Deep Learning Project — Image Classification

Emmanuel Gardin, Joseph-Marie Ngueponwouo

April 19, 2025

Abstract

This project compares Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) for image classification. We evaluate their performance on two specifically designed datasets: a transformed MNIST dataset testing robustness to local changes and noise, and a novel Relational Dataset requiring long-range spatial reasoning. Our findings highlight the distinct strengths of each architecture, demonstrating task-dependent performance differences based on the need for local versus global feature processing.

Contents

1 Introduction

1.1 Background: CNNs vs. Vision Transformers

For years, Convolutional Neural Networks (CNNs) have been the dominant architecture for computer vision tasks, particularly image classification. Their success stems from their hierarchical structure, employing convolutional filters to efficiently learn local patterns (like edges and textures) and pooling layers to build spatial invariance (see Figure 2). This design incorporates strong inductive biases about the spatial nature of images.
Recently, inspired by the success of Transformer models in Natural Language Processing (NLP), the Vision Transformer (ViT) architecture was proposed as an alternative for image tasks [4]. ViTs treat an image as a sequence of flattened patches, adding positional embeddings and processing them through a standard Transformer encoder built upon the self-attention mechanism (see Figure 1). Unlike CNNs, ViTs have weaker built-in spatial biases, but their self-attention mechanism allows them, in principle, to model long-range dependencies between any two parts of an image directly.

1.2 Motivation and Objectives

The emergence of ViTs raises fundamental questions about the trade-offs between these two architectural paradigms. While ViTs offer potential advantages in modeling global context, they often require large datasets or extensive pre-training to match the performance of CNNs, which benefit from their inherent spatial priors.
The primary objective of this project is to conduct an empirical comparison between CNNs and ViTs specifically designed to highlight their differing strengths and weaknesses. We aim to move beyond standard benchmark comparisons by evaluating the models on tasks tailored to probe:

- Robustness to local image transformations and noise.
- Ability to capture long-range spatial relationships.

1.3 Experimental Approach

To achieve these objectives, we designed and utilized two distinct datasets, as detailed in Section 3 and implemented in `data/mnist_dataset.py` and `data/relational_dataset.py`:

1. **Modified MNIST Dataset:** This dataset introduces significant augmentations (rotations, translations, noise, inversions) to the standard MNIST digits. It serves to test the models' ability to handle local feature variations and noise, a scenario where CNNs' inductive biases are hypothesized to be advantageous.
2. **Relational Dataset:** A synthetic dataset where classification requires understanding the relative spatial positioning and orientation of multiple simple geometric shapes scattered across the image. This task is designed to explicitly challenge the models' capacity for long-range spatial reasoning, potentially favoring the global attention mechanism of ViTs. We further explore this by varying image resolution (64x64 vs. 128x128) and CNN depth.

We implemented standard CNN `models/cnn_model.py` and ViT `models/vit_model.py` architectures and trained them using a consistent pipeline `models/model_trainer.py`, comparing their performance using quantitative metrics and qualitative analysis, including ViT attention maps.

1.4 Research Questions

This study aims to provide insights into the following specific questions:

1. How do CNNs and ViTs compare in terms of classification performance and robustness when faced with significant local image transformations and noise, as presented in the Modified MNIST dataset?

2. Which architecture demonstrates superior performance on tasks explicitly requiring the modeling of long-range spatial relationships between image elements, as tested by the Relational Dataset?

3. How does the performance of CNNs on relational tasks depend on the relationship between their architectural depth (and thus receptive field size) and the input image resolution? How does this scalability compare to that of ViTs?

By addressing these questions, we aim to provide a clearer understanding of the conditions under which each architecture excels, guiding the choice of model for specific image classification challenges.

2 Methods: CNN and ViT Architectures

This section provides an overview of the two deep learning architectures compared in this study: Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs). We outline their core operating principles, key processing steps for image classification, and their generally recognized strengths and limitations relevant to this project's scope.

2.1 Vision Transformers (ViTs)

Originally developed for Natural Language Processing (NLP), the Transformer architecture was adapted for computer vision tasks in 2020 with the introduction of the Vision Transformer (ViT) [4]. ViTs represent a departure from the convolutional approach, instead leveraging the self-attention mechanism to model relationships between different parts of an image.
The standard ViT process, as originally proposed, involves the following steps for image classification (illustrated in Figure 1):

1. **Image Patching:** The input image (size $H \times W \times C$) is divided into a sequence of N non-overlapping flattened patches, each of size $P \times P \times C$. The sequence length is $N = (H \times W)/P^2$.

2. **Patch and Position Embedding:** Each flattened patch is linearly projected into an embedding vector of dimension D. Learnable position embeddings are added to these patch embeddings to retain spatial information. A special learnable classification (CLS) token embedding is prepended to the sequence.

3. **Transformer Encoder:** The sequence of embedded patches (including the CLS token) is processed by a standard Transformer encoder. This encoder typically consists of multiple blocks, each containing:
   - Multi-Head Self-Attention (MSA): Allows each patch embedding (and the CLS token) to interact with and aggregate information from all other embeddings in the sequence, capturing global context.
   - Feed-Forward Network (MLP): Applied independently to each position in the sequence.
   - Layer Normalization and Residual Connections: Applied before/after the MSA and MLP layers to stabilize training and facilitate gradient flow.

4. **Classification Head (using CLS token):** The state of the CLS token embedding at the output of the Transformer encoder, which has aggregated information from the entire image via self-attention, is passed through a final classification head (typically a simple MLP) to produce the class probabilities.

Note on Implementation: While the use of a dedicated CLS token is common, the ViT architecture implemented in this project (`models/vit_model.py`) employs a slightly different but functionally equivalent approach for feature aggregation. Instead of relying on a CLS token, our model processes the sequence of patch embeddings through the Transformer encoder and then applies **Global Average Pooling** across the output sequence dimension. This average vector, representing the aggregated features from all patch locations, is then fed into the final classification head. This alternative is also widely used in Transformer-based models and was chosen for its simplicity and effectiveness in summarizing the sequence features for classification without adding an extra learnable token. Both methods aim to produce a single vector representation of the image features for the final classifier.

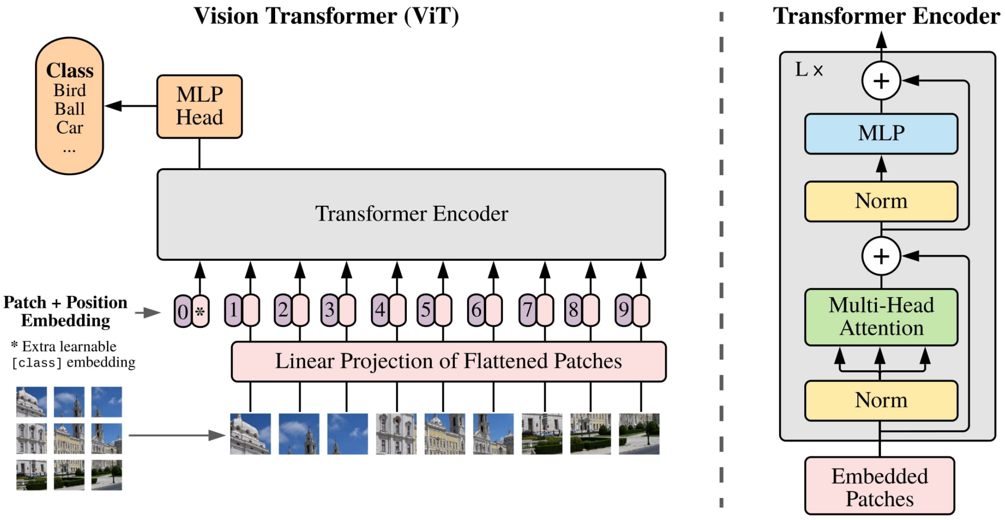


Figure 1: High-level architecture of a Vision Transformer (ViT) for image classification, illustrating the patching, embedding, and Transformer encoder stages. The original paper used a CLS token for classification (as described in step 4), while our implementation uses Global Average Pooling after the encoder.

2.1.1 Strengths and Limitations of ViTs

- **Strengths:**
  - *Global Context Modeling:* The self-attention mechanism enables ViTs to capture long-range dependencies across the entire image, potentially beneficial for tasks requiring understanding of global structure or relationships between distant objects.
  - *Scalability:* Performance can scale well with model and dataset size, although often requiring significant computational resources and data.
- **Limitations:**
  - *Weak Inductive Bias:* Compared to CNNs, ViTs have weaker built-in assumptions about image structure (like locality). This can make them less data-efficient, often requiring large datasets or extensive pre-training to generalize well, especially on smaller datasets.
  - *Local Feature Detail:* May struggle to capture fine-grained local textures or details as effectively as CNNs without sufficient depth or data.
  - *Computational Cost:* The self-attention mechanism has quadratic complexity with respect to the number of patches (sequence length), which can be computationally expensive for high-resolution images.

2.2 Convolutional Neural Networks (CNNs)

CNNs have been the cornerstone of modern computer vision since the breakthrough performance of AlexNet [2] in 2012. They are designed with strong spatial inductive biases, leveraging hierarchical layers of convolutions and pooling to learn representations of visual data.
A typical CNN architecture for image classification includes these primary stages (illustrated in Figure 2):

1. **Convolutional Layers:** Apply learnable filters (kernels) across the input image (or feature maps from previous layers). Each filter detects specific local patterns (e.g., edges, corners, textures). Stacking these layers allows the network to learn increasingly complex features hierarchically.
2. **Activation Functions:** Introduce non-linearity (e.g., ReLU) after convolutional layers, enabling the network to model complex relationships.
3. **Pooling Layers:** Downsample the feature maps (e.g., Max Pooling), reducing spatial dimensions while retaining the most salient information. This provides a degree of translation invariance and reduces computational cost.
4. **Flattening:** The final feature maps are flattened into a one-dimensional vector.
5. **Fully Connected (FC) Layers:** One or more dense layers process the flattened vector, integrating features learned across different spatial locations to make a final prediction.
6. **Classification Layer:** The final layer (often using a Softmax activation) outputs probabilities for each class.

Common architectural variations include ResNet, VGGNet, InceptionNet, and DenseNet, which introduce specific block structures, skip connections, or connectivity patterns to improve training and performance.

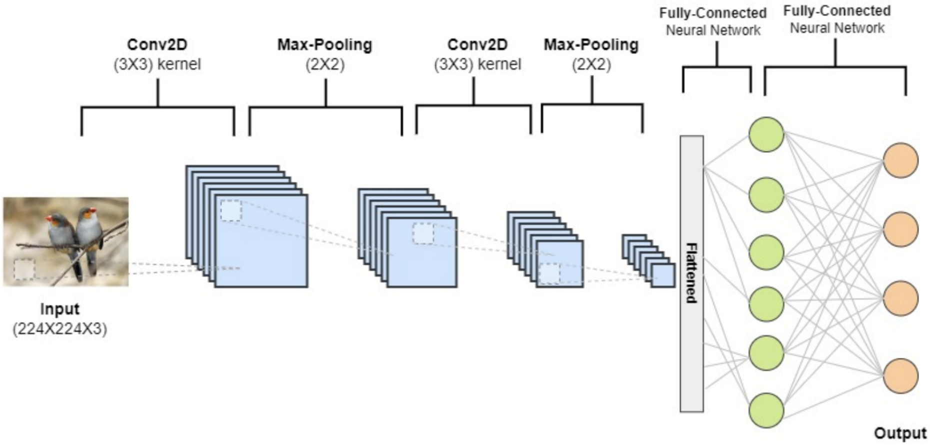


Figure 2: Example architecture of a Convolutional Neural Network (CNN) for image classification.

2.2.1 Strengths and Limitations of CNNs

- **Strengths:**
  - *Strong Inductive Biases:* Built-in assumptions about locality (pixels nearby are related) and translation equivariance (patterns can appear anywhere) make CNNs highly effective at learning spatial hierarchies and local patterns, often with greater data efficiency than ViTs.
  - *Local Feature Extraction:* Excel at capturing fine-grained textures and local details due to the nature of convolutional filters.
  - *Parameter Efficiency (Relative):* Often achieve strong performance with fewer parameters than ViTs, especially on smaller datasets, due to weight sharing in convolutional layers.
- **Limitations:**

- *Limited Global Context:* Standard CNNs struggle to model explicit long-range dependencies due to the local nature of convolutional operations. The effective receptive field grows with depth, but direct long-range interactions are not inherent.
- *Sensitivity to Input Size:* Traditional CNN architectures often require fixed-size inputs, although techniques like adaptive pooling can mitigate this.
- *Handling Rotations/Scale Variations:* While pooling provides some translation invariance, CNNs are not inherently equivariant to significant rotations or scale changes without data augmentation or specialized architectures.

3 Dataset Design for Comparative Analysis

To empirically evaluate the distinct characteristics and performance trade-offs between Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), we designed and utilized two specific datasets. Each dataset targets a different aspect of image understanding, allowing us to probe the hypothesized strengths and weaknesses of each architecture based on the task requirements. The following subsections detail the rationale and construction of these datasets.

3.1 Dataset 1: Modified MNIST for Robustness Testing

Our first experiment utilizes the well-known MNIST dataset [5] of handwritten digits (0-9), sourced via `tensorflow.keras.datasets`. MNIST is a standard benchmark where CNNs typically perform exceptionally well due to their ability to recognize local shapes and patterns.

However, our goal here is not simply benchmark performance, but rather to assess the models' **robustness to local transformations and noise**. We hypothesize that CNNs, owing to their strong inductive biases (locality, translation equivariance via pooling), will maintain high performance even when the input images are significantly altered, whereas ViTs, lacking these biases, might be more sensitive to such local perturbations.

To create this test scenario, we applied a series of augmentations to the standard MNIST images using the `MNISTDataset` class defined in `data/mnist_dataset.py`. These transformations include:

- Random rotations
- Random translations (shifting)
- Addition of Gaussian noise
- Random pixel intensity inversions

The standard MNIST split of 60,000 training images and 10,000 testing images was used. The classification task remains identifying the digit (0-9) depicted in the transformed image. Figure 3 provides examples of images from this modified dataset, illustrating the types of transformations applied. This setup allows us to directly compare how well each architecture generalizes when faced with common visual variations that distort local features.

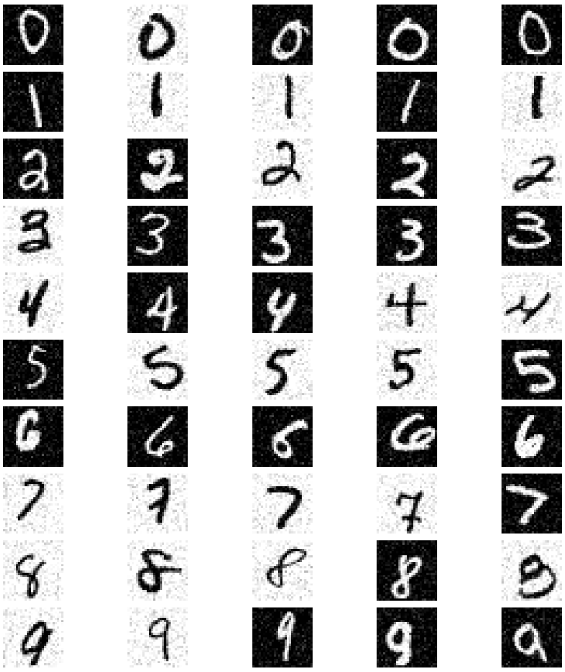


Figure 3: Example images from Dataset 1: The Modified MNIST dataset, showing digits subjected to rotation, translation, noise, and inversion.

3.2 Second Dataset: Relational Reasoning Task

To specifically probe the models' ability to understand long-range spatial relationships—a task hypothesized to favor the global attention mechanism of Vision Transformers (ViTs) over the local receptive fields of Convolutional Neural Networks (CNNs)—we designed a synthetic relational dataset. The core idea, as detailed in the project notebook and implemented in `data/relational_dataset.py`, is that classifying an image requires understanding the relative positioning and orientation of multiple distinct shapes scattered across the canvas, rather than just recognizing local textures or isolated features.

Dataset Rationale: Standard CNNs build hierarchical representations based on local patterns. While effective for texture and local shape recognition (as tested with the modified MNIST dataset), their fixed receptive fields may struggle to capture dependencies between distant elements in an image without becoming excessively deep. ViTs, conversely, use self-attention, allowing any image patch (token) to directly attend to any other patch. This mechanism makes them theoretically better suited for tasks requiring global spatial reasoning. This dataset was constructed to create a scenario where this theoretical advantage of ViTs could be empirically tested against CNNs of varying depths.

Dataset Construction: The dataset generation process, implemented in the `RelationalDataset` class within `data/relational_dataset.py`, involves the following steps for each image:

1. **Initialization:** A blank grayscale image canvas (e.g., 64x64 or 128x128 pixels) is created.
2. **Shape Generation:** Four distinct geometric shapes are drawn onto the canvas:
   - A randomly rotated square (`_draw_square`). This serves as the primary reference shape.
   - A line segment that spans across the image, intersecting the boundaries. Its angle is determined relative to the square based on the target class rules (implemented in `_create_image`).

- A cross shape, potentially with varying arm lengths (`_draw_cross_varying_length`). Its orientation (parallel or 45°) relative to the line segment is determined by the target class.
- A rectangle (`_draw_rectangle`). Its orientation (long side parallel or perpendicular) relative to the line segment is determined by the target class.

3. **Placement:** The square, cross, and rectangle are placed at random positions on the canvas, ensuring a minimum distance between their centers to avoid excessive overlap. The line's position is defined by its calculated angle and a random point it must pass through.
4. **Class Definition:** There are 8 distinct classes (0-7), defined by a combination of three binary rules governing the relative orientations between the shapes. These rules force the model to consider relationships across potentially large distances within the image. The specific rules for each class are summarized in Table 1.
5. **Output:** The process yields the generated images and their corresponding class labels (0-7), forming the training and testing sets.

Table 1: Summary of Class Definitions for the Relational Dataset

Rule	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Class 6**	**Class 7**
Line vs. Square	Side $\|\|$	Side $\|\|$	Side $\|\|$	Side $\|\|$	Diag $\|\|$	Diag $\|\|$	Diag $\|\|$	Diag $\|\|$
Cross vs. Line	$\|\|$	45°	$\|\|$	45°	$\|\|$	45°	$\|\|$	45°
Rectangle vs. Line	$\|\|$	$\|\|$	$\perp$	$\perp$	$\|\|$	$\|\|$	$\perp$	$\perp$

Note: $||$ denotes parallel, $\perp$ denotes perpendicular, Side $||$ means the line is parallel to a side of the square, Diag $||$ means the line is parallel to a diagonal of the square, 45° *means the cross is rotated 45 degrees relative to the line.*

Figure 4 shows example images from each of the 8 classes, illustrating the relational properties the models need to learn. This setup forces the models to integrate information from across the entire image to correctly classify the sample based on the relative configurations of the shapes, directly testing their ability to handle long-range spatial dependencies.

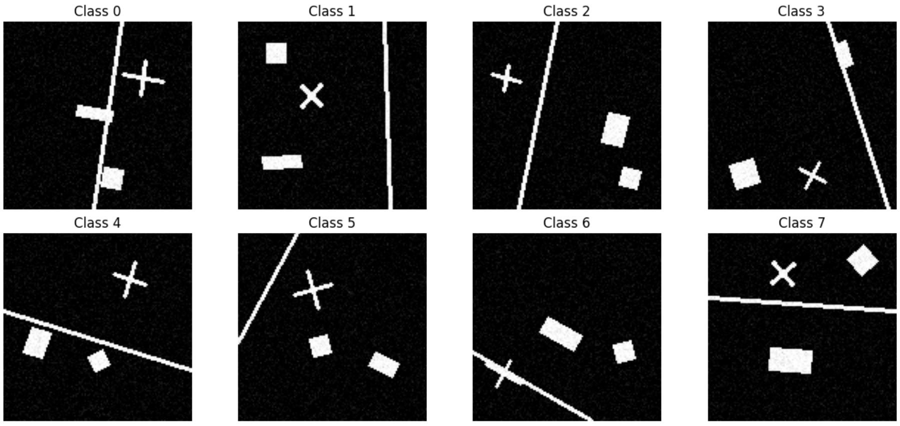


Figure 4: Sample images from the 8 classes of the Relational Dataset. Each class is defined by the relative orientations of the line, cross, and rectangle with respect to the square and each other, as detailed in Table 1.

4 Results and Analysis

4.1 Model Configurations and Hyperparameters

Before presenting the performance results, this section summarizes the key hyperparameters used for the Convolutional Neural Network (CNN) and Vision Transformer (ViT) models across the different datasets and experiments conducted in this study. These configurations were chosen based on common practices observed during lectures and in the literature, refined through hyperparameter tuning specific to each dataset and model, aiming for a fair comparison between the architectures.

Table 2: CNN Model Hyperparameters

Hyperparameter	**MNIST (3L)**	**Relational 64x64 (3L / 4L / 5L)**	**Relational 128x128 (3L / 4L / 5L)**
Input Channels	1	1 / 1 / 1	1 / 1 / 1
Image Size	28x28	64x64	128x128
Num Classes	10	8 / 8 / 8	8 / 8 / 8
Num Layers	3	3 / 4 / 5	3 / 4 / 5
Kernel Size	3	3 / 3 / 3	3 / 3 / 3
First Filters	16	16 / 16 / 16	16 / 16 / 16
Filters Multiplier	2	2 / 2 / 2	2 / 2 / 2
Dense Neurons	64	64 / 64 / 64	64 / 64 / 64
Dropout Rate	0.2	0.1 / 0.1 / 0.1	0.1 / 0.1 / 0.1
Optimizer	AdamW	AdamW	AdamW
Learning Rate	0.001	0.001	0.001
Weight Decay	1e-4	1e-4	1e-4

Note: For Relational datasets, values are shown for 3, 4, and 5 layer variants respectively where applicable.

Table 3: Vision Transformer (ViT) Model Hyperparameters

Hyperparameter	**MNIST**	**Relational 64x64**	**Relational 128x128**
Input Channels	1	1	1
Image Size	28x28	64x64	128x128
Num Classes	10	8	8
Patch Size	7	7	7
Embedding Dim	64	64	64
Num Heads	4	4	4
Num Blocks	2	2	2
Dropout Rate	0.2	0.2	0.1
Optimizer	AdamW	AdamW	AdamW
Learning Rate	5e-4	5e-4	1e-3
Weight Decay	1e-4	1e-4	1e-3

4.2 Results on the First Dataset: Robustness Testing on MNIST

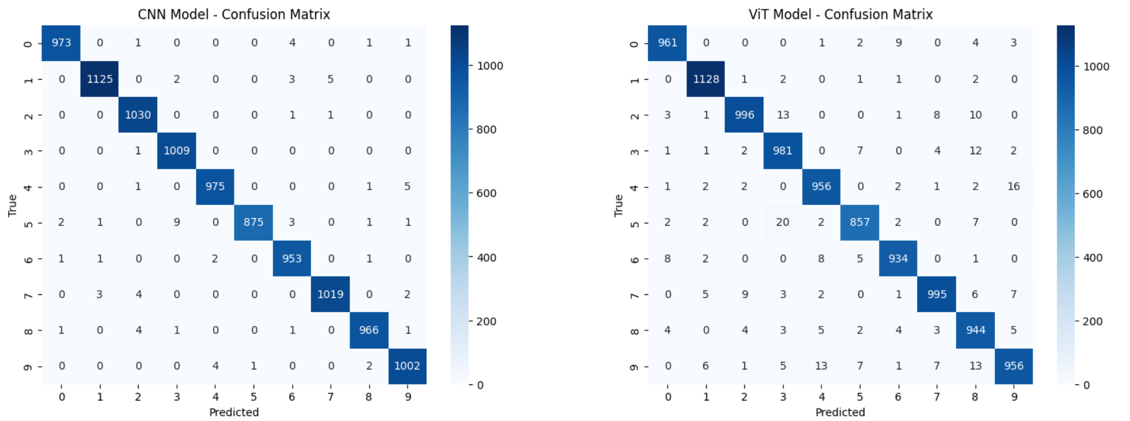
This section presents the performance comparison between the CNN and ViT models on the MNIST dataset, both in its original form and after applying the transformations described in Section 3.1.

4.2.1 Baseline Performance on Original MNIST

First, to establish a baseline, both models were trained and evaluated on the standard, untransformed MNIST dataset. As expected for this task, both architectures achieved high accuracy, confirming their basic capability for digit recognition.

- CNN Model - Validation Accuracy: 99.27%
- ViT Model - Validation Accuracy: 97.08%

The CNN slightly outperformed the ViT on the original dataset, achieving near-perfect accuracy. The ViT also performed well, demonstrating its effectiveness even without large-scale pre-training



(a) CNN - Confusion Matrix

(b) ViT - Confusion Matrix

Figure 5: Confusion Matrices on the original MNIST test set.

on this relatively simple dataset. These results serve as a reference point for evaluating the impact of the introduced transformations.

4.2.2 Performance on Modified MNIST

Next, the models were trained and evaluated on the modified MNIST dataset, which included rotations, translations, noise, and inversions. The goal was to assess robustness to these common local perturbations.

- CNN Model - Validation Accuracy: 97.80%
- ViT Model - Validation Accuracy: 93.71%

As shown by the validation accuracies, both models experienced a performance drop compared to the baseline. However, the impact was significantly more pronounced for the ViT.

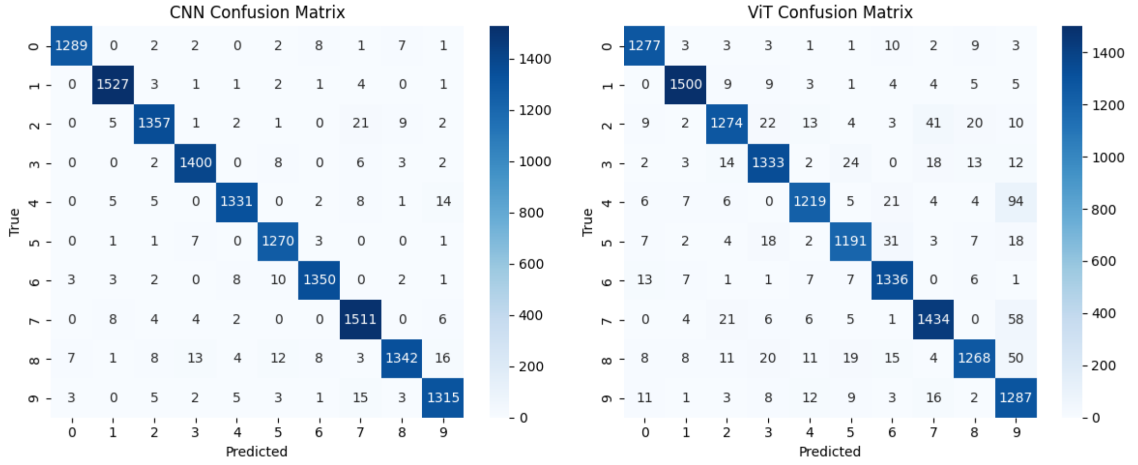


Figure 6: Confusion Matrices on the modified MNIST test set.

The confusion matrices in Figure 6 visually confirm the CNN's superior performance under these challenging conditions, exhibiting fewer off-diagonal entries compared to the ViT.
This performance difference aligns with the architectural characteristics of the models:

- **CNNs:** Their convolutional filters inherently focus on local patterns. Combined with pooling layers that provide some translation invariance, this architecture is well-suited to identifying key digit features even amidst noise and moderate geometric transformations. The strong spatial inductive biases help maintain robustness.

- **ViTs:** Lacking strong built-in spatial biases, ViTs may be more sensitive to local noise and distortions that disrupt the patch representations or the learned relationships between them. While self-attention can model global context, it might struggle to disregard local noise as effectively as CNNs without specific training or pre-training strategies.

4.2.3 Training Dynamics on Modified MNIST

The training history, plotted in Figure 7, provides further insights.

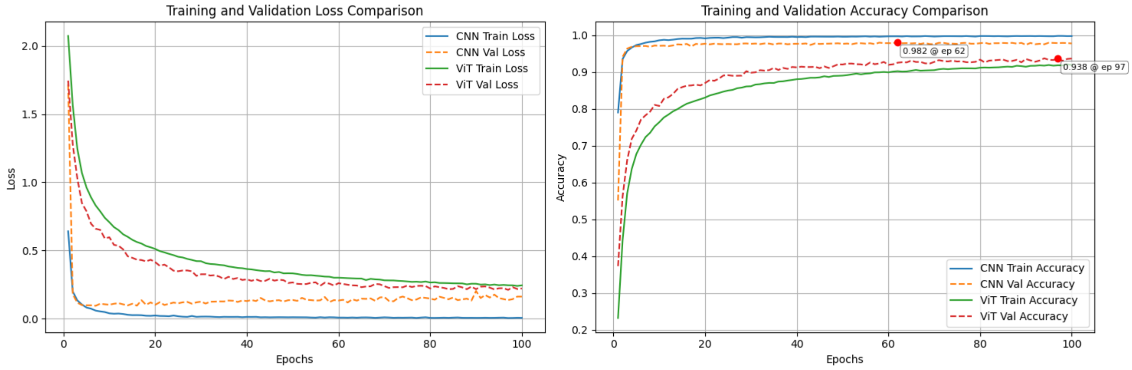


Figure 7: Training and Validation Metrics (Loss and Accuracy) on the modified MNIST dataset.

We observe that the CNN not only achieves higher final validation accuracy but also converges significantly faster. Its validation accuracy starts higher and reaches its plateau in fewer epochs compared to the ViT. The ViT exhibits a slower learning curve, particularly in the initial epochs, consistent with the notion that Transformers often require more training data or time to learn effectively from scratch compared to CNNs, especially on datasets where local features are crucial. The gap between training and validation accuracy also appears slightly larger for the ViT, suggesting it might be marginally more prone to overfitting on this specific task setup, although both models generalize reasonably well.

4.3 Results on the Second Dataset: Relational Reasoning

The second dataset was specifically designed to test the models' ability to capture long-range spatial dependencies, a task hypothesized to favor the global attention mechanism of ViTs over the local receptive fields of CNNs. We conducted experiments on two image resolutions (64x64 and 128x128) and varied the depth of the CNN.

4.3.1 Performance on 64x64 Images

We first trained and evaluated a standard 3-layer CNN and a ViT on the 64x64 version of the Relational Dataset. The hyperparameters for both models were kept consistent with those used for the MNIST experiment to allow for a direct comparison of architectural suitability.
Training Dynamics:

As shown in Figure 8, the training dynamics revealed a stark contrast between the two architectures.

- **CNN (3 Layers):** The CNN exhibited significant overfitting. While training accuracy increased, validation accuracy quickly plateaued around 13% (slightly above random chance for 8 classes), and validation loss started increasing. This indicates a failure to generalize, likely because its limited receptive field (calculated as 15x15 for 3 layers) was insufficient to capture the necessary long-range relationships defining the classes. The model appeared to memorize local patterns in the training data.

- **ViT:** The ViT demonstrated much better generalization. Both training and validation accuracy consistently improved, reaching approximately 62% validation accuracy. Although a gap between training and validation curves suggests some overfitting (which could potentially be addressed with stronger regularization or more data), the ViT clearly learned the underlying relational rules far more effectively than the shallow CNN.

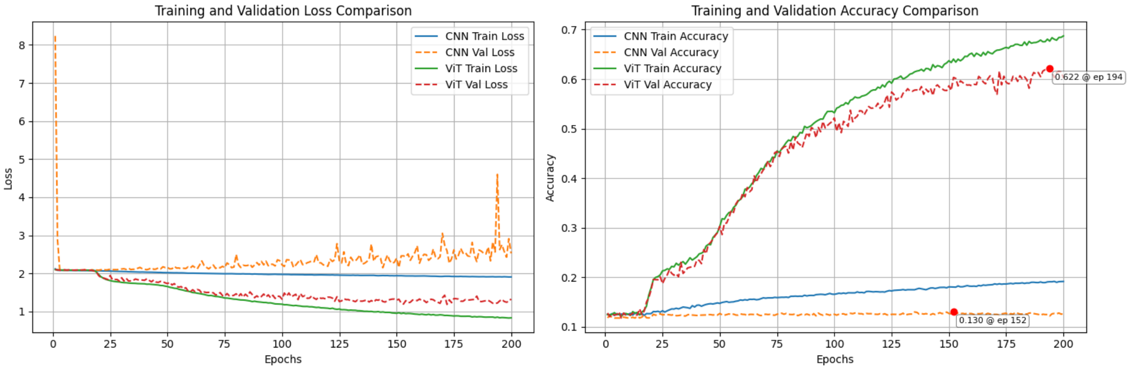


Figure 8: Comparison of Training History (Loss and Accuracy) for CNN (3 layers) and ViT on the 64x64 Relational Dataset.

Evaluation Metrics: The evaluation metrics confirmed the ViT's superiority on this task (Figure 9).

- **Confusion Matrices:** The CNN's confusion matrix showed poor performance, with a strong bias towards predicting a specific class (Class 2). A possible explanation for this behavior could be that during the generation, class 2 had statistically more images with shapes close to each other, therefore biasing the CNN which only has a local receptive field. The ViT's matrix displayed a much stronger diagonal, indicating more accurate and balanced classification across classes.

- **Classification Reports:** The CNN's report reflected low and uneven precision/recall, particularly poor for classes other than the biased prediction. The ViT's report showed significantly higher and more balanced metrics across all 8 classes.

- **ViT Attention Maps:** Visualizing the ViT's attention maps (Figure 10) revealed that the model learned to focus on the regions where shapes intersected or interacted closely. This aligns with the task's requirement to understand relative orientations and positions, explaining the ViT's success in capturing the necessary long-range dependencies.

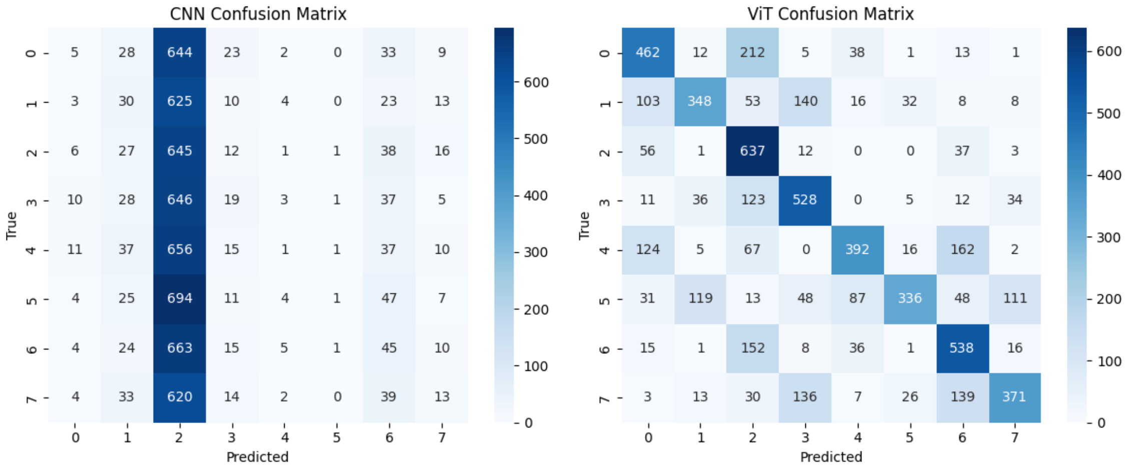


Figure 9: Confusion Matrices for CNN (3 layers) and ViT on the 64x64 Relational Dataset Test Set.

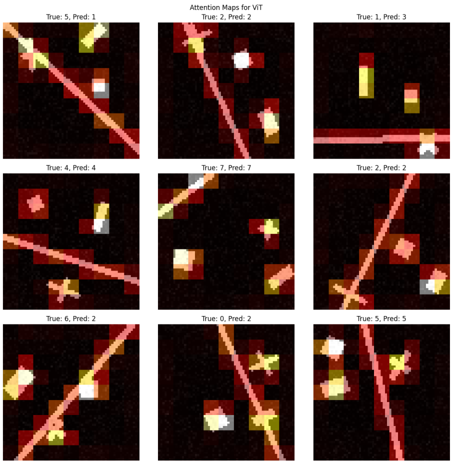


Figure 10: Example ViT Attention Maps on the 64x64 Relational Dataset. High attention (yellow/white) is focused on shape interaction regions.

4.3.2 Impact of CNN Depth on 64x64 Images

Given the 3-layer CNN's failure, we investigated whether increasing its depth, and thus its theoretical receptive field (RF), could improve performance on the 64x64 dataset. We trained CNNs with 4 layers (RF: 31x31) and 5 layers (RF: 63x63, covering almost the entire image).

Results: (See Figure 11)

- **4-Layer CNN:** Showed significant improvement over the 3-layer CNN. Its larger receptive field allowed it to capture more spatial context, leading to better validation accuracy and loss.
- **5-Layer CNN:** Despite having a receptive field covering nearly the entire image, its performance was slightly worse than the 4-layer CNN. It exhibited signs of overfitting, with a gap between training and validation curves. This suggests that while a larger RF is beneficial, simply increasing depth can lead to overfitting on this dataset size/complexity, potentially due to increased model capacity or loss of detail from excessive pooling. A solution to this could be additionnal regularization of the model.

This experiment indicates that while increasing the CNN's receptive field is necessary for this task, simply making the network deeper isn't a guaranteed solution and can introduce other problems like overfitting, especially compared to the ViT's inherent global attention. The 4-layer CNN provided the best balance for this specific resolution.

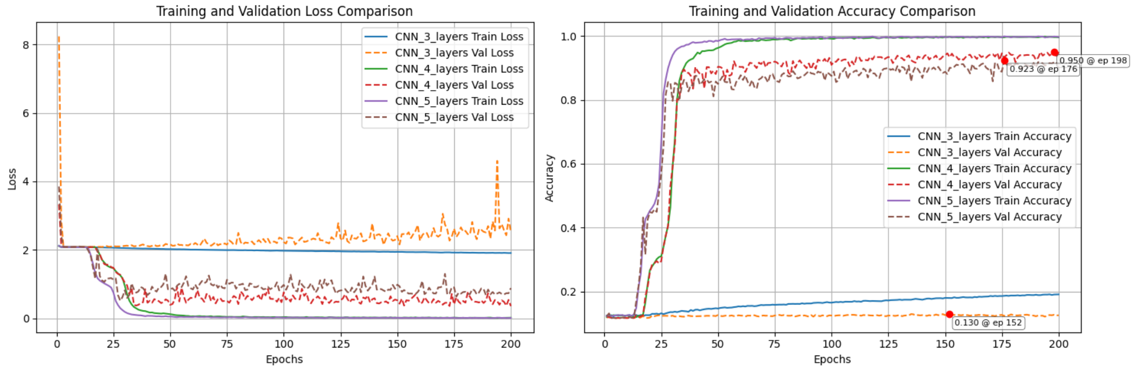


Figure 11: Comparison of Training History for CNNs with 3, 4, and 5 Layers on the 64x64 Relational Dataset.

4.3.3 Performance on 128x128 Images: Scaling Effects

To further test the architectures' ability to handle long-range dependencies and scaling, we repeated the comparison using larger 128x128 images. This setup challenges the models by increasing the distances over which relationships must be captured.

Results: (See Figure 12)

- **3-Layer CNN (RF 15x15):** Performance remained poor, near random chance, confirming its inadequacy as the relative RF size decreased further.
- **4-Layer CNN (RF 31x31):** This model, which performed well on 64x64 images, failed dramatically on 128x128 images. Validation accuracy dropped to 13%. Its fixed receptive field, now covering a much smaller fraction of the input, became insufficient to capture the required global context.
- **5-Layer CNN (RF 63x63):** In contrast to its overfitting on 64x64, the 5-layer CNN performed very well on 128x128 images, achieving high validation accuracy (93.5%). Its receptive field was now appropriately sized relative to the larger input, allowing it to capture the necessary long-range dependencies.
- **ViT:** The ViT maintained strong performance on the 128x128 dataset, demonstrating robust scalability. Its global attention mechanism inherently adapts to the larger input size without requiring architectural changes like increasing depth.

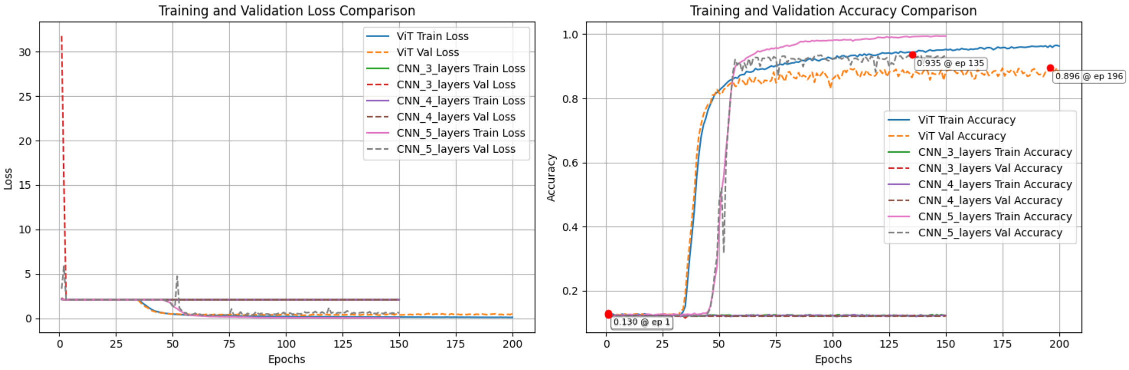


Figure 12: Comparison of Training History for ViT and CNNs (3, 4, 5 Layers) on the 128x128 Relational Dataset.

Analysis Summary: The experiments on the Relational Dataset strongly support the hypothesis that ViTs are better suited for tasks requiring long-range spatial reasoning.

- Standard CNN performance is highly dependent on the receptive field size relative to the input image size. A fixed-depth CNN that works well at one resolution can fail completely at a higher resolution if its RF becomes too small relatively.
- Increasing CNN depth can compensate for larger image sizes but requires careful tuning to avoid overfitting and comes at the cost of increased complexity.
- ViTs demonstrate superior scalability for this type of task due to their global self-attention mechanism, which is less constrained by input resolution compared to the fixed receptive fields of CNNs.

5 Conclusion

This project conducted a comparative study of Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) on image classification tasks, utilizing two distinct datasets designed to highlight their architectural differences.
On the modified MNIST dataset, characterized by local transformations and noise, the CNN outperformed the ViT, achieving higher accuracy and faster convergence. This aligns with the hypothesis that CNNs' inherent inductive biases, such as locality and translation equivariance, provide an advantage for tasks dominated by local patterns and textures, especially when robustness to common augmentations is required. The ViT, while capable, appeared more sensitive to noise and required longer training, consistent with its known reliance on larger datasets or pre-training to compensate for weaker inductive biases.
Conversely, on the custom Relational Dataset designed to test long-range spatial reasoning, the ViT demonstrated a clear advantage. It significantly outperformed shallow CNNs, successfully capturing the global relationships between shapes necessary for classification, as evidenced by its performance and attention maps focusing on interaction points. While increasing CNN depth improved its performance by enlarging the receptive field relative to the image size, this came with challenges:

- Performance was highly sensitive to the match between receptive field size and input resolution. A CNN effective at 64x64 failed at 128x128, while a deeper CNN succeeded at 128x128 but overfit at 64x64.
- Achieving adequate performance required manually adjusting depth based on input size.

The ViT, leveraging its self-attention mechanism, scaled robustly with image resolution without requiring architectural changes, maintaining high performance on both 64x64 and 128x128 relational tasks.

In conclusion, neither architecture is universally superior. CNNs excel when strong local inductive biases are beneficial, such as in tasks involving textures, local patterns, and moderate transformations. ViTs demonstrate strength in tasks requiring global context and long-range dependency modeling, showcasing better scalability with input resolution for such problems due to their attention mechanism. The choice between CNNs and ViTs should therefore be guided by the specific characteristics of the target task and dataset.

6 References

References

[1] Sonia B. et al. (2024). Convolutional Neural Networks and Vision Transformers for Fashion MNIST Classification: A Literature Review. [1]

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

[3] Zhiying Lu, Hongtao Xie, Chuanbin Liu, and Yongdong Zhang. Bridging the gap between vision transformers and convolutional neural networks on small datasets. Advances in Neural Information Processing Systems, 35:14663–14677, 2022.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*, 2020. Available at: https://arxiv.org/abs/2010.11929.

[5] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.