Report: Chess Piece Detection and Move Analysis System

Likhita Yerra

March 26, 2025

1 Description of the Computer Vision Problem

This project tackles the intricate computer vision challenge of detecting and interpreting chess moves from video footage, a task that bridges object detection, motion tracking, and sequence analysis. The system must identify 12 unique chess piece types—pawn, rook, knight, bishop, queen, and king, each in black and white—across an 8x8 chessboard, track their movements frame-by-frame, and translate these into standard algebraic notation (e.g., "e4", "Nf3", "Qxd5"). This involves overcoming obstacles like varying camera perspectives (top-down, angled), inconsistent lighting (indoor vs. outdoor), and occlusions from players' hands or shadows. Additionally, the system distinguishes intentional moves from incidental adjustments, such as a player nudging a piece without completing a move. Beyond technical detection, it generates humorous audio commentary using AI, enhancing its appeal for chess learners, enthusiasts, or casual viewers. This dual purpose—precision in move recognition and creative user engagement—elevates the problem's complexity, requiring a robust pipeline that integrates vision and natural language processing seamlessly.

2 Description of the Computer Vision Algorithm or Methodology

The solution leverages YOLOv8 (You Only Look Once, version 8), a state-of-the-art object detection model renowned for its real-time performance and accuracy. The small variant (`yolov8s.pt`) is fine-tuned for this task, balancing computational efficiency with detection quality. The methodology unfolds as follows:

- **Training Phase**: YOLOv8 is trained on a labeled chess dataset using supervised learning. Images are resized to 640x640, and the model outputs bounding boxes and class probabilities for each piece. Training occurs in Google Colab with a Tesla T4 GPU, using early stopping when mAP50 reaches 0.99 or higher, ensuring optimal performance.
- **Move Detection**: A custom `ChessMoveDetector` class processes video frames: (1) detects pieces with YOLOv8, (2) maps bounding box centroids to chessboard squares (e.g., "a1" to "h8"), (3) compares consecutive frames to infer piece displace-

ments, and (4) validates moves with the `python-chess` library to ensure legality (e.g., rejecting a pawn moving diagonally without capture). Moves are output in SAN.

- **Commentary Generation**: Detected moves are sent to OpenAI's GPT-4, which generates witty, context-aware commentary (e.g., "That pawn to e4 is bolder than a knight in shining armor!"). This text is converted to audio using gTTS (Google Text-to-Speech).
- **User Interface**: A Streamlit app provides an interactive platform, allowing users to upload videos, view annotated outputs, and download commentary audio.

This pipeline combines YOLOv8's speed, chess-specific logic, and AI-driven narration into a cohesive system.

3 Description of the Dataset

The dataset, sourced from Roboflow as `Chess-Piece-Detection-2`, is tailored for chess piece recognition. It consists of numerous images—likely hundreds to thousands—depicting chessboards with annotated pieces across 12 classes (6 per color). Annotations are in YOLO format, specifying bounding boxes (x, y, width, height) and class labels, and the dataset is split into training, validation, and test sets, accessible via Roboflow's API. Stored at `/content/Chess-Piece-Detection-2/` during training, it supports 640x640 preprocessing by YOLOv8. While exact details are unavailable, its diversity—spanning board styles (wooden, plastic), lighting conditions (bright, dim), and perspectives (top-down, angled)—enabled high accuracy (mAP50: 0.991) after 11 epochs. The dataset's robustness suggests potential inclusion of real gameplay images, possibly augmented with synthetic data.

4 Obtained Results

The system achieved strong results across its components:

- **Training**: YOLOv8s training concluded after 11 epochs with an mAP50 of 0.991, reflecting near-perfect piece detection. Model weights (`best.pt`, `last.pt`) and logs were saved to `/content/drive/My Drive/Chess_Model/`, ready for deployment.
- **Move Detection**: The `ChessMoveDetector` processed a sample video (`How To Play The Queen's Gambit.mp4`), producing a move sequence (e.g., "e4", "e5", "Nf3") and an annotated output (`detected_moves.mp4`). Move validation ensured legal outputs, though accuracy depends on video clarity and frame rate.
- **Application**: The Streamlit app successfully displayed the annotated video, move list, and GPT-4-generated audio commentary, offering an engaging user experience. Minor issues, like an mAP50 access error and file path mismatches, were resolved.

While effective on the sample, broader testing across diverse videos is needed to confirm reliability.

5 Possible Future Directions

Future enhancements could include:

1. **Real-Time Processing**: Optimize for live video (e.g., webcam) using YOLOv8 nano or frame subsampling for real-time coaching.
2. **Dataset Expansion**: Incorporate synthetic images (via 3D rendering) or crowd-sourced gameplay videos to improve generalization.
3. **Advanced Analysis**: Integrate Stockfish to evaluate move quality, identifying blunders or brilliant plays with detailed feedback.
4. **Multi-Player Support**: Add hand tracking or multi-camera setups to detect player turns and interactions.
5. **Deployment**: Package as a standalone desktop or web app, with mobile compatibility for wider access.

These advancements could evolve the system into a comprehensive chess analysis and learning tool.