FlexNet and Other CNN Architectures

Jolene Mollenkamp

jrm180012

[jrm180012@utdallas.edu](mailto:jrm180012@utdallas.edu)

Erik Jonsson School of Engineering and Computer Science

Shah Masood

Sam210028

[sam210028@utdallas.edu](mailto:sam210028@utdallas.edu)

Erik Jonsson School of Engineering and Computer Science

*Abstract*— This report presents a comprehensive study on the application of convolutional neural networks (CNNs) for the classification of image data across multiple benchmark datasets: MNIST, Fashion-MNIST, CIFAR-10. We explore the effectiveness of several classic and contemporary CNN Architectures, including Multilayer Perceptron(MLP), LeNet, and a newly proposed model which we call FlexNet, designed for this study. Each model was implemented using PyTorch, and the experiments were conducted using standardized training and validation protocols. The models were evaluated based on their training and validation loss metrics, recorded over multiple epochs, to assess both their learning efficiency and generalization ability. The findings from this study highlight the varying degrees of adaptability of CNN architectures to different image recognition tasks, providing valuable insights into architectural choices and parameter tuning. These results aim to guide the selection of appropriate CNN models for specific image classification tasks in both academic research and practical applications.

1. INTRODUCTION

Convolutional Neural Networks (CNNs) can be helpful in recognizing patterns in images allowing for classification. A CNN is a type of machine learning network that helps to analyze and recognize patterns in visual data using filters, gradients, and backpropagation. A typical CNN is made up of an input layer, an output layer, and any number of hidden layers in between whether they be convolutional layers, max pooling layers, or fully connected layers. The visual data is the input to the model and the output is how the model classifies the image. The structure of a CNN is inspired by biological processes in the brain; the neurons in a CNN are connected in the way the brain’s neurons are. CNNs can be useful for image and video recognition, facial recognition, classification, medical image analysis, natural language processing, and brain-computer interfaces like an EEG or an MRI. There have been many popular architectures created with varying parameters, like LeNet, AlexNet, VGGNet, ResNet, and more that serve as a model for training and testing data.

CNNs are typically made up of convolutional layers, max pooling layers, and fully connected layers. A convolutional layer transforms the width and height of their input by computing the dot product of them and the filter parameters during the forward pass. A max pooling layer will divide the input image into subsections and output the maximum of each subsection. This type of layer is typically used in between successive convolutional layers. A fully connected layer or layers are added to the end of the architecture after the convolutional and max pooling layers to complete the final classification. The neurons in a fully connected layer are connected to every possible source in the previous layer, thus making it prone to overfitting. Overfitting is where the model corresponds too closely to the data and has too many parameters than is justified. There are many methods to prevent or reduce overfitting, like dropout (some nodes are dropped out of the model), stochastic pooling, providing more training data, stopping the training before overfitting occurs, or limiting the number of parameters. After the fully connected layer is the loss layer, which in our case is the LogSoftmax or Softmax loss function, suitable for mutually exclusive multi-classification.

Using existing architectures as a basis for our design, the parameters for the proposed network were determined. This project will utilize three datasets: MNIST, CIFAR10, and Fashion-MNIST as the data used for classification. Using different datasets allows the architecture to perform on different inputs, allowing for a more accurate conclusion on its performance. This project will also use three different architectures to compare the results of two known architectures against a new architecture. The multi-layer perceptron (MLP) and LeNet are going to be compared against a new architecture with unique parameters, layering, and functions. The MLP and the LeNet networks are a control used to compare the results of our model against. This is important so that we can make conclusions about the performance of a novel network through comparison to known networks. The motivation for this project is to create a unique CNN architecture by adjusting parameters and configuration of layers, to receive favorable evaluation metrics, such as a high accuracy score and F1-score.

1. RELATED WORK

The existing literature has many different neural networks with various compositions that relay different results. Some focus on depth, while others focus on preventing overfitting among many other motivations. In recent years research into machine learning is booming and therefore the industry is changing fast as papers are published and experiments are completed. With this in mind, we acknowledge that our model may not be impressive in comparison to a large amount of incredible research, but it is crafted with genuine thought and intent. Our goal was to synthesize the benefits of notable compositions into a new architecture that is highly accurate and performs well with low error rates.

Lecun, Bottou, Bengio, and Haffner created a pivotal neural network for classification [1]. The neural network itself is not overly complicated with only five layers, allowing for easy comprehension and universal usage. The architecture uses a gradient-based learning algorithm for training and results in a model which can classify written characters. At the time of the paper’s publication they were working to eliminate hand-written feature extractors in favor of their neural network, requiring less human input and increasing levels of automation. This is a classic network that has inspired the creation of other architectures, and was used by financial institutions to recognize the numbers written on checks.

Krizhevsky, Sutskever, and Hinton created the AlexNet convolutional neural network [2]. They wanted to improve the performance of previous models and prevent overfitting. They used larger datasets in order to surpass human performance levels; they used the ImageNet LSVRC-2010 which contains 1.2 million images that they classified into 1000 different classes. They received low error rates on their data. Another novel feature of their neural network is that they use ReLU activation function instead of tanh because ReLU results in faster training times. To prevent overfitting they used a method called “dropout” in the fully connected layers. AlexNet showed the benefits of large datasets and of the use of the ReLU activation function on a CNN with five convolutional layers and three fully connected layers.

In contrast to AlexNet and LeNet, VGG Net tested the effect of adding layers on the accuracy of a neural network [3]. While LeNet used a simple five layers, Simonyan and Zisserman tested networks from as small as 3 x 3 convolution filters to 19 layers, and they were able to conclude that more layers were beneficial to the level of classification accuracy. Their results also generalized to other datasets as well.

He, Zhang, Ren, and Sun created ResNet which utilizes an even deeper network than VGG Net does; ResNet consists of up to 152 layers [4]. Along with emphasizing depth, ResNet uses a residual function to reformulate the layers resulting in optimized training. As the depth of the network increases the accuracy degrades, thus the residual learning function curbs this degradation, allowing for optimal accuracy. This CNN improved upon the problem of degradation in deep neural networks by reformulating the layers to make networks this deep possible.

As the complexity and depth of the networks increase, so does the cost of storage. These architectures can contain many layers as well as parameter storage coming at a high cost, so different methods of compression have been proposed in order to filter out what is unnecessary and save space. Basha, Farazuddin, Pulabaigari, Dubey, and Mukherjee proposed to use a history based pruning method that iteratively prunes the redundant filters of the CNN [5]. They implemented the pruning on LeNet-5, VGG-16, ResNet-56, ResNet-110, and ResNet-50 and received an impressive reduction in floating point operations while maintaining a low error rate.

Another study compared the different activation functions to see what the benefits and disadvantages would be using MNIST and Fashion-MNIST datasets on LeNet-5 and VGG-16 [6]. They concluded that the linear activation function obtains the worst results, the sigmoid function results in the slowest convergence, the ELU activation function resulted in the best accuracy but it was not far off from ReLU, leaky ReLU, and PReLU. They repeated the test using different datasets: CIFAR-10 and CIFAR-100 which are noticeably more complex than MNIST and Fashion-MNIST. The best activation functions found for these datasets were leaky ReLU and ELU which resulted in better accuracy and training speed. The final conclusions generated from this research was that multiclassification problems should utilize softmax on its final layer, ReLU and leaky ReLU are suited for hidden layers, and that leaky ReLU’s negative slope can be set to 0.02 for faster training. When choosing a loss function for a CNN model with a softmax layer at the end, it is best to choose cross entropy loss. Ultimately the selection of these important functions depends on the proposed architecture and the data distribution, therefore it is hard to make generalized statements as to which function to choose for which scenario. Deciding which to use will take much trial and error.

1. PROPOSED APPROACH

Building off of the related works and previously built and tested architectures, our architecture consists of the comparison of three different networks, two that have already been created, Multi-layer Perceptron (MLP) and LeNet, with the third being a unique network called FlexNet. This attempts to combine the beneficial aspects of previous architectures, synthesizing them into a new architecture that obtains optimized results. AlexNet discovered the benefit of using the ReLU activation function instead of the tanh activation function, thus the architecture will utilize ReLU activation functions in between the layers. Although the neural network will not be as deep as the ResNet and VGGNet architectures, it will have more layers than LeNet and the MLP. The architecture cannot be too big because of the limitations of the equipment available, which is why ResNet or VGGNet cannot be fully implemented even if it comes with more optimal results.

*A. MLP Model*

The MLP model is a simple network of three linear layers: the first hidden layer, the second hidden layer, and the output layer. TThe first two layers are followed by the ReLU activation function, while the last layer is followed by the Softmax function with ten different classes. The Fashion-MNIST and MNIST datasets both have 28x28 as their input size, while the CIFAR-10 has 32x32 as its input size. The CIFAR-10 and Fashion-MNIST datasets use larger values for their two hidden sizes: 512 and 256, while MNIST uses 128 and 64 as its hidden sizes. All three datasets use the cross entropy loss function and the Adam function as their optimizer with a learning rate of 0.001.

The advantages to the MLP model is that it is easy to understand, and does not contain complex layers. This model can be easily understood so that those who are not within the field of machine learning can understand what a model is and conceptualize what it does. The disadvantages to this model is that it is too simple and will not perform as well as a model including convolutional layers and pooling layers. This model does nothing to prevent overfitting in the fully connected layers as well.

*B. LeNet Model*

The LeNet model is larger than the MLP model because it includes convolutional layers in addition to fully connected layers. For the Fashion-MNIST dataset, the two 5x5 convolutional layers are followed by ReLU activation functions and 2x2 max pooling layers with strides of two. The first convolution has a padding size of two and the second convolution has no padding. Then the fully connected layers transform the features from 400 to 120 in the first layer, 120 to 84 in the second layer, and finally down to 10 for the final layer. The MNIST dataset is very similar to Fashion-MNIST LeNet model, except for a few distinct changes: the padding in the first convolutional layer is zero instead of two, average pooling is used instead of max pooling, and their are 256 features coming into the fully connected layers instead of 400. When using the CIFAR-10 dataset the model only changes the input channel from one to three and the feature vector going into the fully connected layers contains 400 features like the Fashion-MNIST dataset model. All three datasets use the cross entropy loss function and the Adam optimizer function with a learning rate of 0.001.

The advantages of the LeNet neural network is that it has increased the number of layers and complexity in comparison to the MLP model thus resulting in lower error rates. It includes convolutional layers and max pooling layers in order to better process the dataset. This network can be improved by adding dropout to reduce overfitting and batch normalization after the max pooling layers. The disadvantage of this model is its depth, as deeper models often perform better.

*C. FlexNet Model*

The FlexNet model begins with two 3x3 convolutions with a padding of 1 each followed by a ReLU activation function. Then there is a 2x2 max pooling layer with stride of 2 followed by a batch normalization with 64 features. Next is a 3x3 convolution with a padding of 2 and a dilation of 2 followed by a ReLU activation function. Then there is another 3x3 convolution with a padding of 1 followed by a ReLU activation function. Following this is a 2x2 max pooling layer with a stride of 2 and a batch normalization with 128 features. Then the network is flattened to 6,272 features for the Fashion-MNIST and MNIST datasets, and 8,192 for the CIFAR-10 dataset. Then this number is linearly transformed to 512 features followed by a ReLU activation function and a dropout layer. This dropout layer will randomly zero out some of the elements with a 0.5 probability, this is done to prevent overfitting in the fully connected layers. The network is downscaled again with a linear transformation from 512 to 256 features followed by ReLU and dropout layers. A final linear transformation is done to change the 256 features into 10 features which is then fed into the logarithm of the Softmax function for multi-classification. Instead of using Softmax, the LogSoftmax function was chosen because it results in better performance and gradient optimization. The changes made to our architecture are predicted to result in favorable results. The loss function used for all three datasets is the negative log likelihood loss (NLLLoss) because this function works with the results of the LogSoftmax function, while a function like cross entropy loss expects raw data instead of logs. The optimizer used for all three datasets is the Adam function with a learning rate of 0.001.

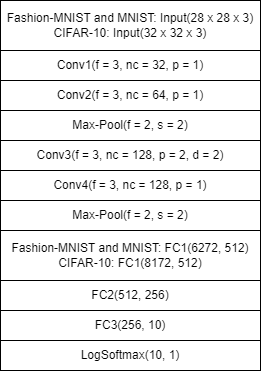


Fig. 1. FlexNet architecture overview

The advantages of the FlexNet model are the increased number of layers, the batch normalization included after each max pooling, the dropout function utilized in the fully connected layers, the LogSoftmax function being used instead of the Softmax function, and the NLLLoss function being used as the loss function. This model has been optimized to fit the three datasets being tested. The FlexNet model could have disadvantages with larger datasets or the model could perform better if more layers are used. Unfortunately we are limited in the depth of our model as running a model as deep as ResNet or VGG Net is not feasible with our machines.

1. METHODOLOGY

The datasets utilized will be the MNIST, CIFAR-10, and Fashion-MNIST datasets. Using three different datasets will allow us to gain a precise picture of how the architecture works with varied inputs. If only one dataset is used, then it is hard to tell if the network would obtain similar results with different data or if the model only works well for some inputs.

*A. MNIST*

The MNIST dataset contains black and white images of handwritten digits, its training set is 60,000 images and the testing set is 10,000 images. Each image is 28x28 pixels and contains a number ranging from zero to nine, meaning there would be ten classes for classification.

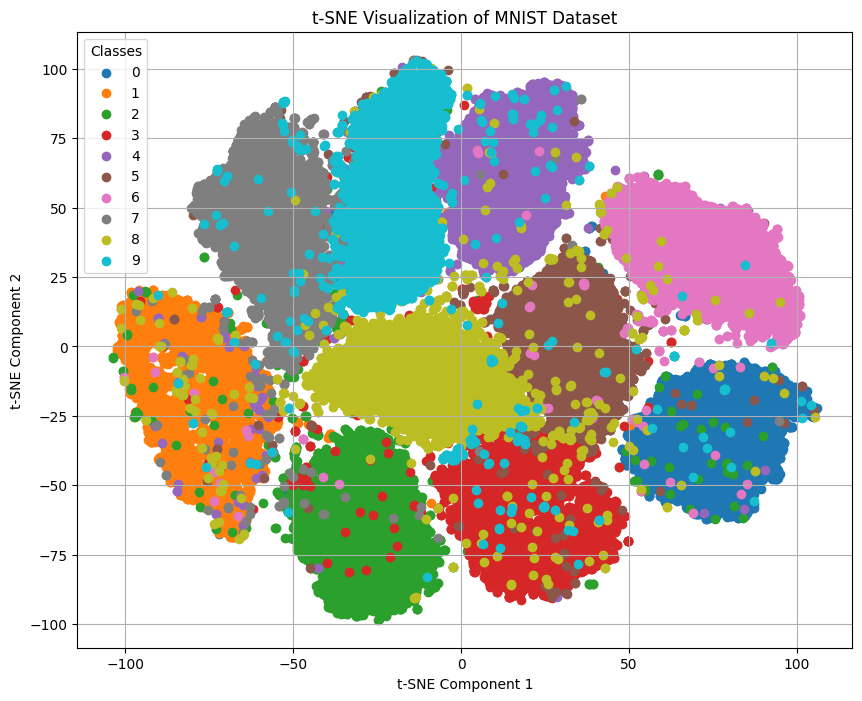


Fig. 2. t-SNE on the MNIST dataset

*B. CIFAR-10*

The CIFAR-10 dataset is made up of a total of 60,000 32x32 color images. These images are divided into 50,000 training images and 10,000 testing images. The images fall into ten mutually exclusive classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

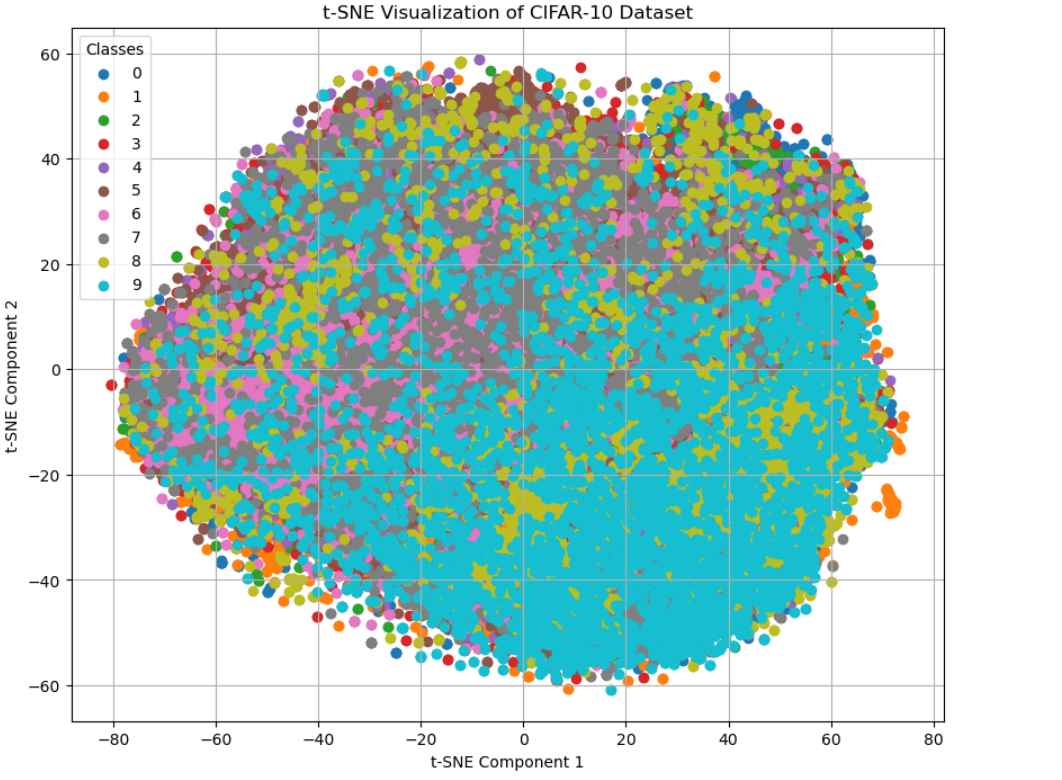


Fig. 3. t-SNE on CIFAR-10 dataset

*C. Fashion-MNIST*

Lastly, the Fashion-MNIST dataset consists of a total of 70,000 28x28 grayscale images of a fashion item, either ankle boots, a bag, a coat, a dress, a pullover, sandals, a shirt, sneakers, a t-shirt/top, or trousers. The training set houses 60,000 images and the testing set houses 10,000 images.

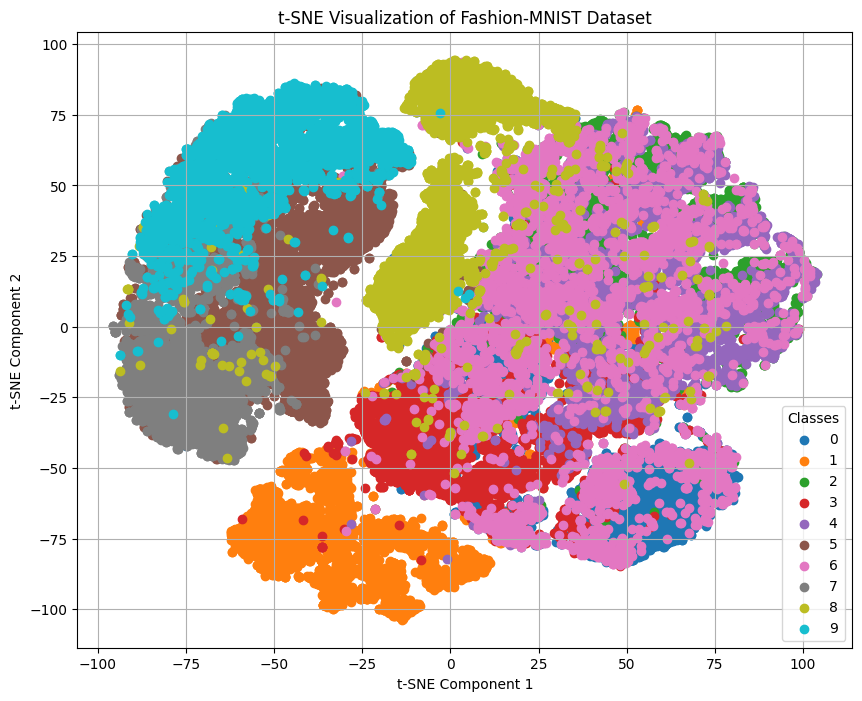


Fig. 4. t-SNE on Fashion-MNIST dataset

*D. The Architecture*

The architecture created consists of four convolutional layers, two max pooling layers, and three fully connected layers in order to reap the benefits of a semi-deep neural network, being careful not to overwhelm a computer with too many layers. Max pooling layers are used after every two convolutional layers to extract features that are important and reduce those that are not. Max pooling will ensure that unnecessary dimensions are not in the model. As fully connected layers are prone to overfitting, the dropout method is used to get rid of some features to curb overfitting. This architecture is more complex than the LeNet model and MLP model, but not as complex as AlexNet, ResNet, or VGG Net. The dropout method and batch normalization methods have been added to obtain better performance.

1. RESULTS

*A. Multilayer perceptron (MLP)*

The MLP model showed varying degrees of success across the datasets. On the MNIST dataset, it achieved an impressive accuracy of 97.8% with the training loss converging to 0.065, demonstrating strong performance on simpler, grayscale images. In contrast, the model’s performance on Fashion-MNIST dropped to an accuracy of 88.5%, reflecting challenges with more complex image patterns. The CIFAR-10 dataset posed significant difficulties for the MLP, where it managed only 54.2% accuracy underscoring its limitations with color and high-dimensional images. As the images increased in complexity and detail, the training accuracy decreased, showing that the MLP model is ill fitted for some datasets. Following is the graph showing each dataset correspondence with the MLP architecture.

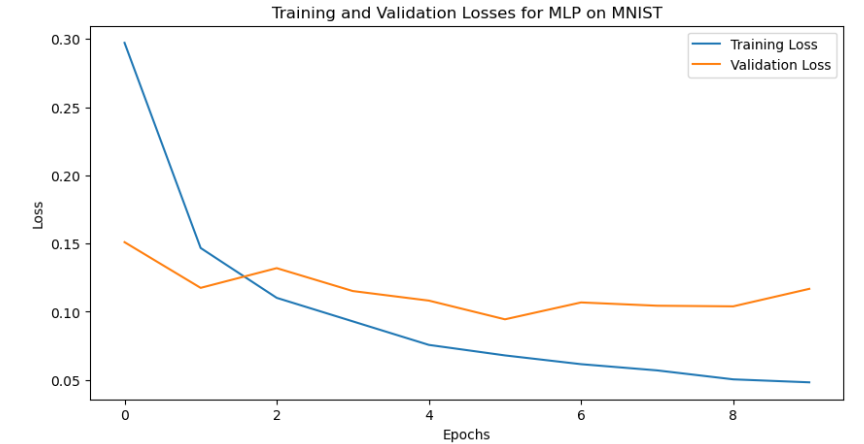
**

Fig. 5. MLP training and validation losses on MNIST

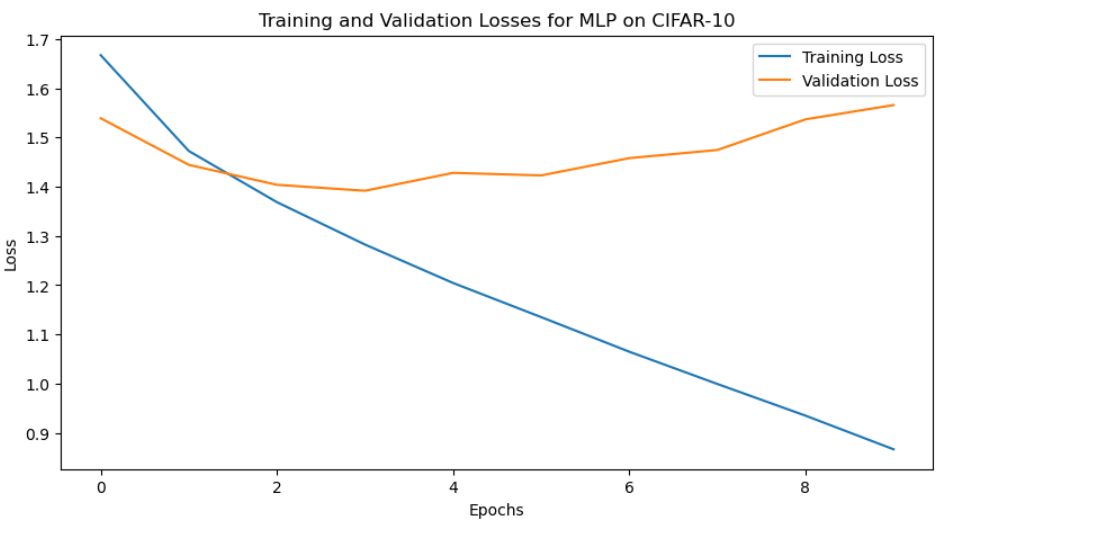
**

Fig. 6. MLP training and validation losses on CIFAR-10

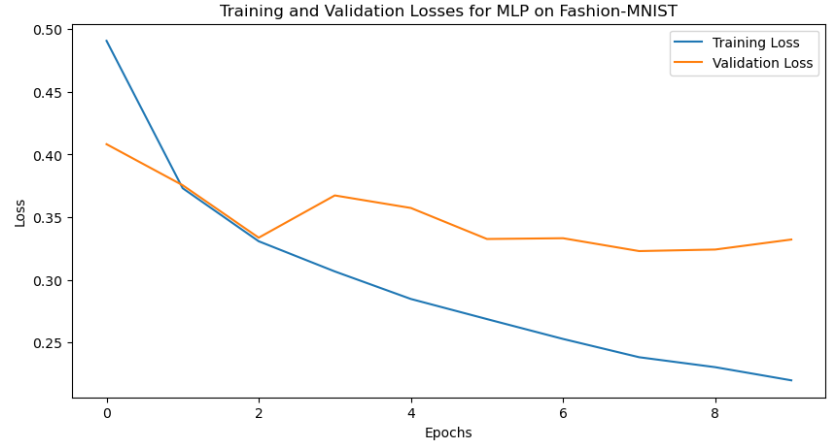
**

Fig. 7. MLP training and validation losses on Fashion-MNIST

*B. LeNet*

LeNet consistently outperformed the MLP model, securing 99% on the MNIST dataset and 90.3% on the Fashion-MNIST dataset. Its performance on the CIFAR-10 dataset was moderately better at 71.6% accuracy. This indicates some capability in handling more complex data but shows that LeNet still is not the most suitable model for high-resolution color images. The following graphs highlight LeNet Utilizating the Datasets and the given training and validation losses of those Datasets.

**

Fig. 8. LeNet training and validation losses on MNIST

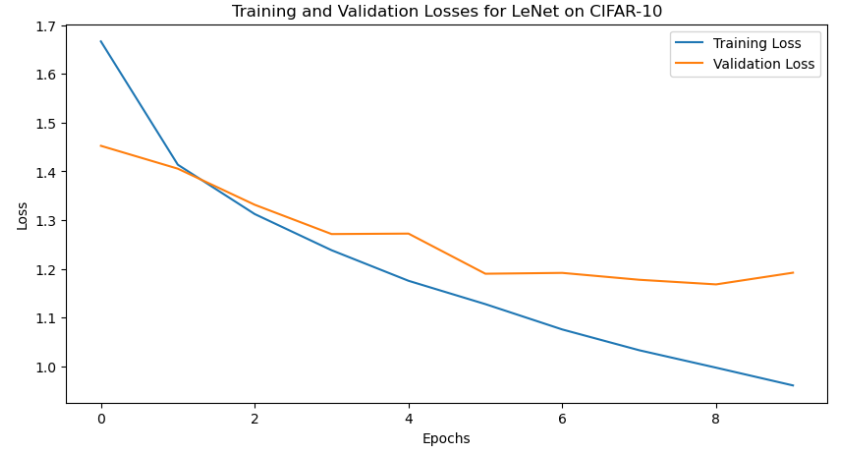
**

Fig. 9. LeNet training and validation losses on CIFAR-10

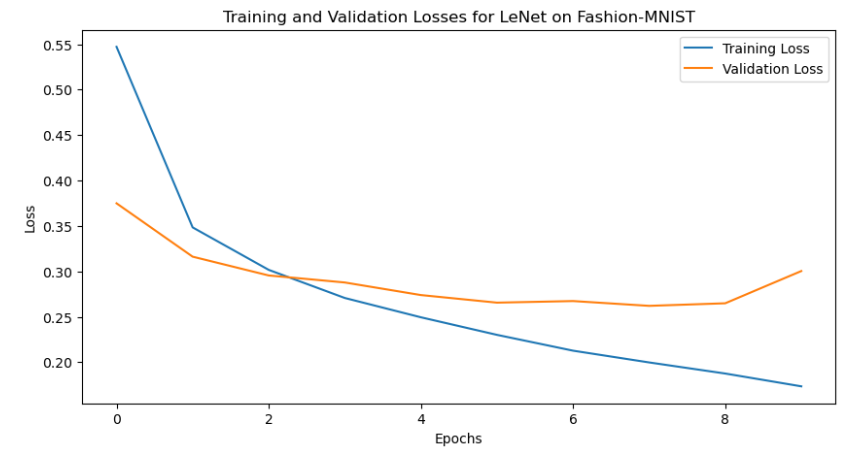
**

Fig. 10. LeNet training and validation losses on Fashion-MNIST

*C. FlexNet*

The FlexNet architecture, specifically designed for this experiment, showed promising results. It achieved accuracies of 99.1% on the MNIST dataset, 91.6% on the Fashion-MNIST dataset and 82% on the CIFAR-10 dataset. The model also achieved the lowest validation losses across all datasets. These outcomes suggest that FlexNet possesses robust generalization capabilities, making it a strong candidate for diverse image processing tasks while balancing performance with computational efficiency. Given below are the graphs for each Dataset the FlexNet architecture used.

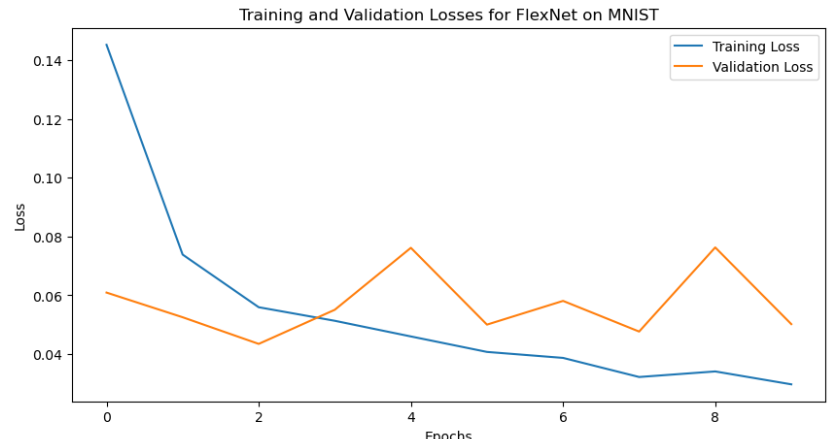
**

Fig. 11. FlexNet training and validation losses on MNIST

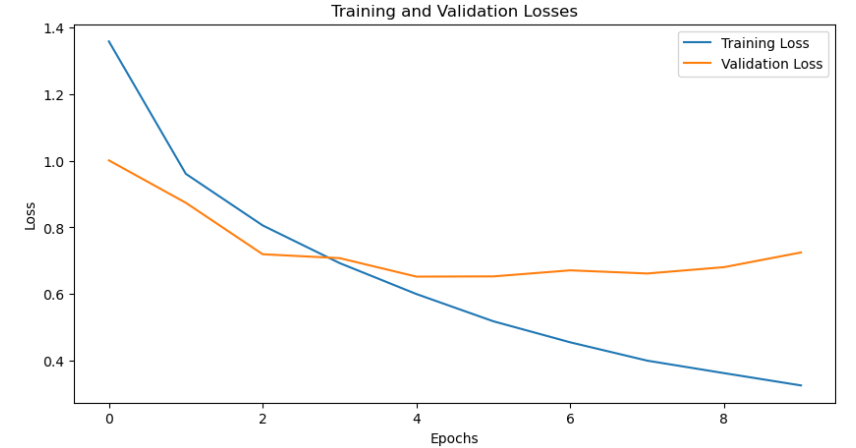
**

Fig. 12. FlexNet training and validation losses on CIFAR-10

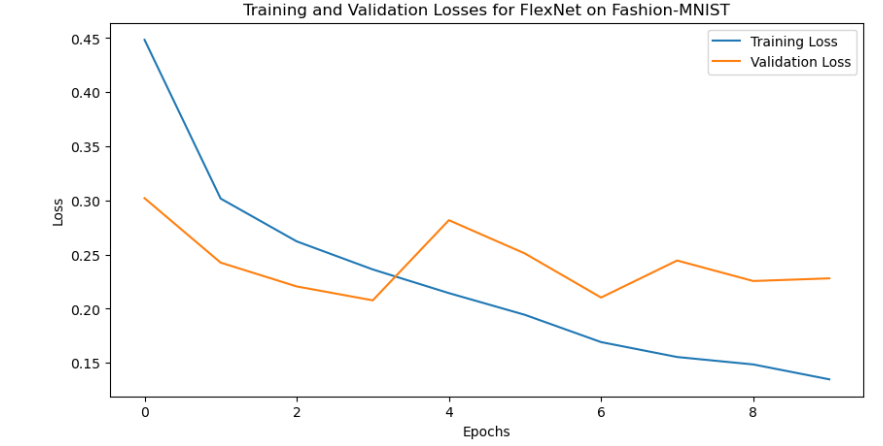
**

Fig. 13. FlexNet training and validation losses on Fashion-MNIST

1. DISCUSSION AND CONCLUSION

The comprehensive experimentation across multiple neural network architectures has provided a deep understanding of how architectural complexities influence performance on diverse image datasets such as MNIST, Fashion-MNIST and CIFAR-10. It showcased different architectures and their limitations, from the simplest, such as MLP to more complex, such as LeNet all the way to more sophisticated Architectures such as the FlexNet.

The newly introduced FlexNet model was designed to strike a balance between computational efficiency and predictive performance. The FlexNet model improved upon the LeNet model substantially, providing more depth, complexity, and methods to prevent overfitting, which were not included in the LeNet network. This study highlights the importance of architectural considerations in neural network design for image classification tasks. Moreover the promising results from the FlexNet model encourage ongoing exploration and development of new models that can offer both efficiency and high accuracy. Future work should focus on further optimizing FlexNet and similar architectures, perhaps by incorporating techniques like attention mechanisms, advanced regularization methods, or pruning techniques to enhance their performance and utility in more demanding real-world applications, as well as compress the architecture as it grows in depth.

In conclusion, the advancement in convolutional neural network design continues to be pivotal in pushing the boundaries of what is possible in image classification, making it an exciting field with vast potential for innovation and application.

REFERENCES

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998. doi:10.1109/5.726791

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional Neural Networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2017. doi:10.1145/3065386

[3] K. Simonyan and A. Zisserman, “International Conference on Learning Representations,” in Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016. doi:10.1109/cvpr.2016.90

[5] S. H. S. Basha, M. Farazuddin, V. Pulabaigari, S. R. Dubey, and S. Mukherjee, “Deep model compression based on the training history,” Neurocomputing, vol. 573, p. 127257, Mar. 2024. doi:10.1016/j.neucom.2024.127257

[6] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of Convolutional Neural Networks: Analysis, applications, and prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022. doi:10.1109/tnnls.2021.3084827