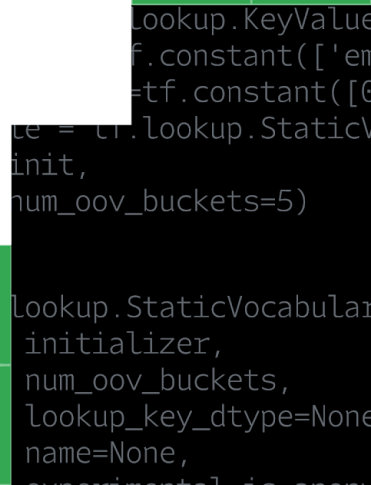
Google Developer Student Clubs

ML/DL Study

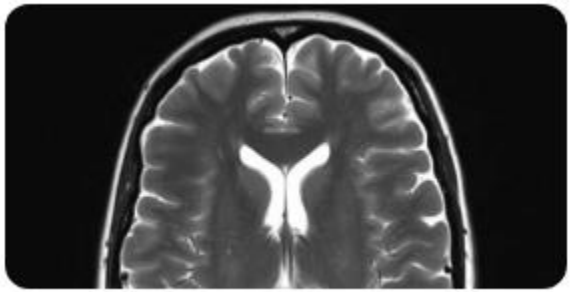


Code Review

Code Review

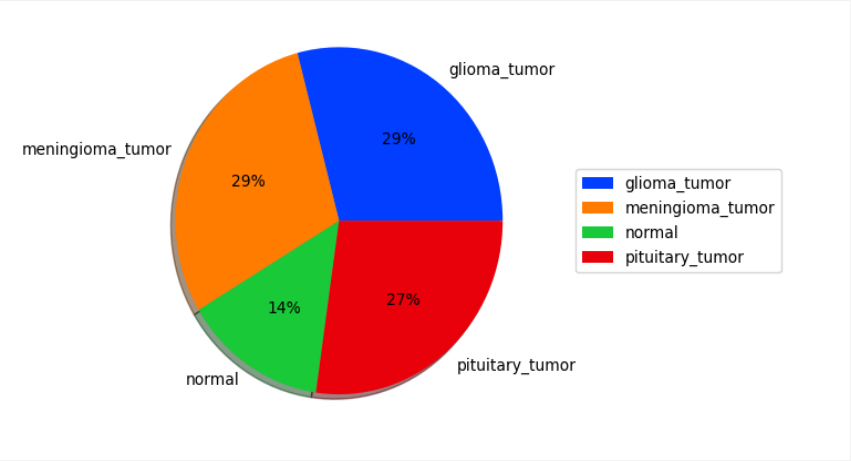
Brain tumors 256x256

A Refined Brain Tumor Image Dataset with Grayscale Normalization and Zoom

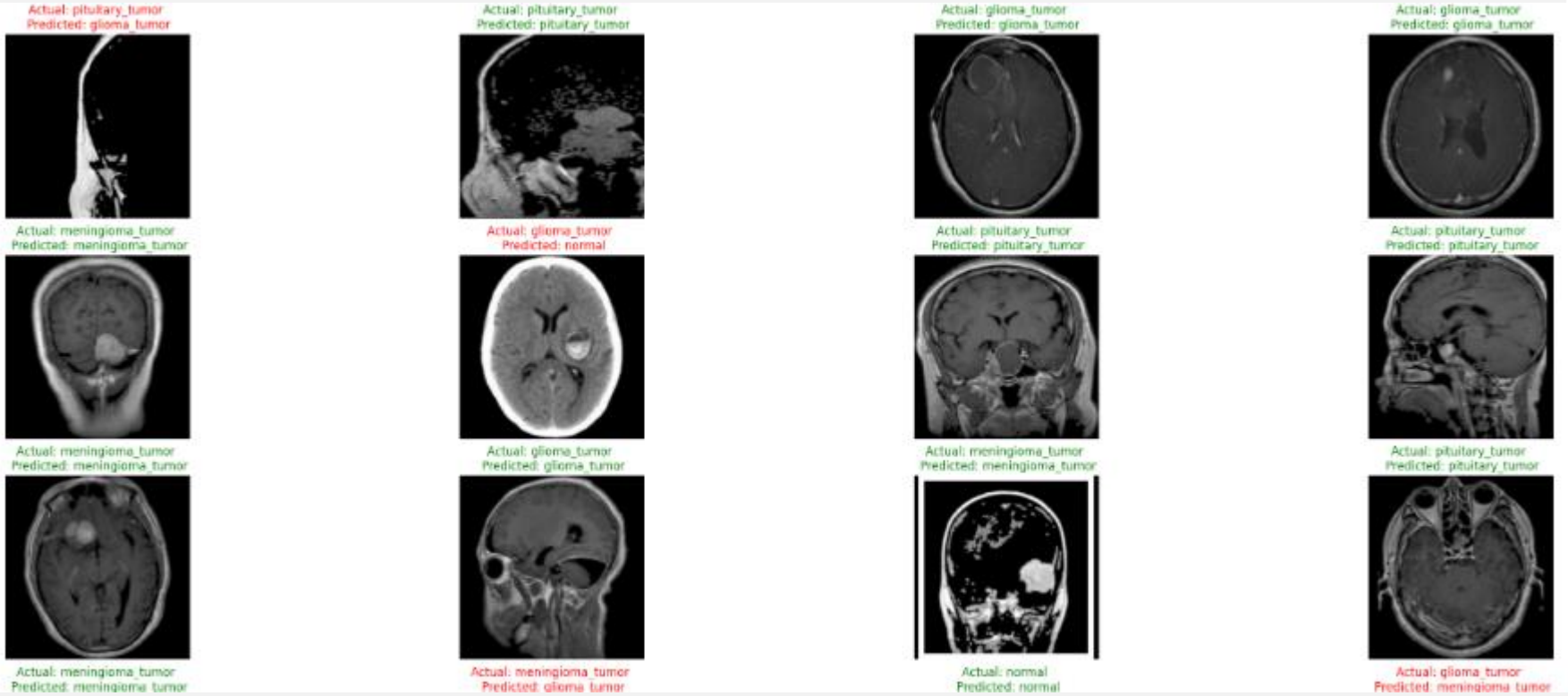


glioma_tumor	2023-11-13 오후 11:56	파일 폴더
meningioma_tumor	2023-11-13 오후 11:57	파일 폴더
normal	2023-11-13 오후 11:58	파일 폴더
pituitary_tumor	2023-11-13 오후 11:59	파일 폴더

Data Preprocesing



	Tumor_Name	glioma_tumor	meningioma_tumor	normal	pituitary_tumor	glioma_tumor_ratio	meningioma_tumor_ratio	normal_ratio	pituitary_tumor_ratio	SUM	Incorrect	Total	Incorrect Ratio
0	glioma_tumor	161	19	4	0	0.875	0.103261	0.021739	0	1	23	184	0.125
1	meningioma_tumor	9	161	0	9	0.050279	0.899441	0	0.050279	1	18	179	0.100559
2	normal	0	1	80	0	0	0.012346	0.987654	0	1	1	81	0.012346
3	pituitary_tumor	2	8	0	165	0.011429	0.045714	0	0.942857	1	10	175	0.057143

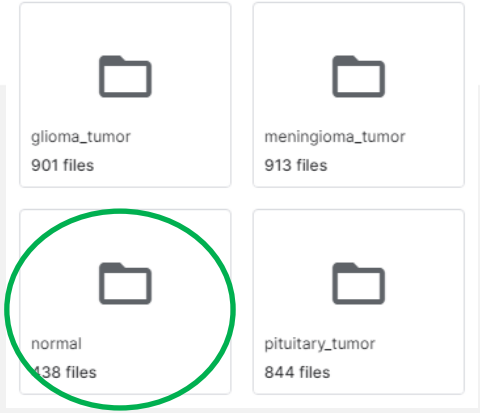

Actual: pituitary_tumor
Predicted: glioma_tumor
Actual: pituitary_tumor
Predicted: pituitary_tumor
Actual: glioma_tumor
Predicted: glioma_tumor
Actual: glioma_tumor
Predicted: glioma_tumor
Actual: meningioma_tumor
Predicted: meningioma_tumor
Actual: glioma_tumor
Predicted: normal
Actual: pituitary_tumor
Predicted: pituitary_tumor
Actual: pituitary_tumor
Predicted: pituitary_tumor
Actual: meningioma_tumor
Predicted: meningioma_tumor
Actual: glioma_tumor
Predicted: glioma_tumor
Actual: meningioma_tumor
Predicted: meningioma_tumor
Actual: pituitary_tumor
Predicted: pituitary_tumor
Actual: meningioma_tumor
Predicted: meningioma_tumor
Actual: meningioma_tumor
Predicted: glioma_tumor
Actual: normal
Predicted: normal
Actual: glioma_tumor
Predicted: meningioma_tumor

Model

```
base_model =tf.keras.applications.resnet.ResNet50(include_top=False)

base_model.trainable = False

inputs = tf.keras.layers.Input(shape=(224, 224,3),name="input_layer")
x = base_model(inputs)

x =
tf.keras.layers.GlobalAveragePooling2D(name="global_average_pooling_layer"
)(x)

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

outputs=tf.keras.layers.Dense(4,
activation="softmax",name="output_layer")(x)

model= tf.keras.Model(inputs, outputs, name="model")
```

```
model.compile(loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              optimizer=tf.keras.optimizers.Adam(lr=0.001),
              metrics=["accuracy"]
             )

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)

history1= model.fit(train_data, epochs=20, validation_data=
val_data,callbacks=[Early_Stopping,learning_rate_reduction], verbose=1)
```

Refactoring



< 방향 설정 >

- 클래스 불균형 해결
- 모델 예측 결과 분석
- ResNet101, 152나 VGGNet, EfficientNet etc 활용
- 코드 재사용성 ↑

  - 데이터 로딩 → 분할하는 처리 과정의 코드 재사용성 높이기
  - model training 파트 코드 재사용성 높이기