BMS COLLEGE OF ENGINEERING, BANGALORE – 560 019

(Autonomous institute, Affiliated to VTU)

Department of Information Science and Engineering

Deep Learning - 20IS6PEDLG

Classification of Alzheimer's Disease Stages

2021 – 2022 – EVEN SEMESTER

Submitted by~

Ankit Das - 1BM19IS025
Deven Prakash Paramaj - 1BM19IS048

BMS COLLEGE OF ENGINEERING, BANGALORE -19

(An autonomous institute, affiliated to VTU)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

2021 – 2022 – EVEN SEMESTER

CERTIFICATE

Certified that Mr. Ankit Das bearing USN 1BM19IS025 and Mr. Deven Prakash Paramaj bearing USN 1BM19IS048 of Sixth Semester belonging to the Department of Information Science and Engineering had successfully completed AAT as a part of the course Deep Learning [20IS6PEDLG].

Faculty Incharge~

Rashmi R

BMS COLLEGE OF ENGINEERING, BANGALORE -19

(An autonomous institute, affiliated to VTU)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
2021 – 2022 – EVEN SEMESTER

CONTENTS

Course: Deep Learning **Course code: 20IS6PEDLG**

Abstract

[1] An early and accurate diagnosis of Alzheimer's disease (AD) and it's stages is crucial for patient treatment and care, so that patients can take precautionary actions before irreversible brain damage develops since they are aware of the severity and progression risks.

The role of structural brain Magnetic Resonance Imaging (MRI) is becoming more and more emphasized in the early diagnostics of Alzheimer's disease (AD).

In this study, we design a Deep Learning architecture, which contains CNN-SVM hybrid model for detection of Alzheimer's disease (AD) as a base model and another CNN-SVM hybrid model is trained with the freezed base model in parallel, stacked over a softmax output layer, to overcome the bottleneck in classification and aid the diagnosis of Alzheimer's disease (AD) and its prodromal stage, Mild Cognitive Impairment (MCI).

Alzheimer's Disease

[2] Alzheimer's Disease affects people in a numerous way. Patients suffer from memory loss, confusion, difficulty in speaking, reading or writing. Eventually, they may forget about their life and could not recognize even their family members. They can forget how to perform daily activities such as brushing teeth or combing hair.

As a result, it makes people anxious or aggressive or to wander away from home. Alzheimer's Disease can even cause death in elder people.

There are three major stages in Alzheimer's Disease - very mild, mild and moderate. Detection of Alzheimer's Disease (AD) is still not accurate until the patient reaches a moderate AD. But early detection and classification of AD are critical for proper treatment and preventing brain tissue damage.

Several things are needed for proper medical assessment of AD. Physical and neurobiological exams, Mini-Mental State Examination (MMSE), and patient's detailed history are required for accurate AD detection and classification.

In recent years, doctors are using brain Magnetic Resonance Imaging (MRI) data for earlier detection of Alzheimer's Disease.

Problem Statement

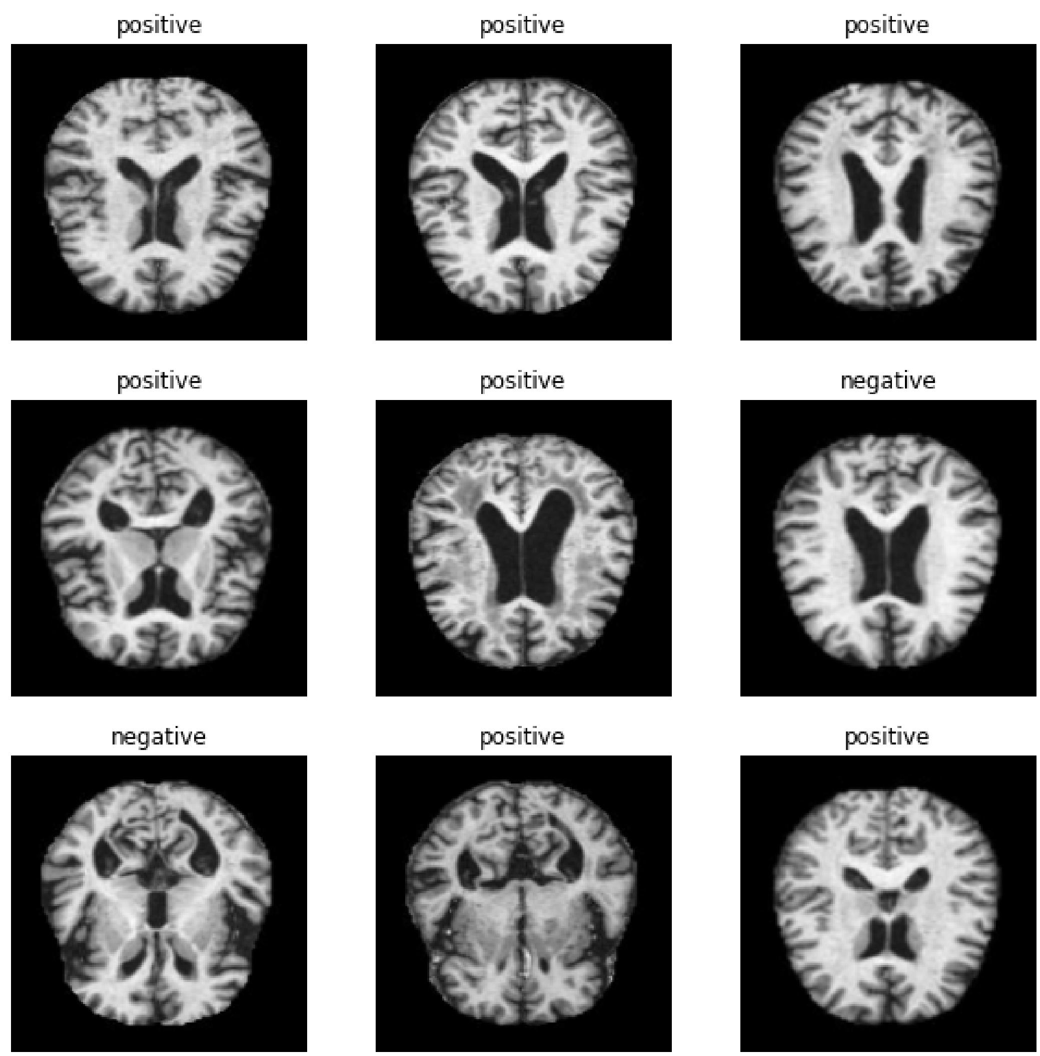
To classify the MRI images into 4 stages of Mild Cognitive Impairment (MCI) i.e. Very Mild Demented, Mild Demented, Moderate Demented and Non Demented

Alzheimer’s Disease MRI Images Dataset

[3] In recent years, doctors are using brain Magnetic Resonance Imaging (MRI) data for earlier detection of Alzheimer’s Disease.

In this study we are using two datasets of MRI Images :-

- Dataset_AD - contains 2 classes of images, positive and negative which depicts if the person has Alzhemer’s Disease or not.
    - Positive Class → 5932 images
    - Negative Class → 5760 images



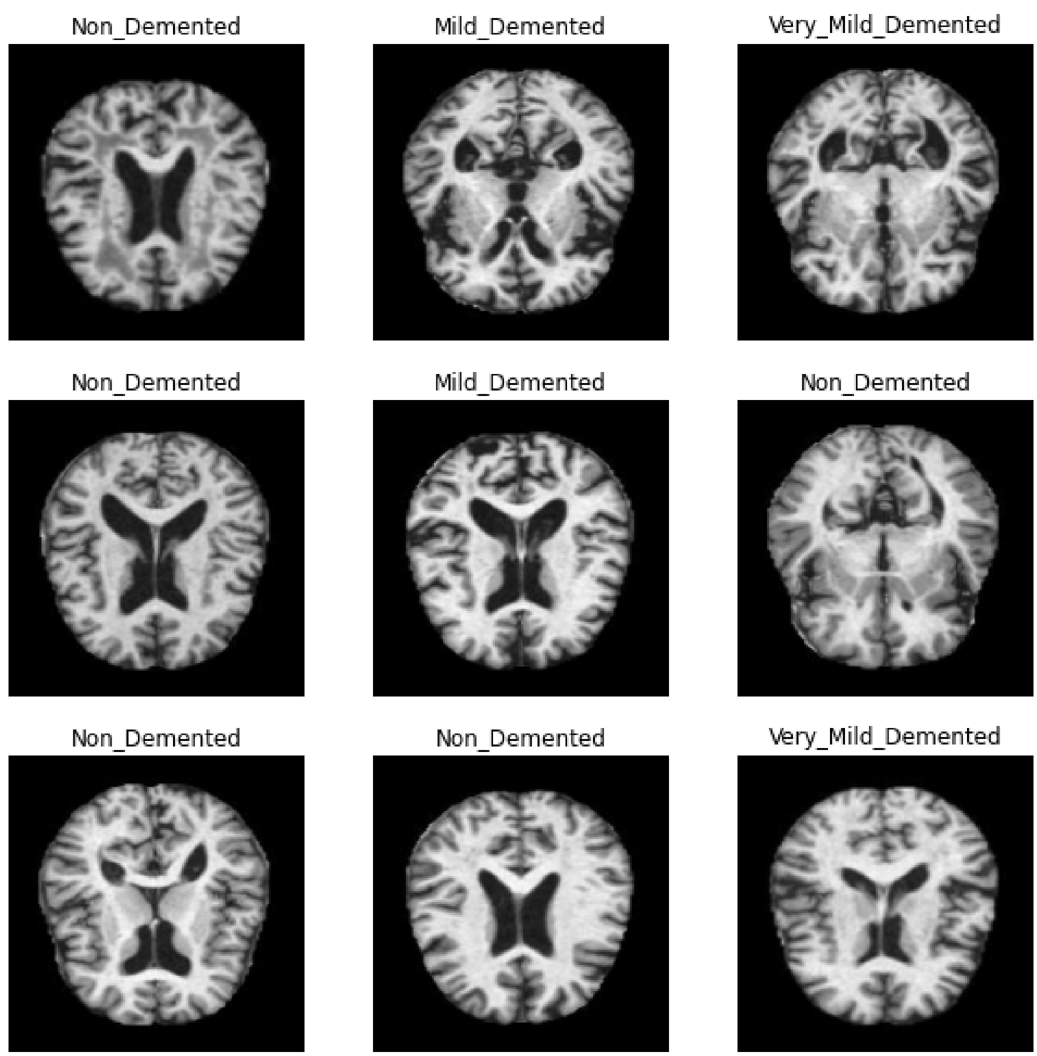
❖ Dataset_AD_stages - contains 2 partition train and test which further contains 4 classes of images i.e. Very Mild Demented, Mild Demented, Moderate Demented and Non Demented which depicts the stages of Mild Cognitive Impairment (MCI).

Train dataset:

- Very Mild Demented → 4032 images
- Mild Demented → 1613 images
- Moderate Demented → 116 images
- Non Demented → 5760 images

Test dataset:

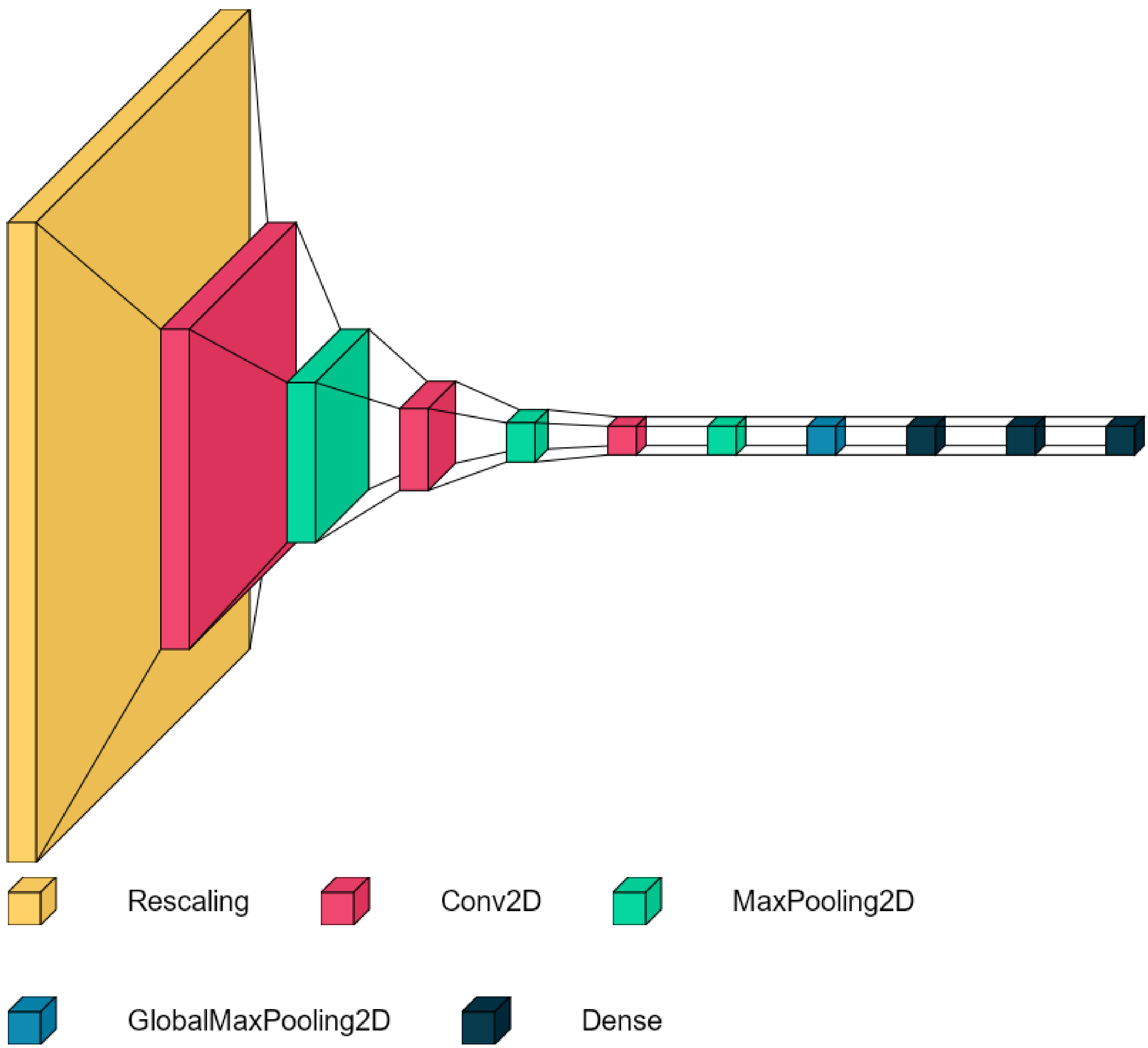
- Very Mild Demented → 448 images
- Mild Demented → 179 images
- Moderate Demented → 12 images
- Non Demented → 640 images



Alzheimer's Disease Detection Model (Base Model)

In this we use **CNN-SVM hybrid model** as the base model to detect if the person has Alzheimer's or not. This model is trained on the **Dataset_AD** dataset mentioned in the previous section.

Architecture of the Model -



model.summary() →

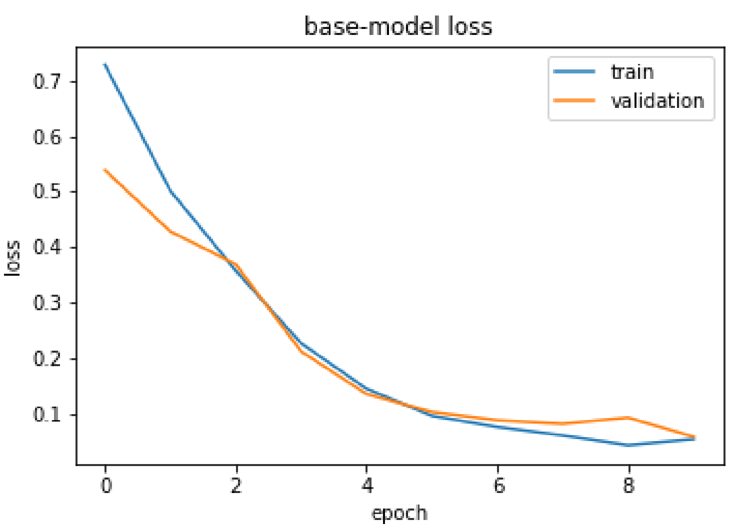
```
Model: "base_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 rescaling_1 (Rescaling)     (None, 180, 180, 3)       0         
                                                                 
 conv2d_3 (Conv2D)           (None, 90, 90, 16)        448       
                                                                 
 max_pooling2d_3 (MaxPooling  (None, 45, 45, 16)       0         
 2D)                                                             
                                                                 
 conv2d_4 (Conv2D)           (None, 23, 23, 32)        4640      
                                                                 
 max_pooling2d_4 (MaxPooling  (None, 11, 11, 32)       0         
 2D)                                                             
                                                                 
 conv2d_5 (Conv2D)           (None, 6, 6, 64)          18496     
                                                                 
 max_pooling2d_5 (MaxPooling  (None, 3, 3, 64)         0         
 2D)                                                             
                                                                 
 global_max_pooling2d_1 (Glo  (None, 64)               0         
 balMaxPooling2D)                                                
                                                                 
 dense_2 (Dense)             (None, 128)               8320      
                                                                 
 SVM (Dense)                 (None, 128)               16512     
                                                                 
 dense_3 (Dense)             (None, 2)                 258       
                                                                 
=================================================================
Total params: 48,674
Trainable params: 48,674
Non-trainable params: 0
_________________________________________________________________
```

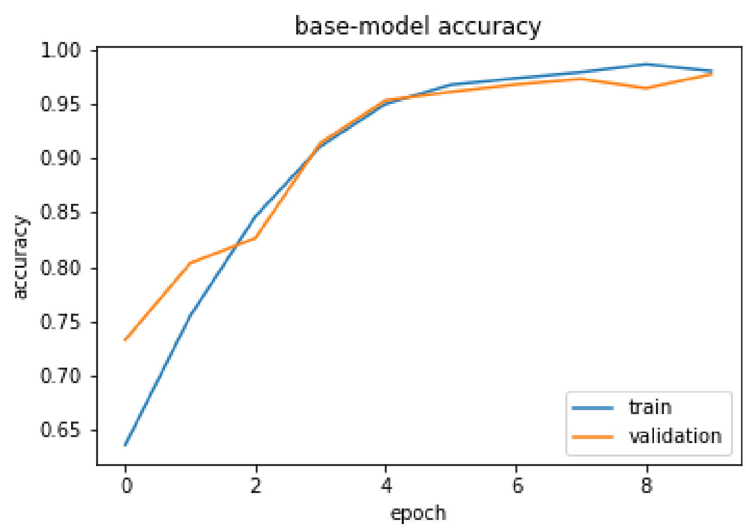
Model training →

```
callback1 = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=1, verbose=1)
callback2 = tf.keras.callbacks.EarlyStopping(monitor='accuracy', patience=1, verbose=1)
callback3 = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=2, verbose=1)
callback4 = tf.keras.callbacks.EarlyStopping(monitor='val_accuracy', patience=2, verbose=1)
```

```
epochs=10
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs,
    shuffle=True,
    callbacks=[callback1, callback2, callback3, callback4]
    # callbacks=[callback1,]
)
```

```
Epoch 1/10
658/658 [==============================] - 96s 24ms/step - loss: 0.7286 - accuracy: 0.6351 - val_loss: 0.5389 - val_accuracy: 0.7322
Epoch 2/10
658/658 [==============================] - 5s 8ms/step - loss: 0.5019 - accuracy: 0.7546 - val_loss: 0.4284 - val_accuracy: 0.8033
Epoch 3/10
658/658 [==============================] - 6s 9ms/step - loss: 0.3580 - accuracy: 0.8463 - val_loss: 0.3696 - val_accuracy: 0.8263
Epoch 4/10
658/658 [==============================] - 5s 8ms/step - loss: 0.2273 - accuracy: 0.9112 - val_loss: 0.2126 - val_accuracy: 0.9145
Epoch 5/10
658/658 [==============================] - 5s 8ms/step - loss: 0.1455 - accuracy: 0.9504 - val_loss: 0.1360 - val_accuracy: 0.9538
Epoch 6/10
658/658 [==============================] - 5s 8ms/step - loss: 0.0966 - accuracy: 0.9681 - val_loss: 0.1035 - val_accuracy: 0.9615
Epoch 7/10
658/658 [==============================] - 5s 8ms/step - loss: 0.0767 - accuracy: 0.9741 - val_loss: 0.0886 - val_accuracy: 0.9683
Epoch 8/10
658/658 [==============================] - 5s 8ms/step - loss: 0.0614 - accuracy: 0.9797 - val_loss: 0.0824 - val_accuracy: 0.9735
Epoch 9/10
658/658 [==============================] - 6s 9ms/step - loss: 0.0440 - accuracy: 0.9872 - val_loss: 0.0933 - val_accuracy: 0.9649
Epoch 10/10
658/658 [==============================] - 6s 9ms/step - loss: 0.0547 - accuracy: 0.9808 - val_loss: 0.0592 - val_accuracy: 0.9778
Epoch 10: early stopping
Epoch 10: early stopping
```

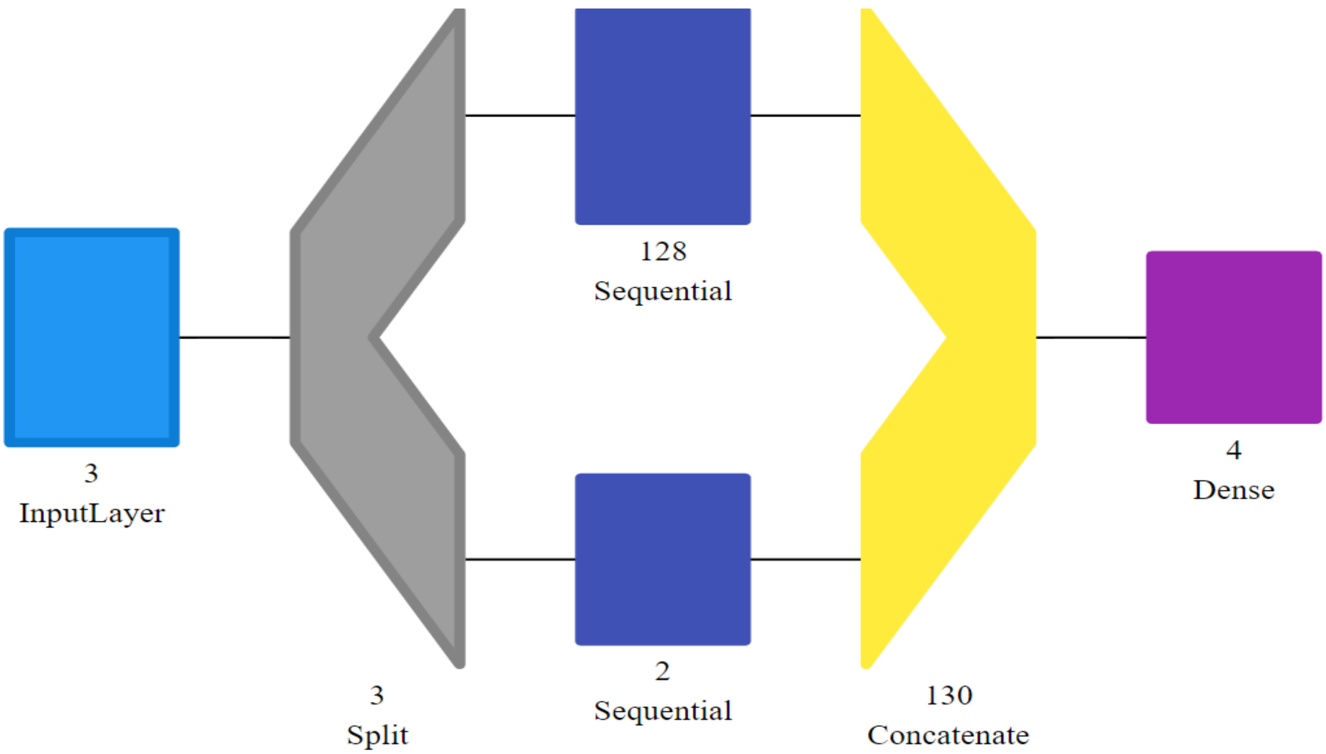


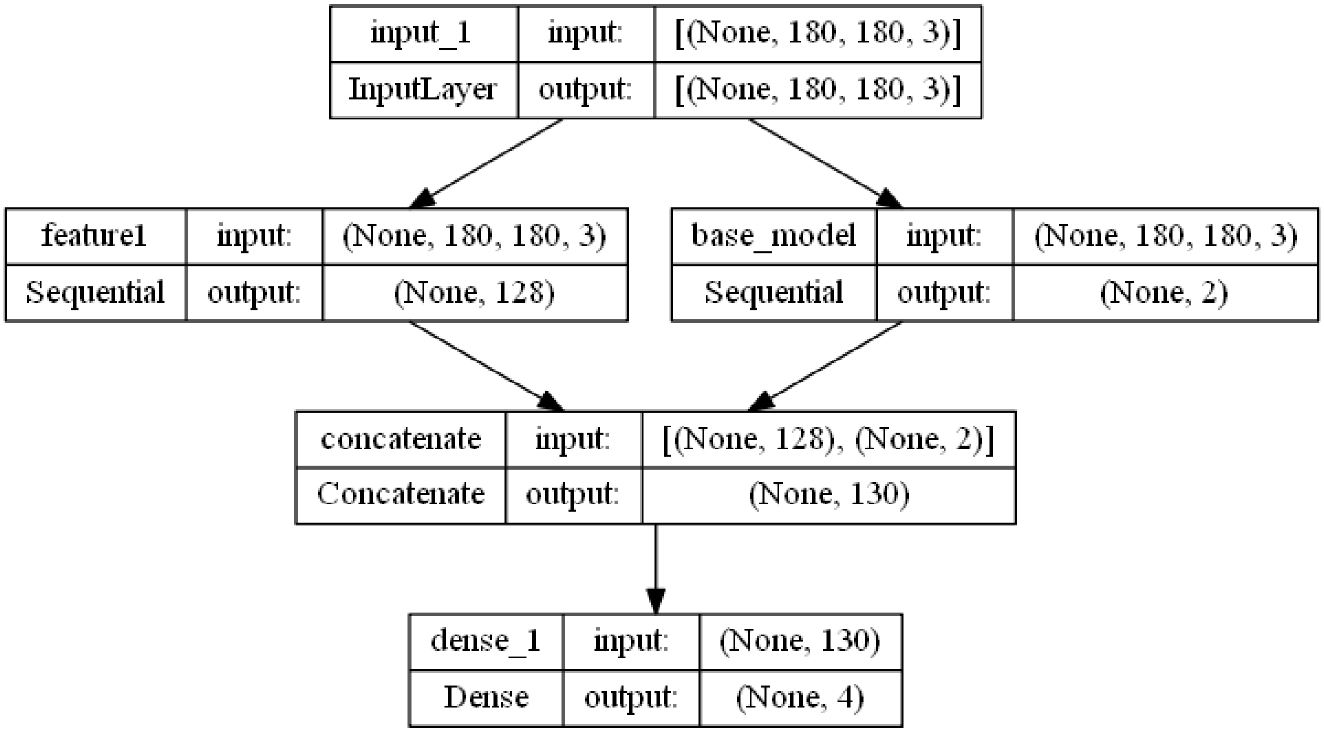


Alzheimer’s Disease Stages Detection Model (Final Model)

In this we use **CNN-SVM hybrid model** and train in parallel to the **freezed base CNN-SVM model** to detect the stages of Alzheimer’s Disease which the patient has in early stages of diagnosis. This is then concatenated and passed on to a softmax output layer. This model is trained on the **Dataset_AD_stages** dataset, mentioned in the previous section.

Architecture of the Model -





The architecture of **feature1 Sequential layer** is the **same as the base model without the softmax output layer**.

model.summary() →

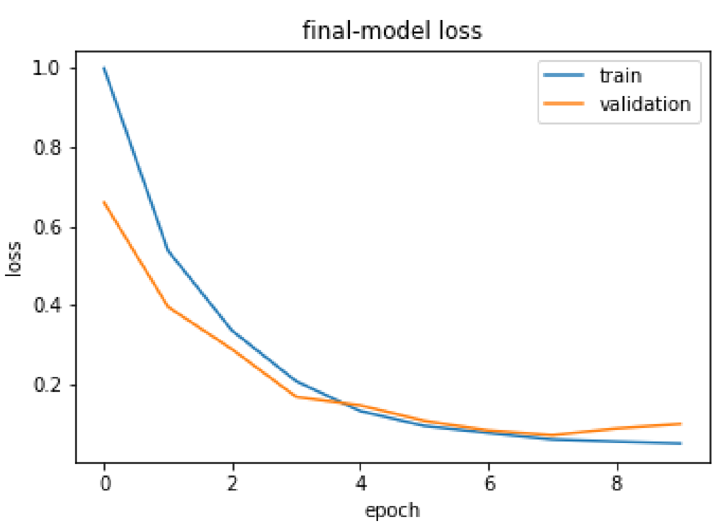
```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 180, 180, 3  0           []
                                )]

 feature1 (Sequential)          (None, 128)          48416       ['input_1[0][0]']

 base_model (Sequential)        (None, 2)            48674       ['input_1[0][0]']

 concatenate (Concatenate)      (None, 130)          0           ['feature1[0][0]',
                                                                  'base_model[0][0]']

 dense_1 (Dense)                (None, 4)            524         ['concatenate[0][0]']

==================================================================================================
Total params: 97,614
Trainable params: 48,940
Non-trainable params: 48,674
__________________________________________________________________________________________________
```

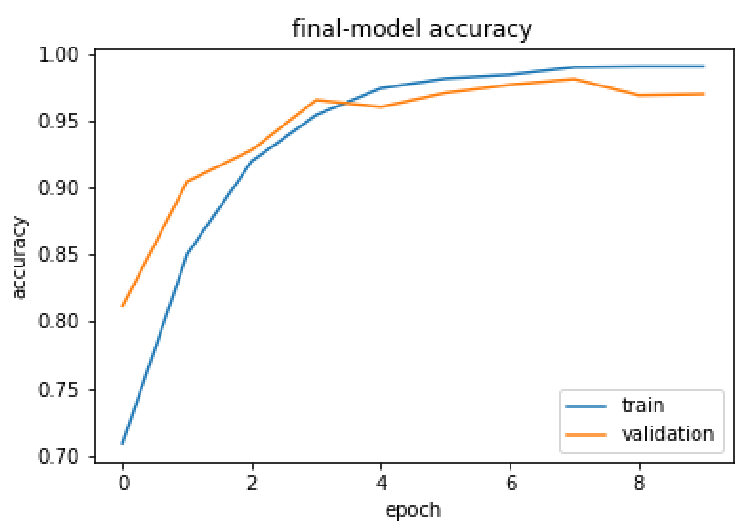
Model training →

```
callback1 = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=1, verbose=1)
callback2 = tf.keras.callbacks.EarlyStopping(monitor='accuracy', patience=1, verbose=1)
callback3 = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=2, verbose=1)
callback4 = tf.keras.callbacks.EarlyStopping(monitor='val_accuracy', patience=2, verbose=1)
```

```
epochs=10
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs,
    shuffle=True,
    callbacks=[callback1, callback2, callback3, callback4]
    # callbacks=[callback1,]
)
```

```
Epoch 1/10
649/649 [==============================] - 59s 30ms/step - loss: 0.9989 - accuracy: 0.7095 - val_loss: 0.6599 - val_accuracy: 0.8116
Epoch 2/10
649/649 [==============================] - 11s 17ms/step - loss: 0.5378 - accuracy: 0.8503 - val_loss: 0.3960 - val_accuracy: 0.9045
Epoch 3/10
649/649 [==============================] - 11s 17ms/step - loss: 0.3357 - accuracy: 0.9197 - val_loss: 0.2885 - val_accuracy: 0.9280
Epoch 4/10
649/649 [==============================] - 11s 17ms/step - loss: 0.2080 - accuracy: 0.9541 - val_loss: 0.1686 - val_accuracy: 0.9653
Epoch 5/10
649/649 [==============================] - 11s 17ms/step - loss: 0.1325 - accuracy: 0.9740 - val_loss: 0.1472 - val_accuracy: 0.9601
Epoch 6/10
649/649 [==============================] - 12s 18ms/step - loss: 0.0950 - accuracy: 0.9812 - val_loss: 0.1077 - val_accuracy: 0.9705
Epoch 7/10
649/649 [==============================] - 11s 17ms/step - loss: 0.0770 - accuracy: 0.9841 - val_loss: 0.0835 - val_accuracy: 0.9766
Epoch 8/10
649/649 [==============================] - 11s 17ms/step - loss: 0.0609 - accuracy: 0.9898 - val_loss: 0.0714 - val_accuracy: 0.9809
Epoch 9/10
649/649 [==============================] - 12s 18ms/step - loss: 0.0553 - accuracy: 0.9904 - val_loss: 0.0884 - val_accuracy: 0.9688
Epoch 10/10
649/649 [==============================] - 11s 17ms/step - loss: 0.0507 - accuracy: 0.9904 - val_loss: 0.1001 - val_accuracy: 0.9696
Epoch 10: early stopping
Epoch 10: early stopping
Epoch 10: early stopping
```

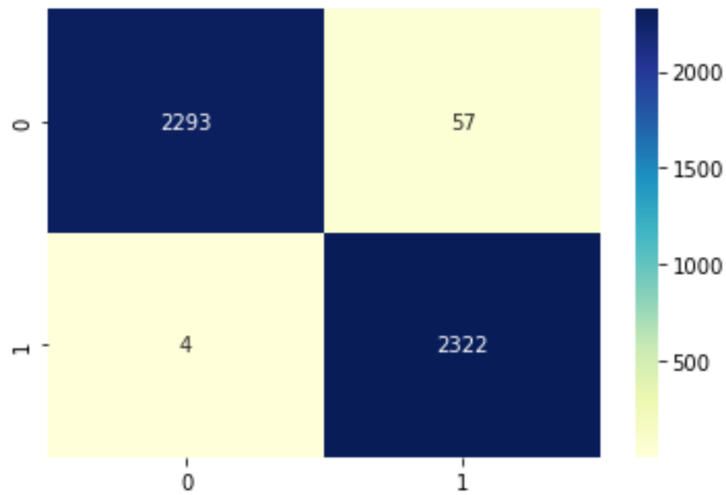




Performance Metrics to determine the models performance

Base Model →

- Confusion matrix -



0 → Negative

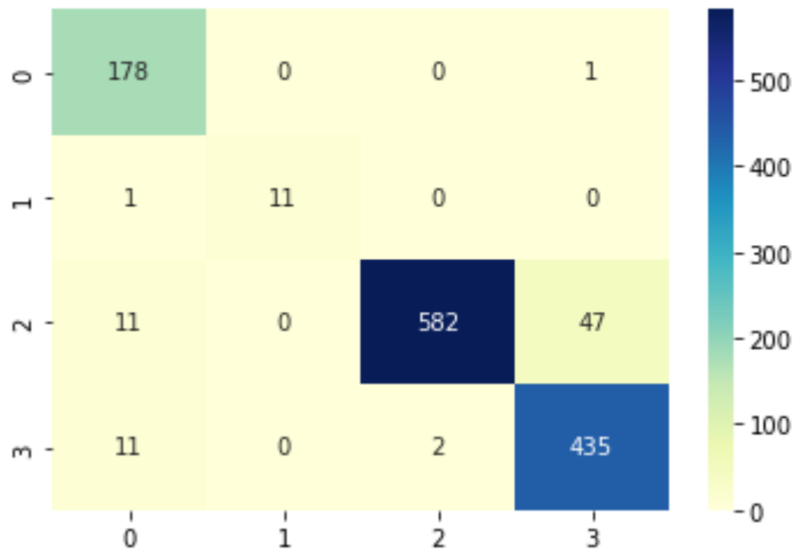
1 → Positive

- Classification Report -

	precision	recall	f1-score	support
0	1.00	0.98	0.99	2350
1	0.98	1.00	0.99	2326
accuracy			0.99	4676
macro avg	0.99	0.99	0.99	4676
weighted avg	0.99	0.99	0.99	4676

Final Model →

- Confusion matrix -



0 → Mild Demented

1 → Moderate Demented

2 → Non Demented

3 → Very Mild Demented

- Classification Report -

	precision	recall	f1-score	support
0	0.89	0.99	0.94	179
1	1.00	0.92	0.96	12
2	1.00	0.91	0.95	640
3	0.90	0.97	0.93	448
accuracy			0.94	1279
macro avg	0.95	0.95	0.94	1279
weighted avg	0.95	0.94	0.94	1279

References

[1] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, “Early diagnosis of Alzheimer’s disease with deep learning,” 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI). IEEE, Apr-2014.

[2] J. Islam and Y. Zhang, “A Novel Deep Learning Based Multi-class Classification Method for Alzheimer’s Disease Detection Using Brain MRI Data,” Brain Informatics. Springer International Publishing, pp. 213–222, 2017.

[3] R. Wolz, V. Julkunen, J. Koikkalainen, E. Niskanen, D. P. Zhang, D. Rueckert, H. Soininen, and J. Lötjönen, “Multi-Method Analysis of MRI Images in Early Diagnostics of Alzheimer’s Disease,” PLoS ONE, vol. 6, no. 10. Public Library of Science (PLoS), p. e25446, 13-Oct-2011.