|  |  |  |  |
| --- | --- | --- | --- |
| **CNN 분류기 개발 팀프로젝트**  **주제 : NVIDIA AGX Xavier Developer Kit 이용한 CNN 분류기 개발 팀프로젝트**  **팀 : 2조**  **제출자 : 전자정보공학부 3학년 박준혁**  **AI융합학부 3학년 김혜빈**  **AI융합학부 3학년 이찬도** |  |  |  |



◦ **Team Introduction**

CNN 분류기 개발 팀프로젝트 ‘2조’는 박준혁, 김혜빈, 이찬도 이상 세명의 팀원으로 이루져있습니다. 저희는 이번 팀 프로젝트에서 주어진 CIFAR 10 데이터셋을 활용하여, 개별 이미지를 해당 레이블로 분류할 수 있는 CNN(Convolutional Nerueal Network) 구조를 설계, 개발하였습니다. 본 목표를 위해 아래와 같이 역할을 분담하였으며, 목표 달성을 위해 꾸준히 회의를 진행하였습니다.

**[ 역할 분담 ]**

- 팀장 박준혁 : 데이터 불러오기 코드 작성 / 데이터셋 통합 코드 작성 / 데이터 전처리 코드 작

성 / CNN 구조 설계 / 평가보고서 작성 / 발표 자료 작성 / 발표 및 시연

- 팀원 김혜빈 : 데이터 전처리 코드 작성 (데이터 증강, Shuffle) / CNN 구조 설계 /

CNN 구조 파라미터 조정 / 모델 학습 및 검정 / 과대적합 문제 해결

- 팀원 이찬도 : NVIDIA AGX Xavier Developer Kit 활용 방법 조사 및 예제 실행 /

CheckPointAndSaveModel 방법 조사 / Helper 코드 활용 방법 조사

**[ 회의 진행 ]**

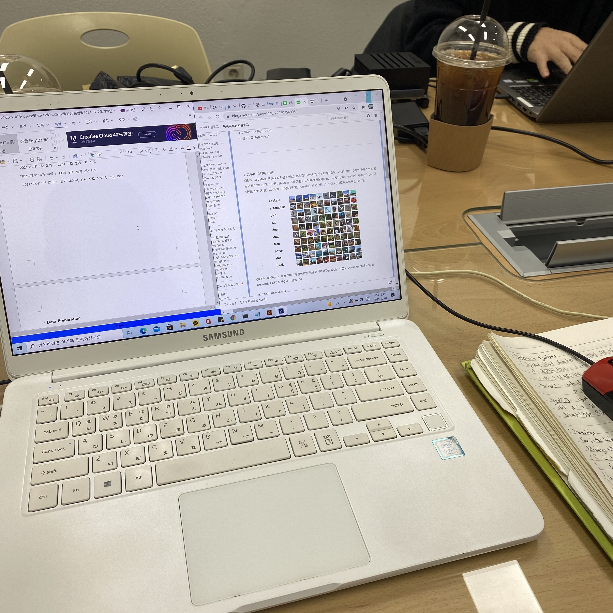
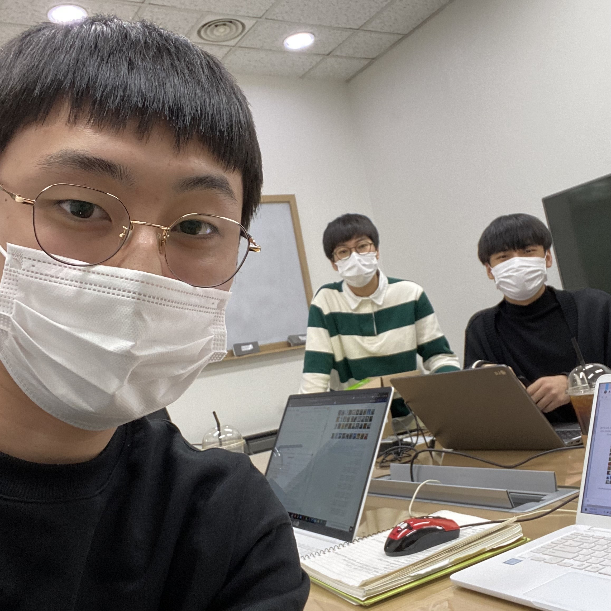
“2021.11.22” 1차 회의 : 전체 프로젝트 계획 수립 및 To Do List 작성

“2021.11.25” 2차 회의 : 데이터 불러오기, 데이터셋 통합 코드 작성 및 점검

“2021.11.28” 3차 회의 : CNN 구조 설계 1차 시도

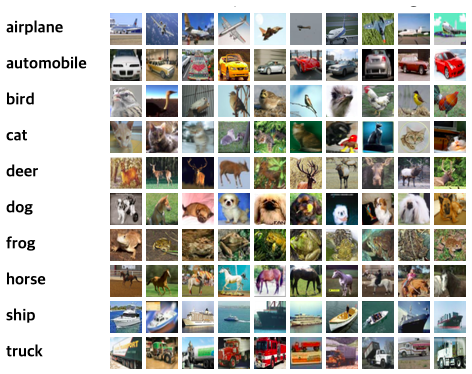
“2021.12.01” 4차 회의 : CNN 구조 설계 2차 시도

“2021.12.05” 5차 회의 : 평가 보고서, 발표 자료 작성 및 시연 준비

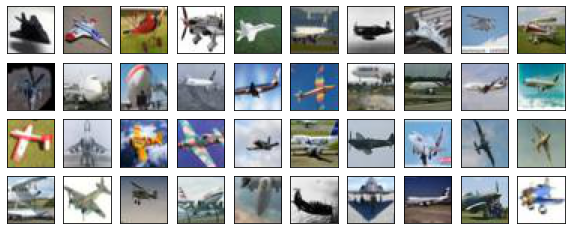


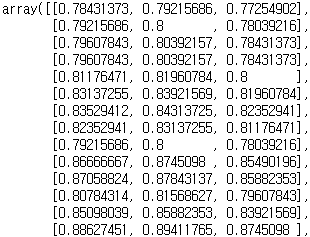
◦ **Data Preparation**

CIFAR-10 dataset은 32x32픽셀의 60,000개 컬러이미지가 포함되어있으며, 각 이미지는 10개의 클래스로 라벨링이 되어있습니다. 10개의 클래스는 아래 그림과 같이 비행기(airplane), 자동차(automobile), 새(bird), 고양이(cat), 개(deer), 개구리(frog), 말(horse), 배(ship), 트럭(truck)으로 각각 0 ~ 9까지의 숫자로 대응되어 있습니다.



본 학습에 이용한 원 데이터는 총 60,000개의 데이터가 카테고리별 6,000개의 데이터로 분할된 형태로 저장되었으며, 각 6,000개의 데이터는 앞에 5,000개까지는 훈련 데이터, 뒤에 1,000개는 시험 데이터로 분류되어 있습니다. 이렇게 해서 총 60,000개의 이미지 데이터를 통해 CNN 분류기를 설계 해보았습니다.

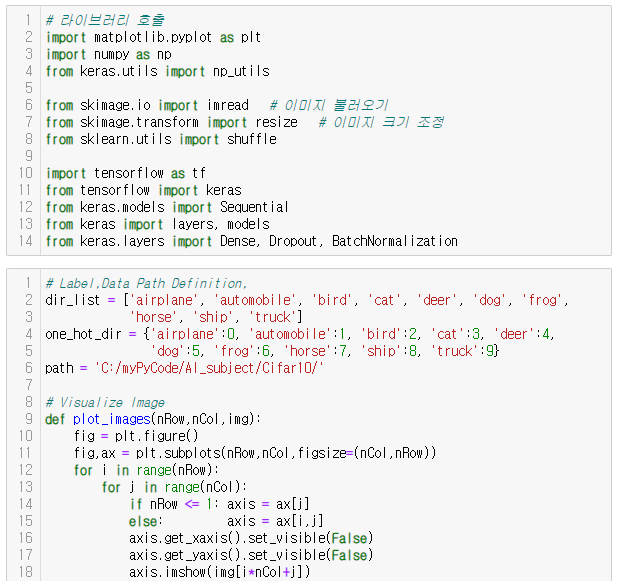
아래 그림은 CIFAR 10 데이터 중 무작위로 40개의 이미지를 추출하여 출력한 결과와 함께 한 이미지의 픽셀 값을 나타낸 것이다. CIFAR 10 데이터셋 내 모든 인스턴스들은 (32, 32, 3)의 구조를 가지고 있는데, 이는 가로 세로 픽셀 값이 모두 ‘32’이며, 3개의 RGB 채널을 가지고 있는 것을 의미한다. 또한, 각 픽셀값은 0 ~ 1 사이의 값을 가지고 있기 때문에 정규화 처리가 기존에 되어 있는 상태이다. 따라서 입력 단계에서 추가적인 정규화 처리는 필요하지 않다.



◦ **Data Load and Preprocessing**

이제 본격적으로 CIFAR 10 데이터를 불러오고, CNN 분류기에 걸맞게 전처리 단계를 수행해야 한다. 데이터 전처리 과정은 데이터를 CNN 분류기에 적합한 형태로 변환하고, 분류 성능을 높이는 것을 목표로 진행하며, 전처리 외에도 전반적인 학습 과정과 결과를 보다 효과적으로 전달하기 위해서 여러 시각화 함수 또한 함께 정의한다.

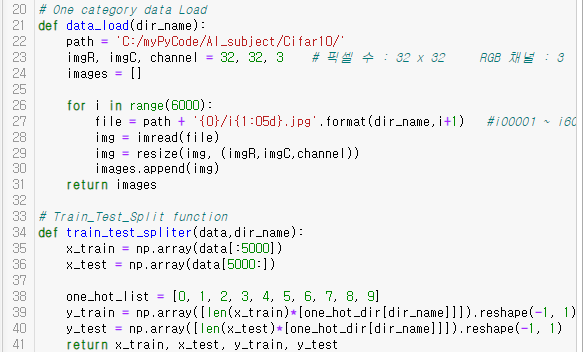
**- Label Object, Data Path Definition. / “plot\_images()” function Definition.**



위 코드에서는 CNN 구조 설계와 시각화 등에 필요한 라이브러리를 호출하였다. 데이터 시각화에 활용되는 ‘matplotlib’, 이미지 데이터를 핸들링할 때 사용되는 ‘skimage’, CNN 구조 설계에 필요한 ‘tensorflow’, ‘keras’ 등의 라이브러리를 호출하였으며, 카테고리 별 레이블 값과 이름을 담은 리스트를 정의하였다. 그리고 데이터를 불러들일 로컬 파일의 경로를 지정하였다.

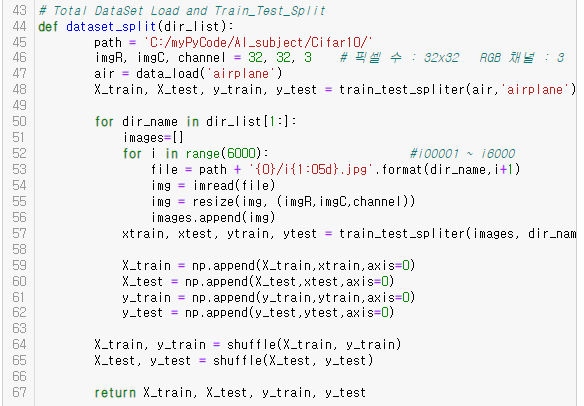
“plot\_images()” 함수에 출력할 이미지의 행과 열 그리고 이미지 객체를 입력하면, 해당 이미지를 설정한 행과 열의 수에 맞게 출력해준다.

**- ‘Data\_Load’ and ‘train\_test\_spliter’ function Definition.**



위 코드에서는 카테고리 별 데이터셋 하나를 불러와 객체에 저장하는 코드를 정의하였다. 데이터 저장 형식에 따라 이미지를 불러온 후, CIFAR 10 데이터의 형식에 맞게 (32, 32, 3)으로 이미지 사이즈를 조정해주었다. 그리고 그 다음 정의된 “train\_test\_spliter()’라는 함수는 이렇게 불러온 카테고리 별 이미지를 훈련 데이터와 시험 데이터로 분할하는 역할을 수행한다. 앞에 5,000개는 훈련 데이터로, 나머지 1,000개는 시험 데이터로 할당하며, 그에 맞는 레이블 객체를 생성해준다.

**- ‘DataSet\_split()’ Function Definition.**



◦ **CNN Structure Design**

블럭 다이어그램 / 파라미터 수 / 실행 시간 / 활성화 함수 / 최적화 함수 / 평가 지표 / 각종 파라미터 /

◦ **Model Training**

Epoch 당 반복 횟수, 소요 시간, loss, 정확도 변화 추이 / loss curve / accuracy curve /

◦ **Evaluate**

100개 샘플에 대한 predict 결과 / plot / 정확도 / 모델 판단 / 과대적합 여부 판단