Leveraging Neuron Activation Patterns to Explain and Improve Deep Learning Classifiers

1. Training of the Auxiliary Model

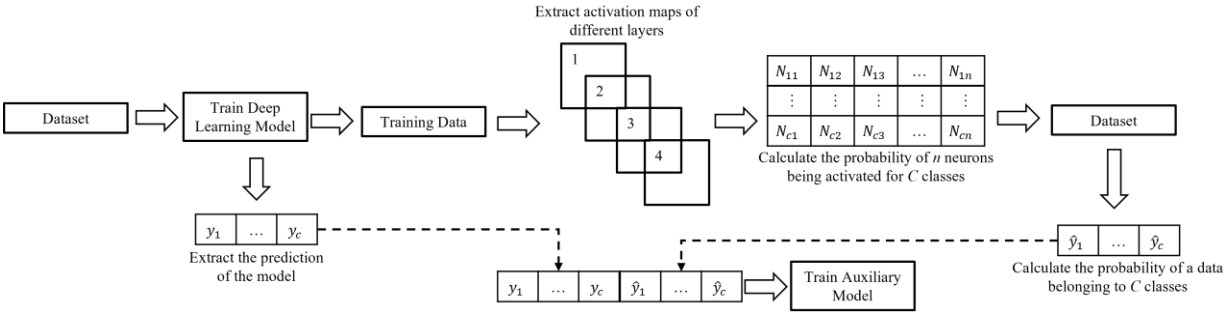

Figure 1: Overview of the proposed performance improvement method and required steps for training the auxiliary model.

We require a trained model to utilize the proposed auxiliary model-based deep learning model's performance enhancement. We start by splitting the data into 80% training data and 20% testing data. We train a model with the training and testing data. Next, we extract the activation maps (neuron activation for FCNN) of the trained model for only the training data. For, CNN we used the activation of the last convolutional layer, and for FCNN, we used the activation of the last hidden layer. Next, we use the activation maps to calculate the probability of activation of different neurons for different classes. Then, using the activation probability matrix we calculate the activation probability vector, i.e., the probabilities of a data belonging to different classes. Next, we calculate the activation probability vector for the dataset. Finally, we combine the activation probability vector and the prediction of the trained model to train the auxiliary model. In the auxiliary model, we use the same training and testing data as the trained model.

Table 1: Details of the architecture of the deep learning models for different datasets.

Dataset	Type of Model	Image Size	Number of Classes	Iterations
MNIST	FCNN	28 × 28	10	20
MNIST Mixed		28 × 28		
Fashion MNIST		28 × 28		
Fashion MNIST Mixed		28 × 28		
CIFAR-10	FCNN	32 × 32		40
	VGG-16	112 × 112		
	ResNet-50	64 × 64		
CIFAR-100	InceptionV3	224 × 224	100	40
Plant Village	FCNN	256 × 256	9	100
	ResNet-50	112 × 112		40

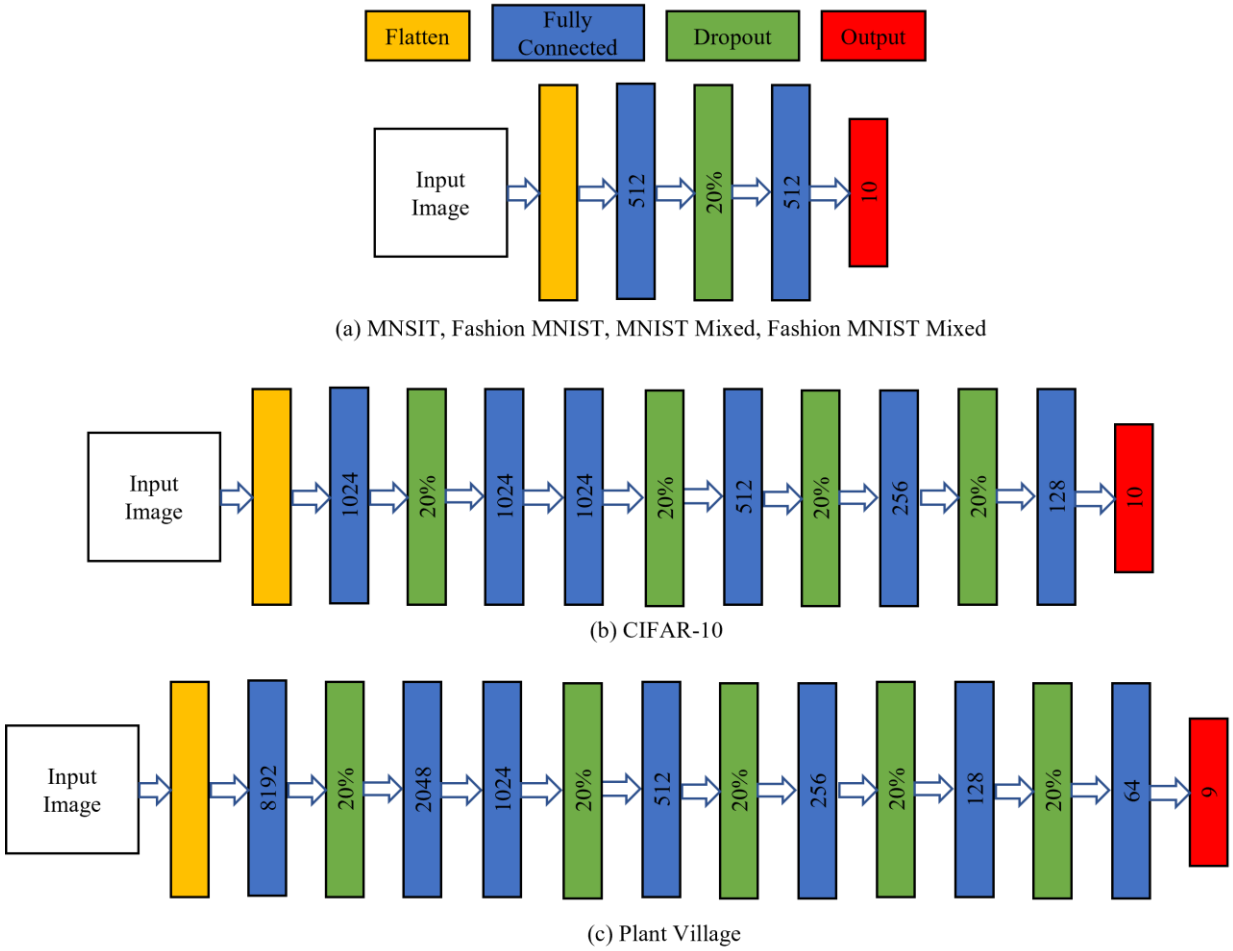

Figure 2: Graphical Representation of the FCNNs used in the study.

Table 2: Pearson correlation coefficient between the entropy and testing accuracy. We used FCNN for MNIST, MNIST Mixed, Fashion MNIST, Fashion MNIST Mixed, ResNet-50 for CIFAR-10 and Plant Village, and InceptionV3 for CIFAR-100. Darker blue represents higher values, and darker red represents lower values.

	MNIST	**MNIST Mxied**	**Fashion MNIST**	**Fashion MNIST Mixed**	**CIFAR-10**	**CIFAR-100**	**Plant Village**
PCC	-0.79	0.17	-0.93	0.35	-0.64	-0.51	0.29

Figure 3: Change of normalized entropy, training and testing accuracy for different classes of the MNIST dataset over different iterations.

Figure 4: Change of normalized training and testing accuracy, and entropy of the MNIST Mixed dataset over different iterations for different classes using FCNN.

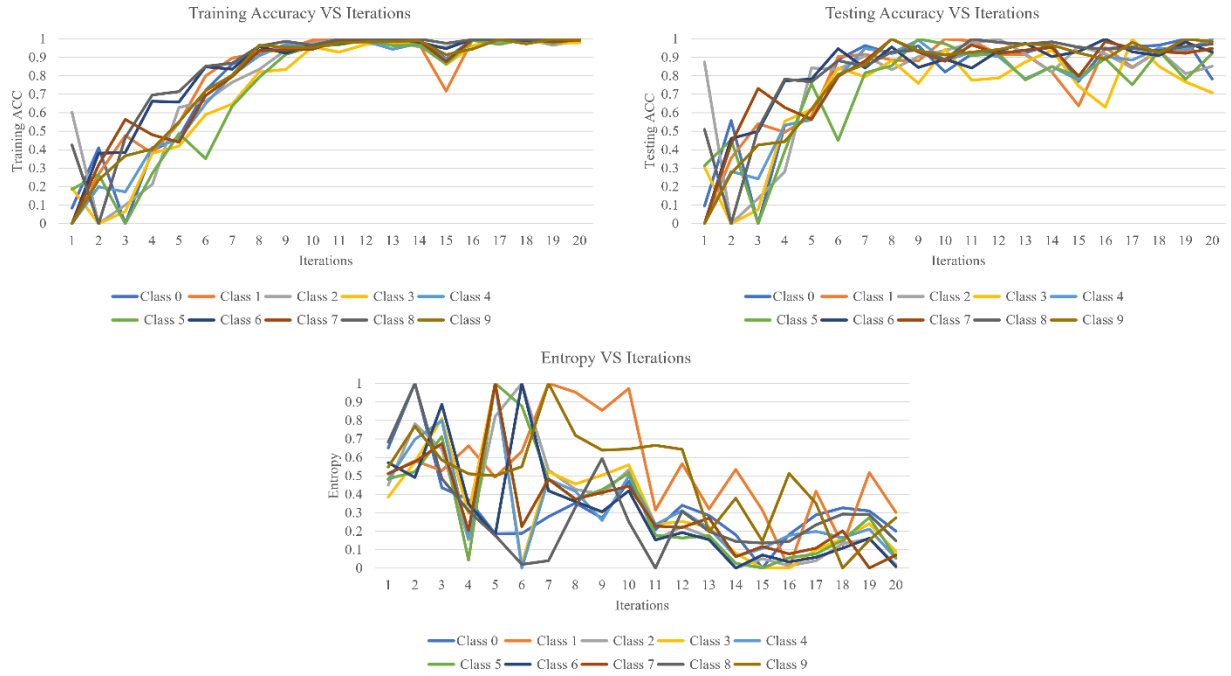

Figure 5: Change of normalized training and testing accuracy, and entropy of the CIFAR-10 dataset over different iterations for different classes using the ResNet-50.

Table 3: Spearman and Pearson correlation coefficient between entropy and training accuracy for individual classes. We used FCNN for MNIST, MNIST Mixed, Fashion MNIST, Fashion MNIST Mixed, CIFAR-10 FCNN, Plant Village and ResNet-50 for CIFAR-10 ResNet-50. Darker blue represents higher values, and darker red represents lower values.

MNIST		**MNIST Mixed**		**Fashion MNIST**		**Fashion MNIST Mixed**		**CIFAR-10 FCNN**		**CIFAR-10 ResNet-50**		**Plant Village**	
PCC	SCC	PCC	SCC	PCC	SCC	PCC	SCC	PCC	SCC	PCC	SCC	PCC	SCC
-0.6	-0.7	-0.1	-0.3	0.1	0.2	-0.1	-0.4	0.2	0.2	-0.5	-0.3	-0.9	-0.9
-0.5	-0.7	0.2	-0.2	-0.2	-0.5	0.8	-0.2	0.7	0.7	-0.1	-0.4	-0.9	-0.9
-0.2	-0.3	0.5	0.0	0.4	0.2	0.5	0.3	0.4	-0.1	-0.6	-0.6	-0.5	-0.4
-0.4	-0.7	0.4	0.1	-0.6	-0.8	0.3	-0.2	0.1	0.2	-0.6	-0.6	-0.9	-0.9
-0.7	-0.7	0.3	-0.1	-0.4	-0.3	0.5	0.1	-0.4	-0.5	-0.6	-0.5	-0.4	-0.3
-0.6	-0.5	-0.1	-0.1	-0.7	-0.7	0.3	-0.2	-0.5	-0.5	-0.6	-0.6	-0.9	-0.9
0.0	0.2	0.2	0.4	-0.8	-1.0	0.7	0.2	0.6	-0.2	-0.6	-0.7	-0.9	-0.9
-0.1	-0.3	0.1	0.3	-0.4	-0.6	0.7	0.3	0.2	0.2	-0.6	-0.7	-0.9	-0.9
-0.7	-0.8	0.5	0.6	-0.9	-1.0	0.6	0.3	0.0	0.1	-0.8	-0.4	-0.8	-0.9
-0.4	-0.8	0.1	0.0	-0.7	-0.6	0.9	0.9	-0.2	-0.3	-0.3	-0.4	-	-