Mateo Restrepo and Tadija Ciric 12/18/2023

Project Report: Chess Vision - Enhancing Chess Piece Recognition Using Deep Learning

Introduction

In this project, we explore the application of deep learning in the realm of chess, specifically focusing on the task of chess piece recognition. The classic game of chess presents a unique challenge for image recognition due to the variety in piece designs and the need for precise identification for game analysis and digitization.

Motivation

The primary motivation for this project is to enhance the integration of physical chess gameplay with digital analysis tools. This integration is crucial for applications such as automated game logging, real-time move tracking, and interactive training programs. By leveraging advanced image recognition techniques, particularly convolutional neural networks (CNNs), the project aims to develop a robust model capable of accurately identifying different chess pieces from images. This task involves overcoming challenges related to the nuances in piece designs and the varying conditions under which chess images might be captured. The successful implementation of this model has the potential to bridge the gap between physical and digital chess experiences, offering a seamless transition for game analysis and player improvement.

The following sections will detail the dataset used, the methodology implemented, the challenges encountered in model development, and an evaluation of the results achieved.

Dataset Composition

For this project, a dataset comprising images of six types of chess pieces — Knights, Bishops, Rooks, Queens, Kings, and Pawns — was utilized. The dataset included a balanced collection of images for each piece, around 110 of each, ensuring a comprehensive representation of all classes. The images were preprocessed to maintain uniformity in size and color formatting, with each resized to 224x224 pixels to align with the input requirements of the VGG16 and VGG19 models used in later stages of classification.

Preprocessing Steps

- **Resizing:** All images were resized to 224x224 pixels, a necessary step to standardize input size for efficient processing by the CNN models.
- **Normalization:** Pixel values were normalized by scaling them to a range of 0 to 1 (by dividing by 255). This normalization aids in faster convergence during model training by maintaining consistency in the scale of input data.

Data Augmentation

Data augmentation played a pivotal role in enhancing the robustness of the models. The augmentation techniques applied included:

- **Rotation:** Images were rotated by up to 20 degrees to simulate variations in piece orientation.
- **Width and Height Shifts:** These shifts (up to 10%) mimicked slight changes in the piece's position on the chessboard.
- **Shear Transformation:** Applied to introduce geometric changes, helping the model learn to recognize pieces from different angles.
- **Zoom:** Slight zooming (up to 10%) exposed the models to variations in piece size.
- **Horizontal Flip:** Images were flipped horizontally to simulate a mirrored view of the pieces, further diversifying the training data.
- **Fill Mode:** The 'nearest' fill mode was used to fill in any new pixels created after a transformation, maintaining the integrity of the chess piece design.

Goal of Preprocessing and Augmentation

The preprocessing and augmentation steps were designed to achieve two primary objectives:

1. **Standardization:** Preprocessing ensured that all input data fed into the models were uniform in size and scale, a crucial factor for the consistency and accuracy of a CNN.
2. **Generalization:** Data augmentation introduced a variety of transformations to the images, thereby simulating real-world variations. This approach aimed to increase the models' generalization capabilities, enabling them to recognize chess pieces under different conditions and reduce the likelihood of overfitting.

3. Related Work

Reference to Previous Study

This project draws inspiration from a previous project focused on predicting music genres. In that project, a similar approach was employed where a multiclass classification problem was deconstructed into multiple binary classification tasks.

Advancements in the Current Project

Building upon the foundations laid by the music genre prediction project, this project extends the binary classification approach in several significant ways:

1. **Integration with VGG16 Architecture:**
    - While the music genre prediction project laid the groundwork for using binary classifiers, our project further enhances this approach by integrating the VGG16 architecture. This well-established CNN architecture is known for its effectiveness in image classification tasks and provides a robust foundation for our binary models.
2. **Addition of Generative Adversarial Networks (GANs):**

    - Our project incorporates GANs to generate synthetic images of chess pieces. This addition addresses the challenge of limited data in specific classes and enhances the diversity of the training dataset, leading to improved model generalization.
3. **Unified Model through Weighted Voting:**
    - Our project combines the outputs of individual binary classifiers into a unified model. This is achieved through a weighted voting mechanism, where the predictions of each binary model are aggregated to determine the final class. This approach not only leverages the strengths of individual classifiers but also provides a cohesive and comprehensive prediction model, altough with some challenges

4. Methodology

4.1 Initial Multiclass Model Approach

- **Initial Strategy:** The project initially adopted a multiclass classification approach using a Convolutional Neural Network. The model aimed to classify images directly into one of the six chess piece categories.
- **Challenges Encountered:** The multiclass model faced challenges, particularly in distinguishing between pieces with subtle differences, leading to lower accuracy rates.

4.2 Transition to Binary Classification Models

- **Adopting Binary Models:** To address the limitations of the multiclass model, the strategy was shifted to develop individual binary models for each chess piece. This approach allowed for more focused and nuanced training on each piece type.
- **VGG16 Integration:** Each binary model was based on the VGG16 architecture, chosen for its proven effectiveness in image classification tasks. Modifications were made to adapt this architecture to binary classification, primarily by altering the final layers to output a single probability score.
- **Challenges in Binary Models:** The primary challenge was balancing the model's complexity with the available data, leading to issues like overfitting. Solutions included implementing the modified VGG16 arcihtecture dropout layers and L2 regularization, adjusting the learning rate, and using early stopping during training.

4.3 Combining Binary Models into a Unified System

- **Weighted Voting System:** The outputs of the six binary models were combined using a weighted voting system. Each model's prediction was assigned a weight based on its validation accuracy, contributing to the final decision.
- **Debugging and Model Calibration:** Extensive debugging was conducted to ensure the accuracy of each binary model post-loading. Calibration techniques were explored to align model confidence with accuracy.
- **Hyperparameter Tuning:** Further adjustments to hyperparameters were made iteratively based on the performance of the combined model.

4.4 Implementation of Generative Adversarial Networks (GANs)

- **Exploratory Implementation:** Generative Adversarial Networks (GANs) were implemented to explore the creation of synthetic chess piece images using the same dataset as for the classification models. This exploration was intended to evaluate the efficacy of GANs in generating visually accurate chess images.

- **Independent Development:** The development of the GANs was kept separate from the main classification task. While initially there was a possibility of integrating GAN-generated images into the training dataset, the focus shifted to assessing the standalone capabilities of the GANs in image generation.

- **Performance of GANs:** The GANs showed promise in generating chess piece images, indicating potential applications in image synthesis and augmentation.
- **Prospective Integration:** Future work may explore the integration of GAN-generated images for advanced data augmentation or developing GANs further as a complementary tool for chess image analysis and generation.

5. Results and Evaluation

Multiclass Model Performance

- **Base Model Training:** The initial training of the base multiclass model showed a learning curve with loss and accuracy over epochs. These graphs indicated how the model's performance evolved during training.
- **Fine-Tuning Results:** Post fine-tuning, additional graphs for loss and accuracy over epochs illustrated improvements in model performance, highlighting the effectiveness of fine-tuning strategies.
- **Accuracy and Confusion Matrix:** The final accuracy graph and confusion matrix provided a clear visualization of the model's classification capabilities, pinpointing areas of strength and weakness in distinguishing between different chess pieces.

Binary Model Evaluations

For each binary model (e.g., Knight, Bishop, etc.), several key metrics and visualizations were used for evaluation:

- **Test Accuracy:** The evaluation of each model on the test dataset provided a direct measure of its accuracy, quantifying its effectiveness in classifying images as either belonging to the target class or not.
- **Classification Report:** The classification report, including precision, recall, and F1-score, offered a detailed view of each model's performance, highlighting its predictive capabilities for both classes.
- **ROC-AUC Curve:** The Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) score depicted each model's diagnostic ability, illustrating the trade-off between true positive rate and false positive rate.

- **Training History:** Plots of accuracy and loss over epochs for both training and validation phases demonstrated the learning progression of each model, revealing trends like overfitting or underfitting.

Interpretation and Thought Process

- **Model Training Observations:** During training, notes were made on the models' responses to different hyperparameters and architectural changes. This iterative process informed decisions on adjustments to improve model performance.
- **Fine-Tuning Insights:** Fine-tuning the models based on initial results led to notable improvements, as seen in the reduced overfitting and enhanced validation accuracy.
- **Binary Model Successes:** Certain binary models achieved high accuracy, validating the approach of treating the problem as a series of binary classifications.
- **Challenges and Adjustments:** Challenges such as overfitting in specific models prompted strategies like increasing dropout rates or adjusting learning rates, underscoring the importance of continuous monitoring and adaptation in model training.

Performance of Individual Binary Models

- **Accuracy Metrics:** Each binary model was evaluated based on accuracy, precision, recall, and F1-score. The models exhibited varied performance, with some achieving high accuracy levels (the Pawn and Rook models) while others required further tuning (the Queen and Bishop models).
- **Overfitting Challenges:** The Knight model, in particular, demonstrated a tendency to overfit, consistently misclassifying non-Knight images as Knights. Adjustments such as increasing dropout rates and implementing L2 regularization were made to mitigate this issue.

Combined Model Performance

- **Weighted Voting System:** The integration of individual binary models into a combined model using a weighted voting system resulted in improved overall performance. This system effectively harnessed the strengths of each binary model to make more accurate final predictions.
- **Model Calibration:** Calibration techniques were explored to enhance the confidence and reliability of the predictions made by the combined model.

Generative Adversarial Networks (GANs) Impact

• **Dataset Enhancement**: GANs were employed to expand the dataset, particularly for underrepresented classes, improving the diversity and richness of the training data.

• **Progressive Refinements**: Continuous refinement of the GAN models led to increasingly realistic image generation, contributing positively to the overall project.

• **Integration Challenges**: Integrating the GAN-generated images into the training set posed challenges, requiring careful balance to avoid skewing the model's performance.

• **Targeted Focus**: Focusing GANs on generating images of specific classes, like Knights, proved beneficial, allowing for targeted improvement in areas where the dataset was lacking.

Enhanced Model Robustness: While the GANs did not fully achieve the goal of creating highly realistic images for direct integration, they provided valuable insights into data augmentation techniques and the potential of synthetic data in model training. Future efforts will focus on refining these techniques for better integration with classification models.

Graphical Representation of Results

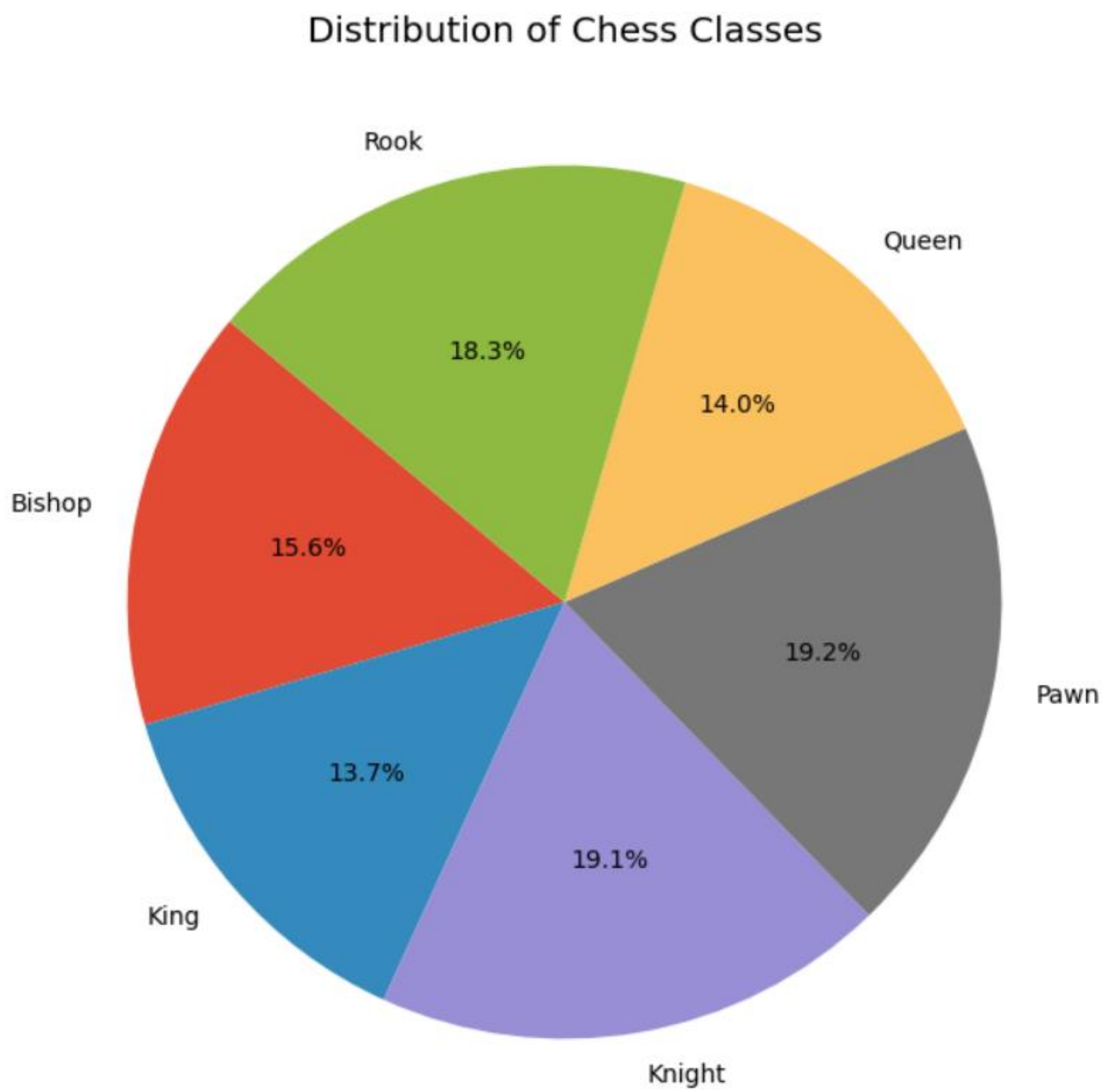
- Graphs and charts are used to visually represent the performance metrics, showing the improvements in model accuracy and other metrics over successive iterations of training and tuning.

Critical Evaluation of Methodology

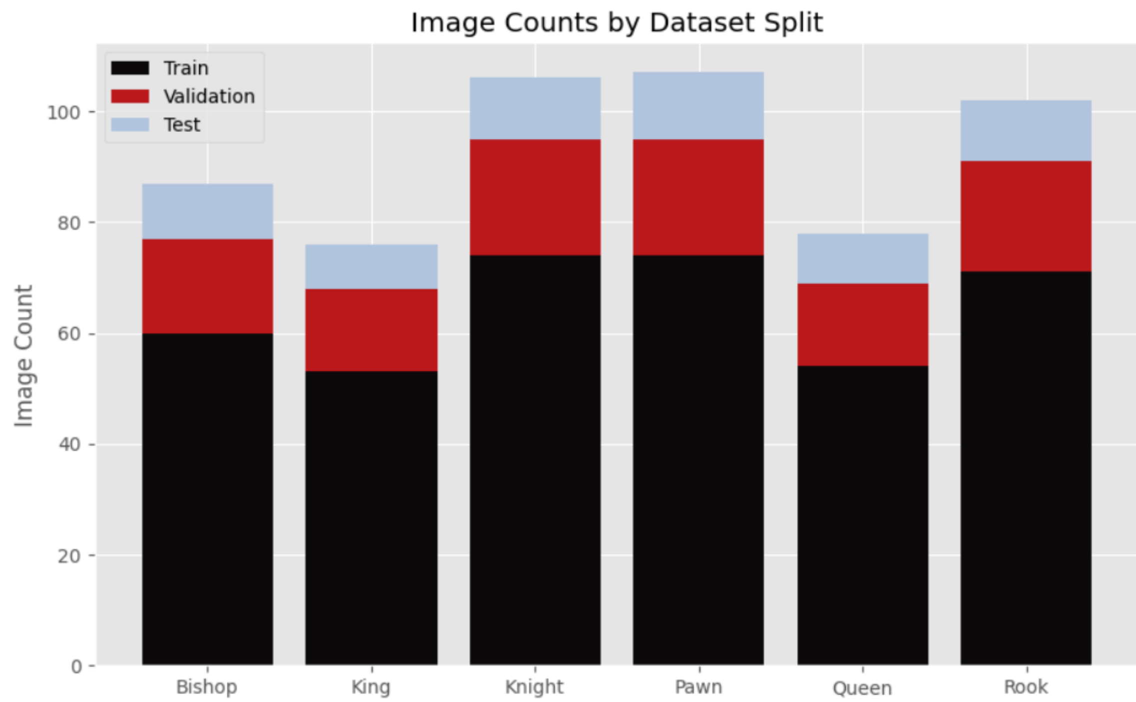
- **Successes:** The project successfully demonstrated the efficacy of using binary models for complex classification tasks and the advantages of a weighted voting system in creating a cohesive combined model.
- **Challenges:** One of the key challenges faced was balancing model complexity with the available data size to prevent overfitting. The project highlighted the importance of data quality and diversity in training effective machine learning models. The model had and still has trouble loading saved models effectivily and achieving good results, even though certain models show promising results.
- **Learnings:** The iterative process of model development, involving continuous testing, debugging, and hyperparameter tuning, underscored the dynamic nature of machine learning projects.

Conclusion

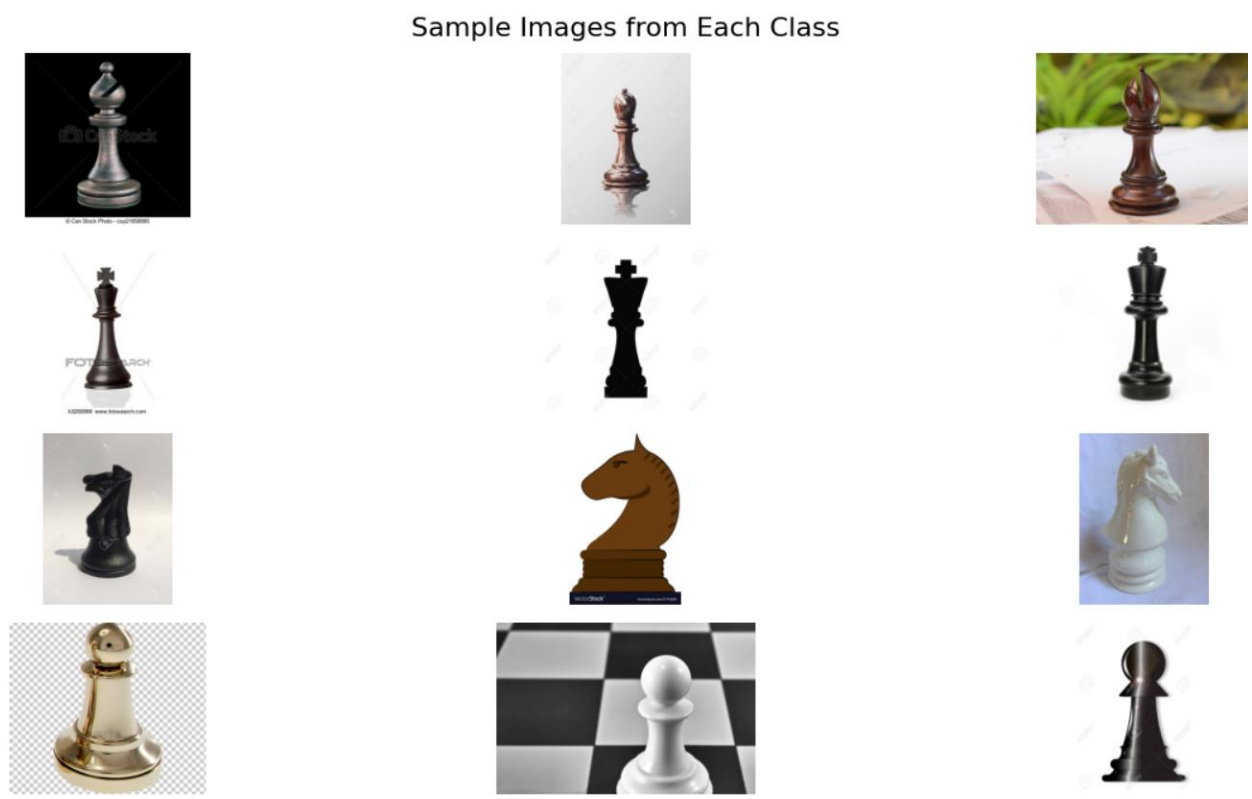
- The project, despite challenges, demonstrated significant progress in chess piece recognition, highlighting the effectiveness of the combined model approach in complex classification tasks. Future work will aim to further refine these models, incorporating lessons learned and exploring new techniques for enhanced performance and accuracy.



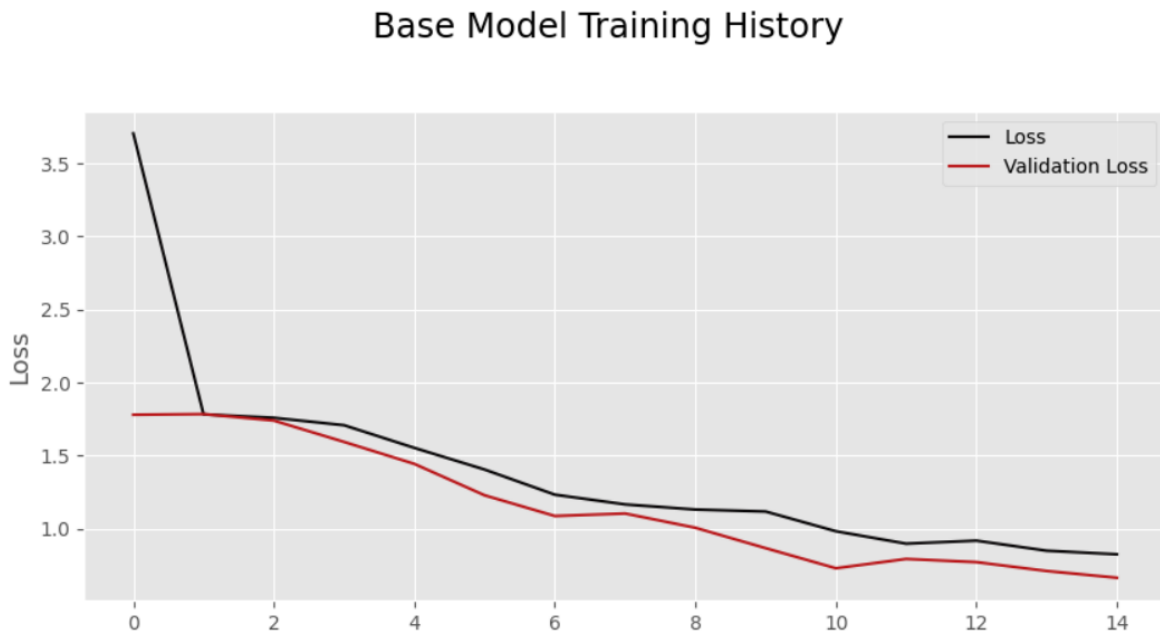
Graph 1. Distribution of images in the dataset



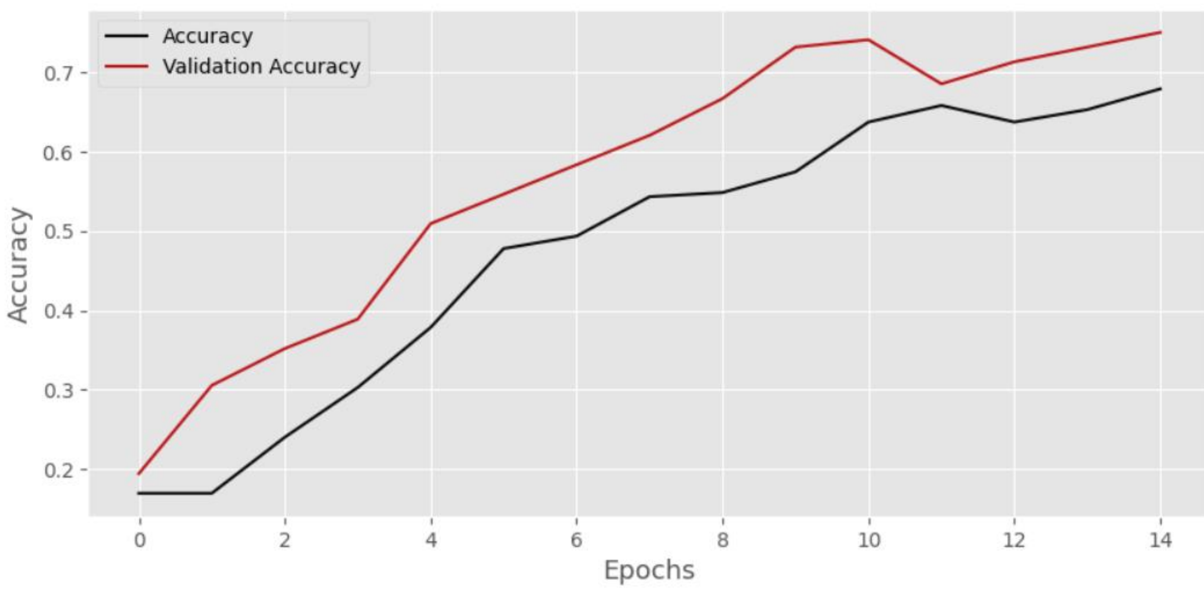
Graph 2. Distribution of validation, traing and test sets in the dataset



Graph 3. Sample images from each class (Bishop, King, Knight and Pawn shown)

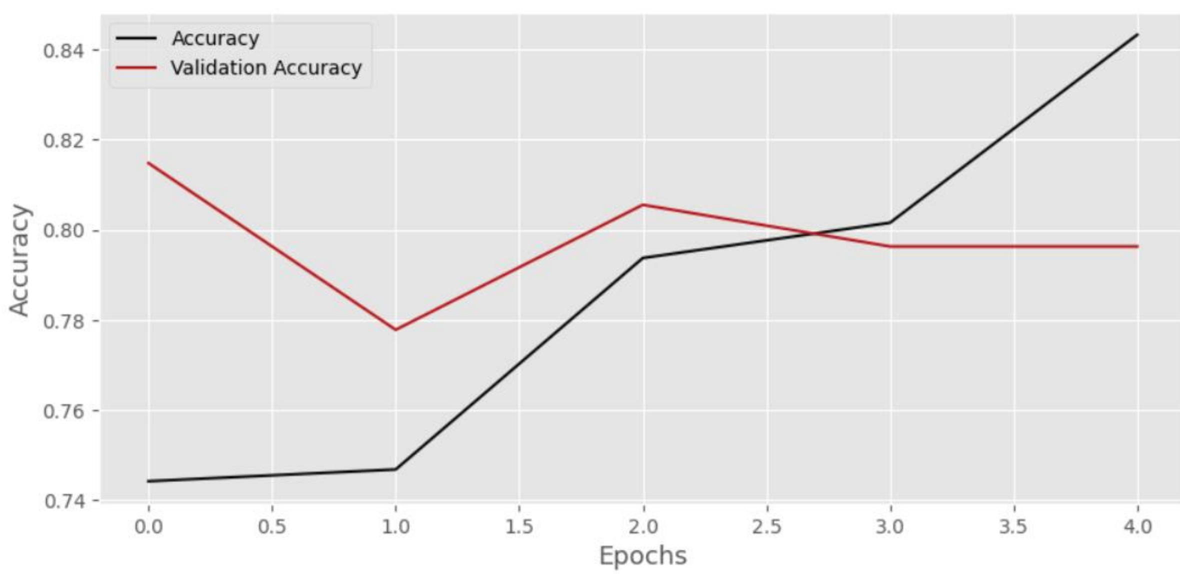


Graph 4. Training history of multiclass model

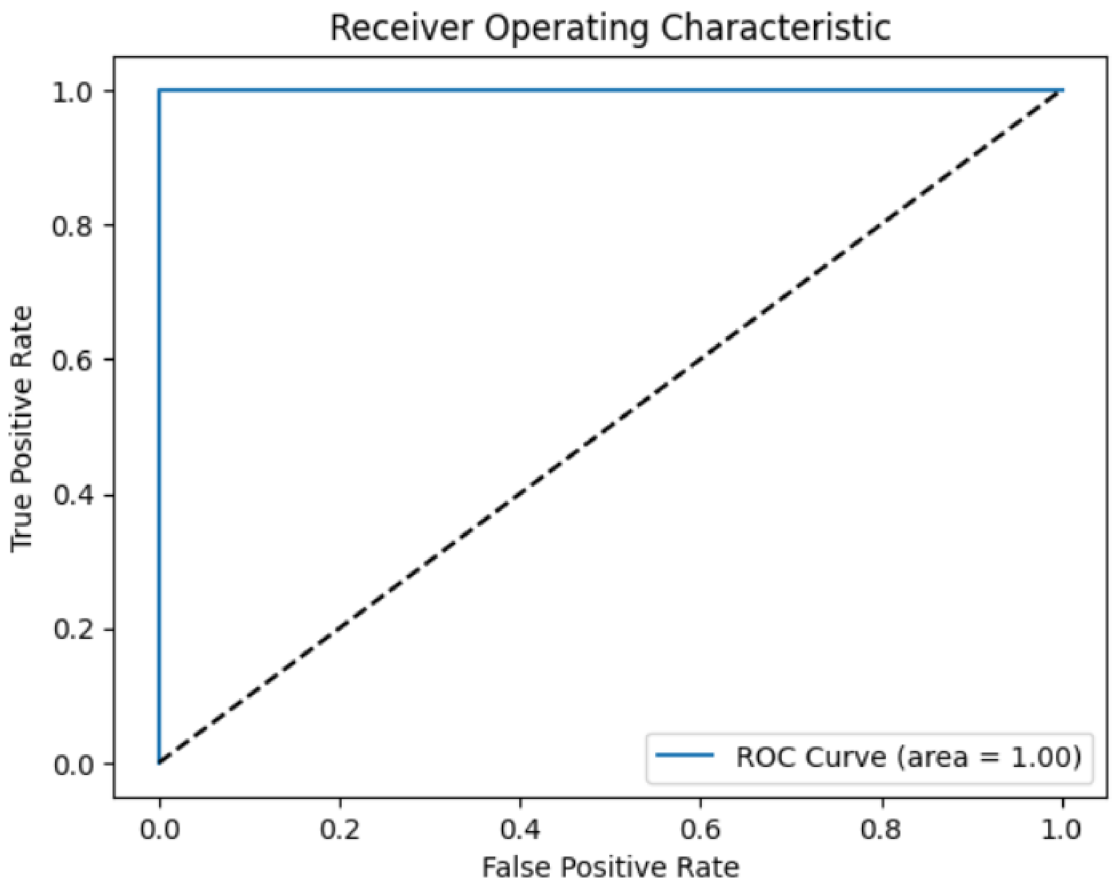


Graph 5. Accuracy over epochs for multiclass model

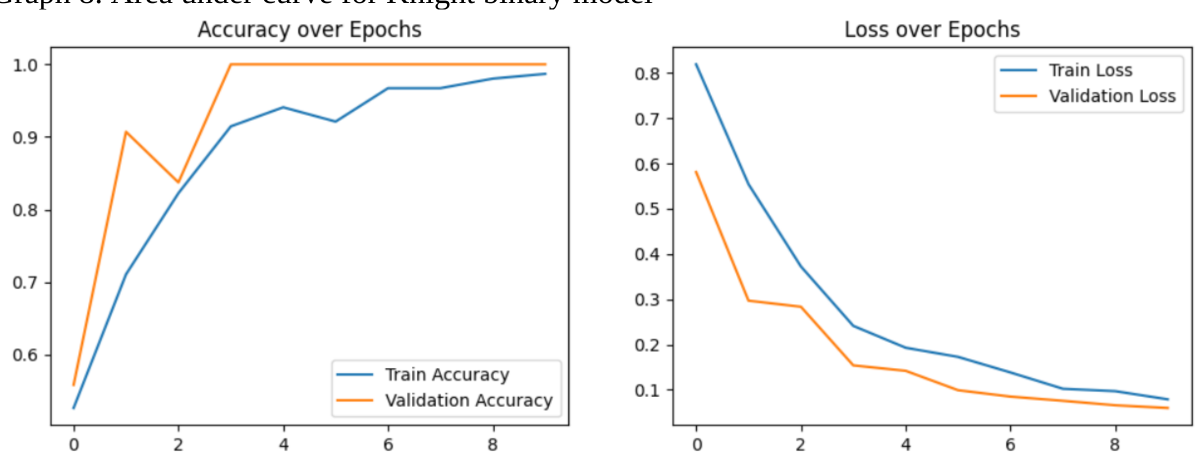
Graph 6. Training history for fine-tuned multiclass model



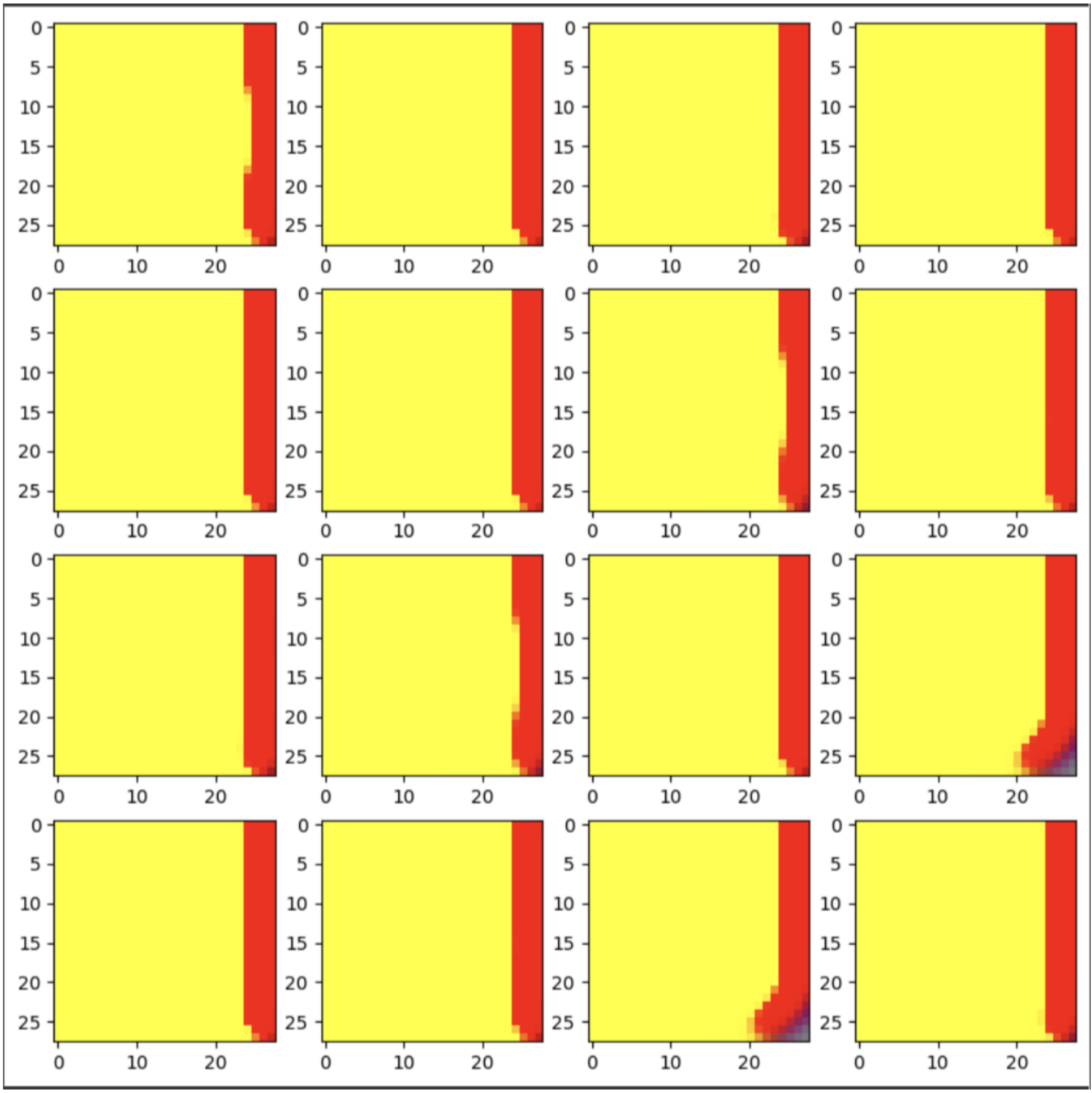
Graph 7. Accuracy over epochs for fine-tuned multiclass model



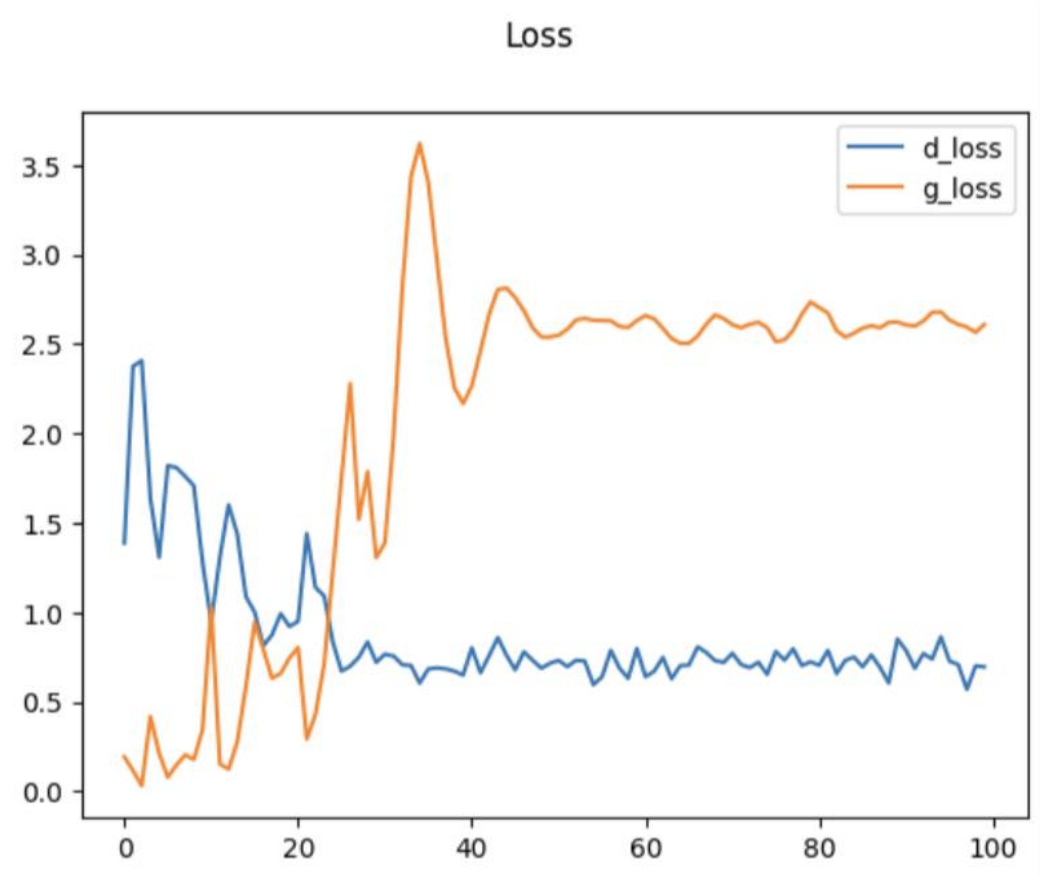
Graph 8. Area under curve for Knight binary model



Graph 9. Accuracy and loss over epochs for knight binary model



Graph 10. Generated images by the GAN model



Graph 11. Discriminator and generator loss for GAN model