|  |  |
| --- | --- |
| 교육 제목 | **OpenCV DNN practice (Final)** |
| 교육 일시 | 2021년 12월 28 일 화요일 |
| 교육 장소 | 영우글로벌러닝 2층 |
| **교육 내용** | |
| 오전 | 1. Understanding functions forward() and getUnconnectedOutLayers()    1. getUnconnectedOutLayers() is used to obtain indexes of the unconnected output layers in order to find out how far function forward() must run through the network.    2. getUnconnectedOutlayers()       * 연결되지 않은 출력 레이어들의 인덱스들을 구하기 위해서 사용됨       * 즉, 얼마나 멀리 forward90 함수가 진행했는가를 확인 할 때 사용 2. net.getUnconnectedOutLayers() gives the position of the layers. the output is an ndarray of shape (1,). So to get the integer we do ln[0]. And to get the index we subtract 1 from the position. 3. Net.getUnconnectedOutLayers()는 레이어들의 위치를 알려준다. 출력은 shape(1,0)로 ndarray이다. 따라서 integer를 얻기위해서는 ln[0]으로 해야한다고 생각한다. 그리고 우리가 원하는 층까지의 개수는 -1을 해야 함 4. YOLO Architecture    1. YOLO (You Only Look Once)는 학습한 물체의 종류와 위치를 실시간으로 파악 할 수 있는 Real-Time Object Detection 모델로 다음과 같은 아키텍쳐로 구성되어 있습니다.   YOLO v3 Architecture<br>출처 - <a href=https://www.cyberailab.com/home/a-closer-look-at-yolov3>https://www.cyberailab.com</a>   1. YOLO v3 Architecture 2. YOLO Output    1. YOLO 모델의 output은 N x N x (C + 5) x B) 형태의 텐서(tensor)를 반환하며, 이 정보를 이용하여 이미지 상에 존재하는 물체의 종류와 2D 좌표를 획득할 수 있습니다.    2. N x N x (C + 5) x B)    3. N x N : Grid    4. C : 학습한 Object 수(Classes)    5. 5 : 탐지한 Object의 Bounding Box 정보(x, y, w, h, confidence)    6. B : Grid cell당 박스 수 3. YOLO 기본 원리    1. 간단히 설명하자면, 예측하고자 하는 이미지를 SxS Grid cells로 나누고 각 cell마다 하나의 객체를 예측한다.    2. 그리고 미리 설정된 개수의 boundary boxes를 통해 객체의 위치와 크기를 파악한다.    3. 이때, 각 cell마다 하나의 객체만을 예측할 수 있기 때문에 여러 객체가 겹쳐있으면 몇몇의 객체는 탐지를 못 하게 될 수 있다. 4. 그럼 이제 각 cell에서 어떻게 객체를 예측하는지 살펴보자.    1. 각 cell은 다음 조건 하에 예측을 진행한다.    2. B개의 boundary boxes를 예측하고 각 box는 하나의 box confidence score를 가지고 있다.    3. 예측된 box 수에 관계없이 단 하나의 객체만 탐지한다.    4. C개의 conditional class probabilities를 예측한다. 5. Non-maximal suppression (NMS)    1. 같은 객체에 대해 여러 개의 탐지(예측된 boundary boxes)가 있을 수 있다. 이것을 고치기 위해, YOLO는 더 낮은 confidence를 가진 중복된 것을 제거하는 non-maximal suppression을 적용한다.    2. 구현 방법은 다음과 같다.       * confidence score 순으로 예측을 정렬한다.       * 제일 높은 score에서 시작해서, 이전의 예측 중 현재 예측과 클래스가 같고 IOU > 0.5인 것이 있었으면 현재의 예측은 무시한다.       * 모든 예측을 확인할 때까지 Step 2를 반복한다. |
| 오후 | * 1. YOLO의 장점   2. (1) YOLO는 매우 빠르다.      1. 우리가 detection을 single regression problem으로 정의하는 것을 통해서, 우리는 복잡한 pipeline이 필요하지 않다.   3. (2) YOLO는 image를 예측시에 globally하다는 특징을 가진다.      1. sliding window나 region proposal 기반의 기술들과는 다르게 학습이나 테스트 중에 전체 이미지를 보기 때문에, 은연중에 각 class의 대표적인 표현들에 대해서 맥락적인 정보를 encoding 한다.      2. 강력한 detection방법인 Fast R-CNN경우에는 배경정보에 따라서 object를 detect하는데 있어서 실수가 있는데, 이것은 Fast R-CNN이 큰 맥락을 보지 못하기 때문이다.      3. YOLO는 Fast R-CNN과 비교했을 때, background error가 절반 정도 밖에 되지 않다.   4. (3) object의 일반적인 특징을 학습한다.   5. object의 일반화를 잘한다.      1. 예술작품이나 자연의 이미지를 학습하거나 테스트할 때, R-CNN같은 강력한 detection방법들보다 학습을 더 잘한다.      2. YOLO는 일반화에 강력하기 때문에, 새로운 도메인이나 예상하지 못한 input이 들어와도 망가질 확률이 적다.   6. Unified Detection:      1. 네트워크의 큰 특징 중 하나는 이미지 전부로부터 features를 뽑아서 각 bounding box를 예측한다는 것이다.      2. 시스템은 input image를 S x S grid로 나눈다.      3. 그리고 만약 어떤 object의 한 가운데가 grid cell에 놓인다면, grid cell은 그 object를 탐지할 의무가 생긴다는 설정이다.      4. 각 grid cell은 B개의 bounding boxes를 예측하고 그 box의 예측 대한 confidence score를 같이 예측한다.      5. confidence score은 얼마나 박스 안에 실제로 object가 존재하는지, 그리고 그 class를 얼마나 잘 반영했는지에 대한 것이다.      6. Confidence 는 형식적으로 Pr(Object) \* IOU(intersection over union) 라고 정의한다.      7. ( IOU란 정답 box와 예측 box의 교집합 크기 / 합집합 크기 이다 )      8. Object가 box안에 아무것도 없을 때에는 confidence가 0이 되어야 한다.      9. 또, predicted box와 ground truth의 IOU가 얼마나 일치하는지가 confidence가 되도록 맞추고 싶다.      10. https://blog.kakaocdn.net/dn/bbqAe7/btqDGgFFgOh/ntjDnIkIlhDyBI1A1aik0K/img.png   7. Objectness score가 confidence값이다. |