Deep Learning Course Project- Gesture Recognition

-Suresh Babu R

-Anugata Mahapatra

-Vibhu Goel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Experiment**  **Number** | **Model** | **Result (Best Epoch)** | **Decision + Explanation** | **Parameters** |
| **0\_0** | **Conv3D** | **Train Accuracy: 0.6783**  **Validation Accuracy: 0.6000** | **Using image/255 to normalize**  **# of frames kept low and reduced image size** | **4,820,133** |
| **0** | **Conv3D** | **Train AUC: 0.9327**  **Validation AUC: 0.9041** | **Normalized image using 95th and 5th percentile to remove the noise which would improve the accuracy and reduce the overfitting.** | **7,187,621** |
| **1** | **Conv3D** | **Train AUC: 0.8857**  **Validation AUC: 0.8672** | **Adding augmentation by cropping the images to get a more targeted gesture and it will also help in removing noise this would further increase the stability** | **7,187,621** |
| **2** | **Conv3D** | **Train AUC: 0.9000**  **Validation AUC: 0.7756** | **Reducing the parameters to optimize the execution time while increasing the accuracy by reducing the image size.** | **4,902,053** |
| **3** | **Conv3D** | **Train AUC: 0.7768**  **Validation AUC: 0.7753** | **Since we see overfitting, we again increase back the image size to 145 and also increase the # of frames and add one more layer to reduce the parameters and further increase the accuracy** | **3,452,709** |
| **4** | **Conv3D** | **Train AUC: 0.7720**  **Validation AUC: 0.7666** | **Further reduce the parameters by reducing back the # of frames to 18 to drop the execution time while maintaining the accuracy** | **1,109,285** |
| **5** | **Conv3D** | **Train AUC: 0.8327**  **Validation AUC: 0.8225** | **Increasing the # of parameters as we saw that higher the parameters higher the accuracy** | **21,400,869** |
| **6** | **Conv2D+LSTM** | **Train AUC:** **0.8319**  **Validation AUC: 0.8066** | **Accuracy doesn’t increase much but we do some overfitting happening** | **10,724,069** |
| **7** | **Transfer Learning MobileNet + LSTM** | **Train AUC: 0.9868**  **Validation AUC: 0.9775** | **We are using a pre-trained Mobilenet model along with LSTM. The Mobilenet weights are not being trained. We saw exceptional improvement in accuracy with hardly any overfitting** | **3,840,453** |
| **8(Final Model)** | **Transfer Learning MobileNet + GRU** | **Train AUC: 0.9998**  **Validation AUC: 0.9797** | **We are using a pre-trained Mobilenet model along with GRU. The Mobilenet weights are not being trained. We saw exceptional high accuracy with hardly any overfitting.** | **3,693,253** |

**A glimpse into the various models that were tried out:**

Model 0: Conv3D(16) > Conv3D(32) > Conv3D(64) > Flatten() > Dense(128) > Dense(64) > Dense(5)

This model has 7 layers. This model has some 7,187,621 trainable parameters. An image size of 145\*145 was used with the no of frames set at 15. A batch size of 65 was maintained throughout the entire of the experiments, as no changes were observed over the experiments with change in batch size. Dropout of .25 was used in the penultimate Dense layer and the Denser layer prior to this layer. A filter of size 5\*5\*5 has been used in the first layer, and filters of size 3\*3\*3 in layers 2 and 3. This model is set to do 20 epochs. The highest clocked training accuracy is 0.94 and validation accuracy for the same is 0.9. The model exhibited a good fit.

Model 1: Conv3D(16) > Conv3D(32) > Conv3D(64) > Flatten() > Dense(128) > Dense(64) > Dense(5)

This model has 7 layers. The same model as model 0 is used which has some 7,187,621 trainable parameters. An image size of 145\*145 was used with the no of frames set at 18. The rest of the parameters remain the same. This model is set to do 20 epochs. The highest clocked training accuracy is 0.94 and validation accuracy for the same is 0.96. This model too exhibited a good fit.

Model 2: Conv3D(16) > Conv3D(32) > Conv3D(64) > Flatten() > Dense(128) > Dense(64) > Dense(5)

This model too has 7 layers. The experiment continues without any change in the layers or the parameters. The image size is brought down to 124\*124 with the no of frames set at 18. This model has some 4,901,445 trainable parameters. This model is set to do 22 epochs. The highest clocked training accuracy is 0.95 and validation accuracy for the same is 0.79. This model exhibited overfitting.

Model 3: Conv3D(16) > Conv3D(32) > Conv3D(64) > Conv3D(128) > Flatten() > Dense(128) > Dense(64) > Dense(5)

This model has 8 layers. An additional layer was introduced to verify model performance. The image size was set at 145\*145. The no of frames is set at 22. The rest of the parameters remained unchanged. This model has some 3,451,845 trainable parameters. This model is set to do 26 epochs. The highest clocked training accuracy is 0.93 and validation accuracy for the same is 0.84. This model exhibits a drop in accuracy in comparison to the previous models though an additional layer was introduced. Thus, the additional layer is of no considerable significance. In fact, the accuracy has dropped considerably in comparison to model 0 and model 1.

Model 4: Conv3D(16) > Conv3D(32) > Conv3D(64) > Conv3D(128) > Flatten() > Dense(128) > Dense(64) > Dense(5)

This model had 8 layers. This is the same as model 3. The image size is now reduced to 140\*140. The no of frames is set at 18. The rest of the parameters remained unchanged. This model has some 1,108,677 trainable parameters. This model is set to do 30 epochs. The highest clocked training accuracy is 0.84 and validation accuracy for the same is 0.80. The accuracy has dropped significantly though no overfit is observed. The additional layer is of no significance and is contributing to loss of accuracy on the contrary.

Model 5: Conv3D(16) > Conv3D(32) > Conv3D(64) > Flatten() > Dense(128) > Dense(128) > Dense(5)

Reverting back to the 7 layered model as model 0, 1, and 2. The penultimate Dense layer is set to 128 neurons. The image size is set at 145\*145. The no of frames is set to 30. This model has some 21,400,389 trainable parameters. This model is set to do 30 epochs. The highest clocked training accuracy is 0.9 and validation accuracy for the same is 0.9. This model has improved accuracy and exhibits best fit.

Model 6: Conv2D(16) > Conv2D(32) > Conv2D(64) > Flatten() > LSTM(128) > Dense(128) > Dense(5)

Changing to Conv2D + LSTM architecture. The model has some 7 layers. The trainable parameters are 10,724,069. The penultimate Dense layer has 128 neurons. The image size is set to 145\*145. The number of frames is 30. This model is set to do 32 epochs. The highest clocked training accuracy is 0.88 and validation accuracy for the same is 0.8. Accuracy is well below Conv3D architecture based models.

Model 7: Mobilenet > Flatten() > LSTM(128) > Dense(128) > Dense(5)

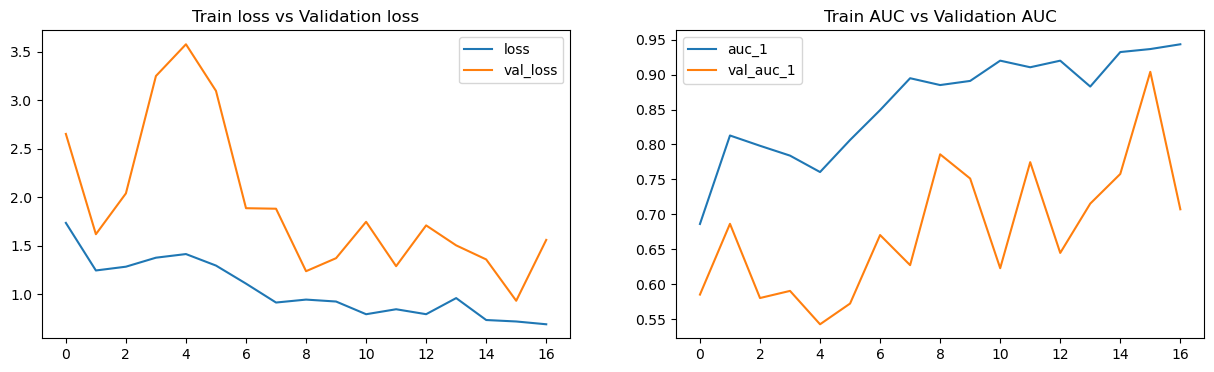
Introducing transfer learning to observe change in model performance when in comparison to building a model from the scratch. Transfer learning of Mobilenet is used along with LSTM. This model has some 3,840,453. The image size is reduced to 124\*124. The no of frames is set to 19. The penultimate Dense layer contains 128 neurons. This model is set to do 27 epochs. The highest clocked training accuracy is 0.99 and validation accuracy for the same is 0.97. The model exhibits superior performance.

Model 8: Mobilenet > Flatten() > GRU(128) > Dense(128) > Dense(5)

Transfer learning continues to be in use in this model as well. This time around transfer learning of Mobilenet is used along with GRU. This model has some 3,693,253; a considerably lesser number of trainable parameters in comparison with the previous model using LSTM. The image size is maintained at 124\*124 as before. The no of frames is set to 19 itself. The penultimate Dense layer contains 128 neurons, same as the previous model. This model is set to do 23 epochs. The highest clocked training accuracy is 0.99 and validation accuracy for the same is 0.97. This model is able to achieve a similar accuracy as the model before, which is on LSTM, with lesser trainable parameters, thus using lesser resources.

**Plotting the loss and AUC metric; both for training and validation, with increasing epochs, for the various models:**

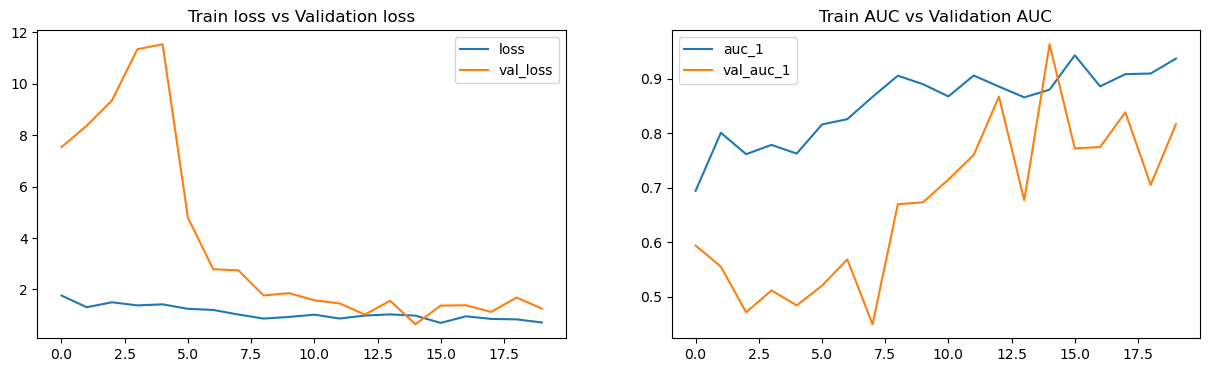
Model 0:



Max. Training Accuracy 0.9435302019119263

Max. Validation Accuracy 0.9040625095367432

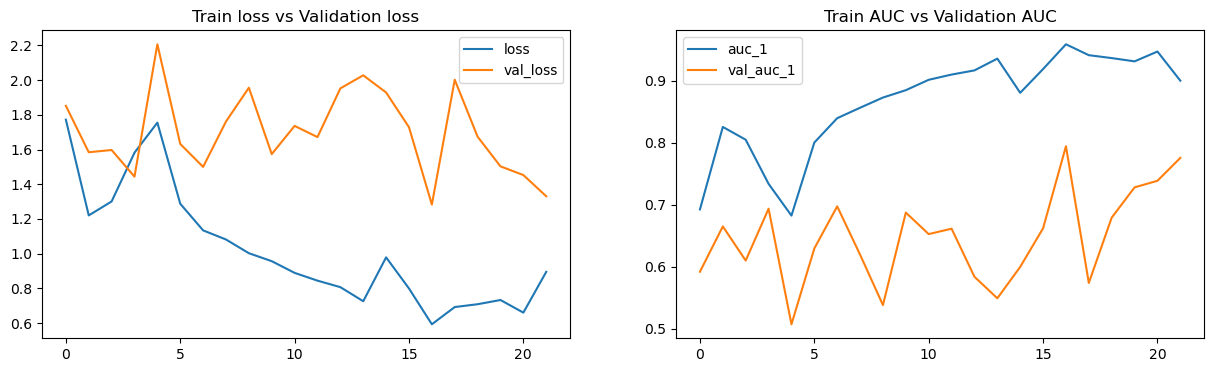
Model 1:



Max. Training Accuracy 0.9432124495506287

Max. Validation Accuracy 0.9634374976158142

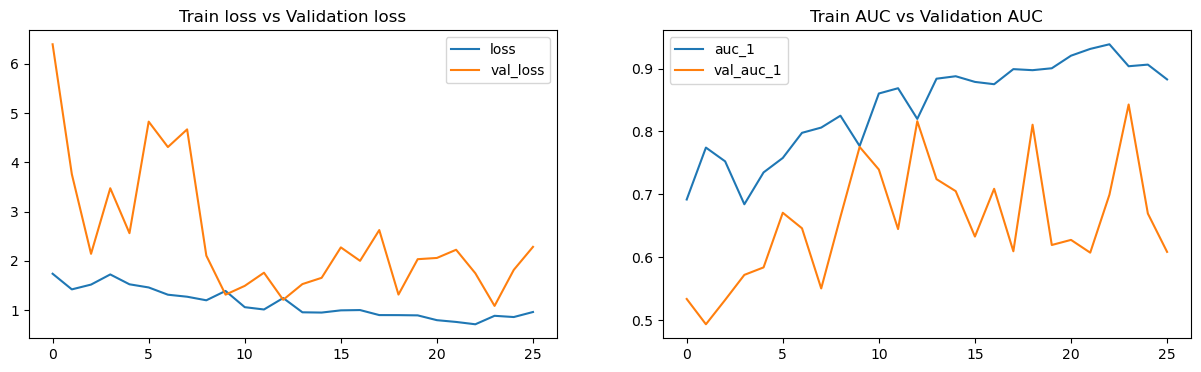
Model 2:



Max. Training Accuracy 0.9587082862854004

Max. Validation Accuracy 0.7943750023841858

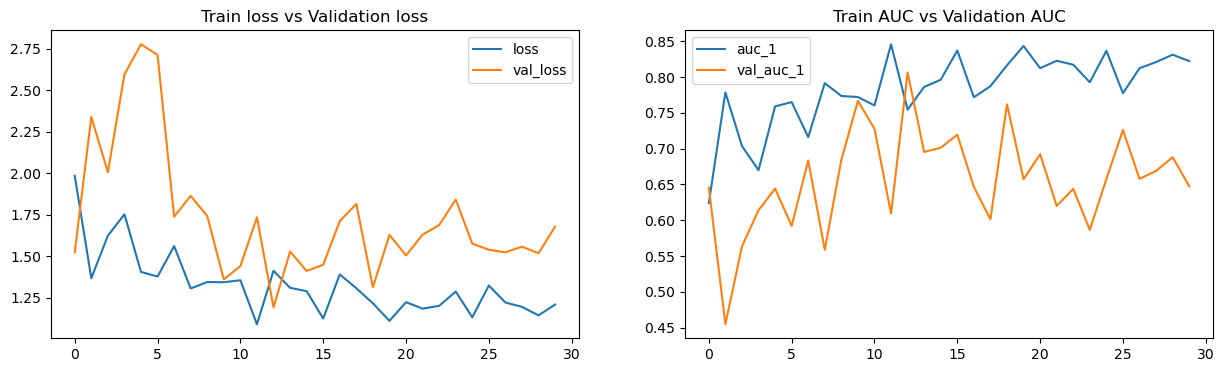
Model 3:



Max. Training Accuracy 0.9385483264923096

Max. Validation Accuracy 0.8428124785423279

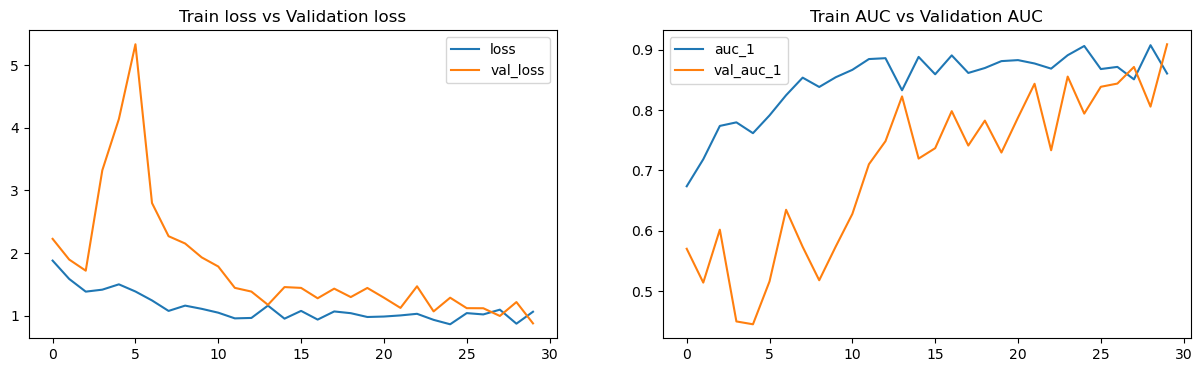
Model 4:



Max. Training Accuracy 0.8456403613090515

Max. Validation Accuracy 0.8062499761581421

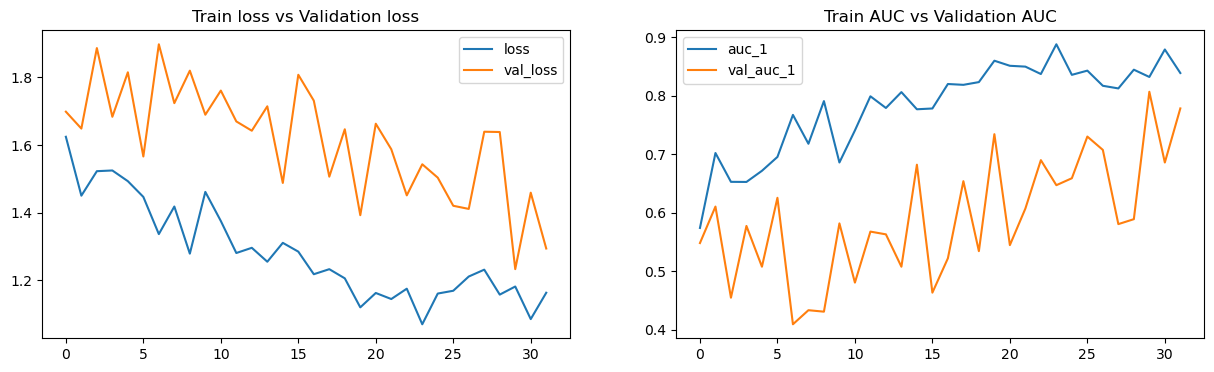
Model 5:



Max. Training Accuracy 0.907244861125946

Max. Validation Accuracy 0.9087499976158142

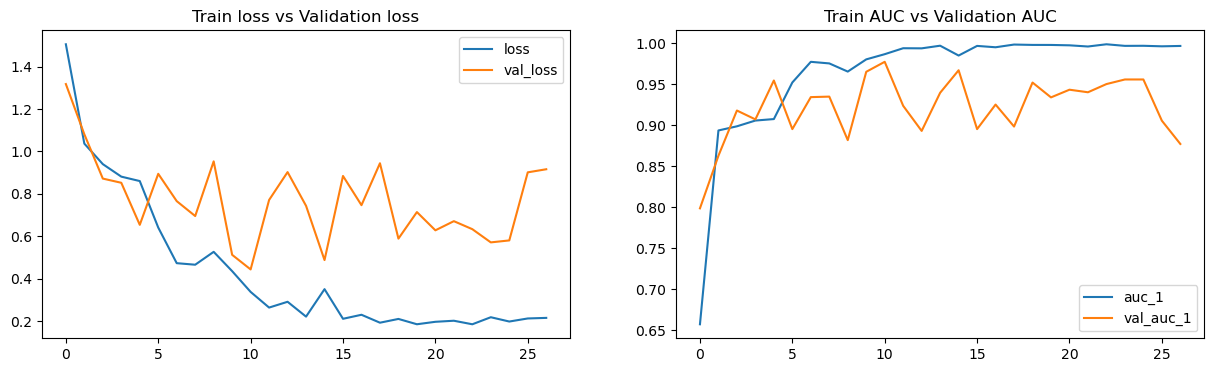
Model 6:



Max. Training Accuracy 0.887775719165802

Max. Validation Accuracy 0.8065624833106995

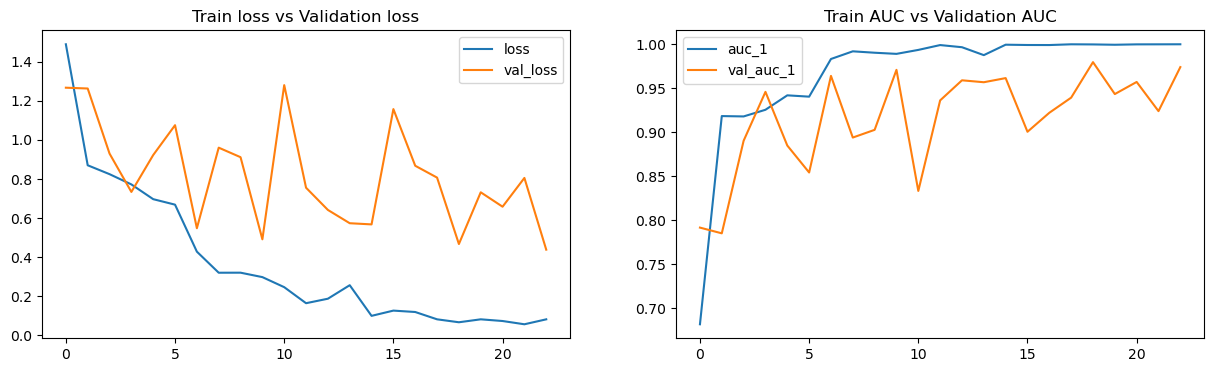
Model 7:



Max. Training Accuracy 0.9989118576049805

Max. Validation Accuracy 0.9775000214576721

Model 8:



Max. Training Accuracy 0.9999756217002869

Max. Validation Accuracy 0.979687511920929

**Different Models and the respective parameters that were used:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Model Architecture** | **No of Trainable Parameters** | **Layers** | **No of Layers** | **Image Size** | **No of Frames** | **Batch Size** | **No of neurons in penultimate Dense layer** | **No of Epochs** | **Dropout** | **Max Training Accuracy Achieved** | **Max Validation Accuracy Achieved** |
| 0 | Conv3D | 7,187,621 | Conv3D(16) > Conv3D(32) > Conv3D(64) > Flatten() > Dense(128) > Dense(64) > Dense(5) | 7 | 145\*145 | 15 | 65 | 64 | 20 | 0.25 | 0.94 | 0.9 |
| 1 | Conv3D | 7,187,621 | Conv3D(16) > Conv3D(32) > Conv3D(64) > Flatten() > Dense(128) > Dense(64) > Dense(5) | 7 | 145\*145 | 18 | 65 | 64 | 20 | 0.25 | 0.94 | 0.96 |
| 2 | Conv3D | 4,901,445 | Conv3D(16) > Conv3D(32) > Conv3D(64) > Flatten() > Dense(128) > Dense(64) > Dense(5) | 7 | 124\*124 | 18 | 65 | 64 | 22 | 0.25 | 0.95 | 0.79 |
| 3 | Conv3D | 3,451,845 | Conv3D(16) > Conv3D(32) > Conv3D(64) > Conv3D(128) > Flatten() > Dense(128) > Dense(64) > Dense(5) | 8 | 145\*145 | 22 | 65 | 64 | 26 | 0.25 | 0.93 | 0.84 |
| 4 | Conv3D | 1,108,677 | Conv3D(16) > Conv3D(32) > Conv3D(64) > Conv3D(128) > Flatten() > Dense(128) > Dense(64) > Dense(5) | 8 | 140\*140 | 18 | 65 | 64 | 30 | 0.25 | 0.84 | 0.8 |
| 5 | Conv3D | 21,400,389 | Conv3D(16) > Conv3D(32) > Conv3D(64) > Flatten() > Dense(128) > Dense(128) > Dense(5) | 7 | 145\*145 | 30 | 65 | 128 | 30 | 0.25 | 0.9 | 0.9 |
| 6 | Conv2D + LSTM | 10,724,069 | Conv2D(16) > Conv2D(32) > Conv2D(64) > Flatten() > LSTM(128) > Dense(128) > Dense(5) | 7 | 145\*145 | 30 | 65 | 128 | 32 | 0.25 | 0.88 | 0.8 |
| 7 | Transfer learning using MobileNet with LSTM | 3,840,453 | Mobilenet > Flatten() > LSTM(128) > Dense(128) > Dense(5) | 5 | 124\*124 | 19 | 65 | 128 | 27 | 0.25 | 0.99 | 0.97 |
| 8 | Transfer learning using MobileNet with GRU | 3,693,253 | Mobilenet > Flatten() > GRU(128) > Dense(128) > Dense(5) | 5 | 124\*124 | 19 | 65 | 128 | 23 | 0.25 | 0.99 | 0.97 |