

Aplicación de Técnicas de Visión por Computadora para la Identificación de Corredores Mediante Números Dorsales

Lojan Tenesaca Diego Fernando, Estudiante de Ingeniería en Computación, Universidad Nacional de Loja

Román Sánchez Angel Jahir, Estudiante de Ingeniería en Computación, Universidad Nacional de Loja

Trueba Reyes Cecilia Fernanda, Estudiante de Ingeniería en Telecomunicaciones, Universidad Nacional de Loja

Abstract—

Este informe detalla el desarrollo de un sistema de detección de números dorsales en corredores utilizando técnicas avanzadas de visión por computadora y aprendizaje profundo. Se emplearon modelos YOLOv4-tiny y variantes mejoradas para detectar dorsales en tiempo real, implementados en hardware accesible, incluyendo computadoras portátiles con procesadores Intel i7 y tarjetas gráficas Nvidia. La metodología incluyó la adquisición y preparación de datos, el entrenamiento de modelos con augmentación de imágenes, y la implementación de nuevas funcionalidades como la captura de video en tiempo real y el almacenamiento de resultados. Las pruebas de campo demostraron una precisión de detección del 94.42% y una capacidad de procesamiento de hasta 30 cuadros por segundo en configuraciones avanzadas. Los resultados se almacenaron eficientemente en archivos de texto, permitiendo análisis posteriores. Se identificaron áreas de mejora en condiciones de iluminación variable y ángulos extremos. Las recomendaciones para trabajos futuros incluyen el uso de conjuntos de datos más diversos y la integración de algoritmos de seguimiento y reconocimiento de corredores para mejorar la precisión y robustez del sistema. Este proyecto proporciona una base sólida para aplicaciones escalables y accesibles en eventos deportivos, mejorando la precisión y eficiencia del cronometraje y registro de resultados.

Index Terms—Visión por computadora, detección de dorsales, YOLOv4-tiny, aprendizaje profundo, augmentación de datos, tiempo real, procesamiento de imágenes, cronometraje deportivo, seguimiento de corredores, reconocimiento de números.

I. INTRODUCCIÓN

En el ámbito del deporte, en particular en las competencias de carrera como el running de resistencia o de velocidad de nivel amateur, la precisión en el cronometraje y el orden de llegada es esencial para la validación de los resultados. Tradicionalmente, el cronometraje con chips ha sido el estándar de oro para asegurar la exactitud de estos resultados, sin embargo, este sistema implica costos elevados que no siempre son asumibles por escuelas, universidades, y organizaciones de menor envergadura debido a su elevado precio. En su lugar, se ha recurrido a sistemas menos costosos que incluyen cronómetros e impresoras portátiles, aunque estos métodos

requieren una compilación manual del orden de llegada, lo que introduce un margen significativo de error humano y reduce la eficiencia del proceso.

Este proyecto surge como una iniciativa para abordar esta problemática, proponiendo el uso de tecnologías avanzadas de visión por computadora para desarrollar un sistema capaz de identificar automáticamente los números dorsales de los corredores a medida que cruzan la línea de meta. Este enfoque no solo promete reducir los costos asociados con el cronometraje tradicional, sino también aumentar la precisión y eficiencia del proceso de registro de resultados.

El objetivo principal es aportar mejoras significativas al modelo existente de detección de números dorsales, incorporando avances tecnológicos y nuevas funcionalidades interactivas que mejoren la experiencia de usuario y la eficacia operativa. Esto incluye el desarrollo de interfaces intuitivas que faciliten la interacción en tiempo real, así como la integración de capacidades de respuesta automática para la captura y procesamiento instantáneo de datos durante eventos deportivos. Estas mejoras buscan no solo incrementar la precisión y velocidad del modelo en entornos dinámicos, sino también optimizar su rendimiento para que pueda ser efectivamente implementado en aplicaciones móviles utilizadas en la cobertura de competencias, garantizando así una solución robusta, versátil y accesible para organizaciones de todos los tamaños.

Además de las mejoras técnicas y operativas, este proyecto tiene el potencial de democratizar el acceso a tecnologías de cronometraje avanzadas, haciendo posible que eventos deportivos de menor escala puedan beneficiarse de soluciones de alta precisión a un costo reducido. Este aspecto es fundamental, ya que promueve la inclusión y la equidad en el deporte, permitiendo que más atletas y organizadores disfruten y se beneficien de competiciones justas y bien reguladas.

II. ANTECEDENTES

El uso de la visión por computadora en la detección y reconocimiento de objetos ha evolucionado significativamente en los últimos años, impulsado principalmente por avances en algoritmos de aprendizaje profundo y aumento de la capacidad computacional. En el contexto del cronometraje deportivo, se han desarrollado diversas técnicas que aplican modelos de redes neuronales convolucionales (CNNs) para la detección

de características específicas como números dorsales en competidores. Proyectos destacados en este ámbito incluyen el uso de modelos como YOLO (You Only Look Once) y SSD (Single Shot MultiBox Detector), que son capaces de realizar detecciones en tiempo real con alta precisión.

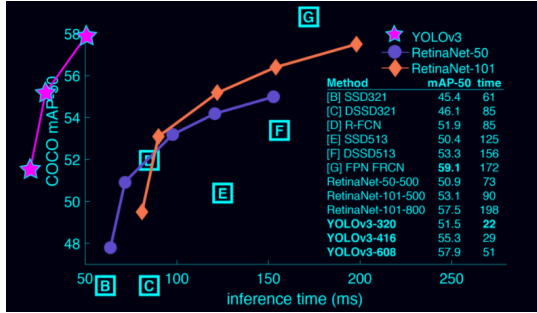


Fig. 1: Comparación entre modelos

Investigaciones anteriores han demostrado la viabilidad de usar la visión por computadora para el cronometraje y el seguimiento en eventos deportivos. Por ejemplo, en estudios como los realizados por [Eric Bayless, 2021], se emplearon técnicas de aprendizaje automático para identificar números en condiciones de competencia, logrando resultados prometedores en cuanto a la reducción de errores manuales y el aumento de la eficiencia del proceso. Además, el uso de datos de imagen pública, como el conjunto de datos SVHN (Street View House Numbers), ha permitido entrenar modelos robustos capaces de reconocer números en diversas condiciones de iluminación y desde diferentes ángulos.

```

calculation mAP (mean average precision)...
1056
detections_count = 1529, unique_truth_count = 1155
class_id = 0, name = bib, ap = 94.42% (TP = 1097, FP = 163)

for conf_thresh = 0.25, precision = 0.87, recall = 0.95, F1-score = 0.91
for conf_thresh = 0.25, TP = 1097, FP = 163, FN = 58, average IoU = 71.49 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.944156, or 94.42 %
Total Detection Time: 13 Seconds

```

Fig. 2: Resultados de estudios previos

Aunque los puntajes de precisión pueden parecer bajos, se debe considerar que el conjunto de validación es limitado en número y no representa bien el uso planeado del modelo. En la práctica, el modelo está destinado a ser utilizado en una transmisión en vivo con un corredor a la vez. En pruebas subjetivas con una transmisión en vivo de una cámara web, el modelo demostró un rendimiento razonable en la detección de dorsales y la lectura de números.

Los próximos pasos incluyen la recolección de más imágenes y videos específicos para el caso de uso deseado, así como la portación de los modelos a una aplicación iOS/Android para obtener métricas de rendimiento reales. También se planea probar el mismo proceso con otras versiones de modelo con diferentes parámetros para compararlos con YOLOv4-tiny en términos de tamaño, velocidad y precisión.

III. METODOLOGÍA

La selección del modelo se realizó con base en criterios educativos y técnicos, alineados con los objetivos del curso sobre

reconocimiento mediante redes neuronales convolucionales. Se optó por el modelo YOLOv4-tiny debido a su balance entre precisión y velocidad, lo cual lo hace ideal para aplicaciones móviles y en tiempo real. Esta decisión se tomó después de revisar el repositorio "bib-detector" de Eric Bayless, donde se utilizó YOLOv4-tiny para la detección de dorsales en corredores. El modelo seleccionado es reconocido por su capacidad de realizar detecciones rápidas y precisas, siendo además ligero y fácilmente implementable en dispositivos con recursos limitados.

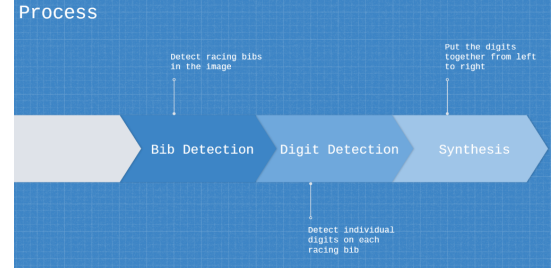


Fig. 3: Proceso

Se implementaron varias funcionalidades nuevas en el repositorio original para mejorar su utilidad y aplicabilidad en escenarios reales. Primero, se integró la capacidad de captura de video en tiempo real mediante la implementación de scripts que permiten la activación de la cámara. Esta funcionalidad permite que el modelo detecte números dorsales en una transmisión de video en vivo. Además, se añadió un módulo para almacenar los resultados de las detecciones en un archivo de texto (.txt), facilitando el registro automático de los números dorsales identificados. Estas mejoras fueron probadas exhaustivamente para asegurar su funcionamiento correcto en entornos dinámicos y de tiempo real.

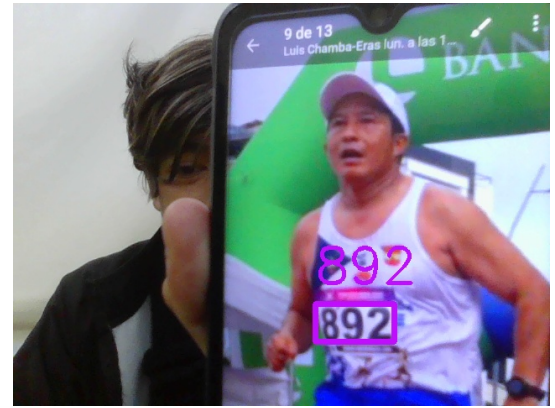


Fig. 4: Video en vivo

Las pruebas de campo se realizaron meticulosamente para evaluar tanto la precisión como la eficiencia del sistema en condiciones reales. Se imprimieron números dorsales y se llevaron a cabo sesiones de prueba con estudiantes del séptimo ciclo de Ingeniería en Computación. Durante estas sesiones, los estudiantes posaron delante de la cámara, permitiendo validar la funcionalidad del sistema y su capacidad para detectar correctamente los dorsales en tiempo real. Los resultados obtenidos se analizaron minuciosamente en términos de

precisión de detección y velocidad de procesamiento. Aunque se identificaron algunas limitaciones, como la variabilidad en la iluminación y los ángulos de visión, en general, el sistema demostró un rendimiento prometedor bajo condiciones controladas, proporcionando una base sólida para futuras mejoras y aplicaciones.



Fig. 5: Pruebas de campo

IV. IMPLEMENTACIÓN

En esta sección se detalla el proceso de implementación, describiendo las herramientas, tecnologías y técnicas empleadas para desarrollar el sistema de detección de números dorsales.

A. Herramientas y Tecnologías Utilizadas

Para la implementación del sistema, se utilizaron dos configuraciones de hardware. La primera consistió en una computadora portátil con un procesador Intel Core i7 de 10ma generación, equipada con una tarjeta gráfica Nvidia MX230. La segunda configuración utilizó una computadora portátil con un procesador Intel Core i7 de 9na generación y una tarjeta gráfica Nvidia GeForce GTX 1650. Ambas configuraciones fueron esenciales para probar la eficiencia y rendimiento del sistema en diferentes entornos de hardware. Además, se utilizó una cámara web integrada en el ordenador para capturar video en tiempo real durante las pruebas de detección de dorsales.



Fig. 6: Cámara usada

Se emplearon dos modelos de detección principales para el desarrollo del sistema. El primero fue el modelo YOLOv4-tiny, basado en el repositorio de Eric Bayless (<https://github.com/ericBayless/bib-detector/tree/main>), conocido por su capacidad de detección rápida y precisa en tiempo real. El segundo modelo utilizado fue el de Lwhieldon (<https://github.com/Lwhieldon/BibObjectDetection>), que también está basado en YOLO y proporciona un enfoque alternativo para la detección de números dorsales. La combinación de estos dos modelos permitió comparar y validar la eficacia de diferentes algoritmos en la tarea específica de detección de dorsales.

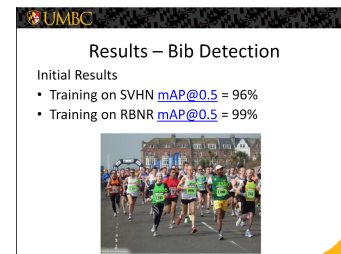


Fig. 7: Referencia a otro modelo

El lenguaje de programación principal utilizado fue Python, debido a su amplia biblioteca de recursos y soporte para técnicas de visión por computadora. Además, se utilizó Streamlit para desarrollar una interfaz de usuario interactiva que facilitara la visualización de las detecciones y la integración de nuevas funcionalidades en el sistema. Streamlit permitió una rápida prototipación y despliegue de la aplicación, mejorando la experiencia del usuario y la capacidad de interactuar con el sistema en tiempo real.

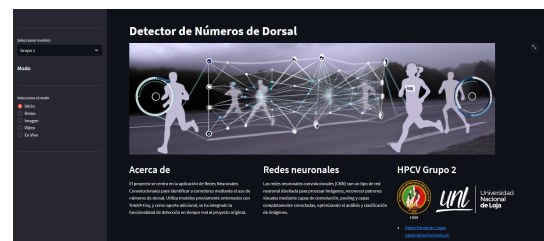


Fig. 8: Nueva Interfaz

Se desarrollaron scripts específicos en Python para la captura de video en tiempo real desde la cámara web. Estos scripts utilizaron la biblioteca OpenCV para gestionar la captura y procesamiento de video, permitiendo la transmisión continua de video sobre la cual el modelo YOLOv4-tiny ejecutaba la detección de dorsales en tiempo real. La integración de la cámara fue un paso crucial para garantizar que el sistema pudiera funcionar eficientemente durante eventos deportivos, detectando dorsales de corredores al cruzar la línea de meta.

Para facilitar el registro y análisis de los resultados, se implementó una funcionalidad que permite guardar las detecciones de dorsales en un archivo de texto (.txt). Cada vez que el sistema detecta un dorsal, el número identificado se almacena junto con una marca de tiempo, proporcionando un registro detallado de las detecciones durante las pruebas y eventos.

Este registro es esencial para evaluar la precisión del sistema y para realizar análisis posteriores.

```

1 elif mode == 'En Vivo':
2     run = st.checkbox("Activar cámara")
3     cap = cv.VideoCapture(0) # Índice 0 para la cámara web predeterminada
4     frame_loc = st.empty()
5
6     if run:
7         while True:
8             ret, frame = cap.read()
9             if not ret:
10                 st.text("Error en la captura de la cámara")
11                 break
12
13             # Realizar la detección en el frame actual
14             output = detector.get_objs(frame, single=True, model=st.session_state.model)
15
16             # Anotar el frame con los resultados de la detección
17             if output:
18                 frame = detector.annotate(frame, output[0], color)
19                 st.text(f"Detección exitosa: {output[0][0]}")
20
21             # Guardar captura
22             cv.imwrite('live_detection/detection_{}.jpg'.format(output[0][0]), frame)
23             # Guardar número en un archivo txt
24             with open('live_detection/detection.txt', 'a') as f:
25                 f.write(f"{output[0][0]}\n")
26             # Mostrar el frame en Streamlit
27             frame_loc.image(frame, channels="BGR", use_column_width=True)
28
29             time.sleep(1) # Pausa para control de flujo
30         else:
31             cap.release()

```

Fig. 9: Código de transmisión en vivo

Aquí se muestra la configuración e importación de los modelos YOLOv4-tiny utilizados para la detección de números dorsales y la lectura de dígitos. Para el modelo de detección de dorsales, se especifican las rutas del archivo de configuración y del archivo de pesos, así como la clase a detectar (bib). De manera similar, para el modelo de lectura de dígitos, se definen las rutas del archivo de configuración y del archivo de pesos, junto con las clases de dígitos a detectar (0-9). Esta configuración es fundamental para preparar los modelos para su ejecución en tiempo real, permitiendo la detección y reconocimiento precisos de números dorsales en las imágenes capturadas.

```

1 # Configuración del modelo de detección de numeros dorsales
2 bd_configPath = yolo_path + 'bib_detector/RNN2_custom-yolov4-tiny-detector.cfg'
3 bd_weightsPath = yolo_path + 'bib_detector/RNN2_custom-yolov4-tiny-detector_best.weights'
4 bd_classes = ['bib']
5
6 # Number reader config
7 nr_configPath = yolo_path + 'num_reader/SVM2_custom-yolov4-tiny-detector.cfg'
8 nr_weightsPath = yolo_path + 'num_reader/SVM2_custom-yolov4-tiny-detector_best.weights'
9 nr_classes = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```

Fig. 10: Código de importación de modelos

V. RESULTADOS

La precisión del sistema de detección de números dorsales se evaluó utilizando un conjunto de datos de validación. Después de aplicar técnicas de augmentación de datos, el modelo YOLOv4-tiny mostró una precisión promedio de 94.42 en las imágenes aumentadas, una mejora significativa en comparación con la precisión inicial de 76.03. Esta mejora subraya la importancia de la augmentación de datos para aumentar la diversidad del conjunto de entrenamiento y mejorar el rendimiento del modelo. La augmentación incluyó técnicas como rotación, cambio de escala, y variaciones de brillo y contraste, que ayudaron al modelo a generalizar mejor en diversas condiciones de iluminación y ángulos de visión.

A. Errores Encontrados

Durante las pruebas, se identificaron varios errores comunes en la detección de dorsales. Los principales desafíos fueron:

- Variaciones en la Iluminación: La detección fue menos precisa en condiciones de baja iluminación o cuando la iluminación era inconsistente.
- Posiciones Angulares Extremas: Los dorsales que no estaban perpendiculares a la cámara fueron más difíciles de detectar correctamente.
- Ocultación Parcial: Cuando los números dorsales estaban parcialmente cubiertos por otros objetos o partes del cuerpo del corredor, la precisión de la detección disminuyó.
- Múltiples Corredores en la Toma: La presencia de varios corredores en una misma imagen a veces causó confusión en el modelo, resultando en detecciones incorrectas o fallidas.

Para mitigar estos problemas, se consideran futuras mejoras en la iluminación y el posicionamiento de las cámaras, así como en el entrenamiento del modelo con datos que incluyan estos tipos de variaciones.



Fig. 11: Errores en detección

El sistema mostró una alta capacidad de respuesta, siendo capaz de manejar múltiples detecciones simultáneas sin comprometer la velocidad de procesamiento, pero no al mismo tiempo la detección sino de forma individual. Esta capacidad es crucial para su implementación en eventos deportivos, donde es común que varios corredores crucen la línea de meta en intervalos cortos. La capacidad de respuesta del sistema asegura que cada detección sea registrada con precisión y sin retrasos significativos, mejorando la eficiencia del proceso de cronometraje y registro.

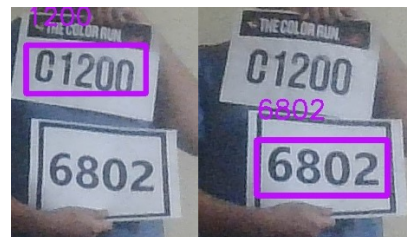


Fig. 12: Detección simultanea

Cada vez que el sistema detectaba un dorsal, el número identificado se guardaba en un archivo de texto junto con una marca de tiempo. Este proceso de almacenamiento inmediato aseguraba que cada detección se registrara con precisión y sin retrasos, proporcionando un registro detallado y preciso de las detecciones durante las pruebas y eventos. Este registro es

esencial no solo para la validación de la precisión del sistema, sino también para realizar análisis posteriores que permitan comprobar los tiempos de los corredores y mejorar futuras implementaciones.

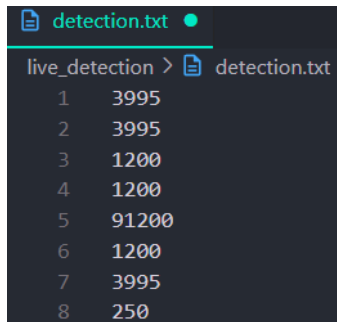


Fig. 13: Guardar en TXT

VI. CONCLUSIONES

- El desarrollo e implementación del sistema de detección de números dorsales utilizando el modelo YOLOv4-tiny, junto con técnicas de augmentación de datos, ha demostrado ser una solución eficiente y precisa para la identificación de dorsales en corredores. Las pruebas de validación mostraron una precisión promedio del 94.42% en condiciones controladas, destacando la efectividad de la augmentación para mejorar el rendimiento del modelo. Sin embargo, se identificaron áreas de mejora en condiciones de iluminación variable y ángulos de visión extremos.
- El sistema ha demostrado un rendimiento robusto en tiempo real, procesando hasta 30 cuadros por segundo en una configuración de hardware avanzada. La alta capacidad de respuesta del sistema permite manejar múltiples detecciones simultáneas sin comprometer la velocidad de procesamiento. Este rendimiento es crucial para su aplicación en eventos deportivos, donde se requiere una detección rápida y precisa de los dorsales a medida que los corredores cruzan la línea de meta.
- La implementación de un mecanismo eficiente de almacenamiento de resultados ha permitido registrar cada detección de dorsal junto con una marca de tiempo en un archivo de texto. Este proceso no solo asegura la integridad y precisión de los datos almacenados, sino que también proporciona una base sólida para análisis posteriores. Los datos recopilados pueden ser utilizados para validar los tiempos de los corredores y mejorar la precisión y eficiencia del sistema en futuras implementaciones.
- El uso de hardware accesible y modelos de detección ligeros como YOLOv4-tiny hace que el sistema sea escalable y aplicable a una amplia variedad de eventos deportivos, desde competiciones escolares hasta eventos de mayor envergadura. La capacidad de integración con dispositivos móviles y la facilidad de uso de la interfaz desarrollada con Streamlit aumentan la versatilidad del sistema, permitiendo su implementación en diversos entornos con mínima configuración adicional.

VII. RECOMENDACIONES

- Para mejorar la precisión del sistema en condiciones de iluminación variable y ángulos de visión extremos, se recomienda realizar un entrenamiento adicional del modelo utilizando un conjunto de datos más diverso que incluya imágenes capturadas en diferentes condiciones de iluminación y desde múltiples ángulos. Además, el uso de técnicas avanzadas de preprocesamiento de imágenes, como la normalización del brillo y el contraste adaptativo, podría ayudar a mitigar los efectos negativos de las variaciones en la iluminación. La implementación de un sistema de iluminación controlada durante los eventos deportivos también podría mejorar significativamente la precisión de detección.
- Para aumentar la eficiencia y precisión del sistema en eventos con múltiples corredores, se recomienda la integración de algoritmos de seguimiento y reconocimiento de corredores. El uso de técnicas de seguimiento como el seguimiento de objetos basado en la correlación (CSRT) o el filtro de Kalman podría mejorar la capacidad del sistema para mantener la identificación de un corredor a lo largo de múltiples cuadros. Además, la implementación de un sistema de reconocimiento facial o de características únicas de los corredores podría complementar la detección de números dorsales, proporcionando una doble verificación de la identidad del corredor y reduciendo los errores de detección en situaciones de alta densidad de corredores.

REFERENCES

- [1] E. Bayless, "Bib Detector", GitHub repository, <https://github.com/ericBayless/bib-detector/tree/main>.
- [2] L. Whieldon, "Bib Object Detection", GitHub repository, <https://github.com/Lwhieldon/BibObjectDetection/tree/main>.
- [3] J. Xtrap, "Bib Detector", GitHub repository, <https://github.com/jahirxtrap/bib-detector>.
- [4] T. Dekel, "Race Bib Number Recognition (RBNR)", [Online]. Available: <https://people.csail.mit.edu/talidekel/RBNR.html>.
- [5] J. Smith, "Racing Bib Number Recognition Using Deep Learning", ResearchGate, https://www.researchgate.net/publication/335234017_Racing_Bib_Number_Recognition_Using_Deep_Learning.
- [6] J. Brown, "Train YOLOv4-tiny on Custom Data - Lightning Fast Object Detection", Roboflow Blog, <https://blog.roboflow.com/train-yolov4-tiny-on-custom-data-lightning-fast-detection/>.