

Mini Project Week 03

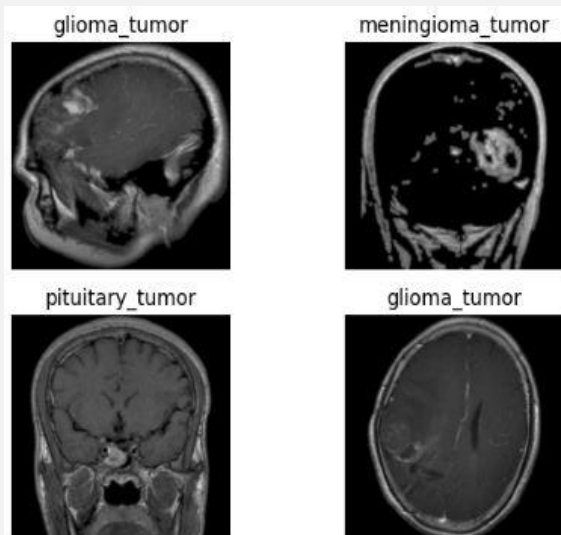
Basic 2조: 김예림, 한유진

```
lookup.KeyValue  
f.constant(['em  
=tf.constant([0  
ce = tf.lookup.StaticV  
init,  
num_oov_buckets=5)
```

```
lookup.StaticVocabular  
initializer,  
num_oov_buckets,  
lookup_key_dtype=None  
name=None,  
experimental_is_open
```

Introduction & Goal

Brain Tumor(Glioma/Meningioma/Pituitary)와 Normal Classification



Recall로 모델의 성능 판단

Base Code

: ResNet50 + (Flatten -> Dense)

Data Preprocess - Data Augmentation

제한된 데이터셋을 최대한 활용하기 위한 방안

```
data_augmentation = tf.keras.Sequential([  
    tf.keras.layers.experimental.preprocessing.Rescaling(1./255),  
    tf.keras.layers.experimental.preprocessing.RandomFlip("horizontal"),  
    tf.keras.layers.experimental.preprocessing.RandomRotation(0.2),  
    tf.keras.layers.experimental.preprocessing.RandomContrast(0.2),  
])
```

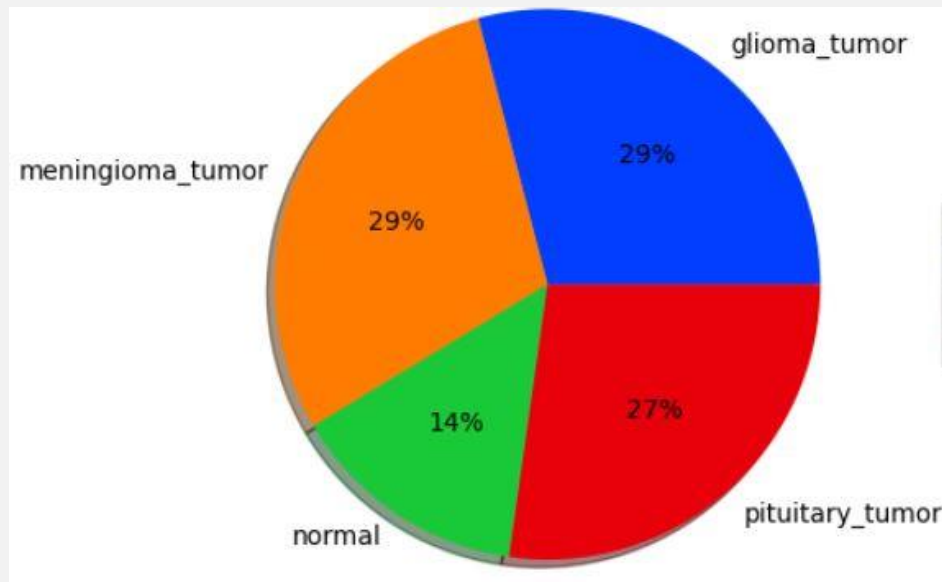
Rescaling: 표준화

Random Flip: 반전

Random Rotation: 회전

Random Contrast: 대비

Data Preprocess - Class Weight



Normal이 전체 데이터셋의 14%로
다른 클래스의 데이터에 비해 적음

Resampling

- Undersampling
- Oversampling

Class Weight

```
3 class_weights = {  
4     0: 1.0, # glioma_tumor  
5     1: 1.0, # meningioma_tumor  
6     2: 1.5, # normal  
7     3: 1.0 # pituitary_tumor  
8 }
```

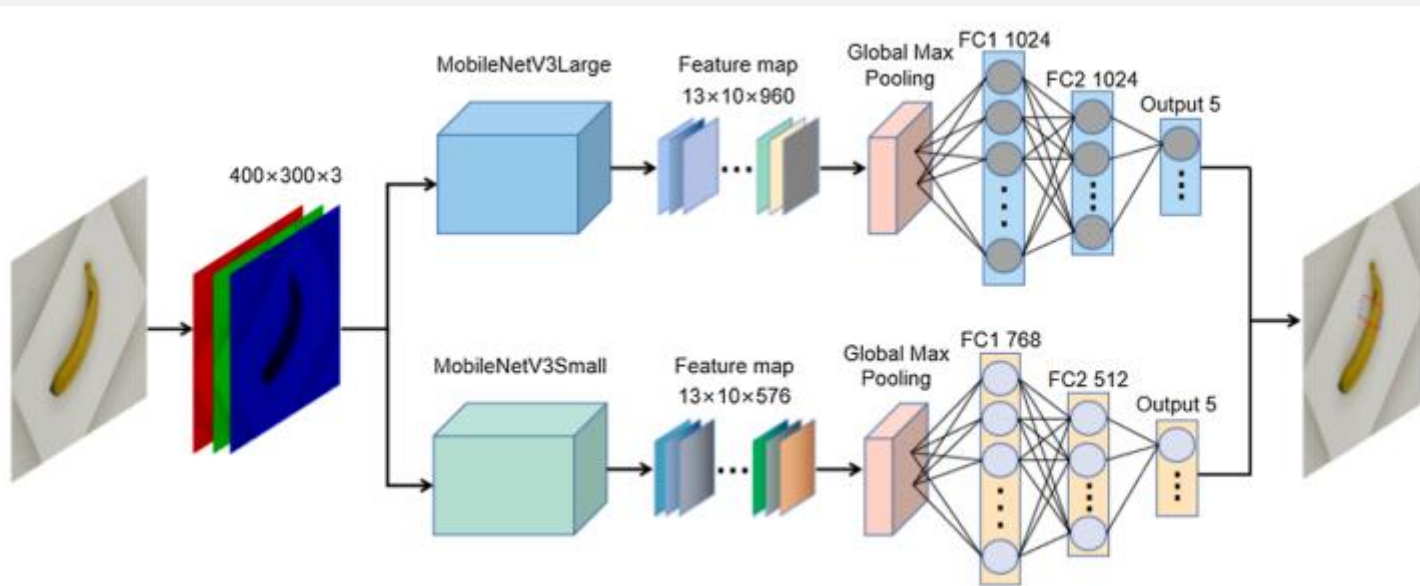
Normal Class: 1.5

Other Class: 1.0

Model

1. ResNet50
2. ResNet101
3. ResNet152
4. VGGNet16
5. VGGNet19
6. MobileNet
7. MobileNetV2
8. MobileNetV3Small
9. MobileNetV3Large

MobileNetV3Large



사용한 기법

1. Class Weight
2. Batch Normalization
3. Callback
4. Bayesian Optimization

Batch Normalization

Batch Normalization

: 각 미니배치의 입력을 평균과 분산으로 정규화

-> 입력 분포 안정화, 층의 출력 범위 조절

```
x=tf.keras.layers.Flatten()(x)
x=tf.keras.layers.Dense(512, activation="relu")(x)
x=tf.keras.layers.BatchNormalization()(x)
x=tf.keras.layers.Dropout(0.5)(x)
```


Callback

Learning Rate Reduction

: Val_loss가 더 이상 감소하지 않을 때, 학습률 감소

Early Stopping

: 성능이 향상되지 않을 때, 중간에 학습 종료

```
learning_rate_reduction = tf.keras.callbacks.ReduceLROnPlateau(  
    monitor="val_loss", patience=2, factor=0.5, min_lr=0.00001, verbose=1)  
early_stopping = tf.keras.callbacks.EarlyStopping(  
    monitor="val_loss", patience=5, restore_best_weights=True)
```

Bayesian Optimization

Bayesian Optimization

: 불필요한 하이퍼 파라미터 반복 탐색을 줄여
빠르게 최적의 하이퍼 파라미터를 찾는 방법

```
space = {  
    'learning_rate': hp.choice('learning_rate', [0.1, 0.01, 0.001, 0.0001]),  
    'dropout_rate': hp.uniform('dropout_rate', 0.0, 0.5),  
    'batch_size': hp.choice('batch_size', [16, 32, 64, 128]),  
    'dense_units': hp.choice('dense_units', [256, 512, 768, 1024]),  
}
```

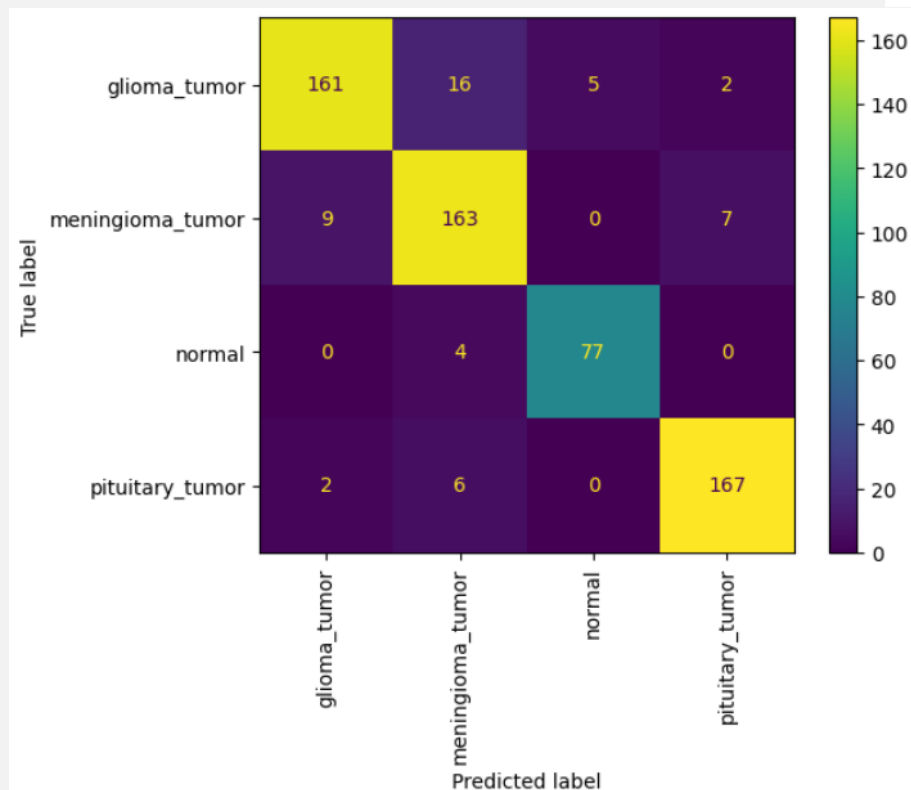
Result

F1 : [0.9216790437201086]

Precision : [0.922629380843999]

Recall : [0.921592100167327]

	precision	recall	f1-score	support
0	0.94	0.88	0.90	184
1	0.86	0.91	0.89	179
2	0.94	0.95	0.94	81
3	0.95	0.95	0.95	175
accuracy			0.92	619
macro avg	0.92	0.92	0.92	619
weighted avg	0.92	0.92	0.92	619



Conclusion

Recall: 0.9215

(Base code의 Recall:0.9052)

- True label이 glioma tumor일 때, 잘못 예측 多
-> Glioma tumor의 특성을 이해한다면 조금 더 발전된 모델을 만들 수 있을 것 같음
- 데이터 전처리로 모델의 성능이 더 많이 향상됨
-> 추가적인 데이터 전처리 기법들에 대해 안다면 더 발전된 모델을 만들 수 있을 것 같음.