A Comparison of Model Architectures for Missing Pixel Predictions

Pittawat Taveekitworachai
Graduate School of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
gr0609fv@ed.ritsumei.ac.jp

***Abstract*—This paper presents a comparative analysis of model architectures for predicting missing pixels in images. The study evaluates the performance of three models: multi-layer perceptron (MLP), convolutional neural network, and MobileNetV2, trained on two datasets. The first dataset contains carefully curated images provided by an instructor, while the second dataset is a 1K sampled subset of Tiny-ImageNet-2000. I also investigate the scalability of the MLP by training a larger architecture on the 1K-Tiny-ImageNet-2000 dataset and evaluate the original MLP on the full Tiny-ImageNet-2000. Additionally, an open-sourced stable diffusion model tuned for inpainting is evaluated for comprehensive comparison.**

Performance is measured using the mean squared error metric. Results demonstrate that models trained on the larger dataset generally outperform those trained on the smaller original dataset. The CNN and MobileNetV2 architectures consistently outperform the traditional MLP in both datasets, showcasing their effectiveness in capturing spatial patterns and hierarchies through convolution layers. Surprisingly, scaling the MLP to a larger architecture does not improve performance. The stable diffusion model shows worse performance for this task, in contrast to general observations where it performs well on general image inpainting tasks. Future research directions include diversifying datasets, exploring novel model architectures, and incorporating data augmentation and transfer learning techniques for further improvement.

***Index Terms*—missing pixels, cnn, mlp, inpainting, stable diffusion**

I. Introduction

Predicting missing pixels in images, also known as image inpainting, is a fundamental problem in computer vision [1]. The main objective of inpainting is to recover missing or corrupted regions in an image, generating visually plausible content that blends seamlessly with the surrounding context. It plays a crucial role in various image processing tasks and applications, such as image restoration, object removal, and image synthesis [2]. In recent years, deep learning models have shown remarkable success in tackling this challenge.

Convolutional Neural Networks (CNNs) have emerged as the dominant architecture for numerous computer vision tasks due to their ability to extract hierarchical and spatial features from images [3]. This inherent property makes them an ideal candidate for image inpainting tasks, as they can better capture local and global relationships between pixels. Moreover, neural network architectures have evolved beyond simple CNNs, and novel models have been introduced. Among these, MobileNetV2 [4], an efficient neural network architecture, has gained attention for its ability to perform well in resource-constrained environments. The exploration of such diverse model architectures is essential to identify the most suitable approach for this task.

I conduct a thorough assessment of different model architectures by comparing three distinct approaches: the traditional multi-layer perceptron (MLP), simple CNN, and MobileNetV2. The evaluation is performed on two datasets to determine how well these models can handle different input distributions and complexities. The first dataset consists of carefully selected images provided by an instructor, allowing for a controlled examination of the models' inpainting accuracy. The second dataset is a 1K sampled subset of Tiny-ImageNet-2000, which is part of the larger ImageNet dataset [5]. This subset presents a more challenging real-world scenario with a wide variety of complex images.

In addition to comparing different model architectures, I am also investigating the scalability of the MLP by training a larger version of the model on the 1K-Tiny-ImageNet-2000 dataset. This assessment aims to understand how well the model architecture scales when dealing with a larger dataset. Furthermore, I am evaluating the original MLP on the full Tiny-ImageNet-2000 dataset to determine if increasing the amount of data leads to improved performance. This analysis provides valuable insights into the potential trade-offs between model complexity and accuracy.

To present a comprehensive view, the diffusion model, such as Stable Diffusion (SD) [6], has recently gained popularity in various computer vision applications, including text-to-image generation [6], image-to-image translation [7], and super-resolution [6]. Therefore, I have included an open-sourced SD model, specifically tuned for inpainting tasks[1]. This model, known for its impressive results in general image inpainting, is tested in the context of this study to see how well it predicts missing pixels. By comparing its performance with others, I am adopting the mean squared error (MSE) metric as my primary performance measurement, widely used for assessing the quality of pixel reconstruction in inpainting tasks.

The contributions of this research lie in offering valuable insights into the comparative performance of various model

[1]https://huggingface.co/runwayml/stable-diffusion-inpainting

architectures for inpainting tasks. My findings can serve as a guide for researchers and practitioners in selecting the most suitable model for this task, facilitating informed decisions for applications in computer vision and image processing. Additionally, this work opens up several future research directions, including the exploration of diverse datasets, the investigation of novel model architectures, and the integration of data augmentation and transfer learning techniques to further enhance the performance of inpainting models.

In the subsequent sections, I provide a detailed description of the methods in Section II, including model architectures and datasets. Next, in Section III, I present experimental results, followed by a comprehensive discussion and analysis of the findings. Finally, I conclude by summarizing the contributions of this research and outlining potential avenues for future investigations in the field of image inpainting in Section IV.

II. Methods

This section outlines the experimental methodology employed in conducting the comparative analysis of model architectures for missing pixel predictions. First, the details of the model architecture are discussed in Section II-A. Next, additional architectures explored are explained in Section II-B. Finally, the details of datasets and the training and evaluation process are discussed in Sections II-C and II-D, respectively.

A. Model Architecture

MLP is a traditional neural network architecture consisting of multiple layers of fully interconnected nodes. For the missing pixel prediction task, the input layer has 120 units with a rectified linear unit (ReLU) activation function, expecting an input shape of `(60)`, which allows the network to model complex non-linear relationships in the data and efficiently handle missing pixel prediction. One hidden layer is utilized, containing 60 units with ReLU activation functions, to further refine the extracted features. The output layer has four nodes with a Sigmoid activation function, as it is a regression task. I noted that this architecture choice was selected and provided by the instructor as a baseline model.

CNN is a deep learning architecture known for its effectiveness in computer vision tasks. The CNN used in this study consists of two convolutional layers, each followed by a max-pooling layer for downsampling. The first convolutional layer serves as an input layer, expecting an input image of shape `(8, 8, 1)` where the middle four pixels are removed. In this layer, 32 filters of size `(5, 5)` are utilized because they are capable of capturing larger patterns and edges. The ReLU activation function is used to introduce non-linearity, same as in MLP, and the option of `same` padding is chosen to retain the spatial dimensions of the input.

The next convolutional layer has 64 filters of size `(3, 3)` with the same ReLU activation function and default padding. The final max-pooling layer retains the same settings as the previous one. After the convolutional layers, the outputs are flattened to serve as inputs for an additional fully connected layer with 64 units and ReLU activation function. The output layer is similar to the MLP, containing four nodes with a Sigmoid activation function.

MobileNetV2 is a lightweight CNN architecture designed for mobile and embedded vision applications, aiming to reduce computation requirements. It employs depth-wise separable convolutions to achieve this reduction in computational complexity. For this study, I use the MobileNetV2 architecture with its default configuration, except for the last layer of the original output. The output layer now contains four nodes using a Sigmoid activation function to match the other architectures. The implementation follows an open-source implementation[2]. This architecture is chosen because it offers several advantages, including utilizing low computation resources due to the low number of parameters and operations. It is also well-tested and optimized for various vision tasks, making it a reliable choice for the study.

B. Additional Exploration

In addition to training the MLP, CNN, and MobileNetV2 on both the original dataset and the 1K-Tiny-ImageNet-2000 dataset, I also investigated the scalability of the MLP model. Specifically, I scaled the MLP to a larger architecture called MLP-L with the following configuration: 256 units in the input layer and three additional hidden layers with 128, 128, and 64 units, respectively, all using ReLU activation functions. The output layer retains the same configuration as the MLP. I then proceeded to train this larger MLP-L model on the 1K-Tiny-ImageNet-2000 dataset. Additionally, I explored the performance of the original MLP model when trained on the full Tiny-ImageNet-2000 dataset, without the 1K sample, aiming to observe any potential improvements with increased training data. Furthremore, I evaluate a pre-trained open-sourced SD model tuned for inpainting as a comparative baseline.

C. Datasets

Original Dataset is provided by an instructor and specifically curated for the missing pixel prediction task. It consists of a diverse set of grayscale images, totaling 55 samples. Out of these, one image is designated as the testing target for evaluation across all models. To ensure consistency, all images are resized to `(256, 256)` before being used in the experiments.

1K-Tiny-ImageNet-2000: Tiny-ImageNet-2000 is a subset of the ImageNet dataset originally designed as part of the Stanford CS231N class. For this study, I created a subset of 1,000 images by randomly selecting from Tiny-ImageNet-2000, encompassing training, validation, and testing images while preserving diversity and complexity. This was done to minimize the computation requirements and make training more manageable. To standardize the input data for the models, all images were pre-processed into grayscale before training. Additionally, it is worth noting that all images from the original Tiny-ImageNet-200 dataset, across training,

[2]https://github.com/Haikoitoh/paper-implementation

testing, and validation sets, were also utilized in one of the experiments. Each image in this subset has a shape of `(64, 64)`. This dataset was selected because ImageNet is a well-established dataset used for many standard evaluations [8].

D. Training and Evaluation

Each model architecture is trained separately on each dataset to compare their performance. The training process is performed using the Adam optimizer [9] with a fixed learning rate. The models are trained for 450 epochs with a batch size of 512 for MLP and 256 for the rest. MSE is selected as the loss function during training, and we assign 10% of the training dataset for validation. All inputs are chosen to have a patch size of 8×8, where the four middle pixels are removed (i.e., set to zero) or not supplied as a part of the input for MLP models. The training process also implements a function to save only the best model, i.e., the model with the lowest loss, and all evaluation processes are done using the saved version of the model.

For evaluation, the models are asked to predict the four missing pixels while all other pixels are automatically filled using the original image. The results from the model predictions are saved as an image file and later loaded to compare with the original image. The value of each predicted pixel is compared with the corresponding original pixel, and the sum of these values is the final metric used for comparison. The chosen evaluation image is a file named `balloon.bmp`, which is part of the original dataset but was exempted from the training. This provides a fair comparison and ensures no data leakage during the training process. For the SD model, there is no additional training or fine-tuning required. The evaluation process provides the model with the entire image as input while masking all pixels that are supposed to be missing. These masked pixels serve as targets for the model to predict.

III. Results and Discussions

In this section, I present the results of the evaluation conducted on five model architectures for missing pixel predictions: MLP, CNN, MobileNetV2, MLP-L, and SD. The performance evaluation of each model is detailed in Section III-A. Following the presentation of results, a comprehensive discussion of the findings and their implications will be provided in Section III-B. This analysis aims to investigate some reasons behind the obtained results, allowing for a deeper understanding of their performance in the context of missing pixel predictions.

A. Model Performance

I assessed every model on the same target image task, as described in the previous section. The results are MSE obtained from the predictions of each model. The lower the MSE, the better the result, indicating that it is closer to the ground truth image. Table I summarizes the results obtained from evaluating each model architecture on different training datasets, and Fig. 1 shows the original images and images obtained from model predictions. It demonstrates the superiority of CNN-based architectures over MLP architectures, while also indicating that scaling up (either architecture or data) worsens the performance for this task. Additionally, SD obtained the worst results across all of the models.

TABLE I: Model performance on missing pixels predictions using various architectures on different datasets.

Model	**Training Dataset**	**MSE↓**
MLP	Original	22.09
CNN	Original	**13.61**
MobileNetV2	Original	15.65
MLP	1K-TinyImageNet-2000	23.61
CNN	1K-TinyImageNet-2000	**13.90**
MobileNetV2	1K-TinyImageNet-2000	17.04
MLP-L	1K-TinyImageNet-2000	64.22
MLP	TinyImageNet-2000	446.78
SD	N/A	312.80

B. Discussions

The findings highlighted interesting observations on the performance of each model that may not depend on the size of the model but rather on the operations that the model performs to learn patterns of data for predictions. The comparative performance was discussed. Next, the scaling of MLP on architecture size and training data was analyzed. Finally, we gave some possible reasons for the poor performance of SD and suggest ways, in general, to investigate further to improve the models.

1) Comparative Performance of Model Architectures: The evaluation results demonstrated that CNN and MobileNetV2, which were CNN-based architectures, consistently outperformed the traditional MLP, regardless of training datasets. This outcome showcased the effectiveness of CNN-based architectures in capturing spatial patterns and hierarchies through convolution layers, which was particularly beneficial for this task.

Surprisingly, the MobileNetV2 architecture, known for its efficiency and performance in resource-constrained environments, did not outperform a simple CNN architecture in this context. This could be due to the fact that the data was not sufficient for the model to learn the patterns, both in terms of the number of samples and the size of the input. It is worth noting that the original MobileNetV2 is generally designed to work with images larger than 32×32, which is 4 times the size of our current inputs.

As for the traditional MLP architecture, it struggled to match the performance of CNN-based architectures, highlighting its limitations in handling spatial relationships effectively. Despite having a single hidden layer and being capable of modeling complex non-linear relationships in data, the MLP's performance lagged behind the CNNs. This suggested that for the missing pixels task, where spatial information was crucial, CNN-based architectures are more suitable.

(a) Original image

(b) A result image from the MLP model trained on the original dataset.

(c) A result image from the CNN model trained on the original dataset.

(d) A result image from the MobileNetV2 model trained on the original dataset.

(e) Original image

(f) A result image from the MLP model trained on the 1K-Tiny-ImageNet-2000 dataset.

(g) A result image from the CNN model trained on the 1K-Tiny-ImageNet-2000 dataset.

(h) A result image from the MobileNetV2 model trained on the 1K-Tiny-ImageNet-2000 dataset.

(i) Original image

(j) A result image from the MLP-L model trained on the 1K-Tiny-ImageNet-2000 dataset.

(k) A result image from the MLP model trained on full Tiny-ImageNet-2000 dataset.

(l) A result image from the Stable Diffusion Inpainting model.

Fig. 1: Comparison of prediction results from various models. *Top row*: Models trained on the original dataset. *Middle row*: Models trained on the 1K sampled subset of Tiny-ImageNet-2000. *Bottom row*: Results from further exploration of models.

It could also be observed that more training data, such as using the 1K-TinyImageNet-2000 as a training dataset, did not increase performance and, in fact, worsened the performance of the already trained models. It is also worth mentioning that all models did not suffer much loss in MSE. However, this was a surprising result as when we compared CNN and MobileNetV2, which is a much larger CNN-based architecture, we did not observe an increase in the performance of MobileNetV2.

The usual understanding is that with a larger architecture and a larger training dataset, the performance should increase in general. However, this was not the case, and it might be due to the fact that the input size with only 8×8 image patches might not provide sufficient information for the model to learn the patterns. On the other hand, it might be more beneficial to utilize a smaller architecture, which result in fewer computational requirements while outperforming in this specific task.

2) Scalability of MLP: To investigate whether the trend of using larger datasets together with larger architectures holds true, we developed MLP-L for comparison with the standard MLP in this task, similar to how CNN was compared with

MobileNetV2. Interestingly, the analysis of MLP's scalability revealed surprising results. When the MLP was scaled up to a larger architecture (MLP-L) and trained on the 1K-Tiny-ImageNet-2000 dataset, its performance significantly degraded. The larger MLP-L achieved a higher MSE compared to the original MLP, indicating that increasing the model's complexity did not lead to improved performance in this context.

Furthermore, training the original MLP on the full Tiny-ImageNet-2000 dataset also resulted in poor performance, with a significantly higher MSE than the MLP trained on the smaller original dataset. This suggests that increasing the training data does not necessarily lead to better results for the traditional MLP architecture in the image inpainting task, especially when the size of the architecture does not scale up to match the size of the dataset.

The observations regarding the scalability of MLP raise questions about the model's ability to effectively capture complex spatial patterns and relationships. It indicates that MLP's limitations may become more prominent when dealing with larger datasets and higher-dimensional data, making it less suitable for image inpainting tasks compared to CNN-based architectures.

3) Performance of SD Model: This study also included an evaluation of the SD model specifically tuned for inpainting tasks. The SD model showed the worst performance compared to the other architectures. This was unexpected, as SD has demonstrated impressive results in general image inpainting tasks and other applications.

The disappointing performance of the SD model in this specific context may indicate that its architecture or training procedure is not well-suited for the intricacies of this task, which involves a challenge in dealing with very limited inputs. It could also suggest that there might be a domain shift between the data used for tuning the SD model and the dataset used in this study, leading to suboptimal performance. Further investigation and experimentation would be necessary to understand the specific reasons behind the SD model's underperformance in this scenario.

IV. Conclusions and Future Work

In this study, I conducted a comparative analysis of model architectures for predicting missing pixels in images, a task known as image inpainting. We evaluated three distinct approaches: MLP, CNN, and MobileNetV2, using two datasets: an instructor-provided curated dataset and a 1K sampled subset of Tiny-ImageNet-2000. Additionally, we explored the scalability of the MLP by training a larger architecture, MLP-L, on the 1K-Tiny-ImageNet-2000 dataset and evaluated the original MLP on the full Tiny-ImageNet-2000. Furthermore, we evaluated the SD model specifically tuned for this task for a comprehensive comparison.

The results showed that CNN-based architectures, including CNN and MobileNetV2, consistently outperformed the traditional MLP when trained on both datasets. Surprisingly, the MLP-L, a scaled-up version of MLP, exhibited degraded performance compared to the original MLP, indicating that increasing the model's complexity did not lead to improved results in this task. Moreover, training the original MLP on the full Tiny-ImageNet-2000 dataset also resulted in poor performance, suggesting that simply increasing the training data may not yield better results for the traditional MLP. The performance of the SD model was unexpectedly poor in this specific context, indicating potential challenges in its architecture or training process for handling limited inputs effectively.

For future work, several directions can be explored to improve the performance of models. First, diversifying the datasets used for training and evaluation may help gain insights into the generalizability and robustness of the models across various scenarios. Second, investigating novel model architectures, particularly those focusing on capturing spatial relationships and handling limited input data, for example, Vision Transformers [10], could lead to more effective inpainting approaches and the potential for leveraging pre-trained models. Third, utilizing larger inputs could potentially enhance the model's performance. Moreover, it is essential to study the domain shift issue between the SD model's tuning data and the target dataset to understand the reasons behind its underperformance. By addressing these research directions, we can further advance in missing pixel prediction.

References

[1] Andreas Lugmayr, Martin Danelljan, Andres Romero, et al. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11461–11471, June 2022.

[2] Xiaobo Zhang, Donghai Zhai, Tianrui Li, et al. Image inpainting based on deep learning: A review. *Information Fusion*, 90:74–94, 2023.

[3] Jaya Gupta, Sunil Pathak, and Gireesh Kumar. Deep learning (cnn) and transfer learning: A review. *Journal of Physics: Conference Series*, 2273(1):012029, may 2022.

[4] Mark Sandler, Andrew Howard, Menglong Zhu, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[5] Jia Deng, Wei Dong, Richard Socher, et al. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, et al. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[7] Chitwan Saharia, William Chan, Huiwen Chang, et al. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022. Association for Computing Machinery.

[8] Olga Russakovsky, Jia Deng, Hao Su, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Appendix

Dataset, trained models, and code are available at https://bit.ly/ru-kbs-2023.