

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
		PAGINA: 1 de 7

16

FECHA	lunes, 24 de febrero de 2020
--------------	------------------------------

Señores
UNIVERSIDAD DE CUNDINAMARCA
 BIBLIOTECA
 Ciudad

UNIDAD REGIONAL	Extensión Chía
TIPO DE DOCUMENTO	Trabajo De Grado
FACULTAD	Ingeniería
NIVEL ACADÉMICO DE FORMACIÓN O PROCESO	Pregrado
PROGRAMA ACADÉMICO	Ingeniería de Sistemas

El Autor(Es):

APELLIDOS COMPLETOS	NOMBRES COMPLETOS	No. DOCUMENTO DE IDENTIFICACIÓN
Murcia Jiménez	Hernán Darío	1070923907

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
 Teléfono (091) 8281483 Línea Gratuita 018000976000
 www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
 NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
 Asegúrese que corresponde a la última versión consultando el Portal Institucional*

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
		PAGINA: 2 de 7

Director(Es) y/o Asesor(Es) del documento:

APELLIDOS COMPLETOS	NOMBRES COMPLETOS
Rojas Albarracín	Luis Gabriel

TÍTULO DEL DOCUMENTO
Sistema de visión artificial para detección y ubicación del golpe de un balón en superficie plana

SUBTÍTULO (Aplica solo para Tesis, Artículos Científicos, Disertaciones, Objetos Virtuales de Aprendizaje)

TRABAJO PARA OPTAR AL TÍTULO DE: Aplica para Tesis/Trabajo de Grado/Pasantía
Ingeniero de sistemas

AÑO DE EDICIÓN DEL DOCUMENTO	NÚMERO DE PÁGINAS
29/05/2019	61

DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS (Usar 6 descriptores o palabras claves)	
ESPAÑOL	INGLÉS
1. Inteligencia artificial	Artificial intelligence
2. Visión artificial	Artificial vision
3. Red neuronal	Neural Network
4. Detección de objetos	Object Detection
5. Aprendizaje profundo	Deep learning
6. Procesamiento de imágenes	Image processing

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
Teléfono (091) 8281483 Línea Gratuita 018000976000
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
Asegúrese que corresponde a la última versión consultando el Portal Institucional*



MACROPROCESO DE APOYO	CÓDIGO: AAAR113
PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
	PAGINA: 3 de 7

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS

(Máximo 250 palabras – 1530 caracteres, aplica para resumen en español):

Resumen:

En este documento se presenta un sistema de visión artificial para la detección del impacto de un balón sobre una superficie plana, entregando mediante socket las coordenadas del impacto para que puedan ser utilizadas por los videojuegos que hacen parte del proyecto de espacios inteligente. El proyecto hace uso de la librería de inteligencia artificial Tensor Flow y keras, y se obtuvo una tasa de acierto del 99% sobre el conjunto de datos de prueba.

Abstract:

This document presents an artificial vision system for the detection of the impact of a ball on a flat surface, delivering the impact coordinates by socket so that they can be used by video games that are part of the intelligent space project. The project makes use of the Tensor Flow and keras artificial intelligence library, and a 99% success rate was obtained on the test data set.

AUTORIZACION DE PUBLICACIÓN

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
Teléfono (091) 8281483 Línea Gratuita 018000976000
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
Asegúrese que corresponde a la última versión consultando el Portal Institucional*



MACROPROCESO DE APOYO	CÓDIGO: AAAR113
PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
	PAGINA: 4 de 7

Por medio del presente escrito autorizo (Autorizamos) a la Universidad de Cundinamarca para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mí (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que, en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autoriza a la Universidad de Cundinamarca, a los usuarios de la Biblioteca de la Universidad; así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado una alianza, son: Marque con una "X":

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La reproducción por cualquier formato conocido o por conocer.	X	
2. La comunicación pública por cualquier procedimiento o medio físico o electrónico, así como su puesta a disposición en Internet.	X	
3. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previa alianza perfeccionada con la Universidad de Cundinamarca para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones.	X	
4. La inclusión en el Repositorio Institucional.	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

Para el caso de las Tesis, Trabajo de Grado o Pasantía, de manera complementaria, garantizo(garantizamos) en mi(nuestra) calidad de estudiante(s) y por ende autor(es) exclusivo(s), que la Tesis, Trabajo de Grado o Pasantía en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi(nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el

	MACROPROCESO DE APOYO	CÓDIGO: AAAr113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL	VIGENCIA: 2017-11-16
	REPOSITORIO INSTITUCIONAL	PAGINA: 5 de 7

derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestra) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “*Los derechos morales sobre el trabajo son propiedad de los autores*”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Universidad de Cundinamarca está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA: (Para Tesis, Trabajo de Grado o Pasantía):

Información Confidencial:

Esta Tesis, Trabajo de Grado o Pasantía, contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de la investigación que se adelanta y cuyos resultados finales no se han publicado.

SI ___ NO _X_.

En caso afirmativo expresamente indicaré (indicaremos), en carta adjunta tal situación con el fin de que se mantenga la restricción de acceso.

LICENCIA DE PUBLICACIÓN

Como titular(es) del derecho de autor, confiero(erimos) a la Universidad de Cundinamarca una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

a) Estará vigente a partir de la fecha de inclusión en el repositorio, por un plazo de 5 años, que serán prorrogables indefinidamente por el tiempo que dure el derecho patrimonial del autor. El autor podrá dar por terminada la licencia solicitándolo a la

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL	VIGENCIA: 2017-11-16
	REPOSITORIO INSTITUCIONAL	PAGINA: 6 de 7

Universidad por escrito. (Para el caso de los Recursos Educativos Digitales, la Licencia de Publicación será permanente).

b) Autoriza a la Universidad de Cundinamarca a publicar la obra en formato y/o soporte digital, conociendo que, dado que se publica en Internet, por este hecho circula con un alcance mundial.

c) Los titulares aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.

d) El(Los) Autor(es), garantizo(amos) que el documento en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro(aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

e) En todo caso la Universidad de Cundinamarca se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.

f) Los titulares autorizan a la Universidad para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.

g) Los titulares aceptan que la Universidad de Cundinamarca pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

h) Los titulares autorizan que la obra sea puesta a disposición del público en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en el "Manual del Repositorio Institucional AAAM003"

i) Para el caso de los Recursos Educativos Digitales producidos por la Oficina de Educación Virtual, sus contenidos de publicación se rigen bajo la Licencia Creative Commons: Atribución- No comercial- Compartir Igual.



	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 3
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2017-11-16
		PAGINA: 7 de 7

j) Para el caso de los Artículos Científicos y Revistas, sus contenidos se rigen bajo la Licencia Creative Commons Atribución- No comercial- Sin derivar.



Nota:

Si el documento se basa en un trabajo que ha sido patrocinado o apoyado por una entidad, con excepción de Universidad de Cundinamarca, los autores garantizan que se ha cumplido con los derechos y obligaciones requeridos por el respectivo contrato o acuerdo.

La obra que se integrará en el Repositorio Institucional, está en el(los) siguiente(s) archivo(s).

Nombre completo del Archivo Incluida su Extensión (Ej. PerezJuan2017.pdf)	Tipo de documento (ej. Texto, imagen, video, etc.)
1. MurciaHernan2020.pdf	Texto
2.	
3.	
4.	

En constancia de lo anterior, Firmo (amos) el presente documento:

APELLIDOS Y NOMBRES COMPLETOS	FIRMA (autógrafa)
Murcia Jiménez Hernán Darío	

21.1-51.20

**SISTEMA DE VISIÓN ARTIFICIAL PARA DETECCIÓN Y UBICACIÓN DEL GOLPE
DE UN BALÓN EN SUPERFICIE PLANA**

**TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE INGENIERO DE
SISTEMAS**

HERNÁN DARÍO MURCIA JIMÉNEZ

UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

FACULTAD DE INGENIERÍA

2019

**SISTEMA DE VISIÓN ARTIFICIAL PARA DETECCIÓN Y UBICACIÓN DEL GOLPE
DE UN BALÓN EN SUPERFICIE PLANA**

HERNÁN DARÍO MURCIA JIMÉNEZ

561214161

DIRECTOR

LUIS GABRIEL ROJAS ALBARRACÍN

UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

FACULTAD DE INGENIERÍA

2019

AGRADECIMIENTOS

DEDICATORIA

RESUMEN

En este documento se presenta un sistema de visión artificial para la detección del impacto de un balón sobre una superficie plana, entregando mediante socket las coordenadas del impacto para que puedan ser utilizadas por los videojuegos que hacen parte del proyecto de espacios inteligente. El proyecto hace uso de la librería de inteligencia artificial Tensor Flow y keras, y se obtuvo una tasa de acierto del 99% sobre el conjunto de datos de prueba.

ABSTRACT

This document presents an artificial vision system for the detection of the impact of a ball on a flat surface, delivering the impact coordinates by socket so that they can be used by video games that are part of the intelligent space project. The project makes use of the Tensor Flow and keras artificial intelligence library, and a 99% success rate was obtained on the test data set.

PALABRAS CLAVE:

- Inteligencia Artificial.
- Visión Artificial.
- Red neuronal.
- Detección de objetos.

KEYWORDS:

- Artificial Intelligence.
- Artificial Vision.
- Neural Network.
- Object Detection.

LISTA DE ILUSTRACIONES

<i>Ilustración 1</i> Flujo de Trabajo del Deep Learnig-----	24
<i>Ilustración 2</i> Descripción grafica de la metodología-----	28
<i>Ilustración 3</i> Etiquetación de las imágenes -----	36
<i>Ilustración 4</i> Estructura del formato PASCAL VOC para la etiquetación de cada balón.	37
<i>Ilustración 5</i> Comandos para la generación de archivos TFRecord-----	38
<i>Ilustración 6</i> Estructura de mapa de etiquetas -----	38
<i>Ilustración 7</i> Configuración de rutas para entrenamiento-----	40
<i>Ilustración 8</i> Comando para lanzar ejecución de entrenamiento. -----	40
<i>Ilustración 9</i> Proceso de entrenamiento. -----	41
<i>Ilustración 10</i> Resultados generales durante el entrenamiento.-----	42
<i>Ilustración 11</i> Comando para la generación del grafico congelado.-----	42

LISTA DE TABLAS

LISTA DE ANEXOS

Anexo A Archivo de configuración de la red neuronal de configuración de la red neuronal

Anexo B Banco de pruebas para el prototipo número 1.

Anexo C Costo del proyecto.

Anexo D Banco de pruebas para el prototipo número 2.

ACRÓNIMOS

IA = Inteligencia Artificial

CNN = Neural Network Convolutional

INTRODUCCIÓN

En la actualidad muchas industrias y organizaciones han acogido a la inteligencia artificial (I.A.) dentro de sus procesos más importantes, diseñando herramientas que permitan reducir tiempos y costos en los procesos de fabricación de productos, o en la verificación de calidad de los mismos [1].

Otro tipo de enfoque uso que se puede dar a la I.A, consiste en la detección e identificación de objetos esto comúnmente se conoce como visión artificial. Esto puede ser utilizado para diferentes sistemas, entre los que se encuentran los sistemas de movilidad, manufactura e incluso, videojuegos como se presentará más adelante.

El presente proyecto muestra un sistema de inteligencia artificial para detectar e identificar un balón, con el fin de poder entregar a un videojuego las coordenadas de impacto del balón contra una pared.

Para el desarrollo del sistema se construyó un banco de imágenes para el entrenamiento de la inteligencia artificial, adicionalmente se utilizó la técnica de transferencia aprendizaje, para reducir los tiempos de entrenamiento.

El sistema presentado en este trabajo logra una tasa de acierto del 99% en la detección de este tipo de objetos.

PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

La universidad de Cundinamarca extensión Chía en su facultad de ingeniería de sistemas, cuenta con dos semilleros de investigación llamados IDCÍ e Invensoft los cual están desarrollando un proyect para cambiar la forma en que se realizan las clases de deportes en los colegios, para lograr un mayor estímulo a nivel sicológico, es decir, que a los alumnos les sea más atractiva la clase y eso conlleve a una cultura de actividad física habitual en los jóvenes. El proyecto consiste en la creación de un área interactiva en un espacio cerrado que puede ser del tamaño de una cancha de baloncesto o de un salón de clases. En las paredes del escenario serán proyectadas imágenes animadas, provenientes de un videojuego.

Sin embargo, es de entender que, para lograr una dinámica más interactiva por parte de los estudiantes, es necesario encontrar algún mecanismo no invasivo, puesto que los jugadores no deberán tener ningún tipo de dispositivo introducido en sus cuerpos.

Para lograr capturar las acciones que realizan los alumnos será necesario incluir tecnologías como visión artificial, la cual se definen como disciplina científica que incluye métodos para adquirir, procesar y analizar imágenes del mundo real, con el fin de producir información que pueda ser tratada por una máquina (contaval,2016). Con base en lo anterior, el sistema de visión artificial deberá ser capaz de realizar un procesamiento de imágenes en tiempo real e identificar las coordenadas donde golpee un balón contra la pared.

El proyecto en su primera etapa a definido como mecánicas para el videojuego, que exista una interacción entre los niños y un balón, es por esta razón que la visión artificial deberá ser capaz de realizar la detección de círculos, mediante algún algoritmo que le permita definir cuáles fueron las coordenadas “X”, “Y”, con las que chocó el balón.

El procesamiento de imágenes para realizar la detección de los círculos contiene múltiples inconvenientes a la hora de realizar la detección de éstos, entré los que se encuentran las variaciones de luz las cuales realizan interferencias y ruidos que provocan tener falsos positivos en la detección, la los frames que puede procesar una cámara, el ángulo de inclinación de las cámaras, entre otros.

FORMULACIÓN DEL PROBLEMA

¿Cómo desarrollar un sistema que utilice técnicas de visión artificial para detectar y ubicar las coordenadas de impacto de un balón contra una superficie plana?

OBJETIVOS

OBJETIVO GENERAL:

Crear un sistema de visión artificial que permita detectar cuando un balón golpea con una pared siendo capaz de determinar cuáles fueron las coordenadas “X”, “Y” con las que chocó.

OBJETIVOS ESPECÍFICOS:

1. Investigar los diferentes algoritmos de detección de círculos.
2. Establecer un banco de pruebas con los videos previamente filmado, que van a ser utilizados para entrenar el sistema de visión artificial.
3. Desarrollar, probar y evaluar prototipos que permitan detectar las coordenadas de choque del balón contra la pared.
4. Documentar el método seleccionado y el despliegue del producto final.

ALCANCES Y LIMITACIONES

ALCANCES

Desarrollo de un sistema no invasivo que sea fácil de implementar en un aula de clase y que reconozca uno o varios balones, además que permita conocer las coordenadas “X”, “Y” las que el objeto (balón) choca contra una pared, estas coordenadas serán relativas dependiendo del grado de inclinación que tenga el proyector del videojuego. El sistema también será capaz de transmitir a un servidor dichas coordenadas.

LIMITACIONES

1. Debido a que el sistema de visión artificial se ve afectado por las variaciones de luminosidad, se determina que la luz debe de ser controlada y monitoreada para obtener los resultados adecuados durante la puesta en marcha del sistema.
2. Para realizar una óptima detección de los balones se tendrán las cámaras a una distancia específica de la pared y con un ángulo de inclinación de 90° en el eje en que se encuentren posicionadas.
3. Debido a que los proceso que se realizan en el procesamiento de imágenes consumen una gran cantidad de recursos será necesario contar características específicas de los dispositivos que permitan la obtención de las imágenes (Ver anexo D).

JUSTIFICACIÓN

Para poder desarrollar el sistema de visión artificial que permita la interacción entre los niños y el video juego con un balón es necesario e importante investigar y conocer sobre los distintos algoritmos de reconocimiento de círculos. Si llegase a no encontrasen algoritmos que no realicen esta tarea, será necesario desarrollar un algoritmo capaz de identificar los círculos (balones) con un margen de error muy bajo.

La creación del sistema de visión artificial es importante porque permitirá que se ejecuten las mecánicas ya propuestas para el videojuego y que los resultados entregados por el sistema de visión artificial se vean en tiempo real, permitiendo crear un nuevo ambiente deportivo, beneficiando a los niños que lo utilicen, y siendo una fuente de apoyo para los profesores al momento de impartir los conocimientos a los niños.

Desde el punto de vista ingenieril con este desarrollo, se podrán sentar bases para desarrollos futuros, donde el tratamiento de imágenes en tiempo real sea el centro de atención, puesto que la documentación que se generará servirá de guía para realizar posteriores detecciones y análisis con distintos objetivos a detectar, pudiendo ser utilizados en otros sistemas de visión artificial.

A nivel industrial actualmente hay muchas empresas y organizaciones que han acogido a la inteligencia artificial (I.A.) dentro de sus procesos más significativos e importantes, diseñando herramientas que permitan optimizar tiempos y costos en los procesos de fabricación de productos, o en la verificación de calidad de los mismos (Benítez, Escudero, Kanaan, & Rodó, 2014), con este desarrollo se hace un aporte significativo al desarrollo de nuevos sistemas inteligentes.

LÍNEA DE INVESTIGACIÓN

SOFTWARE, SISTEMAS EMERGENTES y NUEVAS TECNOLOGÍAS, conjunto de programas, subprogramas subrutinas y menús que se elaboran a manera de aplicaciones y/o paquetes para cumplir con un fin específico. (Acuerdo No 007 de mayo 29 de 2003).

MARCO TEÓRICO

MARCO REFERENCIAL

El ministerio de educación de España en su artículo llamado “Aplicación práctica de la visión artificial en el control de procesos industriales” define la visión artificial “como un campo de la “Inteligencia Artificial”, que mediante la utilización de las técnicas adecuadas permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales (Nogué & Antiga, 2012).

Entre los usos que se le ha dado a la visión artificial se encuentra “A fast and robust circle detection method using isosceles triangles sampling”, el cual es un proyecto que fue creado el 4 de noviembre del 2015 por el Departamento de Física de la Universidad de Umea (Suecia), en este se presenta un método de detección de círculos en imágenes con un alto contenido de ruido y cómo mediante la detección de triángulos isósceles y la utilización de un muestreo aleatorizado se produce reducción significativa de los recursos computacionales y se mejora la detección de estos.

Como es denotado en el artículo presentado por el Departamento de Física de la Universidad de Umea, la detección de círculos en una imagen se puede realizar mediante una estrategia de muestreo de tres puntos, esto consiste en tomar tres píxeles de borde en cada ciclo y representar un círculo o mediante un muestreo circular de cuatro puntos, este cuarto píxel de borde se utiliza para determinar si se encuentra a la distancia correcta del círculo, lo que permite que haya un alto porcentaje de efectividad, pero estos algoritmos aunque son eficientes son susceptibles a ruido y lo que conlleva a que existan falsos positivos durante la detección.

El algoritmo que se plantea cuenta con los criterios trigonométricos de los triángulos isósceles, así como los procesos para aumentar el rendimiento de muestreo aleatorio y aumentar la robustez de este.

Éste está dividido en las siguientes secciones:

- 1) Estimación del gradiente y clasificación del borde
- 2) Estrategia de muestreo
- 3) Refinamiento dependiente del gradiente
- 4) Proceso de validación

Este algoritmo presenta una buena robustez en los criterios que tiene para detectar y definir si se ha encontrado un círculo o no, además de contar con una alta tasa de rechazo ante los falsos positivos. Cuenta con un excelente filtro contra el ruido de Gauss. El método propuesto puede combinarse fácilmente con técnicas modernas de detección de bordes y extracción de contorno, aunque se mejora el rendimiento no es la mejor en tiempos de detección de círculos.

La visión artificial también se ha adentrado en el mundo de los deportes, haciendo uso de la detección de las pelotas de tenis con el sistema llamado ‘Hawk-Eye’ o también conocido como el Ojo de Halcón, es una tecnología de origen británico, que comenzó a utilizarse hace más de una década, se inició a desarrollar a principios del año 2001, donde el críquet, el billar inglés y el tenis fueron los primeros deportes en implementar esta tecnología.

Su principal funcionalidad consiste en establecer el punto donde golpean las bolas, estableciendo si las bolas pegaron dentro o fuera de los límites establecidos en el caso del tenis. El Ojo de Halcón está compuesto por diez cámaras situadas en puntos estratégicos (para el caso del tenis), las cuales están conectas a varios ordenadores encargados de realizar los cálculos correspondientes para encontrar el punto exacto de choque de las bolas.

El sistema realizar los cálculos con base en las imágenes capturadas por las cámaras, a esto se le añade otra variable, el tiempo, con el cual se encarga de crear una animación 3D del evento que se procesó (Owens, Harris, & Stennett, 2003).

En el campo de la educación la IA ha hecho varios aportes, uno de ellos es el proyecto llamado “The Implementation Of Hand Detection And Recognition To Help Presentation Processes”, este proyecto de visión artificial fue creado por la Universidad Bina Nusantara (Indonesia) en el año 2015, está enfocado a utilizar las manos como objeto para manipular una computadora y facilitar la presentación de contenidos por parte de los profesores en sus aulas de clase. Esta apoyado de las librerías FLTK, OpenGL y OpenCv (Sutoyo, Prayoga, Suryani, & Shodiq, 2015).

El sistema cuenta con cinco etapas:

1. Captura
2. Transformación
3. Procesamiento de la imagen

4. Detección de la mano
5. Reconocimiento de las manos.
6. Controlador de los eventos.

Aunque las seis etapas son importantes la etapa más relevante es el proceso es la detección de las manos, puesto que, es en esta etapa donde el objeto (mano) es reconocido como la mano y será localizada para posteriormente para ser utilizada como una entrada para operar la computadora (Sutoyo, Prayoga, Suryani, & Shodiq, 2015).

Otro de los métodos para la detección de objetos con formas circular con visión artificial es realizando análisis de curvaturas isophotes, como el estudio efectuado por el departamento de Ingeniería Mecánica y Automática, Universidad China de Hong Kong, en el año 2014, en el que se efectuó la estimación de la curvatura isophotes, la clasificación de píxeles de borde y la detección de círculos de forma iterativa (De Marco, Cazzato, Leo, & Distante, 2015)

En la actualidad la inteligencia artificial ha evolucionado creando nuevos sistemas más óptimos en tiempos y en efectividad por medio del uso de redes neuronales, como es el caso de Faster R-CNN (Faster Region Based Convolutional Network) (Shaoqing Ren, Kaiming He, Girshick, & Jian Sun, 2017), el cual hace parte de la familia CNN, siendo la evolución de R-CNN (Girshick, Donahue, Darrell, & Malik, Jun 2014) y Fast R-CNN (Girshick, Dec 2015) en las cuales las propuestas de región se generaban mediante la búsqueda selectiva (SS) (Sande, Uijlings, Gevers, & Smeulders, Jan 1, 2011) en lugar de utilizar una red neuronal convolucional (CNN).

Para Faster R-CNN, las tareas de generación de propuestas en las regiones y en la detección de objetos se realizan sobre la misma red convolucional (Shaoqing Ren, Kaiming He, Girshick, & Jian Sun, 2017) al ingresar región proposal network (RPN), una red de propuesta regional, la cual funciona en la última capa de la CNN inicial, con una ventana deslizante de 3x3 que se mueve a través del mapa de características para un tamaño más pequeño (Shaoqing Ren, Kaiming He, Girshick, & Jian Sun, 2017), en el que para cada posición de ventana deslizante, el RPN genera múltiples regiones posibles basadas en uniones espaciales de dimensiones fijas llamadas cajas de anclaje. (anchor boxes), en la que cada caja contiene un puntaje de calificación del objeto en la región y cuatro coordenadas que representan el cuadro delimitador de esa región (Alberti, 2019).

Resumiendo lo anterior, el proceso que realiza Faster RCNN se definen los siguientes pasos:

- En primer lugar, la imagen pasa a través de capas convolucionales en la que se extraen mapas de características.
- Luego se usa una ventana deslizante en RPN para cada ubicación sobre el mapa de características.
- Para la cual, en cada ubicación, se utilizan k ($k = 9$) cajas de anclaje (3 escalas de 128, 256 y 512, y 3 relaciones de aspecto de 1: 1, 1: 2, 2: 1) para generar propuestas de región (Shaoqing Ren, Kaiming He, Girshick, & Jian Sun, 2017).
- Se hace uso de una capa de clasificación para generar $2k$ puntuaciones, ya sea que haya o no objeto para k cajas.
- Una capa de registro genera $4k$ para las coordenadas (coordenadas del centro del cuadro, ancho y alto) de k cuadros.
- Obteniendo un mapa de características $W \times H$, con un total de anclas.

MARCO CONCEPTUAL

En esta sección se elabora una revisión bibliografía de los conceptos generales

INTELIGENCIA ARTIFICIAL:

“Disciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones comparables a las que realiza la mente humana, como el aprendizaje o el razonamiento lógico” (ASALE, 2019).

VISIÓN ARTIFICIAL:

“La Visión por Computador es la disciplina que estudia cómo procesar, analizar e interpretar imágenes de forma automática. Estas técnicas tienen aplicaciones en muchos ámbitos, como la seguridad, la medicina, la inspección automática, o la navegación automática” (UOC, 2019).

CONVOLUTIONAL NEURAL NETWORK:

“Una red neuronal convolucional (CNN o ConvNet) es uno de los algoritmos más populares para Deep learning (aprendizaje profundo), que es un tipo de aprendizaje automático (machine learning) en el que un modelo aprende a realizar tareas de clasificación directamente a partir de imágenes, vídeos, textos o sonidos” (Math Works, 2019).

DEEP LEARNING:

“El Deep Learning es una técnica de aprendizaje automático que enseña a los ordenadores a hacer lo que resulta natural para las personas: aprender mediante ejemplos.” El Deep Learning es una tecnología clave presente en los vehículos sin conductor que les permite reconocer una señal de stop o distinguir entre un peatón y una farola. Resulta fundamental para el control mediante voz en dispositivos tales como teléfonos, tabletas, televisores y altavoces manos libres. El aprendizaje profundo atrae mucha atención últimamente, y hay razones de peso para ello. Está consiguiendo resultados que antes no eran posibles” (Mathworks,2019).

En la ilustración 1 se muestra el flujo de trabajo que maneja Deep learning, el cual consiste en que las imágenes se envían a una CNN en la que debe aprender las características que contengan las imágenes y tener la capacidad de clasificar los objetos de forma automática.

Como se evidencia en la ilustración anterior se puede concluir que el Deep learning es una forma compleja de red neural, que en la actualidad está en auge gracias a la inteligencia artificial y que ha mejorado en las capacidades que tiene de reconocer, detectar, clasificar imágenes, videos o incluso voces.

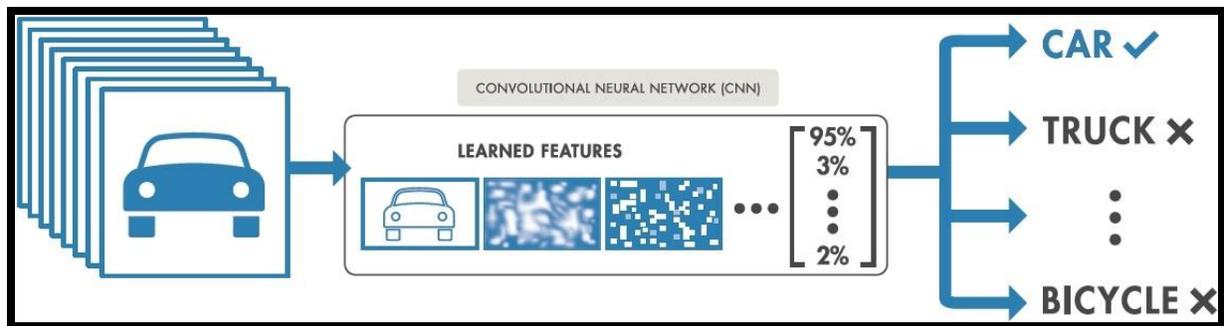


Ilustración 1 Flujo de Trabajo del Deep Learnig

DETECCIÓN DE OBJETOS: La detección de objetos es una técnica relacionada con la visión artificial que consiste en hallar todas las instancias y características que componen a un objeto para poder ser detectado (Amit & Felzenszwalb, 2019).

Otra definición que aclara el concepto de Detección de objetos es el dato por The MathWorks, Inc al definir “El reconocimiento de objetos es una técnica de visión artificial para identificar objetos en imágenes o vídeos. El reconocimiento de objetos constituye una salida clave de los algoritmos de deep learning y machine learning. Cuando las personas miramos una fotografía o vemos un vídeo, detectamos con rapidez personas, objetos, lugares y detalles visuales. El objetivo es enseñar a un ordenador a hacer lo que resulta natural para los humanos: adquirir cierto nivel de comprensión del contenido de una imagen” (MathWorks,2019).

TRANSFERENCIA DE APRENDIZAJE:

Al contar con este conjunto de datos ya pre entrenado, se puede utilizar una técnica llamada transferencia de aprendizaje, esta consiste en un proceso en el que se toma las ponderaciones de un modelo que ya ha sido entrenado con una gran cantidad de datos para realizar una tarea similar, y luego entrenar un nuevo modelo con base a los pesos del modelo previamente entrenado (Weiss, Khoshgoftaar, & Wang, 2016).

MARCO INGENIERIL

TENSORFLOW:

“Es una biblioteca de software de código abierto para cálculos numéricos que utilizan gráficos de flujo de datos. Los nodos del gráfico representan operaciones matemáticas, mientras que los bordes del gráfico representan las matrices de datos multidimensionales (tensores) que fluyen entre ellos. Esta arquitectura flexible le permite implementar la computación en una o más CPU o GPU en una computadora de escritorio, servidor o dispositivo móvil sin reescribir el código. TensorFlow también incluye TensorBoard, un kit de herramientas de visualización de datos (“TensorFlow”, 2019).

ANACONDA:

“Anaconda es una distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático (machine learning). Esto incluye procesamiento de

grandes volúmenes de información, análisis predictivo y cómputos científicos. Está orientado a simplificar el despliegue y administración de los paquetes de software” (Anaconda, 2019).

CUDA:

“Es una plataforma de computación paralela y un modelo de programación desarrollado por NVIDIA para computación general en unidades de procesamiento gráfico (GPU). Con CUDA, los desarrolladores pueden acelerar drásticamente las aplicaciones informáticas aprovechando el poder de las GPU ("CUDA Zone", 2019).

OPENCV:

OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión de computadora y de aprendizaje automático de código abierto. OpenCV fue construido para proporcionar una infraestructura común para las aplicaciones de visión artificial y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia BSD, OpenCV facilita que las empresas utilicen y modifiquen el código ("Opencv.", 2019).

REDES NEURONALES CONVOLUCIONALES (CNN):

Estos algoritmos son utilizados en el Deep Learning (aprendizaje profundo) para la detección de personas, rostros, carros, entre otros objetos, además cuentan con la capacidad de aprender con base a imágenes, haciendo uso de patrones para clasificar la característica directamente de cada imagen (Math Work,2019).

Las CNN se han vuelto cada vez más populares debido a algunos factores que las hacen favoritas para ser usadas en Deep learning, debido a que tienen las siguientes cualidades:

Optimizar el tiempo de aprendizaje mediante el uso de transferencia de aprendizaje de modelos previamente entrenados.

Elimina la extracción manual de las características.

Proporcionan una arquitectura óptima para el funcionamiento y reconocimiento de patrones.

Una red convolucional puede tener una o cientos de capas que van aprendiendo a detectar y a sacar patrones de las características de cada imagen. Algunos de los filtros que se pueden

aplicar para la salida de una capa serán la entrada de la siguiente capa los cuales pueden ir desde algo simple como el color hasta características más que definen inequívocamente el objeto (Math Work,2019).

FLTK

“Fast Light Tool Kit ("FLTK", pronunciado "fulltick") es un kit de herramientas GUI C++ multiplataforma para UNIX (r)/ Linux (r) (X11),Microsoft (r) Windows (r) y MacOS (r) X. FLTK proporciona una funcionalidad GUI moderna y admite gráficos 3D a través de OpenGL (r) y su emulación GLUT incorporada. Originalmente fue desarrollado por el Sr. Bill Spitzak y actualmente es mantenido por un pequeño grupo de desarrolladores en todo el mundo con un repositorio central en los Estados Unidos” (FLTK,2019).

OpenGL

“OpenGL es el entorno principal para desarrollar aplicaciones gráficas 2D e 3D portátiles e interactivas. Desde su introducción en 1992, OpenGL se ha convertido en la interfaz de programación de aplicaciones (API) de gráficos en 2D y 3D más utilizada y admitida en la industria, llevando miles de aplicaciones a una amplia variedad de plataformas informáticas. OpenGL fomenta la innovación y acelera el desarrollo de aplicaciones al incorporar un amplio conjunto de renderizado, mapeo de texturas, efectos especiales y otras potentes funciones de visualización. Los desarrolladores pueden aprovechar el poder de OpenGL en todas las plataformas populares de escritorio y estación de trabajo, asegurando una amplia implementación de aplicaciones.

METODOLOGÍA

La metodología seleccionada para el desarrollo del proyecto se basó en la metodología llamada “por: prototipos”, debido a que el proyecto tiene un fuerte componente de investigación y, aunque los requerimientos están claramente definidos, se pueden presentar cambios en los algoritmos y desarrollos efectuados, por razones de rendimiento, efectividad, integración con otros sistemas o por una nueva tecnología que se encuentre durante la investigación y que se acople mejor a las necesidades del proyecto.

El desarrollo del proyecto consta de 5 etapas:

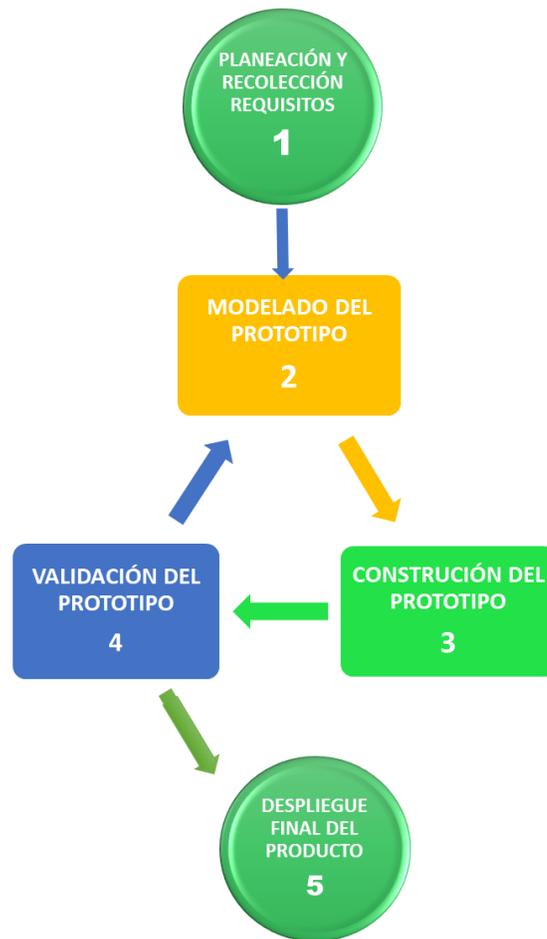


Ilustración 2 Descripción gráfica de la metodología

Para realizar la localización y extracción del balón de una imagen y la identificación de las coordenadas en las que el balón choca con la pared, se seguirán las cinco etapas anteriores, sin embargo, las etapas dos, tres y cuatro se podrán repetir de forma cíclica, es decir, que se podrán repetir cuantas veces sean necesarias (ver ilustración *Ilustración 2* Descripción gráfica de la metodología).

A continuación, se detallarán las actividades que se desarrollarán en cada una de las etapas.

PLANEACIÓN Y RECOLECCIÓN REQUISITOS

- Revisión del estado del arte.
- Asignación de actividades y tiempos
- Selección de algoritmos existentes.

MODELADO DEL PROTOTIPO

- Creación de banco de pruebas del proyecto.
- Diseñar software.

CONSTRUCCIÓN DEL PROTOTIPO

- Creación y desarrollo de prototipos.

VALIDACIÓN DEL PROTOTIPO

- Validación vídeos o imágenes etiquetados con los frames en los cuales aparece el balón para poder realizar pruebas al software.
- Pruebas técnicas y funcionales sobre cada uno de los prototipos desarrollados.

DESPLIEGUE FINAL DEL PRODUCTO

- Entrega de librería que proporciona las coordenadas al videojuego.
- Documento explicativo del método seleccionado y del aplicativo construido.
 - Documento explicando estructura del mensaje y canal, para el envío de coordenadas.

DESARROLLO DEL PROYECTO

Durante el desarrollo del proyecto se realizaron 2 prototipos, el primero utilizando la tecnología .Net y la librería para computación visual OpenCV. El segundo se basó en redes neuronales hechas con el framework tensorflow y keras. A continuación, serán explicados cada uno en detalle.

DESARROLLO PROTOTIPO NÚMERO 1

Para el prototipo número uno se creó un sistema de detección de círculos utilizando la librería Emgu Cv en su versión 3.3.0.2824, éste prototipo utiliza el algoritmo de Hough para realizar la detección y seguimiento a un balón.

ESTRATEGIA DE DETECCIÓN

1. Creación del banco de pruebas.
2. Se realizaron los siguientes métodos de detección:
 - a. Tomado como fuente la imagen original o el modelo HSV.
 - b. Realizando seguimiento al objeto.

Cada uno de los métodos tienen una serie de pasos para realizar la detección, los cuales serán explicados a detalle más adelante.

CREACIÓN DEL BANCO DE PRUEBAS

Para el banco de pruebas se tomaron 5 videos en distintos lugares, en los cuales se grabaron espacios con balones y otros en los que no se encontraban este tipo de objetos, no se hizo control de luminosidad en los lugares, se tomaron videos con diferentes balones, luego de realizar los videos se extrajeron cada uno de los frames por cada video. Obteniendo un total de 500 imágenes con balón y 250 sin balón para probar el sistema. (Ver anexo B).

MÉTODOS DE DETECCIÓN

A continuación, se detallan cada uno de los métodos de detección.

TOMADO COMO FUENTE LA IMAGEN ORIGINAL O EL MODELO HSV

Este método contiene los siguientes pasos para realizar la detección tanto para la imagen original como para el modelo HSV:

1. Configuración de contraste y brillo.
2. Configuración de grises.
3. Pre procesamiento de la imagen final antes de buscar los círculos.
4. Algoritmo de detección Hough

Descripción de los pasos anteriores:

Configuración de contraste y brillo

Puesto que el contraste es la diferencia entre los píxeles más claros y los píxeles más oscuros y el brillo es la claridad de un cuerpo (soto,2018), se realizó una transformación en la cual se toma la imagen de entrada y se convierte a una matriz de 3 canales Bgr (blue, Green and red), en la que se aplicaron los siguientes valores: “0” para el brillo y “0.25” para el contraste de las imágenes entrantes, obteniendo como resultado una matriz de las mismas dimensiones de la imagen entrante.

Configuración de grises.

“Convierte una imagen del espacio de color BGR a escala de grises. Los rangos convencionales para los valores de canal B, G y R son de 0 a 255” (Opencv,2019).

Con esta configuración se reduce el ruido generado por la cámara u otros factores como las variaciones de luz, además de reducir la matriz de la imagen entrante (original) de tres canales a una matriz de un solo canal.

En las ilustraciones Ilustración 3 Resultados obtenidos para el método HSV. Ilustración 4 Resultados obtenidos para la imagen original. se muestran los resultados obtenidos después de los pasos uno y dos, en los cuales se realizaron los ajustes de contraste y paso a escala de grises.



Ilustración 3 Resultados obtenidos para el método HSV.



Ilustración 4 Resultados obtenidos para la imagen original.

Pre procesamiento de la imagen final antes de buscar los círculos

El pre procesamiento consistió en aplicar un filtro de suavizado gaussiano, dilatación y erosión los cuales se explican a continuación.

El desenfoque / suavizado gaussiano es la técnica de suavizado más utilizada para eliminar ruidos en imágenes y videos. En esta técnica, una imagen debe ser convolucionada con un núcleo gaussiano para producir la imagen suavizada (Fernando, 2019).

Puede definir el tamaño del núcleo de acuerdo con sus requisitos. Pero la desviación estándar de la distribución gaussiana en la dirección X e Y debe elegirse cuidadosamente teniendo en cuenta el tamaño del núcleo de manera que los bordes del núcleo estén cerca de cero (Fernando, 2019), para este prototipo se tomó como valores pre determinados “3x3”.

Para la dilatación se convolucionan una imagen A con algún kernel (B), que puede tener cualquier forma o tamaño, generalmente un cuadrado o círculo A medida que B se escanea el núcleo sobre la imagen, se calcula el valor máximo de píxel superpuesto B y se reemplaza el píxel de la imagen en la posición del punto de anclaje con ese valor máximo. Esta operación de maximización hace que se suavicen los bordes de los objetos dentro de una imagen (OpenCv,2019).

En el caso de la erosión la operación consistió en calcular un mínimo local sobre el área del núcleo. A medida que B se escanea sobre la imagen, se calcula el valor mínimo de píxel superpuesto B y se reemplaza el píxel de la imagen debajo del punto de anclaje con ese valor mínimo (OpenCv,2019).

ALGORITMO DE DETECCIÓN HOUGH

Se hizo uso de uno de los métodos contenidos en la librería Opencv, el cual hace uso de la transformada de Hough para detectar circunferencias mediante la siguiente ecuación (Ver ilustración 5).

$$(x - a)^2 + (y - b)^2 = r^2$$

Ilustración 5 Ecuación que describe un circunferencia

Para poder describir la circunferencia debió tener tres parámetros esenciales:

- Centro de la circunferencia (a, b).
- Radio (r).

Al contar con los tres parámetros anteriores se creó un acumulador de las tres dimensiones. Al terminar de completarse el acumulador se fueron buscando los picos, obteniendo el radio y el centro de la circunferencia (OpenCv,2019).

En la ilustración número 6 se ve el resultado de la ejecución del software, con los resultados de cada uno de los pasos, que se ejecutaron para llegar a la detección de balón.

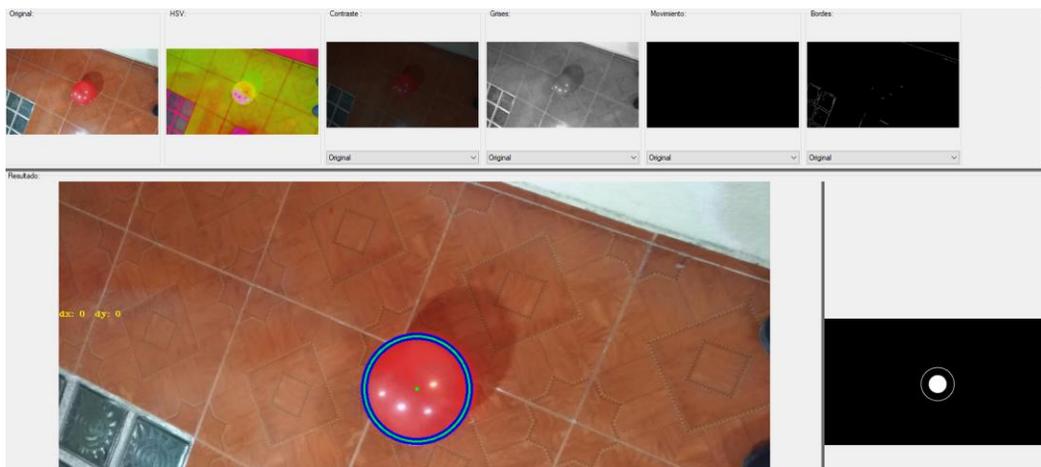


Ilustración 6 Resultados de la detección.

VALIDACIÓN DEL PROTOTIPO

Luego de realizar procesar las carpetas “test con balón” y “test sin balón” se obtuvieron los siguientes resultados:

Tabla 1

Pruebas Del Modelo

Resultados obtenidos de los pasos realizados para la detección junto con el uso del algoritmo de hough

<u>Número de imágenes procesadas con balón</u>	<u>Número de imágenes procesadas sin balón</u>	<u>Fotogramas por segundo</u>	<u>Tasa de acierto para imágenes con balón</u>	<u>Tasa de fallo para imágenes sin balón</u>
500	250	0.185 fps	60%	50%

De acuerdo a los resultados obtenidos, se determinó que el software no es lo suficiente preciso para cumplir con los requerimientos de detección de balones, se evidenció que el sistema de se ve afectado por las variaciones de luz, y con cualquier tipo de ruido que se pueda generar desde la cámara que obtiene los frames.

DESARROLLO PROTOTIPO NÚMERO 2

Desarrollo del prototipo número dos (redes neuronales sobre tensorflow)

ESTRATEGIA DE DETECCIÓN

Se propuso la siguiente estrategia para realizar la detección del balón:

- Creación de un banco de imágenes.
- Diseño de la red neuronal.
- Creación de TFRecord.
- Creación del mapa de etiquetación.
- Configuración de archivo de entrenamiento.
- Entrenamiento del modelo de red diseñada.
- Pruebas del modelo.

CREACIÓN DEL BANCO DE PRUEBAS

Para la creación del banco de pruebas se creó un dataset de imágenes, organizadas en una estructura de carpetas, las cuales se divide en tres: entrenamiento, pruebas y validación, estas contienen 1.834 imágenes de entrenamiento, 475 para pruebas y 500 de validación del modelo. Todas las imágenes de entrenamiento y pruebas tienen unas dimensiones de 300 x 300 píxeles que contienen diferentes objetos, entre los que se encuentran balones de distintos colores, dimensiones e incluso formas como es el caso de los balones de futbol americano.

Para las imágenes de “validación del modelo”, se utilizaron para validar el sistema luego de terminar el entrenamiento, para esto se tomaron 400 imágenes en las cuales se contenían balones y 100 en donde no se encontraban balones, para un total de 500 imágenes distintas con diferentes tamaños.

Para la realización de la etiquetación de cada imagen, tanto para las de pruebas como para las de entrenamiento, se etiqueto cada imagen haciendo uso de la herramienta “LabelImage” en las zonas donde se encontraba el objeto a ser reconocido, en este caso un balón como es evidenciado en la ilustración número 3.

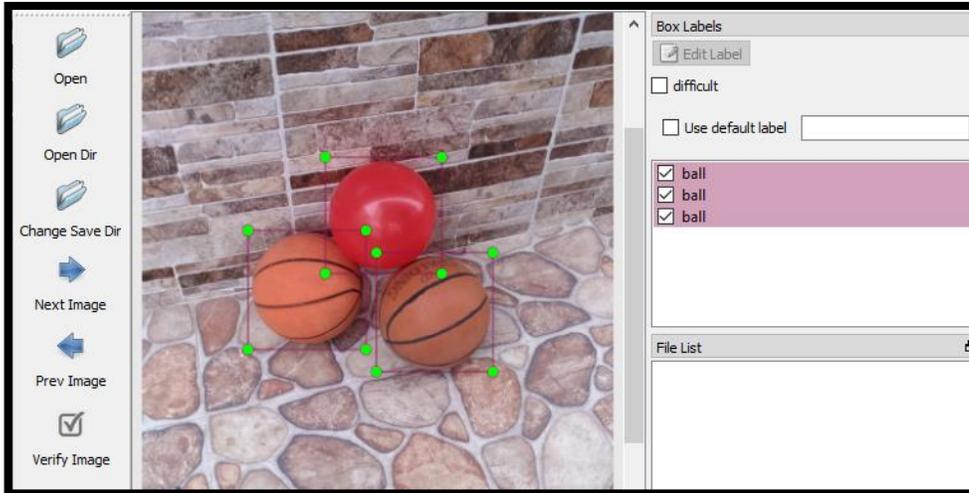


Ilustración 7 Etiquetación de las imágenes

Con la etiquetación correspondiente de cada imagen se obtiene como resultado un archivo con las coordenadas “xmin, ymin” y “xmax, ymax” en un archivo XML en formato PASCAL VOC (PASCAL Visual Object Classes), siendo un estándar que proporciona a las comunidades de visión y aprendizaje automático un conjunto de datos estándar de imágenes y un formato de etiquetación, además de contener procedimientos de evaluación estandarizados (Everingham, Eslami, Van Gool, Williams, Winn, & Zisserman, 2019), a continuación, en la ilustración 4 se presenta un ejemplo de la estructura que se lleva en la etiquetación de cada imagen.

```

<annotation>
+ <folder>test</folder>
+ <filename>21.jpg</filename>
+ <path>C:\tensorflow1\models\research\
+ |   |   object_detection\images\test\21.jpg</path>
+ <source>
+ |   <database>Unknown</database>
+ </source>
+ <size>
+ |   <width>360</width>
+ |   <height>640</height>
+ |   <depth>3</depth>
+ </size>
+ <segmented>0</segmented>
+ <object>
+ |   <name>ball</name>
+ |   <pose>Unspecified</pose>
+ |   <truncated>0</truncated>
+ |   <difficult>0</difficult>
+ |   <bndbox>
+ |   |   <xmin>142</xmin>
+ |   |   <ymin>231</ymin>
+ |   |   <xmax>229</xmax>
+ |   |   <ymax>320</ymax>
+ |   </bndbox>
+ </object>
</annotation>

```

Ilustración 8 Estructura del formato PASCAL VOC para la etiquetación de cada balón.

DISEÑO DE LA RED NEURONAL

El entrenamiento de una red neuronal desde cero conlleva un alto gasto de procesamiento a nivel computacional y requiere de mucho tiempo para completar el entrenamiento, para disminuir ese tiempo de entrenamiento, se hizo uso de The Microsoft Common Objects in Context “coco”, el cual consiste en un conjunto de datos que tiene 91 categorías de objetos comunes, en 2.500.000 imágenes con datos etiquetados y 328.000 instancias de imágenes de referencia (Lin et al., 2014). Se descarta el diseño propio para poder aprovechar las bondades de la transferencia de aprendizaje del modelo mencionado.

Al contar con este conjunto de datos pre entrenado, es posible utilizar una técnica llamada transferencia de aprendizaje, esta consiste en un proceso en el que se toman las ponderaciones de un modelo que ya ha sido entrenado con una gran cantidad de datos para realizar una tarea similar, y luego entrenar el modelo con los datos que se necesitaban para este proyecto (Weiss, Khoshgoftaar, & Wang, 2016).

CREACIÓN DE TFRECORD

Al ya contar con las imágenes previamente etiquetadas, se procedió a la creación de los TFRecord's, los cuales sirven como datos de entrada al modelo de capacitación de la librería TensorFlow. Para la generación de estos archivos se hizo uso de los scripts `xml_to_csv.py` y `generate_tfrecord.py` del conjunto de datos de entrenamiento de TensorFlow haciendo alguna modificación para trabajar con la estructura de directorios definida.

Ejecutando los comandos que se encuentran en la ilustración número 5 se generaron los archivos `train.record` y `test.record` en sobre la ruta `./object_detection`.

Con la generación de estos archivos se entrenó al nuevo clasificador de detección de objetos.

```
python generate_tfrecord.py --csv_input=images\train_labels.csv --image_dir=images\train --output_path=train.record
python generate_tfrecord.py --csv_input=images\test_labels.csv --image_dir=images\test --output_path=test.record
```

Ilustración 9 Comandos para la generación de archivos TFRecord

CREACIÓN DEL MAPA DE ETIQUETACIÓN

Para el banco de imágenes creado, fue necesario crear un mapa de etiquetas, en el cual se realizó una asignación de nombres de clase, en el que se debía contener un id numérico y empezar desde 1, con este mapa de etiquetas se le dice a la red qué es cada objeto al hacer la asignación de los nombres de clase a números de ID de clase.

Para este proyecto solo hay una clase llamada “ball”, el mapa de etiquetas quedo con la siguiente estructura:

```
1  item {
2    id: 1
3    name: 'ball'
4  }
```

Ilustración 10 Estructura de mapa de etiquetas

CONFIGURACIÓN DE ARCHIVO DE ENTRENAMIENTO

Para realizar el entrenamiento de la red neuronal fue necesario realizar una configuración previa al archivo de entrenamiento y al modelo, en la que definieron una serie de parámetros en las que se encuentran:

- Numero de clases las cual fue definida con el número uno puesto que solo se deseaba detectar un objeto.
- Puesto que la red tiene la capacidad de redimensionar las imágenes de entrada se dejaron las variables de redimensionamiento con los mismos tamaños de las imágenes de entrada, puesto que estas ya contenían el tamaño ideal,300 x 300 pixeles, esto con el fin de disminuir los tiempos de procesamiento.
- La tasa de aprendizaje que se defino fue de 0.0002
- Se definió la ruta **/object_detection/ball/model.ckpt** para guardar los puntos de control.
- En la ilustración 7 se presentan las configuraciones de las métricas para el número de ejemplos que se evaluaron, ruta de los archivos record para el entrenamiento y las pruebas, además del archivo label map para la clase definida.

```
train_input_reader: {
  tf_record_input_reader {
    input_path: "C:/object_detection/train.record"
  }
  label_map_path: "C:/object_detection/training/labelmap.pbtxt"
}

eval_config: {
  metrics_set: "coco_detection_metrics"
  num_examples: 475
}

eval_input_reader: {
  tf_record_input_reader {
    input_path: "C:/object_detection/test.record"
  }
  label_map_path: "C:/object_detection/training/labelmap.pbtxt"
  shuffle: false
  num_readers: 1
}
```

Ilustración 11 Configuración de rutas para entrenamiento

ENTRENAMIENTO

Para empezar el entrenamiento de la red neuronal se ejecutó un archivo train.py (ver anexo A). encargado de leer el archivo de configuración de la red.

En la ilustración 8 se aprecia la sentencia encargada de lanzar el proceso de entrenamiento con un parámetro en el cual se le indica la ruta del archivo y el nombre del archivo de configuración, para ver el archivo de configuración,

```
python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_ball.config
```

Ilustración 12 Comando para lanzar ejecución de entrenamiento.

La ejecución del script `train.py` realiza cuatro pasos principales, como se muestra en la ilustración número 9.

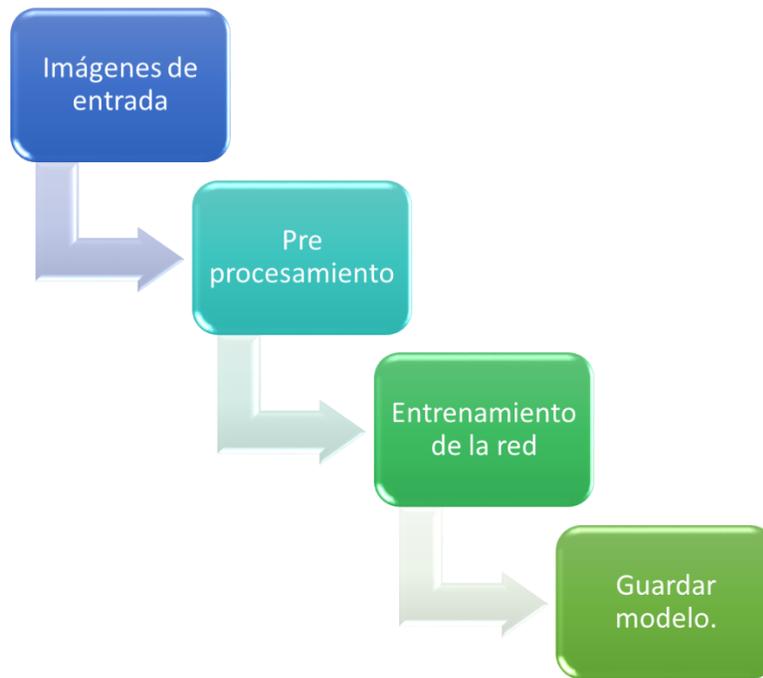


Ilustración 13 Proceso de entrenamiento.

El primer paso, Imágenes de entrada consistió en leer los archivos que se generaron llamados **`train.record`** y **`test.record`**, los cuales contienen los nombres de cada imagen, ancho y alto pixeles, nombre de la clase, `xmin`, `ymin`, `xmax`, `ymax` por cada imagen.

En el segundo paso, pre procesamiento, se normalizaron los valores de entrada, está consistió en re dimensionaron los tamaños de las imágenes.

En el tercer paso, el en entrenamiento la red neuronal, se siguieron los siguientes sub pasos:

1. Se procesa la imagen a través de la CNN para obtener el clasificador.
2. Se ejecutó el mapa de activación a través de una red separada, llamada Red de propuesta de región (RPN), en la que se generó los cuadros / regiones de interés.
3. Para las boxes (cajas) / regiones de interés de la RPN, se usaron varias capas completamente conectadas para así generar las coordenadas más loas cuadros delimitadores.

- Finalmente, se da por terminado el entrenamiento de la red al contar con una pérdida “loss” promedio total de 0.05, este valor se obtuvo luego de 14335 pasos “step” o iteraciones, con una frecuencia de 9.5 segundos por iteración (ver ilustración número 10).

```
Selecionar Anaconda Prompt - python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_ball.config
I0913 13:40:52.422782 15436 learning.py:505] global step 14332: loss = 0.0324 (9.065 sec/step)
INFO:tensorflow:global step 14333: loss = 0.0665 (9.747 sec/step)
I0913 13:41:02.174317 15436 learning.py:505] global step 14333: loss = 0.0665 (9.747 sec/step)
INFO:tensorflow:global step 14334: loss = 0.0324 (9.495 sec/step)
I0913 13:41:11.674691 15436 learning.py:505] global step 14334: loss = 0.0324 (9.495 sec/step)
INFO:tensorflow:global step 14335: loss = 0.0656 (13.267 sec/step)
```

Ilustración 14 Resultados generales durante el entrenamiento.

Como último paso se realizó la generación del gráfico de inferencia congelado, el cual permitió la inferencia, en la ilustración número 11 se muestra el comando para generar el gráfico, tomando el último punto de control el cual fue en el paso 14335, generando como salida el archivo llamado “**frozen_inference_graph.pb**”.

```
python export_inference_graph.py --input_type image_tensor --pipeline_config_path training/faster_rcnn_ball.config --trained_checkpoint_prefix training/model.ckpt-14335 --output_directory inference_graph_20190922
```

Ilustración 15 Comando para la generación del gráfico congelado.

PRUEBAS DEL MODELO

Para verificar a precisión del modelo, se creó un video de a una velocidad de captura 60fps a una resolución de 854 x 480px siendo redimensionadas por la red a un tamaño de 300x300 pixeles, no se grabó bajo un ambiente controlado, es decir, no se modificaron las luces, la cámara no permaneció suspendida en un solo punto, tampoco se hizo ningún tipo de filtro para mejorar la calidad de la imagen, se tomaron como referencia 20 balones de distintos colores y tamaños.

En la realización de las pruebas se usó un hardware con las siguientes características: AMD A10-8700P Radeon R6,10 Compute Cores 4C+ 6G 1.80 GHz con una memoria RAM de 8GB obteniendo los siguientes resultados:

Tabla 2

Prueba Del Modelo

Resultados obtenidos durante la ejecución de la prueba

<u>Tamaño de las imágenes de entrada</u>	<u>Número de frames procesados</u>	<u>Fotogramas por segundo</u>	<u>Tasa de acierto</u>
854 x 480px	10.800	0.24 fps	98 %

En las siguientes ilustraciones se presentan uno de los frames que fue procesado

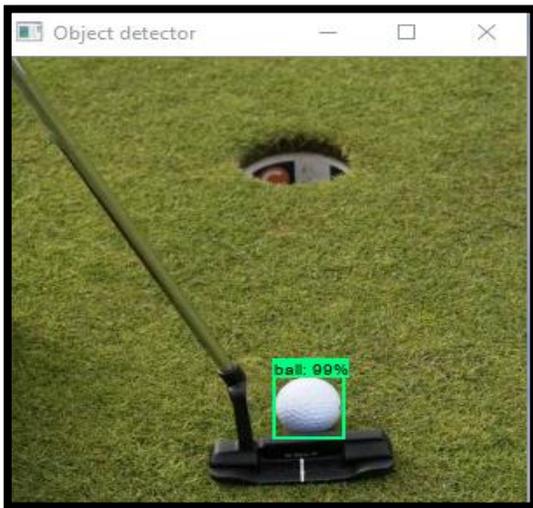


Ilustración 16 Objeto detectado por la red.

```
Anaconda Prompt - python "Object_detection_video.p
FPS: 0.25519941774455357
Coordenadas de detección:
Borde izquierdo: 97.45246767997742
Borde derecho: 207.79207348823547
Borde superior: 130.12137115001678
Borde inferior: 244.63278651237488
FPS: 0.25853267023749565
Coordenadas de detección:
Borde izquierdo: 175.31826496124268
Borde derecho: 194.9873149394989
Borde superior: 134.02752578258514
Borde inferior: 154.62514758110046
FPS: 0.2474623962889703
Coordenadas de detección:
Borde izquierdo: 115.92309772968292
Borde derecho: 183.65680575370789
Borde superior: 134.48638021945953
Borde inferior: 204.56541180610657
FPS: 0.25536347723104014
Coordenadas de detección:
Borde izquierdo: 152.579265832901
Borde derecho: 244.00756359100342
Borde superior: 103.95374894142151
Borde inferior: 193.68215203285217
FPS: 0.24819988240651933
Coordenadas de detección:
Borde izquierdo: 151.55276656150818
Borde derecho: 216.48330688476562
Borde superior: 126.59412324428558
Borde inferior: 189.80791568756104
FPS: 0.24691176622953384
Coordenadas de detección:
Borde izquierdo: 154.86581325531006
Borde derecho: 195.81674337387085
Borde superior: 212.13303208351135
Borde inferior: 252.9996693134308
FPS: 0.246911708088392
```

Ilustración 17 Descripción de coordenadas del objeto detectado y tiempo de proceso por frame.

Luego de la ejecución del video en la cual se contenían los 20 balones, se realizó una segunda prueba, ésta consistió en la creación un video con las mismas características que el anterior, pero en esté no se contenían balones, por el contrario, se grabaron objetos con formas redondas.

Prueba Del Modelo

Resultados obtenidos durante la ejecución de la prueba

<u>Tamaño de las imágenes de entrada</u>	<u>Número de frames procesados</u>	<u>Fotogramas por segundo</u>	<u>Tasa de falsos positivos</u>
854 x 480px	7200	0.24 fps	30 %

De acuerdo con los resultados obtenidos se encuentra que la red detecta algunos objetos como balones entre los que se encuentran platos, pocillos y tapas de gaseosas. Se evidencia que los objetos que fueron detectados como balones cumplen con la mayoría de las características de un balón, como color y forma entre otros, a continuación, se presenta un ejemplo:



Ilustración 19 Detección de plato como balón



Ilustración 18 Detección de tapa de gaseosa como balón

COSTO DEL PROYECTO

Ver anexo C

CONCLUSIONES

El objetivo fundamental de este proyecto consistió en crear un sistema de visión artificial que permitió detectar cuando un balón golpeaba con una pared siendo capaz de determinar cuáles fueron las coordenadas “X”, “Y” con las que chocó.

Así pues, se realizaron dos desarrollos bajo la metodología de prototipos la cual permitió llevar el desarrollo de manera organizada, realizar un refinamiento del software más rápido y reduciendo los tiempos de desarrollo.

Para el primer desarrollo se hizo uso de dos tecnologías; Open cv como librería de visión artificial y .Net (C#) como lenguaje de programación, bajo la implantación de estas dos tecnologías y haciendo uso del algoritmo de hough se obtuvo un prototipo capaz de detectar balones bajo condiciones de luz controladas y contado con una configuración previa para poder llegar a una detección óptima.

Mediante el uso de redes neuronales se logró cumplir con el objetivo del proyecto, cabe mencionar que durante el entrenamiento de la red neuronal la transferencia de aprendizaje cumplió un papel importante, al reducir los tiempos de procesamiento del entrenamiento contado con las ponderaciones de otro modelo previamente entrenado.

Con respecto a las redes neuronales convolucionales se concluye que ofrecen un nivel superior de precisión, aunque requieren de un banco de datos consistente y robusto para realizar predicciones exactas.

Se determinó que las condiciones de luz afectan a los sistemas de visión artificial, causando ruido y pérdida del detalle de los objetos, que a su vez pueden generar falsos positivos en las detecciones.

PROYECCIONES

- Una de las proyecciones que se plantea es realizar la integración con el video juego con el apoyo de los nuevos integrantes de los semilleros de investigación.
- Con base al desarrollo propuesto se aconseja crear un sistema basado en inteligencia artificial el cual permita detectar los niños e identificar quien realiza la jugada y determinar si la está haciendo el ejercicio de manera correcta.

REFERENCIAS

- Benítez, R., Escudero, G., Kanaan, S., & Rodó, D. M. (2014). *Inteligencia artificial avanzada*. Editorial UOC.
- Nogué, A., & Antiga, J. (2012). *Aplicación práctica de la visión artificial en el control de procesos industriales*. Gobierno De España (Ministerio De Educación).
- Owens, N., Harris, C., & Stennett, C. (2003). Hawk-eye tennis system. Paper presented at the 2003 International Conference on Visual Information Engineering VIE 2003, 182-185
- Sutoyo, R., Prayoga, B., Suryani, D., & Shodiq, M. (2015). The implementation of hand detection and recognition to help presentation processes. *Procedia Computer Science*, 59, 550-558.
- De Marco, T., Cazzato, D., Leo, M., & Distanti, C. (2015). Randomized circle detection with isophotes curvature analysis. *Pattern Recognition*, 48(2), 411-421.
- Shaoqing Ren, Kaiming He, Girshick, R., & Jian Sun. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149. doi:10.1109/TPAMI.2016.2577031
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (Jun 2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Paper presented at the 580-587. doi:10.1109/CVPR.2014.81 Retrieved from <https://ieeexplore.ieee.org/document/6909475>
- Girshick, R. (Dec 2015). Fast R-CNN. Paper presented at the 1440-1448. doi:10.1109/ICCV.2015.169 Retrieved from <https://ieeexplore.ieee.org/document/7410526>
- Sande, K. E. A., Uijlings, J. R. R., Gevers, T., & Smeulders, A. (Jan 1, 2011). Segmentation as selective search for object recognition. Paper presented at the Retrieved from <https://www.narcis.nl/publication/RecordID/oai:cwi.nl:20028>
- Alberti, M. (2019). *Uso del aprendizaje profundo para el reconocimiento de objetos | DeepLearningItalia*. Retrieved from <https://www.deeplearningitalia.com/uso-del-aprendizaje-profundo-para-el-reconocimiento-de-objetos/>

- Soto, I. (2018). Diferencia entre Brillo y Contraste - Diferencias.info. Retrieved 29 May 2019, from <https://diferencias.info/diferencia-entre-brillo-y-contraste/>
- ASALE, R. (2019). Inteligencia. Retrieved from <https://dle.rae.es/?id=LqtyoaQ>
- UOC, P. (2019). La visión por computador: Una disciplina en auge - Informatica ++. Retrieved from <http://informatica.blogs.uoc.edu/2012/04/19/la-vision-por-computador-una-disciplina-en-auge/>
- CUDA. (2019). CUDA Zone. Retrieved from <https://developer.nvidia.com/cuda-zone>
- OpenCV. (2019). About opencv.. Retrieved from <https://opencv.org/about/>
- TensorFlow. (2019). Why TensorFlow. Retrieved from <https://www.tensorflow.org/>
- Anaconda. (2019). What is Anaconda. Retrieved from <https://www.anaconda.com/what-is-anaconda/>
- MathWorks (2019). Deep Learning: Tres cosas que es necesario saber. Retrieved from <https://la.mathworks.com/discovery/deep-learning.html>
- MathWorks (2019). Redes Neuronales Convolucionales. Retrieved from <https://la.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>
- Amit, Y., & Felzenszwalb, P. (2019). Object Detection. Retrieved from <http://cs.brown.edu/people/pfelzens/papers/detection.pdf>
- Weiss, K., Khoshgoftaar, T., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3(1), 1-40. doi:10.1186/s40537-016-0043-6
- Everingham, M., Eslami, S., Van Gool, L., Williams, C., Winn, J., & Zisserman, A. (2019). The Pascal Visual Object Classes Challenge: A Retrospective. Retrieved 1 September 2019, from <https://link.springer.com/article/10.1007/s11263-009-0275-4>
- Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., . . . Dollár, P. (2014). Microsoft COCO: Common objects in context Retrieved from https://www.openaire.eu/search/publication?articleId=od_____18::976f1595a605cd4d8c8ea454e79c42c8

Weiss, K., Khoshgoftaar, T., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3(1), 1-40. doi:10.1186/s40537-016-0043-6

INTEL. (2019). Training pipeline [Image]. Retrieved from <https://software.intel.com/sites/default/files/managed/5a/7d/traffic-light-detection-using-tensorflow-object-detection-api-fig4.png>

OpenCV: Graph API: Converting image from one color space to another. (2019). Retrieved 23 September 2019, from https://docs.opencv.org/trunk/dc/d38/group__gapi__colorconvert.html#gad78d0756339b84ed0fed49ac3b7dae01

Fernando, S. (2019). Gaussian Blur. Retrieved 1 October 2019, from <https://www.opencv-srf.com/2018/03/gaussian-blur.html>

OpenCV: Graph API: Converting image from one color space to another. (2019). Retrieved 23 September 2019, from https://docs.opencv.org/trunk/dc/d38/group__gapi__colorconvert.html#gad78d0756339b84ed0fed49ac3b7dae01

Applications - Applications - Fast Light Toolkit (FLTK). (2019). Retrieved 5 July 2019, from <https://www.fltk.org/applications/index.php>

Inc, T. (2019). OpenGL Overview. Retrieved 9 October 2019, from <https://www.opengl.org/about/>

MathWorks, Reconocimiento de objetos. (2019). Retrieved 9 October 2019, from <https://la.mathworks.com/solutions/image-video-processing/object-recognition.html>