**Flash21 인수인계**

**-관광명소 인증 앱 사용을 위한 데이터 학습 방법-**

1. 대상 장소들의 이미지 자료구하기

인증시스템에 적용할 학습데이터를 만들기에 앞서서, 학습의 베이스가 될 이미지가 필요합니다. 이미지는 직접 촬영또는 인터넷 이미지를 가져오는 방식으로 진행합니다. 데이터 학습의 오차를 줄이기위해서는 여러가지 사진 효과들이 들어가지 않고 원상태의 이미지가 필요합니다. 그러므로 이미지는 되도록 원본 상태로 가져오는 것이 좋기 때문에 이미지의 수집이 끝난 뒤에는 전반적인 확인 및 삭제가 필요합니다. 이번 관광명소 인증앱은 인식시스템을 이용하지만, 대상에 대한 인증이 주요기능이기 때문에 이미지는 비슷하거나 미세한 차이라도, 흐릿하거나 깨지지만 않았으면 학습에 사용하시면 됩니다(인증이 중요한 부분이기에 적정량의 중복 이미지도 사용가능합니다.). 인증해야 할 대상이 여러 개라면 각각의 대상에 대한 이미지의 양은 비슷하게 준비해주는 것이 학습결과를 좋게해줍니다. 예시로 대상 A,B가 있다면 A의 이미지가 500장이면 B도 비슷한 양인 500장 정도로 준비하는 것이 좋습니다. 이 때 어느 정도의 장수 차이(50장 내외)는 괜찮으니 완벽히 이미지의 수를 맞추어 준비하지는 않아도 됩니다. 추가적으로 비에 젖어있거나 한밤중에 찍은 이미지를 추가할 경우에는, 상태에 따른 대상의 이미지의 장수가 서로 균일하게끔 처리해주는게 좋습니다.

1. ROBOFLOW에서 라벨링 하기

1번의 과정을 통해서 학습을 위한 이미지 자료가 모였습니다. 모은 이미지들은 컴퓨터에 장소 혹은 대상 별로 정리를 해놓으면, 라벨링 작업을 수월하게 하실 수 있습니다.. 자료가 정리되었다면 일단 roboflow라는 사이트에 접속하여야 합니다.

주소: <https://roboflow.com>

|  |
| --- |

<접속 시 보게 될 이미지>

사이트에 접속을 하게 되면 위의 화면을 보실 수 있습니다. 위 사이트는 학습데이터를 만들기 전에 ‘라벨링’ 작업을 할 수 있게 해주는 사이트로 앞으로 자주 스시게 되실 겁니다. 작업에 들어가기에 앞서서 사이트 이용에는 회원가입이 필요합니다. 가입은 오른쪽 위에 Sign up 버튼을 눌러서 회원가입을 해주시고, 이후에는 사이트 사용시에 로그인하시면 되겠습니다.

|  |
| --- |

<로그인 시 보게 될 첫 화면>

로그인에 성공하셨다면, 위의 화면이 보이실 겁니다. 이 화면에서는 새로운 프로젝트를 만들어서 학습을 위한 데이터 만들기를 진행할 수도 있고, 진행 중이거나 진행했었던 프로젝트에도 추가적인 이미지 자료를 넣어서 새로운 버전의 데이터를 만들 수도 있습니다. 이제부터 예시로 경북대학교 인증프로젝트를 위한 데이터를 만들어보면서 어떻게 진행해야 하는지 알려드리겠습니다.

\*추가적인 정보\*

초기화면의 오른쪽 위를 보면 invite가 있습니다. 여기서 라벨링 작업을 같이 진행할 팀원들을

이메일로 초대하면, 이후 프로젝트 진행시 작업물을 분배하여 할당할 수 있습니다.

위의 화면에 보이는 Create New Project를 클릭해봅시다.

|  |
| --- |

<클릭 시 보이는 화면>

이제 위의 ‘Create Project’라는 화면이 보일 것입니다. 우리는 경북대학교 인증프로젝트였으니 프로젝트 이름은 ‘knuCertification’, 라이선스와 프로젝트 타입은 위와 같이 놔두면 되며, 마지막 ‘What will your model predict?’라는 칸에는 전체적으로 인증할 사물이나 장소에 대한 큰 틀(주제)을 적으면 되며, 우리는 경북대학교 장소 인증을 위한 것이니 Landmark라고 쓰겠습니다.

작성이 끝난 뒤 Create Public Project를 누르시면 앞으로 진행할 프로젝트가 생성됩니다.

|  |
| --- |

<프로젝트 생성 후 보게 될 화면>

프로젝트 생성 후에는 이제 라벨링 작업을 진행할 사진을 프로젝트에 업로드 해주어야 합니다. 사진의 업로드는 위의 중앙에 보이는 select files나 select folder를 클릭해야 자료를 선택하거나 파일에서 드래그하셔도 됩니다.

|  |
| --- |

<사진 파일 혹은 폴더를 선택한 후에 화면>

사진을 선택하여 올렸다면 오른쪽 위의 Finish Uploading이라는 버튼이 활성화되어 있을 겁니다. 버튼을 눌러서 업로드를 합시다. \*업로드 하기전에도 라벨링 작업이 가능하지만, 이 상태에서 할 경우 튕김이 위험이 있습니다. 작업은 업로드까지 하고 나서 하시면 안전하게 작업하실 수 있습니다.\*

|  |
| --- |

<업로드 버튼을 누르고 난 뒤 보게 될 화면>

업로드를 하면 위의 화면이 보이며, 왼쪽의 프로젝트 요약부분에서 upload에서 Annotate로 현재 위치가 바뀐 것을 알 수 있습니다. 여기서 오른쪽에 떠있는 Assign Images에서 작업할 사진을 누구에게 얼마만큼 할당할지를 정할 수 있습니다.. 현재는 팀이 아닌 단독 진행 기준이기에 팀이 1명으로 뜨며, 여기서 밑의 Assign Images 버튼을 눌러 작업할 사진을 할당해 줍시다.

|  |
| --- |

<할당 후 보이게 되는 화면>

작업할 사진들을 할당 하였다면 위의 화면이 보일 것입니다. 여기서부터 ‘라벨링’ 작업을 시작하게 되는데 중간중간 나가도 저장되니 작업도중 파일의 손상은 걱정안하셔도 됩니다. Unannotated는 아직 작업하지 않은 것, Annotated는 작업한 것입니다. 이제 이미지를 하나 클릭해봅시다.

|  |  |
| --- | --- |
|  | |

<경북대학교 석탑을 tower라는 클래스의 라벨링 작업하는 과정>

이미지를 클릭하면 위의 화면이 보이게 될 것입니다. 마우스는 위의 이미지에서 보이는 십자 중앙에 위치합니다. 이를 사진에서 인식할 부분에 대해서 네모 안에 다 들어오게 드래그해 봅시다. 이때 대상의 전체적인 모습이 네모 안에 다 들어오되 너무 네모를 너무 크게까지 키우실 필요는 없습니다. 드래그하면 2번째 이미지처럼 왼쪽 위에 Annotation Editor가 뜰 것입니다. 우리는 지금 석탑을 라벨링 중이니 tower라고 클래스를 선언합시다. 그리고 세이브 버튼을 누르면 첫 번째 사진 라벨링이 끝났습니다. 이 방법은 이후 다른 사진들 작업 시에도 동일하게 진행하며, 하나 유의할 점은 다른 장소나 사물을 라벨링 하게 될 경우 아까의 Annotation Editor에서 그에 대한 클래스 명을 작성해야 합니다.

이때 새로이 작성되는 클래스 명이라면 왼쪽의 그룹 설명에 클래스 리스트에 아래로 나오게 되는데, 이 리스트의 순서를 기억하고 있으면 안드로이드 스튜디오 앱 실행에 필요한 설정을 변경하는데 쓰입니다.

|  |  |
| --- | --- |
|  |  |

<라벨링 작업이 끝난 뒤, 데이터 셋으로 보내고 이후의 결과에 대한 화면>

할당된 이미지에 대한 라벨링 작업이 끝났다면, 크롬창이 아닌 라벨링 작업의 보라색 뒤로가기 버튼을 누르면 위의 화면으로 넘어갑니다. Annotated에 이미지 3개가 옮겨져 있으며, 오른쪽 위에 이미지를 데이터셋으로 옮기기라는 버튼이 활성화 되어있을 겁니다. 이제 이 버튼을 누르면 두 번째 이미지의 데이터셋으로 보낼 때의 방식이 있는데 기본으로 되어있는 Split Images Between Train/Valid/Test를 내버려두고, 분배 퍼센트는 건드리지 않고 아래의 Add Images를 클릭하여 데이터 셋으로 보냅니다. 이제 데이터셋의 항목으로 들어가면 아까의 3개의 이미지가 설정된 분배 퍼센트만큼 나뉘어져 추가된 것을 볼 수 있습니다. 이제 이를 이용하여 데이터 파일을 생성하겠습니다.

|  |
| --- |

<왼쪽의 개요 부분에서 버전을 누르면 보이는 화면>

예시로 보여준 이전의 과정에는 이미지가 적으므로 기존의 프로젝트에서 파일 생성을 알려드리겠습니다. 일단 왼쪽의 프로젝트 개요에서 Versions를 클릭합니다. 그러면 위의 화면이 나오고 이중에서 파일 생성을 위해서 Generate New Version을 클릭합니다.

|  |  |
| --- | --- |
|  |  |

클릭하면 위와 같이 Source Images, Train/Test Split, Preprocessing, Augmentation, Generate가 보일 것입니다. 여기서 앞의 2개는 수정 없이 진행하고, 3번째부터 몇 가지를 확인해야 합니다.

Preprocessing부분에서는 사진의 사이즈를 수정하는데 이 부분에서 우리는 416X416의 데이터들이 필요하므로 416X416으로 되어있는지 확인해야 합니다.

Augmentation에서는 기존의 사진을 변화시켜서 데이터를 늘리는 방법을 정하는 부분인데, 이 부분은 **Rotation:** Between -15° and +15°, **Shear:** ±15° Horizontal, ±15° Vertical, **Exposure:** Between -5% and +5%로 설정해주시면 되겠습니다.(위의 3가지 말고도 사진의 형태를 바꾸면서 데이터를 늘리는 기준들이 있습니다. 상황에 따라서 사용하시데, 너무 많이 적용할 시 학습 결과가 더 안 좋아질수도 있습니다) 마지막으로 Generate는 (이미지 3X)를 골라서 이미지의 양을 3배로 늘리는 것으로 최종 생성 버튼을 눌러주시면 되겠습니다.

\*3배 외에도 이미지의 양을 더 늘릴 수 있지만, 이 부분부터는 사이트의 유료서비스를 사용하거나 해야하며, 이미지를 무작정 늘리는게 도움이 되는지에 대해서 확인이 필요할 수도 있습니다\*

최종적으로 새롭게 생성된 데이터가 보이실 겁니다. 이제 여기서 데이터를 추출한 뒤 Colab에서 사용해야 하기에 Export버튼을 눌러봅시다.

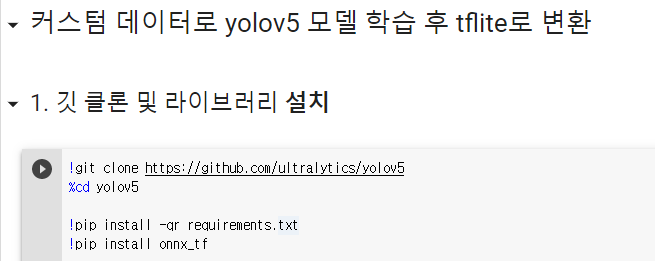
|  |  |
| --- | --- |

<Export 버튼을 누른 뒤 나오는 화면>

Export 버튼을 누르면 위의 창이 뜹니다. 우리 프로젝트에서 사용되는 것은 YOLOv5이므로 형식을 YOLO v5 PyTorch로 고른 뒤, 현재 사용하고 있는 Colab프로그램에서는 데이터를 다운로드 코드로 받아오기 때문에 show download code를 고른 뒤 continue를 눌러 보겠습니다. 이후 당신의 다운로드 코드라는 창이 뜨는데 이 창에서 Raw URL을 눌러서 뜨게 되는 다운로드 주소가 이후 Colab에서 사용하게 될 주소입니다. 이 주소를 복사해서 따로 저장해 놓으셔도 되고, 다운로드는 다시 export클릭하셔도 얻으 실 수 있기에 저장하지 않아도 문제는 없습니다.

1. COLAB을 이용한 데이터 변환

이제 데이터도 준비되었으니 구글 Colab으로 접속하겠습니다. 데이터 학습과 파일 변환의 코드는 나중에 받으시게 될 자료에서 ‘모델학습.ipynb’를 Colab에 업로드해서 여시면 되겠습니다.



<git의 학습용 ipynb를 colab에서 실행하면 보게 될 화면>

파일 실행시 위의 코드가 보일 것입니다. 실행자체는 순차적으로 진행하면 완료되지만, 중간중간 수정해야할 부분이 있으니 과정을 따라서 설명하겠습니다. 위의 코드는 학습을 위해서 YOLOv5의 깃허브에서 복사를 해온뒤 필요한 라이브러리를 설치하는 과정입니다.

|  |
| --- |

<roboflow에서 만들었던 커스텀 데이터를 다운 받는 부분>

라이브러리의 설치가 완료됐다면 다음으로 진행할 것은 2번 과정에서 만들어졌던 데이터를 가져오는 것입니다. 이때 데이터는 다운로드 주소를 통해서 받아오는데 주소는 2번 끝에서 복사해놨던 주소로 수정하여 실행하시면 되겠습니다.

|  |
| --- |
|  |

<모델 셋팅하는 부분>

다음은 모델을 셋팅하는 코드입니다. 이 부분은 학습모델의 셋팅을 진행하는 부분이고, 수정사항은 없기에 그대로 진행하시면 되겠습니다.

|  |
| --- |

<데이터를 학습하는 부분>

제일 중요한 부분입니다. 이제 가져온 커스텀 데이터와 모델을 이용하여 학습을 진행하는 부분이면 데이터가 크기에 따라서 걸리는 시간도 길어지는 부분입니다. 코드에서 변경할 부분은 –epochs 100으로 여기서 숫자는 학습을 몇번 할 것인가를 뜻합니다. 이부분은 보통 200~300정도가 적정선으로 여겨지며, 300을 돌리다가 일찍 종료되는 경우가 종종 있는데, 이는 학습이 너무 잘되서 멈추는 경우입니다.

\*학습에서 주의할 점은 걸리는 시간이 꽤 길기 때문에 주기적(20~30분 간격)으로 Colab의 화면을 클릭해 주어야지 사이트 연결이 끊기지 않습니다. 연결이 끊길경우 처음부터 다시 해야하기에 주의하셔야합니다.\*

|  |
| --- |
|  |

<학습결과 확인 화면>

다음은 학습결과를 확인하는 부분입니다. 첫번째는 텐서보드를 불러서 확인하는 경우, 두번째는 결과 이미지로 그래프를 불러오는 것인데, 텐서보드가 안될 경우에만 두번째 코드를 실행하시면 되겠습니다. 이때 결과는 매트릭스는 1에 수렴하는 직선에 수렴하고, 손실 그래프는 0에 수렴하게 되는 상태라며 훈련이 잘된 것으로 이해하시면 되겠습니다.

|  |
| --- |
|  |

<학습한 데이터와 실제 이미지를 인식해보는 부분>

위는 생성된 학습데이터를 통해서 기존의 이미지들을 인식테스트를 해보는 코드와 그 결과를 이미지로 보여주는 부분입니다. 이 부분은 아까 위의 그래프 값이 잘 나왔다면 인식이미지도 잘 나오셨을 겁니다.

|  |
| --- |

<pt 파일을 tflite 파일로 변환>

학습이 잘 된 것이 확인이 되었다면, 마지막으로 해야 할 것이 있습니다. 안드로이드 스튜디오에서 사용하기 위해선 학습으로 만들어진 best.pt파일을 tflite형식의 파일로 변환해서 넣어줘야하기 때문에 마지막으로 위의 변환코드를 실행하시면 됩니다. 실행 후 tflite파일의 위치를 알려드리겠습니다. 왼쪽의 파일집 모양을 누릅니다.

|  |  |
| --- | --- |

<데이터의 위치>

yolov5/runs/train/yolov5s\_results/weights의 경로를 따라가면 바로 best-fp16.tflite라는 파일을 발견하실 수 있습니다. 이것을 다운로드 하시면 안드로이드 스튜디오 프로젝트에서 사용할 변환파일을 가지고 있는 것입니다.

4. 안드로이드 스튜디오에 넣기 및 수정해야하는 부분

이제 3번까지의 과정을 통해서 안드로이드 스튜디오에 적용할 tflite파일을 가지고 있게 되었습니다. 이제 이를 어디에 넣어야 하고 또 무엇을 수정해야하는지에 대해서 알려드리겠습니다.

|  |
| --- |

안드로이드 스튜디오에서 assets 부분을 클릭하시면 위와같은 파일들이 보일 것입니다. 이 중에서 우리는 아까 받았었던 best-fp16.tflite 파일을 위의 동일한 명의 파일에 덮어 씌워줘야합니다.

|  |
| --- |

덮어씌운 다음에 해야할 것은 customclasses.txt 파일의 내용 수정입니다. 이부분은 인식 부분에서 인식시 그 대상에 대한 구분하게되는 부분으로 내용의 수정은 2번 과정에서 라벨링 작업간에 등록했던 클래스를 그 순서대로 1인 위부터 아래로 하나씩 기입하시면 수정이 끝나게됩니다.

\*중요\*

안드로이드 스튜디오에 데이터를 넣는 부분은 DB연결 방식에 따라서 바꾸셔야 할 수도 있습니다.