**스크립트**

1. (인사)

-안녕하세요 저희는 이번에 social distancing analyzing model을 소개하게 된 5조 18 이민권, 20 문예훈입니다. 먼저 저희가 어떠한 내용에 대해 소개할 것인지 목차를 통해 간략하게 설명해드리겠습니다.

2. 목차

3. 어떤 프로젝트인가?

-앞서 말씀드렸던 것처럼 저희는 social distancing analyzing model에 대해 설명해드릴 것입니다. 먼저 Social distancing analyzer는 해당 영역에 대해 사회적 거리두기 프로토콜이 준수되는 정도를 자동으로 감지하는 모델입니다. CNN, openCV 등을 이용해 사람들 사이간의 거리를 계산해서 거리를 유지하고 있는지 아닌지를 판단하는 것입니다. 먼저 이 모델을 이용해 분석한 영상을 보도록 하겠습니다.

4. 시연영상

-시연영상 재생

5. 계기

-2020년 초 부터 코로나 바이러스로 인해 정부에서는 사회적 거리 두기라는 정책을 시행했습니다. 하지만 사회적 거리두기가 제대로 시행되고 있는가를 생각해보면, 뉴스만 보더라도 그렇지 않다는 것을 알 수 있습니다. 이유는 다양할 수 있으나, 몇몇 사람들은 주변에서 이 정책이 제대로 시행되지 않고 있는 경우들을 볼때면(예. 5인 이상 소모임, 마스크 착용 등) 심리적으로 많이 불안해 하기도 합니다. 따라서 이번 프로젝트를 통해 사람들이 몰려있거나, 자주 다니는 곳 등의 영상을 모델을 통해 테스트하고 사람들이 얼마나 거리유지를 잘 하고 있는지를 판단하는 Social Distance 분석 모델이 어떻게 작동하고 설계되어있는지 배워보고 또한 한동대학교 안의 사회적 거리두기 실태를 파악하고자 이 프로젝트를 시작하게 되었습니다.

6. 인공지능 → CNN 사용 / CNN이란?

-전체적인 코드에 대한 설명에 앞서 먼저 물체를 인식하고 처리하는데 중요한 역할을 해주는 CNN이라는 뉴런 네트워크에 대해 설명해드리겠습니다. CNN은 Convolutional Neuron Network의 약자로 인간의 시신경을 모방하여 만든 딥러닝의 구조 중에 하나 입니다. CNN은 패턴을 찾는데 유용해 이미지 장면을 분류하기도 하며 물체를 감지해 분할하는 등 다양한 이미지 분석 관련 작업을 학습해 처리해줍니다. CNN은 특징을 추출하는 convolution layer와 추출된 특징을 부가 샘플링하는 pooling layer로 구성되어있습니다.

먼저 Convolution layer를 구하는 것에 있어서 중요한 것은 합성곱입니다. 합성곱의 정의는 반전 이동한 값과 두개의 서로 다른 함수를 곱한 다음 적분해 새로운 함수를 구하는 수학 연산자입니다. (그림 넣고) 단일 채널에 대한 합성곱 계층 동작을 표현해보면 (그림 넣고) 여기 보시는 것처럼 입력 데이터 (4X4) 매트릭에 필터를 적용해 N칸씩 옮기면서 계산해 Stride라는 보폭 값을 얻게 되고 최종적으로 feature map을 얻을 수 있는 것입니다. 즉, 필터를 이용해 이미지 특징을 뽑을 수 있게 됩니다.

근데 이런 식으로 하다보면 이미지의 크기가 작아지면서 가장 자리의 픽셀의 정보를 얻기 힘들어지는 문제점이 생기기도 하지만 padding을 이용해 이미지의 가장자리에 특정 값의 픽셀을 추가해 입력 이미지와 출력 이미지의 크기에 큰 차이가 없게 만들어 해결할 수 있습니다.

그리고 이러한 단계들을 통해 정돈된 이미지의 특징을 pooling layer를 통해 특정 데이터를 강조 혹은 핵심적인 데이터를 추출합니다.

전체적인 단계를 요약하자면 입력 데이터가 처음 들어오면 Convolution layer에서 이미지의 특징을 추출 하고 추출된 이미지의 특징을 pooling layer 거쳐 세부적으로 크기 조정, 축소, 특징 강조하는 것입니다. 이러한 과정들을 반복해 결과값을 출력하는 것입니다.

https://velog.io/@seongguk/AI-CNNConvolutional-Neural-Network-%ED%95%99%EC%8A%B5

https://www.aitimes.kr/news/articleView.html?idxno=11294

https://ko.wikipedia.org/wiki/%ED%95%A9%EC%84%B1%EA%B3%B1

https://sungwookkang.com/1408 (여기서 주로)

7. 필요한 모듈,오픈소스 (cv2, darknet), github 주소 등등 다운받는 방법

이 모델을 실행하기 위해 필요한 모듈, 오픈소스가 있습니다. 저희가 기본적으로 필요한 것은 먼저 메인 모델, yolov3.weight, yolov3.cfg 그리고 openCV가 있습니다. 저희 같은 경우에는 이 모델을 github에서 가져왔기에 (사이트 주소 가리키며) 이 사이트에서 한번에 다운받을 수 있었습니다.

먼저 이 깃헙 링크로 들어가셔서 가장 중요한 모델을 clone합니다. 그리고 사이트의 밑으로 내리다보면 Installation 부분이 있을 것입니다. 여기서 두개의 파란생 링크로 들어가셔서 yolov3.weight와 .cfg를 다운 받으시면 됩니다. openCV 같은 경우에는 위의 방법처럼 다운 받는 것이 아니라 cmd를 이용해 다운받습니다. 먼저 cmd 혹은 anaconda prompt를 관리자 권한으로 실행시킨 뒤 명령창에 pip install opencv-python이라고 치신 뒤 다운이 다 받아질때까지 기다리시면 됩니다.

일단 여기까지 모델을 실행시키기 위한 사전 준비는 끝이 났습니다. 이제부터는 이 모델이 어떻게 작동되는지 코드를 통해 살펴보겠습니다.

8\_1. 코드설명

이제 Social Distancing Model이 어떻게 실행되는 것인지 알아보겠습니다. 먼저, 필요한 모듈들을 불러옵니다. Time, numpy모듈과 이 프로젝트의 핵심인 open cv 모듈인 cv2를 import해줍니다.

그 다음에는 input에 사용될 video를 불러오는 작업을 해줍니다. 이 input video의 제목을 ‘video.mp4’라는 형식의 이름을 입력하면 지정해둔 경로에서 비디오 파일을 불러옵니다.

8\_2

그 후에는 몇가지 함수를 선언해줍니다. calibrated\_dist라는 거리를 계산 해주는 함수와 isclose라는 함수가 선언이 되어 있는데요, dist함수는 예상하다시피 두 인자를 가져와서 거리를 계산해주는 함수입니다. 지금은 이미지를 다루기때문에 픽셀에 대해서 거리를 계산해줍니다.

그리고 isclose 함수는 거리계산한 값을 바탕으로 두 좌표의 거리가 가까운지 판단하는 함수입니다. 그 거리 값에 대해 return 값을 3종류로 반환해 나중에 사람 간의 거리가 어떤지를 판단합니다.

8\_3

그 다음에는 ./coco.names라는 파일을 label로 가져오게 되는데, 이것은 이미지에 포착된 객체들의 이름 리스트들입니다. 즉, 이미지에서 보이는 사물이 무엇인지 출력을 해주기 위한 리스트들입니다. 이 label들을 저장하고, weights와 configuration file을 가져옵니다. 이 두개의 파일은 darknet이라는 오픈소스에서 가중치 값과 Neural net 구조를 가져옵니다. 이것은 opencv에서 딥러닝을 하는 과정에서 필요한 값들로, 저희가 힘들게 layer들을 구성하고 pooling을 설정하는 것이 아니고, 미리 구현된 구조를 러닝에 사용하게 됩니다. 그래서 그 값들을 ‘readNetFromDarknet’의 인자로 주어 opencv 전용 신경망을 생성하게 됩니다.

신경망을 생성한 뒤에는 불러온 video를 opencv의 input으로 사용하기 위해서 VideoCapture를 해줍니다.

8\_4

이제 불러온 video를 각 프레임별로 잘라줍니다. 그리고 이미지의 크기, 즉 shape을 저장해 줍니다. 그리고 여기 blobFromImage라는 method가 있는데, 이 method는 각 프레임에 속한 객체, 즉 포착한 사물이나 사람을 뽑아내는 메소드다. 그래서 이 객체들을 blob에 저장하고, 이 객체들이 신경망의 input이 됩니다. 그래서 주어진 신경망의 input으로 객체값을 넣어주고, forward propagation시켜 학습을 시키면, layer Output이 출력됩니다. 이것은 각 객체가 어떤 사물일지에 대한 확률에 해당합니다.

이제 신경망에 output이 person일 경우를 살펴봅니다. person이 포착되었을 경우, ‘사람’객체에게 씌울 ‘박스’를 생성합니다. 그 박스가 위치할 좌표를 구하는 것이 아래 for문의 목적입니다. 이 박스는 output에서 사람의 위치를 알려줄 것입니다. 값들을 얻으면 NMSBoxes라는 opencv 메소드로 입력해줍니다.

8\_5

다음으로는 프레임 내에 사람이 포착되었을 경우 기록되는 내용에 대해 말씀드리겠습니다. 이 if문 안에서는 사람이 있는 위치에 씌워질 박스의 중간 지점을 계산하고 저장 한 뒤, 이 두 값의 거리를 계산하여 많이 가까우면 high\_risk값을 올리고, 덜 가까우면 low\_risk값을 올리고, 적당한 거리를 유지하면 safe하다고 판정합니다.

8\_6

여기에 보이시는 굉장히 긴 코드는 복잡해 보이지만 사실 frame 별로 박스와 박스간의 선을 이어주는 작업을 단순 나열한 것입니다. 박스의 개수만큼 iteration을 하여 output에 그려질 박스와 선을 만들어줍니다. 그리고 출력 단에서 표시될 highrisk lowrisk safe 수 와 전체 포착 인원 수 등을 표시해주는 텍스트를 만듭니다.

8\_7

마지막으로는 output gif 혹은 avi를 생성해줍니다. 이 출력에는 이제 사람들이 위치한 곳에 박스표시가 되어있고, 사람이 가깝다면 선이 표시되며 얼마나 사람들이 가까이 있는지를 표시해줍니다. 다음 보실 영상은 한동대학교 복지동 아래 사람들이 모여있는 영상에 대한 출력입니다.

9. 시연영상 2

이렇게 인식하는 것을 확인할 수 있습니다.

10. 어려웠던 점, 한계

다음은 이 모델의 한계와 겪었던 어려움에 대해 말하겠습니다.

이 프로젝트의 한계는, 사람들이 인식될 수 있는 범위가 제한되어있다는 것입니다. 사람이 인식되기 위해 다른 사물과 혼동되지 않고, 명확하게 표현이 되어있어야 하는데 마스크를 끼는 등의 혼동을 줄 수 있는 요소가 있다면 사람이 어디에 있는지를 모델이 잘 학습하지 못할 수 있습니다. 그 결과 거리가 가깝더라도 safe하다고 판정을 내릴 수 있는 것입니다.

또한, opencv가 output 영상을 출력하는데 코덱을 고려해주어야 합니다. 새로운 비디오로 학습을 시킬 때에는 코덱 에러로 많은 어려움을 겪었었는데, 해당 비디오 input에 맞는 코덱과 파일 이름을 설정해주어야 했습니다.

11. 기대효과

마지막으로 기대효과에 대해 말씀드리겠습니다. 이 모델로 미루어 볼 수 있는 기대효과로는, 5인 이상 집합 금지 명령에 의해 5인 이상 모임을 찾아내는 기능을 추가하여, 그 모임에게 경고하고 코로나 확산 방지할 수 있는 가능성을 높일 수 있습니다.

또한, 이 모델을 시스템을 조금 더 개발시켜 자주 사람들이 모이는 곳의 cctv 영상을 보고 거리유지를 잘 지키고 있는지 실시간으로 확인할 수 있다. 그럴 경우 실시간으로 영상 데이터를 train되어 송출 될 수 있도록 시스템을 구성해야 할 것입니다.

12. 감사합니다

네 이것으로 5조 모인활 프로젝트 발표를 마치겠습니다. 들어주셔서 감사합니다.