# Exploring MLP, CNN and Transformer Neural Networks on OrganAMNIST

COMP551 - Assignment 3

Kaibo Zhang
Email: kaibo.zhang@mail.mcgill.ca

Mingshu Liu
Email: mingshu.liu@mail.mcgill.ca

Alek Bedard
Email: alek.bedard@mail.mcgill.ca

November 18, 2024

**Abstract**

This project explores the impact of architectural and design decisions on neural networks for image classification using the OrganAMNIST dataset, implemented with MLP and CNN models. We specifically experimented with MLPs of varying depths and activations, analyzing how network complexity affects performance. Regularization techniques and input normalization were tested to evaluate their influence on generalization. Additionally, the CNN model was also built to assess their ability to handle spatial hierarchies in medical images and outperformed MLP model due to their feature extraction capabilities.

## 1 Introduction

The impact of architectural decisions, activation functions, regularization techniques, and input resolution on model performance is the focus of this assignment. The analysis began with evaluating MLPs of varying depths and activations, revealing that deeper models generally improve accuracy. Regularization comparisons highlighted L2's ability to enhance generalization compared to L1 regularization. Normalization emerged as a crucial factor, as unnormalized data significantly hindered performance. Increasing image resolution improved MLP performance, though it required greater computational resources. Transitioning to CNNs demonstrated their superior ability to leverage spatial features, significantly outperforming MLPs. These findings underscore the importance of network architecture and preprocessing in medical image classification tasks.

## 2 Data Description

The OrganAMNIST dataset [1] is a medical imaging resource designed for multi-class classification across 11 organ categories, including the bladder, liver, and lungs. It comprises 34,561 training samples, 6,491 validation samples, and 17,778 test samples, with each grayscale image resized to 28 × 28 pixels. Exploratory analysis indicates that pixel values are normalized between 0.0 and 1.0, with a mean of 0.4680 and a standard deviation of 0.2472. The class distribution is notably imbalanced; for instance, the liver class contains over 6,000 images, whereas classes like femur-left and femur-right have fewer than 2,000 images each. Exploration was conducted to confirm the data structure, size, and normalization, with pixel values standardized between 0.0 and 1.0. Most experiments described below were conducted on data with this normalization applied.

## 3 Weight Initialization and Training

We evaluated several initialization methods, observing their training dynamics and selecting the best based on validation loss trends. Poor initializations, such as ”zeros,” hindered learning by failing to break symmetry, while ”uniform” and ”Gaussian” showed slower convergence. ”Xavier” and ”Kaiming” demonstrated faster and more stable training, with ”Xavier” ultimately chosen for its balanced weight scaling and best performance in validation accuracy (0.9056). This method ensures efficient gradient flow, preventing vanishing or exploding gradients, which is particularly effective for the layered structure of this classification task. Table 4 offers a visual comparison of the training loss curves for different initialization methods.

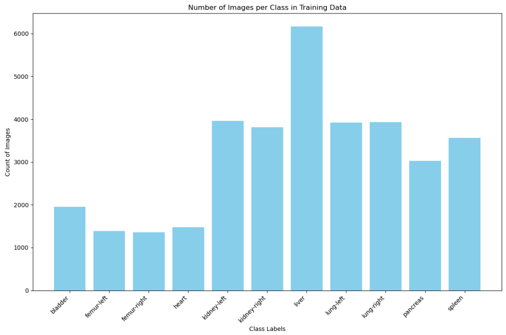

Figure 1: Class distribution in the OrganAMNIST training dataset

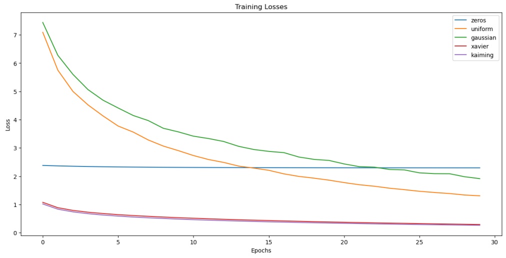

Figure 2: Training loss curve for different weight initialization methods

## 4 Analysis of MLP Architectural Complexity for Grayscale Image Classification

This experiment examines the effect of increasing network depth on the performance of MLP models in image classifications. Three structures were evaluated: a model without hidden layers, a single hidden layer, and two hidden layers (256 neurons each, Relu activation). The learning rate was set to 0.1 based on validation set performance, with observations indicating that the model begins to overfit after 10 training epochs (Figure 5).The results indicate that deeper networks with non-linear activations lead to better performance, with test accuracy improving from 55.41% for the model with no hidden layers to 73.10% for one hidden layer, and further to 75.64% with two hidden layers. The superior performance of the two-hidden-layer model suggests that additional depth facilitates learning more complex patterns. However, the marginal improvement from the second hidden layer may reflect the limits of model capacity given the dataset's size and complexity (Figure 9).

## 5 Evaluation of Activation Functions in MLP Performance

We then compared the performance of Tanh and Leaky ReLU activation functions against the baseline ReLU in MLP models. While ReLU serves as a strong baseline due to its simplicity and effectiveness in mitigating gradient vanishing, Tanh exhibited slightly lower performance, likely constrained by gradient saturation in deeper layers. Leaky ReLU outperformed both, offering better gradient flow for negative values, which supports more stable and efficient learning (Table 1). The improved performance with Leaky ReLU highlights its suitability for the task of grayscale image classification, where nuanced feature extraction is critical, and consistent gradient updates are essential.

| Activation function | Test accuracy |
|---|---|
| ReLu | 0.7564 |
| Tanh | 0.7276 |
| Leaky ReLu | 0.7575 |

Table 1: Effect of the activation function on training loss for the MLP architecture with the best activation function highlighted

## 6 Evaluation of L1 and L2 Regularization on Model Accuracy

The learning rate and epoch were tuned separately on the validation set before assessing the impact of regularization (Figure 10). Both L1 and L2 regularizations required more epochs before overfitting compared to the unregularized model, as they effectively constrain model complexity. However, both regularizations reduced test performance compared to the unregularized model, with L2 achieving a test accuracy of 63.11% (versus 75.64% unregularized) and L1 only 18.48%. This drop occurs because regularization penalizes model weights, potentially oversimplifying the model and limiting its ability to fully capture the intricate features of the dataset. L2 regularization, which preserves weight distribution, outperformed L1, as the latter induces excessive sparsity, severely reducing the network's representational capacity for this image classification task.

# 7 Impact of Unnormalized Data on MLP Performance

Training the MLP on unnormalized data resulted in a test accuracy of 74.79%, slightly lower than the 75.64% achieved with standardized pixel values. The drop in performance stems from the lack of normalization, which leads to unstable gradient updates and slower convergence during training. Normalization ensures that input features are on a similar scale, helping the model learn more efficiently and avoid being biased by larger magnitude inputs. This highlights normalization as a critical preprocessing step for enhancing stability and overall performance.

# 8 Comparison of L2 and L1 Regularizations on 28x28 vs. 128x128 Data

The learning rate was set to 0.001 to prevent overshoot and guarantee stability with more complex input data. The number of epochs was separately tuned on the validation set (L2: 60, L1: 30) (Figure 11). Using 128x128 data improved the test accuracy of the L2-regularized MLP from 63.20% to 72.49%, as the higher resolution allowed the model to capture more detailed features, enhancing generalization. However, L1 regularization did not benefit, maintaining a test accuracy of 18.48%, likely due to its aggressive weight penalization, which constrained the model's capacity to leverage the additional input information (see figure 3). The training curves show that models trained on 128-pixel data progress more gradually compared to those on 28-pixel data. This is due to the increased complexity and higher number of parameters introduced by larger input dimensions, requiring the model to take smaller steps to converge. For L2 regularization, this gradual decrease in loss reflects the model's ability to better capture detailed features over extended epochs. In contrast, L1 regularization fails to leverage the additional resolution, as its sparsity-inducing penalties restrict the model's capacity to adapt to the richer input. Training with 128x128 data also led to longer epoch times, averaging 100-120 seconds compared to 10-15 seconds for 28x28 data, a direct consequence of the increased computational demand from larger input sizes and more extensive parameter updates. Figure 4 illustrates the training times for both regularization techniques and dataset sizes.

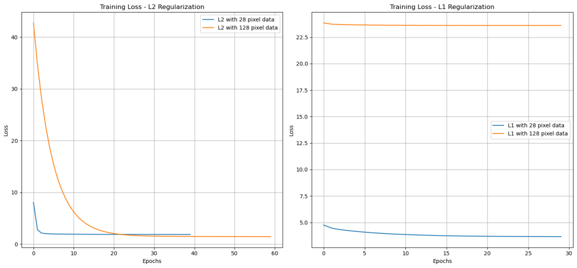

Figure 3: Loss curves for L1 and L2 regularized MLP models trained on 28x28 and 128x128 data

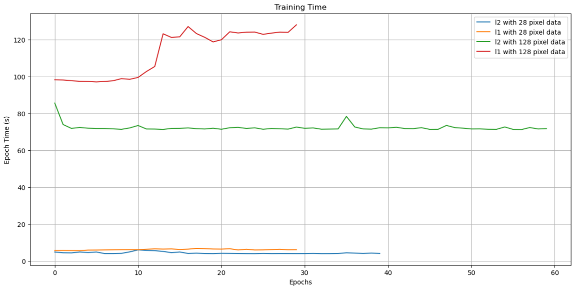

Figure 4: Training times for varying regularization techniques and dataset image sizes

# 9 Comparison of CNN and MLP Performance on 28x28 Data

The CNN model was trained with the same stable learning rate of 0.001 for similar considerations and with early stopping with $patience = 3$ to facilitate the training process. Implementing a CNN with two convolutional layers and a fully connected layer with 256 neurons demonstrated a notable improvement in test accuracy compared to the MLP models (Table 2). This performance boost highlights CNN's ability to effectively capture spatial

| Model | Test accuracy |
|---|---|
| ReLu MLP (2 hidden layers, 256 neurons each) | 0.7564 |
| Leaky ReLu MLP (2 hidden layer, 256 neurons each) | 0.7575 |
| Regular CNN (2 conv. layers + 1 dense layer) | 0.7920 |

Table 2: Architecture comparison between MLP and CNN

hierarchies and local structural patterns inherent in image data, which MLPs cannot exploit due to their lack of spatial awareness. CNNs leverage convolutional operations to extract localized features, such as edges and textures, and build higher-level abstractions as the network deepens. This structural advantage allows CNNs to generalize better on image classification tasks, as evidenced by the stable learning curves and reduced loss observed during training. These results align with the expected strengths of CNNs in handling image data.

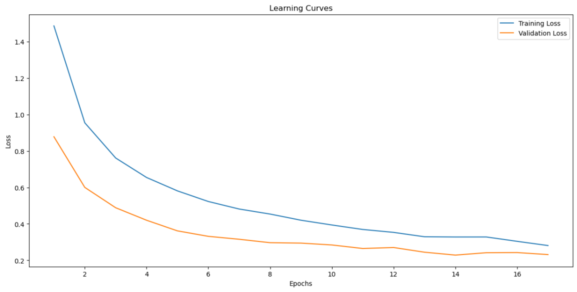

Figure 5: CNN architecture learning curve

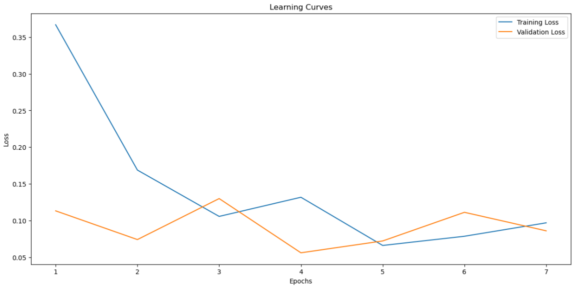

Figure 6: Per-epoch loss curve while training the CNN model on 128X128 images

## 10 Impact of Modified CNN Architecture on 128x128 Images

The CNN was modified by tuning hyperparameters derived from validation set performance: $conv1\_channels = 64$, $conv2\_channels = 256$, $fc\_neurons = 512$, $pool\_kernel = 3$, and $pool\_stride = 3$ (Python dictionary 1, Table 7). The training setup remained the same as the original CNN for consistency. These adjustments enhanced the model's capacity to extract spatial features and generalize effectively. The modified CNN trained on 128x128 images achieved a test accuracy of 88.7%, significantly outperforming both the original CNN (accuracy 79.2%) and the best-performing MLP (accuracy 72.49%). The improvement highlights the effectiveness of the tuned parameters, which increased the model's ability to capture detailed spatial patterns and hierarchies. By adding pooling layers and adjusting neuron counts, the modified CNN effectively reduced the spatial dimensions while preserving key features, enabling it to generalize better on higher-resolution data. The learning curve shows that the modified CNN reaches a local minimum much faster than both the original CNN and the MLP, needing fewer epochs to converge despite the greater input complexity (Figure 6). This efficiency stems from the CNN's ability to capture spatial hierarchies and extract essential features early in the training process, unlike the MLP, which processes each input feature independently and requires more epochs to approximate complex patterns. The addition of pooling layers and optimal parameters further streamlined learning, allowing CNN to handle high-resolution inputs with greater precision and efficiency.

## 11 Comparing with a Pre-Trained Model

resnet101 was used as our pre-trained CNN, that we re-trained with new fully-connected layers while keeping the original convolutional layers. The number of new fully connected layers along with the number of units in each layer was chosen through validation experiments (Table 6). These experiments consisted in varying the number of fully connected layers by starting with an architecture similar to the vanilla CNN, until one layer above the original resnet 101 architecture. The best model that emerged from these experiments was a model with 4 fully connected layers with 4096, 4096, 1000 and 1000 respective units. A ReLu activation function is used between each of these layers. The final testing accuracy of the model was approximately 84.4% with an area under the curve (AUC) of 0.985.

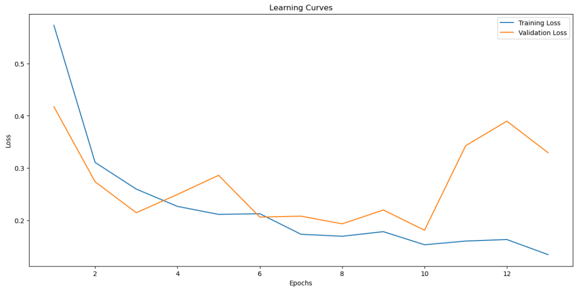

Figure 7: Finetuned ResNet101 train and validation loss

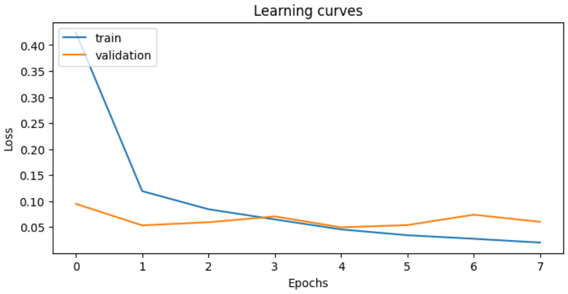

Figure 8: Finetuned ViT train and validation loss

As depicted in Table 3, the resnet101 finetuned model outperforms the MLP, but lags the vanilla CNN model. The comparison between the finetuned CNN and the MLP is coherent with the theory: MLP are not good to capture spatial context, while CNNs can make sense of the surrounding pixels. On the other hand, the vanilla CNN outperformed the finetuned resnet. This result could be explained by the fact that the entire vanilla model architecture was constructed and trained specifically on the OrganAMNIST dataset with large compute capabilities (GPU). The model could offer a good generalization on similar unseen data. The pretrained model had a significant part of its architecture tuned on another dataset, which could explain a reduced accuracy. Finetuning the pretrained ResNet with added dense layers required longer epochs to converge due to the need to adapt its rich, pretrained features to the new dataset while balancing the additional parameters from the dense layers. Unlike the MLP, which took over 30 epochs to converge from scratch, ResNet leverages its hierarchical features for more efficient learning, despite taking longer than the modified CNN. In order to push our reflection further, we finetuned a pre-trained transformer model (ViT - Vision Transformer) and compared its results with the pretrained CNN. The pre-trained transformer model was finetuned with 3 new fully connected layers of 1024, 512 and 256 units respectively. Each of these layers were connected with a ReLu activation function. In order to use the ViT with the original pre-trained convolutional layers, we had to run the training and testing with the 224X224 pixel dataset, which provides more details than the 128X128. The final testing accuracy of the model was approximately 94.5% with a rounded AUC of 1, which outperforms all the other models developed in this assignment. Although the resolution of the images were greater, we can still observe that self-attention mechanisms of transformer models increase the overall accuracy for image classification tasks.

| | MLP | Vanilla CNN | resnet101 | ViT |
|---|---|---|---|---|
| Accuracy | 72% | 89% | 84.4% | 94.5% |
| Epochs | 30 | 4 | 10 | 4 |

Table 3: Comparison between the MLP, Vanilla CNN, finetuned resnet101, and finetuned ViT models

## 12 Conclusion

This study highlights the critical role of architectural choices, preprocessing techniques, and hyperparameter tuning in enhancing neural network performance for medical image classification using the OrganAMNIST dataset. Transitioning from MLPs to CNNs proved highly effective, as CNNs leveraged spatial features to outperform MLPs significantly. Key techniques such as input normalization, optimal weight initialization, and regularization were essential for improved generalization and training stability. Increasing input resolution enhanced accuracy but introduced computational challenges, while fine-tuning pre-trained models like ResNet101 and ViT demonstrated the superior performance of self-attention mechanisms in transformers. Future work could explore hybrid architectures combining CNNs and transformers, advanced attention mechanisms, and multi-resolution models to balance efficiency and accuracy. Additionally, addressing class imbalance through data augmentation and improving interpretability with techniques like Grad-CAM could further refine model performance and applicability. These approaches would enhance the reliability and robustness of neural networks for medical imaging tasks.

## 13 Statement of contributions

Mingshu took the lead in writing the report and contributed with some parts of coding, mainly with dataset exploration. Alek took the lead in formatting the report in Latex and coding some other parts of the experiments and graph developments. Kaibo took the lead in model implementation and coding with a focus on experiments.

## 14 Appendix

| Initialization method | Validation accuracy |
|---|---|
| Zeros | 0.1591 |
| Uniform | 0.8107 |
| Gaussian | 0.8060 |
| Xavier | 0.9056 |
| Kaiming | 0.8976 |

Table 4: Training loss value table for different weight initialization methods with the best technique highlighted

| Learning rate ($\alpha$) | Epochs | Validation accuracy | Overfitting observed |
|---|---|---|---|
| 0.1 | 10 | 0.9162 | No |
| 0.1 | 20 | 0.9128 | Yes |
| 0.01 | 10 | 0.9126 | No |
| 0.01 | 20 | 0.9125 | Yes |
| 0.001 | 10 | 0.9125 | No |
| 0.001 | 20 | 0.9125 | Yes |

Table 5: Effect of different learning rates on validation accuracy and overfitting

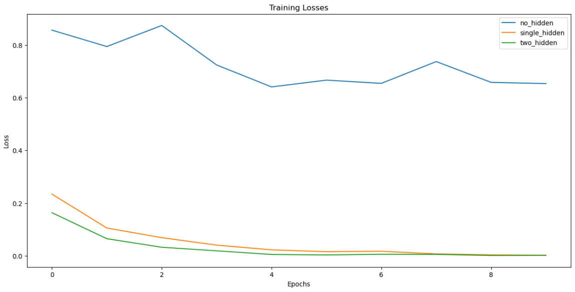

Figure 9: Effect of MLP architecture on training loss

```
param_grid = {
    'conv1_channels': [64, 128],
    'conv2_channels': [128, 256],
    'fc_neurons':     [256, 512],
    'pool_kernel':    [2, 3],
    'pool_stride':    [2, 3]
}
```

Listing 1: Tuned CNN architecture

## References

[1] Patrick Bilic, Patrick Ferdinand Christ, Xuanang Xu, Fugen Zhou, et al. Abdominal ct dataset for multi-class classification (11 classes). Derived from the Liver Tumor Segmentation Benchmark (LiTS) and related works, 2019. 58,830 samples (34,561 training / 6,491 validation / 17,778 testing).

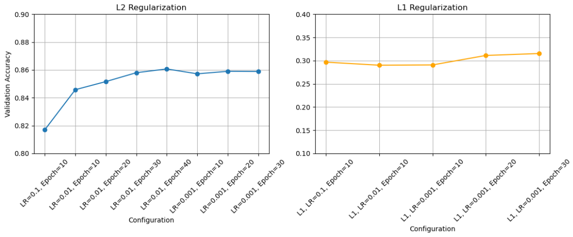

Figure 10: Effect of Lasso (L1) and Ridge (L2) regularization on training loss

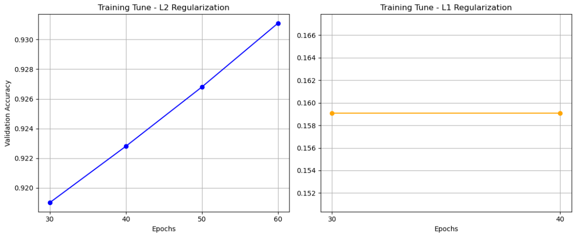

Figure 11: Epoch hyperparameter tuning experiment curve

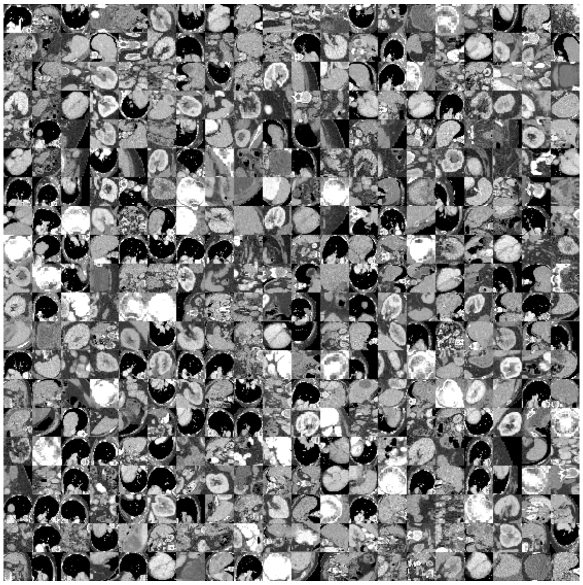

Figure 12: 28X28 normalized and grayscaled images from the OrganAMNIST dataset

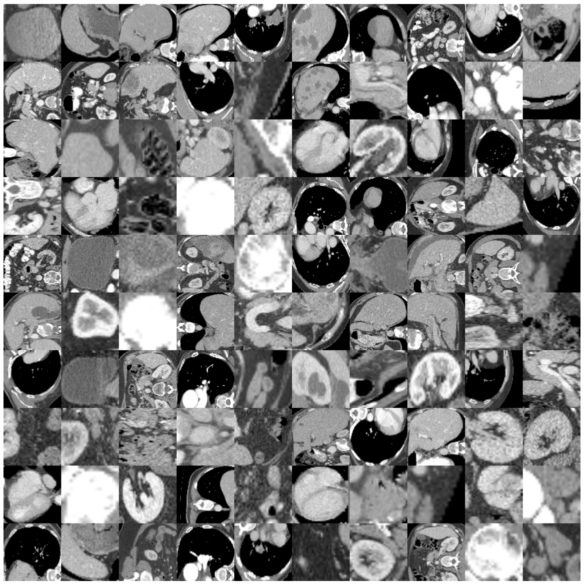

Figure 13: 128X128 normalized and grayscaled images from the OrganAMNIST dataset

| Dense layer architecture | Training Loss | Validation Loss |
|---|---|---|
| 2048, 1024, 2014, 512, 512, 256, 256 | 0.3053 | 0.2039 |
| 4096, 4096, 1000 | 0.2247 | 0.2170 |
| 4096, 4096, 1000, 1000 | 0.1528 | 0.1806 |
| 512, 256, 256, 128, 64 | 0.2102 | 0.2661 |

Table 6: Pre-trained resnet101 CNN architecture tuning with the best model highlighted

| CV1 | CV2 | FC1 | Pooling kernel | Pooling stride | Training Loss | Validation Loss |
|---|---|---|---|---|---|---|
| 64 | 128 | 256 | 2 | 2 | 0.2368 | 0.1005 |
| 64 | 128 | 256 | 2 | 3 | 0.1093 | 0.1135 |
| ... | ... | ... | ... | ... | ... | ... |
| 64 | 256 | 512 | 2 | 2 | 0.0971 | 0.0882 |
| 64 | 256 | 512 | 3 | 3 | 0.1319 | 0.0563 |
| ... | ... | ... | ... | ... | ... | ... |
| 128 | 256 | 512 | 3 | 2 | 0.1092 | 0.0836 |
| 128 | 256 | 512 | 3 | 3 | 0.0687 | 0.0946 |

Table 7: Hyper-parameter tuning for the vanilla CNN architecture with the best model highlighted