빈페이지**목차**

1. **Introduction**
2. 주제명 / 주제에 대한 개요
3. What is CNN?
4. Source Code
5. **Modeling**
6. Data Augmentation
7. How to make CNN model
8. **Classification**
9. Softmax / One-hot encoding
10. **Auto Coloring**
11. 영역별 색이 칠해지는 조건
12. **Analysis & Conclusion**
13. Analyze hypherparameters
14. Hypherparameter 와 loss value 의 상관관계
15. Layer 재구성을 통한 Model의 재정의
16. 아쉬운 점
17. **Reference**
18. **Introduction**
19. **프로젝트 주제**

CNN을 이용하여 단색(검정색) 선으로 구성된 과일 이미지를 정해진 label에 따라 classification을 진행하고 분류한 label에 따라 과일을 자동으로 채색한다.

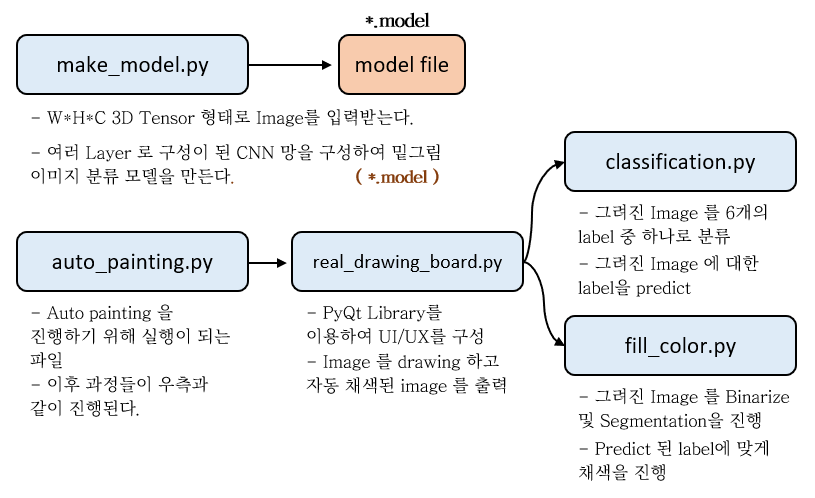
1. **What is CNN?**

CNN(Convolutional Neural Network, 합성곱 신경망)은 시신경 구조를 모방해 만든 기술이다. 기존의 Fully Conneccted Layer로 구성된 Neural Network는 Data를 Vector로 처리하여 Image data의 경우 인근 Pixel과의 관계나 Pattern등 공간 정보를 고려하기 어렵다는 단점이 있었다.

그러나, CNN은 Data를 Vector화가 아닌 3D Tensor로 처리하여 공간 정보를 유지하고 Filter를 이용해 Feature map을 생성하고 학습할 수 있다. 이 때문에 Computer Vision 분야에서 우수한 성능을 보인다.

본 프로젝트는 이러한 CNN의 특성을 이용하여 직접 그린 과일 Image를 분류하고, 추가적으로 Image의 각 영역을 구분하여 자동으로 특정 색으로 채색을 진행하는 프로그램을 구현한다.

1. **Source Code**



**[make\_model.py]**

make\_model.py는 image\_rotate(), make\_npy\_file(), make\_model() 3개의 method 순서로 진행이 된다.

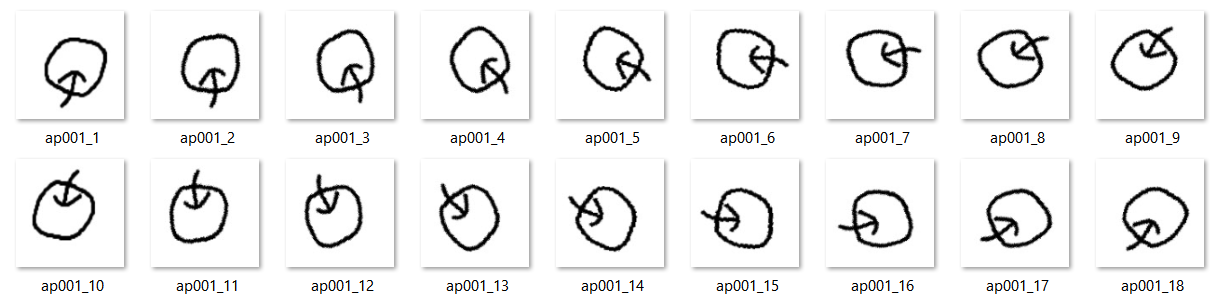
* **image\_rotate() method**

여러 width, height 를 가진 Image를 그리고 이 Image 들을 Train data 로 입력 받는다.

[Train data]

위 Data들은 “apple” label 을 가진 Train data의 일부분이다. 6개의 label 에 대해서 각각 100개의 Train data를 확보하였다. 그 후 Data Augmentation을 진행하였다.

Data Augmentation은 overfitting을 방지하기 위한 Regularization의 일환으로 Data를 다양하게 변형하여 Data의 개수를 증강시키는 기법이다. 구체적인 방법으로는 Horizontal flip, Random crop & Scale, Color jitter, Translation, Rotation, Stretching, Shearing 등 다양한 종류가 있다. 이 중 Rotation 기법을 활용하였다. 총 600개의 직접 그린 Train data가 존재하며, 이 Data 들은 20도를 기준으로 rotation되는 Data Augmentation 이 진행된다.

[rotated\_image]

이 Image 들은  Image 가 -180도부터 160도까지 20도를 단위로 rotation 이 진행된 Image 들이다. 이렇게 Rotation이 진행된 Image 들은 기존의 Train data가 들어있던 폴더와 다른 폴더에 저장이 된다.

* **make\_npy\_file()**

전처리(Resizing, Data augmentation)가 완료된 Image data들의 label을 생성하고, Train data set 과 Validation data set 으로 나눈다.

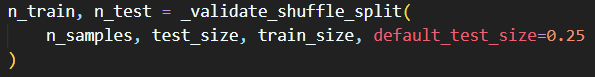
Label은 One-hot vector의 형태로 해당되는 label의 element값을 1, 나머지 element값은 0으로 정의한다. 이후 Image data를 불러와 적용된 label과 짝을 이루도록 배열에 저장한다.



label들을 정의한 위 category 순서대로 각 Image data들에 label이 부여가 된다. 그 후 . Train data set과 Validation data set은 3:1 비율로 랜덤하게 나뉜다.

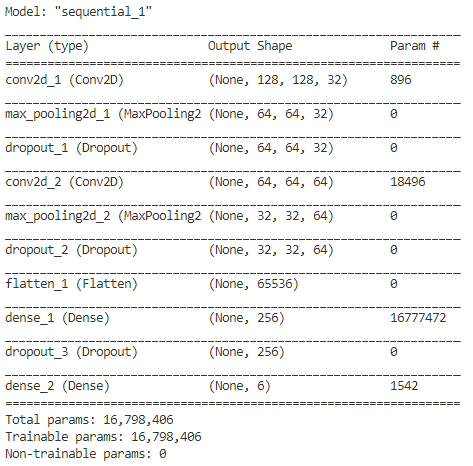


위 부분에서 random하게 train data set, validation data set으로 나뉜다. train\_test\_split(…) method가 사용되며 train\_test\_split(…) method를 확인해 본 결과 아래와 같이 default\_test\_size=0.25로 부여가 된다.



* make\_model()

참고한 Open source 는 아래와 같이 5-Layer 모델을 사용한다.



1. **Modeling**
2. **Data Augmentation**

우선 [‘apple’, ‘tomato’, ‘orientalmelon’, ‘watermelon’, ‘carrot’, ‘strawberry’] 6개의 각각의 label 에 대해서 직접 그린 100개의 data를 통해 train\_data를 확보하였다. 이 Image data들을 모두 하나의 (W\*H\*Channel) 형태를 가지도록 통일한 후 그 후 Image rotation 으로 모든 Image들을 20도씩 rotation 하면서 저장하여 Data Augmentation을 진행 하였다.

1. d
2. d
3. d
4. d
5. d
6. d
7. d

ddd