

# Visión por Ordenador I

Ingeniería Matemática e Inteligencia Artificial

Universidad Pontificia Comillas ICAI

---

## Padel AI System: Control de Seguridad y Arbitraje Autónomo

---

### Proyecto Final

#### Autores:

Jorge Carnicero Príncipe

Andrés Gil Vicente

#### Fecha:

10 de enero de 2026

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Metodología</b>	<b>2</b>
2.1. Calibración de la cámara . . . . .	2
2.2. Diagrama de bloques del sistema . . . . .	3
2.3. Diagrama de bloques del sistema . . . . .	3
2.4. Secuencia de operaciones . . . . .	4
2.5. Sistema de seguridad . . . . .	5
2.5.1. Detección de patrones . . . . .	5
2.5.2. Extracción de información . . . . .	5
2.6. Sistema propuesto: Tracker y Videoarbitraje . . . . .	5
2.6.1. Tracker (Seguimiento) . . . . .	6
2.6.2. Detección de Jugador y Lógica de Juego . . . . .	6
<b>3. Interfaz Interactiva y Visualización (Streamlit)</b>	<b>6</b>
<b>4. Resultados</b>	<b>7</b>
<b>5. Futuros Desarrollos</b>	<b>7</b>
<b>6. Anexo</b>	<b>8</b>
6.1. Seguridad . . . . .	8
6.2. Tracking y Pádel . . . . .	12

## 1. Introducción

El presente proyecto tiene como objetivo documentar el desarrollo e implementación de un sistema integral de visión por ordenador aplicado al deporte del pádel, denominado *Padel AI System*. El alcance del proyecto abarca dos módulos fundamentales: un sistema de control de acceso basado en el reconocimiento de patrones geométricos (módulo de seguridad) y un sistema de arbitraje autónomo que detecta si un saque o servicio de pádel cumple con las reglas establecidas en dicho deporte (módulo de tracking).

El sistema propuesto utiliza una única cámara como sensor de entrada para procesar imágenes en tiempo real, permitiendo validar saques y detectar botes mediante técnicas de visión clásica combinadas con modelos ligeros de detección de pose (YOLOv8). La interfaz de usuario se ha desarrollado a través de la librería de *Streamlit*, proporcionando así al usuario un entorno amigable para la configuración de los parámetros necesarios, utilización de las distintas funcionalidades del sistema y visualización de resultados.

## 2. Metodología

### 2.1. Calibración de la cámara

La calibración de la cámara del sistema se aborda desde dos perspectivas. Por un lado, se ha desarrollado la calibración habitual que se ha enseñado en las prácticas de laboratorio, que nos proporciona los coeficientes de distorsión, los parámetros intrínsecos y los parámetros extrínsecos asociados a nuestro sistema. Por otro lado, también se ha querido incluir un pequeño sistema de calibración del espacio de color, necesario para la parte de la segmentación del color de la pelota de pádel, que además es fácilmente ajustable por el usuario en tiempo real.

Para la calibración de la cámara, se ha utilizado el método del tablero de ajedrez (*chessboard*) para obtener todos los coeficientes y parámetros ya mencionados. A continuación mostramos los resultados de este proceso de calibración realizado offline:

- **Error medio de reproyección (RMS):** 1.6166
- **Matriz de parámetros intrínsecos ( $K$ ) y coeficientes de distorsión ( $D$ ):**  
calculados offline.

$$K = \begin{bmatrix} 1477,08 & 0,00 & 986,46 \\ 0,00 & 1481,48 & 672,36 \\ 0,00 & 0,00 & 1,00 \end{bmatrix} \quad D = [0,067 \ 0,527 \ -0,030 \ 0,001 \ -2,469]$$

- **Parámetros extrínsecos  $[R]$ :** como hay muchos, se muestran a modo de ejemplo la primera y la última matriz, indicando solo la primera y la última fila:

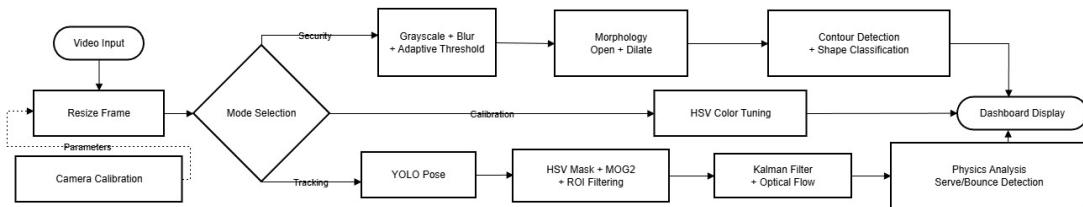
$$[R]_{\text{primera}} = \begin{bmatrix} -0,93 & -0,34 & 0,04 & 126,2 \\ \vdots & \vdots & \vdots & \vdots \\ 0,16 & -0,33 & 0,92 & 523,5 \end{bmatrix} \quad [R]_{\text{última}} = \begin{bmatrix} 0,95 & 0,24 & -0,17 & -177,9 \\ \vdots & \vdots & \vdots & \vdots \\ 0,29 & -0,86 & 0,40 & 728,6 \end{bmatrix}$$

No obstante, como en este caso para el sistema que se ha desarrollado se ha utilizado una cámara de móvil, por lo que al no ser una cámara fisheye ni gran angular, no se aprecia prácticamente que haya distorsión. Es por eso que se ha decidido no corregir dicha distorsión, ya que lejos de mejorar la calidad de la retransmisión de la cámara, la empeoraba.

Adicionalmente, como se ha mencionado, el sistema incluye un módulo de calibración de color en tiempo real (integrado en el script `complete_padel_system_app.py`) que permite ajustar dinámicamente los rangos HSV para segmentar la pelota de pádel según las condiciones de iluminación de cada momento.

## 2.2. Diagrama de bloques del sistema

La arquitectura del software sigue un flujo lineal con ramificación condicional según el estado de la aplicación (*Security Mode* vs *Tracker Mode*).



**Figura 1:** Diagrama de bloques de la arquitectura del sistema Padel AI.

## 2.3. Diagrama de bloques del sistema

El sistema se articula mediante una arquitectura modular gestionada por una máquina de estados. Los componentes principales son:

- **Video Acquisition Module:** Captura asíncrona de imágenes mediante hilos.
- **System Router:** Controlador de flujo que conmuta entre los modos de Seguridad, Calibración y Tracking.
- **Security Preprocessor:** Módulo de visión clásica (umbralización adaptativa y morfología) para resaltar formas geométricas.
- **Shape Classifier Engine:** Algoritmo de decisión basado en propiedades geométricas (vértices, circularidad, *convex hull*) para validar contraseñas visuales.

- **YOLO Pose Estimator:** Para la extracción de puntos clave del esqueleto del jugador.
- **Hybrid Ball Tracker:** Sistema de detección de la pelota mediante fusión de color (HSV), movimiento (MOG2) y flujo óptico.
- **Kalman Filter:** Estimador recursivo para predecir la posición de la pelota y corregir occlusiones momentáneas.
- **Physics Logic Core:** Motor de reglas que evalúa aceleraciones y posiciones relativas para arbitrar saques y botes.
- **Streamlit Dashboard:** Interfaz web para configuración, visualización en tiempo real y presentación de veredictos.

**Conexión:** La entrada de vídeo alimenta al System Router, que deriva el flujo hacia la rama de Security para el control de acceso o hacia la rama de Padel AI (donde operan en paralelo el Pose Estimator y el Ball Tracker) antes de unificar los datos en el *Logic Core* para su visualización final.

## 2.4. Secuencia de operaciones

El flujo de procesamiento por cada fotograma sigue la siguiente secuencia lineal:

1. **Inicialización y Captura:** Arranque de la cámara en hilo independiente y lectura del frame actual.
2. **Preprocesado General:** Redimensionado de la imagen a la resolución de trabajo para optimizar el rendimiento.
3. **Ramificación de Seguridad:**
  - a) Conversión a escala de grises y aplicación de desenfoque gaussiano.
  - b) Umbralización adaptativa y operaciones morfológicas (apertura y dilatación) para consolidar bordes.
  - c) Extracción de contornos y cálculo del Convex Hull.
  - d) Aproximación poligonal y clasificación de la forma (Triángulo, Cuadrado, Pentágono, Círculo, etc.).
4. **Ramificación de Padel AI:**
  - a) **Inferencia de Jugador:** Ejecución de YOLOv8 (cada  $N$  frames) para obtener coordenadas de cintura y pies.
  - b) **Segmentación de Pelota:** Generación de máscaras mediante rango de color HSV y sustracción de fondo (MOG2).
  - c) **Filtrado Espacial:** Aplicación de máscara de zona de interés basada en la posición del jugador.
  - d) **Predicción y Corrección:** Estimación de posición con filtro de Kalman y refinamiento con flujo óptico (Lucas-Kanade).

- e) **Lógica de Eventos:** Cálculo de aceleración de la pelota y comparación de altura respecto a la cintura para validar saques.
  - f) **Detección de Bote:** Análisis de cambios en la trayectoria vertical respecto a la línea de suelo estimada.
5. **Generación de Evidencias:** Captura y guardado automático de imágenes (“Foto Finish”) si se detecta un evento de juego relevante, es decir, un nuevo saque.
  6. **Renderizado:** Superposición de gráficos (esqueleto, trazas, cajas de texto) sobre el frame original.
  7. **Salida:** Codificación JPEG y envío del buffer de vídeo a la interfaz de usuario.

## 2.5. Sistema de seguridad

El módulo de seguridad restringe el acceso al tracker mediante una contraseña visual basada en una secuencia de 4 formas geométricas.

### 2.5.1. Detección de patrones

La detección se basa en el análisis de contornos y la aproximación poligonal aplicada sobre el Convex Hull del objeto, lo que otorga robustez ante imperfecciones en dibujos a mano alzada. Se utilizan las siguientes reglas geométricas tras filtrar por área mínima y solidez:

- **Triángulo:** 3 vértices.
- **Cuadrado/Rectángulo:** 4 vértices. Se diferencian por el *aspect ratio*.
- **Círculo/Pentágono:** 5 o más vértices. Se distinguen por factor de circularidad.
- **Línea:** *Aspect ratio* extremo ( $> 6,5$ ) y baja solidez.

### 2.5.2. Extracción de información

El sistema almacena en memoria una lista con los patrones detectados secuencialmente que se han decidido usar como contraseña. El usuario debe mostrar a la cámara la secuencia correcta (definida previamente en la fase de *Setup*). Si la secuencia de entrada coincide con la almacenada, se desbloquea el acceso al menú principal. Pero en caso de fallar algún paso de la secuencia, se muestra el mensaje de error y se reinicia el proceso de introducción de la contraseña.

## 2.6. Sistema propuesto: Tracker y Videoarbitraje

El núcleo del proyecto es un juez de silla autónomo (videoarbitraje) que combina IA y visión clásica.

### 2.6.1. Tracker (Seguimiento)

Se utiliza un enfoque híbrido para el seguimiento de la pelota:

1. **Detección:** Fusión de máscara de color HSV y sustracción de fondo para obtener el centroide de la pelota.
2. **Predicción:** Un filtro de Kalman estima la posición de la pelota en caso de oclusión momentánea o fallo en la detección visual.
3. **Refinamiento:** Se emplea flujo óptico (Lucas-Kanade) para suavizar la trayectoria visual.

### 2.6.2. Detección de Jugador y Lógica de Juego

Se integra el modelo YOLOv8-pose para detectar al jugador. A partir de los keypoints extraídos, el sistema calcula automáticamente:

- **Línea de cintura:** Promedio de la altura de las caderas.
- **Línea de suelo:** Estimada a partir de los tobillos.

**Lógica de validación del saque:** El sistema calcula la aceleración de la pelota. Al detectar un impacto (pico de aceleración) y verificar que la trayectoria no es vertical (rebote simple), compara la altura de la pelota ( $y_{ball}$ ) con la altura de la cintura ( $y_{waist}$ ).

$$\text{Veredicto} = \begin{cases} \text{VALID} & \text{si } y_{ball} > y_{waist} \text{ (impacto bajo la cintura)} \\ \text{FAULT} & \text{si } y_{ball} \leq y_{waist} \text{ (impacto sobre la cintura)} \end{cases} \quad (1)$$

## 3. Interfaz Interactiva y Visualización (Streamlit)

La visualización y control del sistema se centralizan en un dashboard interactivo desarrollado con Streamlit, diseñado para ofrecer una experiencia de usuario fluida y amigable. Esta interfaz web integra la salida de vídeo procesada con anotaciones de realidad aumentada en tiempo real:

- **Módulo de Seguridad:** Superposición del Convex Hull y etiquetado dinámico de las formas geométricas detectadas ("Triangle", "Square", etc.).
- **Módulo Tracker:** Visualización de bounding boxes para la pelota, esqueleto parcial del jugador y líneas de referencia dinámicas (cintura/suelo) para asistir en la decisión arbitral.

Para garantizar la adaptabilidad del sistema a diferentes entornos, se ha implementado un panel de configuración lateral que permite al usuario ajustar parámetros críticos en tiempo real sin reiniciar la aplicación. Entre estos ajustes se incluyen la selección de la fuente de vídeo (cámara IP o archivo), el control de calidad del streaming y la calibración fina de los umbrales de detección geométrica (área mínima, relación de aspecto y precisión poligonal).

Además de la transmisión en tiempo real, la interfaz gestiona la comunicación bidireccional con el usuario mediante un sistema de notificaciones emergentes. Estas alertas informan sobre eventos del juego (“BOUNCE”, “VALID”, “FAULT”) y confirman acciones del sistema de seguridad, como “Square Captured” o “Access Granted”. Como complemento y apoyo visual, el sistema muestra automáticamente una captura estática de alta resolución (“Foto Finish de cada saque”) en el panel lateral, proporcionando evidencia visual inmediata del momento exacto del impacto.

## 4. Resultados

La validación experimental del sistema desarrollado, realizada tanto en entornos de retransmisión en directo como mediante el análisis offline de clips de vídeo, ha arrojado resultados altamente satisfactorios. El prototipo ha demostrado ser una solución viable y de bajo coste para la digitalización del arbitraje automático. Algunos de los logros más destacados incluyen:

- **Rendimiento Real-Time:** Ejecución fluida y estable (20-25 FPS) gracias a la captura asíncrona.
- **Robustez Geométrica:** Detección fiable de patrones geométricos generados manualmente (independiente de escala/rotación).
- **Precisión Arbitral:** Seguimiento continuo de la bola y validación correcta de saques (Color + Movimiento + Kalman).
- **Usabilidad (UI):** Interfaz intuitiva con feedback visual inmediato, mensajes emergentes y generación automática de evidencias (“Foto Finish”).

## 5. Futuros Desarrollos

Aunque el sistema actual cumple con los requisitos funcionales establecidos, además de tener numerosas ampliaciones y mejoras tanto técnicas como orientadas a la visualización y la experiencia de usuario, se han identificado varias líneas estratégicas para evolucionar este prototipo hacia un producto final más robusto e incluso comercializable. Las líneas estratégicas para la evolución del prototipo podrían ser:

- **Optimización:** Migración a TensorRT/ONNX para maximizar FPS y resolución.
- **Visión 3D:** Sistema multicámara para minimizar el error de tracking.
- **Biometría:** Integración de reconocimiento facial para gestión de usuarios.
- **Analítica Avanzada:** Mapas de calor, estadísticas y métricas de juego.
- **Adaptabilidad:** Normalización para ganar robustez frente a cambios de luz.

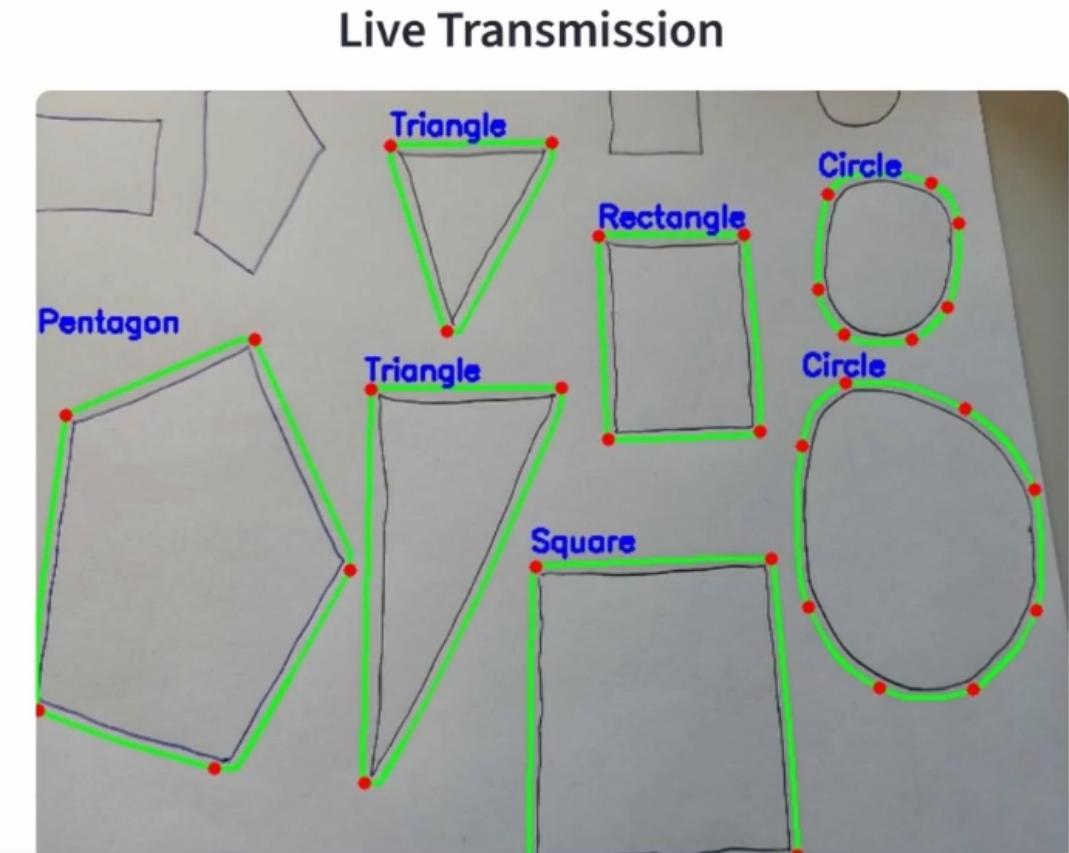
Como conclusión final, este proyecto sienta las bases sólidas para un sistema del videoarbitraje autónomo accesible para todo tipo de usuarios, demostrando el potencial de la Inteligencia Artificial aplicada al deporte del pádel.

## 6. Anexo

Debido al límite de extensión establecido para la documentación técnica del proyecto, se ha optado por incluir este anexo gráfico adicional. Su objetivo es complementar la memoria ilustrando de manera visual la interfaz de usuario, los resultados obtenidos en las distintas pruebas realizadas y el funcionamiento detallado de los distintos módulos del sistema Padel AI. A continuación, se presentan una serie de imágenes organizadas por funcionalidad.

### 6.1. Seguridad

Esta sección recopila imágenes relacionadas con el módulo de control de acceso y seguridad mediante patrones de formas geométricas, mostrando desde la interfaz general hasta casos de éxito y error en la detección.



**Figura 2:** Ejemplo de detección de múltiples formas geométricas simultáneamente

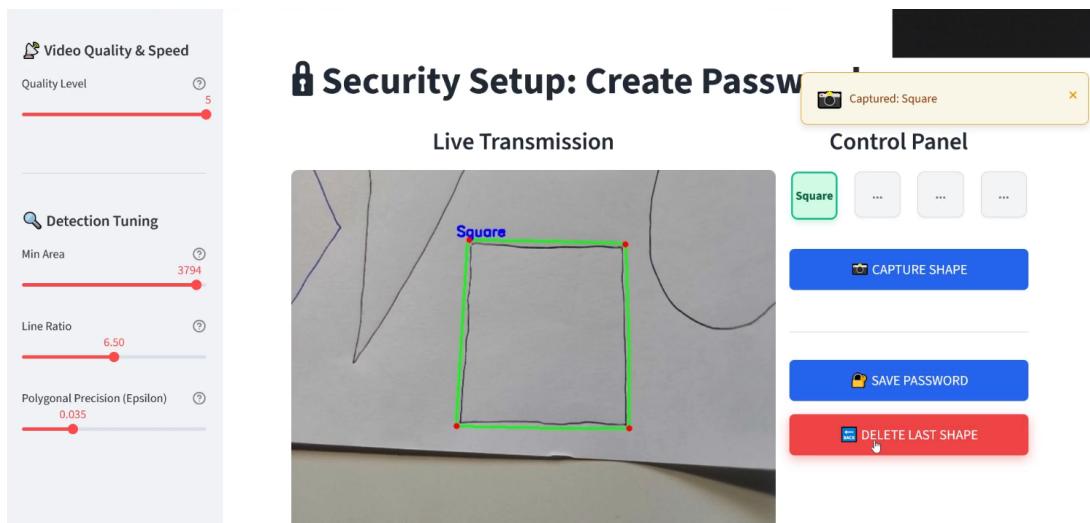


Figura 3: Interfaz general del Sistema de Seguridad

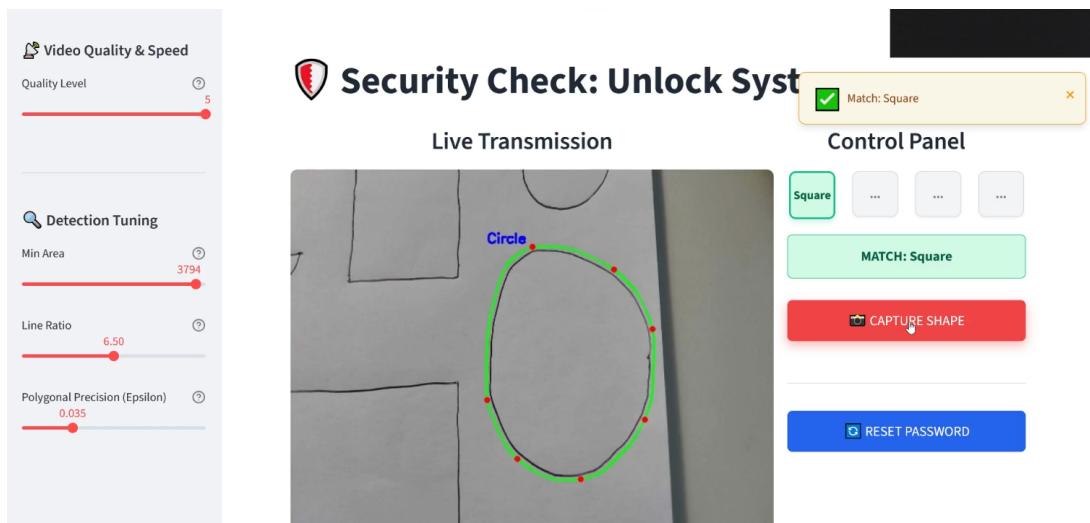


Figura 4: Detección exitosa de un patrón de contraseña

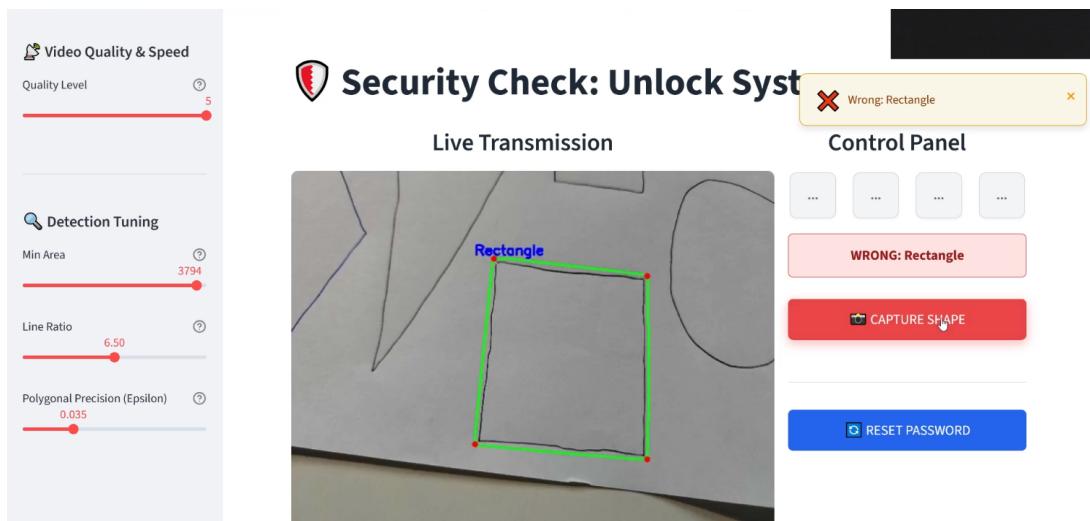
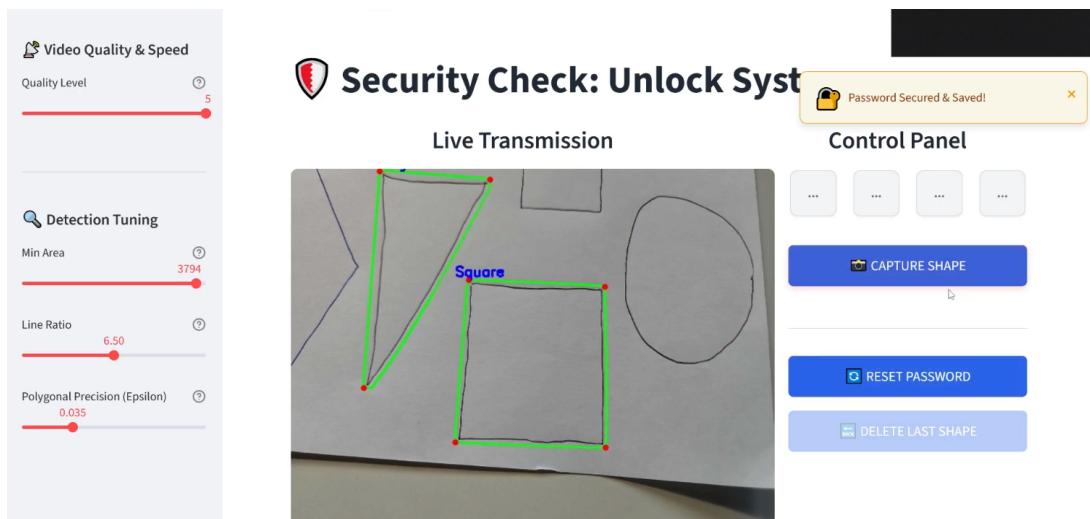
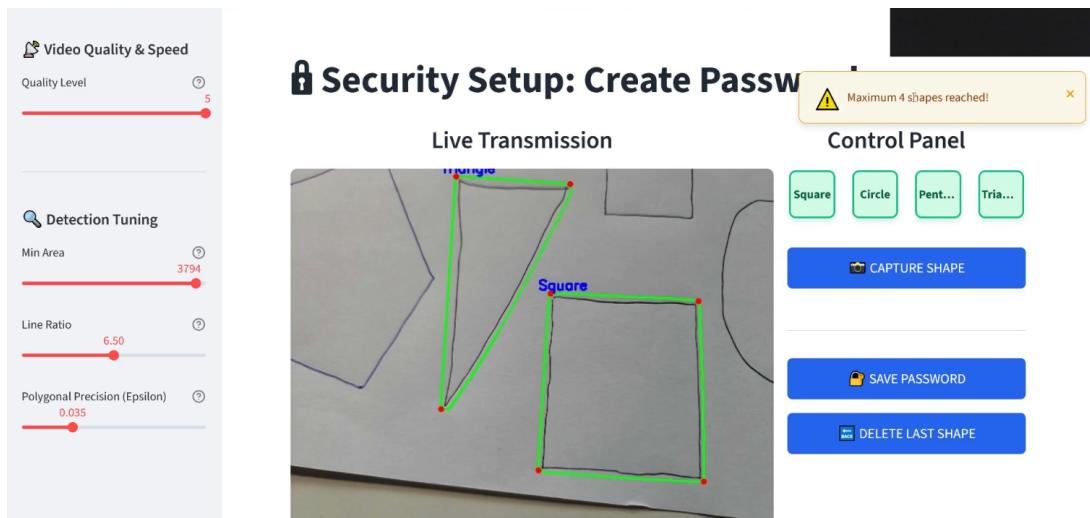


Figura 5: Detección de un patrón de contraseña erróneo



**Figura 6:** Estado del sistema con contraseña guardada y lista para desbloquear



**Figura 7:** Notificación de límite máximo de formas geométricas alcanzado



**Figura 8:** Confirmación de desbloqueo correcto del sistema de seguridad

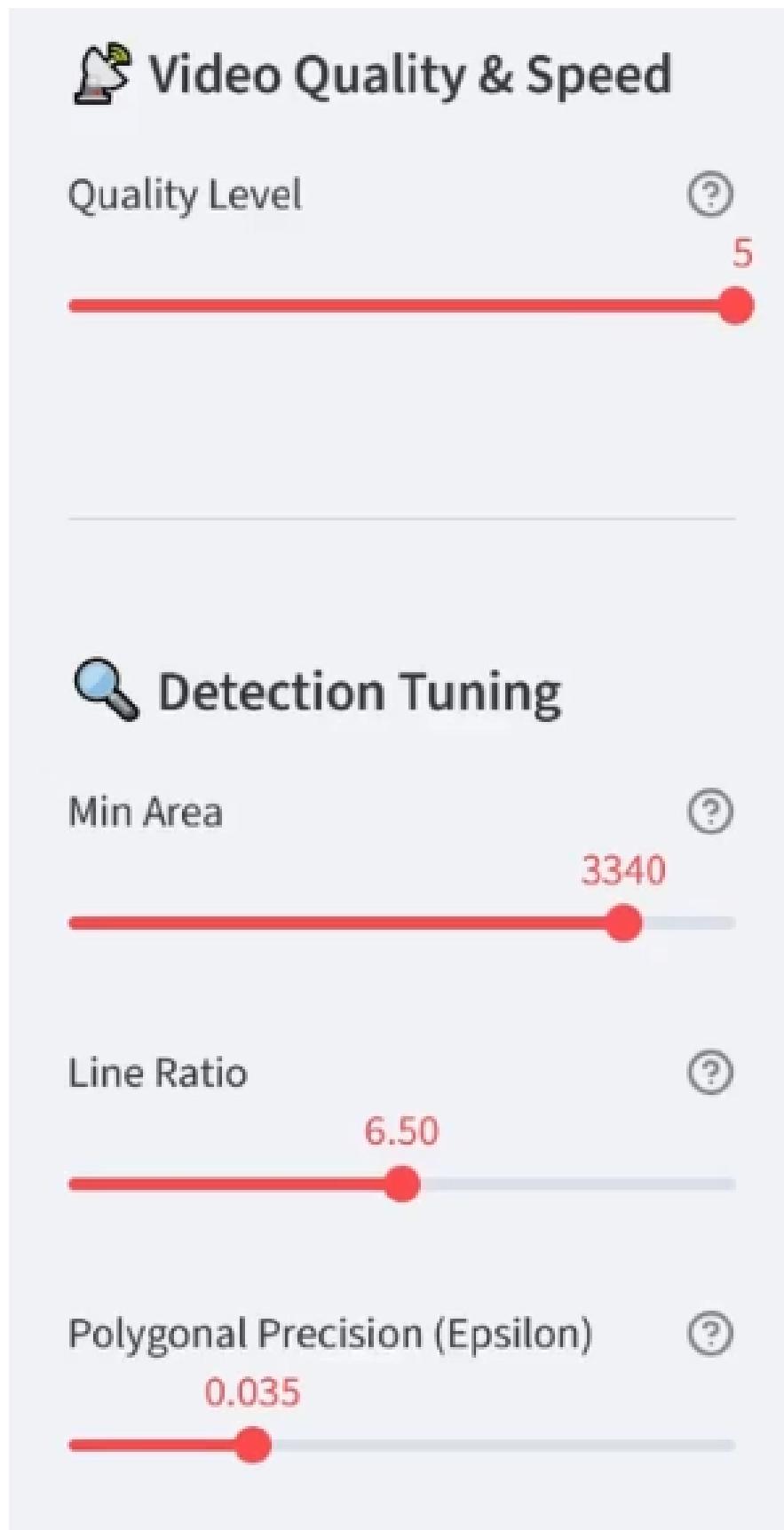


Figura 9: Barra lateral de configuración de parámetros de detección

## 6.2. Tracking y Pádel

En este apartado se muestran las funcionalidades del árbitro autónomo, incluyendo el panel de control principal, la calibración, la detección de eventos de juego (botes y saques) y una secuencia de saques analizados de ejemplo.



**Figura 10:** Panel de control principal (Dashboard) de Padel AI



**Figura 11:** Interfaz de análisis de saques en tiempo real

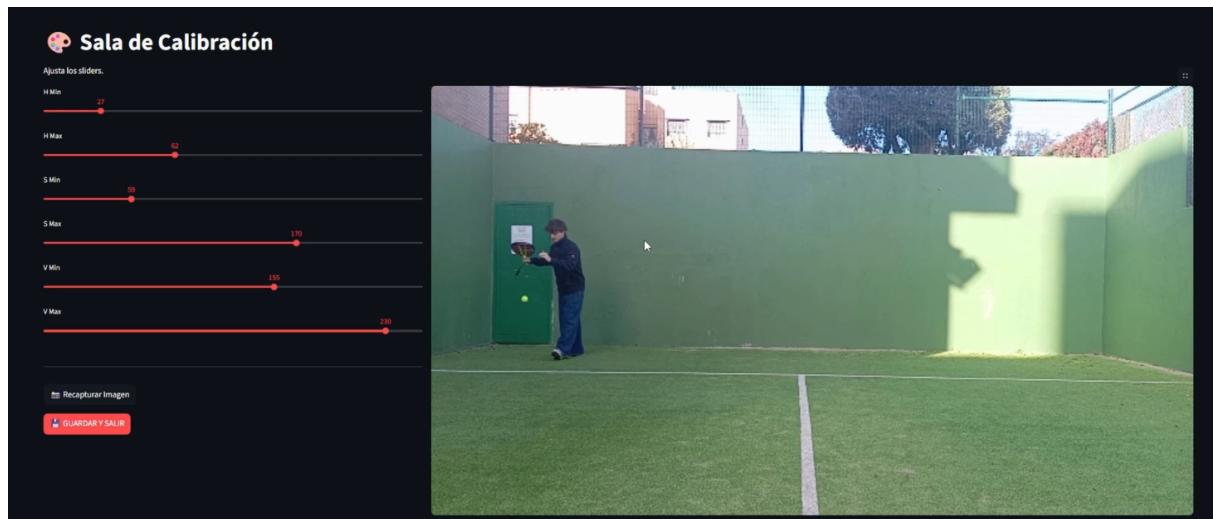


Figura 12: Menú de calibración para segmentación de color en el tracking

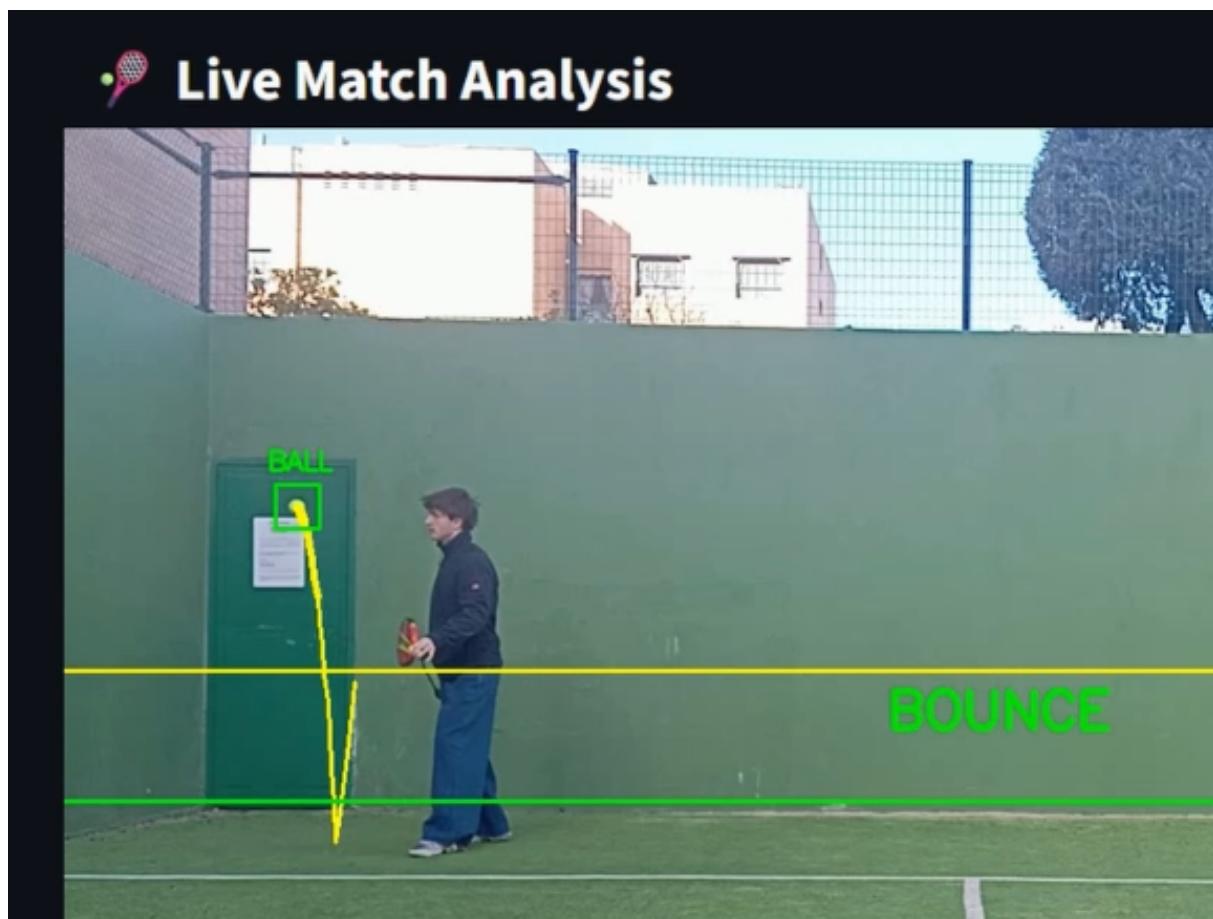


Figura 13: Ejemplo visual de detección de bote de la pelota

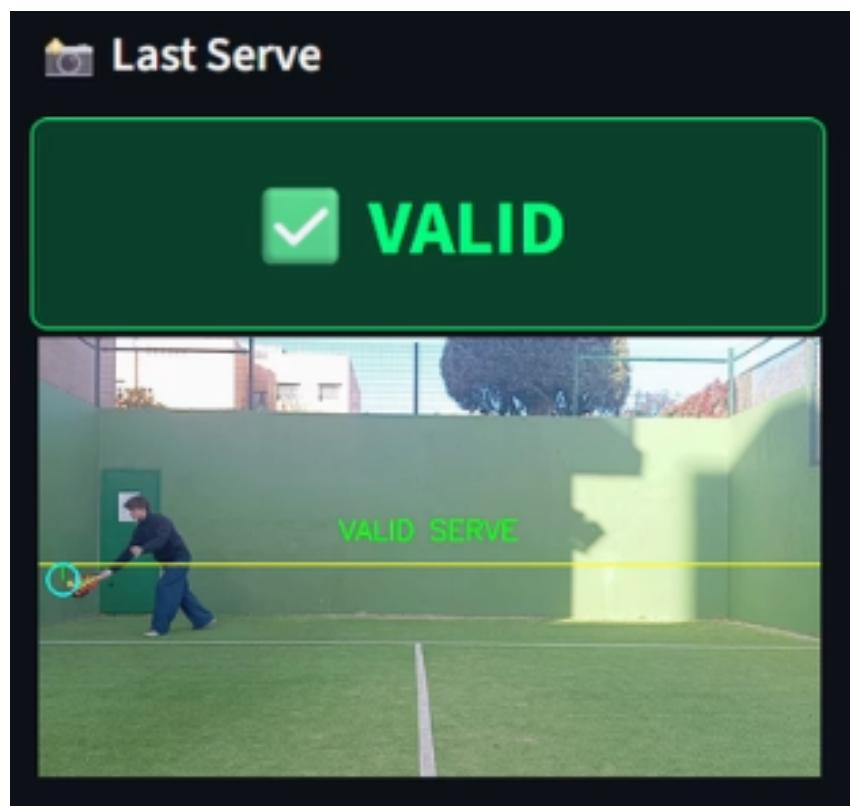


Figura 14: Visualización de un saque clasificado como válido.

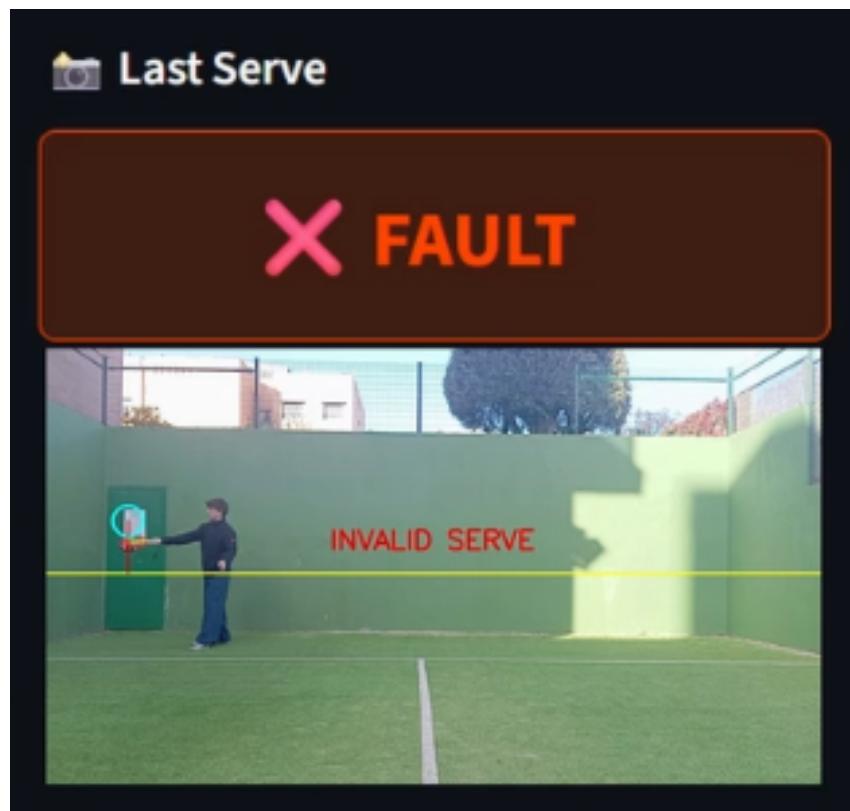


Figura 15: Visualización de un saque clasificado como inválido.



**Figura 1:** Valid Serve



**Figura 2:** Valid Serve



**Figura 3:** Valid Serve



**Figura 4:** Valid Serve



**Figura 5:** Invalid Serve



**Figura 6:** Invalid Serve