Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

DETERMINACIÓN DE LA DISTRIBUCIÓN DE PERSONAS EN UN SALÓN DE CLASES POR MEDIO DE UN ALGORITMO DE VISIÓN POR COMPUTADORA UTILIZANDO CÁMARA OJO DE PEZ, IMPLEMENTADO EN EL DEPARTAMENTO DE INGENIERÍA MECÁNICA, UNIVERSIDAD NACIONAL CHIAO TUNG, HSINCHU, TAIWÁN

Daniel Enrique Alvizures Rodríguez
Gilberto Antonio Rosales Vaquin
Asesorado por el Ing. Marlon Antonio Pérez Türk

Guatemala, octubre de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

DETERMINACIÓN DE LA DISTRIBUCIÓN DE PERSONAS EN UN SALÓN DE CLASES POR MEDIO DE UN ALGORITMO DE VISIÓN POR COMPUTADORA UTILIZANDO CÁMARA OJO DE PEZ, IMPLEMENTADO EN EL DEPARTAMENTO DE INGENIERÍA MECÁNICA, UNIVERSIDAD NACIONAL CHIAO TUNG, HSINCHU, TAIWÁN

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

DANIEL ENRIQUE ALVIZURES RODRÍGUEZ
GILBERTO ANTONIO ROSALES VAQUIN
ASESORADO POR EL ING. MARLON ANTONIO PÉREZ TÜRK

AL CONFERÍRSELES EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton De León Bran
VOCAL IV	Br. Christian Moisés De La Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

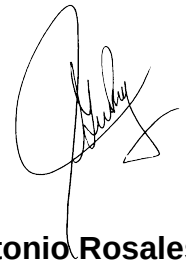
DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. Herman Igor Veliz Linares
EXAMINADOR	Ing. José Ricardo Morales Prado
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presentamos a su consideración nuestro trabajo de graduación titulado:

DETERMINACIÓN DE LA DISTRIBUCIÓN DE PERSONAS EN UN SALÓN DE CLASES POR MEDIO DE UN ALGORITMO DE VISIÓN POR COMPUTADORA UTILIZANDO CÁMARA OJO DE PEZ, IMPLEMENTADO EN EL DEPARTAMENTO DE INGENIERÍA MECÁNICA, UNIVERSIDAD NACIONAL CHIAO TUNG, HSINCHU, TAIWÁN

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha abril de 2019.



Daniel Enrique Alvizures Rodríguez **Gilberto Antonio Rosales Vaquin**

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL. 24439500 EXT. 1534

Guatemala, 21/mayo/2020

Ingeniero Carlos Azurdia
Coordinador Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería, USAC
Presente

Estimado Ing. Azurdia:

Por este medio informo que he revisado y aprobado el Trabajo de Tesis titulado: "DETERMINACIÓN DE LA DISTRIBUCIÓN DE PERSONAS EN UN SALÓN DE CLASES POR MEDIO DE UN ALGORITMO DE VISIÓN POR COMPUTADORA UTILIZANDO CÁMARA OJO DE PEZ, IMPLEMENTADO EN EL DEPARTAMENTO DE INGENIERÍA MECÁNICA, UNIVERSIDAD NACIONAL CHIAO TUNG, HSINCHU, TAIWÁN", de los estudiantes Daniel Enrique Alvizures Rodríguez (Carnet número 201313963) , quien se identifica con DPI número 2709 77236 0105 y Gilberto Antonio Rosales Vaquin (Carnet número 201318565), quien se identifica con DPI número 2712 45670 0101.

Con base en la evaluación realizada hago constar que he evaluado la calidad, validez, pertinencia y coherencia de los resultados obtenidos en el trabajo presentado, por lo cual el trabajo evaluado cuenta con mi aprobación.

Agradeciendo su atención y deseándole éxitos en sus actividades profesionales me suscribo.

Atentamente,

"Id y Enseñad a Todos"

MARLON ANTONIO PEREZ TURK
INGENIERO EN CIENCIAS Y SISTEMAS
COLEGIADO No. 4492

MA. Ing. Marlon Antonio Pérez Türk
Asesor
Colegiado No. 4492

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 29 de mayo del 2020

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **DANIEL ENRIQUE ALVIZURES RODRÍGUEZ** con carné **201313963 y CUI 2709 77236 0105** y **GILBERTO ANTONIO ROSALES VAQUIN** con carné **201318565 y CUI 2712 45670 0101,** titulado: **"DETERMINACIÓN DE LA DISTRIBUCIÓN DE PERSONAS EN UN SALÓN DE CLASES POR MEDIO DE UN ALGORITMO DE VISIÓN POR COMPUTADORA UTILIZANDO CÁMARA OJO DE PEZ, IMPLEMENTADO EN EL DEPARTAMENTO DE INGENIERÍA MECÁNICA, UNIVERSIDAD NACIONAL CHIAO TUNG, HSINCHU, TAIWÁN"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
INGENIERIA EN CIENCIAS Y SISTEMAS
COORDINACION TESIS

ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÌA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación ***"DETERMINACIÓN DE LA DISTRIBUCIÓN DE PERSONAS EN UN SALÓN DE CLASES POR MEDIO DE UN ALGORITMO DE VISIÓN POR COMPUTADORA UTILIZANDO CÁMARA OJO DE PEZ, IMPLEMENTADO EN EL DEPARTAMENTO DE INGENIERÍA MECÁNICA, UNIVERSIDAD NACIONAL CHIAO TUNG, HSINCHU, TAIWÁN",*** *realizado por los estudiantes, Daniel Enrique Alvizures Rodríguez y Gilberto Antonio Rosales Vaquin, aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

Digitally signed by Carlos Gustavo Alonzo
DN: 2.5.4.13=Profesional Titulado, c=GT, l=Guatemala / Guatemala, street=Via 5 3-65 zona 4 Ed. El Angel 5to nivel of 52, 2.5.4.20=22347420, ou=NA, o=NA, title=Ingeniero en Ciencias y Sistemas Colegiado. 6358, serialNumber=2278 03167 0101, 2.5.4.45=29020980, 2.5.4.27=06/03/79, email=carlosalonzo@infoutilitygt.com, cn=Carlos Gustavo Alonzo
Date: 2020.10.07 21:48:52 -06'00'

Alonzo

Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 04 de octubre 2020

DTG. 303.2020.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DETERMINACIÓN DE LA DISTRIBUCIÓN DE PERSONAS EN UN SALÓN DE CLASES POR MEDIO DE UN ALGORITMO DE VISIÓN POR COMPUTADORA UTILIZANDO CÁMARA OJO DE PEZ, IMPLEMENTADO EN EL DEPARTAMENTO DE INGENIERÍA MECÁNICA, UNIVERSIDAD NACIONAL CHIAO TUNG, HSINCHU, TAIWÁN,** presentado por los estudiantes universitarios: **Daniel Enrique Alvizures Rodríguez y Gilberto Antonio Rosales Vaquin,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Inga. Anabela Cordova Estrada
Decana

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANA
FACULTAD DE INGENIERÍA

Guatemala, octubre de 2020

AACE/asga

ACTO QUE DEDICO A:

Dios	Por ser siempre mi guía y darme fuerzas y sabiduría para alcanzar esta meta.
Mis padres	José Pablo Alvizures Muralles y María Verónica Rodríguez Chinchilla por darme siempre su apoyo e inculcarme los valores que me han hecho la persona que soy.
Mis hermanos	José Pablo, Angel Gabriel, Axel Estuardo y Verónica María Alvizures, por siempre darme palabras de aliento y apoyarme en el día a día.
Mi cuñada y sobrinas	Cindy Chocón, Valentina y María José Alvizures, por alegrarme con su presencia.
Mi mejor amiga	Emily Castañón, por siempre darme aliento y apoyarme en cada momento difícil de la carrera.
Mis tíos	Por brindarme su apoyo en cada momento del desarrollo de mi carrera.
Mis primos	Por ser como mis hermanos y ayudarme en todo momento.

Mis abuelos	Por enseñarme a trabajar desde pequeño y siempre mostrarme el camino del bien.
Ing. Ricardo Pontaza	Por su amistad brindada en la estadía en Taiwán y ser un ejemplo de un distinguido estudiante y profesional para mi persona.

Daniel Enrique Alvizures

ACTO QUE DEDICO A:

Dios	Por ser lo primero en mi vida.
Mis padres	Gilberto Rosales y Odilia Vaquin, por ser lo más importante en mi vida.
Mis hermanos	Alejandra Laura y Rene Rosales, por ser una razón en mi esfuerzo de ser siempre un buen ejemplo.
Mis primos	Por ser grandes ejemplos.
Mis tíos	Por enseñarme que en la vida hay que trabajar duro para lograr lo que uno se propone.
Mi abuela	Esperanza García viuda de Rosales, por ser un gran ejemplo de lucha en mi vida.

Gilberto Rosales Vaquin

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser mi casa de estudios y formarme como profesional.
Facultad de Ingeniería	Por brindarme educación de calidad en el campo de la ingeniería en ciencias y sistemas.
Mis amigos de la Facultad	Por compartir conmigo cada momento durante estos años de carrera.
Ing. Marlon Turk	Por brindarme su tiempo y su apoyo en el proyecto de graduación.
National Chiao Tung University	Por ser mi segunda casa de estudios y permitirme adquirir conocimientos durante seis meses en sus laboratorios en Taiwán.

Daniel Enrique Alvizures

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser mi casa de estudios y formarme como profesional.
Facultad de Ingeniería	Por brindarme educación de calidad en el ámbito de la ingeniería.
Mis amigos de la Facultad	Por apoyo durante los años de universidad.
Ing. Marlon Turk	Por el apoyo brindado para obtener la estancia de investigación en Taiwán y el acompañamiento en la realización de este trabajo de graduación.
National Chiao Tung University	Por esos seis meses de la estancia de investigación que sirvieron para poder redactar el proyecto de graduación.

Gilberto Rosales Vaquin

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLA

LISTA DE SÍMBOLOS

Símbolo	Significado
cm	Centímetro
fps	Fotograma por segundo
mAP	Media promedio de precisión
m	Metro
ms	Milisegundos

GLOSARIO

Computer Vision	Disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratados por un ordenador.
Data	Datos, conjunto de datos o información desordenada.
Datasets	Colección de datos habitualmente tabulada.
Deep Learning	Conjunto de algoritmos de clase aprendizaje automático, que intenta modelar abstracciones de alto nivel en datos usando arquitecturas compuestas de transformaciones no lineales múltiples.
Demux	Demultiplexores es un circuito combinacional con una entrada y varias salidas de datos.
FeedBack	Devolución de una señal modificada a su emisor.
FeedForward	También llamada prealimentación, describe un tipo de sistema que reacciona a los cambios en su entorno, normalmente para mantener algún estado concreto del sistema.

Fisheye	Objetivo ojo de pez a aquellos cuyo ángulo de visión es extremadamente grande, de 180° o más.
Kernel	En visión computacional un kernel es un filtro de n*n dimensiones.
Machine Learning	Disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden autónomamente.
Mpeg	Nombre de un grupo de estándares de codificación de audio y vídeo normalizados por el grupo MPEG.
Mux	Multiplexores son circuitos combinacionales con varias entradas y una única salida de datos.
Pooling	Agrupación de recursos con el fin de maximizar la ventaja o minimizar tiempo de procesamiento de imágenes.
XML	XML es un subconjunto de SGML (Estándar Generalised Mark-up Language), simplificado y adaptado a Internet.

RESUMEN

El aprendizaje automático es un tema de las ciencias computacionales con mucho auge en las últimas décadas esto dado al bajo costo de hardware, ha permitido explorar otras herramientas para el procesamiento de imágenes, a través de esta investigación se quiere desarrollar un algoritmo de visión por computación capaz de calcular la distribución de un grupo de personas en un salón de clases.

El trabajo de implementación fue realizado en la Universidad Nacional Chiao Tung de Taiwán (NTCU), en un periodo de seis meses, tiempo que duraba la estancia, donde se realizaron las distintas mediciones, cálculos y pruebas para la mejora y búsqueda de un algoritmo de visión por computación para solventar el problema antes mencionado. Con las modificaciones necesarias a los parámetros del algoritmo se alcanzó un nivel aceptable de los resultados requeridos por el departamento de ingeniería mecánica de la universidad en cuestión.

Durante los primeros meses de la estancia de investigación en Taiwán se realizó una búsqueda entre distintos algoritmos ya desarrollados por otras universidades del mundo, se efectuaron las pruebas a los primeros algoritmos seleccionados, de estos se seleccionó únicamente uno, se puso a prueba y mejora durante el tercer y cuarto mes, los resultados obtenidos de este algoritmo se presentan en este trabajo junto con las pruebas a las que fue sometido durante los últimos meses de la estancia, por poca importancia únicamente se presenta el nombre de los primeros algoritmos.

OBJETIVOS

General

Determinar la distribución de personas en un salón de clases por medio de un algoritmo de visión por computadora utilizando cámara ojo de pez.

Específicos

1. Implementar y entrenar un algoritmo de visión por computadora que sea capaz de reconocer y calcular el número de personas en un salón de clases utilizando una cámara de ojo de pez.

2. Determinar una correlación matemática para aproximar las posiciones reales de las personas en relación a las obtenidas por el algoritmo de visión por computadora.

3. Implementar un servidor de transmisión de video en tiempo real para ser analizado por el algoritmo de visión por computadora.

HIPÓTESIS

A través de los años han surgido innovaciones que han impactado la forma de vida cotidiana de toda la humanidad, cada una de estas innovaciones han formado parte del comportamiento y desarrollo de grandes ciudades e imperios. Cada uno, de estas innovaciones repercuten directamente en la formación de nuevos campos o ramas de conocimiento, en los cuales es necesario la aplicación del conocimiento humano para desarrollarlas e implementarlas a favor de la humanidad.

Una gran innovación fue el descubrimiento de la manipulación de la energía eléctrica la cual ha sido, por más de 100 años, un campo de interés por un numeroso grupo de personas. Con el paso de los años estas innovaciones y la combinación de ellas han creado nuevos campos como la inteligencia artificial, que se ha convertido en una parte de interés común. De la misma forma que todas las innovaciones se espera que la inteligencia artificial se propague a cada aspecto de la vida cotidiana del ser humano.

Siguiendo los pasos de dicha evolución, la implementación de un algoritmo de inteligencia artificial capaz de realizar análisis de visión por computadora, para determinar la distribución de personas en un salón de clases utilizando una cámara de ojo de pez, es una solución que ayudará a optimizar otros sistemas conectados, como el control lumínico del salón de clases y el sistema de aire acondicionado, brindando una posibilidad de optimizar el consumo de energía eléctrica de estos sistemas.

Hipótesis nula:

Es viable la utilización de un algoritmo de inteligencia artificial capaz de realizar análisis de visión por computadora propuesto, para determinar la distribución de personas en un salón de clases utilizando una cámara de ojo de pez, debido a que logra superar un noventa y cinco por ciento de certeza en el reconocimiento de las personas.

Hipótesis alternativa:

La implementación del algoritmo de inteligencia artificial capaz de realizar análisis de visión por computadora, para determinar la distribución de personas en un salón de clases utilizando una cámara de ojo de pez y no es capaz de superar un noventa y cinco por ciento de certeza en el reconocimiento de las personas.

INTRODUCCIÓN

El concepto de inteligencia artificial como tal no es nuevo ya que aparece por primera vez por 1956 cuando John McCarthy, Marvin Minsky y Claude Shannon acuñaron este término durante la conferencia de Darthmounth, esta conferencia reunió un grupo de 10 científicos para discutir acerca de las máquinas y su posibilidad de comportarse de manera inteligente.

En la última década, el desarrollo de la inteligencia artificial ha tomado auge debido al poder computacional y de redes de comunicación de las cuales hoy en día se dispone. A lo largo de la investigación se quiere explorar el entrenamiento de modelos de inteligencia artificial dedicados a visión por computadora, para determinar la distribución de las personas en un salón de clases, con el objetivo de proveer información para controlar y administrar la iluminación de este ambiente.

Hoy en día está muy de moda el concepto de visión computacional, pero ¿hasta dónde es efectivo un computador para organizar y clasificar distintos tipos de objetos?

También se pondrán a prueba los distintos algoritmos de visión computacional que existen de manera tanto comercial como en estudios aun académicos de varias universidades alrededor del mundo, y presentar de manera clara y concisa cual es el mejor algoritmo para identificar y controlar la distribución de las personas en un ambiente cerrado.

En el presente documento se presenta las posibles soluciones y resultados obtenidos del experimento realizado en el departamento de Ingeniería Mecánica de la Universidad Nacional de Taiwán (NTCU).

1. IDENTIFICACION DEL PROBLEMA

En el transcurso de los años, el avance tecnológico desenfrenado que ha tenido el desarrollo del software; ha llevado a la búsqueda de alternativas en las tareas cotidianas del ser humano, con el objetivo de facilitar y optimizar el desarrollo de estás. El campo de la inteligencia artificial es uno de los mayores involucrados, debido a que este campo está interesado en brindar software que puede realizar tareas autónomamente, teniendo como ejemplo, los nuevos modelos de automóviles ofrecidos por la compañía Tesla Inc., a los que se les está incorporando software capaz de realizar las tareas del piloto; tan solo con el uso de sensores y cámaras que captan el entorno del vehículo y alimentan un algoritmo de inteligencia artificial sofisticado, que es el encargado de la toma de decisiones y manejar el automotor. Parte fundamental para que estas funciones implementadas sean posibles, es el avance que se ha logrado en la investigación y descubrimiento de algoritmos de visión por computadora, que al final, son los encargados de llevar el control de las posiciones y ubicaciones de los objetos que se encuentran involucrados en el entorno en que se movilizan estos vehículos.

La visión por computadora generalmente se encuentra conformada por una serie de cámaras, ya sea que capten video o fotografías, son las encargadas de brindar la materia prima a los algoritmos, que se encargan posteriormente de analizar estas fotografías o videos tomando en cuenta las formas o combinación de píxeles para lograr determinar los objetos contenidos en estas fuentes. Estos algoritmos de visión por computadora llevan un proceso inicial de entrenamiento, y se realiza para enseñar a identificar los objetos que se encuentran dentro de las fuentes de entrada.

En la actualidad, en la mayoría de los salones de clases, se utilizan los sistemas de iluminación que no consideran la distribución de las personas que se encuentran dentro, por lo que se genera un gran consumo de energía eléctrica innecesaria. Este problema se ve reflejado en muchas universidades en todo el planeta, causando un gran impacto en la cantidad de energía eléctrica que se consume y no es utilizada eficientemente. Si se considera el escenario donde se encuentran solo el cincuenta por ciento de la capacidad de estudiantes dentro de un salón de clases y se mantiene la totalidad de luces encendidas, se está generando un consumo inapropiado de energía, debido a que se está empleando el doble de energía de la que es necesario.

Este es el caso donde entra en juego la inteligencia artificial y sus algoritmos de visión por computadora, que proveen las herramientas para poder realizar un análisis de las personas que se encuentran dentro de un salón de clases utilizando cámaras, como fuente principal de entrada, y poder determinar la distribución y ubicación de éstas, beneficiando así el control del uso de energía eléctrica conforme a la cantidad de personas que se encuentren en este lugar.

2. APRENDIZAJE DE MÁQUINAS

El aprendizaje de máquinas es la rama de la inteligencia artificial que se encarga del estudio de algoritmos que proveen a las máquinas el aprendizaje necesario para afrontar, llevar a cabo, ciertas actividades para mejorar la ejecución de éstas a través de las distintas pruebas y ejecuciones. El aprendizaje automático utiliza la detección de patrones para realizar predicciones o tomar decisiones en entornos de incertidumbre.

Un ejemplo clásico de este tipo de algoritmos son los detectores de spam de los correos electrónicos, se provee al algoritmo de data sobre rasgos que pueden caracterizar a un correo de spam y en base a los distintos métodos de aprendizaje que existen, este crea un perfil de los correos que podrían calificarse como spam. Existen distintos tipos de algoritmos de *Machine Learning* que se explicarán más adelante.

Definición: campo de estudio que proporciona a las computadoras la capacidad de aprender sin ser programadas explícitamente. El aprendizaje automático es ideal para problemas en los cuales se necesita modificaciones constantes a las reglas o cambios manuales durante la ejecución, como se vio al inicio, esto no es inconveniente para los algoritmos de aprendizaje automático. Otro campo por explorar con estos algoritmos es de las soluciones complejas, para las cuales aún no existe solución, o el enfoque utilizado es muy difícil para los humanos y no para las máquinas, es allí donde se divisa más aplicaciones para el futuro del aprendizaje de las máquinas. Los entornos cambiantes no son problemas para el *Machine Learning* ya que estos se adaptan a la evolución de la data en el tiempo, por último, también los

algoritmos de inteligencia artificial proporcionan una ventaja del uso de grandes cantidades de datos, que como seres humanos se es incapaz de utilizar y dar un sentido útil.

2.1. Consideraciones generales

A continuación, se describen las consideraciones generales.

2.1.1. Data

No hay Inteligencia Artificial, tampoco *Machine Learning*, si no existe data o datos, también éstos ayudan a definir el tipo de problema al que el algoritmo se enfrenta esto ayuda cuando el problema al que se enfrenta el algoritmo es nuevo, ya que problemas con data parecida suelen tener soluciones parecidas. La data puede provenir de distintos medios, desde una base de datos hasta un archivo de texto y desde una simple lista hasta una matriz, no importa de donde venga siempre y cuando alimente el algoritmo y este sea capaz de realizar una correcta inferencia y a la vez aprender en las distintas corridas.

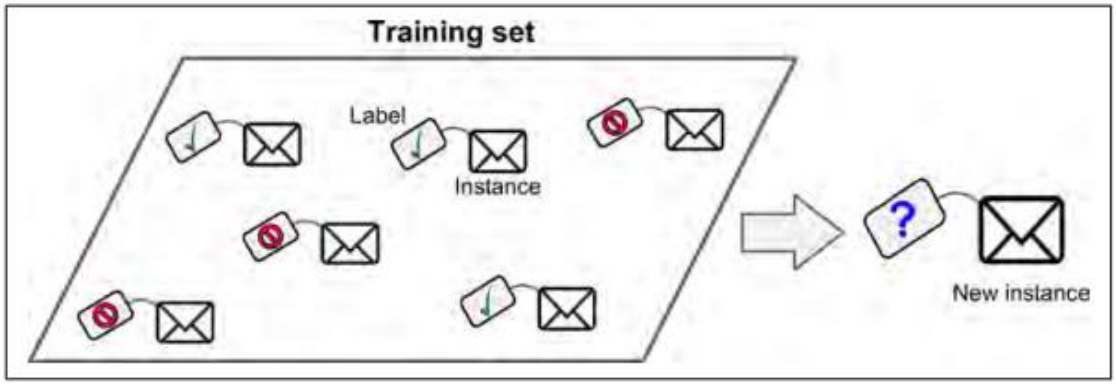
2.1.2. Tipos de *Machine Learning*

Los tipos de *Machine Learning* se pueden clasificar según el tipo de supervisión dado durante las corridas de aprendizaje, entre ellos se encuentra el supervisado, el no supervisado y el semisupervisado.

- Supervisado: en el aprendizaje supervisado, la data de entrenamiento, que se provee al algoritmo también incluye información deseada. Se entrena al algoritmo entregando preguntas llamadas características y asignando su respectiva respuesta llamadas etiquetas. Con las

preguntas y respuestas el algoritmo entrena y da como resultado predicciones para después ser utilizadas por el usuario.

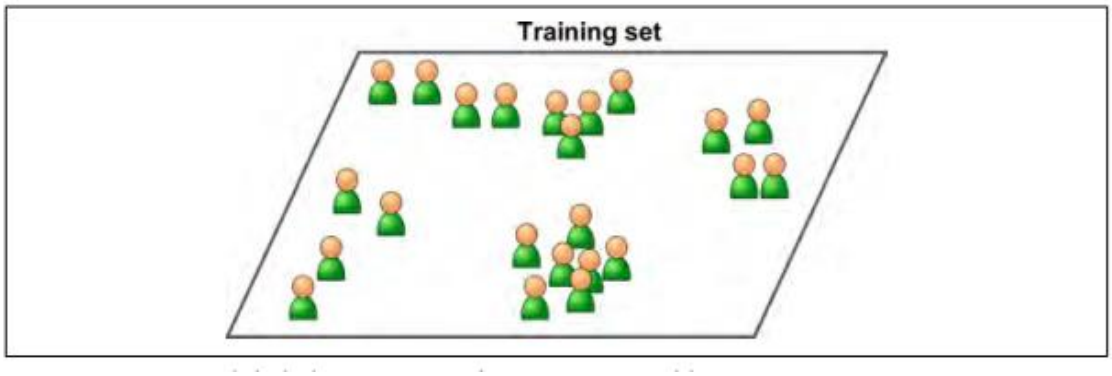
Figura 1. **Correo no deseado**



Fuente: GÉRON, Aurélien. *Hands-on machine learning with Scikit-learn and TensorFlow*. p. 8.

- No supervisado: a diferencia del supervisado este tipo de algoritmos trabajan y aprenden sin necesidad de data con resultados esperados, únicamente se proporcionan los datos de los cuales el algoritmo deberá de realizar las diferentes fases de aprendizaje.

Figura 2. **Training set**



Fuente: GÉRON, Aurélien. *Hands-on machine learning with Scikit-learn and TensorFlow*. p. 10.

2.1.3. Usos del *Machine Learning*

Existen dos razones por las que se decide utilizar un algoritmo de aprendizaje automático sobre un algoritmo programado y desarrollado por un humano: complejidad y adaptación. La capacidad que provee el *Machine Learning* de aprender a partir de distintas y nuevas fuentes de datos, constituye una mejora en los algoritmos de forma autónoma sin necesidad de interferencia humana. Las Tareas Rutinarias también son otro tipo de actividades que se le da muy bien a los algoritmos de *Machine Learning*, por ejemplo, tareas básicas como la conducción, el reconocimiento de imágenes o de caracteres. Los algoritmos que aprenden con su experiencia han logrado buenos resultados en estos tres campos una vez expuestos.

2.2. *Deep Learning*

Otra de las ramas importantes en el campo de la inteligencia artificial es el aprendizaje profundo (*Deep Learning* o DL en inglés), que a diferencia del aprendizaje automático no requiere de tanta supervisión.

El *Deep Learning* tiene su base en las redes neuronales que permiten que el ordenador razone y aprenda por sí mismo sin necesidad de intervención humana. No se requiere de una definición de cada una de las respuestas esperadas en los patrones que se buscan si no automáticamente estas respuestas se generan a partir de la manipulación de los datos.

2.2.1. Aplicaciones

- Reconocimiento de voz
- Reconocimiento de imágenes
- Reconocimiento de manuscritos
- Descubrimiento de componentes farmacéuticos
- Procesamiento del lenguaje natural
- Señales de tiempo

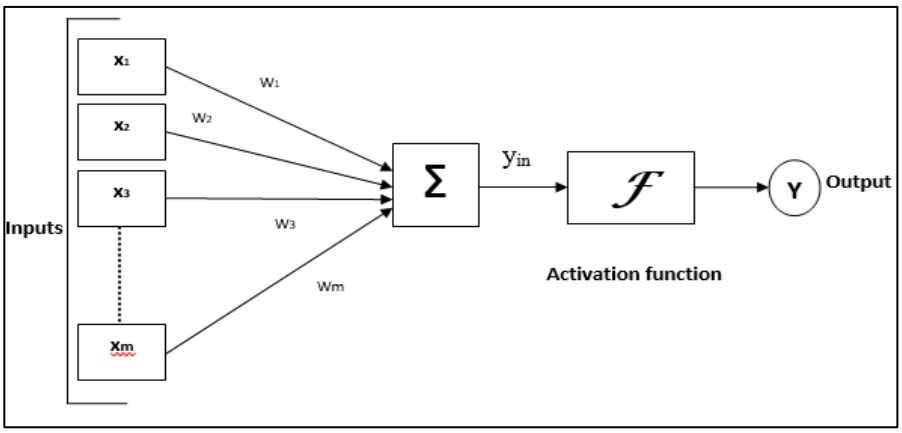
Con el aprendizaje profundo se busca crear algoritmos de aprendizaje capaces de modelar arquitecturas complejas que combinan diferentes transformaciones con ayuda elemental de las neuronas y estas combinadas forman lo que se llama redes neuronales. El crecimiento de esta rama de la inteligencia artificial ha permitido el avance significativo en los campos desde el procesamiento de imágenes y de sonido hasta el campo de visión por computadora y procesamiento de lenguaje.

2.3. Neurona

Es la unidad básica de procesamiento en el *Deep Learning* y las redes neuronales, la neurona no es más que una función matemática que cumple un papel elemental pero importante en todo el proceso de aprendizaje del algoritmo. Al igual que las neuronas biológicas tiene conexiones de entrada a través de las cuales recibe estímulos exteriores, pasa al interior de la neurona donde se realiza el cálculo correspondiente y se obtiene una salida. Dentro de la neurona la función matemática realiza una suma ponderada con todos los valores ingresados junto con el peso de cada uno de los valores w, ver figura 3.

El funcionamiento de la neurona es muy básico y elemental para aspectos y casos más complejos se debe usar la combinación de varias neuronas y es allí donde nace el concepto de red neuronal.

Figura 3. **McCulloh-Pitts model, 1949**



Fuente: TutorialsPoint. *Red neuronal artificial - conceptos básicos*. https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm. Consulta: 25 de febrero de 2020.

2.4. Red neuronal

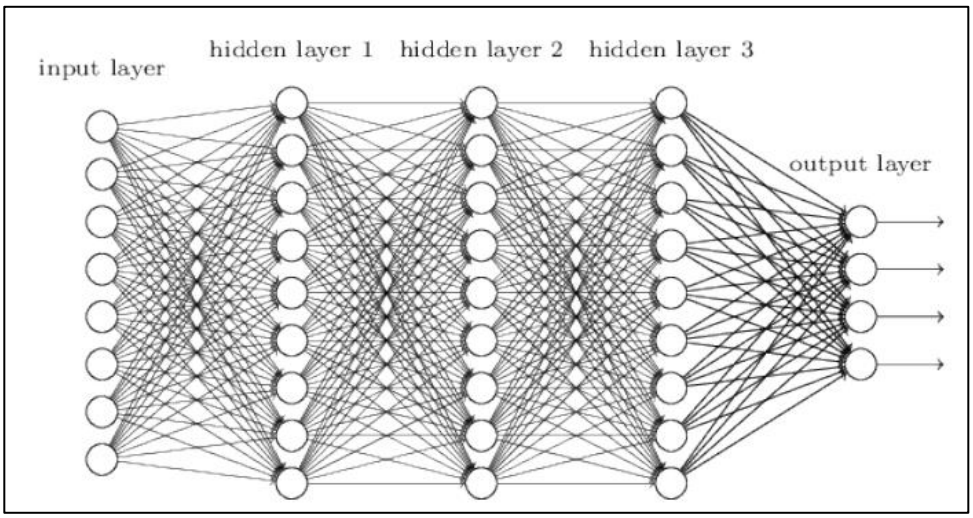
El Dr. Michael Nielsen, define una red neuronal como: "un sistema de computación compuesto por una serie de elementos de procesamiento simples y altamente interconectados, que procesan la información por su respuesta de estado dinámico a entradas externas".[1]

La idea con las redes neuronales es crear artificialmente un dispositivo con funcionamiento semejante al cerebro humano y sus redes de neuronas para aprender y razonar de forma autónoma. Su base es imitar las neuronas y sus dendritas aprovechando el poder computacional que actualmente se dispone y modelar información compleja.

Las redes neuronales artificiales están formadas de nodos que cumplen de forma similar con la función de las neuronas en el cerebro y que comparten información con las demás neuronas de la capa a la que pertenecen. Existe la primera capa de neuronas llamada capa de entrada, por lógica se asume es el conjunto de neuronas que reciben los impulsos de entrada provenientes del exterior y lo envían a la siguiente capa. La siguiente capa que se encuentra puede estar formada por varias capas y es llamada capa oculta, y es la encargada de realizar el trabajo de la red. La última capa que se encontrará en la red neuronal es la capa de salida, esta proporcionará la salida y los resultados al exterior. Ver figura 4.

[1] NIELSEN, Michael. *Neural Networks and Deep Learning*. p. 215.

Figura 4. **Red neuronal**



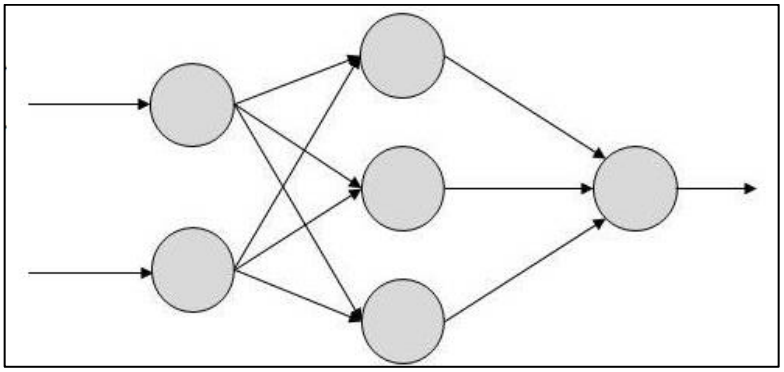
Fuente: NIELSEN, Michael. *Neural Networks and Deep Learning*. p. 169.

Como se vio en la sección anterior las neuronas disponen de una variable llamada peso, en el caso de la red neuronal el aprendizaje se da gracias a la modificación de los pesos de las conexiones entre las neuronas.

2.5. Tipos de redes neuronales

- *FeedForward*: es el tipo de topología de red neuronal en la que las neuronas envían la información en una sola dirección, no existen bucles de retroalimentación.

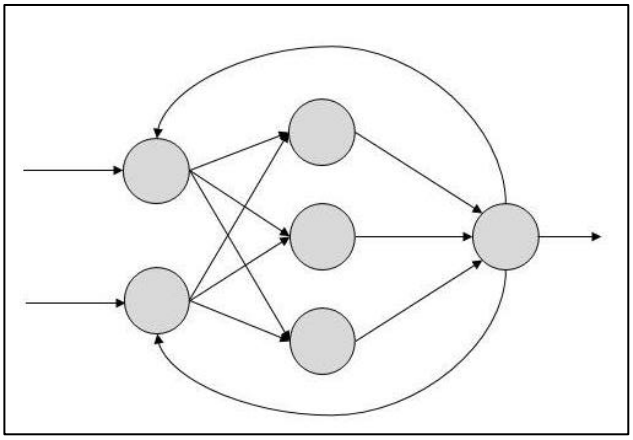
Figura 5. **Red neuronal *FeedForward***



Fuente: TutorialsPoint. *Tutorial de red neuronal artificial*. https://www.tutorialspoint.com/artificial_neural_network. Consulta: 25 de febrero de 2020.

- *FeedBack*: son redes en las que sí están permitidos los ciclos de retroalimentación.

Figura 6. **Red neuronal *FeedBack***



Fuente: TutorialsPoint. *Tutorial de red neuronal artificial*. https://www.tutorialspoint.com/artificial_neural_network. Consulta: 25 de febrero de 2020.

2.6. Visión computacional

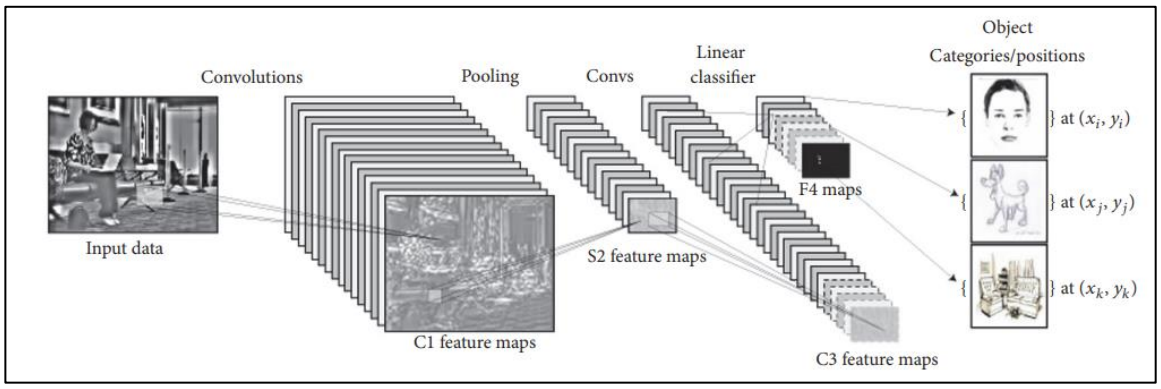
La visión por computación es el subcampo del aprendizaje de máquinas cuyo objetivo es que las computadoras puedan ver y aprender de lo que ven. En su estudio se desarrollan modelos y técnicas por medio de los cuales la computadora pueda entender e interpretar imágenes. Creando varias capas que le permitan al computador aprender y representar datos en múltiples niveles de abstracción que imitan el proceso de aprendizaje del cerebro humano y la forma en que este recibe y visualiza información.

Visión artificial consiste básicamente en la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, a partir de una o varias imágenes bidimensionales de ese mundo.

En los últimos años los algoritmos de visión por computadora han ganado bastante auge gracias a dos factores: el primero es el creciente desarrollo de tecnologías de hardware como las GPU que han permitido el procesamiento en paralelo de las grandes cargas de datos que se necesitan en el entrenamiento de las redes de visión por computación.

El segundo factor es la creciente cantidad de datos etiquetados para entrenar dichas redes, estos bancos de datos junto con el primer factor antes mencionado han permitido que el entrenamiento de las redes de visión por computación sea más rápido y efectivo que en el pasado.

Figura 7. **Red convolucional**



Fuente: Hindawi. *Imagen paper: deep learning for computer vision: a brief review*. https://www.hindawi.com/journals/cin/2018/7068349/. Consulta: 25 de febrero de 2020.

- Aplicaciones de visión por computación

    - Reconocimiento de objetos
    - Reconocimiento de rostros
    - Reconocimiento de acciones
    - *Datasets*

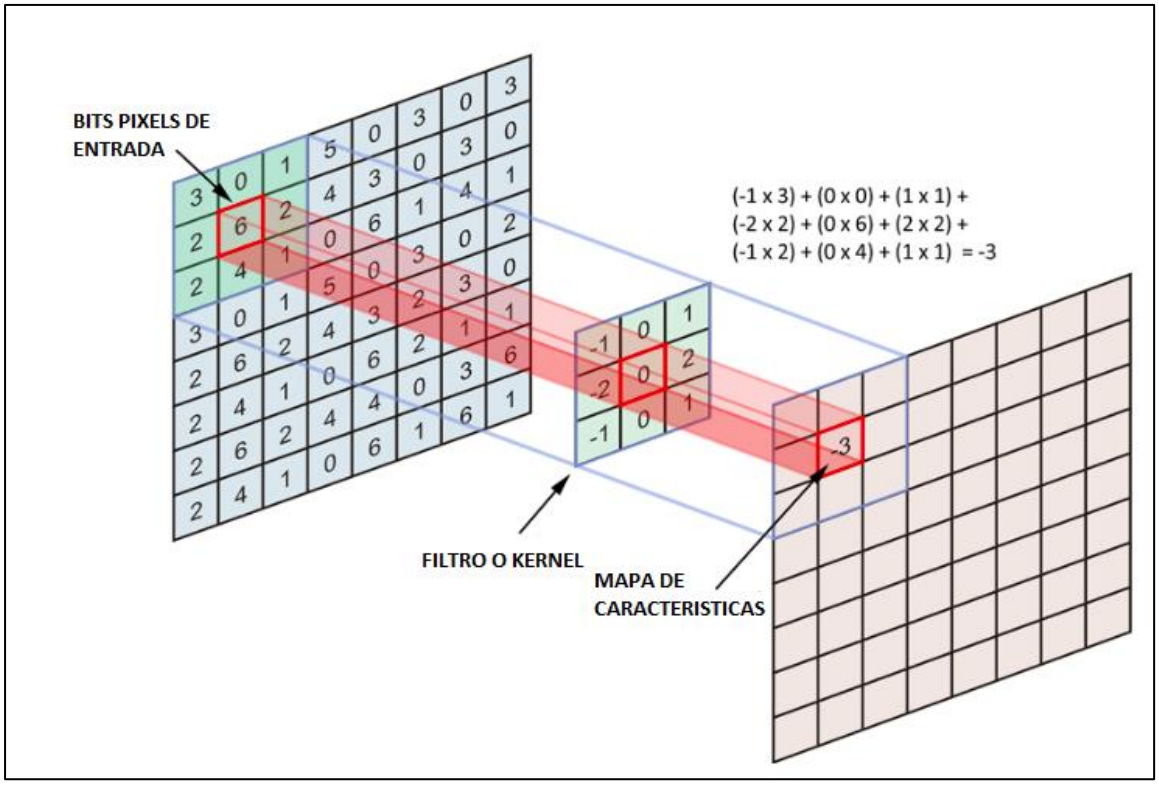
2.6.1. Redes convolucionales

Antes de adentrar en la explicación de las redes Convolucionales es importante mencionar y definir el concepto de convolución. Una convolución, en el ámbito de procesamiento de imágenes, es la operación en la cual dadas dos funciones se obtendrá una tercera función; esta función es un valor ponderado de las dos funciones anteriores.

Las redes convolucionales están inspiradas en la estructura del sistema visual humano y en la arquitectura de una Red convolucional debe mínimo cumplir con las siguientes tres capas:

- Capa de Convolución (Convolutional Layer): en esta capa se aplica el concepto antes mencionado de convolución a las entradas por medio de varios filtros o *kernels* para producir los mapas de características de los bits de la imagen en procesamiento. En la siguiente figura se muestra el proceso de la capa de convolución.

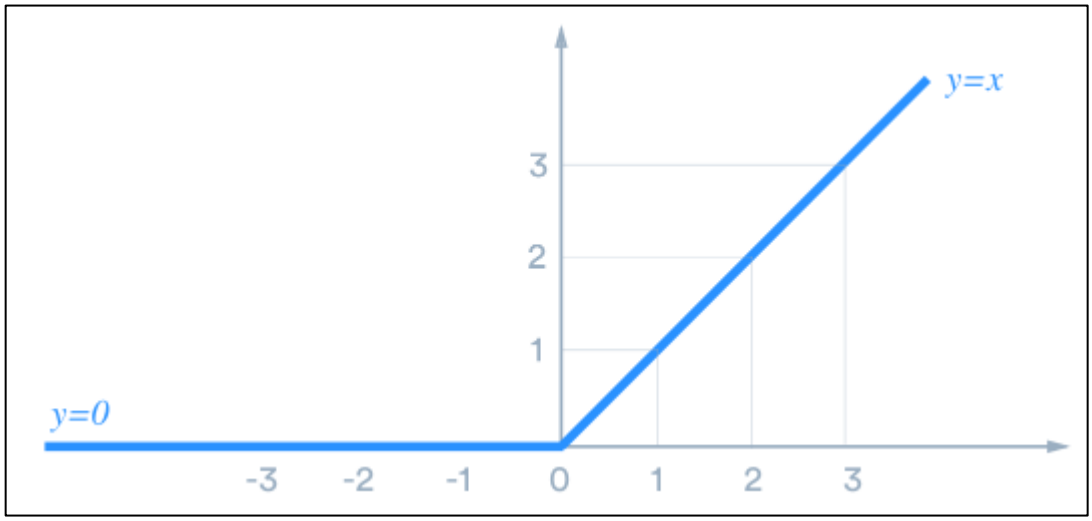
Figura 8. **Funcionamiento capa de convolución**



Fuente: Coursera. *Aprendizaje automático*. https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/. Consulta: 25 de febrero de 2020.

Luego del proceso de convolución, antes de pasar a la siguiente fase de la red los mapas de características pasaran por otro proceso llamado “Activación” en la cual se utiliza la función de activación ReLU. ReLU es la función de activación más comúnmente utilizada en los algoritmos de visión por computación y los algoritmos que conllevan un proceso de convolución. Al final los valores en los mapas de características serán los valores de la convolución que cada uno pasará por la función ReLU como estándar de la arquitectura de las redes convolucionales. Matemáticamente está definida como y = max (0, x).

Figura 9. **Función ReLu**

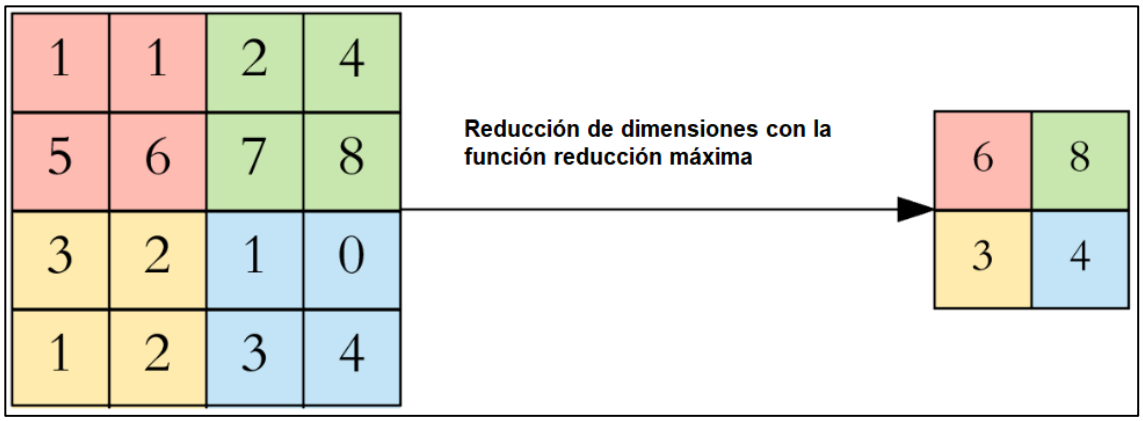


Fuente: Función de activación ReLU. *Aprendizaje automático*. https://www.tinymind.com/learn/terms/relu. Consulta: 25 de febrero de 2020.

- Capa de Agrupación (Pooling Layer): esta es la segunda capa oculta de la cual se compone toda Red Convolucional, en ella se lleva a cabo el proceso de agrupación este es un proceso de disminución de las dimensiones de los mapas de características obtenidos en la capa anterior. En la figura siguiente se observa el proceso de aplicar la función

de valor máximo, en este ejemplo se divide la matriz en 4 partes con la misma cantidad de bits y el algoritmo elegirá el valor máximo de los cuatro valores, se observa en la parte rosa de la matriz que el valor máximo es 6, es el valor seleccionado; luego, se recorre la matriz hasta terminar de obtener una matriz inicial reducida en su cuarta parte. En el proceso pierde algunos bits de información, pero con esta reducción se facilita y se acorta el tiempo del procesamiento de todo el algoritmo.

Figura 10. **Capa de agrupación**



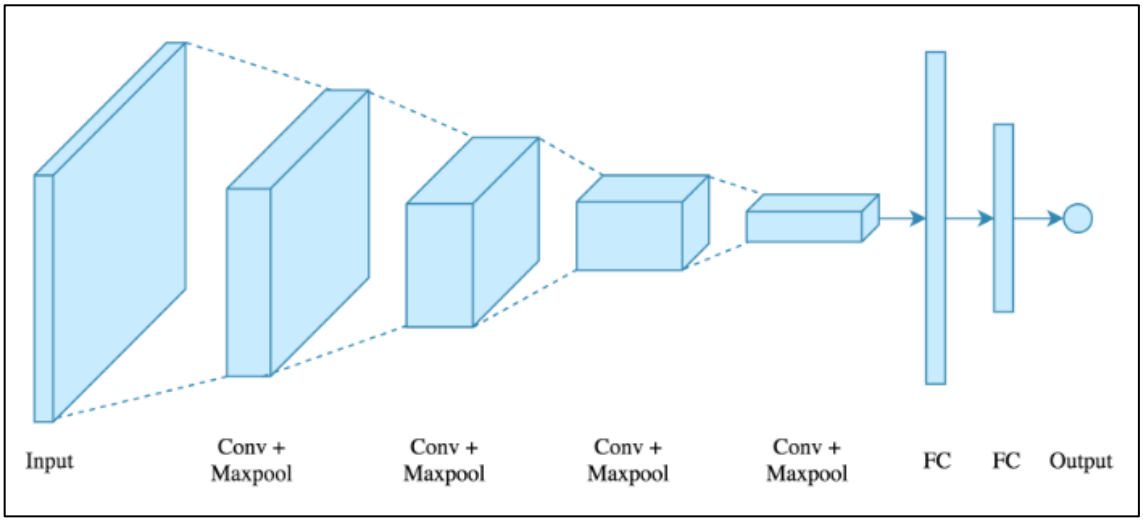
Fuente: Coursera. *Aprendizaje automático*. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#a86a. Consulta: 25 de febrero de 2020.

Clasificación: antes de pasar a la capa final, la matriz reducida pasará a una transformación de 2 dimensiones (en este ejemplo porque pueden ser hasta máximo tres dimensiones cuando la imagen es a color), a una dimensión, esto porque las capas totalmente conectadas solo admiten algoritmo de una dimensión.

- Capas totalmente conectadas (Fully Connected Layers)

En la última capa de toda la red Convolucional es necesario un vector de entrada unidimensional por el cual se puede realizar el razonamiento de alto nivel en toda la red. Este vector termina por convertirse en el vector de características, cada valor de este vector corresponde a un puntaje de clase y cada clase es como una categoría que podrá utilizar el algoritmo para clasificar. Todas las neuronas de esta capa están conectadas a todos los valores de este vector. En la siguiente imagen se puede observar cómo avanza la entrada a través de toda la red y como sus dimensiones van variando hasta llegar a la última capa donde se trabaja con el vector de una dimensión.

Figura 11. **Capa totalmente conectada**



Fuente: Coursera. *Aprendizaje automático*. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#a86a. Consulta: 25 de febrero de 2020.

Cada capa desempeña un papel diferente y muy importante en el proceso de reconocimiento. Cada capa tiene una entrada que convierte en una salida para las respectivas neuronas de la siguiente capa.

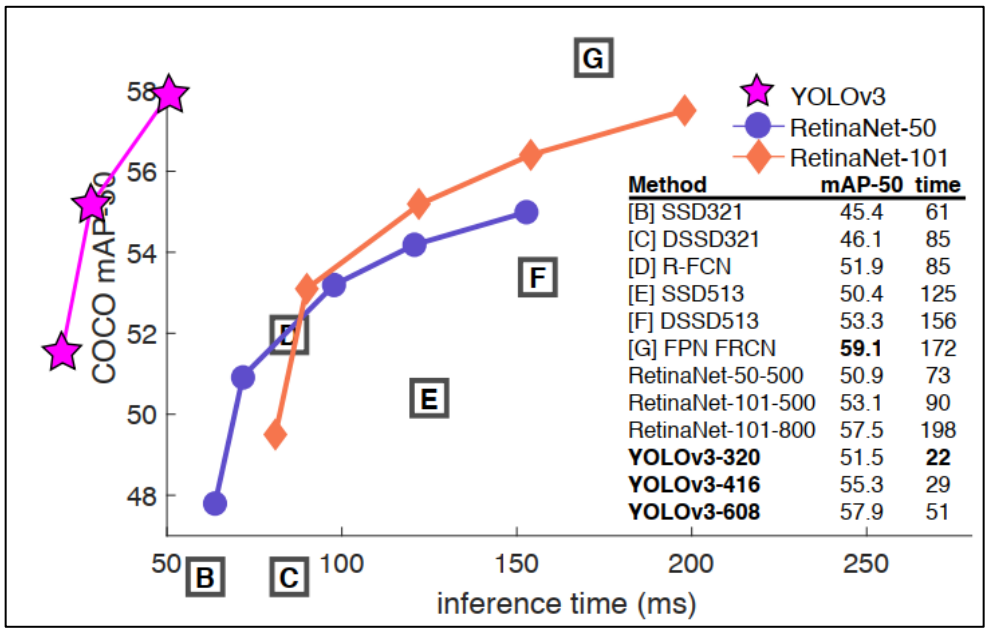
2.6.2. Redes Cnns

Existen actualmente un sin número de algoritmos que utilizan como base las Redes Convolucionales para la visión por computadora, como por ejemplo SSD y YOLO que se caracterizan por ser detectores de captura única a diferencia de RCNN que es un algoritmo de detección por regiones.

En la siguiente gráfica se muestra la comparación en el desempeño de diferentes algoritmos que en la actualidad son los pioneros en la detección de imágenes por computadora. Esta gráfica hace énfasis a la media promedio de precisión de un algoritmo respecto al tiempo de inferencia en el reconocimiento de una figura u objeto en una imagen. Los resultados mostrados por el algoritmo YOLOv3 muestran un 58 % de precisión aproximadamente en un tiempo de 51 milisegundos, un porcentaje que solo se ve superado por el algoritmo FPN FRCN, el cual cuenta con un 59 % de precisión, pero se ve afectado en el tiempo de inferencia siendo 172 milisegundos.

La característica que se resalta en el algoritmo YOLOv3 es que funciona significativamente más rápido que los otros métodos de detección con un rendimiento considerablemente bueno.

Figura 12. **Comparativa Cnns**



Fuente: FARHADI, Ali y REDMON, Joseph. *YOLOv3: An Incremental Improvement*. https://pjreddie.com/media/files/papers/YOLOv3.pdf. Consulta: 4 de abril de 2020.

Debido al desempeño en la media promedio de precisión de cada uno de los algoritmos presentados en la figura 12, es el motivo por el cual se tomó la decisión de utilizar como algoritmo base para el proyecto YOLOv3.

2.6.2.1. YOLO

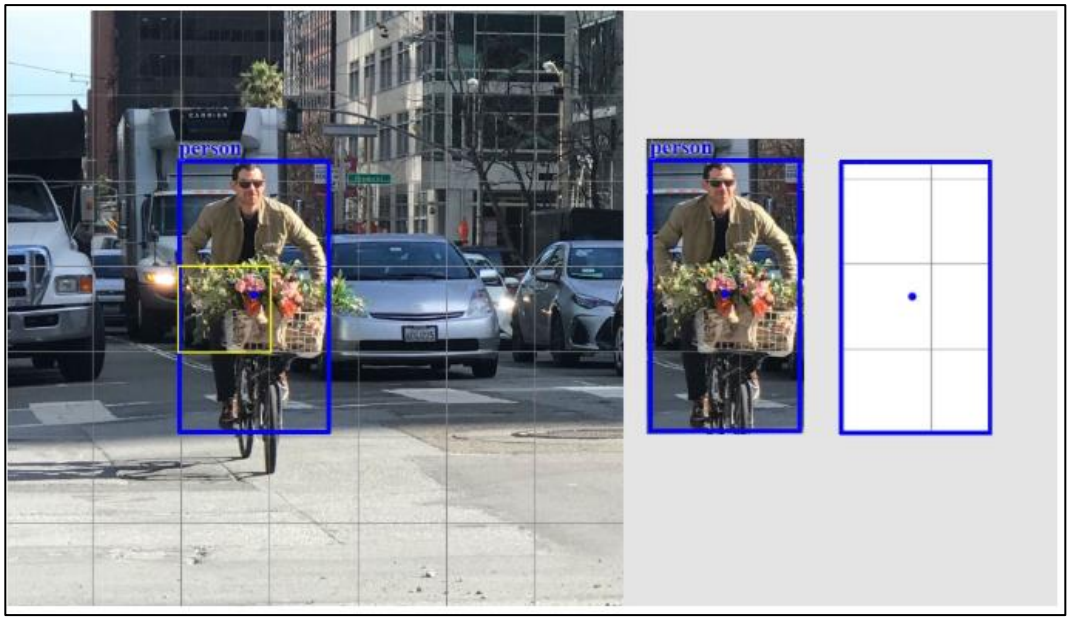
Actualmente el algoritmo más popular y de mejor rendimiento para visión por computación es YOLO (You Only Look Once), como su nombre lo dice traducido al español es como mira una sola vez y es lo que hace interesante a YOLO que únicamente necesita de una imagen para generar la clasificación de los objetos presentes en una imagen o en un video. YOLO no es el algoritmo más preciso en la actualidad, pero si es uno de los más rápidos para el

procesamiento de video, es por ello su popularidad en el campo de conducción autónoma.

- Funcionamiento

La clasificación de YOLO trabaja en dos etapas, en la primera etapa la red divide la imagen en celdas S*S y en cada celda trata de predecir un solo objeto por celda. Y para los posibles objetos detectados crea lo que se llama en YOLO cuadros delimitadores. Esta primera etapa se puede observar en la siguiente imagen, en azul los cuadros delimitadores:

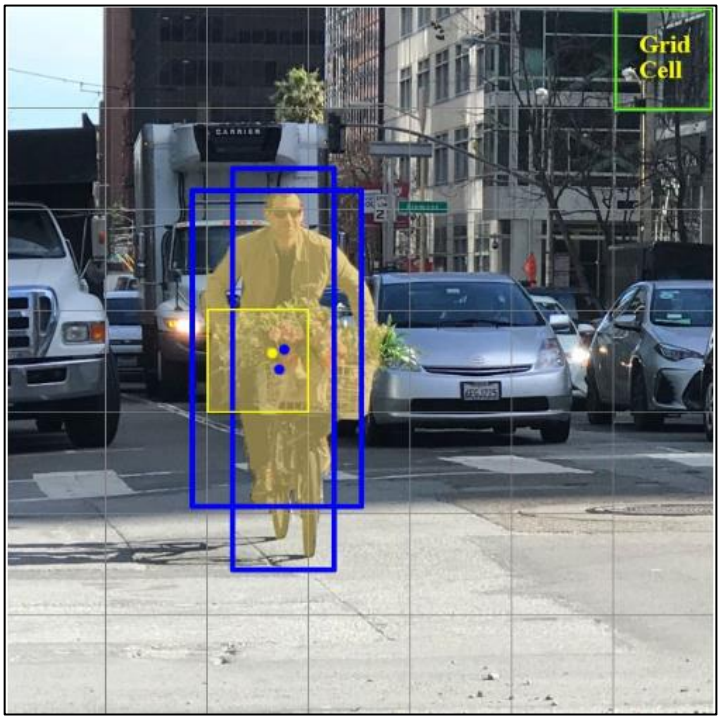
Figura 13. **Funcionamiento YOLO – parte 1**



Fuente: HUI, Jonathan. *Detección de objetos en tiempo real con YOLO, YOLOv2 y ahora YOLOv3*. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088. Consulta: 4 de abril de 2020.

En la segunda etapa el algoritmo realiza una regresión de cada uno de los cuadros delimitadores y la probabilidad que sea una clase identificada por la red.

Figura 14. **Funcionamiento YOLO – parte 2**



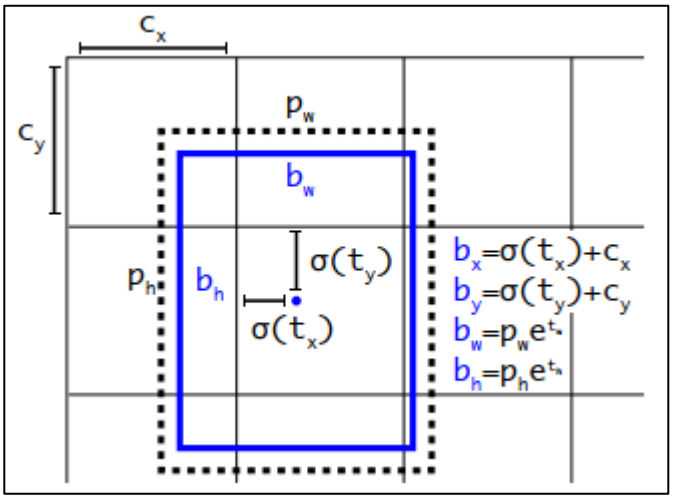
Fuente: HUI, Jonathan. *Detección de objetos en tiempo real con YOLO, YOLOv2 y ahora YOLOv3*. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088. Consulta: 4 de abril de 2020.

- Cuadros delimitadores

Los cuadros delimitadores en YOLO permiten, no solo observar los resultados gráficamente, sino también son una parte importante en el algoritmo y en la clasificación de los objetos dentro de la imagen como dicen sus

creadores: se predice el ancho y la altura del cuadro como compensaciones de los centroides del clúster. Se predicen las coordenadas centrales del cuadro en relación con la ubicación de la aplicación de filtro utilizando una función sigmoidea, ver figura 15.

Figura 15. **Encuadre YOLOv3**



Fuente: elaboración propia, empleando Paper YOLOv3 2018.

- Función de pérdida

El uso de una función es una parte importante en todos los algoritmos de *Deep Learning* y en los algoritmos de Computer Vision no son la excepción, ya que con ella se puede medir el desempeño de dicho algoritmo y qué tan insatisfechos están con las predicciones del modelo. Existen varias funciones de pérdida como, por ejemplo: Error medio cuadrado y Entropía Cruzada. La selección de la función de pérdida depende de la implementación y el nivel de confianza deseado.

En el caso de YOLO la función de pérdida se aplica en tres aspectos diferentes:

- Pérdida de clasificación: si se detecta un objeto, la pérdida de clasificación en cada celda es el error cuadrado de las probabilidades condicionales de la clase.

- Pérdida de localización: mide los errores en las ubicaciones y el tamaño de los cuadros delimitadores.

- Pérdida de confianza: se mide la objetividad de cada cuadro delimitador.

- Supresión no máxima

El proceso de supresión es llevado a cabo en YOLO ya que éste puede realizar predicciones duplicadas en el mismo objeto y con ello se puede eliminar la mayoría de los objetos duplicados, para eliminar los duplicados se elige el cuadro delimitador con el nivel de confianza más bajo. El algoritmo se compone de las siguientes tres etapas:

- En la primera etapa se ordenan las predicciones por los porcentajes de confianza.

- Se recorre todas las cajas delimitadoras y se elimina las que poseen un nivel bajo de confianza.

- Se repite el paso dos hasta que el nivel de confianza supera un valor aceptable.

Ventajas de YOLO

Entre las ventajas de YOLO se pueden encontrar:

- Procesamiento en tiempo real
- Las predicciones se realizan en la misma red
- YOLO ya es un algoritmo estandarizado y general para visión en computación.

3. SISTEMA DE DETECCIÓN

3.1. Aspectos generales sistema de detección

A continuación, se describen los aspectos generales del sistema de detección.

3.1.1. Salón de clases

El proceso de Supresión es llevado a cabo en YOLO ya que este puede realizar predicciones duplicadas en el mismo objeto y con ello se puede eliminar la mayoría de objetos duplicados, para eliminar los duplicados se elige el cuadro delimitador con el nivel de confianza más bajo. El algoritmo se compone de las siguientes tres etapas:

El salón de clases donde se realizaron las configuraciones necesarias y la implementación del sistema de detección por medio de visión computacional fue el salón 132 del edificio 5 de ingeniería mecánica de la National Chiao Tung University. A continuación, se detalla el aula mencionada.

Figura 16. **Salón de clases**

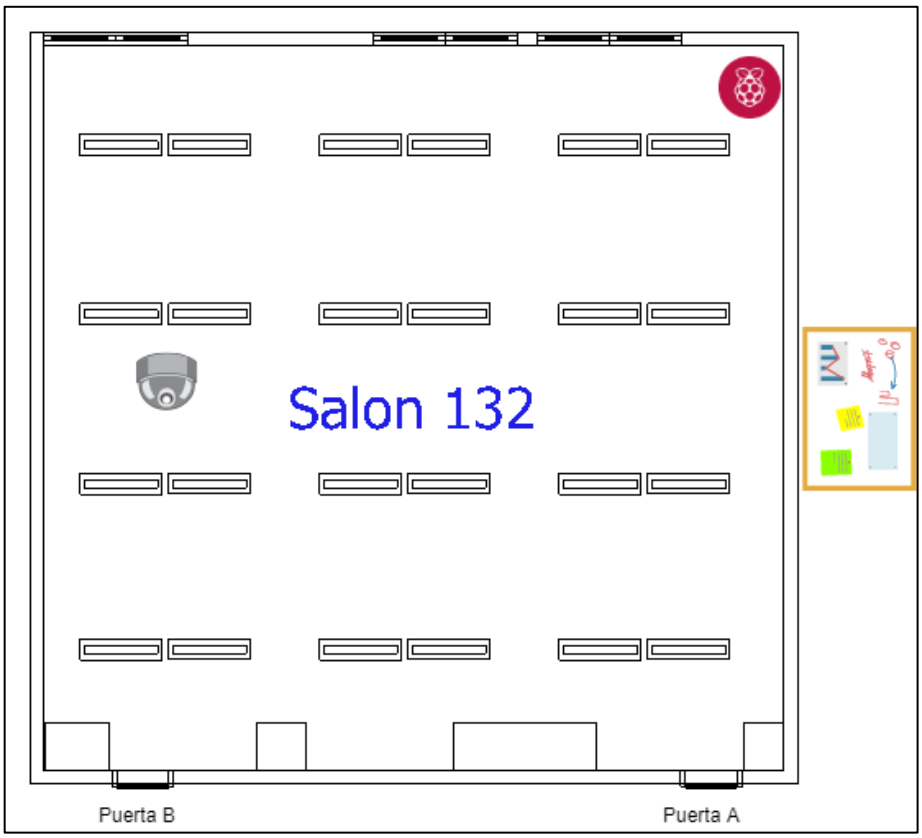
Fuente: elaboración propia.

El aula cuenta con unas dimensiones 30 m por 25 m de largo y capacidad de 90 estudiantes.

- Objetivo del sistema

El objetivo principal del sistema de detección es devolver al algoritmo de control la cantidad de estudiantes presentes en el salón y la ubicación de cada uno dentro de los diferentes escritorios.

Figura 17. **Vista aérea del salón**



Fuente: elaboración propia, empleando Draw.io.

3.1.2. Cámara

La cámara es parte importante del sistema de detección ya que por medio de ella se obtiene el vídeo a procesar en el algoritmo YOLO de visión por computación.

La cámara cuenta con las siguientes características:

Fisheye Wireless Cloud IP Camera - Model DCS-6010L

Marca: D'Link

- 1/3,2” 2 megapixel CMOS progressive sensor
- Fixed fisheye lens: 1,25 mm F2.0
- Built-in microphone and speaker for 2-way communication
- ONVIF compliant
- 802,11n wireless connectivity
- 10/100 Fast Ethernet port
- Micro SD/SDHC Card slot for on board storage

Figura 18. **Cámara**

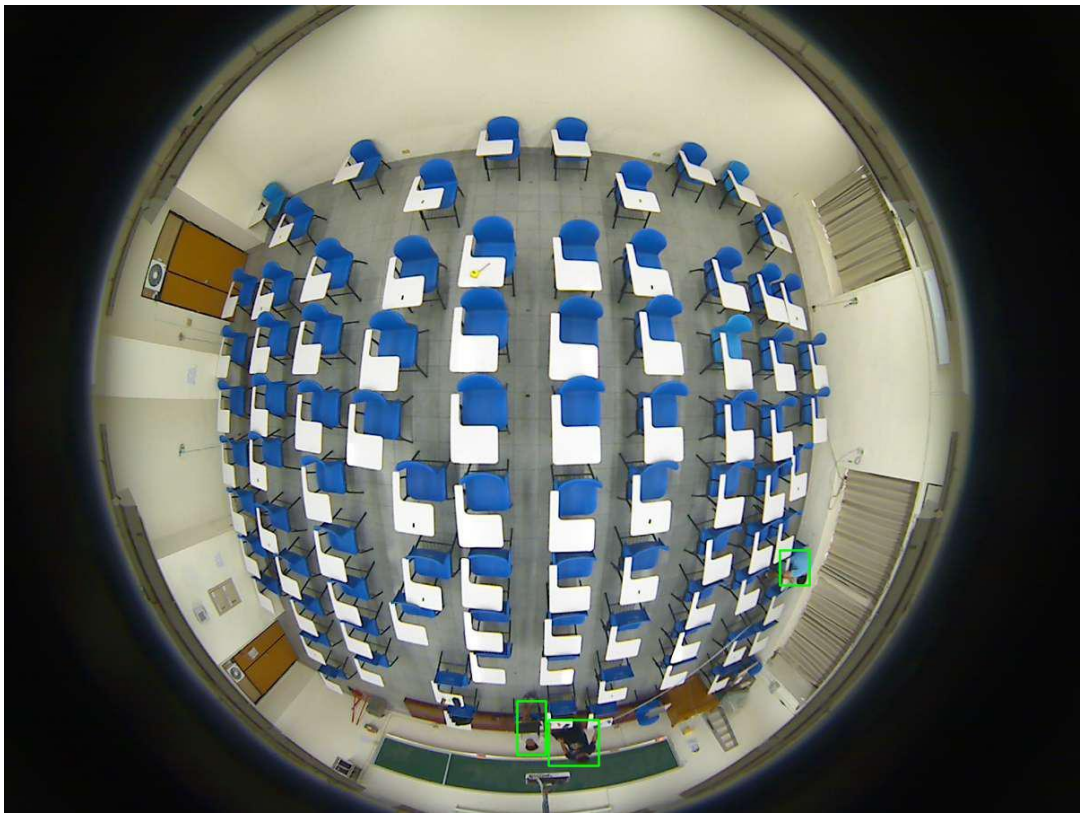
Fuente: D-Link. *Cámara IP inalámbrica en la nube Fisheye – EOL*. https://www.dlink.com.my/product/fisheye-wireless-cloud-ip-camera/. Consulta: 4 de abril de 2020.

El acceso al video de la cámara se realiza a través de la conexión wifi que la cámara permite, más adelante en este mismo capítulo se explica cómo este

video se envía hasta el servidor donde está configurado el sistema de seguimiento.

La cámara por su forma y su función de grabar video 360 grados genera cierta distorsión como se puede observar en la siguiente imagen, esta distorsión es conocida como Fisheye distortion o distorsión de ojo de pez en español.

Figura 19. **Distorsión cámara ojo de pez**



Fuente: elaboración propia, capturada desde la cámara DCS-6010L.

Esta distorsión debe ser corregida antes de realizar el seguimiento correcto de las coordenadas de los estudiantes dentro del salón de clases, es

por ello que se desarrolló el siguiente modelo para la corrección de dicha distorsión y así poder tener un control mucho más preciso de la ubicación de las personas.

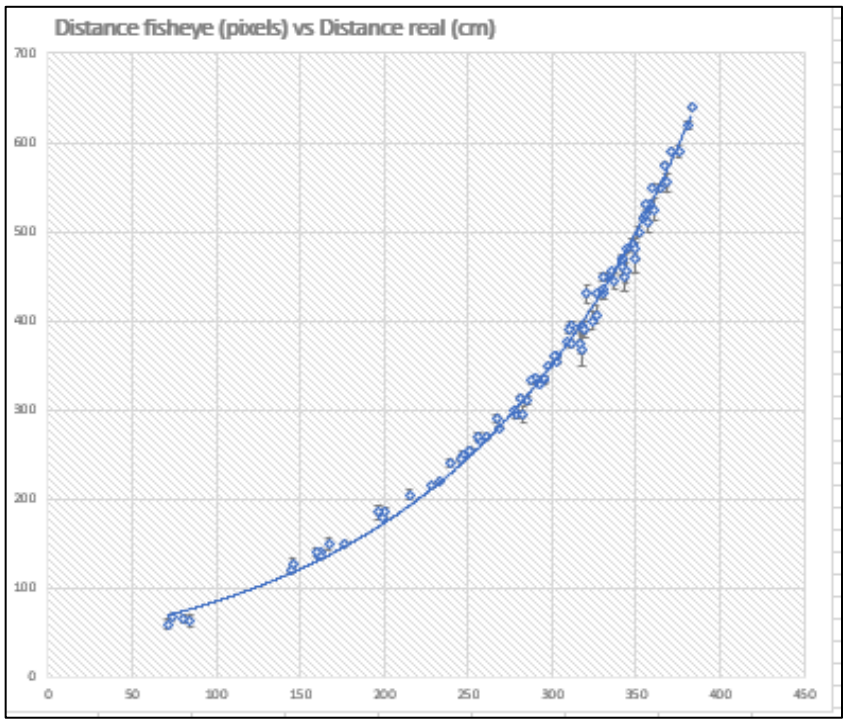
3.1.2.1. Correlación de distorsión de ojo de pez

El modelo matemático elegido fue realizado como parte de la investigación dentro del mismo salón de clases. El modelo seleccionado es un modelo con comportamiento exponencial.

Ecuación: $y = 42{,}165\ e^{0{,}0071*x}$

x: Distancia en fotografía (pixeles)
y: Distancia real en el salón de clases (cm)
Error: 9,88580779 cm

Figura 20. **Gráfica ecuación**



Fuente: elaboración propia, empleando Matlab 9.0 2016.

- Proceso

El proceso por el cual pasan las imágenes después de ser procesadas por el algoritmo es el siguiente.

- Se obtiene del algoritmo una imagen con la posición X y Y en pixeles de la ubicación de la persona.

- Se requiere saber la ubicación en centímetros desde la cámara hasta el escritorio donde dicha persona se ubica.

- Se calcula con la ecuación de distancia la distancia en píxeles de la persona a la cámara.

- Con la fórmula de correlación propuesta se inserta la distancia en píxeles y se obtiene la misma distancia en centímetros dentro del salón.

La siguiente imagen describe las dos distancias y el proceso anteriormente descrito.

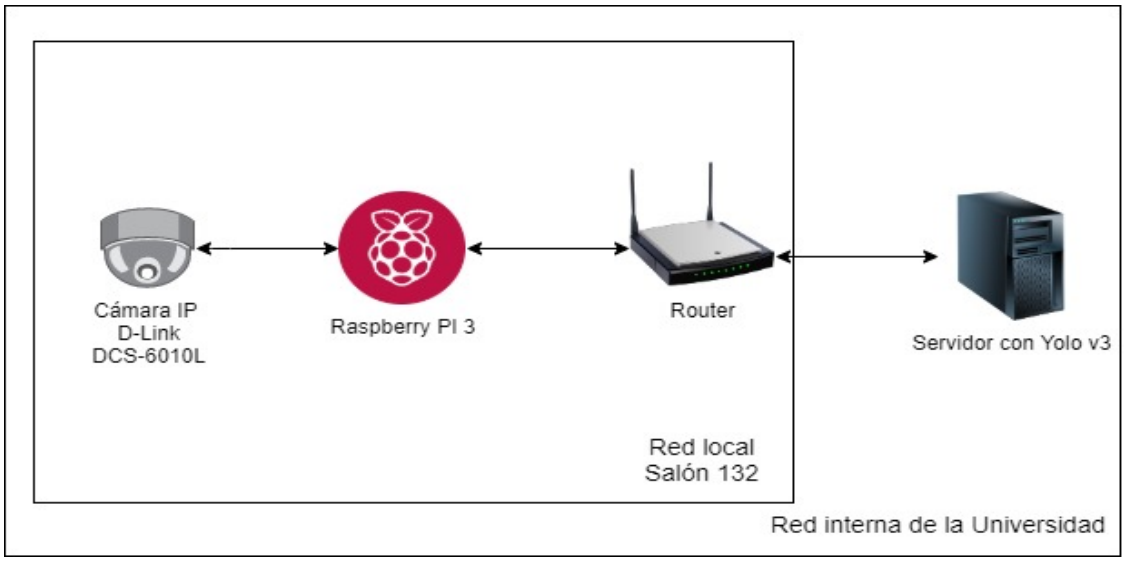
Figura 21. **Funcionamiento ecuación**

Fuente: elaboración propia, capturada desde la cámara DCS-6010L.

3.2. Arquitectura de la solución

La arquitectura de la solución se basa en 4 componentes de software independientes, estos componentes se comunican entre sí para brindar el análisis del video. El primer componente es la cámara ojo de pez, ésta funciona vía red local, esta es la encargada de obtener el video que luego será transmitido a la Raspberry Pi 3, en la cual se encuentra instalada un servidor de transmisión de video (FFServer), y el codificador de video (FFmpeg), encargado de brindar formato de salida al video a transmitir. Se cuenta con un router, que es el encargado de brindar la comunicación entre la cámara Ip y la Raspberry Pi 3, también provee la comunicación a internet. El servidor donde se encuentra instalado el algoritmo de visión por computadora (YOLOv3), se conecta vía internet con el servidor de transmisión de video para obtenerlo y poder realizar el análisis del mismo (ver figura 22).

Figura 22. **Arquitectura de la solución**



Fuente: elaboración propia, empleando herramienta wed Draw.io 2000.

3.3. Herramientas utilizadas para desarrollar la solución

Las herramientas utilizadas en la solución se detallan en el siguiente diagrama.

Figura 23. **Herramientas utilizadas**

Fuente: elaboración propia, empleando herramienta wed Draw.io 2000.

3.3.1. Implementación

La implementación de la solución y el algoritmo de visión por computadora fueron realizados por medio del uso de la herramienta para aprendizaje profundo Tensor Flow y el lenguaje de programación Python, más adelante se detallan las características de cada herramienta:

3.3.1.1. Tensor Flow

Es una biblioteca de código abierto para aprendizaje automático desarrollada por Google. La arquitectura flexible de Tensor Flow le permite

implementar el cálculo a una o más CPU o GPU en equipos de escritorio, servidores o dispositivos móviles con una sola API. Tensor Flow trabaja bajo el concepto de tensores para el desarrollo del aprendizaje y reconocimiento de patrones. La versión utilizada de Tensor Flow es la versión 2.0.

3.3.1.2. Python

> Python es un lenguaje de programación interpretado, interactivo y orientado a objetos. Incorpora módulos, excepciones, tipificación dinámica, tipos de datos dinámicos de muy alto nivel y clases. Python combina una potencia notable con una sintaxis muy clara. Tiene interfaces para muchas llamadas de sistema y bibliotecas, así como para varios sistemas de ventanas, y es extensible en C o C ++. También se puede usar como un lenguaje de extensión para aplicaciones que necesitan una interfaz programable. Finalmente, Python es transportable, es decir, se ejecuta en muchas variantes de Unix, en Mac y en Windows 2000 y versiones posteriores.[2]

Los programas de Python tienen la extensión .py y se pueden ejecutar desde la línea de comandos escribiendo Python file_name.py. La versión utilizada en la implementación es la versión 3.0.

3.3.1.3. Transmisión

Para realizar la transmisión del video y el respectivo procesamiento con YOLO se hizo uso de un Raspberry Pi con el software de FFmpeg, el uso y la configuración de estas herramientas se detalla más adelante, en esta sección solo se da a conocer la versión y las características principales de la herramienta utilizada.

[2] Python. *Preguntas frecuentes generals sobre Python*. https://docs.python.org/3/faq/general.html#what-is-python.

3.3.1.4. Raspberry Pi

> La Raspberry Pi es una computadora de bajo costo del tamaño de una tarjeta de crédito que se conecta a un monitor de computadora o TV, y utiliza un teclado y mouse estándar. Es un pequeño dispositivo que permite a las personas de todas las edades explorar la informática y aprender a programar en lenguajes como Scratch y Python. Es capaz de hacer todo lo que esperaría que hiciera una computadora de escritorio, desde navegar por Internet y reproducir videos de alta definición hasta crear hojas de cálculo, procesamiento de textos y juegos.[3]

La versión Raspberry Pi utilizada es la versión 3.0.

3.3.1.5. FFmpeg

> Es el marco multimedia líder, capaz de decodificar, codificar, transcodificar, mux, demux, transmitir, filtrar y reproducir prácticamente cualquier cosa que los humanos y las máquinas hayan creado. Admite los formatos antiguos más oscuros hasta la vanguardia. No importa si fueron diseñados por algún comité de normas, la comunidad o una corporación. También es muy portátil: FFmpeg compila, ejecuta y pasa nuestra infraestructura de prueba FATE a través de Linux, Mac OS X, Microsoft Windows, BSD, Solaris, etc. bajo una amplia variedad de entornos de construcción, arquitecturas de máquinas y configuraciones.[4]

La versión utilizada es la versión 3.4.

3.4. Proceso de transmisión

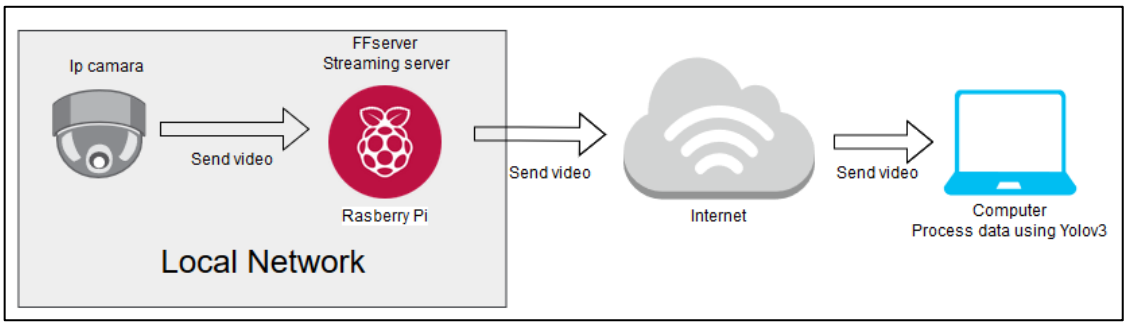
Es uno de los procesos más importantes implementados en la solución del salón inteligente, ya que por medio de él se puede llevar el video para ser analizado fuera del salón de clases y analizarlo en cualquier computador con GPU y Yolo para el análisis correspondiente de las imágenes.

La solución para dicha transmisión se detalla en el siguiente diagrama.

---

[3] Raspberry Pi Programming. *Documentación*. https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/.
[4] FFmpeg. *Documentación oficial*. https://www.ffmpeg.org/about.html.

Figura 24. **Solución**



Fuente: elaboración propia, empleando herramienta wed Draw.io 2000.

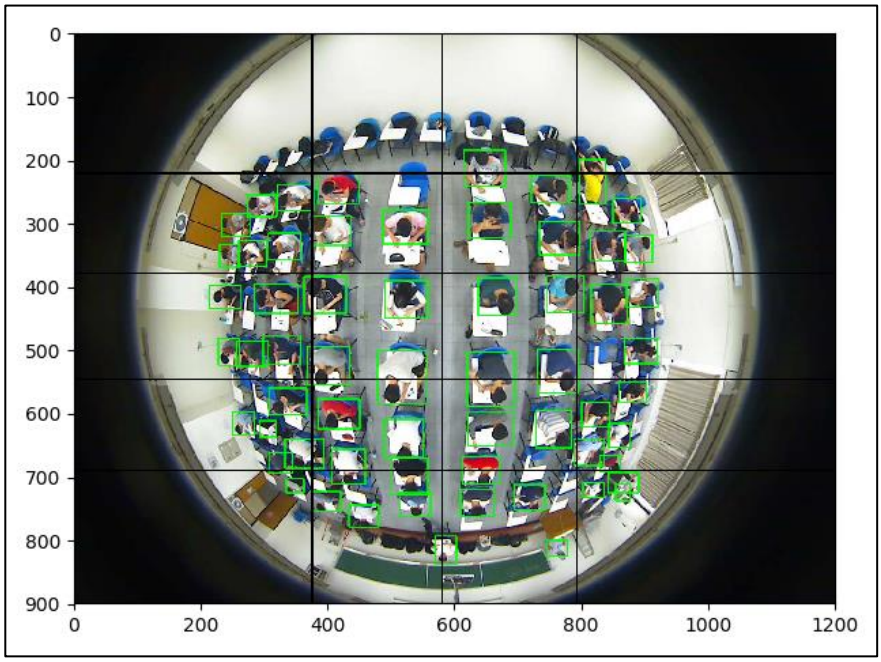
Como se observa en la figura anterior, la cámara es el inicio del proceso de transmisión, ya que por medio de ella se obtiene las imágenes en un formato mpeg, este video es obtenido desde la Raspberry Pi y luego enviado con la ayuda de FFserver al puerto 8090 para que esta data pueda ser obtenida desde la red interna de la universidad desde otro servidor con Yolo configurado. Es importante mencionar que el nombre del video está configurado en un formato en el cual solo usuarios calificados previamente, pueden acceder al video, de esta forma se garantiza la seguridad de los estudiantes en el salón de clases.

3.5. Procesamiento YOLOv3

YOLO es el algoritmo de visión por computación seleccionado para la detección de los estudiantes y su distribución en el salón de clases. YOLO accede al puerto 8090 donde se encuentra el video del proceso anterior y lo procesa para obtener la ubicación de los estudiantes en el salón de clases, luego calcula cada una de las posiciones de los estudiantes dentro de la imagen en píxeles, esta distancia es convertida a distancia real en cm por medio del modelo anteriormente presentado (ver sección 3.1.2.1.). A Continuación, se

muestra una imagen como resultado de la imagen procesada por YOLO y la matriz x, y, z, obtenida de la ubicación de los estudiantes y de la cantidad de personas en el salón de clases.

Figura 25. **Procesamiento YOLOv3**



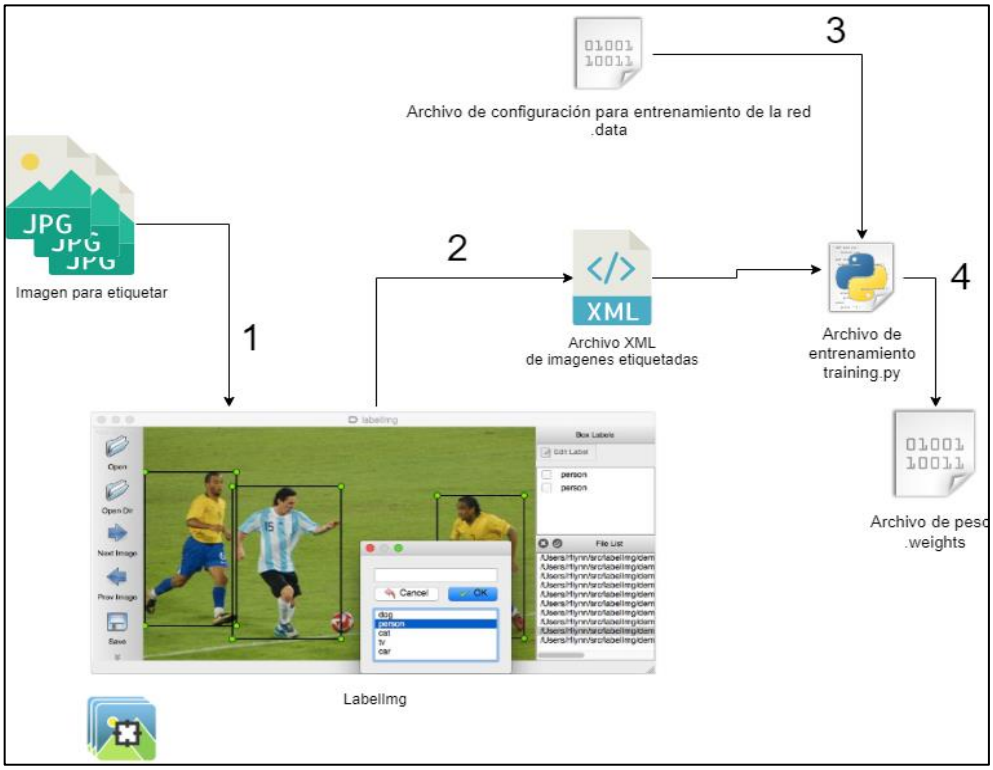
Fuente: elaboración propia, empleando YOLOv3 2018.

3.5.1. Entrenamiento del modelo

Para obtener mejores resultados con el algoritmo Yolo en el sistema de detección, durante la estancia en Taiwán se realizaron varias fases de entrenamiento para incrementar la precisión de la detección de las personas en el salón de clase. En un inicio se disponía de un algoritmo con la posibilidad de detectar personas, pero la precisión de éste fallaba en algunos casos en los que los estudiantes se encuentran de espaldas a la cámara, es por esta situación

que se llevó a cabo el proceso de entrenamiento que se explica gráficamente en la siguiente imagen:

Figura 26. **Proceso de entrenamiento YOLOv3**



Fuente: elaboración propia, empleando herramienta wed Draw.io 2000.

3.5.1.1. Proceso

En la figura 26, se muestra gráficamente el proceso de entrenamiento YOLOv3, a continuación, se detalla la forma en que interactúan los distintos componentes del proceso antes mencionado:

- Primero tomamos un lote de imágenes que serán cargadas a la aplicación LabelIng, dichas imágenes son las imágenes que servirán para entrenar y agregar precisión a nuestro algoritmo YOLO.

- Luego de pasar imagen por imagen y clasificarlas según el tipo o clase de objetos a identificar (en este caso es una sola clase, ya que solo se desea detectar la clase persona) se obtiene el archivo XML con el nombre de las imágenes y la posición de cada una de las etiquetas creadas.

- En el paso tres el archivo training.py es de vital importancia ya que, junto con los archivos de entrada, el XML del paso dos y el nuevo archivo .data que contiene la información y los parámetros relevantes para realizar el entrenamiento, teniendo estos dos archivos se corre el archivo training.py que tarda varias horas en terminar la ejecución, este tiempo depende de los parámetros configurados en el archivo *Data*.

- Después de varias horas de entrenamiento, se obtiene como resultado el archivo weights, este archivo es sustituido en una carpeta específica del algoritmo Yolo y ya se cuenta con una red más precisa y acorde a los objetos y las formas que se desea detectar (en el siguiente capítulo se muestran los resultados y el avance según el entrenamiento de la red).

4. RESULTADOS EXPERIMENTALES

4.1. Resultados algoritmo de detección

A continuación, se describen los resultados de algoritmo de detección.

4.1.1. Durante un período de tiempo sin variación de estudiantes

Durante una clase los estudiantes no varían en un salón de clases, en esta primera prueba se buscaba probar la eficiencia del algoritmo de YOLO durante una clase, los resultados se detallan en la siguiente tabla:

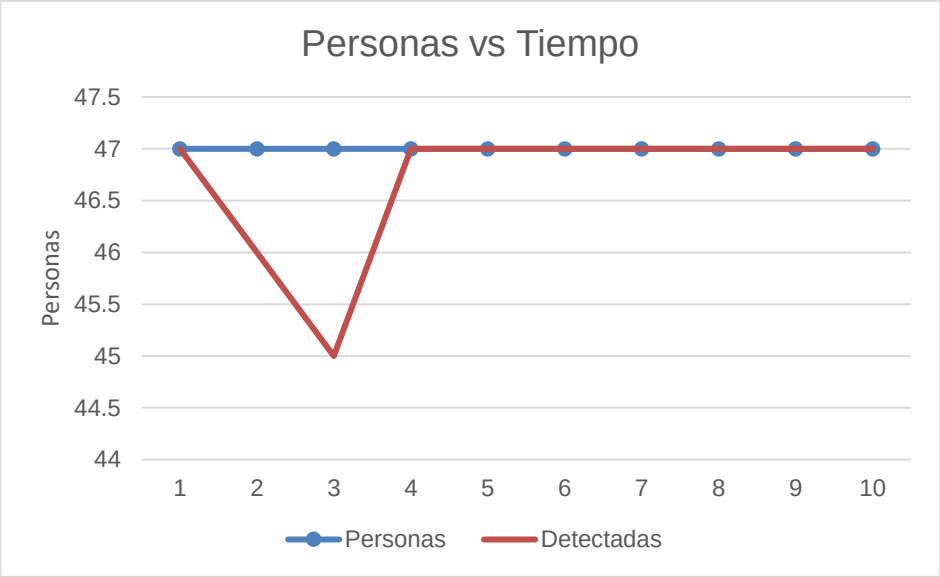
Tabla I. **Casos sin variación de estudiantes**

Caso	Personas en el salón	Personas detectadas	Certeza %	Error %
1	47	47	100,00	0,00
2	47	46	97,87	2,13
3	47	45	95,74	4,26
4	47	47	100,00	0,00
5	47	47	100,00	0,00
6	47	47	100,00	0,00
7	47	47	100,00	0,00
8	47	47	100,00	0,00
9	47	47	100,00	0,00
10	47	47	100,00	0,00

Fuente: elaboración propia.

Durante el período de tiempo en que se tomaron dichos casos únicamente se observa una variación en el caso 2 y en el caso 3, respecto al número real de estudiantes en el salón, luego podemos ver que la efectividad del algoritmo es bastante buena llegando a tener un 99 % de efectividad.

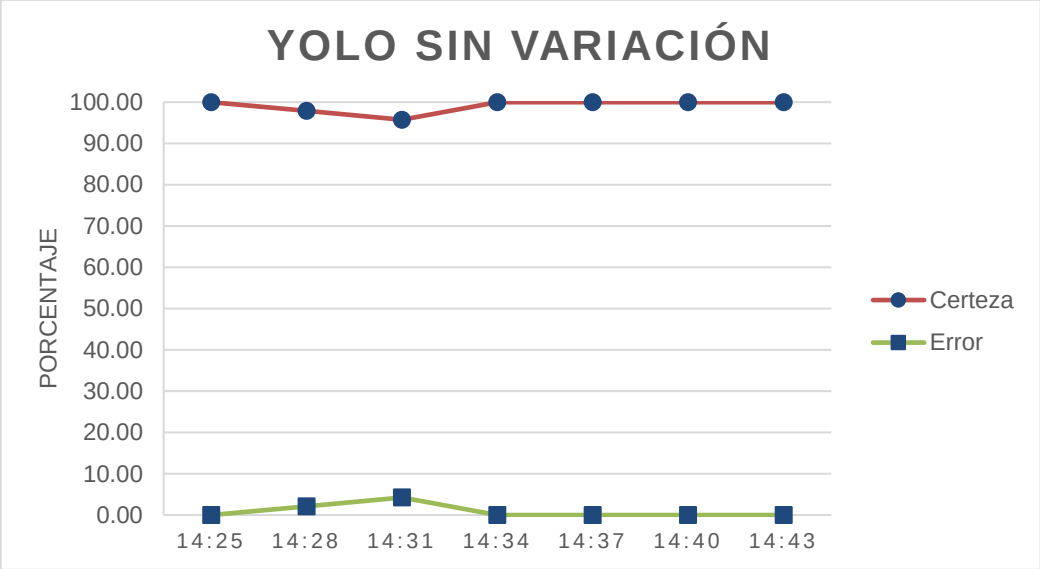
Figura 27. **Gráfica sin variación de personas**



Fuente: elaboración propia.

La figura 27 muestra la gráfica sin variación de personas, como se mencionó anteriormente, únicamente varía en los casos número 2 y 3, sin embargo, el algoritmo puede recuperarse y obtener excelentes resultados.

Figura 28. **Certeza del algoritmo sin variación de personas**



Fuente: elaboración propia.

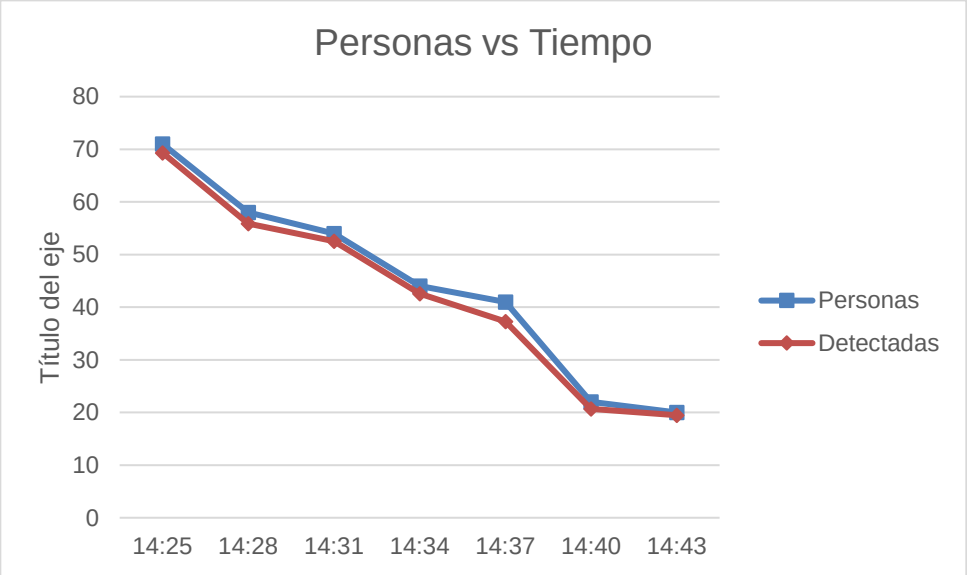
La figura 28 muestra la certeza del algoritmo sin variación de personas, únicamente los casos 2 y 3 no brindan el 100 % de certeza, se debe considerar que para que un modelo sea tomado como válido este debe estar por arriba del 90 % de certeza, pero en ocasiones este porcentaje depende de las especificaciones del usuario, en el caso solicitado en la National Chiao Tung University el 90 % era válido.

4.1.2. Durante un período de tiempo con variación de estudiantes

Las siguientes pruebas fueron efectuadas durante un período aproximadamente de 25 minutos, en este período de tiempo se efectuaba un examen en el salón 132 es por ello que la cantidad de personas variaba en ese

período de tiempo. En el siguiente gráfico se puede observar la variación de estudiantes respecto al tiempo.

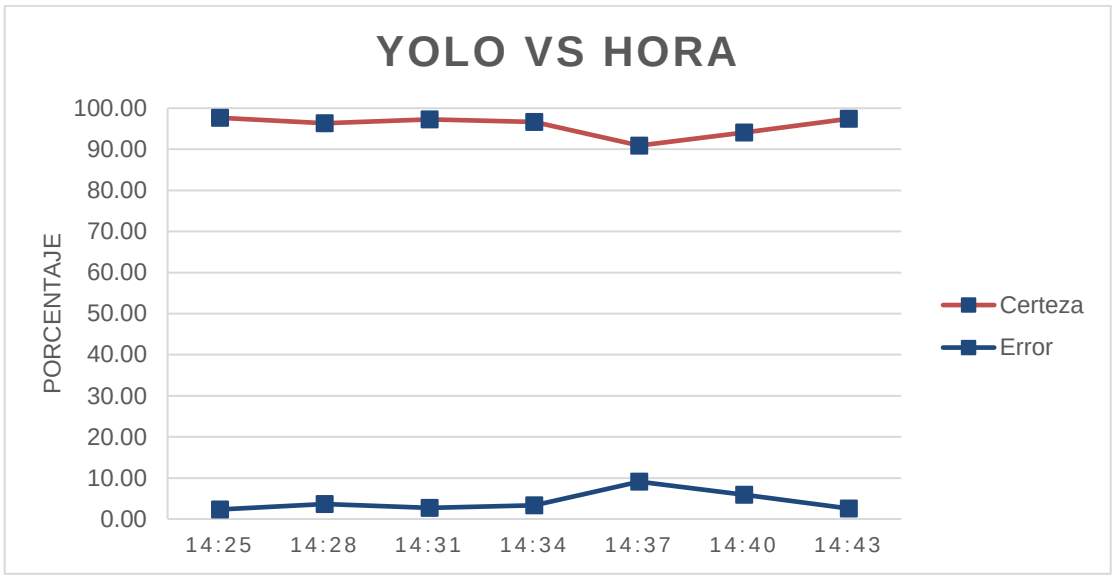
Figura 29. **Detección con variación de personas**



Fuente: elaboración propia.

En azul se puede observar el número real de personas dentro del salón y en anaranjado el número de personas detectadas con el algoritmo, a continuación, se muestra la certeza del algoritmo y cómo respondió a los cambios de personas en un corto período de tiempo.

Figura 30. **Certeza con variación de personas**



Fuente: elaboración propia.

4.1.3. Entrenamiento del algoritmo de YOLOv3

Como se mencionó en el capítulo anterior, el algoritmo de detección fue entrenado por lo menos una vez cada mes para obtener mejores resultados en la detección de los alumnos en el salón de clases

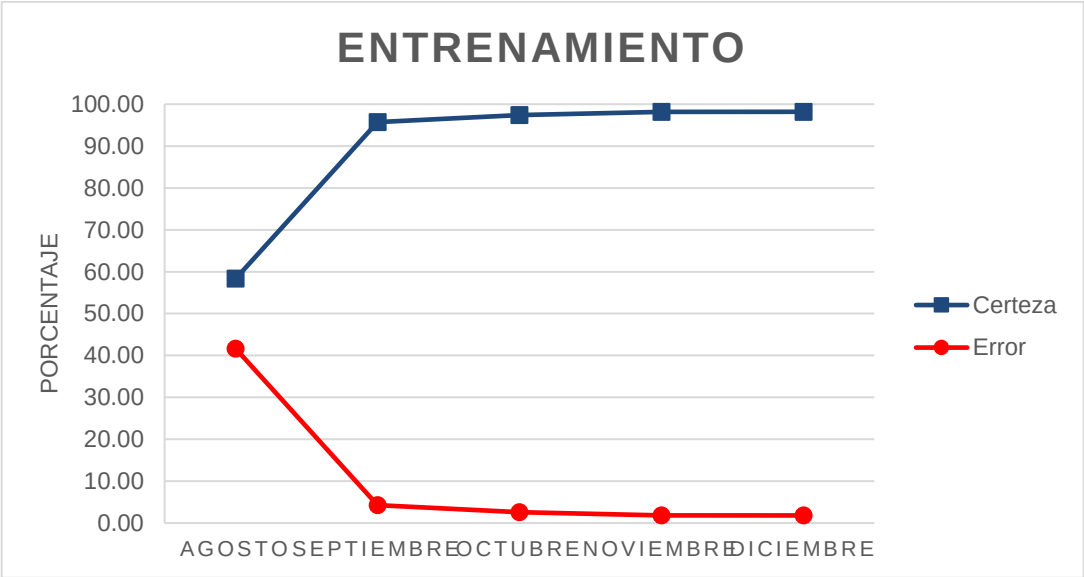
Tabla II. **Entrenamiento del algoritmo YOLOv3**

Caso	Personas detectadas	Personas en el salón	Certeza %	Error %	Meses Año 2019
1	60	35	58,33	41,67	Agosto
2	47	45	95,74	4,26	Septiembre
4	38	37	97,37	2,63	Octubre
5	54	53	98,15	1,85	Noviembre
6	55	54	98,18	1,82	Diciembre

Fuente: elaboración propia.

La figura 31 muestra un gráfico que permite observar cómo a través del tiempo, en los cinco meses de la estancia de investigación en Taiwán, el crecimiento de agosto a septiembre es el más notorio de todo el gráfico debido a que es el primer entreno que se realiza al algoritmo YOLO, de allí en adelante es poco lo que se puede mejorar el algoritmo a menos que se disponga de grandes lotes de imágenes etiquetadas. Una red de visión por computación es considerada válida siempre que es por arriba del noventa por ciento de efectividad o certeza, esta medida es alcanzada desde el mes de septiembre, luego solo se logra mejorar un poco más.

Figura 31. **Entrenamiento algoritmo**



Fuente: elaboración propia.

4.1.4. Comparativa tiempos de YOLOv3 GPU - CPU

Existen actualmente dos métodos de procesamiento el más conocido es llamado CPU es el clásico método de procesamiento Centralizado, nuevas tecnologías como GPU permiten realizar varios procesos con mejoras en el tratamiento de procesos con gráficos, tarjetas especializadas para el *Machine Learning* y la inteligencia artificial.

Durante las pruebas de resultados se realizó una comparativa entre estas dos tecnologías, teniendo la nueva tecnología de GPU notables resultados en el momento de realizar el procesamiento de imágenes del algoritmo de YOLOv3. En la siguiente tabla se puede observar no solo la efectividad del algoritmo, sino también los tiempos de procesamiento.

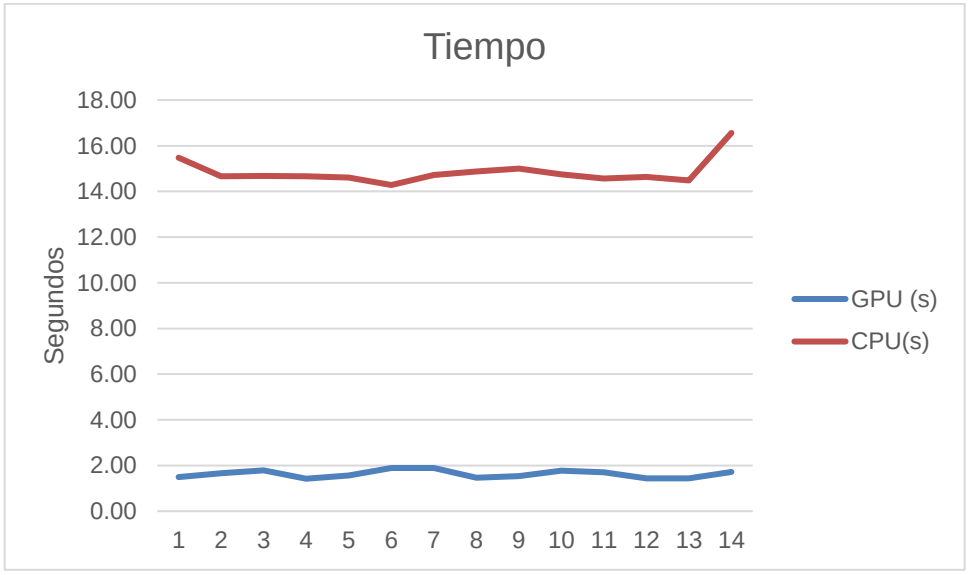
Tabla III. **Tiempo de Yolov3 GPU-CPU**

	GPU				CPU			
Caso	Detectadas	Certeza	Error	GPU (s)	Detectadas	Certeza	Error	CPU(s)
1	47	100,00	0,00	1,49	47	100,00	0,00	15,47
2	46	97,87	2,13	1,66	46	97,87	2,13	14,66
3	45	95,74	4,26	1,79	46	97,87	2,13	14,68
4	47	100,00	0,00	1,42	47	100,00	0,00	14,66
5	47	100,00	0,00	1,56	47	100,00	0,00	14,60
6	47	100,00	0,00	1,89	47	100,00	0,00	14,28
7	47	100,00	0,00	1,89	47	100,00	0,00	14,72
8	47	100,00	0,00	1,46	47	100,00	0,00	14,87
9	47	100,00	0,00	1,54	46	97,87	2,13	15,00
10	47	100,00	0,00	1,77	45	95,74	4,26	14,75
11	47	100,00	0,00	1,71	47	100,00	0,00	14,57
12	47	100,00	0,00	1,44	47	100,00	0,00	14,63
13	47	100,00	0,00	1,43	47	100,00	0,00	14,48
14	47	100,00	0,00	1,72	47	100,00	0,00	16,56
	Resultado	99,54	0,46	1,63	Resultado	99,24	0,76	14,85

Fuente: elaboración propia.

La diferencia de tiempo de procesamiento se puede observar en la siguiente gráfica:

Figura 32. **Comparativa entre GPU Y CPU**



Fuente: elaboración propia.

4.2. Resultados algoritmo de detección

A continuación, de describen los resultados algoritmos de detección.

4.2.1. Modelo de correlación de distorsión ojo de pez

El modelo de correlación ojo de pez busca encontrar la distancia en centímetros de los estudiantes tomando como centro del salón la posición de la cámara, dado el formato de fotografía realizada por dicha cámara era necesario que el modelo no sólo realizara la transformación de pixeles a centímetros, sino también el cálculo de la distorsión de dicha cámara, esto se detalló en el capítulo anterior.

En la siguiente tabla se muestran las pruebas realizadas al modelo antes de su implementación en el algoritmo final.

Tabla IV. **Correlación ojo de pez**

Caso	PosX	PosY	Distancia (pixel)	Distancia modelo(cm)	Distancia medida(cm)	Error (cm)
1	536	648	217,37	197,33	208	10,67
2	643	650	221,98	203,91	197	6,91
3	711	686	280,41	308,74	313	4,26
4	778	397	197,88	171,84	154	17,84
5	639	161	280,45	308,82	302	6,82
6	536	247	195,00	168,36	178	9,64
7	427	339	184,55	156,32	173	16,68
8	368	667	316,25	398,22	385	13,22
					Error Promedio	10,76

Fuente: elaboración propia.

- Detalle de la tabla
    - Caso: el número de prueba efectuada
    - PosX: la posición x en pixeles de la persona, este dato es obtenido desde el algoritmo de YOLOv3.
    - PosY: la posición y en pixeles de la persona, este dato es obtenido desde el algoritmo de YOLOv3.
    - Distancia (pixel): es el cálculo de la distancia en pixeles.

- Distancia modelo: es el resultado obtenido del modelo en centímetros.
- Distancia medida: es la distancia medida en el salón con herramientas de medición.

- Error: valor absoluto de la diferencia entre la distancia del modelo y la medida.

- Error promedio: es el error promedio total de todos los casos, en este experimento el error promedio obtenido es de 10 cm.

CONCLUSIONES

1. El algoritmo de visión por computadora implementado YoloV3, por sus siglas en inglés You Only Look once, alcanzó en la determinación de la distribución de las personas en el salón de clase, utilizando la cámara de ojo de pez, un 98 % de precisión al hacer uso del modelo entrenado.

2. La correlación matemática determinada para la aproximación de la posición real de las personas, la cual se encarga de evaluar y corregir la distorsión causada por la cámara ojo de pez, sigue una distribución exponencial con diez centímetros de error.

3. Para el entrenamiento del modelo del algoritmo YOLOv3 implementado, se hizo la comparación entre la precisión y el tiempo de procesamiento utilizando CPU y GPU, con lo cual se determinó que la precisión mostrada por ambos es similar, pero GPU presenta un menor tiempo de procesamiento para alcanzar el punto de convergencia.

4. Para la transmisión de video en tiempo real se implementó FFServer, el cual es un servidor de transmisión de streaming de video que trabaja en conjunto con Mpeg para la codificación y brindarle formato de salida a la transmisión de la cámara de ojo de pez.

RECOMENDACIONES

1. Seleccionar de forma correcta las imágenes para el entrenamiento del algoritmo es de vital importancia para obtener mejores resultados en el menor tiempo posible.

2. Conocer las especificaciones de la cámara ojo de pez al momento de realizar la implementación, ya que con ello se reduce el tiempo de cálculo de error en la correlación con las dimensiones reales del salón y la posición real de los estudiantes.

3. Al momento de realizar el entrenamiento del algoritmo, es recomendable realizar una selección del ochenta por ciento de las imágenes que se tienen etiquetadas para utilizarlas como muestra de entrenamiento y el veinte por ciento de las imágenes para realizar las pruebas y verificar el correcto funcionamiento del análisis del algoritmo.

BIBLIOGRAFÍA

1. Artificial Neural Network Tutorial. *Tutoriales Point*. [en línea]. <https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm>. [Consulta: marzo de 2019].

2. Central Intelligence Agency. *The world factbook*. [en línea]. <https://www.cia.gov/library/publications/the-world-factbook/rankorder/2233rank.html>. [Consulta: marzo de 2019].

3. Convolutional Neural Networks. *Stanford University*. [en línea]. <http://cs231n.github.io/convolutional-networks/>. [Consulta: julio de 2019].

4. Coursera. *Aprendizaje automático*. [en línea]. <https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/>. [Consulta: 25 de febrero de 2020].

5. D-Link. *Cámara IP inalámbrica en la nube Fisheye – EOL*. [en línea]. <https://www.dlink.com.my/product/fisheye-wireless-cloud-ip-camera/>. [Consulta: 4 de abril de 2020].

6. Fast R-CNN. *Microsoft Research*. [en línea]. <https://arxiv.org/pdf/1504.08083.pdf>. [Consulta: octubre de 2019].

7. FFmpeg. *Documentación oficial*. [en línea]. <https://www.ffmpeg.org/about.html>. [Consulta: 25 de febrero de 2020].

8. Función ReLU. *Aprendizaje automático*. [en línea]. <https://www.tinymind.com/learn/terms/relu>. [Consulta: octubre de 2019].

9. GÉRON, Aurélien. *Hands-On Machine Learning with Scikit-Learn & TensorFlow*. Estados Unidos: O'RELLY, 2017. 437 p.

10. Grupo de Visión del Comité Español de Automática, CEA. *Conceptos y métodos en visión por computador*. [en línea]. <https://intranet.ceautomatica.es/sites/default/files/upload/8/files/ConceptosyMetodosenVxC.pdf>. [Consulta: octubre de 2019].

11. Hindawi. *Imagen paper: deep learning for computer vision: a brief review*. [en línea]. <https://www.hindawi.com/journals/cin/2018/7068349/>. [Consulta: 25 de febrero de 2020].

12. HUI, Jonathan. *Funcionamiento YoloV3*. [en línea]. <https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088>. [Consulta: octubre de 2019].

13. Mc.ai. *Computer visión.* [en línea]. <https://mc.ai/que-es-computer-vision/>. [Consulta: octubre de 2019].

14. MOOR, James. *The dartmouth college artificial intelligence conference: the next fifty years*. [en línea]. <https://pdfs.semanticscholar.org/d486/9863b5da0fa4ff5707fa972c6e1dc92474f6.pdf>. [Consulta: mayo de 2019].

15. NIELSEN, Michael. *Neural Networks and Deep Learning*. Estados Unidos, 2015. 167-170 p.

16. PARAJES, Gonzalo; SANTOS, Matilde. *Inteligencia artificial e ingeniería del conocimiento.* México: Alfaomega, 2006. 233-237 p.

17. Python. *Preguntas frecuentes generals sobre Python*. [en línea]. <https://docs.python.org/3/faq/general.html#what-is-python>. [Consulta: 25 de febrero de 2020].

18. Raspberry Pi Programming. *Documentación*. [en línea]. <https:// www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>. [Consulta: 25 de febrero de 2020].

19. REDMON, Joseph; FARHADI, Ali. *YOLOv3: An Incremental Improvement*. [en línea]. <https://pjreddie.com/media/files/papers/ YOLOv3.pdf>. [Consulta: octubre de 2019].

20. SHALEV-SHWARTZ, Shai. *Understanding Machine Learning*. Estados Unidos: Cambridge University Press, 2014. 449 p.

21. TutorialsPoint. *Red neuronal artificial - conceptos básicos*. [en línea]. <https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_ network_basic_concepts.htm>. [Consulta: 25 de febrero de 2020].

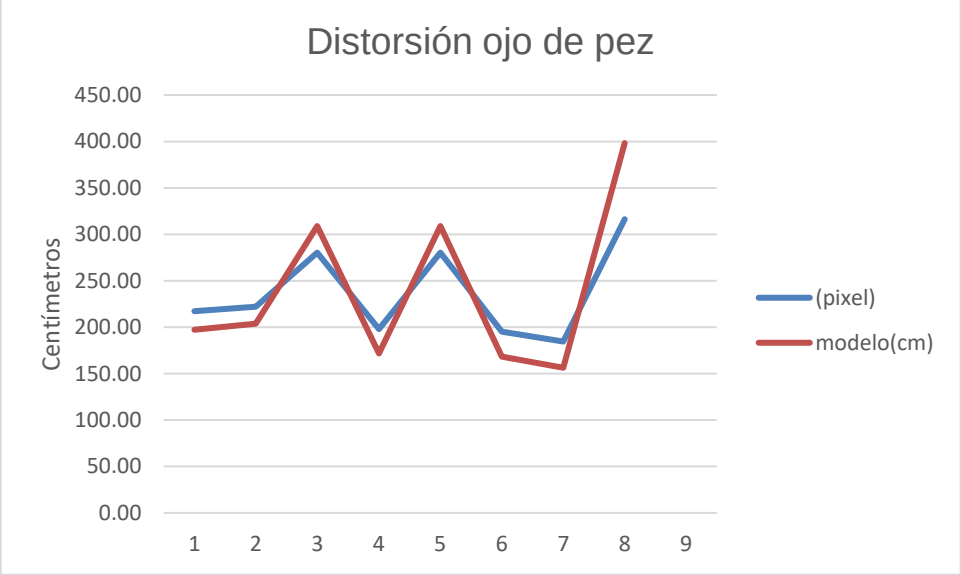
22. _____. *Tutorial de red neuronal artificial*. [en línea]. <https:// www.tutorialspoint.com/artificial_neural_network>. [Consulta: 25 de febrero de 2020].

23. VOULODIMOS, Athanasios. *Deep learning for computer vision: a brief review.* [en línea]. <http://downloads.hindawi.com/journals/cin/2018/7068349.pdf>. [Consulta: junio de 2019].

24. ZEILER, Matthew y FERGUS, Rob. *Visualizing and Understanding Convolutional Networks*. [en línea]. <https://arxiv.org/abs/1311.2901>. [Consulta: junio de 2019].

APÉNDICES

Apéndice 1. **Gráfico comparativa distorsión ojo de pez**

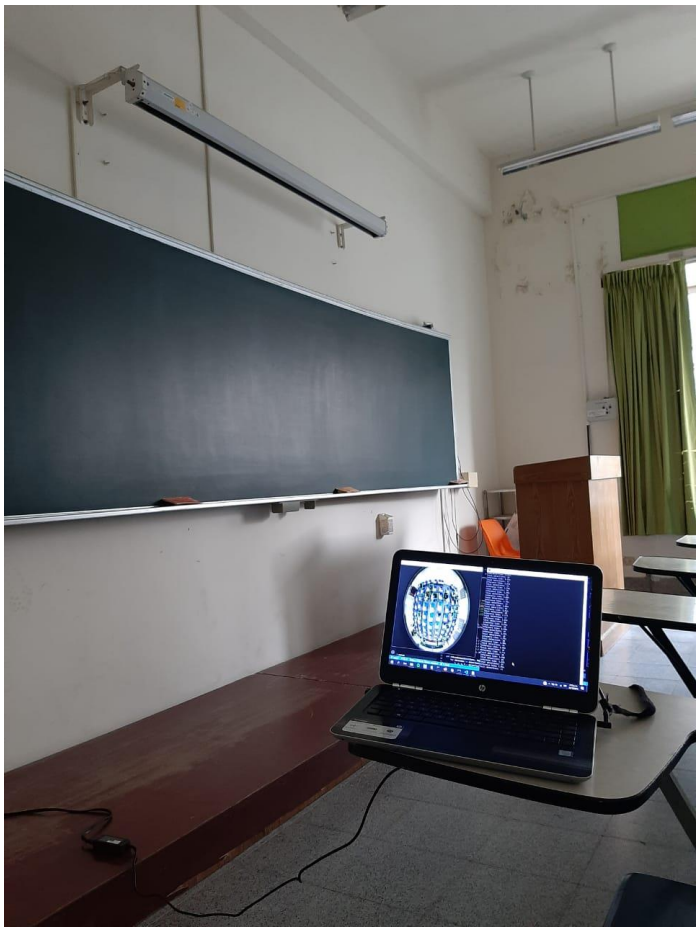


Fuente: elaboración propia.

Apéndice 2. **Presentación del proyecto en Hsinchu Taiwán**

Fuente: elaboración propia.

Apéndice 3. **Pruebas en el salón 132**



Fuente: elaboración propia.