



ESCUELA DE INGENIERÍA EN SISTEMAS

Tema:

SISTEMA DE PROCESAMIENTO DE IMÁGENES CORPORALES MEDIANTE INTELIGENCIA ARTIFICIAL PARA LA GUÍA DEPORTIVA Y NUTRICIONAL

Proyecto de investigación previo a la obtención del título de Ingeniero en Tecnologías de la Información

Línea de Investigación:

Sistemas de información y/o nuevas tecnologías de la información y comunicación y sus aplicaciones

Autor:

Marcelo Alejandro Acuña Martínez

Director:

Mg. Galo Mauricio López Sevilla

Ambato – Ecuador

Mayo 2023

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE AMBATO
HOJA DE APROBACIÓN

Tema:

**SISTEMA DE PROCESAMIENTO DE IMÁGENES CORPORALES MEDIANTE
INTELIGENCIA ARTIFICIAL PARA LA GUÍA DEPORTIVA Y NUTRICIONAL**

Línea de Investigación:

Sistemas de información y/o nuevas tecnologías de la información y
comunicación y sus aplicaciones

Autor:

Marcelo Alejandro Acuña Martínez

Galo Mauricio López Sevilla, Ing. Mg.

CALIFICADOR

f. 

Darío Javier Robayo Jácome, Ing. Mg.

CALIFICADOR

f. 

Santiago Alejandro Acurio Maldonado, Ing. Mg.

CALIFICADOR

f. 

Santiago Alejandro Acurio Maldonado, Ing. Mg.

DIRECTOR ESCUELA DE SISTEMAS

f. 

Dr. Hugo Rogelio Altamirano Villarroel, Dr.

SECRETARIO GENERAL PUCESA

f. 



Pontificia Universidad
Católica del Ecuador

SECRETARIA GENERAL
PROCURADURIA

Ambato Ecuador

Abril 2023



Pontificia Universidad
Católica del Ecuador
BIBLIOTECA *cts*

DECLARACIÓN DE AUTENTICIDAD Y DE RESPONSABILIDAD

Yo, **MARCELO ALEJANDRO ACUÑA MARTÍNEZ** portador de la cédula de ciudadanía **1803751765**, autor del trabajo de graduación intitulado: "SISTEMA DE PROCESAMIENTO DE IMÁGENES CORPORALES MEDIANTE INTELIGENCIA ARTIFICIAL PARA LA GUÍA DEPORTIVA Y NUTRICIONAL", previa a la obtención del título profesional de **INGENIERO EN SISTEMAS DE LA INFORMACIÓN**, en la escuela de **INGENIERÍA EN SISTEMAS**.

1. Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos del autor.
2. Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE Ambato, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad.

Ambato, mayo 2023



Marcelo Alejandro Acuña Martínez

CC. 1803751765

DEDICATORIA

Este trabajo está dedicado a mi familia, amigos, docentes universitarios y compañeros que fueron quienes me acompañaron durante los días que parecían complicados y celebraron mis victorias como propias.

AGRADECIMIENTO

El presente trabajo de grado, si bien ha requerido de esfuerzo y mucha dedicación, no hubiese sido posible su conclusión sin la colaboración de cada persona que forma parte de mi vida, principalmente mis padres, por siempre apoyar este arduo camino llamado vida, siendo constantes en forjarme como persona, hijo, amigo, profesional, estudiante y principalmente como ser humano, porque en estos tiempos la mayoría de gente se ha olvidado que es ser humano, tener valores y principios, ser empático con cada persona que rodea, amables, ser honestos, y sobre todo ser humildes. A mi querida hermana, por mantenerme vivo, por ser esa alegría al llegar a casa, quita cada estrés o problema con tan solo una sonrisa. A mis tíos

Por ser como mis segundos padres, y mi segundo hogar, gracias por cada consejo, inspiración, apoyo y amor que me han brindado en lo largo de este camino, ese apoyo el cual, ha logrado alcanzar de mejor manera mis metas y todo lo que me he propuesto en esta vida.

Agradezco a mis mejores amigos por siempre ser ese apoyo fuera de casa, por cada empujón, consejo y compañía para llegar a ser la persona que soy hoy en día, ser ese amigo, compañero de vida, profesional, y especialmente ser una persona de apoyo para esta sociedad.

Ing. Galo López

Más que un profesor, un gran amigo, los docentes son siempre un pilar fundamental para el aprendizaje dentro de la carrera al igual que en la vida, y usted me demostró ser un excelente docente, su apoyo cuando necesitaba un consejo, un apoyo, y mostrando siempre ese cariño a sus estudiantes, gracias por ser ese pilar dentro de mi vida universitaria. Y como no a cada docente dentro de mi querida facultad. Y por su puesto a mí querida Universidad y a todas las autoridades, por permitirme concluir una etapa más de mi vida, gracias por su paciencia, orientación y siempre guiarme en el desarrollo de este gran trabajo.

RESUMEN

El desarrollo de esta investigación tuvo un diseño experimental, el objetivo de este proyecto es usar esta herramienta para desarrollar un sistema de procesamiento de imágenes corporales mediante inteligencia artificial que brinde una guía nutricional, el desarrollo del sistema de procedimiento de imágenes clasificadas de manera manual según las características requeridas, en este caso: endomorfo, ectomorfo y mesomorfo, posterior a ello se generó un iterador a partir de las imágenes previamente establecidas, seguido de la creación el modelo de red neuronal con capas ocultas de neuronas convolucionales que tienen un núcleo o kernel y una función de activación. Para completar la creación del modelo se aplica la técnica de dropout que consiste en apagar neuronas en cada interacción para evitar el overfitting o el sobre entrenamiento. Finalmente, se exporta el modelo para poder ser utilizado en las aplicaciones requeridas. Para crear un sistema de procesamiento de imágenes con inteligencia artificial que brinden una guía nutricional se crea una aplicación web con HTML y JavaScript. Junto a los datos requeridos como, peso, altura, medidas corporales y edad, calcula las calorías necesarias y la distribución de macronutrientes de acuerdo con el objetivo del usuario. Finalmente, se genera una dieta mediante los resultados obtenidos y la guía entregada.

Palabras clave: inteligencia artificial, kernel, redes neuronales, overfitting.

ABSTRACT

The development of this research had an experimental design, the objective of this project is to use this tool to develop a body image processing system using artificial intelligence to provide nutritional guidance, the development of the system procedure of images manually classified according to the required characteristics, in this case: endomorph, ectomorph and mesomorph, after that an iterator was generated from the previously established images, followed by the creation of the neural network model with hidden layers of convolutional neurons that have a nucleus or kernel and an activation function. To complete the creation of the model, the dropout technique is applied, which consists of turning off neurons in each interaction to avoid overfitting or overtraining. Finally, the model is exported to be used in the required applications. To create an image processing system with artificial intelligence to provide nutritional guidance, a web application is created with HTML and JavaScript. With the required data such as weight, height, body measurements and age, it calculates the necessary calories and the distribution of macronutrients according to the user's objective. Finally, a diet can be generated based on the obtained results and the provided guide.

Keywords: artificial, intelligence, image processing, kernel

ÍNDICE GENERAL DE CONTENIDOS

DECLARACIÓN DE AUTENTICIDAD Y DE RESPONSABILIDAD	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
RESUMEN.....	vi
ABSTRACT.....	vii
INTRODUCCIÓN.....	1
CAPÍTULO I. ESTADO DEL ARTE Y LA PRÁCTICA.....	8
1.1. Procesamiento de imágenes corporales	8
1.2. Industria fitness.....	12
1.2. Inteligencia artificial digital de imágenes en la industria fit.....	16
CAPÍTULO II DISEÑO METODOLÓGICO	19
2.1. Metodología de investigación.....	19
2.2. Metodología de Desarrollo	22
CAPÍTULO III. ANÁLISIS DE LOS RESULTADOS DE LA INVESTIGACIÓN	57
3.1. Fase 6: Producto Final.....	69
CONCLUSIONES	72
RECOMENDACIONES	74
BIBLIOGRAFÍA.....	75
ANEXOS.....	84

INTRODUCCIÓN

La innovación tecnológica ha impulsado grandes cambios a nivel global, gracias a ella la estructura industrial, social y económica se ha visto impulsada a innovar de manera constante, a medida que las TIC's evolucionan el vínculo entre la tecnología, innovación y el trabajo se hace más fuerte. Entre los diversos descubrimientos y avances que ofrece la tecnológica esta la inteligencia artificial.

La inteligencia artificial o en su abreviatura (IA) juega un rol muy importante hoy en día, debido al mundo digitalizado actual, desde el momento que una persona nace tiene acceso a un teléfono, tablet o computador por lo que día a día se desarrollan miles de aplicaciones móviles (Apps) que facilitan las actividades cotidianas; para empezar, Vidal, Madruga & Valdés (2019), afirman que la definición exacta de IA está aún en construcción, sin embargo, la describen como la simulación de la inteligencia humana ejecutada por robots o máquinas diseñadas por humanos, en especial los sistemas informáticos, el fin que tiene la IA es que una máquina tenga la capacidad de comprensión y razonamiento similar a la de una persona.

Por otra parte Muñoz, Cortez (2020) mencionan que la inteligencia artificial es el inicio de la revolución tecnológica, está considerada como un concepto amplio de máquinas que aprenden por si solas y que tienen la capacidad de mejorar automáticamente se partió de experiencias pasadas, sin que sea necesaria una actualización, esto gracias al conjunto de algoritmos y redes neuronales extremadamente precisas.

Según Ponce & otros (2014), la evolución que ha tenido la inteligencia artificial ha enfrentado ciclos de éxito como la disminución de apoyo para su desarrollo paulatino, además, de ser parte de varias ramas científicas entre las más relevantes está: la medicina, psicología, filosofía, economía, educación y ética.

La evolución de la inteligencia artificial ha permitido que el conocimiento en las áreas antes mencionadas se consolide y promueva adelantos significativos,

que ayuden en el progreso de la calidad de vida de los seres humanos, sin embargo, una de las limitantes para que la IA evolucione mucho más rápido es la poca disposición de recursos para su apoyo y la escasa investigación en el tema.

Entre los cambios de gran impacto ocasionados por la IA, Ocaña, Valenzuela & Garro (2019), aseguran que ésta generó una gran brecha cognitiva entre las generaciones anteriores a los 2000, por lo que en los campus universitarios se convirtió en un reto adaptar a los estudiantes a las nuevas herramientas digitales, dando un giro de 180 grados a la educación convencional y su contexto, lo cual, también, modificó los estándares de aprendizaje y los típicos estereotipos que por años fueron muy marcados.

Existen diversos modelos de inteligencia artificial, sin embargo, para Alemán (2017) la IA se divide en cuatro categorías: Sistemas que piensan como humanos, sistemas que actúan como humanos, sistemas que piensan racionalmente y los sistemas que actúan racionalmente . Mientras que Cotino (2019), afirma que las IA denominadas fuertes, hacen referencia a aquellas que son capaces de aplicarse a una gran variedad de problemas y dominios diferentes; además, cumplirá con el requisito de tener la capacidad de realizar tareas en función al juicio de valor y percepción sin que una persona deba intervenir. Hoy todas las IA todavía se clasifican en el primer grupo.

Por otra parte, Ramió (2018), refiere a autores que mencionan la existencia de diversos sistemas IA, pero esto debido a su evolución y a la mezcla de los dos modelos antes mencionados entre los creados están: sistemas que piensan como humanos, sistemas que actúan como humanos, sistemas que actúan racionalmente, sistemas que piensan racionalmente.

La inteligencia artificial ha generado una innumerable cantidad de beneficios desde su aparición, entre los cambios significativos que ha provocado es el ofrecer un desarrollo sostenible, que tiene como fin usar la IA para solucionar problemáticas de pobreza extrema en áreas con características diversas, además, la inteligencia artificial ayudado notablemente a la alfabetización de

las personas que tienen problemas para acceder a una educación digna por medio de la implementación de plataformas educativas que satisfagan las necesidades cognitivas de este tipo de regiones con carencias económicas (Porcelli, 2020).

En el área industrial la inteligencia artificial ayudó a que los costos y salarios se redujeran por lo que la efectividad y productividad incrementaron notablemente, la automatización y la rápida fluidez en las zonas de producción hace, que se produce más por minuto de este modo las empresas sacan al mercado productos de calidad, con menor precio a fin de ser más competitivos (Negro & Pons, 2018).

En el campo de la salud, Martínez, Dalgo, Herrera, Analuisa & Velasco (2019), al realizar su investigación logran determinar que la necesidad de nuevas herramientas informáticas para el desarrollo de fármacos es inminente, aún a pesar de que éstos no den soluciones al 100% en enfermedades catastróficas, los médicos y fármacos guardan la esperanza en que la evolución de la inteligencia artificial permita mejorar la calidad de vida de las personas en un futuro con un menor nivel de incertidumbre.

El área de la salud y cuidado personal es una de las tantas ramas que han conseguido beneficios por la interacción de las matemáticas y computadores, especialmente en la optimización de procesos y recursos. Lugo, Maldonado & Murata (2014), en su publicación afirman que una de las ramas de la inteligencia artificial llamada aprendizaje automático permite que al partir de una base de datos se mejorar y adiestrar sistemas capaces de ejecutar procedimientos de razonamiento basado en casos, redes neurales artificiales, regresiones logísticas múltiples, máquinas de soporte, entre otros.

Mientras que Arias & otros (2019), llegan a determinar que el adecuado funcionamiento de un algoritmo matemático de un computador que depende de la calidad y cantidad de información, que se ingrese en el sistema, por lo que la recopilación de la misma es estratégica para contar con una base de datos confiable y valida. Todo este proceso no se sustituye a personas en el futuro,

sin embargo, ayuda de gran manera al trabajo tedioso que en algún momento generaba carga para los profesionales, por lo que el enfoque de estos es en la atención adecuada para sus clientes.

Entre las diversas formas de ingresar información a la base de datos en la, que se vaya a operar está el procesamiento digital de imágenes, que en palabras de Lira-Baca et al. (2019) es un conjunto de técnicas y procesos que permite el descubrimiento de toda la información que está contenida en una imagen, basada en dos áreas principales: el primero es mejorar la calidad de todo el contenido de la misma y el segundo es el reconocimiento de patrones que permiten identificar las propiedades que pudiesen tener algún significado.

Ahora bien, a la hora de hablar de procesamiento de imágenes, también, es necesario hablar de redes neuronales artificiales por lo que Origel-rivas & Rendón-lara (2020) las define como el conjunto de algoritmos de aprendizaje automático que tienen la capacidad de procesar y entender la información, estos son usados como clasificadores dentro de los sectores industriales, manufactureros, financieros, salud y otros. Añade que la serie de procesadores elementales forman dispositivos simples de calculo que parten de un vector de entrada procedente del exterior o por estímulos recibidos de las neuronas mismas que dieron como resultado una respuesta genuina.

Según Arora & Kaur (2022) existen tres tipos de redes neurales, el primero es de entrada este recibe señales del entorno, es decir, ingresa información a la red, el segundo tipo es el de salida que emite señales afuera y por último están los ocultos cuyas entradas y salidas están dentro del sistema y no tienen ningún contacto con el exterior.

Por otra parte, Rabadan (2019) afirma que la diversidad de información, que se obtiene de una persona o un conjunto de personas gracias a la inteligencia artificial permite que diversas organizaciones generen una ventaja competitiva gracias a estos algoritmos, una de las industrias, que se ha visto beneficiada es la fitness, en la actualidad verse bien es sinónimo de salud lo cual es aprovechado por empresarios especializados en nutrición y entrenamiento

personalizado.

Cárdenas (2018), afirma que la industria fitness digital es tendencia en la actualidad y su desaparición es una realidad muy lejana, el campo nutricional ha tenido resultados increíbles, por medio de los patrones recopilados por la IA se diseña una receta que responde a todas las necesidades individuales de una persona y consigue una mejora notable en la calidad de vida.

Por otra parte, Alvarado & Santana (2021), afirman que actualmente existen muchas aplicaciones dentro de las diferentes tiendas existentes en los sistemas operativos que intentan entregar una ayuda nutricional o entrenamiento para el usuario, sin embargo, no se aprovechan todas las herramientas tecnológicas disponibles tales como el procesamiento de imágenes dentro de dichas aplicaciones para facilitar el proceso de asistencia dentro del sector deportivo para los diferentes usuarios de los sistemas operativos más conocidos como IOS o Android.

En el caso de la comunidad ambateña se ha visto un mayor incremento de personas que buscan un buen estado físico, con el objetivo de tener una mejor salud y verse bien, a pesar del gran avance de inteligencia artificial específicamente en el procesamiento de imágenes en otras industrias, el sector deportivo aún no ha desarrollado todo su potencial.

Definir cuáles son los hábitos de consumo y rutina dentro del mercado fitness tanto en los suplementos alimenticios como las rutinas que aplican las personas para sentirse y verse bien por medio de aplicaciones, marcas deportivas, redes sociales, actividades físicas que practican, esta información permite, que se diseñe un plan adecuado y personalizado.

Objetivo general

- Desarrollar un sistema de procesamiento de imágenes corporales mediante inteligencia artificial que brinde una guía nutricional, para lo cual se emplearan diversas herramientas TIC's para el procesamiento

de imágenes que permita la identificación, los tipos de cuerpo y adicionalmente tome ciertos parámetros ingresados por teclado para entregar una asistencia fitness.

Objetivos específicos

1. Elaborar un estado de arte sobre inteligencia artificial y procesamiento de imágenes para la industria fitness.
2. Desarrollar el sistema de procesamiento de imágenes corporales en Python.
3. Brindar una guía básica de nutrición en base a los datos recogidos del sistema

La investigación se enfoca en la provincia de Tungurahua en el cantón Ambato. Por otro lado, la información que sustenta su desarrollo contiene datos relevantes acerca de inteligencia artificial, aprendizaje automático, interpretación de imágenes, industria fitness, estilo de vida, comportamiento del consumidor.

Adicionalmente, el desarrollo del trabajo responde a las siguientes interrogantes:

- ¿Cuáles son las teorías que sustentan el desarrollo de sistema de procesamiento de imágenes con el uso de inteligencia artificial?
- ¿Cómo se desarrolla un sistema de procesamiento de imágenes?
- ¿Cómo implementar los sistemas de procesamiento de imágenes con inteligencia artificial que brinden una guía nutricional en base a los datos recolectados?

Para lo mencionado en primera instancia se construye el estado del arte que contribuye a la investigación, posterior a ello se profundiza en el lenguaje de programación Python y sus beneficios, seguido se define información acerca de un sistema de procesamiento de imágenes.

La investigación es de tipo descriptivo debido a las revisiones sistemáticas, investigaciones bibliográficas, análisis de los impactos y desarrollo de la IA por medio de fuentes confiables que sustentan el tema, para el diseño del sistema de procesamiento de imágenes corporales mediante inteligencia artificial es necesario ingresar al software una data set que brinde una guía nutricional en base al tipo de cuerpo que tenga cada usuario que cuente con la app.

Finalmente, el sistema de procesamiento de imágenes se encarga de clasificar los tipos de cuerpos existentes, y con una guía básica de nutrición basada en los datos obtenidos del sistema, se genera la orientación adecuada que beneficia a que la industria fitness crezca y otorgue mejores servicios, además, de que los usuarios disfruten de mejores resultados físico-emocionales.

CAPÍTULO I. ESTADO DEL ARTE Y LA PRÁCTICA

1.1. Procesamiento de imágenes corporales

El procesamiento de imágenes en palabras de Mora & Sánchez (2020) es el conjunto de imágenes digitales que busca la mejora en la búsqueda de la información o mejorar la calidad de las mismas. Mientras que Yoelkis (2021) asegura que el procesamiento de imágenes digitales está íntimamente relacionado con conceptos tecnológicos, *software*, hardware, además, hace referencia a como esté ha generado desarrollo de las industrias y sociedad en general, añade, finalmente, que para la solución de cada caso es necesario un determinado grupo de logaritmos.

En el campo de la educación al agregar como metodología didáctica el procesamiento de imágenes según Mitaritonna (2018) permite que las habilidades, destrezas de los estudiantes mejoren notablemente, así como, también, los niveles de asimilación y comprensión cognitiva, lo cual saca de la zona de confort a los docentes, las brechas tecnológicas afectan aquellos docentes longevos que no se han actualizado en temas relacionados a la tecnología.

En investigaciones similares está la de Guadalupe Sánchez-Méndez & Pedro Arguijo-Hernández (2018) misma que a través de las imágenes multiespectrales lograron visualizar las plagas que afecta a los cultivos de café ubicados en el estado de Veracruz, los resultados fueron favorables en las plantaciones, esto a su vez mejoró la producción de las mismas, el autor añade que entre los hallazgos en el futuro es necesario, que se evalúen los patrones de algoritmos a fin de mejorar las redes neurales y árboles de decisión para el alcance de resultados más precisos.

Capraro et al., (2018) en su investigación denominada “ Determinación de CWSI en olivares mediante procesamiento de imágenes termo gráficas” presenta la metodología aplicada para calcular el índice de CWSI con el fin de determinar el estrés hídrico en un olivar por medio del análisis de imágenes

termo gráficas. Para alcanzar lo planteado fue necesario el uso de cámaras Aero transportadoras durante la hidratación de los cultivos en el transcurso del día a fin de, que se consiguiera un riego agrícola con precisión.

En la investigación desarrollada por Díaz (2020) llegó a determinar que el estudio de imágenes ha tomado un papel relevante dentro del área de la salud en los actuales años por la evolución agigantada del desarrollo tecnológico y computacional por medio de algoritmos matemáticos de alta complejidad, al utilizar la IA por medio de imágenes se ha identificado el cáncer de forma temprana, el especialista analiza de manera minuciosa las anomalías cutáneas como el no melanoma y la queratosis actínica a tiempo.

En el campo médico el uso de este tipo de herramientas tecnológicas permite alcanzar con éxito objetivos que en el pasado parecían inalcanzables o difíciles de obtener en la investigación realizada por (Martínez-Castilla et al., 2021) afirma que al aplicar el SPN a la hora de realizar injertos el promedio de éxito alcanzado fue de 98,4% gracias a la medición digital del área donde los médicos injertan piel, lo cual aclaro que la tasa de éxito depende de la parte del cuerpo donde se realice el injerto, además, de que el uso de una computadora para analizar la evolución de la misma disminuye los días de hospitalización de un paciente, por lo que genera un costo – efectivo al hospital o clínica.

Mientras que en la investigación desarrollada por Azevedo (2018) afirma que al aplicar escaneos 3D en el área médica se obtiene un mejor análisis del tamaño, forma corporal, área y volumen del cuerpo a fin de que los pacientes logren superar la etapa de obesidad y alcanzar un peso adecuado, el apoyo que estos software dentro de la medicina no son valorados al 100% por lo que no existen mucho de este tipo, mientras que las existentes no son de gran tamaño en su mayoría pero si de alto costo, además, de que no cuentan con evaluación y descripción y malformaciones congénitas y síndromes.

Por otra parte, Suárez Mejías (2021) menciona que los algoritmos propuestos por el investigador son innovadores, estos fueron usados en las imágenes radiológicas en 3D, permitieron que los oncólogos y cirujanos seleccionen

semiautomáticamente los tumores existentes en los pacientes, además, permite, que se disminuyan las varianzas en procesos tumorales con el fin de planificar la administración de tratamientos quirúrgicos y terapéuticos.

Arselan (2016) se planteó como objetivo la implementación de dos códigos computacionales simples que le permitan a los radiólogos una reconstrucción tomográfica dentro del software que usan frecuentemente para mejorar las imágenes, efectivamente la simplicidad de los valores matemáticos permitió un buen dinamismo entre el software y los modelos matemáticos, el primer logaritmo propuesto es el FDK logra con éxito en los errores del 1% durante el tiempo de prueba de los logaritmos, mientras que con el simrx hubo errores del 2%.

En la publicación de Scarinci et al.,(2020), que se propuso el desarrollo e implementación tanto original como automatizada para identificar tejidos y órganos en imágenes PET-CT, una vez aplicado resultó ser una herramienta muy útil, además, el autor agrega que el uso del logaritmo HDBSCAN permite el procesamiento de grandes cantidades de datos de entrada lo que permitió segmentaciones en tiempos cortos, entre los retos que identificó es el mejorar la calidad de la segmentación en imágenes de cuerpo completo.

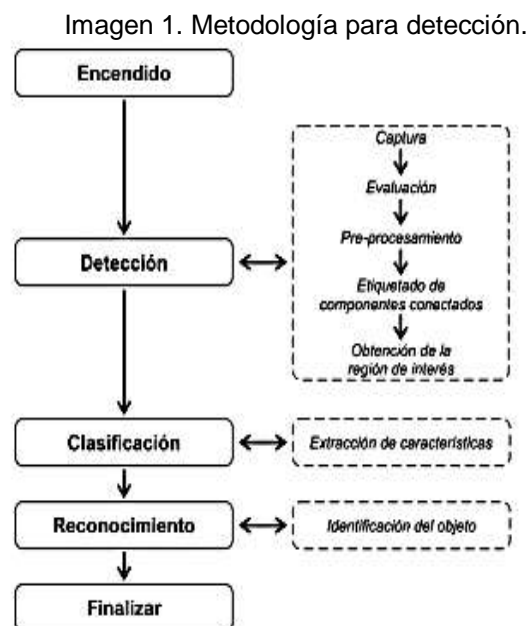
Valero Carvajal et al.(2021) se plantean en su investigación desarrollar e implementar algoritmos de procesamiento digital de imágenes y machine learning, agregados al robot humanoide InMoov, llegó obtener resultados tales como alcanzar una eficiencia del 95% en un ambiente controlado, se cumplió de este modo con el objetivo general, sin embargo, es necesario analizar lo que sucede al aplicar esto en lugares expuestos a la luz, todo lo que tiene semejanzas al color piel se confunde con el entorno.

En cuanto a los dispositivos usados para el procesamiento de imágenes Careta et al., (2019) en su trabajo de investigación el fin fue identificar las condiciones en el hueso al propagarse el sonido en el momento de recopilar la información, en primer lugar, usaron una frecuencia de 256 Hz, se usa gel conductor para la transmisión de sonido en un total de 13 imágenes, para lo cual adicionalmente

fue necesario discriminar 3 colores primarios rojo, azul y verde, para la investigación fue viable analizar los resultados analizados con color rojo porque este refleja la temperatura, llegó a concluir que el hueso es más expuesto al sonido refleja mayor intensidad de color rojo.

En el caso de Buzai & Montes Galbán (2020) que analiza la expansión territorial de Buenos Aires por medio del procesamiento digital de imágenes satelitales nocturnas, estas permitieron comprobar que la ciudad de BA tuvo un crecimiento poblacional y territorial, a la vez se presentó el software diseñado con la cartografía satelital digital obtenida lo cual permite ser un referente para análisis futuros.

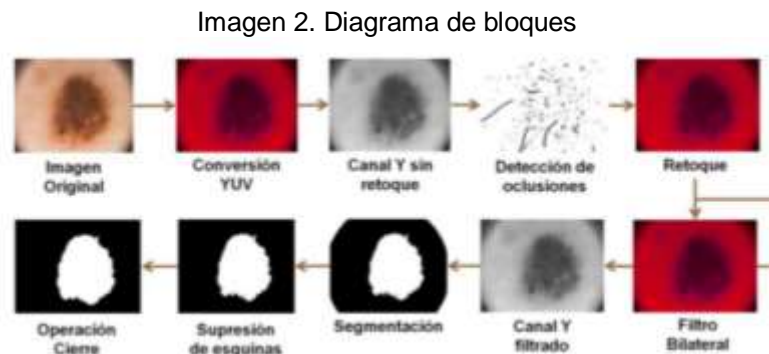
Según (Jerónimo & Sossa, 2018) para el análisis técnico no lineal de tratamiento de las estructuras geométricas de una imagen basada en la teoría de conjuntos ellos proponen la siguiente metodología:



Fuente: tomado de Jerónimo & Sossa (2018)

Por otra parte, Verónica et al., (2021) en su investigación llamada “Extracción y selección de características en imágenes digitales lo cual mostró lesiones de piel” propone que en el pre procesamiento es necesario, que se ilustre por medio de un diagrama de bloques con el fin de alcanzar el procesamiento de

imágenes, en este caso específico dermatoscópica.



Fuente: tomado de Verónica et al., (2021)

García-Estrada et al., (2018) en su investigación titulada “Sistema de clasificación de deformaciones pédicas por procesamiento digital de imágenes y lógica difusa en LabVIEW obtiene los siguientes resultados, en primer lugar, sistematizar y clasificar las malformaciones pédicas, por otra parte la aplicación de cámaras permitió analizar y eliminar información no deseada, a fin de mejorar las líneas y medidas fundamentales para la marcha de los pacientes que fueron parte de la investigación, mientras que los especialistas creen que el uso de este tipo de herramientas permite un diagnóstico más fácil.

El análisis de imágenes, también, es empleado en el área de la construcción Tarazona & Sandoval (2019) afirman que en su investigación llamada “Evaluación de discontinuidades tipo grietas y fisuras en estructuras de hormigón para emplear un analizador de vibraciones y procesamiento digital de imágenes”, la aplicación de esta herramienta les permitió la corroboración de los resultados arrojados por los analizadores de vibración RMS y el coeficiente de Wavelets, el uso de las imágenes localizó de manera precisa las fisuras generadas en el hormigón.

1.2. Industria fitness

En la actualidad es conocido que hacer ejercicio tiene como fin mantener un cuerpo sano, sin embargo, entre los diversos beneficios que este presenta esta mejorar el bienestar y calidad de vida en general, entre los beneficios de

mantener el cuerpo ejercitado esta la generación de endorfinas de manera natural, además, que ayuda al control de la presión arterial, genera colesterol bueno, mantiene un buen flujo sanguíneo y evita, que se genere una pérdida de masa ósea que provoca osteoporosis.

La cultura fitness toma auge a principios del siglo XX por el desarrollo de máquinas mucho más sofisticadas, de este modo las personas que dedicaban tiempo a entrenar lograban alcanzar cuerpos esculturales y con mayor definición, a medida que ha pasado el tiempo el mantenerse saludable y tener un cuerpo ejercitado se ha convertido en una decisión de cierta manera condicionada por la cultura, sociedad y tecnología que han generado auge a los centros de acondicionamiento físico (Espinal-Correa & Estrada-Mesa, 2020).

El mantenerse en forma o saludable ha generado la necesidad de que las personas estén en una búsqueda constante de rutinas, dietas o centros de salud integral para verse y sentirse bien de la Cámara et al. (2019) publicaron que en el caso de España los servicios más buscados son: entrenamiento de fuerza, contratación de profesionales certificados, equipos multidisciplinarios de trabajo, entrenamiento funcional y regulación de ejercicios profesionales fitness. Entre los hallazgos más relevantes esta que los usuarios buscan personas, que sean profesionales certificados en el sector.

Los beneficios de hacer ejercicio son múltiples Torres-Criollo & Ochoa (2021) menciona que entre los más representativos está el cambio metabólico y neuropatológicos de las personas en el caso de enfermedades psicológicas, en donde actividades como el yoga o entrenamiento de la flexibilidad permite retardar tanto su aparición como su desarrollo.

Estany & Vera (2020) en el caso de cardiología llegan a determinar que el beneficio de los programas de Rehabilitación Cardíaca multidisciplinario en el pronóstico de pacientes con disfunción de ventrículo izquierdo de origen isquémico, proporcionaron mejoría de los parámetros ergo métricos que determinaron aptitud física, sin haberse precisado cambios de la FEVI en

reposo o esfuerzo; por tanto, el incremento de la capacidad física se atribuye a adaptaciones periféricas y músculo-esqueléticas ocasionadas por el ejercicio físico.

Mercedes et al. (2021) tras implementar un programa de ejercicio físico combinado en personas con sobre peso llega a determinar que al combinar este por 16 semanas usó el entrenamiento de fuerza y resistencia como ejes estratégicos y como eje transversal el trabajo de la flexibilidad, aplicado de acuerdo a una estructura, dosificación, continuidad, al ejercer controles individualizados por personal idóneo y calificado se convierte en la principal estrategia objetiva ante la mejora de la salud de estos individuos. Lo que demuestra una adherencia objetiva del ejercicio físico como tratamiento activo en personas con obesidad y sobrepeso.

En la publicación de Serra Valdés (2020) donde sugiere ejercicios para adultos durante la etapa de confinamiento donde logro determinar que la práctica de ejercicio físico aerobio de intensidad moderada favorece el fortalecimiento del sistema inmunológico y la respuesta inmune humoral, debido al aumento de las células T, leucocitos e inmunoglobulinas como las IgM, IgE, IgG, e IgA, mismas que actúan como respuesta antiviral, además, disminuye el riesgo de infecciones respiratorias del tracto superior hasta en un 30%; también, aumenta, como respuesta crónica, la cantidad de citosinas antiinflamatorias como IL-10, IL-1 y células de clara (las cuales brindan protección al epitelio del bronquiolo) y, finalmente, es capaz de regular la actividad de las patrocinas, las cuales, en cantidades elevadas (como es el caso de personas que presentan COVID-19).

En el caso europeo el crecimiento de la industria fit ha generado un gran despunte y por lo tanto buena rentabilidad en el 2015 tuvo un incremento de 3,9% centros de cuidado personal lo cual representó 26,7 billones de dólares, en Europa se ha apalancado el crecimiento de este sector económico basados en los estereotipos de cuerpos implementados en la sociedad, sin embargo, al existir tantos centros fit en los clientes se genera un síndrome de infidelidad por lo que el servicio de estos implica a una mejorara constantemente (Zamorano &

García, 2018).

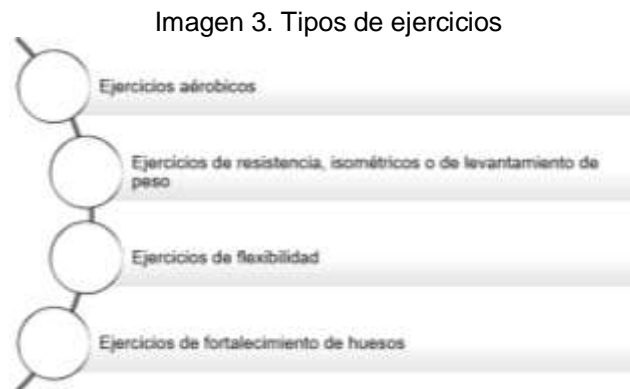
Mientras que en la investigación realizada en Rusia Panasenko et al. (2018) llega a definir que las tendencias de la industria fit tanto al corto como al largo plazo reflejan que esta es sostenible en tiempo entre las formas de innovación de esta industria el autor propone métodos de impacto gamificación en los consumidores, formas interactivas de estimulación de los consumidores con el fin de corregir las tendencias de consumo en base la psicología del marketing y gestión para motivar a la demanda del servicio propuesto.

En México las tendencias en esta industria la perspectiva que la sociedad maneja después de la pandemia del COVID-19 refleja que la cultura fitness llegó para quedarse, Gómez Chávez et al (2022) determino que entre los retos que presenta la industria para los años posteriores esta la prevención de obesidad, hipertensión y diabetes, finalmente, la publicación afirma que entre las actividades más solicitadas a los profesionales están personalizadas, entrenamiento en casa y ejercicio en espacios amplios o abiertos.

Estrada-Marcén et al.,(2019) en el análisis que realizó en los centros fitness de la ciudad Zaragoza donde encontró que la infraestructura de los centros deportivos todas disponen de una sala general y una para actividades dirigidas y el 35,3% tiene dos salas fitness y el 58,8% tiene espacio para SPA, además, de que el 17,6% ofrece a sus clientes una pista polideportiva, mientras que en los espacios al aire libre que ofrecen 38,9% tienen instalaciones físico-deportivas, en cuanto a inclusión el 28% ofrece acceso para personas con capacidades especiales.

Los hábitos de ejercitarse se vieron modificados en época de pandemia por lo que Zamorano & García (2018) afirma que los niveles de prevención incrementaron debido a que los patrones de vida sedentaria ocasionaron la muerte de muchas personas contagiadas de COVID- 19, por lo que se diseñaron rutinas para ejecutarse en casa, aun así, por medio de las encuestas se evidenciaron que el 46,35% de las personas realiza escasa actividad física, mientras que solo el 31,76% tiene una rutina establecida.

Los tipos de ejercicios son los siguientes:



Fuente: tomado de Rosa Guillamón (2019)

1.2. Inteligencia artificial digital de imágenes en la industria fit.

Tras el impacto que generó la pandemia entre los diversos negocios, que se caracterizaban por el contacto físico o lugares de alta concurrencia tomaran un giro impresionante, dentro de los cambios a los, que se vieron enfrentados está el sector de deportivo y todas las ramas que están íntimamente relacionados a este, entre las herramientas que les fue de utilidad para que los entrenadores físicos entrenen a las personas en sus hogares está el uso de redes sociales, así como plataformas de reunión virtual, por lo que el rol de la inteligencia virtual dentro del deporte en casa jugó un papel muy importante.

Actualmente se han hecho muchos avances con respecto a la visión artificial y a los sistemas de procesamiento de imágenes, por el hecho de que este es un campo que tiene mucho potencial y muchas instituciones, tanto universidades como industrias, investigaron como aplicar y como mejorar los actuales sistemas de visión artificial y procesamiento de imagen; en esta sección se muestra algunos trabajos relacionados con el tema de la visión artificial y su utilidad en el desarrollo de la investigación.

El apoyo de la inteligencia artificial en el ámbito educativo ha generado que aquellas barreras que en su momento fueron un limitante en la actualidad desaparezcan, el que un docente de cultura física logre impartir clases por medio del desarrollo de imágenes descriptivas de la ejecución de ejercicios

físicos, permitió que tanto niños como familias enteras se mantengan en actividad durante el tiempo de confinamiento, por lo que para los docentes, que se dedican a esta profesión se les hiciera mucho más fácil impartir clases por medio de un computador (Blas Yoel & Rodríguez Hernández, 2021).

El desarrollo de algoritmos para el procesamiento de imágenes ha cobrado mayor impulso en el ámbito científico, en especial en el campo de la inteligencia artificial para aplicaciones relacionadas con visión por computadora. Este conocimiento se nutre de la digitalización de datos en dispositivos para captar videos, fotos, etcétera. Las imágenes, entonces, se convierten en el núcleo de la descripción de la realidad. Asimismo, el uso de piezas publicitarias y su digitalización en el mercado retail abren una oportunidad para la aplicación de técnicas de análisis de imágenes en lógicas para identificar oportunidades en los procesos comerciales, operativos y generación de conocimiento.

Flores Echaiz et al.,(2021) llega a determinar que el acceso a la IA y a otras TIC avanzadas es importante para superar las nuevas brechas digitales, que se registran entre los países y dentro de ellos. En este sentido se ve varias tendencias importantes, el acceso a los datos fortalecerse a través de repositorios de datos abiertos para asegurar el acceso gratuito a los datos para el entrenamiento de algoritmos. Sin embargo, hay que prestar más atención a los temas relacionados con la interoperabilidad representativa de los datos y las normas de datos. Por último, es necesario poder acceder a hardware de computación para correr los algoritmos de IA. Este aspecto requiere un mayor esfuerzo de parte de los gobiernos y del sector privado para hacer que las unidades de procesamiento adecuadas estén disponibles, a pesar de que la banda ancha asequible ofrece una solución alternativa parcial.

Herrera Torres et al. (2019) en España tras aplicar su investigación en 215 empresas, que se dedican al servicio Fit a la hora de analizar los resultados llegaron a determinar que las plataformas virtuales por las, que se alcanzó a sus clientes fue, en primer lugar, Facebook, seguida por Instagram lo que permitió mantener, a pesar de las restricciones sanitarios por temporada de

pandemia.

El objetivo principal del proyecto de Sánchez (2021) fue el de desarrollar un sistema de visión artificial para realizar la corrección de posturas en el ejercicio de levantamiento de potencia mediante un algoritmo, la utilización de una cámara y el monitoreo cardiaco donde se concluye, que se cumplió con este objetivo de manera satisfactoria, lo que mantuvo los grados de flexión en los rangos correctos de dicho ejercicio y establecidos en los estándares de los organismos de salud y deporte.

El “Sistema de Visión Artificial para la Detección y Corrección de Posturas en Ejercicios realizados por Fisicoculturistas” desarrollada por José Martínez y Juan Manuel en el 2018 contempló: la vista, con el reconocimiento de objetos o patrones, el seguimiento del cuerpo humano y la captura de movimiento (Mo Cap). Para lo cual se usó como herramienta principal elementos computacionales denominados exergame, el sistema tenía como objetivo primordial la programación de algoritmos que buscan solucionar los movimiento, posturas y tiempos empleados por los fisiculturistas mediante la posición adecuada de sus articulaciones en el sistema de ejes de tres dimensiones.

En la Institución Universitaria de Salazar y Herrera en el año 2018, S. Amaya, I. Torres y D. Velásquez con el tema “Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura” usó una cámara web unida al software Pyton, el objetivo principal de esta investigación es usar una cámara web integrada al software libre Python, mismo que trabaja en conjunto de datos de las librerías Open CV, se base en el funcionamiento del posicionador de objetos, configurado previamente con el sistema de colores, que se presenta al frente de la cámara, misma que captura la imagen para procesarla y clasificarla de manera eficaz.

CAPÍTULO II DISEÑO METODOLÓGICO

2.1. Metodología de investigación

El desarrollo del presente inicio con la definición de los términos básicos relacionados al tema, así como los antecedentes investigativos y documentación. Un análisis pertinente detallado del problema encontrado y su posible solución, que sería el desarrollo de un sistema de procesamiento de imágenes corporales mediante inteligencia artificial que brinde una guía nutricional. Así como detallar el entorno en el, que se va a usar el software a fin de que este brinde resultados eficaces y convenientes, para determinar es necesario los resultados de esta herramienta, que se pruebe el funcionamiento por medio de un prototipo en el cual, esta una base de datos e imágenes de los diferentes tipos de cuerpos que existen.

Para el desarrollo de una interfaz natural de interacción entre una persona y una herramienta informática es necesario el uso de diversas técnicas y métodos investigativos, estos a su vez permiten, que se obtenga información de utilidad y relevancia de diversas fuentes bibliográficas ya sean físicas o digitales.

Método general: analítico sintético

Para el desarrollo de la presente investigación se utilizó el método analítico sintético que contribuye a la guía deportiva y nutricional de las personas que buscan mejorar su calidad de vida, es necesario, que se construyan conocimientos sobre el *software*

y todas las herramientas, que se van a usar, estos a partir de investigaciones previas, el sintetizar información bibliográfica permite, que se organice de manera ordenada toda la información, que se obtiene a fin de alcanzar todos los objetivos propuestos.

Método específico: experimental

La investigación tiene como metodología la experimental, esta consta de un grupo de actividades metodológicamente aceptables y técnicas que ayudaron a obtener la información necesaria sobre el tema en desarrollo a fin de, que se optimicen recursos tecnológicos, económicos y humanos. Para lo cual la persona que desarrolla la investigación genera situaciones en las que se controla o deliberar como las variables en análisis se manipula para alcanzar el objetivo general del estudio.

La presente es experimental debido a que el desarrollo del prototipo parta de las necesidades del usuario y por medio de varias pruebas, a fin de que poco a poco, se llegue a cumplir todas las expectativas planteadas y se cumplan los objetivos planteados.

Tipo de investigación

Documental

Según Tancara Q, (1993) este tipo de investigación constituye una serie de técnicas y diversos métodos que permiten proporcionar una base teórica para toda la información, que se presenta en un determinado documento.

Para desarrollar la presente investigación, se usa la investigación de tipo documental y bibliográfica por la revisión de todas las investigaciones previas, que se han realizado ya sean similares o sean de utilidad para el desarrollo y cumplimiento de los objetivos previamente planteados.

Campo

En palabras de Araica Zepeda & Hernández (2020) este tipo de investigaciones, se basan en encuestas, entrevistas, cuestionarios o por el registro de observaciones que el investigador realice, en este tipo de investigaciones, se cree necesario, que se tomen en cuenta que los cuestionarios ya hayan sido aplicados con anticipación a fin de que estos

hayan sido correctamente avalados.

En el caso de la presente es de campo debido a que se lleva un registro a modo de bitácora de todo lo que se logre observar al momento de ejecutar el prototipo hasta alcanzar el producto final o se cumpla con las expectativas planificadas.

Técnicas e instrumentos de recolección de la información

La técnica usada para recolectar la información requerida durante el desarrollo del proceso es documental y bibliográfica, esta permite que los estudios más relevantes e importantes ayuden a conocer las diversas teorías y perspectivas que estén relacionadas al tema en desarrollo.

Los instrumentos para usarse dentro de la investigación son las fichas técnicas, esta herramienta de investigación permite la identificación, clasificación, organización y el correcto registro de las diversas fuentes de información, que se encuentra. La ficha bibliográfica permite, que se tenga un enfoque de visión general y ordenado de las fuentes de donde se tomó la información.

Otra de las técnicas usadas fue la Revisión de registros, la elaboración del instrumento de registro no solo es una etapa del proceso de investigación, sino que, y principalmente, es un momento de la producción de los datos. Es un recurso del que dispone en toda investigación social con fuentes primarias para contribuir al pasaje del hecho al dato. Y esto lo posiciona como un recurso que exige especial atención por parte de quien lo construye Cohen & Gómez (2019). Para el proyecto los registros de información para la creación de la base de datos de cuerpos masculinos, se realizó mediante la toma de información de internet, con la precaución de que sean imágenes sin copyrigh.

Población y muestra

La población objetivo, que se beneficia del sistema son personas de género masculino interesadas en tener un estilo de vida saludable, al respecto, para la

toma de información para crear la base de datos de experimentación, no se trabajó directamente con dicha población, la información requerida es de uso delicado y en vista de ello, los datos se extrajeron de internet por cuanto se requirieron imágenes corporales frontales.

En base a lo explicado, no se calcula tampoco una muestra, pues como se explicó, la base de datos se construyó con imágenes obtenidas del internet, esto debido a que por cuestiones de seguridad no se lograron obtener de manera directa de los gimnasios de la ciudad, las imágenes empleadas para la presente son únicamente masculinas.

2.2. Metodología de Desarrollo

Para el desarrollo del proyecto, se trabajó con el Modelo de Prototipos, para empezar, se afirma que un prototipo es un modelo experimental ya sea de un componente o de un sistema, conformado por los elementos necesarios a fin de que este se use (Martín, Garibello, Rojas, 2021).

Jima et al.(2019), llega a determinar que uno de sus objetivos es que es un medio eficaz para que se aclaren los requerimientos de los usuarios para identificar las características que el prototipo tiene a fin de que se cambie, modifique o añada lo necesario para su correcto funcionamiento.

Por otra parte Arbey (2018), menciona que entre las características que tiene un prototipo está, que la aplicación sea funcional, se crea de manera rápida, este evoluciona por medio de un proceso iterativo, su desarrollo es barato y tiene como finalidad varios supuestos característicos de un sistema.

Enfoques de desarrollo

Existen teres tipos de enfoques en los que se desarrolla un prototipo, las características y diferencias se mencionan en la tabla, a continuación:

Tabla 1. Enfoques de desarrollo

Característica	Desechable	Evolutivo	Mixto
Enfoque de desarrollo	Sin rigor y rápido de elaborar	Riguroso	
¿Qué construir?	Solo partes complicadas	Se elabora sobre una base sólida, se desarrolla primero las partes bien entendidas	Se aplican técnicas convencionales para los requisitos bien conocidos.
Directrices del diseño	Optimizar el tiempo del desarrollo	Optimizar la modificación	Evolutivos para los requisitos poco conocidos.
Último objetivo	Desecharlo	Incluirlo en el sistema	Combinación de prototipos.

Fuente: tomado de García et al. (2019)

Para el desarrollo del proyecto, se utilizó el enfoque de prototipo desechable, debido a que según Intriago-Plaza et al. (2021) son ideales para identificar los requisitos del software, especialmente los puntos generales solicitados por el usuario, además, de que presenta cierta flexibilidad a la hora de introducir información. Por otra parte, tiene un proceso iterativo, esto sucede al momento prototipo ya está a punto de satisfacer todas las necesidades del beneficiario del *software*, por lo que la persona que lo desarrolla comprende mejor procesamiento del mismo. Este tipo de prototipos para García et al. (2019), se usan normalmente se aplican en: interfaz de usuario, formatos de informes, formatos de gráficos, organización de bases de datos, rendimiento de bases de datos, precisión e implementación de cálculos complejos, partes con respuesta crítica en el tiempo en sistemas de tiempo real, rendimiento de sistemas interactivos y la viabilidad de partes del sistema en las que no se tiene experiencia.

Imagen 4. Ejemplo del modelo



Fuente: tomado de García et al.(2019)

El desarrollo de prototipos en las palabras usadas por Vidal-Silva et al. (2021), representa un enfoque de desarrollo basado principalmente en la producción y adaptación del prototipo diseñado, todo esto en las fases antes referidas.

Por otra parte, Mendoza & Id, (2018) menciona que un prototipo tiene las siguientes formas:

- Es ejecutable en la computadora o de manera escrita, en donde se detallan el paso a paso del cómo se va a generar la interacción entre el hombre, la máquina y los listados de aplicación.
- Además, de que un prototipo, es aquel que realice un subconjunto de la función, que se necesita a fin de que se evalúe la utilidad de un algoritmo, velocidad o capacidad de almacenamiento del programa final.
- Otro de los tipos de prototipos que existen, esta aquel que tiene la capacidad de realizar la función requerida ya se de manera parcial o de manera íntegra, además, de que tenga características de consideración y buen rendimiento.

En el desarrollo de la presente investigación, es necesario que el prototipo ejecute de manera adecuada a fin de que por medio del análisis de imágenes corporales se logre alcanzar un sistema que brinde una guía nutricional.

Según Cadenas (2015) es necesario, que se use el modelo de prototipos, este modelo sustenta la prueba de manera anticipada a fin de determinar los errores, que a fin de ir probar las diferentes alternativas y mejorar las ideas para que el objetivo general planteado se cumpla, para esto se cumplirán las siguientes fases:

Fase 1: Obtención de requisitos

Pineda Ballesteros et al.,(2020) define a la obtención de requisitos de un sistema, como el conjunto de actividades, que se relacionan de manera

científica y documentada de las necesidades y limitantes de los usuarios o futuros beneficiarios del prototipo a desarrollarse, el autor afirma que existen las siguientes técnicas para la obtención de los requisitos: Prototipado, Análisis de interfaces, Storyboards, Casos de uso, Análisis de documentos, escenarios y Software heredado.

Esta primera fase de desarrollo consiste en: seleccionar, recolectar y hacer un análisis detallado de todos los elementos funcionales que tiene la aplicación, se toma como base la investigación inicial realizada, además, del hardware y software que van a formar parte del sistema.

En primera instancia mediante la técnica de análisis de registros, se recopilaron las imágenes de los diferentes tipos de cuerpos masculinos existentes, información que fue cargada en la Base de datos seleccionada.

Como elementos de software empleados fueron: Python, Google Colab, finalmente, los elementos del hardware usado fueron Computadora y cámara.

Dentro de los requisitos de la aplicación, se identificaron aspectos funcionales importantes para análisis de los diferentes tipos de cuerpos masculinos, y se tomó como base el análisis documental, así como información tomada de la internet, se requiere cumplir los siguientes aspectos:

- Creación de una base de datos con un conjunto de patrones de cuerpos masculinos, que sirve de línea base para el análisis
- La captura de imágenes en vivo de los usuarios que deseen analizar sus cuerpos, esta imagen se captura a través de la cámara del computador
- El análisis comparativo de las imágenes cargadas con la base de datos mediante la aplicación de algoritmos de inteligencia artificial y redes neuronales.
- El cálculo de distintos índices de análisis corporal para tener información importante para la toma de decisiones
- El valor agregado, que se requiere es que el sistema emita unas

sugerencias para mejorar el entrenamiento del usuario según su tipo de cuerpo, así como sugerencias nutricionales basadas en los datos obtenidos.

Fase 2: Diseño rápido

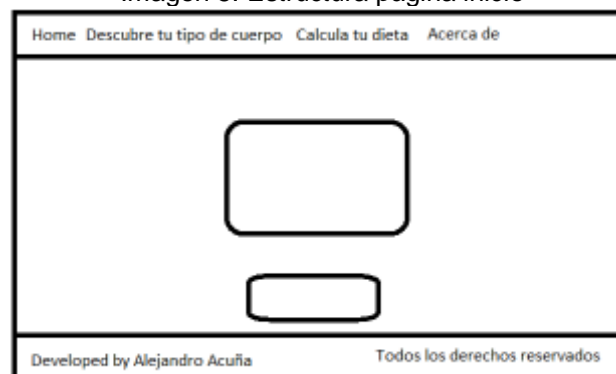
En este punto se determina como va a interactuar la herramienta software con el usuario. La idea general del sistema es funcionar como una herramienta de análisis de imágenes mediante inteligencia artificial, con el valor agregado de ser un asesor nutricional y de entrenamiento básico gracias a la información manipulada por el sistema.

La idea general del software es poder determinar el tipo de cuerpo que tiene el usuario del sistema, por esta razón la interfaz del mismo es sencilla, con exiguos elementos y muy intuitiva para un fácil manejo por parte del usuario.

Al partir del análisis anteriormente realizado, debido a que se estructuró una base de información fotográfica de cuerpos, la interfaz de la aplicación para interactuar sería la siguiente:

Página 1(Home)

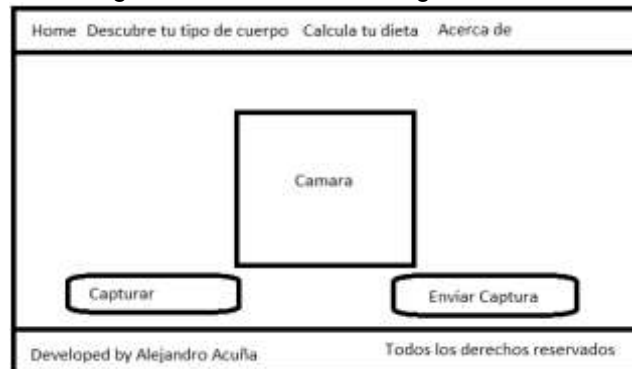
Imagen 5. Estructura página inicio



Fuente: elaboración propia

Página 2 (Descubre tu tipo de cuerpo)

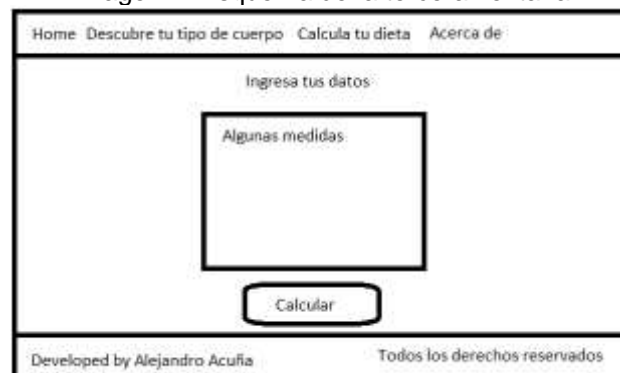
Imagen 6. Estructura de la segunda ventana



Fuente: elaboración propia

Página 3 (Calcula tu dieta)

Imagen 7. Esquema de la tercera ventana.



Fuente: elaboración propia

Fase 3: Construcción del prototipo

Para la construcción del prototipo, se seleccionaron las siguientes herramientas:

Phyton

El lenguaje Phyton creado por Guido Van Rossum, está basado en el lenguaje de programación ABC, en la actualidad es uno de los más usados, especialmente en el área educativa, lo que caracteriza a este sistema es su sintaxis simple, alta legibilidad, interpreta interactivamente lo que el usuario requiere o necesita, genera un mayor nivel de expresividad, cuenta con una librería estándar de gran calidad que permite el uso de librerías de externos,

además, de que es de libre acceso.

Tabla 2. Sistema Phyton requerimientos de hardware y *software*

Sistema Phyton	
¿Qué es?	Es el lenguaje más popular en la elaboración de herramientas visuales, tanto para la programación directa como para la de visualización de la ejecución de programas.
Requerimientos mínimos de hardware	Procesador de doble núcleo de 2,66 Ghz. Bus USB 2.0 2 Gb de RAM
Requerimientos de <i>software</i>	Windows 7 u 8 (32,64 Bits) Visual Studio 2010 o versiones superiores

Fuente: tomado de García Monsálvez, (2017)

En el caso de Phyton el enfoque es evolutivo, este se basa en los ajustes arquitectónicos, que se hacen en base a toda la información, que se obtiene del internet, por lo tanto, este se expaña innumerable número de veces.

Imagen 8. Sistema Phyton



Fuente: tomado de Kanani & Padole, (2019)

Phyton es 100% compatible para ser programada en cualquier lenguaje incluso en entornos táctiles como smartphones y tablets por lo que se implementa en un computador con un sistema operativo Windows, como es de acceso libre el costo de su aplicación es inferior con relación a otros sistemas de igual características.

Google Colab

Para el presente trabajo, se empleó Google Colab el cual, es un Notebook que permite programar y ejecutar Python en el navegador. Los Notebooks de Colab permiten combinar código ejecutable y texto enriquecido en un mismo

documento, además, de imágenes, HTML y LaTeX. Los cuadernos que creas en Colab se almacenan en tu cuenta de Google Drive.

Con Colab, se importa un conjunto de datos de imágenes (*dataset*), entrenar un clasificador de imágenes con dicho conjunto de datos y evaluar el modelo con tan solo unas pocas líneas de código. Los notebooks de Colab ejecutan código en los servidores en la nube de Google, lo que permite aprovechar la potencia del hardware de Google, incluidas las GPU y TPU, independientemente de la potencia del equipo, que se conecta al navegador.

Imagen 9. Google Colab



Fuente: tomado de Kanani & Padole, (2019)

Colab es una herramienta muy utilizada en la comunidad de aprendizaje automático. Permite utilizar las librerías de TensorFlow para desarrollar y entrenar redes neuronales. En la siguiente ilustración se puede observar a facilidad de esta herramienta para importar *datasets* desde tensor Flow. Además, de otras librerías importantes para el entrenamiento de redes neuronales para visión artificial. Tales como matplotlib y cv2.

A continuación, se muestran los requerimientos para trabajar con esta herramienta:

Google Colab

Tabla 3. Google Colab

Google Colab	
¿Qué es?	Es una herramienta usada en la comunidad de aprendizaje automático, es de utilidad para importar data set desde Tensor Flow para desarrollar y entrenar redes neurales.
Requerimientos mínimos de hardware	Intel i5-4590 o AMD FX 8350 o Superior
	2 GB de espacio disponible
	4 Gb de RAM
Requerimientos de <i>software</i>	Windows 7 SP1
	Nvidia GeForce GTX970 o superior

Fuente: tomado de Kanani & Padole, (2019)

Visual Studio Code

Según López (2019) es un software desarrollado por Microsoft Windows, macOS y Linux, este viene incluido con el soporte para depurar, además, de tener el control integrado de Git, resaltado de sintaxis, finaliza inteligentemente el código introducido por el diseñador, los fragmentos y por último este software permite la refactorización del código si fuese necesario.

Tabla 4. Visual Studio Code

Visual Studio Code	
¿Qué es?	Es un software desarrollado por Microsoft Windows, macOS y Linux, este viene incluido con el soporte para depurar, además, de tener el control integrado de Git, resaltado de sintaxis, finaliza inteligentemente el código introducido por el diseñador, los fragmentos y por último este software permite la refactorización del código si fuese necesario.
Requerimientos mínimos de hardware	Intel i5-4590 o AMD FX 8350 o Superior
	1 GB de espacio disponible
	1 Gb de RAM
Requerimientos de <i>software</i>	Windows 10 en ARM, Windows 10 Enterprise LTSC, Windows Server 2012 y Windows Server 2008 R2 SP1
	En el caso de x86 o AMD64/x64, requiere un procesador de 1,6 GHz o superior

Fuente: tomado de López, (2019)

Imagen 10. Visual Studio Code



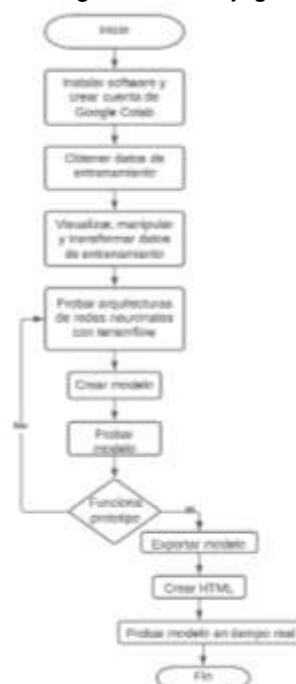
Fuente: tomado de López, (2019)

Visual Studio Code

Es un lenguaje de programación, multiparadigma y multinivel, además, de que cuenta con un soporte en programación direccionada a objetivos de manera imperativa y funcional. Con este tipo de lenguaje, se crea aplicaciones nativas e híbridas, por lo que es fácil de acceder por su sencillo lenguaje de programación.

Una vez analizado de manera general de cada uno de los elementos requeridos para el prototipo del sistema, el siguiente paso es integrarlos.

Imagen 11. Diagrama de Flujoograma



Fuente: elaboración propia

Creación del DataSet

En primer lugar, se prepara la data set, que sirva para entrenar el modelo que clasifica los tipos de cuerpo. Para lograrlo, se realizan los siguientes pasos:

Se definen los parámetros a tener en cuenta para la selección de imágenes. En este caso, para realizar la clasificación de tipos de cuerpo, es necesario que las imágenes muestren el torso completo de las personas, sin ningún tipo de vestimenta y en posición frontal.

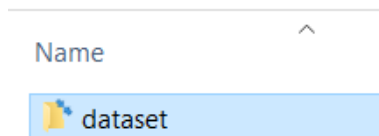
Se realiza la recolección de imágenes a través de formularios de Google, y descargas directas del internet.

Una vez se cuenta con las imágenes se estructura el *dataset*:

1. Se crea la carpeta que contiene el *dataset*.

Imagen 12. Carpeta del dataset

Proyectos > Tesis

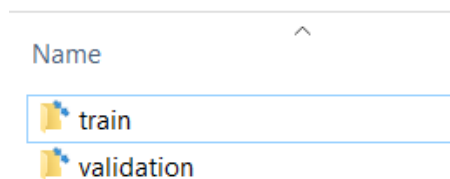


Fuente: elaboración propia

2. Se divide entre datos de entrenamiento (train) y datos de validación(validation). Los datos de validación representan un 15-20% del total de los datos recolectados.

Imagen 13. Datos recolectados

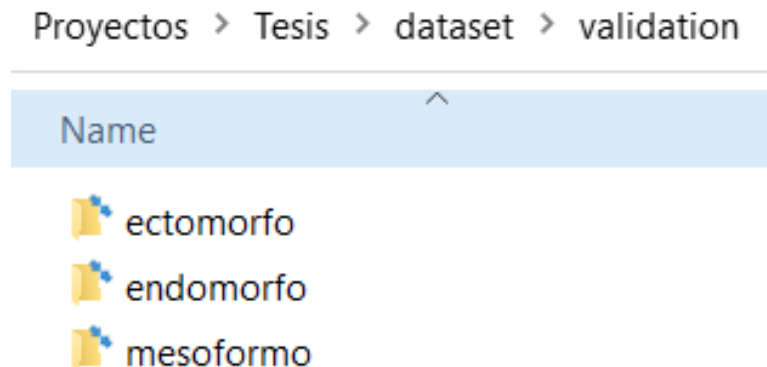
Proyectos > Tesis > dataset



Fuente: elaboración propia

3. Dentro de cada carpeta (“train” y “validation”), se crean las clases, una por cada tipo de cuerpo: endomorfo, ectomorfo y mesomorfo. Es decir, se crea una carpeta por cada tipo de cuerpo. Se asegura que los nombres de clases sean exactamente los mismos tanto para los datos de entrenamiento como para los datos de validación.

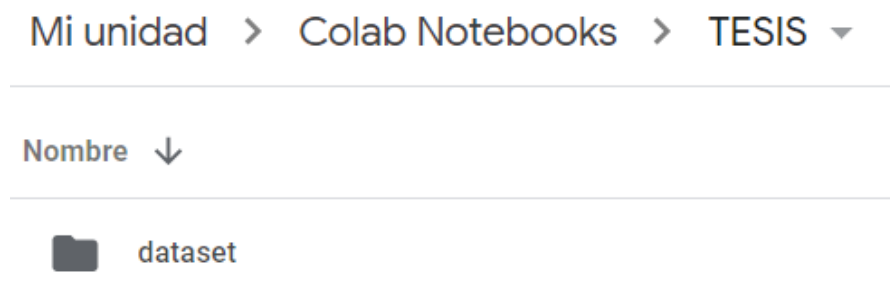
Imagen 14. Tipos de cuerpo



Fuente: elaboración propia

4. Se guarda las imágenes recolectadas dentro de cada carpeta según el tipo de cuerpo.
5. Se carga el *dataset* a Google drive.

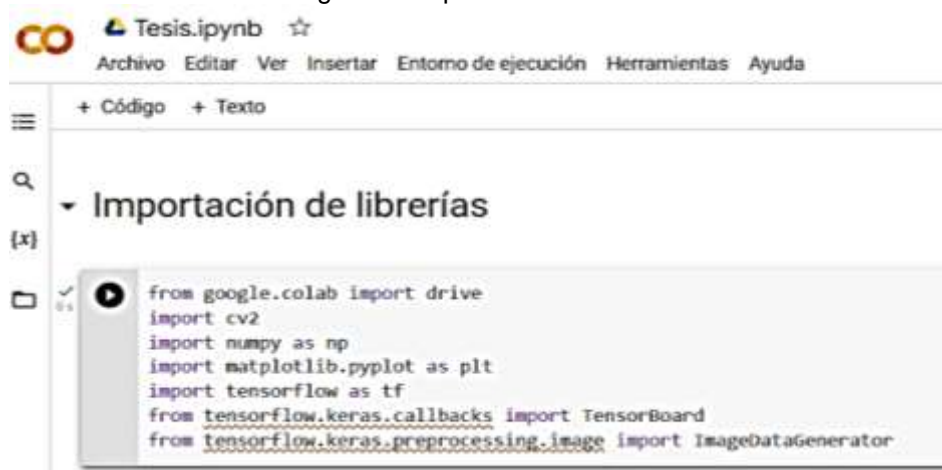
Imagen 15. Cargar el dataset a Google drive.



Fuente: elaboración propia

Ahora el *dataset* está cargado y listo para usarse en los siguientes pasos. A continuación, se abre un notebook de Google Colab e importamos las librerías necesarias para el desarrollo del proyecto:

Imagen 16. Importación de librerías



```

from google.colab import drive
import cv2
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.callbacks import TensorBoard
from tensorflow.keras.preprocessing.image import ImageDataGenerator

```

Fuente: elaboración propia

Librerías usadas

Drive de Google Colab: Permite activar Google Drive en el entorno de ejecución de Google Colab para acceder a sus archivos en la nube. Esto representa una ventaja respecto a la carga local de archivos, debido a que no es necesario cargar el set de datos cada vez que se desee correr el modelo.

Cv2: Es una librería de OpenCV originalmente desarrollada por Intel. OpenCV significa Open Computer Vision (Visión Artificial Abierta). Permite el análisis y tratamiento de datos.

Numpy: NumPy es un paquete de Python que significa “Numerical Python”, es la librería principal para la informática científica, proporciona potentes estructuras de datos, implementa matrices y matrices multidimensionales. Estas estructuras de datos garantizan cálculos eficientes con matrices. (<https://aprendeia.com/introduccion-a-numpy-python-1/>)

matplotlib.pyplot: es una colección de funciones que hacen que matplotlib funcione como MATLAB. Cada función de pyplot realiza algún cambio en una figura: por ejemplo, crea una figura, crea un área de trazado en una figura, traza algunas líneas en un área de trazado, decora el trazado con etiquetas, etc (<https://matplotlib.org/3.5.0/tutorials/introductory/pyplot.html>).

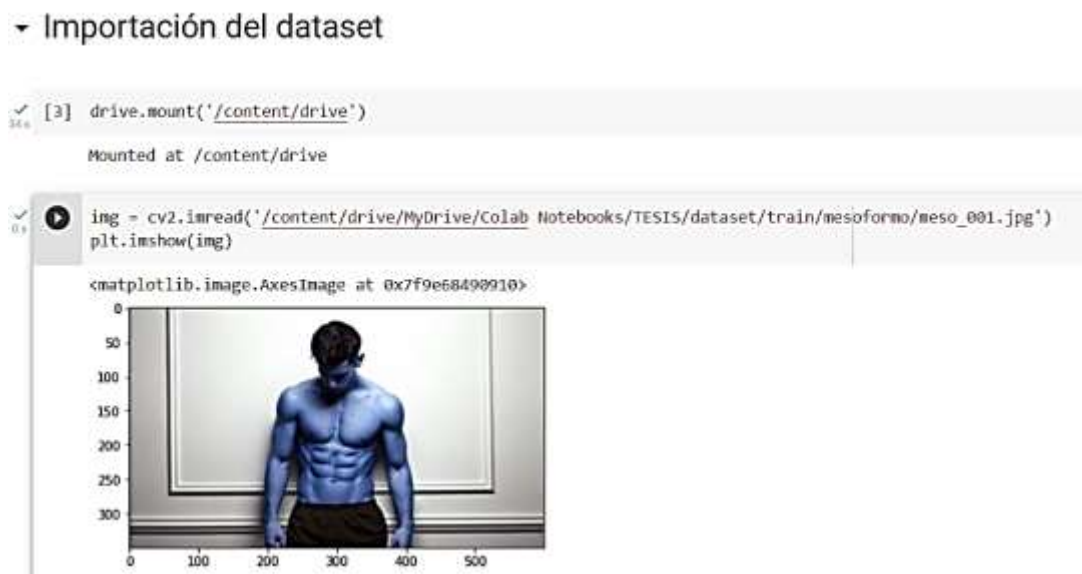
Tensorflow: es la principal biblioteca de código abierto para desarrollar y entrenar modelos de IA. (<https://www.tensorflow.org/?hl=es-419>)

tensorboard: proporciona la visualización y las herramientas necesarias para experimentar con el aprendizaje automático. En este proyecto se usará para evaluar el modelo entrenado.

ImageDataGenerator: es un paquete de TensorFlow que permite generar lotes de datos de imágenes con aumento de datos en tiempo real.

Una vez se han importado las librerías, se procede a montar el drive en el entorno de ejecución para poder acceder al *dataset*. Mediante el uso de cv2 se muestra una imagen del set de datos.

Imagen 17. Importación dataset



Fuente: elaboración propia

Como se observa en la ilustración, al leer un archivo de imagen con la función `imread()` de OpenCV, el orden de colores es BGR (blue, green, red – azul, verde, rojo). Sin embargo, la función `imshow()` de matplotlib lee la imagen en el orden de colores RGB. Por lo que es necesario convertir los colores con la función `cvtColor` de OpenCV.

Imagen 18. Conversión de colores



Fuente: elaboración propia

Debido a la pequeña cantidad de datos, se realiza un aumento de datos. La técnica de aumento de imágenes es una buena forma de ampliar el tamaño del dataset. De esta manera, se obtienen nuevas imágenes a partir del dataset original. Es común emplear un enfoque conservativo a la hora de aumentar imágenes, es decir, aumentar las imágenes y almacenarlas éstas en un array de Numpy o en una nueva carpeta. Sin embargo, en este proyecto se emplea otro enfoque a través de la clase ImageDataGenerator de Keras.

Esta clase permite aumentar las imágenes en tiempo real mientras el modelo continúa en entrenamiento. Se aplican transformaciones aleatorias a cada imagen del *dataset* de entrenamiento, esta se pasa al modelo. Esto permite conseguir un modelo más robusto, también, ahorrar memoria. De esta manera se crea un gran grupo de imágenes similares sin la necesidad de recolectar nuevas imágenes.

La clase ImageDataGenerator de Keras se asegura de que el modelo reciba nuevas variaciones de las imágenes del dataset en cada época. Pero solamente regresa las imágenes transformadas y no las añade al conjunto de datos original. De esta manera, se evita un posible overfitting.

De acuerdo con la metodología utilizada, se ha documentado en el presente capítulo la parte correspondiente al Análisis de Requisitos, al diseño y al proceso de construcción del prototipo, y en el capítulo III se muestra el Refinamiento del prototipo, la Evaluación y el producto final.

A continuación, se presentan algunas opciones de aumento de datos:

Rotaciones aleatorias: permite rotar aleatoriamente las imágenes en un rango de 0 a 360 grados. El iterador Flow del ImageDataGenerator acepta vectores con 4 dimensiones. Las imágenes del set de datos tienen 3 dimensiones por lo que se incrementa una dimensión a la imagen con Numpy.

Imagen 19. Dimensión de la imagen



The screenshot shows a Jupyter Notebook interface with the following content:

- Header: Tesis.ipynb
- Menu: Archivo, Editar, Ver, Insertar, Entorno de ejecución, Herramientas, Ayuda
- Buttons: + Código, + Texto
- Image: A photograph of a shirtless man standing in front of a whiteboard, with a coordinate grid overlaid on it. The x-axis ranges from 0 to 500, and the y-axis ranges from 0 to 300.
- Section: Aumento de datos
- Code Cell [22]: `img.shape` resulting in `(350, 600, 3)`
- Code Cell [25]: `#aumentar una dimension al array de la imagen`
`ex_img = np.expand_dims(img, 0)`
`ex_img.shape` resulting in `(1, 350, 600, 3)`

Fuente: elaboración propia

A continuación, se crea el generador de imágenes con el nombre de clase ImageDataGenerator y se le asigna un valor de 90 a "rotation_range". Lo que quiere decir, que se generaran imágenes con una rotación aleatoria entre 0 a 90 grados.

Se crea el iterador con el generador previamente creado. Se le pasan los datos que emplea como base para generar aumentos (`ex_img`) y el tamaño del lote (`batch_size`) en este caso como solamente se experimentó con una imagen se le asigna el valor de 1.

Se asigna el tamaño de la figura que contenga la muestra de imágenes generadas para tener una mejor visualización. Se usa `figure` de `matplotlib`.

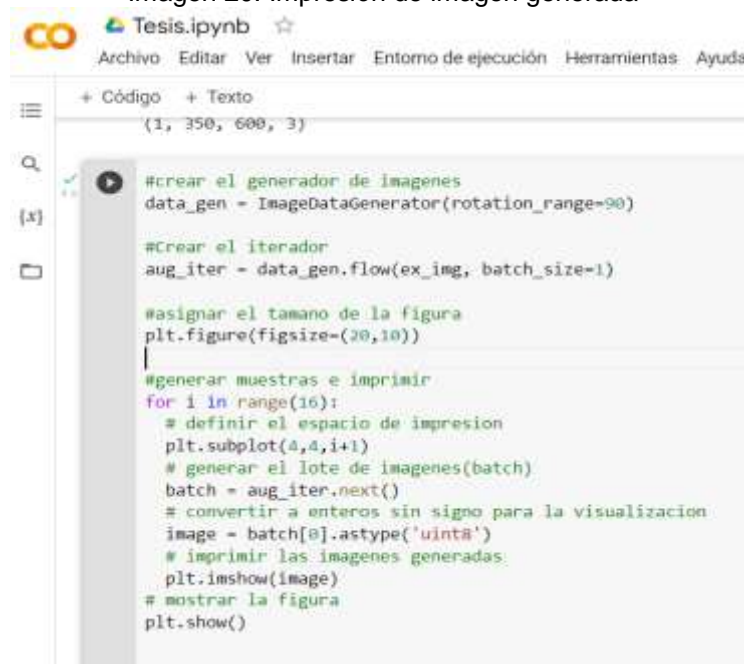
Se quiere generar 16 aumentos a partir de la imagen original, por lo tanto, se crea un ciclo `for` con un rango de 16.

A continuación, se define el espacio de impresión para cada imagen en cada iteración. Con `subplot`, se especifica, que se desean 4 imágenes en el eje x, 4 en el eje y y finalmente, la posición en la que se imprime cada imagen.

Se genera el lote de imágenes mediante el uso del comando `next()` para recorrer el iterador.

El lote se convierte a enteros sin signos para que se visualice con `matplotlib`. Para esto se emplea `astype("uint8")`. Finalmente, se imprime la imagen generada. Una vez completado el ciclo `for` se muestra la figura.

Imagen 20. Impresión de imagen generada



```

Tesis.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

+ Código + Texto
(1, 350, 600, 3)

#crear el generador de imagenes
data_gen = ImageDataGenerator(rotation_range=90)

#Crear el iterador
aug_iter = data_gen.flow(ex_img, batch_size=1)

#asignar el tamaño de la figura
plt.figure(figsize=(20,10))

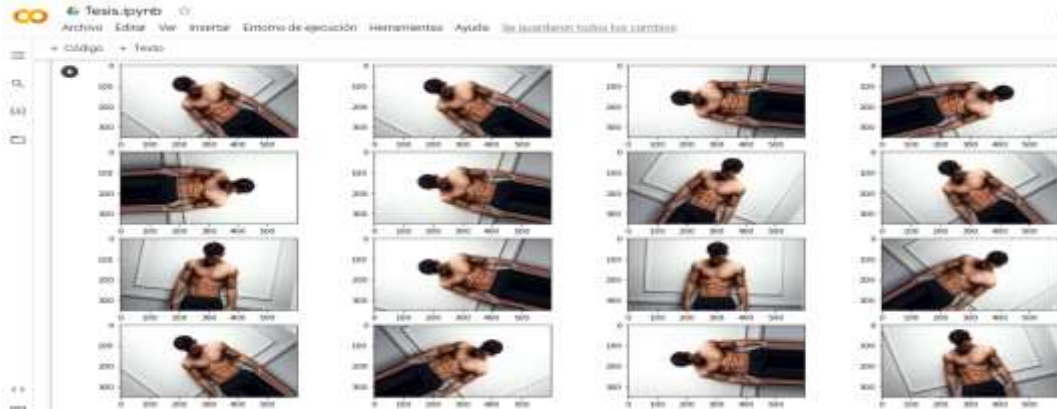
#generar muestras e imprimir
for i in range(16):
    # definir el espacio de impresion
    plt.subplot(4,4,i+1)
    # generar el lote de imagenes(batch)
    batch = aug_iter.next()
    # convertir a enteros sin signo para la visualizacion
    image = batch[0].astype('uint8')
    # imprimir las imagenes generadas
    plt.imshow(image)
# mostrar la figura
plt.show()

```

Fuente: elaboración propia

Como se observa 90 grados es un valor demasiado elevado y genera imágenes con una rotación que no se encontró en situaciones reales.

Imagen 21. Rotación de imágenes



Fuente: elaboración propia

Un valor de 20 grados es mucho más real y se adapta mejor a este caso concreto de estudio.

Imagen 22. Ajuste de la rotación



Fuente: elaboración propia

Aumento de datos

Debido a la pequeña cantidad de datos, se realiza un aumento de datos.

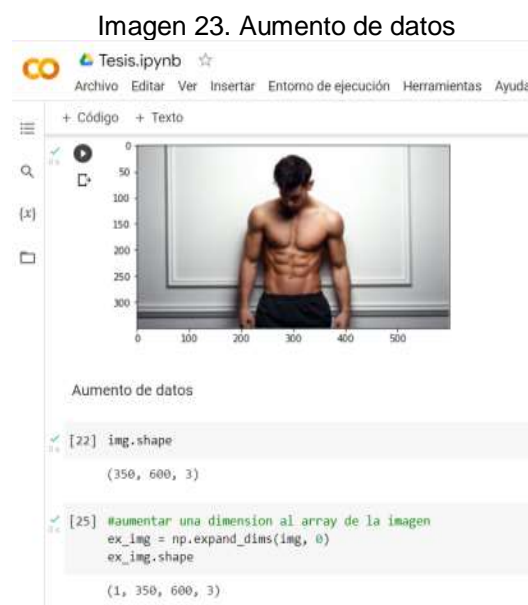
La técnica de aumento de imágenes es una buena forma de ampliar el tamaño del dataset. De esta manera, se obtienen nuevas imágenes a partir del dataset original.

Es común emplear un enfoque conservativo a la hora de aumentar imágenes, es decir, aumentar las imágenes y almacenarlas éstas en un array de Numpy o en una nueva carpeta. Sin embargo, en este proyecto se emplea otro enfoque a través de la clase ImageDataGenerator de Keras. Esta clase permite aumentar las imágenes en tiempo real mientras el modelo continúa en entrenamiento. Se aplican transformaciones aleatorias a cada imagen del dataset de entrenamiento al pasarse al modelo. Esto permite conseguir un modelo más robusto y también, ahorrar memoria. De esta manera, se crea un gran grupo de imágenes similares sin la necesidad de recolectar nuevas imágenes.

La clase ImageDataGenerator de Keras, se asegura de que el modelo reciba nuevas variaciones de las imágenes del dataset en cada época. Pero solamente regresa las imágenes transformadas y no las añade al conjunto de datos original. De esta manera se evita un posible overfitting.

A continuación, se presentan algunas opciones de aumento de datos:

El iterador Flow del ImageDataGenerator acepta vectores con 4 dimensiones. Las imágenes del set de datos tienen 3 dimensiones por lo que se incrementa una dimensión a la imagen con Numpy. Rotaciones aleatorias: permite rotar aleatoriamente las imágenes en un rango de 0 a 360 grados.



Fuente: elaboración propia

A continuación, se crea el generador de imágenes denominado a la clase `ImageDataGenerator` y se le asigna un valor de 90 a `rotation_range`. Lo que quiere decir, que se generaran imágenes con una rotación aleatoria entre 0 a 90 grados.

Se crea el iterador con el generador previamente creado. Se le pasan los datos que emplea como base para generar aumentos (`ex_img`) y el tamaño del lote (`batch_size`) en este caso como solamente se experimentó con una imagen se le asigno el valor de 1.

Se asigna el tamaño de la figura que contenga la muestra de imágenes generadas para tener una mejor visualización. Se usa `figure` de `matplotlib`.

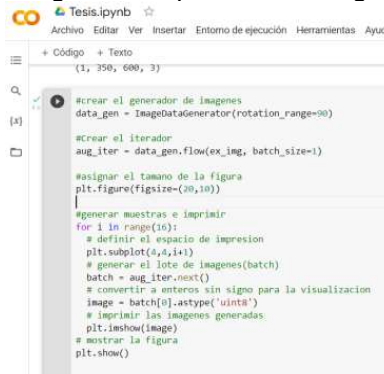
Se quiere generar 16 aumentos a partir de la imagen original, por lo tanto, se crea un ciclo `for` con un rango de 16.

A continuación, se define el espacio de impresión para cada imagen en cada iteración. Con `subplot`, se especifica, que se desean 4 imágenes en el eje x, 4 en el eje y finalmente, la posición en la que se imprime cada imagen.

Se generó el lote de imágenes mediante el uso del comando `next()` para recorrer el iterador.

El lote se convierte a enteros sin signos para que se visualice con `matplotlib`. Para esto, se emplea `astype("uint8")`. Finalmente, se imprime la imagen generada. Una vez completado el ciclo `for` se muestra la figura.

Imagen 24. Impresión de imagen



```

Tesis.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
+ Código + Texto
(1, 350, 600, 3)
#crear el generador de imagenes
data_gen = ImageDataGenerator(rotation_range=90)

#crear el iterador
aug_iter = data_gen.flow(ex_img, batch_size=1)

#asignar el tamaño de la figura
plt.figure(figsize=(20,10))

#generar muestras e imprimir
for i in range(16):
    #definir el espacio de impresion
    plt.subplot(4,4,i+1)
    #generar el lote de imagenes(batch)
    batch = aug_iter.next()
    #convertir a enteros sin signo para la visualizacion
    image = batch[0].astype('uint8')
    #imprimir las imagenes generadas
    plt.imshow(image)
#mostrar la figura
plt.show()

```

Fuente: elaboración propia

Como se observa 90 grados es un valor demasiado elevado y genera imágenes con una rotación que no se encuentra en situaciones reales.

Desplazamientos aleatorios

Aplicar un desplazamiento a una imagen significa mover todos los píxeles de la imagen en una dirección, son horizontal o verticalmente, mientras que las dimensiones de la imagen se mantienen.

Los argumentos `width_shift_range` y `height_shift_range` del constructor `ImageDataGenerator` controlan la cantidad de desplazamiento vertical u horizontal.

Imagen 25. Calidad de desplazamiento

```
data_gen2 = ImageDataGenerator(width_shift_range=0.2, height_shift_range=0.2)
iter = data_gen2.flow(ex_img, batch_size=1)

plt.figure(figsize=(20,10))

for i in range(16):
    plt.subplot(4,4,i+1)
    batch = iter.next()
    image = batch[0].astype('uint8')
    plt.imshow(image)

plt.show()
```

Fuente: elaboración propia

Al igual que en las rotaciones se tiene cuidado con la magnitud del cambio. En este caso el 20% para desplazamientos verticales resulta perjudicial debido a que se pierden detalles importantes como las caderas, que es una de las características fundamentales para la clasificación de los tipos de cuerpo.

Imagen 26. Características fundamentales en la clasificación de imágenes



Fuente: elaboración propia

Se realiza el cambio hacia un 10% en desplazamientos verticales y se obtiene un resultado mejor para el presente proyecto

Imagen 27. Desplazamiento de imagen



Fuente: elaboración propia

Volteos aleatorios

Un volteo de imagen significa invertir las filas o columnas de píxeles en un volteo vertical u horizontal respectivamente. La clase ImageDataGenerator tiene los parámetros: `horizontal_flip` y `vertical_flip` para voltear a lo largo del eje vertical u horizontal.

Imagen 28. Volteos aleatorios

```
data_gen3 = ImageDataGenerator(horizontal_flip=True, vertical_flip=True)
iter = data_gen3.flow(ex_img, batch_size = 1)
plt.figure(figsize=(20,10))

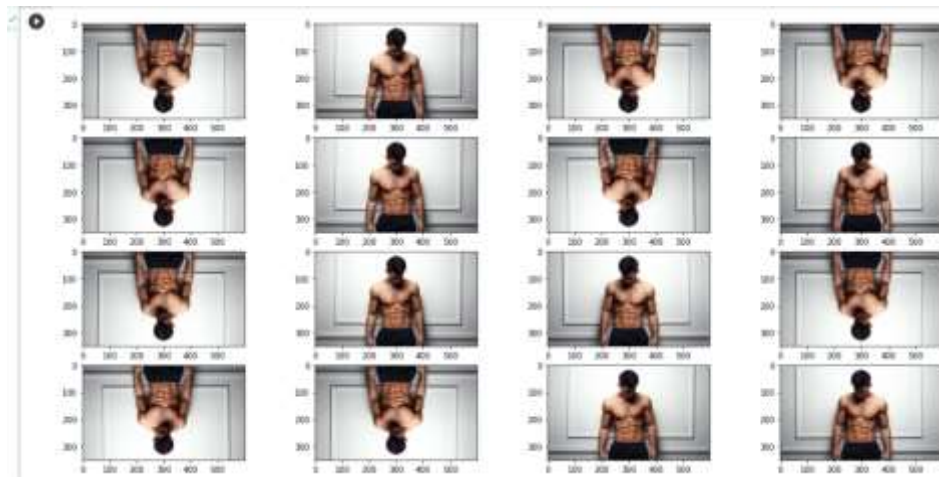
for i in range(16):
    plt.subplot(4,4,i+1)
    batch = iter.next()
    image = batch[0].astype('uint8')
    plt.imshow(image)

plt.show()
```

Fuente: elaboración propia

Como se ve en el código para aplicar estos volteos empleamos la sentencia (`horizontal_flip` y `vertical_flip`) y se da el valor de `true`. Los resultados se muestran en la siguiente imagen:

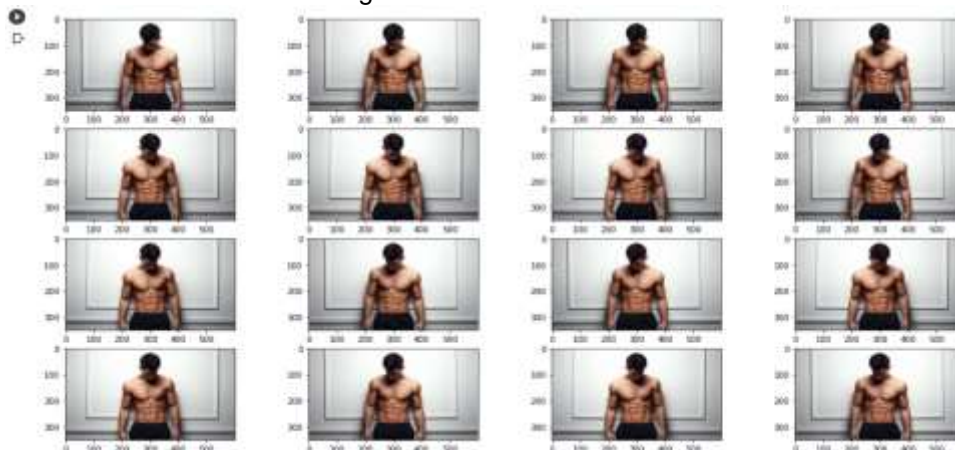
Imagen 29. Volteos aleatorios



Fuente: elaboración propia

Como se observa un volteo vertical no tiene sentido en el presente caso de estudio. Por lo tanto, se omite este valor y se obtiene el siguiente resultado.

Imagen 30. Volteos aleatorios



Fuente: elaboración propia

Brillo aleatorio

Es una técnica de aumento muy útil porque muchas veces el objeto a detectar no está bajo condiciones de iluminación perfectas. Por lo tanto, es muy importante entrenar el modelo con imágenes bajo diferentes condiciones de iluminación.

El brillo se controla en la clase ImageDataGenerator a través del parámetro `brightness_range`. Acepta una lista de dos valores flotantes y escoge un valor

de cambio de brillo de ese rango. Los valores inferiores a 1,0 oscurecen la imagen, mientras que los valores superiores a 1,0 la aclaran. (<https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>)

Imagen 31. Brillo aleatorio

```
data_gen3 = ImageDataGenerator(brightness_range=[0.3,1.3])
iter = data_gen3.flow(ex_img, batch_size=1)
plt.figure(figsize=(20,10))

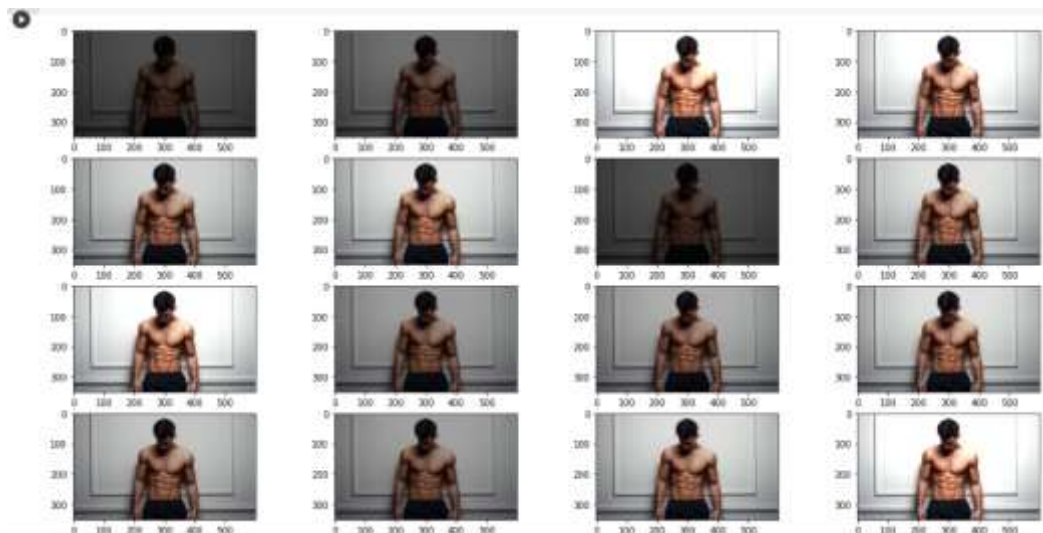
for i in range(16):
    plt.subplot(4,4,i+1)
    batch = iter.next()
    image = batch[0].astype('uint8')
    plt.imshow(image)

plt.show()
```

Fuente: elaboración propia

Se definen valores de 0.3 y 1.3 para la variación del brillo y se obtienen resultados satisfactorios, que se muestran, a continuación.

Imagen 32. Brillo aleatorio



Fuente: elaboración propia

Zoom aleatorio

Un aumento del zoom amplía aleatoriamente la imagen y añade nuevos valores de píxeles alrededor de la imagen o interpola los valores de los píxeles respectivamente. El porcentaje del zoom se especifica como un float o un

rango como una matriz o tupla. Si se especifica un float, el rango del zoom es $[1-\text{valor}, 1+\text{valor}]$. Por ejemplo, si se especifica 0,3, entonces el rango es $[0,7, 1,3]$, o entre el 70% (zoom in) y el 130% (zoom out).

Imagen 33. Zoom aleatorio

```

▶ data_gen4 = ImageDataGenerator(zoom_range=0.3)

iter = data_gen4.flow(ex_img, batch_size=1)

plt.figure(figsize=(20,10))

for i in range(16):
    plt.subplot(4,4,i+1)
    batch = iter.next()
    image = batch[0].astype('uint8')
    plt.imshow(image)

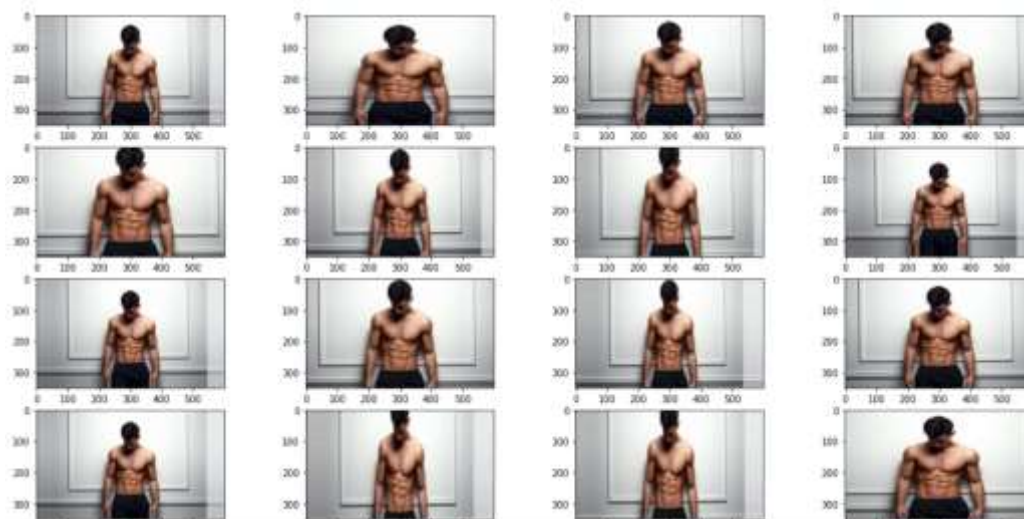
plt.show()

```

Fuente: elaboración propia

El resultado de aplicar zoom resulta perjudicial para el entrenamiento del modelo debido a que las proporciones de la imagen se distorsionan, ocasiona que un ectomorfo se transforme en un mesomorfo. Por lo tanto, no se aplica este aumento de datos

Imagen 34. Zoom aleatorio.



Fuente: elaboración propia

Finalmente, se aplican los cuatro tipos de aumento de imágenes revisadas previamente.

Imagen 35. Aumento de imágenes

```

#Aplicación conjunta de las técnicas de aumento de datos
data_gen_final = ImageDataGenerator(rotation_range=30,
                                   width_shift_range=0.2,
                                   height_shift_range=0.1,
                                   brightness_range=[0.3,1.3],
                                   horizontal_flip=True)

iter = data_gen_final.flow(ex_img, batch_size=1)

plt.figure(figsize=(20,30))

for i in range(10):
    plt.subplot(0,0,i+1)
    batch = iter.next()
    image = batch[0].astype('uint8')
    plt.imshow(image)

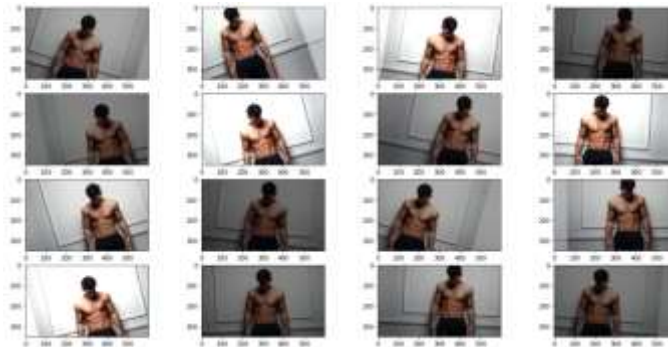
plt.show()

```

Fuente: elaboración propia

Los aumentos realizados a la imagen a través de ImageDataGenerator se los pasa a un interador el cual, muestra los siguientes resultados.

Imagen 36. Aumento de imágenes



Fuente: Autor

Se genera unas tantas iteraciones como se desee. Se observa, como de una sola imagen, se genera 36 aumentos de imágenes que robustecen el modelo.

Imagen 37. Incremento de imágenes



Fuente: elaboración propia

Una vez realizadas las pruebas de aumento de datos se procede a aplicar las técnicas a los datos, que es usado en el modelo. Adicionalmente, se toman las siguientes consideraciones:

- Se re escala la imagen pasar los pixeles de un valor entre 0-255 a un valor entre 0-1.
- Se normalizó todos los datos a una altura y ancho predeterminados.
- Se asignó un batch size o tamaño de lotes, que se pasarán al modelo.
- Se estableció el modo de color en escala de grises para facilitar el entrenamiento, debido a que para detectar el tipo de cuerpo no importa el color de la piel del usuario.

Imagen 38. Incremento de datos

▾ Aumento de datos para el modelo

▾ Creacion de generadores

```
[36] #Creacion de los generadores para el modelo
train_aug = ImageDataGenerator(      rescale = 1/255,
                                     rotation_range=20,
                                     width_shift_range=0.2,
                                     height_shift_range=0.1,
                                     brightness_range=[0.3,1.3],
                                     horizontal_flip=True)

validation = ImageDataGenerator(rescale = 1/255)
```

Fuente: elaboración propia

A continuación, se genera el iterador que tomara las imágenes del *dataset* y aplicara los aumentos definidos anteriormente en el generador para pasarlos al modelo.

Imagen 39. Incrementar datos

```
#dimensiones de las imagenes
img_height = 230
img_width = 200
batch_size = 32

#creacion del iterador para ser usado en el modelo

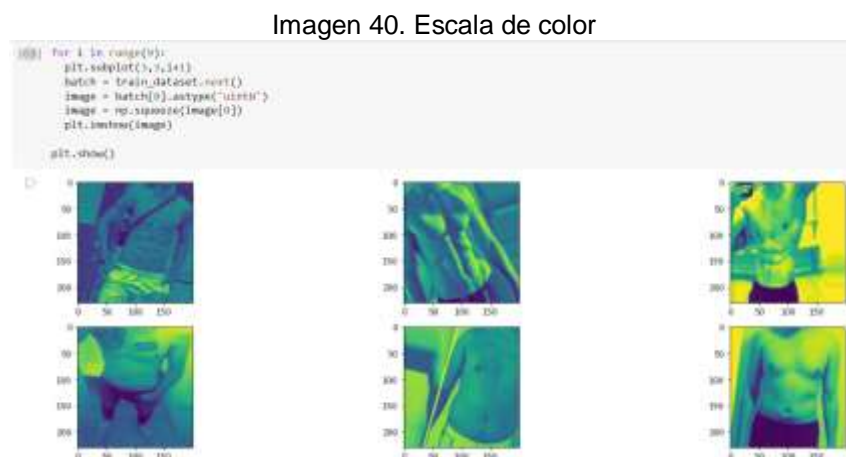
train_dataset = train_aug.flow_from_directory('/content/drive/MyDrive/Colab Notebooks/TESIS/dataset/train',
                                             target_size = (img_height,img_width),
                                             batch_size = batch_size,
                                             shuffle=True,
                                             color_mode = 'grayscale',
                                             class_mode = 'categorical')

validation_dataset = validation.flow_from_directory('/content/drive/MyDrive/Colab Notebooks/TESIS/dataset/validation',
                                                  target_size = (img_height,img_width),
                                                  color_mode = 'grayscale',
                                                  class_mode = 'categorical')
```

Found 337 images belonging to 3 classes.
Found 68 images belonging to 3 classes.

Fuente: elaboración propia

Se imprime una muestra de los datos generados a través de ImageDataGenerator para verificar su correcto funcionamiento. Es importante destacar dos aspectos: en primer lugar, para poder observar la muestra se deshabilita rescale. En segundo lugar, si bien en color mode, se establece el modo de color en escala de grises, el modelo matplotlib interpreta que solo existe un canal de colores y toma el valor medio de RGB en este caso G(green) por esta razón, se visualiza en color verde.



Fuente: elaboración propia

A continuación, se accede a los índices de cada clase creada a través de `.class_indices`. Esto arroja como resultado un diccionario cuya llave es el nombre de la clase y su valor el índice de esta. Este valor ayuda al momento de exportar el modelo al navegador.

Imagen 41. Clases creadas

```

[54] train_dataset.class_indices
      {'ectomorfo': 0, 'endomorfo': 1, 'mesomorfo': 2}
  
```

Fuente: elaboración propia

Creación del Modelo

Para la creación del modelo si bien existen sugerencias no existe una metodología completamente definida por lo que es necesario experimentar y observar los resultados.

Se emplea keras de tensorflow para crear el modelo y se selecciona un modelo secuencial, es decir, que las capas de neuronas se colocarán una a continuación de otra secuencialmente.

Para la primera prueba, se define la capa de entrada con el tamaño de las imágenes normalizadas. A continuación, se definen 3 capas ocultas con neuronas convolucionales con su respectiva capa de agrupación. Se aplanan las capas para pasarlas a una capa densa para, finalmente, pasar a la capa de salida.

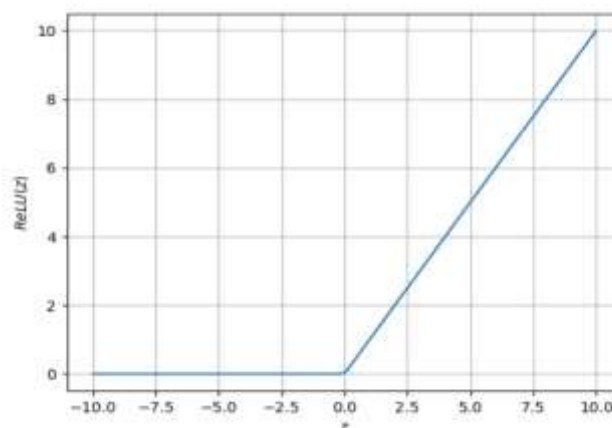
En el presente proyecto, se emplean las funciones de activación RELU para las capas ocultas y una función de activación SOFTMAX para la capa de salida. El principal inconveniente de otras funciones de activación como la sigmoideal y tanh es la saturación de sus valores de salida. Esto dificulta el proceso de entrenamiento al no permitir la rápida minimización de la función de error.

La función de activación RELU (Rectified Linear Unit) generó una salida igual a cero, la entrada (z) al ser negativa y una salida igual a la entrada la misma que es positiva.

$$\phi = \max(0, x)$$

El comportamiento de esta función se muestra en la siguiente figura:

Imagen 42. Comportamiento de la función



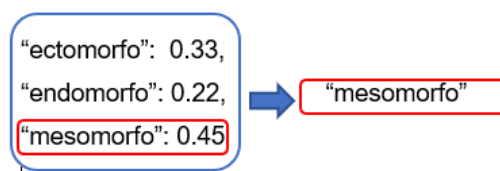
Fuente: elaboración propia

La capa de salida emplea una función de activación softmax que soporta sistemas de clasificación multinomial, por lo que, se convierte en el recurso principal utilizado en las capas de salida de un clasificador como en el caso del presente proyecto. Esta función de activación devuelve la distribución de probabilidad de cada una de las clases soportadas en el modelo. La función Softmax calcula la distribución de probabilidades del evento sobre 'n' eventos diferentes. En términos generales, esta función calcula las probabilidades de cada clase objetivo sobre todas las clases objetivo-possibles.

$$\phi(x_i) = \frac{e^{x_i}}{\sum_{j=0 \dots k} e^{x_j}} \quad i = 0, 1, 2, \dots, k$$

En el presente proyecto la capa de salida tiene 3 neuronas que equivalen a las tres opciones posibles "ectomorfo": 0, "endomorfo":1, "mesomorfo":2. Esta vez, se emplea una función de activación softmax. Esta función de activación calcula la probabilidad de que la imagen analizada pertenezca a una de las clases previamente definidas y devuelve la clase con el valor de probabilidad más alto. Por ejemplo:

Imagen 43. Función de activación



Fuente: Autor

Para la primera prueba, se define la capa de entrada con el tamaño de las imágenes normalizadas y un solo canal. A continuación, se definen 3 capas ocultas con neuronas convolucionales con su respectiva capa de agrupación. Se aplanan las capas para pasarlas a una capa densa para, finalmente, pasar a la capa de salida.

La primera capa oculta consta de 16 filtros con un kernel de 3 x 3 y una función de activación "relu".

La segunda capa oculta consta de 32 filtros con un kernel de 3 x 3 y una función de activación "relu".

La tercera capa oculta consta de 64 filtros con un kernel de 3 x 3 y una función de activación “relu”.

La función Flatten aplana los datos, transforma a una sola dimensión.

La última capa oculta tiene un modelo de red denso con 512 neuronas y una función de activación “relu”.

La capa de salida tiene tres neuronas y una función de activación SOFTMAX.

Finalmente, se imprime el modelo con la función plot_model de utils de keras.

Imagen 44. Impresión del modelo

```

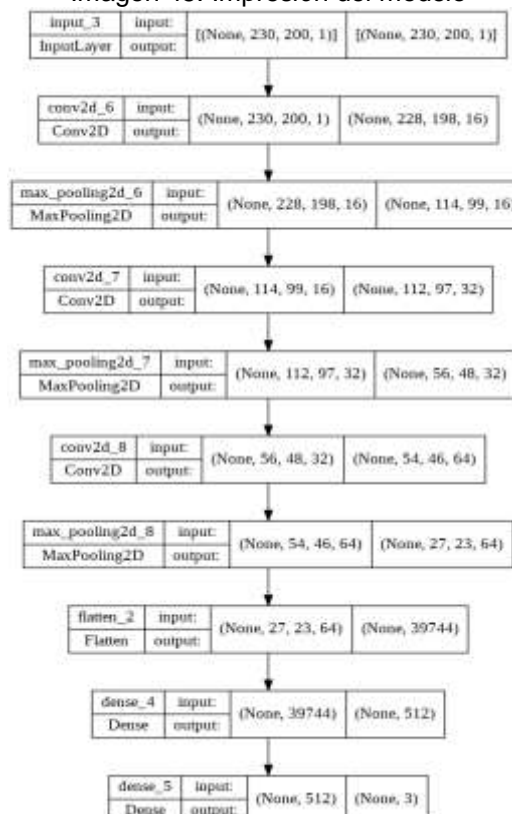
# Creamos el modelo de la red neuronal
model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 1)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplana los datos
    tf.keras.layers.Dense(512, activation = 'relu'), #capa oculta 4
    tf.keras.layers.Dense(3, activation = 'softmax') #capa de salida
])

tf.keras.utils.plot_model(model, show_shapes=True)

```

Fuente: elaboración propia

Imagen 45. Impresión del modelo



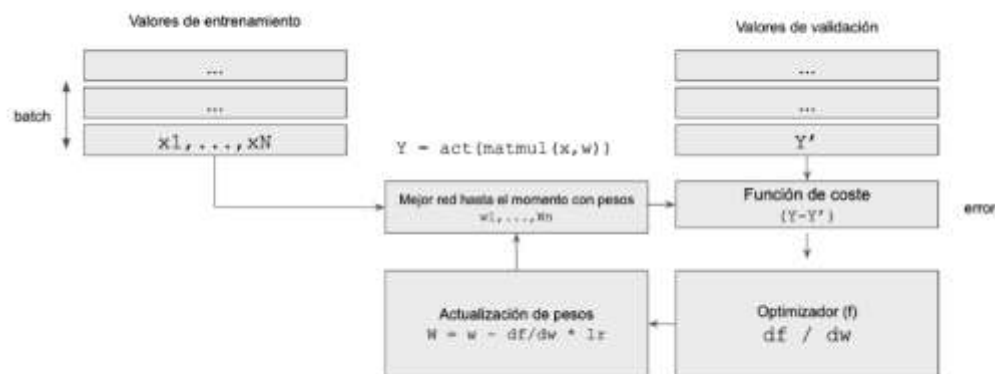
Fuente: elaboración propia

Compilación del modelo

Al compilar el modelo, se define el optimizador, la pérdida y las métricas del modelo.

El objetivo del entrenamiento de redes neuronales es minimizar la función de coste al encontrar pesos adecuados para la red, también, asegurar la generalización. El descubrimiento de estos pesos se lleva a cabo mediante un algoritmo numérico llamado backpropagation.

Imagen 46. Compilación del modelo



Fuente: elaboración propia

En este sentido el optimizador es el encargado de generar estos pesos y mejorarlos continuamente. Su funcionamiento, se basa en calcular el gradiente de la función de coste (derivada parcial) por cada peso (parametro/dimension) de la red. Se desea minimizar el error, por lo tanto, se modifica cada peso en la dirección (negativa) del gradiente:

Para agilizar la convergencia de la función de coste hacia su mínimo, se multiplica el vector de gradiente por un factor llamado factor de entrenamiento o aprendizaje.

La función de coste o pérdida (loss) estima la pérdida actual del modelo de modo que el optimizador actualice los pesos para reducir la pérdida en la siguiente evaluación. Para modelos de clasificación, se usa una Función de pérdida de clasificación multiclase.

Por otro lado, la métrica es una función, que se utiliza para calificar el rendimiento del modelo. Las funciones métricas son similares a las funciones de pérdida, excepto que los resultados de la evaluación de una métrica no se utilizan al entrenar el modelo. Los valores de las métricas son mostrados en pantalla durante el entrenamiento.

Keras tiene optimizadores que realizan todo este proceso automáticamente, el presente proyecto emplea el optimizador "ADAM" (Adaptive moment estimation). Así mismo, se emplea la función de pérdida "CategoricalCrossEntropy" y la métrica "accuracy" de keras.

Imagen 47. Compilador del modelo

▼ Compilar el modelo

```
model.compile(optimizer='Adam',
              loss='CategoricalCrossentropy',
              metrics=['accuracy'])
```

Fuente: elaboración propia

Después de compilar el modelo se importa tensorboard que permita visualizar los resultados del modelo de mejor manera para tomar decisiones.

Imagen 48. Importación del tensoboard

```
#llamamos a tensorboard
from tensorflow.keras.callbacks import TensorBoard
tensorboardface = TensorBoard(log_dir='logs/body')
```

Fuente: elaboración propia

Entrenamiento del modelo

Para entrenar o ajustar el modelo, se pasa los datos, que se emplearán para entrenar al modelo. En este caso, se llama al iterador creado previamente con ImageDataGenerator.

Se definen las épocas, es decir, cuántas veces se recorre todo el dataset y actualiza los pesos de las neuronas.

Se asignan los datos para la validación del modelo con los cuales, compara la precisión del modelo y finalmente, se realiza un callback a tensorflow para poder visualizar los datos del entrenamiento.

Imagen 49. Entrenamiento del modelo

Entrenamiento del modelo

```
[10] model_fit = model.fit(train_dataset,
                          epochs = 100,
                          validation_data = validation_dataset,
                          callbacks = [tensorboardface])
```

Epoch 1/100
 11/11 [=====] - 42s 3s/step - loss: 1.3461 - accuracy: 0.4748 - val_loss: 1.1250 - val_accuracy: 0.5294
 Epoch 2/100

Fuente: elaboración propia

Para visualizar los resultados, se carga tensorboard:

Imagen 50. Carga del tensorboard

```
%load_ext tensorboard
%tensorboard --logdir logs
```

Fuente: elaboración propia

Exportación del modelo

Se exporta el modelo en formato .h5, este archivo incluye:

- Los modelos de arquitectura
- Los valores de peso del modelo (que se aprendieron durante el entrenamiento)
- La configuración de entrenamiento del modelo (lo que pasó a 'compilar'), si corresponde
- El optimizador y su estado, si corresponde (esto le permite reiniciar el entrenamiento donde lo dejó)

Para guardar el modelo se emplea el comando save y le damos un nombre en formato .h5

Instalamos la librería de tensorflowjs para convertirlo a un formato, que se usa

en el navegador.

Finalmente descargamos el model.json y los archivos bin desde la sección de carpetas.

Imagen 51. Descarga del model.json

```
[ ] #Exportar el modelo en formato h5
    model.save('cuerpo.h5')

[ ] #El equipo es Linux. Listemos el contenido de la carpeta actual para ver que se exporto el modelo
    ls

    cuerpo.h5  drive  logs  model.png  sample_data

▶ #Para convertirlo a tensorflow.js, primero debemos instalar la libreria
    !pip install tensorflowjs

[ ] #Crear carpeta donde se colocaran los archivos resultantes
    !mkdir carpeta_modelo

[ ] #Realizar la exportacion a la carpeta de salida
    !tensorflowjs_converter --input_format keras cuerpo.h5 carpeta_modelo

[ ] #Confirmar que en la carpeta de salida se hayan generado los archivos. Deben aparecer archivos "bin" y "json"
    ls carpeta_modelo

    group1-shard1of2.bin  group1-shard2of2.bin  model.json
```

Fuente: elaboración propia

CAPÍTULO III. ANÁLISIS DE LOS RESULTADOS DE LA INVESTIGACIÓN

Fase 4 y 5: Evaluación del prototipo y Refinamiento del Prototipo

En tensorboard, se analiza los resultados obtenidos tanto con los datos de entrenamiento (naranja) como con los datos de validación(azul). En las gráficas, se muestran la precisión del modelo por cada época, así como su valor de pérdida.

En la primera instancia, se crea el modelo con cuatro capas ocultas, tres de ellas son convolucionales y una capa densa. Tanto la capa de entrada como la de salida se mantiene durante todo el proceso de refinamiento.

Imagen 52. Modelo cuatro capas ocultas

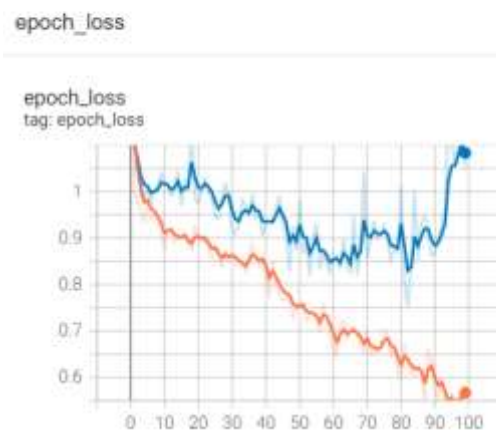
```
#creamos el modelo de la red neuronal
model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width,3)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'),#capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dense(512, activation = 'relu'), #capa oculta 4
    tf.keras.layers.Dense(3, activation = 'softmax') #capa de salida
])

tf.keras.utils.plot_model(model, show_shapes=True)
```

Fuente: elaboración propia

Como se observa en la siguiente imagen los valores de precisión obtenidos no son los mejores. Adicionalmente, a partir de la época 90 el valor de pérdida de los datos de validación sube drásticamente, mientras que el valor de pérdida de los datos de entrenamiento continuó en decrecimiento. Esto indica que existe un overfitting. Por lo, que se procede a aplicar una técnica llamada “dropout” para tratar de reducir el overfitting. Esta técnica consiste en apagar ciertas neuronas para evitar que “memoricen” ciertos pesos y se las obliga a continuar en aprendizaje (actualizar sus pesos) con cada nueva iteración.

Imagen 53. Técnica dropout



Fuente: elaboración propia

Para contrarrestar el overfitting analizada en la imagen anterior se añade un dropout de 0.5 a la capa de salida mientras que la estructura se mantiene del modelo anterior.

Imagen 54. Overfitting

```
#Creamos el modelo de la red neuronal
model = tf.keras.models.Sequential([
    tf.keras.layers.Input(shape=(img_height, img_width, 3)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplana los datos
    tf.keras.layers.Dense(512, activation = 'relu'), #capa oculta 4
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(1, activation = 'softmax') #capa de salida
])
tf.keras.utils.plot_model(model, show_shapes=True)
```

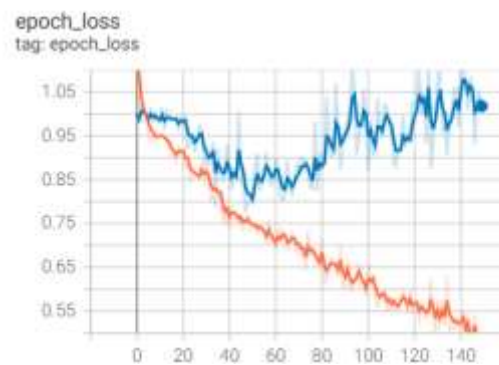
Fuente: elaboración propia

Los resultados muestran que el dropout aplicado no tuvo un efecto favorable sobre la red. Por lo tanto, se toma otro tipo de corrección.

Imagen 55. Resultados del dropout



epoch_loss



Fuente: elaboración propia

A continuación, se añade una nueva capa oculta de tipo convolucional, con 128 filtros lo que le da una mayor profundidad al modelo. El dropout aplicado en la prueba anterior se mantiene.

Imagen 56. Desarrollo modelo

```

model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 1)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(512, activation = 'relu'), #capa oculta 4
    tf.keras.layers.Dense(3, activation = 'softmax') #capa de salida
])
tf.keras.utils.plot_model(model, show_shapes=True)

```

Fuente: elaboración propia

Al incrementar una capa a la red neuronal se evidencia que la precisión del modelo mejora. Sin embargo, la pérdida se mantiene elevada con una clara indicación que existe overfitting.

Imagen 57. Resultados del nuevo modelo



Fuente: elaboración propia

Para disminuir la pérdida, se aplica una nueva capa oculta de tipo convolucional al modelo con 128 filtros.

Imagen 58. Integración de capa oculta convolucional

```

model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 3)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 5
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(512, activation = 'relu'), #capa oculta 6
    tf.keras.layers.Dense(1, activation = 'softmax') #capa de salida
])

```

Fuente: elaboración propia

Los resultados obtenidos muestran una pequeña mejora. Sin embargo, la pérdida continúa elevada, y se mantiene el problema del overfitting.

Imagen 59. Análisis del modelo



Fuente: elaboración propia

Para mejorar el modelo, se incrementan los filtros o la profundidad de la capa oculta añadida previamente de 128 a 256.

Imagen 60. Mejora del modelo

```

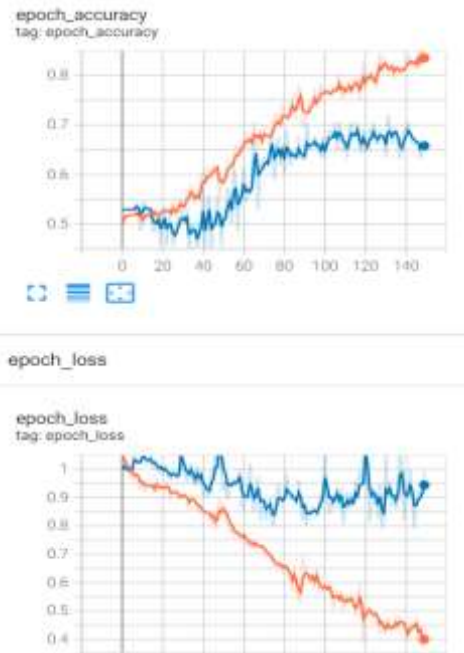
model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 1)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(256, (3,3), activation = 'relu'), #capa oculta 5
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(512, activation = 'relu'), #capa oculta 6
    tf.keras.layers.Dense(10, activation = 'softmax') #capa de salida
])

```

Fuente: elaboración propia

Al incrementar el número de filtros o profundidad del modelo, este no mejoró, por el contrario, empeoró. Por lo tanto, una mayor profundidad en el modelo no necesariamente va a mejorar el performance del modelo. Sino incrementa el overfitting.

Imagen 61. Análisis del modelo



Fuente: elaboración propia

A continuación, se reduce el tamaño de la última capa y se añade otro dropout para manejar el problema del overfitting.

Imagen 62. Añadidura de dropout

```

model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 1)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(512, activation = 'relu'), #capa oculta 4
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(3, activation = 'softmax') #capa de salida
])

```

Fuente: elaboración propia

Con los ajustes realizados en el modelo anterior se reduce la pérdida, sin embargo, aún es alta y todavía existe el problema del overfitting.

Imagen 63. Análisis del modelo



Fuente: elaboración propia

En general para combatir el overfitting, se recomienda reducir la profundidad del modelo por lo tanto se reducen las neuronas de la última capa de entrada a la mitad.

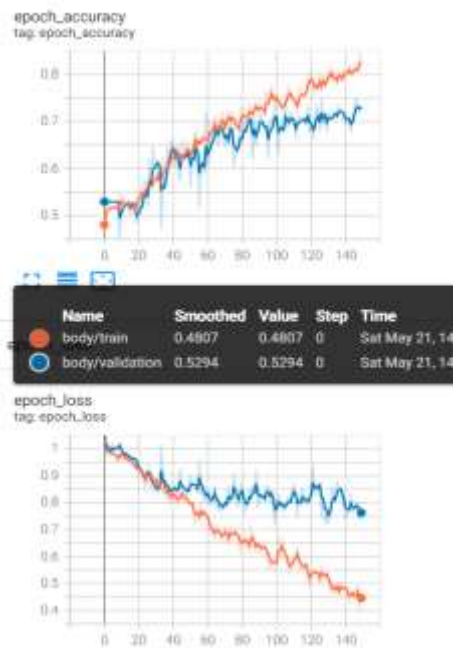
Imagen 64. Ingreso de nuevas capas

```
//model = tf.keras.models.Sequential([
    tf.keras.layers.Input(shape=(img_height, img_width, 3)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (7,7), activation = 'relu'), #capa oculta 5
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #ajustar los datos
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(256, activation = 'relu'), #capa oculta 6
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(1, activation = 'softmax') #capa de salida
])
```

Fuente: elaboración propia

Al reducir las neuronas de la última capa no se mejora de una manera sustancial el modelo y continua el problema del overfitting debido a que con cada nueva interacción las perdidas con los valores de entrenamiento disminuyen, sin embargo, la perdida con los datos de validación se mantiene elevada.

Imagen 65. Resultados del modelo



Fuente: elaboración propia

Para mejorar el modelo, se aplica la técnica de dropout de manera más agresiva incremento el primer dropout de 0.5 a 0.8.

Imagen 66. Mejorar el dropout

```

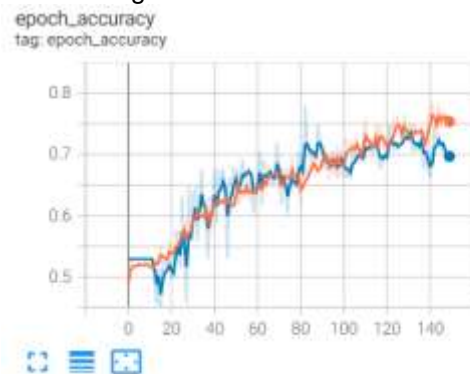
model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 1)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3, 3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(64, (3, 3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(128, (3, 3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(128, (3, 3), activation = 'relu'), #capa oculta 5
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dropout(0.8), #dropout
    tf.keras.layers.Dense(256, activation = 'relu'), #capa oculta 4
    tf.keras.layers.Dropout(0.3), #dropout
    tf.keras.layers.Dense(1, activation = 'softmax') #capa de salida
])

```

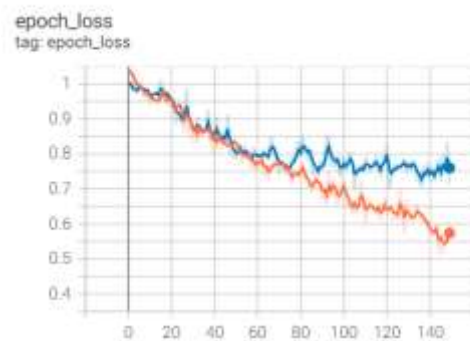
Fuente: elaboración propia

Al elevar el dropout, se mejora sustancialmente el problema del overfitting. Sin embargo, la pérdida aún es elevada.

Imagen 67. Resultados del modelo



epoch_loss



Fuente: elaboración propia

Se reduce nuevamente a la mitad el número de neuronas de la última capa oculta para mejorar la precisión del modelo.

Imagen 68. Análisis del modelo

```

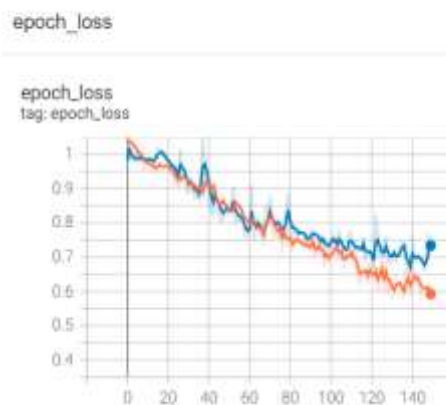
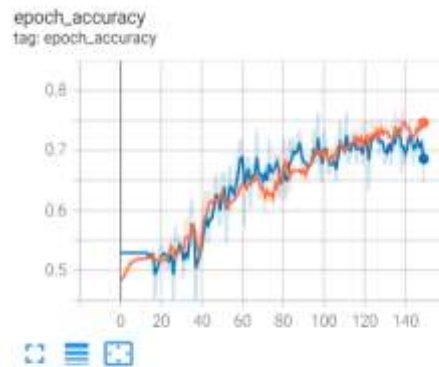
model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 1)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 5
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(128, activation = 'relu'), #capa oculta 6
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(3, activation = 'softmax') #capa de salida
])

```

Fuente: elaboración propia

Al reducir el número de neuronas de la última capa se reduce considerablemente el overfitting y la precisión alcanza un rango aceptable.

Imagen 69. Resultados del modelo



Fuente: elaboración propia

Para intentar reducir aún más el overfitting se procede a reducir nuevamente el número de neuronas de la última capa oculta a la mitad de 128 a 64.

Imagen 70. Análisis del modelo

```

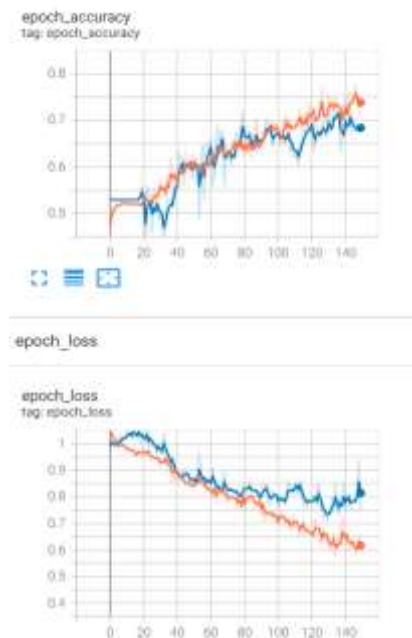
model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 3)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 5
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplazar los datos
    tf.keras.layers.Dropout(0.5), #dropout1
    tf.keras.layers.Dense(64, activation = 'relu'), #capa oculta 6
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(3, activation = 'softmax') #capa de salida
])

```

Fuente: elaboración propia

Se se observa que la reducción del tamaño de la última capa no tuvo un resultado favorable en el modelo.

Imagen 71. Resultados del modelo



Fuente: elaboración propia

Se procede a incrementar el número de neuronas de la última capa oculta a 512.

Imagen 72. Análisis del modelo

```

model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 3)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 5
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dropout(0.5), #dropout
    tf.keras.layers.Dense(512, activation = 'relu'), #capa oculta 6
    tf.keras.layers.Dropout(0.3), #dropout
    tf.keras.layers.Dense(3, activation = 'softmax') #capa de salida
])

```

Fuente: elaboración propia

Al incrementar las neuronas de la última capa oculta no se obtiene ninguna mejora. Por lo que, se concluye que el tamaño ideal para la última capa oculta es de 128 neuronas.

Imagen 73. Resultados del modelo



Fuente: elaboración propia

Para mejorar el modelo, se incrementa el número de capas ocultas al añadir una capa convolucional de 128 filtros.

Imagen 74. Análisis del modelo

```

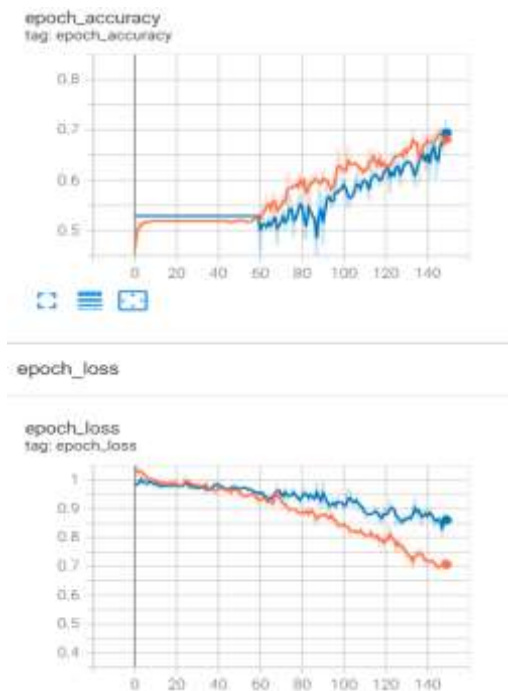
model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(img_height, img_width, 3)), #capa de entrada
    tf.keras.layers.Conv2D(16, (3,3), activation = 'relu'), #capa oculta 1
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(32, (3,3), activation = 'relu'), #capa oculta 2
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(64, (3,3), activation = 'relu'), #capa oculta 3
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 4
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 5
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Conv2D(128, (3,3), activation = 'relu'), #capa oculta 6
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(), #aplanar los datos
    tf.keras.layers.Dropout(0.8), #dropout
    tf.keras.layers.Dense(256, activation = 'relu'), #capa oculta 7
    tf.keras.layers.Dropout(0.3), #dropout
    tf.keras.layers.Dense(3, activation = 'softmax') #capa de salida
])

```

Fuente: elaboración propia

Como, se observa en la gráfica. Al incrementar el número de capas ocultas el modelo no mejora, sino que regresa el problema del overfitting. Por lo tanto, se emplea el modelo que obtuvo los mejores resultados. Se lo exporta y se carga el modelo en el navegador.

Imagen 75. Resultados del modelo



Fuente: elaboración propia

3.1. Fase 6: Producto Final

El sistema, se realizó por medio de un sitio web HTML, como se muestra en la construcción del prototipo, además, de que cuenta con estilos con ccs y el empleo de java script, estas herramientas permiten que el software arroje un resultado al usuario, muestra el tipo de cuerpo que posee a fin de recomendar una alimentación acorde a sus necesidades o deseos.

En primera instancia, se accede al link, que se describe a continuación:

<http://127.0.0.1:5500/index.html>

Ventana de inicio

Una vez en la página, se despliega la primera ventana de inicio, que se refleja a continuación, se eligió la primera pestaña como Home.

Dentro del Home, se encuentra un botón para Empezar, el cual, lleva a la pestaña de Descubrir tu tipo de cuerpo.

Imagen 76. Ventana de inicio del sitio web



Fuente: elaboración propia

Descubrir tu tipo de Cuerpo

Esta ventana, es donde se encuentra cargado el modelo entrenado que permite identificar el tipo de cuerpo, para lo cual se accede a la cámara web del usuario y se entrega un resultado en tiempo real.

Imagen 77. Activación de la cámara



Fuente: elaboración propia

Calcula tu dieta

En esta ventana, se divide en dos secciones:

Una donde se ingresa los datos del usuario y otra donde se muestra los resultados.

En la sección del ingreso de datos se inicia con la elección del tipo de cuerpo, si el usuario no ha identificado su tipo de cuerpo, existe un hipervínculo, que redirecciona a la ventana de (Descubre tu tipo de cuerpo).

A continuación, se requiere datos de medidas del usuario, como: altura, cuello, cintura, peso. Junto a su nivel de actividad física, adicionalmente se le pregunta la meta del usuario. Una vez, que se tiene estos datos, el programa calcula la cantidad de calorías necesaria, así como su distribución en macronutrientes para entregar una guía nutricional personalizada.

Imagen 78. Datos del usuario

Ingresa tus datos

Tipo de cuerpo

Si no sabes cuál es tu tipo de cuerpo, [descubre cuál](#)

Selecciona uno:

Ectomorfo Mesomorfo Endomorfo

Algunas medidas

Altura:

Cintura:

Cintura:

Peso:

Actividad

Indica tu nivel de Actividad:

Baja Media Alta

Elige tu meta

Tu meta es: Aumentar masa muscular Perder porcentaje de grasa

Calcular

Desarrollado by Alejandro Martínez Todos los derechos reservados

Fuente: elaboración propia

CONCLUSIONES

- La elaboración de un estado de arte sobre inteligencia artificial y procesamiento de imágenes para la industria fitness, permite concluir que las computadoras procesan las imágenes de diferente manera que el ojo humano, donde el ojo humano ve patrones como líneas, la computadora procesa píxeles, cada uno de estos píxeles se conforman de valores numéricos que indican un nivel de color, para diferenciar los píxeles en una imagen y encontrar patrones es necesario compararlos con los píxeles de su alrededor, este proceso se lo conoce como convolución. En las convoluciones se usa núcleos o kernels para comparar varios píxeles alrededor de un pixel principal. Para el presente proyecto se usó un núcleo de 3x3 con los cuales se pudo diferenciar líneas o ejes en las imágenes. A continuación, con una capa de agrupación se determinan patrones, se determinan patrones a través de un Maxpooling de 2 dimensiones de 2x2.
- Para el desarrollo de un sistema de procesamiento de imágenes corporales en Python; el primer paso, es la preparación del data set el cual, consta de varias imágenes clasificadas manualmente de acuerdo con las categorías requeridas, en este caso: endomorfo, ectomorfo y mesomorfo. En segundo lugar, se realiza una técnica conocida como aumento de datos, la cual le da más robustez al dataset, la misma que un iterador a partir de las imágenes previamente establecidas. Este proyecto emplea el ImageDataGenerator para realizar el aumento de imágenes a través de la rotación, desplazamientos horizontales y los desplazamientos verticales. Se crea el modelo de red neuronal con capas ocultas de neuronas convolucionales que tienen un núcleo o kernel y una función de activación. A continuación, se generan capas de agrupación que reconocen patrones dentro de las imágenes, cada nueva capa oculta permite reconocer nuevos patrones, sin embargo, aumenta el costo computacional por lo que es necesario encontrar un equilibrio. Para completar la creación del modelo se aplica la técnica de dropout que consiste en apagar neuronas en cada interacción para evitar el

overfitting o el sobre entrenamiento. Una vez creado el modelo se lo compila y entrena. En la compilación, se define el optimizador junto a la función de pérdida. En el entrenamiento, se determinan las épocas durante las cuales se va a entrenar el modelo, en este caso se utilizaron 150 épocas. Finalmente, se exporta el modelo para poder ser utilizado en las aplicaciones requeridas.

- La entrega de una guía básica de nutrición en base a los datos recogidos del sistema, se determina con un sistema de procesamiento de imágenes con inteligencia artificial que brinden una guía nutricional se crea una aplicación web con HTML y JavaScript que permite usar el modelo previamente entrenado en TensorFlow, esta aplicación web toma la imagen a través de la cámara de un dispositivo para procesarla en tiempo real y clasificarla de acuerdo con el tipo de cuerpo. Junto a los datos requeridos como, peso, altura, medidas corporales y edad, calcula las calorías necesarias y la distribución de macronutrientes de acuerdo con el objetivo del usuario. Finalmente, se genera una dieta mediante los resultados obtenidos y la guía entregada.

RECOMENDACIONES

- En el procesamiento de imágenes, al trabajar con las redes neuronales densas no generan un resultado óptimo porque da un mayor overfitting, es decir, las neuronas aprenden las posiciones de los píxeles mas no los patrones, es por eso recomendable realizar el procesamiento de imágenes con redes neuronales convolucionales.
- Para futuras investigaciones con el objetivo de mejorar el desempeño del sistema se recomienda poner a prueba el modelo con un mayor número de épocas de entrenamiento y un dropout mayor para evitar el overfitting.
- Para el entrenamiento de la red neuronal, se inicia con pocas neuronas e ir incrementa de a poco, esto debido a que, se ingresa muchas de ellas en un solo instante, el performance no genera los resultados deseados, por lo cual dejo en evidencia que es falsa la hipótesis que a mayor redes neuronales mejores resultados se obtendrán.
- En el caso del overfitting de la investigación al contar con una base de datos muy limitada fue necesario la aplicación métodos como el ImageDataGenerator, lo que permite tener mejor data set y que mejore notablemente el modelo, a partir de estas se generaron los drop sets en este caso fueron dos, donde el 80% de las neuronas se apagaran para dar un mejor modelo.
- Se mejora la aplicación web si se agrega una mayor especificidad a la dieta arrojada por el sistema, utiliza la programación orientada a objetos, al crear una clase por cada macronutriente y genera las dietas como objetos en la cual el usuario selecciona el tipo de alimento que prefiere para su dieta.

BIBLIOGRAFÍA

- Alemán, D. (2017). *Técnicas de inteligencia artificial aplicadas a problemas de ingeniería civil*. *Revista de Arquitectura e Ingeniería*, 11(3), 1–7.
<https://www.redalyc.org/articulo.oa?id=193955164005>
- Araica Zepeda, R., & Hernández, M. M. V. (2020). *Hacia la definición conceptual del constructo Calidad de la Educación Superior en el contexto de la Universidad Nacional Agraria*. *Revista Educación*, 44.
<https://doi.org/10.15517/revedu.v44i2.40522>
- Arbey, W. (2018). *Prototype of Monitoring and Alarm for the Detection of White Frosts in Rural Sectors of Sogamoso*. 18(1), 57–64.
- Arora, S., & Kaur, R. (2022). *An escalated convergent firefly algorithm*. *Journal of King Saud University - Computer and Information Sciences*, 34(2), 308–315. <https://doi.org/10.1016/j.jksuci.2018.10.007>
- Arselan, M. (2016). *Algoritmos de procesamiento y reconstrucción tomográfica para imágenes radiológicas*. 63.
- Aumentada, R., Extranjera, L., Aumentada, R., Burs, F., & Neuronales, R. (2020). *COMPUTACIÓN*. 18.
- Azevedo, D. (n.d.). *PYDATA__Escaneo_corporal_3D_de_bajo_costo_para_monitoreo_y_seguimiento_remos_de_sobrepeso.pdf*. 10–13.

- Blas Yoel, J. G., & Rodríguez Hernández, C. (2021). *Educación física en tiempos de Covid-19. Valoraciones a partir de la utilización de las TIC. Revista Conrado*, 17(79), 32–40. <https://orcid.org/0000-0002-1665-4485>
- Buzai, G. D., & Montes Galbán, E. (2020). *Megaciudad Buenos Aires: Cartografía de su última expansión y conurbación mediante el procesamiento digital de imágenes satelitales nocturnas. Revista Cartográfica*, 100, 215–238. <https://doi.org/10.35424/rcarto.v0i100.667>
- Capraro, F., Serman, F. V., & Tosetti, S. (n.d.). *Determinación de CWSI en olivares mediante procesamiento de imágenes termográficas. 125–138.*
- Careta, E. P., Córdova-Fraga, T., Pérez, M. G., Delgado, J. H., Solís, C. G., & Cabrera, R. G. (2019). *Digital thermal image processing to identify bone propagation characteristics. Computacion y Sistemas*, 23(4), 1469–1473. <https://doi.org/10.13053/CyS-23-4-3214>
- de la Cámara, M., Valcarce-Torrente, M., & Veiga, O. L. (2019). *National survey of fitness trends in Spain for 2020. Retos*, 40(9), 434–441. <https://doi.org/10.47197/RETOS.V37I37.74179>
- Diáz, J. E. M. (2020). *Deep Artificial Vision Applied to the Early Identification of Non-Melanoma Cancer and Actinic Keratosis. Computacion y Sistemas*, 24(2), 751–766. <https://doi.org/10.13053/CyS-24-2-2901>

Dr. Vladimir, V. F. (1967). 濟無No Title No Title No Title. *Gastronomía Ecuatoriana y Turismo Local.*, 1(69), 5–24.

Enfoque, U. N., El, H., & Realpe-muñoz, P. (2020). *Tendencias tecnológicas basadas en fpga* : 17, 3–14.

Espinal-Correa, C. E., & Estrada-Mesa, D. A. (2020). *El fitness: estilo de vida saludable o biopolítica sobre el cuerpo desde la racionalidad neoliberal. Iatreia*, 33(2), 184–190. <https://doi.org/10.17533/udea.iatreia.48>

Estany, E. R., & Vera, N. C. (2020). *Cardiología*. 27(4), 344–350.

Estrada-Marcén, N. C., Sanz-Gonzalo, G., Simón-Grima, J., Casterad-Seral, J., & Roso-Moliner, A. (2019). *The fitness centres of the city of Zaragoza. Apunts. Educacion Fisica y Deportes*, 135, 118–136. [https://doi.org/10.5672/APUNTS.2014-0983.CAT.\(2019/1\).135.08](https://doi.org/10.5672/APUNTS.2014-0983.CAT.(2019/1).135.08)

Flores Echaiz, L., Neupane, B., Rivera Lam, M., Sibal, P., & Hu, X. (2021). *El aporte de la inteligencia artificial y las TIC avanzadas a las sociedades del conocimiento: una perspectiva de derechos, apertura, acceso y múltiples actores. In UNESCO Publishing.* <https://unesdoc.unesco.org/ark:/48223/pf0000375796?1=null&queryId=221d02da-a91f-4ce4-ab1c-44799cf472d1>

García-Estrada, H., Linares-Escobar, O.-A., Pastrana-Manzanero, A., Martínez-

- Ruiz, L.-C., Ramírez-Sotelo, M.-G., & Cabrera-Llanos, A.-I. (2018). *Sistema de clasificación de deformaciones pédicas por procesamiento digital de imágenes y lógica difusa en LabVIEW. Research in Computing Science, 147(12), 31–39.* <https://doi.org/10.13053/racs-147-12-3>
- García, F. J., Moreno, M. N., García, A., & Vázquez, A. (2019). *Ingeniería de Software I. 23.* [https://repositorio.grial.eu/bitstream/grial/1940/1/IS_I_Tema_3 - Modelos de Proceso.pdf](https://repositorio.grial.eu/bitstream/grial/1940/1/IS_I_Tema_3_-_Modelos_de_Proceso.pdf)
- García Monsálvez, J. C. (2017). *Python como primer lenguaje de programación textual en la Enseñanza Secundaria. Education in the Knowledge Society(EKS),18(2),147–62.* <https://doi.org/10.14201/eks2017182147162>
- Gómez Chávez, L. F. J., López-Haro, J., Pelayo Zavalza, A. R., & Aguirre Rodríguez, L. E. (2022). *Encuesta Nacional de Tendencias Fitness para México en 2022 (National Survey of Fitness Trends in Mexico for 2022).* *Retos, 44, 1053–1062.* <https://doi.org/10.47197/retos.v44i0.92028>
- Guadalupe Sánchez-Méndez, A., & Pedro Arguijo-Hernández, S. (2018). *Análisis de imágenes multiespectrales para la detección de cultivos y detección de plagas y enfermedades en la producción de café Analysis of Multispectral Images for the Detection of Crops and Detection of Pests and Diseases in Coffee Production. Research in Computing Science, 147(7), 309–317.*

- Herrera Torres, L., Pérez Tur, F., Valcarce Torrente, M., & García Fernández, J. (2019). *Análisis de las redes sociales en el sector del fitness español: un estudio longitudinal. Materiales Para La Historia Del Deporte*, 18, 122–128.
- Intriago-Plaza, R., Palma-Zambrano, W., & Mielles-Bravo, Y. (2021). *Diseño y evaluación de un prototipo para vivienda de interés social con mampostería confinada en ladrillo. Revista Científica INGENIAR: Ingeniería, Tecnología e Investigación*, 4(7), 17–30. <https://doi.org/10.46296/ig.v4i7.0019>
- Jerónimo, J., & Sossa, H. (2018). *Reconocimiento de objetos cuasi-planos mediante un sistema de tratamiento digital de imágenes embebido en una plataforma tipo Raspberry Pi. Research in Computing Science*, 147(4), 159–169. <https://doi.org/10.13053/rcs-147-4-13>
- Jima, S., Salvatierra, R., & Terán, I. (2019). *Diseño Y Construcción De Un Prototipo Recolector De Residuos Marinos Para El Muelle De La Ciudad De Manta. Revista de Ciencias Del Mar y Acuicultura "YAKU,"* 2, 10–19.
- Kanani, P., & Padole, M. (2019). *Deep learning to detect skin cancer using google colab. International Journal of Engineering and Advanced Technology*, 8(6), 2176–2183. <https://doi.org/10.35940/ijeat.F8587.088619>
- Lira-Baca, T., Borboya Melchor, M., Hernández Sánchez, R., Tinoco Varela, D.,

- & Gudiño Peñaloza, F. (2019). *Mapeo de incidencia criminal en el Estado de México mediante procesamiento digital de imágenes. Research in Computing Science, 148(8), 173–185. <https://doi.org/10.13053/rcs-148-8-13>*
- Martínez-Castilla, J. M., Zapata Córdoba, D., Dávalos Dávalos, A. A., & Manzur Ayala, R. (2021). *Cuantificación digital de integración de injertos de piel de espesor parcial ocluidos con sistema de presión negativa. Cirugía Plástica Ibero-Latinoamericana, 47(1), 95–104. <https://doi.org/10.4321/s0376-7892.2021.0001.00013>*
- Mendoza, L., & Id, O. R. C. (n.d.). *Simulación. 0–2.*
- Mercedes, K., Theran, D., Carazo, R. M., & Marín, J. G. (2021). *Impacto de programa de ejercicio físico combinado en marcadores bioquímicos asociados a sobrepeso y obesidad Impact of a combined physical exercise program on biochemical markers associated with overweight and obesity. Revista Cuatrimestral “Conecta Libertad,” 5, 13–23.*
- Mitaritonna, A. D. (2018). *Tecnologías emergentes en la educación: la realidad aumentada. Revistas.Ub.Edu.Ar, 1, 87. <https://revistas.u b.edu.ar/index.php/Perspectivas/article/view/38>*
- Origel-rivas, C. G., & Rendón-lara, E. (2020). *Redes neuronales artificiales y árboles de decisión para la clasificación con datos categóricos Artificial*

Neural Networks and Decision Trees for. 149(8), 541–554.
[https://rcs.cic.ipn.mx/2020_149_8/Redes neuronales artificiales y arboles de decision para la clasificacion con datos categoricos.pdf](https://rcs.cic.ipn.mx/2020_149_8/Redes%20neuronales%20artificiales%20y%20arboles%20de%20decision%20para%20la%20clasificacion%20con%20datos%20categoricos.pdf)

Panasenko, S. V., Nikishin, A. F., Mayorova, E. A., Boris, O. A., & Murtuzalieva, T. V. (2018). *Innovative approach to fitness industry development.* *Espacios*, 39(41).

Pineda Ballesteros, E., Tellez Acuña, F. R., & Medina Cruz, J. (2020). *Software requirements: prototyping, legacy software, and document analysis.* *Ingeniería y Desarrollo*, 37(2), 327–345. <https://doi.org/10.14482/ingen.de.37.2.1053>

Rosa-Guillamón, A. (2019). *Análisis de la relación entre salud, ejercicio físico y condición física en escolares y adolescentes.* *Revista Ciencias de La Actividad Física*, 20(1), 1–15. <https://doi.org/10.29035/rcaf.20.1.1>

Scarinci, I. E., Pérez, P., & Valente, M. (2020). *Heuristic algorithm for pet images' segmentation using artificial intelligence techniques.* *Anales de La Asociacion Fisica Argentina*, 31(4), 165–171. <https://doi.org/10.31527/analesafa.2020.31.4.165>

Serra Valdés, M. Á. (2020). *Sugerencias de ejercicio físico en casa para adultos durante confinamiento por virus SARS-CoV-2.* *Revista Habanera De Ciencias Medicas*, 6(5), 1–15. <https://scielo.sld.cu/scielo.php?>

id=S1729-519X2008000300003&script=sci_arttext

Suárez Mejías, C. (2021). *Segmentación de tejidos con contornos difusos en imágenes radiológicas*. 1–20.

Tancara Q, C. (1993). *La Investigacion Documental*. *Temas Sociales*, 17, 91–106. http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S0040-29151993000100008&lng=es&nrm=iso&tlng=es

Tarazona, B., & Sandoval, C. (2019). *Evaluación de discontinuidades tipo grietas y fisuras en estructuras de hormigón empleando un analizador de vibraciones y procesamiento digital de imágenes*. *Entre Ciencia e Ingeniería*, 13(25), 85. <https://doi.org/10.31908/19098367.4018>

Torres-Criollo, L. M., & Ochoa, R. A. (2021). *Efectividad de la intervención basada en el ejercicio físico en pacientes con enfermedad de Alzheimer*. *Revista Ecuatoriana de Ciencia, Tecnología e Innovación En Salud Pública.*, 5(3), 13. <https://www.researchgate.net/publication/355177025>

Valero Carvajal, D. A., Orozco Isaza, J. E., Córdoba Puerto, J. N., Noreña Hemelberg, O. S., Gómez Alarcón, M. A., Jinete Gómez, M. A., & Gutiérrez Alfonso, J. A. (2021). *Algoritmos para el procesamiento de imágenes implementados en el Robot Humanoide InMoov*. *Revista EIA*, 18(36), 1–6. <https://doi.org/10.24050/reia.v18i36.1495>

- Verónica, I., Organista, D., Antonio, J., Valverde, M., Martínez, D. M., & Cadena, M. T. I. E. (2021). *Extracción y selección de características en imágenes digitales mostrando lesiones de piel*. 13, 91–104.
- Vidal-Silva, C. L., Sánchez-Ortiz, A., Serrano, J., & Rubio, J. M. (2021). *Academic experience in rapid development of web information systems with Python and Django | Experiencia académica en desarrollo rápido de sistemas de información web con Python y Django*. *Formacion Universitaria*, 14(5), 85–94.
- Yoelkis, V. (2021). *Algoritmos para el procesamiento de imágenes con artefactos de endurecimiento de haz en tomografía computarizada*. *Revista Cubana de Ciencias Informáticas*, 15(2), 96–117.
- Zamorano, S., & García, J. (2018). *the Importance-Performance Analysis According To Gender and*. *Materiales Para La Historia Del Deporte*, 16(2340–7166), 24–35.

ANEXOS

Anexo N1. Código Fuente

Ventana de inicio

Calculo:

El css de esta ventana es:

```
<!
es">
<head> DOCTYPE html>
<html lang="
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AI Fitness</title>
  <link rel="stylesheet" href="css/main.css">
</head>
<header>
  <nav>
    <ul class="nav-left-section">
      <li>
        <a href="">Home</a>
      </li>
      <li>
        <a href="cuerpo.html">Descubre tu tipo de cuerpo</a>
      </li>
      <li>
        <a href="dieta.html">Calcula tu dieta</a>
      </li>
      <li>
        <a href="about.html">Acerca de</a>
      </li>
    </ul>
  </nav>
</header>
```

```

<main>
  <section class="main-logo">
    

  </section>

  <section class="main-description">
    <p>Descubre tu tipo de cuerpo, calcula tu dieta y ponte en forma.</p>
    <p id="pia">Todo esto con la ayuda de la inteligencia artificial...!!!</p>
  </section>

  <section class="main-button">
    <div>
      <a href="cuerpo.html">
        <button>
          Empezar
        </button>
      </a>
    </div>
  </section>
</main>

<footer>
  <p class="footer-left">Developed by Alejandro Martinez</p>
  <p class="footer-right">Todos los derechos reservados</p>

</footer>
<body>

</body>
</html>

```

Anexo N3:

Funcionamiento de la cámara

'use strict'; // Para hacer que el código sea más seguro.

// Definimos las constantes que vamos a utilizar

```
const videoFrame = document.getElementById('video_frame');
const canvasFrame = document.getElementById('canvas_frame');
const snapFrame = document.getElementById("snap_frame");
const errorMsgElement = document.querySelector('span#errorMsg');
```

// Definimos tamaño del video y si queremos audio o no

```
const constraints = {
  audio: false,
  video: {
    width: 480, height: 720
  }
};
```

// Comprobamos acceso a la Webcam

```
async function init() {
  try {
    const stream = await navigator.mediaDevices.getUserMedia(constraints);
    handleSuccess(stream);
  } catch (e) {
    errorMsgElement.innerHTML = `navigator.getUserMedia
error:${e.toString()}`;
  }
}
```

// En caso de que el acceso sea correcto, cargamos la webcam

```
async function handleSuccess(stream) {
  window.stream = stream;
  videoFrame.srcObject = stream;
}
```

```
// Iniciamos JS
init();

// Hacemos captura de pantalla al hacer click
var context = canvasFrame.getContext('2d');
snapFrame.addEventListener("click", function() {
    context.drawImage(videoFrame, 0, 0, 140, 320);
});
```

Anexo N4:

Ventana de altura, medidas del cuello, cintura y peso.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AIFIT Dieta</title>
    <link rel="stylesheet" href="css/main.css">
</head>
<body>
    <header>
        <nav>
            <ul class="nav-left-section">
                <li>
                    <a href="index.html">Home</a>

                </li>
                <li>
                    <a href="cuerpo.html">Descubre tu tipo de cuerpo</a>
                </li>
                <li>
                    <a href="dieta.html">Calcula tu dieta</a>
```

```

    </li>
    <li>
        <a href="about.html">Acerca de</a>
    </li>

</ul>
</nav>
</header>
<main>

<h1>Ingresa tus datos</h1>

<section class="section-form">

    <form action="main.js" method="get">
        <section class="section-tipocuerpo">
            <h2>Tipo de cuerpo</h2>
            <p>Si no sabes cual es tu tipo de cuerpo, descubrelo <a
href="cuerpo.html">aqui</a>. </p>
            <p>Selecciona uno:</p>
            <p>
                <input type="radio" name="tipo" value="ec" > Ectomorfo
                <input type="radio" name="tipo" value="me" > Mesomorfo
                <input type="radio" name="tipo" value="en" > Endomorfo
            </p>
        </section>

        <section class="section-medidas">
            <h2>Algunas medidas</h2>
            <p>Altura(cm)
                <input id="input_altura" type="number" name="altura"
step="any" min=100 max=220>
            </p>

```

```

    <p>Cuello(cm)
      <input id="input_cuello" type="number" name="cuello"
step="any" min="0" max="80">
    </p>
    <p>Cintura(cm)
      <input id="input_cintura" type="number" name="cintura"
step="any" min="0" max="200">
    </p>
    <p>Peso(Kg)
      <input id="input_peso" type="number" name="peso" step="any"
min="0" max="200">
    </p>
  </section>

```

```

<section class="section-actividad">
  <h1>Actividad</h1>
  <p>Indica tu nivel de Actividad</p>
  <p>
    <input type="range" name="actividad" id="input_nivel_actividad"
min="1.2" max="1.9" step="0.175">
  </p>
  <ul class="nivel-actividad-label">
    <li id="label-bajo">Bajo</li>
    <li id="label-medio">Medio</li></div>
    <li id="label-alto">Alto</li>

  </ul>
</section>

```

```

<section class="section-meta">
  <h1>Elige tu meta</h1>
  <p>Tu meta es subir o bajar de peso?:</p>

```

```

    <p>
      <input type="radio" name="meta" value="subir" > Aumentar
masa muscular
      <input type="radio" name="meta" value="bajar" > Bajar
porcentaje de grasa
    </p>
  </section>

```

```

<section class="main-button">
  <div>
    <button type="button" onclick="onClickCalcularDieta()">
      Calcular
    </button>
  </div>
</section>

```

```

</form>

```

```

</section>

```

```

<section class="section-resultados">
  <h1 id="title_reultados"></h1>
  <div class="div-resultados">
    <ul class="resultados">
      <li>
        <p id="out_porcentajeGrasa"></p>
      </li>
      <li>
        <p id="out_masamagra"></p>
      </li>
      <li>
        <p id="out_bmr"></p>
      </li>
    </ul>
  </div>
</section>

```

```
</li>
  <p id="out_caloriasMantemimiento"></p>
</li>
<li>
  <p id="out_caloriasMeta"></p>
</li>
<li>
  <p id="out_proteinas"></p>
</li>
<li>
  <p id="out_carbohidratos"></p>
</li>
<li>
  <p id = "out_grasas"></p>
</li>
</ul>

</div>

</section>
</main>

<footer>
  <p class="footer-left">Developed by Alejandro Martinez</p>
  <p class="footer-right">Todos los derechos reservados</p>
</footer>

<script src="dieta.js"></script>
</body>
</html>
```

Anexo N5:**Actividad física**

```
function calcularPorcentajeGrasa(altura, cuello, cintura){
    var    grasa    =    86.01*Math.log10((cintura/2.54)-((cuello/2.54)))-
70.041*Math.log10(altura/2.54)+36.76;
    return grasa
};
```

```
function calcularMasaMagra(peso, grasacorporal){
    return peso*(1-(grasacorporal/100))
};
```

```
function calcularBMR(masamagra){
    return (masamagra*21.6)+370
};
```

```
function calcularCaloriasMantenimiento(bmr, factorActividad){
    return bmr*factorActividad
};
```

```
function calcularCaloriasMeta(meta, caloriasMantenimiento, tipocuerpo){
    var metaSubir; //kilos por mes
    var metaBajar; //kilos por semana

    switch (tipocuerpo){

        case "ec":
            metaBajar = 0.5;
            metaSubir = 1.3;
            break;
```

```
    case "me":
        metaBajar = 0.8;
        metaSubir = 1.0;
        break;

    case "en":
        metaBajar = 1.2;
        metaSubir = 0.5;
        break;
}

if(meta == 'subir'){
    return caloriasMantenimiento + (metaSubir*440);
}
else if (meta == 'bajar'){
    return caloriasMantenimiento - (metaBajar*1100)
}

};

function calcularMacronutrientes(meta, caloriasMeta, tipocuerpo, masamagra){
    var proteinasperkilo;
    var carbohidratosporkilo;
    var grasasporkilo;
    var macros = [];

    switch (tipocuerpo){

        case "ec":
            if(meta = "bajar"){
                proteinasperkilo = 3.0;
```

```

    grasasporkilo = 1.2;
    carbohidratosporkilo = ((caloriasMeta -
(proteinasporkilo*4*masamagra + grasasporkilo*9*masamagra))/4)/masamagra;

    } else{
        proteinasporkilo = 2.2;
        grasasporkilo = ((0.3*caloriasMeta)/9)/masamagra;
        carbohidratosporkilo = ((caloriasMeta -
(proteinasporkilo*4*masamagra + grasasporkilo*9*masamagra))/4)/masamagra;
    }
    break;

    case "me":
        if(meta = "bajar"){
            proteinasporkilo = 2.5;
            grasasporkilo = 1.1;
            carbohidratosporkilo = ((caloriasMeta -
(proteinasporkilo*4*masamagra + grasasporkilo*9*masamagra))/4)/masamagra;

        } else{
            proteinasporkilo = 2.0;
            grasasporkilo = ((0.25*caloriasMeta)/9)/masamagra;
            carbohidratosporkilo = ((caloriasMeta -
(proteinasporkilo*4*masamagra + grasasporkilo*9*masamagra))/4)/masamagra;
        }
        break;

    case "en":
        if(meta = "bajar"){
            proteinasporkilo = 2.3;
            grasasporkilo = 0.9;
            carbohidratosporkilo = ((caloriasMeta -
(proteinasporkilo*4*masamagra + grasasporkilo*9*masamagra))/4)/masamagra;

```

```

    } else{
        proteinas porkilo = 2.2;
        grasas porkilo = ((0.20*caloriasMeta)/9)/masamagra;
        carbohidratos porkilo = ((caloriasMeta -
(proteinas porkilo*4*masamagra + grasas porkilo*9*masamagra))/4)/masamagra;
    }
    break;

}

```

```

var proteinas = proteinas porkilo * masamagra;
var carbohidratos = carbohidratos porkilo * masamagra;
var grasa = grasas porkilo * masamagra;

macros.push(proteinas, carbohidratos, grasa);

return macros
};

```

```

function onClickCalcularDieta(){
    //Traer variables de HTML
    var cuello = document.getElementById("input_cuello").value;
    var altura = document.getElementById("input_altura").value;
    var cintura = document.getElementById("input_cintura").value;
    var peso = document.getElementById("input_peso").value;
    var factorActividad =
document.getElementById("input_nivel_actividad").value;
    var meta = document.querySelector('input[name=meta]:checked').value;
    var tipodecuerpo =
document.querySelector('input[name=tipo]:checked').value;

    //Calcular valores a mostrar

```

```

var porcentajeGrasa = calcularPorcentajeGrasa(altura, cuello, cintura);
var masamagra = calcularMasaMagra(peso, porcentajeGrasa);
var bmr = calcularBMR(masamagra);
var          caloriasMantenimiento          =
calcularCaloriasMantenimiento(bmr,factorActividad);
var caloriasMeta = calcularCaloriasMeta(meta, caloriasMantenimiento,
tipodecuerpo);
var macronutrientes = calcularMacronutrientes(meta, caloriasMeta,
tipodecuerpo, masamagra); //calculo proteinas, carbohidratos y grasas

//Mostrar valores en HTML
var          out_porcentajeGrasa          =
document.getElementById("out_porcentajeGrasa");
out_porcentajeGrasa.innerHTML = "Tu porcentaje de grasa es: " +
porcentajeGrasa.toFixed(0) + "%";

var out_masamagra = document.getElementById("out_masamagra");
out_masamagra.innerHTML = "Tu masa magra es de: " +
masamagra.toFixed(1) + " Kg";

var          out_caloriasMantemimiento          =
document.getElementById("out_caloriasMantemimiento");
out_caloriasMantemimiento.innerHTML = "Tus calorias de mantenimiento son:
" + caloriasMantenimiento.toFixed(0) + " cal";

var out_caloriasMeta = document.getElementById('out_caloriasMeta');
out_caloriasMeta.innerHTML = "Tus calorias meta son: " +
caloriasMeta.toFixed(0) + ' cal';

var out_proteinas = document.getElementById('out_proteinas');
out_proteinas.innerHTML = "Proteinas: " + macronutrientes[0].toFixed(0) + "
gr";

var out_carbohidratos = document.getElementById('out_carbohidratos');

```

```
    out_carbohidratos.innerHTML = "Carbohidratos: " +  
macronutrientes[1].toFixed(0) + " gr";
```

```
    var out_grasas = document.getElementById('out_grasas');  
    out_grasas.innerHTML = "Grasas: " + macronutrientes[2].toFixed(0) + " gr";
```

```
};
```