Assignment 2: Computer Vision on CIFAR-10 Data Set

Gavyn Gallagher

Northwestern University

MSDS 458: Artificial Intelligence and Deep Learning

Syamala Srinivasan

October 9, 2022

Abstract

Company X is seeking to implement a facial recognition login on their application. Facial recognition requires a technology that can distinguish and classify faces. Computer vision using CNN and DNN models can be used to classify images. The data set CIFAR-10 was chosen for Company X to create a model on. If the model is proven to be successful then Company X will use the model for the facial recognition login. In 5 experiments, 10 models of various parameters were created. CNNs outperoformed DNN models. The highest accuracy of a model was 77.26%. This was deemed not high enough to be considered a success. Recomddations is that Company X continue building new CNN model of different parameters and number of layers. Also to work on a dataset that includes human faces. Company X will not have a facial recognition until a proven model is presented.

Introduction and Problem Statement

Our Company X has developed an application (app) that currently uses a four digit key code to log into. In the future, Company X seeks to have the users login with facial recognition technology. This would allow for ease of the user as they do not have to perform an action to login. Another added benefit would be security, as it prevents hackers from logging into users accounts. Only the user, with their unique face would have the ability to login to their account. Company X asks for a model to be created on an image data set prior to implementing the facation region feature into their app.

The Canadian Institute for Advanced Research 10 (CIFAR-10) data set has been selected by Company X to test computer vision on. 60,000 images of an equal sized 10 classes are included in the CIFAR-10 data set. Classes include : airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Neural network models will be used to classify these 60,000 images into the correct class. If deemed successful the best model can then be used on Company Xs' app for their facial recognition login.

In this assignment 5 experiments were conducted. A total of ten models were created throughout the 5 experiments. Regularization is not used in the first 4 models and then used in the remaining 6 models. Experiment 1 is a dense neural network, "model1" of 2 layers. Similarly Experiment 2 is a dense neural network, "model2", but with 3 layers. Experiments 3 and 4, "model3" & "model4", use convolution neural networks. With model3 having 2 convolution and max pooling layers and model4 having 3 convolution and max pooling layers. Model5 is a copy of model1 with regularization applied. Then model6 is a copy of model2 with regularization. Model7 and model8 are copies of model3 and model4 with regularization. Model9 uses 2 convolution layers, a larger number of filters in the output layer and a larger stride size. Lastly model10 uses the same stride size as models1-8 but used a smaller drop out rate. All models with regularization use the same L2 rate of 0.001 due to time constraints.

Each model will be evaluated on its performance. In the data preparation, training and testing data will be split with 50,000 images in training and 10,000 images in testing. A confusion matrix will be produced to see how each models predictions are compared against the test data set. Metrics used in evaluation of the model will be accuracy and root mean squared error. Accuracy and loss will each be shown in line graphs displaying their values over each iteration of the neural network model. Each model's confusion matrix will be displayed along with a t-Distributed Stochastic Neighbor Embedding (T-SNE) plot. Recommendations on what Company X should be included in the conclusion of the paper. After this assignment Company X will have its first start at testing for a facial recognition login.

Literature Review

How to Develop a CNN From Scratch for CIFAR-10 Photo Classification

Jason Browlee uses computer vision on the CIFAR-10 data set (Brownlee, 2019). The goal of the model is create a model that can accuracy classify all the images into the 10 classes. First the author starts out with a baseline model then creates other models that are more complex.

In the data preparation the images were also rescaled by dividing htem by 255. The baseline model was a CNN with 6 convoutional layers 3 max pooling layers. A learning rate of 0.001 was used. All of the acitrivations use relu and the optimizer was SGD.

Through this article, it is learned to start off with a basic model and then make more models to compare them to the baseline. Also that multiple convoulational layers can be ran in a row before a max pooling layer. This article used CNN and achieved 73% accuracy.

Classification of Image using Convolutional Neural Network (CNN)

This article by Anware Hossain and Shahriar Alam Sajib also focusses on using CNN models on the CIFAR-10 data set. The goal of the article is to explain how CNN models work as well as create a successful model in classifying the CIFAR-10 images. Also MatConvNet is explored.

The arthcieteure of CNN and how each layer works is explained in the article. MatConvnet, a CNN for MATLAB is introduced and explained. The paratmers of the model are chosen and the reasoning behind htem. For example, the learning rate was set to 0.0001. Ultiamely a model was created that had 93% accuracy.

From this article, understanding of how each layer operates is learned. Knowing how much of each type of model to have and the order can lead to the success of an accurate model. Lastly, it was proved that a CNN model can be built that achieve high accuracy.

Method(s)

Data Preparation was performed on the CIFAR-10 data set. Of the 60,000 total images, 10,000 was randomly assigned to testing data. While the remaining 50,000 would be used as training data for the models. All images are a size of 32 by 32. Next Images were rescaled and all divided by 255.

Model methods included two types of neural networks: dense neural networks(DNN) and convoulational neural networks(CNN). In the dense layer all inputs are used to form the output, while in the convultional layer only a small amount of the inputs are used (Haque et al., 2022). Regualzation was also used in six of the ten models, is it used for simplifying the model and combatting overfitting (Haque et al., 2022). Using L2 regularization and dropout was used for regularzation in the models. Also, in the CNN models max polling layers were used to grab the more important features and ignore the others. Lastly, batch normalization was used in the CNN models to normalize the output of previous layers.

For evaluating the models a variety of charts were produced. Line charts of accuracy and loss shows how the model did at each iteration of the models fit. Accuracy and Root mean square error scores for the test data are recorded as well as the amount of time it took to fit the model. Confusion matrices compares the predicted classes against the truth and how many of each class was correct/inocrrect. Valuable insights such as which classes are being confused for eachother will be noted. Lastly a T-SNE plot was used to visualize how the data points cluster.

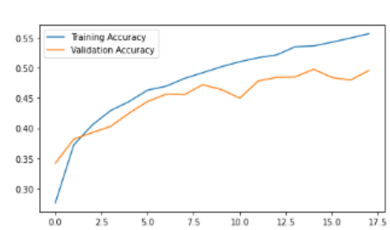
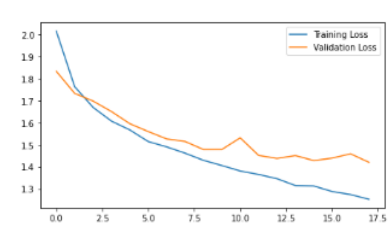
Results

Experiment 1

First experiment, uses Model1 to classify the data sets images. What distinguishes Model1 from other models is that it uses DNN, 2 hidden layers, and no regularization. Evaluation of all models will be expressed by accuracy and loss scores, confusion matrix, and a T-SNE Plot.

Accuracy and loss for both the training and the validation data was recorded for each epoch ran. Model1 ran for 18 epochs after which the results flattened and the early stop functionality stopped the model from continuing to run. The loss starts and accuracy starts off poor but gradually becomes medium at the end.

Figure 1
Model1 Accuracy and Loss Graphs

Model Name	Accuracy	Loss
Model1		

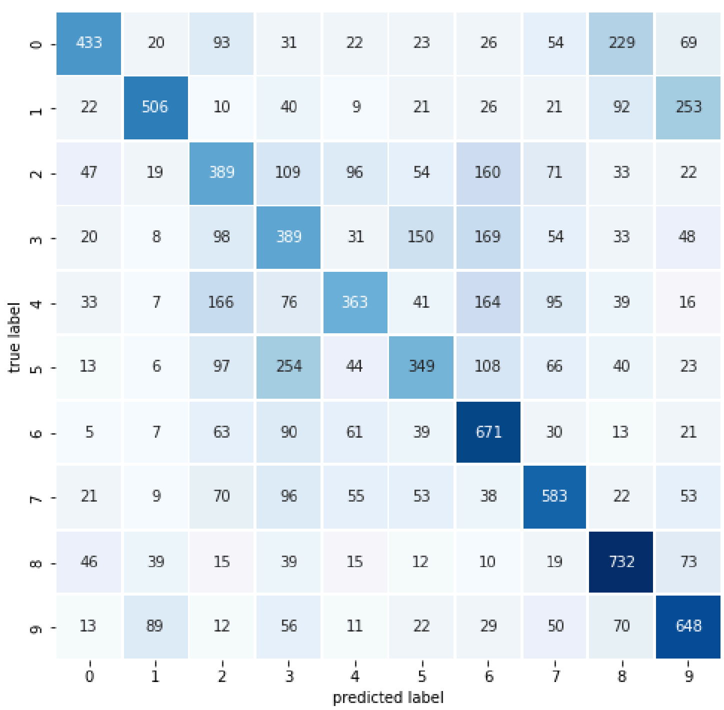
Next, model 1s' predictions were tested against the observed data or truth. Scores can be seen below in figure 2. The layout of the table in figure 2 will be used for other DNN models, its field will help distinguish one model from another. Accuracy of the model 1s; predictions were medium to poor at .5063. The Root Mean Square Error was equally medium to poor.

Figure 2
Model1s' Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Regularization	L2	Process Time
Model1	0.5063	3.0801	DNN	2	No	-	14.94

Looking further into Model1s accuracy a confusion matrix was produced, shown in figure 3. Ideally there should be a dark blue downward diagonal line left to right with all other squares displaying a white color. In all confusion matrix figures there will be a key to the right of it. Many of the squares in model1s' confusion matrix have a light to medium blue representing misclassification. Examples of model1s' misclassifying include dogs as cats, airplanes as ships, and automobiles as trucks.

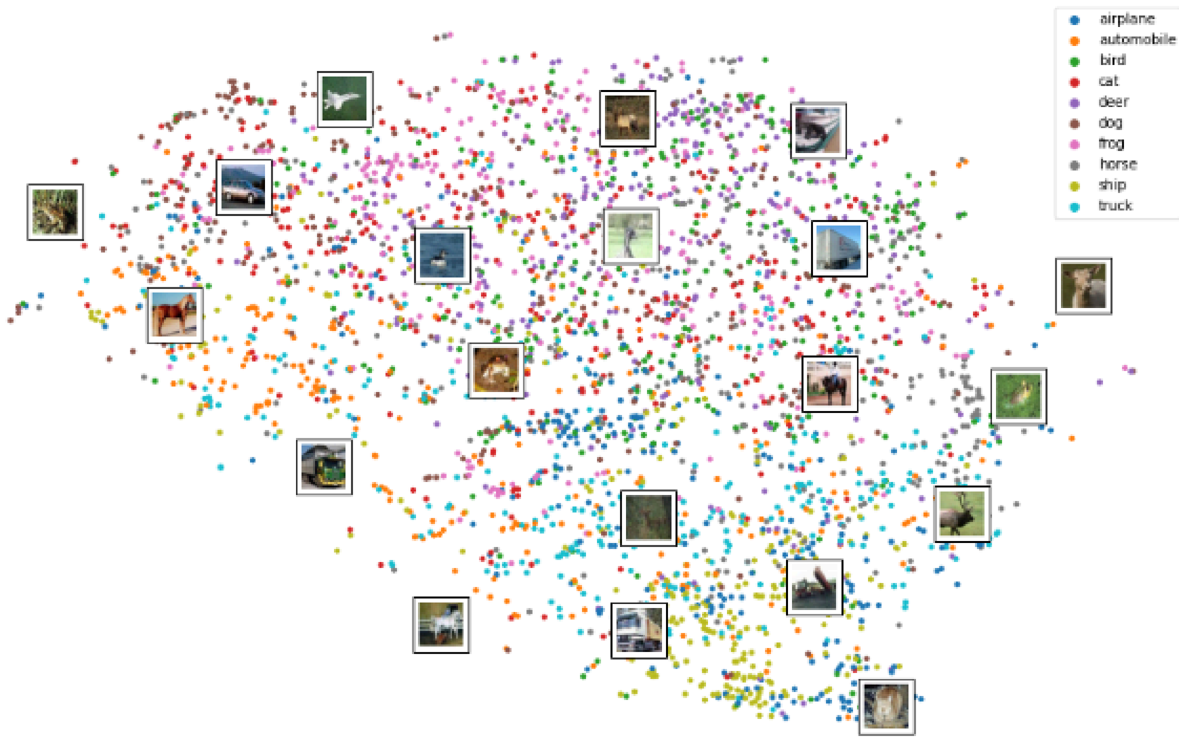
Figure 3
Model1s' Confusion Matrix



Label	Class_
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

The last way we will be evaluating model1 and other models is through a T-SNE plot. Each data point The images displayed are representing the data that is near. There are more images than classes for there is an image generated at a specific interval of distance. Figure 4, shows that there are multiple images of trucks, horses, frogs, and deer. Following the chart by color, the range of distance between images from the same class are widespread. THere is a cluster of ships near the bottom of the chart, meaning the model did cluster ships well.

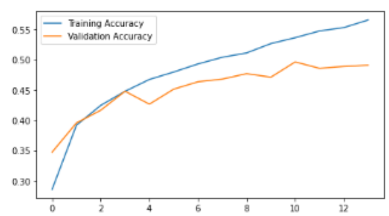
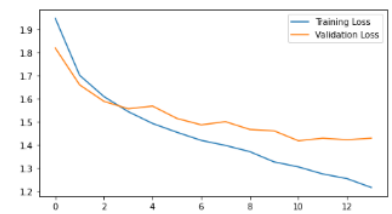
Figure 4
Model1s' T-SNE Plot



Experiment 2

Model2 was conducted in experiment 2. Model2 is similar to Model1 except for that it uses 3 layers instead of 2. All other parameters of the models are identical. The extra third layer uses 512 units or nodes. Model2 went 14 iterations before stopping. Accuracy and loss of each iteration is shown below in figure 5. The vladiation accuracy at first performed as high as the training accuracy and then quickly dropped off. The shape of the accuracy and loss is similar to that of model1.

Figure 5
Model2 Accuracy and Loss Graphs

Model Name	Accuracy	Loss
Model2		

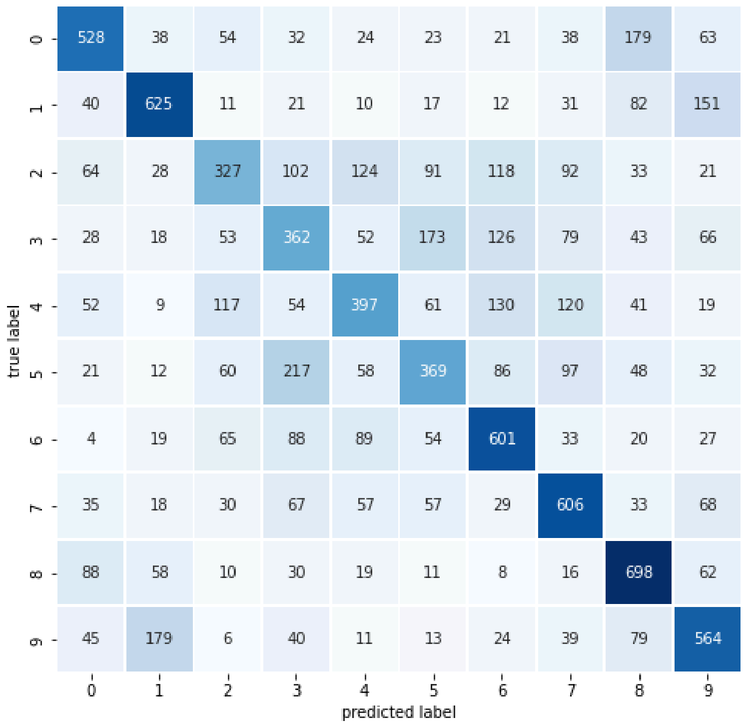
Furthermore, the accuracy of and root mean square of model2 was generated. The accuracy of model2 was slightly higher than models1'. Due to the randomness of what data will be included in the training and test testing set, this small improvement is not an indication that model2 did better than model1. The root mean square error of model 2 is slightly worse than model1. With model2 having a slightly greater accuracy and a slightly lower root mean squared error the models are nearly equal to each other.

Figure 6
Model2s' Test Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Regularization	L2	Process Time
Model2	0.5077	3.096	DNN	3	No	-	11.14

A confusion matrix of Model2 predictions versus observed values is shown in figure 7. Here the missfications are common as well. Model2 did better at classifying airplanes compared to model1. Top 3 misclassified classes include cats for dogs, dogs for cats, and automobiles for ships. The overall most misclassified class was cats. Other successful classifications were for automobiles and ships.

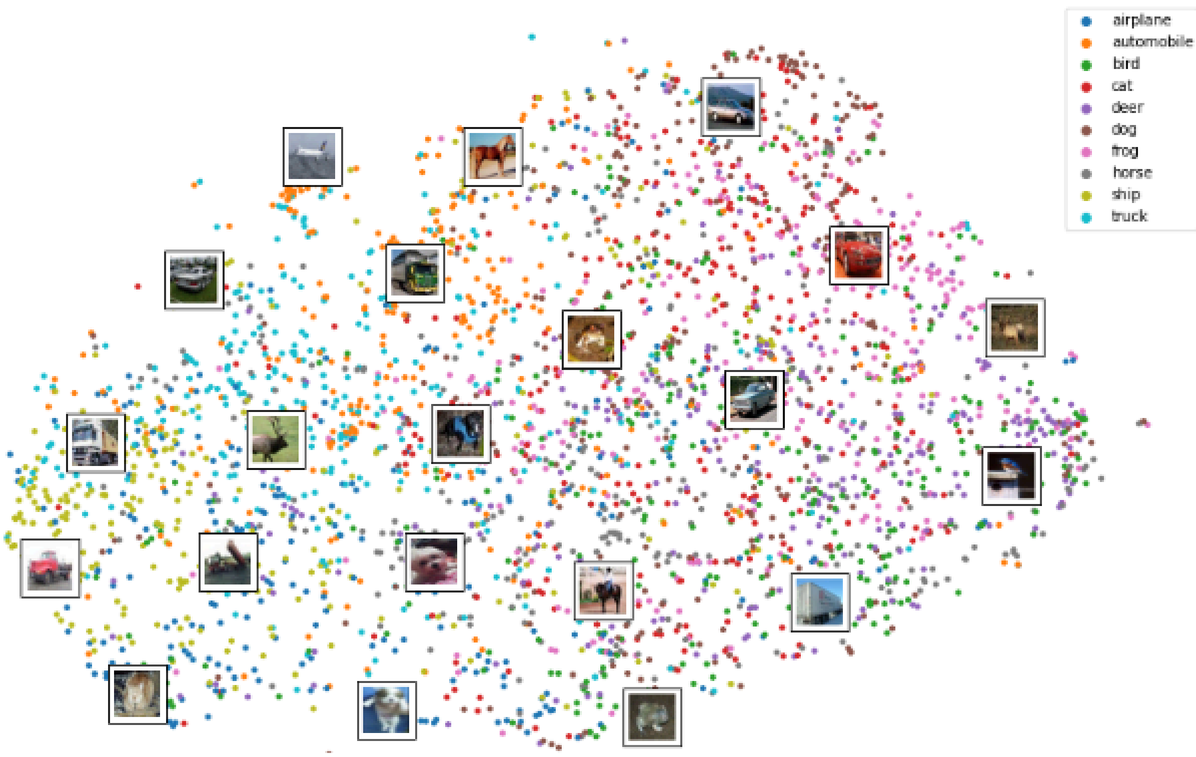
Figure 7
Model2s' Confusion Matrix



Label	Class_
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

To close out model2's evaluation a T-SNE was produced. The shape of this T-SNE is much different than model1 even though they performed similarly. Here are new images that did not appear in model1s' T-SNE such as the red car and red truck. There are several images of automobiles located through the plot. Ideally the images of automobiles would all be clumped together. Also of note, is that all classes appear at least once. The dots of trucks and ships alike to the confusion matrix show that they predicted well.

Figure 8
Model2s' T-SNE Plot

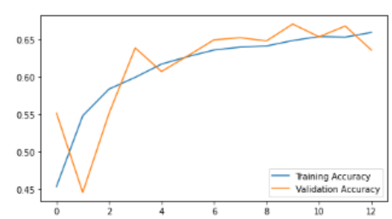
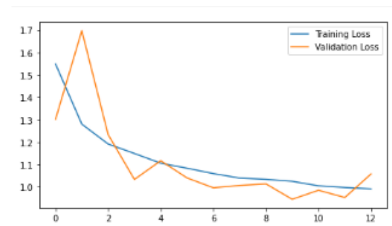


Experiment 3

In Experiment 3, a CNN model was used rather than a DNN. Model3 is a CNN with 2 convolution and max pooling layers. Although similar in design to model1 with 2 layers the CNN aspect of the model dramatically changed the models performance. Figure 9, shows a higher amount of accuracy than previous models at only its second iteration. Model3 went for 13 iterations. Interestingly the valdication outperformed the training at certain iterations for both accuracy and loss.

Figure 9

Model3s' Accuracy and Loss Graphs

Model Name	Accuracy	Loss
Model3		

To see if model3 did outperform the previous models their scores were collected in figure 10. Here the table contains more fields such as strides, dropout rate, and regulationization. The last two fields of regularization and L2 will not apply until experiment 5. The accuracy for model 3 is moderate as is higher than model1 and model2s'. The Root mean squared error is of the same vain.

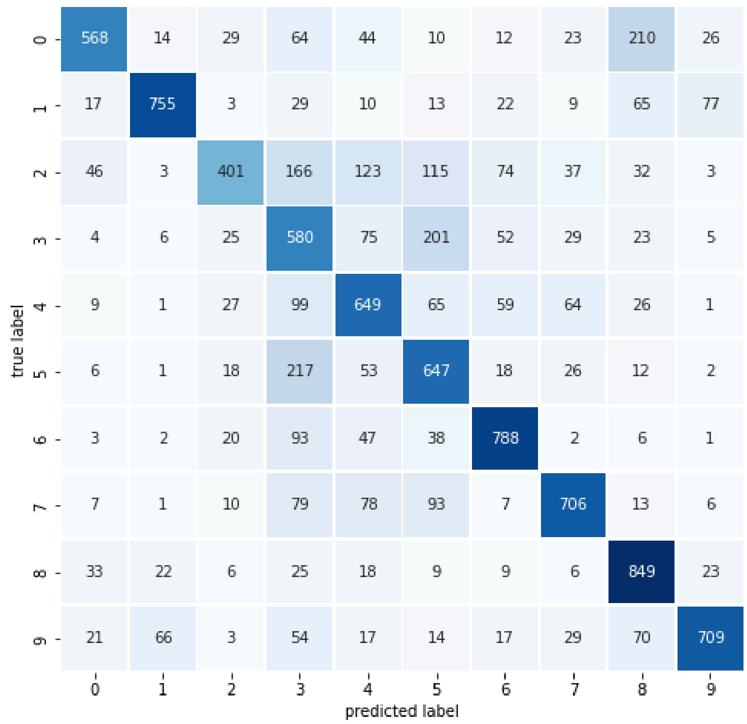
Figure 10

Model3s' Test Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Strides	Dropout	Regularization	L2	Process Time
Model3	0.6652	2.3952	CNN	2	2	0.3	No	-	97

From the start, Model3s' confusion matrix has fewer squares of a blue color. The first 5 class predictions were more accurate than models 1 and 2. Once again ships and automobiles classify well. The weakest classification is for birds. Common misclassifications in model3 were for dogs as cats, airplanes as ships, and cats as dogs. These were also common errors in the previous models. Model3 did a significantly better job at classifying automobiles, cats, ships.

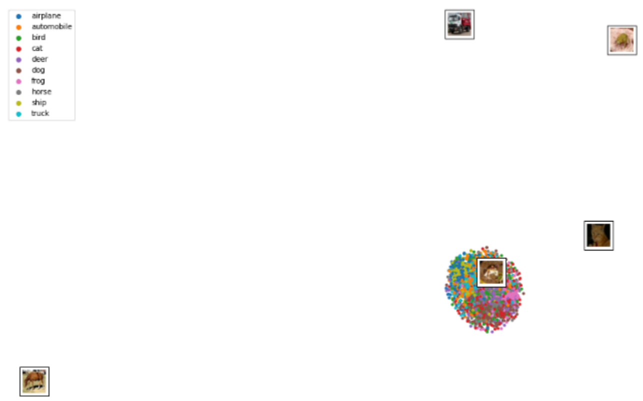
Figure 11
Model3s' Confusion Matrix



Label	Class_
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

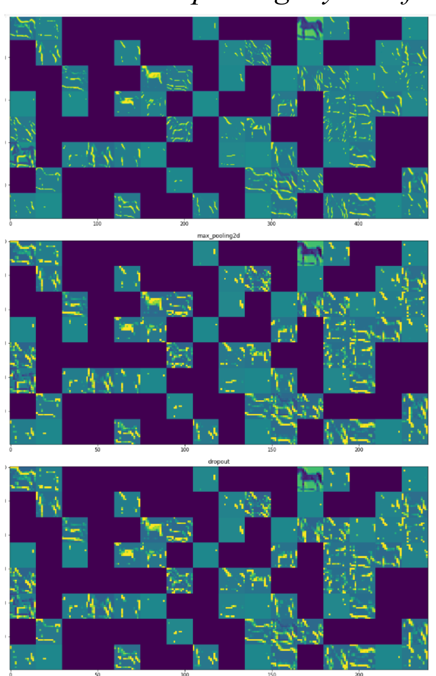
The T-SNE plot for model3 resulted in a bizarre way where one image appears over the majority of the data points. Then a few other images are displayed. It is believed what is happening is that there are outlier data points outside the main cluster that result in images produced and a zoomed out T-SNE plot. It is difficult to analyze this plot from this far away. The confusion matrix and scores will have to suffice for model3.

Figure 12
Model3s' T-SNE Plot



There is one additional plot for Model3. Here we take the filters from the maxpooling layers and show their outputs on a grid. The lightened images should show some of the original images of the dataset. All that can be seen is a horse image, which is only truly clear in one of the squares.

Figure 13
Model3s' Max pooling layers s filters output

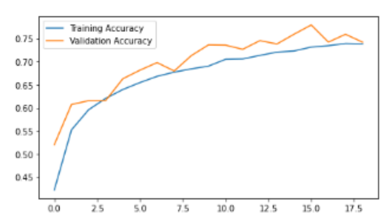
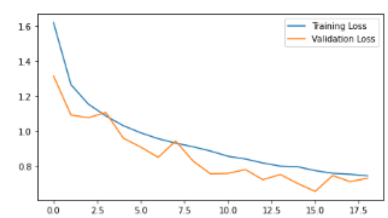


Experiment 4

Akeen to model3, model4 in this experiment will use CNN. This time 3 layers will be used instead of 2. We have already learned that CNN models without regularization outperform DNN models without regularization. Now we will find out if adding another layer will improve the predictions. In DNN the added layer added no value, CNN may perform better.

Model4 ran for 19 iterations, the largest so far. Figure 13 details how the accuracy and loss are the best we have seen out of all the experiments. The validation consistently performs better than the training. Based on these charts, this proves to be the best model yet made in this assignment.

Figure 14
Model4s' Accuracy and Loss Graphs

Model Name	Accuracy	Loss
Model3		

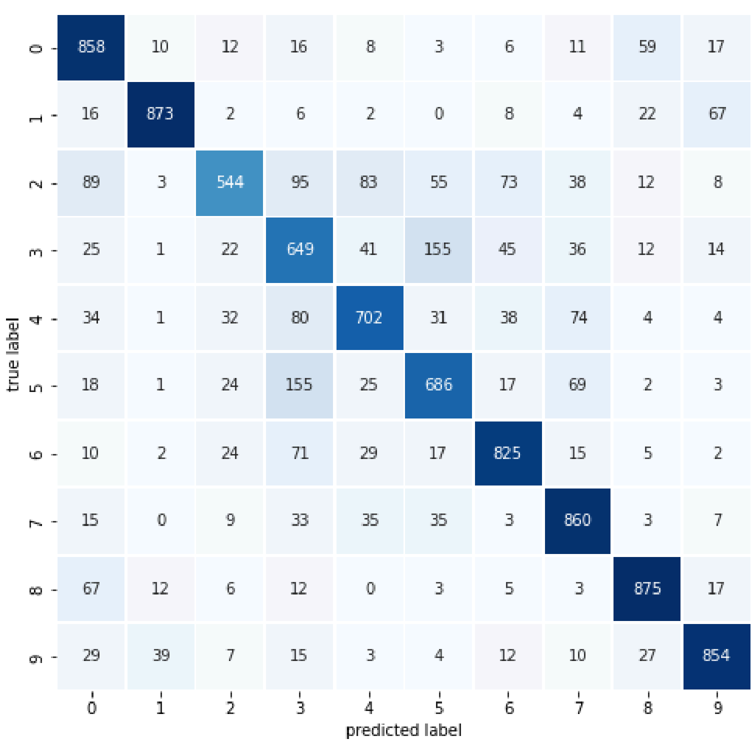
Further proof of model4s' greatness is shown in figure 14. The accuracy is still in the moderate range and has yet to enter into the strong threshold. Loss was less than 2. The extra layer gave model 4 11% more accuracy than model3. With model3 already being 16% ahead of the DNN Models 1 and 2.

Figure 14
Model4s' Test Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Strides	Dropout	Regularization	L2	Process Time
Model4	0.7726	1.938	CNN	3	2	0.3	No	-	168

Majority of the squares in figure 15 possess a dark blue hue representing that the images had many correct predictions. Automobiles and ships had the largest amount of correct predictions. Birds had the lowest amount of correct predictions but still performed well. THe largest misclassification is a tie between dogs as cats and cats as dogs. These animals must have too similar sizes and features to differentiate

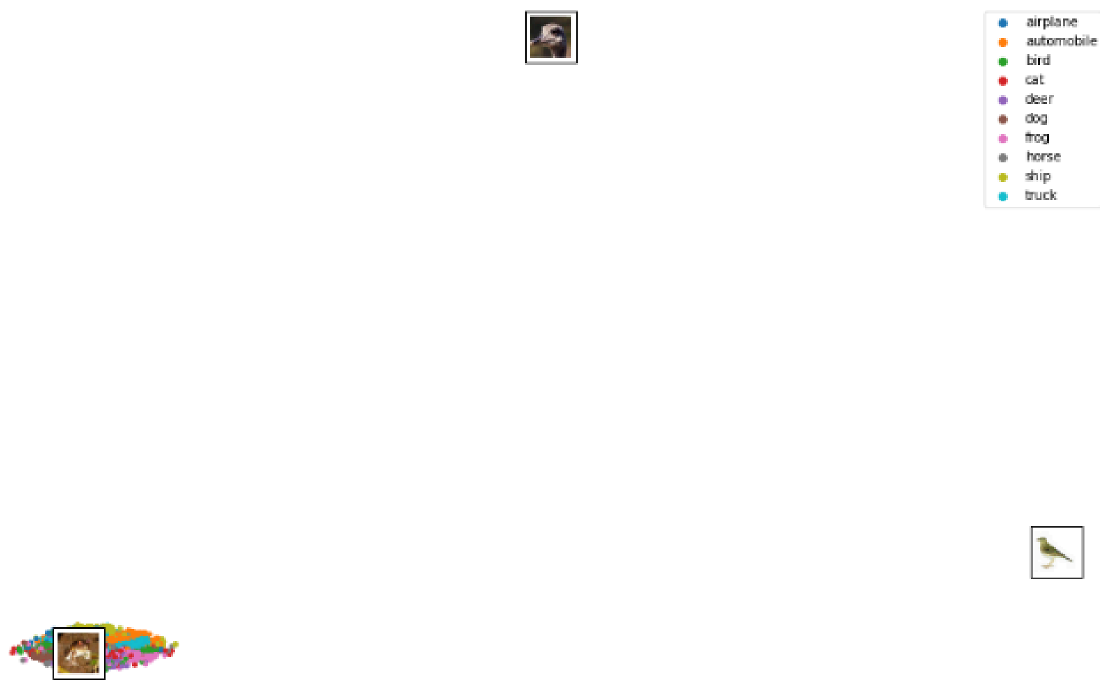
Figure 15
Model4s' Confusion Matrix



Label	Class_
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

Unfortunately the same problem that models3' T-SNE faced occurred for model4s'. This time only two images were left outside of the main cluster. Those two images of both birds. The image on the main cluster is of a frog. Evaluation of model4 must rely on the confusion matrix and scores.

Figure 16
Model4s' T-SNE Plot



Experiment 5

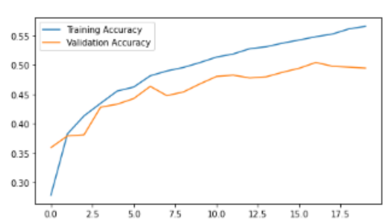
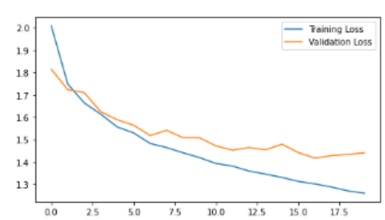
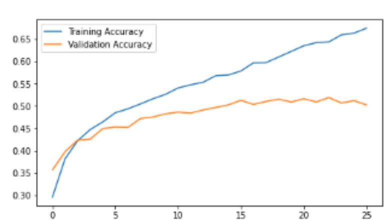
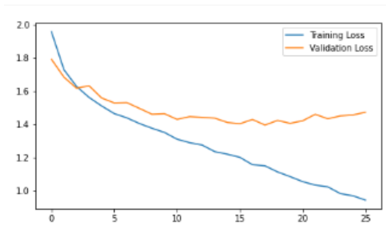
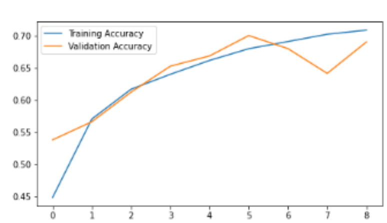
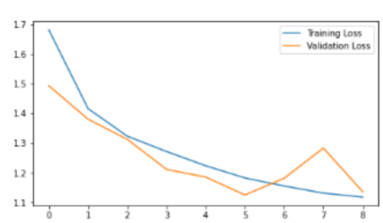
6 Models were made in experiment 5. Models 5-8 are replicas of models 1-4 but use regularization. Models 9 and 10 are CNN with alternative parameters. Model 9 has 2 convolutional layers with an increased number of units or nodes in the dense layer while also

having an increased strides size of 3. The last model, model10, has 3 convolutional layers, an increased number of units or nodes in the dense layer, and a smaller dropout rate of 0.2. To see a visualization of all the differences between each model please refer to figure 19.

Accuracy and Loss at each iteration were recorded for all models in experiment 5. Model6 had the most iterations ran out of any of the 10 models created with 26. For training accuracy, model10 reached the highest. While validation accuracy was the highest for model8. Of models 5-10, the one with the lowest training loss was a close tie between model9 and model10. With the model8 having the lowest validation loss. From these charts, we can infer that models 8, 9, and 10 are the best-performing models in experiment 5.

Figure 18

Model 5-10s' Accuracy and Loss Graphs

Model Name	Accuracy	Loss
Model5		
Model6		
Model7		

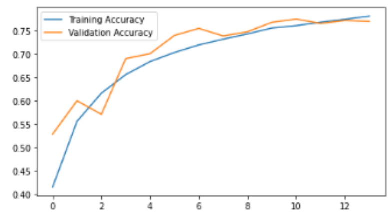
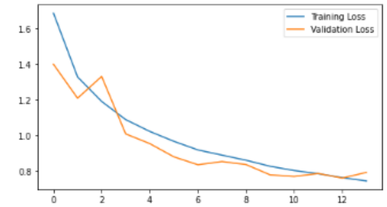
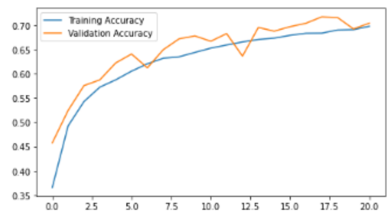
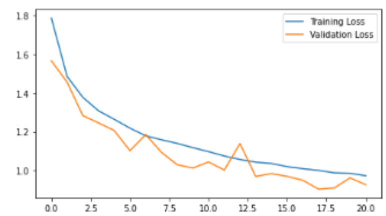
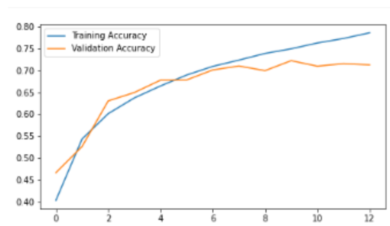
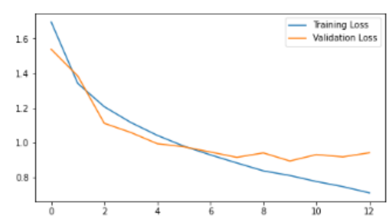
<table>
<tr><td>Model8</td><td></td><td></td></tr>
<tr><td>Model9</td><td></td><td></td></tr>
<tr><td>Model10</td><td></td><td></td></tr>
</table>

Figure 19 confirms that models 8, 9, and 10 did perform well and outperformed the other models. Model9 and model10 performed equally in the sense one model does slightly better on accuracy while the other wins on loss. Overall Model8 did the best. Based on these results, the extra third convolutional layer did improve model scores. Lowering the dropout rate only seemed to hurt the model. Model9 also increased the number of strides and that could of helped improve the scores. All of these models had regularization, models 5, 6, and 7 improved upon their non-regulziation counterparts. While model8 did slightly worse. The best model of all the models is model4. With regularization improving ¾ models but not having the best model, it is difficult to deem it a success. More models will be needed to be created.

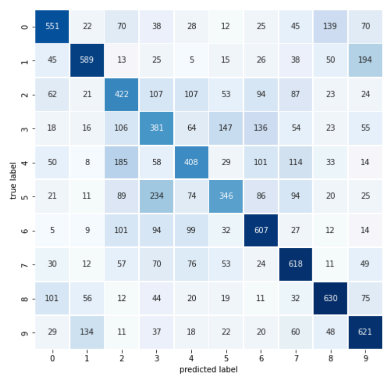
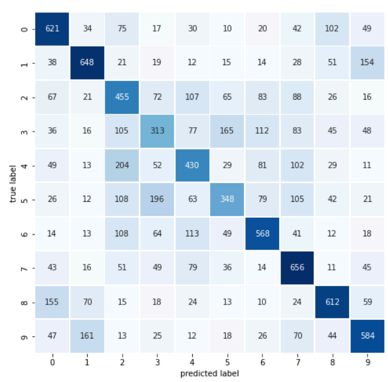
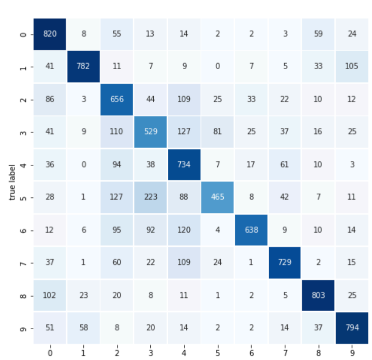
Figure 19
Model 5 - 10s' Test Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Strides	Dropout	Regularization	L2	Process Time
Model5	0.5173	3.0176	DNN	2	-	-	Yes	0.001	17.62
Model6	0.5235	3.0137	DNN	3	-	-	Yes	0.001	19.34
Model7	0.695	2.309	CNN	2	2	0.3	Yes	0.001	81
Model8	0.77	1.9989	CNN	3	2	0.3	Yes	0.001	145
Model9	0.7218	2.219	CNN	2	3	0.3	Yes	0.001	121
Model10	0.7141	2.2142	CNN	3	2	0.2	Yes	0.001	133

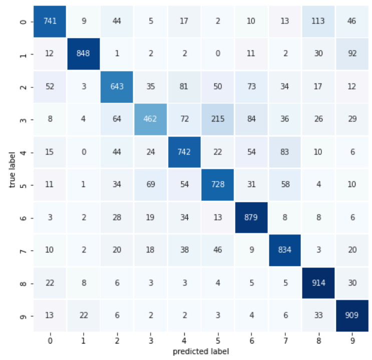
With confusion matrices of models 5-10 we can tell which models made the most accurate predictions and what misclassifications occurred. Model5 had regularization and was better at classifying than model1 without regularization. This is a theme for model6 vs. model2 and model7 vs. model3 as well. Model8 was not as successful as model4 by accuracy score; although it classified 6 of the 10 classes better than model4 did. The 4 classes it did misclassify are a larger discrepancy thus weighting the accuracy down slightly. Even on the best model of experiment 5, model8, did struggle to classify cats. This has been the consistently hardest class to classify amongst classes. Model8 was the best at classifying ships and trucks. More models will need to be made to master identifying cats as cats and not cats as dogs.

Figure 20

Model 5-10s' Confusion Matrices

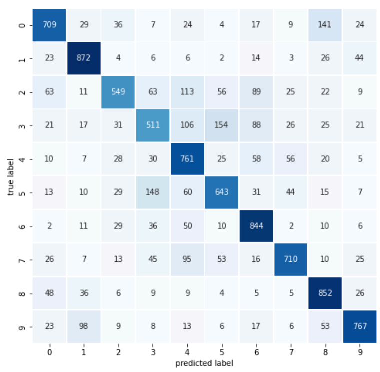
Model Name	Confusion Matrix
Model5	 **Label** **Class_** 0 airplane 1 automobile 2 bird 3 cat 4 deer 5 dog 6 frog 7 horse 8 ship 9 truck
Model6	 **Label** **Class_** 0 airplane 1 automobile 2 bird 3 cat 4 deer 5 dog 6 frog 7 horse 8 ship 9 truck
Model7	 **Label** **Class_** 0 airplane 1 automobile 2 bird 3 cat 4 deer 5 dog 6 frog 7 horse 8 ship 9 truck

Model8



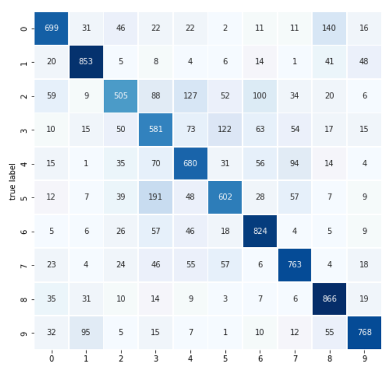
Label	Class_
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

Model9



Label	Class_
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

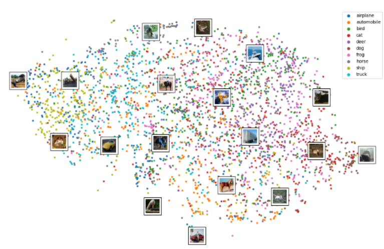
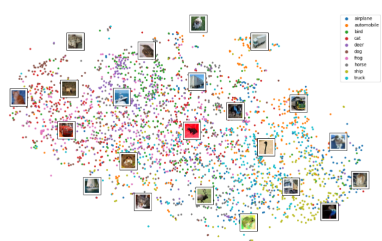
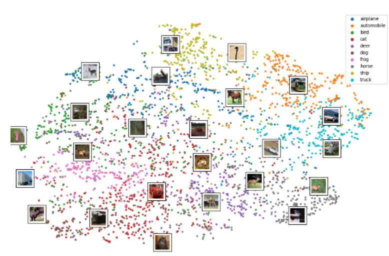
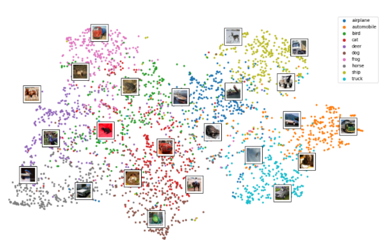
Model10

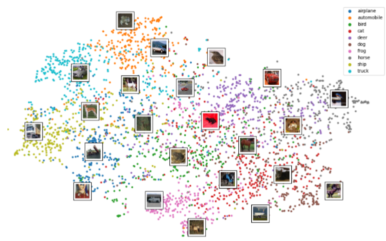
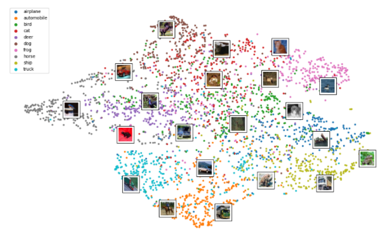


Label	Class_
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

All models in experiment 5 were successful in producing a legible T-SNE plot. This a great improvement of model3 and model4s' plots, It is possible the addition of regularization helped. It is fascinating how the shape of the T-SNE changed upon each model as well as the location of each class's data points. Focusing on the best model of the experiment, modle8 has gaps in its shape. This is indicating a stronger border between the classes, as the ships in the light green color form a peninsula of data points. Other successful clusters in model8 were trucks, automobiles, and frogs. With all the T-SNE plots next to each other, it is easy to distinguish which models clustered better than others.

Figure 21
Model 5-10s' T-SNE Plots

Model Name	T-SNE
Model5	
Model6	
Model7	
Model8	

Model9	
Model10	

Conclusion

In Conclusion, Computer vision through DNN and CNN models were conducted on the CIFAR-10 data set. If a model was proven to be a success than Company X would use the model for facial recognition for users to login to their app. Once facial recognition is implemented a user can have an easy and secure way to login.

Five experiments were held and 10 models were ultimately created. 4 of which were DNN and 6 CNN. The first four models did not have regulzatization and were then remade with regulzattion. Three of the 4 models improved with regulzation, however the model with the highest accuracy was without regulzation. For this, it is difficult to determine if regulzation improved the models. Models generated greater scores with more layers as well as an increase in nodes in the hidden layers. Dropout rate drecase and an increase in stride did not improve the model. Overall Model4 has the greatest accuracy with a score of 0.7726.

All models struggled to classify all 10 of the classes. Specifically Cats were often mistaken for dogs. As well as dogs for cats. The models did become quite successful at indetfying types of transportation such as airplanes, ships, and automobiles. Ultimately the fical recognition that Company X will be used on Humans. SInce the best model only produced moderate accuracy and the models could not distrinigh cats and dogs well, more models will be needed to be made. Incorporating human faces in the dataset would also be great practice for the future models. Through mor iterations of modeling COmpany X will eventually have a facial recognition login on their app.

References

Brownlee, J. (2019, May 12). *How to Develop a CNN From Scratch for CIFAR-10 Photo Classification*. Machine Learning Mastery. https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/

Haque, R. U., Khan, R. H., Shihavuddin, A. S. M., Syeed, M. M. M., & Uddin, M. F. (2022). Lightweight and Parameter-Optimized Real-Time Food Calorie Estimation from Images Using CNN-Based Approach. *Applied Sciences*, *12*(19), 9733. https://doi.org/10.3390/app12199733

Appendix

Figure 1
Model1 Accuracy Graph

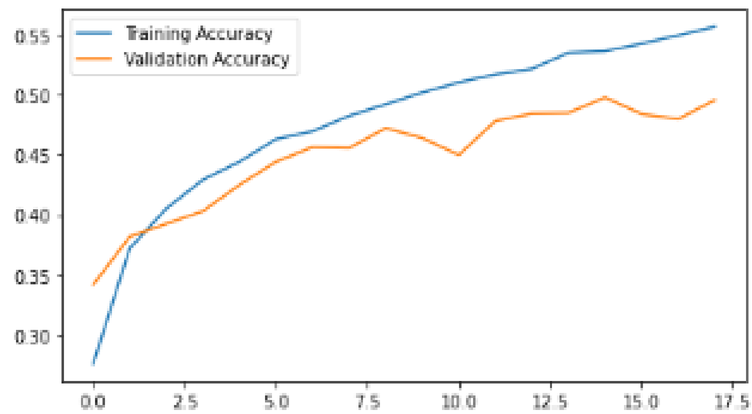


Figure 2
Model1 Loss Graph

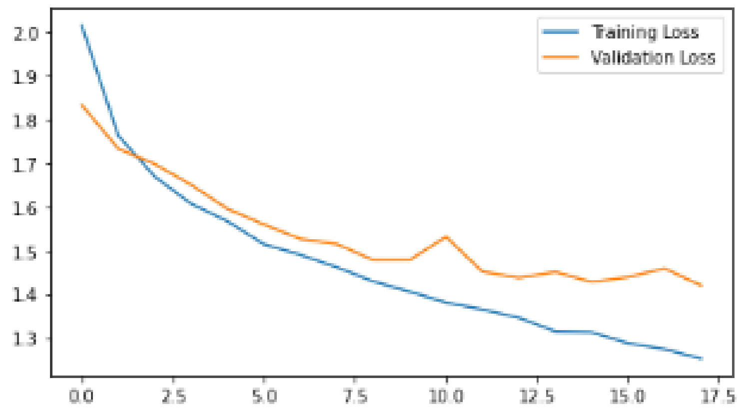


Figure 3
Model1s' Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Regularization	L2	Process Time
Model1	0.5063	3.0801	DNN	2	No	-	14.94

Figure 4

Model1s' Confusion Matrix

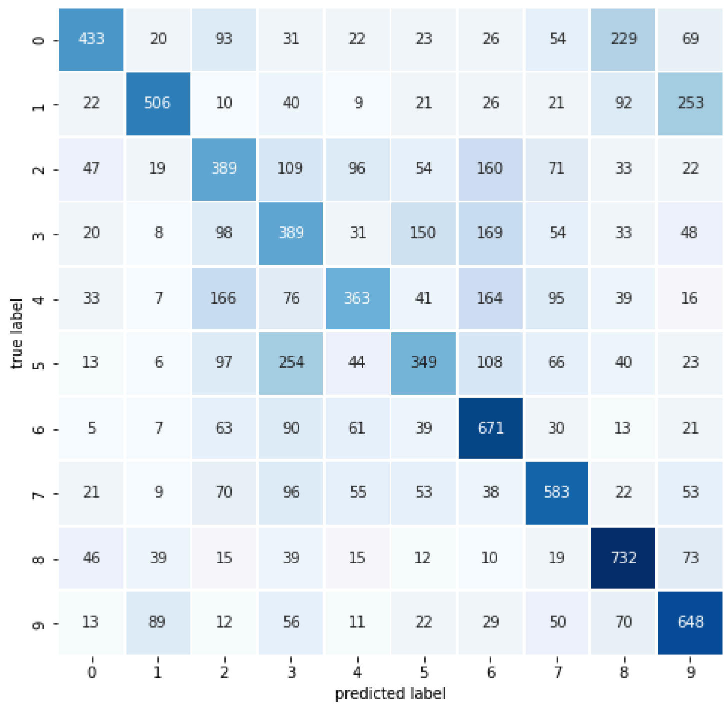


Figure 5
Model1s' T-SNE Plot

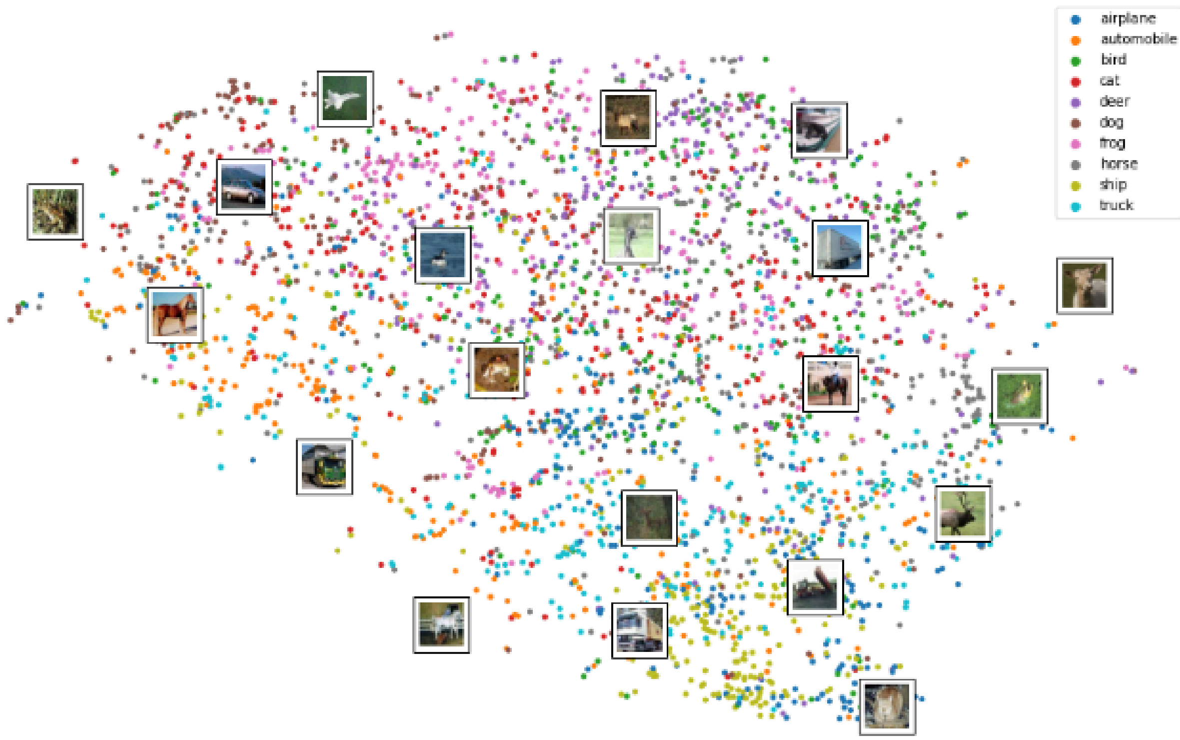


Figure 6
Model2 Accuracy Graph

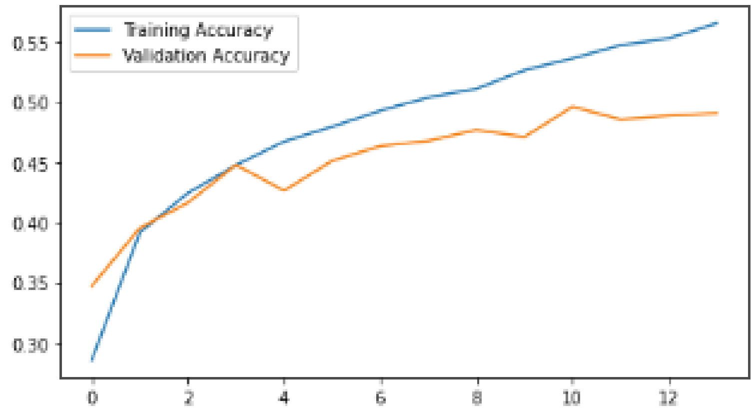


Figure 7
Model2 Loss Graph

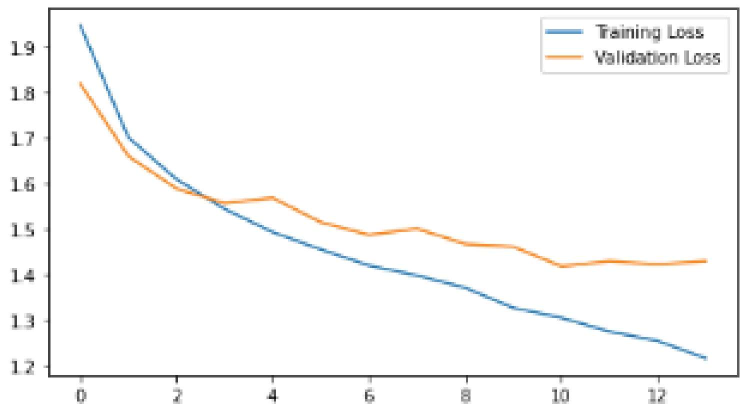


Figure 8
Model2s' Test Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Regularization	L2	Process Time
Model2	0.5077	3.096	DNN	3	No	-	11.14

Figure 9
Model2s' Confusion Matrix

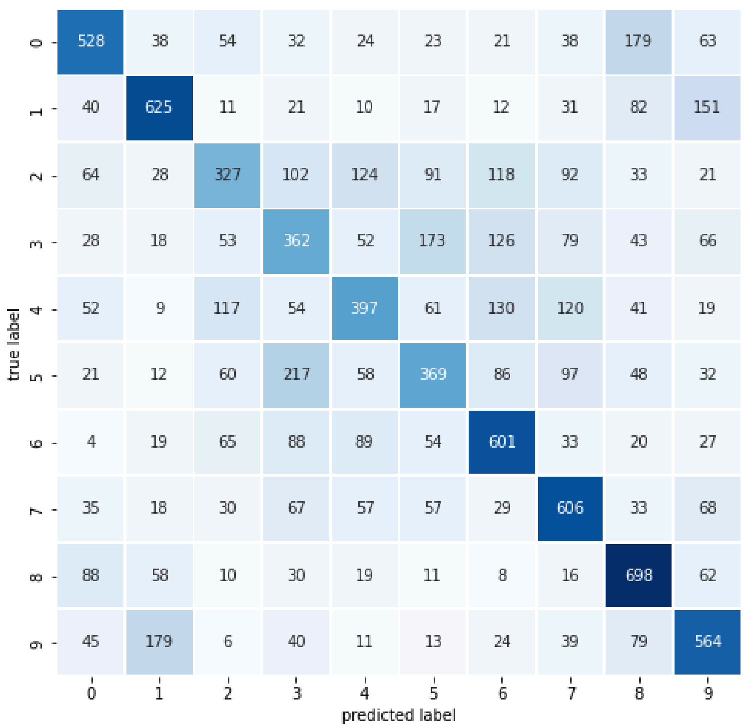


Figure 10
Model2s' T-SNE Plot

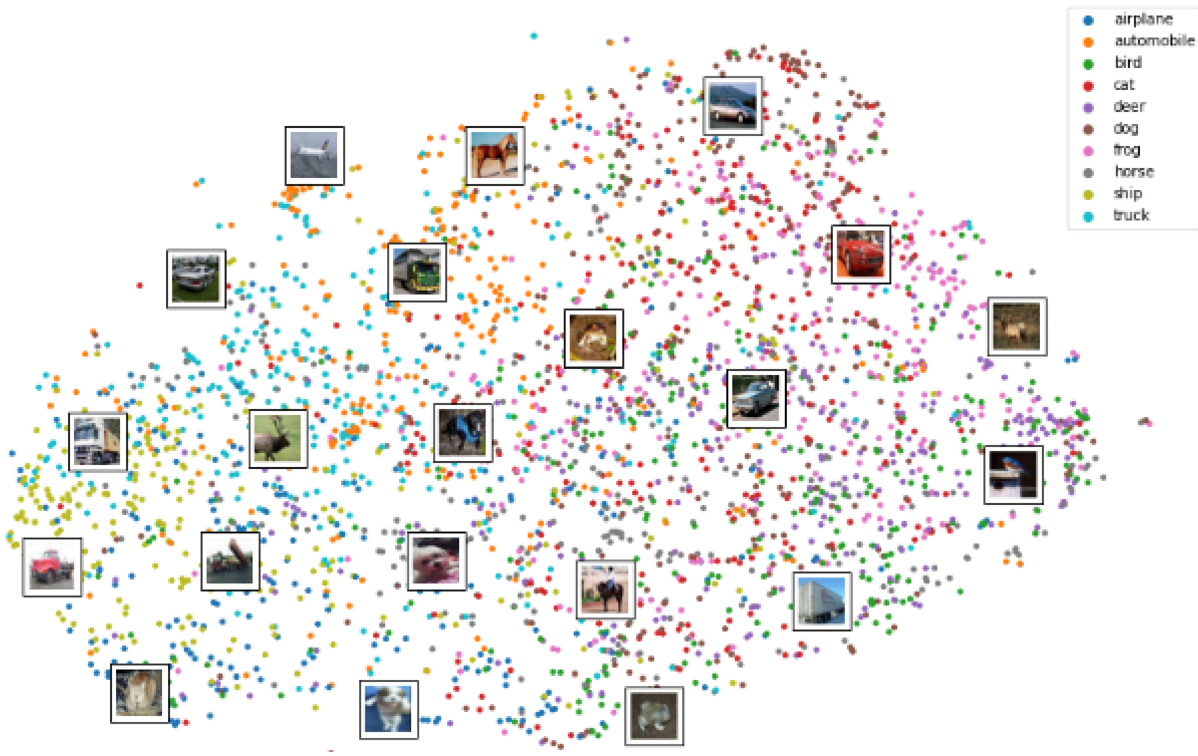


Figure 11
Model3s' Accuracy Graph

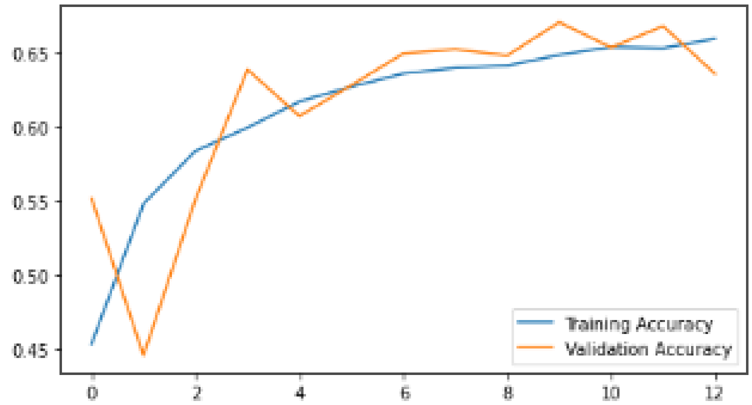


Figure 12
Model3s' Loss Graph

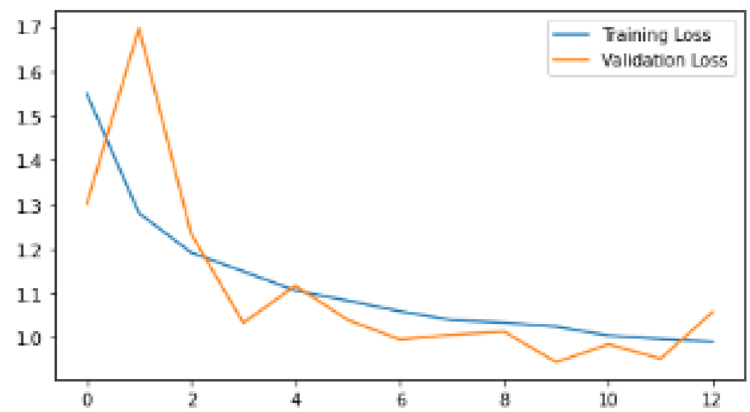


Figure 13
Model3s' Test Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Strides	Dropout	Regularization	L2	Process Time
Model3	0.6652	2.3952	CNN	2	2	0.3	No	-	97

Figure 14

Model3s' Confusion Matrix

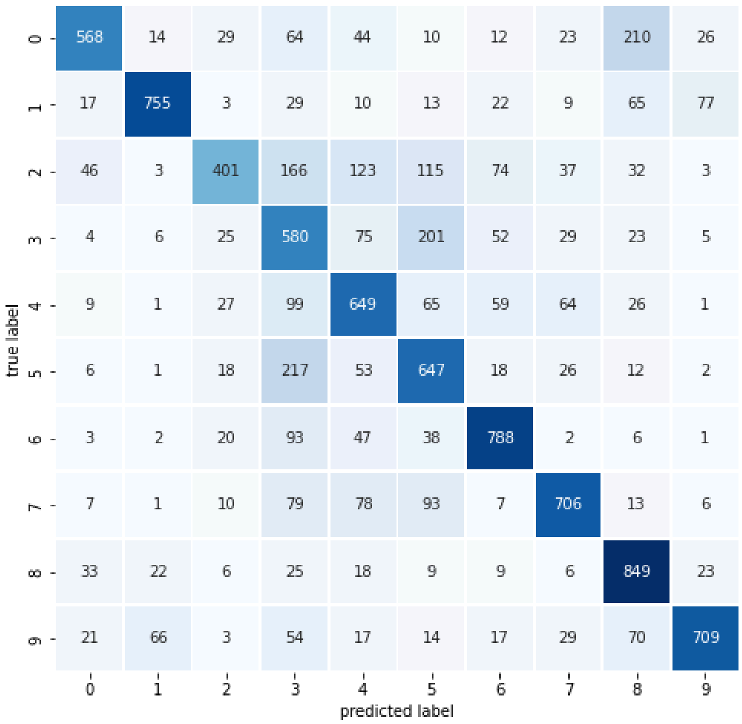


Figure 15
Model3s' T-SNE Plot

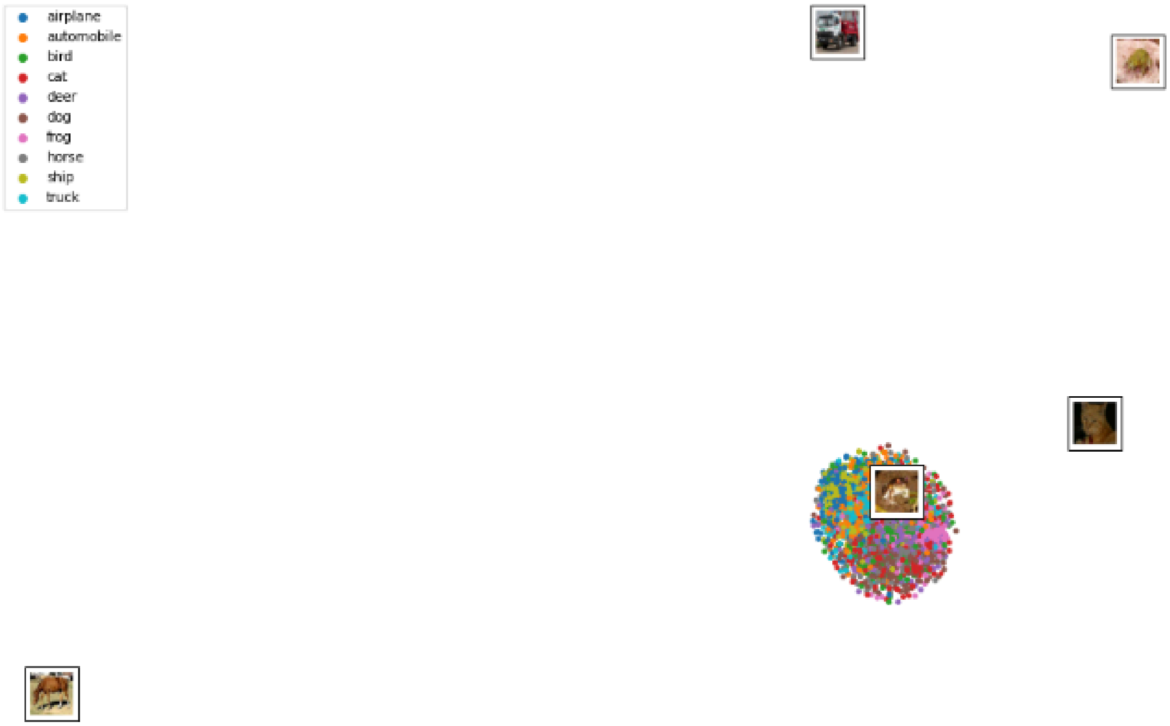


Figure 16
Model3s' Max pooling layers s filters output

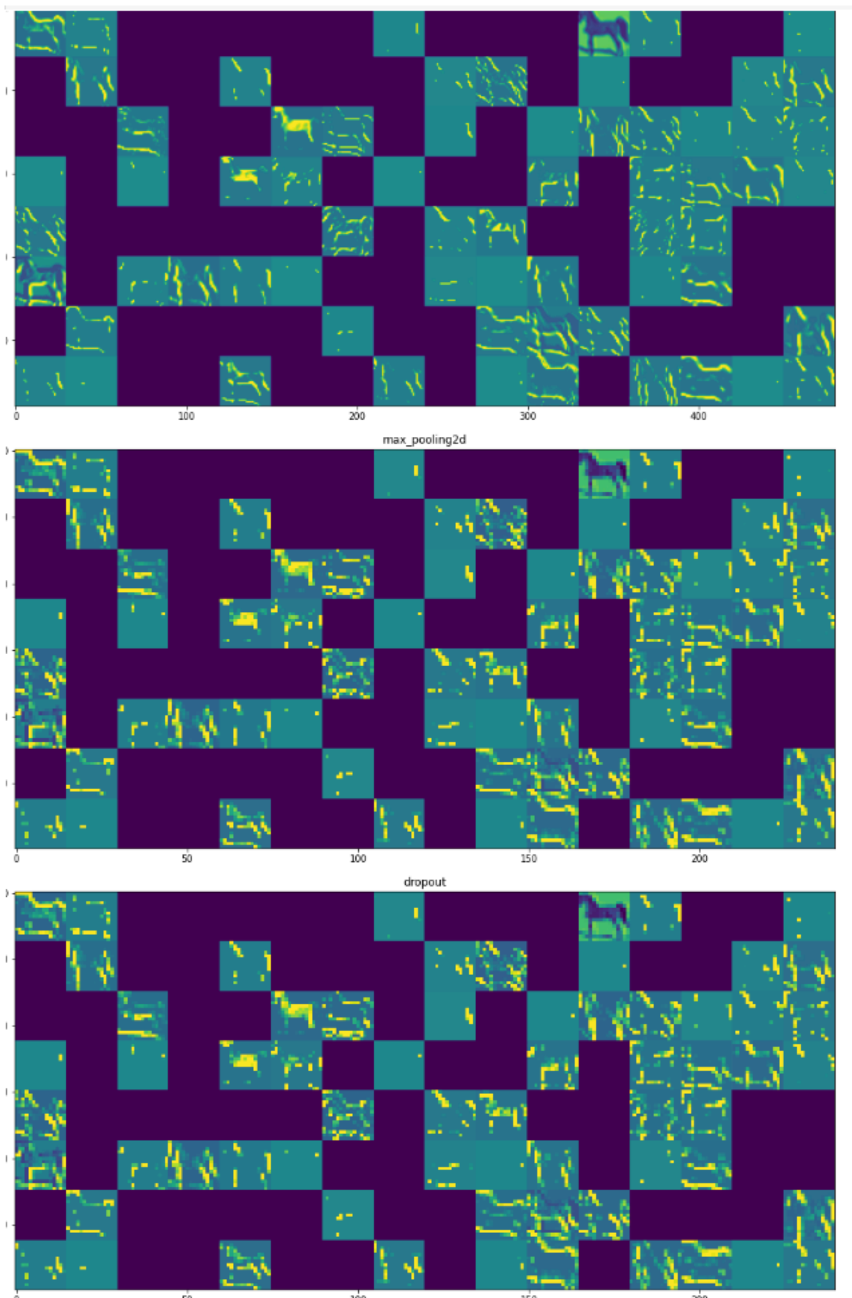


Figure 17

Model4s' Accuracy and Loss Graphs

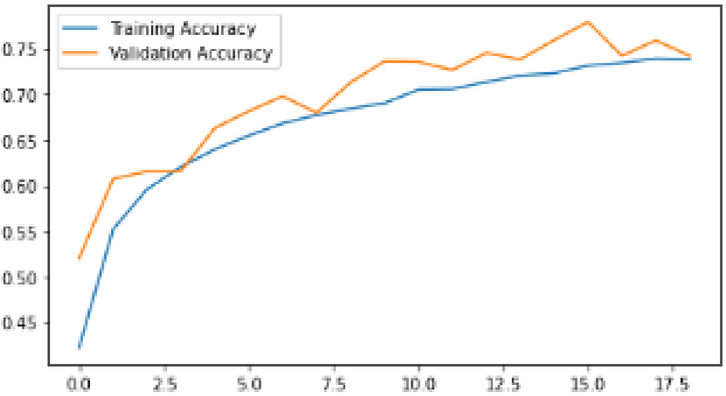


Figure 18
Model4s' Accuracy and Loss Graphs

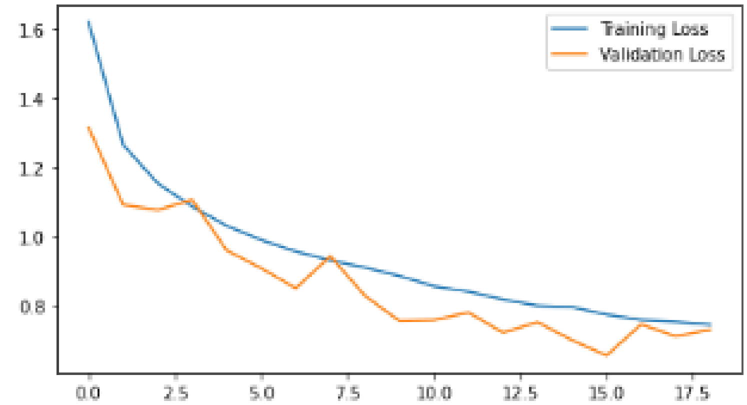


Figure 19
Model4s' Test Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Strides	Dropout	Regularization	L2	Process Time
Model4	0.7726	1.938	CNN	3	2	0.3	No	-	168

Figure 20

Model4s' Confusion Matrix

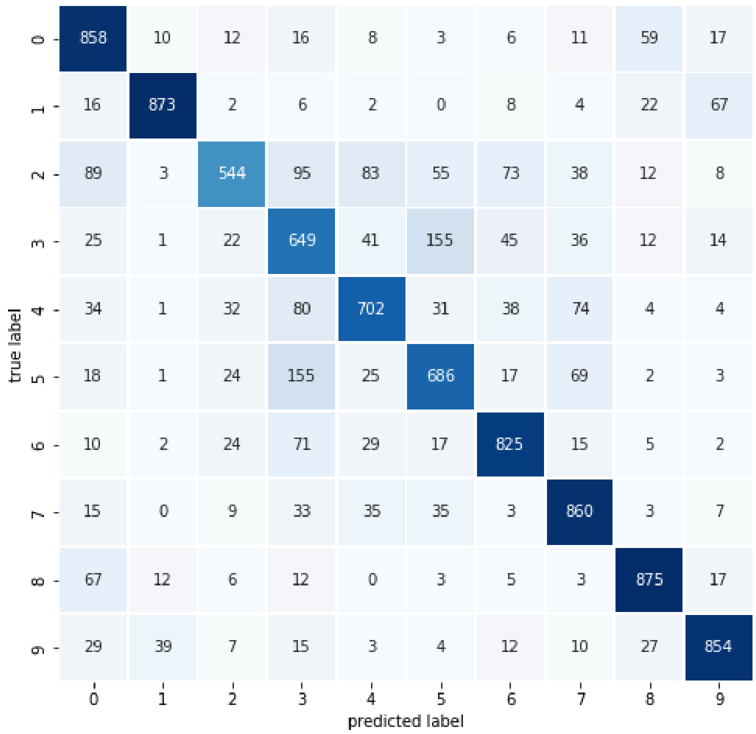


Figure 21
Model4s' T-SNE Plot

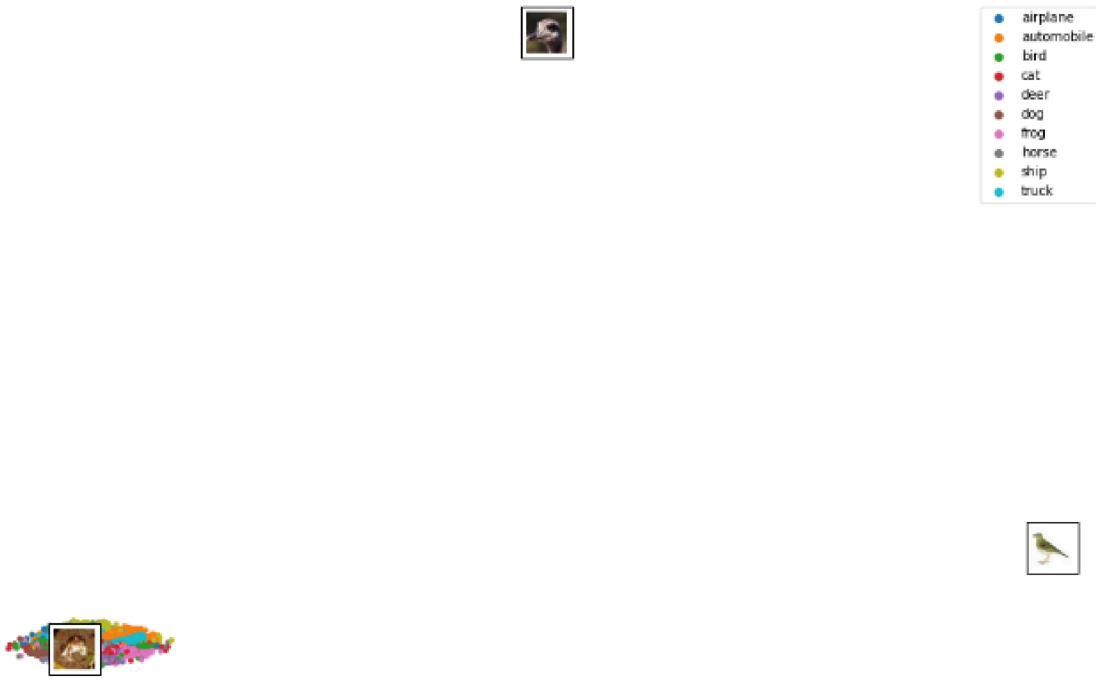


Figure 22
Model 5 Accuracy Graph

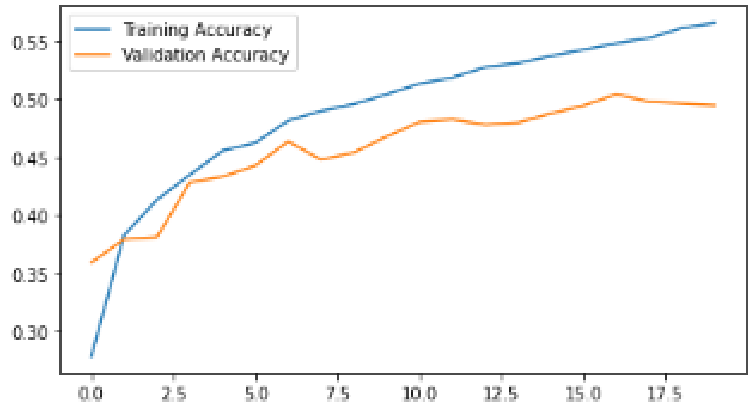


Figure 23
Model 5 Loss Graph

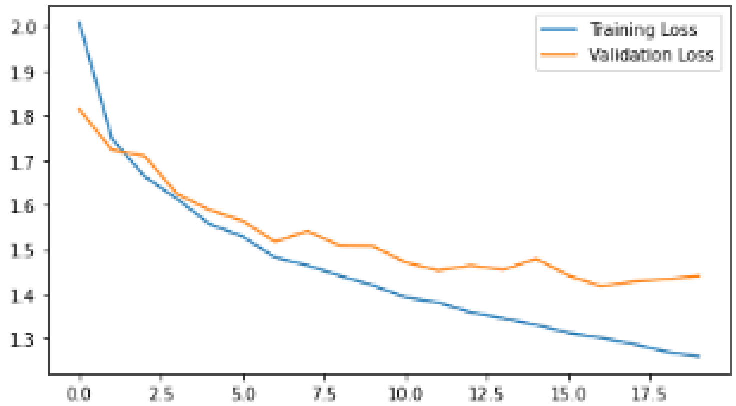


Figure 24
Model 6 Accuracy Graph

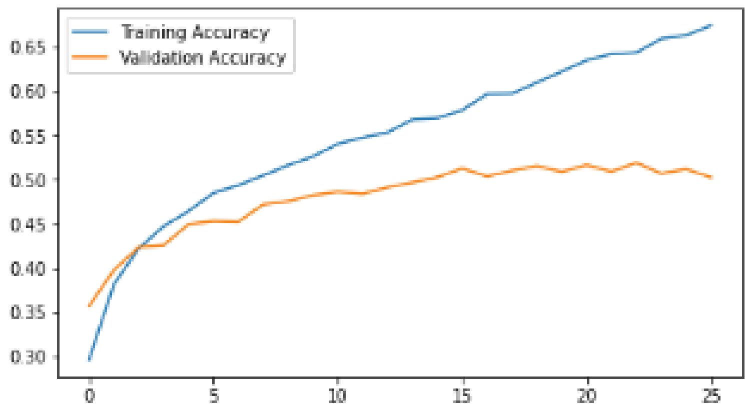


Figure 25
Model 6 Loss Graph

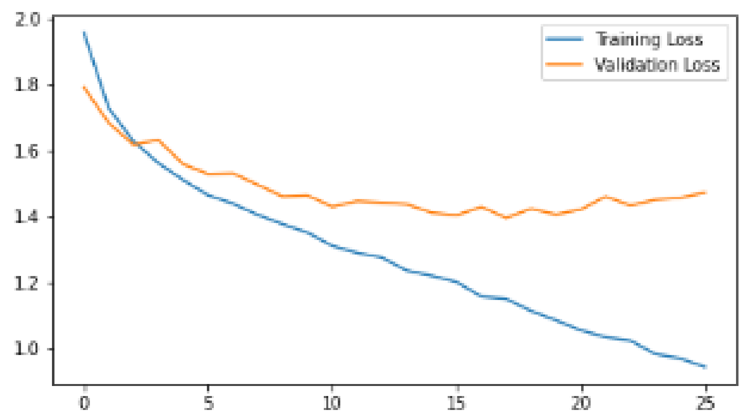


Figure 26
Model 7 Accuracy Graph

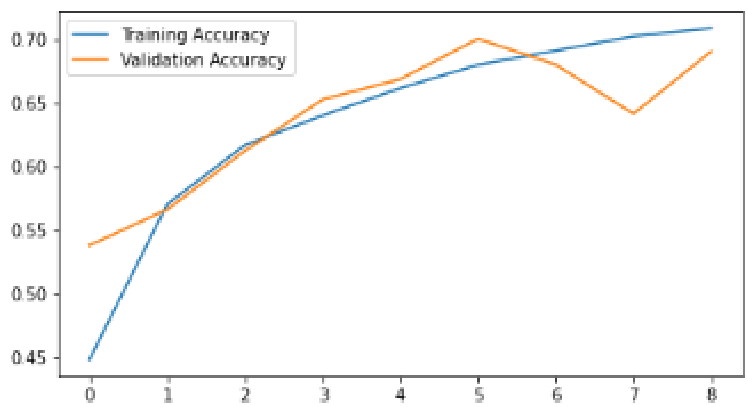


Figure 27
Model 7 Loss Graph

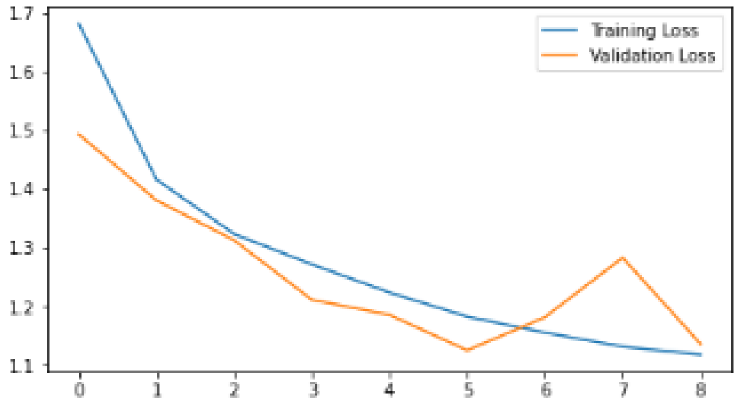


Figure 28
Model 8 Accuracy Graph

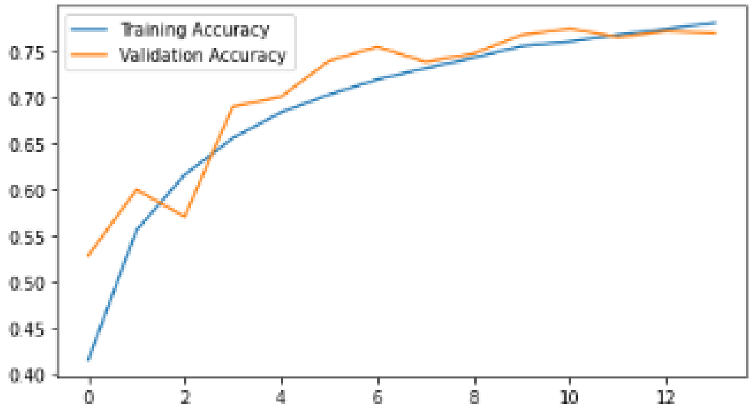


Figure 29

Model 8 Loss Graph

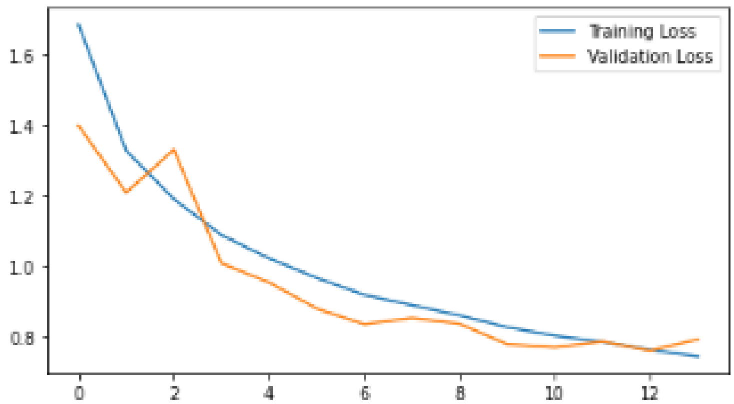


Figure 30

Model 9 Accuracy Graph

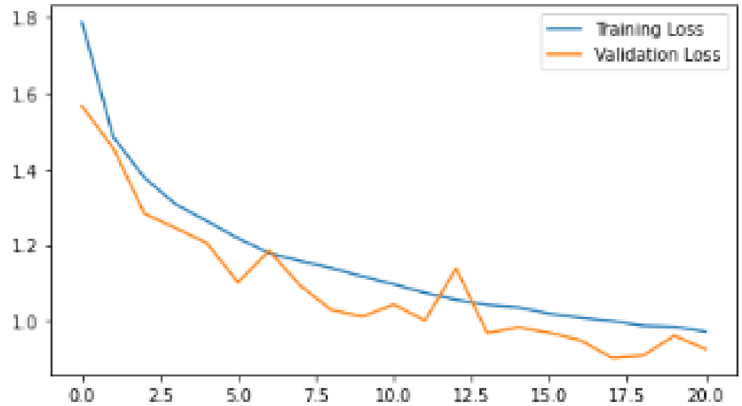


Figure 31
Model 9 Loss Graph

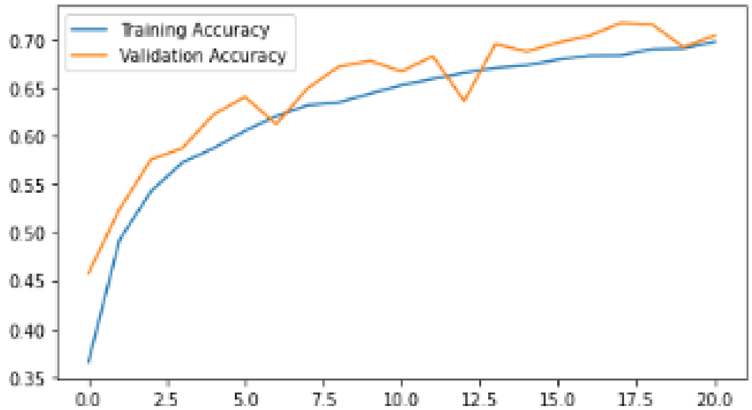


Figure 32
Model 10 Accuracy Graph

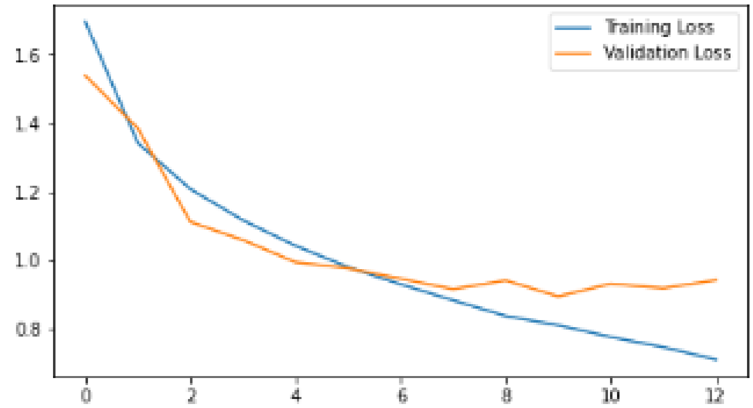


Figure 33
Model 10 Loss Graph

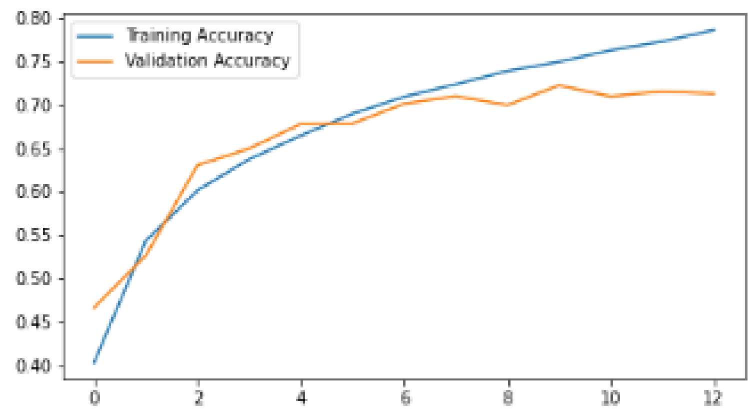


Figure 34
Model 5 - 10s' Test Scores

Model Name	Accuracy Score	Root Mean Square Error	Model Type	Number of Layers	Strides	Dropout	Regularization	L2	Process Time
Model5	0.5173	3.0176	DNN	2	-	-	Yes	0.001	17.62
Model6	0.5235	3.0137	DNN	3	-	-	Yes	0.001	19.34
Model7	0.695	2.309	CNN	2	2	0.3	Yes	0.001	81
Model8	0.77	1.9989	CNN	3	2	0.3	Yes	0.001	145
Model9	0.7218	2.219	CNN	2	3	0.3	Yes	0.001	121
Model10	0.7141	2.2142	CNN	3	2	0.2	Yes	0.001	133

Figure 35

Model 5s' Confusion Matrix

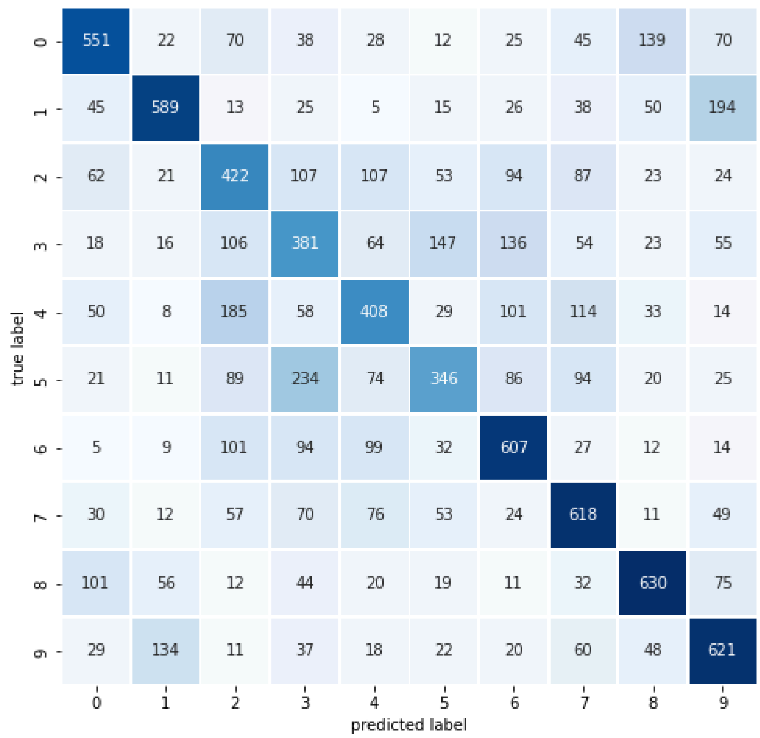


Figure 36
Model 6s' Confusion Matrix

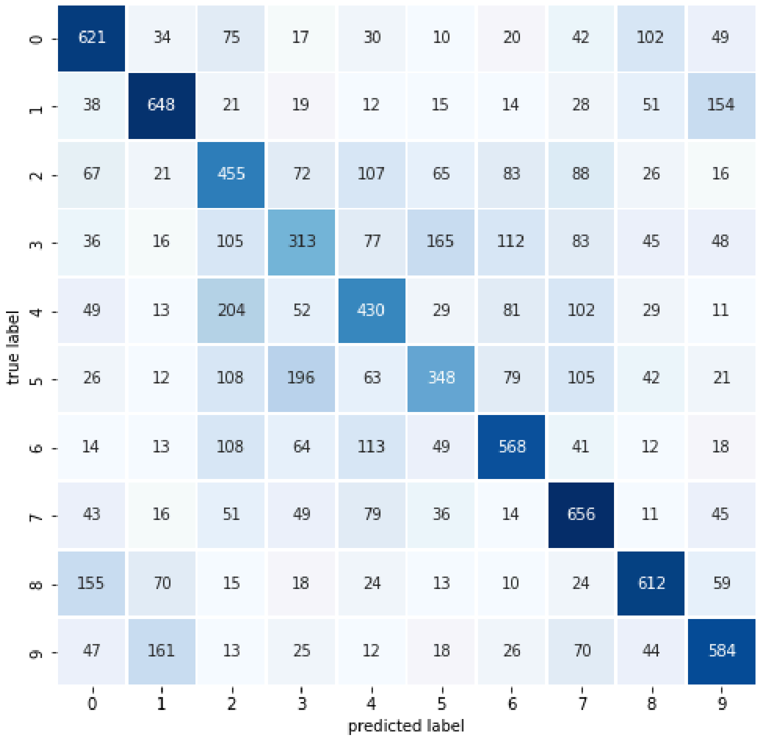


Figure 37

Model 7s' Confusion Matrix

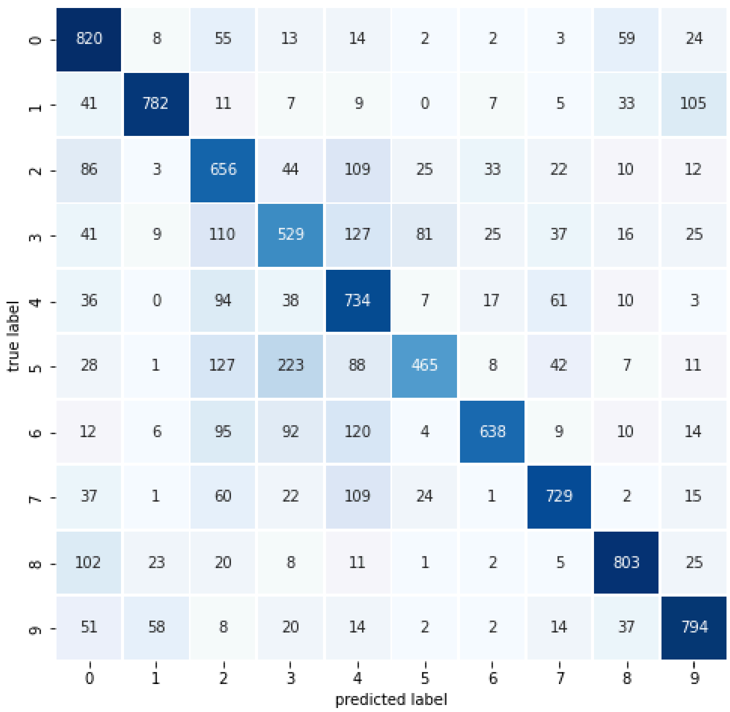


Figure 38
Model 8s' Confusion Matrix

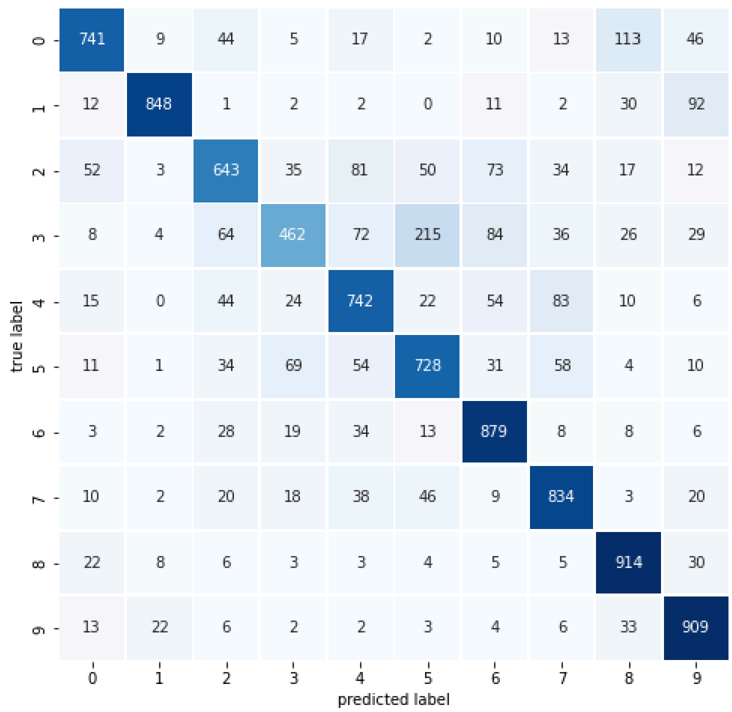


Figure 39

Model 9s' Confusion Matrix

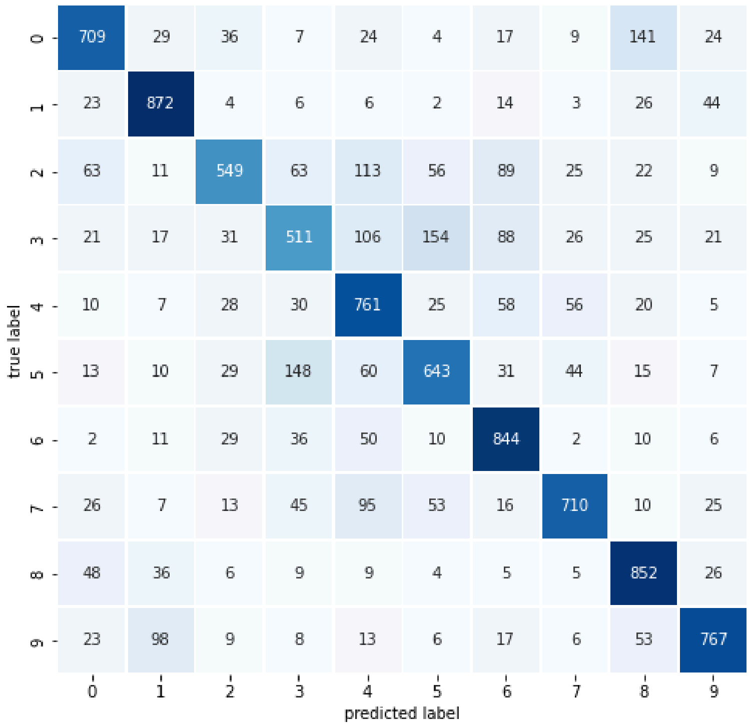


Figure 40

Model 10s' Confusion Matrix

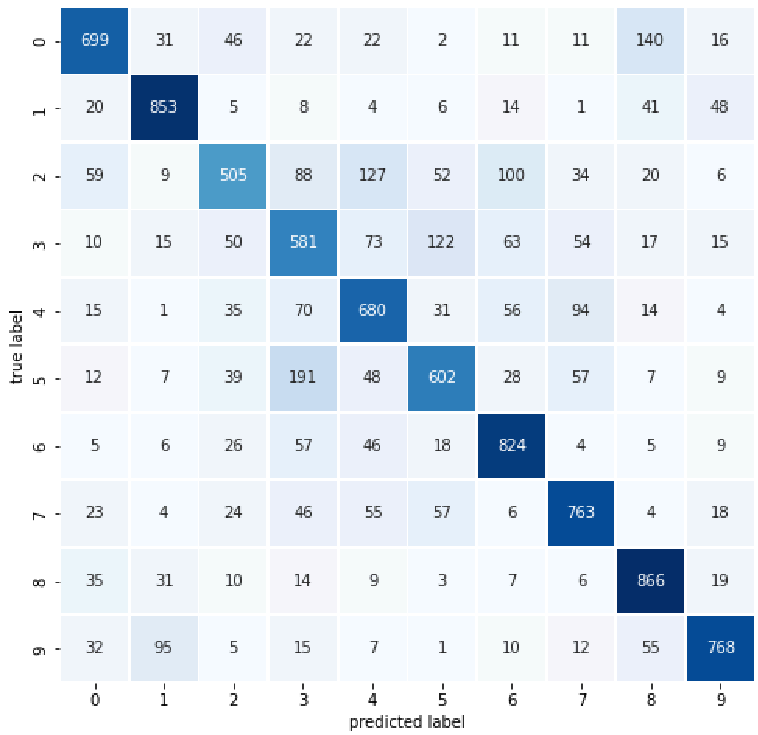


Figure 41

Model 5s' T-SNE Plots

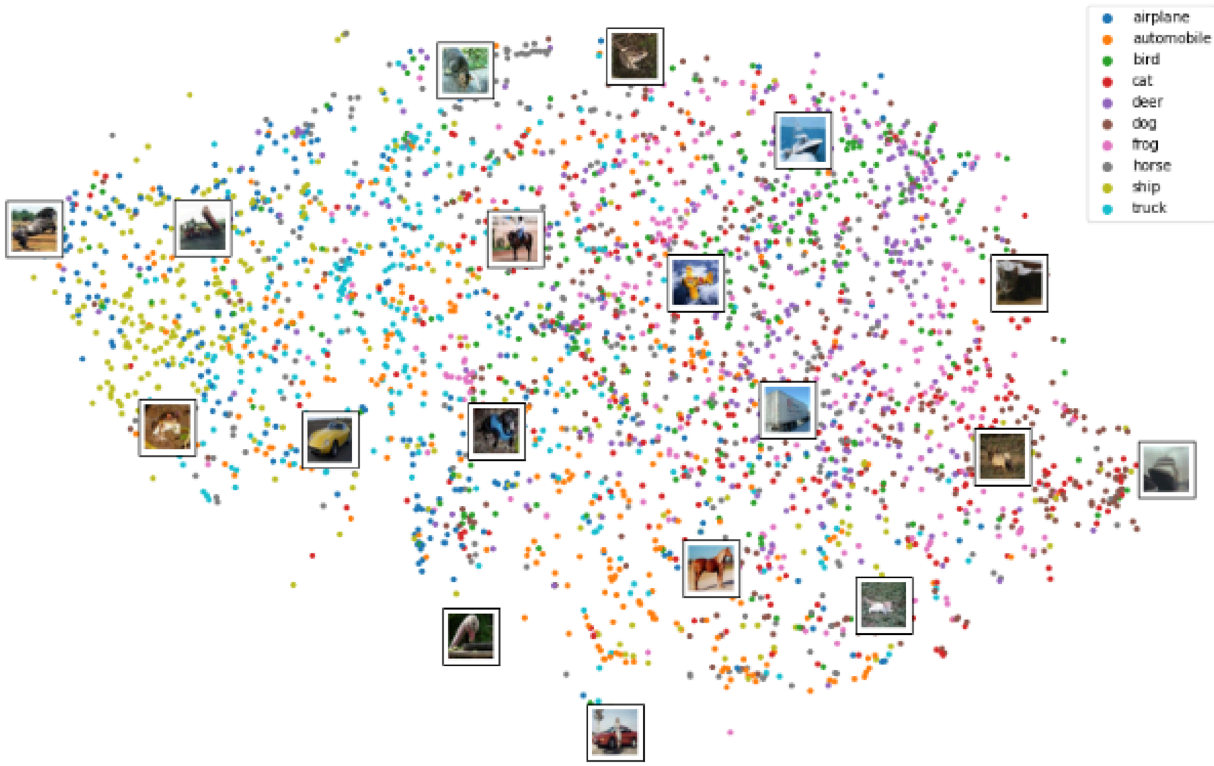


Figure 42
Model 6s' T-SNE Plots

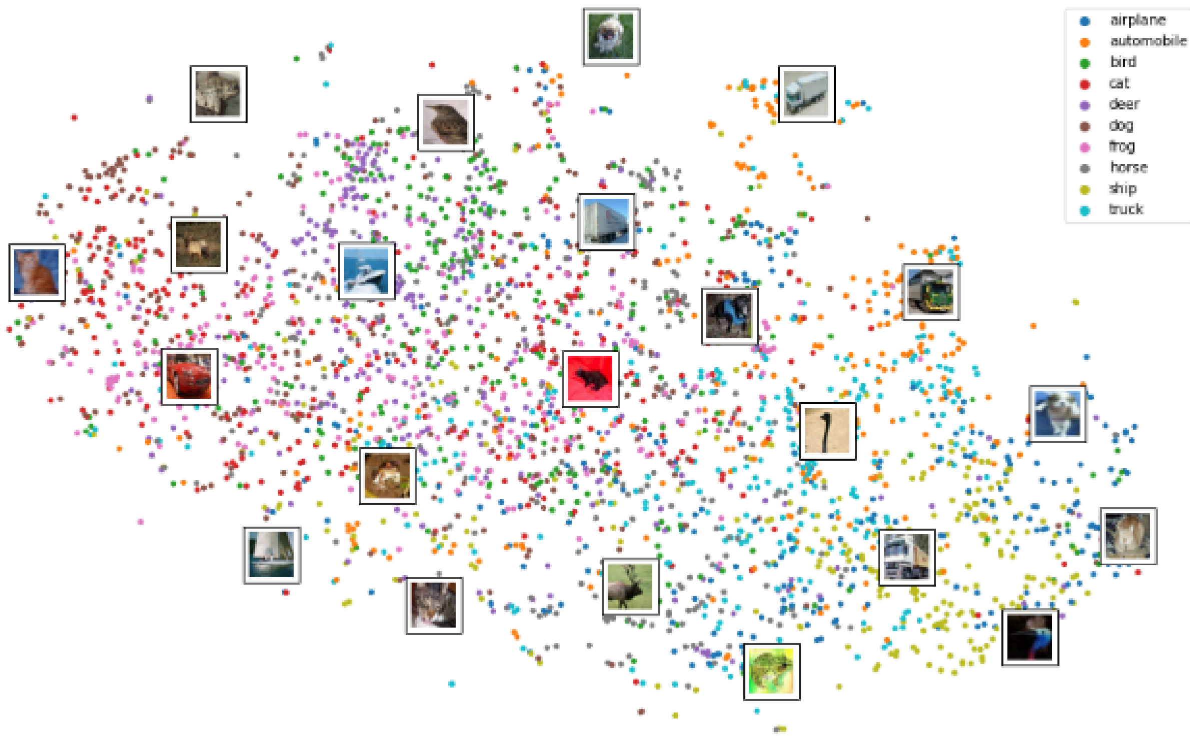


Figure 43
Model 7s' T-SNE Plots

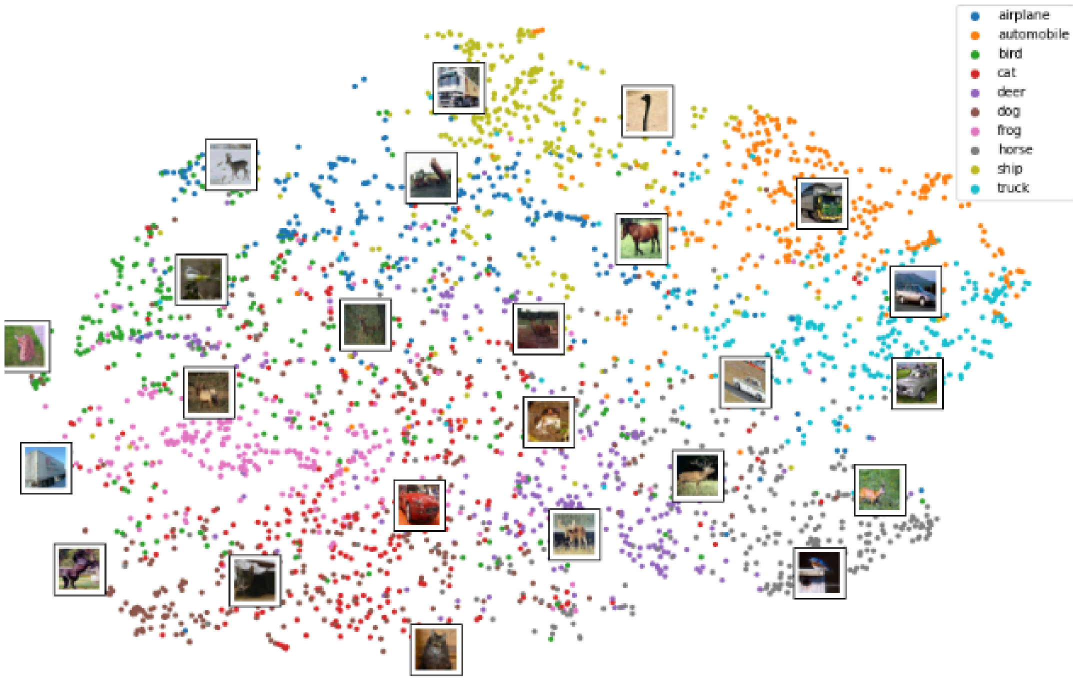


Figure 44
Model 8s' T-SNE Plots

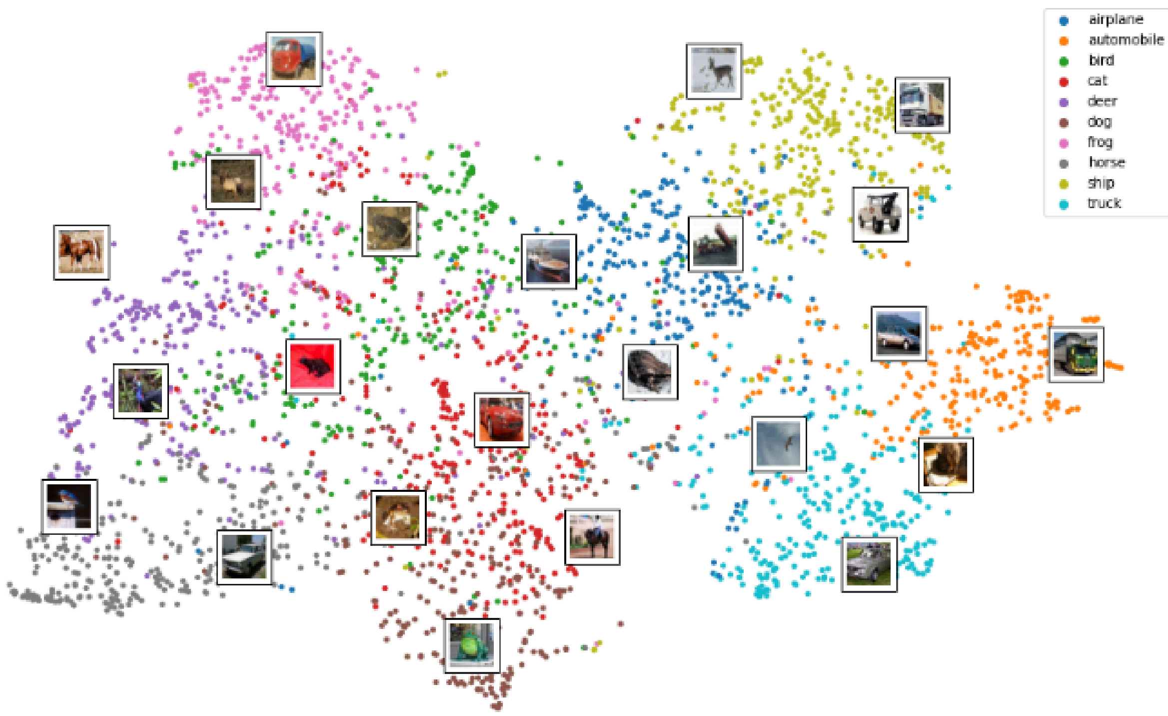


Figure 45
Model 9s' T-SNE Plots

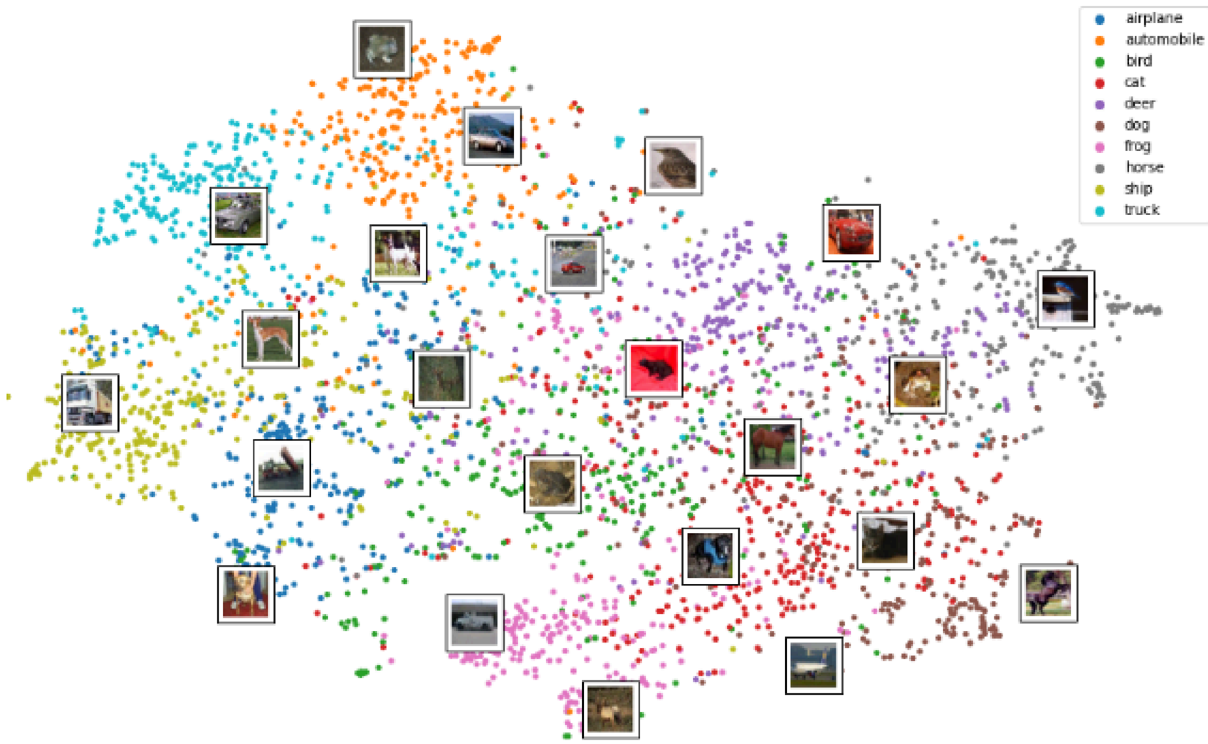


Figure 46
Model 10s' T-SNE Plots