1. image2solution 모델 구축

본 논문에서는 이미지를 입력값으로 하여 수송모델(추후 수정 가능)의 최적해를 도출하는 과정을 자동화한 image2solution 모델을 제안한다. 모델의 전체적인 프로세스는 ***<Figure 1>***과 같다. image2solution은 다음의 5단계의 과정을 거친다. 먼저 (주제와 관련된) 이미지를 수집한 후 이미지에 존재하는 객체(object)마다 수작업으로 주석을 달아준다. 둘째, 객체 검출 알고리즘 중 실시간 탐지가 가능하면서도 성능이 우수하다고 알려진 YOLO(You Only Looks Onces) 알고리즘을 이미지에서 올바르게 객체들을 검출하도록 학습시킨다. 셋째, YOLO의 결과로서 산출된 이미지의 객체정보를 seq2seq 알고리즘의 입력값으로 하여 이미지에 대한 캡션을 생성한다. 넷째, 각 이미지마다 생성된 캡션 이용하여 MySQL 데이터베이스에서 관련 데이터들을 조회하여 파이썬 변수로 할당한다. 마지막으로 변수에 담긴 값을 이용하여 선형계획법(Linear Programming, LP)의 최적해(최저 비용)를 산출하고 화면에 그 값을 출력한다.

1.1. 이미지 수집 및 주석표시(labeling)

***[국제 운송이 중요한 이유, 예: 국가간 무역의 증대]*** 국제운송의 형태로 해상운송(shipping), 육상운송(land trasnportation), 항공운송(air transportation)있다. 이 중 항공운송은 주로 소형, 경량의 고부가가치 제품을 취급하며 운송기간이 짧아 정시성이 높다는 특징 및 장점이 있다. 그러나 다른 운송형태에 비해서 운송비용이 높다는 단점이 있다. 따라서 항공운송 비용의 최적화는 최대 이윤을 추구하고자하는 기업입장에서 주요 관심대상이다. 이에 착안하여 공항간의 수송비용 최적화 문제를 자동화하기 위한 학습 대상인 이미지를 수집하기로 하였다. 이미지 수집 과정은 다음과 같다.

첫째, 미국 공항 중에서 주요 5개의 공항으로 Dallas, Chicago, Los Angeles, New York, Miami 공항을 선정하였다. 선형계획법의 풀이를 단순화 하기 위해서 5개 공항 중 Dallas, Chicago 공항은 '출발' 공항으로, Los Angeles, New York, Miami 공항은 '도착' 공항으로 분류하였다.

둘째, 대량의 이미지를 자동으로 수집하기 위하여 파이썬 기반의 크롤러(crawler)를 개발하였다. 크롤러는 구글 이미지 검색 페이지에서 [공항이름+airport+이∙착륙상태+plane]과 같은 규칙에 기반한 질의(search query)를 통해 공항 당 10장, 총 100장의 이미지를 수집한다. 이와 같은 규칙을 생성한 이유는 출발 공항의 경우에 공항 전경에 이륙하는 항공기가 함께 나타야하며 도착 공항의 경우에도 공항 전경에 착륙하는 항공기가 함께 나타나야 함을 강제하기 위함이다. 그러나 수집된 이미지 중 연구 목적과 맞지 않는 이미지가 있을 수 있기 때문에 연구자가 직접 확인하여 그러한 이미지를 제거하는 과정을 거친다. ***<Table1>***은 수집된 이미지와 사용여부에 대한 예시를 보여준다.

셋째, 수집한 이미지를 YOLO의 입력데이터로 활용하기 위해서는 각 이미지마다 객체의 위치를 나타내는 경계박스를 그리고 그 객체의 이름을 나타내는 주석을 표시해야한다. 본 연구에서 탐지하고자 하는 객체는 공항이름과 비행기의 이∙착륙 상태이다. 주석을 표시하기 위해서는 오픈소스 라벨링(labeling) 도구인 'labelImg'를 사용하였다. 주석 표시 예시는 ***<Figure 2>***에 나타나 있다.

1.2. YOLO를 통한 이미지의 객체 검출

YOLO(You only look once)는 CNN기반의 객체검출(object detection) 알고리즘이다. region proposal base 모델에 비해서 검출 정확도는 떨어지나 속도면에서 우위가 있다. (특정 객체가 이미지의 어느 부분에 있는지에 대한 지역정보(localization)를 가질 수 있어서 성능 향상에 유리할 것이라는 판단.) YOLO는 오픈소스로 모든 코드가 공개되어 있으므로 본 연구에서는 이를 활용하였다. 학습반복횟수는 N회로 하였고 이미지 검출 결과는 아래 사진과 같다.

1.3. seq2seq를 통한 이미지 캡셔닝(captioning)

seq2seq 알고리즘은 딥러닝 알고리즘 중 하나인 RNN에 기반한 것이다. RNN은 다양한 상황에서 활용되는데 이러한 RNN 알고리즘을 2개를 이어 붙여서 machine translation에 주로 활용된다. 본 연구에서는 seq2seq의 특성에 주목하여 단순한 객체명을 입력 데이터로하여 이미지에 대한 캡션을 출력할 수 있다고 판단하였다.

1.4. MySQL 데이터베이스에서 관련 데이터 추출

DBMS에 SQL 쿼리를 이용하여 관련 데이터를 꺼내온 후 이를 이용하여 수송모델의 최적해를 구하는 단계이다. 자세한 과정은 다음과 같다. 먼저 DBMS으로 MySQL을 사용하여 데이터베이스 환경을 구축한 후 데이터베이스 내부에 제품의 단위당 운송비용, 공급량, 수요량 테이블을 작성하였다. 그 다음 파이썬의 pymysql 모듈을 활용하여 위에서 구축한 데이터베이스에 접근한다. 데이터베이스에 접근한 후에는 SQL 쿼리 문법을 이용하여 각 테이블에서 필요한 정보를 추출하여 파이썬 변수에 할당한다.

1.5 선형계획법을 통한 최저비용과 최적 운송량 산출

마지막으로 추출된 정보들을 파이썬의 pup 모듈을 활용하여 선형계획법의 입력 데이터로 할당하여 최적해를 도출하고 그 결과를 엑셀파일로 최종 출력한다.