Image Classification for Caltech101 with Convolutional Neural Networks

HENRY LUO and XIAOXIAO FU, University of British Columbia

The current project attempts to explore different implementation of convolutional neural network (CNN) suitable for image classification task on the Caltech101 dataset. Two classic architectures, AlexNet and ResNet50, are examined. The models are first attempted to be trained from scratch, exploring various data augmentation and hyperparameter adjustments, to increase the performance on classification of Caltech101 images. Transfer learning is then applied to ResNet50, in the form of importing a pre-trained ResNet50 model. The performance of the models trained from scratch achieved accuracy in a ranged between 66% and 77%. The pre-trained model achieved the best performance, with the fastest and most stable validation loss convergence and the highest accuracy of 97%. It is concluded that it appears that the more complex model with transfer learning is the most effective implementation for the image classification task for Caltech101.

Additional Keywords and Phrases: Image classification, convolutional neural network, AlexNet, ResNet50

1 INTRODUCTION

The current project attempts to identify a suitable implementation of convolutional neural network (CNN) to effectively perform image classification on the Caltech101 image dataset. The project examines the performance of two classic implementations of CNN: AlexNet and ResNet50. Various attempts to improve model performance are made. The models are first trained from scratch with images in the Caltech 101 dataset. Data augmentation and adjustments of hyperparameters are implemented as parts of the effort to increase classification performance by increasing training data's diversity and countering overfit. The project first implemented a default AlexNet architecture and attempted to improve its performance by adjusting hyperparameter values and enlisting additional data augmentation. ResNet50 is then implemented to as a deeper architecture to improve classification performance. Hyperparameter values are again adjusted with additional attempts of changing optimization method. Transfer learning is then applied on ResNet50 by employing a pre-trained model and fine tune it with the Caltech 101 images to make it suitable for the specific image classification problem at hand. As the complexity of the CNN model increases, the classification performance increased, while transfer learning with the more complex model achieved the best result for the classification of images in Caltech 101.

2 BACKGROUND

Image classification is an important task in computer vision industry. It involves categorizing an image into a set of pre-defined classes or categories based on its content. With the development of image classification technology, its related applications are becoming more and more widespread. At present, the more commonly used ones are object recognition, facial recognition, medical imaging, autonomous vehicles, and quality control.

In recent years, with the development of deep learning algorithms, significant progress has been made in image classification. One of the most important breakthroughs occurred in 2012 with the appearance of AlexNet, which is a deep convolutional neural network (CNN) and has a great success in ImageNet classification challenge. AlexNet achieved an error rate of 15.3%, beating the 2nd place by 10.8% and it consists of 5 convolution layers, 3 max-pooling layers and 2 fully connected layers [1]. AlexNet's success shows the potential of deep learning for image classification and paved the way for further research and development in this area [2]. With the use of convolutional layers, it allows the network to learn spatial features from the images. Moreover, the network was trained on a large dataset of labeled images, which helped it to learn generalizable features that could be applied to new images [1].

Another advancement in image classification occurred in 2015, with the appearance of ResNet50.ResNet50 use residual learning and allows the network to learn more complex and deeper architectures [5]. The biggest advantage of residual learning is that it eases the training of networks, so that it allows for training very deep networks with hundreds of layers. Meanwhile, it also alleviates the gradient disappearance problem that often occurs in deep networks [3].

Transfer learning is another important concept in image classification, where pre-trained models are used as a starting point for a new task. In other words, with the use of transfer learning, we try to exploit what has been learned in one task to improve the generalization of another [4]. Since transfer learning allows the use of knowledge from pre-trained models trained on large dataset, so it can save significant time and resources when training a new model for a specific dataset. It’s currently very popular in deep learning.

As mentioned above, image classification is a critical task with numerous applications in computer vision, and recent advances in deep learning and transfer learning have significantly improved accuracy and performance. The current project will use AlexNet, ResNet50 and transfer learning to improve the accuracy and performance of image classification tasks.

3 METHOD

3.1 Data

The Caltech 101 dataset contains 9,144 images of 101 categories. Each category includes around 30 to 800 images. The format and size of the images are mixed. RGB and gray-scale images of various sizes are present.

3.2 Data processing

All images are resized to 244 times 244. Gray-scale images are converted to RGB images to make them compatible with the model implementation. Images are normalized according to the ImageNet mean and standard deviation to increase the efficiency of training and improve generalization.

The images are split into three separate sets: training, validation, and testing. The training set contains 70% of the images in Caltech101. The models are trained on this set of images. The validation set contains 15% of the images. In each training epoch, the model is validated on the validation set of training with the training set as a way to monitor the training process in real time. The testing set contains the remaining 15% if images. The testing images are never seen by the models during training and serves as the ultimate test for the model's performance on novel images.

Basic data augmentation is performed. Random horizontal flip is implemented. About half of the images in the training set are randomly flipped horizontally in each batch as the default probability is set 0.5. Images in the training set are also randomly cropped after adding 4 pixels of padding at the edges during training, increasing the diversity of the training set. The augmentation allows the model to see he same images in different orientations, position, and scale, so that it can generalize better for novel images capturing the same object.

3.3 Model Implementation

A classic architecture of AlexNet is implemented. The implementation consists of 5 convolutional layers with the first layer having a kernel size of 11 times 11 and the subsequent layers having a kernel size of 3 times 3. A padding of 2 is added to the first and second layers while a padding of 1 is added to the subsequent layers. ReLu (Rectified Linear Unit) is implemented as the activation function. 3 max-pooling are implemented each with a kernel size of 3 and stride of 2 to reduce dimension and improve computation efficiency. 3 fully connected layers are employed with the last layer having 101 nodes, matching the number of image categories of the Caltech 101 dataset. Two drop out layers are incorporated, each with a default dropout rate of 0.5 to reduce overfit.

A batch size of 32 is chosen for the model. Cross Entropy Loss function and stochastic gradient decent (SGD) optimization method are employed for the model. Learning rate is set at 0.001. A momentum of 0.9 is incorporated for the SGD method to speed up convergence by considering the both the previous and the current gradient when updating the parameters.

A classic architecture of ResNet50 is implemented. There are 50 layers with weights in the model. The initial convolutional layer has a kernel size of 7 times 7 followed by a max pooling layer with a 3 times 3 kernel. ReLu activation function is used, both layers has stride of 3. Four main blocks containing residual units are implemented subsequently. Each main block includes several residual units which contains various convolutional layers. Each residual unit, also referred to as bottleneck blocks, contain 3 convolutional layers with kernel size of 1, 3, and 1 subsequently. The input of each residual unit is added to the output before passing to the activation function. ReLu is used for activation function and batch normalization is applied after each layer before the activation function to increase training efficiency. The first main block has 3 residual units, the subsequent main blocks contain 3, 6, and 3 residual units respectively. A global average pooling layer is applied after the main blocks. The final fully connected layer with 101 nodes is then applied.

A batch size of 32 is again chosen for the model. Cross Entropy Loss and SGD optimization method are also employed again for consistency. Learning rate is set at 0.001 and momentum at 0.9.

A pre-trained model of ResNet50 is imported from `torchvision`. The final fully connected layer is adjusted to have 101 nodes to match the image categories of the Caltech101 dataset. The pre-trained model is already trained on the ImageNet images thus the weights are already tuned. The Caltech101 images are passed to the model to fine tune it for the specific dataset.

3.4 Procedure

A total of seven trials are performed. Three trials are for AlexNet, three for ResNet50, and the final trial for pre-trained ResNet50. The first trial for AlexNet is training the default implementation from scratch and the results are used as a performance benchmark for subsequent trails. The second and third AlexNet trials adjusted weight decay and incorporated additional data augmentation as attempts to increase the performance of AlexNet. In each trial of AlexNet, the model is trained for 100 epochs.

The first trial of ResNet50 is training the default implementation and the results are again used as a benchmark for subsequent trials and to compare with AlexNet. The second trail adjusted the optimization method to Adam, an adaptive learning rate method that dynamically adjust the learning rate during the training process. The initial learning rate is 0.001. The third trial reduced the learning rate to 0.001. The first ResNet50 trial trained the model for 100 epochs. The second and third trials trained the model for 60 epochs. The final pre-trained ResNet50 trial inherited the default hyperparameters of the from the firs ResNet50 trial while. The model is trained for 25 epochs.

4 EXPERIMENTS AND RESULTS

4.1 **AlexNet Trial 1**

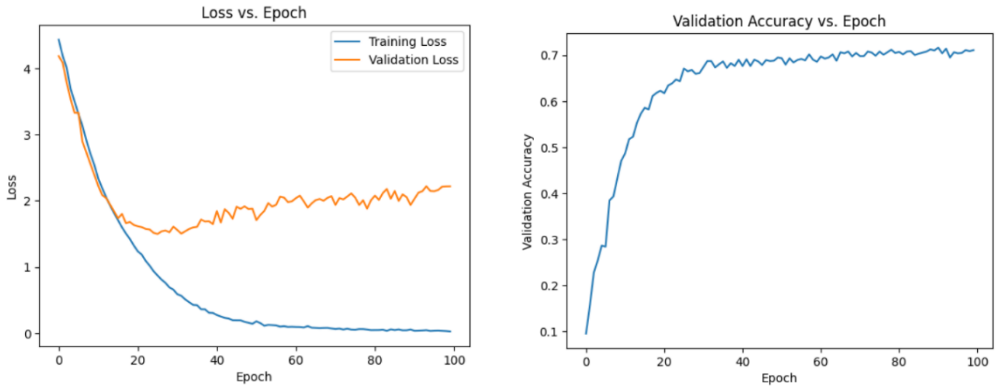

Figure 1: AlexNet trial 1 loss and validation accuracy

Figure 1 demonstrates the training loss and validation loss in each epoch of training in the plot on the left, and validation accuracy of the mode after training on the right. From the loss plot on the left, it is evident that there is an increasing trend for validation loss while the training loss is decreasing after around 40 epochs. It could be a sign of overfitting. The training loss is the in-sample training loss which measures how well the model classifies the images of the training set. Decreasing training loss means that the model is getting better at classifying the training images. However, as training loss decreases, the model could be picking up noise in the training set, hindering its ability to generalize to novel images. The validation set could monitor for potential overfit. If validation loss shows an increasing trend while the training loss is still decreasing, it indicates that the model is not generalizing well for novel images, which is a sign of overfit. The default AlexNet achieved a accuracy of around 67.58%, F1 score: 0.66, precision:0.69, and recall: 0.67, with signs of overfit.

4.2 AlexNet Trial 2

The address the potential overfit in trial 1, a weight decay of 0.0001 is introduced, adding a regularization term to the loss function to prime the model towards learning smaller weights. It helps reduce overfitting, improving generalization of the model.

Figure 2 shows the training and validation loss in each epoch on the left and the test accuracy on the right. Contrary to the expectation, the increasing trend of the validation loss during training is still present. The model achieved an accuracy of around Accuracy: 68.51%, F1 score: 0.67, precision: 0.69, and recall: 0.69. The performance of the model is not improved significantly and potential overfit is still present.

4.3 AlexNet Trial 3

To tackle overfit, further data augmentation is introduced to increase the diversity of the training data. Random rotation between -10 to 10 degrees is carried out for each training image. The brightness, contrast, saturation, and hue of each image are also randomly altered up to 10% of the original. The model now sees the images in different orientations and colours. Increasing the diversity of the dataset in addition of weight decay could improve the generalizability of the model.

Figure 3 shows the results of the trial. Although the validation loss for the early epochs improved and the that for the later epochs became more stable, the general increasing trend is still present. The model achieved an accuracy of 67.81%, F1 score: 0.67, precision: 0.72, recall: 0.68. The performance again did not increase significantly.

4.4 RestNet50 Trail 1

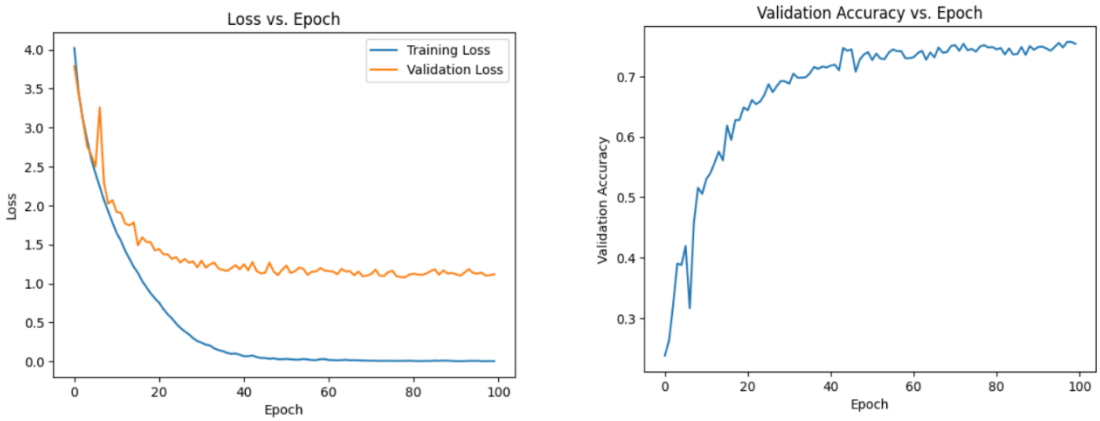

Figure 4: ResNet50 trial 1 loss and validation accuracy

It seems that a more complex architecture is needed to improve classification performance. Figure 4 outlines the results of the default implementation of ResNet50 after 100 epochs. Validation loss does not exhibit an increasing trend, which is a good sign, indicating that the model is not overfitting. The validation loss and training loss appears to have converged after around 50 to 60 epochs. The model achieved an accuracy of Accuracy: 74.52%, F1 score: 0.74, precision: 0.76, recall: 0.76. The performance has been improved.

4.5 ResNet50 Trail 2

There are significant oscillations during the early epochs for the validation loss during trial 1, suggesting that the initial learning rate may be larger than needed, causing the optimization algorithm to bounce back and forth. To handle this problem, an adaptive learning rate method, Adam, is employed to dynamically adjust the learning rate during training. Figure 5 shows the results. Since the ResNet50 model apparently converged around 50 epochs in trial 1, the subsequent trials adopted 60 epochs due to time constraints and lack of GPU resources. The initial large oscillations have disappeared, and the model converged at roughly the same level of validation and training loss. The performance of the model is as follows: Accuracy: 75.83%, F1 score: 0.76, precision: 0.79, recall: 0.76.

4.6 ResNet50 Trial 3

To explore if a smaller initial learning rate could improve model performance, the initial learning rate for Adam is reduced to 0.0001. A smaller initial learning rate may allow the adaptive method to make finer adjustments and pick up more complex structures in the images. The results are outlined in figure 6. Although the validation loss converged faster than in the previous trials, the classification accuracy remains largely the same as before. Accuracy: 77.60%, F1 score: 0.77, precision: 0.80, recall: 0.78

4.7 Pre-trained ResNet50

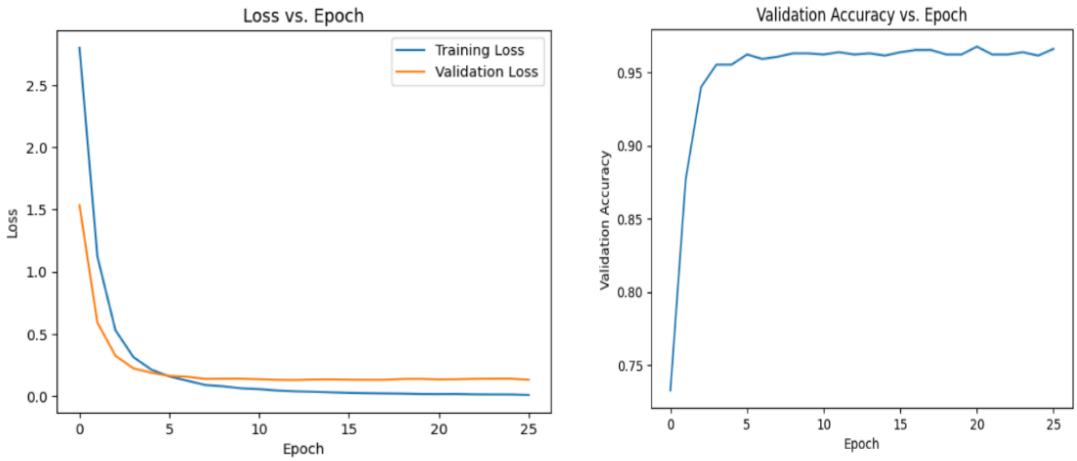

Figure 7: pre-trained ResNet50 loss and validation accuracy

Transfer learning is applied with ResNet50. A pre-trained model is imported from the torchvision model trained on the Caltech101 dataset. The results are demonstrated in figure 7. Significant improvement of loss convergence and classification accuracy is observed. Validation loss converged much faster than the previous models. Since the weights of the pre-trained model is pre-trained on ImageNet dataset, the model only needs to be fine tuned to adapted to Caltech101. The process only took less than 5 epochs for the validation loss to converge and there is no sign of overfit. The model achieved the best performance thus far: Accuracy: 96.70%, F1 score: 0.97, precision: 0.97, recall: 0.967.

DISCUSSION

With limited prior exposure to machine learning projects, difficulties arose frequently throughout the project from loading the data set to implementing the models and recording the results. Constant online searching and relentless learning from trials and errors enabled this project's completion. One notable challenge involved processing of the images in Caltech101 before inputting to the models. The dataset contains a mixture of gray-scale and colour images entailing that the number of channels of the original images are different, which resulted in numerus errors when attempting to implement the models. Handling this problem allowed more thorough understanding of the expected input and output of each CNN layer and how the convolution process works. The models explored are computationally intensive, time constraints and the lack of computing resources such as access to GPUs was another major challenge.

In conclusion, the current project attempted to explore suitable implementations of convolutional neural networks for classifying images in the Caltech101 dataset. Two classic architectures, AlexNet and RestNet50 are trained from scratch with various adjustments data augmentation and model hyperparameters to improve model performance. The

Performance of models trained from scratch ranged between 67% to 77% accuracy. Transfer learning is employed for ResNet50 and achieved the best result of 97% accuracy. Hyperparameter tuning aiming did not yield significant improvement for the models for this project. With more time and resources, it is possible to explore more combinations of hyperparameter values that could improve model performance. It appears that increasing model complexity with more advanced architecture and applling transfer learning was the most effective strategy to improve model performance in the scope of this project.

REFERENCES

[1] Chen Yanhui. 2021. *From AlexNet to NASNet: A Brief History and Introduction of Convolutional Neural Networks*. Retrieved 30 April, 2023 from https://towardsdatascience.com/from-alexnet-to-nasnet-a-brief-history-and-introduction-of-convolutional-neural-networks-cf63bf3320e1

[2] Md Zahangir Alom1 , Tarek M. Taha1 , Chris Yakopcic1 , Stefan Westberg1 , Paheding Sidike2 , Mst Shamima Nasrin1 , Brian C Van Essen3 , Abdul A S. Awwal3 , and Vijayan K. Asari. 2018. *The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches*. https://doi.org/10.48550/arxiv.1803.01164

[3] Guadenz Boesch. 2023. *Deep Residual Networks (ResNet, ResNet50) – 2023 Guide*. viso.ai. Retrieved 30 April, 2023 from https://viso.ai/deep-learning/resnet-residual-neural-network/. [Accessed: Apr. 30, 2023].

[4] Jason Brownlee. 2017. *A Gentle Introduction to Transfer Learning for Deep Learning*. Machine Learning Mastery. Retrieved 30 April, 2023 from https://machinelearningmastery.com/transfer-learning-for-deep-learning/.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun 2016. *Deep residual learning for image recognition*. IEEE Computer Society. DOI: 10.1109/CVPR.2016.90.

APPENDIX

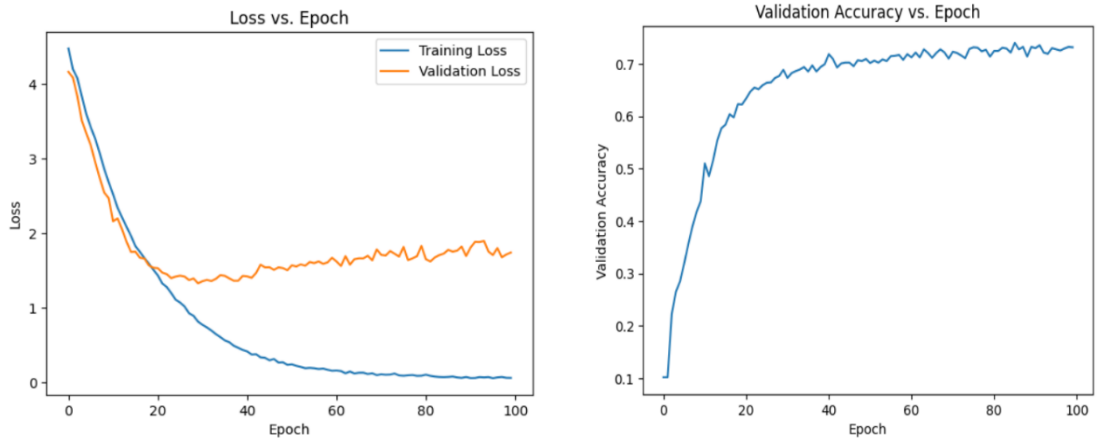

Figure 2: AlexNet trial 2 loss and validation accuracy

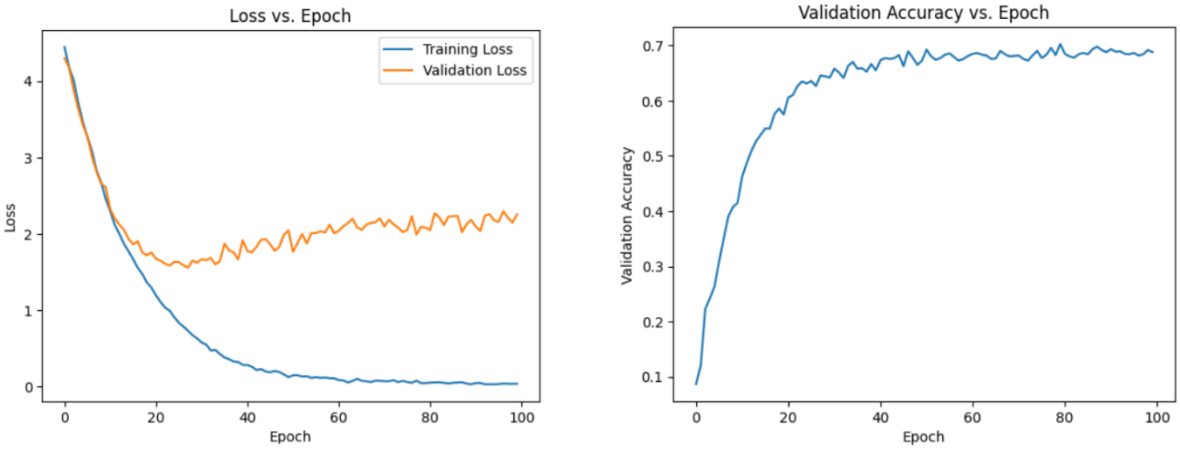

Figure 3: AlexNet trial 3 loss and validation accuracy

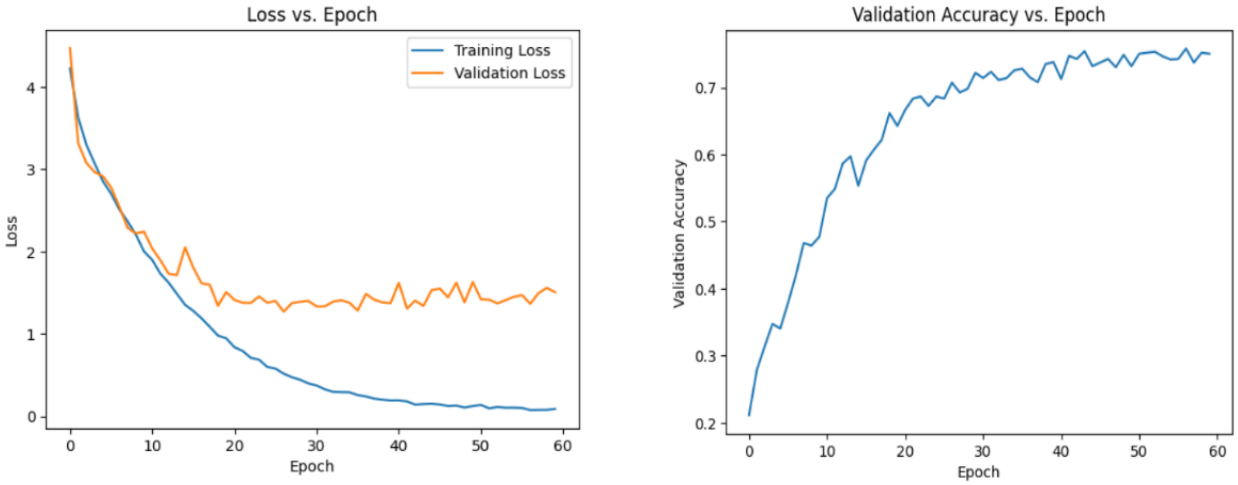

Figure 5: ResNet50 trial 2 loss and validation accuracy

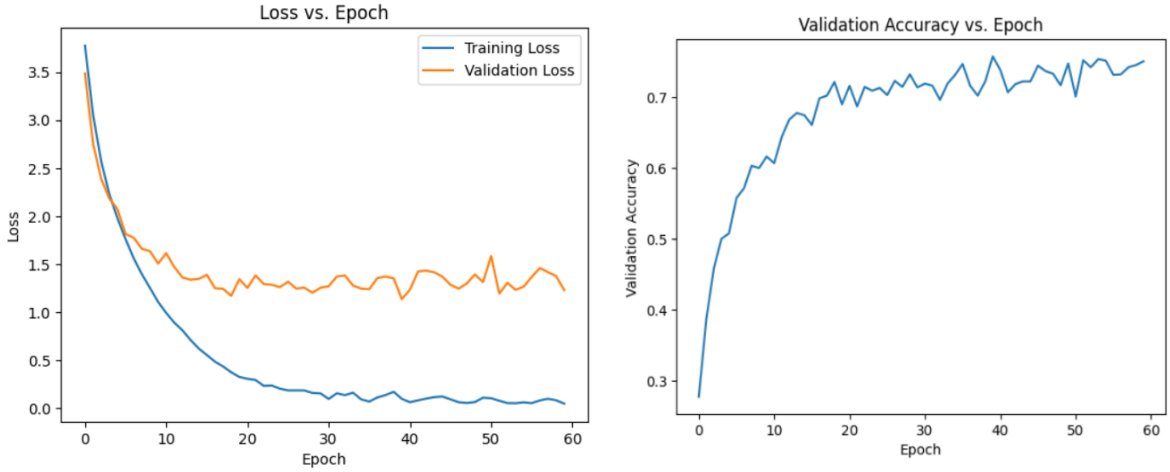

Figure 6: ResNet50 trial 3 loss and validation accuracy