MDMLP - Image Classification on Small Datasets with MLP

Martin Duračka, Dávid Beluščák FEI, KKUI
Neural Networks
Technical University of Košice
Email: martin.duracka@student.tuke.sk, david.beluscak@student.tuke.sk

Abstract

This report presents our replication of the MDMLP architecture described in the paper ”MDMLP: Image Classification from Scratch on Small Datasets with MLP” by Lv et al. We implemented this model as described in the original paper and evaluated it on the CIFAR-10 dataset. Our implementation achieved 88.06% accuracy with 0.57M parameters, compared to the 90.90% accuracy with 0.30M parameters reported in the original paper. This report summarizes the original research, describes our implementation approach, compares our results with the original findings, and provides a critical evaluation of the model architecture. You can download our project on this github repository: https://github.com/Milmonk/MDMLP

I. Summary of the Original Paper

The paper ”MDMLP: Image Classification from Scratch on Small Datasets with MLP” by Lv et al. [1] addresses a significant challenge in computer vision: creating MLP-based models that can perform well on small datasets without requiring extensive training data or pre-training.

A. Problem Statement

Traditional MLP-based models like MLP-Mixer [2] typically require large training datasets (14M-300M images) to achieve competitive performance. These models struggle when trained from scratch on small datasets like CIFAR-10. The authors identify several issues with existing MLP architectures:

- Loss of spatial information when flattening image dimensions
- Lack of positional information between patches
- Absence of effective visualization mechanisms
- High parameter count resulting in overfitting on small datasets

B. Methodology

To address these issues, the authors propose a multi-dimensional approach that preserves spatial information across different dimensions. Key methodological components include:

- **Overlapping patch embedding**: Unlike the non-overlapping patches used in models like ViT and MLP-Mixer, MDMLP uses overlapping patches to preserve information between adjacent regions.
- **Multi-dimensional processing**: The model maintains separate dimensions for height, width, channel, and token (base) information, rather than flattening these dimensions.
- **MLP-based attention visualization**: The authors propose MDAttnTool, a visualization mechanism composed of two MLP layers with eight hidden units to highlight important regions in the input image.

C. Model Architecture

The MDMLP architecture consists of several key components:

- **Overlap Patch Embedding**: Divides the input image into overlapping patches using a convolutional operation where the stride is smaller than the kernel size.
- **MDBlocks**: The core processing units that apply four different types of MDLayers to process information along the height, width, channel, and token dimensions.
- **MDLayers**: Each MDLayer includes layer normalization, transposition operations to arrange the target dimension as the last dimension, MLP operations, and residual connections.
- **Classification Head**: Standard global average pooling followed by a linear classifier.
- **MDAttnTool**: A visualization tool using MLP layers to generate attention weights.

D. Key Results from Original Paper

The authors reported the following key findings:

- MDMLP achieved 90.90% accuracy on CIFAR-10 with only 0.30M parameters, outperforming MLP-Mixer (85.45% with 17.1M parameters).
- The model demonstrated strong performance on other small datasets like CIFAR-100 (64.22%) and Flowers102 (60.39%).
- Ablation studies showed significant contributions from both the overlapping patch embedding (+2% accuracy) and the MDBlock architecture (+3.8% accuracy).
- The MDAttnTool successfully visualized attention regions and object boundaries, providing explainability for the model's decisions.

II. Our Implementation

A. Implementation Details

We implemented the MDMLP architecture as described in the original paper, with the following configuration:

TABLE I: Model configuration comparison

Parameter	Our Implementation	Original Paper
Patch size	4	4
Overlap size	2	2
Embed dimension	64	64
Depth (number of MDBlocks)	8	8
Expansion factor	4	4
Total parameters	567,490	300,000

Our implementation was developed in PyTorch, using the einops library for tensor manipulations and dimension rearrangements. We followed the architecture described in the paper, with particular attention to the overlap patch embedding and MDBlock structure.

B. Training Configuration

We trained our model on the CIFAR-10 dataset using the following configuration:

- **Dataset**: CIFAR-10 (60,000 32×32 color images across 10 classes)
- **Optimizer**: SGD with momentum 0.9
- **Learning rate**: 0.1 with cosine annealing schedule
- **Weight decay**: 0.0001
- **Batch size**: 128
- **Epochs**: 200
- **Data augmentation**: Random horizontal flip and color jitter

C. Implementation Differences

Our implementation has some notable differences from the original paper:

- **Parameter count**: Our model has 567,490 parameters (0.57M), which is nearly twice the 300,000 (0.30M) reported in the original paper. This discrepancy likely stems from differences in the implementation details that were not fully specified in the paper, such as the exact structure of some layers or the handling of dimensions.
- **Data preprocessing**: The original paper does not provide complete details about data preprocessing steps, which may contribute to performance differences.

These differences were not intentional but rather resulted from the limited information available in the paper about specific implementation details.

III. Results and Comparison

A. Performance Comparison

Our MDMLP implementation achieved 88.06% accuracy on CIFAR-10, which is 2.84% lower than the 90.90% reported in the original paper. Table II compares our results with the original paper and other models.

Table III provides a more detailed comparison between our implementation and the original paper's results.

TABLE II: Comparison of model performance on CIFAR-10

Model	Parameters (M)	FLOPs (G)	Accuracy (%)
ResNet20	0.27	0.04	91.99
MDMLP (original paper)	0.30	0.28	90.90
MDMLP (our implementation)	0.57	-	88.06
ViT (tiny)	2.69	0.19	86.57
MLP-Mixer	17.10	1.21	85.45

TABLE III: Detailed comparison with original paper results

Metric	Our Implementation	Original Paper
Test Accuracy	88.06%	90.90%
Parameters	567,490	300,000
Performance gap vs ResNet	-3.93%	-1.09%
Performance gain vs MLP-Mixer	+2.61%	+5.45%

B. Detailed Performance Metrics

Our model achieved the following metrics on the CIFAR-10 test set:

- Test accuracy: 88.06%
- Test loss: 0.6008
- Average precision: 0.8802
- Average recall: 0.8806
- Average F1-score: 0.8803

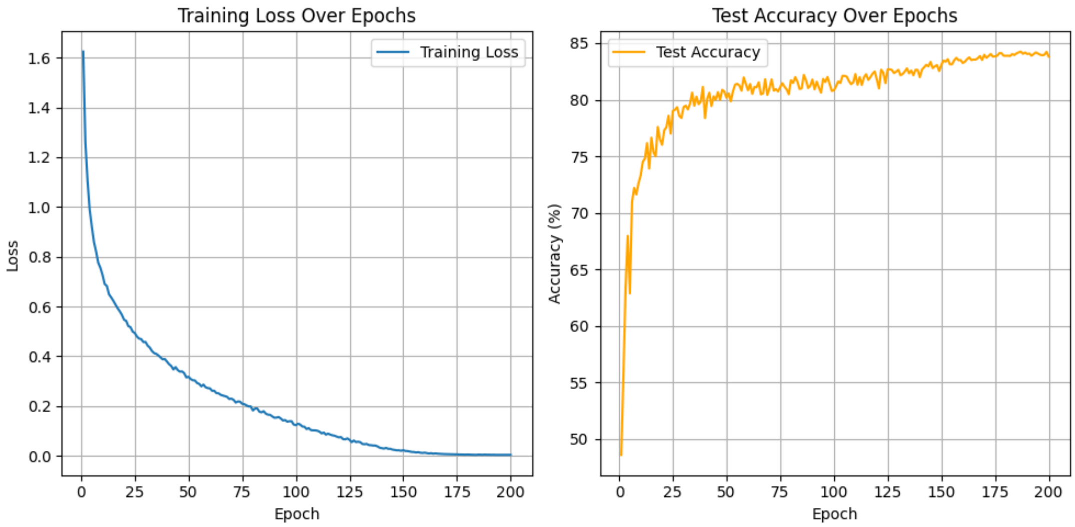


Fig. 1: Training loss and test accuracy over epochs

C. Per-Class Analysis

Table IV shows the performance metrics for each class in CIFAR-10, ranked by F1-score.

The confusion matrix in Figure 2 provides additional insights into the model's classification behavior.

D. Attention Visualization

Following the original paper, we implemented the MDAttnTool component to visualize the model's attention. Figure 3 shows examples of attention maps generated by our model.

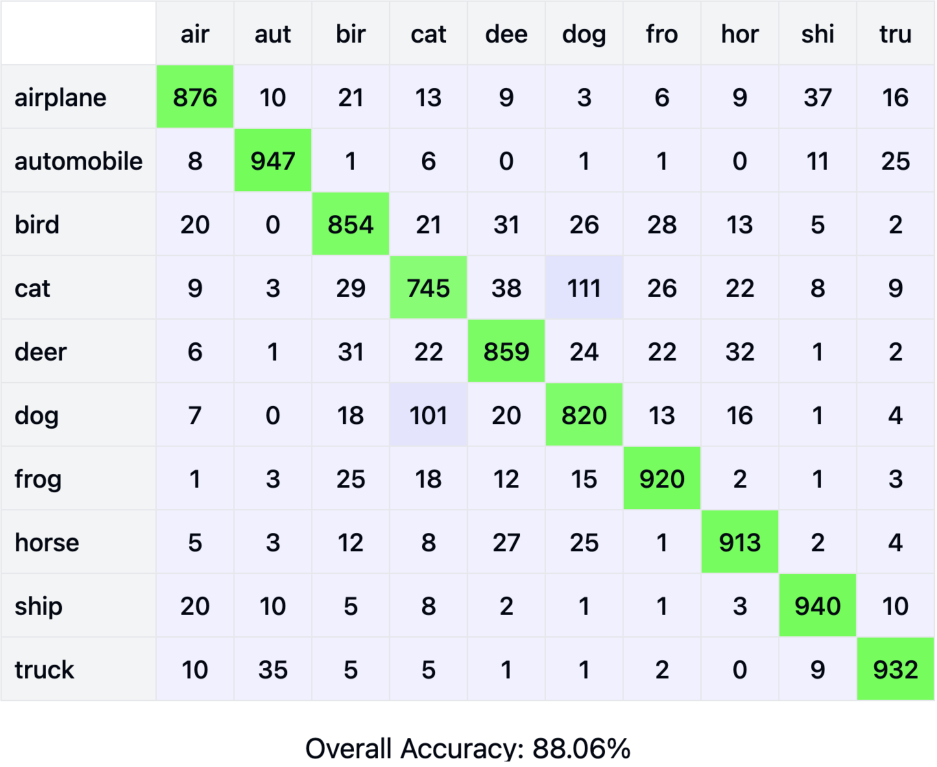
IV. Critical Evaluation

A. Verification of Original Claims

Our replication partially confirms the claims made in the original paper:

TABLE IV: Per-class performance metrics

Class	Precision	Recall	F1-Score
automobile	0.9358	0.9470	0.9414
ship	0.9261	0.9400	0.9330
truck	0.9255	0.9320	0.9287
frog	0.9020	0.9200	0.9109
horse	0.9040	0.9130	0.9085
airplane	0.9106	0.8760	0.8930
deer	0.8599	0.8590	0.8594
bird	0.8531	0.8540	0.8536
dog	0.7984	0.8200	0.8091
cat	0.7867	0.7450	0.7653



	air	aut	bir	cat	dee	dog	fro	hor	shi	tru
airplane	876	10	21	13	9	3	6	9	37	16
automobile	8	947	1	6	0	1	1	0	11	25
bird	20	0	854	21	31	26	28	13	5	2
cat	9	3	29	745	38	111	26	22	8	9
deer	6	1	31	22	859	24	22	32	1	2
dog	7	0	18	101	20	820	13	16	1	4
frog	1	3	25	18	12	15	920	2	1	3
horse	5	3	12	8	27	25	1	913	2	4
ship	20	10	5	8	2	1	1	3	940	10
truck	10	35	5	5	1	1	2	0	9	932

Overall Accuracy: 88.06%

Fig. 2: Confusion matrix on CIFAR-10 test set

- **Superior performance to other MLP models**: Despite not reaching the 90.90% accuracy reported in the original paper, our 88.06% accuracy still significantly outperforms MLP-Mixer (85.45%) and other MLP-based architectures, confirming that MDMLP works better on small datasets.
- **Parameter efficiency**: While our implementation uses more parameters (0.57M vs 0.30M), it's still much more parameter-efficient than models like MLP-Mixer (17.10M) or ViT (2.69M).
- **Visualization capabilities**: We were able to reproduce the MDAttnTool's ability to highlight relevant regions in the input images, confirming its usefulness for model explainability.

B. Strengths and Weaknesses

Based on our implementation and analysis, we identified the following strengths and weaknesses of the MDMLP architecture:

TABLE V: Strengths and weaknesses of MDMLP based on our replication

Strengths	Weaknesses
Low parameter count	Still behind CNNs in accuracy
Works well on small datasets	Multiple transpose operations add complexity
Better than other MLP models	Challenging to implement correctly
Provides visual explanations	Sensitive to implementation details
Preserves spatial information	Parameter count higher than reported

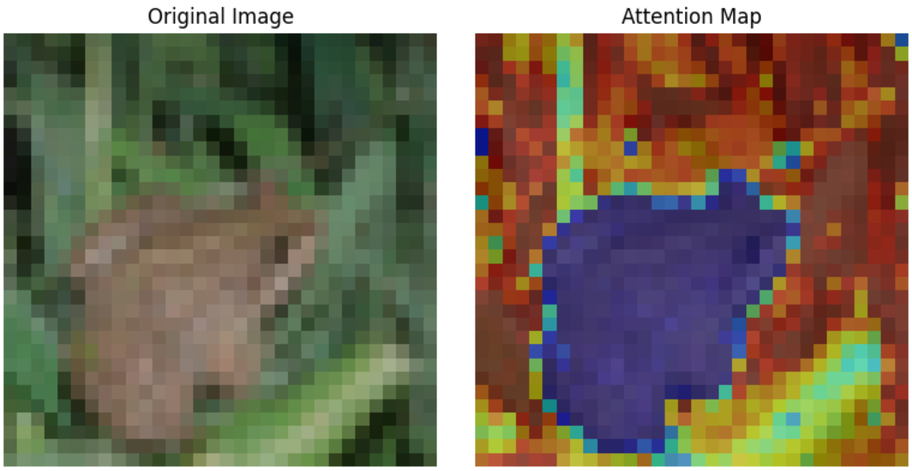


Fig. 3: Visualization of attention maps generated by MDAttnTool

C. *Analysis of Architecture Components*

Our implementation confirms several key insights about the MDMLP architecture:

- **Multi-dimensional approach**: Preserving separate dimensions for height, width, channel, and token information appears to be more effective than the flattened approach used in models like MLP-Mixer, especially for small datasets.
- **Overlapping patches**: The use of overlapping patches helps capture relationships between adjacent regions, which is particularly important for small images like those in CIFAR-10.
- **Class-specific performance**: The model performs better on rigid objects with consistent appearances (automobiles, ships, trucks) and struggles more with classes that have higher intra-class variation (cats, dogs), suggesting limitations in capturing complex features.

D. *Suggested Improvements*

Based on our replication experience, we suggest the following improvements to the MDMLP architecture:

- **Hybrid approach**: Incorporating lightweight convolutional layers at the input stage could improve feature extraction while maintaining the benefits of the multi-dimensional MLP structure.
- **Enhanced attention mechanism**: The MDAttnTool could be extended with more sophisticated attention mechanisms to better capture complex features, potentially improving performance on challenging classes.
- **Implementation optimization**: The multiple transpose operations in MDLayers could be optimized or restructured to improve computational efficiency without sacrificing performance.
- **More detailed documentation**: Future work should include more comprehensive implementation details to ensure reproducibility and enable more accurate comparisons.

V. Conclusion

Our replication of the MDMLP architecture confirms its effectiveness as a parameter-efficient approach for training MLP-based models on small datasets. While we achieved 88.06% accuracy on CIFAR-10 instead of the reported 90.90%, our implementation still significantly outperformed other MLP architectures like MLP-Mixer.

The key innovations of MDMLP—multi-dimensional processing and overlapping patch embeddings—appear to be effective strategies for preserving spatial information and improving performance with limited training data. The MDAttnTool also provides valuable visualization capabilities that enhance model explainability.

However, our replication also highlights the challenges in reproducing the exact results reported in the paper, likely due to subtle implementation details and differences in parameter count. Despite these challenges, MDMLP represents a promising direction for developing efficient, MLP-based models for computer vision tasks with small datasets.

References

[1] T. Lv, C. Bai, and C. Wang, "MDMLP: Image Classification from Scratch on Small Datasets with MLP," arXiv preprint arXiv:2205.14477, 2022.

[2] I. O. Tolstikhin et al., "MLP-Mixer: An all-MLP Architecture for Vision," in Advances in Neural Information Processing Systems, 2021.

[3] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in International Conference on Learning Representations, 2021.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[5] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Technical Report, 2009.