓰면서 전체적인 네트워크를 구성했다.

2.2. 전이학습(transfer learning)

전이학습[18]이란 인풋데이터의 특징을 추출하고 학습하여 얻은 트레인 모델을 가지고 다른 데이터를 학습하는 것을 말한다. 다양한 분야에서 학습의 효율성을 높이기위해 전이학습을 사용한다. 특히 CNN기반의 딥러닝 네트워크 분야에서는 noise없는 충분한 데이터셋을 얻는 것이 쉽지 않다. 데이터셋을 얻기 위한 시간과 비용이 많이 들기 때문에 전이학습으로 비용과 시간을 많이 절감할 수 있다.

개인의 데이터셋을 학습해서 편향된 model을 얻는 것보다, 검증된 데이터인 cocodataset[14], ImageNet과 같은 데이터셋을 VGG모델[11], ResNet[12]과 같은 검증되고 견고한 모델로 학습을 시켜 얻은 pretrain모델의 weight값을 이용해서 개인 데이터를 학습하면 학습속도도 빨라지고, 정확도 또한 향상된다. 사람의 경우에도 똑같은 예를 들 수가 있다. 사과 깎는 법을 가르치고 싶을 때, 배를 깎는 법을 가르치고 사과 깎는 법을 가르치면 효율성이 좋다. 기존의 train모델을 사용해서 파인튜닝 학습을 하는 것을 전이학습이라 한다.

딥러닝을 포함한 기계학습은 train data와 test data가 비슷하거나 같은 특징과 분포를 갖고있을 때 효과적이며, 그 외에는 학습에 어려움을 겪는 경우가 많다. 따라서 데이터의 특징이나 분포가 변화하면 기존 수학적 모델들을 새로 생성된 train data로 다시 재학습 시켜야 한다. 이는 비용과 시간이 많이 들고 불가능한 경우가 많다, 따라서 우리는 이런 비용과 시간을 줄이기 위해 knowledge transfer 또는 transfer learning을 통하여 해결해야 한다.

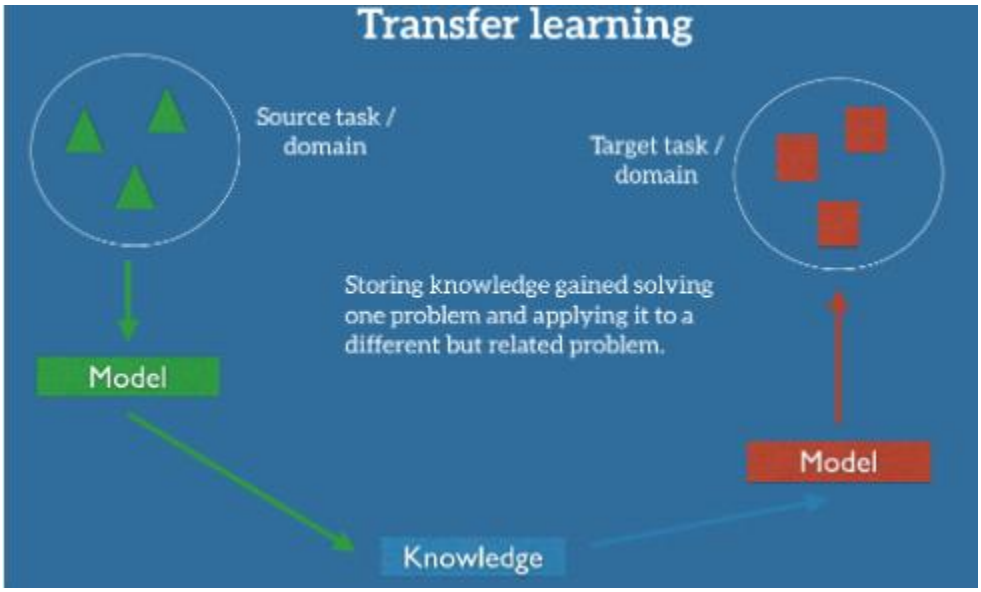

그림2-5. pre-train을 활용한 Transfer Learing 구조

그림2-5 가 본 연구에서 사용된 전이학습의 한 방법의 구조를 나타낸다. 방대하고 보증된 데이터로 학습모델을 추출해서, 모델의 weight 즉 지식을 다른 데이터를 학습할 때 넣어주는 것이다. 전이학습의 장점은 위에서 언급했듯이, 바로 학습을 시키는 것보다 효율적이지만, 학습 시키고자 하는 customData 양이 적을 때에도 효율적으로 학습을 할 수 있기 때문에 detection, classification 분야에서 많이 쓰이고 있다.

데이터의 양이 많아 지도학습을 통해 모델을 견고하게 뽑으면 좋겠지만, 대부분의 경우 데이터 수집에 어려움을 겪고 한정적이기 때문에 비지도 학습과, 준 지도 학습관련 연구가 많이 이루어지며, 전이학습에도 어떠한 모델을 붙여서 쓸지 연구가 이루어지고 있다.

2.3 Opticalflow

영상은 2차원 이미지들이 시간의 흐름 순으로 나열된 것이다. 이미지 프레임과 이전 이미지 프레임과의 차이를 통해서 영상에 존재하는 물체의 움직임을 추정할 수 있다. 즉 시퀀스 프레임을 통해 그 물체가 무엇이든 이전 프레임과의 비교를 통해 픽셀의 좌표 값이 달라지면 opticalflow[16]를 통해서 움직임을 추정할 수 있다. 단 픽셀의 밝기 값이 주기적으로 변화할 경우에는 물체의 움직임을 추정하기 어렵다. 이를 수식으로 표현하면 다음 그림2-6 과 같다.

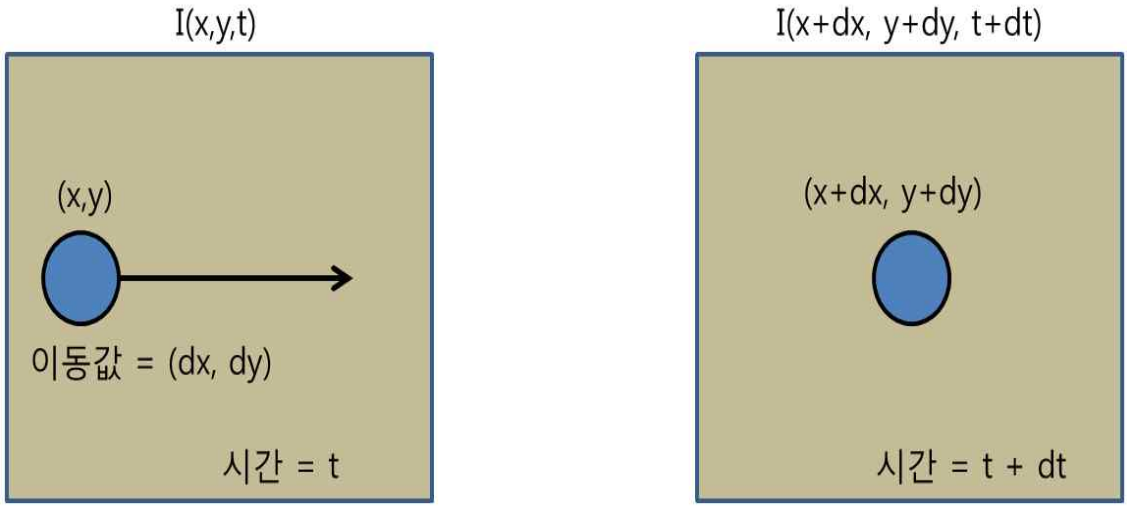

그림2-6. 시간에 따른 물체 이동식

그림2-6에서 I는 intensity의 약자이며 밝기 값을 뜻한다. x,y는 이미지의 공간좌표 값이며 t는 시간을 나타낸다. 우리는 이러한 식으로 그림3의 파란색 공의 움직임의 방향과 이미지에서 거리의 ground truth값이 있다면, 속도추정까지 가능하다.

opticalflow에는 크게 Sparse opticalflow와 Dense opticalflow가 있다.

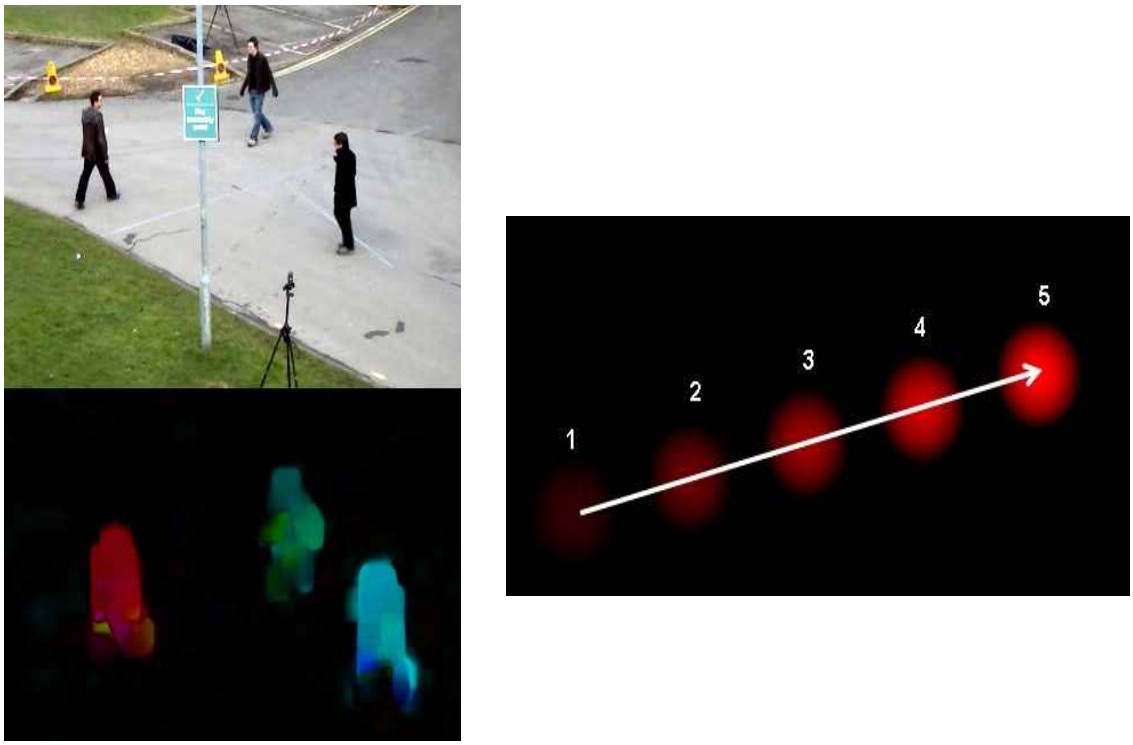

그림2-7. Dense opticalflow

그림2-8. Sparse opticalflow

그림2-7 에서처럼 Dense opticalflow는 물체 전체의 흐름벡터를 제공한다. 프레임의 모든 픽셀에 대해 벡터 값을 계산하는데, 계산양이 많아서 느리지만 정확한 결과 값을 도출할 수 있다. 그림2-8 에서처럼sparse opticalflow는 일부 물체의 가장자리나 모서리 포인트의 픽셀 값만 계산하여 나타내기 때문에 정확성은 떨어지지만. 빠르다는 장점이 있다.

제3장 행동 검출 및 방향성 인식 방법

본 장에서는 연구를 위해 사용된 데이터와 사람의 행동 검출 방법과 효율적인 검출 방법을 위한 물체의 흐름을 통한 방법과 객체 인식을 통한 방법에 대해 설명한다.

3.1 이미지데이터 수집

데이터는 개인촬영 영상과 “AI HUB” 로 부터 제공받은 데이터를 사용하였습니다. “AI HUB”는 AI기술 및 제품 서비스 개발에 필요한 AI인프라를 지원함으로써 누구나 활용하고 참여하는 AI 통합 플랫폼입니다. 그 중에서 사람의 동작영상 데이터는 ㈜스위트케이 로 부터 연구목적 사용으로 허가를 받고 본 논문의 연구에 활용하였습니다.

사용된 영상데이터는 사람의 걷기, 뛰기, 쓰러짐 3개의 동작에 대한 영상을 활용해서 이미지로 변환 후, 어노테이션(annotation)작업을 진행 하였습니다.

	사용된 영상 개수	추출한 이미지 개수	활용된 사람 수
걷기	17	600	7
뛰기	17	600	7
쓰러짐	26	600	12

<표3-1> 활용데이터 양

<표3-1>에서 보는 것과 같이 사용된 영상의 개수는 총 60개이며, 이미지의 추출 개수는 1800장이다. 영상 촬영에 활용된 사람은 총 26명이다. 쓰러짐의 영상을 다른 동작보다 많이 사용한 이유는 쓰러짐의 과정을 이미지로 추출할 경우 데이터의 양이 적기 때문에 다른 동작들보다 많은 영상의 수를 사용해서 이미지 데이터를

추출했다.

어노테이션 작업은 YOLOv4에 맞게 동작에 바운딩박스를 쳐서 레이블을 구분했으며, 툴은 DarkLabel Tool을 활용했다.

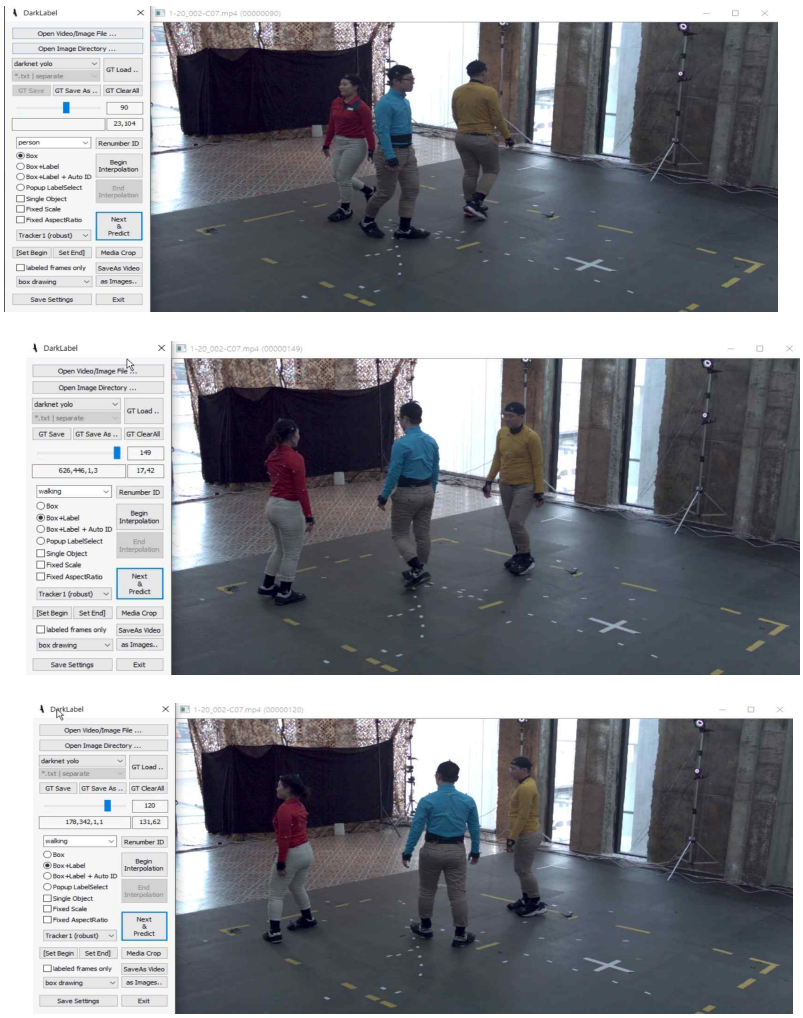

그림3-1. DarkLabel Tool

그림3-1 이 DarkLabel을 활용해서 걷기 영상을 불러오는 화면이다. 영상데이터를 프레임 단위로 넘기면서 학습데이터 어노테이션 작업을 할 수가 있다.

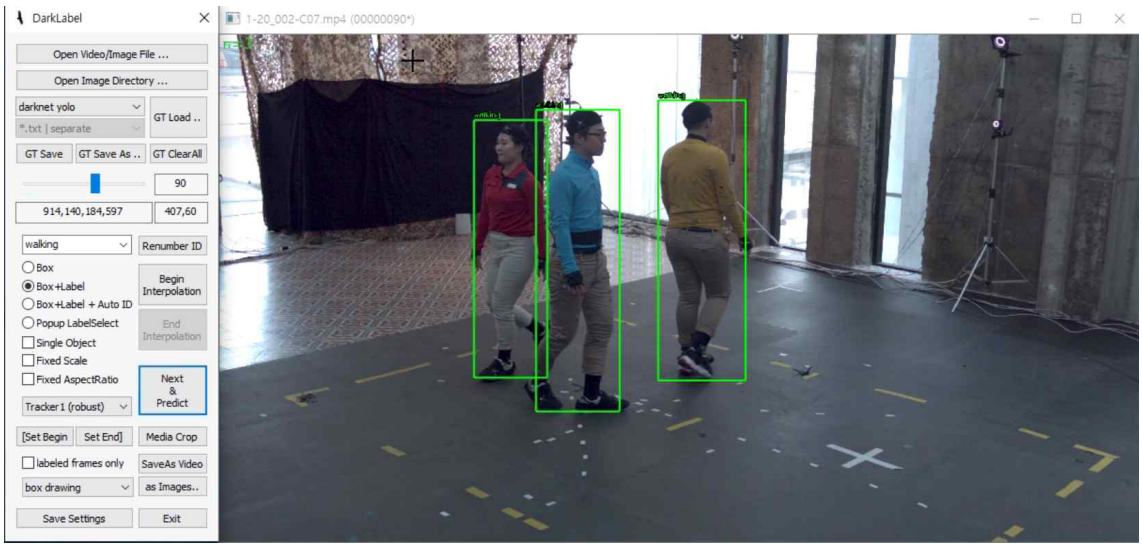

그림3-2. DarkLabel을 활용해서 bounding box 작업한 이미지

그림3-2에서 보이는 초록색 네모를 바운딩 박스라 부르며 이러한 작업을 어노테이션 이라고 한다. 박스안에 학습시키고 싶은 물체를 넣어서 학습데이터로 활용되게끔 하는 작업이다. 이미지와 함께 박스의 픽셀 위치 좌표 값과 class number가 있는 .txt 파일도 함께 준비한 후 학습데이터에 넣어줘야 제대로 된 학습 데이터로 활용을 할 수가 있다

3.2. 사람행동 검출 방법

기존 YOLOv3[2]에서 인풋이미지 사이즈에 한계가 있었지만, 올해 새로 발표한 YOLOv4[3]에서는 그 문제점이 쉽게 해결되었다. 따라서 같은 양의 데이터를 학습시켜도 사이즈를 크게 학습시키면 오래걸리는 단점은 있지만 정확하게 학습하는 장점이 있다.

Train image	1800
Image input size	416 x 416
Batchsize	48
Subdivision	12
Learning rate	0.001
Max_batch	6000
Filter	12
Padding	1
Stride	1

<표3-2 학습 설정값>

기본적인 네트워크는 YOLOv4[3]논문에서 Backbone이라 불리는 단계에서 CSP(Cross-Stage Partial connections)DarkNet-53[21]을 사용한다.

표3-2은 학습 시 사용된 데이터양과 네트워크 설정값을 나타낸다. 한번에 네트워크에 넘겨주는 양을 뜻하는 batchsize는 48로 설정했으며, 48로 크게 설정해도 subdivision(mini-batch)값을 12로 설정해서 4번에 나눠서 쪼개서 들어가기 때문에 단일 gpu에서도 큰 무리 없이 학습을 진행 하였다. 최종적인 출력 층을 나타내는 필터의 개수도 클래스의 값에 맞게 24개로 설정하였으며, Iteration을 뜻하는 max_batch는 클래스의 개수에 맞게 6000으로 설정하였고, learning rate는 0.001로 설정하고, 그 밖에 stride 와 padding값은 default 값인 1로 설정 하고 학습을 시켰습니다.

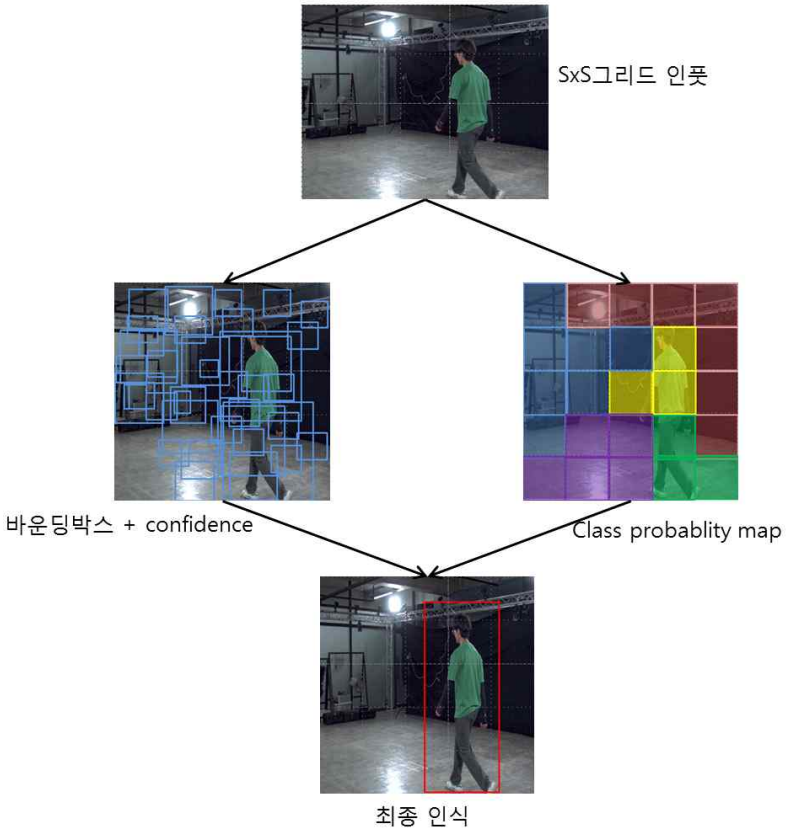

그림3-3. YOLO의 객체인식 시스템

그림3-3은 YOLO의 이미지인식 기법의 구조를 보여준다. YOLOv4[3]논문에서는 Head단계 라고 불리는 인식 단계이다.

본 연구에서는 YOLOv4를 사용하였으며, BoF 기법의 데이터증강을 통해 인풋이미지에 광도변화 및 기하변형을 통한 가변성을 주어, 학습모델이 이미지의 변형이나 환경에 의한 픽셀 변화에도 견고하게 하였다. 또한 BoS와같이 기존 네트워크 모듈과 활성화 함수들을 변경하고, backbone 단계에 기존의FPN 대신 PANet 레이어를 추가해서 기존 YOLO에 비해 정확성과 속도를 모두 끌어올렸다.

3.3. 사람의 방향성과 움직임 판단 방법

사람의 방향성을 판단하기 위해서 sparse opticalflow를 활용해서 이동방향과 속력을 대략 측정할 수 있었다. 방향성을 판단하는 것은 영상만으로 가능하지만, 속력을 알기 위해서는 영상에 대한 ground truth값이 필요하므로 본 연구에서는 걷기와 뛰기의 판단[19],[20]을 위해 움직임을 판단하고 opencv를 활용하여 출력 text 값으로 방향성과 “walk” 와 “run”의 값이 출력 되도록 하였다.

그림3-4 걷기와 뛰기 방향성 판단

그림3-4에서 보이듯이 프레임을 비교해서 걸을 때와 뛸 때를 추정하고 방향성도 판단한다. 이 와 마찬가지로 프레임에서 움직임이 없을 때는 쓰러짐으로 판단 할 수 있다.

제4장 실험결과

본 장에서는 YOLOv4[3] 이용해서 전이학습을 통해 얻은 모델을 테스트 하고, detection의 성능 문제점을 Opticalflow[16] 용해서 보완하는 방법을 제안하였다.

4.1. 전이학습을 통해 얻은 weight모델의 성능

기존 YOLOv4 모델을 사용해서 MS-COCO dataset 을 학습 시켜서 얻은 YOLO_pre_train 모델을 기반으로 전이학습(Trransfer Learning)을 실행하였다. 학습과정과 결과 및 시간은 하단의 표와 그림을 통해서 설명 하겠다.

```
 (next mAP calculation at 1100 iterations)
 Last accuracy mAP@0.5 = 72.52 %, best = 72.52 %
 1078: 10.339696, 14.696595 avg loss, 0.001000 rate, 11.320904 seconds, 51744 images, 2.151644 hours left
Loaded: 0.000054 seconds

 (next mAP calculation at 1100 iterations)
 Last accuracy mAP@0.5 = 72.52 %, best = 72.52 %
 1079: 14.570712, 14.684007 avg loss, 0.001000 rate, 11.919492 seconds, 51792 images, 2.159122 hours left
Loaded: 0.000074 seconds

 (next mAP calculation at 1100 iterations)
 Last accuracy mAP@0.5 = 72.52 %, best = 72.52 %
 1080: 14.978015, 14.713408 avg loss, 0.001000 rate, 11.605806 seconds, 51840 images, 2.168025 hours left
Resizing, random_coef = 1.40

 576 x 576
 try to allocate additional workspace_size = 52.43 MB
 CUDA allocate done!
Loaded: 0.000045 seconds

 (next mAP calculation at 1100 iterations)
 Last accuracy mAP@0.5 = 72.52 %, best = 72.52 %
 1081: 20.477385, 15.289805 avg loss, 0.001000 rate, 5.985724 seconds, 51888 images, 2.176004 hours left
Loaded: 0.000167 seconds

 (next mAP calculation at 1100 iterations)
 Last accuracy mAP@0.5 = 72.52 %, best = 72.52 %
 1082: 16.080744, 15.368899 avg loss, 0.001000 rate, 5.904441 seconds, 51936 images, 2.169525 hours left
Loaded: 0.000050 seconds
```

그림 4-1. 학습과정중인 dataset

위의 그림은 현재 학습중인 코드 출력 값이며 iteration 1000번대 에서의 mAP와 평균loss과 인풋된 이미지의 수를 보여준다. mAP@0.5 의 뜻은 50%만 예측을 해도 정답으로 인정하겠다는 뜻으로 ground truth바운딩박스와 예측 바운딩박스의 교집합이 50%이상이면 물체를 예측했다고 하는 지표이다. loss값은 손실 값으로 정확도 값과 반비례합니다.

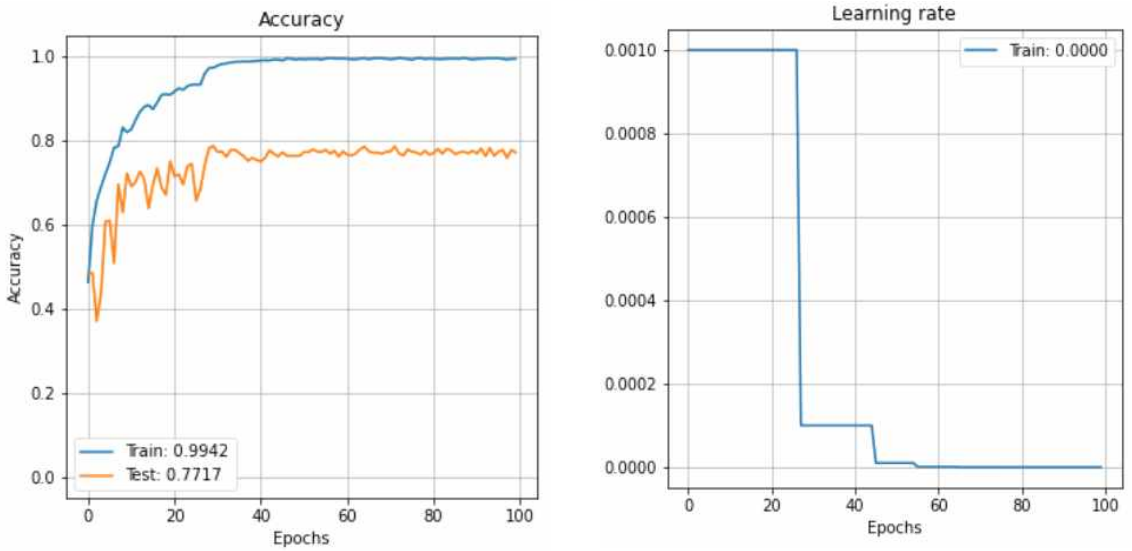

그림4-2. epoch에 따른 정확도와 learning rate

그림4-2 첫 번째 그래프에서는 학습 진행에 따른 학습 정확도와 테스트정확도를 나타낸다. 두 번째 그래프에서는 학습진행에 따른 lreaning rate의 감소율을 나타낸다. 최종 F1-score는 0.77을 기록했다. x축의 Epoch는 학습 횟수를 나타내는데 Iterataion과의 차이점은 전체 학습 횟수인지 gpu에 한번에 들어가는 이미지를 학습한 횟수인지에 따라 다르다. 즉, 1800장의 이미지를 배치사이즈 48로 설정했으므로 1800 / 48 = 37.5 이므로 38번의 Iteration이 한 번의 Epoch라고 할 수 있다.

4.2. opticalflow를 같이 활용한 실험결과

본 4.2 의 실험은 YOLOv4만을 통해 테스트한 결과의 인식률을 높이기 위해 Opticalflow를 접목시켰다. 객체를 인식한 boundingbox를 opticalflow포인터로 잡는 문제점이 생겨서 boundinbox는 지웠다.

Opticalflow를 사용해서, 영상 속에 움직이는 사람이 한 명일 경우와 또는 여러 명이 같은 속도의 움직임을 갖는 경우 100%의 정확도로 걷는지, 뛰는지를 판단하고 쓰러짐 판단까지 하였다. 그림4-3이 Opticalflow를 활용했을 때의 결과 값이다.

그림4-3. 한명일 때의 opticalflow결과

하지만 영상 속에 사람들이 서로 다른 방향을 향해 움직일 경우 opticalflow만으로 객체를 인식하고 라벨링 하여 움직임 판단이 불가능 하다.

그림4-4. 여러 명일 때의 opticalflow결과

그림4-4 와 같이 두 명이상이 서로 다른 속도로 움직일 때는 정확하게 물체의 흐름 속도를 추정하지 못한다. 결과 값인 text가 누구를 초점으로 잡고 결과를 도출하는지 알 수 없다.

이런 경우는 YOLOv4를 활용해서 해결할 수 있다. 객체인식의 장점은 객체를 인식함과 동시에 객체의 위치 값 까지 알 수 있다는 점을 활용했다.

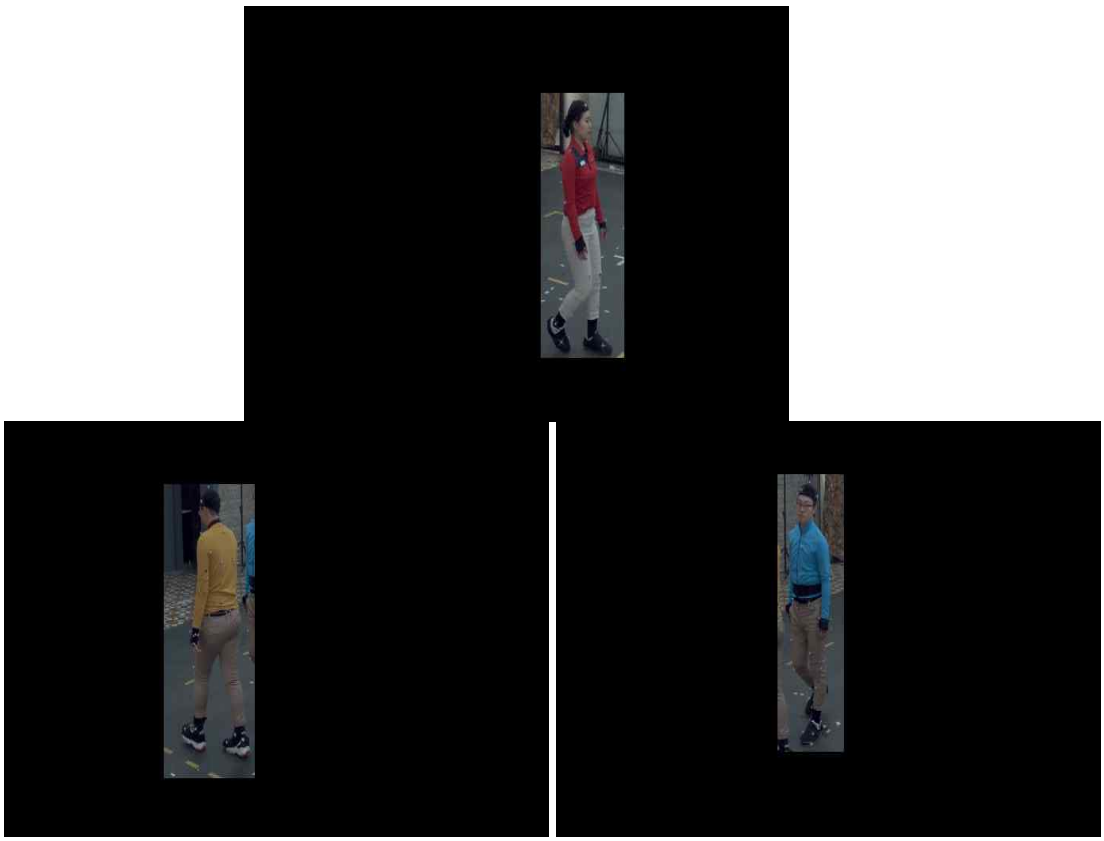

그림4-5. boundingbox 외에 검은화면 뿌리기

그림4-4의 문제점을 해결하기 위해 다음과 같이 객체로 인식된 물체의 바운딩박스의 중심좌표와 박스의 높이와 넓이 값을 알 수 있다. 그 값을 이용해서 바운딩박스 외에는 배경에 블랙화면을 뿌려서 각각의 이미지의 opticalflow값을 계산 후, 결과 값을 최종 출력 영상에 뿌려주는 방식으로 문제를 해결 할 수 있다.

그림4-6. 최종 출력영상의 결과

그림4-6은 최종 YOLOv4와 opticalflow를 붙여서 서로의 약점을 보완해서 출력한 이미지입니다. 각각의 검출된 객체마다 좌표 값을 받고 배경픽셀을 무시하기 위해 블랙화면을 뿌린 후, 픽셀의 움직임을 계산하기 때문에 계산양은 많지만 검출된 객체가 잘못 라벨링 된 경우를 정확하게 잡아낼 수 있다.

YOLOv4와 Opticalflow를 같이 사용함으로써, YOLO의 잘못된 객체 검출을 보완해서 결과 값을 출력할 수 있으며, 원하는 객체마다 opticalflow를 적용하기위해 바운딩박스의 좌표 값을 받아와서 분석을 하므로 영상 속 원하는 각각의 객체의 행동을 검출 할 수 있다.

5장 결론

5.1. 결론

본 연구는 이미지만으로 사람의 움직임과 방향을 예측할 수 있으며, 픽셀 움직임을 분석해서 쓰러짐을 판단할 수 있다.

본 연구를 통해 딥러닝 기반의 객체인식 시스템으로 사람의 동작이 검출 가능하다는 것을 보였다. 하지만 사람의 행동을 검출할 때 객체인식 시스템인 YOLOv4만을 활용할 때는 본 연구 환경에서 그림 4-2.처럼 77.17%로 낮은 정확도를 나타낸다. 이러한 낮은 정확도를 보완하기 위해 객체인식 시스템을 통해 각 라벨링 되는 객체의 중앙 좌표 값과 박스의 크기를 이용해서 배경에 블랙 픽셀을 뿌리고, 포인터를 객체에 집중시켰다.

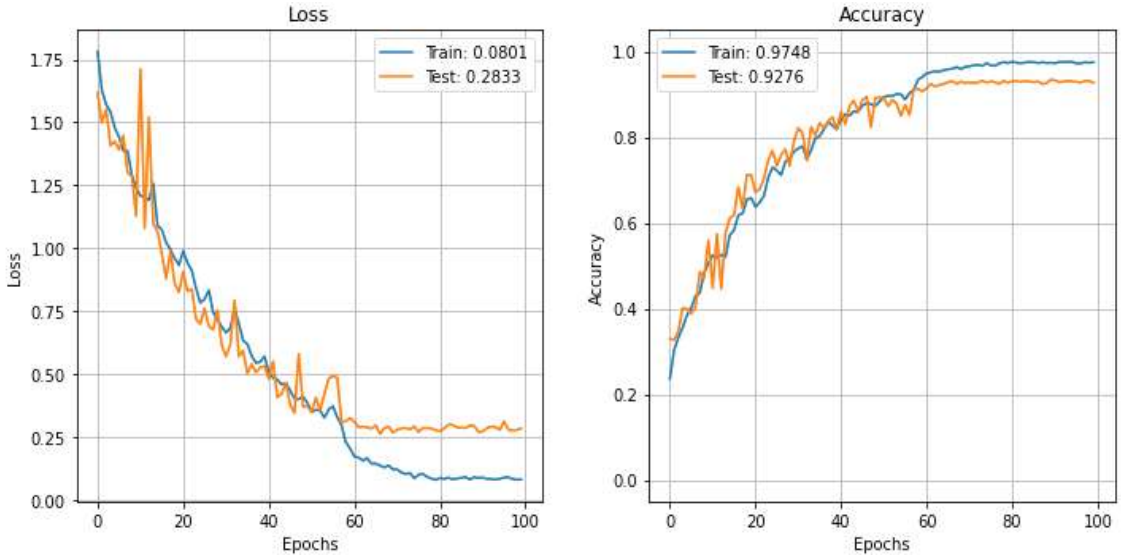

그림5-1. Opticalflow접목 후 최종 결과값

기존 77% 얻은 결과와 같은 테스트 데이터를 활용해서, opticalflow 코드와 배경에 블랙 픽셀 값을 뿌리는 코드를 접목한 결과 그림5-1과 같이 92.7%의 정확도를 도출해 냈다.

향후 연구에서는 학습데이터를 좀 더 다양한 환경에서 추출해서, 기존 객체 검출 정확도를 높이고 행동의 Class를 좀 더 세분화 해서 연구를 해 볼 수 있을 것이다.

참 고 문 헌

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788

[2] Joseph Redmon, Ali Farhadi, “YOLOv3: An Incremental Improvement”, arXiv:1804.02767, 2018

[3] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection”, arXiv:2004.10934, 2020

[4] Ross Girshick, “Fast R-CNN”, IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448

[5] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, Advances in Neural Information Processing Systems 28 (NIPS 2015)

[6]] J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013 - 2016.

[7] R Vinayakumar; K P Soman; Prabaharan Poornachandran, “Applying convolutional neural network for network intrusion detection”,International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017

[8] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, Jiaya Jia, “Path Aggregation Network for Instance Segmentation”, IEEE Conference on Computer Vision

and Pattern Recognition (CVPR), 2018, pp. 8759-8768

[9] Luis Alvarez, Joachim Weickert & Javier Sánchez, “Reliable Estimation of Dense Optical Flow Fields with Large Displacements”, International Journal of Computer Vision volume 39, pages41 - 56(2000)

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778

[11] Karen Simonyan, Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, arXiv:1409.1556, 2015

[12] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, Kaiming He, “Aggregated Residual Transformations for Deep Neural Networks”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1492-1500

[13] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, “Feature Pyramid Networks for Object Detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117-2125

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and ´C Lawrence Zitnick. Microsoft COCO: Common objectsin context. In Proceedings of the European Conference on Computer Vision (ECCV), pages 740 - 755, 2014

[15] Gabriele Bleser; Harald Wuest; Didier Stricker, “Online camera pose

estimation in partially known and dynamic scenes", IEEE/ACM International Symposium on Mixed and Augmented Reality 2006

[16] Berthold K.P. Horn, Brian G. Schunck, "Determining Optical Flow", Proceedings Volume 0281, Techniques and Applications of Image Understanding; (1981) https://doi.org/10.1117/12.965761

[17]Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich "Going deeper with convolutions", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9

[18] Hoo-Chang Shin; Holger R. Roth; Mingchen Gao; Le Lu; Ziyue Xu; Isabella Nogues; Jianhua Yao; Daniel Mollura, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning", IEEE Transactions on Medical Imaging Volume: 35, Issue: 5, May 2016

[19] Jinhui Lan, Jian Li, Guangda Hu, Bin Ran, Ling Wang; "Vehicle speed measurement based on gray constraint optical flow algorithm", ELSEVIER Volume 125, Issue 1, January 2014, Pages 289-295

[20] Joel Janai, Fatma Guney, Jonas Wulff, Michael J. Black, Andreas Geiger, "Slow Flow: Exploiting High-Speed Cameras for Accurate and Diverse Optical Flow Reference Data", ; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3597-3607

[21] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, I-Hau Yeh, “CSPNet: A New Backbone That Can Enhance Learning Capability of CNN”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020, pp. 390-391

[22] Tae Hyong Kim, Ahnryul Choi, Hyun Mu Heo, Kyungran Kim, Kyungsuk Lee, Joung Hwan Mun, “Machine Learning-Based Pre-Impact Fall Detection Model to Discriminate Various Types of Fall”, J Biomech Eng. Aug 2019, 141(8): 081010 (10 pages)

[23]Lai Jiang, Mai Xu, Zulin Wang, Predicting Video Saliency with Object-to-Motion CNN and Two-layer Convolutional LSTM, arXiv:1709.06316 [cs.CV]

ABSTRACT

Pedestrian behavior detection using YOLOv4 and optical flow

Yonghyun Kim

Bio Mechatronic Engineering

Sungkyunkwan University

Recently, as hardware performance and communication service technology are gradually developing, many studies that can be done with videos rather than just images are in progress. Among them, anomaly detection research has been actively conducted recently. Machine learning and deep learning are often used to detect abnormalities in voice, text, and image fields.

From 2012 onward, the number of unrelated deaths has been increasing year by year, and in 2019, 2,536 people are aged 65 or older, accounting for 45%. Manpower for solving these problems is insufficient and there are no special solutions. Recently, many types of camera services have appeared to care for infants, dogs, and the elderly. However, the problem is that although people are recognized, they cannot judge their actions and directions. It is difficult to

recognize a person's collapse with only an image without a wearable device for measuring heart rate or other external devices, since it is not accurate to judge abnormal behavior such as a fall with only the image. In this study, by grasping the object recognition system and the pixel movement of the object, it is possible to distinguish walking and running, and to determine the direction, and by analyzing the movement of the pixel, it is possible to determine whether or not to fall without an external device.

In this paper, the accuracy of the recognition rate was improved when dividing the behavior of pedestrians into walking and running in the image, and accurately judge the collapse in the situation of falling while walking. The technology used in the recognition system uses the YOLOv4 model to detect objects in images released in April of this year, and learns behavior data by dividing it into three classes: walking, running, and falling. After that, in order to compensate for the decrease in the recognition rate when the behavior changes, a method of increasing the accuracy of abnormal behavior detection by incorporating Opticalflow into the recognition system was proposed. Through this, the change of running from walking can be detected through pixel motion comparison between frames, and even when recognizing a motionless action such as a fall, it can be recognized more accurately than before.

In this study, the pretrain model extracted by learning YOLOv4's Cocodataset was tested based on the extracted custom_model by transfer learning on the image data of a total of 1800 annotations (bounding box), each of 600 walking, running, and falling images.