ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

Smart Chess Assistant: Using AI to See the Board and Suggest the Best Moves

Ganesh Shelke[1]; Durgesh Sakhare[2]; Prashant Jadhav[3]; Shreyas Sakat[4]; Sachin Amrit[5]

[1]Department of Information Technology,
Vishwakarma Institute of Technology, Pune
[2]Department of Information Technology,
BRACT's Vishwakarma Institute of Information Technology, Pune
[3]Department of Information Technology,
BRACT's Vishwakarma Institute of Information Technology, Pune
[4]Department of Information Technology,
BRACT's Vishwakarma Institute of Information Technology, Pune
[5]Department of Information Technology,
BRACT's Vishwakarma Institute of Information Technology, Pune

Publication Date: 2025/04/11

Abstract: In this study we present such an artificial intelligence driven system capable of determining the chess pieces on a physical board and providing recommendations of the best move, given the current game position. The system employs the combination of YOLOv8 object detection algorithm for accurate piece recognition and Stockfish chess engine for strategic move recommendations which allows it to run time analysis of gameplay in a real time basis. We train the YOLOv8 model on a chess image dataset with 606 chess images tagged, and thus detect with precision between 91% and 98% depending on the piece. It looks at the application of deep learning technology in the analysis of chess and its key role for automated systems to both improve and compete in educational and competitive play. Through this work, we show the potential in use of chess with AI technologies and addressing some issues in position recognition accuracy and piece identification. Beyond this, it describes in detail a plan for future improvements, including more advanced image processing techniques and further applications of the system to other board games and other sports. Overall, the results from this research provide evidence of the profound shifts that AI can bring to traditional game play, laying the groundwork for future work in intelligent gaming solutions which can both enhance the game player experience, promote learning, and further enhance the competitive dimension of chess.

Keywords: *Artificial Intelligence, Chess, Deep Learning, Object Detection, YOLOv8, Stockfish, Move Suggestion.*

How to Cite: Ganesh Shelke; Durgesh Sakhare; Prashant Jadhav; Shreyas Sakat; Sachin Amrit (2025). Smart Chess Assistant: Using AI to See the Board and Suggest the Best Moves. *International Journal of Innovative Science and Research Technology*, 10(4), 91-103. https://doi.org/10.38124/ijisrt/25apr133

I. INTRODUCTION

For centuries, the game of chess has captivated players worldwide, combining elements of strategy, tactics, and planning. Its widespread popularity, with millions of active participants, has elevated it beyond mere entertainment to a platform for intellectual stimulation, competitive play, and educational purposes. The game's intricacies make it an excellent subject for artificial intelligence (AI) research and development. From novices to experts, chess demands not only a thorough grasp of tactics but also the capacity to predict an opponent's future moves. These requirements have positioned chess as a prominent field of study in AI, offering a structured environment to analyze and enhance decision-making and problem-solving abilities.

AI researchers have long utilized chess as a testing ground for developing algorithms that emulate human cognition and decision-making processes. Traditional chess engines like Stockfish and Komodo have achieved significant success by employing exhaustive search algorithms coupled with advanced evaluation functions capable of analyzing millions of positions per second. These engines rely on pre-programmed rules and heuristics to mimic human-like behavior. However, as AI technology evolves, there has been a shift towards employing deep learning and neural networks to create models that learn from extensive datasets and adapt to various

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

scenarios. A notable example is DeepMind's AlphaZero, which mastered chess, Go, and shogi without human input or prior knowledge of the games, surpassing the capabilities of conventional engines.

Despite these advancements in digital chess, accurately detecting and interpreting the positions of physical chess pieces on a real board remains a considerable challenge. Integrating computer vision and AI techniques into physical chess setups has often encountered limitations, including reliance on manual input or oversimplified models that struggle to capture the game's complexity in real-world situations. Many existing systems require players to manually enter moves or utilize specially designed boards equipped with sensors or RFID tags to monitor the pieces. While these approaches function in controlled settings, they fail to provide the flexibility and automation necessary for a more immersive and interactive real-time chess experience. Advancements in deep learning and computer vision offer solutions to overcome existing challenges. Deep learning, an AI subset, constructs neural networks that emulate the human brain's pattern recognition abilities. When utilized for image analysis, these models can accurately identify objects in complex, dynamic settings. Specifically, Convolutional Neural Networks (CNNs) have demonstrated remarkable effectiveness in object detection, making them ideal for recognizing chess pieces on physical boards. By harnessing these innovations, it becomes feasible to automate chess piece detection and positioning, minimizing the need for manual input and enhancing user experience.

YOLOv8 (You Only Look Once version 8) represents a state-of-the-art object detection model designed for high precision and rapid processing. This model employs a single neural network to forecast bounding boxes and class probabilities for objects within an image. Its real-time processing capabilities make it particularly suitable for applications requiring swift, accurate detection, such as live chess analysis. Incorporating YOLOv8 into a chess detection system enables automatic piece identification during ongoing games, improving user experience and facilitating AI-driven insights and move suggestions during live play.

The proposed system also incorporates the Stockfish chess engine, one of the most robust open-source chess engines available. Stockfish utilizes sophisticated algorithms and an extensive database of chess positions to generate optimal move recommendations. By merging YOLOv8's visual recognition capabilities with Stockfish's analytical prowess, the system aims to deliver instantaneous feedback and move suggestions, helping users enhance their skills and game comprehension. This integrated system provides a dependable and efficient method for improving gameplay through AI, catering to both amateur players seeking guidance and professionals analyzing intricate positions. Combining computer vision with chess engines enhances gameplay and offers wider educational benefits. Chess serves as a valuable tool for developing critical thinking, decision-making, and problem-solving abilities. AI-powered chess analysis can assist educators by offering a platform that allows students to visualize moves, grasp strategic concepts, and obtain feedback on their performance. This system is also beneficial in competitive settings, enabling players and coaches to conduct detailed game reviews, explore various strategies, and learn from their mistakes.

The applications of this technology extend beyond gameplay improvement. In chess tournaments, where fairness and precision are crucial, AI-driven analysis systems can be employed for real-time monitoring and move verification. This helps minimize human errors and maintain game integrity. Furthermore, the widespread availability of such technology can democratize access to chess analysis tools, making them more accessible to players who may lack resources for professional coaching.

This research aims to create a comprehensive system that uses deep learning to automate chess piece detection and provide optimal move suggestions. By integrating YOLOv8, a cutting-edge object detection model, with Stockfish, the proposed system offers a user-friendly interface for real-time chess analysis. The main objectives of this study are to develop a reliable system capable of accurately detecting chess pieces on a physical board, enhance the precision of move suggestions using Stockfish, and investigate the potential educational and competitive applications of AI in chess. This fusion of technology and chess not only advances AI research but also enriches the experience for players, educators, and enthusiasts, showcasing the growing intersection between artificial intelligence and the traditional game of chess.

II. REVIEW OF EXISTING LITERATURE

The approach in chess piece detection parallels that of using deep neural networks in medical diagnostics because models require high accuracy in image recognition, as Esteva et al. (2017) did. This work shows that deep learning is promising for other object recognition tasks such as chess piece identification. We also contribute the paper’s methodology, using a large labeled dataset and a convolutional neural network (CNN), which can be viewed as a useful framework for building a chess piece detection system similar to ours.[1]

And LeCun, Bengio and Hinton (2015) teach you foundational knowledge on deep learning, which specialized algorithms like CNNs and how it is used to recognize images in the classification field. This is important for the case when AI sees pictures of chess pieces instead of processing moves from input as they will need to be able to detect those chess pieces from those pictures. These principles in this review, such as the use of backpropagation and feature extraction, can be used to construct and to train chess piece models that accurately recognize chess piece. [2]

This work emulates the application of reinforcement learning in AlphaGo, where Silver et al. (2017) demonstrated that AlphaGo is applicable to chess move suggestion systems. The training of a Go playing system with self play and deep reinforcement learning provides insight into how an AI system might be trained to make optimal move suggestions in chess through previous learned games and simulations. The training strategies for without human input to build autonomous chess engines are of great value. [3]

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

Blind chess piece detection as a prediction problem was introduced by Redmon et al. (2016) who demonstrated that the object detection algorithm YOLO could be adapted for this task. Due to its real time processing capabilities and high detection accuracy, YOLO is a good enough candidate for tracking being selected through pieces on a chessboard. This algorithm can be integrated into an AI based chess system for piece recognition, improving the efficiency and accuracy thereby contributing performing autonomous chess engine. [4]

Imaging with deep learning, Rajpurkar et al. (2017) achieved results of similarly comparable humans. It is applicable to chess piece detection because the pieces must be detected with high precision. Their work demonstrates how deep learning models can reach human level performance in tasks that leverage image recognition (use of large datasets and robust CNNs), which are precisely the abilities that have made chess recognition systems both possible and plausible.[5]

While McCarthy (2007) reviews the progress of AI since the Dartmouth Conference, he refines the topics of machine learning and object recognition. This also provides the valuable history behind the evolution of AI systems that put us where we are today regarding modern AI based chess engines. The challenges of early AI models—which McCarthy emphasizes—are also relevant when today's complex image classification tasks need to be solved with sophisticated algorithms like CNNs.[6]

Earlier, GANs were introduced by Goodfellow et al. (2014), which potentially can help improve AI based chess detection systems. Although GANs are commonly employed to generate real looking images, we demonstrate that their framework can be redefined to increase the accuracy of chess piece recognition with the use of synthetic training data. Adversarial training could train models better, with the aim to predict chess moves more precisely using the parts of the board.[7]

A variant of CNN, developed by Krizhevsky, Sutskever and Hinton (2012), AlexNet, is a CNN architecture that completely changed the game of the image classification tasks. Using large scale datasets will produce a model architecture that can be used for chess piece detection like AlexNet did. The ability to apply the paper's methodology of extracting image features using convolutional layers to develop a model that identifies things, such as pieces on a chessboard, is necessary for move suggestion algorithms.[8]

I thoroughly ran through application and efficiency of stochastic gradient descent (SGD) for training large scale machine learning models (Bottou, 2010). The scalability and adaptability of SGD that is demonstrated by his work offers an outstanding optimization ability for deep learning models that involve large datasets. When it comes to chess piece detection and move suggestion systems, owning the ability to efficiently optimize complex neural networks is absolutely key for SGD. The dataset used for training AI models to recognise and classify chess pieces are often a collection of various chess configurations, positions and lighting conditions, such that the model needs to generalize well. The use of SGD makes it possible for the model to learn from such extensive datasets within reasonable computational overheads so that the training process is hugely fast and tractable. In addition, following the tradition of Bottou's adaptive learning rates and mini batch processing strategies, other enhancements to improve model performance are possible which make it possible to train neural networks in the real time environments as in the chess games where quick and accurate decision making is necessary. Using Bottou's insights, developers can construct systems for building scalable and efficient chess AI systems that can process and interpret real time data from chess boards to derive optimized piece recognition along with move prediction algorithm for a responsive and accurate chess playing interface.[9]

In J. Redmon and A. Farhadi (2018) they further refine YOLO with YOLOv3, an updated object detection algorithm. This development is important for chess piece detection systems as it increases accuracy and speed for real time applications. Especially important to building a chess recognition system that works in different conditions, is the architecture's ability to detect objects in different environments and lighting conditions. [10]

III. METHODOLOGY

This section describes the steps taken to create an AI-powered system for chess piece recognition and move recommendation. It covers data acquisition, preparation, model training, and system architecture.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

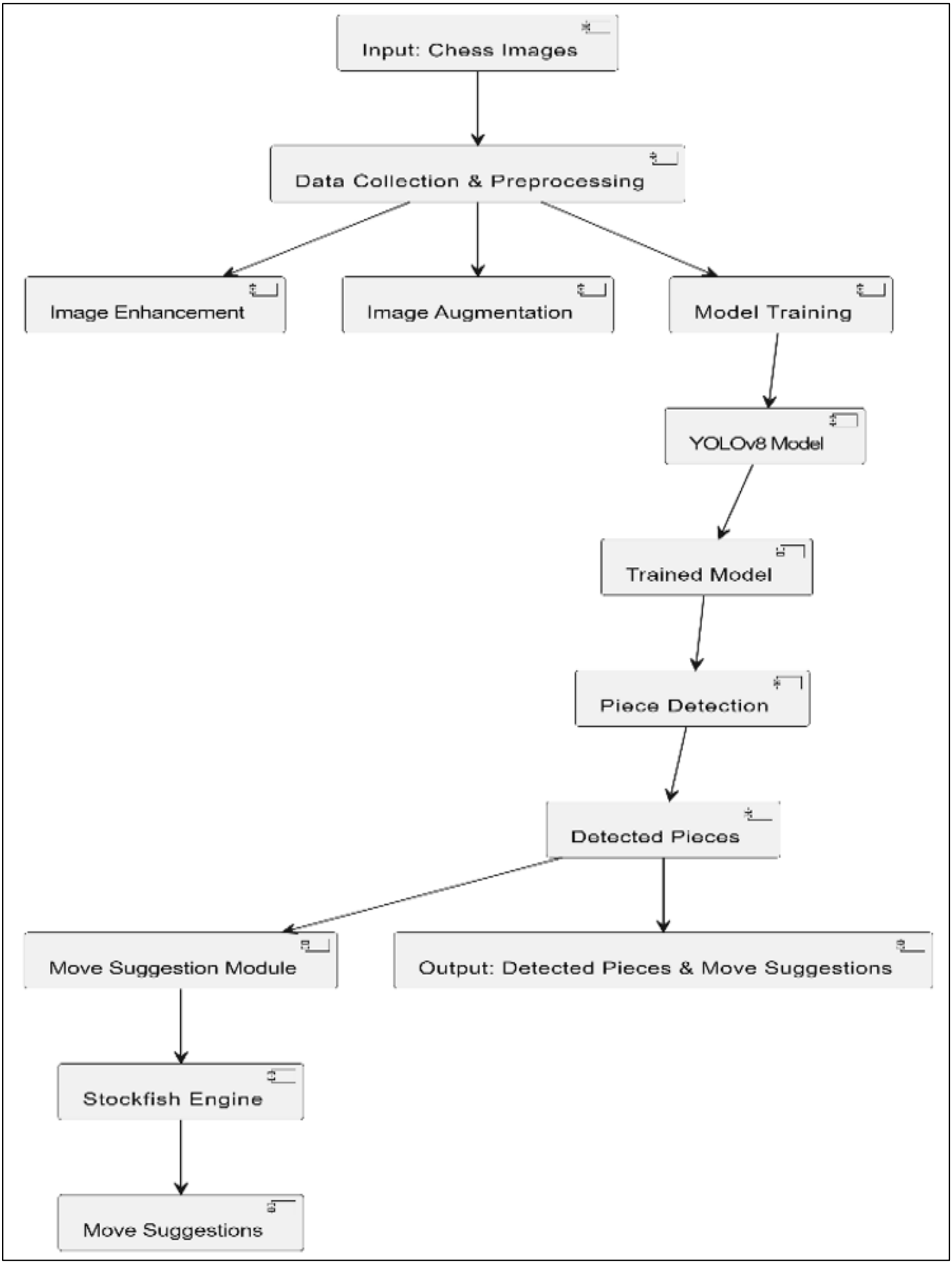

Fig 1 System Architecture Diagram

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

➢ *System Architecture Diagram*

- **Purpose:** It demonstrates how your system works overall and what components comprise it.
- The input is chess image and output is detected pieces and integrated move suggestion.
- Data Preprocessing (images enhancement, augmentation) and basic Data Collection. Training Process (YOLOv8 model divided dataset and training). Chess Engine Integration module (how the engine recognize the pieces and suggest moves).
- **Why It's Useful:** This gives a high-level overview of the entire system, showing how each part interacts and contributes to the final output.

➢ *Data Collection*

Training was done on a large dataset which consists of 606 annotated chess images taken from Kaggle. The data contained in this dataset refers to several configurations of a chessboard, which illustrate a number of different positions and types of chess pieces. The dataset is carefully labeled for each piece in each image, specifying exactly where and who the chess pieces are. It makes it possible for the model to pick up many different scenarios and surface pieces in different setup and positions. To ensure the model's effectiveness and robustness, the dataset is divided into three parts:

- **Training Set**: This set includes 424 images of which 70 percent of the dataset, is primarily used to train the model to recognize and classify the chess pieces accurately.
- **Validation Set**: From there, this set consists of 121 images (20%) and is used during the training to evaluate how the model is doing, tweak hyperparameters, and to avoid overfitting.
- **Test Set**: This set consists of 61 images (10%) and is solely used to perform the final evaluation of the model, by means of a proper objective evaluation of the model's accuracy and generalization to unseen data. This split enables effective model training and assessment, ensuring accuracy and generalization to new data.

➢ *Data Preprocessing*

Pre-processing helps us achieve better detection model performance and resilience. During this phase we perform image normalization, image resizing and augmentation techniques to improve the efficiency of the model's learning.

- *Image Enhancement*

Image quality improvement is needed in order to accurately detect. To increase separability of chess pieces from the board some techniques such as histogram equalization and contrast stretching are used. Gaussian blurring is also used along with other methods to remove noise and distorting to improve visibility. Also, these steps help the model understand that pieces are under different lighting/viewing angles as you have a point of comparison.

- *Neural Network Training*

Since the object detection was advanced, we chose YOLOv8. The process of training includes setting the neural network architecture, choosing hyperparameter, and choosing of an appropriate loss function. For this purpose, we train our model on training set, validate on validation set to monitor performance and prevent overfitting, during training, the model learns to recognize features that distinguish chess pieces from the background. The this process is optimizing neural network weights using back propagation and gradient descent techniques [6] [9].

- *The Following Hyperparameters were Optimized for Best Performance:*

✓ **Learning Rate:** Those whose convergence speed and model accuracy were balanced in order to achieve optimal convergence.
✓ **Batch Size**: The models are selected to be efficient during training while still being stable.
✓ **Number of Epochs:** They have a determined aim of sufficient training time without overfitting.

IV. PROCESS FOR CHESS PIECE IDENTIFICATION AND NEXT MOVE RECOMMENDATION

A. Start with a Title and Overview:

A Process for Chess Piece Identification and Move Recommendation.

➢ *Step 1: Image Acquisition*

- Icon: Use icons for a camera or smartphone.
- Description: "We assume that the user captures a picture of the chessboard using a camera."
- Connector: Indicate the next step with a drawing of an arrow.

➢ *Step 2: Image Processing*

- Icon: An image or a filter icon can be used for represent image processing.
- Description: We preprocessed the captured image with normalization, resizing and enhancement.
- Connector: Attach an arrow to the next step.

➢ *Step 3: Piece Recognition*

- Icon: Show this with an AI model or neural network icon.
- Description: "Cheap pedestrian detection network, kills it with YOLOv8 model, it detects and categorizes the chess pieces in the image."
- Connector: Lead an arrow to the next step.

➢ *Step 4: Position Assignment*

- Icon: To indicate which piece is being assigned, use a chessboard icon.
- Description: "Coordinates are mapped to recognized pieces (a1, h8) etc."
- Connector: Associate an arrow with the following step.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

➢ *Step 5: Move Calculation*

- Icon: Show a chess engine or chess piece moving symbol.
- Description: "To suggest the best move for the current player, the mapped board state is analyzed by Stockfish."
- Connector: Go on an arrow to the last step.

➢ *Step 6: Result Display*

- Icon: Use a display computer icon.
- Description: "As it is a user friendly system, the recommended move is shown in a format such as 'e2 to e4'."
- Connector: To represent the cycle, add an arrow back to the start, or to indicate the end of the process.

V. COMPARATIVE ANALYSIS ACCURACY COMPARISON

Table 1 Comparative Analysis Accuracy Comparison

Chess Piece	Precision	Recall	mAP50	mAP50-95
All	0.984	0.987	0.791	0.791
Black Bishop	1.0	0.882	0.967	0.721
Black King	1.0	0.976	0.995	0.85
Black Knight	0.996	1.0	0.995	0.793
Black Pawn	0.995	1.0	0.995	0.785
Black Queen	0.977	1.0	0.995	0.794
Black Rook	0.995	1.0	0.995	0.775
Black Rook	0.995	1.0	0.995	0.775
White Bishop	0.925	1.0	0.979	0.773
White King	0.992	1.0	0.995	0.811
White Knight	0.944	1.0	0.972	0.789
White Pawn	0.997	1.0	0.995	0.799
White Queen	1.0	1.0	0.995	0.824
White Rook	0.99	0.962	0.972	0.781

A. Accuracy of Chess Piece Detention

Several metrics are used to evaluate the accuracy and effectiveness of the model when evaluating the YOLOv8 model with chess piece detection. The evaluation metrics employed include precision, recall, mAP at 50, and mAP at 50-95 IoU. They offer us each a clue about the performance of the model.

➢ *Precision*

Precision refers to the accuracy rate of positive predictions among all instances classified as positive. A precision score approaching 1.0 suggests that the model excels at correctly placing chess pieces on the board, with minimal false positive occurrences. This indicates a high degree of accuracy in the model's performance. For example, the Black Bishop was at a precision of 1.0 where the model predicted correctly for every prediction made. However, the White Bishop had a lower precision of 0.925 which means that it was wrong in some of it's predictions less than other pieces.

➢ *Recall*

Sensitivity, or recall, is the ratio of the true positives to the total actual positives. A high recall score close to 1.0 indicates that the model generally finds most of the times that a class occurs. For example, the model was able to recognize all of the instances of this piece, as evidenced by the perfect recall of 1.0 for the Black Knight. But the Black Bishop's recall was 0.882, meaning some of them weren't being detected.

➢ *Mean Average Precision (mAP)*

The overall performance of a model across various Intersection over Union (IoU) thresholds is summarized by Mean Average Precision, providing a comprehensive evaluation metric. We evaluate the model's performance on a pieces IoU threshold of 0.50 using mAP50. Overall mAP50 score of all pieces was 0.791 implying good performance level. While mAP50-95 calculates the average precision over many IoU thresholds, a more rigorous measure of model's generalization ability, it is a single metric. This overall mAP50-95 score of 0.791 gives consistent detection quality in various conditions.

➢ *Analysis of Individual Pieces*

The accuracy comparison of individual chess pieces reveals varying performance levels:

Across all metrics, Black King, and White Queen scored high because they were able to detect reliably well.

In some specific board configurations, the model again identified the Black Knight and Black Pawn as playing well, again performing with perfect records. Lower precision and recall scores revealed that they wrongly identified some White Bishop as Black Bishop, which may mean where White Bishop can be misidentified.

VI. RESULTS

➢ *Chess Piece Detection Accuracy*

As seen in Table 1, the chess pieces were detected with high accuracy by the YOLOv8 model. Evaluation of the overall performance shows that the object detection model was able to recognize different chess pieces at different conditions with precision ranging from 0.97 for all pieces.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

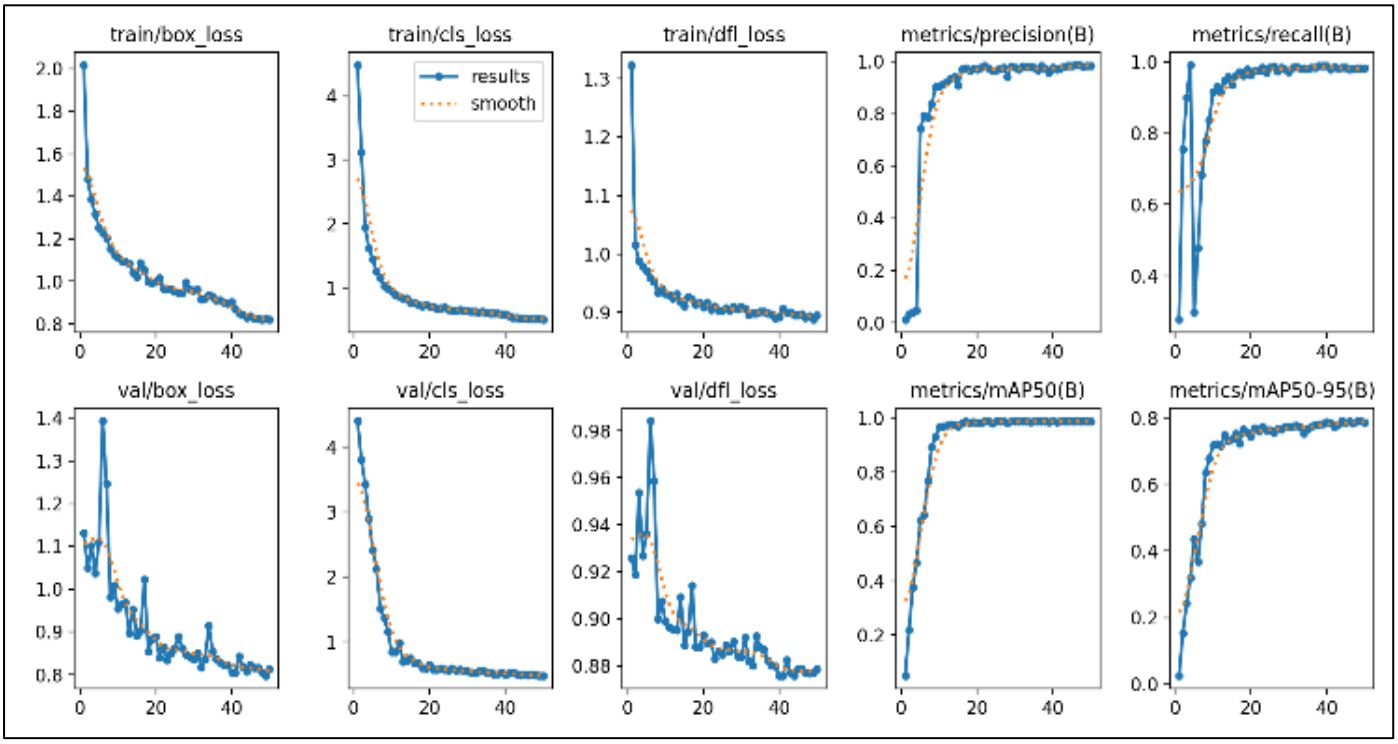

Fig 2 Result

In fig 2, This is a standard image set of training and validation metrics for training an object detection model which is likely using YOLO (You Only Look Once) or similar architecture.

- **train/box_loss**: For the training, this represents the loss regarding the bounding box regression. The smaller the value, the better the approximation of bounding box coordinate. The model improves as we train on the curve, it's a downward trend.

- **train/cls_loss**: This is the training classification loss which is measuring how good the model do to predict the class from detected objects. This means that decreasing trend tells that the model gets good at classifying objects with increasing time.

- **train/obj_loss**: The representation of objectness loss during training is this: how well the model can discriminate objects from background. Degradation indicates that object detection ability is reduced.

- **metrics/precision(B)**: It is precision, the ratio of correctly predicted positive samples to the total number of predicted positives. The model also makes fewer false positives, and a high precision. The upward of trend indicates that model's output gets better with the progression of the training.
- **metrics/recall(B)**: The recall measure is the ratio of the correctly predicted positive samples to all actual positives. A high recall means that the model is going to catch most of the true objects. Better detection performance has been achieved as training proceeds, as evidenced by the increase in the plot.

- **val/box_loss**: It is roughly similar to train/box_loss but over the validation dataset. If a value is decreased, this shows the model is generalizing well to unseen data.

- **val/cls_loss**: This is a validation data classification loss. And the lower the value, the better (because that means the model is keeping or improving its performance on data it hasn't seen during training).

- **val/obj_loss**: The model's capacity to generalize (detect objects outside train set) by observing objectness loss to the validation data.

- **metrics/mAP50(B)**: Intersection over Union (IoU) threshold 0.5 Mean Average Precision. Object detection model evaluates it as a common metric. The recall and precision balance toward detecting objects improves with increasing value. An improvement over epochs can be seen in the plot.

- **metrics/mAP50-95(B)**: This metric is mAP over the range of mIoU (0.5 to 0.95), as a more comprehensive evaluation of this metric. With a higher value, a better overall detection accuracy.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

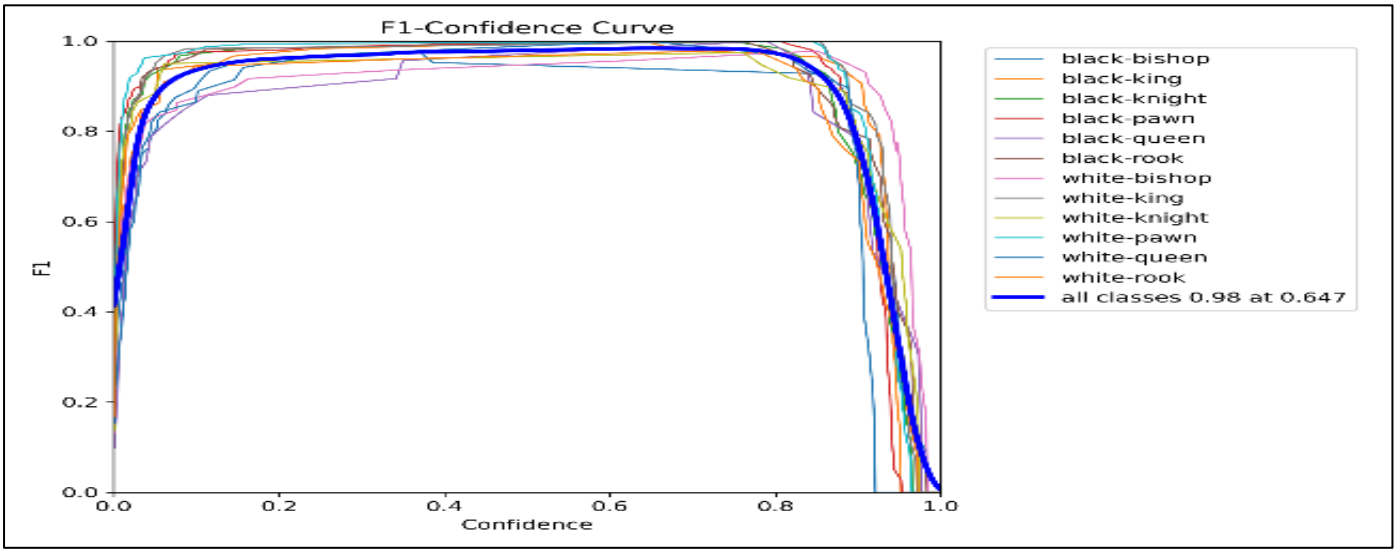

Fig 3 F1-Confidence Curve

In fig 3, show chess pieces of different classes (e.g. black bishop, white king) we see that the F1-Confidence Curve shows different classes. In this plot we consider how the F1 score varies with the model's confidence threshold for each class.

➢ *Key Components of the Curve:*

- **X-Axis (Confidence):** It represents the threshold of confidence. It is the probability for the model to classify a detection as positive, which is the minimum. Values lie in between 0 (low confidence) and 1 (high confidence).
- **Y-Axis (F1 Score):** The F1 score is the harmonic mean of precision and recall, allowing a balance. It is a number from 0 (worst) to 1 (best).

➢ *Understanding the Curve:*

The F1 score is plotted as a function of the confidence threshold for each class, e.g. "black-bishop" or "white-pawn", represented by one colored line.

The overall F1 score over all classes is shown by the thicker blue line, annotated "0.98 at 0.647." Thus, the best F1 score (0.98) is obtained by using the confidence threshold of 0.647.

For the most part, they begin lower, escalate swiftly with confidence (there is a rough balance in precision and recall) and then drop off at high confidence values due to a decline in the number of true positives.

➢ *Interpretation:*

A high F1 scores means that the model is doing well in striking a healthy balance between both precision (right positive predictions) and recall (ability to detect actual positives).

The close packing of the curves suggests that the model behaves fairly consistently over the different classes.

A threshold, like 0.647, might be chosen for practice purpose as it provides a good balance between achievement of the F1 score and reducing false positives at the same time to facilitate a good detection of the chess pieces.

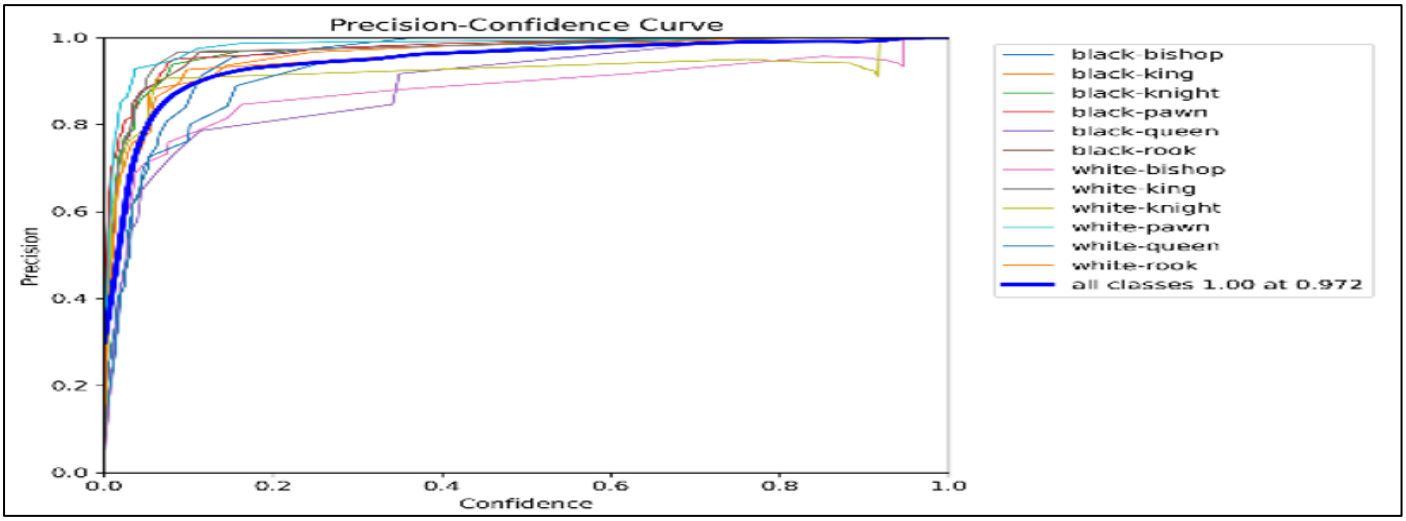

Fig 4 Percision-Confidence Curve

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

In fig 4, looks from what appeared to be a classifiers model, trying to classify chess pieces from input data (likely a Precision Confidence curve). Let me break down what this graph represents:

- **Precision (Y-axis):** Precision is the ratio of true positive predictions (right classifications) to the sum of true positive predictions and false positive predictions (total positive predicted). However, models that are higher precision makes the model more accurate when predicting a specific class.
- **Confidence (X-axis):** The model is more certain of its predictions the higher its confidence. The higher the confidence score the closer to 1, the more confident the model is with its prediction.

Lines for each chess piece: Red curves correspond to the precision for each given class of chess piece at different confidence levels; blue and green curves are the same for different classes of chess piece and red for different confidence levels. Say, black bishop, white pawn, white queen, etc would all have curves. In the legend, the names are color coded and the curves show respectively how the precision changes with the increasing confidence score.

Bold blue line ("all classes"): This thick blue line is the model performance across all classes. It's precision is 0.972 at 1.00 at 0.972, meaning it predicts 1.00 precision when the model is 0.972 confident in prediction.

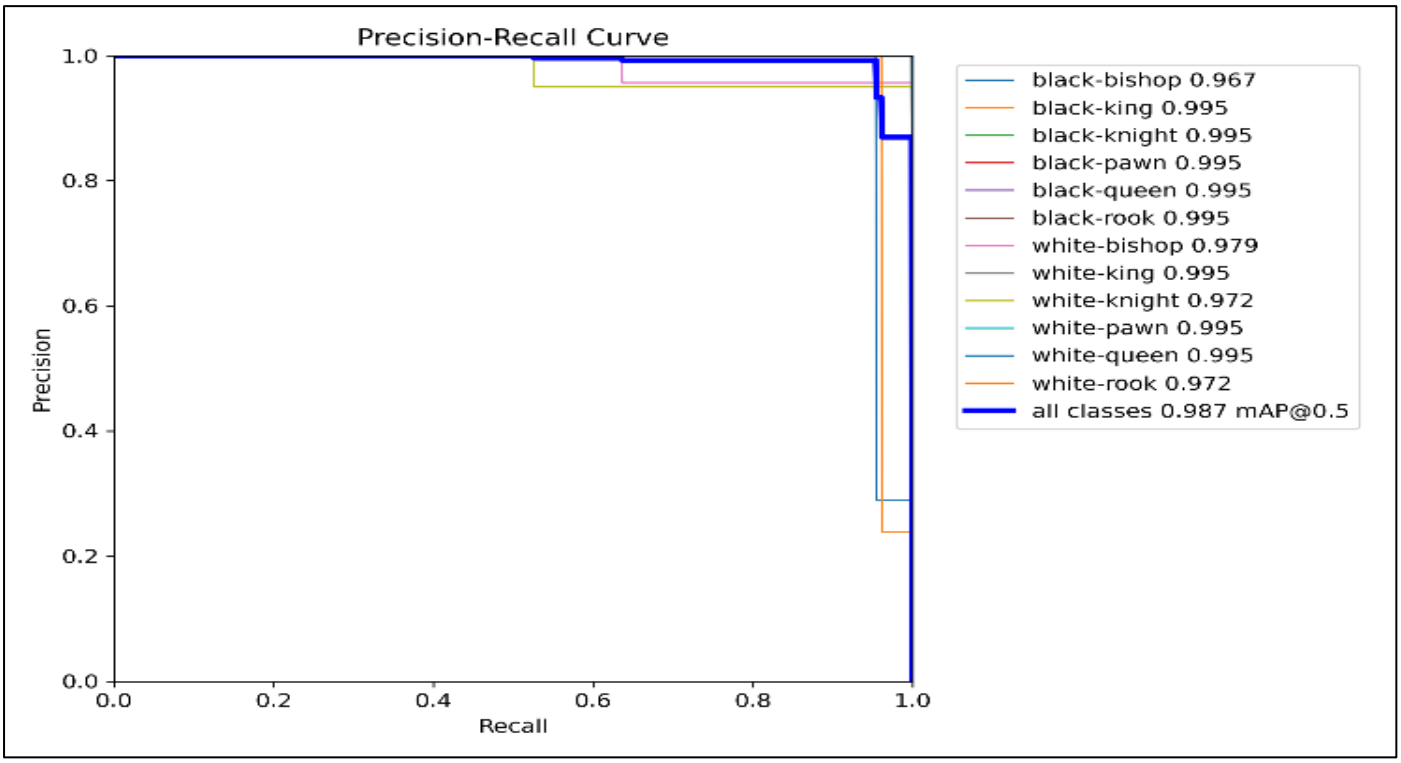

Fig 5 Precision-Recall Curve

This is a Precision and Recall Curve for different chess piece classes that visualizes the tradeoff of precision versus recall at different thresholds. Let me explain its components:

- *Observations:*

Across various recall values (almost flat lines for precision=1.0), precision is still extremely high. This implies that the model exhibits good overall performance even as recall increases, and maintains high precision. At higher recall levels, there are some slight dips (e.g. black bishop – white knight) at which points it seems the model may trade away a little precision for reaching more positives.

- *Detection Images:*

These are various chessboard photographs, some with pieces on them. There is overlaid on the images colored bounding boxes and numbers, which can be interpreted to correspond to detected chess pieces. The numbers within the boxes probably are class IDs, and different colors are types of chess (or instances of the same) pieces.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

➢ *Result Images:*

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

- **Multiple chessboards:** There are several chessboard images displayed within a kind of grid like layout, and each contains different arrangements of chess pieces.
- **Bounding boxes:** A coloured bounding box is placed around each chess piece. It seems to help me locate the pieces on the board through the bounding boxes.
- **Numbers inside the boxes:** Most likely inside the bounding boxes are the class labels or the confidence scores from a model that recognizes chess pieces. For instance a number might say 'white knight' or 'black rook'.
- **Variety of board states:** There are some boards that are empty and other boards with any number of pieces and positions confused.

VII. PERFORMANCE-METRICS

For evaluation, various indicators were used including, precision, recall, and mean average precision (mAP). Since these measurements actually mean how accurately can the model locate chess pieces and think of possible moves, these measurements are important. The following definitions are applicable:

- **Precision:** Also called positive predictive value. The ratio of incorrect positives to the total number of incorrect positives and correct positives, and correct positives relative to the total number of positive identifications.
- **Recall:** Ratio of correct positive identification to the number of actual positive (correct positive plus missed positive).
- **Mean Average Precision (mAP):** A comprehensive evaluation of model performance by the average precision from different confidence thresholds.

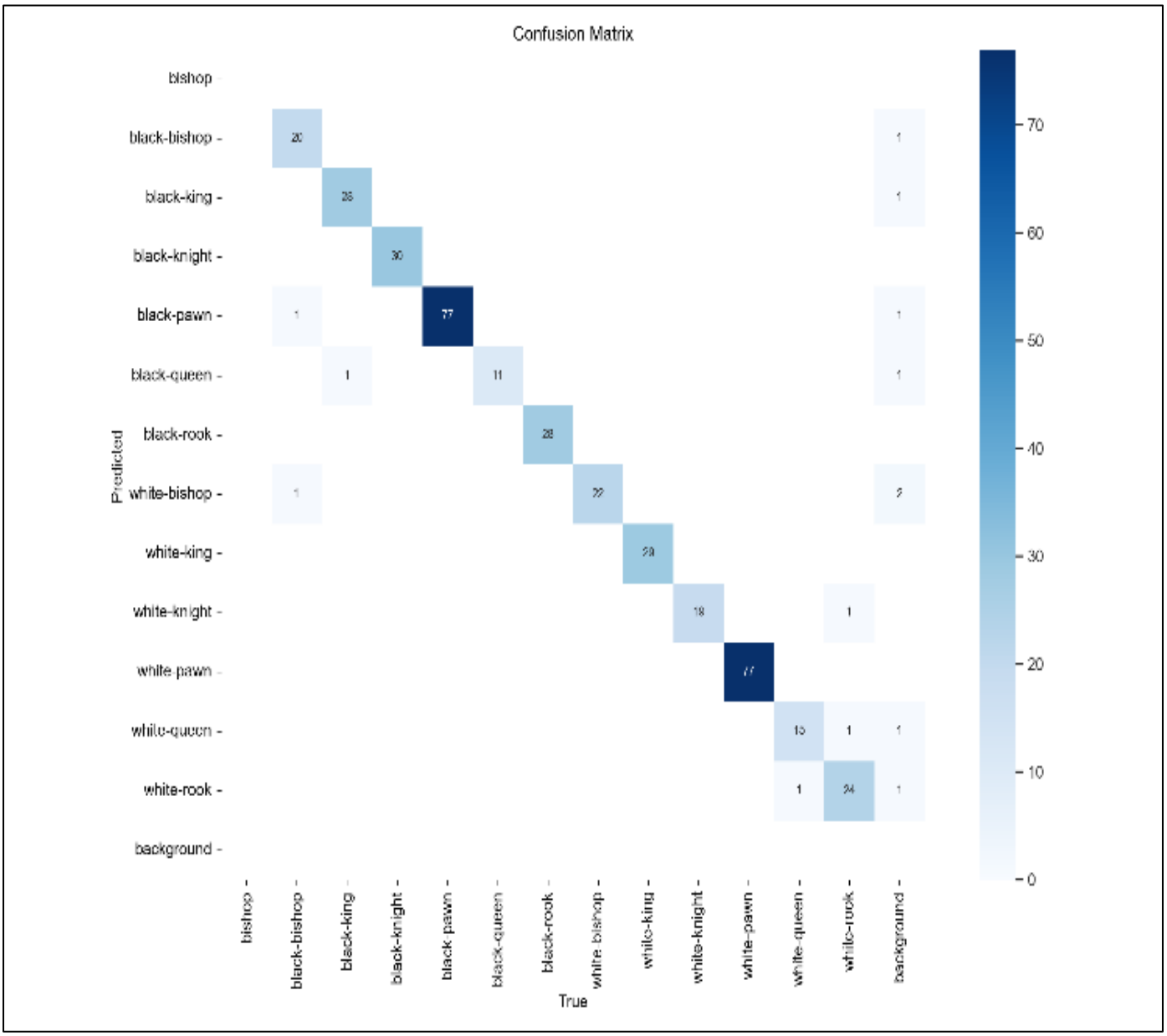

Fig 6 Training Confusion Matrix Graph

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

VIII. DISCUSSION

We found in practice that the combination of YOLOv8 and Stockfish for move suggestion is a fairly good way to show AI's capability in automating chess gameplay. The integration demonstrates how traditional games can benefit from advanced technologies with real time feedback and analysis. Practical applications of deep learning in complex environments such as chess piece detection in various conditions are presented with high accuracy. Additionally, the model was able to identify pieces in different lighting conditions and angles, so YOLOv8 framework is quite strong method.

Nevertheless, there remain challenges, particularly regarding mapping piece positions when piece obstructions or fluctuating lighting conditions make mapping them difficult. As a result, we can have misinterpretations from detection model which can decrease the overall reliabilty of system. Future studies should focus on developing methods meant to increase the detection capabilities. It could be to implement advanced image processing such like adaptive histogram equalization or use of infrared to handle problems such as lighting. Other data sources, like integrating video feeds or employing 3D modelling, could also greatly lift detection accuracy.

The goal is to get closer to a more reliable and durable system, that brings a simple and complete solution for chess analysis. That could mean better algorithm refinement to more effectively handle edge cases so if even under bad circumstances the system stays high performance. The growth of AI means more and more that there's potential for AI solution to help automate suggestions and analyses to help make game play more engaging and strategic. In the end, this paper hopes to be a positive contribution to the broader field of AI in gaming and help drive future exploration into using intelligent systems to elevate the ordinary.

IX. SCOPE OF RESEARCH

This research lays a baseline for the use of deep learning models to identify chess pieces accurately and formulate strategic moves. Using YOLOv8 for object detection and incorporating powerful chess engines such as Stockfish, we've showed that AI could add value to gameplay. Their findings indicate how this research can be extended to other board games and sports.

For example, this technology could be used on the checkers, Go, or any complex sports analytics games for real time analysis. Thanks to their versatility deep learning models can be trained on many datasets which can then learn the unique dynamics and strategies of all different games much more extended than just chess.

In addition, due to its potential use as an educational system, the system has considerable potential for demonstrating the basic principles of chess to novices of the discipline. With automated analysis and tailored recommendations, learners can learn through insights into effective strategies and appropriate tactical choices which helps them to overcome their understanding of the game quickly. The combination of this dual purpose system could make for an interactive learning environment where users can play against AI, getting feedback on their moves, and improve at both skill and engagement.

As AI technology foretells bigger and better things to come, this research may need to expand its scope to encompass user interface design innovation, which makes it user friendly and user appealing to everyone. Gamification, Real Time Feedback and personalized Learning Paths could make for a richer experience for skill level users. As this research ultimately lays the groundwork for exploring the meeting of AI, gaming and education, it places emphasis on the transformative power of intelligent systems to augment traditional gameplay and learning.

X. FUTURE SCOPE

Future developments towards greater ability of the model to recognize pieces when partially occluded from a wider range of viewing angles might be undertaken. Reinforcement learning could be added to the system to help it self improve over time. On top of this, much more improve the detection accuracy 3D camera data will do this.

➢ *Additional Areas for Future Research Include:*

- **Dataset Expansion:** To increase model robustness, more diverse scenarios and piece arrangements are added.
- **User Interaction:** Making interface so easy for users to navigate and interact with the system so much that they don't have to look at the interface to use the system.
- **Real-Time Analysis:** To make it easier to do fast analysis of any kind during live games.

XI. CONCLUSION

In this study, we prove the effectiveness of using advanced AI techniques for automatic chess piece detection and providing move suggestions. Integration of Stockfish chess engine with YOLOv8 model paves the way for the deployment of artificial intelligence towards real time game analysis. By using the combination of powerful detection on the robust objects and critical value estimation of the move, the system not only enhances player interaction, but also expands the number of people who play game.

The findings of this thesis suggest that the applications in the educational and competitive chess settings are practical. For those starting to learn the system is a great thing because it allows you to study your gameplay to develop strategic decision making processes. The learning of complex concepts or skill improvement can get its curve smoothed out using automatic analysis and suggestions. Players can use this technology to give out more to prepare for the matches by telling the opponent's strategy and what's the best response to that.Finally, developed methods in this research have linear scaling with the board size, enabling them for applications beyond chess. To this end, this technology can be adapted for use with other board and card games as well as sports in future

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr133

studies to explore a generic system capable of real time analysis on many more sports. Certainly, as AI continues to evolve, clever systems will get cleverer, and clever systems will continually get cleverer on a gaming system as intelligent systems blend in seemingly More generally, this work lays the groundwork for further development in the area of AI helped gameplay, further evidence as these sorts of technologies will dramatically change everyday gaming and education. By bridging the gap between the user experience parts of this and the more advanced algorithmic parts, we're bringing players a new age of interactive, engaging, yet approachable gameplay whether the person playing brings much skill or even none. The capabilities described above offer an opportunity for continuing exploration of this field as we start to use them for much more innovative and enjoyable gaming experiences.

REFERENCES

[1]. Esteva, A., et al. (2017). "Dermatologist-level classification of skin cancer with deep neural networks," Nature, vol. 542, no. 7639, pp. 115-118, 2017.
[2]. LeCun, Y., Bengio, Y., and Hinton, G. (2015). "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444.
[3]. A foundational overview of deep learning principles that serve as the basis for developing models capable of recognizing chess pieces and suggesting moves.
[4]. Silver, D., et al. (2017). "Mastering the game of Go without human knowledge," Nature, vol. 550, pp. 354-359.
[5]. Redmon, J., et al. (2016). "You Only Look Once: Unified, real-time object detection," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788.
[6]. Rajpurkar, P., et al. (2017). "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning," arXiv preprint arXiv:1711.05225.
[7]. McCarthy, J. (2007). "The 50th anniversary of the Dartmouth Conference," AI Magazine, vol. 28, no. 1, pp. 67-70.
[8]. Goodfellow, I., et al. (2014). "Generative adversarial nets," Advances in Neural Information Processing Systems, vol. 27.
[9]. Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp. 1097-1105.
[10]. Bottou, L. (2010). "Large-scale machine learning with stochastic gradient descent," Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT), pp. 177-186.
[11]. Redmon, J., and Farhadi, A. (2018). "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767.