Chessboard Image Analysis for Game State Extraction

Ethan McKeen
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, B.C., Canada
ethanrmckeen@gmail.com

Abstract

This project explores the use of Vision Transformers (ViTs) for classifying individual chess pieces from images of chessboard squares. Unlike traditional approaches that rely on convolutional neural networks (CNNs) to directly predict board states, this method introduces a preprocessing pipeline that segments full-board images into square-level inputs, enabling fine-grained classification. The model is trained on a diverse dataset of synthetically generated chess positions featuring various board and piece styles. Results show that the ViT architecture achieves high classification accuracy while requiring less training data and time, thanks to its superior generalization capabilities. Additionally, attention maps offer valuable interpretability into the model's decision-making process.

1 Introduction

Chess is a game of strategy that has been extensively studied in machine Learning. Using powerful chess engines such as Stockfish and AlphaZero, players can receive optimal move suggestions[8]. These engines, however, require an accurate representation of the board state to function effectively. While online chess platforms like Chess.com and Lichess provide built-in engine analysis, users may wish to use external engines for independent analysis, training, or research purposes [2][5].

The challenge here is extracting the board state externally. This can be achieved by taking an image of a digital chessboard and automating the process required to visually recognize each chess piece and its position. Variations in board themes, piece styles, and colours further complicate this task. The goal of this project is to develop a computer vision system that accurately detects the positions of all chess pieces from an image of a digital chessboard and translates this information into a structured format like Forsyth-Edwards Notation (FEN). This tool can assist players in game review, training, and AI-assisted strategy development by enabling real-time external analysis.

2 Related Work

2.1 Traditional Computer Vision Approaches

Early attempts at chessboard recognition relied on classical image processing techniques, such as edge detection, Hough transforms, and contour analysis, to identify board boundaries and individual squares. OpenCV-based approaches, including template matching and color segmentation, have been used to classify chess pieces [1]. While these methods perform reasonably well in controlled environments, they often struggle with variations in board themes, lighting conditions, and perspective distortions.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2.2 Deep Learning-Based Methods

Deep learning, particularly Convolutional Neural Networks (CNNs), has significantly improved object detection and classification. Models like YOLO (You Only Look Once) and Faster R-CNN enable real-time piece recognition with high accuracy [3] [7]. These architectures allow end-to-end learning of chess piece positions from labeled datasets, reducing the need for handcrafted features [10][6]. However, training these models requires extensive annotated data and computational resources.

3 Methodology

Most existing approaches rely on CNNs to directly map input images to FEN representations. While effective, this method typically requires a large amount of training data and significant training time to achieve high accuracy.

In contrast, this project leverages a Vision Transformer (ViT) architecture, made feasible through a carefully designed preprocessing pipeline. The preprocessing reduces the complexity of the input data, enabling the use of a more sophisticated model without the need for extensive training resources. The transformer model demonstrates similar performance to other traditional CNN-based approaches, while requiring less data and training time due to its improved generalization capabilities. Furthermore, the attention mechanisms inherent in transformer models allow for the visualization of attention heatmaps which can further improve the interpretability of the model's predictions. A transformer architecture was also chosen as it is a discriminative model and, considering the classification task, a lightweight model that can focus directly on modeling the decision boundary between classes is beneficial. Also complicated inference like sampling is not required in this problem context.

3.1 Dataset

The dataset used in this project is provided by Pavel Koryakin and consists of 100,000 images of randomly generated chess positions [4]. It features 5 to 15 pieces, including two kings and a variable number of pawns and other pieces. The dataset includes: 28 distinct chessboard styles and 32 unique piece styles, resulting in 896 possible board-piece combinations, image dimensions of 400x400 pixels, a structured split into training (80,000 images) and test (20,000 images) sets, a probabilistic distribution of pieces: 30% Pawns, 20% Bishops, 20% Knights, 20% Rooks, 10% Queens, and two kings (one per color) always present.

Some example chessboard images are shown below in figure 1 to demonstrate the amount of variance in the chessboard training set. The FEN notation for each image is "1b1b2k1-K2B1q2-R3B2p-3b1NR1-5p2-3N4-8-5N2", "1B1bq2B-1P2R2B-K7-6B1-2N5-5r2-N7-5Rk1", and "1b1n4-3q4-6n1-8-5K2-3Q4-8-1r5k" for the images from left to right respectively.

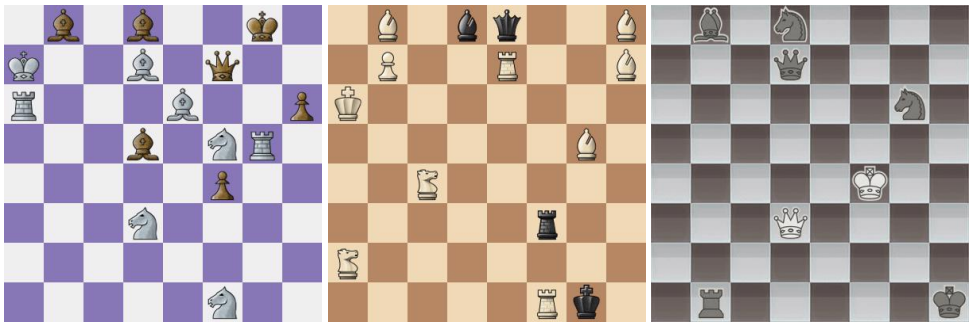


Figure 1: Example Training Set Chessboard Images

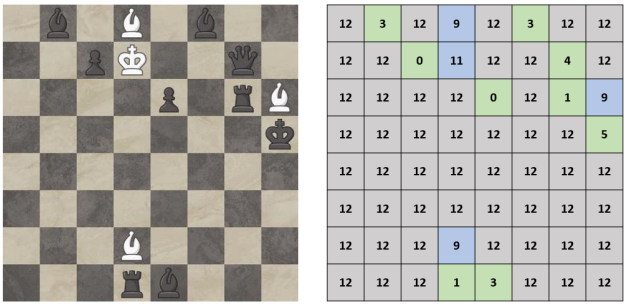
3.2 Data Preprocessing

This model has an extensive preprocessing step. Each 400x400 pixel chess board image is divided into 64 individual 50x50 pixel images representing each square of the chess board. This turns the

problem into multiple smaller classification problems as the pieces on each square are classified independently. This simplification sacrifices some spacial context like the position of the piece on the board. This information could be re-introduced using positional encodings which, in some contexts, may be useful. For example, in many board states it might be more common for white pieces to be surrounded by white pieces and black pieces to be surrounded by black pieces. However, this information was not taken into consideration in this methodology as it may hinder generalizeability and avoid overfitting to common board states.

This problem redefinition also means there is a required conversion step where the FEN representation of the board is first translated into an 8x8x13 tensor and, after classification, the model's outputs are converted back into a standard FEN string for downstream use. A chessboard example and its corresponding 8x8x13 tensor can be seen in figure 2. This tensor represents the 8x8 grid of a chess board and the 13 different states each square can be in. These states include:

1. States 0-5: The square has one of the 6 types of white chess pieces on it.
2. States 6-11: The square has one of the 6 types of black chess pieces on it.
3. State 12: The square is empty.



12	3	12	9	12	3	12	12
12	12	0	11	12	12	4	12
12	12	12	12	0	12	1	9
12	12	12	12	12	12	12	5
12	12	12	12	12	12	12	12
12	12	12	12	12	12	12	12
12	12	12	9	12	12	12	12
12	12	12	1	3	12	12	12

Figure 2: Example Chessboard and Tensor Representation

3.3 Model Architecture

The model used is a vision transformer to leverage self attention. The architecture is built using Hugging Face's ViTForImageClassification with a customized configuration [9]. The input image is divided into non-overlapping patches of size 4×4 pixels, which are then linearly embedded and passed through a transformer encoder consisting of 3 hidden layers. Each transformer layer employs a single self-attention head, making the model lightweight and suitable for the relatively low-resolution inputs typical of individual chessboard squares. Regularization is applied using dropout in both the hidden layers and attention mechanisms. The model is trained using a typical supervised learning loop with cross-entropy loss and an Adam optimizer. Accuracy is measured by evaluating the proportion of correctly predicted labels across the test dataset.

4 Results

4.1 Classification Accuracy

The performance of the Vision Transformer model is evaluated using classification accuracy on a per chessboard square level. To evaluate the models generalization ability the classification accuracy was calculated for varying training set sizes. At each training set size the model's accuracy was always evaluated using the entire test set of size 20,000. The results are outlined below:

Table 1: Classification Accuracy

Epochs	Training Set Size	Accuracy (%)
10	10	83.71
10	50	88.32
10	100	97.91
10	500	99.92
10	1000	99.95

As seen by the results, the model is able to perform very well, with a classification accuracy of 83.71%, despite only having 10 chessboard images to train from. As expected, the model improves as more training data is provided. Despite having a training set of 80,000 images the vision transformer model was able to achieve almost 100% accuracy after training on only 500 images.

4.2 Attention Maps

A key motivation for using Vision Transformers in this task is their inherent interpretability through attention mechanisms. The attention matrices generated by the transformer layers provide visual insights into which parts of the input the model is focusing on during classification. Early in training, these attention heatmaps often resembled the general shape or outline of the pieces they were attempting to classify, providing intuitive and human-interpretable visual explanations. However, as training progressed and the model improved, the attention maps became increasingly abstract and less interpretable. This shift is likely due to the model learning more efficient, specialized attention patterns that focus only on the most discriminative regions of the input image (such as specific edges, contours, or color features) rather than capturing the full shape of the piece. This phenomenon suggests a trade-off between model interpretability and raw performance, especially as the model begins to overfit to the training data.

The attention heatmaps for classifying a bishop are shown below for training set sizes of 10, 50 and 1000 in figures 4, 5 and 6 respectively. The bishop was chosen for this demonstrations as the variation between bishop design styles across chess boards is very high. The full images showing the attention map across all layers and the attention maps for training set sizes of 100, and 500 can be found in appendix A. In the figures it can be seen that the attention heatmap becomes increasingly more abstract as the model trains more. It is also interesting to note that after 1000 training images the model has learned to identify black chess pieces by the background shape rather than the piece shape. This discovery is more apparent later when looking at the attention map of the full chessboard.

Figure 3: * Example Bishop From Chessboard Training Set

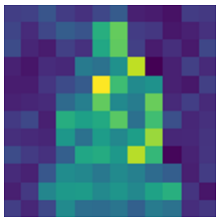


Figure 4: * Attention Heatmap (Trained on 10 Images)

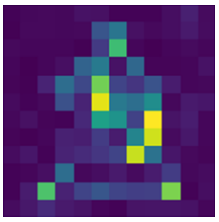


Figure 5: * Attention Heatmap (Trained on 50 Images)

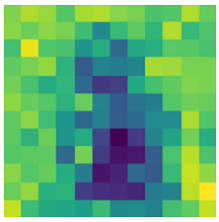


Figure 6: * Attention Heatmap (Trained on 1000 Images)

The attention heatmaps for classifying a rook are shown below for training set sizes of 10, 50 and 1000 in figures 8, 9 and 10 respectively. In contrast to the bishop, the rook was chosen for this demonstrations as the variation in design styles across chess boards is much lower. The full attention maps can also be found in appendix B. The figures again show that the attention heatmap becomes increasingly more abstract as the model trains more and that after 1000 training images the model has learned to identify black chess pieces by the background shape rather than the piece shape.

Figure 7: * Example Bishop From Chessboard Training Set

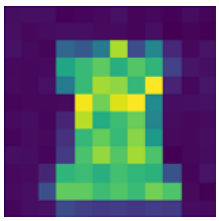


Figure 8: * Attention Heatmap (Trained on 10 Images)

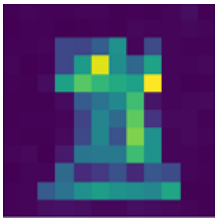


Figure 9: * Attention Heatmap (Trained on 50 Images)

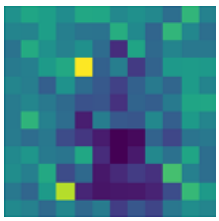


Figure 10: * Attention Heatmap (Trained on 1000 Images)

The attention heatmaps of a full chessboard are shown below for training set sizes of 10, 50 and 1000 in figures 12, 13 and 14 respectively. The heatmaps for training set sizes of 100 and 500 can be found in Appendix C. Figure 12 in particular clearly shows the heatmap's interpretability as the attention matrices show clear outlines of the chess piece shapes. Figure 14 shows the result previously discussed, where black pieces and darker chess squares use the shape of the background square to identify the piece.

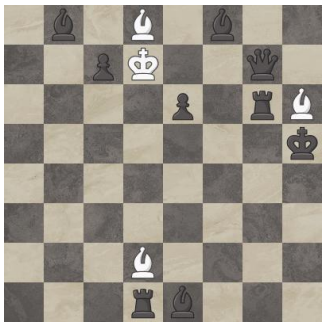


Figure 11: * Example Chessboard Layout From Training Set

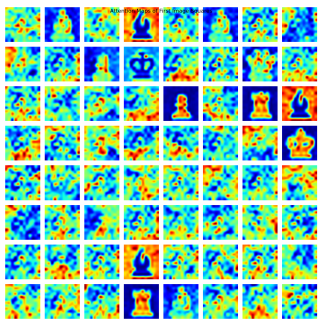


Figure 12: * Attention Heatmap (Trained on 10 Images)

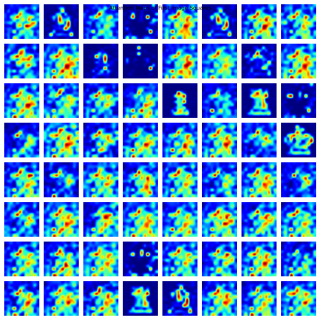


Figure 13: * Attention Heatmap (Trained on 50 Images)

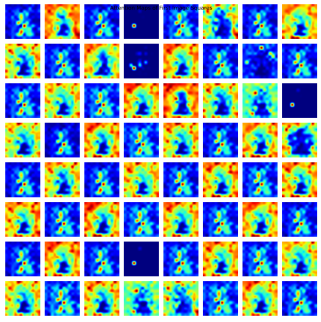


Figure 14: * Attention Heatmap (Trained on 1000 Images)

5 Discussion

5.1 Negative Results and Insights

5.1.1 Attention Heads

Initial experiments included testing multi-head attention to explore whether multiple perspectives on the same input could enhance classification performance. However, results indicated that the attention matrices across different heads showed minimal variation, suggesting redundancy. This implies that additional attention heads did not provide meaningful diversity or improved feature representation in this specific task. This makes sense, as the task of classifying individual chess pieces from small image patches may not require complex, multi-faceted relationships between input regions.

5.1.2 Grayscale Images

Another experiment involved converting the input images to grayscale in an effort to simplify the input and potentially improve generalization. However, the model's performance declined under this preprocessing step. This indicates that color is an important feature for distinguishing between white and black pieces, especially given the variety in the dataset. While grayscale preprocessing could be beneficial in more uniform datasets (e.g., with consistent color schemes where brightness could reliably indicate team color), the variability in board and piece designs in this dataset makes color a critical signal for accurate classification.

5.2 Threats to Validity

A fundamental limitation of the current methodology is that each square is classified independently, with no information about the surrounding board state. As a result, the model is unable to account for the global structure or legality of a chess position. This can lead to implausible or illegal board configurations (for example, predicting three kings on the board). While the per-square classification approach simplifies training, incorporating positional or contextual awareness would be necessary for applications requiring valid game states.

The reported classification accuracy may be artificially high due to the imbalance in class distribution, particularly the large number of empty squares on the board. Since empty squares make up a significant proportion of the dataset, a model that is simply good at detecting emptiness could achieve a deceptively high accuracy.

5.3 Future Work

Incorporating positional encodings or spatial relationships between squares could allow the model to better capture global game context, which is important for ensuring legal and coherent board states. Additionally, ensemble methods or hybrid models that combine CNN-based feature extraction with transformer-based classification could balance the benefits of both architectures. Further analysis using class-wise metrics and confusion matrices would also help in identifying specific areas of weakness, such as misclassifications between visually similar pieces. Exploring methods to address data imbalance, either through resampling techniques or loss function adjustments, may also improve overall robustness.

6 Conclusion

This project demonstrates the feasibility and effectiveness of using Vision Transformers for the classification of chess pieces from individual board squares. By leveraging a well-structured preprocessing pipeline and transformer-based architecture, the model achieves high accuracy while requiring significantly less training data than traditional CNN-based approaches. Moreover, the use of attention mechanisms offers greater interpretability, although this benefit diminishes as the model becomes more specialized. Despite some limitations, such as loss of global board context and class imbalance, the model lays a strong foundation for further research in interpretable, efficient chessboard state recognition from visual inputs.

References

[1] G. Bradski. *The OpenCV Library*. Dr. Dobb's Journal of Software Tools, 2000.

[2] chess.com. "Chess.com - Play Chess Online - Free Games," Chess.com, 2019. https://www.chess.com/.

[3] R. Girshick J. Redmon, S. Divvala and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] P. Koryakin. Chess positions, 2019. tps://www.kaggle.com/datasets/koryakinp/chess-positions.

[5] lichess.org. "The best free, adless Chess server," lichess.org, Jun. 27, 2019. https://lichess.org/.

[6] A. Laskowski M. A. Czyzewski and S. Wasik. Chessboard and chess piece recognition with the support of neural networks, 2020.

[7] R. Girshick S. Ren, K. He and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[8] stockfishchess.org. "Home - Stockfish - Open Source Chess Engine," stockfishchess.org. https://stockfishchess.org/.

[9] Vision Transformer (ViT). huggingface.co. https://huggingface.co/docs/transformers/en/model_doc/vit.

[10] G. Wölflein and O. Arandjelović. *Determining Chess Game State from an Image*. Journal of Imaging, 2021.

A Bishop Heatmaps Across All Layers

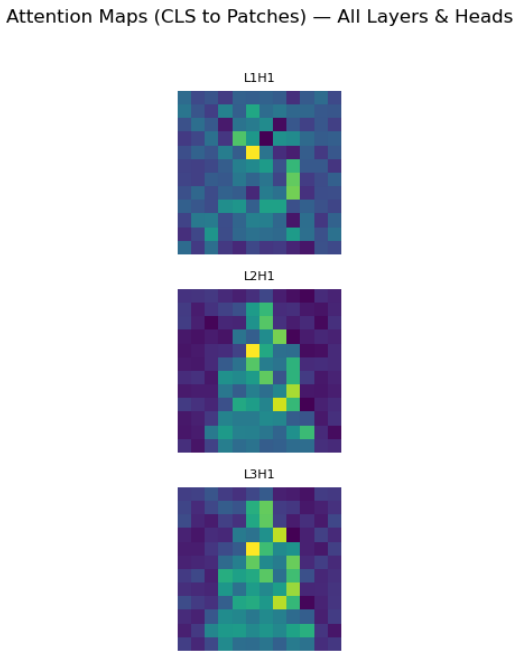


Figure 15: Attention Heatmap (Trained on 10 Images)

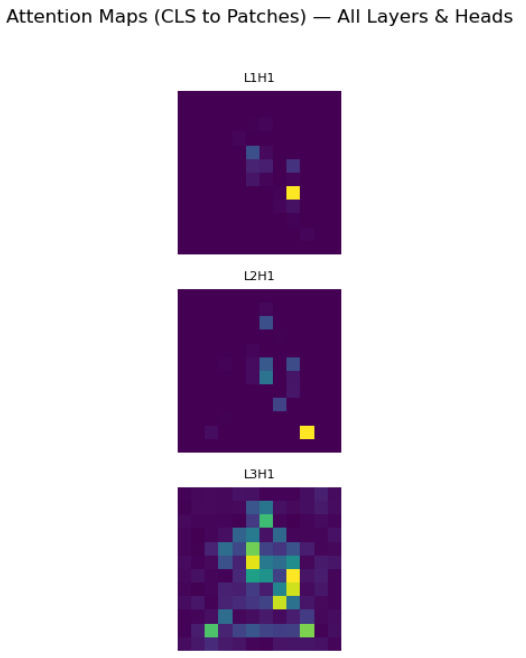


Figure 16: Attention Heatmap (Trained on 50 Images)

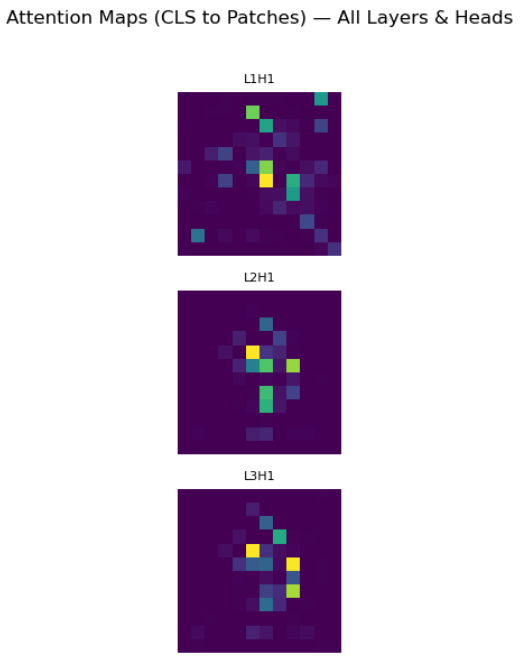


Figure 17: Attention Heatmap (Trained on 100 Images)

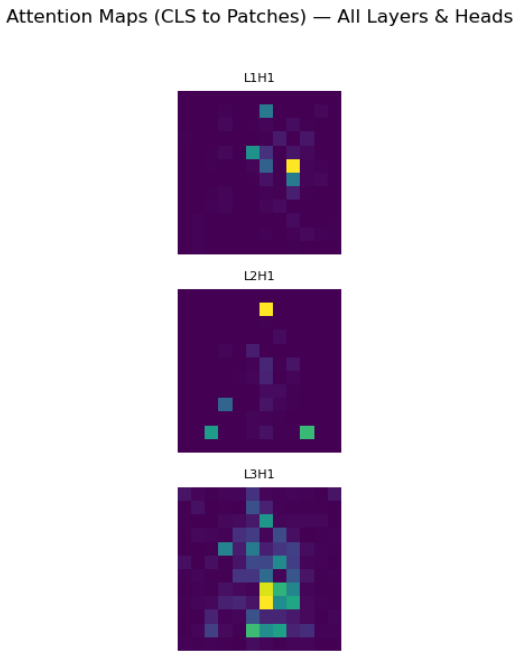


Figure 18: Attention Heatmap (Trained on 500 Images)

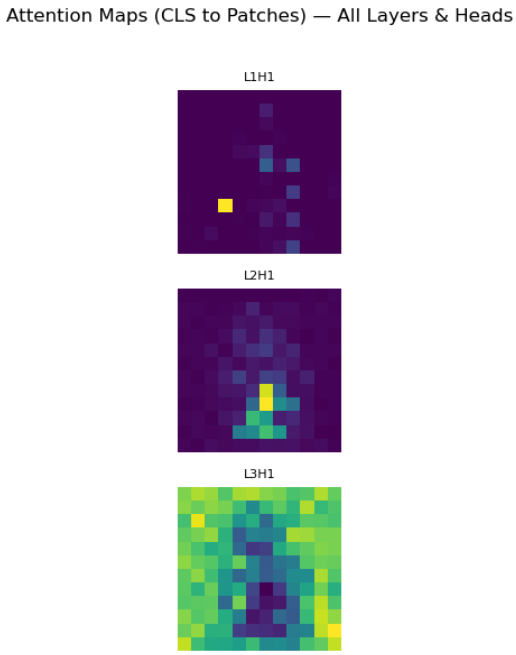


Figure 19: Attention Heatmap (Trained on 1000 Images)

B Rook Heatmaps Across All Layers

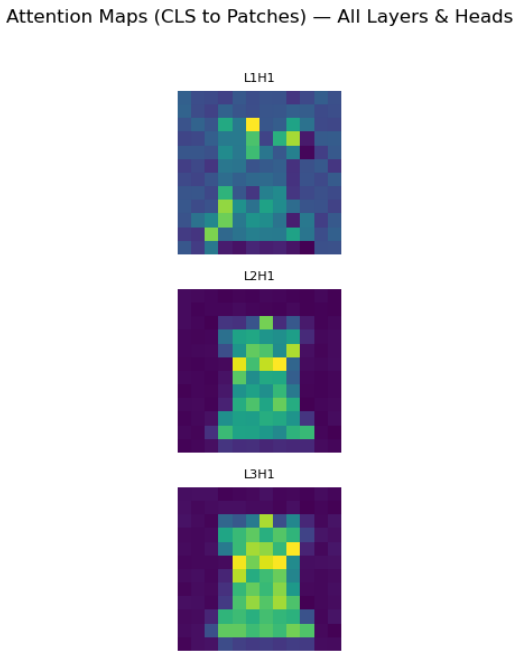


Figure 20: Attention Heatmap (Trained on 10 Images)

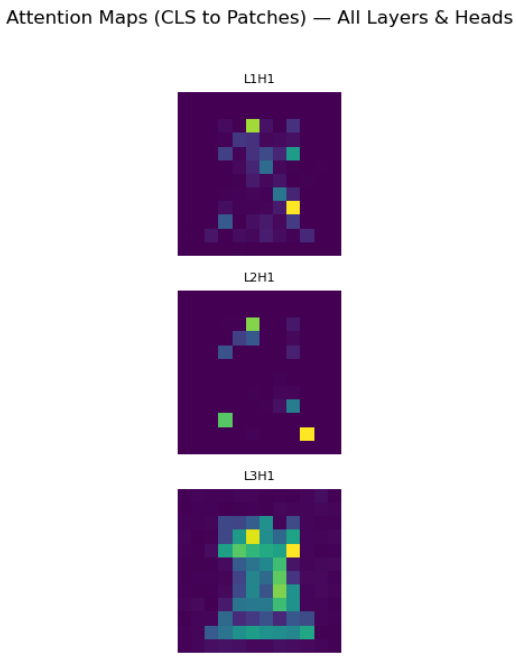


Figure 21: Attention Heatmap (Trained on 50 Images)

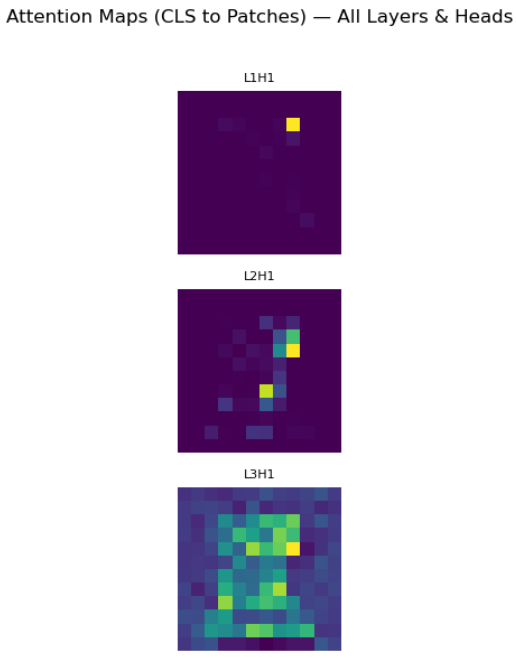


Figure 22: Attention Heatmap (Trained on 100 Images)

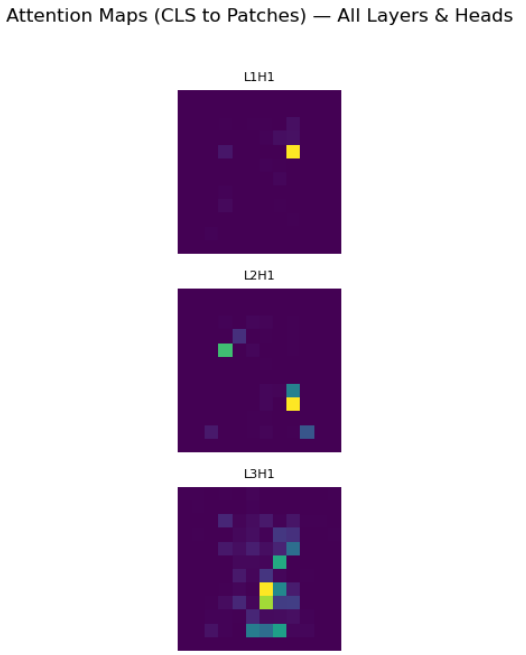


Figure 23: Attention Heatmap (Trained on 500 Images)

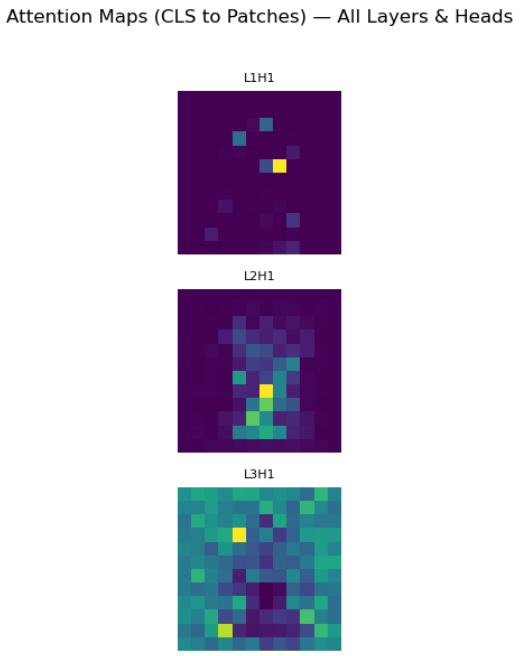


Figure 24: Attention Heatmap (Trained on 1000 Images)

C Chessboard Heatmaps

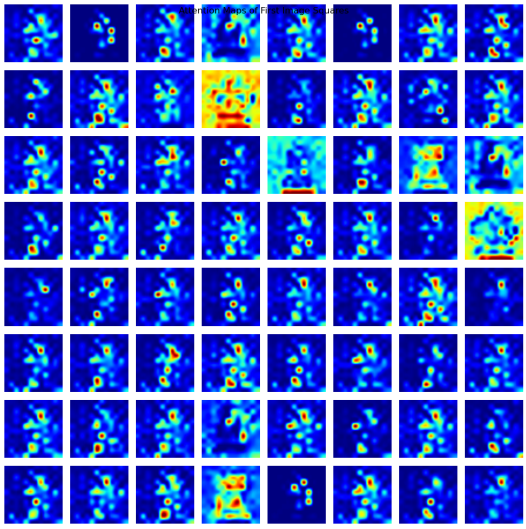


Figure 25: Attention Heatmap (Trained on 100 Images)

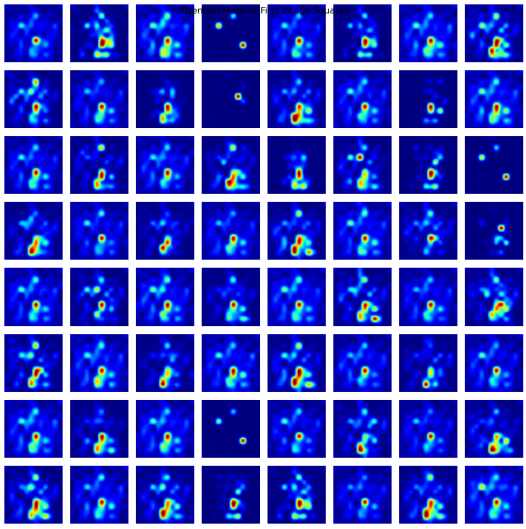


Figure 26: Attention Heatmap (Trained on 500 Images)

D Source Code

Source code is available here: https://github.com/EthanRMcKeen/ViT-Chess-Position_Analysis