# UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

*La Universidad Católica de Loja*

## ÁREA TÉCNICA

## MAGÍSTER EN CIENCIAS Y TECNOLOGÍAS DE LA COMPUTACIÓN

TRABAJO DE TITULACIÓN

# Modelo para detectar la atención de los estudiantes en un salón de clases basado en redes neuronales convolucionales.

**Autor**: Saavedra García, Diego Medardo

**Director**: Cordero Zambrano, Jorge Marcos

LOJA - ECUADOR

2021



*2021*

**Aprobación del director del trabajo de titulación**

Loja, 22 de marzo de 2021

Doctor.
Rommel Torres Tandazo
**Coordinador de la Maestría en Ciencias y Tecnologías de la Computación**
Ciudad. -

De mi consideración:

El presente trabajo de titulación denominado: **"Modelo para detectar la atención de los estudiantes en un salón de clases basado en redes neuronales convolucionales**" realizado por Saavedra García Diego Medardo, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación de este. Así mismo, doy fe que dicho trabajo de titulación ha sido revisado por la herramienta anti-plagio institucional.

Particular que comunico para los fines pertinentes.

Atentamente,

Jorge Marcos Cordero Zambrano.
C.I: 0703084707

**Declaración de autoría y cesión de derechos**

"Yo, Diego Medardo Saavedra García, declaro y acepto en forma expresa lo siguiente:

- Ser autor(a) del Trabajo de Titulación denominado: Modelo para detectar la atención de los estudiantes en un salón de clases basado en redes neuronales convolucionales., del Programa de posgrado Maestría en Ciencias y Tecnologías de la Computación, específicamente de los contenidos comprendidos en: Capítulo 1. Introducción, Capítulo 2. Marco Teórico, Capítulo 3. Propuesta, Capítulo 4. Metodología, Capitulo 5. Experimentación y Análisis de Resultados, Conclusiones y Recomendaciones, siendo Jorge Marcos Cordero Zambrano, director del presente trabajo; y, en tal virtud, eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones judiciales o administrativas, en relación con la propiedad intelectual. Además, ratifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo son de mi exclusiva responsabilidad.
- Que mi obra, producto de mis actividades académicas y de investigación, forma parte del patrimonio de la Universidad Técnica Particular de Loja, de conformidad con el artículo 20, literal j), de la Ley Orgánica de Educación Superior; y, artículo 91 del Estatuto Orgánico de la UTPL, que establece: "Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad".
- Autorizo a la Universidad Técnica Particular de Loja para que pueda hacer uso de mi obra con fines netamente académicos, ya sea de forma impresa, digital y/o electrónica o por cualquier medio conocido o por conocerse, sirviendo el presente instrumento como la fe de mi completo consentimiento; y, para que sea ingresada al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública, en cumplimiento del artículo 144 de la Ley Orgánica de Educación Superior.

Firma: .....

Autor: Diego Medardo Saavedra García.

C.I.: 1104520281

## Dedicatoria

Este trabajo de tesis lo dedico a mi hija, mi principal motor para poder tener la motivación necesaria que me impulsa día a día a seguir adelante.

A mis padres, sin su apoyo no habría sido posible para mí haber podido continuar mis estudios de cuarto nivel

A mis estudiantes quienes ven en mí un reflejo para de superación constante y sin importar el sacrificio que sea necesario hacer, las innumerables noches, días y hasta semanas en vela tienen su recompensa

A cada una de las personas que pudo colaborar con un tutorial, un vídeo, un curso, una url o tan solo palabras de aliento, son pequeños pasos que me impulsaron para seguir, convirtiéndose en el pilar fundamental de apoyo en el proceso de desarrollo de la presente investigación.

## Agradecimiento

A la universidad que me abrió sus puertas para poder conocer el camino de la investigación y las nuevas tendencias en tecnología.

A los docentes de maestría que supieron poner de manifiesto todo su saber en las aulas de clase para que pudiera apropiarme de dicho conocimiento y lo pueda replicar en los trabajos actuales o en un futuro cercano.

Al director de la maestría, ya que con sus consejos siempre me brindó un apoyo constante en el transitar de las aulas o con algún trámite que necesitaba desarrollar para poder avanzar.

A Yefferson Torres y Juan Pinzón, amigos, académicos e investigadores, que aun conociendo todas las ocupaciones por las que atraviesan en sus distintas responsabilidades de padres, profesionales, entre otras, supieron darse el tiempo necesario para revisar, reescribir, modificar los scripts que utilizaba tanto en la arquitectura de la RNC y en el script de VA, los cuales supieron guiarme de la mejor forma para poder crear los mismos en las etapas más importantes.

A quienes lean este trabajo de investigación y puedan sacar provecho de él, a ustedes con mucho afecto y dedicación.

# Índice de Contenidos

## Índice de Tablas

## Índice de Figuras

**Resumen**

En esta investigación se emplearon técnicas de Visión Artificial (VA) utilizando una arquitectura de Red Neuronal Convolucional (RNC) para poder determinar los niveles de atención. La RNC crea un modelo de Inteligencia artificial (IA) que permite detectar la atención. Esta investigación se desarrolla con casos de estudio aislados, utilizando el sensor de la cámara del computador en tiempo real o a través de grabaciones en video. Se implementa la metodología SCRUM, así como diversas herramientas para la creación de RNC y sus diferentes etapas de clasificación, preentrenamiento, entrenamiento, compilación, almacenamiento, evaluación, pruebas y experimentación. Los resultados permiten observar que el modelo de IA alcanza un aprendizaje superior al 90%, si el reconocimiento facial supera el 0.5% de detección del rostro. El modelo de IA alcanza una confiabilidad de 99.64% para la detección de atención y 64.26% para la detección de inatención. Finalmente, este modelo podría permitir a las autoridades educativas en distintos niveles a tomar mejores decisiones respecto a las clases que se imparten en las aulas de forma presencial o virtual.

*Palabras claves***:** Atención, detección de atención, visión artificial, Redes Neuronales Convolucionales.

**Abstract**

In this research, Artificial Vision (VA) techniques were used using a Convolutional Neural Network (RNC) architecture to determine the levels of attention. The RNC creates an Artificial Intelligence (AI) model that allows attention to be detected. This research is developed with isolated case studies, using the computer's camera sensor in real time or through video recordings. The SCRUM methodology is implemented, as well as various tools for the creation of RNC and its different stages of classification, pre-training, training, compilation, storage, evaluation, testing and experimentation. The results allow us to observe that the AI model achieves a learning higher than 90% if facial recognition exceeds 0.5% of face detection. The AI model achieves a reliability of 99.64% for attention detection and 64.26% for inattention detection. Finally, this model could allow educational authorities at different levels to make better decisions regarding the classes that are taught in the classrooms in person or virtually.

Keywords: Attention, attention detection, artificial vision, Convolutional Neural Networks.

# Capitulo uno

## 1. Introducción.

Según Altinkaya & Yalçin. (2020) la atención es un proceso cognitivo que se produce en nuestro cerebro el mismo permite procesar estímulos que se adquiere del entorno, como pensamientos que surgen de nuestro interior y acciones relevantes, de la misma forma permite ignorar acciones irrelevantes o distractores. Por tal motivo, se procede a analizar la atención considerando algunos conceptos de distintos investigadores pertenecientes a áreas como la psicología y educación.

Para Díaz. (2017) la atención y la concentración tienen como propósito que nuestro cerebro impida la sobrecarga de información en la actividad mental, según Raymond & O'Brien, (2009) esto permite que el cerebro humano se sobrecargue, y la información pueda ser analizada de forma efectiva. Por consiguiente, este proceso es analizado desde un enfoque psicológico.

Citando a Khan et al, (2019) la atención se gana a través de la percepción (a través de los sentidos), la excitación o mediante eventos (sucesos, fenómenos) que resulten novedosos, sorprendentes o inciertos, de la misma forma permite mantener la atención de estudiantes. Los docentes pueden obtener la atención utilizando preguntas o problemas desafiantes que permitan estimular la curiosidad, asimismo es posible captar la atención utilizando métodos que puedan influir en la participación por ejemplo el humor o conflicto.

Posner et al, (1982) crean el concepto de modelos atencionales, en investigaciones más recientes como la de Santalla, (2017) se define los modelos atencionales como un mecanismo central en nuestro cerebro que actúa directamente sobre la activación de procesos y operaciones como la selección, la distribución y el mantenimiento de la actividad psicológica; es decir la atención es un proceso psicológico que se produce al interior de nuestro cerebro.

Acorde a Lemonnier et al, (2020) la atención visual es la asignación de recursos atencionales de abajo hacia arriba, es decir, es impulsada por datos lo cual depende de las

características de la escena visual que se puede percibir. La atención visual es aquella aproximación que nos permite determinar si una persona está poniendo atención o está inatenta.

Citando a Leon B., (2008, p. 3) las medidas de atención son un buen punto de partida como predictor escolar, aquellos alumnos que alcanzan buenas notas tienen mayor atención selectiva, su atención dividida es mejor y tienen menos errores. Recíprocamente aquellos alumnos que son inquietos y distraídos logran un bajo rendimiento escolar, en las pruebas de atención también suelen salir con un bajo resultado, de igual manera en aquellos alumnos con problemas de déficit de atención.

Según Madsen et al, (2021) la inatención es un problema latente en las aulas de clase, y se evidencia cuando los estudiantes no pueden comprender el material didáctico relevante propuesto por los docentes, una de las causas es no prestar atención, lo cual repercute en un bajo rendimiento en sus calificaciones. La inatención es un problema muy común en las aulas de clase, pese a que se esfuerzan los docentes en generar mejores clases, no se puede llegar a todo el alumnado, cada día los docentes buscan nuevos métodos, técnicas y estrategias para captar el interés por las clases.

Para Anaya-Jaimes et al, (2020) los modelos de atención visual creados hasta el momento no han podido medir bien la atención visual, uno de los inconvenientes presentados es no haber podido tomar en cuenta los entornos y contextos. Por lo tanto, los modelos de tención visual hasta el momento han sido ineficaces.

Según Gupta et al, (2019, p. 1) los procesos de aprendizaje permiten que nuestro cerebro desarrolle asociaciones a partir de estímulos visuales de recompensa o castigo, complementando Raymond & O'Brien, (2009) menciona que cuando la atención es limitada, el procesamiento visual está sesgado a favor de los estímulos asociados a la recompensa. En las aulas de clase la recompensa y castigo suele ser utilizada en el proceso de enseñanza/aprendizaje.

Citando a Lateef & Ruichek, (2019, p. 1) menciona que existe el campo de las Redes Neuronales de IA, un campo muy prometedor en tareas de aprendizaje supervisado y no

supervisado (redes neuronales artificiales que pueden aprender con la interacción del ser humano o sin ella), De la misma forma dentro de este campo existe el aprendizaje profundo (Deep Learning), según Yann Lecun et al, (2015) este campo permite a modelos computacionales que están compuestos de múltiples capas de procesamiento, los cuales pueden aprender representaciones de datos con deferentes niveles de abstracción. Por otra parte Massiris et al, (2020) en su investigación menciona que las RNC son aquellas redes las encargadas de extraer las características de una imagen y luego usar dichas características para detectar o clasificar los objetos en una imagen. Las RNC son un subcampo de las redes de aprendizaje profundo, que a su vez se encuentran dentro de la Redes Neuronales de IA.

Por lo tanto, la presente investigación toma en cuenta la atención de los estudiantes y propone como punto de partida la atención visual, el movimiento ocular y la posición de la cabeza como parámetros concluyentes para poder medir la atención, y lo cual, propone la creación de una arquitectura de RNC que permita la generación de una IA, utilizando tecnología de VA la IA generada pueda predecir la atención de un estudiante en un aula de clases.

Utilizando reconocimiento facial se pueda analizar un rostro siempre y cuando este pueda superar el 0.5% de detección, se podrá utilizar como parámetro la IA, a su vez esta podrá determinar si el rostro detectado está prestando atención o siendo objeto de inatención.

Después de realizar las fases de pruebas y experimentación el modelo creado obtiene una confiabilidad superior al 90% al predecir la atención y un 64.26% al predecir la inatención.

## Planteamiento del problema.

Según Altinkaya & Yalçin, (2020) la atención es un proceso cognitivo que se produce al interior de nuestro cerebro, el mismo permite procesar estímulos que se adquiere del entorno, como pensamientos que surgen de nuestro interior, por otra parte permite ignorar acciones irrelevantes o distractores. Por lo tanto, existen estímulos que surgen desde nuestro exterior, así como de nuestro interior, al igual que distractores, lo cual simplifica la forma de entender cómo se produce la atención en nuestro cerebro.

Citando a Zuñiga. (2007) la atención en educación despierta tanto en profesores como en autoridades la necesidad de propiciar mejores condiciones escolares que permitan estimular el interés de sus estudiantes. Si la enseñanza es apropiada existe interés por parte del estudiante para atender, con las limitaciones de sus habilidades y capacidad, Por lo tanto, no basta con proponer estrategias, estas se deben aplicar de forma precisa y clara de acuerdo con el programa de estudios, la organización y los lineamientos propuestos para poder mejorar la atención del estudiante.

Existen muy pocas investigaciones que permitan detectar la atención de alumnos en un aula de clases, Madsen et al, (2021); Papoutsaki et al, (2016) realizan experimentos de atención utilizando el seguimiento ocular, otras aproximaciones existen desde el campo de la psicología, sin embargo, son test a lápiz y papel, en otros casos a través de electroencefalogramas, estas técnicas se consideran intrusivas, ya que el usuario debe ser sometido a sensores en la piel, lo cual resulta un poco molesto, sin embargo solo determinan algunas características en diferentes tiempos como milisegundos, minutos (Rodríguez Artacho, 2011).

Mora, (2014) expresa que “Sólo se puede enseñar a través de la alegría” y resalta además que “Sólo se puede aprender lo que se ama”. La alegría con la que un docente se relaciona con sus saberes es una de las más eficaces herramientas para derrotar la apatía y generar atención. La intervención de la pasión en relación con lo nuevo genera una apertura que facilita el aprendizaje. “La atención, ventana del conocimiento, despierta cuando hay algo nuevo en el entorno”, sin embargo, ese algo debe ser percibido como un objeto que se

relaciona directamente con el sujeto porque “La atención nace de algo que tiene que ver con nuestra propia vida”. Puede que quizá no exista un lazo más profundo que el afecto y la emoción, por eso resume Mora: “Sin emoción no hay curiosidad, no hay atención, no hay aprendizaje”.

## 1.2. Objetivo General.

Definir un modelo para analizar el nivel de atención de los estudiantes en un salón de clases para coadyuvar en el proceso de enseñanza-aprendizaje.

### 1.2.1. *Objetivos Específicos*.

- Revisión de literatura actualizada referente a la atención en clases, para determinar la relación entre los niveles de atención con el rendimiento académico.
- Analizar y seleccionar redes neuronales convolucionales, para identificar el nivel de atención de los estudiantes.
- Analizar el nivel de atención en el aula, para emprender acciones para recuperar la atención.
- Experimentar y analizar los resultados.

## 1.3. Alcance.

La presente investigación pretende la creación de una arquitectura de RNC, que permita obtener un modelo de IA, el cual permita a través de técnicas de Visión Artificial detectar la atención.

Este modelo entrenado a través de RNC permitirá no solo detectar la atención sino también la inatención de estudiantes, utilizando el sensor de la cámara web en tiempo real o a través de un vídeo pregrabado en clase, en la actualidad las clases se realizan de forma virtual, lo cual también será posible analizar mediante la arquitectura propuesta en esta investigación.

El modelo de IA resultado de la presente investigación permitirá a directivos de instituciones educativas tomar mejores decisiones que se relacionen directamente con el proceso de enseñanza/aprendizaje que se desarrolla en clase.

## 1.4. Organización de la Tesis.

En el capítulo uno se conoce la introducción de la presente investigación, el problema a investigar, los objetivos de la presente investigación, el alcance que busca y la organización de la tesis.

En el capítulo dos se encuentra el marco teórico conceptual donde se analiza algunos conceptos importantes de varios autores, se realiza una revisión sistemática de literatura, la misma que permite conocer investigaciones relacionadas, se conoce métodos, técnicas, algoritmos, avances tecnológicos dentro del área de estudio y se define un camino para encontrar la solución al problema planteado.

En el capítulo tres se desarrolla la propuesta de la presente investigación, se define la arquitectura de implementación, las herramientas a utilizar, los lenguajes de programación, los entornos de desarrollo, el proceso que se lleva a cabo para el desarrollo de la RNC, los dataset a utilizar, la generación de criterios de selección de la data, la clasificación, el tratamiento, la arquitectura de la RNC, y como se produce la detección de la atención a través de VA.

En el capítulo cuatro se desarrolla la metodología utilizada en la presente investigación, la implementación, como se realiza la recolección de información, la creación de la RNC y el script de VA, como también los resultados y conclusiones previas.

En el capítulo cinco se desarrolla la etapa de experimentación y análisis de resultados, como se produce la obtención de resultados, la detección de la atención y la inatención.

Finalmente se redacta la discusión, conclusiones y recomendaciones de la presente investigación.

# Capítulo dos

## 2. Marco Teórico.

En este capítulo se presentan los conceptos relacionados con la atención, inatención, niveles de atención, arquitecturas de redes neuronales convolucionales, configuración de parámetros para las diferentes etapas de creación y entrenamiento de las RNC y con ello la generación de modelos de IA, evaluación y aplicación a través de VA. Iniciando por la Revisión Sistemática de Literatura.

### 2.1. La Atención.

Según Altinkaya & Yalçin, (2020) La atención es un proceso cognitivo que se produce en nuestro cerebro, permite procesar estímulos que se adquiere del entorno, pensamientos que surgen de nuestro interior y acciones relevantes, así como ignorar acciones irrelevantes o distractores. Al ser un proceso cognitivo que se produce a lo interno del cerebro humano es necesario poder determinar cómo se puede medir la atención, y determinar cuáles son los medios que permitan determinar cuando una persona está atendiendo o está siendo objeto de inatención.

Citando a Gazzaniga et al. (2008) el problema que presenta el estudio de la atención desde el punto de vista biológico es como cerebro es capaz de seleccionar cierto tipo de información y desecha otra que le parece irrelevante, en la presente investigación se pretende analizar cuáles son los factores que influyen en el ser humano para poder prestar atención.

Según Moraine. (2014) de acuerdo con las funciones ejecutivas del estudiante, la atención reside en el corazón de cada una de las experiencias que vivimos todos los días, y también es una de las más destacadas de nuestras funciones ejecutivas. Es la base de nuestras experiencias, es la esencia de lo que nos hace ser humanos. Por lo tanto, aquello que nos permite sentirnos humanos es aquello a lo que prestamos atención la mayor parte de nuestras vidas.

Según C. Davidson. (2011) en su libro “Ahora lo ves: Cómo la ciencia del cerebro de Atención transformará la manera de vivir, trabajar, y aprender” menciona que las maneras en las que prestamos atención en el siglo XXI son muy diferentes a las que utilizaron las generaciones anteriores. Se puede decir que en la actualidad existen más distractores que en el pasado, nuestros sentidos están llenos de estímulos que fácilmente hacen que los seres humanos perdamos el foco de atención de aquello que nos interesa aprender. Esto es muy habitual

#### *2.1.1. Tratar de controlar la atención.*

Como educadores y padres nos esforzamos intensamente intentando dirigir y controlar la atención de otros. “¡Presta atención! ¿Estás atendiendo? ¿Me estás escuchando? ¿Por qué no me escuchas cuando te hablo?”. Son algunas de las interrogantes que decimos cuando se busca que la atención se dirija a algún punto en particular. Cuando más pequeño es un niño, más control intentan ejercer los padres sobre su atención. Al crecer los niños los padres se sienten más frustrados, porque disminuye año tras año su habilidad para controlar la atención del niño. Cualquier padre de un adolescente está de acuerdo en lo complicado que es intentar que su hijo preste atención específicamente. No es sorprendente que los adolescentes se resistan a que se les diga a qué tienen que hacer caso. De hecho, no sería una sorpresa que cualquier persona, de cualquier edad, se resista a que le digan cómo usar su atención.

En las clases, es complicado poder captar la atención de los estudiantes. Los profesores pasan gran parte del tiempo intentando que los estudiantes presten atención a lo que están diciendo, haciendo o explicando, mucho de ello tiene que ver con los métodos, técnicas y estrategias de aprendizaje que se aplica, la lucha por la atención empieza cuando el docente se levanta en la mañana, y no termina hasta que el sueño pone fin a la lucha, al final del día.

Se puede caer en la trampa de intentar estructurar la atención, pidiendo que el niño esté atento a lo que se le dice que debería hacer. Que el mismo Intente experimentarlo por sí mismo, empiece un día tratando de escuchar todas las diferentes formas en las que intenta que alguien dirija la atención en algún sentido. Sí le parece un período de tiempo demasiado largo, lo intenta al menos por una hora, o incluso durante 15 minutos. Los comentarios de dirigir la atención son fáciles de reconocer en nuestra interacción con los niños. Y a la vez permite obtener buenos resultados si la didáctica utilizada es la más adecuada al contexto, temática o interés por parte del niño.

Casi todo lo que se le dice al niño tiene un elemento de dirección. Actívelo y escuche las maneras en que intenta controlar la atención de los adultos, amigos y colegas. Si se tiene algún adolescente cerca, ¿suele cambiar su lenguaje con ellos? ¿Cambia el tono que usa cuando habla con varias personas? ¿Intenta que los demás escuchen lo que está diciendo por la forma en que participa en la conversación?

La atención se basa en dirigir la atención del niño a un contenido específico, a unas acciones específicas y a unas interacciones específicas. Nuestra labor como educadores y padres es dirigir nuestros esfuerzos a poder guiar, dirigir, formar, encauzar, estructurar y educar la manera en que los

niños usan su atención. Es también nuestro trabajo hacer lo mismo con nuestra propia atención. ¿Qué es más difícil? ¿Es más fácil intentar cambiar la atención de otra persona o la nuestra? Hay diferentes respuestas a esta pregunta, pero si se responde honestamente, estará en el camino correcto para entender los factores más importantes en la atención.

Lo que trata de decir en estas líneas (LeCun, Yann, Bengio, 2017) es que a menudo como padres o docentes tratamos de conducir (controlar) la atención de nuestros hijos y/o estudiantes hacia algún evento que consideramos aportará a su formación.

### *2.1.2. La atención y la educación.*

En la década anterior (Fredricks et al., 2004, pp. 71, 78) Los estudios del campo del aprendizaje han demostrado los beneficios de rendimiento de uso de la estrategia. Los niños que utilizan estrategias metacognitivas, como regular su atención y esfuerzo, relacionando nueva información con el conocimiento existente, y activamente monitorear su comprensión, mejorar en varios indicadores de rendimiento académico… Los estudiantes pueden completar asignaciones prestando atención y permaneciendo concentrado en la tarea y utilizando estrategias de aprendizaje para memorizar en lugar de estrategias más profundas para comprender lo que está siendo enseñado.

Según (Piaget, 2005) la atención referente a las cuestiones afectivas sobre el desarrollo del conocimiento, no puede explicar por qué se inicia y se concluye la experiencia de dicho conocimiento, por qué nos interesan unas cosas y no otras y también por qué nos interesan unas personas y no otras.

Esto tiene que ver con los intereses del niño, si algo no le llama la atención lo suficiente como para poner a disposición sus recursos atencionales implicará que el mismo ocupe su tiempo en otras actividades más excitantes de lo que el docente intenta impartir y en este punto se produce la inatención que puede derivar en distintos factores, una mala práctica docente, falta de planificación, falta de conocimientos por parte del mismo o la forma en que muestra su expertis frente al público al que se dirige.

Según (Epigeum, 2011) tenemos 2 tipos de atención en el aula de clases, la atención pasiva y la atención activa. Las conferencias generalmente están programadas para durar alrededor de una hora, y lo que los estudiantes hacen durante la mayoría de las conferencias convencionales implica atención “pasiva” (como escuchar y grabar algo de lo que se dice o se muestra visualmente). Cuando estamos siendo en gran parte pasivos, el cerebro tiene una capacidad limitada para mantener la atención y maximizar el rendimiento durante mucho tiempo.

Las emociones se pueden considerar como un sentimiento que resulta en efectos físicos y psicológicos. cambios que controlan nuestro comportamiento. El progreso en el reconocimiento de emociones ha avanzado significativamente en las últimas dos décadas, con la contribución de muchos campos disciplinarios como la psicología, la neurociencia, la endocrinología, la medicina, la sociología e incluso la informática. Para el reconocimiento de las emociones, la actividad cerebral es esencial, involucrando motivación, percepción, experiencia, conocimiento, cognición, creatividad, atención, aprendizaje y toma de decisiones. Básicamente, hay dos clases amplias de reconocimiento de emociones con tres modalidades diferentes.

- Unimodal: reconocimiento de emociones utilizando una única modalidad como entrada al sistema.
- Bimodal: el sistema de reconocimiento incluye dos modalidades como entrada al sistema.
- Multimodal: incluye más de dos modalidades de reconocimiento de emociones.

Estudios recientes han demostrado que las amplitudes de ECoG en ciertas bandas de frecuencia transportan información sustancial sobre la actividad relacionada con la tarea, como la ejecución y planificación motora, el procesamiento auditivo y la atención visual-espacial (Ehrotra, 2018).

La tención a nivel cerebral según (Ehrotra, 2018) considera que una onda gamma es la actividad cerebral más rápida. Es responsable del funcionamiento cognitivo, aprendizaje, memoria y procesamiento de información. La prominencia de esta ola conduce a ansiedad, alta excitación y estrés; mientras que su supresión puede conducir al trastorno por déficit de atención con hiperactividad (TDAH), depresión y problemas de aprendizaje. En condiciones óptimas, las ondas gamma ayudan con la atención, el enfoque, la unión de los sentidos (olfato, vista y oído), la conciencia, el procesamiento mental y la percepción. Esto es de suma importancia para conocer cómo se produce la atención a través de los sentidos.

### *2.1.3. Inhibición*.

Según (Korzeniowski, 2018, p. 13) la inhibición es un constructo multidimensional que trata de explicar una serie de operaciones mentales tendientes a suprimir una conducta inapropiada, o una tendencia atencional hacia estímulos no relevantes o distractores que pueden interferir en la resolución deliberada de un problema. El control inhibitorio está construido por diferentes aspectos disociables entre sí. La mayoría de los expertos en el tema señalan que puede subdividirse en: inhibición

conductual e inhibición de la atención (Barkley & Wasserstein, 2000; Diamond, 2013, pp. 135–168; Friedman & Miyake, 2004, pp. 101–135). La inhibición conductual comprende de tener una respuesta dominante, suprimir una respuesta en curso y el cambio de un patrón de respuesta a otro (Barkley & Wasserstein, 2000; Diamond, 2013). La inhibición de la atención implica inhibir estímulos relevantes optimizando los procesos de focalización, sostenimiento y cambio atencional (Barkley & Wasserstein, 2000; Verbruggen et al., 2006, pp. 190–203; Wright & Diamond, 2014, pp. 1–9). Uno de los procesos clave en el cambio atencional es el control de la interferencia, el cual involucra procesos de atención selectiva e inhibición cognitiva (Diamond, 2013) necesarios para la resolución de tareas que implican dirimir conflictos entre estímulos competitivos. La inhibición de la atención posibilitaría entonces implica suprimir la activación, el procesamiento o la expresión de información que potencialmente pueden interferir con el logro de una meta (Christ et al., 2006, pp. 845–864).

Dawson y Gare, 2010 sugieren pautas de modificación del entorno para aumentar el nivel de atención y mejora en el control de inhibición de respuesta en las actividades de clase. Propone modificaciones del entorno y de las actividades que se le proponen de modo de disminuir el uso de las habilidades ejecutivas deficitarias lo que se combina con un trabajo conjunto en dichas habilidades para realizar paulatinamente una reducción gradual de estas adecuaciones. La intervención procura que el niño o joven tome consciencia de los tiempos que demanda una tarea, en la división de tareas complejas y concesión de descansos luego de completar cada subetapa, también resulta central la toma de conciencia sobre los estímulos o aspectos que contribuyen a su motivación para mantener su atención (Yoldi, 2019).

### 2.2. Distracción.

Los estímulos que afectan la atención selectiva es el caso de la distracción social cuando las personas no son objeto de atención. Los rostros de los distractores muestran efectos de interferencia en la búsqueda visual incluso bajo una alta carga perceptiva, lo que sugiere un procesamiento obligatorio (Lavie et al., 2003). Los efectos de tal distracción no han sido investigados más allá de la atención selectiva dentro de las tareas perceptivas (Ruth Doherty et al., 2017).

Los rostros también muestran un papel especial en la atención selectiva (P Vuilleumier et al., 2001; Patrik Vuilleumier, 2000), incluso en las tareas de detección de cambios (Ro et al., 2001) y las señales tareas (Langton & Bruce, 1999).

Según la investigación de (Ruth Doherty et al., 2017) basado en la literatura previa sobre sesgos atencionales. Concluye que los encuentros con estímulos sociales que distraen no solo afectan el aquí y ahora, sino también los recuerdos posteriores construidos a partir de interacciones perceptivas de aprendizaje. Los sesgos de atención no solo operan en un dominio de atención selectiva, sino que también tienen consecuencias funcionales sobre la memoria, que a su vez pueden reforzar estos sesgos. Los sesgos de atención a menudo están implicados en estos trastornos. Este estudio proporciona nuevos conocimientos sobre el papel de los sesgos de atención en general, y el estado privilegiado de los estímulos sociales en particular, para la memoria.

Por otra parte la investigación de (Wetzel et al., 2019) describe el curso del tiempo de desarrollo del control de la atención audiovisual durante la primera infancia y más allá. El desempeño exitoso del paradigma de distracción actual requiere funciones cognitivas básicas como la atención sostenida y enfocada en los estímulos objetivo, la memorización de las características objetivo y las relaciones estímulo-respuesta, la evaluación de la información entrante para la relevancia de la tarea, así como la selección de tarea relevante y la inhibición del procesamiento de estímulos irrelevantes para la tarea. Estas funciones se superponen con los componentes de la función ejecutiva que se desarrollan considerablemente durante la primera infancia y mejoran aún más en los niños en edad escolar (M. C. Davidson et al., 2006; Hughes et al., 1998). El control de la atención en situaciones audiovisuales a menudo se requiere en la vida diaria, y se sugiere que el nivel de maduración de los mecanismos subyacentes afecta el desempeño en una variedad de tareas. El control de la atención en situaciones audiovisuales a menudo se requiere en la vida cotidiana, y se sugiere que el nivel de maduración de los mecanismos subyacentes afecta el desempeño en una variedad de tareas. El conocimiento mejorado sobre el curso del tiempo de desarrollo del control de la atención en el contexto de sonidos ambientales que ocurren sorprendentemente podría contribuir a adaptar las condiciones de aprendizaje a las necesidades de los niños que son diferentes en los diferentes grupos de edad. Los resultados también podrían influir en la investigación sobre trastornos que incluyen déficits en el control de la atención o la autorregulación.

William James (1890) afirmaba: “Todo el mundo sabe lo que es la atención. Es tomar posesión de la mente, de una forma clara y vivida, de uno de los que parecen ser diferentes objetos o líneas de pensamiento que suceden de forma simultánea. Su esencia son la localización y la concentración de

la conciencia. Implica dejar de lado algunas cosas para poder tratar de forma efectiva otras" (James, 2007).

La atención se puede definir como la capacidad de seleccionar y concentrarse en los estímulos más relevantes. Por lo tanto, la atención es el proceso cognitivo que nos permite orientar los recursos atencionales hacia los estímulos relevantes y procesarlos para responder en consecuencia.

La capacidad de generar, seleccionar, dirigir y mantener un nivel de activación adecuado para procesar la información relevante. Dicho de otra forma, la atención es un proceso que tiene lugar a nivel cognitivo y que permite orientarnos hacia aquellos estímulos que son relevantes, ignorando los que no lo son para actuar en consecuencia (Bitbrain, 2018).

La atención ha sido estudiada con numerosas metáforas. Ha sido tratada como si representara un filtro (Broadbent, 1958), esfuerzo (Kahneman, 1973), recursos (Shaw y Shaw, 1977), como un proceso de control de la memoria operativa (Shiffrin y Schneider, 1977), orientación (Posner, 1980), como conexión entre diversas características de los estímulos (Treisman y Gelade, 1980), como un foco (Eriksen y St. James, 1986; Tsal, 1983), y como un proceso de selección más una actividad preparatoria (LaBerge y Brown, 1989).

Dentro del campo de la psicología existe el término Atención Selectiva. Es un término complejo para el que no contamos con una definición consensuada y unívoca (Cowan, 1995). Dada su complejidad

Dentro del ámbito de brain-computer interfaz (Interfaz Cerebro-Computadora) e Internet of Things (Internet de las Cosas) se utiliza *selective attention mechanism* (mecanismo de atención selectiva). Según (Zhang et al., 2018), en su investigación proponen un framework de deep learning para unir una interfaz cerebro-computadora e internet de las cosas, utilizan WAS-LSTM para extraer la dependencia inter-dimensional entre la señal de entrada de las actividades del cerebro humano que son seleccionadas por el mecanismo de atención selectiva.

Wertheimer en 1912 (Psychologie et al., 1910) describió por primera vez uno de los fenómenos que se incluyen en esta categoría, y ahora a menudo se lo denomina movimiento aparente basado en la atención por ejemplo, (Verstraten et al., 2000, 2001) o movimiento basado en la atención en el caso de un estímulo continuo (Otting et al., 1992). En un típico experimento de seguimiento atento, se requiere que los observadores mantengan la fijación en un punto en el centro.

### 2.3. *Test de Atención.*

Los psicólogos para sus investigaciones utilizan test que les permite evaluar la atención de los cuales se toma en cuenta los más relevantes y con mejores resultados obtenidos a lo largo de los años. A continuación, se describen algunos test para determinar la atención.

### *2.3.1. Prueba de cinco dígitos (FDT).*

La prueba de cinco dígitos (5D), propuesta por Sedó 2004, es una adaptación MVT del SCWT. Al realizar esta prueba, el sujeto debe conocer solo los primeros cinco números y sus símbolos correspondientes. La prueba mide el desempeño verbal continuo en diferentes niveles de la red de atención porque prueba tanto un proceso más "automático" como un proceso más "controlado", en el cual el sujeto debe inhibir una rutina automatizada de procesamiento en favor de una secundaria, modo intuitivo de procesamiento (Paula & Ávila, 2011).

### *2.3.2. Test de Red Atencional (Attentional Network Test ANT).*

La ANT es una prueba relativamente breve, de unos 20 minutos aproximadamente de administración, que proporciona una medida de la eficiencia de las redes atencionales implicados en alerta, orientación y atención ejecutiva. Está diseñada para ser utilizado con niños (a los cuales se les presenta una variación de la prueba en la cual se utilizan peces de colores como estímulos, mucho más estimulantes, ante los que tienen que responder), adultos, pacientes con diversas alteraciones de la atención y primates no humanos, ya que no está influida por el idioma (Sáez et al., 2016).

### *2.3.3. Test de Atención D2.*

La prueba “d2” pertenece a la categoría de los instrumentos que pretenden medir estos procesos básicos. En Alemania son conocidos como test de concentración o test de atención selectiva, y en los Estados Unidos reciben la denominación de prueba de amplitud atencional, de atención selectiva o de atención sostenida. En particular, la atención selectiva ha sido ampliamente estudiada en la neuropsicología americana. A menudo definida como concentración, la atención selectiva puede definirse como la capacidad para centrarse en uno o dos estímulos importantes, mientras se suprime deliberadamente la consciencia de otros estímulos distractores. El constructo de vigilancia o atención sostenida, con el que la atención selectiva está relacionada, se refiere a la capacidad de mantener una actividad atencional durante un periodo de tiempo. La prueba d2 es una medida concisa de la atención selectiva y la concentración mental. El constructo de atención y concentración alude a una selección de estímulos enfocada de modo continuo a un resultado. La parte central de estos procesos es la capacidad de atender selectivamente a ciertos aspectos relevantes de una tarea mientras se ignoran

los irrelevantes y, además, hacerlo de forma rápida y precisa. De acuerdo con esta definición, el d2 supone una actividad de concentración con respecto a estímulos visuales. Una buena concentración requiere un funcionamiento adecuado de la motivación y del control de la atención. Estos dos aspectos, aplicados al d2, se reflejan en tres componentes de la conducta atencional (Brickenkamp, 2012)

### 2.4. La Atención y el rostro.

A partir del 2000 se realizan investigaciones para encontrar relación entre el rostro humano y la atención, la investigación de (Ro et al., 2001) demuestra que los rostros juegan un papel especial en la atención.

Esto permite conocer que ciertos rasgos que se producen en el rostro como la mirada y la dirección de esta se producen cuando una persona está poniendo atención a cierto estímulo que permite que los recursos atencionales en este caso la vista fija a cierto fenómeno, suceso o acontecimiento se da cuando el individuo presta atención, la inhibición de otros factores demuestra que es una atención sostenida. Esto se produce por el interés, la motivación o curiosidad que la persona tiene acerca del acontecimiento que está sucediendo frente a sus ojos, esto coadyuva a que la inhibición de este pueda ser superior a la inatención de la que podría ser parte.

### 2.5. La atención, la posición de la cabeza y el seguimiento ocular.

La investigación de (Langton & Bruce, 1999) establece claramente que los estímulos faciales, que indican la dirección en virtud de la posición de la cabeza y los ojos, producen una respuesta de orientación reflexiva en nombre del observador. Esto demuestra que uno de los indicadores para poder asegurar que un ser humano está prestando atención es hacia donde dirige la mirada y la posición de la cabeza. Asimismo, (Becker et al., 2007) determina que las fijaciones oculares indican la captura de atención en escenas complejas, por lo tanto hacia donde se encuentra la dirección de la mirada es hacia donde nuestro cerebro está prestando atención en determinado tiempo.

### 2.6. Deep Learning.

En base a (Yann Lecun et al., 2015) se define el Deep Learning como algoritmos que permite la creación de modelos computacionales que están compuestos de múltiples capas de procesamiento, los mismos que permiten aprender de representaciones de datos con diferentes niveles de abstracción. Por lo tanto, la conceptualización de Deep Learning indica que algoritmos que pueden aprender de diferentes fuentes de información (dataset) para encontrar la solución a determinado problema que se investiga.

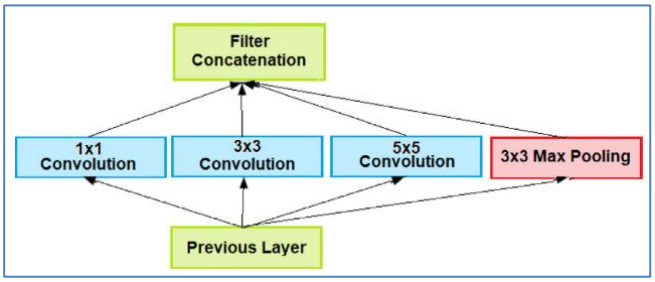
La historia de las redes neuronales data desde 1940, de acuerdo a (Lateef & Ruichek, 2019) por algunas décadas no tuvieron mucha atención, sin embargo, en la década de 1990 tuvieron un gran progreso, esto se debe al aparecimiento de las cámaras digitales, los teléfonos celulares y la potencia informática crece con el aparecimiento de las GPU que poco a poco se ha convertido en una herramienta informática para uso general.

En la actualidad las redes neuronales profundas son muy efectivas en la segmentación semántica, es decir etiquetar cada región o pixel con una clase de objetos, esto es fundamental para las tareas de análisis de imágenes. El aprendizaje profundo (deep learning) es una división del campo del aprendizaje automático (machine learning). Es un campo en constante crecimiento que difícilmente logra mantenerse actualizado, inclusive para realizar un seguimiento de los trabajos relacionados, cada día surgen nuevos métodos, mejoras de los métodos existentes y su implementación en nuevos dominios y aplicaciones (Lateef & Ruichek, 2019).

Gracias al éxito que han tenido las redes neuronales convolucionales profundas (CNN) desde la primera aplicación exitosa creada por Choi et al. (2005). Que introdujeron una arquitectura llamada LeNet5 para leer códigos postales, dígitos y extraer características en varias ubicaciones de la imagen. Más tarde Alex Krizhevsy lanzó una gran CNN profunda AlexNet  gracias a Gonzalez. (2007), esta se considera una de las publicaciones más importantes dentro del campo de las CNN, AlexNet es una versión mucho más amplia y profunda de LeNet, se la utiliza para aprender objetos complejos y jerarquías de estos objetos. Zeiler y Fergus (Suah, 2017) hicieron conocer al mundo ZFNet, un ajuste de la estructura de AlexNet. Ellos propusieron una técnica para visualizar mapas de características en cualquier capa del modelo de red. Esta técnica utiliza una red desconvolutiva multicapa para proyectar las activaciones de características de regreso al espacio de píxeles de entrada. Por otra parte Lin y Col (Lin et al., 2014), proponen un modelo de red en red basado en micro redes neuronales, que es el perceptrón multicapa (MLP) (White & Rosenblatt, 1963), esto quiere decir que su arquitectura consta de múltiples capas completamente conectadas con las funciones de activación no lineales. Szegedy y col. (Garcia-Perez et al, 2019) propuso una red neuronal profunda eficiente llamada GoogleNet. Introdujeron un módulo de inicio como se muestra en la Figura 1, que es una combinación de filtros convolucionales 1 × 1, 3 × 3 y 5 × 5 y una capa de agrupación. Redujo la cantidad de funciones y operaciones en cada capa, lo que ahorró tiempo y costos computacionales.

**Figura 1**

*Redes Neuronal Profunda GoogleNet.*




La Figura 1 es tomada de Garcia-Perez et al, 2019.

En la Figura 1 se puede evidenciar la arquitectura de una Red Neuronal profunda de forma gráfica para entender como los parámetros de entrada van pasando de una capa a otra.

Los mismos autores (Joseph, 2016) propusieron un algoritmo llamado BN-Inception para construir, entrenar y realizar inferencias con el método de normalización por lotes. Szegedy y col (Szegedy et al., 2016), introdujo además dos nuevos módulos Inception V2 e Inception V3 con algunas modificaciones como factorizar las convoluciones y usar la técnica de reducción de cuadrícula de su módulo anterior. Posteriormente, Szegedy et al. (Szegedy et al., 2017) reemplazó la etapa de concatenación de filtros de la arquitectura Inception con conexiones residuales para aumentar la eficiencia y el rendimiento. Propusieron Inception-ResNet-v1, Inception-ResNet-v2 y una variante pura de Inception llamada Inception V4. Chollet y col (Chollet, 2017) propuso un módulo llamado Xception, que significa inicio extremo. Se reemplazaron los módulos de inicio con convoluciones separables en profundidad propuestas en (Mamalet & Garcia, 2012).

El aprendizaje profundo permite que los modelos computacionales de múltiples capas de procesamiento aprendan y representen datos con múltiples niveles de abstracción imitando cómo el cerebro percibe y comprende la información multimodal, capturando así implícitamente estructuras intrincadas de datos a gran escala. El aprendizaje profundo es una rica familia de métodos, que abarca redes neuronales, modelos probabilísticos jerárquicos y una variedad de algoritmos de aprendizaje de características supervisados y no supervisados. Según Voulodimos et al., (A. Voulodimos et al., 2018, p. 1) el reciente aumento de interés en los métodos de aprendizaje profundo se debe al hecho de que se ha demostrado que superan las técnicas de vanguardia anteriores en varias tareas, así como a la abundancia de datos complejos de diferentes fuentes (por ejemplo, visual, audio, médico, social y sensor).

### *2.6.1. Desarrollos y métodos de Aprendizaje Profundo (AP).*

Las RNC se inspiraron en la estructura del sistema visual y, en particular, en los modelos propuestos en (G. Li et al., 2021). Los primeros modelos computacionales basados en estas conectividades locales entre neuronas y en transformaciones jerárquicamente organizadas de la imagen se encuentran en Neocognitron (Fukushima, 1980), que describe que cuando se aplican neuronas con los mismos parámetros en parches de la capa anterior en diferentes ubicaciones, se produce una forma de invariancia tradicional. Yann LeCun y sus colaboradores más tarde diseñaron redes neuronales convolucionales empleando el gradiente de error y obteniendo muy buenos resultados en una variedad de tareas de reconocimiento de patrones (LeCun et al., 1989; Y Lecun et al., 1998; Tygert et al., 2016).

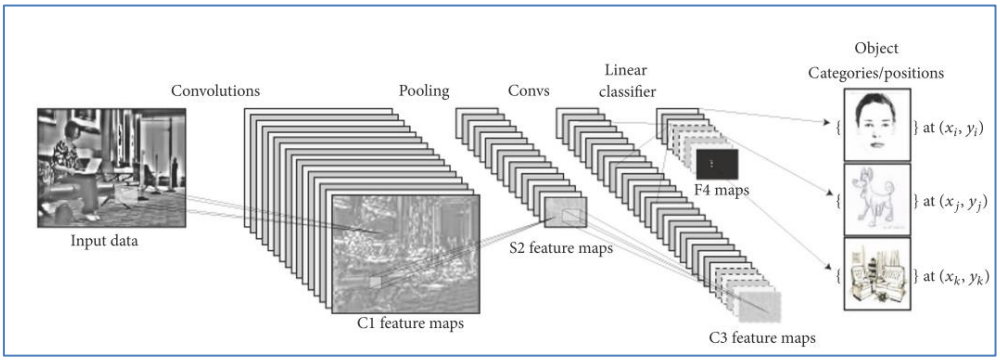
Una RNC comprende tres tipos principales de capas neuronales, capas convolucionales, capas agrupadas y capas completamente conectadas. Cada tipo de capa juega un papel diferente. Cada capa de una RNC transforma el volumen de entrada en un volumen de salida de activación de neuronas, lo que eventualmente conduce a las capas finales completamente conectadas, esto da como resultado un mapeo de los datos de entrada a un vector de características de una dimensión. Las RNC han tenido un gran éxito en aplicaciones de VC, como el reconocimiento facial, la detección de objetos, la potencia de la visión en robótica y los automóviles autónomos (A. Voulodimos et al., 2018). En la presente investigación se propone la arquitectura de RNC por el reconocimiento facial.

## 2.7. Redes Neuronales Convolucionales (RNC).

En la Figura 2 se puede observar una arquitectura de RNC para una tarea de visión artificial, donde se puede evidenciar los parámetros de entrada (imágenes), las capas ocultas intermedias, dentro de ellas las convoluciones, el MaxPolling, capas lineales, capas full conectadas y finalmente las categorías de salida obtenidas por la arquitectura.

**Figura 2**

*Arquitectura de una RNC para una tarea de visión artificial (detección de objetos).*



A. Voulodimos et al, 2018.

Una RNC es un tipo especial de red que reduce significativamente el número de parámetros en una red neuronal profunda con muchas unidades sin perder demasiado en la calidad del modelo. Las CNN han encontrado aplicaciones en el procesamiento de imágenes y texto en las que superan muchos puntos de referencia previamente establecidos. Las CNN se inventaron teniendo en cuenta el procesamiento de imágenes (Burkov, 1997, p. 66).

Las CNN no son sensibles a la posición de los objetos en los datos, por lo que es sencillo aplicarlas a grandes encuestas. Lo más importante es que las CNN no requieren características extraídas manualmente como entradas. En cambio, las CNN extraen características automáticamente de los datos aplicando diferentes filtros en diferentes capas durante el entrenamiento (Xu et al., 2020, p. 3).

El propósito de RNC es extraer las características de una imagen y luego usar dichas características para detectar o clasificar los objetos en una imagen (Massiris et al., 2020).

La optimización de modelos tan grandes es un problema computacionalmente intensivo. Cuando nuestros ejemplos de entrenamiento son imágenes, la entrada es de muy alta dimensión. Si desea aprender a clasificar imágenes, es probable que el problema de optimización se vuelva intratable.

Las RNC son inspiradas en la corteza visual, la cual consiste en mapas de campos receptivos locales que responden a los estímulos únicamente en una región del campo visual, y disminuyen a medida que la corteza se mueve hacia adelante; los campos receptivos se superponen de modo que cubren todo el campo visual que es percibido por los ojos. Las RNC toman la arquitectura de la red neuronal profunda multicapa perceptrón (DMLP), pero su principal diferencia es que cada neurona solo se encuentra conectada a un parche local de neuronas de la siguiente capa (Suah, 2017). Esta arquitectura aprovecha el patrón jerárquico de los datos para ensamblar patrones más complejos usando unos patrones más pequeños y simples. Su nombre indica que la red usa el operador matemático conocido como convolución, que permite realizar operaciones complejas de forma más sencilla. En esta red, los datos de entrada son convolucionados por diferentes filtros durante el desplazamiento a través de las neuronas. Los filtros convolucionales comparten los mismos parámetros en cada pequeña porción la red, reduciendo los parámetros del algoritmo. Su principal

aplicación son las tareas de VA, como la clasificación y segmentación de imágenes (Krizhevsky et al., 2012).

Las RNC según Gonzalez, (2007); Simonyan & Zisserman, (2015) Con el éxito de las RNC utilizando el análisis de datos de cuadrícula regular, muchas arquitecturas basadas en se han diseñado para analizar nubes de puntos 3D. En la actualidad se siguen buscando nuevas arquitecturas, métodos, técnicas y algoritmos que permitan reducir la capacidad computacional que consumen este tipo de redes, y conseguir mejores resultados al momento de resolver problemas puntuales.

Una técnica de aprendizaje profundo de uso común es la RNC. Las RNC son redes neuronales artificiales que involucran una serie de operaciones matemáticas (por ejemplo, convolución, inicialización, agrupación, activación, etc.) y esquemas de conexión (por ejemplo, conexión completa, conexión de acceso directo, apilamiento simple, conexión de inicio, etc.) (G. Li et al., 2021). Los factores más importantes que contribuyen al gran impulso de las técnicas de aprendizaje profundo son la aparición de conjuntos de datos grandes, de alta calidad y disponibles públicamente, junto con el empoderamiento de las unidades de procesamiento de gráficos (GPU), que permiten un cálculo paralelo masivo y aceleran el entrenamiento y prueba de modelos de aprendizaje profundo (A. Voulodimos et al., 2018).

Citando a Zhou et al. (2017) el deep learning, traducido al español significa aprendizaje profundo consiste en un nuevo campo y en rápido crecimiento los últimos años, cada día consigue nuevos investigadores que se dedican al desarrollo de arquitecturas de RNC para la solución de problemas de clasificación.

En comparación con las arquitecturas poco profundas (Redes Neuronales), las RNC tiene grandes ventajas en como la extracción de características y ajuste para la creación de modelos.

Es óptimo para descubrir representaciones de características cada vez más abstractas cuya capacidad de generalización es fuerte a partir de los datos de entrada sin procesar.

Las RNC para solucionar problemas principalmente de clasificación aplican métodos, técnicas en diferentes casos. Cabe recalcar que no son el único tipo de redes neuronales que existen, sin embargo, es importante destacar su importancia. a continuación, se detallan algunos de los más utilizados en estas investigaciones.

### *2.7.1. Métodos de la RNC basados en codificador de funciones.*

Métodos como VGG Simonyan & Zisserman. (2015) y ResNet propuestos por He et al. (2016) son los enfoques que más se utilizan en la extracción de características. Detrás del concepto es extraer mapas de características basados en capas de convolución apiladas, capas ReLu y capas agrupadas. Este método es muy utilizado en investigaciones de vanguardia que su objetivo principal es la obtención de características sin determinar cuáles son las que se obtendrán, esto se remplaza con el uso de parámetros de configuración en la RNC.

### *2.7.2. Red VGG.*

La Red VGG (Simonyan & Zisserman, 2015) es introducida por el Grupo de Geometría Visual (Visual Geometry Group). Por otra parte LeNet (Yann Lecun et al., 2015) y AlexNet (Kingma & Ba, 2015), VGGNet utiliza una gran cantidad de parámetros y una capacidad de aprendizaje debido al efecto de campos receptivos más grandes, p. Ej. 5 × 5 y 7 × 7. Sin embargo, es una convolución múltiple de 3 × 3 en la secuencia que puede coincidir con los clasificadores grandes.

### *2.7.3. Marco de Aprendizaje Residual.*

Los marcos de aprendizaje residual suelen incluir métodos que utilizan el bloque residual (He et al., 2016) como un bloque de construcción fundamental en la creación de la arquitectura. ResNet (He et al., 2016) es la red neuronal más popular y ampliamente utilizada para la segmentación semántica. Es muy difícil lograr entrenar una red neuronal profunda con un gran número de capas, cuanto más aumenta la profundidad, su rendimiento se satura lo cual implica un mayor consumo computacional en la etapa de entrenamiento, He et al., (He et al., 2016) resuelve el problema del gradiente de desaparición de una manera efectiva al introducir una conexión de acceso directo de identidad (omitiendo una o más capas). Propusieron un bloque residual variante de preactivación en el que los gradientes pueden fluir fácilmente a través de la conexión de atajo sin obstrucción durante el paso de retroceso de propagación de retorno. Varias arquitecturas actuales se basan en ResNet, sus variantes e interpretaciones.

Paszke y col. (Paszke et al., 2016) presenta en 2016 un esquema de red de codificador y decodificador llamado red neuronal eficiente (ENet). Esta red es muy similar al enfoque de ResNet, creado para tareas que requieren una operación de baja latencia, como teléfonos móviles o dispositivos que funcionan con baterías. En los artículos de Deng et al., y Veit et al., (Deng et al., 2020; Veit et al., 2016) se propone una forma contraria a la intuición de entrenamiento de una red profunda mediante la eliminación aleatoria de sus capas y el uso de la red completa en el tiempo de prueba. Wu et al., (Wu

et al., 2019) presenta una red neuronal llamada ResNet-38, en la que se agrega y elimina capas en redes residuales en el momento del entrenamiento y prueba. Se analizan las profundidades efectivas de las unidades residuales y se señala que ResNet se comporta como conjuntos lineales de redes poco profundas. Pohlen et al. (2017) propone una red residual de resolución completa (FRRN) con un fuerte rendimiento de localización y reconocimiento para la segmentación semántica. FRRN exhibe las mismas propiedades de entrenamiento superiores que ResNet, con dos flujos de procesamiento, residual y agrupación. El flujo residual transporta información a la resolución de imagen completa y permite un cumplimiento preciso de los límites de los segmentos. La corriente de agrupación se somete a una secuencia de operaciones de agrupación para obtener características robustas para el reconocimiento. Los dos flujos se acoplan a la resolución de imagen completa al utilizar residuos para conseguir un fuerte reconocimiento y rendimiento de localización para la segmentación semántica. Xie et al. (2017) propone una ResNet modificada llamada ResNeXt, siguiendo la estrategia dividir, transformar y fusionar como módulos de inicio (Garcia-Perez et al., 2019; Szegedy et al., 2016), excepto que las salidas de las diferentes rutas están enlazadas y todas las rutas comparten la misma topología de red. Esto quiere decir que permite que el diseño se extienda a una gran cantidad de transformaciones. Adaptando la idea de ResNet-50 He et al. (2016); Valada et al. (2017) propone una arquitectura denominada Red Adaptativa o AdapNet.

### *2.7.4. Métodos basados en propuestas regionales.*

Los algoritmos de propuestas regionales son muy influyentes en VA sobre todo para la detección de objetos. La idea principal es poder detectar las regiones de acuerdo con la variedad de espacios de color y métricas de similitud, solo después de este paso poder realizar la clasificación (propuestas de región que pueden contener un objeto). Girshick et al. (2014) en la Universidad de Berkeley propuso una región red neuronal convolucional (R-CNN) para tareas de detección de objetos. El R-CNN consta de tres módulos; generador de propuesta regional en el que utilizaron el método de búsqueda selectiva (Uijlings et al., 2013) realizando la función de generar 20 0 0 regiones diferentes que tienen la mayor probabilidad de contener un objeto; red neuronal convolucional (Choi et al., 2005) para extraer características de cada región; finalmente, estas características de CNN se utilizan como entrada para un conjunto de SVM lineales específicas de clase. Las características también se introducen en el regresor del cuadro delimitador para obtener las coordenadas más precisas y reducir los errores de localización. A continuación, se detalla los tipos de redes neuronales que se aplica en la

presente investigación y su importancia, así como parámetros más importantes que se toma en cuenta, al conocer su arquitectura se podrá evidenciar por que se las elige el punto neurálgico de la alternativa de solución del presente trabajo de fin de master.

*2.7.5. Métodos basados en Redes Neuronales Recurrentes.*

Las redes neuronales recurrentes (RNN) se introdujeron para tratar secuencias (Graves et al., 2013; I. Goodfellow, 2016). Algunos de los logros de este tipo de redes son la escritura a mano y reconocimiento de voz, los RNN son muy exitosos en tareas de VA (manejo de imágenes). Los modelos de red que adoptan RNN en imágenes 2D (integran las capas de convolución con RNN). La red neuronal recurrente formada por bloques de memoria a corto y largo plazo (LSTM) (Aksoy et al., 2018). La capacidad de RNN para aprender las dependencias a largo plazo de los datos secuenciales y la capacidad de mantener la memoria a lo largo de la secuencia hace que sea aplicable en muchas tareas de VA, incluida la segmentación semántica (Visin et al., 2015, 2016), la segmentación y el etiquetado de escenas (L. C. Chen et al., 2016; Fan et al., 2018), basadas en el uso de la combinación RNN.

*2.7.6. Transfer Learning.*

El desarrollo del aprendizaje por transferencia (Transfer Learning), una técnica que transfiere pesos previamente entrenados de grandes conjuntos de datos públicos a aplicaciones personalizadas, y la aparición de Frameworks como TensorFlow, PyTorch, Theano, rompen las barreras entre la informática y otras ciencias (A. Voulodimos et al., 2018).

Según Voulodimos et al., (A. Voulodimos et al., 2018) una de las fortalezas de las RNC es el hecho de que pueden ser invariables a transformaciones como traslación, escala y rotación. La invariancia a la traducción, rotación y escala es uno de los activos más importantes de las RNC, especialmente en problemas de visión por computadora, como la detección de objetos, porque permite abstraer la identidad o categoría de un objeto de los detalles de la entrada visual (por ejemplo, posiciones/orientación relativas de la cámara y el objeto), lo que permite que la red reconozca eficazmente un objeto en los casos en los que los valores de píxeles reales de la imagen pueden diferir significativamente.

## 2.8. Visión por Computadora (VC).

El área de visión por computadora busca describir el mundo que vemos imágenes y reconstruir sus propiedades, tales como: forma, iluminación y distribuciones de color (Miranda, 2018). Este ámbito es utilizado posteriormente al proceso de entrenamiento de la RNC, se utiliza esta

tecnología para capturar mediante vídeo a través de un sensor (cámara web) o un archivo (vídeo pregrabado) poder analizar si el modelo de IA permite o no detectar la atención o inatención de estudiantes en un aula de clases.

Recientemente, los métodos basados en CNN han demostrado su rendimiento superior en muchas tareas de VC, incluida la depuración de imágenes (Yin et al., 2021, p. 2). Otro campo muy utilizado en VC es la tecnología en 3D, Según Li et al., (R. Li et al., 2021) el requisito de capturar características e información geométrica de nubes de puntos en 3D es cada vez más urgente en la VC. Sin embargo, comprender y analizar las nubes de puntos 3D son tareas desafiantes porque las nubes de puntos son irregulares y desordenadas.

Los componentes principales de un sistema de visión por computadora incluyen cámaras, unidades de grabación, unidades de procesamiento y modelos (G. Li et al., 2021). Estos componentes se destacan en la aplicación de tecnologías de visión por computadora especialmente con el uso de RNC. Hasta el día de hoy esta tecnología se puede aplicar a diferentes campos, en la presente investigación la Visión Artificial permite utilizar imágenes para el entrenamiento de modelos a través de RNC lo que es muy importante ya que es un punto en el que la psicología, educación y la tecnología se encuentran para dar solución a este tipo de problemas.

### *2.8.1. Aplicaciones de la Visión Artificial.*

En esta sección se describen algunas investigaciones que han utilizado métodos de aprendizaje profundo utilizados en VA, entre ellos se puede destacar el reconocimiento facial, el reconocimiento de acciones y actividades por estimación de la pose humana.

#### 2.8.1.1. Detección de Objetos.

La detección de objetos consiste en el proceso de poder detectar instancias de objetos semánticos de una determinada clase, por ejemplo, humanos, aviones, pájaros, entre otros. Esto lo hacen a través de imágenes o vídeo digitales. Las RNC son utilizadas por los Frameworks para la detección de objetos, esto incluye la creación de ventanas en secuencia clasificada (A. Voulodimos et al., 2018). Según Girshick et al. (2014) realiza una búsqueda selectiva (Uijlings et al., 2013) lo cual permite derivar propuestas de objetos, extraer características de RNC para cada propuesta y luego enviar dichas características a un clasificador SVM lo cual toma la decisión de incluir ventanas o no. Por otra parte tenemos las regiones con características de RNC propuesto por (Girshick et al., 2014), estos trabajos basados en el paradigma de Regiones con características de RNC suelen obtener excelentes

precisiones, como ejemplo tenemos (Girshick, 2015; Ren et al., 2017). Sin embargo, existen otros enfoques que no logran determinar con precisión las regiones aproximadas de los objetos (Dong et al., 2014; Hariharan et al., 2014; Hosang et al., 2014; Zhu et al., 2015).

#### 2.8.1.2. Reconocimiento Facial.

El reconocimiento facial es una de las aplicaciones de visión por computadora más populares con gran interés comercial (A. Voulodimos et al., 2018). Existe una variedad de sistemas de reconocimiento facial basados en la extracción de rasgos hechos a mano (Berg & Belhumeur, 2012; Cao et al., 2013; D Chen et al., 2013; Dong Chen et al., 2012).

#### 2.8.1.3. Reconocimiento de Actividad y acciones.

El reconocimiento de la acción y la actividad humana es un tema de investigación que ha recibido mucha atención por parte de los investigadores (A. S. Voulodimos et al., 2012, 2014). Se utilizó el aprendizaje profundo para la detección y el reconocimiento de eventos complejos en secuencias de vídeo: primero, se utilizaron mapas de saliencia para detectar y localizar eventos, y luego se aplicó el aprendizaje profundo a las características previamente entrenadas para identificar los frame más importantes que corresponden al evento subyacente. En (Kautz et al., 2017) se logra aplicar con éxito un enfoque basado en RNC para el reconocimiento de actividades en el voleibol de playa, de la misma forma el enfoque de (Karpathy et al., 2014) para clasificar eventos a partir de video a gran escala.

#### 2.8.1.4. Estimación de la pose humana.

Según Voulodimos et al., (A. Voulodimos et al., 2018) el objetivo de la estimación de la pose humana es determinar la posición de las articulaciones humanas a partir de imágenes, secuencias de imágenes, imágenes de profundidad o datos del esqueleto proporcionados por el hardware de captura de movimiento (Kitsikidis et al., 2014). La estimación de la pose humana es una tarea muy desafiante debido a la amplia gama de siluetas y apariencias humanas, la iluminación difícil y el fondo desordenado.

#### 2.8.1.5. Dataset.

La aplicación de los enfoques de aprendizaje profundo se ha evaluado en numerosos conjuntos de datos, cuyo contenido varió mucho, según el escenario de aplicación. Independientemente del caso investigado, el dominio de aplicación principal son las imágenes (naturales) (A. Voulodimos et al., 2018).

## 2.9. Herramientas utilizadas para la Arquitectura de Red Neuronal Convolucional.

A continuación, se describe una serie de herramientas utilizadas para la creación del modelo de IA creado a través de la arquitectura de RNC.

### *2.9.1. Python.*

Tomando en cuenta la documentación de Python[2], (2020) se puede definir como un lenguaje de programación muy potente y fácil de aprender puesto que la curva de aprendizaje es bastante simple debido a su sintaxis, por otra parte cuanta con un tipado dinámico, además permite el desarrollo rápido de aplicaciones en múltiples áreas, de esta forma también para un gran número de plataformas. Algunas de las ventajas que ofrece este lenguaje de programación es su extensiva librería estándar, así como el desarrollo de terceros, se encuentra disponible su código fuente como también de forma binaria para su fácil instalación en diferentes entornos de trabajo.

Su sitio web cuenta con múltiples módulos para el desarrollo de programas y algunas herramientas adicionales, sin embargo, lo que hace de Python uno de los principales lenguajes de programación utilizados en la actualidad des su documentación robusta debido a los programadores que aportan con su granito de arena para mantenerlo actualizado y a disposición de cualquier persona.

La versión de Python utilizada para la presente investigación es 3.7.7. Se eligió este lenguaje de programación por su sencillez, facilidad y potencia en el desarrollo de Deep Learning, además de ser uno de los primeros en la lista del análisis científico de datos

### *2.9.2. Anaconda.*

Según la documentación oficial de (Anaconda Documentation, 2020) es una herramienta administrador de paquetes, que además cuenta con un administrador de entornos, también es considerada una distribución de ciencia de datos para los lenguajes de programación Python y R actualmente cuenta con una colección de más de 7500 paquetes de código abierto, uno de los plus adicionales que tiene es que está completamente gratis, es muy fácil de instalar, su soporte comunitario es gratuito. Por otra parte, se considera un sistema de gestión de paquetes de código abierto, en la actualidad es un sistema de gestión del entorno que se puede instalar en los principales sistemas operativos (Microsoft Windows, GNU/Linux y Mac OS.

De la misma forma Anaconda cuenta con (Conda Documentation, 2017) un administrador de paquetes, que permite la instalación de paquetes sobre todo cuando se necesita de versiones diferentes, no es necesario desinstalar e instalar una versión diferente, con el mismo administrador es posible instalar distintas versiones del lenguaje para probar en diferente entornos. La versión de

Anaconda utilizada para la presente investigación es 1.9.12. El trabajo desarrollado con Anaconda permitió crear el entorno de trabajo, la instalación de los paquetes necesarios y desarrollar las pruebas pertinentes.

### *2.9.3. Spyder.*

Spyder[1] según su documentación oficial (Spyder 4 Documentation, 2020) es un entorno científico de escritorio en Python, fue creado para científicos, ingenieros y analistas de datos. La facilidad de poder desarrollar desde la edición, análisis, depuración y creación de perfiles hacen de Spyder una herramienta de desarrollo integral con la posibilidad de poder realizar la exploración de datos, se utiliza mucho su ejecución interactiva, como también la inspección profunda y permite tener capacidades de visualización. En la presente investigación se utilizó la versión de Spyder 4.1.4 con el objetivo de desarrollar el código fuente para la creación de la red neurona[2]l convolucional como para las pruebas necesarias a desarrollar.

### *2.9.4. Google Colaboratory.*

Según su documentación oficial (Colaboratory – Google, 2020) es llamado Colab, el mismo es un producto desarrollado por de Google Research, esta herramienta permite que cualquier persona sea capaz de escribir y ejecutar código de Python a través del navegador, es muy utilizado en proyectos de aprendizaje automático, análisis de datos y en la educación. Esta herramienta es más un servicio que se ejecuta como un notebook de Jupyter[3], no requiere ningún tipo de configuración previa para poder ser utilizado, además brinda acceso gratuito a recursos computacionales como GPU. Este servicio pertenece a Google, los equipos con los que se cuenta actualmente son de recursos limitados tanto para el entrenamiento como para el procesamiento de los datos al contar solo con CPU[4] y GPU[5], en la presente investigación se utilizó los TPU[6] para poder acelerar el proceso de entrenamiento y pruebas en la creación del modelo de la RNC.

---

[1] Su página oficial es https://www.spyder-ide.org/
[2] Su página oficial es https://colab.research.google.com
[3] Jupyter Notebook es un enfoque basado en la consola de computación, proporciona una aplicación basada en la web adecuada para capturar el proceso de computación: desarrollar, documentar y ejecutar código, así como presentar los resultados. El portátil Jupyter combina la Aplicación Web y los Documentos de cuaderno en uno solo.
[4] CPU son las siglas de Central Process Unit (Unidad Central de Procesamiento).
[5] GPU son las siglas de Graphics Processing Unit (Unidad Central de Procesamiento Gráfico), en la presente investigación se utiliza para el trabajo con la librería TensorFlow de Google.
[6] TPU son las siglas de Tensor Processing Units (Unidad de Procesamiento Tensorial). Es un circuito integrado de aplicación específica y acelerador de IA desarrollado por Google para el aprendizaje automático con redes neuronales artificiales y más específicamente optimizado para usar TensorFlow.

### *2.9.5. Kaggle.*

Kaggle[7] (Kaggle Data Science Resources, 2020), es más una comunidad que una herramienta, es una subsidiaria de Google LLC conformada por científicos de datos y profesionales en constante aprendizaje, se utiliza dentro del campo de aprendizaje automático. Esta comunicad permite a los usuarios controlar y publicar conjuntos de datos, así como poder explorar y construir modelos en un entorno de ciencia de datos basado directamente desde la web, por otra parte, permite trabajar con otros científicos de datos e ingenieros, recrear entornos previamente desarrollados por miembros de la comunicad y poder y participar en concursos para resolver desafíos de ciencias de datos. En la presente investigación fue utilizado para obtener varios Dataset con imágenes que tienen importancia relevante en la presente investigación.

### *2.9.6. TensorFlow.*

TensorFlow[8] (TensorFlow Aprendizaje Automático, 2020) es la librería principal de Google, está orientada al desarrollo de algoritmos de aprendizaje automático y análisis de datos, en esta herramienta es muy fácil la creación de modelos de aprendizaje automático para computadoras de escritorio, así como dispositivos móviles, puede ser ejecutado en la web y en la nube, está orientado para para principiantes como personas experimentadas.

### *2.9.7. Keras.*

Keras[9] (Keras, 2020) es más bien una API de aprendizaje profundo, la misma fue escrita en Python, se puede ejecutar sobre la plataforma de aprendizaje automático TensorFlow. Este API fue desarrollado con un enfoque que permite una experimentación rápida, es posible obtener resultados muy rápidos y eficientes, como su lema lo dice “Pasar de la idea al resultado lo más rápido posible, lo cual es la clave de una buena investigación”. En la presente investigación permitió en poco tiempo crear la arquitectura de RNC para generar el modelo de IA, así mismo poder realizar la configuración de parámetros, preprocesamiento de los datos, etapa de entrenamiento, evaluación y almacenamiento.

### *2.9.8. OpenCV.*

Open Source Computer Visión Library (OpenCV, 2020) con su abreviatura OpenCV[10] es una biblioteca de software utilizada para la creación de aplicaciones utilizando tecnología de VA y

[7] Su página oficial es https://www.kaggle.com
[8] Su página oficial es https://www.tensorflow.org/
[9] Su página oficial es https://keras.io
[10] Su página oficial es https://opencv.org

aprendizaje automático, cabe recalcar que esta biblioteca es de código abierto. En la presente investigación se utilizó como una de las librerías principales para la generación del script de VA.

## Capitulo tres.

# 3. Propuesta.

La presente propuesta de investigación realiza la explicación detallada que se utiliza en cada una de las etapas necesarias para generar la arquitectura de RNC, de la misma forma el script de VA que permiten predecir la atención.

## 3.1. Generación de la arquitectura de la Red Neuronal Convolucional.

El presente proceso tiene diferentes etapas que se debe tomar en cuenta, las mismas que inician con la búsqueda, selección, descarga, clasificación, categorización de la data, hasta la generación de la arquitectura de red neuronal, La misma cuenta con diferentes capas ocultas que harán posible la creación del modelo de IA que posteriormente a través de técnicas de VA permitirán la predicción de la atención de una o varias personas en un aula de clases.

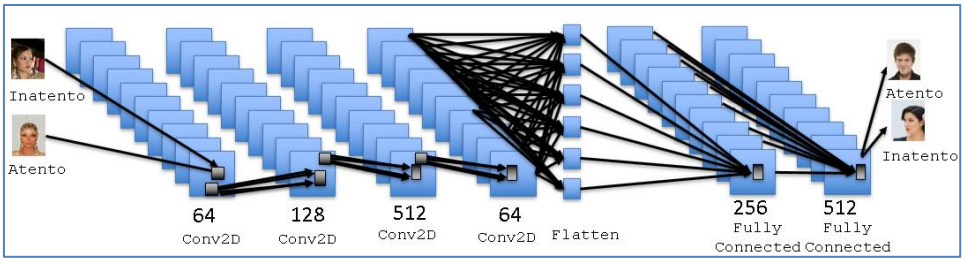
El código generado de la presente propuesta se encuentra disponible en línea en la siguiente dirección https://github.com/5t4t1ck/attentionFinal

## 3.2. Modelo para detectar la atención.

En la Figura 3 se puede evidenciar cual es la arquitectura propuesta para la creación de la RNC utilizada en la presente investigación que permite detectar la atención o inatención a través de técnicas de VA.

**Figura 3**

*Creación de la arquitectura de Red Neuronal Convolucional.*



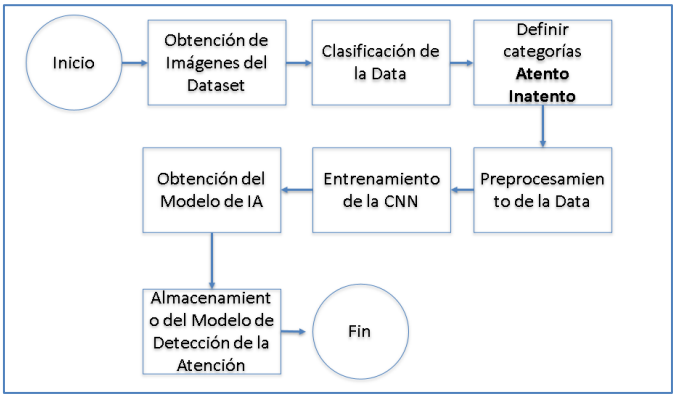

En la Figura 4 se puede observar cuáles fueron las etapas necesarias que se utiliza para para la crear la arquitectura de la Red Neuronal Convolucional. Según Yann Lecun et al, (2015) las redes neuronales convolucionales profundas han producido grandes avances en el aprendizaje para el procesamiento de imágenes, vídeo, voz y audio, en conclusión este tipo de red tiene como propósito la creación de una Inteligencia Artificial que pueda aprender a predecir la atención.

En la Figura 5 se puede apreciar cómo se describe el funcionamiento del algoritmo propuesto el cual cubre la obtención de imágenes del dataset, clasifica la data (imágenes), define las categorías

Atento e Inatento, realiza el preprocesamiento de la data (modifica tamaño, dirección, acercamiento, entre otras), realiza el entrenamiento de la Red Neuronal Convolucional, se obtiene el modelo de Inteligencia Artificial, se almacena el modelo de Inteligencia Artificial y finaliza, cabe recalcar que posteriormente es evaluado para determinar la confiabilidad del aprendizaje de la misma.

**Figura 4**

*Algoritmo para la creación de la RNC.*

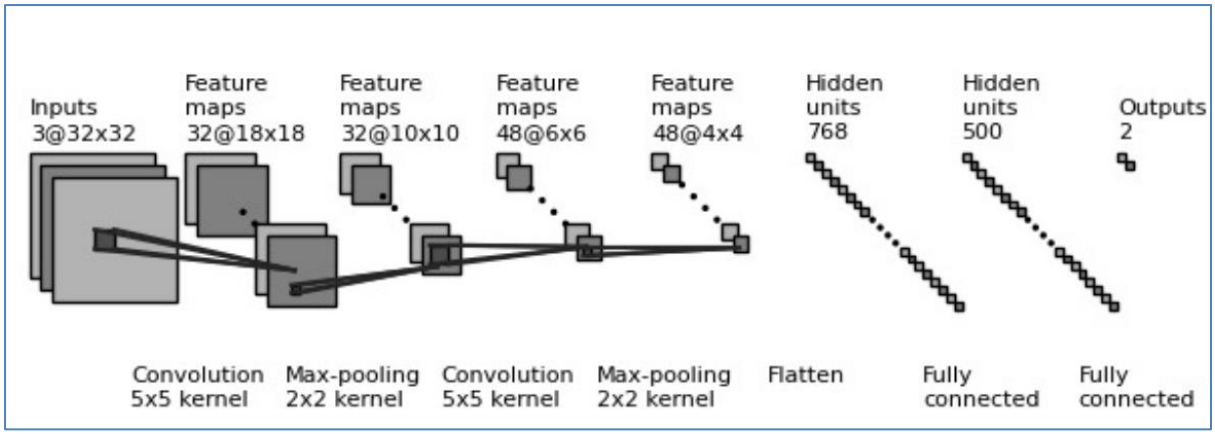



En las Figuras 5 se puede apreciar la arquitectura propuesta en esta investigación y las interacciones en las diferentes capas ocultas generadas utilizando el estilo Alexnet y FCNN.

Según Progrmador Clic. (2019) en los últimos años las Redes Neuronales Convolucionales han podido lograr la creación de aplicaciones muy exitosas en el reconocimiento de imágenes, las mismas que se han permitido convertir en un punto muy importante del aprendizaje profundo. Desde su creación han existido múltiples variantes que han surgido desde los modelos clásicos (LeNet, Alexnet, Googlenet, VGG, DRL, entre otros), de la misma forma en la Figura 6 se puede observar la arquitectura de la Red Neuronal Convolucional para la detección de atención propuesta con un estilo FCNN.

**Figura 5**

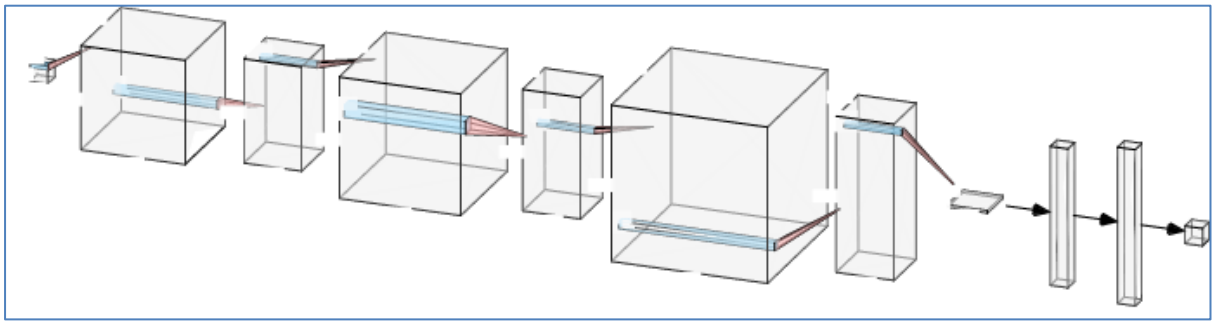
*Arquitectura de la RNC para la detección de atención.*




En la Figura 6 también se puede apreciar la arquitectura de red neuronal convolucional para la detección de atención con un estilo AlexNet, la vista es diferente a la Figura 6, sin embargo, las capas internas que constan son las mismas.

**Figura 6**

*Arquitectura de la RNC para la detección de atención estilo AlexNet.*



*3.2.1. Identificar la data a utilizar.*

Para poder crear la arquitectura de Red Neuronal Convolucional propuesta es necesario conocer como aprenden las redes neuronales convolucionales, según Imma Grau, (2018) estas redes son un tipo de red neuronal artificial donde las neuronas pueden responder a campos repetitivos de forma muy similar al funcionamiento de la corteza visual, son muy efectivas en la clasificación y segmentación de imágenes entre otras cosas. Después de analizar las ventajas que ofrece este tipo de tecnología en la presente investigación se investiga de donde se puede utilizar dataset de información que permita entrenar un modelo de Red Neuronal Convolucional.

Según Madsen et al, (2021) la inatención es un problema latente en las aulas de clase, y se evidencia cuando los estudiantes no pueden comprender el material didáctico relevante propuesto por los docentes, y una de las causas es no prestar atención, lo cual repercute en un bajo rendimiento en sus calificaciones. Las imágenes seleccionadas para el aprendizaje del modelo de la Red Neuronal

Convolucional propuesta utiliza como punto de partida el seguimiento ocular, la posición del rostro y la atención.

De acuerdo a XinZhe Jin, (2020) la posición inicial de la cara, el grado de apertura de la boca y de los ojos presenta algunos inconvenientes al momento de realizar la detección facial, sin embargo, esto no ha podido impedir que los avances tecnológicos a lo largo de los años puedan proponer diferentes métodos para identificar la superficie facial. En la presente investigación se utiliza el reconocimiento facial para la detección de atención, citando a Wang et al, (2015) se presenta una investigación donde ha podido ser entrenado una red que tiene una precisión del 90% en un conjunto de datos de gran tamaño, en la presente investigación se analiza la posición de la cara para determinar si una persona presta atención.

Investigaciones como la de Saeed et al, (2015) analizan la pose y estimación de la postura de la cabeza de la misma forma el análisis de rostro humano es muy utilizado en sistemas de Visión Artificial para reconocer expresiones faciales, gestos con la cabeza, entre otros. En la presente investigación se analiza la postura de la cabeza para determinar si una persona está poniendo atención.

De la misma forma XinZhe Jin, (2020) citando a Viola & Jones, (2001) señala que para poder realizar el reconocimiento facial se utiliza la librería OpenCV, la misma cuenta con la función de recorte de fotogramas, utiliza un método muy conocido como *Harr Cascade*, cuyo propósito inicial era la detección de objetos, sin embargo debido a su gran efectividad pudo expandir su uso para más campos, y uno de ellos es el reconocimiento facial y ocular. En la presente investigación se analiza la posición de los ojos para determinar si una persona está poniendo atención.

Lo más importante en esta sección es poder identificar la fijación ocular (hacia donde está mirando la persona) y la posición del rostro y la postura de la cabeza, por ejemplo podemos encontrar a una persona mirando hacia el lado izquierdo con la posición del rostro hacia el frente, o viceversa, la posición del rostro hacia el lado izquierdo y la fijación ocular dirigida hacia el centro, este tipo de variantes son los que posteriormente sirven para detecta si una persona está prestando atención o está siendo objeto de inatención por distintos factores (no le llama la atención el suceso, fenómeno o evento que se produce).

En la literatura revisada se encontró *Kaggle Data Science Resources*, (2020) una comunidad de inteligencia artificial en donde sus miembros publican dataset de información para ser utilizados por

cualquier miembro de su comunicad, es así que se obtiene algunos dataset para poder ser descargados, analizados y posteriormente propone un dataset de información que permita obtener las imágenes necesarias para posteriormente ser utilizadas en la arquitectura de red neuronal convolucional propuesta.

*3.2.2. Identificar fuentes para la obtención de la Data.*

Para la búsqueda de imágenes que permitan continuar con la etapa de entrenamiento y validación de la RNC se analizaron diferentes alternativas, empezando por la búsqueda de imagen por imagen, la descarga de dataset de diferentes repositorios, la aplicación de técnicas como web scraping[11], sin embargo al final se tomó la decisión de utilizar los dataset de *Kaggle Data Science Resources*, (2020) como se describe en la Tabla 1.

**Tabla 1**

*Etapas de creación del Dataset.*

| Pasos | Descripción |
|---|---|
| **Búsqueda de Dataset** | Se empieza a buscar dataset de con distintos tipos de imágenes que permitan ser discriminados y seleccionar las imágenes que sirvan para los propósitos de la presente investigación, para ello fue fundamental *Kaggle Data Science Resources*, (2020) debido a que la mayor parte de dataset se obtuvo de esta gran comunidad. |
| **Selección de imágenes** | Se analizó cada uno de los dataset obtenidos, discriminando cada imagen de acuerdo con las categorías de atento e inatento. |
| **Organización de los Dataset** | Se organizó las imágenes en distintos directorios para poder tener una distribución heterogénea de imágenes. |

Para la construcción del dataset de imágenes se tomó en consideración los criterios de la posición de la cabeza (Saeed et al., 2015), la posición de la cara (Wang et al., 2015) y el seguimiento ocular (Viola & Jones, 2001), de la misma forma Lemonnier et al, (2020) en su investigación destaca la atención visual en un movimiento que describe una trayectoria de arriba hacia abajo, la misma que es impulsada por datos, los cuales dependen de las características de la escena visual que se puede

[11] Web Scraping: Web scraping o raspado web, es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.

percibir. Cabe mencionar que se incluyó los efectos de la atención tomados en cuenta por Madsen et al, (2021) obteniendo la correlación entre los participantes y sus movimientos oculares motivados por la atención.

Finalmente se definió en base a los criterios expuestos que la mejor solución para las posteriores etapas de la creación de la arquitectura de la red neuronal convolucional, la generación del dataset debía contener 2 categorías, las mismas que son fundamentales para determinar la atención en la inteligencia artificial. Estas son Atento e Inatento, sin embargo, se hace indispensable conocer cómo se seleccionan las imágenes pertenecientes a cada categoría. En la siguiente sección se establece una serie de criterios de selección que permitirán organizar el dataset de imágenes.

### *3.2.3. Generación de Criterios de Selección.*

La generación de criterios de selección es una de las etapas más importantes que diferencia esta investigación de otras relacionadas, principalmente por la creación de las categorías que permitirán las posteriores etapas de entrenamiento y validación de la Red Neuronal Convolucional. Para poder seleccionar las imágenes que permitan predecir la atención se propone dos categorías “atención” e “inatención”, a continuación, se describe cada uno de los criterios tomados en cuenta para la selección y organización de estas dos categorías en la Tabla 2.

**Tabla 2**

*Criterios tomados en cuenta para la generación de las categorías.*

| Atención | Imagen ejemplo | Inatención | Imagen ejemplo |
|---|---|---|---|
| Imágenes con el rostro en dirección al frente y la vista al frente |  | Imágenes con el rostro a diferente al frente y la vista diferente al frente |  |
| Imágenes con el rostro en dirección al lado derecho y la vista al frente |  | Imágenes con el rostro en dirección a un lado derecho y la vista diferente al frente |  |
| Imágenes con el rostro en dirección al lado izquierdo y la vista al frente |  | Imágenes con el rostro en dirección a un lado izquierdo y la vista diferente al frente |  |

| | | | |
|---|---|---|---|
| Imágenes con el rostro en dirección al lado hacia arriba y la vista al frente |  | Imágenes con el rostro en dirección a un lado hacia arriba y la vista diferente al frente |  |
| Imágenes con el rostro en dirección al lado hacia abajo y la vista al frente |  | Imágenes con el rostro en dirección a un lado hacia abajo y la vista diferente al frente |  |
| Imágenes con el rostro en dirección al lado derecho/arriba y la vista al frente |  | Imágenes con el rostro en dirección a un lado derecho/arriba y la vista diferente al frente |  |
| Imágenes con el rostro en dirección al lado izquierdo/arriba y la vista al frente |  | Imágenes con el rostro en dirección a un lado izquierdo/arriba y la vista diferente al frente |  |
| Imágenes con el rostro en dirección al lado hacia abajo/derecha y la vista al frente |  | Imágenes con el rostro en dirección a un lado hacia abajo/derecha y la vista diferente al frente |  |
| Imágenes con el rostro en dirección al lado hacia abajo/izquierda y la vista al frente |  | Imágenes con el rostro en dirección a un lado hacia abajo/izquierda y la vista diferente al frente |  |

Imágenes tomadas del perfil de Drobotdean, (2019) en la plataforma Freepik.

*3.2.4. Clasificación de la Data.*

Se realizó un proceso manual de clasificación de cada imagen tomando en cuenta los criterios de selección descritos en la Tabla 2.

Se seleccionó 2002 imágenes para la categoría atento y 2001 para la categoría inatento. El tamaño del dataset fue variando donde se pudo determinar que debía estar balanceado, de acuerdo a Pulgar et al, (n.d.) el objetivo principal de poder balancear los datos es obtener un modelo que permita

clasificar correctamente nuevos ejemplos. Se realizaron algunas pruebas que permitieron determinar el número adecuado de imágenes para el aprendizaje, una limitante en esta etapa fue el hardware del equipo de pruebas con el que se contaba y los tiempos límites de Colab.

*3.2.5. Tratamiento de la Data.*

Para poder crear el script de entrenamiento y test se utilizó Anaconda, Spyder y las librerías de TensorFlow y Keras entre otras, sin embargo, el tiempo de entrenamiento que se produce en CPU y GPU además de ralentizar el equipo de pruebas, un Intel® Core™ i7-8700 CPU @3.20GHz 3.19 GHz con 8,00 Gb (721 GB utilizable), Procesador de 64 bits con Tarjeta Gráfica GeForce GT 710 y sistema operativo Windows 10 Pro N. Provocaba constantes reinicios por sobrecargar los recursos del equipo, se optó por migrar a Google Colab, aunque los recursos utilizados eran mejores que los del equipo físico que se disponía para este tipo de entrenamiento y testeo existían limitaciones de entrenamiento, como ventaja reduce el tiempo de entrenamiento de un entorno local, pero como desventaja el límite de entrenamiento que produce Google Colab puede significar un inconveniente en los resultados que se intenta obtener. Después de realizar las pruebas necesarias en Colab, se determinó que finalmente se realizarían cada una de las etapas de la arquitectura en el equipo físico.

Se importar las librerías necesarias para el entrenamiento. Cabe mencionar que las librerías finales que se utiliza en esta etapa fueron seleccionadas después de analizar la necesidad de cada etapa de creación, preentrenamiento, entrenamiento, compilación, evaluación y almacenamiento de la RNC.

En la Figura 7 se puede observar cómo se importa las librerías necesarias para la creación del modelo de atención a través de redes neuronales convolucionales.

**Figura 7**

*Importación de las librerias necesarias para la creación de la arqutectura de la RNC.*

```
#import tensorflow
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense
from keras.layers import Dropout,Input
from keras.layers import Flatten
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import BatchNormalization
from keras.layers import Activation
from keras.models import Model,Sequential
from keras.optimizers import Adam
from keras.preprocessing.image import load_img,img_to_array
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator
from sklearn import metrics
import matplotlib.pyplot as plt
import os
import zipfile
from google.colab import drive
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
```

En la Figura 8 se puede observar el almacenamiento en las variables train_Atento y train_Desatento de ambos directorios para las etapas posteriores. De la misma forma se lo realiza con las etapas de test, definiendo ambas categorías para que el modelo de IA no solo aprenda a reconocer cuando una persona está atento o inatento, sino también para validar dicho modelo y las etapas posteriores no presenten mayores inconvenientes.

**Figura 8**

*Se accede a las 2 categorías en las variables train_Atento y train_Desatento.*

```
base_dir = '/tmp/attention/data/'
train_dir = os.path.join(base_dir)

train_Atento = os.path.join('/tmp/attention/data/atento')
train_Desatento = os.path.join('/tmp/attention/data/desatento')
```

En la Figura 9 se puede observar la verificación de la cantidad de imágenes cargadas en ambas categorías, en esta sección se cargaron 2002 imágenes en Atento y 2001 en Desatento para las pruebas pertinentes como se menciona en la subsección 3.2.4.

**Figura 9**

*Se verifica el número de imágenes de acuerdo con cada categoría.*

```
print('total training Atento images :', len(os.listdir(train_Atento)))
print('total training Desatento images :', len(os.listdir(train_Desatento)))
```

```
total training Atento images : 2002
total training Desatento images : 2001
```

En la Figura 10 se observa las distintas configuraciones y transformaciones que se producen en el Dataset, al finalizar se obtienen 4003 imágenes en 2 clases que son las categorías atento e inatento.

**Figura 10**

*Se realiza el procesamiento del Dataset.*

```
# number of images to feed into the NN for every batch
batch_size = 256

datagen_train = ImageDataGenerator()
#datagen_validation = ImageDataGenerator()
train_generator = datagen_train.flow_from_directory(base_dir,
                                                    target_size=(pic_size,pic_size),
                                                    color_mode="grayscale",
                                                    batch_size=batch_size,
                                                    class_mode='categorical',
                                                    shuffle=True)
```

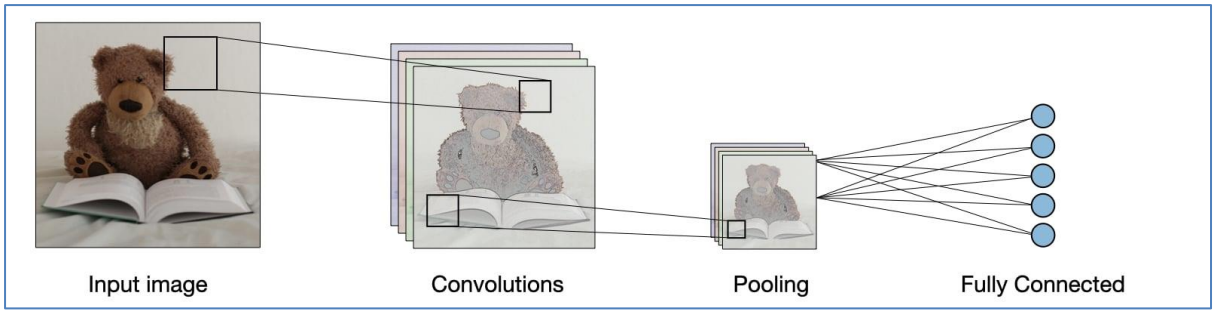
### *3.2.6. Creación de la arquitectura de la RNC.*

Cada una de las etapas anteriores preparan la información (dataset de imágenes) de la cual será alimentada la red neuronal convolucional para su etapa de entrenamiento y validación, “Una arquitectura de Red Neuronal Convolucional es como un conjunto de capas de procesamiento, de modo que puede verse y entenderse como un diagrama de bloques secuencial”. (Maeda Gutierrez, 2019) citando a (Vedaldi et al., 2015).

En la imagen 11 se puede destacar las etapas más importantes de una red neuronal convolucional, estas son la imagen de entrada, las capas de convolución, las capas de Pooling y la capa Fully Connected.

**Figura 11**

*Arquitectura de una Red Neuronal Convolucional tradicional*

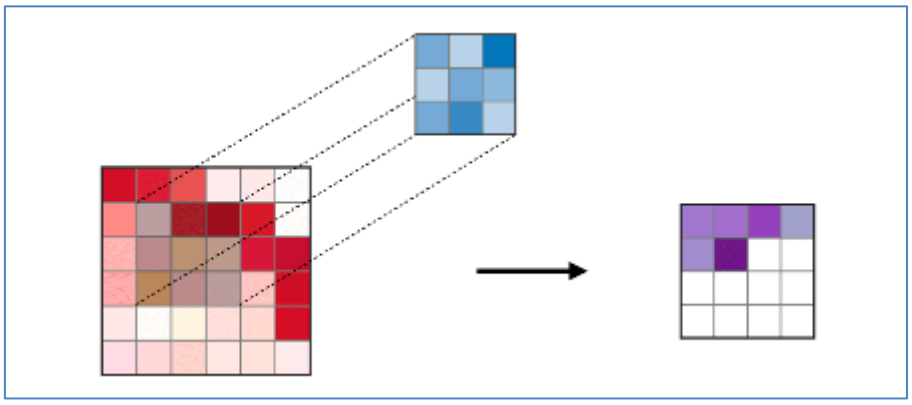



Tomado de la (Shervine A, 2019) en la Universidad de Stanford.

La Capa de Convolución “La capa de convolución (CONV) utiliza filtros que realizan operaciones de convolución mientras escanea la entrada con respecto a sus dimensiones. Sus hiperparámetros incluyen el tamaño de filtro FFF y stride SSS. La salida resultante OOO se denomina mapa de características o mapa de activación” como se puede observar en la Figura 12. (Shervine A, 2019).

**Figura 12**

*Capa de Convolución.*



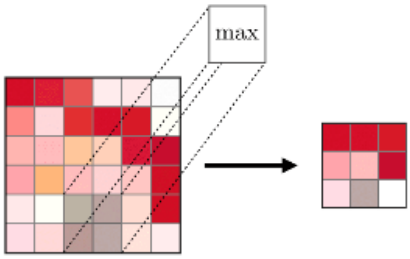
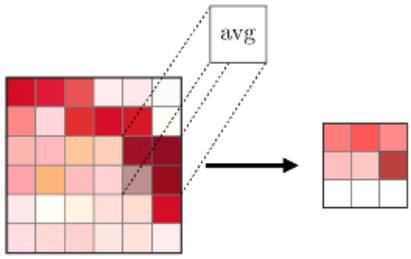
Shervine A, 2019 en la Universidad de Stanford.

La capa de Pooling (POOL) “Es una operación de submuestreo, que normalmente se aplica después de una capa de convolución, que realiza cierta invariancia espacial. En particular, la agrupación máxima y media son tipos especiales de agrupación en los que se toman el valor máximo y medio, respectivamente”. Como se puede observar en la Tabla 3. Se hace la comparación de 2 operaciones que se utilizan en la creación de redes neuronales convolucionales, cabe recalcar que en la presente arquitectura se utilizó Max Pooling. (Shervine A, 2019).

**Tabla 3**

*Capa Max Pooling.*

| **Tipo** | Max Pooling | Averange Pooling |
|---|---|---|

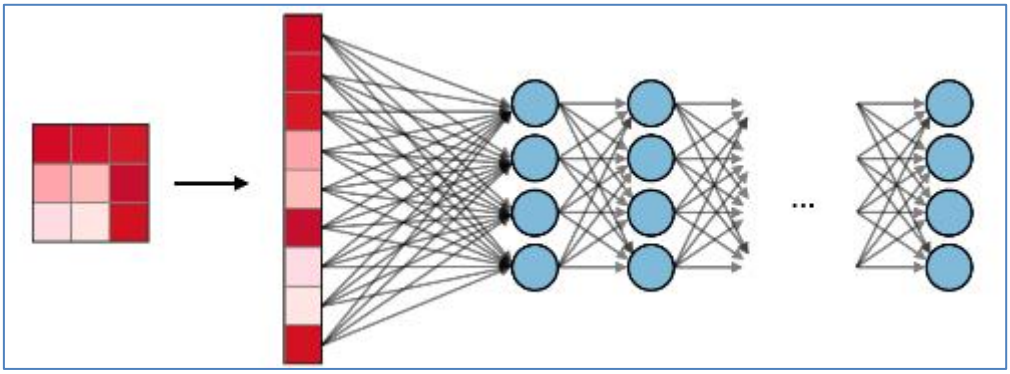
| | | |
|---|---|---|
| **Propósito** | Cada operación de agrupación selecciona el valor máximo de la vista actual | Cada operación de agrupación promedia los valores de la vista actual |
| **Ilustración** |  |  |
| **Comentarios** | • Conserva las características detectadas.<br>• Más comúnmente utilizado. | • Mapa de funciones de muestras reducidas.<br>• Utilizando en LeNet |

Tomado de la Shervine A, 2019 en la Universidad de Stanford.

La capa completamente conectada (FC) “Opera en una entrada plana donde cada entrada está conectada a todas las neuronas. Si están presentes, las capas FC se encuentran generalmente hacia el final de las arquitecturas CNN y se pueden usar para optimizar objetivos tales como puntajes de clase”. Como se puede observar en la Figura 13. (Shervine A, 2019). Finalmente se procede a la creación de la arquitectura de la Red Neuronal Convolucional en la cual se describe cada una de las secciones.

**Figura 13**

*Capa Completamente Conectada.*



Tomado de la (Shervine A, 2019) en la Universidad de Stanford.

3.2.6.1. Primera Sección de la Red Neuronal Convolucional.

La presente Red Neuronal Convolucional como se muestra en la Figura 16, inicia con

Secuential ya que es apropiado para una pila simple de capas donde cada capa tiene exactamente un tensor de entrada y un tensor de salida. Continúa agregando la primera capa convolucional con 2 entradas de las categorías atento e inatención, genera 64 capas ocultas, un kernel

de 3x3 (especifica la altura y el ancho de las Convolution2D con lo que se va realizando la convolución de cada imagen y se repite en las 64 capas ocultas) y un padding “same” (es decir igual da como resultado un relleno uniforme hacia la izquierda/derecha o hacia arriba/debajo de la entrada, de modo que la salida tiene la misma dimensión de altura y ancho que la entrada), con un imput_shape de 48x48x1 lo cual indica que tiene imágenes de 48x48 pixeles 1 solo canal (gris).

**Figura 14**

*Primera Capa de la Arquitectura de la RNC.*

```
#1
model.add(Conv2D(64,(3,3), padding = 'same', input_shape = (48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))
```

Se agrega una capa de Badch Normalization la misma como su nombre lo indica busca la normalización de las entradas, dicha normalización por lotes aplica una transformación que mantiene la salida media cercana a 0 o la desviación estándar de salida a 1, es importante destacar que la normalización por lotes funciona de manera diferente durante el entrenamiento y durante la inferencia.

Se agrega una capa de activación en este caso Relu, lo cual permite generar las activaciones y estas se puedan utilizar a través de una capa de activación o mediante el argumento de activación admitido por todas las capas de avance.

Se agrega una capa de MaxPooling2D con un pool_size de 2 x 2, esta es una operación de agrupación máxima para los datos en 2 Dimensiones, permite reducir la representación de entrada tomando el valor máximo sobre la ventana definida por pool_size para cada dimensión a lo largo del eje de características. La ventana se desplazca por pasos en cada dimensión. La salida resultante cuando se usa la operación de relleno “válida” tiene una forma (de acuerdo al número de filas y columnas) de output_shape = (input_shape – pool_size +1)/strides) es decir es un kernel (segmento de la matriz de la imagen) que va pasando por toda la imagen recuperando las características más sobresalientes de la misma para poder minimizar la imagen al máximo y que la capacidad computacional para dicho análisis de características no requiera muchos recursos y el procesamiento sea válido de acuerdo a la investigación que se necesita, En este caso los c criterios tomados en cuenta para determinar el entrenamiento en la Tabla 2.

Se agrega una capa de Dropout, de acuerdo a Xia et al (2021) la técnica de Dropout fue propuesta por primera vez por Hinton et al. (2012), para poder mejorar el problema del sobreajuste (sobre entrenamiento), de las redes neuronales. Tomando el parámetro 0.25 consiste en el abandono o desconexión de neuronas en las redes neuronales lo que permite mejorar el aprendizaje de la red al desconectar ciertas neuronas en este caso el 25% del total de las neuronas con el aprendizaje de características de las 2 categorías señaladas.

3.2.6.2. Segunda Sección de la Red Neuronal Convolucional.

La segunda sección no varía mucho de la primera, se agregan las mismas capas, excepto que en la capa Conv2D los parámetros cambian a 128 capas ocultas, lo demás se mantiene igual con las mismas capas (Batch Normalization, Activation Relu, MaxPolling2D y Droput). Este aumento de capas ocultas permite generar un entrenamiento más optimo, ya que toma en cuenta el doble de características necesarias para que el entrenamiento de la RNC pueda ser validado de mejor forma, según las investigaciones similares en el desarrollo de modelos de IA está es la sugerencia más acertada.

3.2.6.3. Tercera Sección de la Red Neuronal Convolucional.

En la tercera sección es similar a la segunda, con la variante de la capa Conv2D donde los parámetros cambian a 512 capas ocultas, al igual que en la primera sección las demás capas se mantienen intactas (Batch Normalization, Activation Relu, MaxPolling2D y Droput).

3.2.6.4. Cuarta Sección de la Red Neuronal Convolucional.

En la cuarta sección es similar a la segunda y tercera, con la variante de la capa Conv2D donde los parámetros cambian a 64 capas ocultas (como en la primera sección), las demás capas se mantienen intactas (Batch Normalization, Activation Relu, MaxPolling2D y Droput).

La arquitectura de la RNC agrega una capa Flatten, de acuerdo a AbuRass et al. (2020) la capa de salida en un solo vector de valores, se aplana cada uno de estos valores representa una probabilidad de que determinada característica pertenezca a una etiqueta. En la presente propuesta toma características propias de cada imagen como la posición de los ojos y la dirección de la mirada, la nariz, la boca, entre otras características que el modelo ha podido considerar pertinentes para poder catalogar la probabilidad de que el rostro detectado pueda pertenecer una etiqueta (categoría) sea está atento o inatento según corresponda en las etapas de entrenamiento y validación que se desarrollarán

posteriormente, la organización de estas capas determina la confiabilidad con valores entre 0 y 1 (p. 8).

3.2.6.5. Quinta Sección de la Red Neuronal Convolucional.

La siguiente sección se refiere a una Capa Full Connected, esto significa que todas las neuronas de la última salida (la capa anterior) se conectan con todas, existe una capa Dense con 256 neuronas, una capa de BatchNormalization, una capa de activación Relu y un Droput de 0.25.

En la siguiente sección se agrega una segunda Capa Full Connected, muy similar a la anterior con la única variante que en la capa Dense cambia el parámetro a 512 capas ocultas. Lo demás se mantiene tal y como está en la sección anterior (Capa Full Connected Layer 1), se agrega una capa Dense con 256 neuronas, una capa de Batch Normalization, una capa de activación Relu y un Droput de 0.25, hasta este punto se conoce como las capas ocultas.

En la siguiente sección se utiliza la capa de activación softmax, según Asriny et al. (2020) se aplica en la clasificación para proporcionar resultados más intuitivos y que puedan ser fáciles de clasificar de acuerdo a interpretaciones probabilísticas para todas las etiquetas producidas. En la presente investigación permite obtener solo 2 probabilidades, que el rostro detectado sea atento o inatento (p. 4).

3.2.6.6. Optimizador Adam.

En la siguiente sección se utiliza el optimizador Adam, citando a Andika et al. (2020) la optimización de Adam se utiliza para mejorar la precisión del aprendizaje de los modelos que se ha creado, es un algoritmo de optimización de aprendizaje adaptativo (p. 5).

Otro parámetro que se hace presente es Learning Rate (lr) según Melinte et al. (2020) es la tasa de aprendizaje, un parámetro clave y para determinar el valor ideal de aprendizaje de la red neuronal convolucional, la misma tiene estrecha relación con el número de épocas utilizadas, en la presente investigación después de realizar varias pruebas se determinó para la taza de aprendizaje es 0.0004, la misma presenta mejores resultados en el entrenamiento (p. 4).

Definidos cada uno de los parámetros descritos en la presente arquitectura, se procede a la compilación del modelo de Inteligencia Artificial.

*3.2.7. Compilación del modelo.*

Utilizando la arquitectura de red neuronal convolucional propuesta en el apartado 3.2.6 el siguiente paso es compilar el modelo, previa al entrenamiento y validación obteniendo las métricas que permitan evaluar el aprendizaje de la inteligencia artificial creada.

Utilizando la documentación oficial de *Keras.* (2020), se procede a explicar cada una de las métricas utilizadas para evaluar el modelo.

3.2.7.1. Perdida.

Uno de los parámetros que determina la función de pérdidas esto permite conocer el grado de error entre salidas calculadas y las salidas de los datos de entrenamiento.

El optimizador utilizado en nuestra investigación es Adam como se lo revisó en la sección anterior. La pérdida se la calcula utilizando Categorical Crosentropy que significa Calculo de pérdida de entropía cruzada entre las etiquetas y las predicciones. Esta función se utiliza cuando hay 2 o más clases de etiquetas.

3.2.7.2. Metrics.

Las métricas utilizadas en la presente red neuronal convolucional tienen algunos parámetros que se procede a describir a continuación.

Accuracy (aprendizaje) se calcula la frecuencia con la que las predicciones son iguales a las etiquetas. Esta métrica crea dos variables locales, total y count, que se utilizan para calcular la frecuencia con la modelo puede predecir los aciertos. Esta frecuencia suele devolverse finalmente como precisión binaria (0, 1).

AUC calcula el área bajo la curva aproximadamente una suma de Riemann (Un tipo de aproximación del valor de una integral mediante una suma finita, se llama así en honor al matemático alemán del siglo XIX Bernhard Riemann). Esta métrica crea cuatro variables locales, true_positives, true_negatives, falese_positives y fales_negatives. Esto significa que la curva AUC, se utiliza un conjunto de umbrales espaciados linealmente para calcular pares de valores de recuperación y la precisión, esto quiere decir que el área bajo la curva ROC se calcula utilizando la altura de los valores de recuperación por la tasa de falsos positivos. Mientras que el área bajo la curva PR se calcula utilizando la altura de los valores de precisión mediante la recuperación. Este valor se devuelve finalmente mediante AUC, una operación que calcula el área bajo una curva de precisión versus valores de recuperación.

*3.2.8. Entrenamiento de la Red Neuronal Convolucional.*

El entrenamiento de la RNC es un punto muy importante de esta propuesta, ya que permite generar el entrenamiento y la etapa de validación del modelo de inteligencia artificial.

Después de definir los diferentes parámetros necesarios en la red neuronal (Número de Capas, métodos de activación, métodos de desconexión de las redes para evitar Overfitting) podemos compilar el modelo de IA.

Si todo funciona correctamente se podrá obtener el modelo de IA, cada uno de los parámetros configurados fueron mejorados después de una serie de pruebas donde fue necesario aumentar o disminuir parámetros en las etapas anteriores, de la misma forma en las capas ocultas y los parámetros de configuración, aumentando o disminuyendo el número de capas para poder obtener los mejores resultados.

Como se puede observar en la Figura 15 se realiza el entrenamiento de la IA, fue necesario modificar más de una ocasión las configuraciones definidas en las etapas anteriores para poder evitar el sobreajuste (sobre entrenamiento), obteniendo un modelo con un aprendizaje superior al 90% y una perdida menor al 13%.

**Figura 15**

*Entrenamiento de la RNC.*

```
7/7 [==============================] - 125s 18s/step - loss: 0.1688 - accuracy: 0.9280
Epoch 120/130
7/7 [==============================] - 129s 18s/step - loss: 0.1513 - accuracy: 0.9416
Epoch 121/130
7/7 [==============================] - 125s 18s/step - loss: 0.1391 - accuracy: 0.9453
Epoch 122/130
7/7 [==============================] - 124s 18s/step - loss: 0.1520 - accuracy: 0.9351
Epoch 123/130
7/7 [==============================] - 128s 18s/step - loss: 0.1367 - accuracy: 0.9469
Epoch 124/130
7/7 [==============================] - 125s 18s/step - loss: 0.1335 - accuracy: 0.9491
Epoch 125/130
7/7 [==============================] - 124s 18s/step - loss: 0.1230 - accuracy: 0.9525
Epoch 126/130
7/7 [==============================] - 125s 18s/step - loss: 0.1250 - accuracy: 0.9456
Epoch 127/130
7/7 [==============================] - 125s 18s/step - loss: 0.1381 - accuracy: 0.9450
Epoch 128/130
7/7 [==============================] - 124s 18s/step - loss: 0.1322 - accuracy: 0.9491
Epoch 129/130
7/7 [==============================] - 125s 18s/step - loss: 0.1355 - accuracy: 0.9445
Epoch 130/130
7/7 [==============================] - 125s 18s/step - loss: 0.1285 - accuracy: 0.9485
```

### *3.2.9. Evaluación del entrenamiento y compilación del modelo de la IA.*

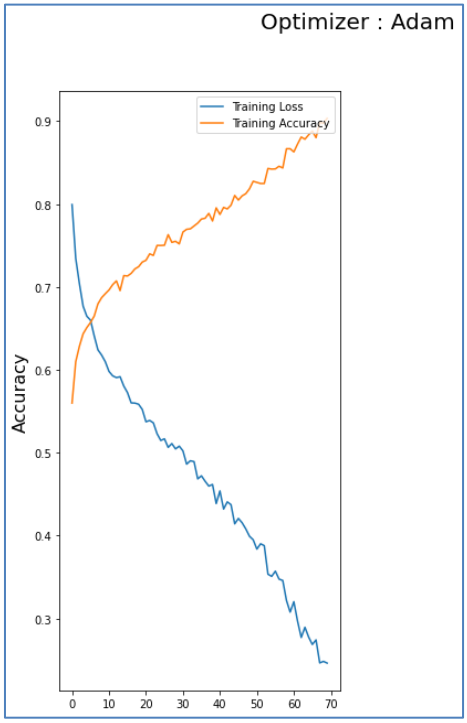
En esta etapa se procede a analizar los resultados de la etapa de entrenamiento y compilación del modelo de Inteligencia Artificial creado, tanto para el aprendizaje como para la perdida en las etapas

de entrenamiento y validación que genera la RNC, cuando el aprendizaje alcanza un aprendizaje superior al 90% y la perdida desciende al 3% se puede considerar que se ha tenido éxito. Como se puede observar el aprendizaje (Accuracy) sobrepasa el 0.96%, mientras que la perdida (Loss) alcanza un 2% lo cual demuestra que el modelo de inteligencia artificial desarrollado es se debe someter a las etapas de prueba con tecnología de Visión Artificial.

En la Figura 16 se puede apreciar que el aprendizaje del entrenamiento alcanza un aprendizaje mayor al 90% y la perdida desciende al 3% muy similar a la validación con ligeras variaciones. El entrenamiento mejora los parámetros permitió alcanzar el 96% en Accuracy y en perdida 2%.

**Figura 16**

*Resultados del entrenamiento del modelo (Etapa de entrenamiento).*

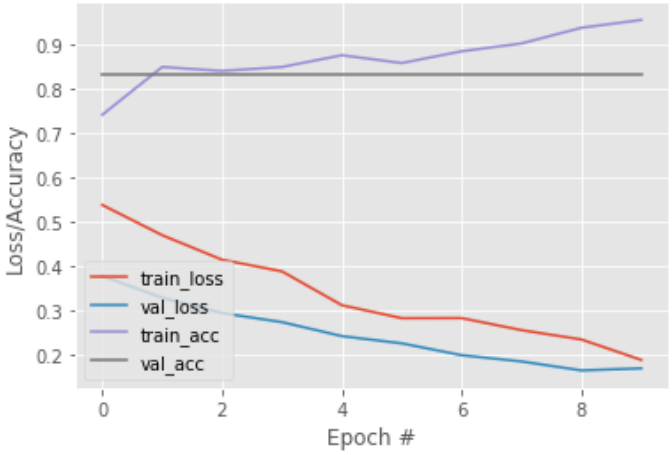



De la misma forma como se puede apreciar en la Figura 18 los resultados del entrenamiento en las etapas de entrenamiento y validación siguen caminos muy similares como los descritos en la Figura 17. Mientras que el aprendizaje supera el 90% en la etapa de entrenamiento, la validación supera el 80%, por otra parte, el entrenamiento de la perdida desciende hasta el 3%, y la validación desciende a un valor similar. Esto implica que el modelo de inteligencia artificial creado alcanza los objetivos de la investigación planteada, estas métricas se definen en el apartado 3.2.7.2 donde se describe cada una de las métricas utilizadas, sin embargo, para poder aclarar de mejor forma se

presenta solamente las gráficas de Accuracy y Loss tanto en entrenamiento como en validación, demostrando que el modelo creado alcanzó los objetivos propuestos.

**Figura 17**

*Resultados del Entrenamiento del Modelo (Etapa de Validación vs Entrenamiento).*




*3.2.10.* Validación del Modelo.

En la Tabla 4 se puede observar los resultados del entrenamiento de la RNC y el modelo de IA generado para las predicciones, tanto en Accuracy, sensibilidad y especificación.

**Tabla 4**

*Resultados de la Matriz de Confusión.*

| |
|---|
| **[[160 0]** |
| **[ 66 0]]** |
| **acc: 0.8387** |
| **sensitivity: 1.000** |
| **specificity: 0.000** |

En la Tabla 6 se puede evidenciar los datos resultantes de la evaluación del modelo de IA, donde la precisión obtiene un resultado de 0.90 para la clase inatento y en atento 1.00.

**Tabla 5**

*Resultados de la Evaluación del Modelo cargado.*

| | **precision** | **recall** | **f1-score** | **support** |
|---|---|---|---|---|
| **atento** | 0.90 | 1.00 | 0.95 | 201 |
| **inatento** | 1.00 | 0.40 | 0.57 | 200 |

| | | | | |
|---|---|---|---|---|
| **accuracy** | | | 0.90 | 401 |
| **macro avg** | 0.95 | 0.70 | 0.76 | 401 |
| **weithted avg** | 0.91 | 0.90 | 0.89 | 401 |

En la Tabla 6 se puede observar el resultado de la predicción de Atento con una imagen diferente a las imágenes utilizadas en la etapa de entrenamiento y validación del modelo de IA.

**Tabla 6**

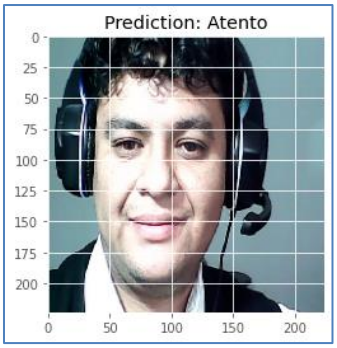
*Resultado de la Predicción de Atento.*

| |
|---|
| **[[1.0000000e+00 1.4374905e-13]]** |
| **[0]** |

A través de la Figura 18 se puede observar que el modelo de IA funciona correctamente al poder predecir si una persona está Atenta.

**Figura 18**

*Predicción: Atento*




En la Tabla 7 se puede apreciar el resultado de la predicción de la clase inatento del modelo de IA creado.

**Tabla 7**

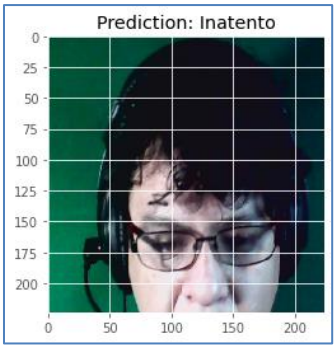
*Resultado de la Predicción de Inatento.*

| |
|---|
| **[[2.6846156e-11 1.0000000e+00]]** |
| **1** |

En la Figura 19 se puede evidenciar que el modelo de IA funciona correctamente al poder predecir si una persona está Inatenta.

**Figura 19**

*Predicción: Inatento*



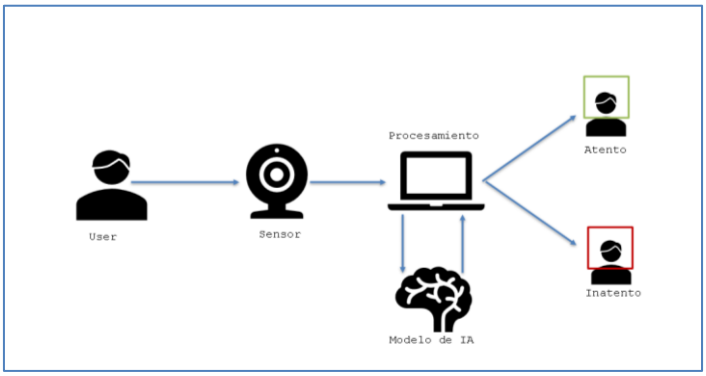

3.3. Detección de la Atención a través de Visión Artificial.

A continuación, se describe la arquitectura propuesta en la presente investigación para la detección de la atención e inatención.

En las Figuras 20 se define la arquitectura que permitirá la detección de la atención o inatención a través de VA, se empieza a describir el usuario que se encuentra frente a un sensor de la cámara, cuando se capta un rostro a través de vídeo es enviado como parámetro a la inteligencia artificial.

**Figura 20**

*Arquitectura para la detección de la atención utilizando el modelo de IA.*




En la Figura 21 se observa la función Detectar y Predecir la Atención (detect_and_predict_attention), la misma permite generar el frame que podrá capturar en tiempo real en vídeo la cámara que se pueda detectar.

**Figura 21**

*Función de detección y predicción de la atención.*

```
def detect_and_predict_attention(frame, faceNet, attentionNet):
    # grab the dimensions of the frame and then construct a blob
    # from it
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
        (104.0, 177.0, 123.0))

    # pass the blob through the network and obtain the face detections
    faceNet.setInput(blob)
    detections = faceNet.forward()
    print(detections.shape)

    # initialize our list of faces, their corresponding locations,
    # and the list of predictions from our face attention network
    faces = []
    locs = []
    preds = []
```

En la Figura 22 se genera el código que permite detectar la atención, se utiliza como parámetros el modelo de inteligencia artificial para poder realizar las predicciones, si un rostro es detectado por el sensor de la cámara, al predecirse que el rostro está atento se crea alrededor del rostro un recuadro de color verde, caso contrario si rostro que se detecta no está prestando atención se genera un recuadro de color rojo lo cual significa que el rostro está inatento. Para este código se utilizó la librería *OpenCV*. (2020) de acuerdo con la documentación oficial que se puede encontrar en la página oficial.

**Figura 22**

*Detección de la Atención.*

```
# loop over the frames from the video stream
while True:
    # grab the frame from the threaded video stream and resize it
    # to have a maximum width of 1024 pixels
    frame = vs.read()
    frame = imutils.resize(frame, width=1024)

    # detect faces in the frame and determine if they are wearing a
    # face attention or not
    (locs, preds) = detect_and_predict_attention(frame, faceNet, attentionNet)

    # loop over the detected face locations and their corresponding
    # locations
    for (box, pred) in zip(locs, preds):
        # unpack the bounding box and predictions
        (startX, startY, endX, endY) = box
        (atento, desatento) = pred

        # determine the class label and color we'll use to draw
        # the bounding box and text
        label = "Atento" if atento > desatento else "Inatento"
        color = (0, 255, 0) if label == "Atento" else (0, 0, 255)

        # include the probability in the label
        label = "{}: {:.2f}%".format(label, max(atento, desatento) * 100)

        # display the label and bounding box rectangle on the output
        # frame
        cv2.putText(frame, label, (startX, startY - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
        cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

    # show the output frame
    cv2.imshow("Attention", frame)
    key = cv2.waitKey(1) & 0xFF
```

# Capítulo cuatro

## 4. Metodología.

### 4.1. Metodología de Investigación.

En el desarrollo de la presente investigación se da inicio con la revisión bibliográfica de literatura, se obtiene investigaciones relacionadas con la detección de la atención, inatención y la construcción de modelos de IA a través de redes neuronales convolucionales.

Se propone la metodología sugerida por (Torres-Carrion et al., 2018) en su trabajo denominado “Methodology for systematic literature review applied to engineering and education”, ya que en el mismo propone un método para la revisión sistemática de la literatura científica

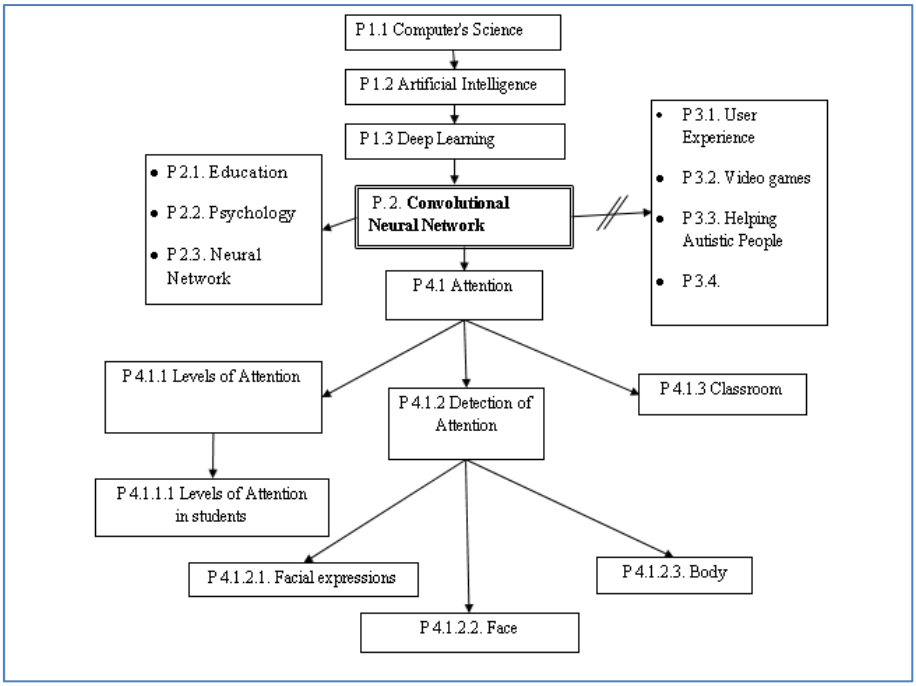
### 4.2. Revisión Sistemática de Literatura.

En la presente revisión sistemática de literatura se realizó una búsqueda de información con los siguientes constructos: Atención, CNN (Convolutional Neural Network), de los cuales se utilizó la búsqueda de información en la base de datos científica Springer, las investigaciones relacionadas al objeto de estudio de esta investigación se describen a continuación.

Esta es la primera fase, la cual permitió tener una mejor comprensión de la concepción general respecto a la detección de la “Atención” y las “Redes Neuronales Convolucionales”. Al realizar una revisión sistemática de literatura (SLR) se permite justificar una investigación, y dar respuesta a las preguntas planteadas en un inicio. Un SLR es útil para poder identificar las publicaciones científicas relacionadas con el tema investigado y servirán para la adquisición de conocimiento de otros investigadores (Torres-Carrion et al., 2018).

**Figura 23**

*Mentefacto Conceptual sobre “Atención” y “Redes Neuronales Convolucionales”.*




Inicialmente se construye el mentefacto conceptual como se describe en la Figura 23, ya que este permitirá visualizar de forma gráfica las principales características del objeto de estudio teniendo en cuenta el enfoque central del estudio, esto permitirá conocer los términos relacionados con el objeto de estudio de la investigación y poder dar mayor profundidad (Torres-Carrion et al., 2018).

Mediante la revisión sistemática de literatura se pudo identificar y evaluar los trabajos relacionados con la presente investigación, tomando en cuenta el nivel de similitud e importancia de estos. En la Figura 45 se puede observar el mentefacto conceptual que permite tener una visión muy general, dentro del campo del Deep Learning las redes neuronales convolucionales y como estas se relacionan con la atención. Con este precedente se procedió a realizar la búsqueda de investigaciones en la base de datos indexada como Scopus[12] donde se seleccionaron los trabajos que sirvieron de base para el desarrollo de la presente investigación.

Al seguir la metodología de Torres-Carrion et al. (2018) se puede asegurar que la búsqueda de información es confiable, se recomienda utilizar bases de datos científicas. El siguiente paso consiste en elaborar el tesauro y la sinonimia basada en el mentefacto conceptual, esto permitió tener la mayor cantidad de resultados de los campos relacionados a la investigación en el script de búsqueda. Se obtuvo bibliografía reciente que permitió adquirir criterios científicos para generar una

[12] Scopus: https://www.scopus.com/search/form.uri?display=basic

investigación de calidad obteniendo los mejores resultados, se define el mejor camino a seguir en las etapas posteriores.

*4.2.1. Investigaciones relacionadas con la Atención.*

Según Díaz. (2017), en su investigación analiza los niveles atencionales y la concentración a través de distintas herramientas, esta investigación se relaciona con el objeto de estudio de la presente investigación que busca conocer ¿Qué es la atención?, analizar los niveles atencionales existentes y su relación con la concentración podría definir métricas para poder predecir cuando una persona está atenta.

Citando a Burnik et al. (2018) analiza la problemática de mantener la atención de estudiante, como experimentación se analiza las conferencias. La atención varía de un estudiante a otro, depende de la capacidad de los conferencistas para mantener motivados a los estudiantes, se utiliza anotaciones manuales y web, se analiza señales visuales, se obtiene métricas por estudiante de acuerdo con las anotaciones realizadas, también se utiliza la evaluación como métrica de atención. En esta investigación el procedimiento es muy anticuado y sujeto a errores que se pueden presentar por hacerlo de forma manual.

Zaletelj & Košir, (2017) utiliza un sensor denominado Kinect One, el mismo permite estimar el nivel de atención de estudiantes, utiliza la observación visual de señales de comportamiento y lo correlaciona con la atención, se crea un modelo de predicción de la atención con un conjunto de 18 personas obteniendo una precisión de 0.753, relaciona con los propósitos de la presente investigación, sin embargo el objeto de estudio es diferente, ya que utiliza como parámetros señales de comportamiento, lo cual requiere de un historial del departamento de bienestar estudiantil o similares que organizan la información en portafolios individualizados por cada estudiante.

De acuerdo a Paletta et al. (2013) utiliza tecnología en 3D para determinar la atención mediante el seguimiento ocular mediante un sistema de múltiples componente que permite un mapeo generalizado de la atención humana. Su metodología realiza un mapeo de la mirada, utilizan tecnología de descriptores, utiliza anotaciones automatizadas mediante regiones de interés. Se relaciona con la presente propuesta ya que involucra el estudio de la atención, sin embargo, no ofrece métricas relevantes a ser tomadas en cuenta como se propone en la presente investigación.

Asteriadis et al. (2014) realiza una investigación para estimar el foco de atención de una persona, analiza lógica difusa para la rotación de la cabeza, las estimaciones de la mirada. El sistema

es capaz de reconocer el movimiento de la cabeza, movimientos de traslación, utiliza el sensor de la cámara. Esta investigación se relaciona con la presente propuesta, sin embargo, utiliza tecnología de lógica difusa, lo cual se aleja del objeto de estudio de la presente investigación que propone el desarrollo de un modelo de Inteligencia Artificial a través de una arquitectura utilizando Redes Neuronales Convolucionales.

Dinesh, (2016) propone una investigación que utiliza 5 estados afectivos (activo, transcriptivo, inútil, distraído y en transición), se analiza vídeos pregrabados de clases desde una ubicación remota que permite proporcionar retroalimentación al maestro. Se analiza el seguimiento del rostro y el flujo óptico de cada estudiante, el sistema puede detectar el nivel de atención del estudiante, define el nivel de interés por los temas tratados, a los docentes les permite mejorar el estilo de instrucción. Esta investigación tiene estrecha relación con el objeto de estudio de la presente investigación.

*4.2.2. Investigaciones relacionadas con Redes Neuronales Convolucionales.*

El aprendizaje automático también nos brinda un gran aporte, según Whitehill et al, (2014) proponen una investigación que permite desarrollar detectores de participación alta y baja utilizando modelos de aprendizaje automático, los experimentos propuestos permiten demostrar que el aprendizaje automático funciona con mayor precisión que los observadores humanos.

Según C. M. Chen et al. (2017) proponen una investigación que analiza si los estudiantes prestan atención en un ambiente en línea, desarrollan un sistema de atención consciente para determinar si los estudiantes permanecen enfocados, el sistema es capaz de reconocer los niveles de atención utilizando señales electro encefálicas, el sistema permite a los instructores evaluar los niveles de atención de sus estudiantes y les permite mejorar el desempeño en este tipo de tutorías.

Gracias a Mills et al, (2016) se demuestra que la mirada es un indicador que permite medir la atención, también incluye las divagaciones o distractores mentales, por otra parte Monkaresi et al. (2017) concluye que el seguimiento ocular se ve afectado por los movimientos de la cabeza y determina que no es fácilmente escalable en el mundo real, esta investigación se relaciona con el objeto de estudio de la presente propuesta ya que en la misma se busca determinar la atención a través del seguimiento ocular, la posición de la cabeza y la posición del rostro.

*4.2.3. Investigaciones relacionadas con la Distracción.*

Según Bixler & D’Mello. (2016) se propone una investigación donde se analiza la mirada y las señales para detectar las divagaciones de la mente durante la lectura utilizando la interfaz de la

computadora. Detecta una precisión del 72%, al finalizar la página alcanza una precisión del 67%, la taza de divagación mental detectada se relaciona negativamente con las medidas de aprendizaje, esta investigación da indicios a la presente propuesta de analizar no solo la atención sino estos periodos de divagación que se pueden presentar en el desarrollo del modelo de IA.

Saeed et al. (2015) propone una investigación donde se utiliza Kinect, se analiza el reconocimiento facial, los gestos de la cabeza, la detección de bostezos, para ello utiliza la tecnología descrita por Vila y Jones introduciendo un descriptor de características, esto da una pauta para profundizar en tecnologías como el reconocimiento facial y la detección de posturas de la cabeza, la detección de bostezos puede ser un indicador de inatención.

*4.2.4. Investigaciones relacionadas con las Redes Neuronales Convolucionales.*

Mukherjee & Robertson. (2015) propone un modelo basado en Redes Neuronales Convolucionales que permite medir la estimación de la pose de la cabeza, profundiza en la clasificación de la dirección de la mirada, utiliza la combinación de dos modelos, la clasificación y la regresión. Esta investigación se relaciona con el objeto de estudio de la presente propuesta al tomar en cuenta la dirección de la mirada como un posible indicador para medir la atención.

Según T. S & Guddeti. (2020) proponen una arquitectura basada en Redes Neuronales Convolucionales utilizando 2 modelos, el primero analiza los estados afectivos de un solo estudiante, el segundo utiliza múltiples estudiantes, por lo tanto, es posible analizar el estado afectivo de toda la clase. Esta arquitectura analiza expresiones faciales, gestos de las manos, posturas del cuerpo, demuestran una efectividad del 86% y 70% en la predicción si el estudiante está comprometido, aburrido o neutral. Esta investigación se relaciona por los estados que puede predecir, no se había considerado el estado neutral, hasta ahora solo atento e inatento en la presente propuesta.

*4.2.5. Investigaciones relacionadas con la atención y el seguimiento ocular.*

(Madsen et al., 2021) realizan una investigación que involucra los efectos de la atención en los movimientos oculares. Los resultados obtienen una correlación entre los participantes de sus movimientos oculares motivados por la atención para ver los vídeos. Esta investigación se relaciona directamente con el objeto de estudio de la presente propuesta al demostrar la correlación entre el seguimiento ocular y la atención.

A continuación, se describe la metodología de desarrollo de software que se utilizó para la presente investigación y el desarrollo de la propuesta de solución al problema de detectar la atención

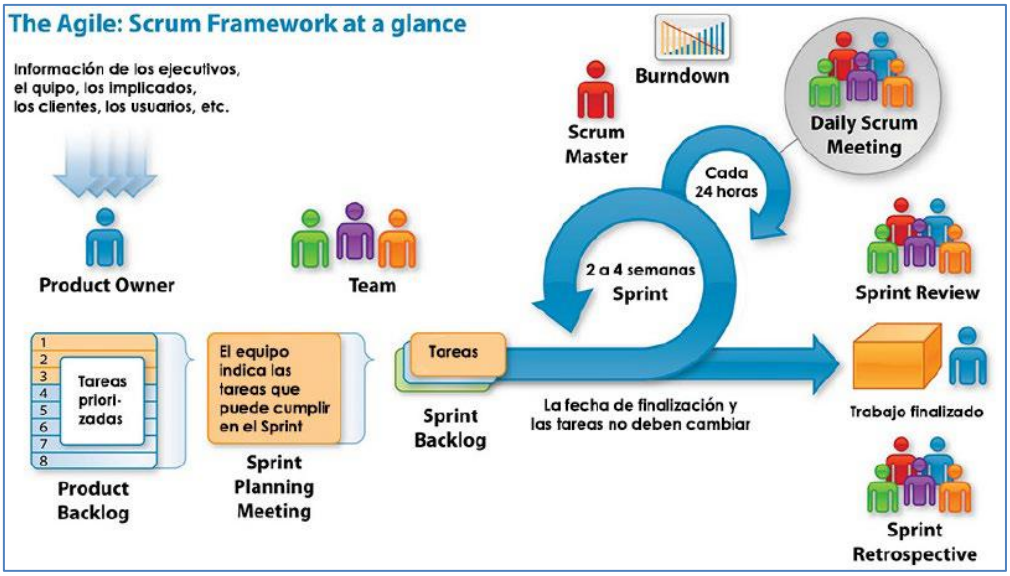
mediante la creación de una RNC para la generación de una IA que permita detectar la atención de estudiantes en un aula de clases.

## 4.3. Metodología - SCRUM

Para el proceso de desarrollo de la propuesta de investigación se propone la metodología SCRUM, la misma posee un método de gestión de proyectos interactivo e incremental, se implementa tres pilares de la metodología: transparencia, inspección y adaptación (*SCRUM*, 2021).

**Figura 24**

*Modelo de Trabajo de la Metodología SCRUM.*




Modelo de trabajo de la metodología Scrum, por (*SCRUM*, 2021).

En la Figura 24 se puede observar el proceso de la metodología para obtener un proceso o producto y desempeño de cada una de las personas que cumplen en la metodología Scrum. A continuación, se detalla el uso de la metodología dentro de la presente investigación.

### *4.3.1. Product Backlog*

Para empezar con el desarrollo se procede a describir una lista de funcionalidades que el sistema de detección de atención e inatención debe cumplir, este proceso es definido como Product Backlog. Esta lista debe tener los lineamientos con los que se desea obtener el producto final, los que a continuación se presenta:

- El sistema debe ejecutarse en un ambiente amigable para el usuario final, sin provocar métodos intrusivos como sensores para la detección de la epidermis del cuerpo (sensores en la piel).

- La detección de la atención e inatención solo se puede realizar siempre y cuando el haya sido detectado un rostro.
- La dirección de la mirada y la posición de la cabeza serán fundamentales para que el modelo de IA pueda determinar si el usuario está prestando atención o inatención.
- La detección de la atención o inatención puede variar en el proceso si cambia la dirección de la mirada o el rostro, se debe definir una serie de criterios de selección de las imágenes para la etapa de entrenamiento de la RNC.
- La implementación del sistema debe permitir detectar la atención e inatención en tiempo real como también a través de la grabación de vídeos.

*4.3.2. Spring Backlog.*

Tomando en cuenta los parámetros definidos en la etapa producto backlog, se puede empezar con el desarrollo del sistema. Es necesario definir las actividades que se deben cumplir de forma ordenada. A este proceso se lo denomina backlog, dichas actividades se describen a continuación.

- Entrenamiento de la RNC.
    - Recolección de Información.
    - Generación de Criterios de Selección.
    - Clasificación de la Data.
    - Tratamiento de la Data.
    - Creación de la arquitectura de la RNC.
    - Entrenamiento de la RNC.
    - Compilar el Modelo de IA.
    - Evaluación del entrenamiento y compilación del modelo de la IA.
    - Validación del Modelo
    - Almacenamiento del modelo de IA.
- Detección de la Atención a través de VA.
    - Detección de la Atención a través de la Predicción

En cada una de las etapas mencionadas es necesario culminar con éxito la etapa anterior para poder continuar con la siguiente, caso contrario es necesario modificar los parámetros que sean necesarios para la creación de la RNC, el modelo de IA y el script de VA que permitirá la detección de la atención o inatención a través de reconocimiento facial.

4.3.2.1. Entrenamiento de la RNC.

Para realizar el entrenamiento de la RNC es necesario definir las herramientas que se utilizarán en todo el proceso, estas herramientas se describen en la sección 3.1., definidas dichas herramientas servirán para poder generar la Arquitectura de la RNC que se describe en el apartado 3.2 de forma general.

4.3.2.2. Recolección de Información.

Como se describe en la sección 3.2.1. se procede a identificar las fuentes de información en donde se puede obtener la Data necesaria que servirá como base en las etapas de entrenamiento y testeo en la creación de la RNC, en la sección 3.2.2. se describe los métodos y técnicas utilizados para la obtención de la data, es necesario determinar las etapas de creación del dataset que se describen en la Tabla 1.

4.3.2.3. Generación de Criterios de Selección.

En esta etapa como se describe en la sección 3.2.3. en particular en la Tabla 2, utilizando las referencias bibliográficas de diferentes psicólogos se procede a generar una serie de criterios de selección de las imágenes que posteriormente serán utilizadas en las etapas de entrenamiento y testeo de la RNC, esta serie de criterios es una de las partes más importantes de la creación de la RNC, ya que, si no se determinan de forma adecuada, posiblemente en la etapa de testeo no será posible la detección de atención o inatención.

4.3.2.4. Clasificación de la Data.

En esta etapa como se describe en la sección 3.2.5. se puede observar cómo se procede a hacer una clasificación manual de cada una de las imágenes que formarán parte del dataset de información para las categorías de atento e inatento.

4.3.2.5. Tratamiento de la Data.

En esta etapa se describe las herramientas utilizadas, las características del equipo donde se crearon las pruebas y la tarjeta gráfica que permite el uso de la GPU como se describe en la sección 3.2.5, se inicia con el uso de Colab como herramienta para acelerar los procesos de carga de librerías y la carga del dataset de las categorías de Atento e Inatento.

4.3.2.6. Creación de la Arquitectura de la RNC.

En esta etapa como se describe en la sección 3.2.6, se procede a la creación de la RNC, se hace énfasis en las diferentes capas ocultas, las convoluciones, la normalización de la data, los

métodos de activación y el MaxPooling utilizado, así como la técnica de Droput que permite desconectar las diferentes neuronas para evitar el sobre entrenamiento (overfitting) o el poco entrenamiento (underfitting).

4.3.2.7. Entrenamiento de la RNC.

Esta es una de las etapas más importantes, ya que es necesario regresar a las etapas anteriores si no se obtiene los resultados esperados (Un entrenamiento aceptable superior al 90% de aprendizaje y menos de 3% de perdida), como se describe en la sección 3.2.7, de la propuesta y en la Figura 33, es necesario realizar ajustes en las etapas previas de Tratamiento de la Data y Clasificación de la Data.

4.3.2.8. Compilar el Modelo de IA.

En esta etapa como se describe en la sección 3.2.7. se describe el proceso que se lleva a cabo para la compilación del modelo y la generación de las métricas que permitirán evaluar tanto el aprendizaje como la pérdida del modelo de IA que permitirá la detección de la atención.

4.3.2.8.1. Evaluación del entrenamiento y compilación del modelo de la IA.

A través de la Figura 34 se puede observar las etapas de entrenamiento del modelo de forma gráfica, donde se puede evidenciar que el entrenamiento de la RNC alcanza un nivel aceptable superior al 96% de aprendizaje y la perdida es menor al 3%, en la figura 35 se puede observar tanto la etapa de entrenamiento como la de validación siguen caminos similares.

4.3.2.8.2. Validación del Modelo.

En esta etapa como se describe en la sección 3.2.5, se procede a la validación del modelo de la RNC tanto para la categoría atento como inatento. En la Tabla 4 se puede observar los resultados de la Matriz de Confusión, una métrica utilizada para analizar el desempeño de los modelos de IA desarrollados a través de RNC, en la Tabla 5 se puede observar los resultados de la evaluación del modelo con una precisión cercana a 0.90 para atento y de 1.00 para inatento. En la Figura 44 se puede observar una predicción del modelo de IA para Atento y en la Figura 45 la predicción para Inatento.

4.3.2.8.3. Almacenamiento del modelo de IA.

Mediante la Figura 38 se puede observar cómo se procede al almacenamiento del modelo de IA creado para la detección de atención, esto se lo realiza con el objetivo de pasar a las siguientes etapas con el algoritmo de visión artificial para analizar en tiempo real o a través de vídeos pregrabados.

4.2.2.2. Creación del Script de VA.

En esta sección se describe como el Script de VA permite detectar la atención utilizando el modelo de IA previamente entrenado.

En la sección 3.3, se explica cómo a través de una cámara web de una laptop o PC es posible detectar la atención o inatención. Las pruebas de la eficiencia de este modelo se describen en el capítulo 5 de esta investigación.

4.2.2.2.1. Detección de la atención a través de la predicción.

En la sección 3.3 se describe todo el proceso de creación del Script de VA para la detección de la atención, en la Figura 52 se puede observar cómo se detecta la atención a través del reconocimiento de un rostro, en la Figura 53, se puede observar cómo es cargado el modelo de IA que es utilizado para las predicciones de atención o inatención. En la Figura 43 se puede observar cómo se realiza el proceso de detección de la atención o inatención siempre y cuando es posible detectar un rostro y dicha detección supera el 0.5, en la Figura 54 se destaca el uso de VA para la detección de la atención a través del modelo de IA que permite predecir si se detecta atención o inatención. En la Figura 46 se define los colores para diferenciar la atención de inatención, el color verde determina que un rostro está prestando atención y el color rojo determina la inatención. Esto se puede determinar tanto en tiempo real como en vídeos grabados de clases virtuales, presenciales o una reunión a través de medios digitales. Debido al periodo de aislamiento no se pudo hacer pruebas en tiempo real en un aula de clases, para evitar contagios.

*4.2.3. Spring Planning meeting.*

Las Spring Planning Meeting son las reuniones con el personal que está interesado en el resultado. En este caso se mantiene reuniones periódicas con el director de este trabajo de investigación para poder determinar cómo se va enfocando el proyecto. Además, también se mantienen reuniones esporádicas con clientes finales (Docentes) que van analizando los resultados obtenidos. Se va mostrando avances en cada reunión.

*4.2.4. Spring Review.*

En esta etapa se presentan avances realizados hasta determinada fecha. En esta investigación se procedió a mostrar avances al termino de cada una de las actividades del sprint backlog, se procede a evaluar cada etapa de la generación de la RNC, también se pide la opinión de programadores

externos al proyecto para tener en cuenta sus sugerencias para mejorar dichas etapas. De la misma forma se procede a determinar errores que deben ser corregidos previo a la siguiente etapa propuesta.

#### *4.2.5. Spring retrospective.*

En esta etapa se procede a finalizar con el proyecto, previo a realizar una revisión de cada uno de los objetivos propuestos, en el presente sistema se cumple con el objetivo principal para el cual fue creado, se realiza un análisis sobre los errores cometidos para evitarlos en el futuro. En el diseño final se puede hacer pruebas con diferentes tipos de vídeos pregrabados, se aprovechó algunos sugeridos por internet que son públicos como clases virtuales a través de plataformas como Zoom Meeting, Google Meet, entre otras. Esto último debido a encontrarnos en tiempos de pandemia y tener las clases presenciales suspendidas de forma indefinida hasta la fecha de desarrollo de la presente investigación. Dentro de los cambios sugeridos está el cambiar la técnica de transfer learning por la creación de una RNC desde 0 ya que el aprendizaje con esta técnica mediante el dataset definido con los criterios de selección no superaba el 70% de aprendizaje de la IA.

# Capítulo cinco

## 5. Experimentación y Análisis de Resultados.

### 5.1. Experimento 1, Detección de la atención e inatención de 1 persona.

Después de la investigación desarrollada se observó que el modelo de IA desarrollado a través de RNC permite la detección de la atención en tiempo real para una persona a través del sensor de la cámara, el script desarrollado con VA consume muchos recursos del equipo de prueba.

#### *5.1.1. Detección de la Atención e Inatención en Tiempo Real con un usuario.*

En la Tabla 8 se describe las pruebas realizadas en el Experimento 1, las mismas que permiten predecir diferentes estados de atención e inatención en una sola persona con diferentes tamaños del frame utilizados (400, 800, 1024). En la misma tabla se puede apreciar que cuando se detecta que una persona está atenta se grafica un recuadro de color verde alrededor del rostro, de la misma forma cuando se detecta que una persona está inatenta se grafica un recuadro de color rojo con el título de inatento.

Las pruebas describen la detección de atención con objetos en el rostro (lentes, audífonos) lo cual no es un impedimento para poder predecir si la persona está atenta o inatenta, esto se debe a que los criterios de selección propuestos en la sección 3.2.3 son validados para la detección de la atención e inatención con una sola persona, esto supera el experimento 1 con excelentes resultados.

Con respecto a las métricas de la precisión con la que se realiza la predicción, se puede evidenciar que, al detectar la atención, la misma supera el 90% y cuando se detecta inatención consigue una predicción supera el de 60%. Inclusive con objetos en el rostro que pueden impedir que se determine si una persona está atenta o inatenta.

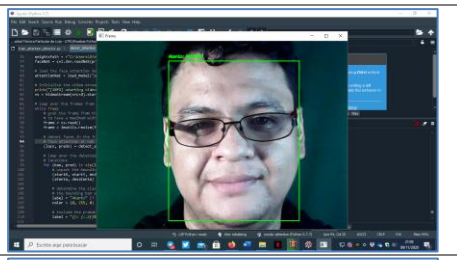
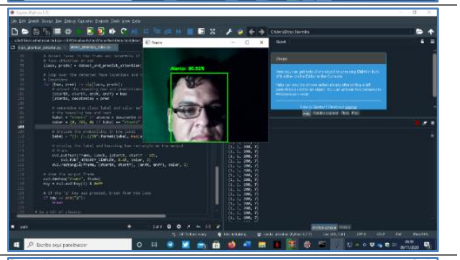
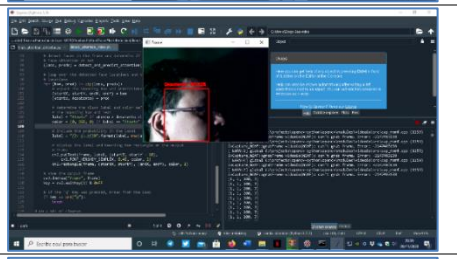
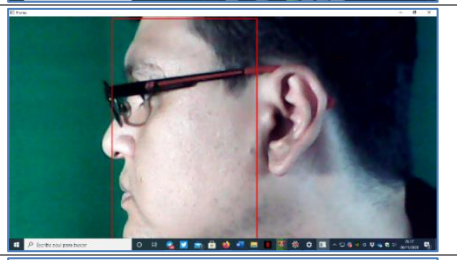
Se utilizó una cámara web básica la misma que se puede encontrar en computadores portátiles como también se puede adquirir en cualquier lugar de venta de accesorios de computadores a precios bajos si se trata de un computador de escritorio.

La iluminación en este experimento no fue un impedimento para el desarrollo de este, ya que se realizó experimentos con la luz natural del día y también en horas de la noche con luz de una lámpara. Esto demuestra que no es necesario tener una fuerte iluminación para que el modelo funcione adecuadamente.

Como prueba adicional se realizó la detección de la atención e inatención de un estudiante a través de un vídeo pregrabado por la plataforma Zoom Meeting.

**Tabla 8**

*Detección de la atención e inatención.*

| Descripción | Predicción |
|---|---|
| Detección de la Atención en tiempo real con un frame de 800 |  |
| Detección de la Atención en tiempo real con un frame de 400 |  |
| Detección de la Atención con objetos en el rostro |  |
| Detección de la Inatención en tiempo real con un frame de 400. |  |
| Detección de la Atención con un frame de 1024 |  |
| Detección de la inatención con objetos en el rostro |  |

| | |
|---|---|
| Detección de la Inatención de un estudiante en la plataforma zoom. |  |

5.2. Experimento 2, Detección de la atención e inatención con 2 personas.

En el experimento 2 se procede a realizar pruebas de detección de la atención e inatención con más de una persona.

*5.2.1. Detección de la Atención e Inatención en Tiempo Real con 2 usuarios.*

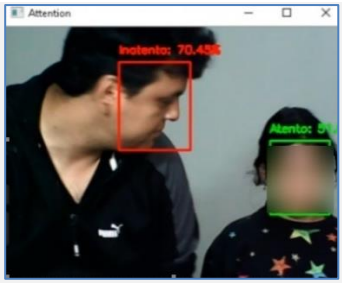
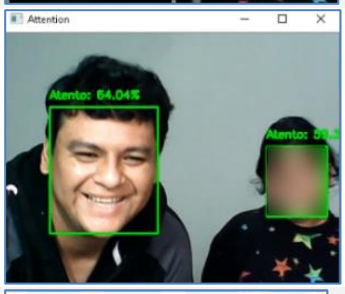
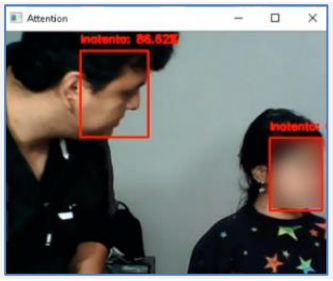
En esta sección se realizan pruebas con más de un usuario a través del sensor de la cámara en tiempo real. Se oculta el rostro del infante para precautelar su integridad, sin embargo, se puede observar el porcentaje de atención o inatención en más de una prueba realizada, el sensor de la cámara utilizada es muy básico para determinar si se necesita mejores prestaciones en este tipo de sensores en tiempo real en clases virtuales. El tema de aislamiento por el temor a contagio se maximizó e hizo imposible poder conseguir los permisos necesarios para poder hacer pruebas en tiempo real con clases en niños con plataformas como Zoom Meeting o similares. Así que las pruebas en tiempo real lo hice con familiares y personas cercanas, es por ello por lo que como precaución adicional se oculta el rostro de los participantes en las siguientes imágenes donde aparecen niños en las pruebas realizadas, el código de la niñez y adolescencia es muy estricto al referirse a menores, así sea en temas académicos se procuró hacer lo mejor por su integridad.

En la Tabla 9 la se puede evidenciar en los 3 casos, la detección de la atención e Inatención de 2 usuario, y la atención e inatención al mismo tiempo, la detección de la atención de 2 usuarios en tiempo real y la detección de la Inatención de 2 usuarios en tiempo real. Como se puede apreciar el modelo de IA permite detectar la atención e inatención con más de un usuario en tiempo real. Al igual que en el Experimento 1, las pruebas se las realiza con una cámara básica y la luz natural del día, lo cual no influye en la predicción que se puede realizar para de tectar la atención e inatención.

**Tabla 9**

*Detección de la Atención e Inatención con más de 1 usuario.*

| DESCRIPCIÓN | PREDICCIÓN |
|---|---|

| | |
|---|---|
| **DETECCIÓN DE LA ATENCIÓN E INATENCIÓN DE 2 USUARIOS EN TIEMPO REAL** |  |
| **DETECCIÓN DE LA ATENCIÓN DE 2 USUARIOS EN TIEMPO REAL** |  |
| **DETECCIÓN DE LA INATENCIÓN DE 2 USUARIOS EN TIEMPO REAL** |  |

5.3. Experimento 3, Detección de la atención e inatención con más de 2 personas.

En este experimento se realiza la detección de la atención e inatención con más de 2 personas en clases pregrabadas a través de distintas plataformas virtuales.

*5.3.1. Detección de la Atención en Tiempo Real con más de 2 usuarios.*

En esta sección se realiza pruebas con vídeos públicos a través de la red social YouTube, los mismos que fueron descargados para poder utilizarlos como parámetro al script de VA y conocer si el modelo de IA puede predecir la atención o inatención de los usuarios. En algunos casos por temas de iluminación, objetos en el rostro o sensibilidad del sensor de la cámara con los que fueron grabados no es posible la detección de un rostro, esto imposibilita que el script de VA creado pueda dar paso a los siguientes procesos de predicción de la IA.

En la Tabla 10 se puede observar cómo el modelo de IA en el Caso 1 puede detectar la atención de más de un usuario en clases pregrabadas, sin embargo, no detecta todos los usuarios, esto debido a que la detección de rostro no supera el 0.5% de detección.

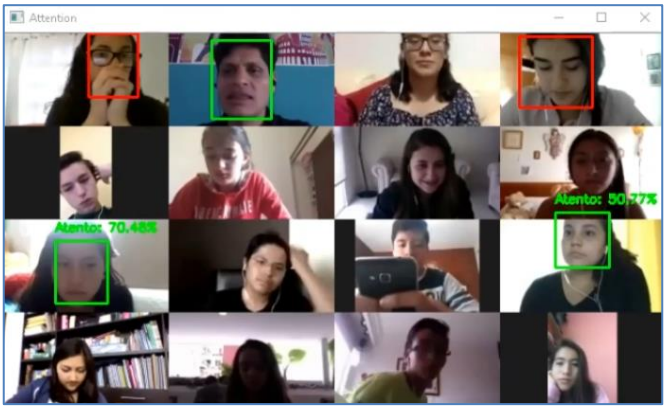
En el Caso 2 puede detectar la atención de 3 estudiantes y la inatención de 2 estudiantes, para el resto de los estudiantes por problemas de iluminación, baja resolución de las cámaras, objetos, no es posible detectar el rostro.

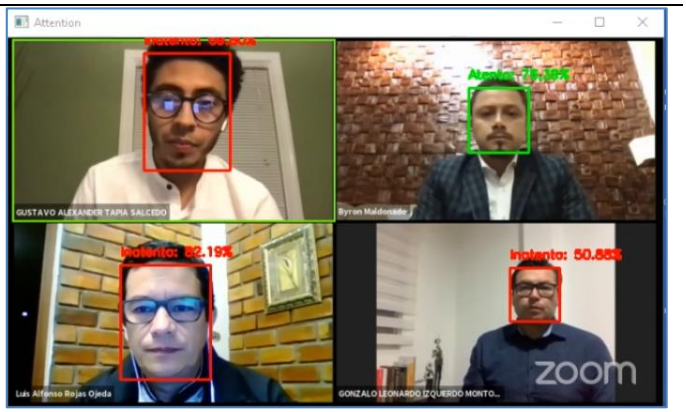
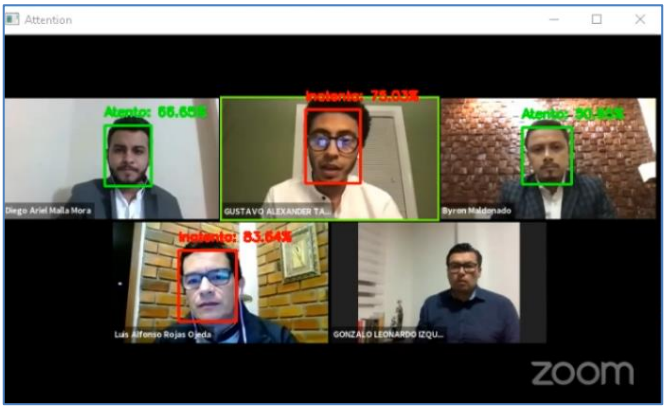
En el Caso 3 se puede observar cómo el modelo de IA puede detectar la atención e inatención de 4 personas conectadas a través de la plataforma Zoom, en la misma se puede evidenciar que solo 1 persona está observando directamente al sensor de la cámara mientras que las otras 3 personas miran a la pantalla, pero no a la cámara, también se evidencia que una luz se refleja en los lentes de 2 de las personas lo cual puede ser un factor para que el modelo lo categorice como inatención.

En el Caso 4 se puede observar cómo el modelo de IA es capaz de detectar la atención de 2 personas, la inatención de 2 personas y no detecta el rostro de la última persona. Esto se debe a que los 2 primeros usuarios miran al sensor de la cámara de su computador, los 2 usuarios donde se detecta la inatención se refleja una luz en sus lentes lo cual el modelo lo cataloga como inatención y en el caso donde no se detecta la atención o inatención, la detección de rostro no supera la base de 0.5.

**Tabla 10**

*Detección de la Atención e Inatención con más de 2 usuarios.*

| Casos | Descripción | Predicción |
|---|---|---|
| **Caso 1** | Detección de la Atención e Inatención en clases pregrabadas por Zoom |   |
| **Caso 2** | Detección de la Atención e Inatención en clases pregrabadas diferente a Zoom |   |

| | | |
|---|---|---|
| **Caso 3** | Detección de la Atención e Inatención de varias personas por Zoom |  |
| **Caso 4** | Detección de la Atención e Inatención de varias personas por Zoom. |  |

5.4. Detección de la Atención en Tiempo en un aula de clases.

Una de las pruebas principales realizadas en esta investigación fue el utilizar clases presenciales pregrabadas antes del periodo de pandemia por Covid-19, lo cual también se utilizó dentro de una de las pruebas de la presente investigación, sin embargo la posición del rostro y la mirada están en diferente perspectiva del sensor de la cámara, lo cual imposibilita la detección del rostro y hace imposible que el modelo de IA pueda predecir si los estudiantes están atentos o inatentos, se considera que en otra perspectiva de grabación donde se determine la posición al frente tanto en rostro como mirada será posible determinar si el estudiante está atento o inatento.

En la Tabla 11 se puede observar como el Caso 1 el modelo de IA puede predecir la atención de un estudiante en un aula de clases, esto debido a que la dirección de la mirada y la dirección de la cabeza no miran directamente al sensor de la cámara, debido a que la posición de la pizarra, el docente o lo que llama la atención está en una dirección diferente a donde se encuentra el sensor de la cámara. Sin embargo, el modelo puede detectar la atención de un estudiante y esto es muy favorable, es necesario recalcar que la posición de la cámara en la grabación es muy importante ya que el modelo de IA propuesto en la presente investigación al ser alimentado por los criterios de selección determinará que considera atención o inatención independientemente si estos criterios sean válidos o no, los

mismos que se contraste con la literatura, los test de detección de atención y la selección de parámetros como el seguimiento ocular y la posición del rostro.

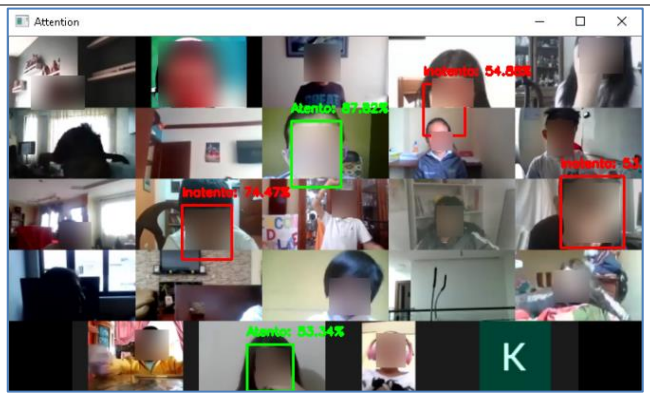
En el Caso 2 se puede observar como el modelo de IA es capaz de detectar la inatención de un estudiante, debido a diferentes factores anteriormente descritos en la Figura 64. Sin embargo, cabe recalcar como se puede apreciar en la imagen la perspectiva del rostro y la mirada son muy distintos a la ubicación de la cámara, posiblemente porque el escritorio o la pizarra están en una posición diferente al sensor de la cámara.

En el Caso 3 se puede observar una clase pregrabada con niños donde se puede detectar algunos rostros y el modelo de IA procede a categorizar los niveles de atención, donde no es posible detectar el rostro no se puede predecir la atención o inatención de los estudiantes. Para precautelar la integridad de los infantes se procede a anonimizar los rostros mediante la difuminación de la sección donde se encuentran los mismos en el vídeo pregrabado. Los trámites para poder conseguir un comité de ética que apruebe la experimentación con menores son muy complicados por lo que se procede a anonimizar los rostros de los participantes.

**Tabla 11**

*Detección de la Atención e Inatención en Aulas de Clase.*

| Casos | Descripción | Predicción |
|---|---|---|
| Caso 1 | Detección de la atención en un aula de clases. |   |

| | | |
|---|---|---|
| Caso 2 | Detección de la Inatención de un estudiante en un aula de clases. |   |
| Caso 3 | Detección de la atención e inatención en una clase virtual con niños. |   |

5.5. Discusión

Una de las investigaciones que se analizó a fondo previo al desarrollo de la presente propuesta de investigación es la de (Madsen et al., 2021) que involucra los efectos de la atención en los movimientos oculares durante la presentación de vídeos. Mientras su hipótesis plantea que los movimientos oculares dependen del estado de atención del sujeto, en esta investigación se añade el factor tiempo, porque una persona deja de poner atención y es objeto de inatención dependiendo del grado de motivación que le permita concentrarse en un fenómeno, suceso o evento que los sentidos en especial la vista pueden captar del entorno que lo rodea, si la motivación es lo suficientemente satisfactoria, la inhibición de lo que los sentidos capten permiten que el sujeto dirija su vista a lo que le interesa atender. Mientras sus experimentos proponen que el sujeto visualice vídeos pregrabados, en esta investigación solo se detecta la atención a través de algoritmos de visión artificial y un modelo de IA creado con arquitectura de RNC. Se coincide con esta investigación en que existe una correlación en los movimientos oculares motivados por la atención, y se suma la posición del rostro en la misma perspectiva del movimiento ocular.

## Conclusiones.

La presente investigación se concluye que es si posible crear un modelo de inteligencia artificial que permita analizar los niveles de atención de los estudiantes en un salón de clases, este modelo propuesto permite predecir la atención de estudiantes siempre y cuando el sensor de la cámara este ubicado en una posición adecuada donde se puedan ser captados los rostros.

La revisión sistemática de literatura permite analizar diferentes investigaciones referentes al análisis de la atención, lo cual permite encontrar la correlación entre los niveles de atención y el rendimiento académico.

La búsqueda de diferentes investigaciones donde se aplica RNC permite conocer métodos, técnicas, algoritmos, parámetros, configuraciones, herramientas, y resultados que son indispensables en la creación de la arquitectura de la RNC.

La revisión bibliográfica de literatura permite analizar diferentes investigaciones relacionadas con el objeto de estudio de la investigación, y determinar cuál es la arquitectura más adecuada para el desarrollo de un modelo de inteligencia artificial que permite medir la atención de estudiantes en un aula de clases.

La atención visual selectiva se puede determinar mediante parámetros como el seguimiento ocular, la posición del rostro y la posición de la cabeza, demostrando cuándo una persona está atenta o inatenta.

La creación de criterios de selección permite relacionar la atención con las imágenes que alimentan el modelo de inteligencia artificial, esto se verifica en las etapas de entrenamiento y validación del modelo de inteligencia artificial.

Definir una arquitectura de Red Neuronal Convolucional desde 0 contribuye de mejor forma para obtener un aprendizaje superior al 90%, en relación con la arquitectura que utiliza la técnica de Transfer Learning que no supero el 70% del aprendizaje.

La implementación del modelo de IA generado a partir de la RNC incide positivamente en la detección de la atención o inatención porque define los requisitos necesarios basados en la teoría de la atención visual selectiva, utilizada para determinar cuando una persona o

conjunto de personas están atentas o inatentas. El modelo propuesto es posible modificarlo para definir parámetros de funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad, adaptabilidad, tiempo de respuesta.

Las pruebas realizadas permiten determinar que la posición del rostro, la cabeza y la dirección de la mirada influyen en la determinación de la atención o inatención.

El uso del modelo de Inteligencia Artificial a través de Visión Artificial determina que es posible detectar la atención visual selectiva en una o más personas en tiempo real o mediante videos pregrabados.

La precisión del modelo de IA generado a través de la RNC consigue una confiabilidad de la precisión superior al 90%, por ejemplo, en el caso de experimentación con una persona, el nivel de precisión es superior a 85% en atención, cuando es más de uno se encuentra entre 70% y 80%, los factores como la iluminación, la calidad del sensor de la cámara, la distancia de la o las personas, entre otros factores, en algunos casos la confiabilidad llegó a 99.64% en atención y en inatención 64,26%

## Recomendaciones.

La detección de atención es muy difícil de calcular con test de psicología que se encuentra en la literatura relacionada a la atención, además de ser imprecisa en tiempos y características utilizadas por los psicólogos, se recomienda utilizar el modelo de Inteligencia Artificial creado para predecir la atención.

En nuevas investigaciones relacionadas a la presente propuesta se recomienda analizar otros parámetros como la postura del cuerpo, el tiempo que el estudiante está atento y los factores que permiten que pierda el foco de atención.

En futuras investigaciones se recomienda determinar un umbral de atención que permita obtener un rango para aumentar la confiablidad de la predicción.

Se recomienda analizar la influencia de otros factores como las emociones, Déficit de Atención, Hiperactividad, Síndrome de Down, Asperger entre otros.

Los directivos de instituciones educativas pueden implementar este modelo de IA obtenido en sistemas de VC para mejorar la toma de decisiones que permitan para recuperar la atención de estudiantes en las aulas de clases.

Se recomienda realizar investigaciones con un Dataset de información que contenga una sola categoría, atención, y todo lo que no esté contemplado en el entrenamiento sea considerado como inatención, esto posiblemente mejore la detección de inatención en futuras investigaciones.

Al existir pocas investigaciones relacionadas con la presente investigación se abre un abanico de posibilidades para investigar a fondo las razones por las cuales una persona está atenta o inatenta.

Se recomienda implementar sistema de detección de la atención en aulas de clases para mejorar el proceso de enseñanza y aprendizaje en los centros educativos que lo consideren pertinente.

## Referencias

AbuRass, S., Huneiti, A., & Belal, M. (2020). Enhancing Convolutional Neural Network using Hu's Moments. *International Journal of Advanced Computer Science and Applications*, *11*(12), 130–137. https://doi.org/10.14569/ijacsa.2020.0111216

Aksoy, A., Ertürk, Y. E., Erdoğan, S., Eyduran, E., & Tariq, M. M. (2018). Estimation of honey production in beekeeping enterprises from eastern part of Turkey through some data mining algorithms. *Pakistan Journal of Zoology*, *50*(6), 2199–2207. https://doi.org/10.17582/journal.pjz/2018.50.6.2199.2207

Altinkaya, Ş., & Yalçin, S. (2020). Some applications of generalized srivastava-attiya operator to the bi-concave functions. In *Miskolc Mathematical Notes* (Vol. 21, Issue 1, pp. 51–60). https://doi.org/10.18514/MMN.2020.2947

*Anaconda documentation*. (2020).

Anaya-Jaimes, J., García-Castro, A., & Gutiérrez-Carvajal, R. E. (2020). Performance of bottom-up visual attention models when compared in contextless and context awareness scenarios. *Proc.SPIE*, *11433*. https://doi.org/10.1117/12.2557135

Andika, L. A., Pratiwi, H., & Sulistijowati Handajani, S. (2020). Convolutional neural network modeling for classification of pulmonary tuberculosis disease. *Journal of Physics: Conference Series*, *1490*(1). https://doi.org/10.1088/1742-6596/1490/1/012020

Asriny, D. M., Rani, S., & Hidayatullah, A. F. (2020). Orange Fruit Images Classification using Convolutional Neural Networks. *IOP Conference Series: Materials Science and Engineering*, *803*(1). https://doi.org/10.1088/1757-899X/803/1/012020

Asteriadis, S., Karpouzis, K., & Kollias, S. (2014). Visual focus of attention in non-calibrated environments using gaze estimation. *International Journal of Computer Vision*, *107*(3), 293–316. https://doi.org/10.1007/s11263-013-0691-3

Barkley, R. A., & Wasserstein, J. (2000). ADHD and The Nature of Self-Control. *Journal of Cognitive Psychotherapy*, *14*(1), 111–113. https://doi.org/10.1891/0889-8391.14.1.111

Becker, M. W., Pashler, H., & Lubin, J. (2007). Object-intrinsic oddities draw early saccades.

*Journal of Experimental Psychology: Human Perception and Performance*, *33*(1), 20–30. https://doi.org/10.1037/0096-1523.33.1.20

Berg, T., & Belhumeur, P. N. (2012). Tom-vs-Pete classifiers and identity-preserving alignment for face verification. *BMVC 2012 - Electronic Proceedings of the British Machine Vision Conference 2012*, 1–11. https://doi.org/10.5244/C.26.129

Bitbrain. (2018). *Qué es la atención, tipos y alteraciones | Bitbrain*.

Bixler, R., & D'Mello, S. (2016). Automatic gaze-based user-independent detection of mind wandering during computerized reading. *User Modeling and User-Adapted Interaction*, *26*(1), 33–68. https://doi.org/10.1007/s11257-015-9167-1

Brickenkamp, R. (2012). d2, test de atención (adapt. Nicolás Seisdedos Cubero). *Tea*.

Burkov, A. (1997). The Hundred Page Machine Learning. *Computer*, *2005*(April), 414.

Burnik, U., Zaletelj, J., & Košir, A. (2018). Video-based learners' observed attention estimates for lecture learning gain evaluation. *Multimedia Tools and Applications*, *77*(13), 16903–16926. https://doi.org/10.1007/s11042-017-5259-8

Cao, X., Wipf, D., Wen, F., Duan, G., & Sun, J. (2013). A Practical Transfer Learning Algorithm for Face Verification. *2013 IEEE International Conference on Computer Vision*, 3208–3215. https://doi.org/10.1109/ICCV.2013.398

Chen, C. M., Wang, J. Y., & Yu, C. M. (2017). Assessing the attention levels of students by using a novel attention aware system based on brainwave signals. *British Journal of Educational Technology*, *48*(2), 348–369. https://doi.org/10.1111/bjet.12359

Chen, D, Cao, X., Wen, F., & Sun, J. (2013). Blessing of Dimensionality: High-Dimensional Feature and Its Efficient Compression for Face Verification. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 3025–3032. https://doi.org/10.1109/CVPR.2013.389

Chen, Dong, Cao, X., Wang, L., Wen, F., & Sun, J. (2012). *Eccv_2012_Bayesian.Dvi*. *1*, 1–14.

Chen, L. C., Yang, Y., Wang, J., Xu, W., & Yuille, A. L. (2016). Attention to Scale: Scale-Aware Semantic Image Segmentation. *Proceedings of the IEEE Computer Society Conference*

*on Computer Vision and Pattern Recognition*, *2016-Decem*, 3640–3649. https://doi.org/10.1109/CVPR.2016.396

Choi, K. S., Shin, J. S., Lee, J. J., Kim, Y. S., Kim, S. B., & Kim, C. W. (2005). In vitro trans-differentiation of rat mesenchymal cells into insulin-producing cells by rat pancreatic extract. *Biochemical and Biophysical Research Communications*, *330*(4), 1299–1305. https://doi.org/10.1016/j.bbrc.2005.03.111

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 1800–1807. https://doi.org/10.1109/CVPR.2017.195

Christ, S. E., Steiner, R. D., Grange, D. K., Abrams, R. A., & White, D. A. (2006). Inhibitory control in children with phenylketonuria. *Developmental Neuropsychology*, *30*(3), 845–864. https://doi.org/10.1207/s15326942dn3003_5

*Colaboratory – Google*. (2020).

*Conda documentation*. (2017).

Cowan, N. (1995). *Attention and Memory: An Integrated Framework - Oxford Scholarship*.

Dansilio, S. (2000). Introducción : modelos recientes de los procesos ejecutivos y desarrollo. *Fnc*.

Davidson, C. (2011). *Now You See It | Cathy N. Davidson*.

Davidson, M. C., Amso, D., Anderson, L. C., & Diamond, A. (2006). Development of cognitive control and executive functions from 4 to 13 years: Evidence from manipulations of memory, inhibition, and task switching. *Neuropsychologia*, *44*(11), 2037–2078. https://doi.org/10.1016/j.neuropsychologia.2006.02.006

Deng, L., Yang, M., Li, H., Li, T., Hu, B., & Wang, C. (2020). Restricted Deformable Convolution-Based Road Scene Semantic Segmentation Using Surround View Cameras. *IEEE Transactions on Intelligent Transportation Systems*, *21*(10), 4350–4362. https://doi.org/10.1109/TITS.2019.2939832

Diamond, A. (2013). Executive functions. *Annual Review of Psychology*, *64*, 135–168. https://doi.org/10.1146/annurev-psych-113011-143750

Díaz, A. (2017). Funciones básicas y atención - concentración en niños y niñas del 2° grado de una I.E estatal distrito de Huanchaco de la provincia de Trujillo. *Universidad Privada Antenor Orrego*, 1–45.

Dinesh, D. (2016). *Student Analytics for productive teaching / learning*. 97–102.

*Documentación de Python - 3.9.1*. (2020).

Dong, J., Chen, Q., Yan, S., & Yuille, A. (2014). Towards unified object detection and semantic segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8693 LNCS*(PART 5), 299–314. https://doi.org/10.1007/978-3-319-10602-1_20

Drobotdean. (2019). | *Freepik*. https://www.freepik.es/drobotdean

Ehrotra, S. U. C. M. (2018). *Introduction To Eeg- and Emotion Recognition*.

Epigeum. (2011). *Student attention over an hour*.

Fan, H., Mei, X., Prokhorov, D., & Ling, H. (2018). Multi-level contextual RNNs with attention model for scene labeling. *IEEE Transactions on Intelligent Transportation Systems*, *19*(11), 3475–3485. https://doi.org/10.1109/TITS.2017.2775628

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Educational Research*, *74*(1), 59–109. https://doi.org/10.3102/00346543074001059

Friedman, N. P., & Miyake, A. (2004). The Relations Among Inhibition and Interference Control Functions: A Latent-Variable Analysis. *Journal of Experimental Psychology: General*, *133*(1), 101–135. https://doi.org/10.1037/0096-3445.133.1.101

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202. https://doi.org/10.1007/BF00344251

Garcia-Perez, A., Gheriss, F., Bedford, D., Garcia-Perez, A., Gheriss, F., & Bedford, D. (2019). Measurement, Reliability, and Validity. *Designing and Tracking Knowledge Management Metrics*, 163–182. https://doi.org/10.1108/978-1-78973-723-320191012

Gazzaniga, M. S., Ivry, R. B., & Mangun, G. R. (2008). Cognitive Neuroscience: The Biology

Of Mind (excerpt). In *Cognitive Neuroscience The Biology Of Mind*.

Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, *2015 Inter*, 1440–1448. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 580–587. https://doi.org/10.1109/CVPR.2014.81

Gonzalez, T. F. (2007). Handbook of approximation algorithms and metaheuristics. *Handbook of Approximation Algorithms and Metaheuristics*, 1–1432. https://doi.org/10.1201/9781420010749

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *1*, 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947

Gupta, R., Raymond, J. E., & Vuilleumier, P. (2019). Priming by motivationally salient distractors produces hemispheric asymmetries in visual processing. *Psychological Research*, *83*(8), 1798–1807. https://doi.org/10.1007/s00426-018-1028-1

Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2014). Simultaneous detection and segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8695 LNCS*(PART 7), 297–312. https://doi.org/10.1007/978-3-319-10584-0_20

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. 1–18. http://arxiv.org/abs/1207.0580

Hosang, J., Benenson, R., & Schiele, B. (2014). How good are detection proposals, really?

*BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 1–25. https://doi.org/10.5244/c.28.24

Hughes, C., Dunn, J., & White, A. (1998). Trick or treat?: Uneven understanding of mind and emotion and executive dysfunction in “hard-to-manage” preschoolers. *Journal of Child Psychology and Psychiatry and Allied Disciplines*, *39*(7), 981–994. https://doi.org/10.1017/S0021963098003059

I. Goodfellow, Y. B. C. (2016). *Deep Learning, MIT Press*.

Imma Grau. (2018, May 30). *Las máquinas aprenden a diagnosticar*. https://www.fundacionisys.org/es/blogs/profesional/profesional/422-las-maquinas-aprenden-a-diagnosticar

James, W. (2007). *The Principles of Psychology*. Cosimo, Incorporated.

Joseph, S. (2016). Australian Literary Journalism and “Missing Voices”: How Helen Garner finally resolves this recurring ethical tension. *Journalism Practice*, *10*(6), 730–743. https://doi.org/10.1080/17512786.2015.1058180

*Kaggle Data Science Resources*. (2020).

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Li, F. F. (2014). Large-scale video classification with convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1725–1732. https://doi.org/10.1109/CVPR.2014.223

Kautz, T., Groh, B. H., Hannink, J., Jensen, U., Strubberg, H., & Eskofier, B. M. (2017). Activity recognition in beach volleyball using a Deep Convolutional Neural Network. *Data Mining and Knowledge Discovery*, *31*(6), 1678–1705. https://doi.org/10.1007/s10618-017-0495-0

Keane, P. J. (P. J., Caletka, A. F. (Anthony F. ., & Ph, R. O. (2008). *No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title*. *8*, 276.

*Keras*. (2020).

Khan, T., Johnston, K., & Ophoff, J. (2019). The Impact of an Augmented Reality Application

on Learning Motivation of Students. *Advances in Human-Computer Interaction*, *2019*. https://doi.org/10.1155/2019/7208494

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Kitsikidis, A., Dimitropoulos, K., Douka, S., & Grammalidis, N. (2014). Dance analysis using multiple Kinect sensors. *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, *2*, 789–795.

Korzeniowski, C. (2018). Las funciones ejecutivas en el estudiante: Su compresión e implementación desde el salón de clases. *Dirección General de Escuelas*.

Krizhevsky, B. A., Sutskever, I., & Hinton, G. E. (2012). Cnn实际训练的. *Communications of the ACM*, *60*(6), 84–90.

Langton, S. R. H., & Bruce, V. (1999). Reflexive visual orienting in response to the social attention of others. *Visual Cognition*, *6*(5), 541–567. https://doi.org/10.1080/135062899394939

Lateef, F., & Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, *338*, 321–348. https://doi.org/10.1016/j.neucom.2019.02.003

Lavie, N., Ro, T., & Russell, C. (2003). The role of perceptual load in processing distractor faces. *Psychological Science*, *14*(5), 510–515. https://doi.org/10.1111/1467-9280.03453

LeCun, Yann, Bengio, Y. (2017). *Las funciones ejecutivas del estudiante: Mejorar la atención, la memoria, la organización y otras funciones para facilitar el aprendizaje*. Narcea Ediciones.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

Lecun, Y, Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to

document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Lecun, Yann, Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lemonnier, S., Désiré, L., Brémond, R., & Baccino, T. (2020). Drivers' visual attention: A field study at intersections. *Transportation Research Part F: Traffic Psychology and Behaviour*, *69*, 206–221. https://doi.org/10.1016/j.trf.2020.01.012

Leon B. (2008). Atención plena y rendimiento académico en estudiantes de enseñanza secundaria. *European Journal of Education and Psychology*, *1*(3).

Li, G., Huang, Y., Chen, Z., Chesser, G. D., Purswell, J. L., Linhoss, J., & Zhao, Y. (2021). Practices and applications of convolutional neural network-based computer vision systems in animal farming: A review. *Sensors*, *21*(4), 1–42. https://doi.org/10.3390/s21041492

Li, R., Zhang, Y., Niu, D., Yang, G., Zafar, N., Zhang, C., & Zhao, X. (2021). PointVGG: Graph convolutional network with progressive aggregating features on point clouds. *Neurocomputing*, *429*, 187–198. https://doi.org/10.1016/j.neucom.2020.10.086

Lin, M., Chen, Q., & Yan, S. (2014). Network in network. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 1–10.

Madsen, J., Júlio, S. U., Gucik, P. J., Steinberg, R., & Parra, L. C. (2021). Synchronized eye movements predict test scores in online video education. *Proceedings of the National Academy of Sciences*, *118*(5). https://doi.org/10.1073/pnas.2016980118

Maeda Gutierrez, V. (2019). *Comparación De Arquitecturas De Redes Neuronales Convolucionales Para La Clasificación De Enfermedades En Tomate*. *September*, 77.

Mamalet, F., & Garcia, C. (2012). Simplifying ConvNets for fast learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7553 LNCS*(PART 2), 58–65. https://doi.org/10.1007/978-3-642-33266-1_8

Massiris, M., Delrieux, C., & Fernández, J. Á. (2020). *Detección de equipos de protección*

*personal mediante red neuronal convolucional YOLO*. 1022–1029. https://doi.org/10.17979/spudc.9788497497565.1022

Melinte, D. O., Travediu, A. M., & Dumitriu, D. N. (2020). Deep convolutional neural networks object detector for real-time waste identification. *Applied Sciences (Switzerland)*, *10*(20), 1–18. https://doi.org/10.3390/app10207301

Mills, C., Bixler, R., Wang, X., & D'Mello, S. K. (2016). Automatic gaze-based detection of mind wandering during narrative film comprehension. *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*, 30–37.

Miranda, J. C. (2018). Clasificación automática de naranjas por tamaño y por defectos utilizando técnicas de visión por computadora. *Facultad Politécnica, Universidad Nacional de Asunción*, *March*.

Monkaresi, H., Bosch, N., Calvo, R. A., & D'Mello, S. K. (2017). Automated Detection of Engagement Using Video-Based Estimation of Facial Expressions and Heart Rate. *IEEE Transactions on Affective Computing*, *8*(1), 15–28. https://doi.org/10.1109/TAFFC.2016.2515084

Mora, F. (2014). *E U Ro E D U C a C I Ó N*. *3*(2), 259–262.

Moraine, P. (2014). *Las funciones ejecutivas del estudiante: Mejorar la atención, la memoria, la organización y otras funciones para facilitar el aprendizaje*. Narcea.

Mukherjee, S. S., & Robertson, N. M. (2015). Deep Head Pose: Gaze-Direction Estimation in Multimodal Video. *IEEE Transactions on Multimedia*, *17*(11), 2094–2107. https://doi.org/10.1109/TMM.2015.2482819

*OpenCV*. (2020).

Otting, G., Reson, J. M., & Cavanagh, P. (1992). *Attention-Based Motion Perception*. *257*(September), 1563–1565.

Paletta, L., Santner, K., Fritz, G., Hofmann, A., Lodron, G., Thallinger, G., & Mayer, H. (2013). *FACTS - A Computer Vision System for 3D Recovery and Semantic Mapping of Human Factors BT - Computer Vision Systems* (M. Chen, B. Leibe, & B. Neumann (Eds.); pp. 62–72). Springer Berlin Heidelberg.

Papoutsaki, A., Daskalova, N., Sangkloy, P., Huang, J., Laskey, J., & Hays, J. (2016). WebGazer: Scalable webcam eye tracking using user interactions. *IJCAI International Joint Conference on Artificial Intelligence*, *2016-Janua*, 3839–3845.

Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). *ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation*. 1–10.

Paula, J. de, & Ávila, R. de. (2011). Assessing Processing Speed and Executive Functions in Low Educated Older Adults: the Use of the Five Digit Test in Patients. *Clinical Neuropsychiatry*, 339–346.

Piaget, J. (2005). *Teligencia Y Afectividad*.

Pohlen, T., Hermans, A., Mathias, M., & Leibe, B. (2017). Full-resolution residual networks for semantic segmentation in street scenes. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 3309–3318. https://doi.org/10.1109/CVPR.2017.353

Posner, G. J., Strike, K. A., Hewson, P. W., & Gertzog, W. A. (1982). Accommodation of a scientific conception: Toward a theory of conceptual change. *Science Education*, *66*(2), 211–227. https://doi.org/10.1002/sce.3730660207

Progrmador Clic. (2019). *El modelo clásico de CNN: LeNet - programador clic*. https://programmerclick.com/article/35441799811/

Psychologie, Z., Johann, V. Von, & Barth, A. (1910). Zeitschrift Für Psychologie. *Mind*, *XIX*(1), 143–144. https://doi.org/10.1093/mind/xix.1.143

Pulgar, F. J., Rivera, A. J., Charte, F., & Jesus, J. (2018). *Análisis del impacto de datos desbalanceados en el rendimiento predictivo de redes neuronales convolucionales*. 1213–1218.

Raymond, J. E., & O'Brien, J. L. (2009). Selective visual attention and motivation: The consequences of value learning in an attentional blink task. *Psychological Science*, *20*(8), 981–988. https://doi.org/10.1111/j.1467-9280.2009.02391.x

Ren, S., He, K., Girshick, R., & Sun, J. (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. https://doi.org/10.1109/TPAMI.2016.2577031

Ro, T., Russell, C., & Lavie, N. (2001). Changing faces: A detection advantage in the flicker paradigm. *Psychological Science*, *12*(1), 94–99. https://doi.org/10.1111/1467-9280.00317

Rodríguez Artacho, M. (2011). *Diferencias en Flexibilidad Cognitiva medidas mediante el Paradigma de Cambio de Tarea en Sinestesia y Esclerosis*. 1–195.

Rosselli, M., Jurado, M. B., & Matute, E. (2008). Las Funciones Ejecutivas a través de la Vida. *Revista Neuropsicología, Neuropsiquiatría y Neurociencias*, *8*(1), 23–46.

Ruth Doherty, B., Patai, E. Z., Duta, M., Nobre, A. C., & Scerif, G. (2017). The functional consequences of social distraction: Attention and memory for complex scenes. *Cognition*, *158*, 215–223. https://doi.org/10.1016/j.cognition.2016.10.015

Saeed, A., Al-Hamadi, A., & Ghoneim, A. (2015). Head pose estimation on top of haar-like face detection: A study using the Kinect sensor. *Sensors (Switzerland)*, *15*(9), 20945–20966. https://doi.org/10.3390/s150920945

Sáez, D. C., Dpto, Z., & Psicobiología, : (2016). *Datos normativos de la prueba ANT en muestra española*.

Santalla, Z. (2017). *Capítulo 4 : El mecanismo atencional*. *November*.

*SCRUM*. (2021). https://www.scrum.org/resources/scrum-guide

Shervine A. (2019). *CS 230 - Convolutional Neural Networks Cheatsheet*. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

*Spyder 4 documentation*. (2020).

Stuss, D. T., & Levine, B. (2002). Adult clinical neuropsychology: Lessons from studies of the frontal lobes. *Annual Review of Psychology*, *53*, 401–433. https://doi.org/10.1146/annurev.psych.53.100901.135220

Suah, F. B. M. (2017). Preparation and characterization of a novel Co(II) optode based on polymer inclusion membrane. *Analytical Chemistry Research*, *12*, 40–46. https://doi.org/10.1016/j.ancr.2017.02.001

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 4278–4284.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

T. S, A., & Guddeti, R. M. R. (2020). Automatic detection of students' affective states in classroom environment using hybrid convolutional neural networks. *Education and Information Technologies*, *25*(2), 1387–1415. https://doi.org/10.1007/s10639-019-10004-6

*TensorFlow Aprendizaje Automático*. (2020).

Torres-Carrion, P. V., Gonzalez-Gonzalez, C. S., Aciar, S., & Rodriguez-Morales, G. (2018). Methodology for systematic literature review applied to engineering and education. *IEEE Global Engineering Education Conference, EDUCON*, *2018-April*, 1364–1373. https://doi.org/10.1109/EDUCON.2018.8363388

Tygert, M., Bruna, J., Chintala, S., LeCun, Y., Piantino, S., & Szlam, A. (2016). A mathematical motivation for complex-valued convolutional networks. *Neural Computation*, *28*(5), 815–825. https://doi.org/10.1162/NECO_a_00824

Uijlings, J. R. R., Van De Sande, K. E. A., Gevers, T., & Smeulders, A. W. M. (2013). Selective search for object recognition. *International Journal of Computer Vision*, *104*(2), 154–171. https://doi.org/10.1007/s11263-013-0620-5

Valada, A., Vertens, J., Dhall, A., & Burgard, W. (2017). AdapNet: Adaptive semantic segmentation in adverse environmental conditions. *Proceedings - IEEE International Conference on Robotics and Automation*, 4644–4651.

https://doi.org/10.1109/ICRA.2017.7989540

Vedaldi, A., Lux, M., & Bertini, M. (2015). MatConvNet. *ACM SIGMultimedia Records*, *10*(1), 9–9. https://doi.org/10.1145/3210241.3210250

Veit, A., Wilber, M., & Belongie, S. (2016). Residual networks behave like ensembles of relatively shallow networks. *Advances in Neural Information Processing Systems*, 550–558.

Verbruggen, F., Liefooghe, B., & Vandierendonck, A. (2006). The effect of interference in the early processing stages on response inhibition in the stop signal task. *Quarterly Journal of Experimental Psychology*, *59*(1), 190–203. https://doi.org/10.1080/17470210500151386

Verstraten, F. A. J., Cavanagh, P., & Labianca, A. T. (2000). Limits of attentive tracking reveal temporal properties of attention. *Vision Research*, *40*(26), 3651–3664. https://doi.org/10.1016/S0042-6989(00)00213-3

Verstraten, F. A. J., Hooge, I. T. C., Culham, J., & Van Wezel, R. J. A. (2001). Systematic eye movements do not account for the perception of motion during attentive tracking. *Vision Research*, *41*(25–26), 3505–3511. https://doi.org/10.1016/S0042-6989(01)00205-X

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, *1*, I–I. https://doi.org/10.1109/CVPR.2001.990517

Visin, F., Kastner, K., Cho, K., Matteucci, M., Courville, A., & Bengio, Y. (2015). *ReNet: A Recurrent Neural Network Based Alternative to Convolutional Networks*. 1–9.

Visin, F., Romero, A., Cho, K., Matteucci, M., Ciccone, M., Kastner, K., Bengio, Y., & Courville, A. (2016). ReSeg: A Recurrent Neural Network-Based Model for Semantic Segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 426–433. https://doi.org/10.1109/CVPRW.2016.60

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, *2018*.

https://doi.org/10.1155/2018/7068349

Voulodimos, A. S., Doulamis, N. D., Kosmopoulos, D. I., & Varvarigou, T. A. (2012). IMPROVING MULTI-CAMERA ACTIVITY RECOGNITION BY EMPLOYING NEURAL NETWORK BASED READJUSTMENT. *Applied Artificial Intelligence*, *26*(1–2), 97–118. https://doi.org/10.1080/08839514.2012.629540

Voulodimos, A. S., Kosmopoulos, D. I., Doulamis, N. D., & Varvarigou, T. A. (2014). A top-down event-driven approach for concurrent activity recognition. *Multimedia Tools and Applications*, *69*(2), 293–311. https://doi.org/10.1007/s11042-012-0993-4

Vuilleumier, P, Armony, J. L., Driver, J., & Dolan, R. J. (2001). Effects of attention and emotion on face processing in the human brain. *Neuron*, *30*, 829–841.

Vuilleumier, Patrik. (2000). Faces call for attention: Evidence from patients with visual extinction. *Neuropsychologia*, *38*(5), 693–700. https://doi.org/10.1016/S0028-3932(99)00107-4

Wang, W., Yang, J., Xiao, J., Li, S., & Zhou, D. (2015). *Face Recognition Based on Deep Learning BT - Human Centered Computing* (Q. Zu, B. Hu, N. Gu, & S. Seng (Eds.); pp. 812–820). Springer International Publishing.

Wetzel, N., Scharf, F., & Widmann, A. (2019). Can't Ignore—Distraction by Task-Irrelevant Sounds in Early and Middle Childhood. *Child Development*, *90*(6), e819–e830. https://doi.org/10.1111/cdev.13109

White, B. W., & Rosenblatt, F. (1963). Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. *The American Journal of Psychology*, *76*(4), 705. https://doi.org/10.2307/1419730

Whitehill, J., Serpell, Z., Lin, Y. C., Foster, A., & Movellan, J. R. (2014). The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Transactions on Affective Computing*, *5*(1), 86–98. https://doi.org/10.1109/TAFFC.2014.2316163

Wright, A., & Diamond, A. (2014). An effect of inhibitory load in children while keeping working memory load constant. *Frontiers in Psychology*, *5*(MAR), 1–9.

https://doi.org/10.3389/fpsyg.2014.00213

Wu, Z., Shen, C., & van den Hengel, A. (2019). Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *Pattern Recognition*, *90*, 119–133. https://doi.org/10.1016/j.patcog.2019.01.006

Xia, H., Shao, L., Zhao, J., & Cao, Z. (2021). Improved deep learning techniques in gravitational-wave data analysis. *Physical Review D*, *103*(2), 1–13. https://doi.org/10.1103/PhysRevD.103.024040

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 5987–5995. https://doi.org/10.1109/CVPR.2017.634

XinZhe Jin. (2020). *Universidad Politécnica de Madrid Trabajo Fin de Máster Seguimiento ocular para análisis del comportamiento mediante ANN*.

Xu, D., Offner, S. S. R., Gutermuth, R., & van Oort, C. (2020). Application of convolutional neural networks to identify stellar feedback bubbles in CO emission. *ArXiv*. https://doi.org/10.3847/1538-4357/ab6607

Yin, S., Wang, Y., & Yang, Y. H. (2021). Attentive U-recurrent encoder-decoder network for image dehazing. *Neurocomputing*, *437*, 143–156. https://doi.org/10.1016/j.neucom.2020.12.081

Yoldi, A. (2015). Las Funciones Ejecutivas: Hacia Prácticas Educativas Que Potencien Su Desarrollo. *Páginas de Educación*, *8*(1), 72–98. https://doi.org/10.22235/pe.v8i1.497

Yoldi, A. (2019). *PUBLICADO EN REVISTA ARBITRADA : Páginas de Educación N ° 8 , pp . POTENCIEN SU DESARROLLO . THE EXECUTIVE FUNCTIONS , TOWARDS EDUCATIVE PRACTICES THAT POTENCE THEIR DEVELOPMENT . May 2015*.

Zaletelj, J., & Košir, A. (2017). Predicting students' attention in the classroom from Kinect facial and body features. *Eurasip Journal on Image and Video Processing*, *2017*(1). https://doi.org/10.1186/s13640-017-0228-8

Zhang, X., Yao, L., Huang, C., Wang, S., Tan, M., Long, G., & Wang, C. (2018). Multi-modality

sensor data classification with selective attention. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 3111–3117. https://doi.org/10.24963/ijcai.2018/432

Zhou, F. Y., Jin, L. P., & Dong, J. (2017). Review of Convolutional Neural Network. *Jisuanji Xuebao/Chinese Journal of Computers*, *40*(6), 1229–1251. https://doi.org/10.11897/SP.J.1016.2017.01229

Zhu, Y., Urtasun, R., Salakhutdinov, R., & Fidler, S. (2015). SegDeepM: Exploiting segmentation and context in deep neural networks for object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June*, 4703–4711. https://doi.org/10.1109/CVPR.2015.7299102

Zuñiga, M. (2007). *Análisis de la Propuesta de Atención para Alumnos Sobresalientes en el Estado de Hidalgo*.

# Apéndice

Apéndice 1: Modelo de RNC con Transfer Learning.

La primera opción que se tomó en cuenta para el desarrollo de la arquitectura de la RNC fue con el uso de la técnica Transfer Learning, a continuación, se detallan cada uno de los pasos necesarios para la creación de esta.

Empezamos por la importación de cada una de las librerías necesarias para cada una de las etapas que se utilizará.

```
# importar los paquetes necearios
from tensorflow.keras.preprocessing.image import ImageDataGenerator, im
g_to_array, load_img
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D, Dropout, Flatten,
 Dense, Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.utils import to_categorical
#from tensorflow_hub import hub
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.decomposition import PCA
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os
from google.colab import drive
```

El siguiente paso es definir las variables como la taza de aprendizaje, las épocas y el tamaño del lote.

```
# inicializar la tasa de aprendizaje inicial, el número de épocas para
entrenar,
# y tamaño de lote
INIT_LR = 1e-4
EPOCHS = 100
BS = 256
```

A continuación, se carga el dataset previamente subido a Drive.

```
#montando Google Drive para el dataset
drive.mount('/content/drive')

DIRECTORY = r"/content/drive/My Drive/dataset"
CATEGORIES = ["atento", "desatento"]
```

```
Mounted at /content/drive
```

Se carga las inicializa las imágenes y se las convierte en clase según las categorías, en este caso atento y desatento.

```
"""toma la lista de imágenes en nuestro directorio de conjuntos de dato
s desde google drive, luego inicializa
la lista de datos (es decir, imágenes) e imágenes de clase"""

print("[INFO] cargando imagenes...")
data = []
labels = []

for category in CATEGORIES:
    path = os.path.join(DIRECTORY, category)
    for img in os.listdir(path):
      img_path = os.path.join(path, img)
      image = load_img(img_path, target_size=(224, 224))
      image = img_to_array(image)
      image = preprocess_input(image)

      data.append(image)
      labels.append(category)
print("imagenes cargadas")
```

```
[INFO] cargando imagenes...
imagenes cargadas
```

El siguiente paso es la codificación one-hot en etiquetas, aquí se define cada uno de los dataset tanto para la etapa de entrenamiento como para la de testeo.

```
# realizar codificación one-hot en las etiquetas
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

data = np.array(data, dtype="float32")
labels = np.array(labels)

(trainX, testX, trainY, testY) = train_test_split(data, labels,
    test_size=0.1, stratify=labels, random_state=42)
```

Ahora es la etapa del preprocesamiento de las imágenes, aquí se modifica los hiperparámetros.

```
# construya el generador de imágenes de entrenamiento para el aumento d
e datos
aug = ImageDataGenerator(
    #rotation_range=0.1,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.4,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest")
```

El siguiente paso es cargar la red neuronal que ya está entrenada y utilizaremos ese conocimiento aprendido para aplicar la técnica Transfer Learning.

```
# cargue la red MobileNetV2, asegurándose de que los conjuntos de capas
 FC principales estén detenidos
baseModel=MobileNetV2(
    input_shape=(224,224,3),
    alpha=1.0,
    include_top=False,
    weights="imagenet",
    input_tensor=None,
    pooling=None,
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224_no_top.h5
9412608/9406464 [==============================] - 0s 0us/step
```

El siguiente paso es la creación de la red neuronal base y modificar las últimas capas de entrenamiento.

```
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.2)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
model = Model(inputs=baseModel.input, outputs=headModel)
```

Se hace un recorrido por cada una de las capas para verificar que se incluye toda la arquitectura de la Red Neuronal descargada y se cambia el parámetro de entreamiento de True por False.

```
# loop over all layers in the base model and freeze them so they will
# *not* be updated during the first training process
for layer in baseModel.layers:
  layer.trainable = False
```

Se procede a la compilación del modelo

```
# construya el modelo
print("[INFO] compilando el modelo...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / 60)
model.compile(loss="binary_crossentropy", optimizer=opt,
  metrics=["accuracy"])
```

```
[INFO] compilando el modelo...
```

Se coloca el modelo base como cabecera, este proceso convierte la red neuronal que se entrenará con nuestras capas modificadas al final.

```
# coloque el modelo FC del cabezal encima del modelo base (esto se conv
ertirá el modelo real que entrenaremos)
model = Model(inputs=baseModel.input, outputs=headModel)
```

Se realiza un recorrido para conocer si realmente se han incluido las últimas capas y se congela las capas originales del modelo descargado.

```
# recorrer todas las capas del modelo base y congelarlas para que
# no se actualice durante el primer proceso de entrenamiento
for layer in baseModel.layers:
  layer.trainable = False
```

Se compila el modelo de Red Neuronal Convolucional

```
# compilando nuestro modelo
print("[INFO] compilando el modelo...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
  metrics=["accuracy"])
```

```
[INFO] compilando el modelo...
```

Se procede a las etapas de entrenamiento y testeo.

```
# entrenado la cabecera de la red
print("[INFO] entrenando la cabecera...")
H = model.fit(
    aug.flow(trainX, trainY, batch_size=512),
    steps_per_epoch=len(trainX) // 512,
    validation_data=(testX, testY),
    validation_steps=len(testX) // 512,
    epochs=EPOCHS)
```

```
[INFO] entrenando la cabecera...
Epoch 1/100
7/7 [==============================] - 158s 20s/step - loss: 0.8100 -
accuracy: 0.5126
Epoch 2/100
7/7 [==============================] - 152s 20s/step - loss: 0.7321 -
accuracy: 0.5326
Epoch 3/100
7/7 [==============================] - 169s 23s/step - loss: 0.7036 -
accuracy: 0.5848
Epoch 4/100
7/7 [==============================] - 159s 26s/step - loss: 0.6745 -
accuracy: 0.5982
Epoch 5/100
7/7 [==============================] - 171s 24s/step - loss: 0.6607 -
accuracy: 0.6222
Epoch 6/100
7/7 [==============================] - 172s 24s/step - loss: 0.6506 -
accuracy: 0.6427
Epoch 7/100
7/7 [==============================] - 170s 23s/step - loss: 0.6430 -
accuracy: 0.6436
Epoch 8/100
7/7 [==============================] - 179s 25s/step - loss: 0.6527 -
accuracy: 0.6362
Epoch 9/100
7/7 [==============================] - 147s 20s/step - loss: 0.6290 -
accuracy: 0.6587
Epoch 10/100
7/7 [==============================] - 151s 20s/step - loss: 0.6245 -
accuracy: 0.6595
Epoch 20/100
7/7 [==============================] - 159s 26s/step - loss: 0.5952 -
accuracy: 0.6902
Epoch 21/100
7/7 [==============================] - 152s 20s/step - loss: 0.5938 -
accuracy: 0.6799
Epoch 22/100
7/7 [==============================] - 155s 21s/step - loss: 0.5915 -
accuracy: 0.6864
Epoch 23/100
7/7 [==============================] - 174s 24s/step - loss: 0.5811 -
accuracy: 0.6862
Epoch 24/100
7/7 [==============================] - 151s 25s/step - loss: 0.5655 -
accuracy: 0.7054
Epoch 25/100
7/7 [==============================] - 154s 25s/step - loss: 0.5998 -
accuracy: 0.6500
Epoch 26/100
7/7 [==============================] - 152s 25s/step - loss: 0.5939 -
accuracy: 0.6794
Epoch 27/100
7/7 [==============================] - 159s 22s/step - loss: 0.5823 -
accuracy: 0.6953
Epoch 28/100
7/7 [==============================] - 154s 21s/step - loss: 0.5713 -
accuracy: 0.7080
Epoch 29/100
7/7 [==============================] - 156s 21s/step - loss: 0.5692 -
accuracy: 0.7025
```

```
Epoch 30/100
7/7 [==============================] - 150s 20s/step - loss: 0.5807 -
accuracy: 0.6853
Epoch 31/100
7/7 [==============================] - 163s 27s/step - loss: 0.5662 -
accuracy: 0.6966
Epoch 32/100
7/7 [==============================] - 153s 20s/step - loss: 0.5795 -
accuracy: 0.7111
Epoch 33/100
7/7 [==============================] - 154s 21s/step - loss: 0.5639 -
accuracy: 0.7088
Epoch 34/100
7/7 [==============================] - 172s 24s/step - loss: 0.5731 -
accuracy: 0.7030
Epoch 35/100
7/7 [==============================] - 164s 21s/step - loss: 0.5649 -
accuracy: 0.7043
Epoch 36/100
7/7 [==============================] - 153s 21s/step - loss: 0.5695 -
accuracy: 0.7066
Epoch 37/100
7/7 [==============================] - 149s 20s/step - loss: 0.5729 -
accuracy: 0.6996
Epoch 38/100
7/7 [==============================] - 154s 25s/step - loss: 0.5691 -
accuracy: 0.7068
Epoch 39/100
7/7 [==============================] - 186s 24s/step - loss: 0.5638 -
accuracy: 0.7159
Epoch 50/100
7/7 [==============================] - 184s 26s/step - loss: 0.5532 -
accuracy: 0.7104
Epoch 51/100
7/7 [==============================] - 155s 21s/step - loss: 0.5605 -
accuracy: 0.7141
Epoch 52/100
7/7 [==============================] - 177s 24s/step - loss: 0.5651 -
accuracy: 0.6994
Epoch 53/100
7/7 [==============================] - 150s 20s/step - loss: 0.5539 -
accuracy: 0.7166
Epoch 54/100
7/7 [==============================] - 185s 25s/step - loss: 0.5628 -
accuracy: 0.7134
Epoch 55/100
7/7 [==============================] - 154s 21s/step - loss: 0.5452 -
accuracy: 0.7237
Epoch 56/100
7/7 [==============================] - 174s 24s/step - loss: 0.5582 -
accuracy: 0.6991
Epoch 57/100
7/7 [==============================] - 169s 23s/step - loss: 0.5487 -
accuracy: 0.7218
Epoch 58/100
7/7 [==============================] - 154s 21s/step - loss: 0.5593 -
accuracy: 0.7075
Epoch 59/100
7/7 [==============================] - 173s 24s/step - loss: 0.5495 -
accuracy: 0.7184
Epoch 60/100
```

```
7/7 [==============================] - 174s 24s/step - loss: 0.5495 -
accuracy: 0.7252
Epoch 90/100
7/7 [==============================] - 154s 21s/step - loss: 0.5429 -
accuracy: 0.7128
Epoch 91/100
7/7 [==============================] - 154s 25s/step - loss: 0.5481 -
accuracy: 0.7251
Epoch 92/100
7/7 [==============================] - 166s 23s/step - loss: 0.5436 -
accuracy: 0.7206
Epoch 93/100
7/7 [==============================] - 155s 21s/step - loss: 0.5360 -
accuracy: 0.7199
Epoch 94/100
7/7 [==============================] - 155s 21s/step - loss: 0.5534 -
accuracy: 0.7114
Epoch 95/100
7/7 [==============================] - 149s 20s/step - loss: 0.5311 -
accuracy: 0.7304
Epoch 96/100
7/7 [==============================] - 166s 23s/step - loss: 0.5362 -
accuracy: 0.7326
Epoch 97/100
7/7 [==============================] - 154s 21s/step - loss: 0.5425 -
accuracy: 0.7170
Epoch 98/100
7/7 [==============================] - 173s 24s/step - loss: 0.5286 -
accuracy: 0.7286
Epoch 99/100
7/7 [==============================] - 153s 21s/step - loss: 0.5320 -
accuracy: 0.7267
Epoch 100/100
7/7 [==============================] - 170s 23s/step - loss: 0.5314 -
accuracy: 0.7380
```

Se procede a hacer algunas predicciones con el set de entrenamiento

```
# hacer predicciones con el set de entrenamiento
print("[INFO] evaluando la red...")
predIdxs = model.predict(testX, batch_size=512)
```

Se etiqueta la mayor probabilidad.

```
# para cada imagen en el conjunto de prueba, necesitamos encontrar el í
ndice de la
# etiqueta con la mayor probabilidad que se predice correspondiente
predIdxs = np.argmax(predIdxs, axis=1)
# mostrar un informe de clasificación con un formato agradable
print(classification_report(testY.argmax(axis=1), predIdxs,
  target_names=lb.classes_))
```

Se puede observar que el aprendizaje no supera el 69% en atento y en desatento llega a 0.76%

```
              precision    recall  f1-score   support

      atento       0.69      0.80      0.74       201
   desatento       0.76      0.64      0.70       200
```

```
              precision    recall  f1-score   support

    accuracy                         0.72       401
   macro avg       0.73      0.72      0.72       401
weighted avg       0.73      0.72      0.72       401
```

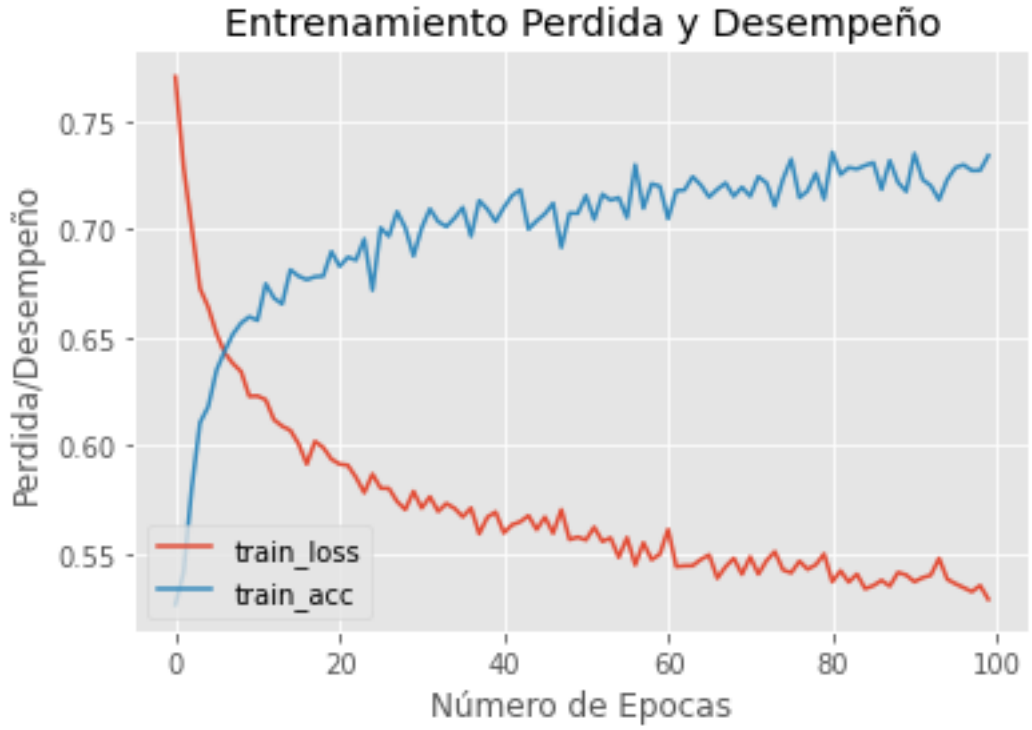
Se almacena el modelo para analizar con tecnología de visión artificial.

```
# serializar el modelo en disco
print("[INFO] guardando el modelo detector de la atención...")
model.save("attention_detector.model", save_format="h5")
```

```
[INFO] guardando el modelo detector de la atención...
```

Se generan las gráficas de entrenamiento.

```
# graficar la pérdida y la precisión del entrenamiento
N = EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
#plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")
#plt.plot(np.arange(0, N), H.history["val_accuracy"], label="val_acc")
plt.title("Entrenamiento Perdida y Desempeño")
plt.xlabel("Número de Epocas")
plt.ylabel("Perdida/Desempeño")
plt.legend(loc="lower left")
plt.savefig("plot.png")
```



Apéndice 2: RNC sin Transfer Learning.

En este apéndice se puede observar las distintas etapas que se toma en cuenta para desarrollar la arquitectura de la RNC sin Transfer Learning.

```
#import tensorflow
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense
from keras.layers import Dropout,Input
from keras.layers import Flatten
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import BatchNormalization
from keras.layers import Activation
from keras.models import Model,Sequential
from keras.optimizers import Adam
from keras.preprocessing.image import load_img,img_to_array
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import os
import zipfile
from google.colab import drive
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
```

Se procede a montar el dataset comprimido en Google Drive

```
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Se descomprime el dataset

```
local_zip = 'drive/My Drive/Attention/attention.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp/attention')
zip_ref.close()
```

Se crean las categorías de Atento y Desatento

```
base_dir = '/tmp/attention/data/'
train_dir = os.path.join(base_dir)

train_Atento = os.path.join('/tmp/attention/data/atento')
train_Desatento = os.path.join('/tmp/attention/data/desatento')
```

Se procede a verificar las imágenes cargadas en las categorías de Atento y Desatento

```
train_Atento_names = os.listdir(train_Atento)
print(train_Atento_names[:10])

train_Desatento_names = os.listdir(train_Desatento)
print(train_Desatento_names[:10])
```

```
['attention15649.jpg', 'attention15861.jpg', 'attention16557.jpg',

'attention14735.jpg', 'attention13771.jpg', 'attention16342.jpg',

'attention15177.jpg', 'attention16404.jpg', 'attention16347.jpg',

'attention15525.jpg']

['attention32349.jpg', 'attention32157.jpg', 'attention36607.jpg',

'attention31553.jpg', 'attention31414.jpg', 'attention33667.jpg',

'attention34068.jpg', 'attention33976.jpg', 'attention33586.jpg',

'attention31682.jpg']
```

Se muestra el número total de imágenes en ambas categorías

```
print('total training Atento images :', len(os.listdir(train_Atento)))
print('total training Desatento images :', len(os.listdir(train_Desaten
to)))
```

```
total training Atento images : 2002
total training Desatento images : 2001
```

Se verifica las categorías cargadas.

```
for attention in os.listdir(base_dir):
    #['Atento', 'Desatento']
    #print(str(len(os.listdir(base_dir))))
    print(str(len(os.listdir(base_dir + attention))) + " " + attention
+" images")
```

```
2001 desatento images
2002 atento images
```

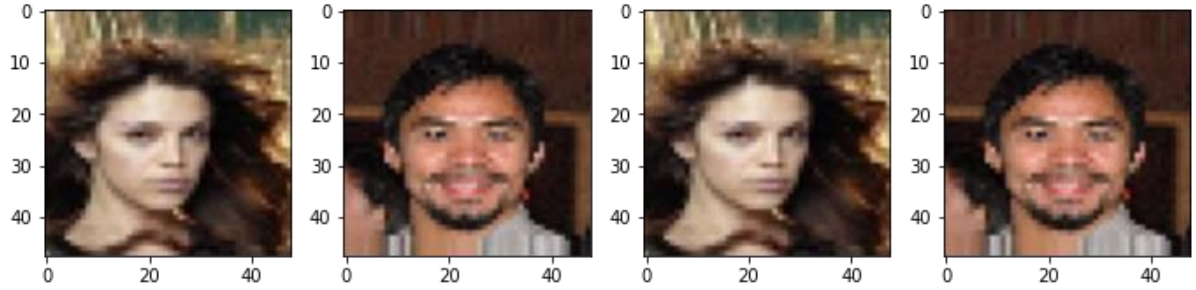
Se analiza las imágenes en ambos dataset

```
# size of the image: 48*48 pixels
pic_size = 48

plt.figure(0, figsize=(12,20))
cpt = 0

for expression in os.listdir(base_dir):
    for i in range(1,3):
        cpt = cpt + 1
        plt.subplot(7,5,cpt)
        img = load_img(base_dir + attention + "/" +os.listdir(base_dir
+ attention)[i], target_size=(pic_size, pic_size))
        plt.imshow(img, cmap="gray")
```

```
plt.tight_layout()
plt.show()
```



Se realiza el preprocesamiento de la data

```
# number of images to feed into the NN for every batch
batch_size = 512

datagen_train = ImageDataGenerator()
#datagen_validation = ImageDataGenerator()
train_generator = datagen_train.flow_from_directory(base_dir,
                  target_size=(pic_size,pic_size),
                  color_mode="grayscale",
                  batch_size=batch_size,
                  class_mode='categorical',
                  shuffle=True)
```

```
Found 4003 images belonging to 2 classes.
```

Se genera la arquitectura de la RNC y compilación del modelo.

```
#No. of possible label values
nb_classes = 2
#Initialising the CNN
model = Sequential()
#1
model.add(Conv2D(64,(3,3), padding = 'same', input_shape = (48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#2
model.add(Conv2D(128,(5,5), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#3
```

```
model.add(Conv2D(512,(3,3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#4
model.add(Conv2D(64,(3,3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#Flattening
model.add(Flatten())

#Full connected layer 1
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

#Full connected layer 2
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(nb_classes, activation = 'softmax'))

opt = Adam(lr = 0.0004)

model.compile(optimizer = opt, loss = 'categorical_crossentropy', metri
cs = ['accuracy'])
```

Se genera la etapa de entrenamiento y validación.

```
# number of epochs to train the NN
epochs = 120
from keras.callbacks import ModelCheckpoint

#checkpoint = ModelCheckpoint("model_weights.h5", monitor='val_acc', ve
rbose=1, save_best_only=True, mode='max')
#callbacks_list = [checkpoint]
#steps per epoch = 28273/128 = 220
history = model.fit_generator(generator=train_generator,
                              steps_per_epoch=train_generator.n//trai
n_generator.batch_size,
```

```
                                epochs = epochs,
                                verbose = 1)
```

```
Epoch 1/120
7/7 [==============================] - 112s 16s/step - loss: 0.8741 -
accuracy: 0.5278
Epoch 2/120
7/7 [==============================] - 111s 16s/step - loss: 0.7723 -
accuracy: 0.5584
Epoch 3/120
7/7 [==============================] - 109s 15s/step - loss: 0.7414 -
accuracy: 0.6111
Epoch 4/120
7/7 [==============================] - 109s 15s/step - loss: 0.7264 -
accuracy: 0.6083
Epoch 5/120
7/7 [==============================] - 108s 15s/step - loss: 0.6870 -
accuracy: 0.6363
Epoch 6/120
7/7 [==============================] - 117s 17s/step - loss: 0.6863 -
accuracy: 0.6448
Epoch 7/120
7/7 [==============================] - 113s 16s/step - loss: 0.6760 -
accuracy: 0.6466
Epoch 8/120
7/7 [==============================] - 110s 16s/step - loss: 0.6686 -
accuracy: 0.6636
Epoch 9/120
7/7 [==============================] - 110s 16s/step - loss: 0.6486 -
accuracy: 0.6667
Epoch 10/120
7/7 [==============================] - 109s 16s/step - loss: 0.6305 -
accuracy: 0.6914
Epoch 20/120
7/7 [==============================] - 110s 15s/step - loss: 0.5960 -
accuracy: 0.6954
Epoch 21/120
7/7 [==============================] - 110s 16s/step - loss: 0.5918 -
accuracy: 0.7083
Epoch 22/120
7/7 [==============================] - 109s 15s/step - loss: 0.5770 -
accuracy: 0.7040
Epoch 23/120
7/7 [==============================] - 123s 18s/step - loss: 0.5657 -
accuracy: 0.7131
Epoch 24/120
7/7 [==============================] - 110s 16s/step - loss: 0.5796 -
accuracy: 0.7260
Epoch 25/120
7/7 [==============================] - 110s 16s/step - loss: 0.5602 -
accuracy: 0.7089
Epoch 26/120
7/7 [==============================] - 112s 16s/step - loss: 0.5600 -
accuracy: 0.7195
Epoch 27/120
7/7 [==============================] - 110s 16s/step - loss: 0.5551 -
accuracy: 0.7338
Epoch 28/120
```

```
7/7 [==============================] - 111s 16s/step - loss: 0.5447 -
accuracy: 0.7358
Epoch 29/120
7/7 [==============================] - 110s 16s/step - loss: 0.5514 -
accuracy: 0.7332
Epoch 30/120
7/7 [==============================] - 111s 16s/step - loss: 0.5584 -
accuracy: 0.7300
Epoch 50/120
7/7 [==============================] - 117s 17s/step - loss: 0.4658 -
accuracy: 0.7830
Epoch 51/120
7/7 [==============================] - 113s 16s/step - loss: 0.4744 -
accuracy: 0.7780
Epoch 52/120
7/7 [==============================] - 112s 16s/step - loss: 0.4578 -
accuracy: 0.7786
Epoch 53/120
7/7 [==============================] - 111s 16s/step - loss: 0.4743 -
accuracy: 0.7754
Epoch 54/120
7/7 [==============================] - 110s 16s/step - loss: 0.4541 -
accuracy: 0.7876
Epoch 55/120
7/7 [==============================] - 110s 16s/step - loss: 0.4567 -
accuracy: 0.7844
Epoch 56/120
7/7 [==============================] - 110s 16s/step - loss: 0.4552 -
accuracy: 0.7839
Epoch 57/120
7/7 [==============================] - 109s 15s/step - loss: 0.4413 -
accuracy: 0.7792
Epoch 58/120
7/7 [==============================] - 113s 16s/step - loss: 0.4602 -
accuracy: 0.7873
Epoch 59/120
7/7 [==============================] - 110s 16s/step - loss: 0.4350 -
accuracy: 0.7959
Epoch 90/120
7/7 [==============================] - 110s 16s/step - loss: 0.2744 -
accuracy: 0.8808
Epoch 91/120
7/7 [==============================] - 108s 15s/step - loss: 0.2467 -
accuracy: 0.8957
Epoch 92/120
7/7 [==============================] - 110s 16s/step - loss: 0.2722 -
accuracy: 0.8826
Epoch 93/120
7/7 [==============================] - 114s 16s/step - loss: 0.2399 -
accuracy: 0.8978
Epoch 94/120
7/7 [==============================] - 108s 15s/step - loss: 0.2694 -
accuracy: 0.8893
Epoch 95/120
7/7 [==============================] - 108s 15s/step - loss: 0.2374 -
accuracy: 0.8994
Epoch 96/120
7/7 [==============================] - 109s 15s/step - loss: 0.2264 -
accuracy: 0.9091
Epoch 97/120
```

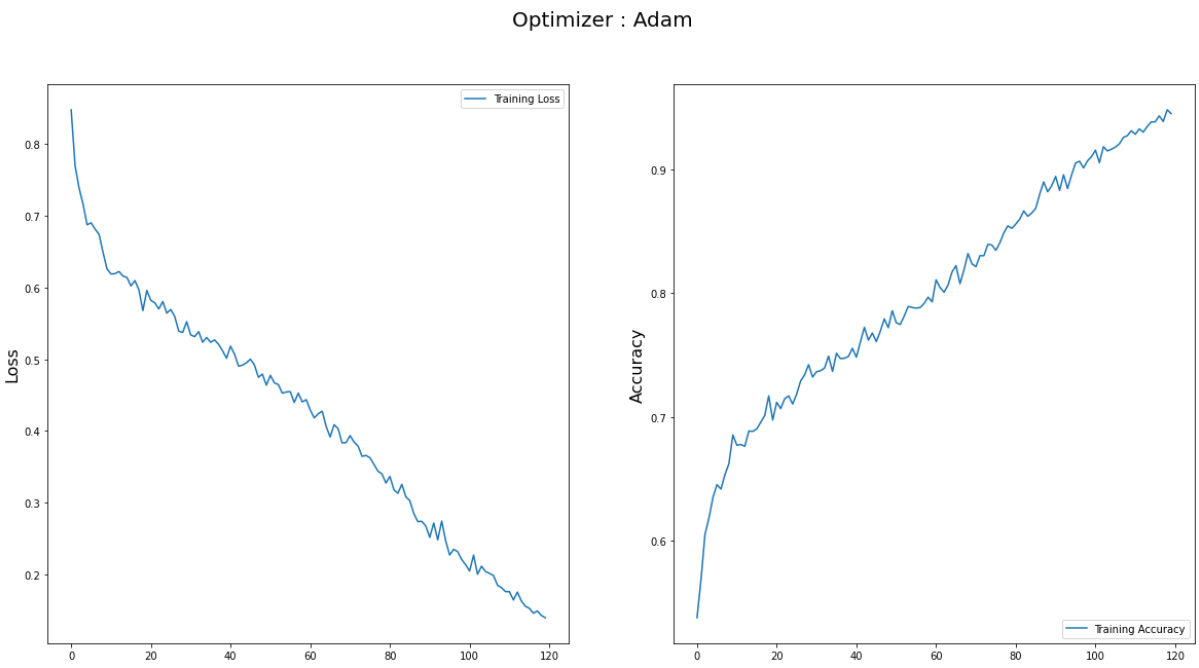
```
7/7 [==============================] - 108s 15s/step - loss: 0.2308 -
accuracy: 0.9060
Epoch 98/120
7/7 [==============================] - 111s 16s/step - loss: 0.2373 -
accuracy: 0.8987
Epoch 99/120
7/7 [==============================] - 109s 15s/step - loss: 0.2272 -
accuracy: 0.9046
Epoch 100/120
7/7 [==============================] - 109s 16s/step - loss: 0.2144 -
accuracy: 0.9115
Epoch 101/120
7/7 [==============================] - 110s 16s/step - loss: 0.1987 -
accuracy: 0.9189
Epoch 102/120
7/7 [==============================] - 109s 16s/step - loss: 0.2197 -
accuracy: 0.9099
Epoch 103/120
7/7 [==============================] - 109s 15s/step - loss: 0.2103 -
accuracy: 0.9141
Epoch 104/120
7/7 [==============================] - 109s 15s/step - loss: 0.2145 -
accuracy: 0.9147
Epoch 105/120
7/7 [==============================] - 112s 16s/step - loss: 0.2009 -
accuracy: 0.9184
Epoch 106/120
7/7 [==============================] - 110s 15s/step - loss: 0.1900 -
accuracy: 0.9236
Epoch 107/120
7/7 [==============================] - 110s 16s/step - loss: 0.1998 -
accuracy: 0.9177
Epoch 108/120
7/7 [==============================] - 110s 16s/step - loss: 0.1814 -
accuracy: 0.9258
Epoch 109/120
7/7 [==============================] - 109s 15s/step - loss: 0.1756 -
accuracy: 0.9281
Epoch 110/120
7/7 [==============================] - 110s 16s/step - loss: 0.1818 -
accuracy: 0.9274
Epoch 111/120
7/7 [==============================] - 110s 15s/step - loss: 0.1737 -
accuracy: 0.9291
Epoch 112/120
7/7 [==============================] - 109s 15s/step - loss: 0.1666 -
accuracy: 0.9298
Epoch 113/120
7/7 [==============================] - 109s 15s/step - loss: 0.1713 -
accuracy: 0.9329
Epoch 114/120
7/7 [==============================] - 110s 16s/step - loss: 0.1594 -
accuracy: 0.9341
Epoch 115/120
7/7 [==============================] - 113s 16s/step - loss: 0.1600 -
accuracy: 0.9339
Epoch 116/120
7/7 [==============================] - 111s 16s/step - loss: 0.1525 -
accuracy: 0.9374
Epoch 117/120
```

```
7/7 [==============================] - 111s 16s/step - loss: 0.1432 -
accuracy: 0.9452
Epoch 118/120
7/7 [==============================] - 111s 16s/step - loss: 0.1430 -
accuracy: 0.9422
Epoch 119/120
7/7 [==============================] - 111s 16s/step - loss: 0.1362 -
accuracy: 0.9512
Epoch 120/120
7/7 [==============================] - 113s 16s/step - loss: 0.1295 -

accuracy: 0.9498
```

Se grafica el aprendizaje y pérdida

```
plt.figure(figsize=(20,10))
plt.subplot(1, 2, 1)
plt.suptitle('Optimizer : Adam', fontsize=20)
plt.ylabel('Loss', fontsize=16)
plt.plot(history.history['loss'], label='Training Loss')
#plt.plot(history.history['val_loss'], label='Validation Loss')
plt.legend(loc='upper right')
################################################################################
##
plt.subplot(1, 2, 2)
plt.ylabel('Accuracy', fontsize=16)
plt.plot(history.history['accuracy'], label='Training Accuracy')
#plt.plot(history.history['val_acc'], label='Validation Accuracy')
plt.legend(loc='lower right')
plt.show()
```



Apéndice 3: Script de VA para la detección de la atención en tiempo real.

En este apéndice se muestra el código que hace posible la detección de la atención utilizando tecnología de visión artificial.

```
# import the necessary packages
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import imutils
import time
import cv2
import os

def detect_and_predict_attention(frame, faceNet, attentionNet):
  # grab the dimensions of the frame and then construct a blob
  # from it
  (h, w) = frame.shape[:2]
  blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
    (104.0, 177.0, 123.0))

  # pass the blob through the network and obtain the face detections
  faceNet.setInput(blob)
  detections = faceNet.forward()
  print(detections.shape)

  # initialize our list of faces, their corresponding locations,
  # and the list of predictions from our face attention network
  faces = []
  locs = []
  preds = []

  # loop over the detections
  for i in range(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with
    # the detection
    confidence = detections[0, 0, i, 2]

    # filter out weak detections by ensuring the confidence is
    # greater than the minimum confidence
    if confidence > 0.5:
      # compute the (x, y)-coordinates of the bounding box for
      # the object
      box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
      (startX, startY, endX, endY) = box.astype("int")
```

```
      # ensure the bounding boxes fall within the dimensions of
      # the frame
      (startX, startY) = (max(0, startX), max(0, startY))
      (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

      # extract the face ROI, convert it from BGR to RGB channel
      # ordering, resize it to 224x224, and preprocess it
      face = frame[startY:endY, startX:endX]
      face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
      face = cv2.resize(face, (224, 224))
      face = img_to_array(face)
      face = preprocess_input(face)

      # add the face and bounding boxes to their respective
      # lists
      faces.append(face)
      locs.append((startX, startY, endX, endY))

  # only make a predictions if at least one face was detected
  if len(faces) > 0:
    # for faster inference we'll make batch predictions on *all*
    # faces at the same time rather than one-by-one predictions
    # in the above `for` loop
    faces = np.array(faces, dtype="float32")
    preds = attentionNet.predict(faces, batch_size=32)

  # return a 2-tuple of the face locations and their corresponding
  # locations
  return (locs, preds)

# load our serialized face detector model from disk
prototxtPath = r"C:\Users\Diego Saavedra\Desktop\face_detector\deploy.p
rototxt"
weightsPath = r"C:\Users\Diego Saavedra\Desktop\face_detector\res10_300
x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face attention detector model from disk
attentionNet = load_model(r"\attention_detector.model")

# initialize the video stream
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()

# loop over the frames from the video stream
while True:
  # grab the frame from the threaded video stream and resize it
  # to have a maximum width of 1024 pixels
```

```
  frame = vs.read()
  frame = imutils.resize(frame, width=1024)

  # detect faces in the frame and determine if they are wearing a
  # face attention or not
  (locs, preds) = detect_and_predict_attention(frame, faceNet, attentio
nNet)

  # loop over the detected face locations and their corresponding
  # locations
  for (box, pred) in zip(locs, preds):
    # unpack the bounding box and predictions
    (startX, startY, endX, endY) = box
    (atento, desatento) = pred

    # determine the class label and color we'll use to draw
    # the bounding box and text
    label = "Atento" if atento > desatento else "Inatento"
    color = (0, 255, 0) if label == "Atento" else (0, 0, 255)

    # include the probability in the label
    label = "{}: {:.2f}%".format(label, max(atento, desatento) * 100)

    # display the label and bounding box rectangle on the output
    # frame
    cv2.putText(frame, label, (startX, startY - 10),
      cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
    cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

  # show the output frame
  cv2.imshow("Attention", frame)
  key = cv2.waitKey(1) & 0xFF

  # if the `q` key was pressed, break from the loop
  if key == ord("q"):
    break

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

**Apendice 4. Código que permite detectar la atención en vídeos pregrabados**

En el presente código se puede observar como el modelo de inteligencia artificial puede detectar la atención de un vídeo pregrabado.

```
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import cv2
import os
import cvlib as cv

# load model
model = load_model(r'C:\Users\Diego Saavedra\OneDrive - Universidad Téc
nica Particular de Loja - UTPL\Pruebas Python\attention_tesis\attention
_detector.model')

# open webcam
webcam = cv2.VideoCapture(0)

classes = ['atento','inatento']

# loop through frames
while webcam.isOpened():

    # read frame from webcam
    status, frame = webcam.read()

    # apply face detection
    face, confidence = cv.detect_face(frame)


    # loop through detected faces
    for idx, f in enumerate(face):

        # get corner points of face rectangle
        (startX, startY) = f[0], f[1]
        (endX, endY) = f[2], f[3]

        # draw rectangle over face
        cv2.rectangle(frame, (startX,startY), (endX,endY), (0,255,0), 2
)

        # crop the detected face region
        face_crop = np.copy(frame[startY:endY,startX:endX])

        if (face_crop.shape[0]) < 10 or (face_crop.shape[1]) < 10:
            continue
```

```
        # preprocessing for gender detection model
        face_crop = cv2.resize(face_crop, (96,96))
        face_crop = face_crop.astype("float") / 255.0
        face_crop = img_to_array(face_crop)
        face_crop = np.expand_dims(face_crop, axis=0)

        # apply gender detection on face
        conf = model.predict(face_crop)[0] # model.predict return a 2D
matrix, ex: [[9.9993384e-01 7.4850512e-05]]

        # get label with max accuracy
        idx = np.argmax(conf)
        label = classes[idx]

        label = "{}: {:.2f}%".format(label, conf[idx] * 100)

        Y = startY - 10 if startY - 10 > 10 else startY + 10

        # write label and confidence above face rectangle
        cv2.putText(frame, label, (startX, Y),  cv2.FONT_HERSHEY_SIMPLE
X,
                    0.7, (0, 255, 0), 2)

    # display output
    cv2.imshow("Attention", frame)

    # press "Q" to stop
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# release resources
webcam.release()
cv2.destroyAllWindows()
```

Cabe recalcar que se debe llamar al script y pasar como parámetro el vídeo pregrabado.

Ejemplo:

python detect_attention_video.py --input nombrevideo.mp4