

ex01. 데이터 수집 (Web Crawling)

- keywords.txt 파일에서 키워드를 읽고 구글 이미지에 키워드를 자동 검색 후 이미지 저장 (대략 각 키워드 100 장)
- keyword.txt (banana, kiwi, orange, pineapple, watermelon 5 가지)

```
2022.12 > 12.19_d54_data > ≡ keyword.txt
```

```
1 banana
2 kiwi
3 orange
4 pineapple
5 watermelon
```

ex02. Data Split

- ex01 에서 저장한 이미지 데이터를 읽고 train 과 validation 데이터를 9:1 비율로 분리
- 분리 후 train 폴더와 val 폴더에 이미지 저장

ex03. Image Pretreatment

- expand2square() 함수를 사용해 기존 이미지를 정사각형화 후, padding하여 400 x 400 사이즈로 Resize -> dataset/images 폴더에 저장
- 각 카테고리 별 데이터의 수를 출력 (총 460장)

```
Banana      : 129
Kiwi        : 58
Orange      : 81
Pineapple   : 111
Watermelon  : 81
Total       : 460
```

ex04. CustomDataset 생성

- data/train 에 있는 5 개 카테고리의 모든 이미지 데이터를 불러와 리스트 한 후, 이미지에 해당하는 label 값을 읽어와 image, label 데이터 반환

ex05. Main

- 서로 다른 augmentation transform 모델을 불러오고 훈련이 진행
- hy_parameter.py 에서 parameter 값을 가져옴
- transform_list.py 에서 image augmentation 리스트를 불러와 훈련 모두 실행
- utils.py의 train() 함수를 사용하여 모델 훈련 후 model save 폴더에 pt 파일 저장

hy_parameter.py

```
1 lr = 0.001
2 batch_size = 8
3 epoch = 100
4 num_classes = 5
```

hy_parameter.py

```
transform_train0 = A.Compose([
    A.Resize(height= 256, width= 256),
    ToTensorV2()])

transform_train1 = [
    A.Resize(height=256, width=256),
    A.RandomCrop(height= 224, width= 224), # 랜덤 crop 224 x 224
    A.HorizontalFlip(p= 0.5),             # 좌우
    ToTensorV2()]

transform_train2 = [
    A.Resize(height=256, width=256),
    A.RandomCrop(height= 224, width= 224), # 랜덤 crop 224 x 224
    A.OneOf([
        A.HorizontalFlip(p = 1),          # 좌우 반전
        A.VerticalFlip(p = 1),            # 상하 반전
        A.RandomRotate90(p = 1),          # 회전
    ], p= 1),
    ToTensorV2()]

transform_train3 = [
    A.Resize(height=256, width=256),
    A.RandomCrop(height= 224, width= 224), # 랜덤 crop 224 x 224
    A.OneOf([
        A.MotionBlur(p=1),                # 모션 블러 필터
        A.OpticalDistortion(p=1),          # 광학 왜곡
        A.GaussNoise(p=1),                 # 가우시안 블러 필터
    ], p= 1),
    ToTensorV2()]
```

```

transform_train4 = [
    A.Resize(height=256, width=256),
    A.RandomCrop(height= 224, width= 224), # 랜덤 crop 224 x 224
    A.OneOf([
        A.HorizontalFlip(p = 1),      # 좌우 반전
        A.VerticalFlip(p = 1),        # 상하 반전
        A.RandomRotate90(p = 1),      # 회전
    ], p= 1),
    A.OneOf([
        A.MotionBlur(p=1),            # 모션 블러 필터
        A.OpticalDistortion(p=1),      # 광학 왜곡
        A.GaussNoise(p=1),            # 가우시안 블러 필터
    ], p= 1),
    ToTensorV2()]

transform_val = [
    A.Resize(height=256, width=256),
    ToTensorV2()]

```

ex06. Test

- model_save 폴더의 서로 다른 augmentation 을 적용해 훈련시켜 저장한 pt 파일들을 읽고 정확도 출력
- pt 모델들을 비교하여 정확도가 가장 높은 pt 파일을 best.pt 로 저장

```

===== The Result of transform0 =====
acc for 49 image : 91.84%
transform0이 None보다 정확도가 높습니다

===== The Result of transform1 =====
acc for 49 image : 95.92%
transform10이 transform0보다 정확도가 높습니다

===== The Result of transform2 =====
acc for 49 image : 93.88%
transform1의 정확도가 더 높습니다

===== The Result of transform3 =====
acc for 49 image : 95.92%
transform30이 transform1보다 정확도가 같습니다

===== The Result of transform4 =====
acc for 49 image : 93.88%
transform1의 정확도가 더 높습니다

```

Result

		Transform 0	Transform 1	Transform 2		Transform 3		Transform 4	
적용한 Albu 종류	1	Resize(256,256)	Resize(256,256)	Resize(256,256)		Resize(256,256)		Resize(256,256)	
	2		RandomCrop(224, 224)	RandomCrop(224, 224)		RandomCrop(224, 224)		RandomCrop(224, 224)	
	3		HorizontalFlip(p=0.5)	OneOf(p=1)	HorizontalFlip(p=1)	OneOf(p=1)	MotionBlur(p=1)	OneOf(p=1)	HorizontalFlip(p=1)
					VerticalFlip(p=1)		OpticalDistortion(p=1)		VerticalFlip(p=1)
					RandomRotate90(p=1)		GaussNoise(p=1)		RandomRotate90(p=1)
	4							OneOf(p=1)	MotionBlur(p=1)
							OpticalDistortion(p=1)		OpticalDistortion(p=1)
							GaussNoise(p=1)		GaussNoise(p=1)
Accuracy		91.84%	95.92%	93.88%		95.92%		93.88%	