Brain Tumor Classification (MRI)

Creating and testing different models for Brain Tumor Classification using Tensorflow library in Python

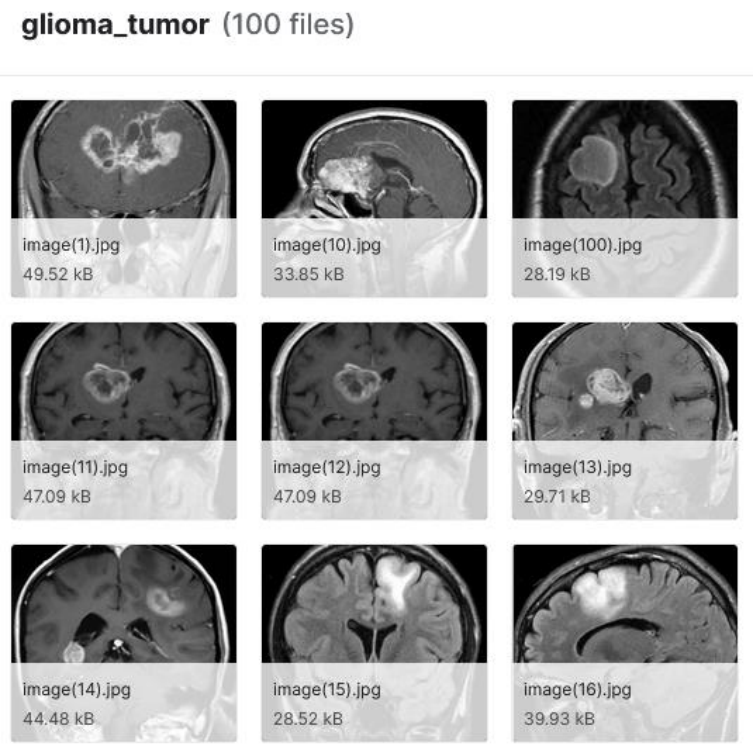
Brain Tumor Classification (MRI)

https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri

3264 files:

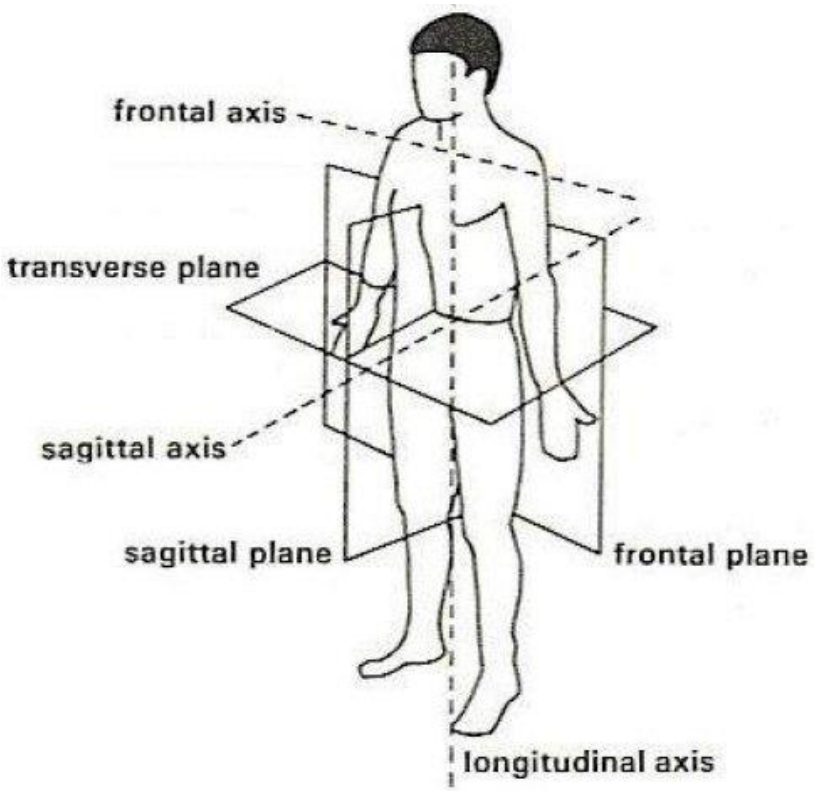
- Training:
  - -no tumor (395)
  - - glioma tumor (826)
  - - meningioma tumor (822)
  - - pituitary tumor (827)
- Testing:
  - -no tumor (105)
  - - glioma tumor (100)
  - - meningioma tumor (115)
  - - pituitary tumor (74)

Problem: Classification



Images are taken from 3 planes:

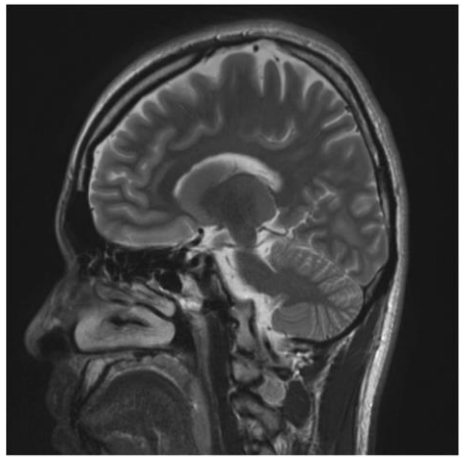
- - transverse
- - saggital
- - frontal



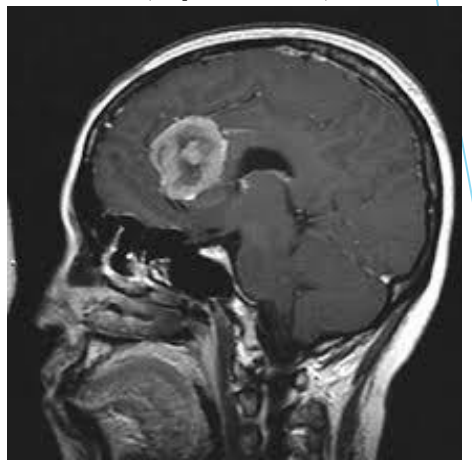
EXAMPLE IMAGES

(saggital plane)

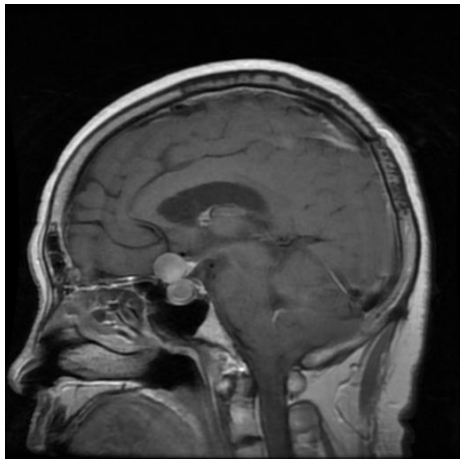
No tumor



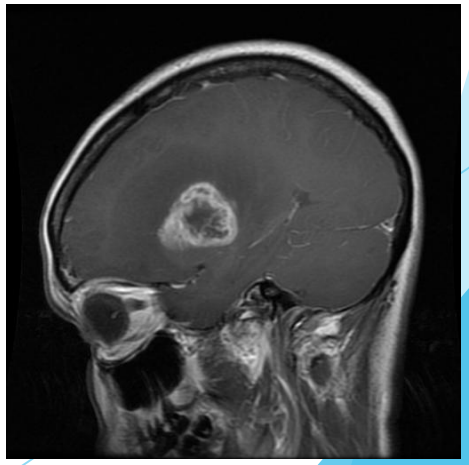
Meningioma tumor
(oponiak)



Pituitary tumor
(guz przysadki)



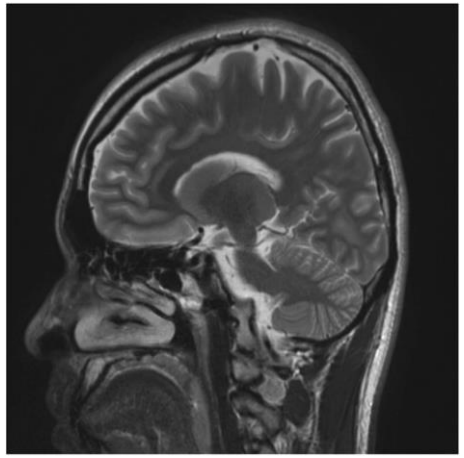
Glioma tumor
(glejak)



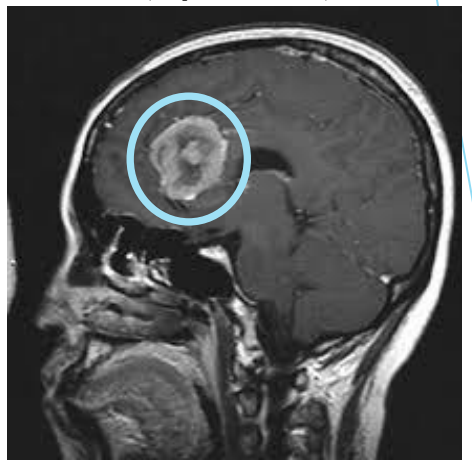
EXAMPLE IMAGES

(saggital plane)

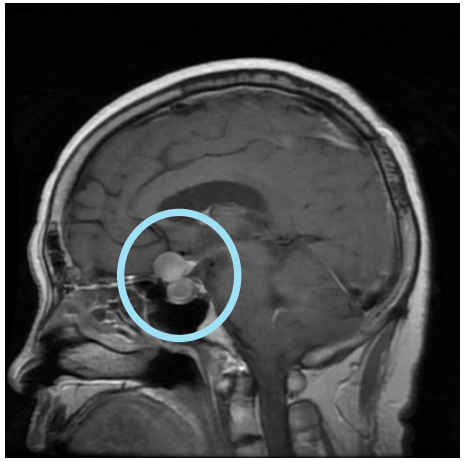
No tumor



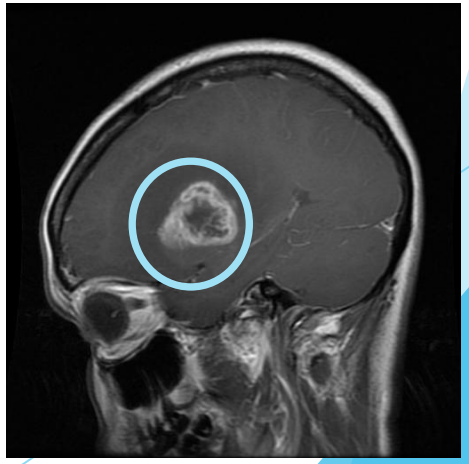
Meningioma tumor
(oponiak)



Pituitary tumor
(guz przysadki)



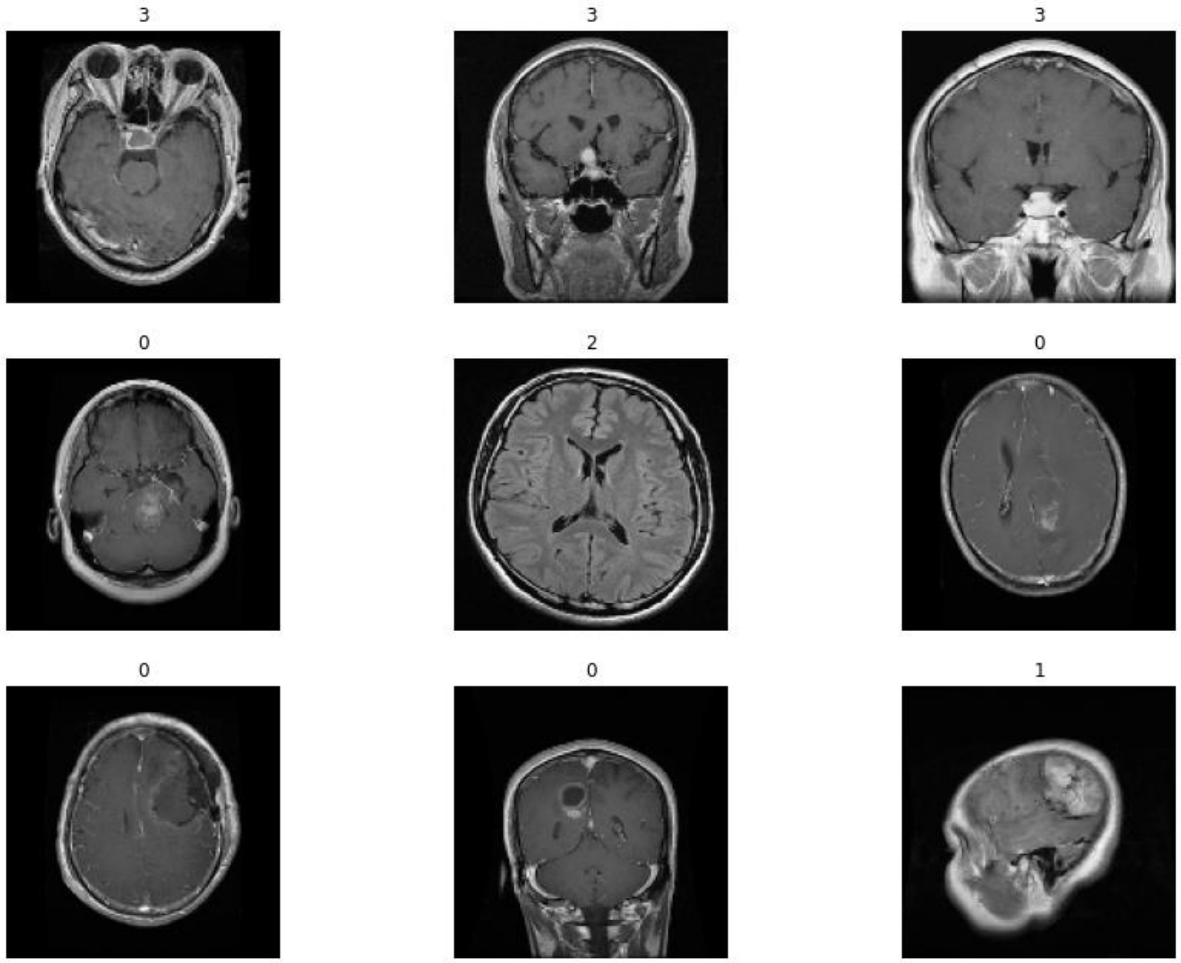
Glioma tumor
(glejak)



Loaded images (Greyscale)

```
mapping = {'no_tumor’:0, 'pituitary_tumor':1, 'meningioma_tumor':2,
'glioma_tumor’:3}
```

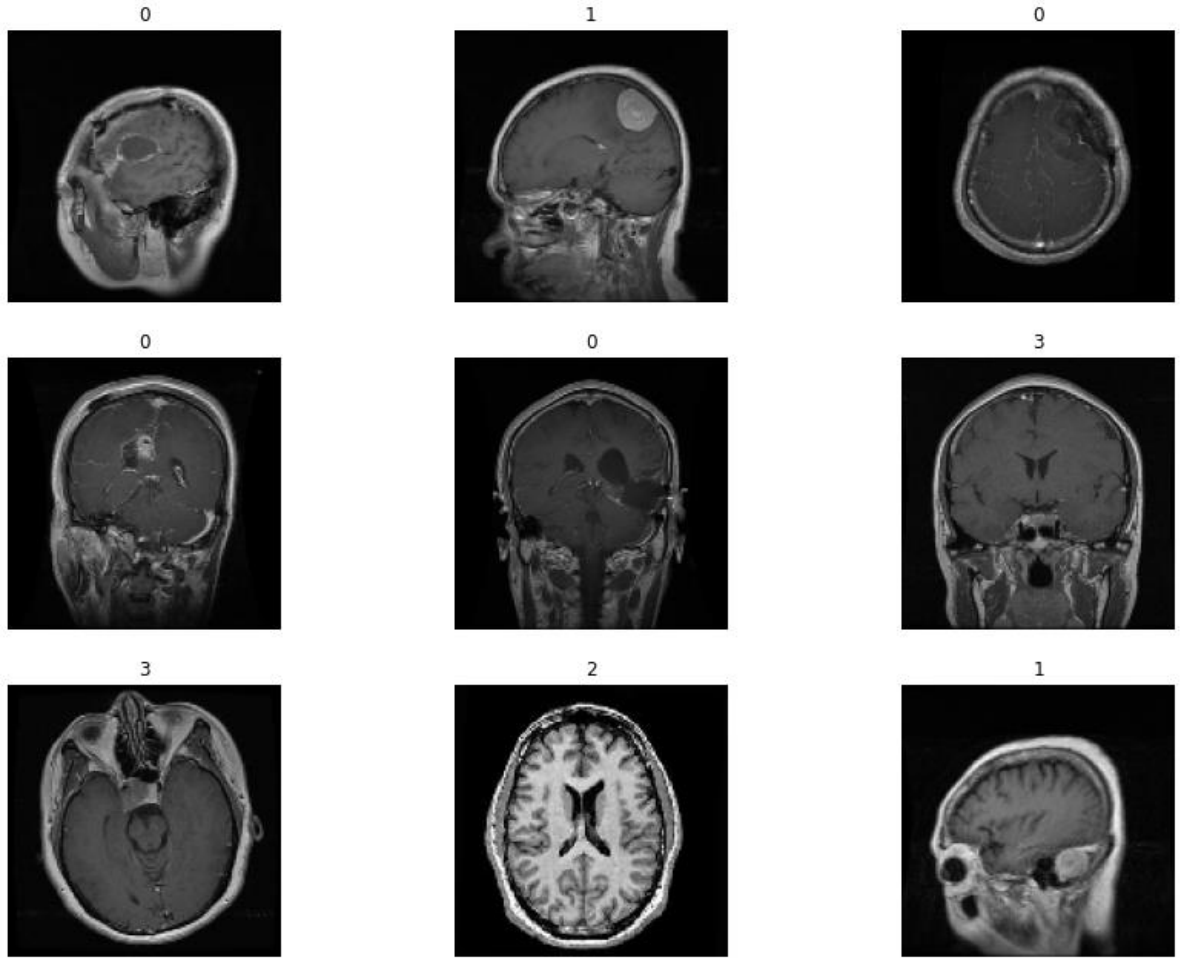
- Resolution:128x128px
- Normalized



Loaded images (RGB)

```
mapping = {'no_tumor’:0, 'pituitary_tumor':1, 'meningioma_tumor':2,
'glioma_tumor’:3}
```

- Resolution:128x128px
- Normalized

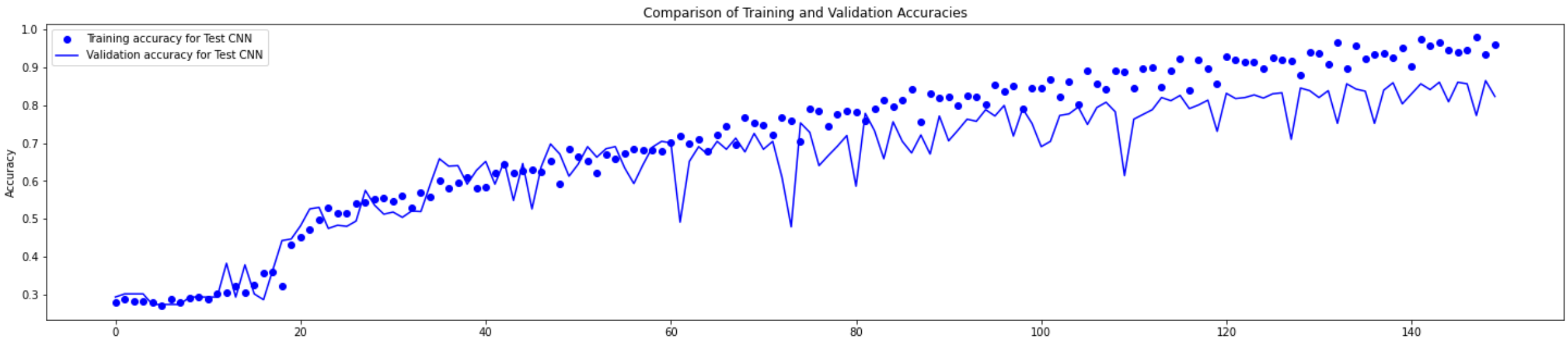


Base network

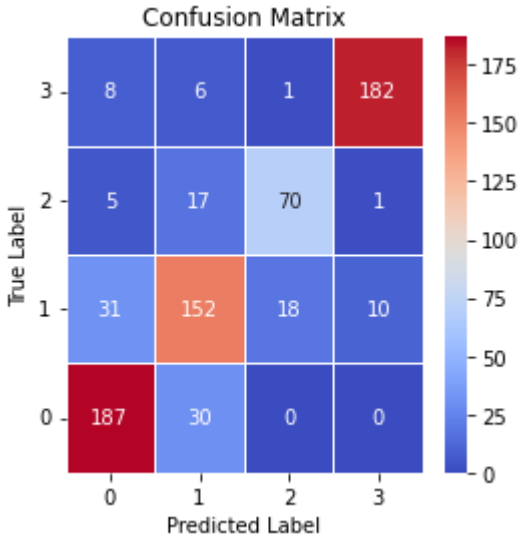
```
model1 = models.Sequential()
model1.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(RES, RES, 1)))
model1.add(layers.MaxPooling2D((2, 2)))
model1.add(layers.Conv2D(64, (3, 3), activation='relu'))
model1.add(layers.MaxPooling2D((2, 2)))
model1.add(layers.Conv2D(64, (3, 3), activation='relu'))
model1.add(layers.Flatten())
model1.add(layers.Dense(64, activation='relu'))
model1.add(layers.Dense(4, activation='softmax'))

model1.summary()
```

```
hist_df = pd.DataFrame(history1.history)

hist_csv_file = 'history_CNN2_V3.csv'
with open(hist_csv_file, mode='w') as f:
    hist_df.to_csv(f)
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 32)      320

max_pooling2d (MaxPooling2D  (None, 63, 63, 32)        0
)

conv2d_1 (Conv2D)            (None, 61, 61, 64)        18496

max_pooling2d_1 (MaxPooling  (None, 30, 30, 64)        0
2D)

conv2d_2 (Conv2D)            (None, 28, 28, 64)        36928

flatten (Flatten)            (None, 50176)             0

dense (Dense)                (None, 64)                3211328

dense_1 (Dense)              (None, 4)                 260

=================================================================
Total params: 3,267,332
Trainable params: 3,267,332
Non-trainable params: 0
_________________________________________________________________
```



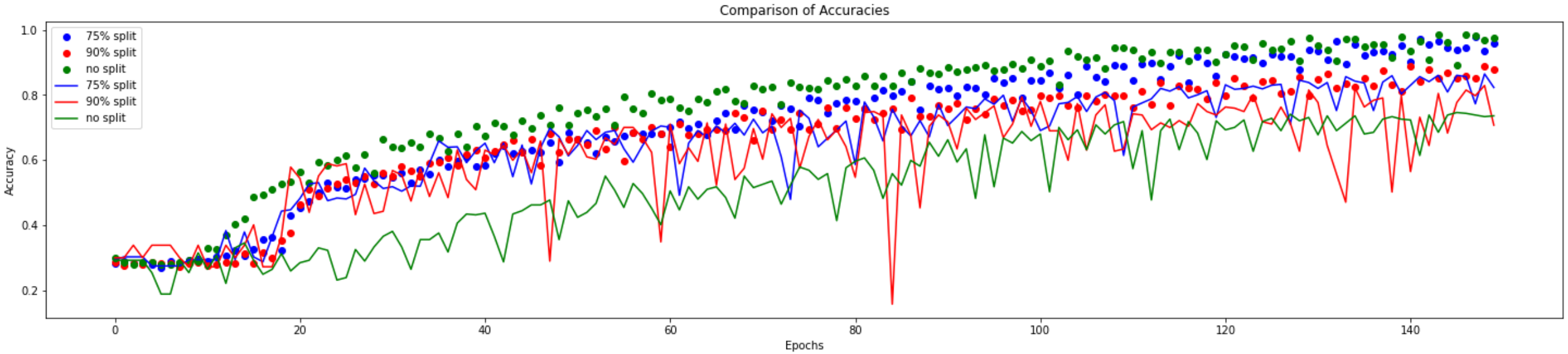
no_tumor:0
pituitary_tumor:1
meningioma_tumor:2
glioma_tumor:3



	precision	recall	f1-score	support
0	0.81	0.86	0.83	217
1	0.74	0.72	0.73	211
2	0.79	0.75	0.77	93
3	0.94	0.92	0.93	197
accuracy			0.82	718
macro avg	0.82	0.81	0.82	718
weighted avg	0.82	0.82	0.82	718

Different size of training dataset

```
x_train_splitRES, x_validation_splitRES, y_train_splitRES, y_validation_splitRES = train_test_split(x_train_cnRES, y_train_cnRES, shuffle=True, random_state=101)
```



Model with dropout

```
probdrop=0.5

model2 = models.Sequential()
model2.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(RES, RES, 1)))
model2.add(layers.MaxPooling2D((2, 2)))
model2.add(layers.Dropout(probdrop))
model2.add(layers.Conv2D(64, (3, 3), activation='relu'))
model2.add(layers.MaxPooling2D((2, 2)))
model2.add(layers.Dropout(probdrop))
model2.add(layers.Conv2D(64, (3, 3), activation='relu'))
model2.add(layers.Flatten())
model2.add(layers.Dense(64, activation='relu'))
model2.add(layers.Dropout(probdrop))
model2.add(layers.Dense(4, activation='softmax'))
```

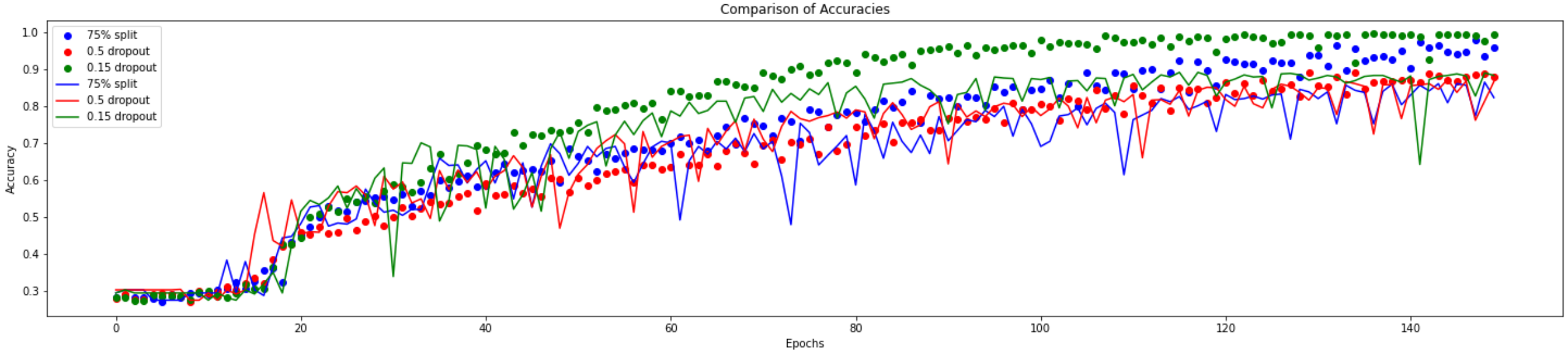
Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 126, 126, 32)	320
max_pooling2d_6 (MaxPooling 2D)	(None, 63, 63, 32)	0
dropout (Dropout)	(None, 63, 63, 32)	0
conv2d_10 (Conv2D)	(None, 61, 61, 64)	18496
max_pooling2d_7 (MaxPooling 2D)	(None, 30, 30, 64)	0
dropout_1 (Dropout)	(None, 30, 30, 64)	0
conv2d_11 (Conv2D)	(None, 28, 28, 64)	36928
flatten_3 (Flatten)	(None, 50176)	0
dense_6 (Dense)	(None, 64)	3211328
dropout_2 (Dropout)	(None, 64)	0

...

Total params: 3,267,332

Trainable params: 3,267,332

Non-trainable params: 0


Comparison of Accuracies
75% split
0.5 dropout
0.15 dropout
75% split
0.5 dropout
0.15 dropout
Accuracy
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
120
140

Different convolution layer size

9,5,3

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 120, 120, 32)	2624
max_pooling2d (MaxPooling2D)	(None, 40, 40, 32)	0
dropout_1 (Dropout)	(None, 40, 40, 32)	0
conv2d_1 (Conv2D)	(None, 36, 36, 64)	51264
max_pooling2d_1 (MaxPooling2D)	(None, 18, 18, 64)	0
dropout_2 (Dropout)	(None, 18, 18, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	36928
flatten (Flatten)	(None, 16384)	0
dense_1 (Dense)	(None, 64)	1048640
dropout_3 (Dropout)	(None, 64)	0

...

Total params: 1,139,716
Trainable params: 1,139,716
Non-trainable params: 0

9,9,9

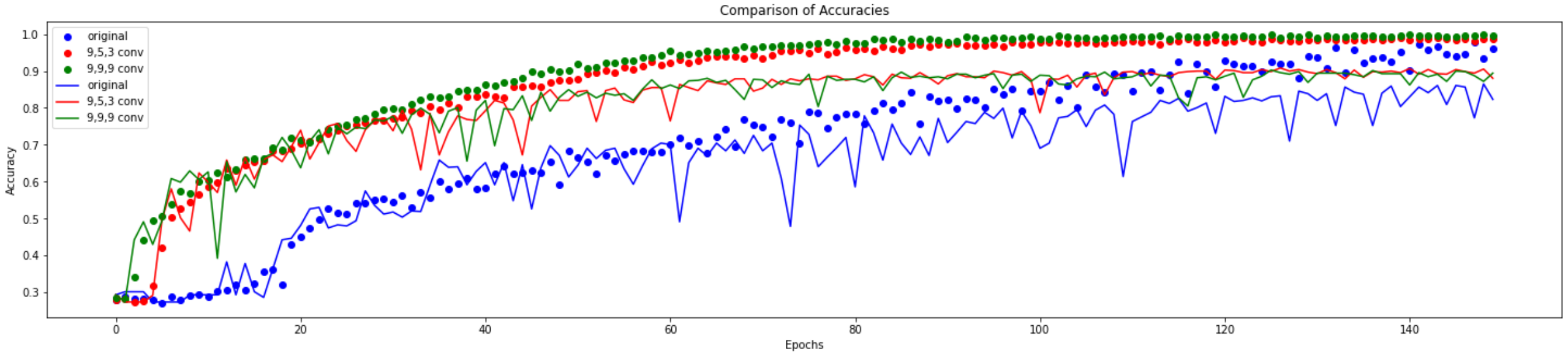
Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 120, 120, 32)	2624
max_pooling2d_8 (MaxPooling2D)	(None, 40, 40, 32)	0
dropout_11 (Dropout)	(None, 40, 40, 32)	0
conv2d_13 (Conv2D)	(None, 32, 32, 64)	165952
max_pooling2d_9 (MaxPooling2D)	(None, 10, 10, 64)	0
dropout_12 (Dropout)	(None, 10, 10, 64)	0
conv2d_14 (Conv2D)	(None, 2, 2, 64)	331840
flatten_2 (Flatten)	(None, 256)	0
dense_5 (Dense)	(None, 64)	16448
dropout_13 (Dropout)	(None, 64)	0

...

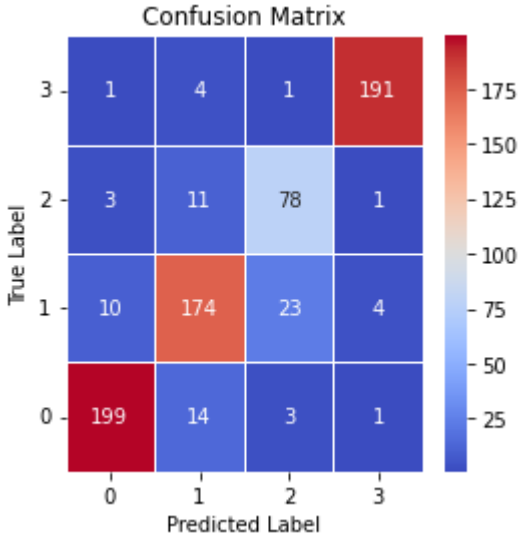
Total params: 517,124
Trainable params: 517,124
Non-trainable params: 0

original

Total params: 3,267,332
Trainable params: 3,267,332
Non-trainable params: 0



no_tumor:0
pituitary_tumor:1
meningioma_tumor:2
glioma_tumor:3

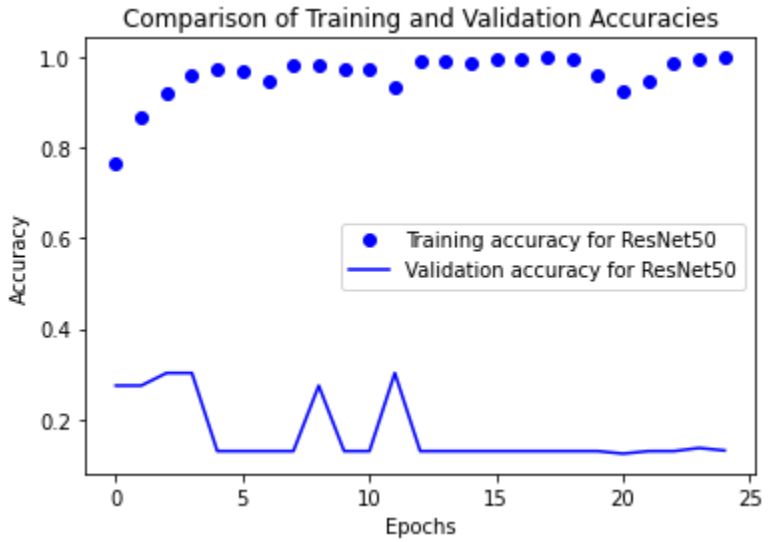


	precision	recall	f1-score	support
0	0.93	0.92	0.93	217
1	0.86	0.82	0.84	211
2	0.74	0.84	0.79	93
3	0.97	0.97	0.97	197
accuracy			0.89	718
macro avg	0.88	0.89	0.88	718
weighted avg	0.90	0.89	0.89	718

ResNet50

```
from tensorflow.keras.applications.resnet50 import ResNet50
resnet = ResNet50(weights='imagenet', include_top=False,input_shape=(RES,RES,3))
```

```
modelRes = resnet.output
modelRes = tf.keras.layers.GlobalAveragePooling2D()(modelRes)
modelRes = tf.keras.layers.Dropout(rate=0.5)(modelRes)
modelRes = tf.keras.layers.Dense(4,activation='softmax')(modelRes)
modelRes = tf.keras.models.Model(inputs=resnet.input, outputs = modelRes)
```




Comparison of Training and Validation Accuracies
Accuracy
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
Training accuracy for ResNet50
Validation accuracy for ResNet50