Cervical Spine Fracture Detection

- **Introduction:**

Over 1.5 million spine fractures occur annually in the United States alone resulting in over 17,730 spinal cord injuries annually. The most common site of spine fracture is the cervical spine. There has been a rise in the incidence of spinal fractures in the elderly and in this population, fractures can be more difficult to detect on imaging due to superimposed degenerative disease and osteoporosis. Imaging diagnosis of adult spine fractures is now almost exclusively performed with computed tomography (CT) instead of radiographs (x-rays). Quickly detecting and determining the location of any vertebral fractures is essential to prevent neurologic deterioration and paralysis after trauma.

In this project, I will try to develop CNN models that match the radiologists' performance in detecting and localizing fractures to the seven vertebrae that comprise the cervical spine.

- **Dataset**:

Dataset Description

The goal of this competition is to identify fractures in CT scans of the cervical spine (neck) at both the level of a single vertebrae and the entire patient. Quickly detecting and determining the location of any vertebral fractures is essential to prevent neurologic deterioration and paralysis after trauma.

This competition uses a hidden test. When your submitted notebook is scored the actual test data (including a full length sample submission) will be made available to your notebook.

<u>Files</u>

train.csv :

Metadata for the train test set.

StudyInstanceUID - The study ID. There is one unique study ID for each patient scan.

patient_overall - One of the target columns. The patient level outcome, i.e. if any of the vertebrae are fractured.

C[1-7] - The other target columns. Whether the given vertebrae is fractured. See this diagram for the real location of each vertbrae in the spine.

test.csv :

Metadata for the test set prediction structure. Only the first few rows of the test set are available for download.

row_id - The row ID. This will match the same column in the sample submission file.

StudyInstanceUID - The study ID.

prediction_type - Which one of the eight target columns needs a prediction in this row.

[train/test]_images/[StudyInstanceUID]/[slice_number].dcm :

The image data, organized with one folder per scan. Expect to see roughly 1,500 scans in the hidden test set.
Each image is in the dicom file format. The DICOM image files are ≤ 1 mm slice thickness, axial orientation, and bone kernel. Note that some of the DICOM files are JPEG compressed. You may require additional resources to read the pixel array of these files, such as GDCM and pylibjpeg.

sample_submission.csv :

A valid sample submission.

row_id - The row ID. See the test.csv for what prediction needs to be filed in that row.

fractured - The target column.

train_bounding_boxes.csv Bounding boxes for a subset of the training set.

Segmentations:

Pixel level annotations for a subset of the training set. This data is provided in the nifti file format.

A portion of the imaging datasets have been segmented automatically using a 3D UNET model, and radiologists modified and approved the segmentations. The provided segmentation labels have values of 1 to 7 for C1 to C7 (seven cervical vertebrae) and 8 to 19 for T1 to T12 (twelve thoracic vertebrae are located in the center of your upper and middle back), and 0 for everything else. As we focused on the cervical spine, all scans have C1 to C7 labels but not all thoracic labels.

Be aware that the NIFTI files consist of segmentation in the sagittal plane, while the DICOM files are in the axial plane.

Datasets can be reached in:

https://www.kaggle.com/competitions/rsna-2022-cervical-spine-fracture-detection/data

https://www.kaggle.com/datasets/ujjwalkar/cervical-spine-segmentation (additional data set)

- **Method:**

Model Architecture: The model architecture in the provided code is based on transfer learning using the ResNet50 neural network. ResNet50 is a well-known architecture originally designed for image classification tasks. The architecture consists of convolutional layers followed by residual blocks, which help alleviate the vanishing gradient problem and enable training of very deep networks. In this code, the pre-trained ResNet50 model is used as the base model, followed by additional layers for classification.

The model architecture can be summarized as follows:

The base model: ResNet50 with pre-trained weights on ImageNet is loaded as the base model.

Flatten layer: The output from the base model is flattened to a 1D tensor.

Dense layer: A fully connected layer with 256 units and ReLU activation function is added.

Output layer: A dense layer with a single unit and sigmoid activation function is added for binary classification.

Model Functionality: The model takes cervical spine images as input and predicts the presence or absence of a specific clinical condition related to the spine. The ResNet50 base model extracts relevant features from the input images, which are then passed through the additional layers for classification. The sigmoid activation function in the output layer produces a probability score indicating the likelihood of the condition being present in each input image.

Choice of Model: The ResNet50 model is a popular choice for many computer vision tasks due to its effectiveness in capturing features from images and its ability to handle deep networks. Transfer learning with pre-trained models like ResNet50 saves significant computational resources and training time. By leveraging the knowledge learned from a large dataset (ImageNet), the model can generalize well to new tasks, even with limited labeled data. Therefore, choosing ResNet50 as a base model for cervical spine segmentation is a reasonable approach.

Model Training Strategy: The model training strategy in the provided code is supervised learning. The model is trained using labeled data, where each input image is associated with a corresponding binary label indicating the presence or absence of the clinical condition. The rationale behind supervised learning is that the model can learn to generalize patterns and make predictions based on the labeled examples. Since the task is binary classification, supervised learning is appropriate.

Loss Function and Optimizer: The loss function used is Binary Crossentropy, which is commonly used for binary classification problems. It measures the dissimilarity between the predicted probabilities and the true labels. The choice of Binary Crossentropy is appropriate because it encourages the model to optimize the predicted probabilities to match the true labels.

The optimizer used is Adam with a learning rate of 0.0005. Adam is an adaptive optimization algorithm that adjusts the learning rate during training to update the model parameters effectively. It combines the benefits of both AdaGrad and RMSProp algorithms. The learning rate of 0.0005 is a commonly used value, although the optimal learning rate can vary depending on the specific dataset and problem domain.

Overall, the choice of Binary Crossentropy loss function and Adam optimizer is reasonable for the binary classification task, as they are widely used and have proven to be effective in similar scenarios.

- **Experiments:**

This code is a combination of data preprocessing, visualization, model training, and evaluation for cervical spine segmentation.

Loading the dataset:

The code specifies the directory paths for the dataset (datadir), mask images (mask_dir), and train images (image_dir).

It creates lists of file paths for images and masks using os.listdir and os.path.join.

Visualization functions:

show_spine(n): Loads and displays an image and its corresponding masks for the nth example in the dataset.

crop_seg(n): Loads and displays an image with cropped regions of interest (based on masks) for the nth example.

save_crop_seg(n): Crops and saves the segmented regions of interest for the nth example.

Data processing and organization:

The code creates a list called dataset to store the processed data.

The save_crop_seg function is used to generate and save cropped masks for all examples in the dataset.

The code creates a pandas DataFrame called data to store information about the processed data.

Augmentation:

The code does not include explicit augmentation techniques. Augmentation is a common approach to increase the size and diversity of the training data by applying various transformations such as rotation, scaling, flipping, or adding noise to the images. Augmentation can help improve the model's generalization ability and robustness. However, in this case, augmentation techniques are not implemented in the code.

Dataset usage and data splitting:

The code splits the dataset into a training set and a test set using the train_test_split function from the sklearn library. The default test size is set to 0.33, meaning 33% of the data will be used for testing. The training set is used for training the model, and the test set is used for evaluating the model's performance.

Weight initialization:

In this case, the ResNet50 model is likely initialized with pre-trained weights on ImageNet. This is a common practice in transfer learning, where the weights learned from a large dataset (ImageNet) are used as a starting point for training on a different task (cervical spine segmentation).

Model building and training:

build_model(data, tl_name='resnet', batch_size=16): This function builds and trains a ResNet-based model for each category of cervical spine segmentation (C1, C2, C3, etc.).

The function splits the data into train and test sets and loads the images and labels accordingly.

It then creates a ResNet50 model and compiles it.

Training is performed using the loaded images and labels, with early stopping to prevent overfitting.

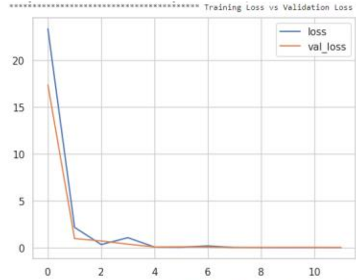
The function evaluates the model on the test set, displays training and validation loss/accuracy, and prints a classification report.

Finally, it plots the training/validation loss and accuracy and visualizes the confusion matrix.

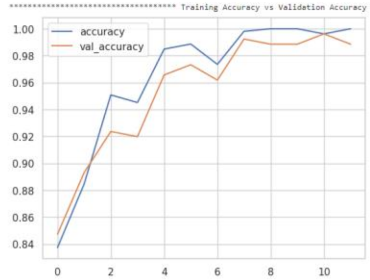
Model results:

Cervical Spine Fracture Detection in C1 vertbrae Results

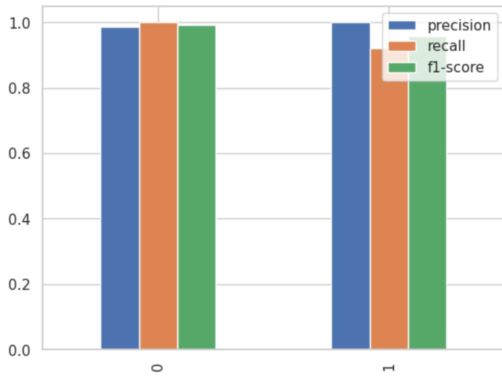
```
**************************** Category : C1, Output file name: spine_C1.h5 ***************************************
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 16, 16, 2048)      23581440

 flatten_1 (Flatten)         (None, 524288)            0

 dense_2 (Dense)             (None, 256)               134217984

 dense_3 (Dense)             (None, 1)                 257

=================================================================
Total params: 157,799,681
Trainable params: 157,746,561
Non-trainable params: 53,120
_________________________________________________________________
None
**********************************************  Training Data... **************************************************
Epoch 1/50
34/34 [==============================] - 69s 1s/step - loss: 23.3485 - accuracy: 0.8374 - val_loss: 17.3556 - val_accuracy: 0.8473
Epoch 2/50
34/34 [==============================] - 31s 911ms/step - loss: 2.1616 - accuracy: 0.8847 - val_loss: 0.9648 - val_accuracy: 0.8931
Epoch 3/50
34/34 [==============================] - 32s 953ms/step - loss: 0.3310 - accuracy: 0.9509 - val_loss: 0.7150 - val_accuracy: 0.9237
Epoch 4/50
34/34 [==============================] - 30s 878ms/step - loss: 1.0621 - accuracy: 0.9452 - val_loss: 0.3703 - val_accuracy: 0.9198
Epoch 5/50
34/34 [==============================] - 32s 953ms/step - loss: 0.0646 - accuracy: 0.9849 - val_loss: 0.0723 - val_accuracy: 0.9656
Epoch 6/50
34/34 [==============================] - 31s 902ms/step - loss: 0.0501 - accuracy: 0.9887 - val_loss: 0.0925 - val_accuracy: 0.9733
Epoch 7/50
34/34 [==============================] - 30s 877ms/step - loss: 0.1863 - accuracy: 0.9735 - val_loss: 0.0684 - val_accuracy: 0.9618
Epoch 8/50
34/34 [==============================] - 31s 904ms/step - loss: 0.0204 - accuracy: 0.9981 - val_loss: 0.0141 - val_accuracy: 0.9924
Epoch 9/50
34/34 [==============================] - 32s 952ms/step - loss: 7.2477e-04 - accuracy: 1.0000 - val_loss: 0.0276 - val_accuracy: 0.9885
Epoch 10/50
34/34 [==============================] - 30s 879ms/step - loss: 4.8174e-06 - accuracy: 1.0000 - val_loss: 0.0303 - val_accuracy: 0.9885
Epoch 11/50
34/34 [==============================] - 30s 878ms/step - loss: 0.0052 - accuracy: 0.9962 - val_loss: 0.0285 - val_accuracy: 0.9962
Epoch 12/50
34/34 [==============================] - 32s 949ms/step - loss: 2.5447e-04 - accuracy: 1.0000 - val_loss: 0.0258 - val_accuracy: 0.9885
******************************************  Dataset Trained Successfully  ******************************************
9/9 [==============================] - 7s 370ms/step
9/9 [==============================] - 4s 384ms/step - loss: 0.0276 - accuracy: 0.9885
***************************************** Training Loss vs Validation Loss *****************************************
```



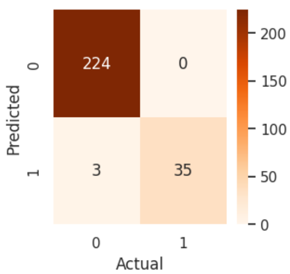

```
************************************ Training Accuracy vs Validation Accuracy ************************************
```



```
Classification Report:

              precision    recall  f1-score   support

           0       0.99      1.00      0.99       224
           1       1.00      0.92      0.96        38

    accuracy                           0.99       262
   macro avg       0.99      0.96      0.98       262
weighted avg       0.99      0.99      0.99       262

****************************************************** Plotting The Result ********************************************************
```

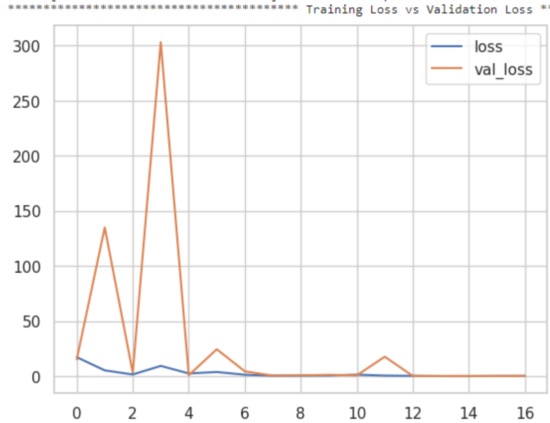



```
************************************************** Confusion Matrix **************************************************
```

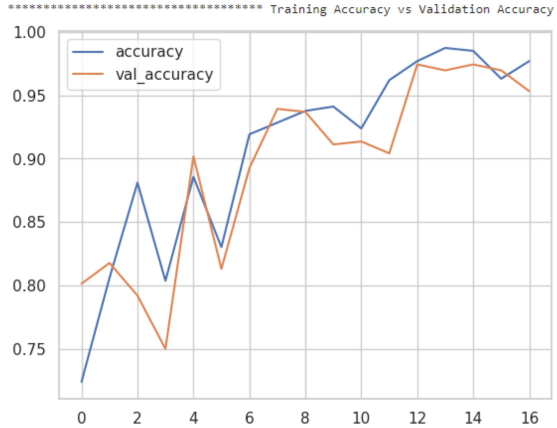


Cervical Spine Fracture Detection in C2 vertbrae Results

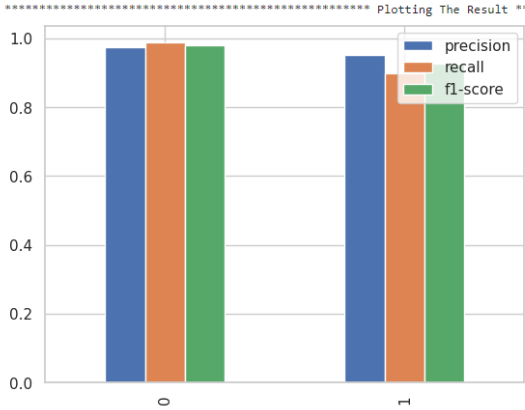
```
**************************** Category : C2, Output file name: spine_C2.h5 ***************************************
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 16, 16, 2048)      23581440

 flatten_2 (Flatten)         (None, 524288)            0

 dense_4 (Dense)             (None, 256)               134217984

 dense_5 (Dense)             (None, 1)                 257

=================================================================
Total params: 157,799,681
Trainable params: 157,746,561
Non-trainable params: 53,120
_________________________________________________________________
None
**********************************************  Training Data... **************************************************
Epoch 1/50
55/55 [==============================] - 90s 1s/step - loss: 17.2555 - accuracy: 0.7240 - val_loss: 15.2880 - val_accuracy: 0.8014
Epoch 2/50
55/55 [==============================] - 54s 986ms/step - loss: 5.2739 - accuracy: 0.8060 - val_loss: 134.9459 - val_accuracy: 0.8178
Epoch 3/50
55/55 [==============================] - 54s 988ms/step - loss: 1.5613 - accuracy: 0.8811 - val_loss: 3.2596 - val_accuracy: 0.7921
Epoch 4/50
55/55 [==============================] - 53s 968ms/step - loss: 9.3457 - accuracy: 0.8037 - val_loss: 302.9967 - val_accuracy: 0.7500
Epoch 5/50
55/55 [==============================] - 50s 903ms/step - loss: 2.5025 - accuracy: 0.8857 - val_loss: 0.8414 - val_accuracy: 0.9019
Epoch 6/50
55/55 [==============================] - 53s 968ms/step - loss: 3.7986 - accuracy: 0.8303 - val_loss: 24.4293 - val_accuracy: 0.8131
Epoch 7/50
55/55 [==============================] - 49s 901ms/step - loss: 1.3263 - accuracy: 0.9192 - val_loss: 4.4422 - val_accuracy: 0.8925
Epoch 8/50
55/55 [==============================] - 54s 984ms/step - loss: 0.4922 - accuracy: 0.9284 - val_loss: 0.5000 - val_accuracy: 0.9393
Epoch 9/50
55/55 [==============================] - 50s 902ms/step - loss: 0.5287 - accuracy: 0.9376 - val_loss: 0.7308 - val_accuracy: 0.9369
Epoch 10/50
55/55 [==============================] - 54s 983ms/step - loss: 0.6322 - accuracy: 0.9411 - val_loss: 1.2465 - val_accuracy: 0.9112
Epoch 11/50
55/55 [==============================] - 49s 886ms/step - loss: 1.4137 - accuracy: 0.9238 - val_loss: 0.7165 - val_accuracy: 0.9136
Epoch 12/50
55/55 [==============================] - 49s 899ms/step - loss: 0.5615 - accuracy: 0.9619 - val_loss: 17.6770 - val_accuracy: 0.9042
Epoch 13/50
55/55 [==============================] - 49s 901ms/step - loss: 0.3125 - accuracy: 0.9769 - val_loss: 0.2094 - val_accuracy: 0.9743
Epoch 14/50
55/55 [==============================] - 54s 983ms/step - loss: 0.0844 - accuracy: 0.9873 - val_loss: 0.1560 - val_accuracy: 0.9696
Epoch 15/50
55/55 [==============================] - 49s 885ms/step - loss: 0.0676 - accuracy: 0.9850 - val_loss: 0.2055 - val_accuracy: 0.9743
Epoch 16/50
55/55 [==============================] - 53s 966ms/step - loss: 0.2614 - accuracy: 0.9630 - val_loss: 0.1019 - val_accuracy: 0.9696
Epoch 17/50
55/55 [==============================] - 54s 982ms/step - loss: 0.2582 - accuracy: 0.9769 - val_loss: 0.2414 - val_accuracy: 0.9533
*******************************************  Dataset Trained Successfully  *****************************************
14/14 [==============================] - 6s 420ms/step
14/14 [==============================] - 6s 404ms/step - loss: 0.1560 - accuracy: 0.9696
*******************************************  Training Loss vs Validation Loss *****************************************
```



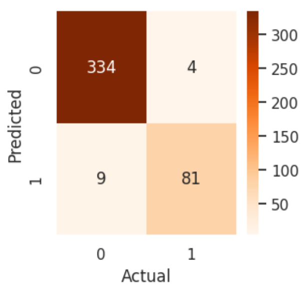

```
************************************** Training Accuracy vs Validation Accuracy ***************************************
```



```
****************************************************** Result ********************************************************
Classification Report:

              precision    recall  f1-score   support

           0       0.97      0.99      0.98       338
           1       0.95      0.90      0.93        90

    accuracy                           0.97       428
   macro avg       0.96      0.94      0.95       428
weighted avg       0.97      0.97      0.97       428

****************************************************** Plotting The Result ********************************************************
```



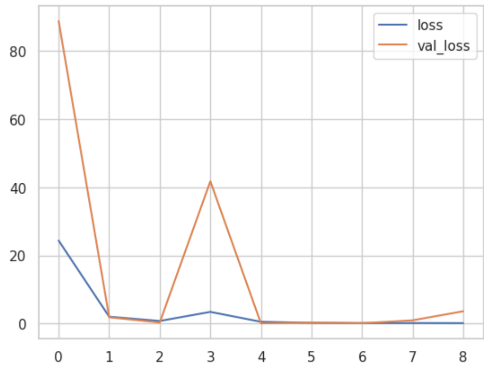

```
************************************************** Confusion Matrix ***************************************************
```



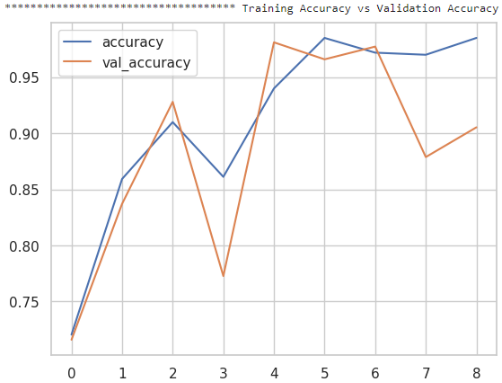
Cervical Spine Fracture Detection in C3 vertbrae Results

```
***************************** Category : C3, Output file name: spine_C3.h5 ***************************************
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 16, 16, 2048)      23581440

 flatten_3 (Flatten)         (None, 524288)            0

 dense_6 (Dense)             (None, 256)               134217984

 dense_7 (Dense)             (None, 1)                 257

=================================================================
Total params: 157,799,681
Trainable params: 157,746,561
Non-trainable params: 53,120
_________________________________________________________________
None
**********************************************  Training Data... **************************************************
Epoch 1/50
67/67 [==============================] - 74s 582ms/step - loss: 24.3452 - accuracy: 0.7205 - val_loss: 88.8398 - val_accuracy: 0.7159
Epoch 2/50
67/67 [==============================] - 33s 491ms/step - loss: 1.9853 - accuracy: 0.8593 - val_loss: 1.8058 - val_accuracy: 0.8371
Epoch 3/50
67/67 [==============================] - 33s 499ms/step - loss: 0.7325 - accuracy: 0.9099 - val_loss: 0.2874 - val_accuracy: 0.9280
Epoch 4/50
67/67 [==============================] - 33s 501ms/step - loss: 3.4064 - accuracy: 0.8612 - val_loss: 41.7688 - val_accuracy: 0.7727
Epoch 5/50
67/67 [==============================] - 34s 513ms/step - loss: 0.4967 - accuracy: 0.9400 - val_loss: 0.0532 - val_accuracy: 0.9811
Epoch 6/50
67/67 [==============================] - 34s 514ms/step - loss: 0.1870 - accuracy: 0.9850 - val_loss: 0.2420 - val_accuracy: 0.9659
Epoch 7/50
67/67 [==============================] - 32s 480ms/step - loss: 0.1139 - accuracy: 0.9719 - val_loss: 0.0791 - val_accuracy: 0.9773
Epoch 8/50
67/67 [==============================] - 33s 501ms/step - loss: 0.1339 - accuracy: 0.9700 - val_loss: 0.9228 - val_accuracy: 0.8788
Epoch 9/50
67/67 [==============================] - 34s 511ms/step - loss: 0.1007 - accuracy: 0.9850 - val_loss: 3.5873 - val_accuracy: 0.9053
*****************************************  Dataset Trained Successfully  *****************************************
9/9 [==============================] - 4s 424ms/step
9/9 [==============================] - 5s 386ms/step - loss: 0.2420 - accuracy: 0.9659
```

************************************ Training Loss vs Validation Loss ************************************

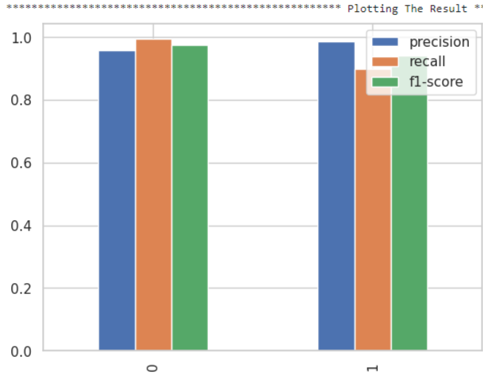


********************************* Training Accuracy vs Validation Accuracy *********************************

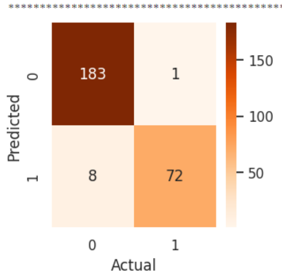



```
Classification Report:

              precision    recall  f1-score   support

           0       0.96      0.99      0.98       184
           1       0.99      0.90      0.94        80

    accuracy                           0.97       264
   macro avg       0.97      0.95      0.96       264
weighted avg       0.97      0.97      0.97       264
```

** Plotting The Result **

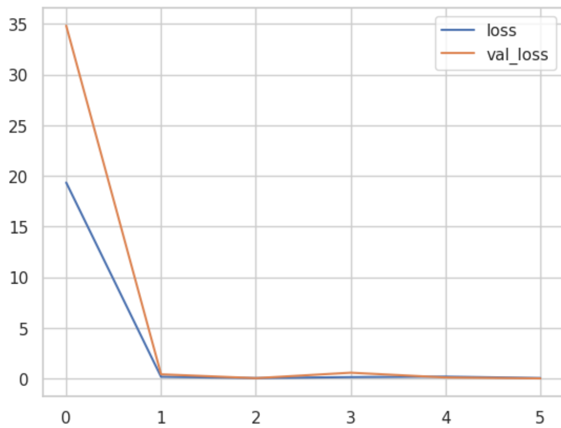


*** Confusion Matrix ***

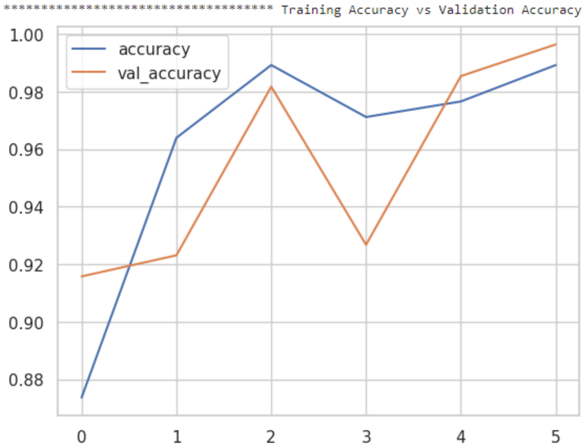


Cervical Spine Fracture Detection in C4 vertbrae Results

```
**************************** Category : C4, Output file name: spine_C4.h5 ***************************************
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 16, 16, 2048)      23581440

 flatten_4 (Flatten)         (None, 524288)            0

 dense_8 (Dense)             (None, 256)               134217984

 dense_9 (Dense)             (None, 1)                 257

=================================================================
Total params: 157,799,681
Trainable params: 157,746,561
Non-trainable params: 53,120
_________________________________________________________________
None
**********************************************  Training Data... **************************************************
Epoch 1/50
70/70 [==============================] - 70s 542ms/step - loss: 19.3454 - accuracy: 0.8736 - val_loss: 34.8305 - val_accuracy: 0.9158
Epoch 2/50
70/70 [==============================] - 34s 485ms/step - loss: 0.1765 - accuracy: 0.9639 - val_loss: 0.4179 - val_accuracy: 0.9231
Epoch 3/50
70/70 [==============================] - 34s 490ms/step - loss: 0.0454 - accuracy: 0.9892 - val_loss: 0.0389 - val_accuracy: 0.9817
Epoch 4/50
70/70 [==============================] - 35s 495ms/step - loss: 0.1427 - accuracy: 0.9711 - val_loss: 0.5771 - val_accuracy: 0.9267
Epoch 5/50
70/70 [==============================] - 33s 476ms/step - loss: 0.1886 - accuracy: 0.9765 - val_loss: 0.1068 - val_accuracy: 0.9853
Epoch 6/50
70/70 [==============================] - 34s 489ms/step - loss: 0.0481 - accuracy: 0.9892 - val_loss: 0.0144 - val_accuracy: 0.9963
*******************************************  Dataset Trained Successfully  ****************************************
9/9 [==============================] - 6s 676ms/step
9/9 [==============================] - 4s 402ms/step - loss: 0.0389 - accuracy: 0.9817
*************************************** Training Loss vs Validation Loss *****************************************
```

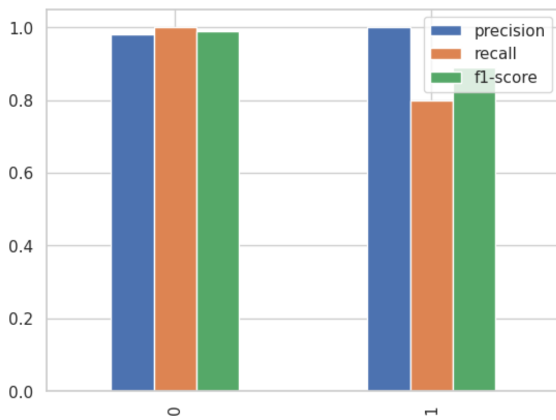



```
*********************************** Training Accuracy vs Validation Accuracy ************************************
```

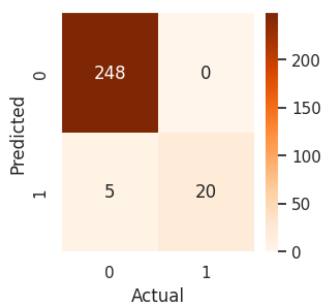


```
Classification Report:

              precision    recall  f1-score   support

           0       0.98      1.00      0.99       248
           1       1.00      0.80      0.89        25

    accuracy                           0.98       273
   macro avg       0.99      0.90      0.94       273
weighted avg       0.98      0.98      0.98       273
```

```
**************************************************** Plotting The Result ****************************************************
```

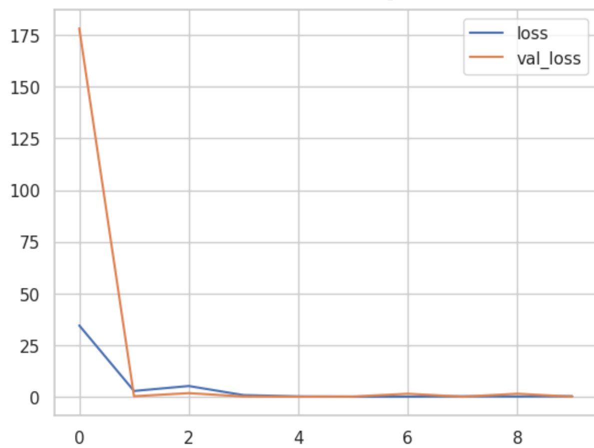



```
************************************************ Confusion Matrix ************************************************
```

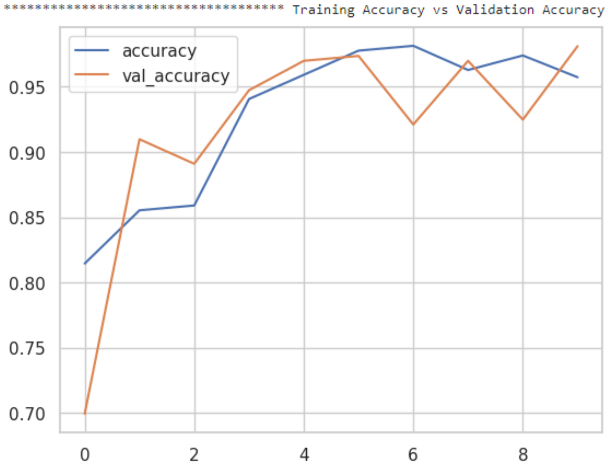


Cervical Spine Fracture Detection in C5 vertbrae Results

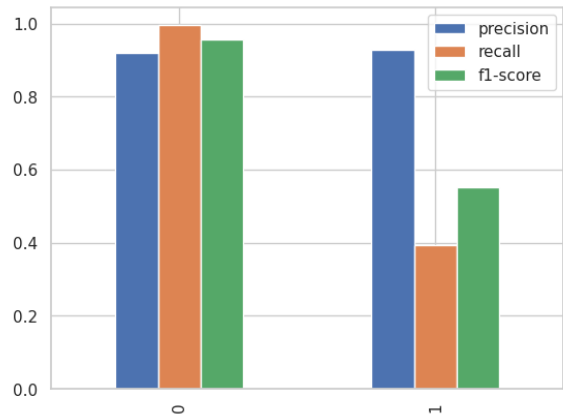
```
**************************** Category : C5, Output file name: spine_C5.h5 ***************************************
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 16, 16, 2048)      23581440

 flatten_5 (Flatten)         (None, 524288)            0

 dense_10 (Dense)            (None, 256)               134217984

 dense_11 (Dense)            (None, 1)                 257

=================================================================
Total params: 157,799,681
Trainable params: 157,746,561
Non-trainable params: 53,120
_________________________________________________________________
None
*********************************************  Training Data... *************************************************
Epoch 1/50
68/68 [==============================] - 71s 576ms/step - loss: 34.5623 - accuracy: 0.8145 - val_loss: 178.1740 - val_accuracy: 0.6992
Epoch 2/50
68/68 [==============================] - 35s 509ms/step - loss: 2.8739 - accuracy: 0.8553 - val_loss: 0.3398 - val_accuracy: 0.9098
Epoch 3/50
68/68 [==============================] - 35s 508ms/step - loss: 5.2682 - accuracy: 0.8590 - val_loss: 1.8226 - val_accuracy: 0.8910
Epoch 4/50
68/68 [==============================] - 35s 510ms/step - loss: 0.8707 - accuracy: 0.9406 - val_loss: 0.2205 - val_accuracy: 0.9474
Epoch 5/50
68/68 [==============================] - 35s 509ms/step - loss: 0.2397 - accuracy: 0.9592 - val_loss: 0.0934 - val_accuracy: 0.9699
Epoch 6/50
68/68 [==============================] - 34s 507ms/step - loss: 0.1111 - accuracy: 0.9777 - val_loss: 0.1817 - val_accuracy: 0.9737
Epoch 7/50
68/68 [==============================] - 33s 491ms/step - loss: 0.1047 - accuracy: 0.9814 - val_loss: 1.5204 - val_accuracy: 0.9211
Epoch 8/50
68/68 [==============================] - 34s 497ms/step - loss: 0.2860 - accuracy: 0.9629 - val_loss: 0.1071 - val_accuracy: 0.9699
Epoch 9/50
68/68 [==============================] - 34s 495ms/step - loss: 0.1818 - accuracy: 0.9740 - val_loss: 1.5373 - val_accuracy: 0.9248
Epoch 10/50
68/68 [==============================] - 33s 490ms/step - loss: 0.3393 - accuracy: 0.9573 - val_loss: 0.1137 - val_accuracy: 0.9812
******************************************  Dataset Trained Successfully  ****************************************
9/9 [==============================] - 6s 610ms/step
9/9 [==============================] - 4s 391ms/step - loss: 1.5204 - accuracy: 0.9211
********************************************* Training Loss vs Validation Loss *********************************************
```



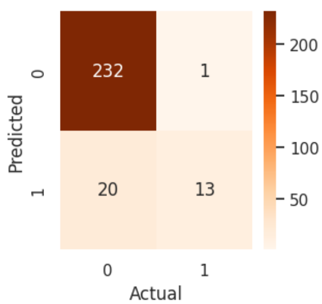

```
************************************ Training Accuracy vs Validation Accuracy *************************************
```



```
****************************************************** Result ******************************************************
Classification Report:

              precision    recall  f1-score   support

           0       0.92      1.00      0.96       233
           1       0.93      0.39      0.55        33

    accuracy                           0.92       266
   macro avg       0.92      0.69      0.75       266
weighted avg       0.92      0.92      0.91       266
****************************************************** Plotting The Result ******************************************************
```

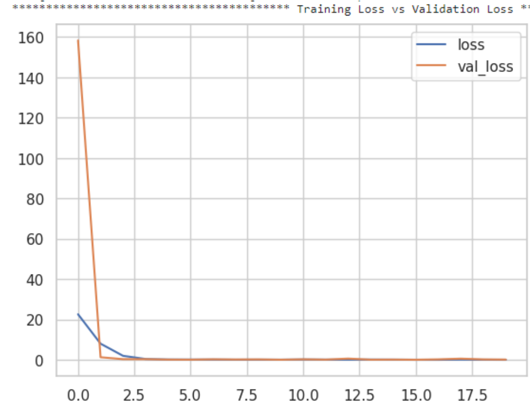



```
****************************************************** Confusion Matrix ******************************************************
```

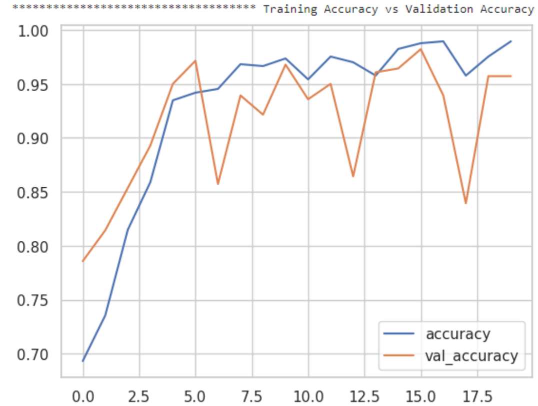


Cervical Spine Fracture Detection in C6 vertbrae Results

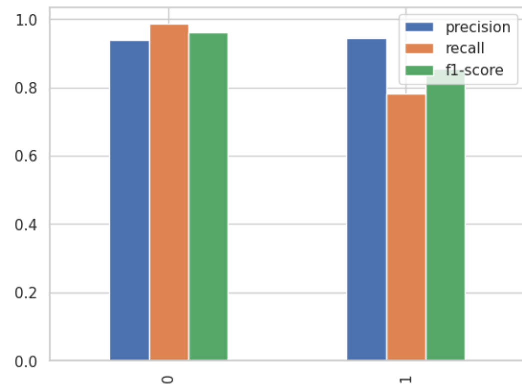
```
*************************** Category : C6, Output file name: spine_C6.h5 **************************************
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 16, 16, 2048)      23581440

 flatten_6 (Flatten)         (None, 524288)            0

 dense_12 (Dense)            (None, 256)               134217984

 dense_13 (Dense)            (None, 1)                 257

=================================================================
Total params: 157,799,681
Trainable params: 157,746,561
Non-trainable params: 53,120
_________________________________________________________________
None
********************************************  Training Data... *************************************************
Epoch 1/50
142/142 [==============================] - 80s 294ms/step - loss: 22.5924 - accuracy: 0.6931 - val_loss: 158.2847 - val_accuracy: 0.7857
Epoch 2/50
142/142 [==============================] - 39s 275ms/step - loss: 8.0269 - accuracy: 0.7354 - val_loss: 1.2805 - val_accuracy: 0.8143
Epoch 3/50
142/142 [==============================] - 39s 277ms/step - loss: 1.9857 - accuracy: 0.8148 - val_loss: 0.3382 - val_accuracy: 0.8536
Epoch 4/50
142/142 [==============================] - 39s 275ms/step - loss: 0.4174 - accuracy: 0.8589 - val_loss: 0.3291 - val_accuracy: 0.8929
Epoch 5/50
142/142 [==============================] - 38s 270ms/step - loss: 0.1641 - accuracy: 0.9347 - val_loss: 0.1281 - val_accuracy: 0.9500
Epoch 6/50
142/142 [==============================] - 38s 269ms/step - loss: 0.1297 - accuracy: 0.9418 - val_loss: 0.0971 - val_accuracy: 0.9714
Epoch 7/50
142/142 [==============================] - 39s 276ms/step - loss: 0.1424 - accuracy: 0.9453 - val_loss: 0.2587 - val_accuracy: 0.8571
Epoch 8/50
142/142 [==============================] - 38s 268ms/step - loss: 0.0758 - accuracy: 0.9683 - val_loss: 0.1687 - val_accuracy: 0.9393
Epoch 9/50
142/142 [==============================] - 37s 263ms/step - loss: 0.0964 - accuracy: 0.9665 - val_loss: 0.2133 - val_accuracy: 0.9214
Epoch 10/50
142/142 [==============================] - 39s 276ms/step - loss: 0.0641 - accuracy: 0.9735 - val_loss: 0.0706 - val_accuracy: 0.9679
Epoch 11/50
142/142 [==============================] - 38s 269ms/step - loss: 0.1752 - accuracy: 0.9541 - val_loss: 0.2544 - val_accuracy: 0.9357
Epoch 12/50
142/142 [==============================] - 39s 275ms/step - loss: 0.1019 - accuracy: 0.9753 - val_loss: 0.1171 - val_accuracy: 0.9500
Epoch 13/50
142/142 [==============================] - 37s 263ms/step - loss: 0.1001 - accuracy: 0.9700 - val_loss: 0.6180 - val_accuracy: 0.8643
Epoch 14/50
142/142 [==============================] - 37s 263ms/step - loss: 0.1385 - accuracy: 0.9577 - val_loss: 0.0918 - val_accuracy: 0.9607
Epoch 15/50
142/142 [==============================] - 39s 276ms/step - loss: 0.0965 - accuracy: 0.9824 - val_loss: 0.1320 - val_accuracy: 0.9643
Epoch 16/50
142/142 [==============================] - 39s 275ms/step - loss: 0.0363 - accuracy: 0.9877 - val_loss: 0.0540 - val_accuracy: 0.9821
Epoch 17/50
142/142 [==============================] - 39s 275ms/step - loss: 0.0376 - accuracy: 0.9894 - val_loss: 0.2355 - val_accuracy: 0.9393
Epoch 18/50
142/142 [==============================] - 37s 263ms/step - loss: 0.1395 - accuracy: 0.9577 - val_loss: 0.5884 - val_accuracy: 0.8393
Epoch 19/50
142/142 [==============================] - 38s 269ms/step - loss: 0.0705 - accuracy: 0.9753 - val_loss: 0.2093 - val_accuracy: 0.9571
Epoch 20/50
142/142 [==============================] - 38s 268ms/step - loss: 0.0395 - accuracy: 0.9894 - val_loss: 0.1337 - val_accuracy: 0.9571
*******************************************  Dataset Trained Successfully  ****************************************
 ***************************************** Training Loss vs Validation Loss ****************************************
```



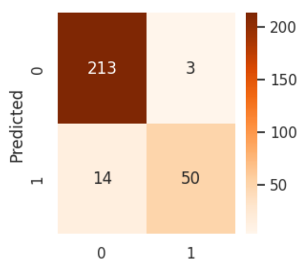

```
 *********************************** Training Accuracy vs Validation Accuracy ************************************
```



```
****************************************************** Result ********************************************************
Classification Report:

              precision    recall  f1-score   support

           0       0.94      0.99      0.96       216
           1       0.94      0.78      0.85        64

    accuracy                           0.94       280
   macro avg       0.94      0.88      0.91       280
weighted avg       0.94      0.94      0.94       280

****************************************************** Plotting The Result ******************************************************
```



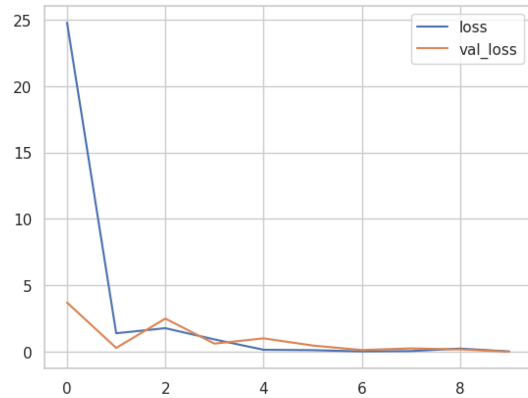

```
************************************************** Confusion Matrix **************************************************
```



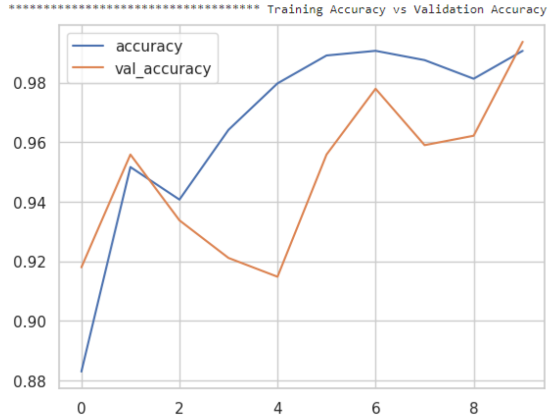
Cervical Spine Fracture Detection in C7 vertbrae Results

```
**************************** Category : C7, Output file name: spine_C7.h5 ****************************************
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 16, 16, 2048)      23581440

 flatten_7 (Flatten)         (None, 524288)            0

 dense_14 (Dense)            (None, 256)               134217984

 dense_15 (Dense)            (None, 1)                 257

=================================================================
Total params: 157,799,681
Trainable params: 157,746,561
Non-trainable params: 53,120
_________________________________________________________________
None
*********************************************** Training Data... *****************************************************
Epoch 1/50
161/161 [==============================] - 79s 286ms/step - loss: 24.7683 - accuracy: 0.8830 - val_loss: 3.7061 - val_accuracy: 0.9180
Epoch 2/50
161/161 [==============================] - 43s 264ms/step - loss: 1.3980 - accuracy: 0.9516 - val_loss: 0.2964 - val_accuracy: 0.9558
Epoch 3/50
161/161 [==============================] - 42s 263ms/step - loss: 1.7797 - accuracy: 0.9407 - val_loss: 2.4989 - val_accuracy: 0.9338
Epoch 4/50
161/161 [==============================] - 43s 268ms/step - loss: 0.9348 - accuracy: 0.9641 - val_loss: 0.6192 - val_accuracy: 0.9211
Epoch 5/50
161/161 [==============================] - 43s 266ms/step - loss: 0.1565 - accuracy: 0.9797 - val_loss: 1.0134 - val_accuracy: 0.9148
Epoch 6/50
161/161 [==============================] - 43s 267ms/step - loss: 0.1274 - accuracy: 0.9891 - val_loss: 0.4762 - val_accuracy: 0.9558
Epoch 7/50
161/161 [==============================] - 43s 268ms/step - loss: 0.0352 - accuracy: 0.9906 - val_loss: 0.1352 - val_accuracy: 0.9779
Epoch 8/50
161/161 [==============================] - 42s 261ms/step - loss: 0.0604 - accuracy: 0.9875 - val_loss: 0.2608 - val_accuracy: 0.9590
Epoch 9/50
161/161 [==============================] - 42s 263ms/step - loss: 0.2451 - accuracy: 0.9813 - val_loss: 0.1849 - val_accuracy: 0.9621
Epoch 10/50
161/161 [==============================] - 43s 267ms/step - loss: 0.0317 - accuracy: 0.9906 - val_loss: 0.0111 - val_accuracy: 0.9937
******************************************* Dataset Trained Successfully ******************************************
10/10 [==============================] - 8s 732ms/step
10/10 [==============================] - 4s 421ms/step - loss: 0.1352 - accuracy: 0.9779
```

************************************** Training Loss vs Validation Loss **************************************

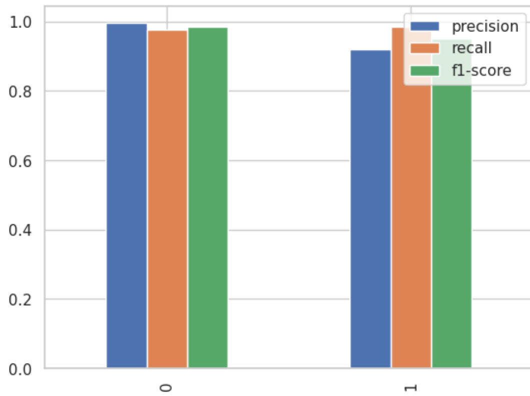


********************************** Training Accuracy vs Validation Accuracy **********************************

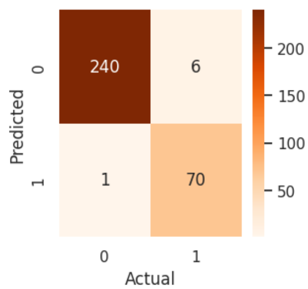


```
Classification Report:

              precision    recall  f1-score   support

           0       1.00      0.98      0.99       246
           1       0.92      0.99      0.95        71

    accuracy                           0.98       317
   macro avg       0.96      0.98      0.97       317
weighted avg       0.98      0.98      0.98       317
```

*** Plotting The Result **



*** Confusion Matrix ***



- **Conclusion and Discussion:**

Model is generalize succesfuly.

Additional Steps: Data Augmentation can be applicable for better generalized models. Also, instead of binary class classification, multiclass classification can be used.