Team02. 권태윤, 손병구, 여익수, 이승윤, 최유연

Rock-Scissor-Paper Image Classification

1. 데이터 구축 방법

1) Laurence Moroney - The AI Guy. : Train 과 Valid 로 활용

https://laurencemoroney.com/datasets.html



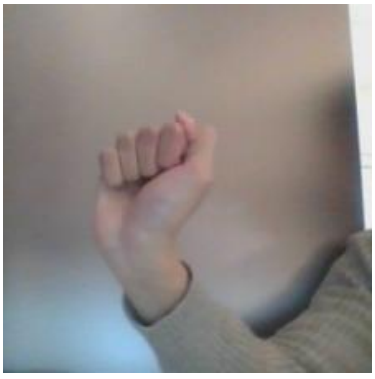
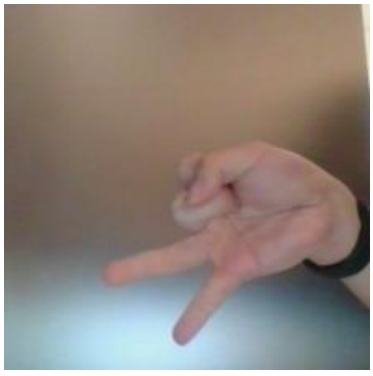
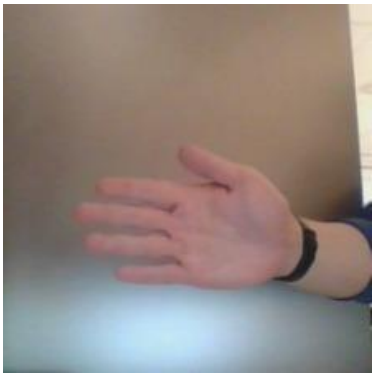
2) Kaggle Data: Train 과 Valid 로 활용

https://www.kaggle.com/datasets/drgfreeman/rockpaperscissors?resource=download



3) Webcam Data: Webcam 으로 직접 촬영하여 Test 로 활용

https://teachablemachine.withgoogle.com/

2. 데이터 구축 현황

- 총 5698 장, Train, Valid, Test 를 대략 8 : 1 : 1 로 split 실행

Train (4571)			Valid (509)			Test (618)		
Rock	Scissor	Paper	Rock	Scissor	Paper	Rock	Scissor	Paper
1521	1542	1508	169	172	168	197	210	211

3. 모델 선정

Swin_t, ResNet50, ReXNetV1, VGG11 로 대표적인 Image Classification 4 가지 모델 선정

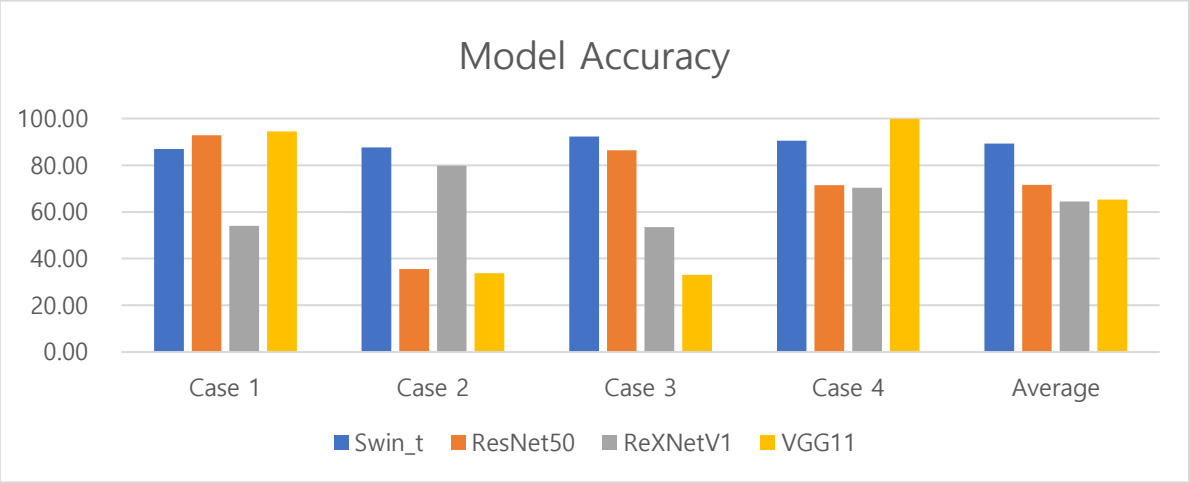
4. Augmentation 및 하이퍼파라미터 설정

```
train_aug = A.Compose([
    A.SmallestMaxSize(max_size= 224),
    A.Resize(width= 200, height= 200),
    A.RandomCrop(width= 180, height= 180),
    A.HorizontalFlip(p=0.6),
    A.VerticalFlip(p=0.6),
    A.ShiftScaleRotate(shift_limit= 0.05, scale_limit= 0.06,
                       rotate_limit=20, p=0.5),
    A.RGBShift(r_shift_limit=10, g_shift_limit=10, b_shift_limit=10, p=1),
    A.RandomBrightnessContrast(p= 0.5),
    A.Normalize(mean=(0.485, 0.456, 0.406), std= (0.229, 0.224, 0.225)),
    ToTensorV2()
])

valid_aug = A.Compose([
    A.SmallestMaxSize(max_size= 224),
    A.Resize(width= 200, height= 200),
    A.CenterCrop(width= 180, height= 180),
    A.Normalize(mean=(0.485, 0.456, 0.406), std= (0.229, 0.224, 0.225)),
    ToTensorV2()
])
```

No.	Case 1	Case 2	Case3	Case 4
Train Batch	64	64	64	**128**
Valid Batch	64	64	64	128
Epoch	**50**	10	10	10
Learning Rate	0.0001	**0.001**	0.0001	0.0001
Loss Function	LabelSmoothingCrossEntropy			
Optimizer	AdamW			

6. 결과 화면



No.	Swin_t	ResNet50	ReXNetV1	VGG11	Average	비교 지표
Case 1	86.89	92.88	54.05	**94.50**	82.08	epoch: 50
Case 2	**87.70**	35.60	79.77	33.79	59.22	lr: 0.001
Case 3	**92.23**	86.41	53.56	33.01	66.30	평균
Case 4	90.45	71.52	70.39	**99.80**	**83.04**	batch: 124
Average	**89.32**	71.60	64.44	65.28	72.66	

1) Learning Rate 비교분석 (Case 2 vs 다른 Case)

- Case 2 에서 Learning Rate 를 0.001 로 설정하였을 때 대부분의 모델의 정확도가 상대적으로 낮은 것을 확인하여, 다른 Case 에서 Learning Rate 를 낮춰 0.0001 로 설정한 경우 더 높은 정확도를 기록.

2) Epoch 비교분석 (Case 3 vs 다른 Case)

- Case 1 에서 Epoch 을 50 으로 설정하였을 때 Epoch 을 10 으로 설정한 다른 Case 에 비해 더 높은 정확도를 기록.

3) Batch Size 비교분석 (Case 4 vs 다른 Case)

- Case 4 에서 Batch Size 를 124 로 설정하였을 때 Batch Size 을 64 로 설정한 다른 Case 에 비해 비교적 높은 정확도를 기록.

4) 서로 다른 4 가지의 Classification 모델 비교분석

- Swin_t 가 모든 Case 에서 큰 폭 없이 높은 정확도를 보이며 성능이 가장 좋은 분류 모델로 판단.

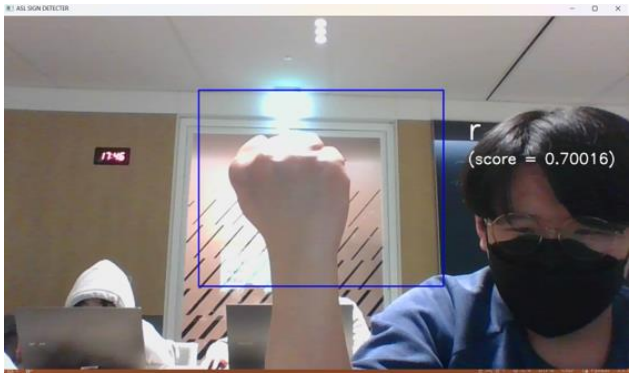
- VGG11 의 경우 Case 에 따라 가장 큰 정확도 차이를 보이며 hyperparameter 의 가장 큰 영향을 받는 것으로 판단.

- ResNet50 은 Case 2 를 제외하고 모든 Case 에서 고른 정확도를 보이고 ReXNetV1 의 경우 모든 Case 에서 좋지 않은 정확도를 기록.

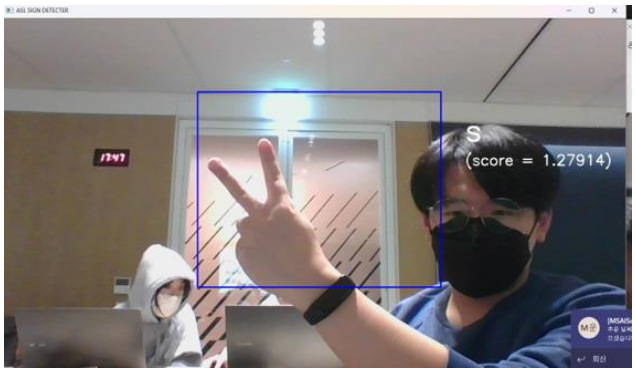
- 전체적인 정확도 부분에 있어 Train 과 Valid 의 모든 사진은 손등(손의 뒷면)이 사용되었지만 실시간 webcam 으로 촬영하여 구축한 test 의 사진의 경우 손의 앞면이 포함 되어있어 모델이 성능이 조금 미흡한 것으로 판단.

7. Real-time Webcam Object Detection & Classification

- Rock



- Scissor



- Paper