

---

# PROJECT PROPOSAL: COUNTING CARD SYSTEM FOR BLACKJACK

---

**Davide Baggio 2122547**

Department of Computer Engineering  
University of Padua

**Zoren Martinez**

Department of Computer Engineering  
University of Padua

**Francesco Pivotto**

Department of Computer Engineering  
University of Padua

## Introduction

In the realm of casino gaming, ensuring fair play and detecting potential cheating behaviors is of paramount importance. One such method of cheating that has persisted over time is card counting in the game of Blackjack. While not inherently illegal, card counting can give players a significant advantage over the house, leading to potential revenue losses for casinos. As a response to this challenge, this project proposes the development of an automated card counting detection system using C++, OpenCV, and potentially deep learning techniques.

The core objective of the project is to build a real-time video analysis system capable of observing and interpreting the flow of cards dealt in Blackjack games. By leveraging the capabilities of OpenCV for image processing and card recognition, combined with the power of C++ for performance and control, the system will track the sequence of cards on the table. Additionally, machine learning or deep learning algorithms may be integrated to analyze patterns in player behavior and betting in correlation with card distributions, flagging potential instances of card counting.

This automated system has practical implications for modern casinos, offering an intelligent and scalable solution to monitor gameplay, support security teams, and uphold the integrity of the game without intrusive surveillance. Through this project, we aim to contribute to the development of smart surveillance tools in gaming environments using cutting-edge computer vision and AI technologies.

## Visual Output and Card Value Representation

To facilitate intuitive monitoring and real-time analysis, the system will process the video feed and overlay bounding boxes around detected playing cards. Each bounding box will be color-coded based on the card's value in the Hi-Lo counting system:

- **Green** bounding boxes will indicate cards valued at +1 (typically 2 through 6).
- **Blue** bounding boxes will indicate neutral cards with a value of 0 (typically 7 through 9).
- **Red** bounding boxes will indicate high-value cards that subtract from the count, assigned a value of -1 (typically 10, face cards, and Aces).

This visual overlay will not only aid in debugging and performance evaluation but will also allow casino personnel to interpret the system's counting logic in real time. By observing the count trend and the corresponding card values, operators can better assess whether a player's betting behavior aligns suspiciously with card distribution—potentially identifying card counting activity.

This color-coded visualization enhances transparency and provides immediate visual cues without requiring technical expertise to understand the underlying card tracking mechanism.



Figure 1: (a) Base image

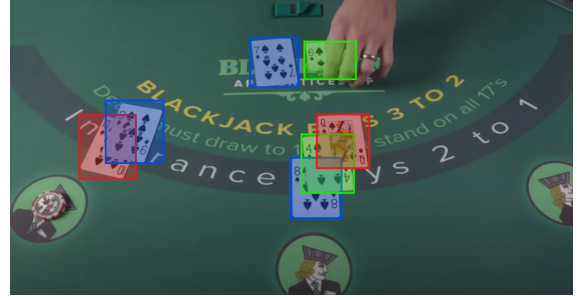


Figure 2: (b) Detection output

## Dataset and Real-World Data Sources

To train and evaluate the card detection and recognition system, we will utilize a combination of a structured dataset and real-world video footage:

### The Complete Playing Card Dataset

We will employ the Complete Playing Card Dataset [1] available on Kaggle. This dataset comprises approximately 50 images for each playing card, including Jokers. Each card serves as a distinct class, providing a diverse set of images that capture various positions and rotations of the cards. This diversity is crucial for training a robust model capable of recognizing cards under different orientations and perspectives.

### Real-World Video Footage

To simulate real-time scenarios and evaluate the system's performance in practical environments, we will use publicly available Blackjack gameplay videos sourced from platforms like YouTube. These videos offer a variety of real-world conditions, including different lighting, backgrounds, and card arrangements, which are essential for testing the system's robustness and adaptability.

By combining the structured dataset with real-world video footage, we aim to develop a comprehensive card counting detection system that performs reliably in both controlled and dynamic environments.

## Performance Measurement

To ensure the reliability and practicality of the proposed card counting detection system, a comprehensive evaluation strategy will be implemented. The performance of the system will be assessed across several key metrics, focusing on both the card recognition accuracy and the effectiveness of the card counting logic in real-world conditions.

### Card Detection and Recognition Accuracy

The first layer of performance measurement involves evaluating how accurately the system detects and classifies individual playing cards:

- Precision and Recall: These will be measured to assess the system's ability to correctly identify cards (true positives) while minimizing false detections.
- F1-Score: A harmonic mean of precision and recall to provide a balanced evaluation metric.
- Confusion Matrix: To visualize misclassifications across different card types and values.

Testing will be performed on:

- Held-out images from the annotated dataset (to test generalization).
- Frames extracted from real gameplay videos, where ground truth labels are manually annotated for evaluation purposes.

### **Accuracy of Card Counting Logic**

The system's ability to maintain an accurate running count will be verified by comparing its output with manually computed counts based on recorded game sessions. Metrics include:

Running Count Error Rate: Average deviation of the system's count from the ground truth over time.

Event-Based Accuracy: Correct detection of count transitions (e.g., from neutral to positive count, or negative to neutral).

### **References**

- [1] Pradip Shah Jay. The complete playing card dataset. Kaggle, 2020. URL: <https://www.kaggle.com/datasets/jaypradipshah/the-complete-playing-card-dataset>.