NORTHWESTERN UNIVERSITY | SCHOOL OF PROFESSIONAL STUDIES

A.2 Second Research/Programming Assignment: Deep Neural Networks & Convolutional Neural Networks

MS DSP 458 - Artificial Intelligence and Deep Learning

Sachin Sharma

October 20, 2024

Abstract

This research explores the performance of Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) on image classification tasks, using the CIFAR-10 dataset. The primary objective is to understand how feature extraction and learning differ between simple DNN architectures and deep CNNs. We experiment with different configurations of DNNs and CNNs, incorporating regularization techniques like batch normalization, L2 regularization, and dropout, to improve model performance. We also visualize the feature extraction process of CNNs to examine how convolutional filters capture meaningful patterns. Our results show that CNNs outperform DNNs significantly in terms of accuracy, with further gains observed when applying regularization methods. The study demonstrates that deeper models with convolutional layers extract more abstract and useful features, as evidenced by improved classification accuracy and feature visualizations.

Introduction

As the need for automated image recognition grows, understanding how neural networks learn to extract features from raw image data becomes critical. Neural networks, particularly CNNs, have shown tremendous success in tasks like image classification, but the mechanisms of how they extract and learn features as they deepen are not always intuitive. This research investigates the learning process of DNNs and CNNs to identify the differences in feature extraction capabilities and classification performance.

The specific aim is to conduct a series of experiments on the CIFAR-10 dataset to compare the effectiveness of simple DNNs and more complex CNN architectures. By doing so, we aim to determine the role of multiple hidden layers, pooling operations, and regularization techniques in learning feature representations, improving classification accuracy, and reducing overfitting.

Literature Review

The field of deep learning has rapidly evolved, particularly in image recognition, where CNNs have become the de facto standard. **LeCun et al. (1998)** laid the groundwork for CNNs with their work on handwritten digit classification (LeNet), which showcased the effectiveness of convolutional layers in extracting spatial hierarchies. Since then, **Krizhevsky et al. (2012)** demonstrated the breakthrough of deep CNNs with the AlexNet model on the ImageNet competition, which significantly improved classification accuracy. Researchers have since introduced various architectures, such as **VGGNet (Simonyan & Zisserman, 2014)**, **ResNet (He et al., 2016)**, and **InceptionNet (Szegedy et al., 2015)**, all of which pushed the limits of accuracy using deeper models and novel architectural innovations. Regularization techniques like dropout (**Srivastava et al., 2014**) and batch normalization (**Ioffe & Szegedy, 2015**) have also been shown to improve the generalization capabilities of deep models by preventing overfitting and stabilizing the training process.

For DNNs, the performance often saturates as additional layers are added without convolution, particularly on image data. Thus, CNNs remain the gold standard for image-related tasks due to their ability to extract meaningful hierarchical features, as explored by **Goodfellow et al. (2016)** in their book *Deep Learning*.

Methods

Our research focused on comparing DNN and CNN architectures on the CIFAR-10 dataset to understand their performance in image classification tasks. The research involved several key steps:

1. **Baseline Models:** We initially built and trained four baseline models: two Deep Neural Networks (DNN) with 2 and 3 layers, and two Convolutional Neural Networks (CNN) with 2 and 3 convolutional/max pooling layers.

2. **Regularized Models:** We then introduced various regularization techniques—batch normalization, dropout, and L2 regularization—to the baseline models to evaluate their impact on classification accuracy and generalization.
3. **Performance Evaluation:** For each model, we tracked performance metrics such as training time, training accuracy, validation accuracy, and loss. We plotted learning curves to analyze the models' behaviors across epochs, and employed early stopping to prevent overfitting. Confusion matrices provided insights .
4. **Feature Visualization:** We visualized the outputs from filters in the convolutional layers to observe how the models learned to extract patterns from the images.

Implementation

The models were developed using the **Keras** and **TensorFlow** frameworks in Python, chosen for their flexibility and ease of use. Key configurations included:

- **Activation Functions:** We used the ReLU activation function for hidden layers and softmax for the output layer.
- **Optimizer and Loss Function:** The Adam optimizer was employed alongside SparseCategoricalCrossentropy for multi-class classification.
- **Hyperparameters:** Consistent hyperparameters, such as batch size (64) and epochs (200), were maintained for fair comparisons.

Dataset: CIFAR-10

The **CIFAR-10** dataset comprises 60,000 32x32 color images across 10 categories. The dataset was split into training (45,000), validation (5,000), and testing (10,000) sets, with pixel values normalized between 0 and 1.

Exploratory Data Analysis (EDA)

During the EDA phase, we visualized sample images and analyzed label distribution to ensure a balanced dataset across all classes. This step helped us understand the dataset's characteristics and identify potential challenges in image classification.

Regularization Techniques

To address overfitting and enhance model performance, we utilized:

- **Batch Normalization:** This normalized inputs to layers, stabilizing learning and allowing for higher learning rates.
- **L2 Regularization (Ridge Regression):** Applied to penalize large weights, promoting simpler model representations.
- **Dropout:** This randomly ignored certain neurons during training, aiding generalization.

Model Training and Early Stopping

Training was conducted on a GPU-accelerated machine featuring an **NVIDIA L4 Tensor Core GPU**, 16 vCPUs, and 64 GB of RAM. Early stopping was employed to halt training if validation accuracy did not improve after a set number of epochs, ensuring optimal weights were retained before overfitting occurred.

Results

A total of eleven experiments were conducted, comparing the performance of models with varying architectures and regularization techniques. The key performance metrics, including training time, test accuracy, test loss, training accuracy, training loss, validation accuracy, and validation loss, revealed notable patterns across different network types and configurations. Detailed results for each experiment can be found in the appendix, providing a comprehensive view of model behaviour and performance.

Key Findings

1. **DNN vs. CNN Performance:**
    - **DNN Models (Experiments 1 and 2)** struggled with classification accuracy, both achieving a test accuracy of only **47.4%**. Although deeper layers (3-layer

DNN) improved training accuracy significantly, this did not translate to improved test accuracy, with a notable gap between training and validation performance. This indicates that DNNs, even with added layers, tend to overfit to the training data in complex tasks like image classification, where spatial relationships in pixels are key.

- **CNN Models (Experiments 3 and 4)** clearly outperformed the DNN models, achieving **71.3%** and **73.6%**test accuracy for the 2-layer and 3-layer CNNs, respectively. CNNs excelled at extracting important visual features from images, leveraging convolutional filters and pooling operations. These results affirm the superiority of CNNs in tasks involving visual data due to their ability to capture local and global patterns, making them much more suited for the CIFAR-10 dataset.

2. **Impact of Regularization Techniques:**
   - The application of **L2 Regularization, Batch Normalization, and Dropout** in later experiments significantly enhanced the models' robustness. Notably, **Experiment 10**, a 3-layer CNN with 2 fully connected layers, L2 regularization, and dropout, achieved the highest test accuracy of **81.1%**. This model balanced complexity and regularization, maintaining high accuracy while preventing overfitting.
   - The **Dropout and Batch Normalization** techniques (used in Experiments 9 and 10) were particularly effective, promoting generalization and leading to improved test and validation accuracy. The success of these methods highlights the importance of regularization in deep learning models, particularly when dealing with complex datasets like CIFAR-10 that can easily lead to overfitting in deeper architectures.

3. **Training Time and Model Complexity:**
    - As expected, the **training time increased** with more complex architectures and the addition of regularization techniques. The simplest DNN (Experiment 1) completed training in **21.78 seconds**, while the most complex CNN (Experiment 10) required **284.72 seconds**. While more advanced CNNs with regularization required significantly longer training times, the substantial gains in test accuracy and model stability justify the added computational cost.

Conclusions

This study provided a comparative analysis of Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) on the CIFAR-10 image classification task, highlighting the strengths and weaknesses of both architectures, as well as the impact of various regularization techniques. The key takeaway from this research is the clear advantage CNN architectures hold over DNNs in handling image data. CNNs, with their ability to capture spatial hierarchies through convolutional layers, outperformed DNNs significantly in terms of accuracy and generalization. Even with additional layers and without regularization, DNNs struggled to extract the complex features needed to classify images accurately, demonstrating their limitations in image classification tasks.

Regularization techniques such as L2 regularization, batch normalization, and dropout were crucial in improving model performance and mitigating overfitting. The best-performing model, a CNN with 3 convolutional layers, 2 fully-connected layers, L2 regularization, dropout, and batch normalization, achieved an impressive **test accuracy of 81.1%**. This result shows the importance of carefully designed network architectures and regularization techniques in enhancing model robustness and generalization to unseen data.

References

Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791 https://ieeexplore.ieee.org/document/726791

Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386. https://www.researchgate.net/publication/267960550_ImageNet_Classification_with_Deep_Convolutional_Neural_Networks

Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. https://www.researchgate.net/publication/319770291_Very_Deep_Convolutional_Networks_for_Large-Scale_Image_Recognition

He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90. https://www.researchgate.net/publication/311609041_Deep_Residual_Learning_for_Image_Recognition

C. Szegedy *et al*., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594. https://ieeexplore.ieee.org/document/7298594

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1 (January 2014), 1929–1958. https://dl.acm.org/doi/10.5555/2627435.2670313

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15). JMLR.org, 448–456. https://dl.acm.org/doi/10.5555/3045118.3045167

Goodfellow, I., et al. (2016) Deep Learning. MIT Press, Cambridge, MA. http://www.deeplearningbook.org

Appendix A – Accuracy Results from Experiments

The table below displays the training, validation, and testing accuracy of each of the 11 models developed for classification of CIFAR-10 images.

Result1: Table with the accuracy and loss for train/test/validation & process time for ALL the models

	Architecture	Train Time	Test Accuracy	Test Loss	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss
Experiment1	• DNN with 2 layers • no regularization	21.78 seconds	0.474	1.478	0.53	1.311	0.461	1.55
Experiment2	• DNN with 3 layers • no regularization	36.06 seconds	0.474	1.466	0.762	0.655	0.454	2.307
Experiment3	• CNN with 2 layers/max pooling layers • 1 full-connected layer • no regularization	60.82 seconds	0.713	0.865	0.97	0.085	0.685	2.025
Experiment4	• CNN with 3 layers/max pooling layers • 1 full-connected layer • no regularization	119.0 seconds	0.736	0.794	0.985	0.048	0.723	2.322
Experiment5	• DNN with 2 layers (384, 768) • Batch Normalization • L2 Regularization(0.001)	41.89 seconds	0.49	1.482	0.65	0.991	0.477	1.637
Experiment6	• DNN with 3 layers • Regularization: batch normalization	56.52 seconds	0.483	1.48	0.792	0.581	0.478	2.413
Experiment7	• CNN with 2 layers/max pooling layers • L2 Regularization(0.001)	145.38 seconds	0.701	0.907	0.993	0.022	0.724	1.894
Experiment8	• CNN with 3 layers/max pooling layers • L2 Regularization(0.001)	101.77 seconds	0.699	0.944	0.984	0.047	0.701	2.117
Experiment9	• CNN with 3 layers/max pooling layers • Dropout(0.3) • L2 Regularization(0.001) • Batch Normalization	185.3 seconds	0.798	0.743	0.853	0.574	0.795	0.746
Experiment10	• CNN with 3 layers/max pooling layers • 2 Fully-Connected Hidden Layers • Dropout(0.3) • L2 Regularization(0.001) • Batch Normalization	284.72 seconds	0.811	0.741	0.88	0.49	0.804	0.772
Experiment11	• CNN with 3 layers/max pooling layers • 2 Fully-Connected Hidden Layers • Dropout(variable) • L2 Regularization(0.001) • Batch Normalization	173.08 seconds	0.772	0.944	0.741	1.046	0.771	0.956

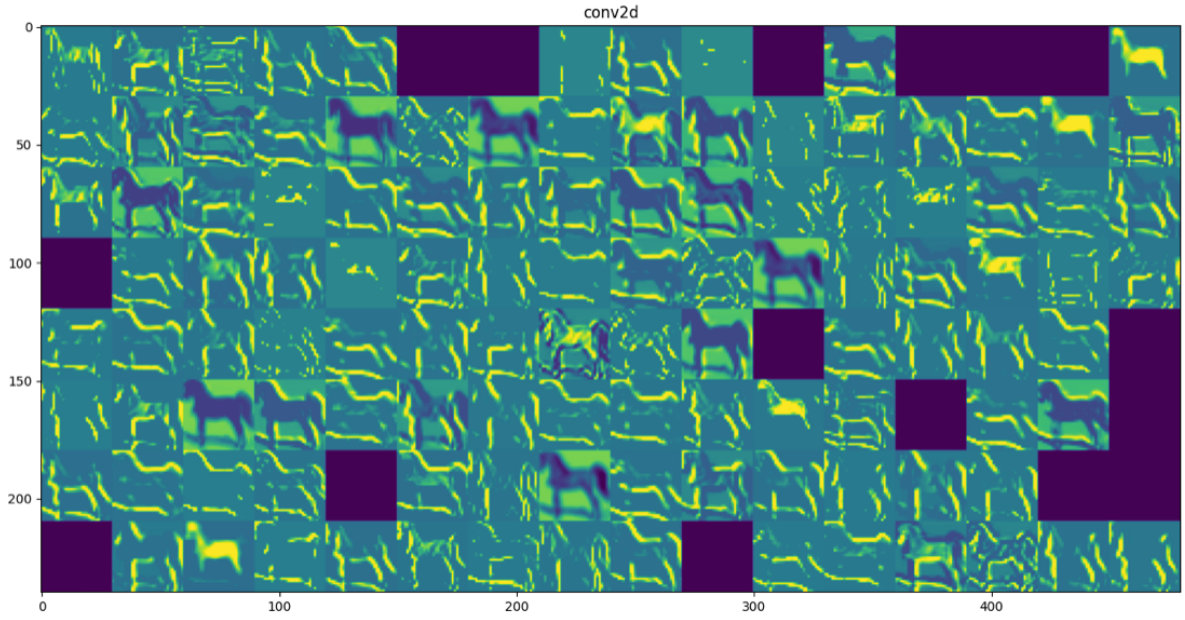
Result2

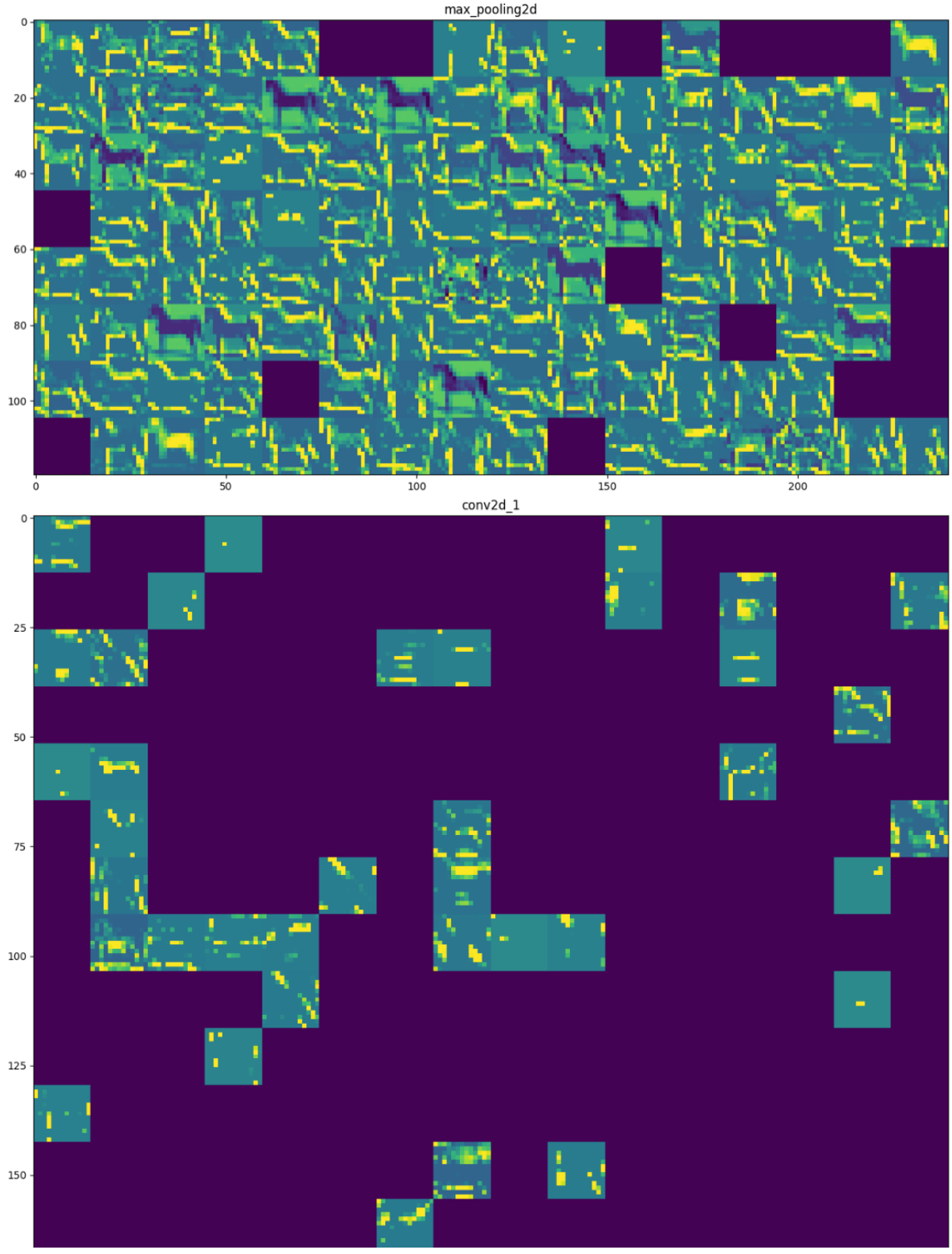
In Experiment 3 (CNN with 2 convolutional/max pooling layers), we extracted the outputs from two filters in each of the max pooling layers to visualize how the network learned to detect features from the input images. After training, we visualized the feature maps by displaying the "lighted up" regions (i.e., the areas of the image that activated the filter most strongly).

Observation:

Upon visualizing the feature maps, we observed that the "lighted up" regions in the filters corresponded to distinct features of the original images. For example, the filters in the first max pooling layer detected simple patterns such as edges and corners, while the filters in the second max pooling layer identified more abstract and higher-level features like textures or object shapes. These visualized regions align well with meaningful parts of the original images, demonstrating that the CNN effectively learned to extract hierarchical features as it progressed through the layers.

This confirms that the convolutional layers are capable of learning useful feature representations that are abstracted as the network deepens, which is crucial for improved classification accuracy.

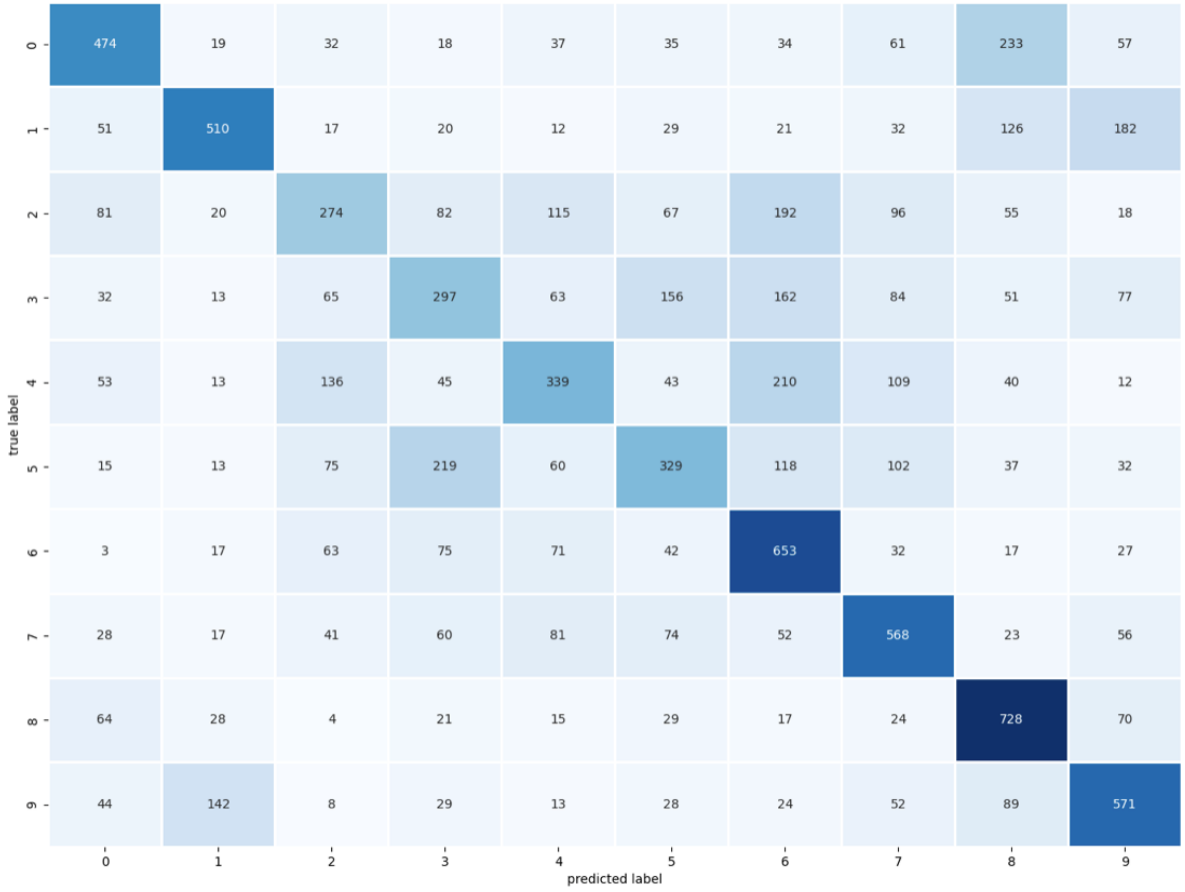



max_pooling2d
0
20
40
60
80
100
0
50
100
150
200
conv2d_1
0
25
50
75
100
125
150

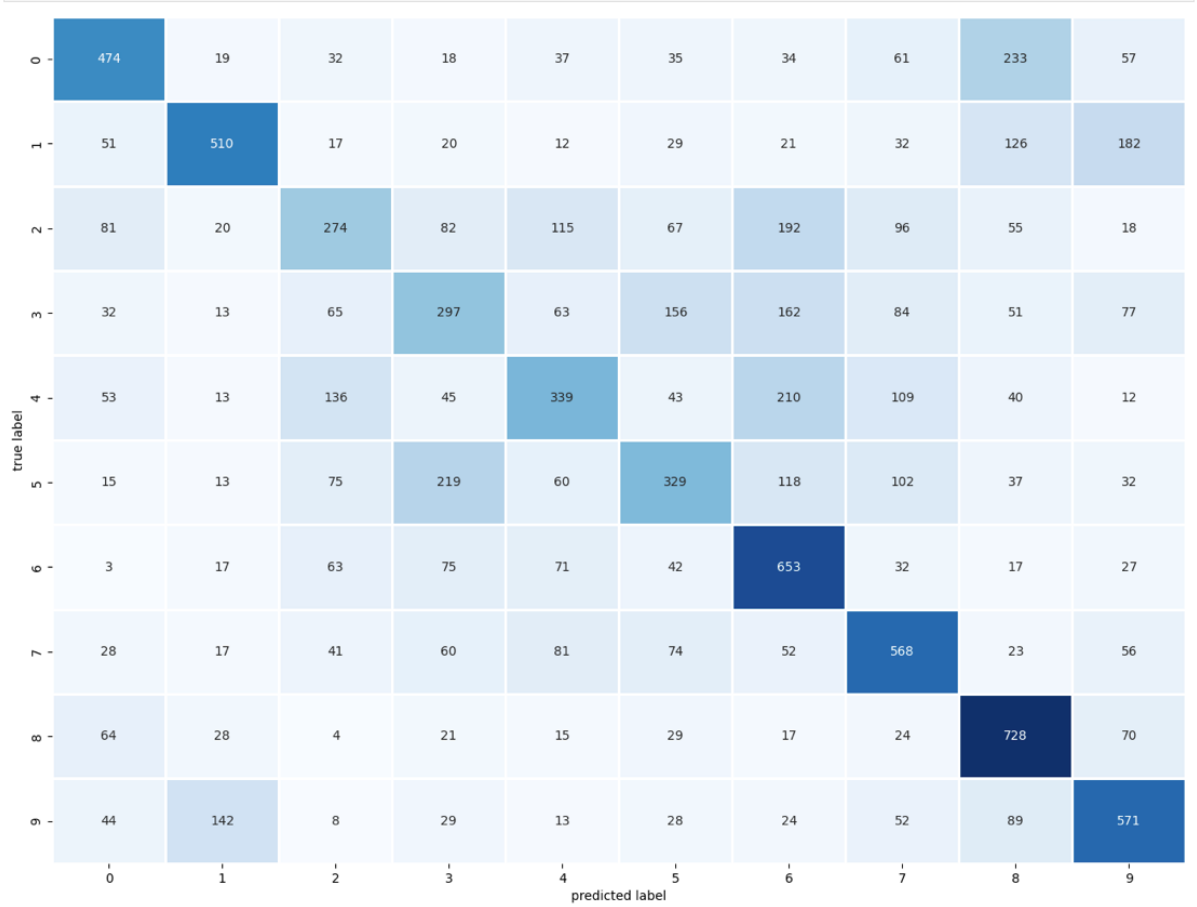
Appendix B – Confusion Matrices Resulting from Experiments

The images below display the confusion matrices resulting from the application of each CIFAR-10 classification model the testing dataset.

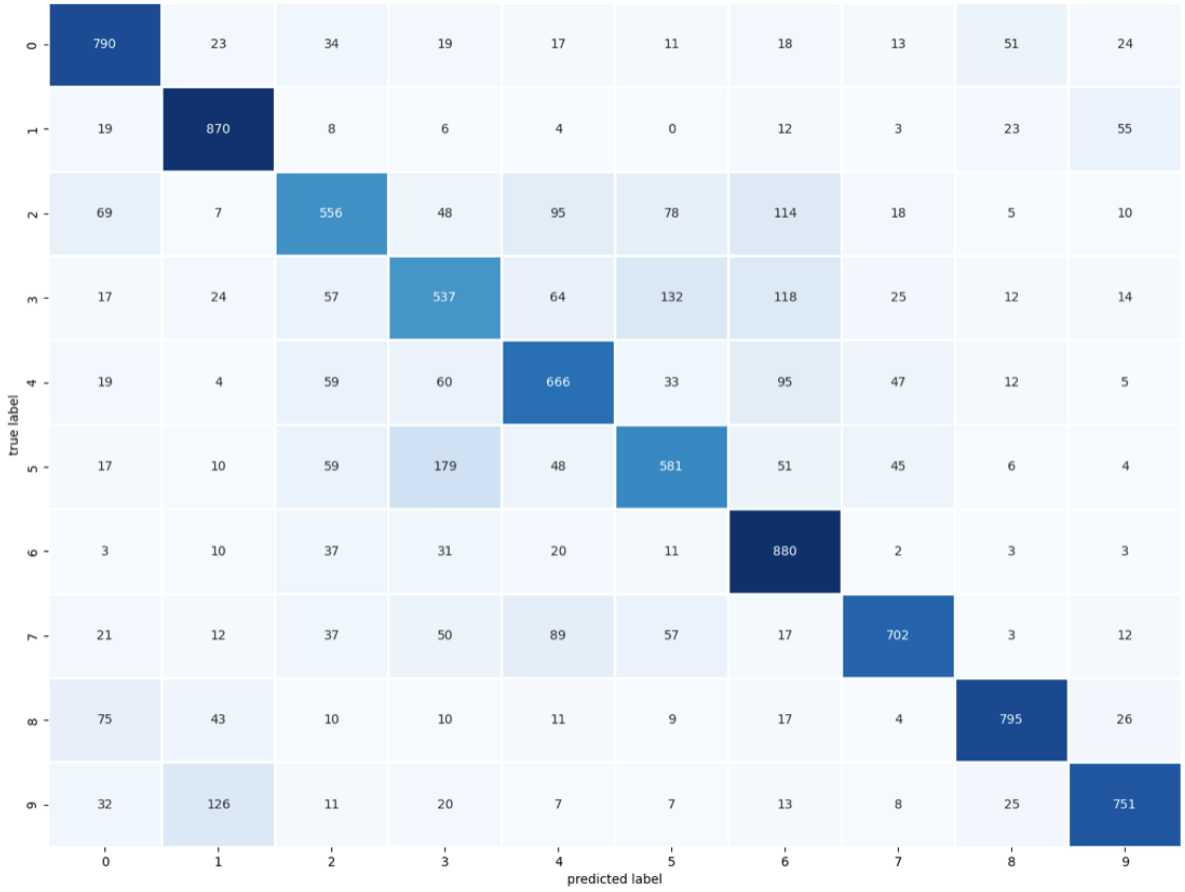
Experiment 1



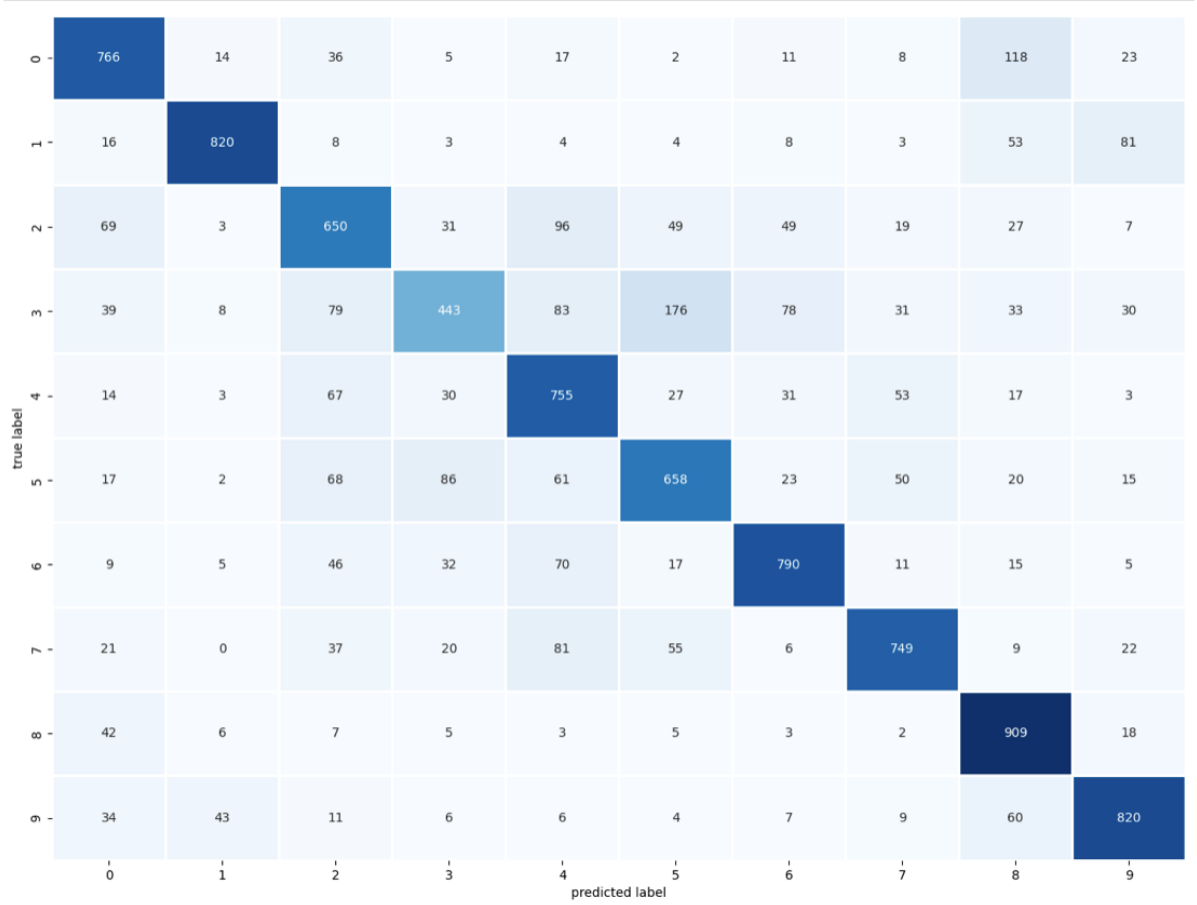
Experiment 2



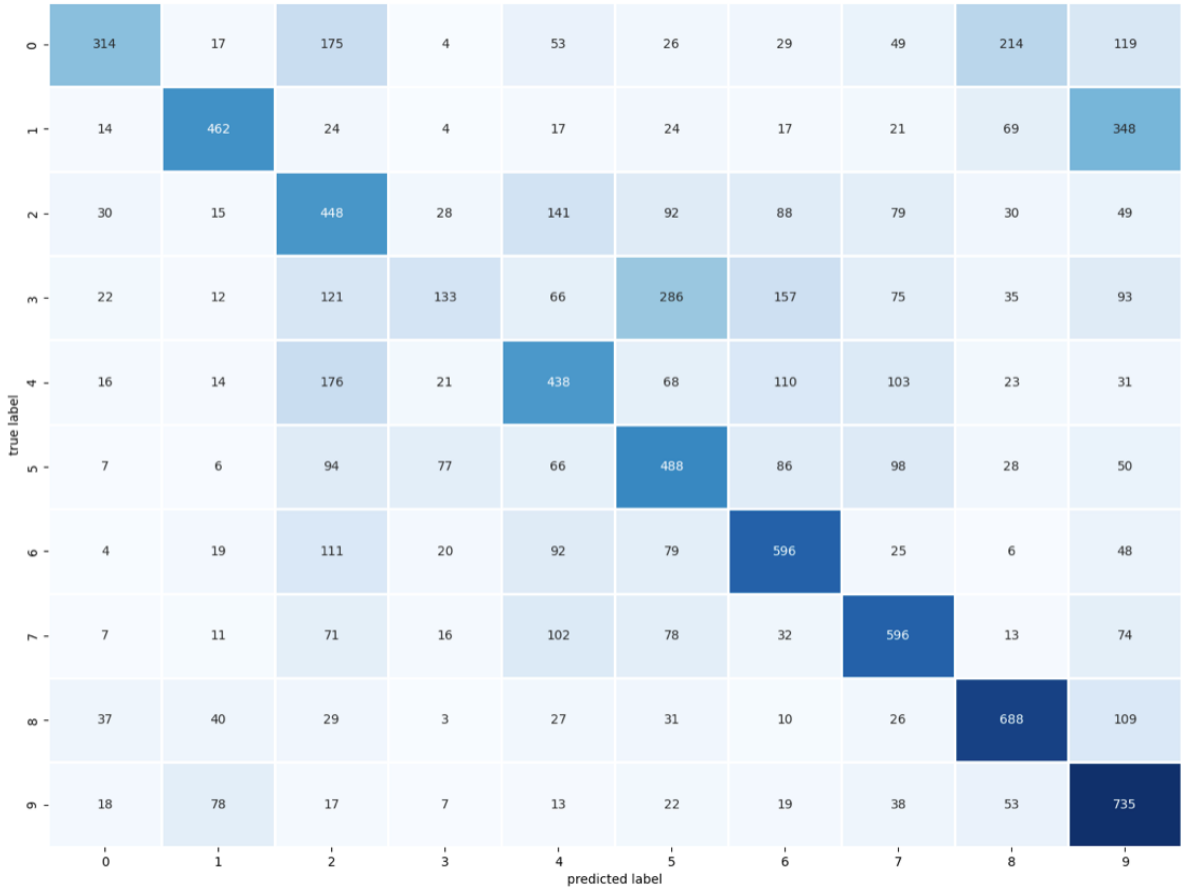
Experiment 3



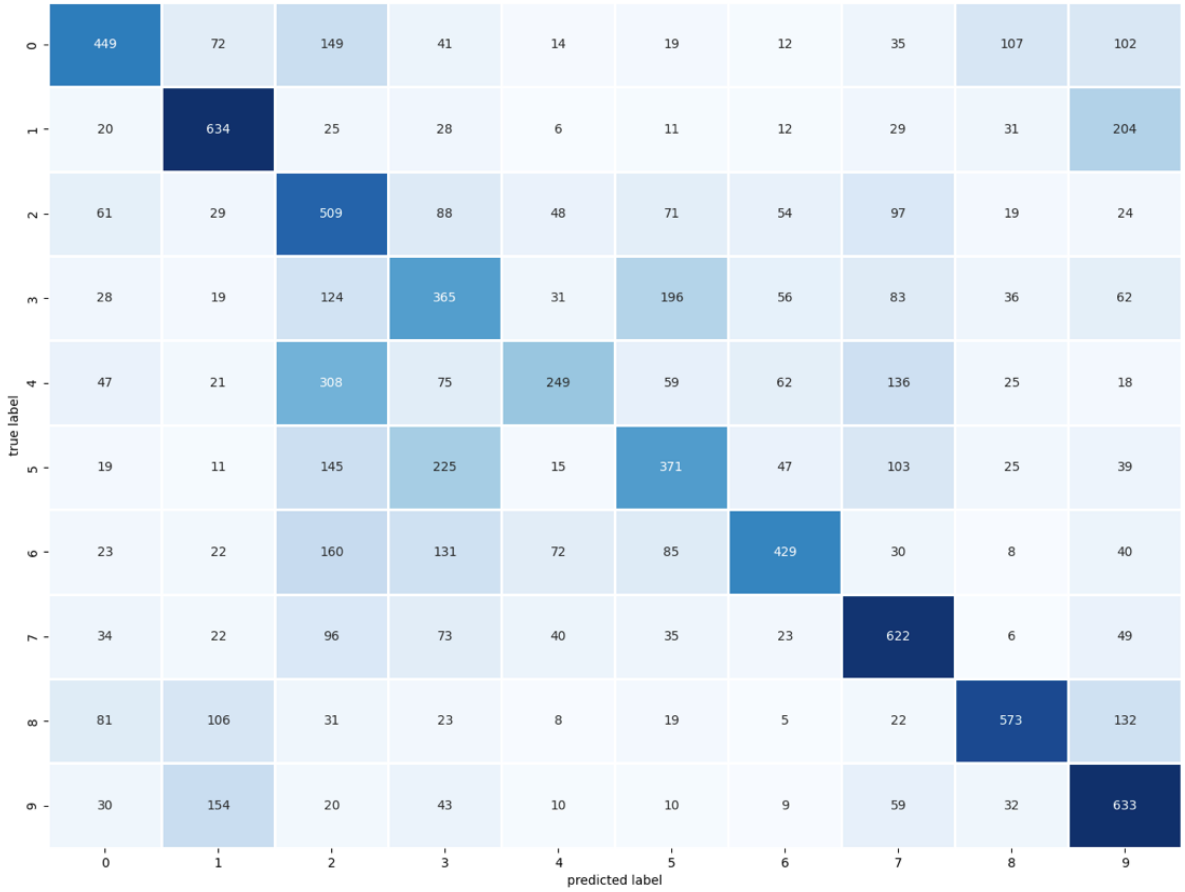
Experiment 4



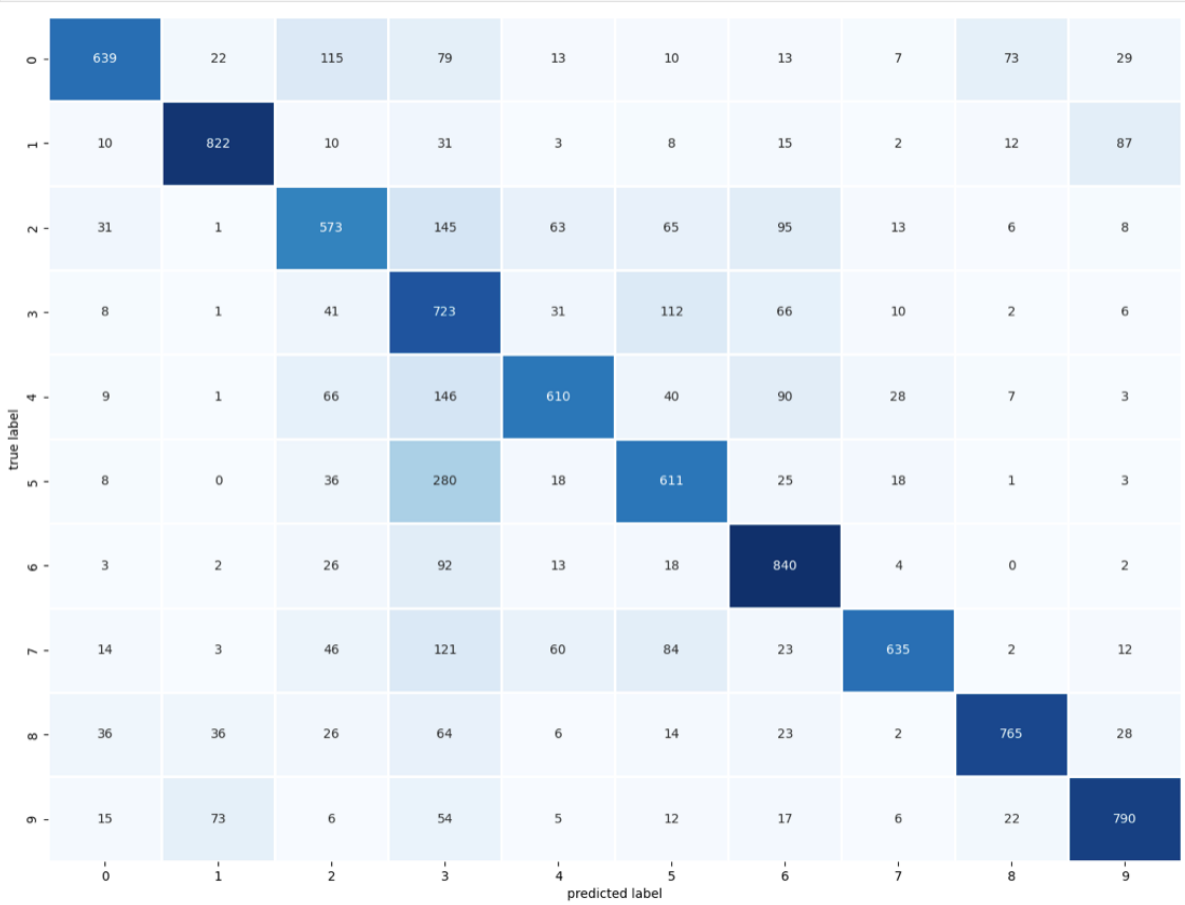
Experiment 5



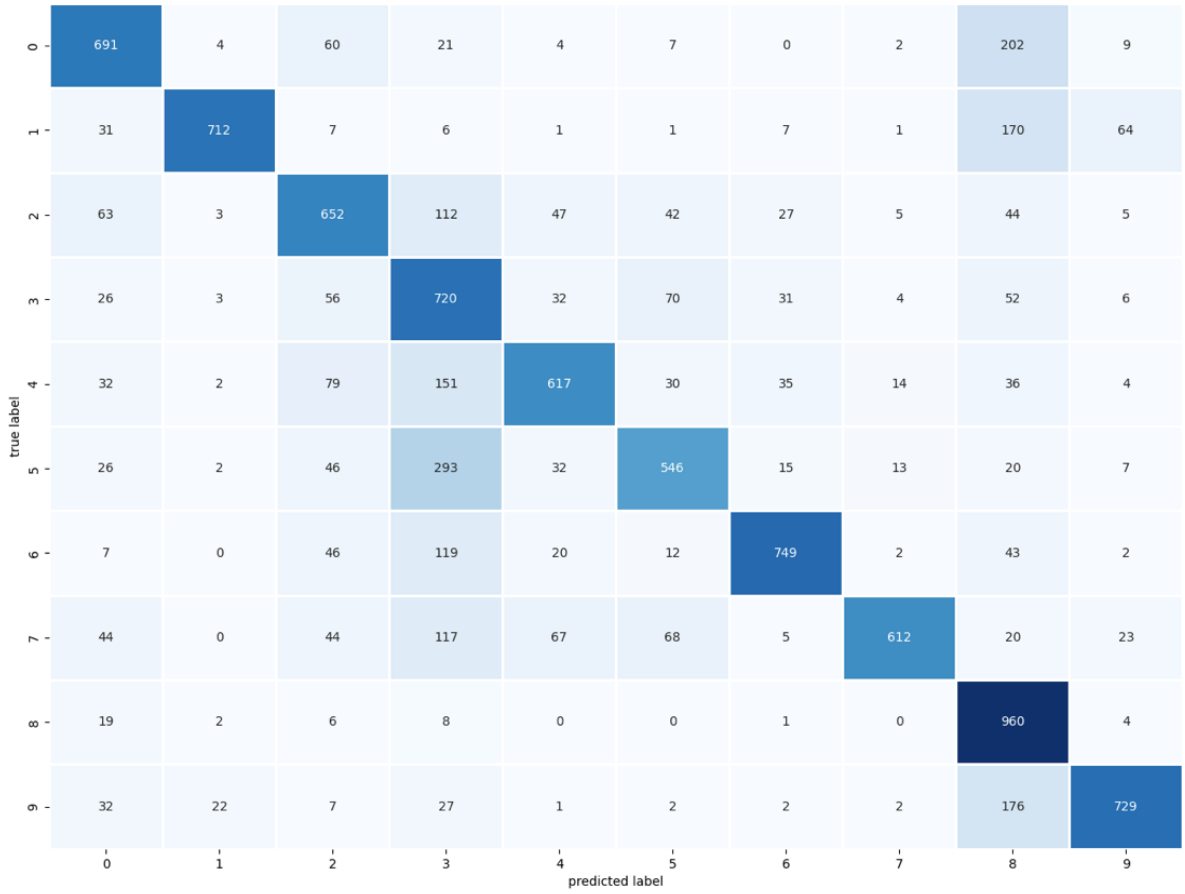
Experiment 6



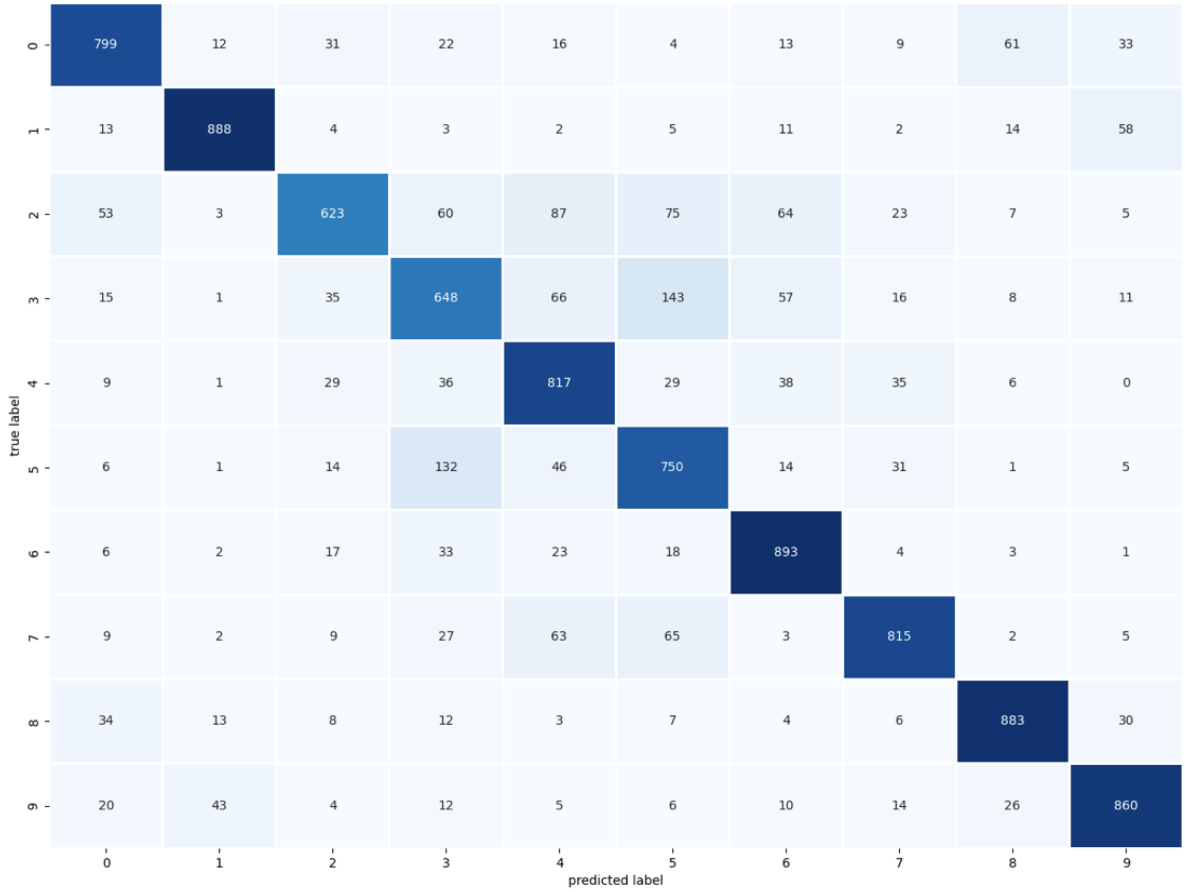
Experiment 7



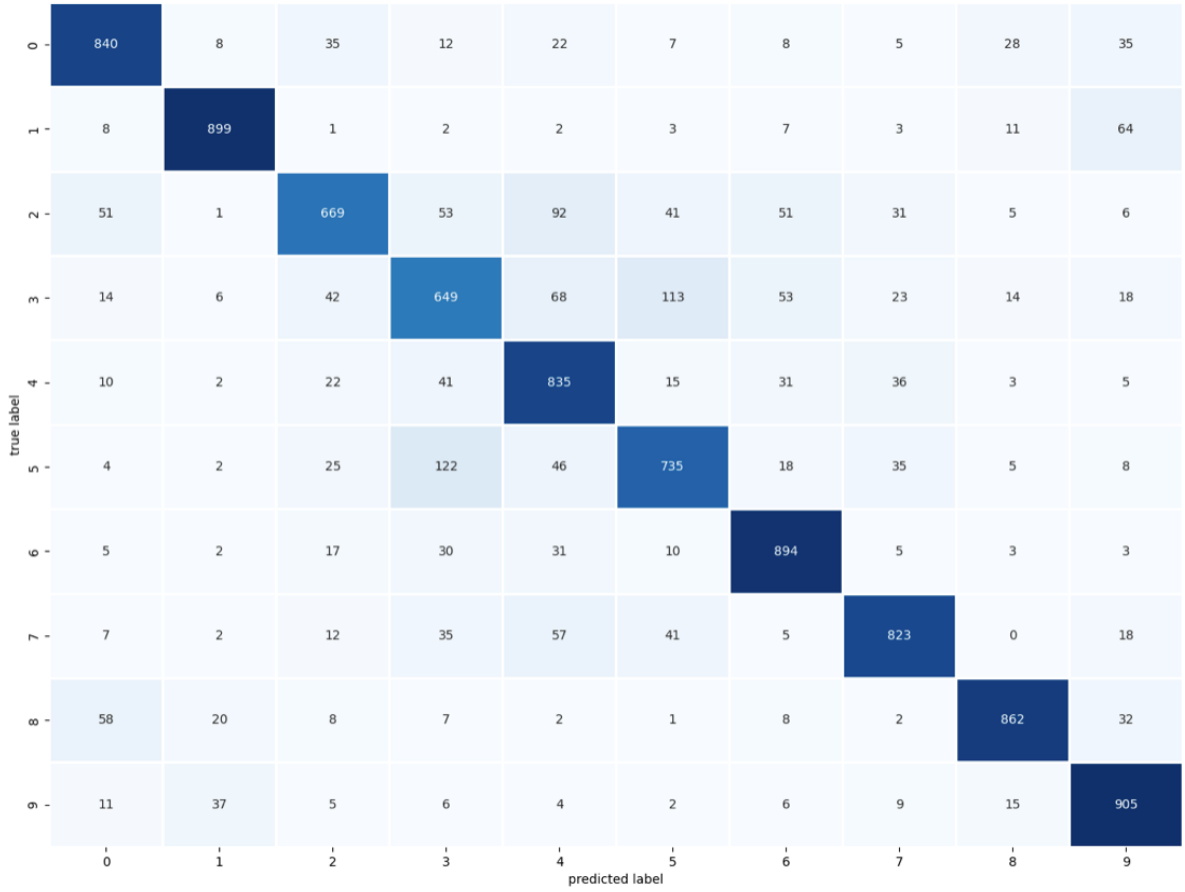
Experiment 8



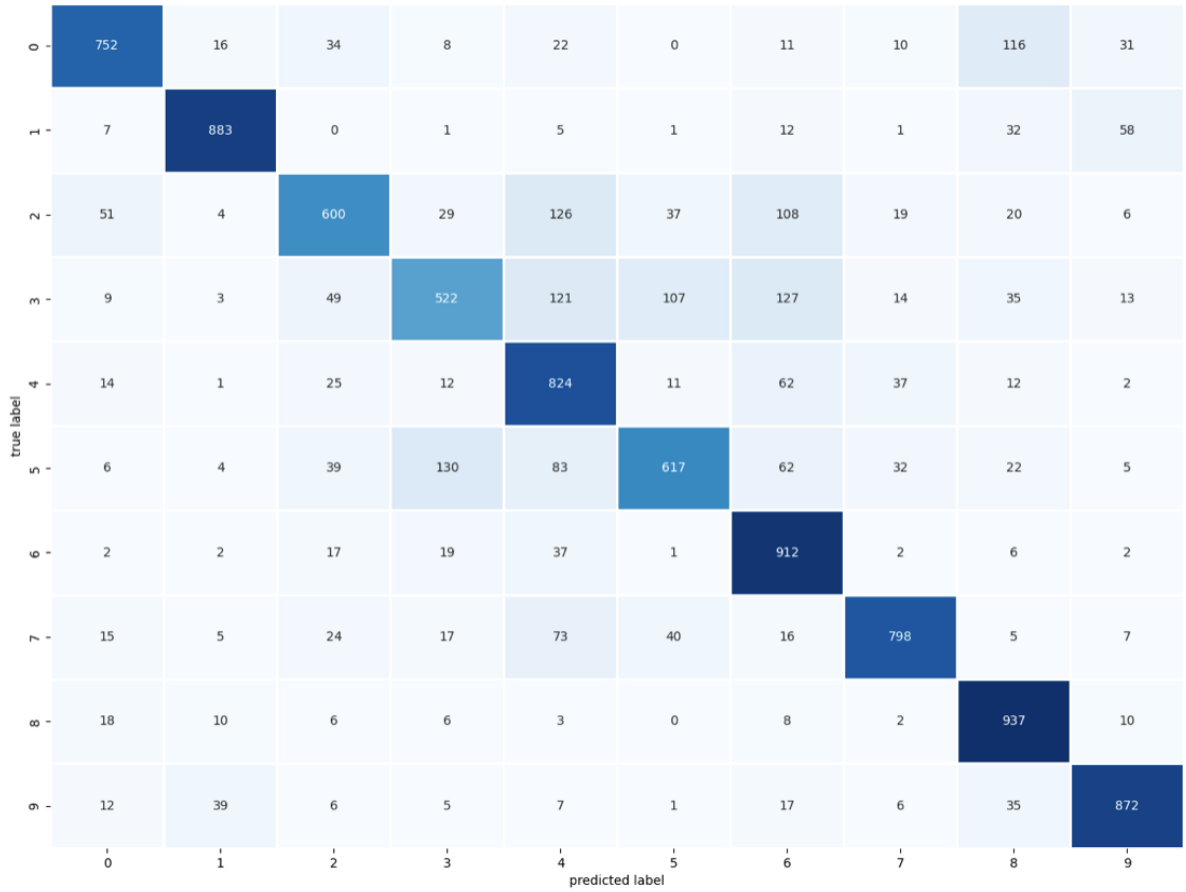
Experiment 9



Experiment 10



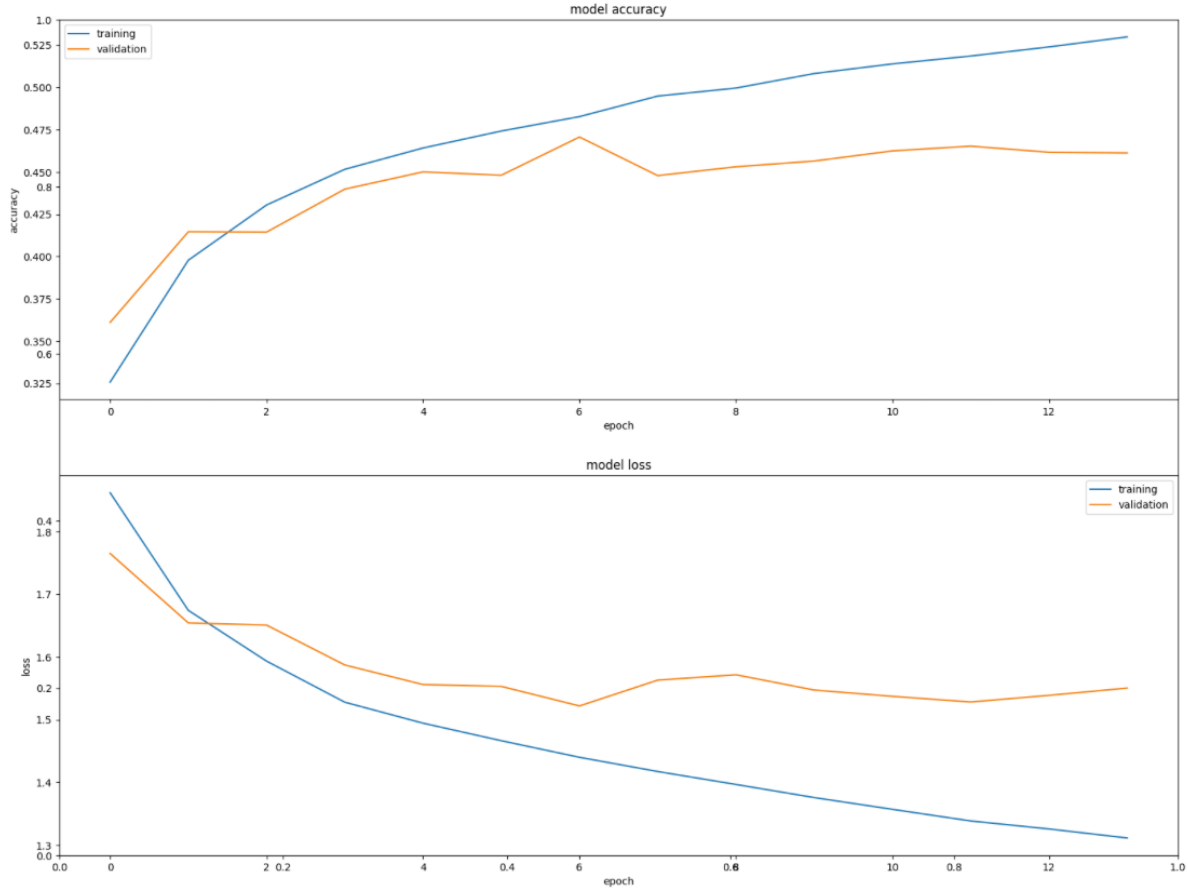
Experiment 11



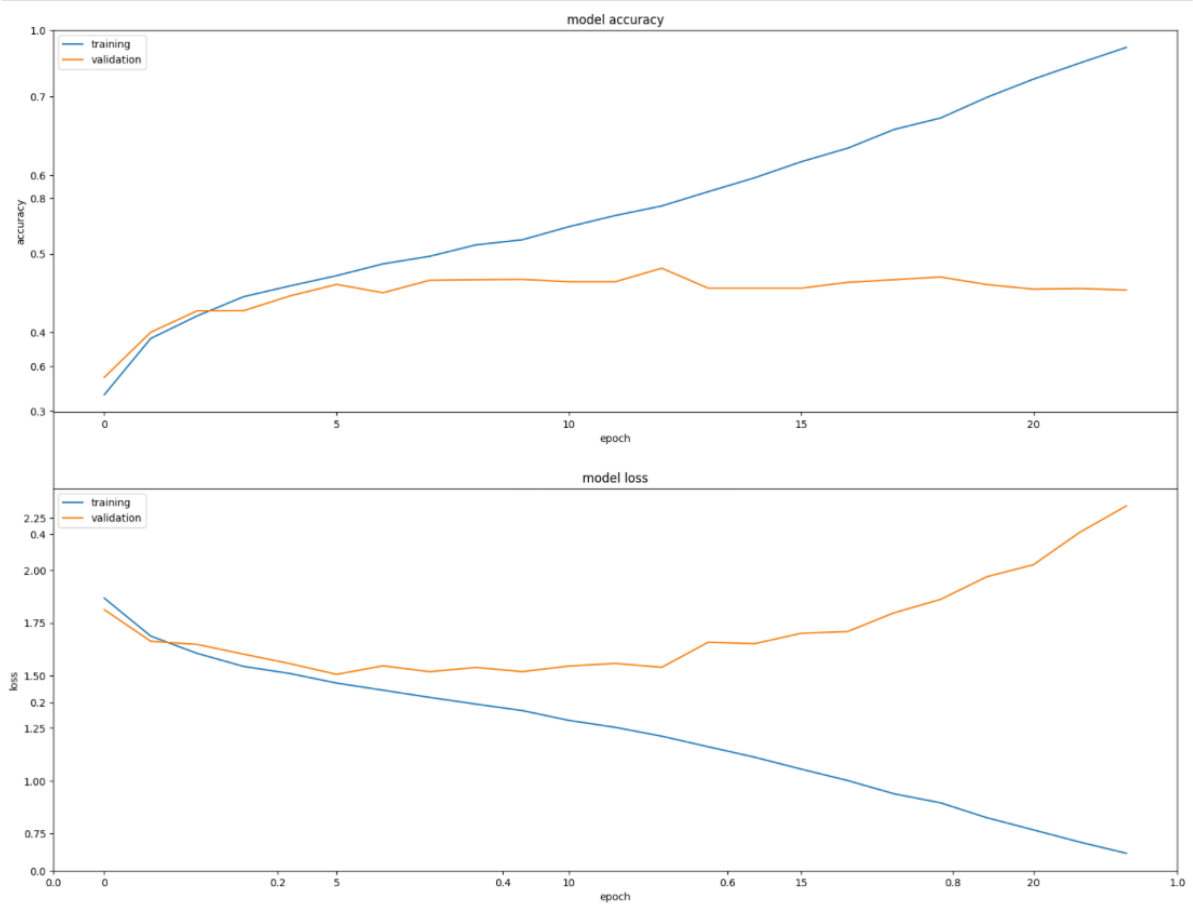
Appendix C – Accuracy and Loss Trends by Epoch Resulting From Experimental Model Fitting

The graphs below display the training and validation accuracies generated throughout each epoch of training each of the CIFAR-10 image classification models.

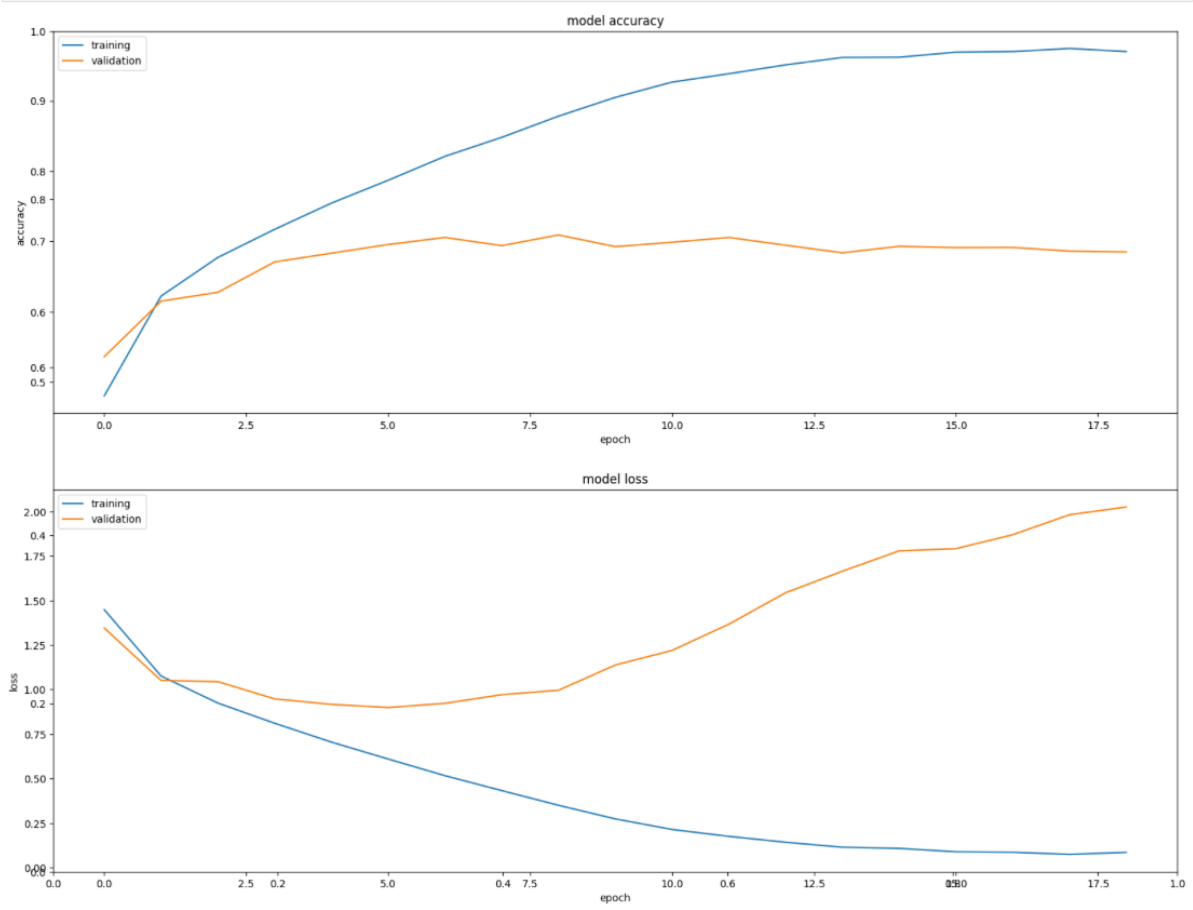
Experiment 1 – Accuracy and Loss Trends During Model Fitting



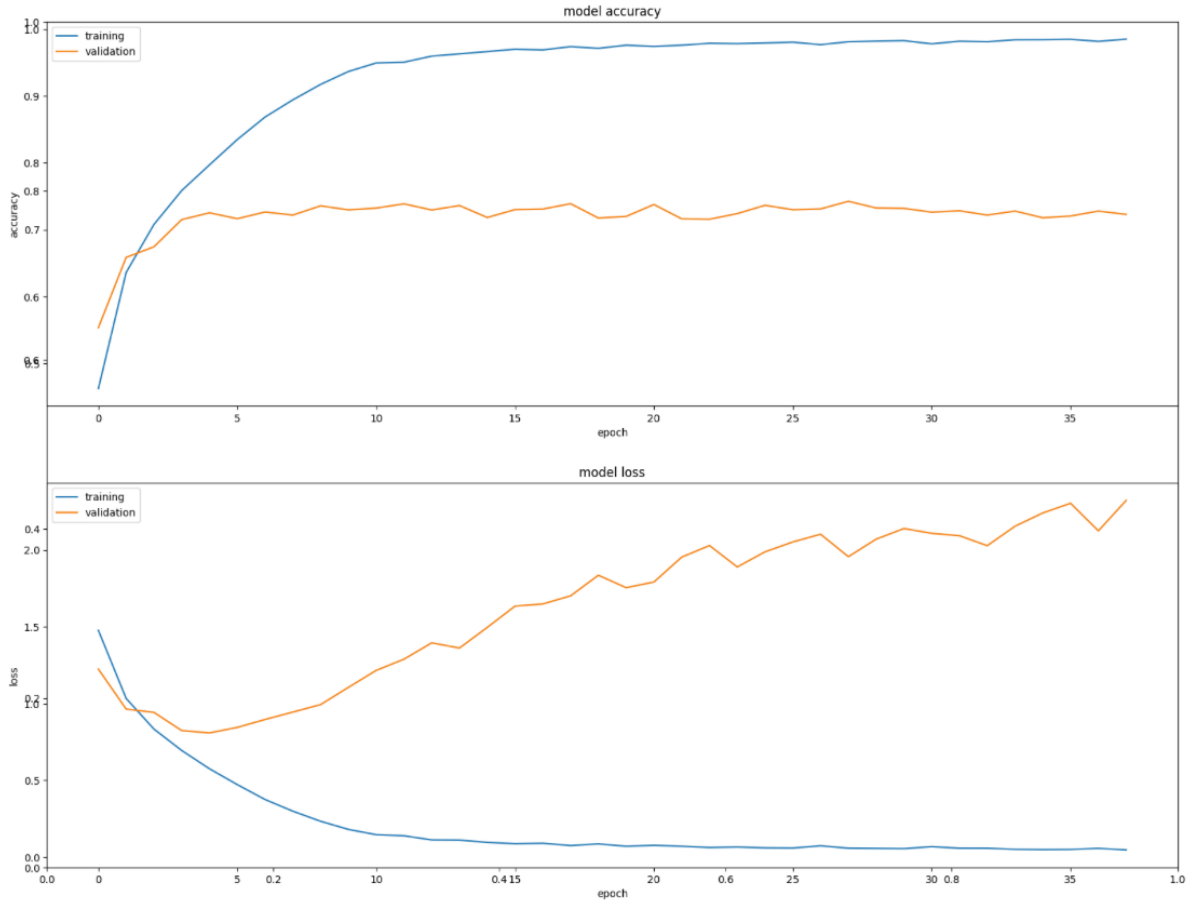
Experiment 2 – Accuracy and Loss Trends During Model Fitting



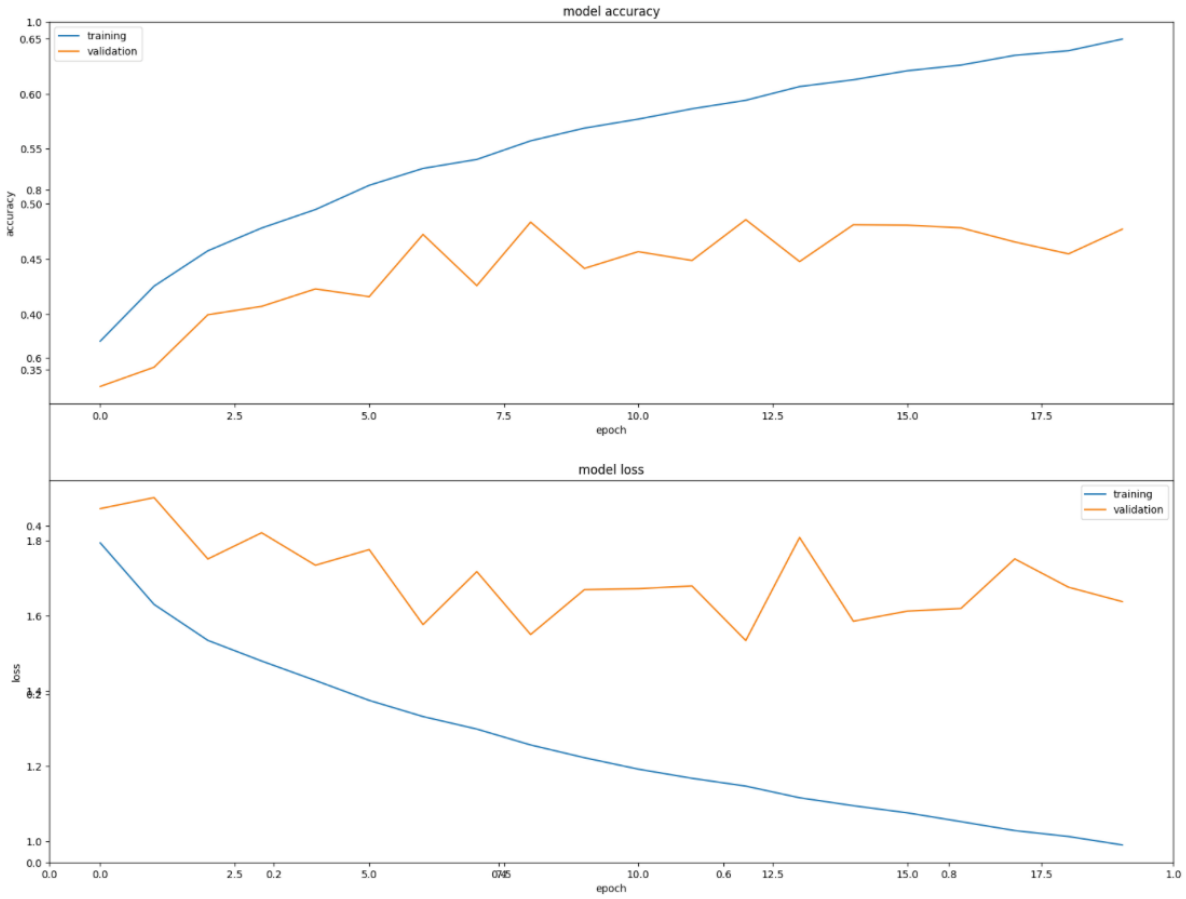
Experiment 3 – Accuracy and Loss Trends During Model Fitting



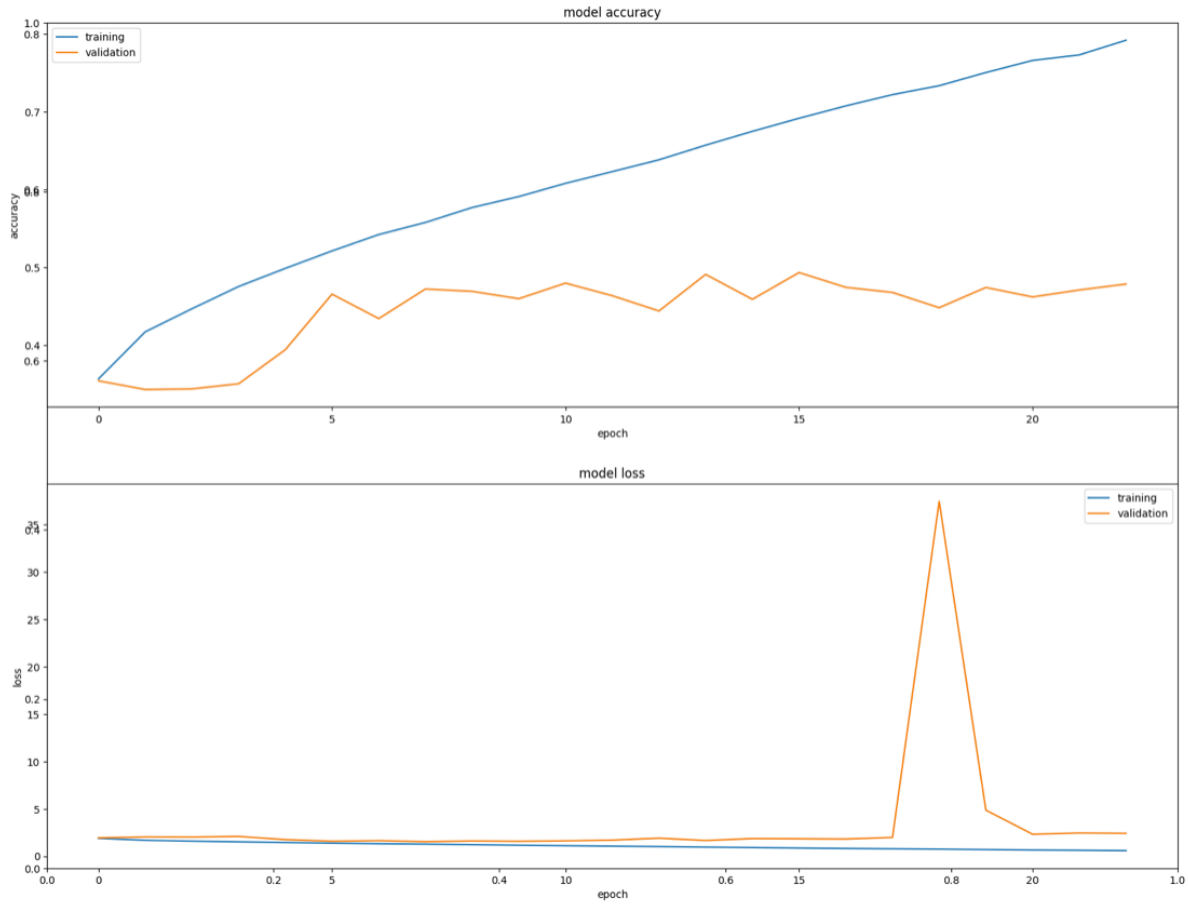
Experiment 4 – Accuracy and Loss Trends During Model Fitting



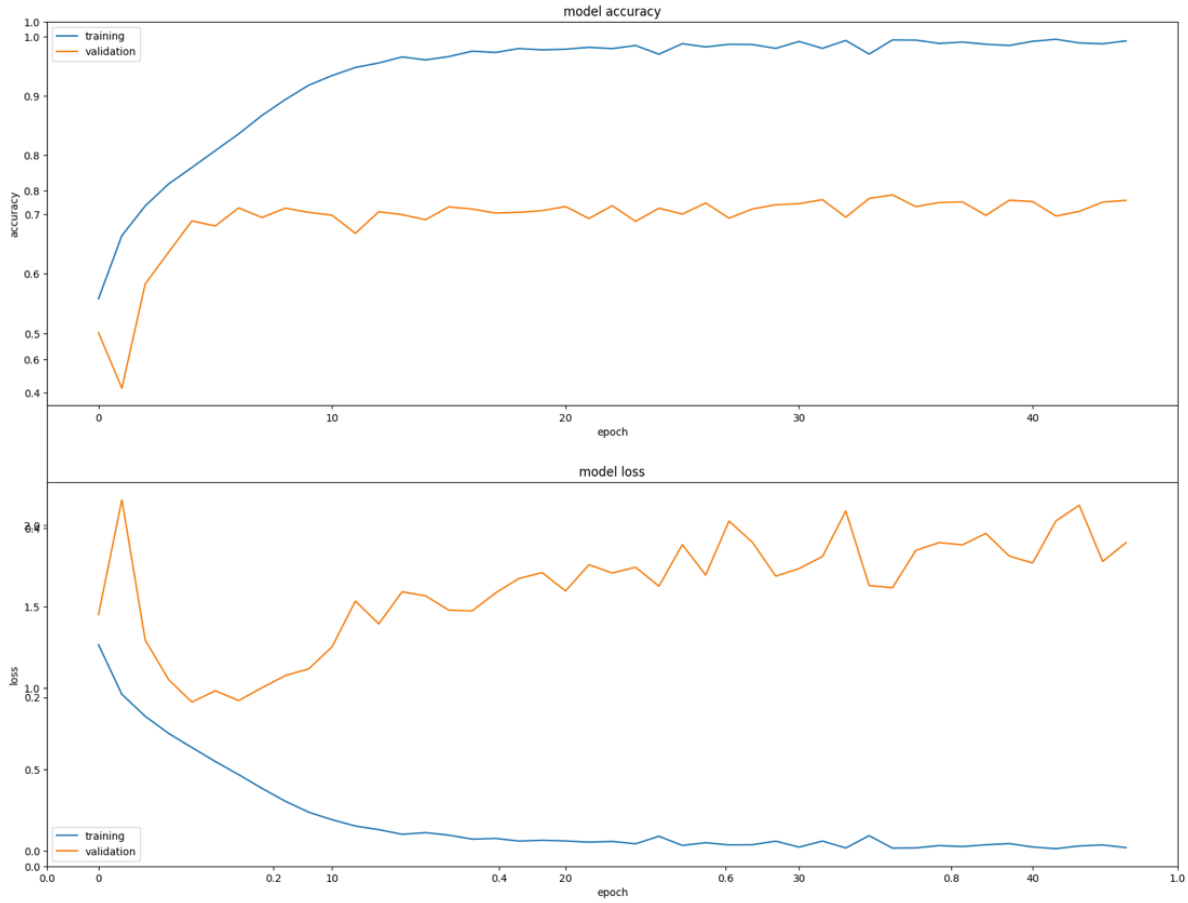
Experiment 5 – Accuracy and Loss Trends During Model Fitting



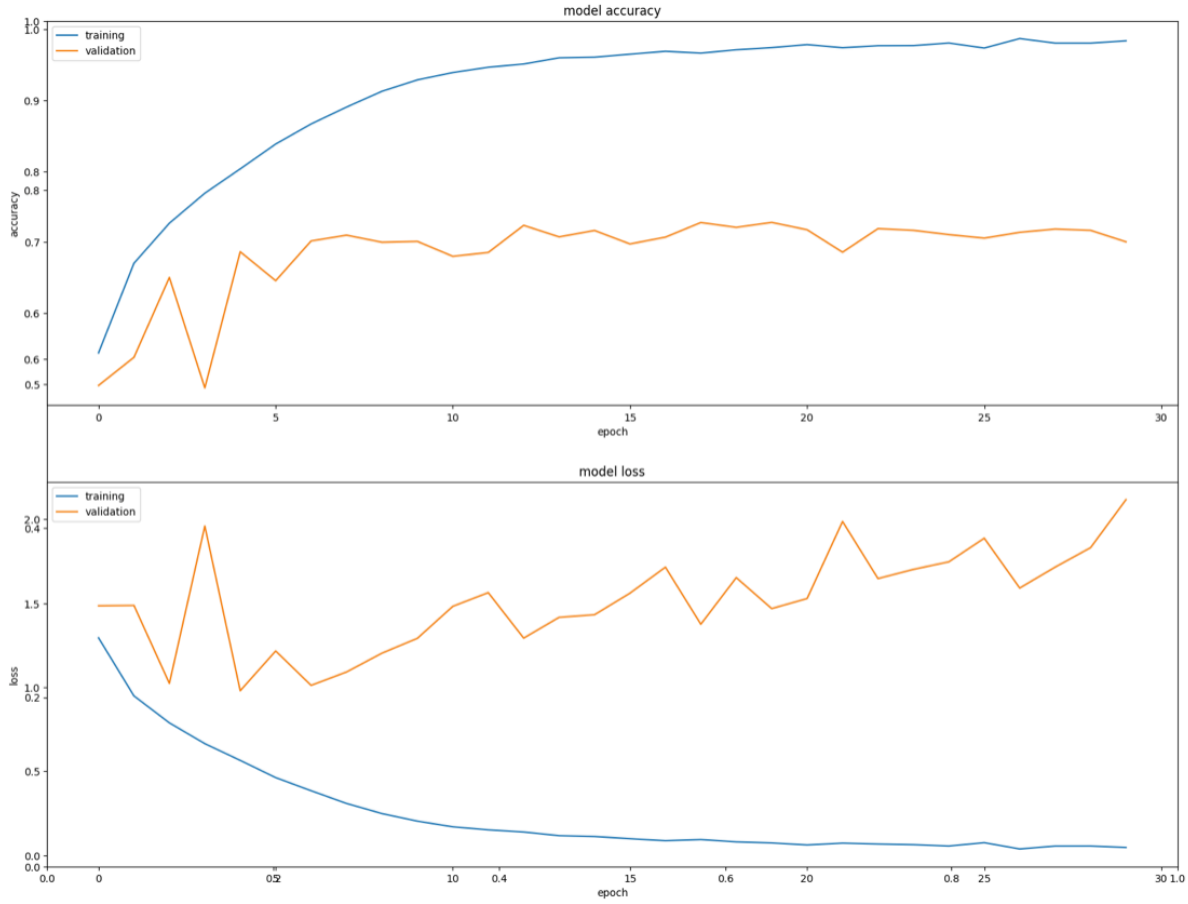
Experiment 6 – Accuracy and Loss Trends During Model Fitting



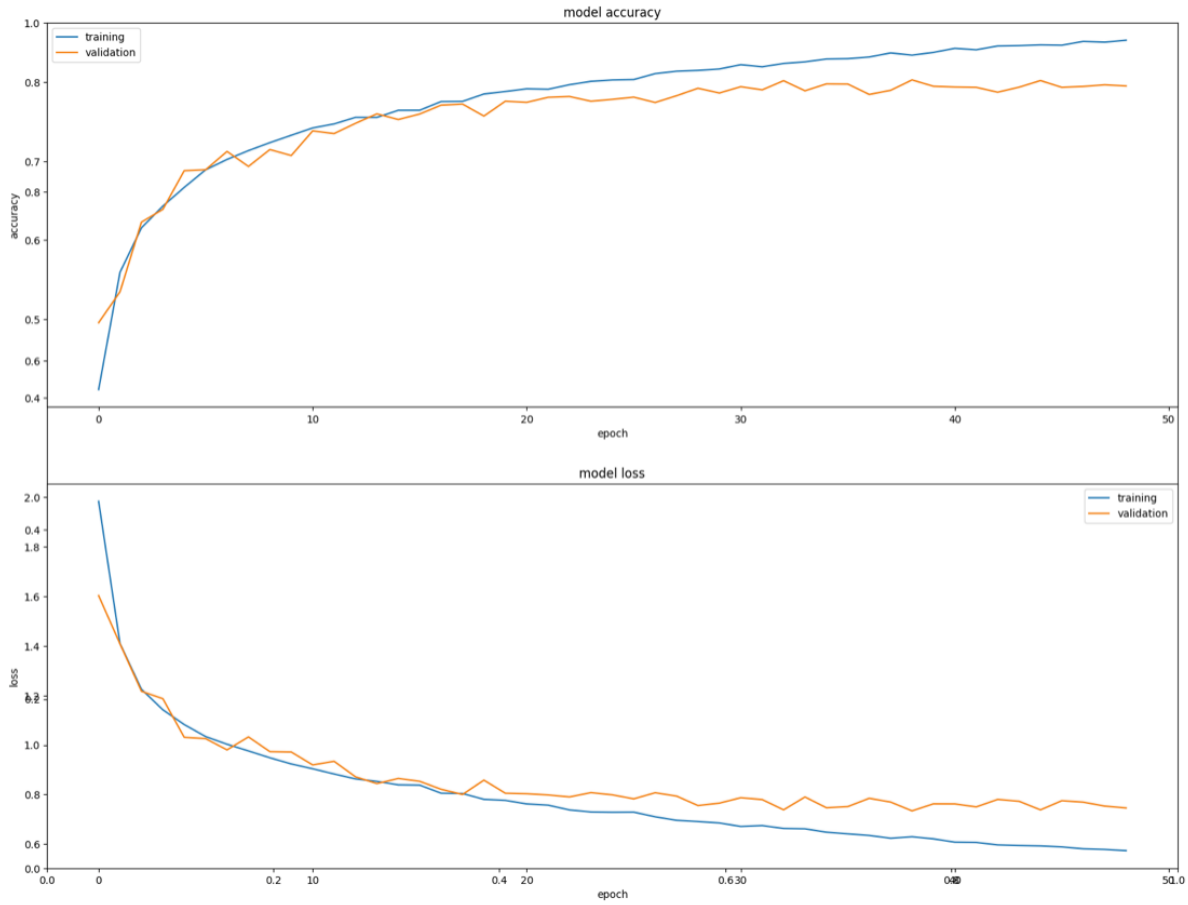
Experiment 7 – Accuracy and Loss Trends During Model Fitting



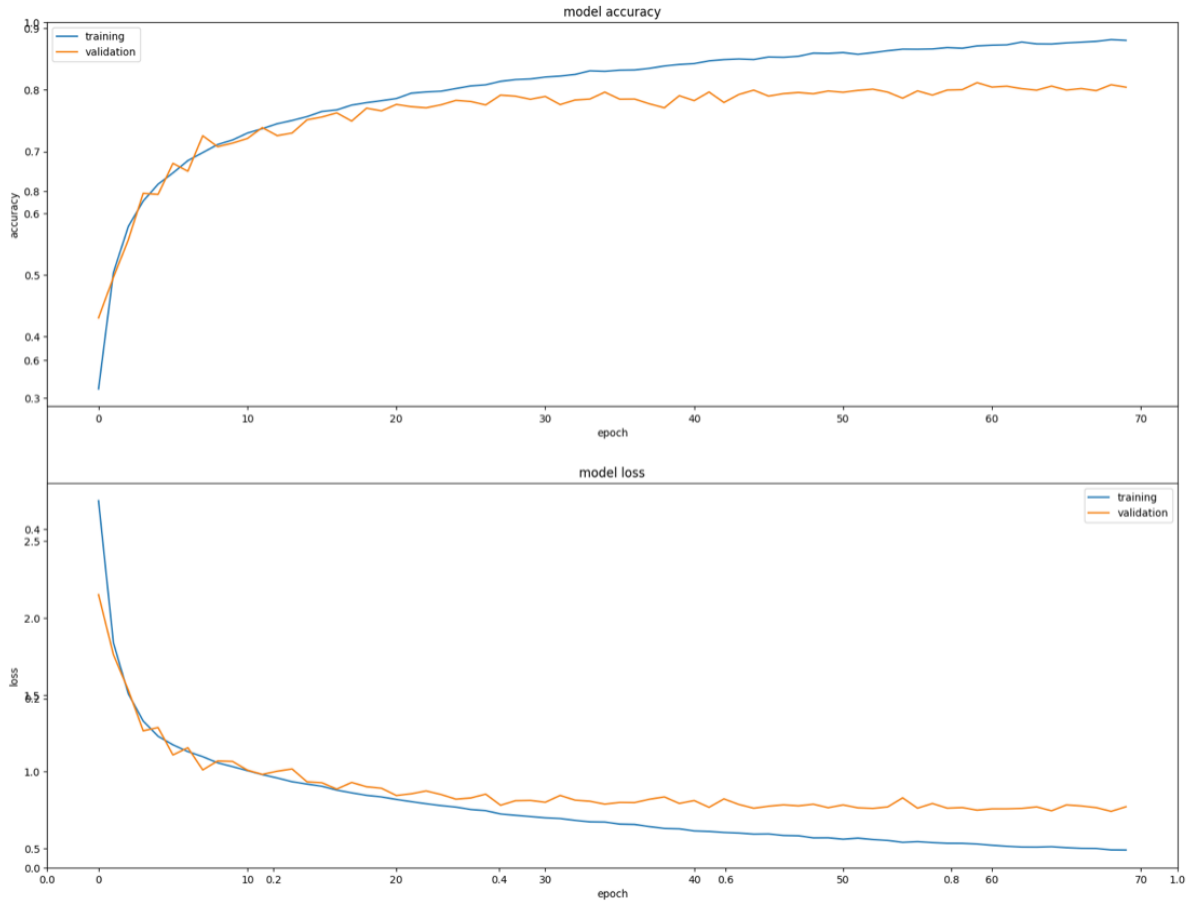
Experiment 8 – Accuracy and Loss Trends During Model Fitting



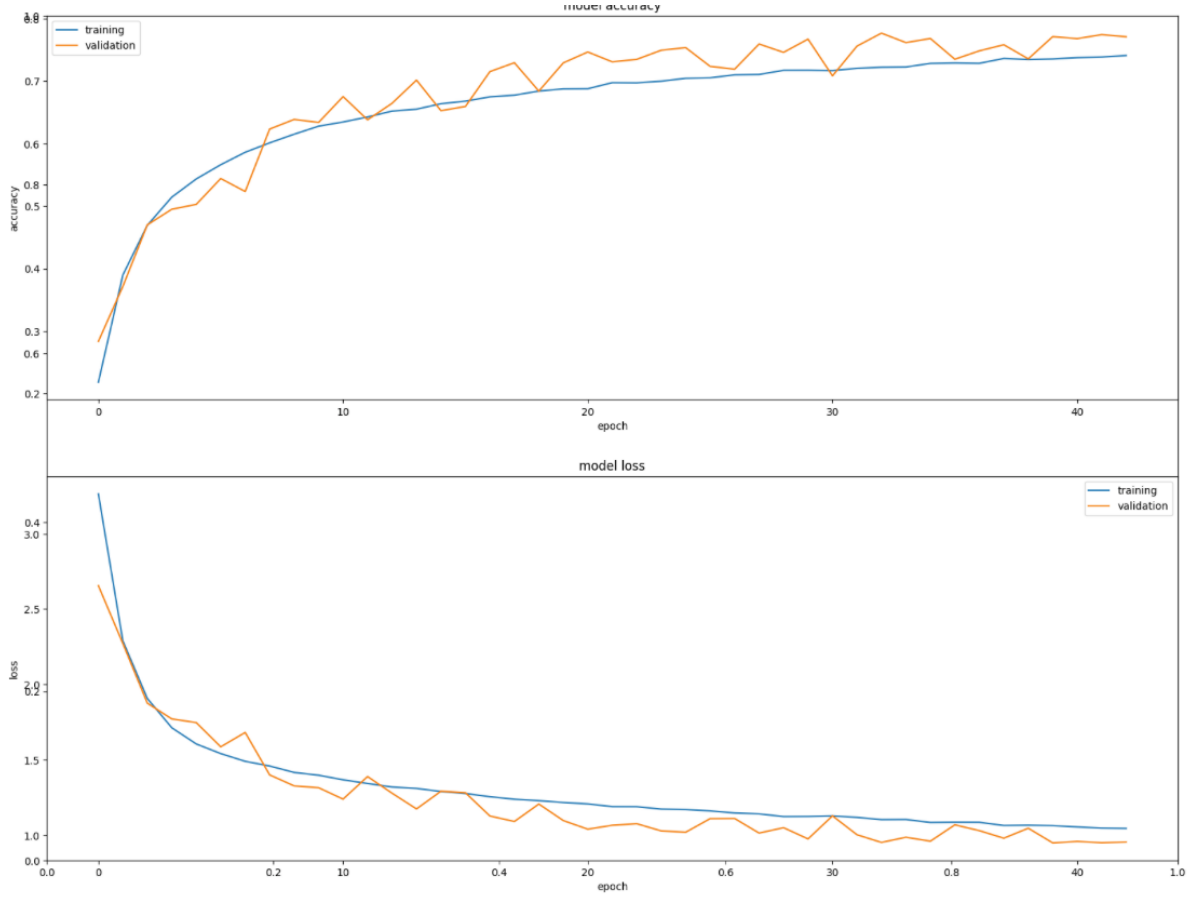
Experiment 9 – Accuracy and Loss Trends During Model Fitting



Experiment 10 – Accuracy and Loss Trends During Model Fitting



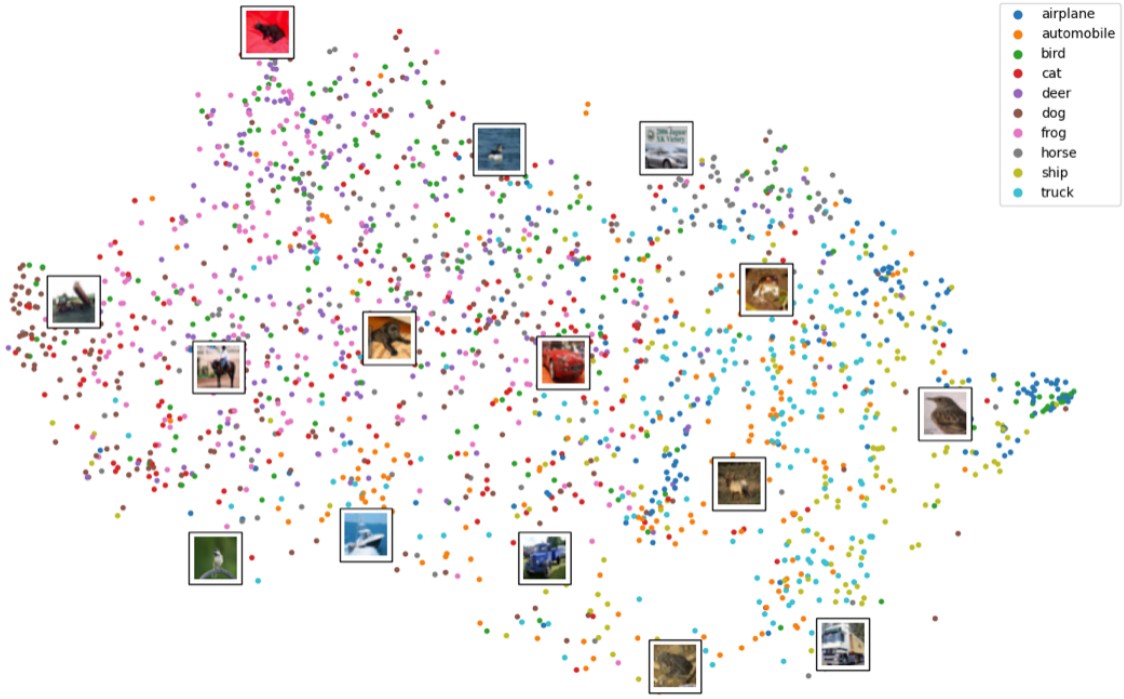
Experiment 11 – Accuracy and Loss Trends During Model Fitting



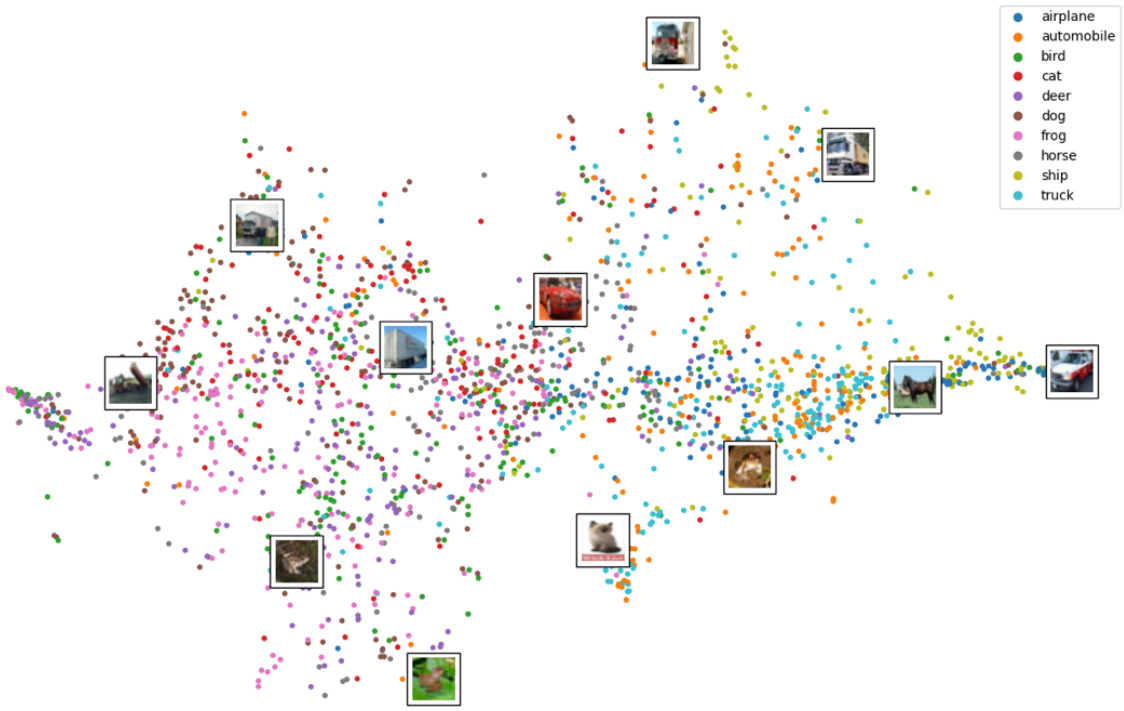
Appendix D – Scatterplot to observe the distribution of a sample of model classifications.

The graphs below display the scatterplots of model classifications.

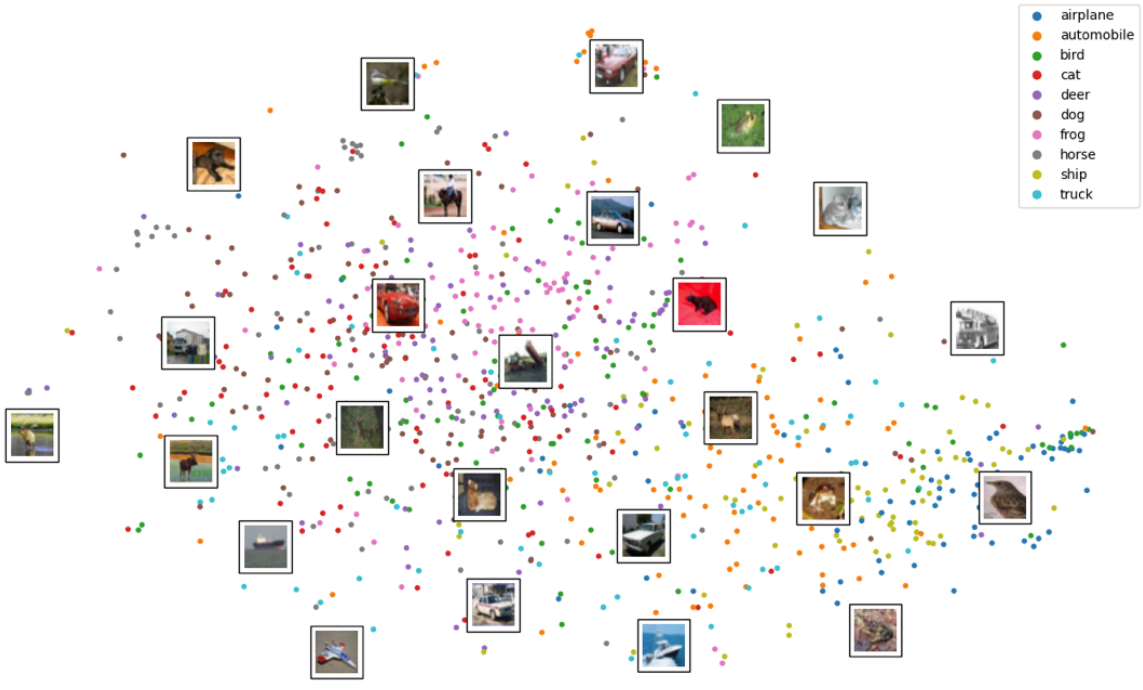
Experiment 1



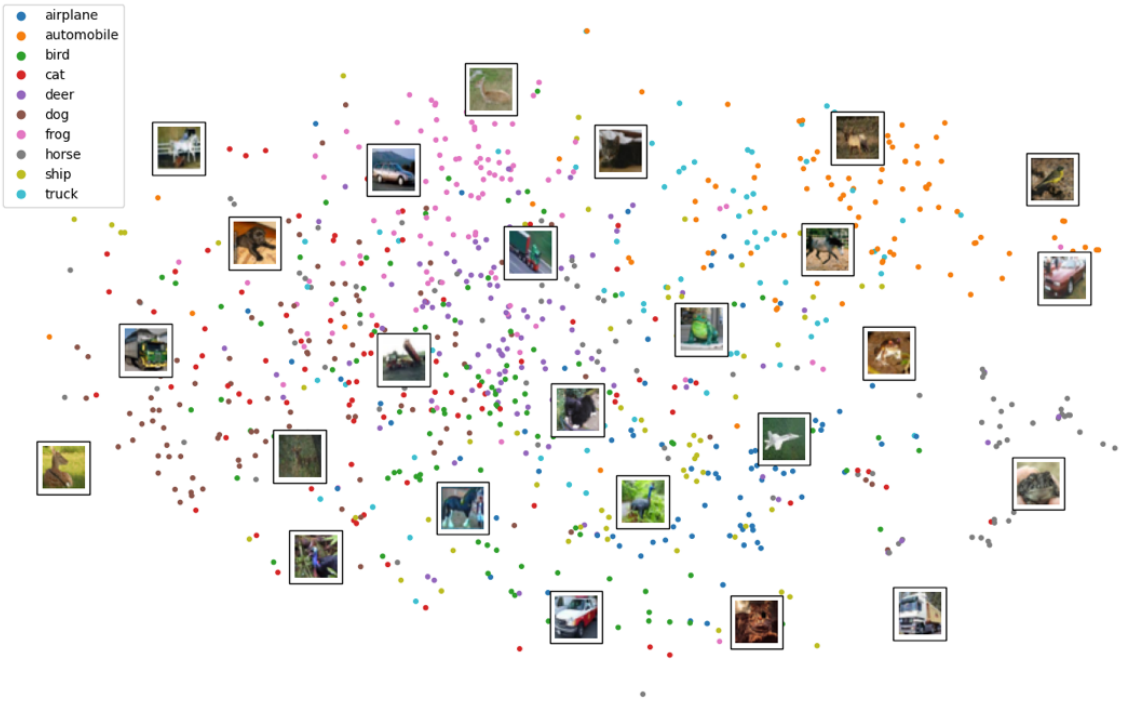
Experiment 2



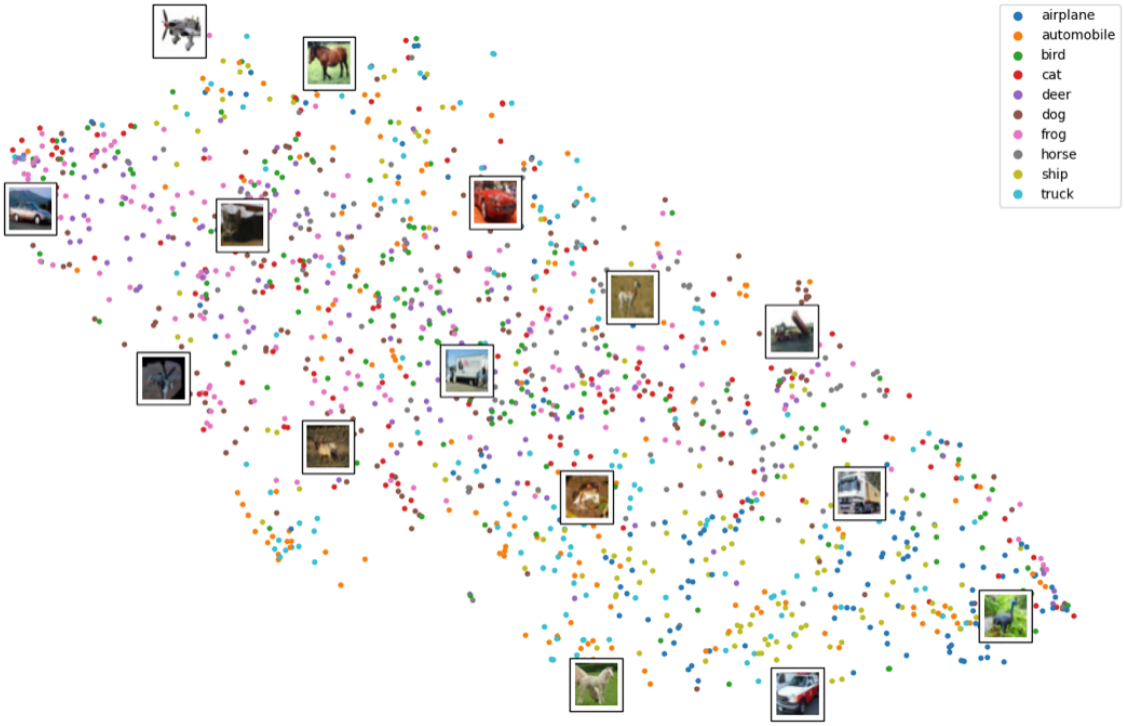
Experiment 3



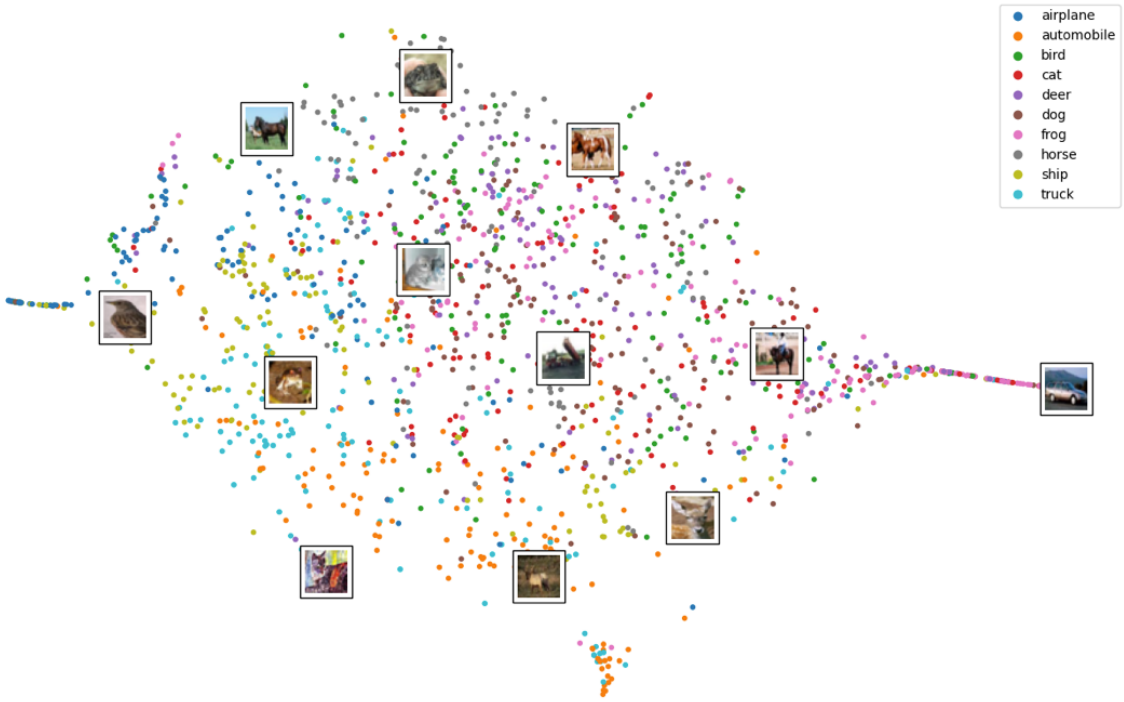
Experiment 4



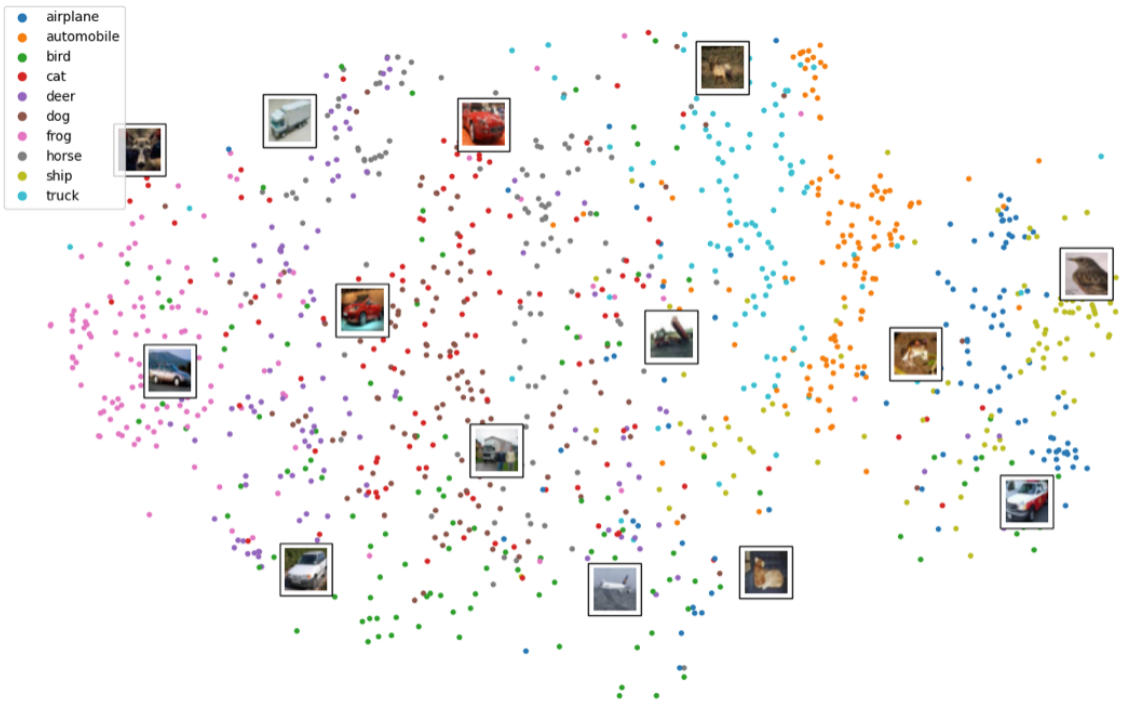
Experiment 5



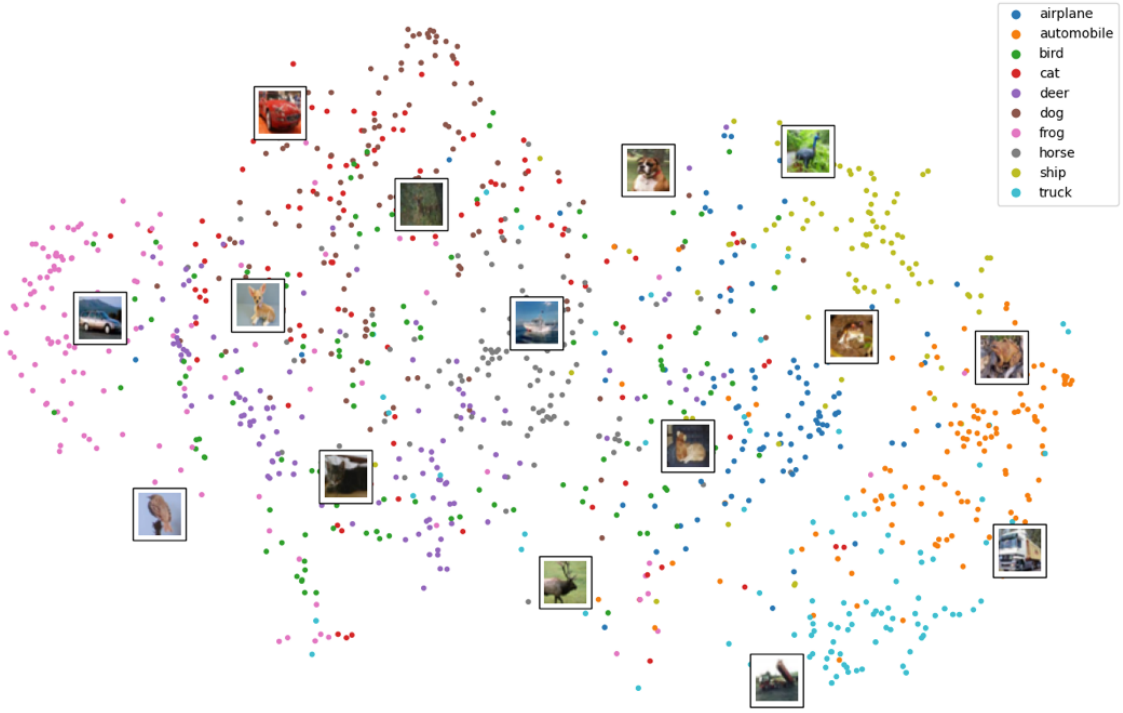
Experiment 6



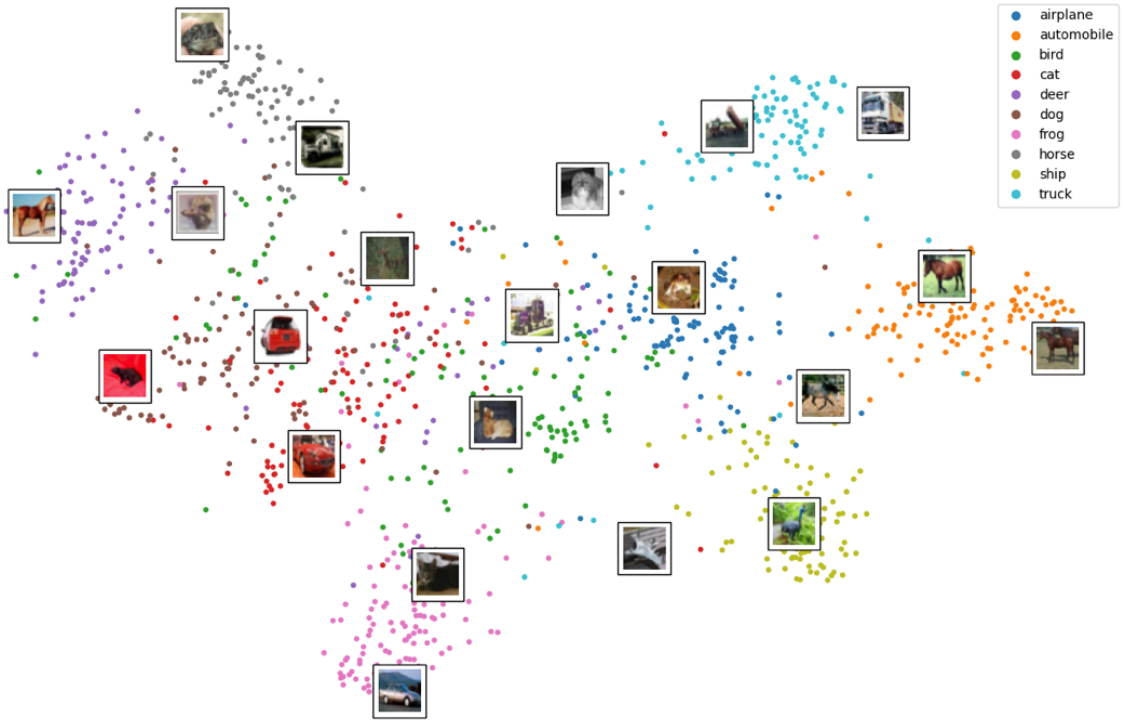
Experiment 7



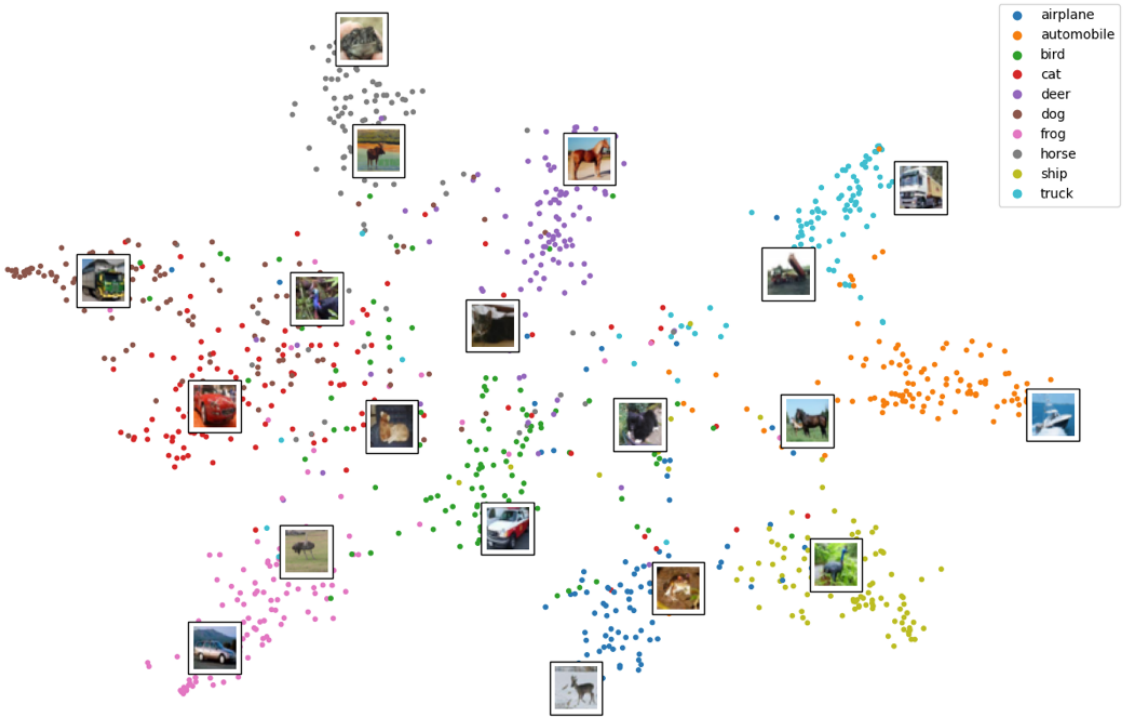
Experiment 8



Experiment 9



Experiment 10



Experiment 11