CNN Image Classification Project

MATH 6373 - Deep Learning and Neural Networks

Ravik Chand

Introduction

This project is an exercise in the construction and optimization of a Convolution Neural Network deployed in a KerasTensorflow environment with the objective of automatically classifying images. For this project we used the OrganAMNIST dataset from the MedMNIST v2 data collection, hosted on Zenodo.org.

https://medmnist.com/
https://zenodo.org/record/6496656

This dataset contains 58,850 CT scanned images of the abdominal region. Initially 3D, the images were converted to single-channel grayscale and cropped into 2D, 28x28 pixel slices. Specifically, the OrganA set contains images fixed from an axial perspective.

Preparation

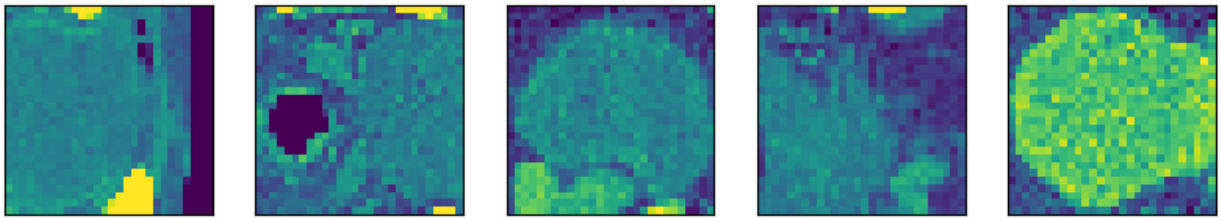
The data was downloaded as an .npz file and opened in python using Numpy. The intensity of each pixel of a given image was stored as a value on a scale from 0 to 255 in a 28x28 matrix. The matrices of all 58,850 images were stored in three arrays: 34,581 for training, 6,491 for validation, and 17,778 for testing. These arrays were combined to form the **variables** array. An additional three arrays of equal lengths contained single value labels ranging from 0 through 10 representing the eleven distinct classes that comprised the dataset. These were combined into the **labels** array. The number of occurrences of each class in the dataset was determined.

CLASS	COUNT
0	3313
1	2425
2	2377
3	2651
4	6595
5	6419
6	10482
7	6699
8	6751
9	5182
10	5956

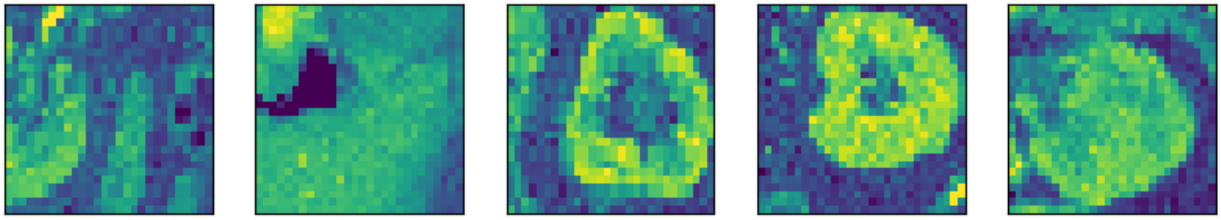
The matrices in the **variables** array were flattened to form variable vectors, each of length 784. The values from the **labels** array were then attached to the vectors of the **flat_variables** array based on index to associate variables with labels. With the exception of classes 1, 2, and 3 which contained the fewest observations, the remaining classes were isolated into eight distinct class arrays. To prevent issues which would arise from indexing, these classes were relabeled as follows:

CLASS	ARRAY	RELABEL
0	**cl_0**	0
4	**cl_1**	1
5	**cl_2**	2
6	**cl_3**	3
7	**cl_4**	4
8	**cl_5**	5
9	**cl_6**	6
10	**cl_7**	7

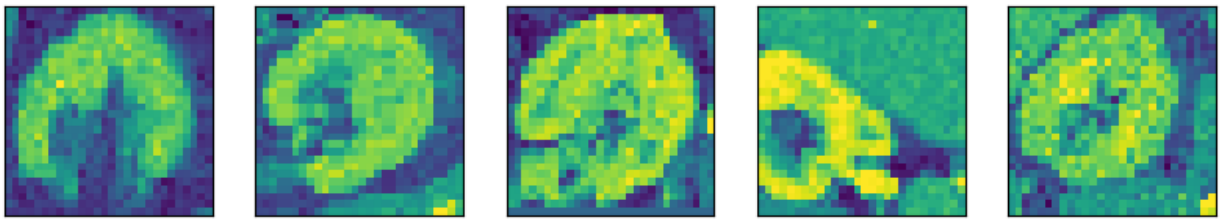
Of the larger classes, **cl_0** was the smallest and established the base size for **cl_1**, **cl_2**, **cl_3**, **cl_4**, **cl_5**, **cl_6**, and **cl_7**. 3313 observations were randomly sampled from each of the classes to create eight reduced class arrays of equal length.



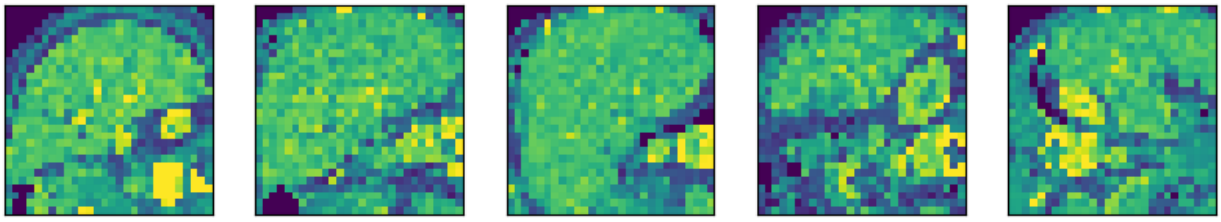
cl_0



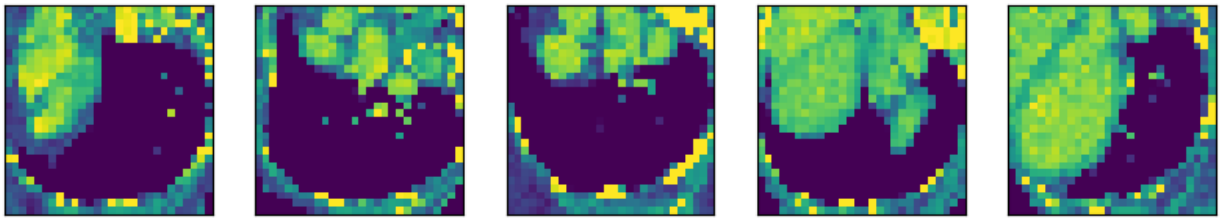
cl_1



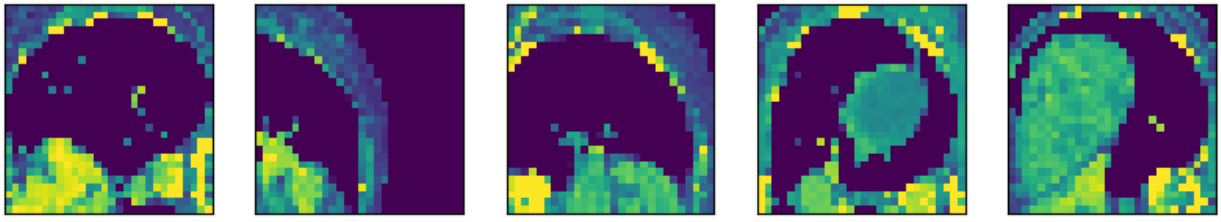
cl_2



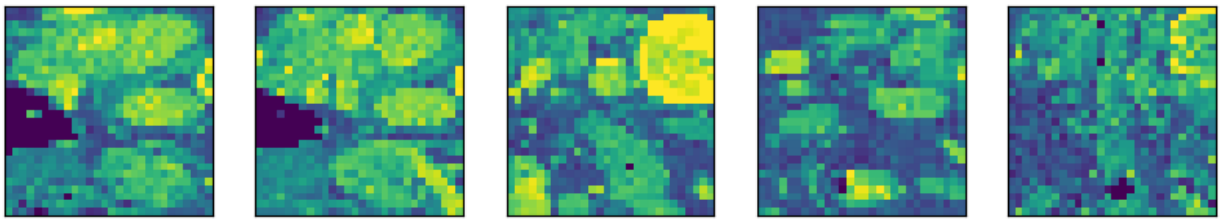
cl_3



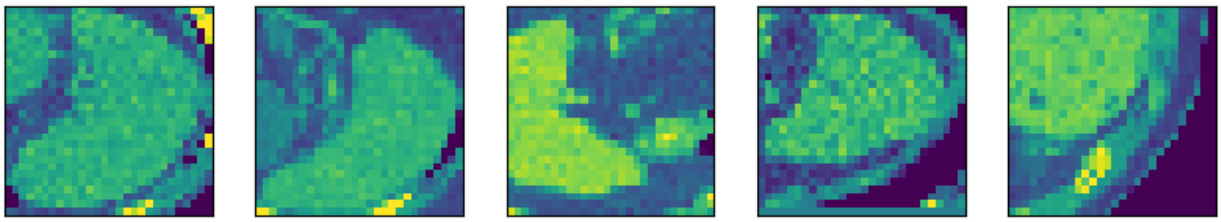
cl_4



cl_5



cl_6



cl_7

The arrays **red_cl_0** through **red_cl_7** were then randomly separated using *train_test_split()* function to create their respective trainval and test set arrays for each class with 10% allocation to the test sets. The trainval arrays were split again to create the training and validation sets with 5% allocation to the validation sets. The individual class arrays **cl_0_train** through **cl_7_train** were combined to form the array **train**. Likewise, the class arrays for validation and test sets were combined to form the arrays **val** and **test** respectively. Labels were then transferred from the **train**, **val** and **test** arrays to the **y_train**, **y_val** and **y_test** arrays. Through application of one-hot encoding using the *to_categorical()* function, the individual observation values in the y arrays were expanded into vectors of length 8. The variables from the **train**, **val** and **test** arrays were deposited into the arrays **x_train_flat**, **x_val_flat** and **x_test_flat**. These were then reshaped back into 28x28 matrices in the **x_train**, **x_val** and **x_test** arrays.

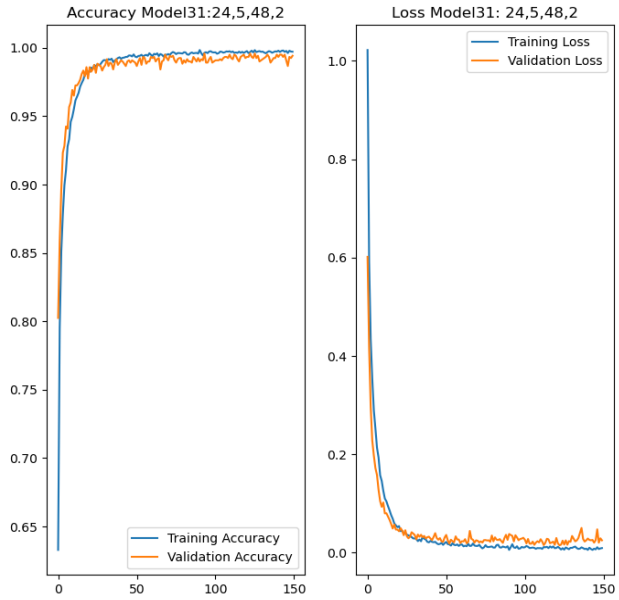
Model

A model class CNN was established with a sequential structure. Its architecture consisted of an input layer followed immediately by a 2D convolution layer. Channel sizes for the input layers varied between 8, 16, and 24 which determined the number of features that could be recognized from the input variables. Filter sizes were also varied between 3, 5, and 7 which determined the focus of the channels on global or local features. The strides were fixed at 1, and padding was not present. The next layer consisted of a maxpool layer which was used to downsample the output of the previous layer. The window size of the maxpool layer was fixed at 2x2. Another convolution layer followed, with channel size either equal to or double that of the previous convolution layer, and window size of either 2 or 3. This was followed by another maxpool layer identical to the previous maxpool layer. A dropout layer randomly deactivated neurons as a means to mitigate overfitting by reducing likelihood of overreliance on specific feature combinations. The amount of neurons dropped at every update was fixed to 50%. At this point a flatten layer was used to convert the matrix output of the preceding convolutions into a vector which would be passed into a hidden layer of dimension *h*. The value of *h* was defined as half the length of the flatten layer's output vector and adjusted per model. Finally, an output layer was created that would return a vector of equal dimension to the y arrays and pass through a softmax function before evaluation.

Thirty-six models were run with variations in hyper-parameter combinations. The ‘adam’ optimizer was selected and categorical cross entropy was used to define loss. Accuracy was the target metric. Epochs were set to their minimum allowance for this project, 150. The batch size was set to the average of the minimum allowance of 80 and the square-root of the length of the train data, which ultimately amounted to 115. Total compute time for the thirty-six models amounted to about 41 minutes, with compute times of individual models ranging from 63.48 to 75.36 seconds.

Parsimony ratios, defined as the product of the count of observations in the training data and the length of each one-hot encoded class label, divided by the total count of parameters in a model, were calculated for each model. The average parsimony ratio was 1.06, with a median of 0.83. All models exhibited high accuracy with a minimum validation accuracy of 96.17%.

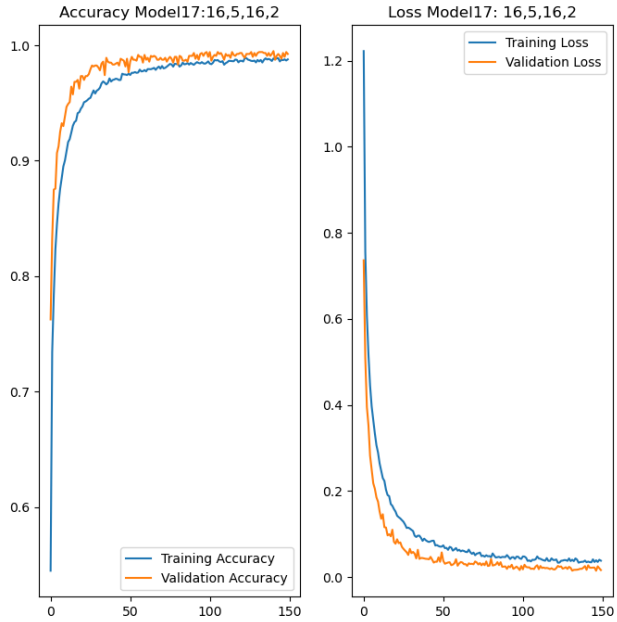
The highest accuracy was attributed to model_31, which was configured with a channel size of 24 and a window size of 5 for the first convolution layer, and a channel size of 48 with a window size of 2 for its second convolution layer. It demonstrated an accuracy of 99.42% against validation data, with only 2.51% loss. The model was composed of 479,216 parameters, but expressed a low parsimony ratio of 0.38.



Model_31 train/val accuracy & loss

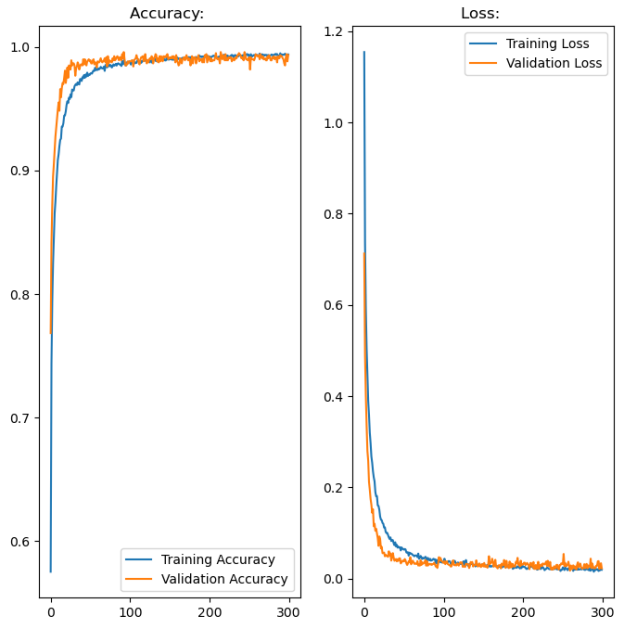
Model_17 exhibited a lower accuracy of 99.25%, and loss of 1.65%. However, this model also demonstrated a considerably higher parsimony ratio of 1.12. Because of the combination of high accuracy, low loss, and reasonable parsimony ratio, the

configuration of model_17 was selected as the optimal model for further analysis. The dimension of model_17's flatten layer was found to be 400.



Model_17 train/val accuracy & loss

A model_opt was created with identical parameters to model_17 and trained again with the number of epochs extended to 300. This resulted in an accuracy of 98.67%, and a loss of 3.5% against the validation set. Additionally, model_opt was evaluated against the test data for which it returned an accuracy of 98.98%, and loss of 4.2%.



Model_opt train/val accuracy & loss

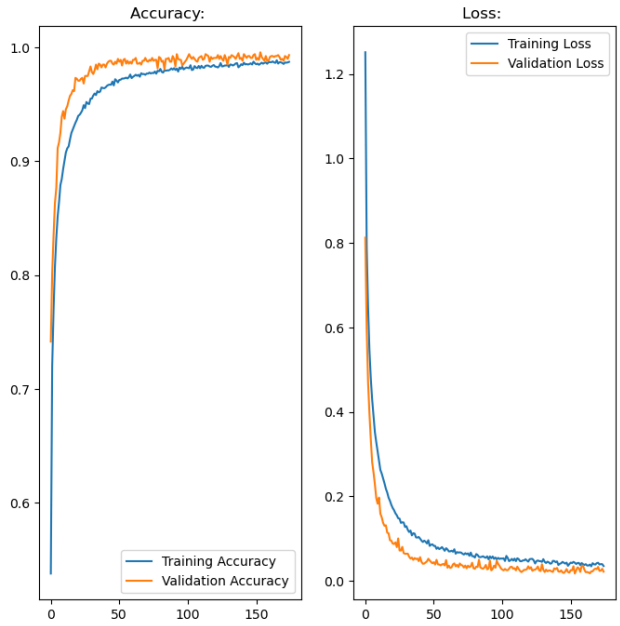
The following confusion matrix generated by model_opt against the test data demonstrates very low confusion with classes 0, 3, 4, and 5. The model appears to converge prior to the 200th epoch.

	cl_0	cl_1	cl_2	cl_3	cl_4	cl_5	cl_6	cl_7
cl_0	99	1	0	0	0	0	0	0
cl_1	0	99	0	0	0	0	0	1
cl_2	0	2	98	0	0	0	0	0
cl_3	0	0	0	99	0	1	0	0
cl_4	0	0	0	0	100	0	0	0
cl_5	0	0	0	0	0	100	0	0
cl_6	0	1	0	0	0	0	98	1
cl_7	0	0	1	0	0	0	0	99

Model_opt test confusion

We extracted the output of the sixth layer of model_opt upon which we performed PCA to discern the number of components necessary to explain 90% of variance. This returned a value of 321, which was used to define the size of the hidden layers for subsequent models. For model_opt2, the hidden layer was set equal to the number of principal components, whereas for model_opt3 the hidden layer was twice the number

of principal components. For model_opt2, the count of epochs was reduced to 175. We observed a slight improvement in validation accuracy to 99.33% and test accuracy to 98.95% relative to model_opt. Likewise, we saw a reduction in validation loss to 2.18%, but no significant change regarding the test loss. There was a slight drop in the parsimony ratio to 0.69. We noted a significant decrease in compute time likely attributable to the reduction of epochs.



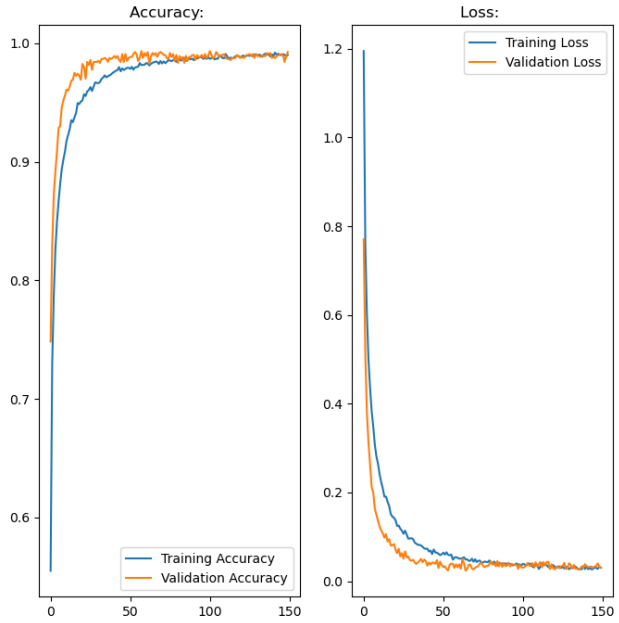
Model_opt2 train/val accuracy & loss

The matrix below produced against test data also shows less confusion between classes.

	cl_0	cl_1	cl_2	cl_3	cl_4	cl_5	cl_6	cl_7
cl_0	100	0	0	0	0	0	0	0
cl_1	0	98	1	0	0	0	0	1
cl_2	1	2	96	0	0	0	0	1
cl_3	0	0	0	100	0	0	0	0
cl_4	0	0	0	0	100	0	0	0
cl_5	0	0	0	0	0	100	0	0
cl_6	0	0	0	0	0	0	99	0
cl_7	0	1	1	0	0	0	0	98

Model_opt2 test confusion

Model_opt3 had its number of epochs reduced further back to 150. Summary details revealed an increase in total parameters 264,042, which is roughly 100,000 more parameters than other iterative models based on the optimal configuration. We observed a slight decrease in accuracy regarding validation and test data, to 98.91% and 98.93% respectively. We observed a significant drop in parsimony ratio, down to 0.35.



Model_opt3 train/val accuracy & loss

As observed earlier, classes 0, 3, 4 and 5 consistently display high accuracy while classes 1, 2, 6 and 7 appear to result in some confusion.

	cl_0	cl_1	cl_2	cl_3	cl_4	cl_5	cl_6	cl_7
cl_0	100	0	0	0	0	0	0	0
cl_1	0	98	1	0	0	0	0	0
cl_2	1	2	96	0	0	0	0	0
cl_3	0	0	0	100	0	0	0	0
cl_4	0	0	0	0	100	0	0	0
cl_5	0	0	0	0	0	100	0	0
cl_6	0	1	0	0	0	0	99	0
cl_7	0	1	1	0	0	0	0	99

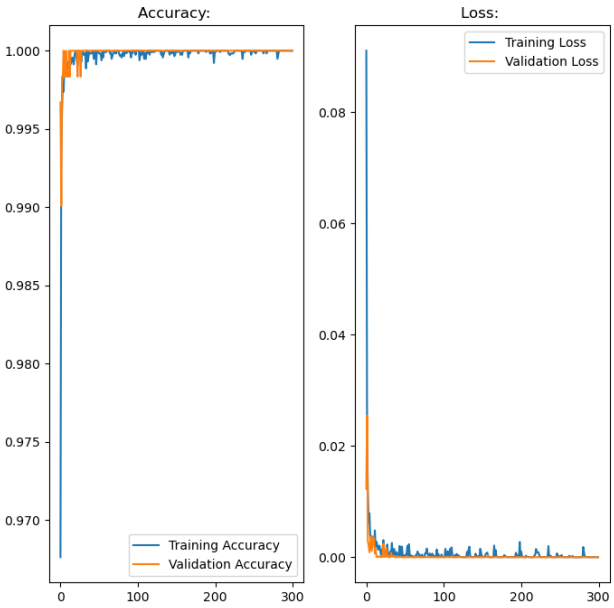
Model_opt3 test confusion

We selected model_opt as the best of the three optimal models. Channel and window sizes, as well as its hidden layer dimension and epochs were used to create a CNN-based binary classifier.

Due to their low confusion, classes 4 and 5 were selected for application towards the binary classifier, model_optb. The previous data preparation steps were taken again, with the exception of only including two of the previous eight classes. Due to the larger sizes of classes 4 and 5, a larger sample size of 6,699 was established from which binary test train, and validation data was derived. Relabeling was performed as before.

CLASS	ARRAY	RELABEL	b_ARRAY	b_RELABEL
7	**cl_4**	4	**b_cl_0**	0
8	**cl_5**	5	**b_cl_1**	1

Validation and test accuracy were recorded at 100%. Simultaneously, 0% loss was reported for both validation and test sets. Parsimony reduced to 0.57.



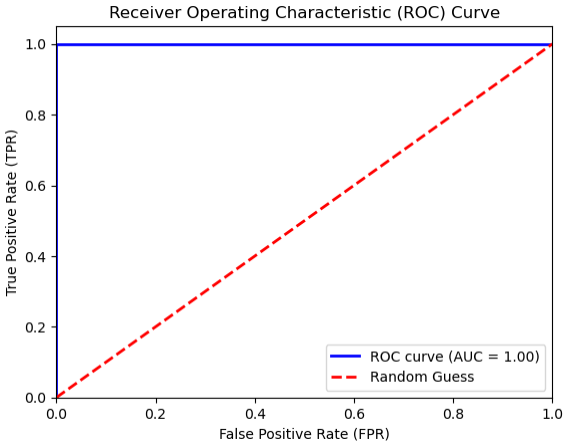
Model_optb train/val accuracy & loss

No confusion was recorded when model_optb was provided with binary test data.

	cl_0	cl_1
cl_0	100	0
cl_1	0	100

Model_optb test confusion

The following ROC Curve was generated based on model_optb:

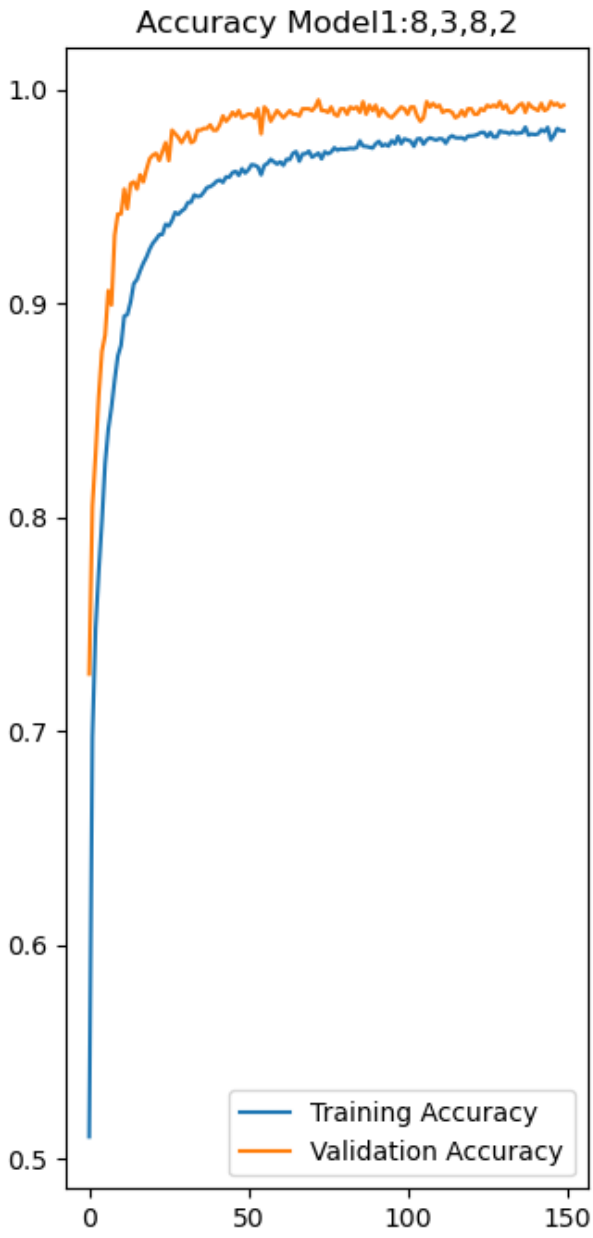


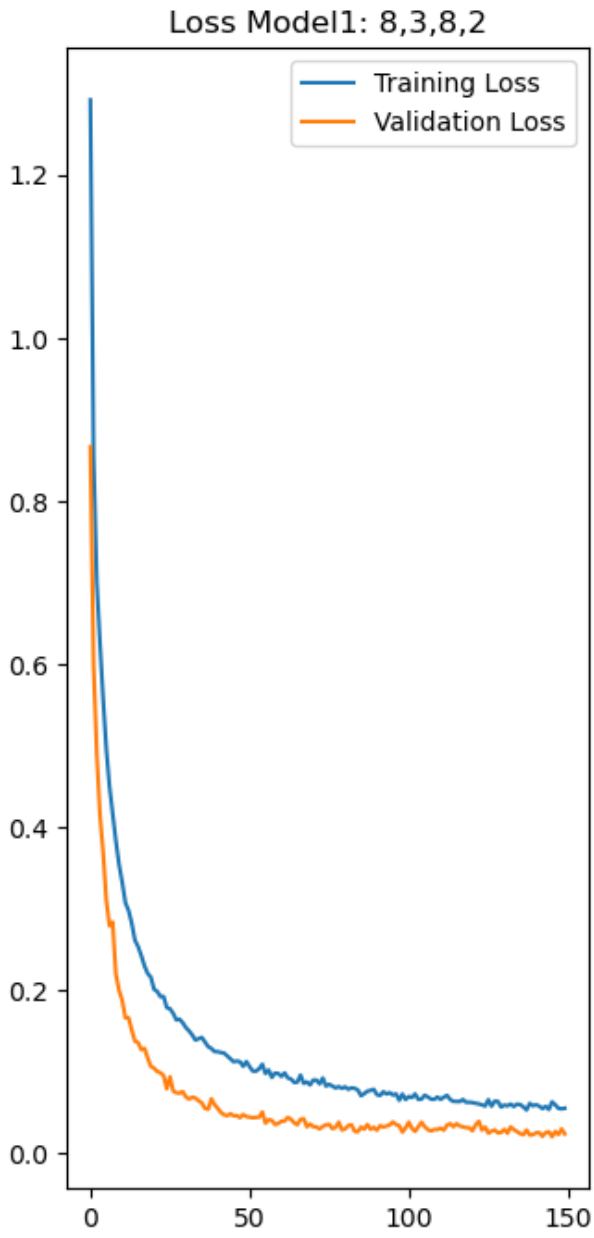
Graphs tracking accuracy and loss for models 1 through 36 against the training and validation data, as well as a table of information regarding individual model performance are included below.

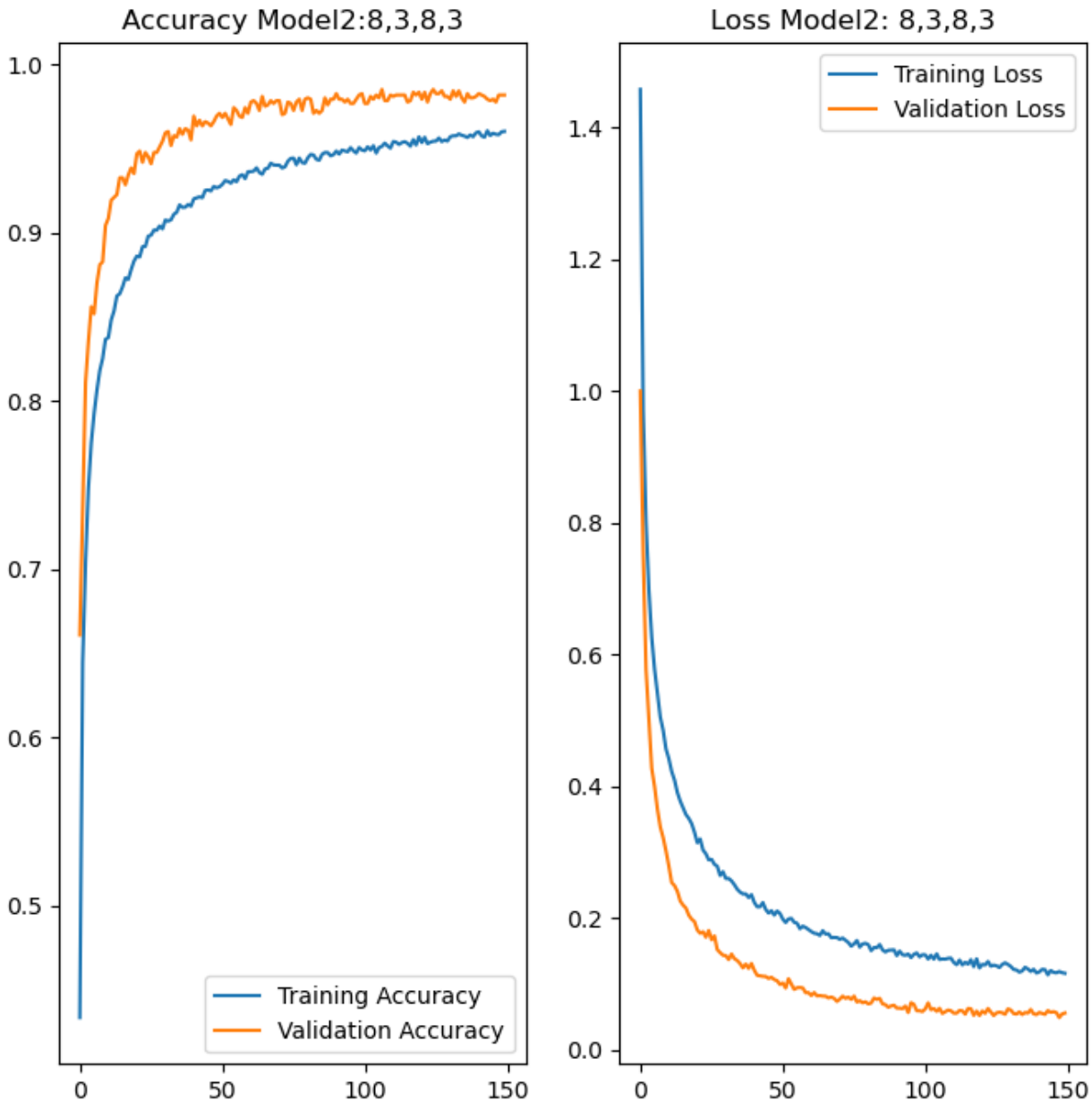
Model	Configuration	Epochs	Batch Size	Train Time	Parameters	Parsimony Ratio	Train Acc (%)	Train Loss (%)	Validation Acc (%)	Validation Loss (%)
1	8,3,8,2	150	115	71.12	116,776	1.55	99.94	0.35	99.25	2.40
2	8,3,8,3	150	115	69.94	82,600	2.19	99.74	1.67	98.17	5.56
3	8,3,16,2	150	115	69.65	229,936	0.79	100.00	0.02	99.08	3.58
4	8,3,16,3	150	115	70.46	161,584	1.12	99.98	0.17	98.83	3.98
5	8,5,8,2	150	115	68.42	82,408	2.20	99.74	1.59	98.00	6.02
6	8,5,8,3	150	115	68.68	82,728	2.19	99.81	1.40	99.08	4.61
7	8,5,16,2	150	115	68.38	161,072	1.12	99.92	0.33	98.50	4.79
8	8,5,16,3	150	115	66.74	161,712	1.12	99.97	0.18	98.33	3.74
9	8,7,8,2	150	115	67.75	82,600	2.19	99.74	1.65	98.08	6.01
10	8,7,8,3	150	115	69.34	54,696	3.31	98.03	7.95	96.17	12.43
11	8,7,16,2	150	115	67.22	161,264	1.12	99.99	0.15	98.92	3.96
12	8,7,16,3	150	115	66.10	105,456	1.72	99.90	0.91	97.75	5.40
13	16,3,16,2	150	115	67.44	230,528	0.79	100.00	0.04	99.25	2.32
14	16,3,16,3	150	115	67.83	162,816	1.11	99.97	0.12	99.17	3.07
15	16,3,32,2	150	115	67.38	457,360	0.40	100.00	0.00	99.33	2.65
16	16,3,32,3	150	115	66.96	321,936	0.56	100.00	0.01	98.92	2.64
17	16,5,16,2	150	115	65.19	161,792	1.12	99.95	0.21	99.25	1.65
18	16,5,16,3	150	115	66.13	163,072	1.11	99.99	0.12	99.17	3.03
19	16,5,32,2	150	115	67.16	319,632	0.57	99.99	0.03	99.17	3.64
20	16,5,32,3	150	115	66.47	322,192	0.56	100.00	0.01	99.25	3.27
21	16,7,16,2	150	115	64.84	162,176	1.12	99.99	0.11	99.17	3.06
22	16,7,16,3	150	115	63.48	107,008	1.69	99.90	0.79	98.33	4.84
23	16,7,32,2	150	115	65.17	320,016	0.57	100.00	0.02	99.17	2.89
24	16,7,32,3	150	115	65.69	209,680	0.86	99.96	0.22	98.67	4.45
25	24,3,24,2	150	115	70.88	344,792	0.53	100.00	0.01	99.00	3.96
26	24,3,24,3	150	115	69.79	244,184	0.74	100.00	0.04	99.17	2.69
27	24,3,48,2	150	115	73.69	685,808	0.26	100.00	0.00	99.42	2.71
28	24,3,48,3	150	115	75.36	484,592	0.37	100.00	0.01	99.08	3.02
29	24,5,24,2	150	115	67.53	241,688	0.75	99.99	0.06	99.17	2.20

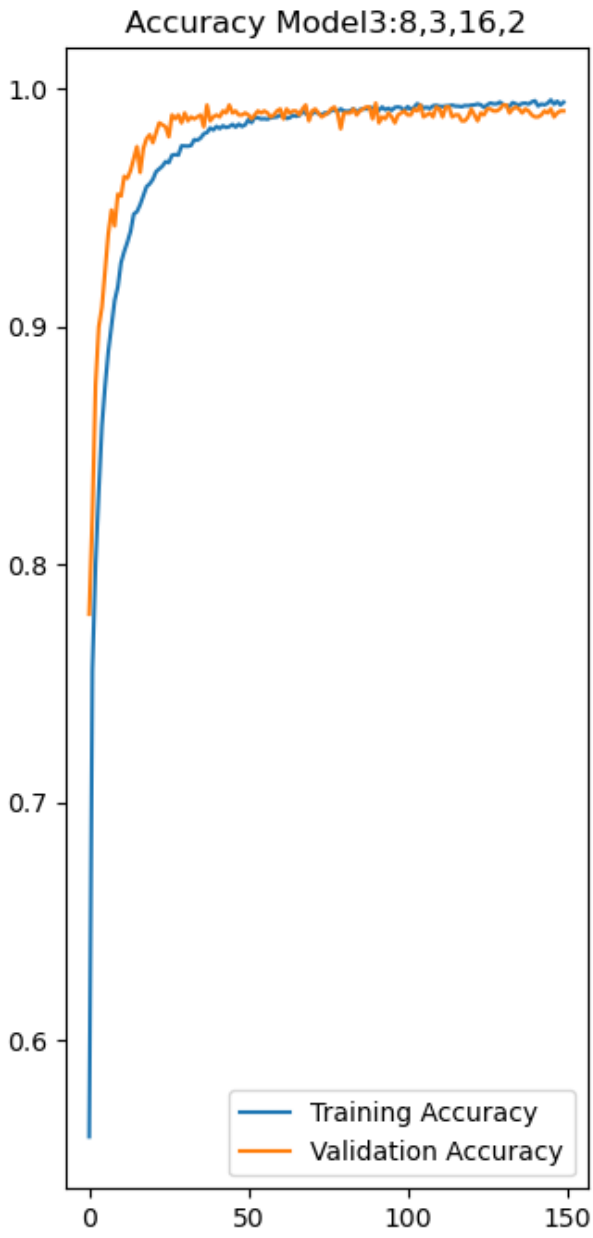
30	24,5,24,3	150	115	67.10	244,568	0.74	99.99	0.06	99.08	1.91
31	24,5,48,2	150	115	69.92	479,216	0.38	100.00	0.03	99.42	2.51
32	24,5,48,3	150	115	71.49	484,976	0.37	100.00	0.01	99.42	2.62
33	24,7,24,2	150	115	66.39	242,264	0.75	99.96	0.18	99.00	2.52
34	24,7,24,3	150	115	66.62	160,472	1.13	99.95	0.29	98.08	4.61
35	24,7,48,2	150	115	70.82	479,792	0.38	100.00	0.01	99.25	2.26
36	24,7,48,3	150	115	68.16	316,208	0.57	100.00	0.08	98.67	4.28

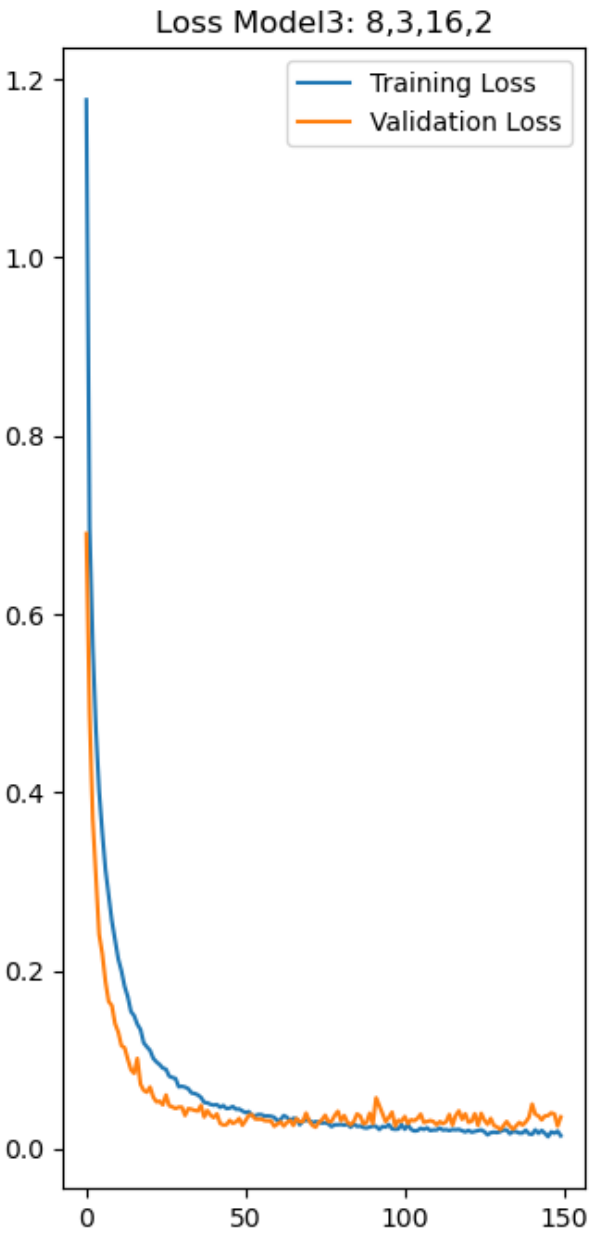
Summary of Models 1 - 36

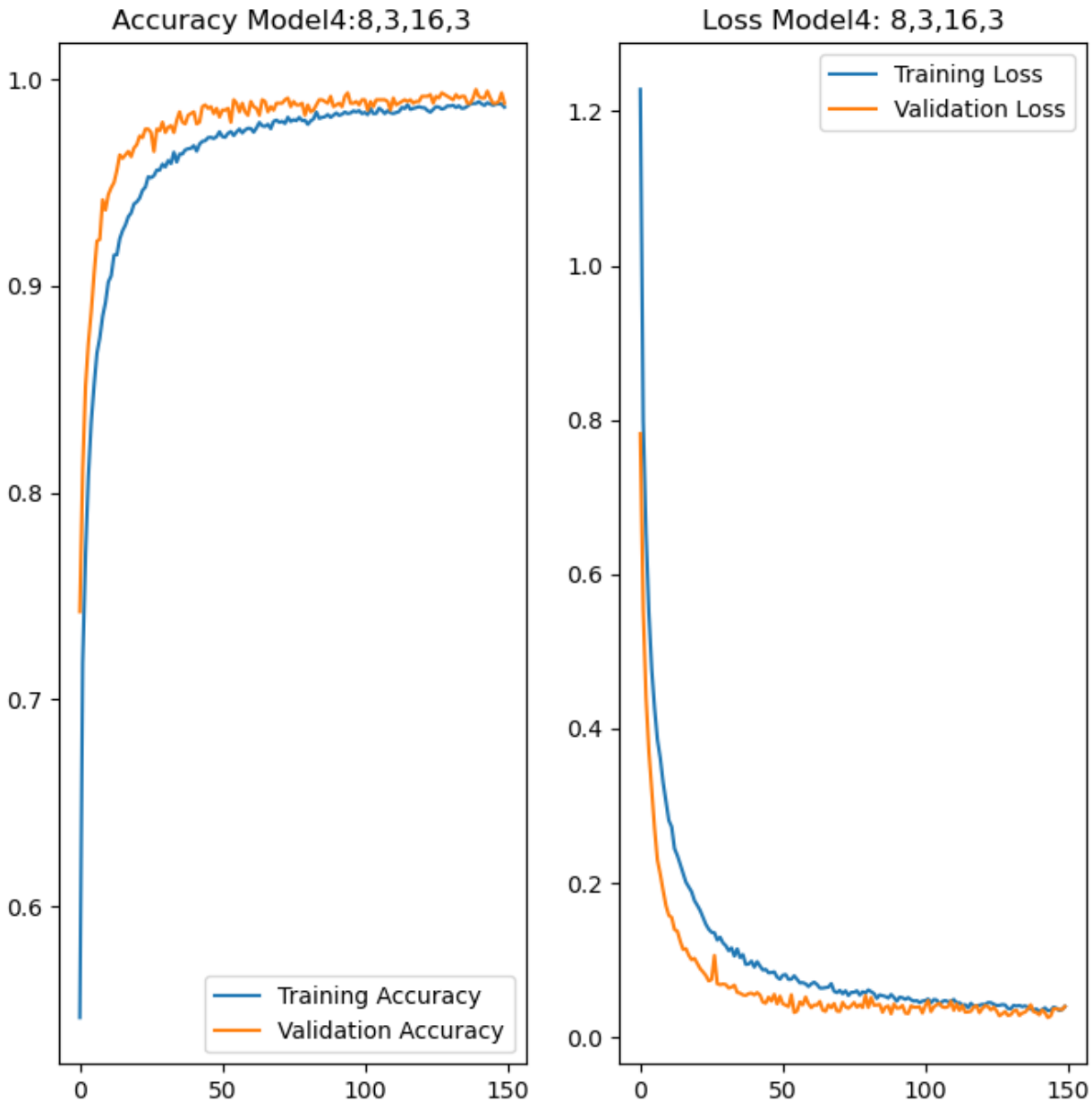

Accuracy Model1:8,3,8,2
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
Training Accuracy
Validation Accuracy

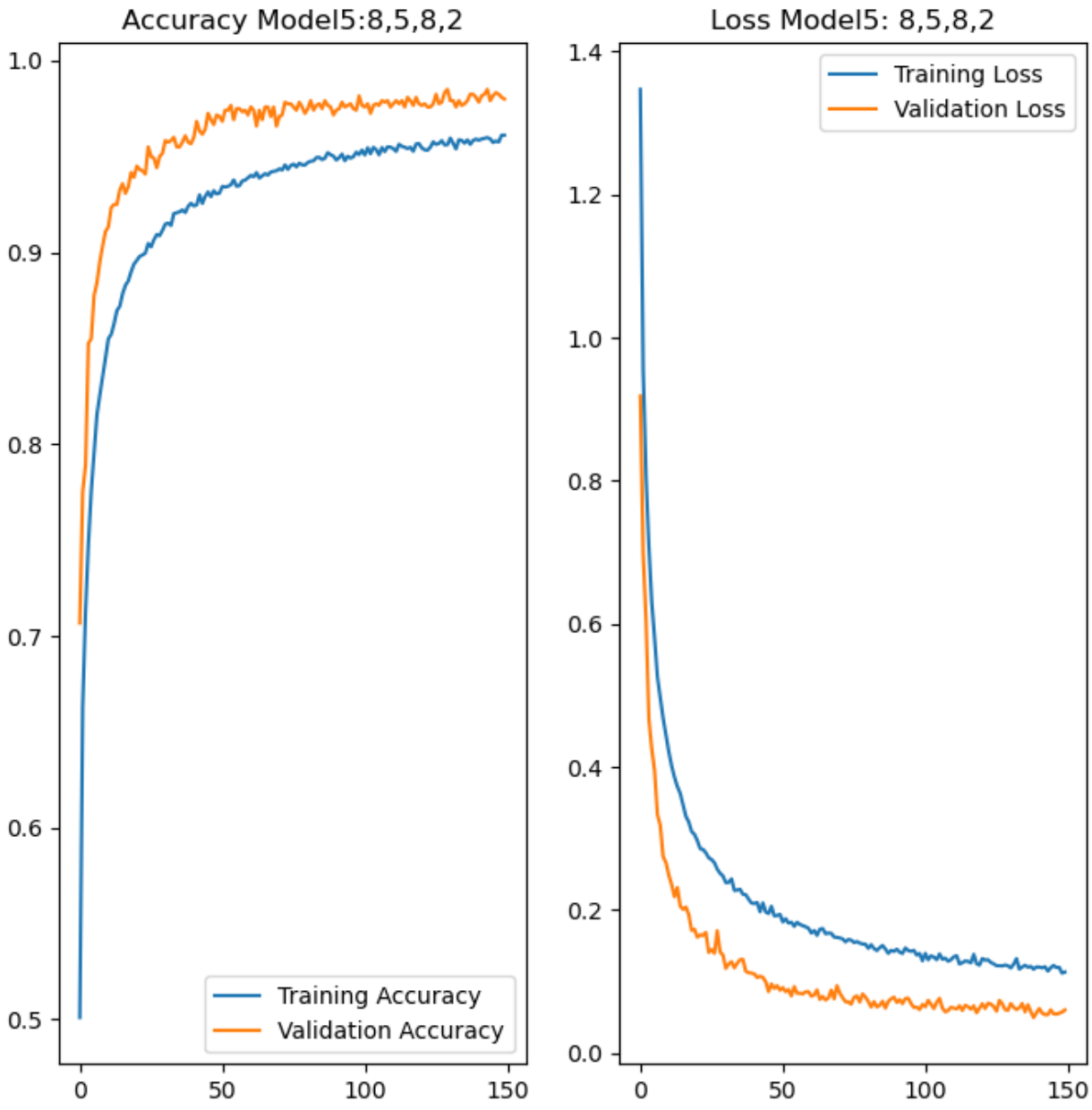

Loss Model1: 8,3,8,2
Training Loss
Validation Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

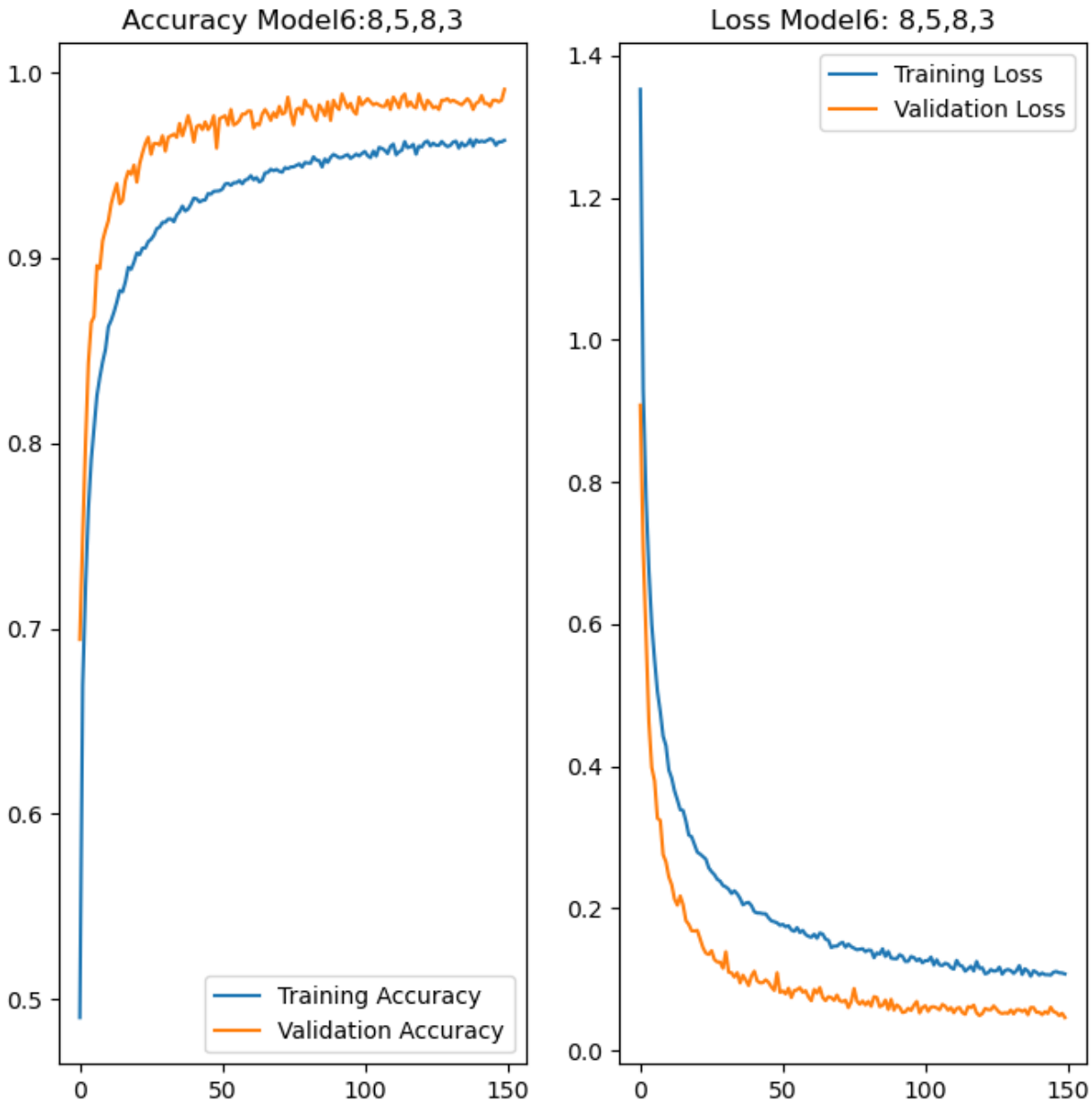

Accuracy Model2:8,3,8,3
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model2: 8,3,8,3
Training Loss
Validation Loss
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

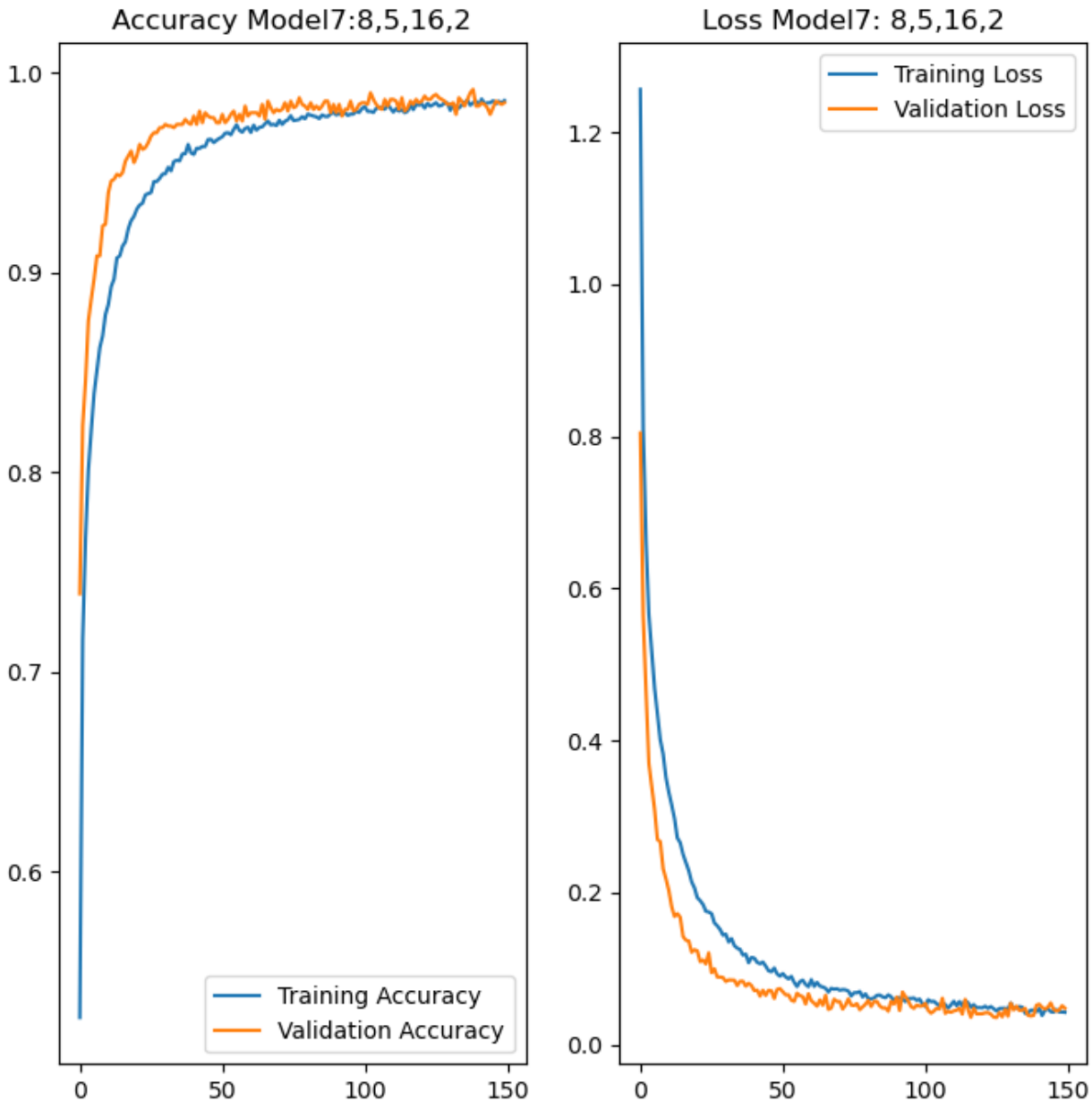

Accuracy Model3:8,3,16,2
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy

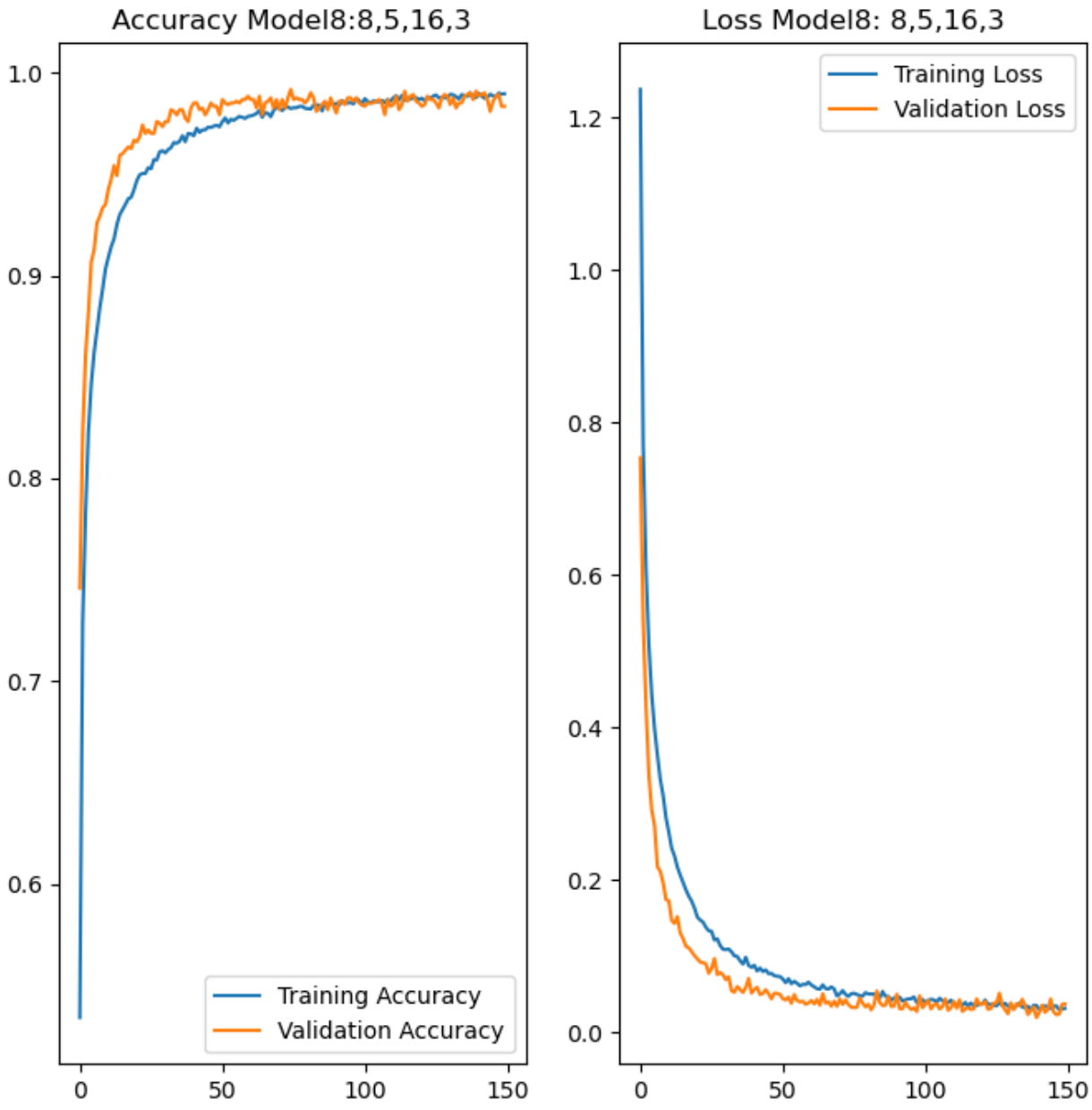

Loss Model3: 8,3,16,2
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

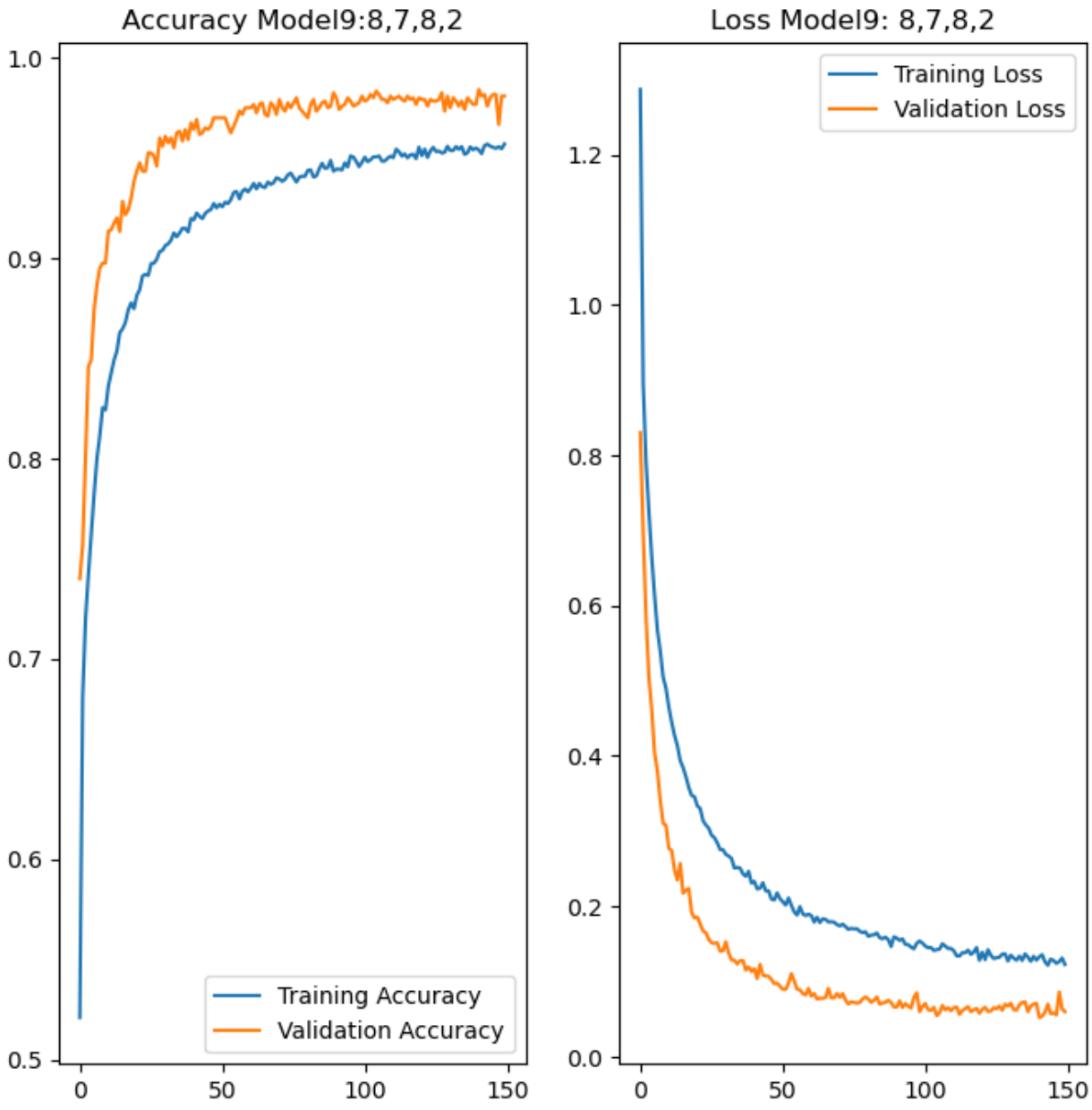

Accuracy Model4:8,3,16,3
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model4: 8,3,16,3
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

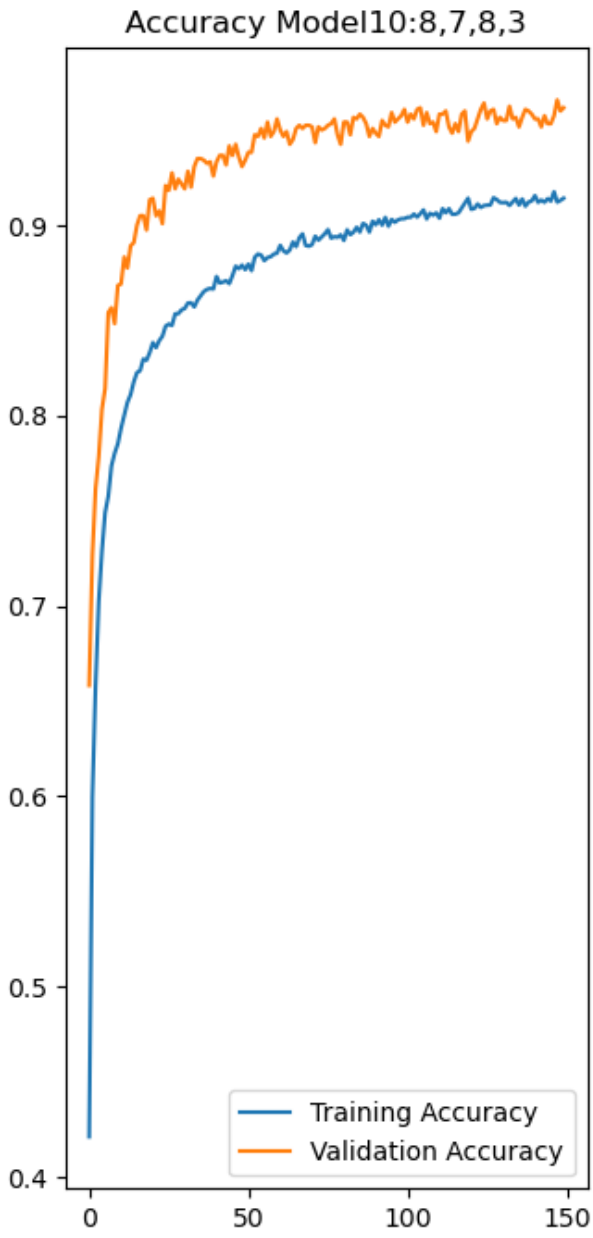

Accuracy Model5:8,5,8,2
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model5: 8,5,8,2
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

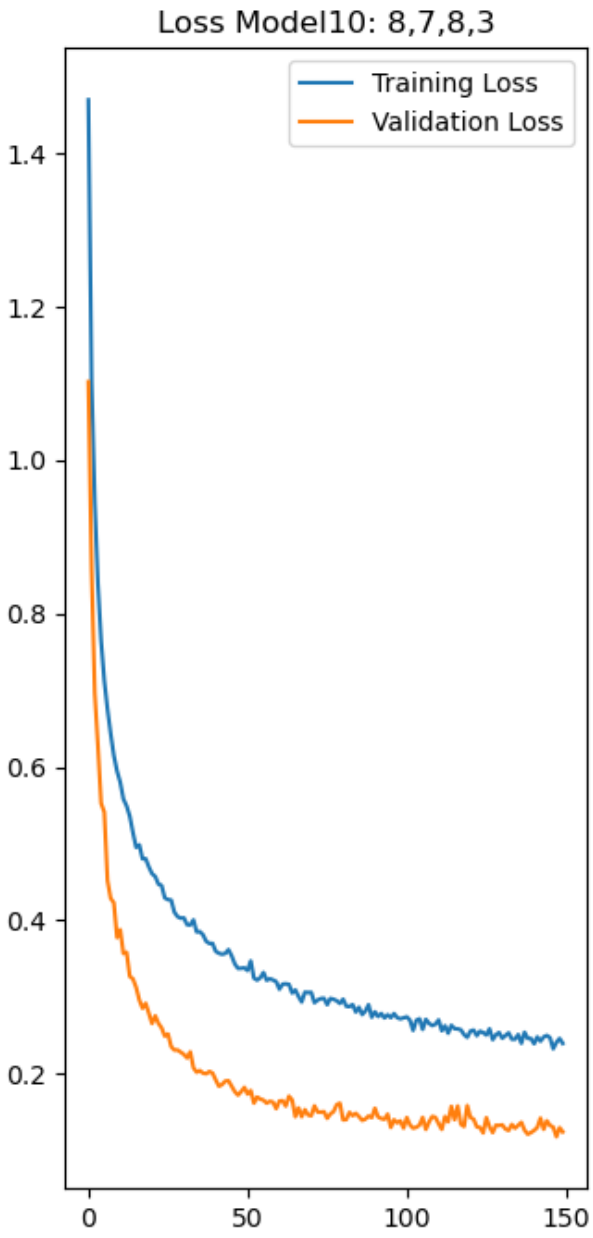

Accuracy Model6:8,5,8,3
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model6: 8,5,8,3
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

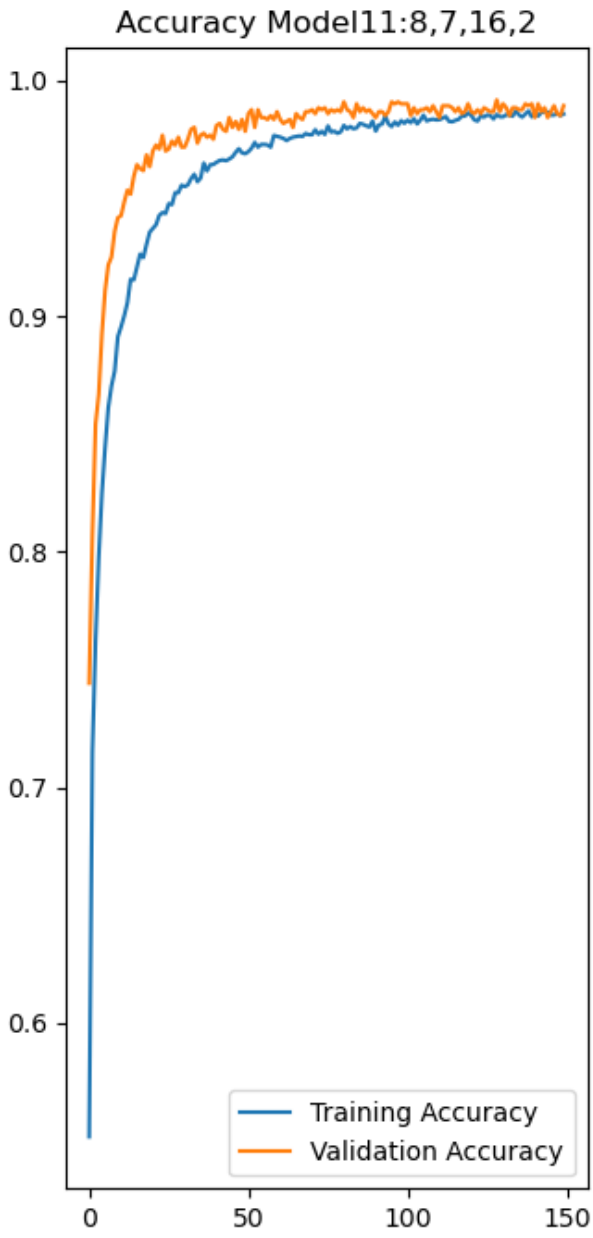

Accuracy Model7:8,5,16,2
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model7: 8,5,16,2
Training Loss
Validation Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

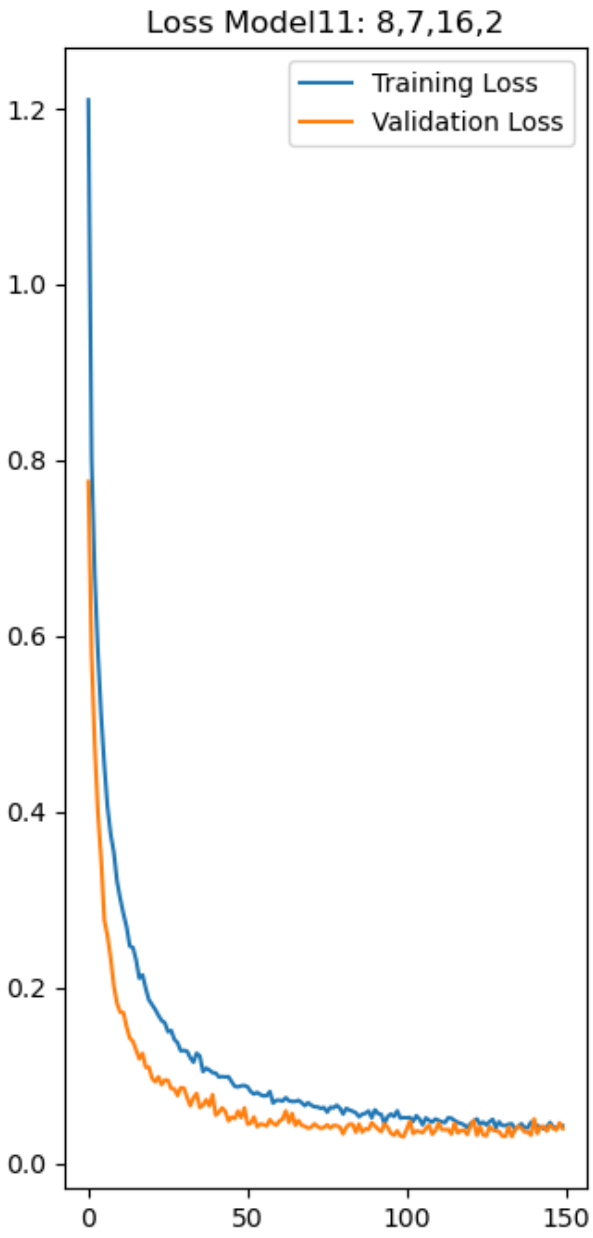

Accuracy Model8:8,5,16,3
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model8: 8,5,16,3
Training Loss
Validation Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

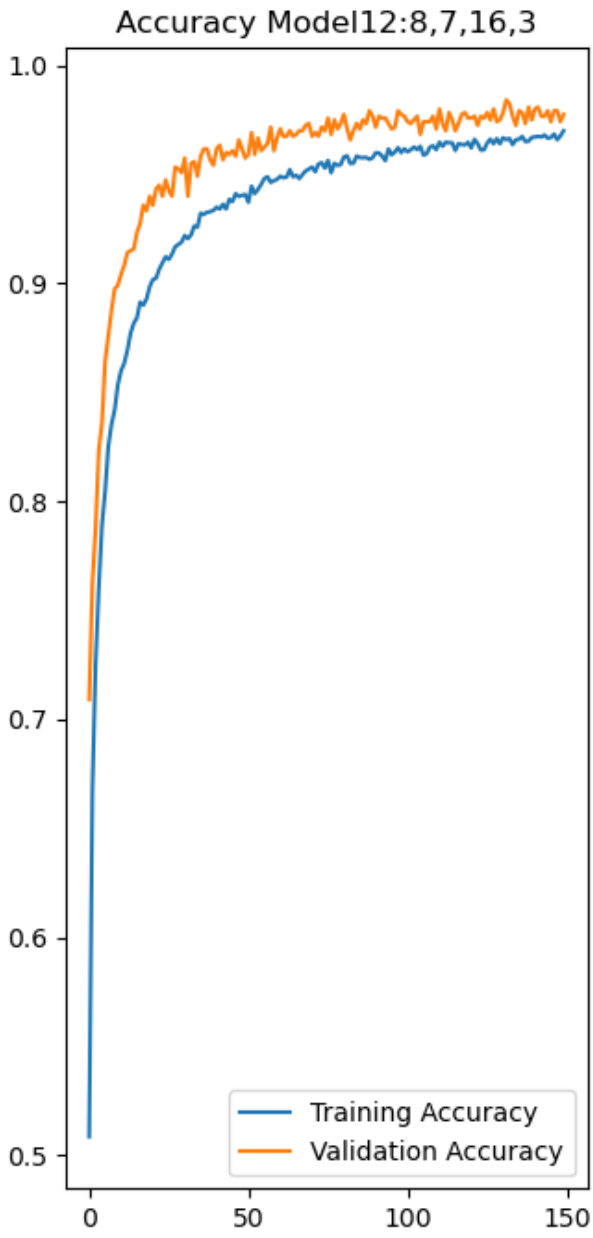

Accuracy Model9:8,7,8,2
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model9: 8,7,8,2
Training Loss
Validation Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

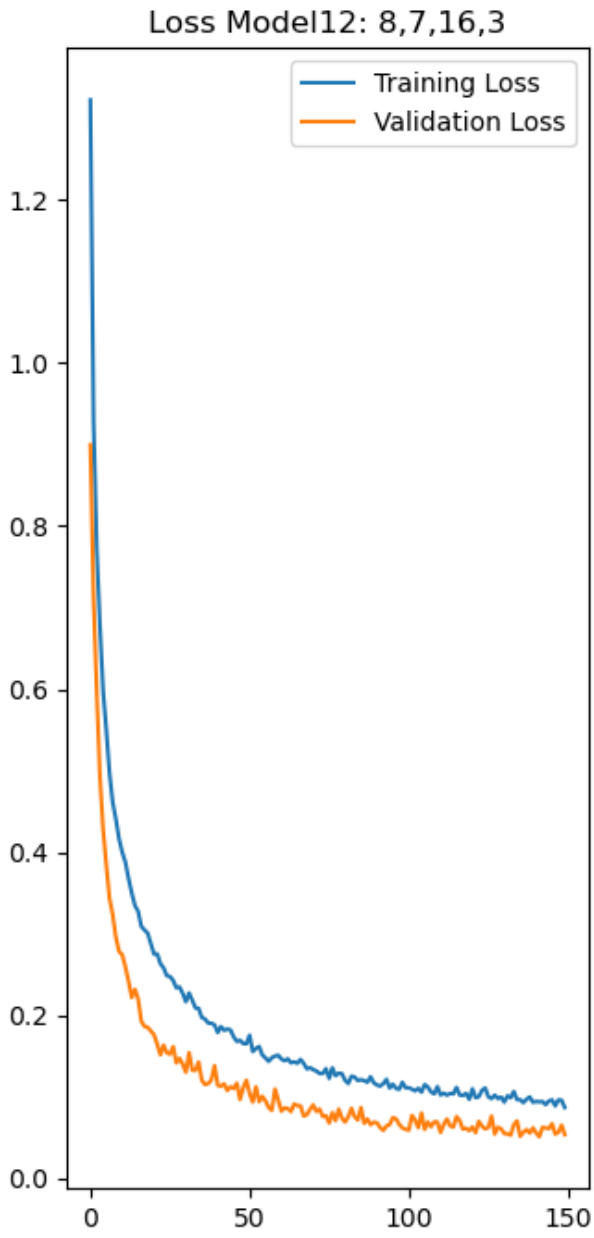

Accuracy Model10:8,7,8,3
0.9
0.8
0.7
0.6
0.5
0.4
0
50
100
150
Training Accuracy
Validation Accuracy

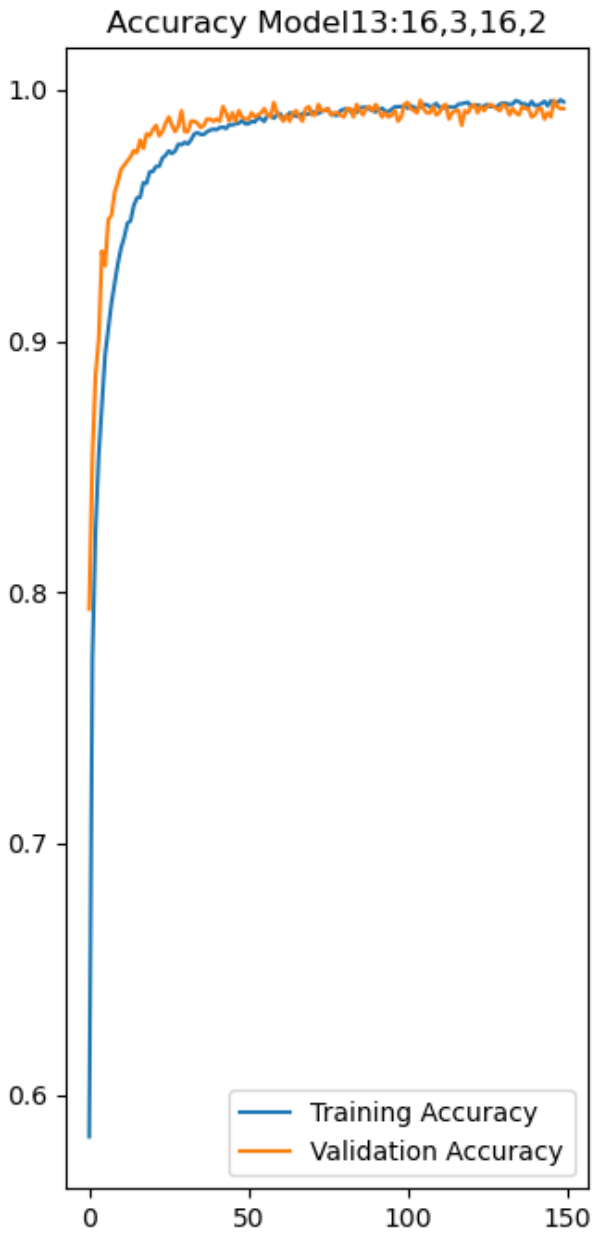

Loss Model10: 8,7,8,3
Training Loss
Validation Loss
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
50
100
150

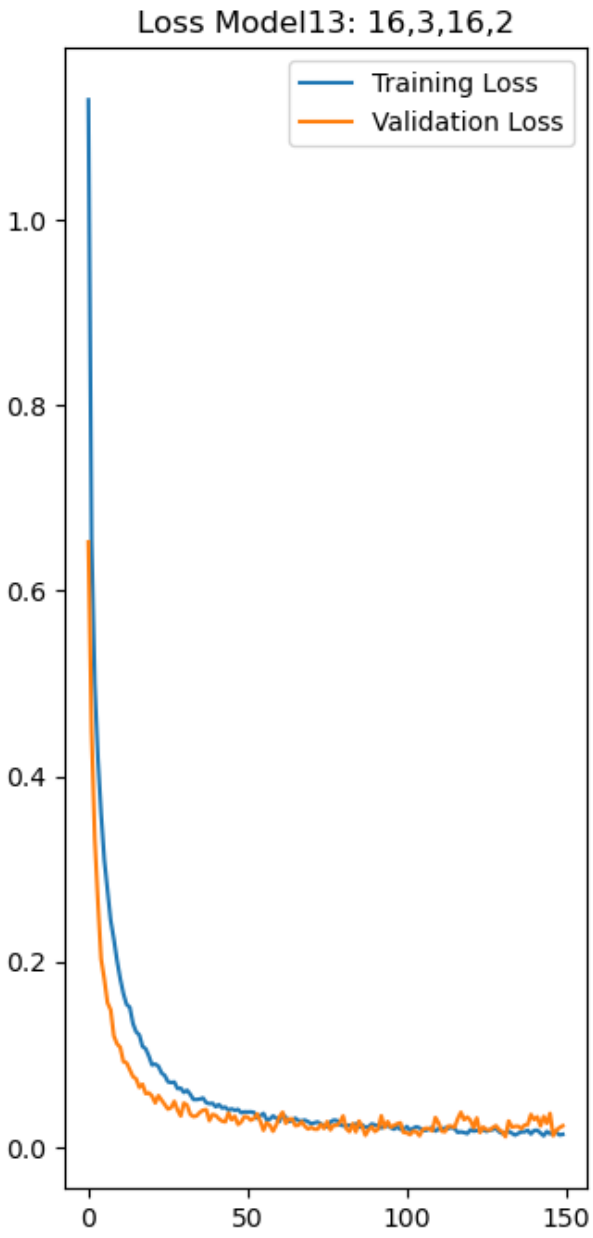

Accuracy Model11:8,7,16,2
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy

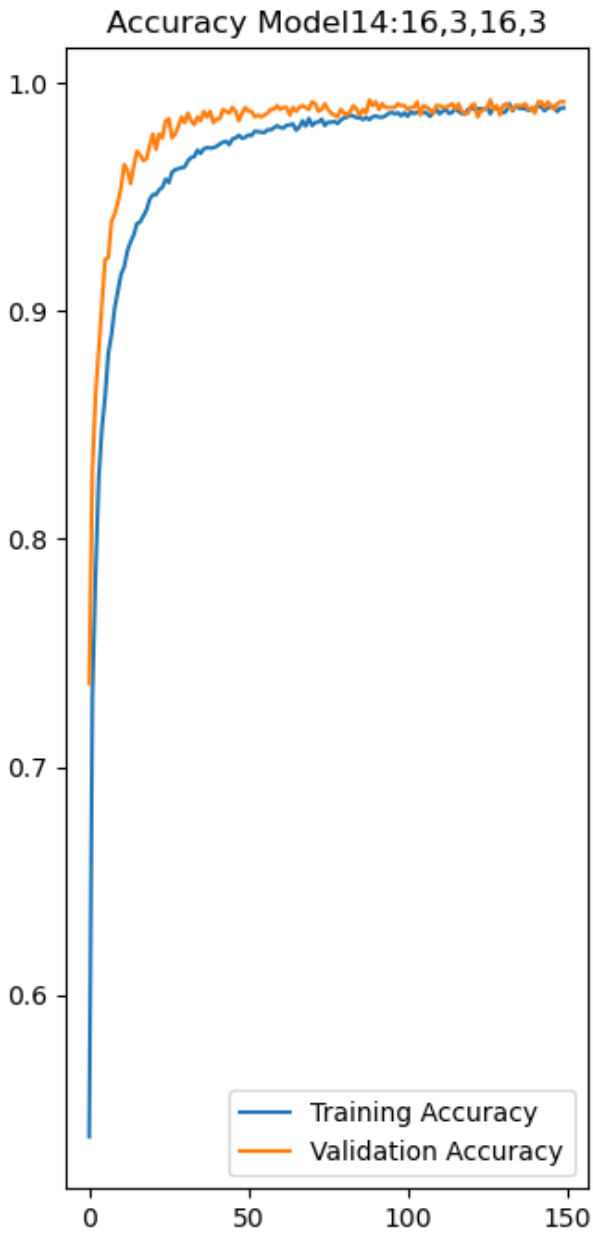

Loss Model11: 8,7,16,2
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

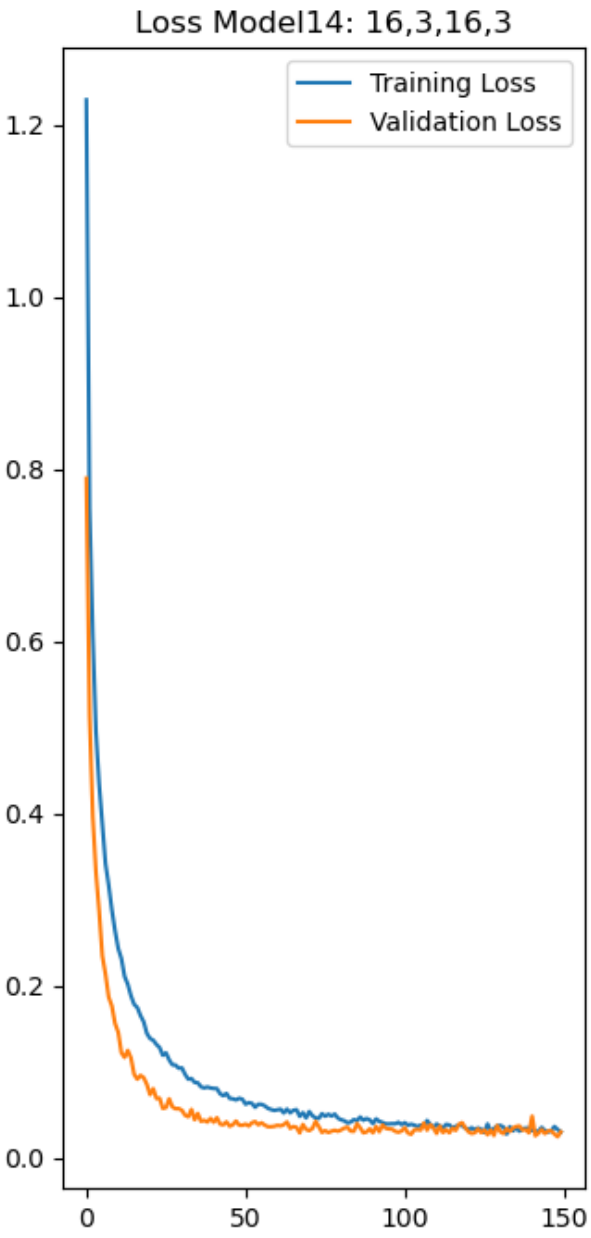

Accuracy Model12:8,7,16,3
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
Training Accuracy
Validation Accuracy

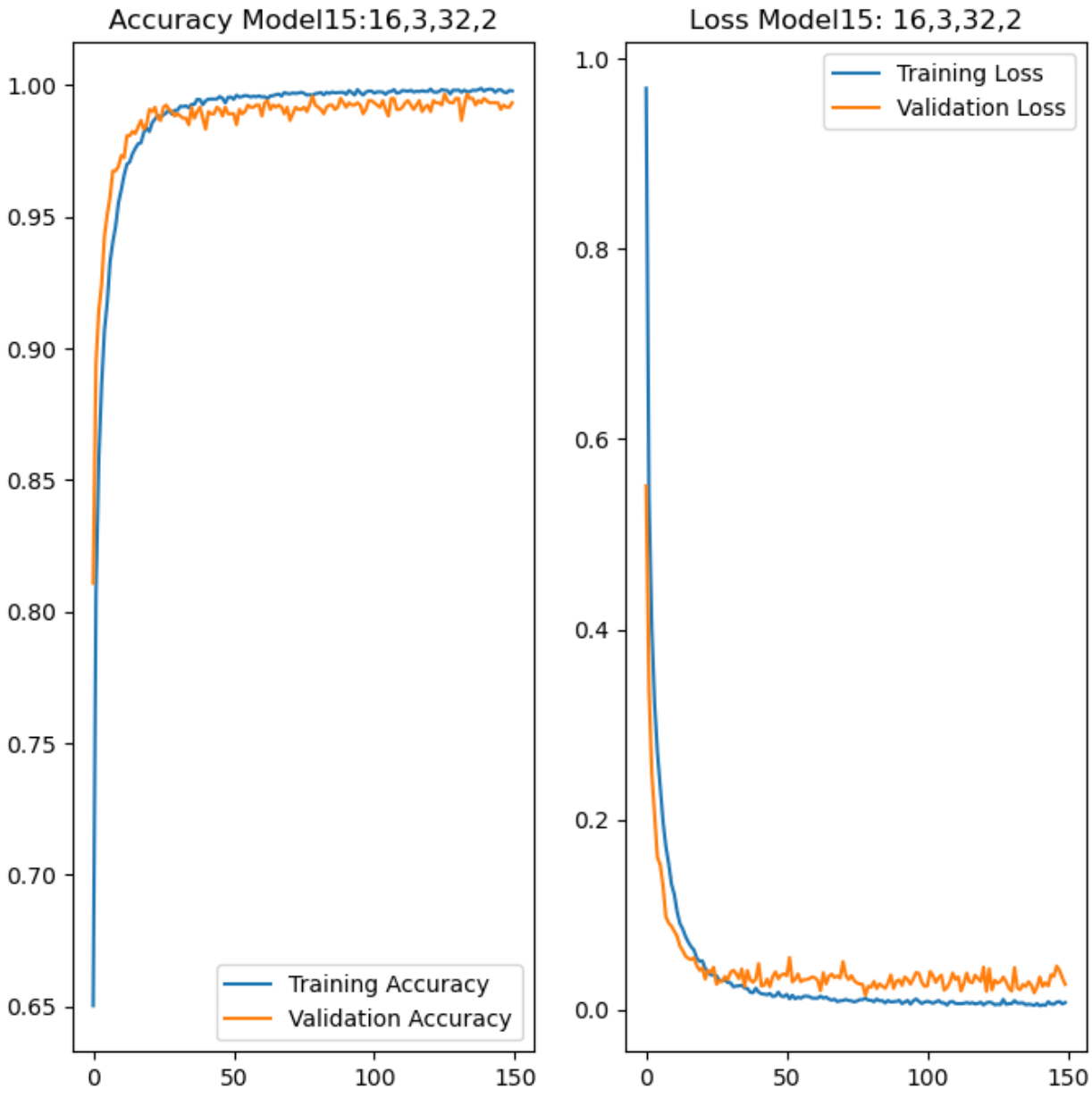

Loss Model12: 8,7,16,3
Training Loss
Validation Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

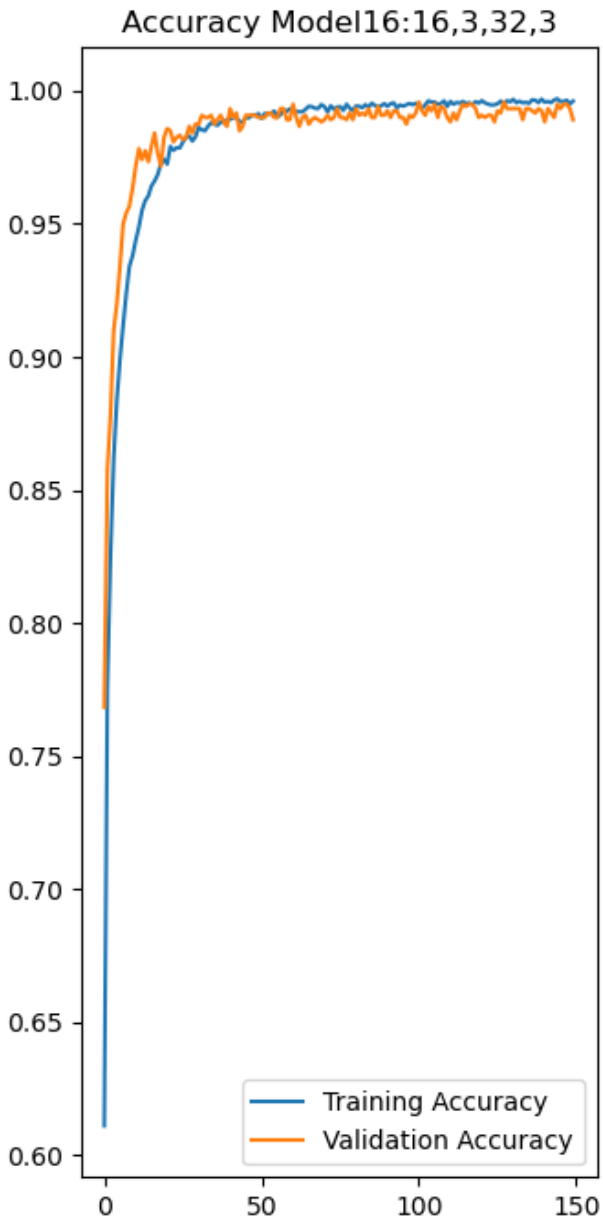

Accuracy Model13:16,3,16,2
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy

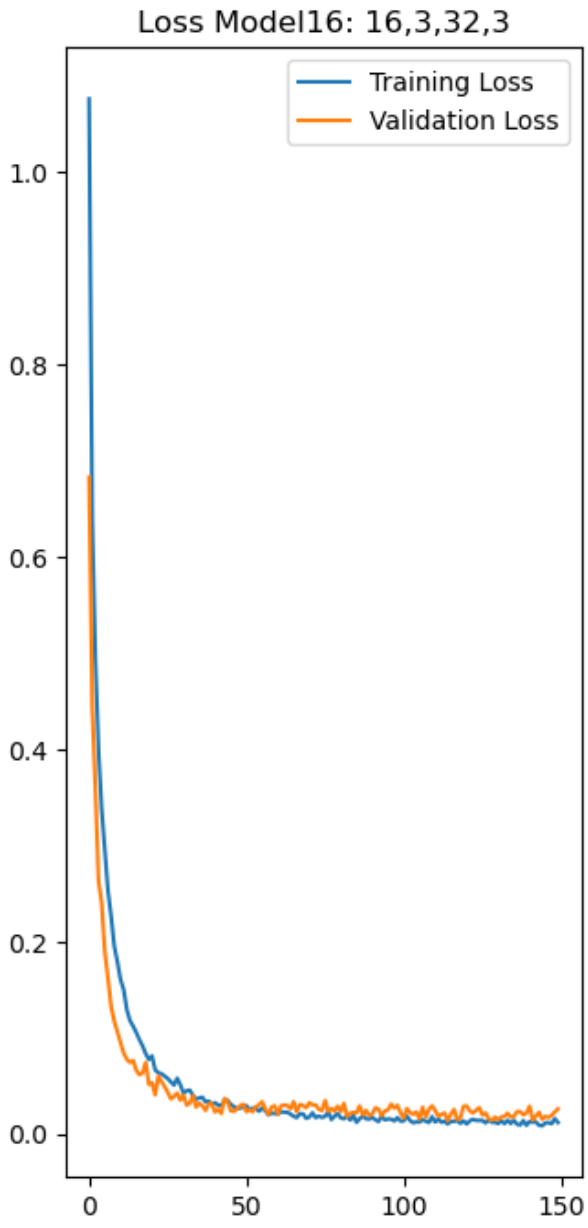

Loss Model13: 16,3,16,2
Training Loss
Validation Loss
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

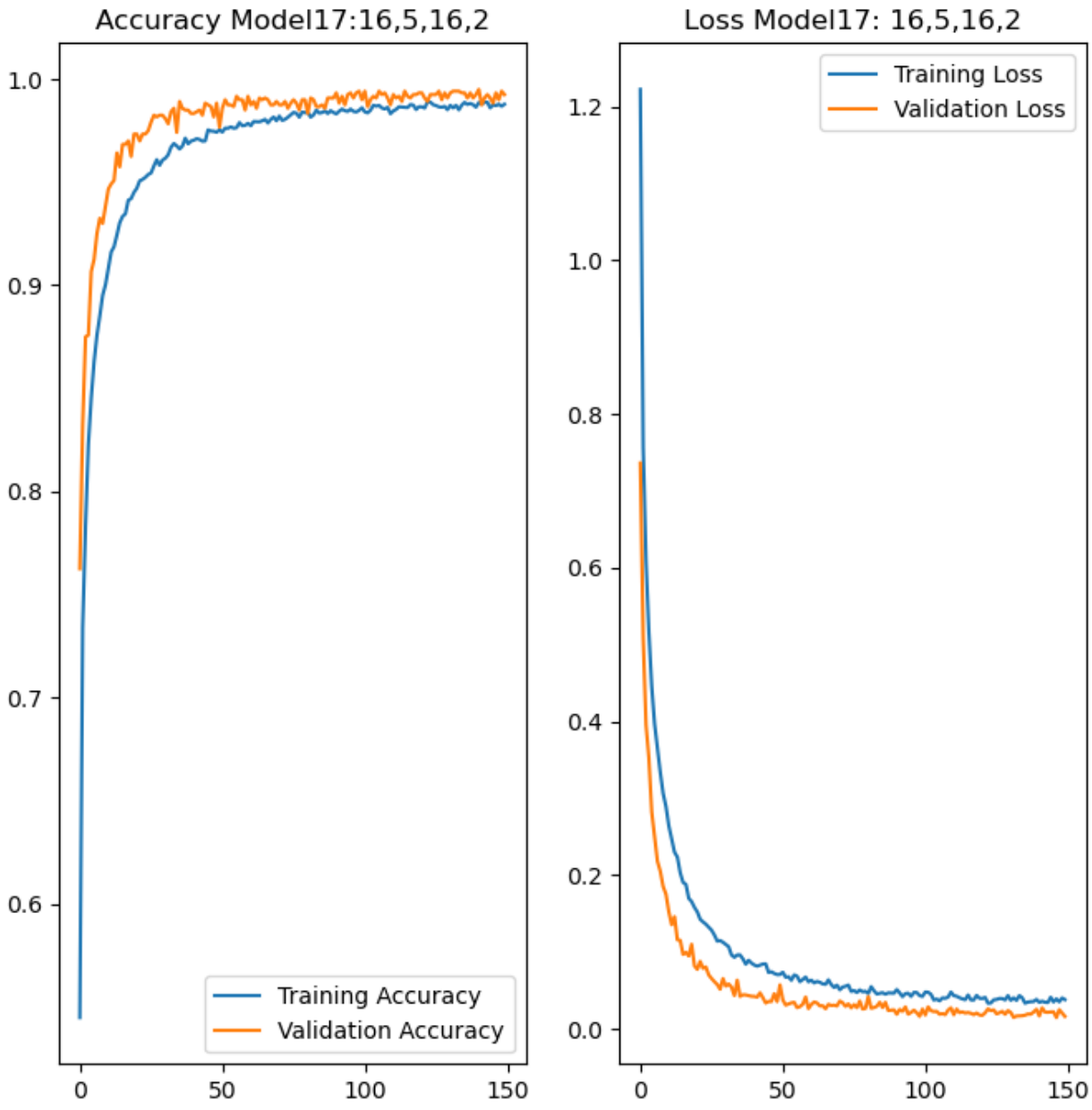

Accuracy Model14:16,3,16,3
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy

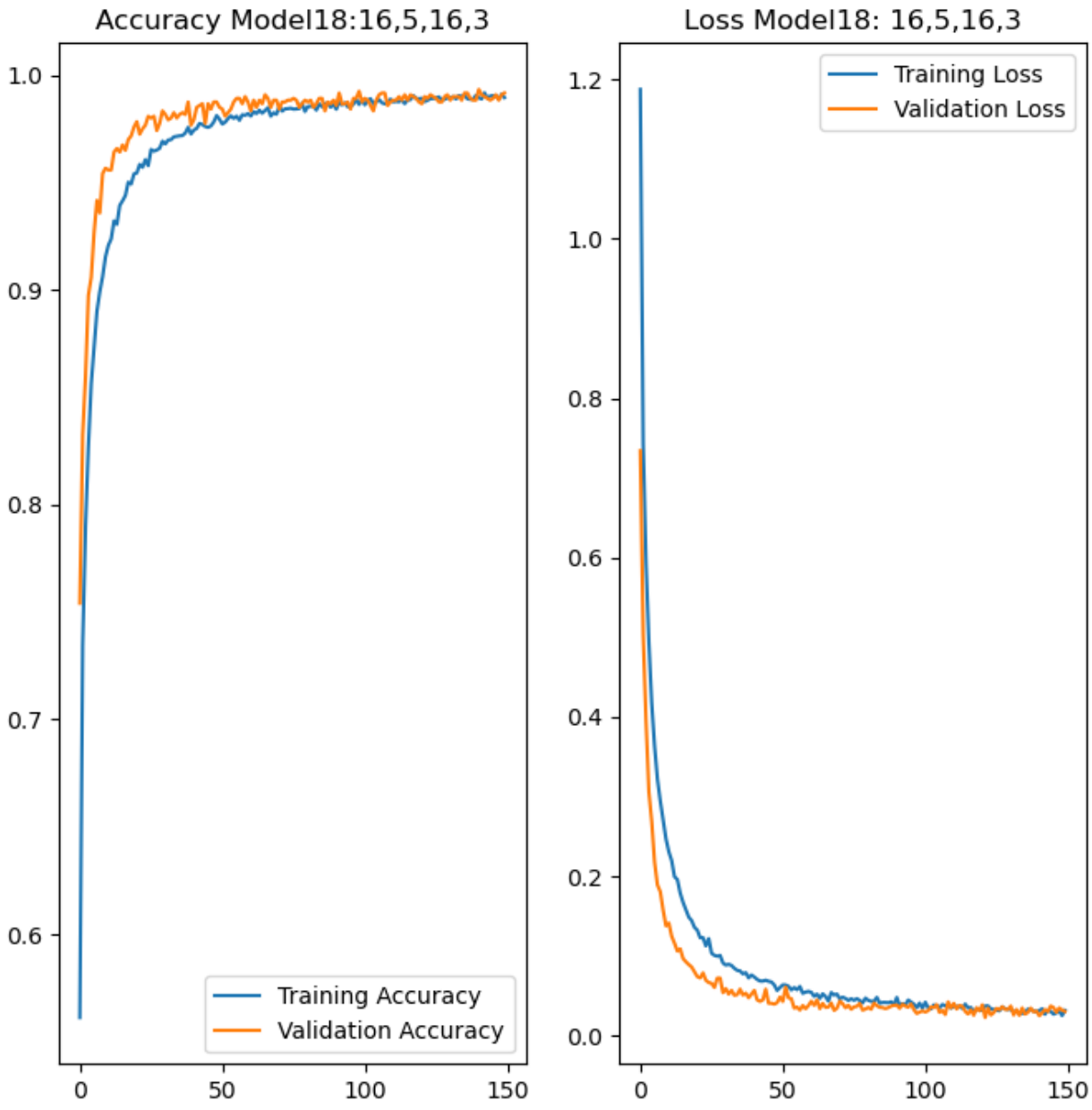

Loss Model14: 16,3,16,3
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

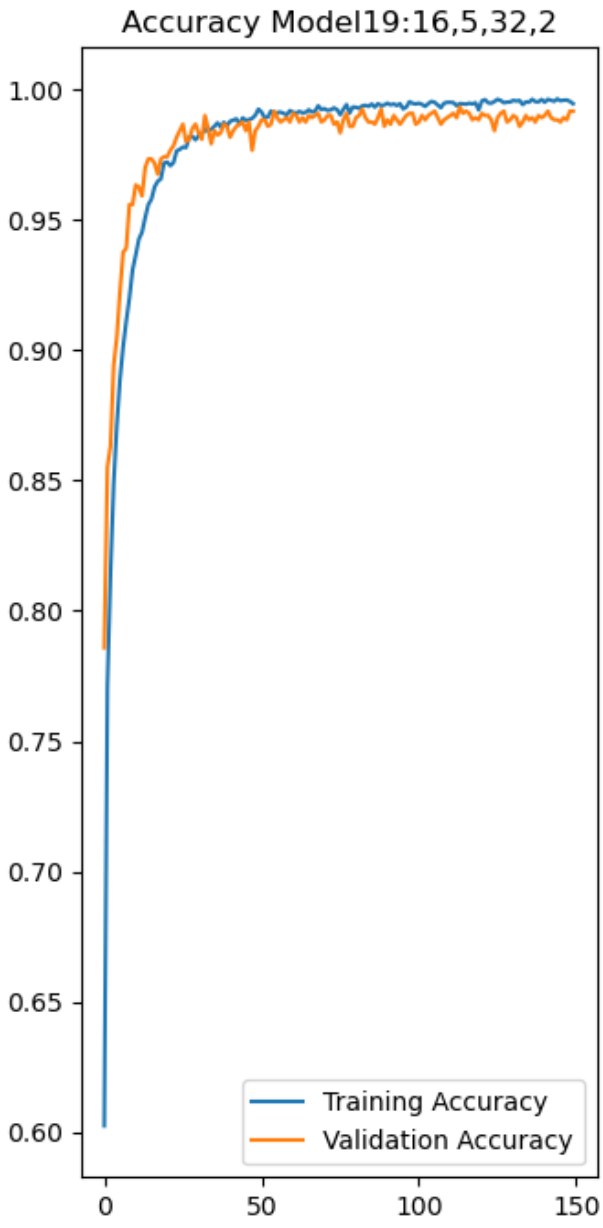

Accuracy Model15:16,3,32,2
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model15: 16,3,32,2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

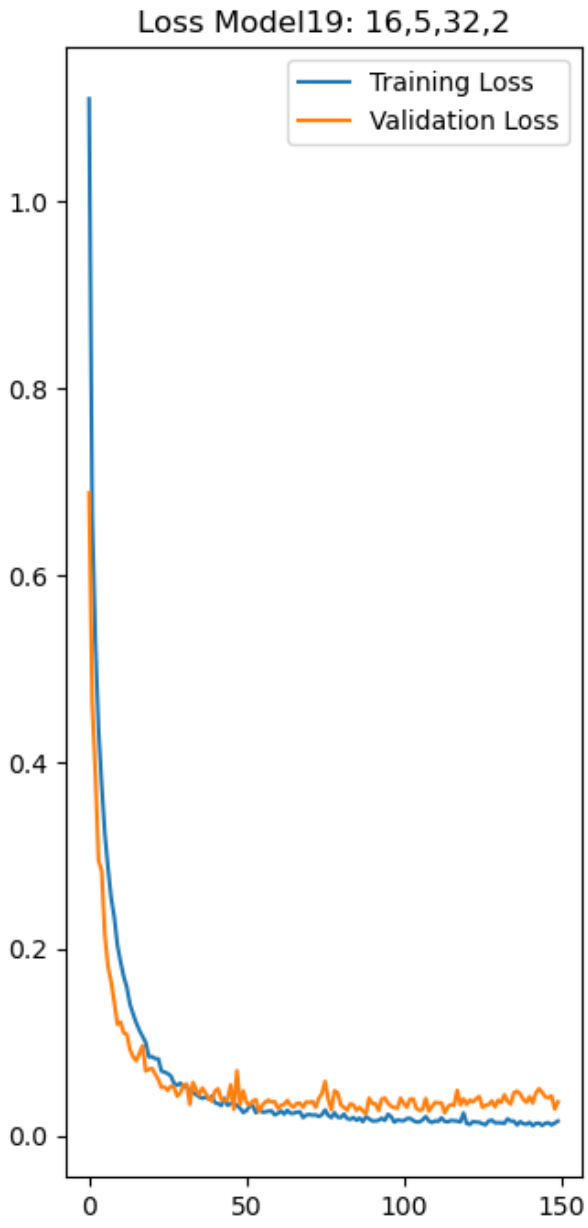

Accuracy Model16:16,3,32,3
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
50
100
150
Training Accuracy
Validation Accuracy

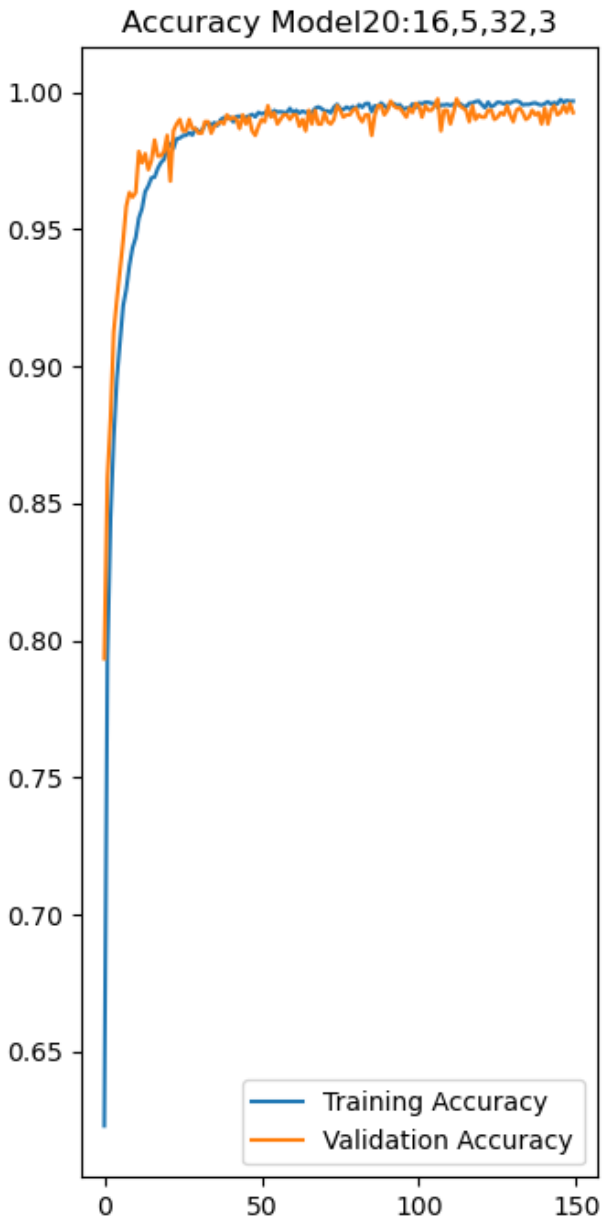

Loss Model16: 16,3,32,3
Training Loss
Validation Loss
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

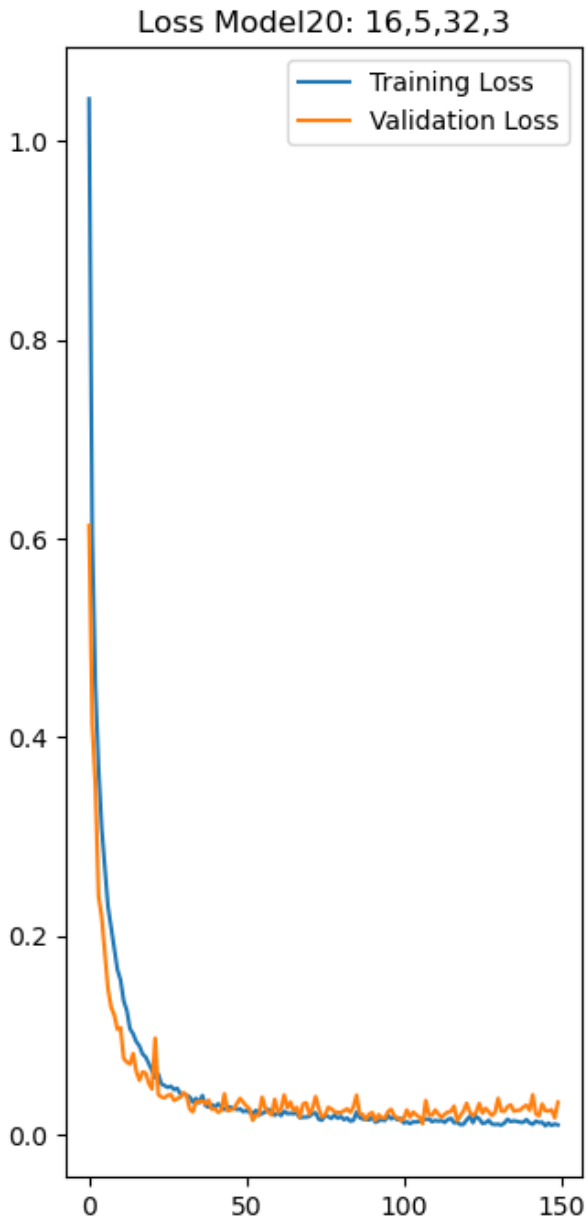

Accuracy Model17:16,5,16,2
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model17: 16,5,16,2
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

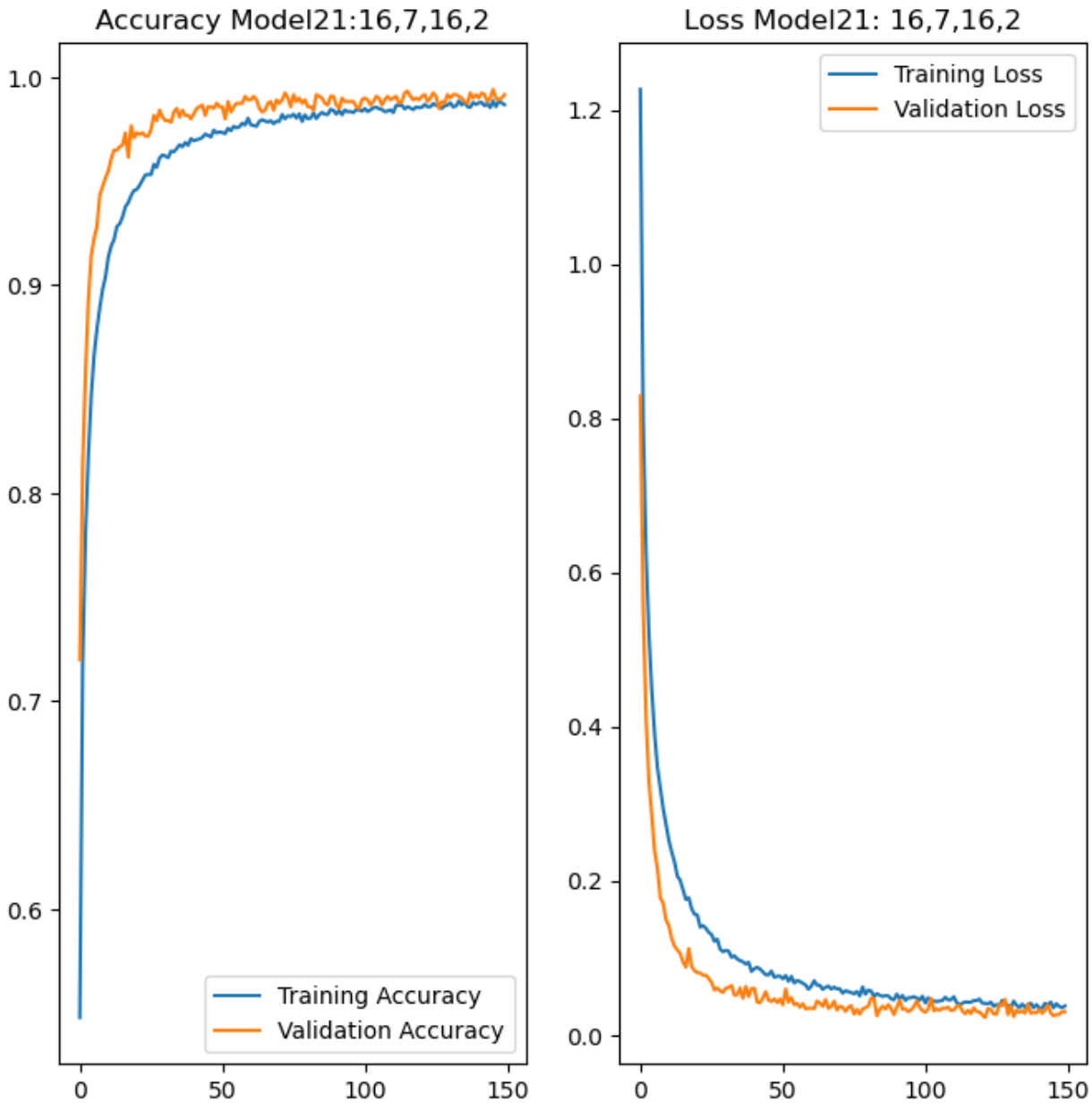

Accuracy Model18:16,5,16,3
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model18: 16,5,16,3
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

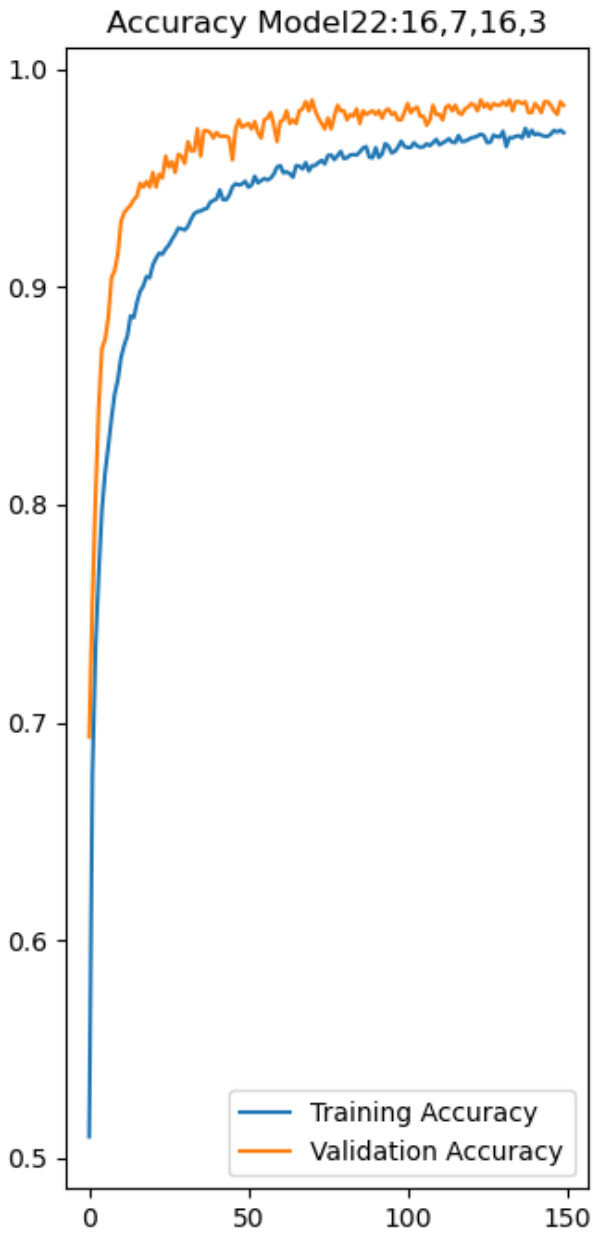

Accuracy Model19:16,5,32,2
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
50
100
150
Training Accuracy
Validation Accuracy

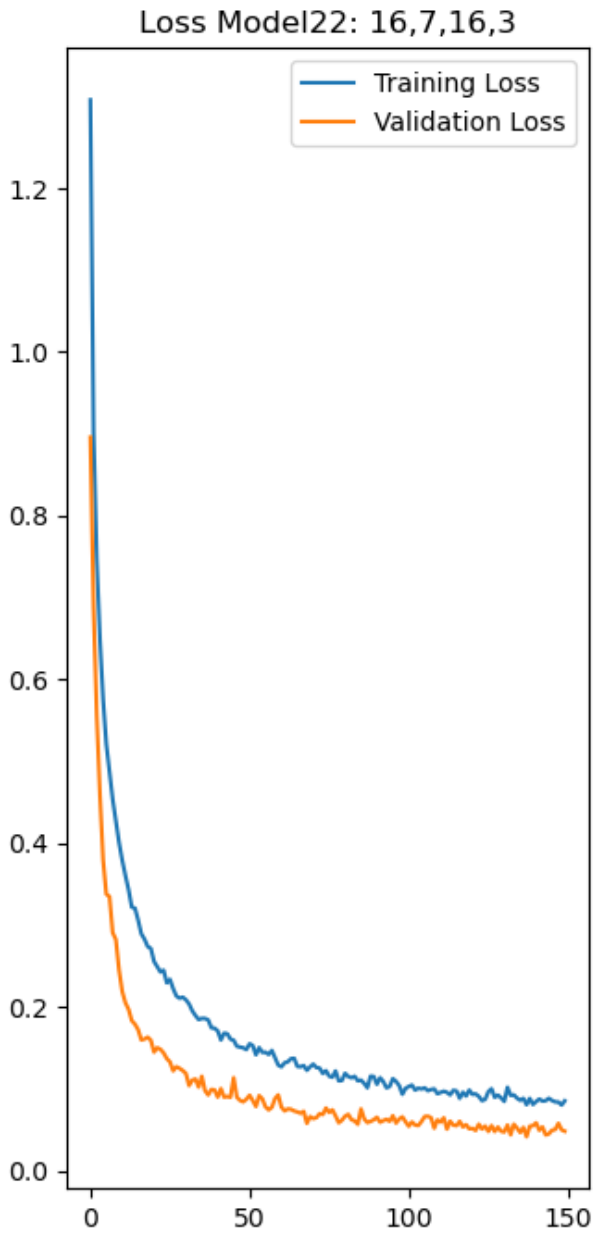

Loss Model19: 16,5,32,2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

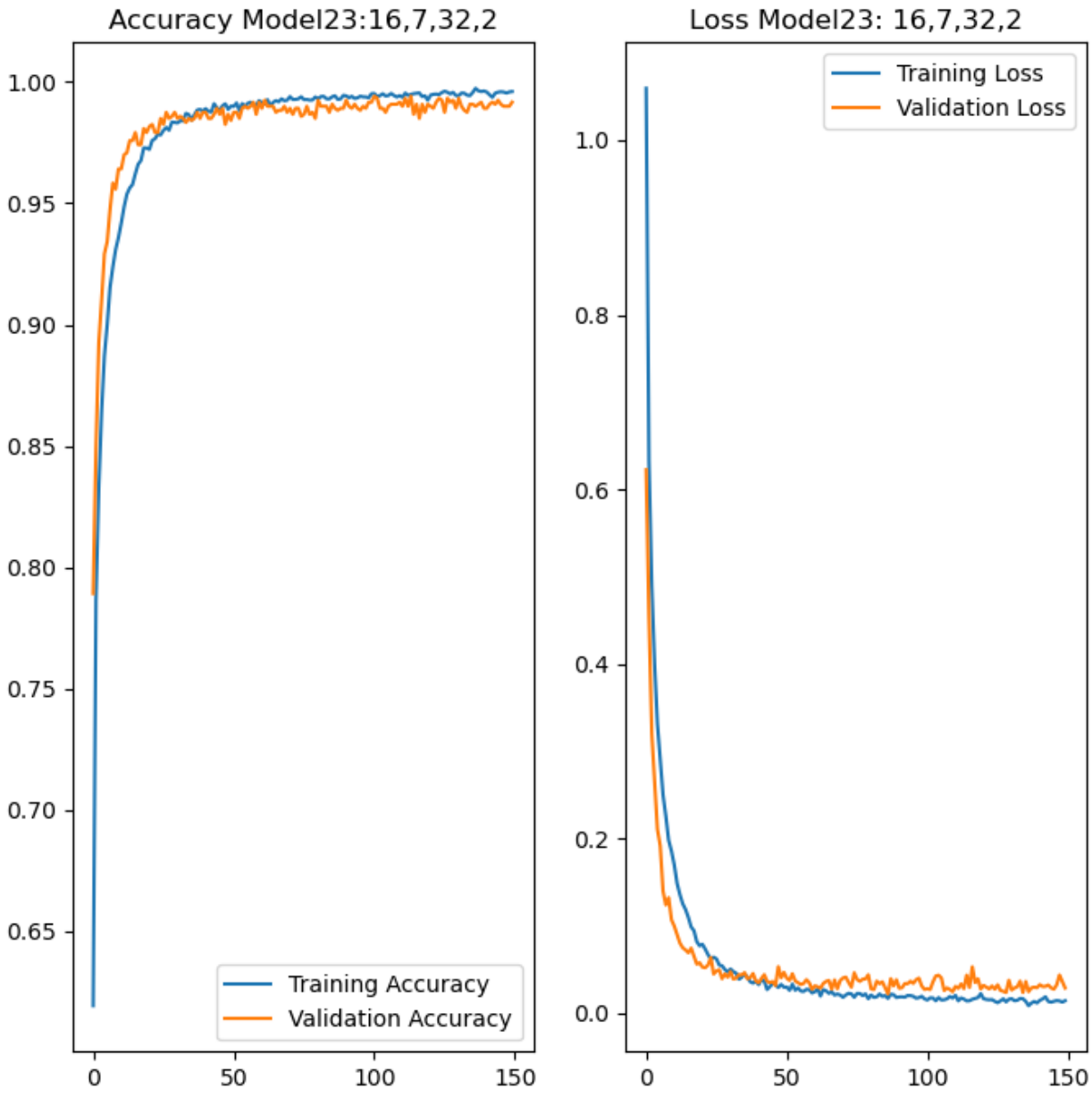

Accuracy Model20:16,5,32,3
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
50
100
150
Training Accuracy
Validation Accuracy

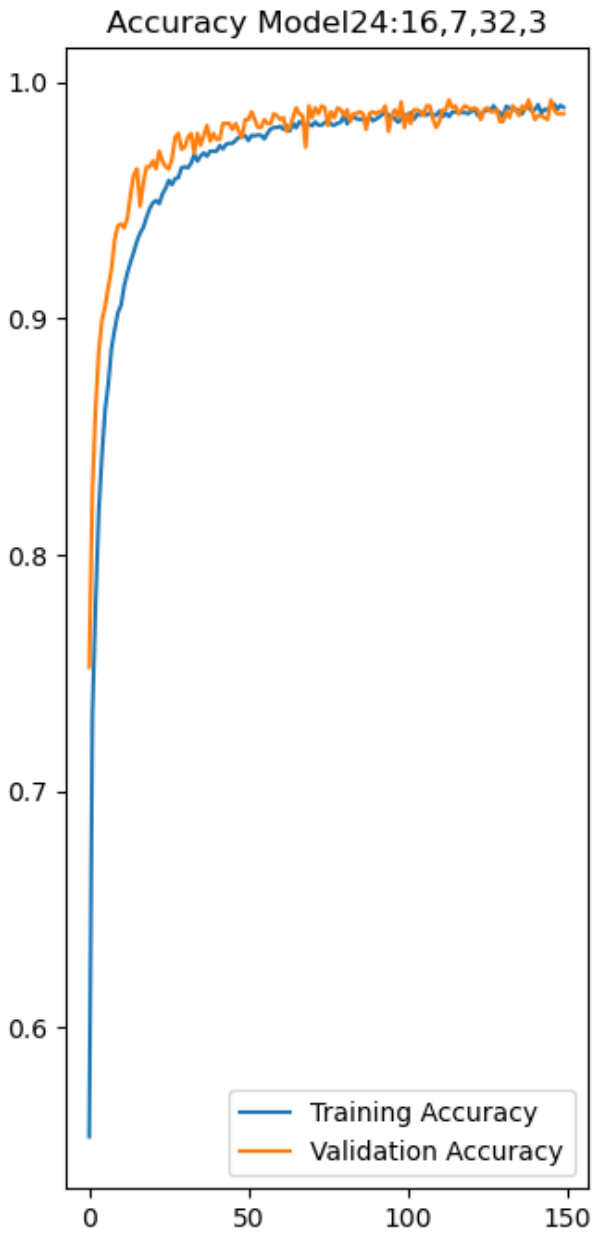

Loss Model20: 16,5,32,3
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

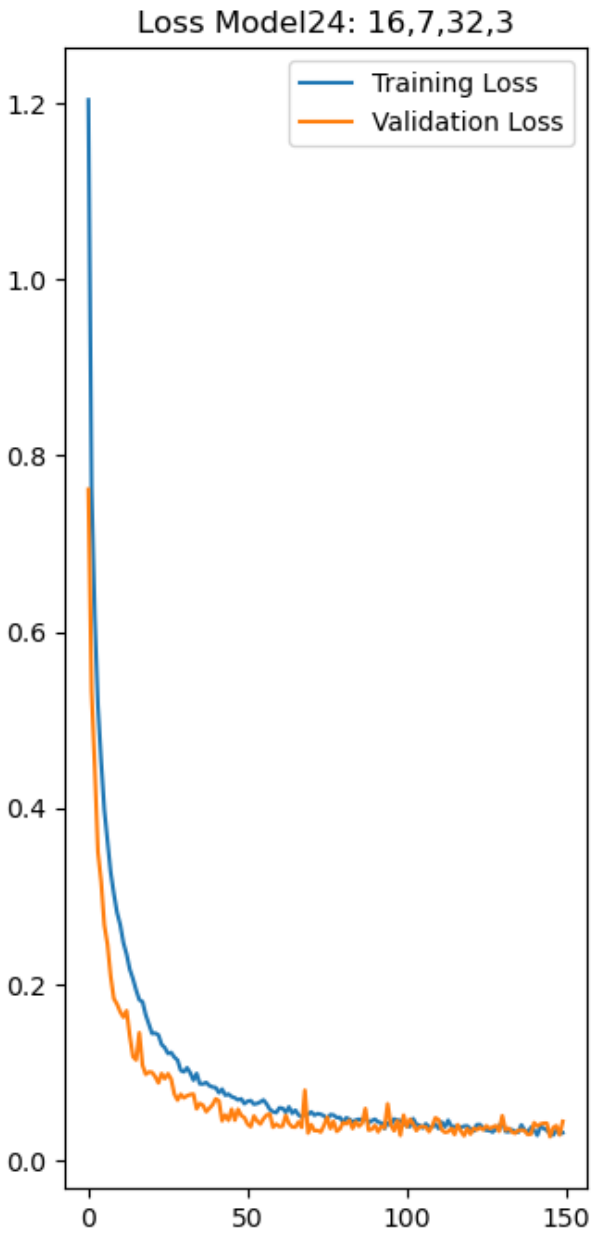

Accuracy Model21:16,7,16,2
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model21: 16,7,16,2
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

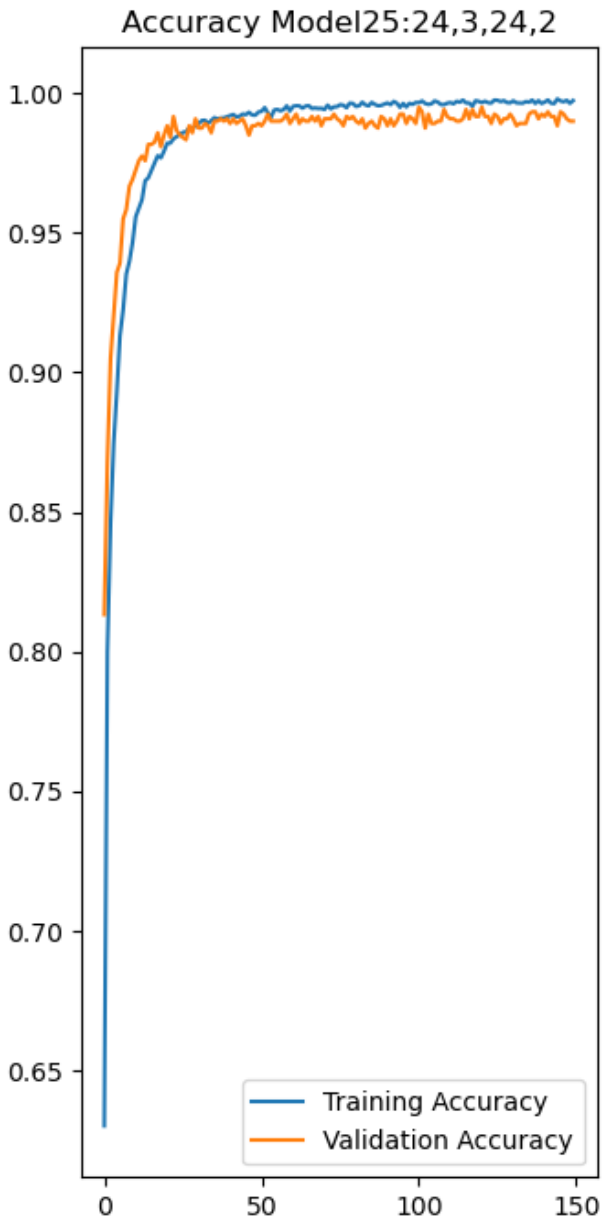

Accuracy Model22:16,7,16,3
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
Training Accuracy
Validation Accuracy

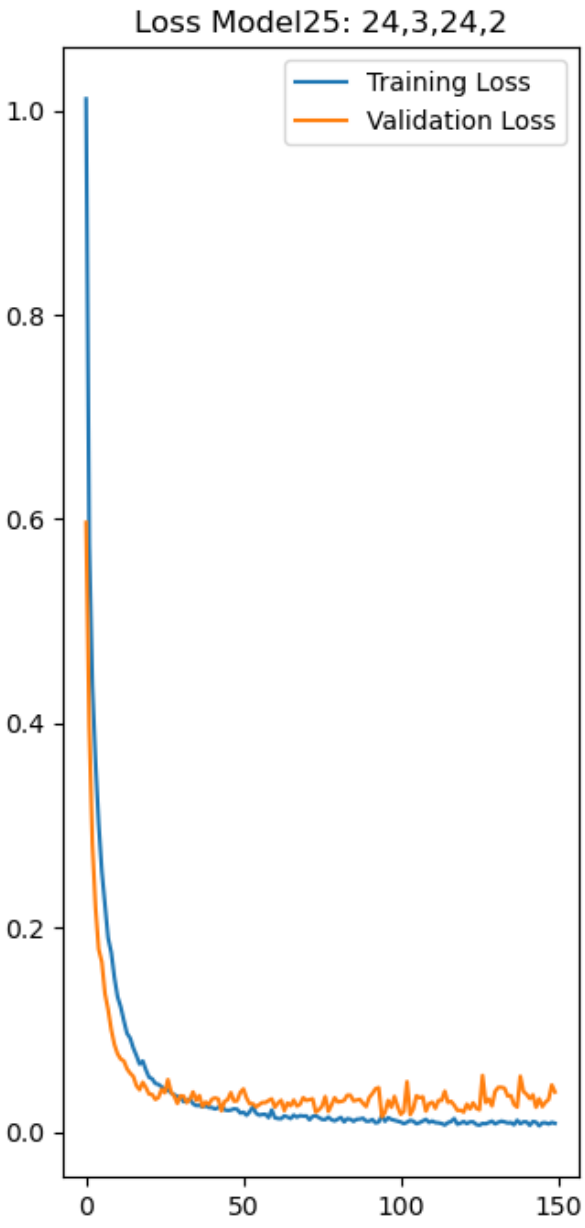

Loss Model22: 16,7,16,3
Training Loss
Validation Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

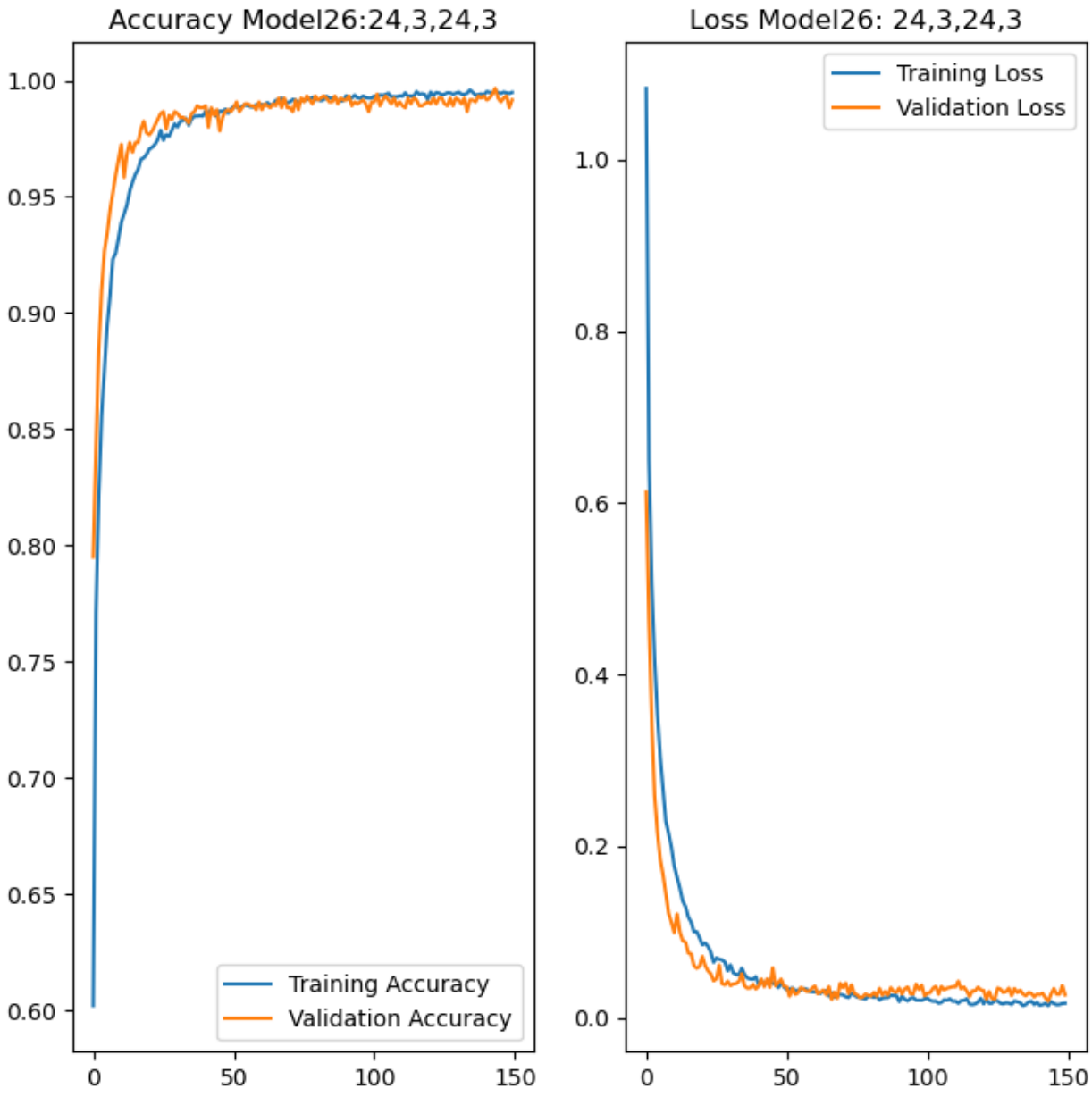

Accuracy Model23:16,7,32,2
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model23: 16,7,32,2
Training Loss
Validation Loss
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

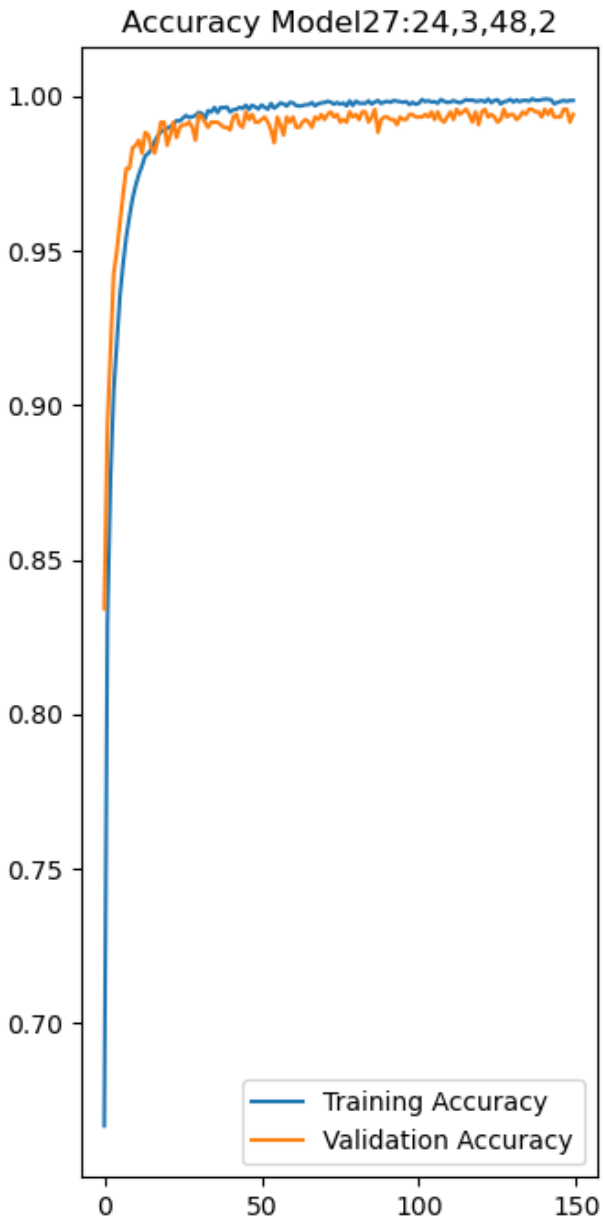

Accuracy Model24:16,7,32,3
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy

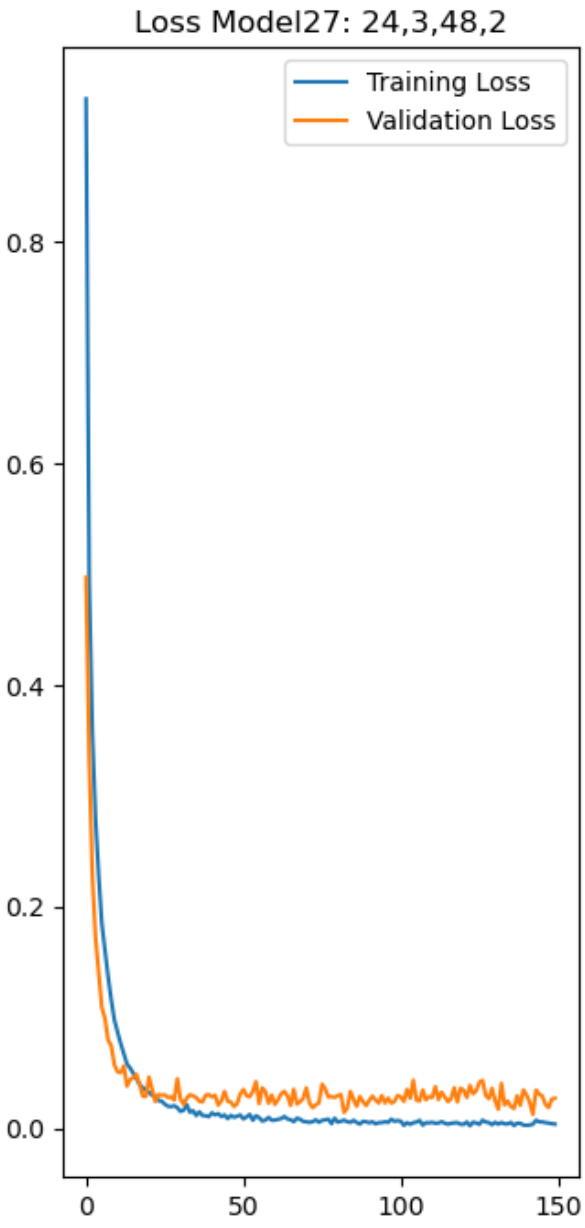

Loss Model24: 16,7,32,3
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

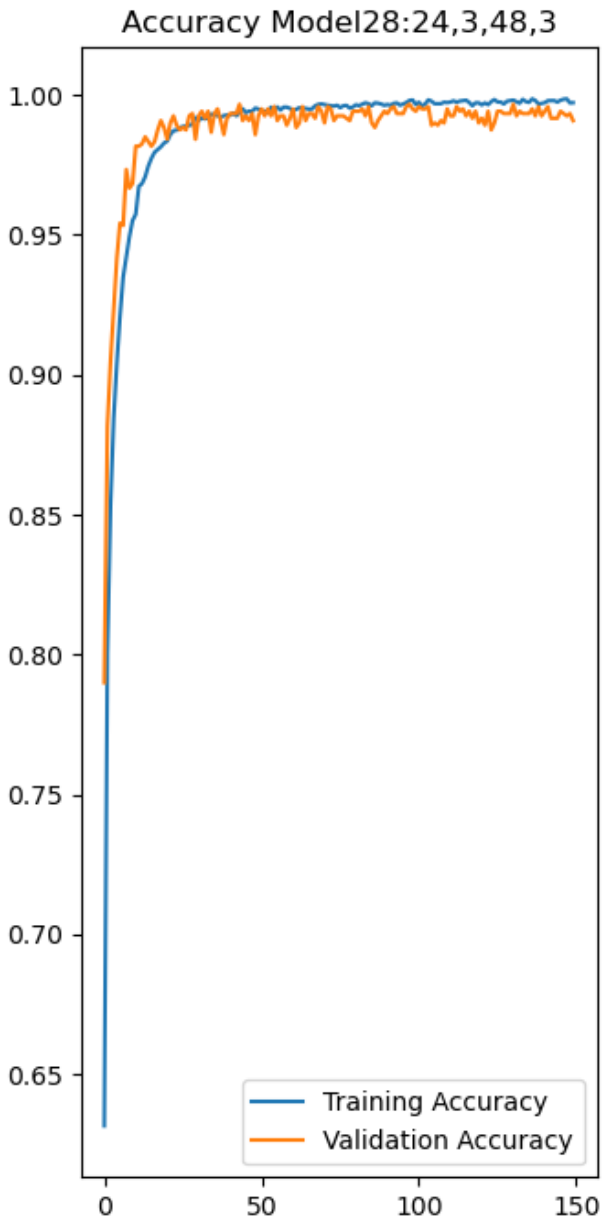

Accuracy Model25:24,3,24,2
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
50
100
150
Training Accuracy
Validation Accuracy

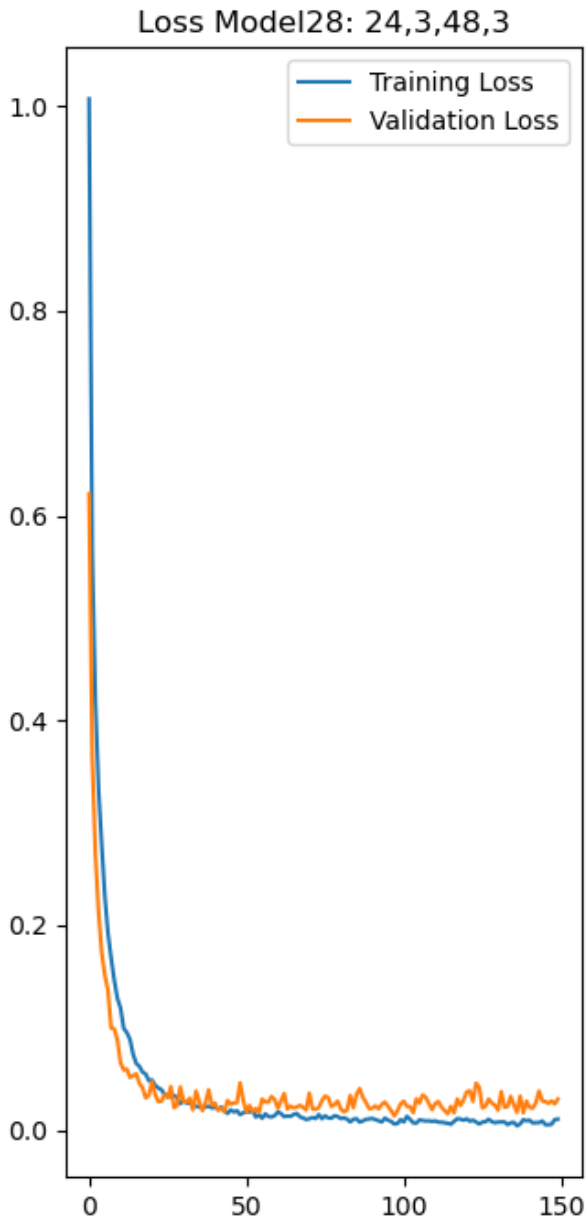

Loss Model25: 24,3,24,2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

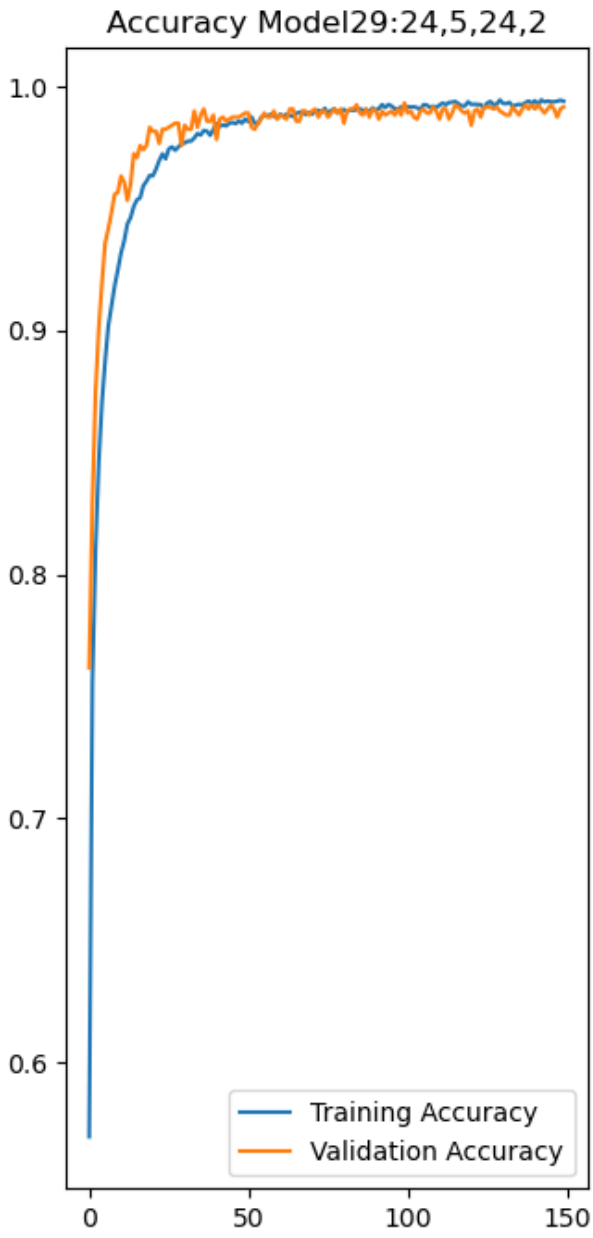

Accuracy Model26:24,3,24,3
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model26: 24,3,24,3
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

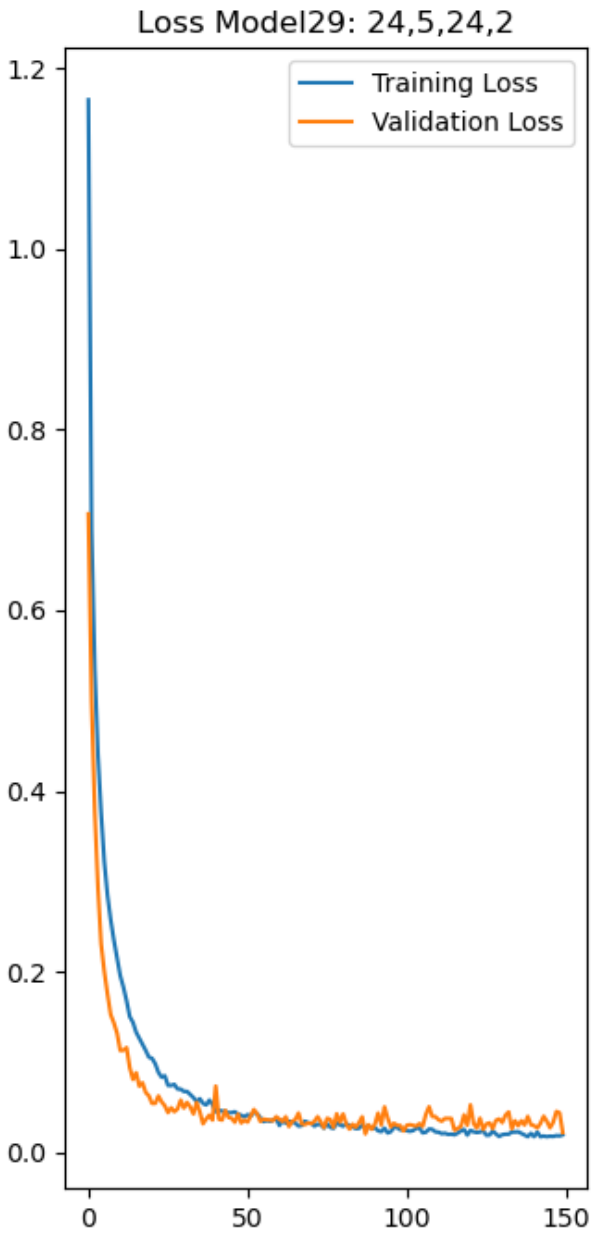

Accuracy Model27:24,3,48,2
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0
50
100
150
Training Accuracy
Validation Accuracy

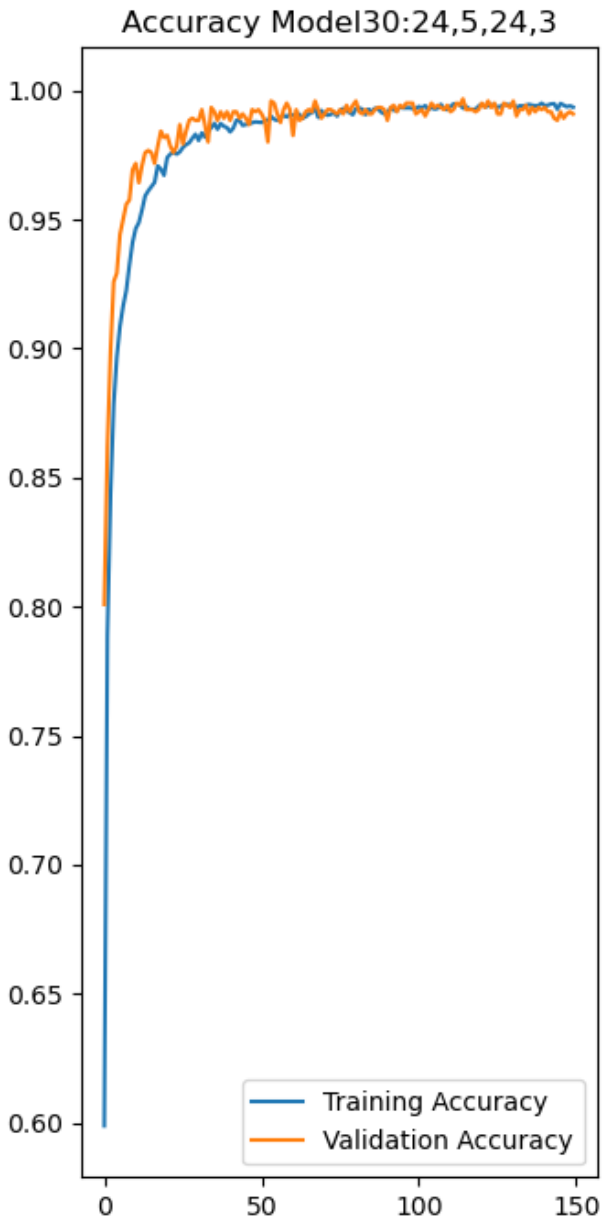

Loss Model27: 24,3,48,2
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

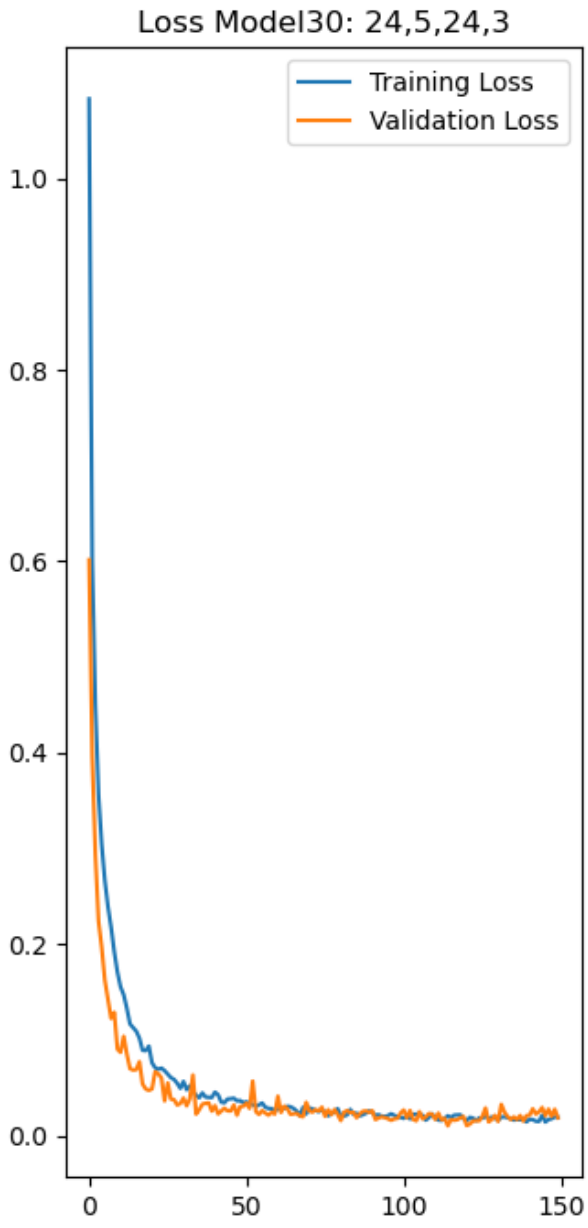

Accuracy Model28:24,3,48,3
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
50
100
150
Training Accuracy
Validation Accuracy

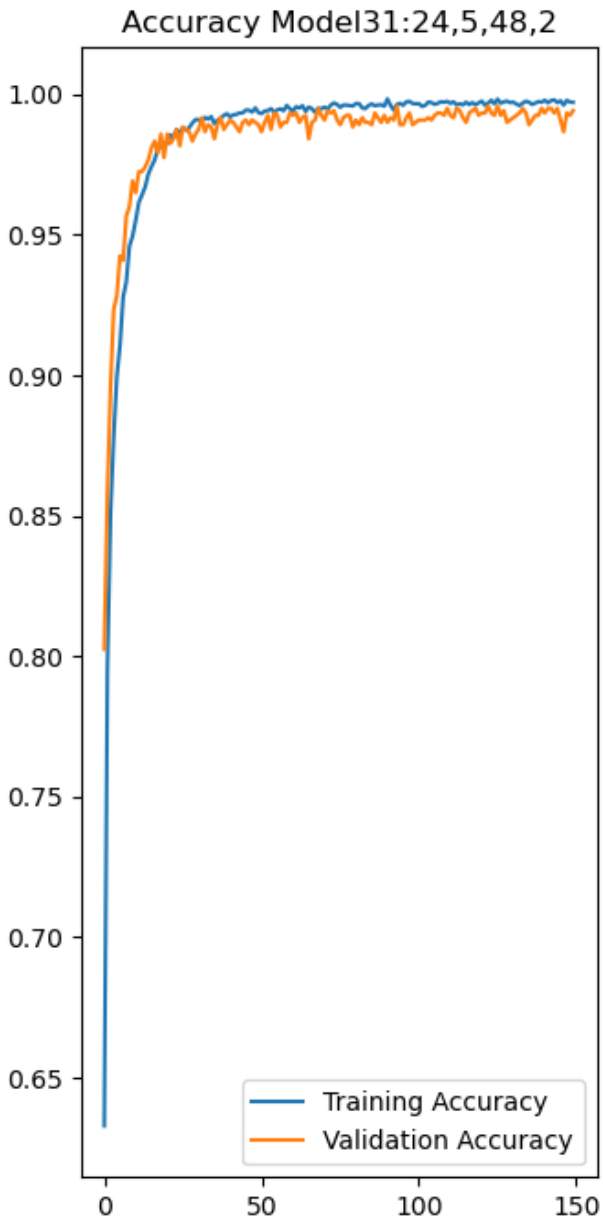

Loss Model28: 24,3,48,3
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

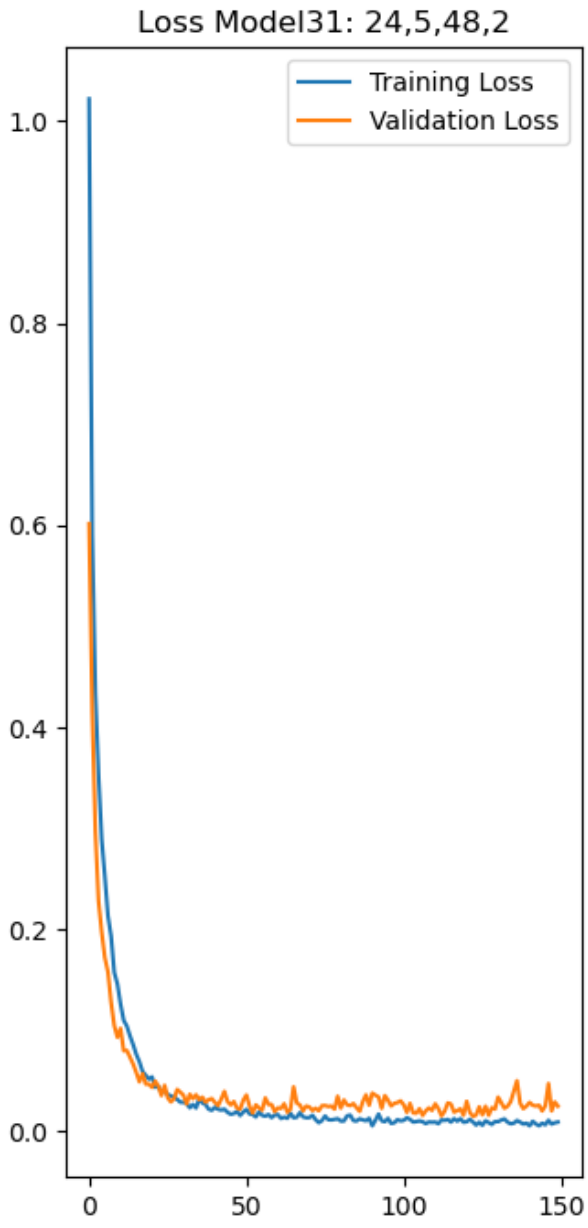

Accuracy Model29:24,5,24,2
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy

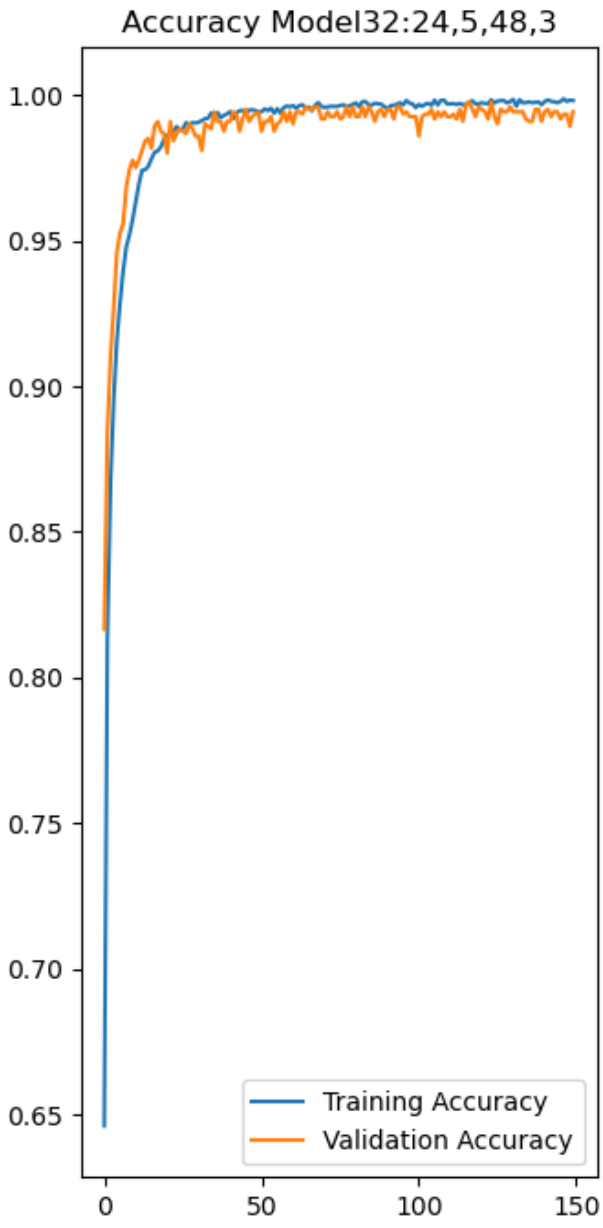

Loss Model29: 24,5,24,2
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

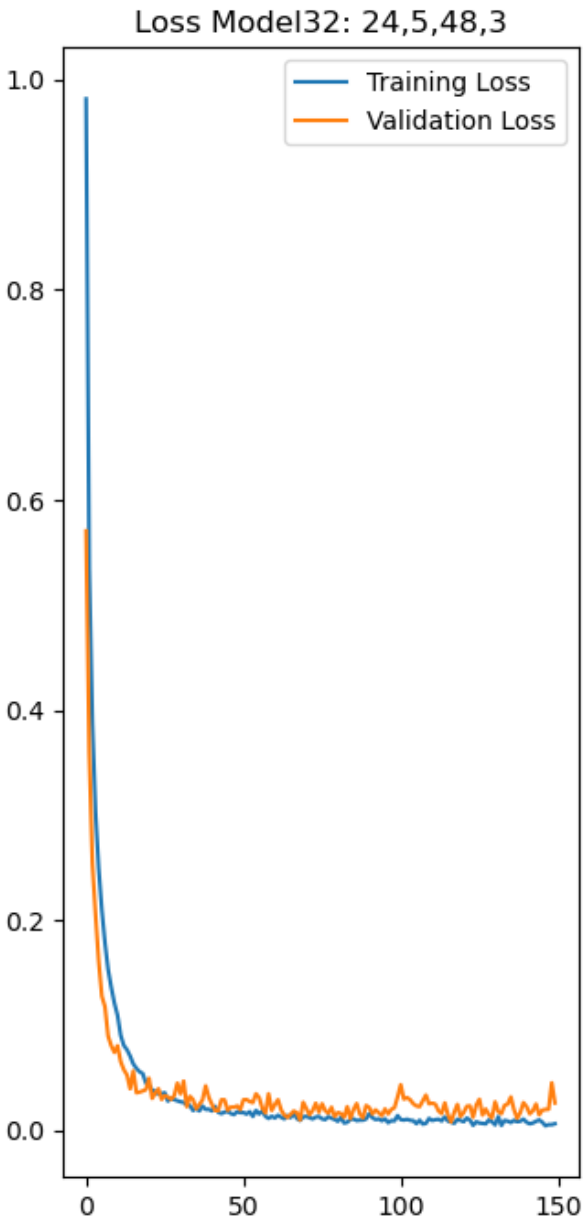

Accuracy Model30:24,5,24,3
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
50
100
150
Training Accuracy
Validation Accuracy

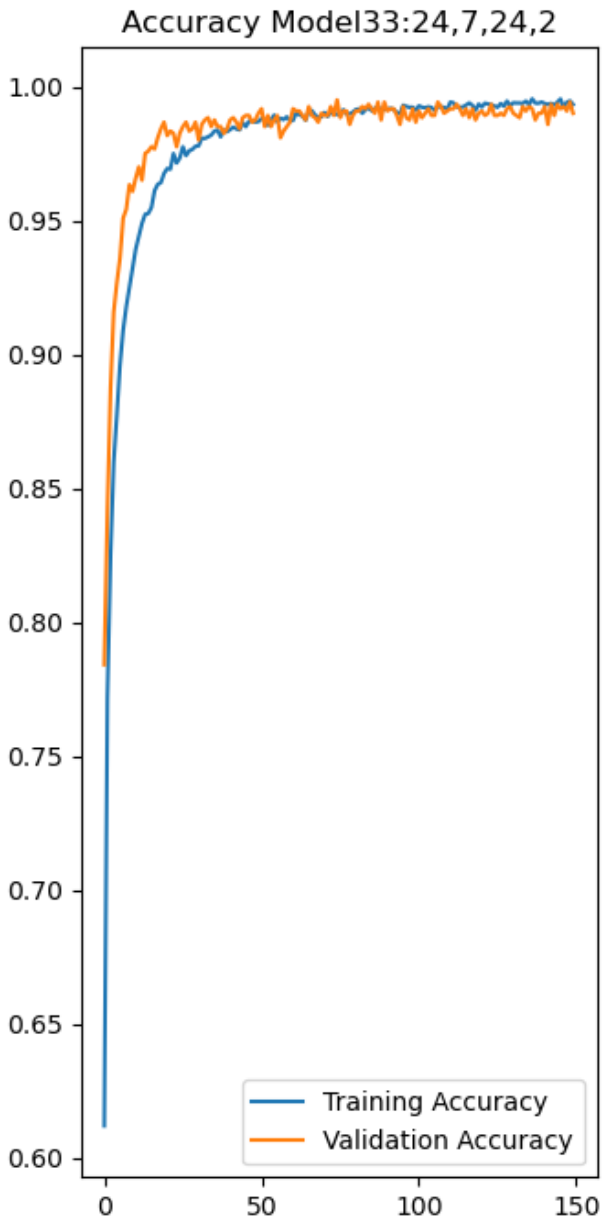

Loss Model30: 24,5,24,3
Training Loss
Validation Loss
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

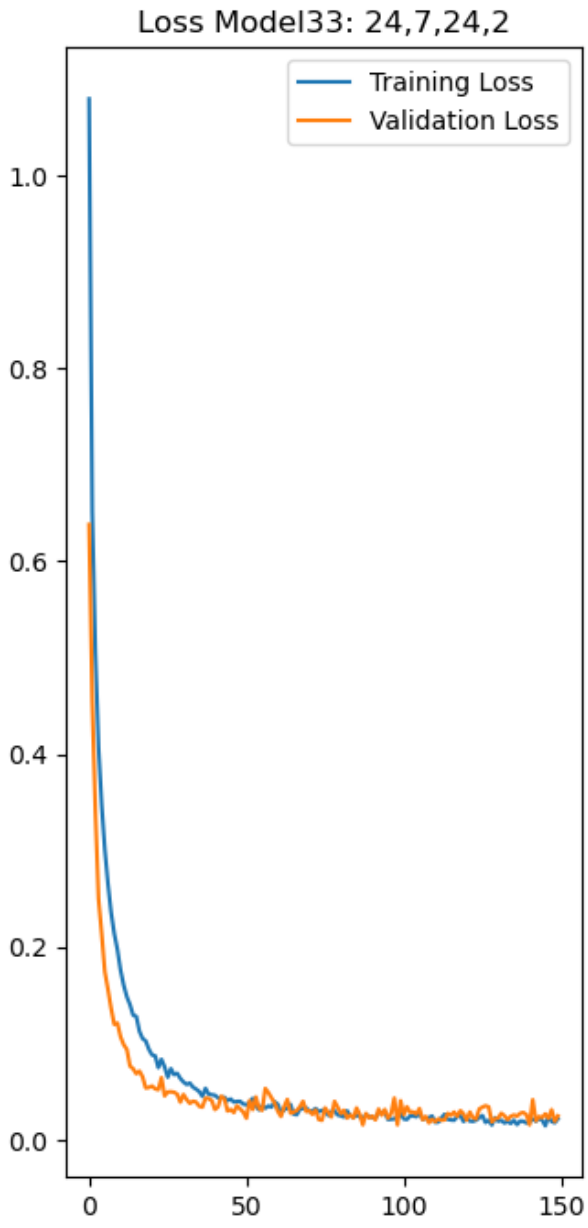

Accuracy Model31:24,5,48,2
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
50
100
150
Training Accuracy
Validation Accuracy

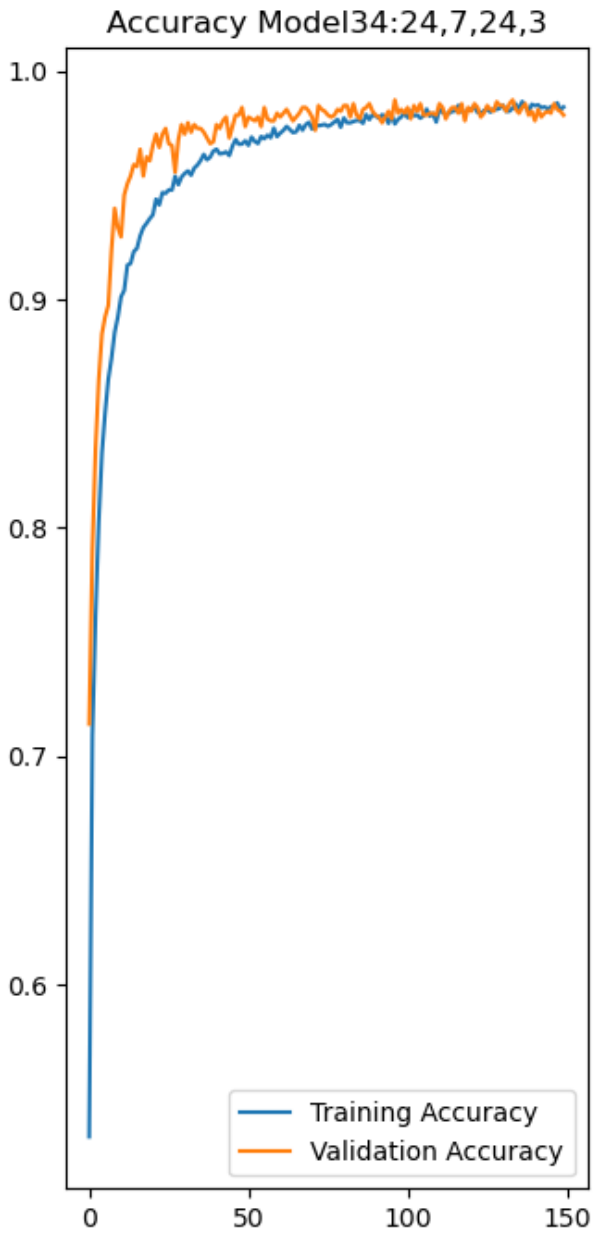

Loss Model31: 24,5,48,2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

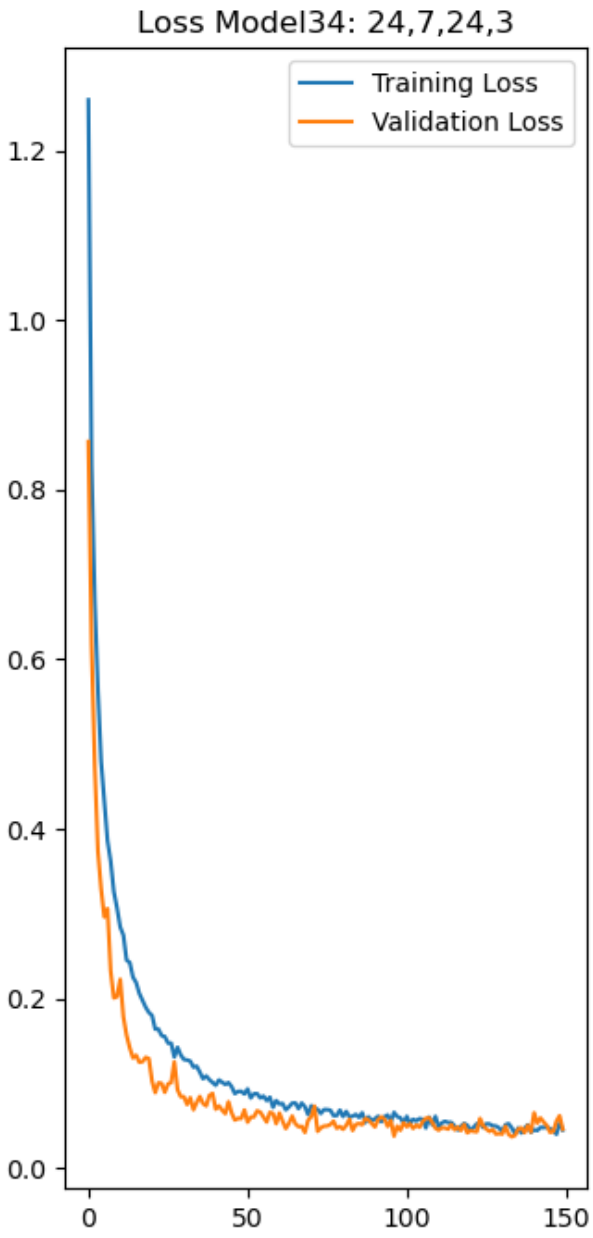

Accuracy Model32:24,5,48,3
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
50
100
150
Training Accuracy
Validation Accuracy

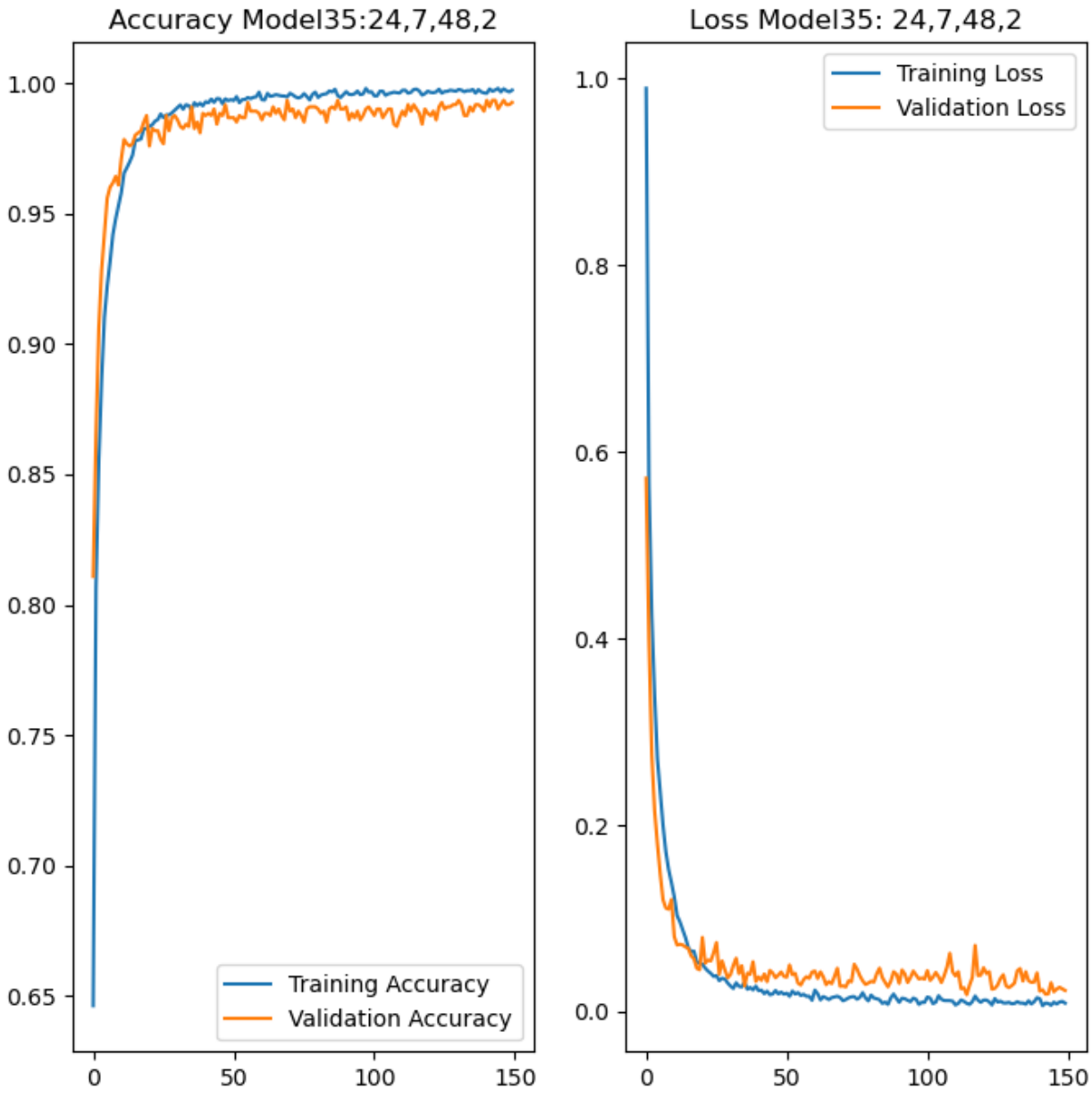

Loss Model32: 24,5,48,3
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss

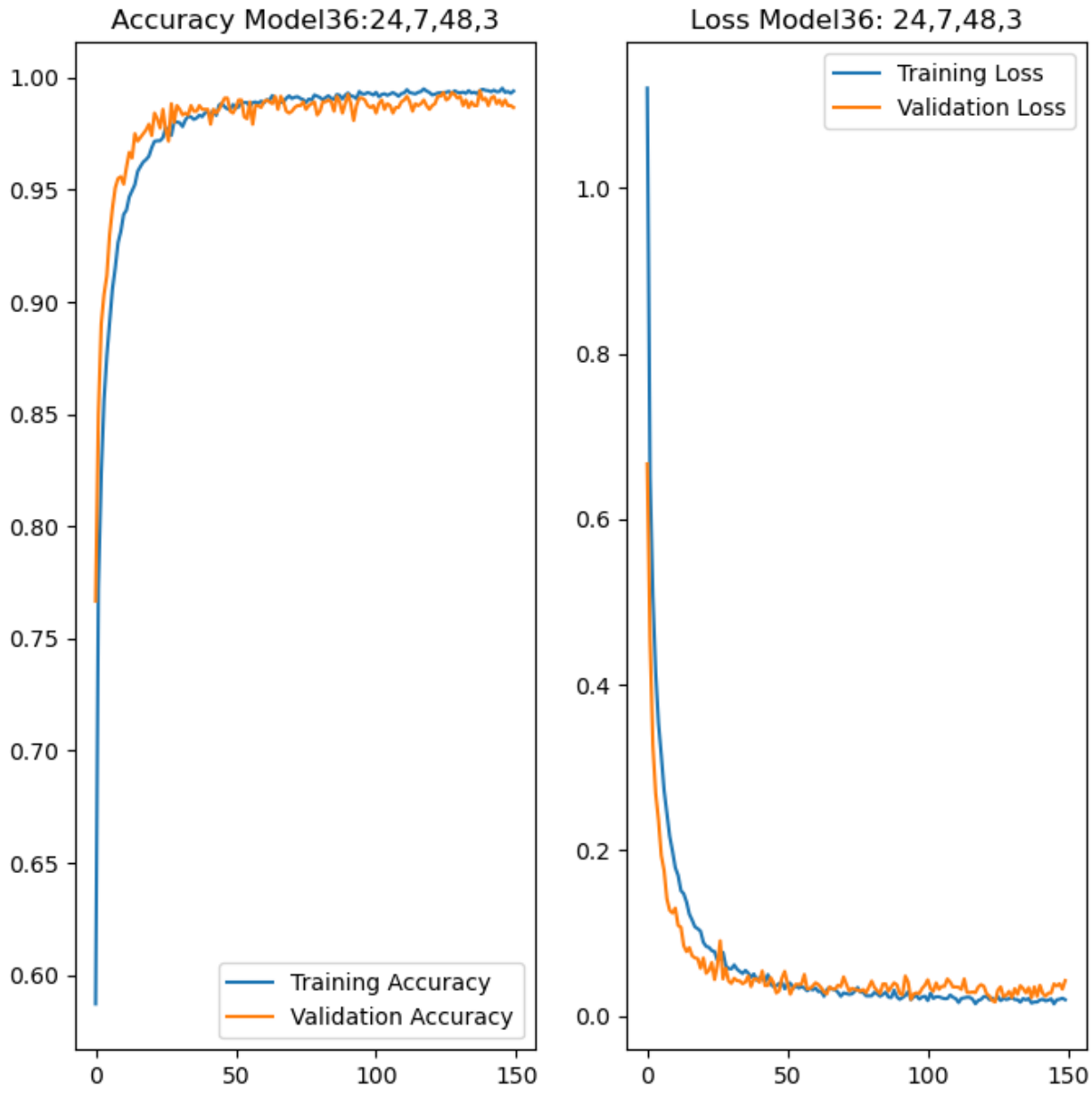

Accuracy Model33:24,7,24,2
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
50
100
150
Training Accuracy
Validation Accuracy


Loss Model33: 24,7,24,2
Training Loss
Validation Loss
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

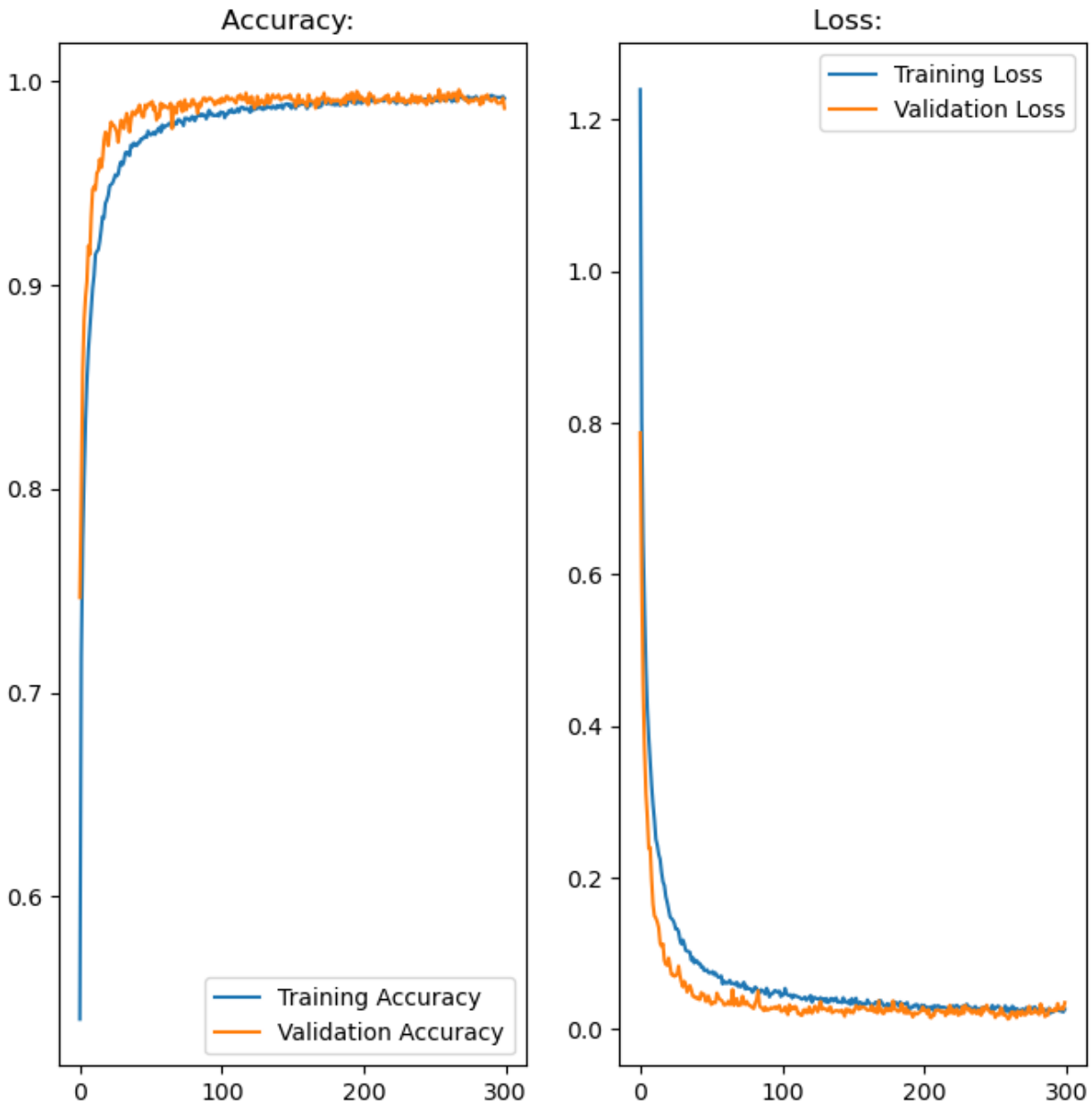

Accuracy Model34:24,7,24,3
1.0
0.9
0.8
0.7
0.6
0
50
100
150
Training Accuracy
Validation Accuracy


Loss Model34: 24,7,24,3
Training Loss
Validation Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150

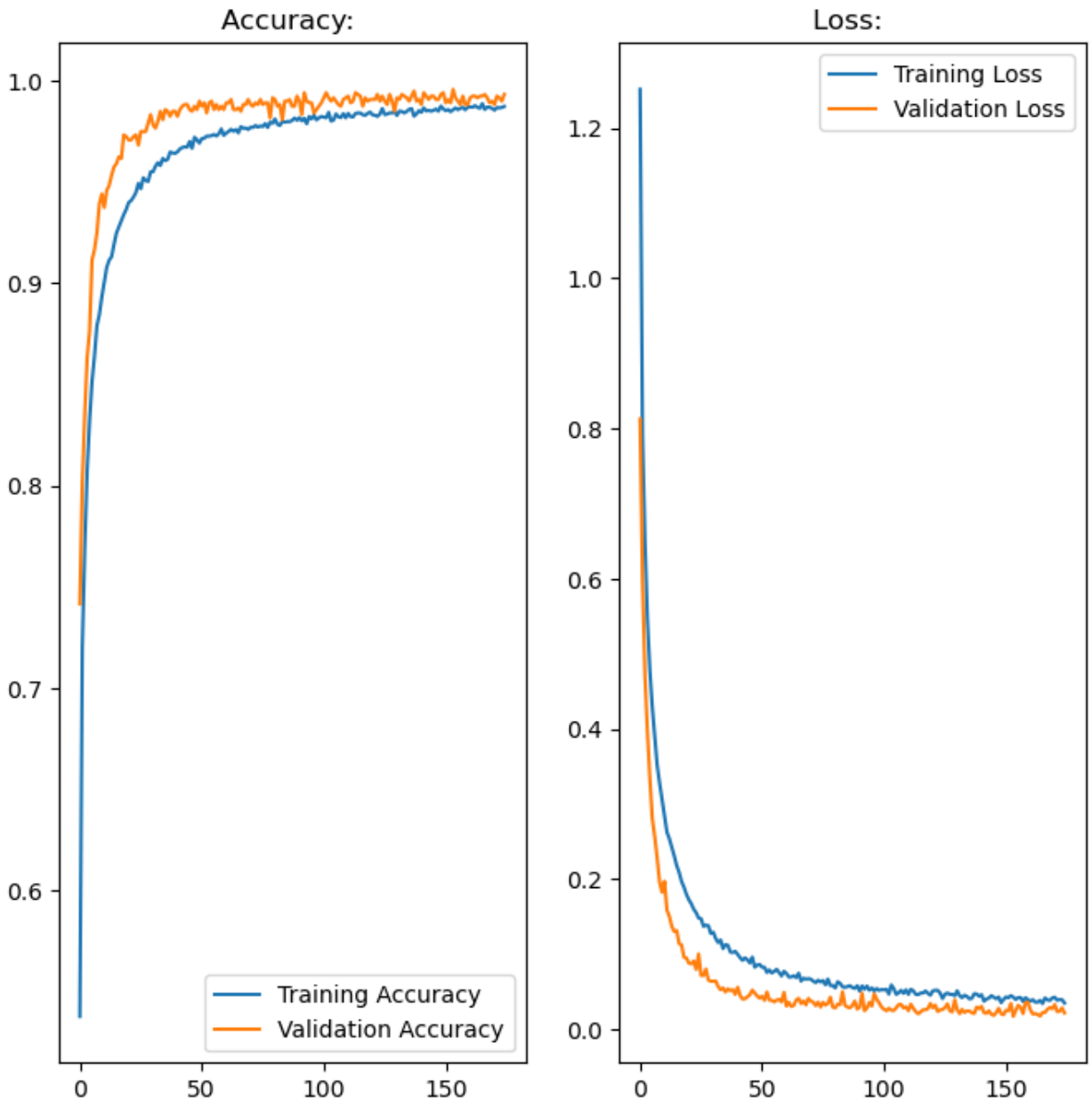

Accuracy Model35:24,7,48,2
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
50
100
150
Training Accuracy
Validation Accuracy
Loss Model35: 24,7,48,2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Training Loss
Validation Loss



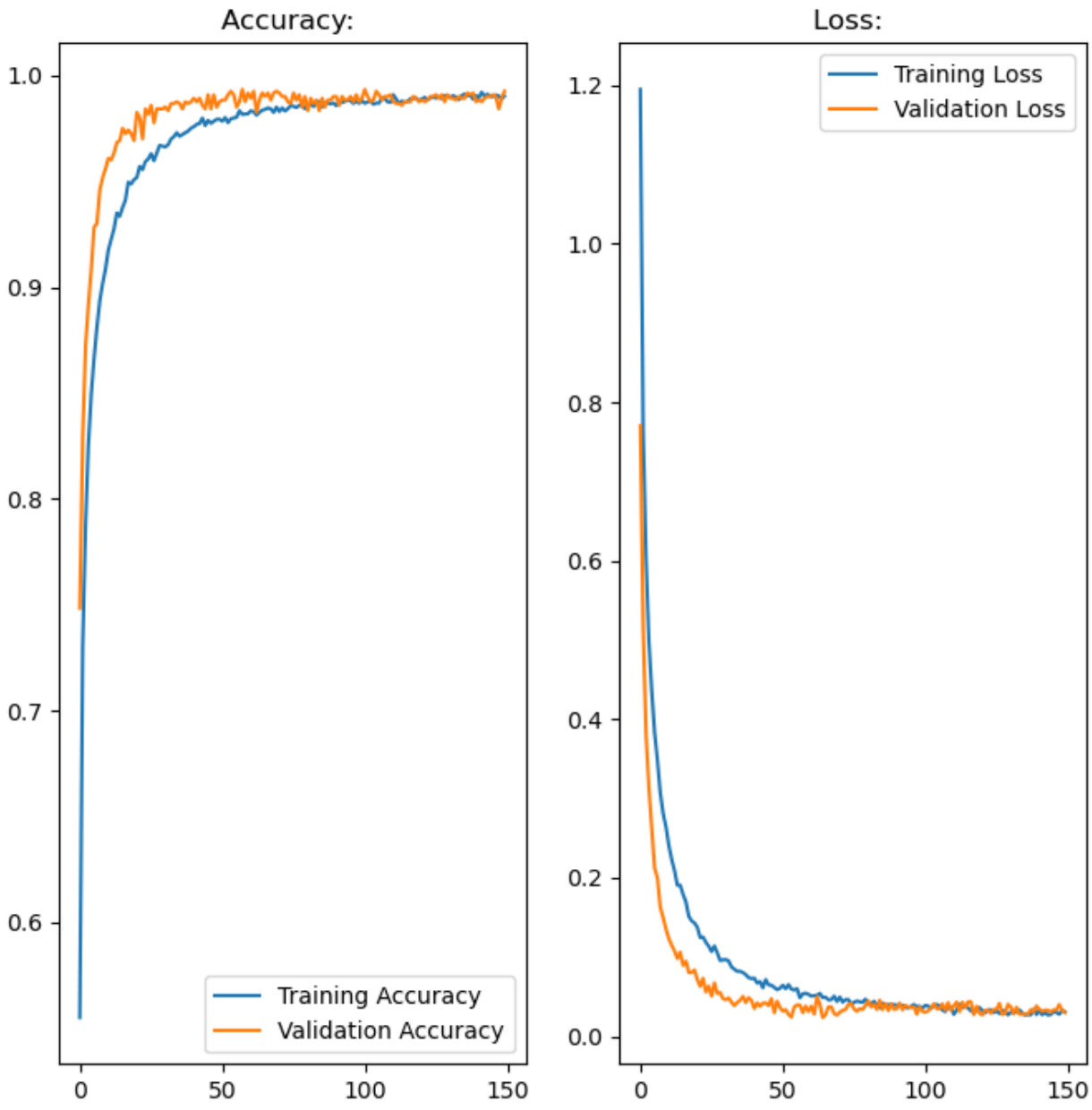
Graphs tracking accuracy, loss for model_opt, model_opt2, model_opt3 and model_optb against the training as well as validation data are included below.



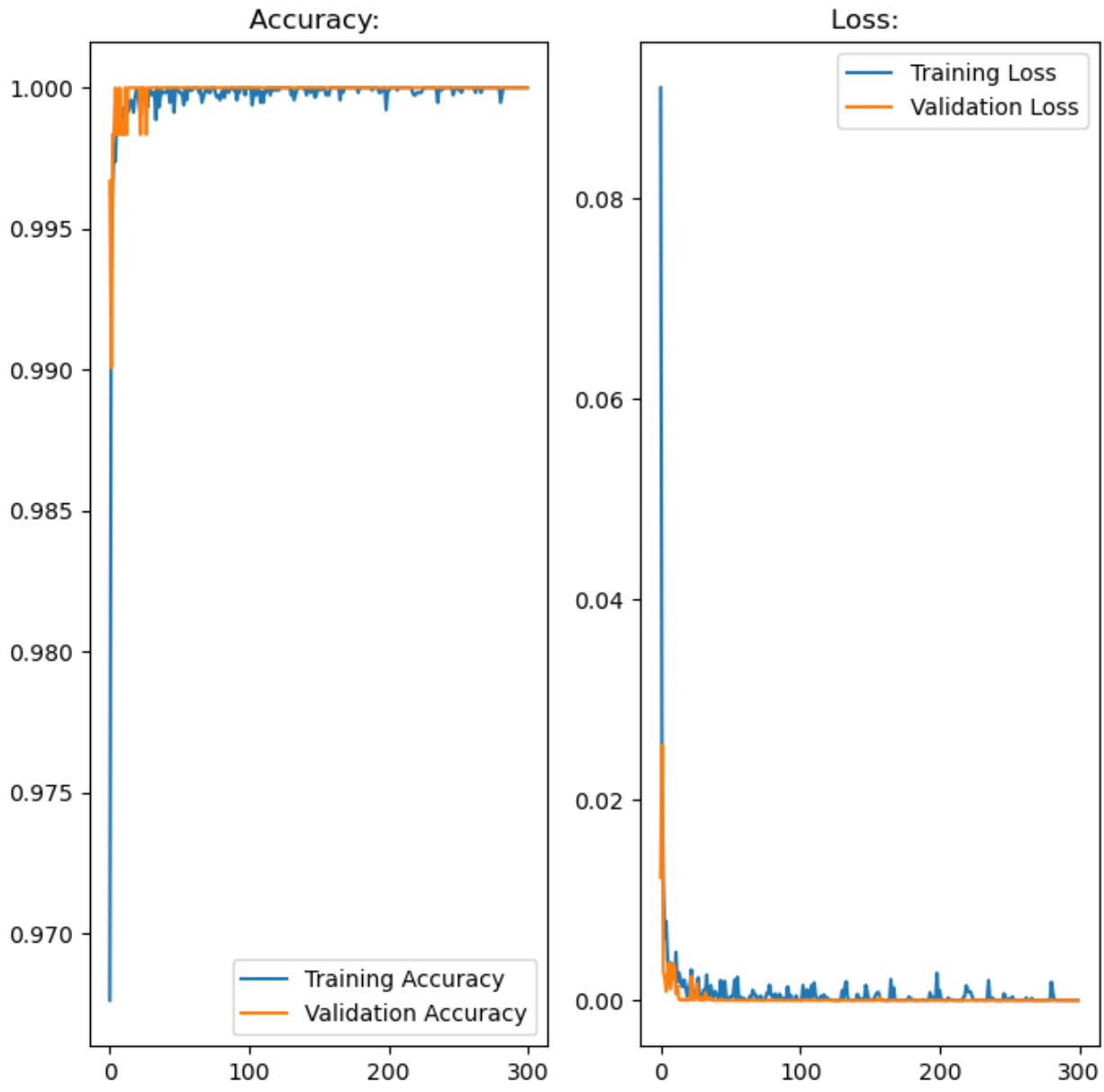
Model_opt train/val accuracy & loss



Model_opt2 train/val accuracy & loss



Model_opt3 train/val accuracy & loss



Model_optb train/val accuracy & loss