Selected 2 Project

FCAI-HU

Team Number: 14

Team Members Names & IDs:

Team Member Name	Team Member ID	
محمد عبد الرحيم ابراهيم محمد	201900698	
مصطفى عصام عبدالفتاح ابوشامه	201900824	
احمد مصطفى اسماعيل علام	201900103	
أحمد فرغلى ثابت عبد الرحمن	201900076	
احمد اشرف عبدالمنعم محمود	201900016	
عمر محمد كامل عبدالمالك	201900530	

Paper details

Paper Name: Brain Tumor Classification Using Convolutional Neural Network (CNN).

Paper authors: Sunanda Das, Nishat Nayla Labiba
Paper link:
Brain-Tumor-Classification-Using-Convolutional-Neural-Network.pdf (researchgate.net)

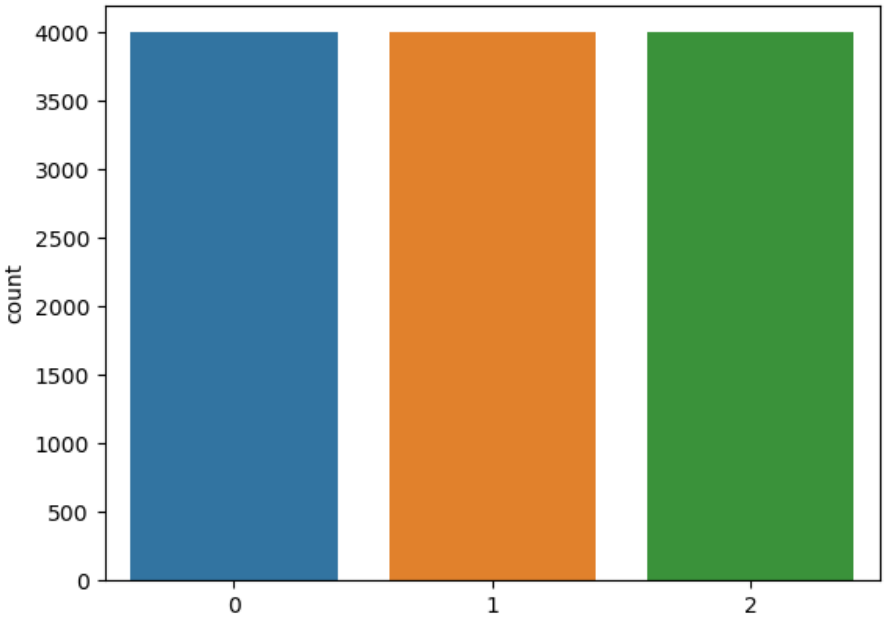
Dataset name: Brain Cancer
Dataset link:
https://www.kaggle.com/datasets/obulisainaren/multi-cancer

Number of samples: 15000
[Train : Test] Ratio -> 12000 image : 3000 image = **80% : 20%**

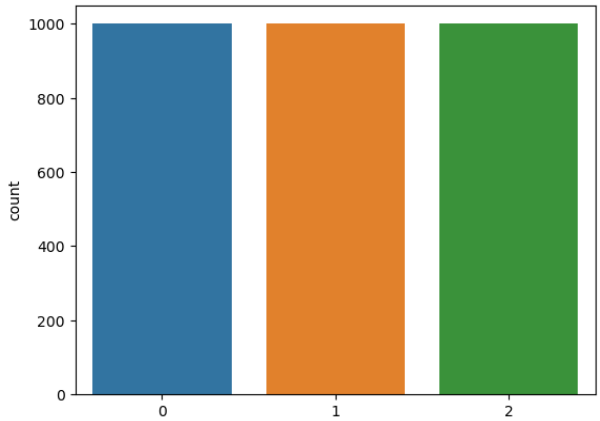
Diminution of image: 112*112
Classes number: 3
Classes Labels: brain glioma, brain menin, brain pituitary

Analysis & Distribution Data

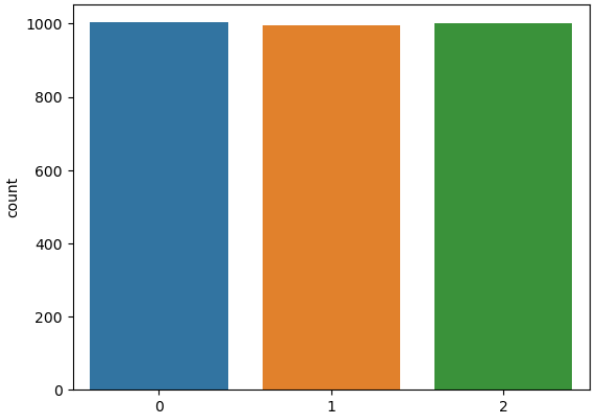
Training Data



Testing Data



Predicting Data



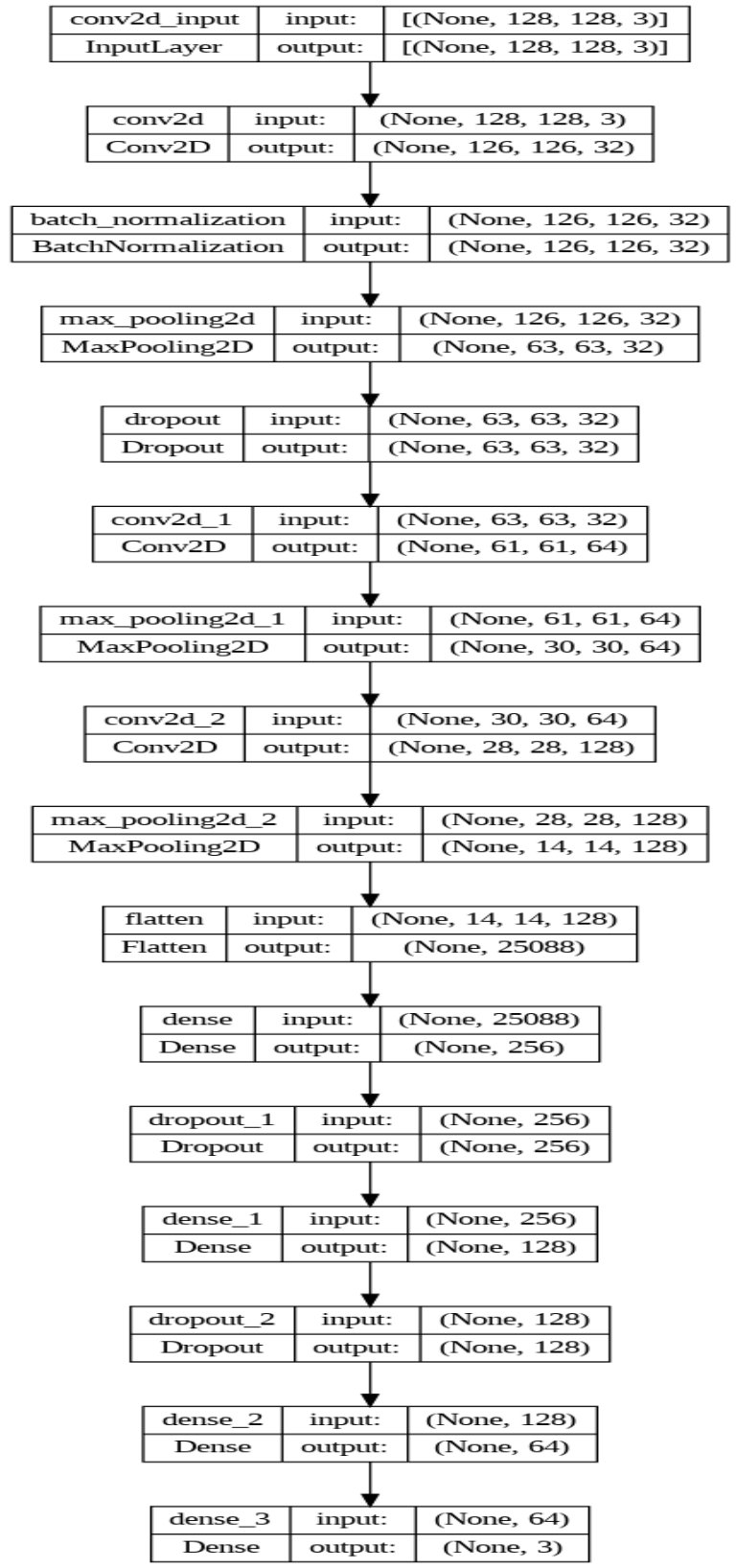
These graphs show that data is balanced

Model

```
model = Sequential()

model.add(Conv2D(filters = 32, kernel_size = (3, 3), activation = 'relu', input_shape = (IMGSIZE, IMGSIZE, 3)))
model.add(BatchNormalization())
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(filters = 64, kernel_size = (3, 3), activation = 'relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(filters = 128, kernel_size = (3, 3), activation = 'relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(filters = 256, kernel_size = (3, 3), activation = 'relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256, activation = 'relu'))
model.add(Dropout(0.25))

model.add(Dense(128, activation = 'relu'))
model.add(Dropout(0.25))

model.add(Dense(64, activation = 'relu'))
model.add(Dense(3, activation = 'softmax'))

model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])
```

```
history = model.fit(X_train, y_train, validation_data = (X_test, y_test), epochs = 25)
# val_loss, val_acc = model.evaluate(X_test, y_test)

y_pred = model.predict(X_test)

model.summary()
```

Block Diagram:



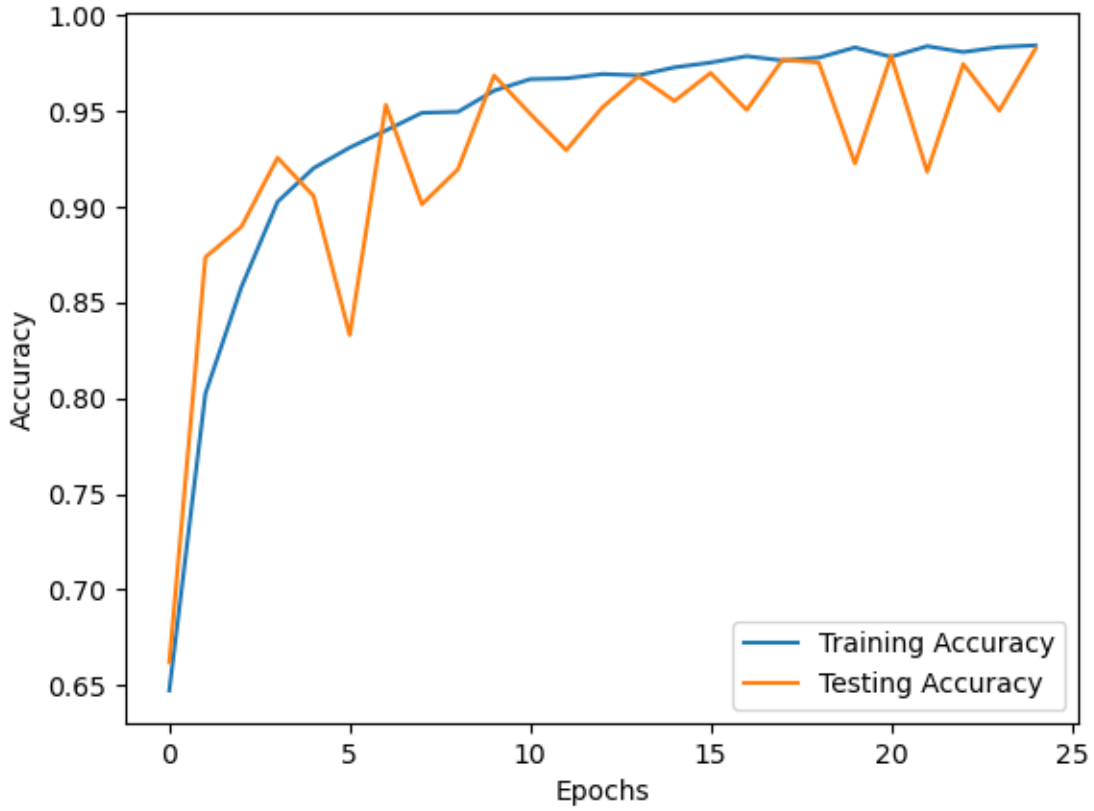
Hyper-parameters:

```
numOfCNNLayers = 4 & Activation_functhion = ‘relu’
kernal_size = (3,3)& MaxPooling = (2,2)
numOfDenseLayers = 4 & Activation_functhion = ‘relu’, ‘softmax’ for output
optimizer = 'adam'
loss = 'sparse_categorical_crossentropy'
epochs = 25
batch_size = 32 (Defualt)
image_size = 128 * 128
channels = 3 (RGB)
```

We have changed the hyperparameters in model many times to get the best results.

Result Details:

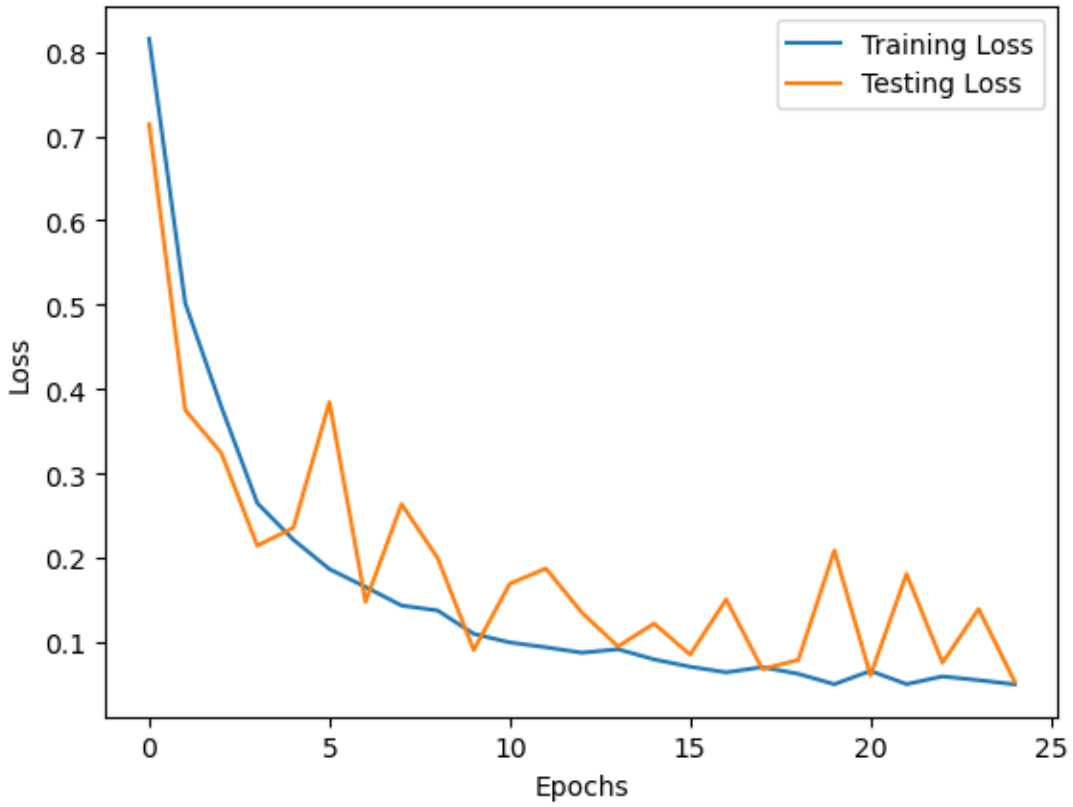
1) Accuracy Graph:



Training Accuracy = 98.44 %

Validation Accuracy = 98.27 %

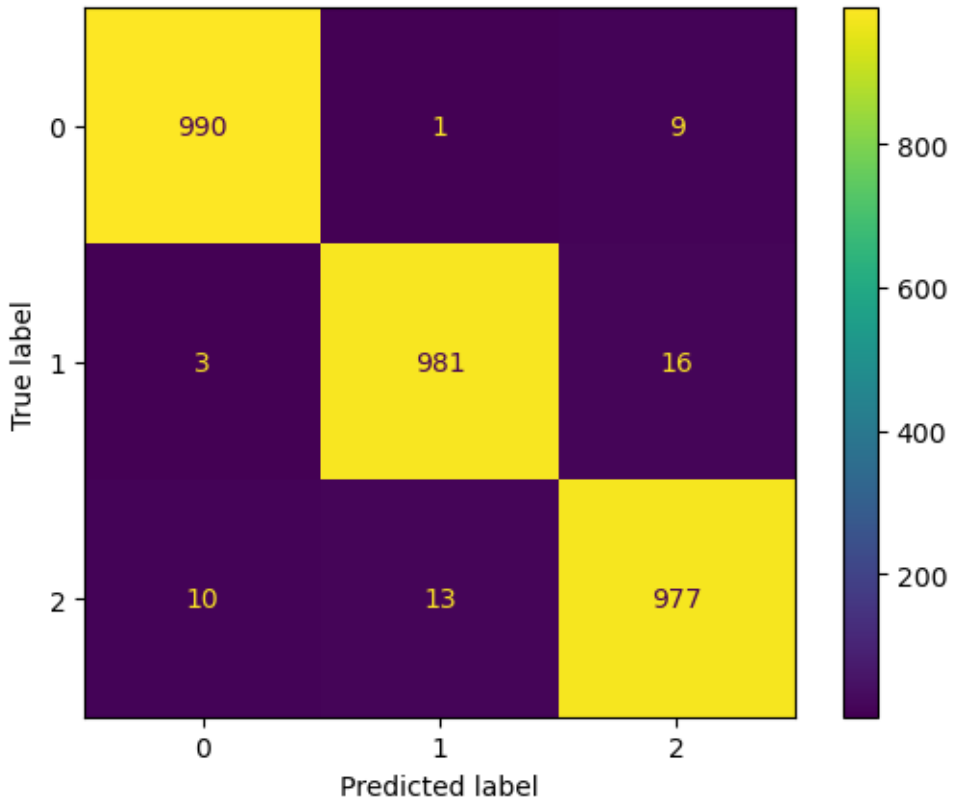
2) Loss Graph:



Training Loss = 5.00 %

Validation Loss = 5.30 %

3) Confusion Matrix:



4) Zero One Loss: -> Only 52 of 3000 image

There are 2948 of 3000 image that Model Predicted Correctly.

.....