UNIVERSIDAD POLITÉCNICA SALESIANA
UNIDAD DE POSGRADOS

MAESTRÍA EN CONTROL Y AUTOMATIZACIÓN INDUSTRIALES

Proyecto de investigación y desarrollo
previo a la obtención del Grado de Magister
en Control y Automatización Industriales

DISEÑO Y DESARROLLO DE UN MÓDULO PARA DETERMINAR LA POSTURA HUMANA EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL Y RECONOCIMIENTO DE PATRONES COMO HERRAMIENTA DE SOPORTE EN EL DESARROLLO DE LA MOTRICIDAD GRUESA DE NIÑOS CON DISCAPACIDAD.

Autor: Lenin Germán Aguilar Siguenza

Dirigido por: Vladimir Espartaco Robles Bykbaev

DISEÑO Y DESARROLLO DE UN MÓDULO PARA DETERMINAR LA POSTURA HUMANA EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL Y RECONOCIMIENTO DE PATRONES COMO HERRAMIENTA DE SOPORTE EN EL DESARROLLO DE LA MOTRICIDAD GRUESA DE NIÑOS CON DISCAPACIDAD.

DISEÑO Y DESARROLLO DE UN MÓDULO PARA DETERMINAR LA POSTURA HUMANA EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL Y RECONOCIMIENTO DE PATRONES COMO HERRAMIENTA DE SOPORTE EN EL DESARROLLO DE LA MOTRICIDAD GRUESA DE NIÑOS CON DISCAPACIDAD

AUTOR:

LENIN GERMAN AGUILAR SIGUENZA
Ingeniero Electrónico
Egresado de la Maestría Control y Automatización Industriales

DIRIGIDO POR:

PhD. VLADIMIR ESPARTACO ROBLES BYKBAEV
Ingeniero en Sistemas
Master en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital.
Coordinador del grupo de Investigación en Inteligencia Artificial y Tecnologías de asistencia.
Doctor en Tecnologías de la Información y Comunicación de la Universidad de Vigo.
Docente de la carrera de Ingeniería en Sistemas.
Docente de la Maestría en Métodos Matemáticos y Simulación Numérica en Ingeniería.

Cuenca-Ecuador

Datos de catalogación bibliográfica

AGUILAR SIGUENZA LENIN GERMAN

DISEÑO Y DESARROLLO DE UN MÓDULO PARA DETERMINAR LA POSTURA HUMANA EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL Y RECONOCIMIENTO DE PATRONES COMO HERRAMIENTA DE SOPORTE EN EL DESARROLLO DE LA MOTRICIDAD GRUESA DE NIÑOS CON DISCAPACIDAD.

Universidad Politécnica Salesiana, Cuenca – Ecuador, 2020

MAESTRÍA EN CONTROL Y AUTOMATIZACIÓN INDUSTRIALES

Formato 170 x 240 mm Páginas: 57

Breve reseña de los autores e información de contacto

Autor:
LENIN GERMAN AGUILAR SIGUENZA
Ingeniero Electrónico
lenin.aguilar.s@gmail.com.ec

Dirigido por:
VLADIMIR ESPARTACO ROBLES BYKBAEV
Ingeniero en Sistemas.
Master en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital.
Coordinador del grupo de Investigación en Inteligencia Artificial y tecnologías de asistencia.
Doctor en Tecnologías de la Información y Comunicación de la Universidad de Vigo.
Docente de la Universidad Politécnica Salesiana sede Cuenca.
Docente de la Maestría en Modelos Matemáticos y Simulación Numérica para Ingeniería.
vrobles@ups.edu.ec

IMPRESO EN ECUADOR - PRINTED IN ECUADOR

Contenido

INDICE DE FIGURAS

INDICE DE TABLAS

AGRADECIMIENTO

A Dios, Quien por Su gracia me concedió la fortaleza para la realización de este posgrado.

A mis padres, quienes con incondicional amor, se han esforzado siempre sin escatimar esfuerzos por darme lo mejor de ellos.

A los integrantes de mi hogar, que siempre han sido el motor que me impulsa a concluir lo que empiezo.

Al Ing. Daniel Andrade, que con su apoyo y experiencia se convirtió en parte fundamental para retomar conceptos indispensables para la realización del presente proyecto.

Al Instituto de Parálisis Cerebral del Azuay, IPCA, que me brindó total apretura para el diseño, desarrollo y evaluación de este trabajo.

Lenin Aguilar Sigüenza

PREFACIO

Este trabajo de tesis presenta los resultados de exactitud y precisión de un módulo que realiza la estimación de la postura humana, empleando algoritmos de visión por computadora, mismo que sirve como herramienta de soporte en el desarrollo de la motricidad gruesa de niños con discapacidad.

Este módulo fue diseñado y desarrollado con software libre y su algoritmo usa Tensorflow que es una librearía de acceso público, creada por Google.

PROLOGO

En el presente trabajo de tesis se presenta el Diseño y desarrollo de un módulo para determinar la postura humana empleando técnicas de visión artificial y reconocimiento de patrones como herramienta de soporte en el desarrollo de la motricidad gruesa de niños con discapacidad. Se propone un algoritmo que infiere la postura de los usuarios, empleando redes neuronales con técnicas de visión artificial y haciendo uso únicamente de la cámara integrada en el mismo computador en el que corre el algoritmo.

Para el buen desempeño del módulo propuesto, fue necesario realizar las siguientes etapas:

- Determinar las posturas a estimar.
- Seleccionar una técnica a usar en el algoritmo.
- Construir un corpus de imágenes de las posturas a estimar para entrenar la red neuronal.

Finalmente se presentan los resultados de validación de la propuesta

INTRODUCCION

El instituto de parálisis cerebral del Azuay IPCA es un centro de atención multisectorial e integral de carácter fiscomisional que cuenta con un equipo interdisciplinario en habilitación y rehabilitación medico terapéutico y educación especializada y ofrece programas de gestión acorde a las reales necesidades de los niños, niñas, adolescentes y jóvenes con discapacidad, uno de sus objetivos es conseguir rehabilitar a sus pacientes tanto en motricidad como aprendizaje, para que puedan desenvolverse por sí solos en sus actividades además de lograr la inclusión social para ellos (IPCA, 2019).

El desarrollo de rehabilitación para seres con discapacidad depende de varios factores, como la economía, lugar de nacimiento, residencia de la persona, etc. Así como el soporte emocional que aporten los seres que los rodean. En la actualidad se estima que el 15% de la población total, es decir más de mil millones de personas viven en el mundo con algún tipo de discapacidad, donde la mayor parte de ellos se encuentran en países de ingresos bajos (OMS, 2013)

La forma en que los pacientes pueden realizar actividades por sí solos, disminuye la carga de las personas que están al cuidado de ellos, tanto de sus familiares como los mismos terapeutas del centro, mejorando su calidad de vida. Por ello es de gran importancia el desarrollo de herramientas que permitan brindar un soporte en el aprendizaje motriz de los pacientes con discapacidad, y gracias al avance tecnológico es posible llevar a cabo el desarrollo de dichas herramientas que los motivan durante su rehabilitación.

El presente trabajo busca desarrollar un sistema inteligente para la rehabilitación motriz de personas con discapacidad, y se encuentra estructurado de la siguiente manera: Capitulo 1 Fundamentos teóricos y estado del arte, Capitulo 2 Descripción de la propuesta, Capitulo 3 Experimentación y resultados, Capitulo 4 Conclusiones y recomendaciones.

1. CAPITULO 1: FUNDAMENTOS TEÓRICOS Y ESTADO DEL ARTE.

Hoy en día la estimación de la postura humana tiene un gran número de aplicaciones que van desde el ámbito de la medicina hasta el de la seguridad. En tal virtud, este capítulo se enfoca en brindar a los lectores una breve revisión del estado del arte de las técnicas usadas en visión por computador para la estimación de la postura humana, su esqueletización o seguimiento en ambientes ya sean controlados o no.

Luego de ello se realiza un análisis de las investigaciones y aportes más recientes y se busca determinar las principales dificultades que existen en este campo de la inteligencia artificial a fin de compartir una noción general de cómo los autores han superado dichas dificultades.

La estimación de la postura humana es una técnica utilizada para determinar las acciones que realiza una persona o animal, con la finalidad de discriminar aquellas acciones que pueden resultar de interés para un determinado problema o ámbito de estudio.

Cabe indicar que existen estimaciones que se realizan en ambientes controlados y no controlados, pues en esto juega un papel muy importante la iluminación, las oclusiones, la velocidad de los movimientos, el número de individuos a analizar y la distancia a la que se encuentre la persona desde el dispositivo que capta las imágenes.

La estimación parte de la detección de la persona en cuestión para luego aislarla de los demás objetos de la escena y concentrarse en la extracción de sus características. Como acto siguiente, se realiza el análisis de su cuerpo para identificar articulaciones y su estructura corporal. Una vez obtenida esta información, puede ser entrenada con un corpus de posturas, que no es más que la relación de la posición entre los miembros de su cuerpo, para inferir la acción ejecutada.

Determinar de forma automática la acción que realiza una persona, permite que un dispositivo pueda generar alertas sobre conductas que deben ser atendidas prioritariamente. Ejemplos de estas conductas podrían relacionarse con los ámbitos de la seguridad, el cuidado de la salud, la educación, etc. Por otra parte, es importante mencionar que algunas de estas acciones no son sencillas de detectar cuando existen varios objetos en una escena.

En el campo de la seguridad, la estimación de la postura presta grandes beneficios, tales como la identificación de hurtos en tiendas, bancos, supermercados, actividades como venta de narcóticos, intercambio de mercancías en aeropuertos, etc.

La visión por computador es una de las ramas de la inteligencia artificial que estudia cómo procesar, analizar, clasificar e interpretar imágenes de forma automática. En los

últimos años esta rama ha experimentado una evolución impresionante a un ritmo realmente acelerado. En esta línea, debemos recalcar que esta disciplina aporta en gran medida a los más variados ámbitos, como la seguridad, la medicina, la educación, la inclusión de personas con discapacidad, entre otras. Esto se debe a que permite llevar a cabo procesos para la inspección de imágenes y videos con el fin de determinar, reconocer, clasificar o discriminar acciones, ambientes, objetos, etc.

Por otra parte, el aprendizaje profundo es un mecanismo de clasificación que tiene la valiosa característica de no necesitar descriptores y está fundamentado en redes neuronales recurrentes, convolucionales, aprendizaje competitivo o por refuerzo y otras técnicas más, que son capaces de inferir sin supervisión las características más relevantes de una imagen y las emplean como filtros para realizar la clasificación (Arista-Jalife, Calderón-Azua, Fierro-Radilla, & Nakano, 2017).

Con este capítulo se pretende ofrecer una idea general de la investigación que se ha desarrollado dentro del campo de la visión por computador y el tratamiento de imágenes, para luego elegir la técnica que el autor crea conveniente para desarrollar el presente proyecto.

1.1. Fundamentos teóricos.

1.1.1. Redes neuronales convolucionales.

Al igual que las redes neuronales, permiten detectar patrones en los datos de entrada, pero específicamente con imágenes.

Sus filtros permiten extraer las características relevantes de la imagen imitando el cerebro humano y como procesan esa información.

Usa varias capas de las cuales, cada una se encarga de extraer características específicas tales como líneas, bordes, colores, texturas, formas, etc. que a medida que avanza la profundidad de las capas, detectan características más complejas como ojos, cejas, labios, etc. hasta que llega a la capa que es capaz de determinar que tenemos un rostro o un número, según sea el caso de la imagen digital de entrada.

Cada capa puede estar conformada por etapas, cuya misión es obtener las características más relevantes de la imagen, la primera etapa se trata de un filtro convolucional, dicho filtro se aplica a la imagen para obtener un mapa de características, si se aplican varios filtros de convolución se obtienen diferentes mapas, los cuales se van apilando uno tras otro como se puede apreciar en la Figura 2.

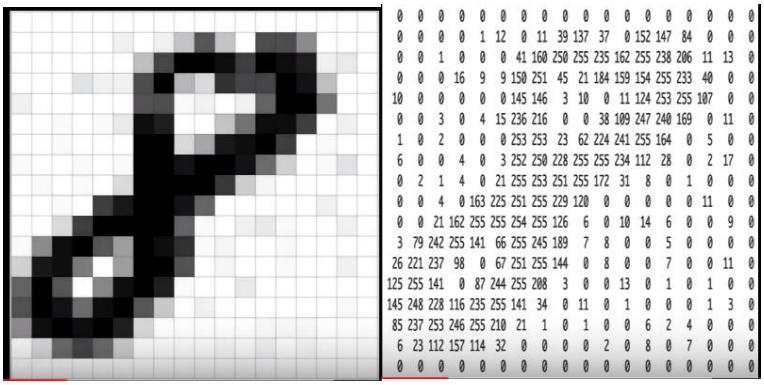


Figura 1 *Imagen que contiene el número 8 y representación como matriz*

La etapa de submuestreo está conformada por dos eventos principales, el primero se trata de una función cuya intención es brindar robustez a la información que se propaga de capa a capa, haciéndola invariante a la translación local, el segundo evento se trata de la reducción de parámetros que se transmiten a la siguiente capa, el objetivo de realizar la etapa de submuestreo es disminuir el número de parámetros que se propagan de capa a capa. Los eventos mencionados se llevan a cabo mediante un proceso similar, mediante la compresión de la información la cual se trata de realizar un promedio de valores en pequeñas cantidades o grupos de valores, lo que se conoce como pooling, estos valores son provenientes de los mapas de características y realiza un promedio de los grupos de valores para reducir los parámetros. De una forma similar a la convolución, el pooling se lleva a cabo mediante el barrido a lo largo de todos los mapas de características, la técnica más usada para submuestreo es la de Max pooling la cual se explica de mejor manera en la sección 1.1.4.

El éxito de las redes convolucionales radica principalmente en los filtros que usa en cada capa y extrae las características de la imagen. Podemos ver en la imagen a continuación que la imagen de entrada se la separa por medio de los filtros en 3 capas (RGB), luego se aplican filtros que extraen otras características y ahora tenemos el mismo tamaño de la imagen, 224x224, pero con una profundidad mayor, 64, y luego se aplica Max pooling y dos filtrados más, con esto la imagen queda reducida a un tamaño de 112x112 pero con una profundidad aun mayor, que en este caso es 128 y de esta manera se van aplicando filtros hasta llegar a la parte donde tomando las características recogidas en los filtros, se forma un vector que luego, empleando una capa softmax realiza la clasificación.

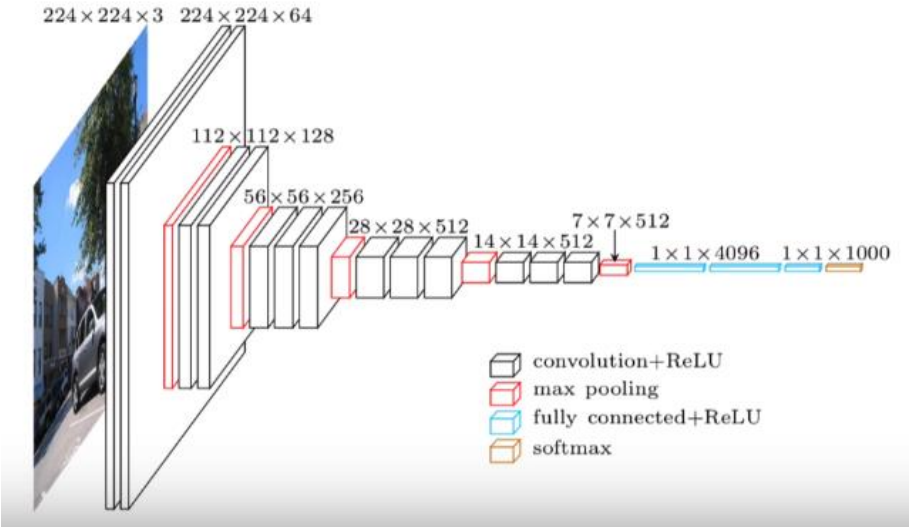


Figura 2 *Red convolucional*

1.1.2. Convolución

La convolución a través de un kernel o filtro que se aplica a la imagen permite extraer unas características, por ejemplo, bordes como se muestra en la siguiente Figura:

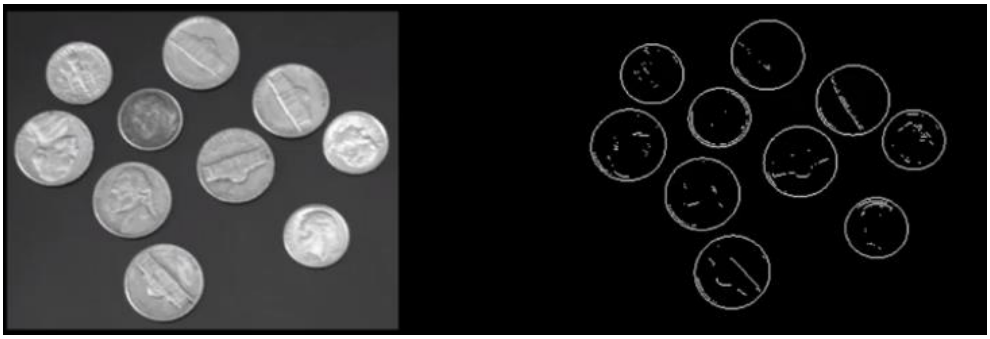


Figura 3 *Imagen en escala de grises y extracción de bordes aplicando convolución*

La convolución es un proceso matemático con muchas aplicaciones en campos de la ciencia y tecnología. Uno de los usos más importantes es el filtrado de imágenes, que en resumidas cuentas lo que hace es modificarlas, realzando o mejorando las características que son de nuestro interés con el objetivo de extraer información relevante.

En nuestro campo, usamos la convolución de matrices como método para realizar el filtrado de imágenes, que a fin de cuentas es una matriz.

Una imagen será para nosotros una función bidimensional $z = F(x, y)$, donde x e y son coordenadas espaciales que ubican el pixel en cuestión, mientras que z es la intensidad del color en dicha coordenada.

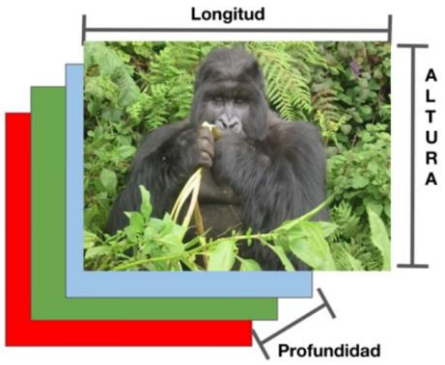


Figura 4 *Dimensiones de imagen a RGB*

El valor de cada pixel va de 0 a 255 y cuando se trabaja con una imagen digital a color, tendremos 3 matrices, R, G y B, que representan la intensidad del color rojo, verde y azul respectivamente (Gimenez Palomarez, Monsoriu, & Alemany-Martinez, 2016).

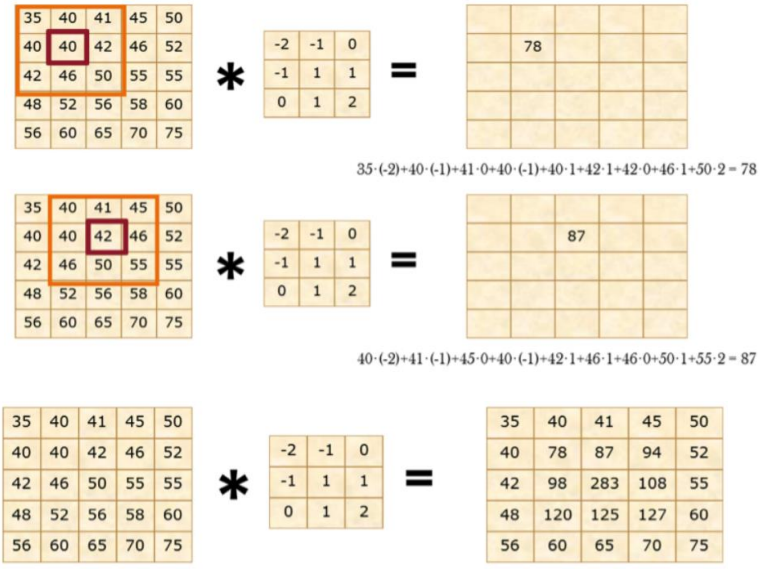


Figura 5 *Proceso de convolución 2D*

El proceso de convolución se describe como sigue:

El filtro elegido va a ir barriendo toda la imagen (matriz) y ejecutando la operación de convolución, lo que nos da como resultado otra matriz que tendrá alto y ancho menor pero mayor profundidad dependiendo del número de filtros aplicados.

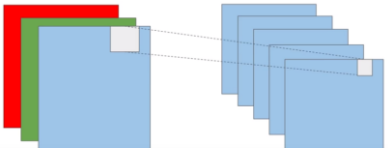


Figura 6 *Convolución de imagen*

La profundidad viene dada por el número de filtros aplicados a la imagen:

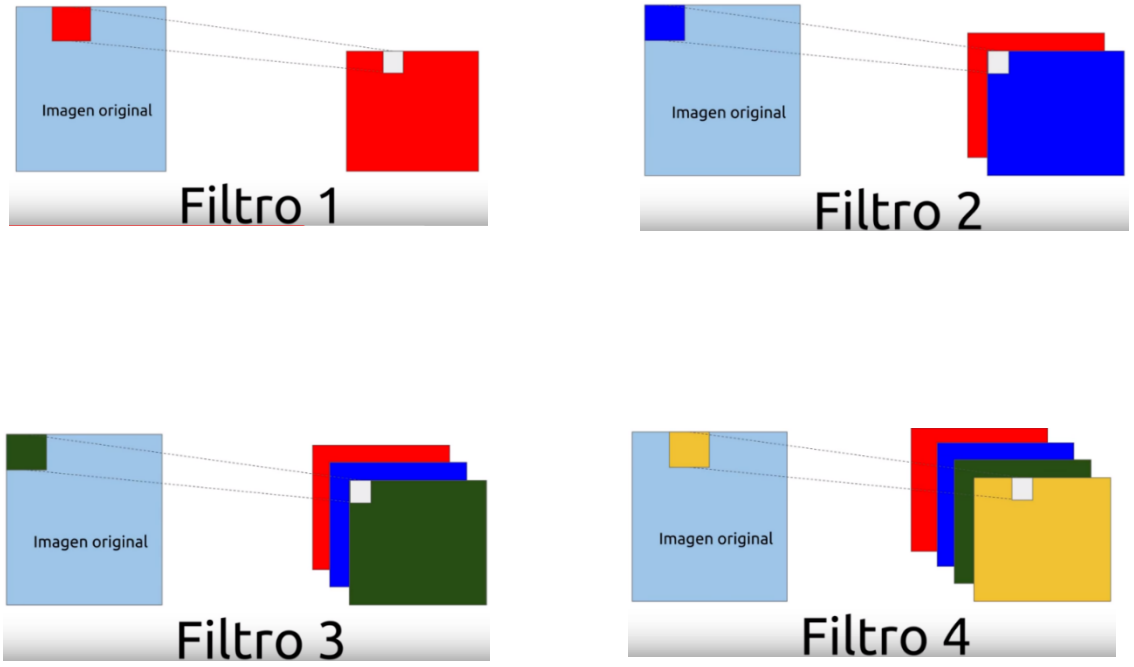


Figura 7 *Aplicación de filtros y formación de profundidad*

1.1.3. Stride o paso.

Es el parámetro que especifica qué tanto vamos a ir recorriendo nuestro filtro, por ejemplo, de pixel en pixel o de dos pixeles en dos pixeles.

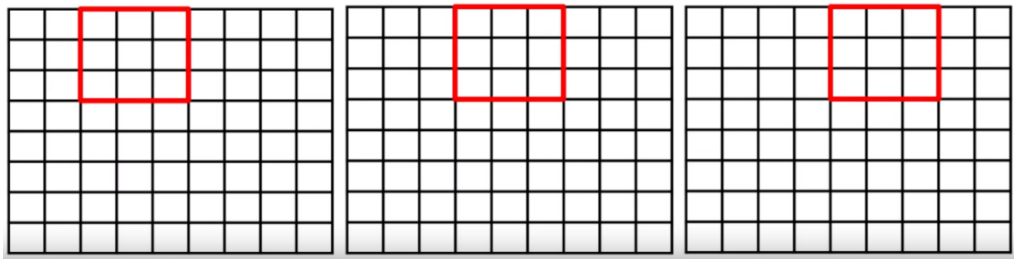


Figura 8 *Barrido del filtro de convolución con stride=1, el filtro recorre 1 pixel*

1.1.4. Pooling.

Es un filtro que recorre la imagen, obteniendo valores, para formar una nueva matriz y de esta forma proporciona una nueva, con características promediadas o las más relevantes, lo cual proporciona una matriz con tamaño espacial menor. Y por lo tanto nuestra red tendrá menos parámetros que aprender y reduce significativamente el coste computacional.

Si hablamos de Max pooling, su filtro forma una nueva matriz con la obtención del número mayor de cada porción de la matriz original que va iterando esta función se realiza sin solapamiento y en grupos de 2x2 en imágenes.

Average pooling, forma una nueva matriz con el promedio de los números que ocupan la porción de matriz original que va abarcando su filtro.

Figura 9 *Max Pooling y Average pooling*

1.1.5. Relu.

Relu (rectified linear unit), es una función de activación que permite el paso de todos los valores positivos sin alterarlos, mientras que anula los números negativos, asignando como cero la salida que corresponda a una entrada menor que cero.

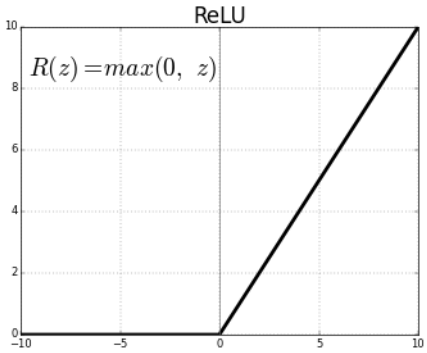


Figura 10 *Función de activación RELU*

Ya que, en el procesamiento de imágenes, los valores negativos no son importantes; pero los valores positivos, después de la convolución, deben pasar a la siguiente capa para su procesamiento, se utiliza generalmente la función de activación RELU o una de sus variantes (Noren, 2019).

1.1.6. Capas fully-connected (DNN)

Las capas Fully connected, es la estructura más básica de las redes neuronales, en donde, las salidas se calculan a partir de los valores de entrada (xi), ponderados por pesos

entrenables (wi) y sumados a un valor de bias (b), mismo que luego se aplican como argumento a una función de activación

$$n = \sum_i (w_i x_i) + b$$

$$y = f(n) = f(\sum_i (w_i x_i) + b)$$

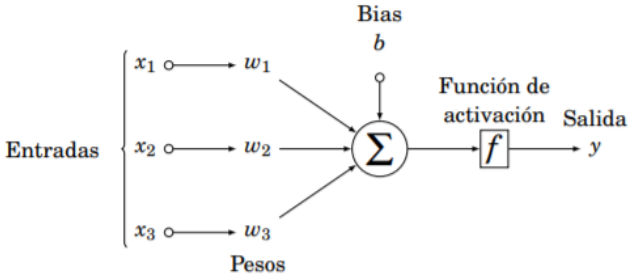


Figura 11 *Diagrama del cálculo de la salida de una neurona en una capa DNN*

Como podemos ver, la estructura de las capas fully connected, es equivalente al modelo de aprendizaje del perceptrón multicapa. Además, estas capas pueden utilizarse como parte de arquitecturas más complejas (Benito Gorron, 2018).

1.1.7. Softmax.

Es una función de activación para clasificar data, que es capaz de decirnos si pertenece a una u otra determinada clase.

Nos da un porcentaje de probabilidad de pertenencia a cada clase. Normalmente va en la última capa. Y viene dada por la siguiente expresión:

$$\frac{e^{Y_j}}{\sum_{k=1}^{K} \left(e^{Y_k}\right)}$$

Y = Salida de las capas ocultas

K = Número de clases en nuestro modelo

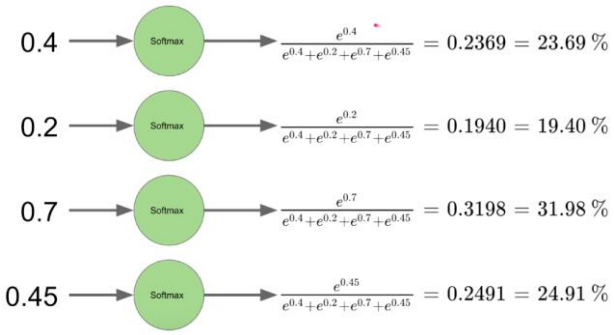


Figura 12 *ejemplo de función SoftMax*

1.2. Software utilizado

1.2.1. Tensorflow.

Es una librería de código abierto, especialmente desarrollada para calculo numérico, con una arquitectura flexible y permite un rápido desarrollo para plataformas como CPU's, GPU's y TPU's. La forma de programación de esta librería es por medio de grafos de flujo de datos, cuyos links representan los conjuntos de datos multidimensionales que tienen el nombre de tensores (Valle Barrio, 2018).

Fue desarrollada por Google para emplearla en redes neuronales que detecten y descifren patrones y correlaciones, similares al aprendizaje y razonamiento usado por humanos (Fundación Wikimedia, Inc., 2019).

Tensoflow tiene la virtud de optimizar las variables del grafo ya que calcula automáticamente los gradientes necesarios para esto, ya que el grafo es una combinación de expresiones matemáticas sin mucha complejidad.

1.2.2. Anaconda.

Es un repositorio de código abierto de cientos de paquetes populares de ciencia de datos, junto con el paquete conda y el administrador de entorno virtual para Windows, Linux y MacOS. Conda facilita y agiliza la instalación, ejecución y actualización de entornos complejos de ciencia de datos y aprendizaje automático como scikit-learn, TensorFlow y SciPy (Inc., 2019).

1.2.3. Python.

Está desarrollado bajo una licencia de código abierto aprobada por OSI, por lo que es de libre uso y distribución, incluso para uso comercial. La licencia de Python es administrada por la Python Software Foundation. El Python Package Index (PyPI) aloja

miles de módulos de terceros para Python. Tanto la biblioteca estándar de Python como los módulos aportados por la comunidad permiten infinitas posibilidades para poder programar en un lenguaje de alto nivel, una de sus debilidades es la creación de objetos para definir escalares, esto aumenta la sobrecarga de cálculo numérico, por lo cual es indispensable utilizar herramientas que trabajan en conjunto con Python como numpy o theano (Phyton, 2019).

1.2.4. PoseNet.

La estimación de postura se refiere a las técnicas que detectan figuras humanas en una imagen o video, esta tecnología no necesita saber quién necesariamente está en la imagen, basta con saber en dónde se encuentran partes específicas del cuerpo humano como articulaciones o partes del rostro de las personas.

PoseNet es una herramienta de TensorFlow que nos permite estimar puntos clave de una o varias personas en una imagen, esta herramienta ha sido desarrollada en conjunto con Google. Se trata de un modelo preentrenado el cual puede ser utilizado libremente desde el navegador, o desde un dispositivo móvil.

La estimación realizada por PoseNet ocurre en dos fases:

- Una imagen RGB alimenta la entrada de una red neuronal convolucional.
- Se utiliza un algoritmo de decodificación para decodificar poses, puntajes de confianza de pose, posiciones de puntos clave y puntajes de confianza de puntos clave de las salidas del modelo.

A continuación, se presenta una breve descripción de los parámetros más importantes en el desarrollo del trabajo presente:

Pose: Es el resultado final que entrega PoseNet, se trata de una lista que contiene los puntos clave de una persona detectada en una imagen junto con las probabilidades de su estimación.

Punto Clave: Parte de una persona que se desea estimar como la nariz, ojos, orejas, articulaciones etc. y contiene una posición y un puntaje de confianza. PoseNet actualmente detecta 17 puntos clave como se puede apreciar en el desarrollo del modelo descrito en el presente documento.

1.2.4.1.Puntuación de detección para Key-Points

PoseNet ha experimentado con diferentes métodos para asignar una puntuación de nivel de puntos clave e instancia a las estimaciones generadas por el algoritmo de decodificación. El primer método de puntuación a nivel de puntos clave sigue y asigna a cada punto detectado una puntuación de confianza, $s_{j,k} = h_k\,(y_{j,k})$. Un inconveniente de este enfoque es que los puntos clave faciales bien localizables

generalmente reciben puntajes mucho más altos que los puntos clave mal localizables como la cadera o la rodilla (Papandreou, 2018)

Para calibrar las puntuaciones, se tiene una métrica de evaluación de similitud de puntos clave object keypoint similarity (OKS), que utiliza diferentes umbrales de precisión para penalizar los errores de localización de dichos puntos

y se define mediante:

$$s_{j,k} = E\{OKS_{j,k}\} = p_k(y_{j,k}) \int_{x \in \mathcal{D}_{R(y_{j,k})}} \hat{h}_k(x) \exp\left(-\frac{(x - y_{j,k})^2}{2\lambda^2 k_k^2}\right) dx$$

Donde $\hat{h}_k(x)$ son las puntuaciones de Hough normalizadas en $\mathcal{D}_{R(y_{j,k})}$ y λ es la raíz cuadrada del área del cuadro delimitador que contiene todos los puntos detectados.

Como puntaje de nivel de instancia final, se usa la suma de los puntajes de los puntos clave que aún no han sido reclamados por instancias de mayor puntuación, normalizado por el número total de puntos clave:

$$s_j = \left(\frac{1}{K}\right) \sum_{k=1:K} s_{j,k} \left[\left|\left|y_{j,k} - y_{j',k}\right|\right| > r, para\ todo\ j' < j\right]$$

Donde r es el radio de supresión no máxima, estas métricas han sido evaluadas dentro de las competencias de COCO, el cual es un conjunto de datos de detección, segmentación y subtitulación de objetos a gran escala (COCO, 2019).

Scoring	NMS	AP	$AP^{.50}$	$AP^{.75}$	AP^M	AP^L
Hough [33]	hard	0.632	0.838	0.693	0.593	0.698
Expected-OKS	hard	0.647	0.843	0.703	0.599	0.718
Hough [33]	soft	0.645	0.853	0.703	0.610	0.702
Expected-OKS	**soft**	0.665	0.862	0.719	0.623	0.732

Figura 13 *Rendimiento de PoseNet en los puntos clave de COCO val split. Evaluación del modelo ResNet-101 a escala única para diferentes puntos clave y configuraciones de supresión no máxima.*

1.2.5. Mobilenet.

Modelo de red que se enfoca tanto en disminuir el tamaño de la red como también en su exactitud y optimización de la latencia.

Mobilnet ha cambiado la estructura básica de ejecución de las capas convolucionales tradicionales, ya que se aplica la convolución de un solo kernel, con las dimensiones deseadas a la matriz de entrada y este resultado es la entrada de otra capa de convolución

de tamaño fijo de filtro 1x1 con el número de filtros deseados, este proceso se llama Depthwise Separable Convolution.

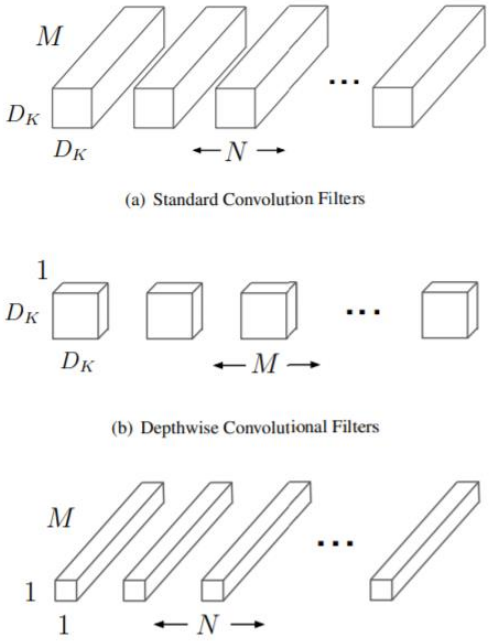


Figura 14 *Conversión de una capa de convolución convencional a una capa depthwise*

En la figura anterior podemos ver la repristinación de la conversión de una capa de convolución convencional a una capa de Depthwise Separable Convolution (Howard, et al., 2017).

Con este nuevo proceso se reduce considerablemente el tamaño de la red y el esfuerzo computacional necesario para correr el modelo sin afectar su exactitud.

En un proceso de convolución normal, siendo D_I la dimensión de la matriz de la imagen de entrada, M su profundidad, D_F la dimensión del filtro y N la profundidad de la salida de acuerdo con el número de filtros empleados, tenemos que su coste computacional es:

$$D_I \cdot D_I \cdot M \cdot N \cdot D_F \cdot D_F$$

Mientras que el trabajo computacional al ejecutar el modelo Depthwise Separable Convolution está dado por la expresión:

$$D_I \cdot D_I \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_I \cdot D_I$$

Como se puede observar, el realizar la Depthwise Separable Convolution es $N + {D_I}^2$ veces mas ligera que ejecutar la convolución convencional (Gamino del Rio & Sanchez, 2018).

1.2.6. Pose estimation.

Es un modelo de visión que nos entrega la estimación de la postura a partir de la determinación de las articulaciones del cuerpo en una imagen o video.

Identifica la figura humana en una imagen o video y nos entrega las coordenadas de ubicación de los puntos clave del cuerpo de dicha figura, como por ejemplo los hombros, codos, ojos, nariz, etc.

Posenet procesa la imagen de entrada, cualquiera que sea su tamaño, pero la exactitud de predicción va de acuerdo al paso de salida que elijamos y por supuesto depende mucho del rendimiento del procesador que ejecute el algoritmo.

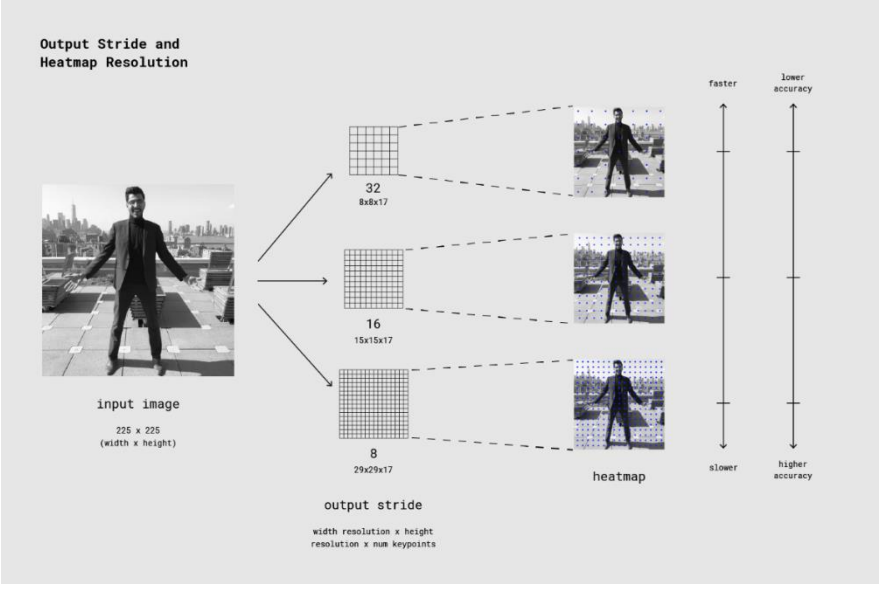


Figura 15 *Posenet de TensorFlow*

1.3. Métricas de evaluación.

Dentro de la fase de evaluación de todo modelo, se realiza la cuantificación de la eficacia en las estimaciones, para lo cual se tienen métricas que nos ayudan a determinar si el modelo obtenido trabaja de manera eficiente o si comete equivocaciones al

momento de realizar las estimaciones, para evaluar el modelo se tienen las siguientes métricas:

1.3.1. Matriz de confusión

También conocida como matriz de error, es una tabla en la cual se visualiza el rendimiento del algoritmo llevado a cabo mediante aprendizaje supervisado, cada columna representa el número de estimaciones de cada clase, mientras que cada fila representa a las instancias en la clase real (Stehman, 1997).

valor estimado ⇩

valor real ⇨

$N_{muestra}$	$Clase_1$	$Clase_2$	… .	$Clase_n$
$Clase_1$				
$Clase_2$				
….				
$Clase_n$				

Tabla 1 *Matriz de confusión de nxn*

1.3.2. Exactitud (Accuracy)

Se refiere a la capacidad que tiene el modelo de aproximar las señales en su entrada a su valor real, en términos de repetitividad es el grado en el que mediciones repetidas en las mismas condiciones, muestran los mismos resultados (Bièvre, 2009). Matemáticamente se define como:

$$Acc(y, \hat{y}) = \frac{1}{n_{muestras}} \sum_{i=0}^{n-1} 1(y_i = \hat{y}_i) \qquad (1)$$

Donde $y_i$es el valor real de la i-ésima muestra, y $\hat{y}_i$ es el valor estimado correspondiente.

1.3.3. Precisión Recall y F1-Score

Las métricas de Precisión y Recall ayudan a verificar la habilidad del clasificador para encontrar todas las muestras positivas, y no calificar como de manera errónea algo que está mal o es negativo, respectivamente; su calificación varía entre cero y uno (Flach, 2015).

$$RECALL = \frac{t_p}{t_p + f_n} \; ; \; PRECISION = \frac{t_p}{t_p + f_P} \qquad (2)$$

Donde $t_p$es el número de verdaderos positivos, f_n es el número de falsos negativos y f_P es el número de falsos positivos.

Mientras que la métrica F1-Score puede ser interpretada como una media ponderada de la precisión y recall. (Flach, 2015)

$$F1 = \frac{2*(PRECISION*RECALL)}{PRECISIÓN+RECALL} \qquad (3)$$

1.4. Estado del arte

Una vez que se ha revisado de forma breve las definiciones de visión artificial y aprendizaje profundo, a continuación, se recogen algunos de los trabajos más relevantes relacionados con la estimación de la postura humana.

Un trabajo aportado por Yang en 2018, estima la postura humana en 2D o 3D a partir de una imagen o video dado. Se basa en dos redes: una como estimador y otra como discriminador, utiliza un generador condicional para la estimación de la pose desde una imagen de entrada. El generador está entrenado para generar muestras de una manera que confunde al discriminador, que a su vez intenta distinguirlas de las muestras reales.

Este trabajo propone el adversarial learning, el mismo que toma en consideración dos factores clave:

- La descripción en la correspondencia imagen- postura.
- Restricciones de las articulaciones del cuerpo humano.
- Usa estimaciones tridimensionales existentes como base, ya que las poses tridimensionales predichas a partir de imágenes en ambientes no controlados, son usadas para el mejor aprendizaje de su estimador y funcionan efectivamente incluso con tareas discriminatorias muy duras (Yang, et al., 2018).

Por otro lado, con el uso de redes convolucionales Bogo en 2016, presenta la estimación 3D completamente automática de la postura y forma del cuerpo a partir de una imagen. Lo realiza en dos pasos:

1. Usando una red neuronal convolucional, estima en 2D las junturas (articulaciones) y para cada juntura la red proporciona un valor de confianza.
2. Estima la postura 3D y la forma o silueta de la persona a partir del paso anterior usando un modelo generativo 3D llamado SMPL.

El proceso en general es llamado SMPLify, que básicamente lo que realiza es una estimación ascendente con la red neuronal Convolucional y una verificación descendente con el modelo Generativo, luego realiza una aproximación de la silueta usando cápsulas como eslabones, para formar el cuerpo uniendo las junturas.

Realiza una optimización basada en el gradiente de sus radios y longitudes de eje para minimizar la distancia bidireccional entre las cápsulas y la superficie del cuerpo (Bogo, et al., 2016).

Un interesante enfoque se presentó en 2017 por parte de Xiaoqiang, ya que en este artículo, se expone un punto de vista para el reconocimiento de acciones humanas extraídas de las poses claves de un esqueleto con SVM (support vector machines).

Los resultados experimentales demostraron que las acciones humanas pueden ser reconocidas por solo unos pocos cuadros de imágenes, es decir, con algunas posturas esqueléticas representativas se pueden describir y distinguir acciones tales como agitar las manos, aplaudir, caminar, sentarse, levantarse entre otras.

A partir de la extracción de características utilizando las posiciones relativas pares de las articulaciones, las posiciones de las poses clave de la esqueletización se encuentran con la ayuda de SVM.

En el procedimiento de prueba se provee un método para calcular el puntaje de cada posición inicial para reconocer la acción.

La prueba del modelo fue validada con tres conjuntos de datos de referencia, los cuales son: el MSR Action 3d, UTKinect y el Florence 3D, observando que su rendimiento se degrada cuando la acción es poco variada y no informativa (Xiaoqiang, Zhang, & Liao, 2017).

Otro trabajo que también usa redes convolucionales es el paper "Combinación de imágenes y marcas de segmentación para esqueletización en la naturaleza.". Este documento propone, otra referencia alternativa para la extracción de esqueletos de imágenes no binarias, que son más complejas, por su forma, fondo, apariencia, colores, textura, tamaño y escala. Para esto emplea una red neuronal convolucional (CNN) de dos flujos de datos, que usa la imagen original y su correspondiente mapa de probabilidad de segmentación semántica como entradas, partiendo de la primicia de que el mapa de probabilidad de segmentación semántica es complementario a la imagen de color correspondiente.

Mediante funciones de múltiples escalas se lleva a cabo experimentos con el Dataset SK-LARGE.

Para manejar el problema de la diversidad de escala, emplean la red de arquitectura U-Net y FPN que utilizan una red descendente para fusionar características desde el plano superior al inferior, capa por capa.

También se utiliza el modelo Deeplab para obtener la probabilidad de segmentación correspondiente al mapa de una imagen, y luego pasar la imagen original y el mapa de probabilidad de segmentación a la red convolucional para predecir el esqueleto (Xiaolong, Pengyuan, Xiang, & MIng-MIng, 2017).

En el ámbito de la estimación de posturas, son muy utilizadas las redes convolucionales, es así que en el artículo de Tompson en 2014, se utiliza una combinación de una red convolucional y un modelo espacial, MRF, que muestra la ubicación espacial de cada parte del cuerpo, en un marco de aprendizaje unificado.

El modelo de estimación, en una primera etapa realiza la localización de partes del cuerpo mediante una arquitectura de red Convolucional profunda, cuya entrada es una imagen RGB que contiene una o más personas y la salida es un mapa de calor, que arroja una probabilidad por píxel de las ubicaciones de las articulaciones más importantes en el esqueleto humano.

La red de convolución se desplaza sobre la imagen de entrada para producir una salida de mapa de calor denso para cada articulación del cuerpo. Una ventaja del modelo de ventana deslizante es que el detector es invariante a la traslación.

El modelo espacial es usado para hacer cumplir la consistencia de la pose global, que aún contiene falsos positivos, que son anatómicamente incorrectos.

Luego de realizar la detección de partes con la red convolucional y obtener los mapas de calor, entrena el modelo espacial con éstos y finalmente combina los dos métodos y propaga hacia atrás toda la red, logrando una sintonización más fina y aumentando aún más el rendimiento.

Esta propuesta usa hardware de nivel básico y se ejecuta a velocidades de cuadros cercanas a tiempo real, lo que la vuelve interesante para muchas áreas de aplicación (Tompson, Jain, LeCun, & Bregler, 2014).

En 2017, se publicó un trabajo donde los autores construyen un modelo sobre Redes Neuronales Recurrentes (RNN) con Long-Term Memory (LSTM) que aprende a enfocarse selectivamente en las articulaciones más relevantes del esqueleto humano en cada cuadro de imagen de entrada y asigna diferentes niveles de atención a las salidas de dichos cuadros.

Con el módulo de atención espacial se lleva a cabo la extracción de forma automática y adaptativa de las articulaciones dominantes y se asigna un nivel de importancia diferente a cada articulación.

Con el módulo de atención temporal el método asigna diferentes pesos de atención a cada cuadro dentro de la secuencia.

Por ejemplo, las articulaciones mano, codo y cabeza se relacionan con la acción "beber", mientras que las articulaciones de las piernas se pueden considerar como ruido.

Para concluir hace un entrenamiento combinando la atención espacial y temporal con una pérdida de entropía cruzada.

Sus experimentos los realizaron con dos conjuntos de datos: el conjunto de datos de interacción SBU Kinect (conjunto de datos de interacción con dos sujetos) y el conjunto de datos RGB + D (Song, Lan, Xing, Zeng, & Liu, 2017).

En el artículo "Estimación de pose 2D de múltiples personas en tiempo real utilizando campos de afinidad de partes." Los desarrolladores se enfocan en un algoritmo para detectar la pose en 2D de múltiples personas en tiempo real, partiendo de la identificación de articulaciones y luego asociando las mismas al individuo que corresponde, mediante campos de afinidad de partes, para así formar su esqueleto.

Los autores realizan una representación no paramétrica y explícita de los puntos claves que contienen información tanto de la posición como la orientación de las extremidades humanas. Una red neuronal de propagación hacia adelante obtiene un mapa de confianza bidimensional de la ubicación de las partes del cuerpo y un conjunto de vectores de campo de la afinidad de partes que contiene información de los grados de asociación entre ellas.

Estos mapas de confianza y campos de afinidad son analizados por un algoritmo que realiza la inferencia y arroja como salidas los puntos clave bidimensionales de los sujetos en la escena. Seguidamente emplea una arquitectura conjunta para el aprendizaje, la detección y asociación de partes.

Figura 16 ***Mapas de confianza para detección de puntos clave***

Finalmente, evalúa el método con dos puntos de referencia para la estimación de pose multifactorial: El MPII Human Pose Database y el COCO 2016. Estos dos conjuntos de datos recopilan imágenes en diversas circunstancias de la vida real, como el hacinamiento, la oclusión y el contacto entre individuos que permiten evaluar los algoritmos como si estuvieran haciendo una prueba de campo (Cao, Tomas, Shih-En, & Yaser, 2017).

Continuando con el uso de redes convolucionales para la estimación de la postura humana, en el artículo "Estimación monocular de la postura humana en 3D utilizando aprendizaje por transferencia y supervisión mejorada de redes neuronales convolucionales." los autores proponen un método basado en CNN (Convolutional neural network) para hacer una estimación de la postura del cuerpo humano en 3D a partir de una imagen RGB en un entorno genérico.

Dada una imagen RGB, estima la postura humana 3D en el sistema de coordenadas de una cámara previamente calibrada, la postura se representa como un vector de posiciones conjuntas pero segmentadas con un nivel jerárquico.

Posteriormente se estima la posición general de las articulaciones del esqueleto con el algoritmo planteado que consta de los siguientes tres pasos:

1. El sujeto se localiza en el cuadro delimitador a partir de mapas de calor 2D calculados con una red neuronal convolucional conocida 2DPoseNet.
2. La postura 3D centrada se calcula con la ayuda de un segundo cálculo realizado con una CNN conocida 3DPoseNet.

3. Se realiza la corrección de la perspectiva y las coordenadas de la postura en 3D (Dushyant, et al., 2017).

En cuanto a la estimación en 2D, un trabajo de Belagiannis & Zisserman en 2017, presenta una estimación recurrente en donde la arquitectura de ConvNet consiste en dos módulos: un módulo de Propagación hacia Adelante que se ejecuta una vez y un módulo Recurrente que, como su nombre lo indica, se puede ejecutar varias veces.

El módulo de Propagación hacia Adelante, basado en una arquitectura de regresión de mapa de calor, actúa principalmente como un sistema detector que reacciona a los cambios en su entorno, para mantener algún estado concreto del sistema.

Por su parte, el módulo Recurrente, que mejora las predicciones del mapa de calor, aporta progresivamente más contexto cada vez que se ejecuta, porque el campo receptivo efectivo aumenta con cada iteración. Puede ejecutarse iterativamente para aumentar la eficacia receptiva de la red y así mejorar el rendimiento.

Las tres principales contribuciones de este modelo son las siguientes:

1. Combinar una red de propagación hacia adelante con módulo recurrente.
2. El modelo puede ser entrenado de extremo a extremo y desde cero, y las pérdidas auxiliares pueden ser incorporadas para mejorar el rendimiento.
3. Una investigación preliminar para mejorar la estimación de la postura humana ante oclusiones (Belagiannis & Zisserman, Recurrent Human Pose Estimation, 2017).

Para la estimación de la postura humana en videos los autores, Pfister, Charles, & Zisserman, presentan en 2015 un enfoque con Deep Convolutional Networks (ConvNets), que plantea una arquitectura de red neuronal profunda para regresiones de mapas de calor, capas de fusión espacial, flujo óptico, que es usado para alinear las predicciones de los mapas de calor a partir de los frames vecinos y una capa de agrupación paramétrica que combina los mapas de calor alineados y forma un mapa de confianza.

Al hacer la regresión de un mapa de calor de las posiciones (que se hacen por separado para cada articulación), los mapas de calor de los frames vecinos pueden deformarse y alinearse usando un flujo óptico, confirmando la posición efectiva temporalmente.

Las capas de fusión espacial, son capas de convolución capaces de aprender las dependencias entre las partes del cuerpo humano, con lo que determinan las posturas que son cinemáticamente imposibles y las eliminan.

La idea clave del enfoque de este trabajo es que, dado la localización los objetivos de predicción se pueden usar vectores de flujo óptico denso para tramar las posiciones predichas sobre una imagen (Pfister, Charles, & Zisserman, 2015).

1.5. Aprendizaje mediante posturas

El IPCA, Instituto De Parálisis Cerebral Del Azuay, es un centro de atención multisectorial e integral que atiende a niños, niñas, adolescentes y jóvenes con discapacidad. Uno de sus servicios es la terapia física, que hace uso de técnicas y medios físicos, naturales y cinéticos con el fin de habilitar, rehabilitar y reintegrar al paciente a la sociedad.

Para el desarrollo de la motricidad gruesa, el departamento de terapia física desarrolla el entrenamiento de adopción de posturas en sus pacientes, mediante las cuales desarrollan su destreza para realizar actividades, aumentar su capacidad de prestar atención, y estimular los músculos y articulaciones para facilitar las actividades que necesitan realizar cotidianamente. Entre estas posturas se tienen:

- ✓ **Postura de inhibición:** Esta postura consiste en sentarse en el piso, con las piernas y brazos cruzados de tal forma de contraer varios músculos de las extremidades que envían señales a sus músculos opuestos para que se relajen o se inhiban, de tal manera que se mantenga el juego de las articulaciones, se trata de un mecanismo protector.
- ✓ **Postura de gateo:** En esta posición los niños tienen que simular movimientos básicos de un bebe, en toda persona es una etapa normal en el desarrollo que ayuda a prepararse para etapas posteriores, su importancia se debe a que en esta posición los dos hemisferios cerebrales deben estar coordinados para poder mover tanto las extremidades del lado derecho como las del lado izquierdo.
- ✓ **Postura de caballero:** La postura consiste en simular la forma en la que una persona se incorpora al levantarse del suelo tratando de no forzar demasiado las articulaciones y músculos del cuerpo.
- ✓ **Postura levantar peso:** Esta es una posición en la cual se busca que los niños puedan levantar objetos del piso, para lo cual tienen que inclinarse hacia el objeto desplazando una rodilla hacia el piso, mientras la otra pierna se prepara para volver a levantarse luego de haber atrapado el objeto con sus manos.
- ✓ **Postura reposo de pie:** en esta posición se encuentran todos los músculos en estado de relajación, el objetivo de esta posición es enseñar a los niños a mantenerse de pie sin generar demasiada tensión en la planta de sus pies.
- ✓ **Postura Sentado:** En esta posición se pretende que los niños se encuentren más cómodos y calmados de tal forma que puedan prestar atención a las instrucciones de los docentes o terapeutas.

2. CAPITULO 2: DESARROLLO DEL MODELO CONVOLUCIONAL

En base a la revisión del estado del arte, se decidió optar por el uso de redes neuronales convolucionales junto con la herramienta PoseNet. El modelo propuesto en este trabajo consta de tres etapas principales, la primera se trata de la generación de un set de imágenes con las posturas propuestas, en la segunda etapa se obtienen los esqueletos de cada imagen mediante PoseNet y finalmente en la tercera etapa se realiza la inferencia de imágenes nuevas, es decir diferentes a las usadas en el entrenamiento para estimar la posición de una persona.

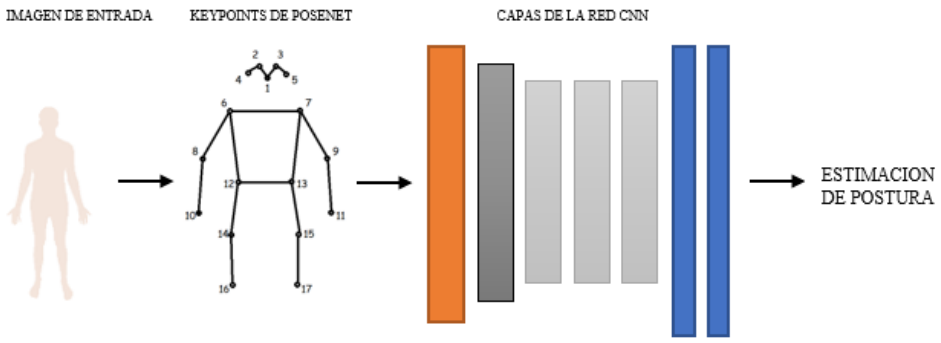


Figura 17 *Diagrama de estimación de posturas mediante redes CNN*

Las posturas mencionadas son aprendidas por los pacientes, con cierto grado de dificultad, por su condición de déficit de atención en ciertos casos, o por el limitado desarrollo de su motricidad gruesa, lo que genera en ellos un sentimiento de frustración.

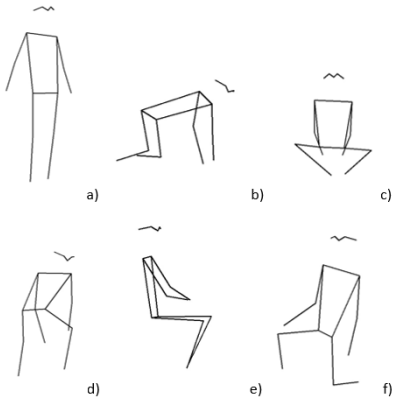


Figura 18 *Posturas obtenidas mediante PoseNet de TensorFlow a) Reposo de pie, b) Gateo, c) Inhibición, d) Mecánica corporal levantar peso, e) Sentado, f) Postura de caballero*

En tal virtud y conociendo de parte del IPCA que los niños, niñas, adolescentes y jóvenes con discapacidad interactúan con mayor facilidad y motivación frente a un equipo electrónico, se plantea el Diseño y desarrollo de un módulo para determinar la postura humana empleando técnicas de visión artificial y reconocimiento de patrones como herramienta de soporte en el desarrollo de la motricidad gruesa de niños con discapacidad, misma que propone la postura que debe realizar el paciente y asigna un tiempo para que éste la realice y desde ahí empieza a ejecutar la comparación de la postura propuesta con la capturada por la cámara del computador, y cuando el algoritmo determina un 85%, de similitud devuelve un mensaje de éxito. Este valor fue elegido debido a que en algunas ocasiones los pacientes necesitan hacer un esfuerzo demasiado alto para lograr colocar todas sus articulaciones en la posición correcta para alcanzar la postura deseada. El porcentaje de similitud es proporcionado por la función de activación Softmax de la última capa del modelo, que muestra el porcentaje de probabilidad de pertenencia a cada clase.

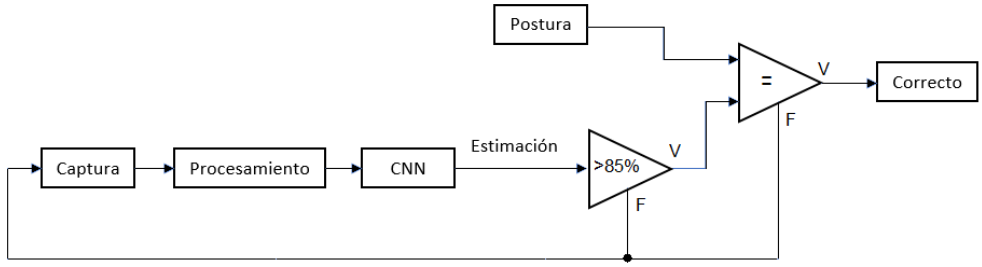


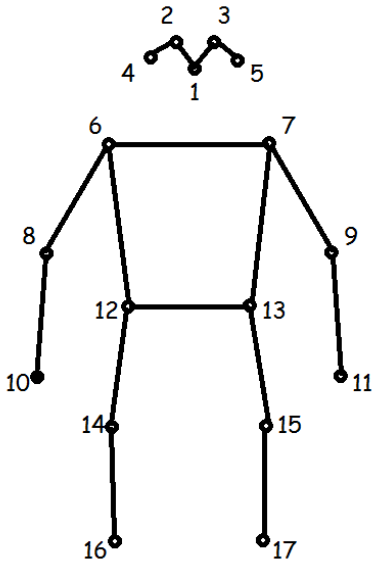
Figura 19 *Diagrama de funcionamiento propuesto*

2.1. Esqueletización

Para realizar la comparación de la postura ejecutada por el usuario con la postura propuesta por el terapeuta, nuestro modelo realiza una captura de la escena por medio de la cámara incorporada en el computador y de cada imagen obtiene los puntos clave usando la herramienta PoseNet. Actualmente PoseNet está diseñado para detectar 17 puntos clave de una persona, y nos da de regreso las posiciones de cada punto clave junto con su nivel de probabilidad de que pertenezca a la parte del cuerpo estimada, el conjunto de puntos clave es conocido como pose y contiene una lista con los 17 puntos estimados por Posenet para hacer la inferencia con los esqueletos que constan en un corpus, con el que se entrena el modelo de nuestra propuesta.

Con los puntos clave obtenidos con PoseNet, se genera un esqueleto con la postura estimada de la persona que se encuentra en la imagen, como se puede apreciar en la

Figura 20, el proceso para la generación del corpus se describe más detalladamente en el algoritmo mostrado en la **Figura *21***, para la generación del corpus se han obtenido capturas con diferentes personas cuyas edades varían desde 10 años en adelante, con niños muy pequeños el modelo de PoseNet no detecta los puntos clave con precisión debido a su pequeña estatura y cercanía entre puntos. Este proceso se realizó con niños en condiciones normales, es decir que son capaces de prestar atención sin importar las distracciones del entorno y tienen facilidad en realizar las posturas propuestas, para luego junto con los terapeutas ayudar a los niños con parálisis cerebral a desarrollar los ejercicios propuestos, de tal forma que intenten hacerlo voluntariamente mejorando así su desempeño motriz y su capacidad de prestar atención a las tareas propuestas.



Puntos clave	Nombre
1	Nariz
2	Ojo Izquierdo
3	Ojo derecho
4	Oído Izquierdo
5	Oído derecho
6	Hombro Izquierdo
7	Hombro derecho
8	Codo Izquierdo
9	Codo derecho
10	Muñeca Izquierda
11	Muñeca derecha
12	Cadera Izquierda
13	Cadera derecha
14	Rodilla Izquierdo
15	Rodilla derecha
16	Tobillo derecho
18	Tobillo derecho

Figura 20 *Lista de puntos clave del esqueleto*

Debido a la inexistencia de un corpus que contenga las imágenes necesarias para detectar las posturas recomendadas, por los terapeutas del IPCA, con la ayuda de la herramienta PoseNet de TensorFlow, el cual nos entrega los puntos clave apreciados en la **Figura 20** se ha diseñado un algoritmo que en tiempo real obtiene un extenso banco de imágenes, generando un corpus con aproximadamente mil imágenes por cada una de las seis posturas.

El algoritmo diseñado para la obtención de dicho banco se puede apreciar en el diagrama de flujo de la **Figura *21*** en el cual se van a realizar capturas continuas de imágenes de

personas que se encuentran en las posturas deseadas, durante 1000 iteraciones por cada posición, variando los ángulos y posición de las personas que se encuentran en las imágenes obtenidas por la cámara del ordenador a través de OpenCV, que es una herramienta de Python, para poder trabajar con cámaras de video. Cada captura es procesada por PoseNet y del conjunto de puntos clave se obtiene un esqueleto por cada captura, todos estos esqueletos obtenidos son almacenados en una carpeta con el nombre de la posición.

Al finalizar las 1000 capturas se repite el proceso para cada una de las 6 posturas mencionadas en este trabajo. Cada esqueleto es guardado con un nombre conformado por el "número de iteración" + Nombre de la posición.

Luego de obtener todos los esqueletos, de las 6 clases, se obtendrán 6 carpetas, de las cuales se separarán en 2 conjuntos, para las fases de entrenamiento y validación, tomando 800 imágenes para el entrenamiento y 200 para validación de cada clase, es decir el entrenamiento se realizará con 4800 y la validación con 1200 imágenes.

Mientras que para la fase de prueba se realizara directamente con los niños del IPCA ya que ellos serán los usuarios finales.

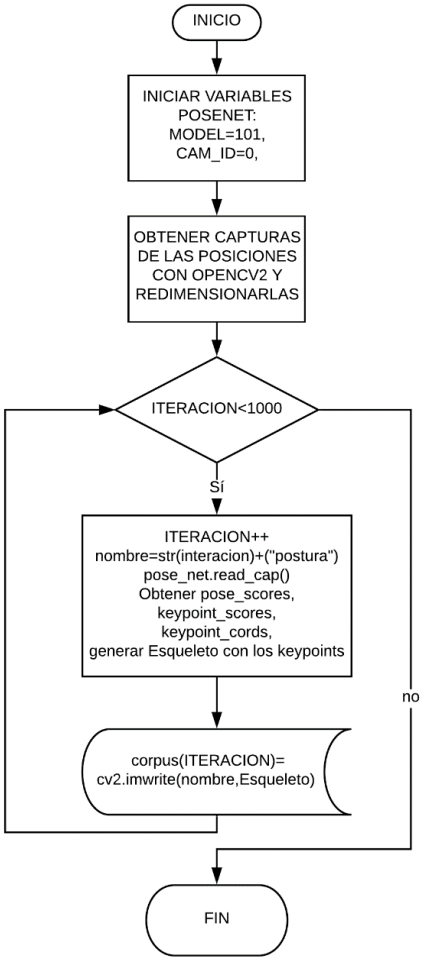


Figura 21 *Generación de corpus para cada postura*

En la **Figura 21** se tiene el diagrama de flujo correspondiente al proceso de esqueletización, donde inicialmente se carga el modelo preentrenado de PoseNet junto con sus pesos por defecto y se definen los parámetros de la cámara que va a obtener las capturas mediante la herramienta de OpenCV, estas imágenes son procesadas previamente ajustando sus dimensiones para que luego ingresen a un bucle que durante 1000 iteraciones adquiere puntos clave y nos entrega sus coordenadas y precisión de estimación de dichos puntos, para luego generar un esqueleto con los puntos adquiridos y guardarlo en una base de datos conformada de 6 clases, es decir que se tienen 1000 imágenes por cada una de las 6 posturas.

```python
def main():
    with tf.Session() as sess:
        model_cfg, model_outputs = posenet.load_model(args.model, sess)
        output_stride = model_cfg['output_stride']

        if args.file is not None:
            cap = cv2.VideoCapture(args.file)
        else:
            cap = cv2.VideoCapture(args.cam_id)
        cap.set(3, args.cam_width)
        cap.set(4, args.cam_height)

        start = time.time()
        frame_count = 0
        captura=0
        while True:
            captura=captura+1
            nombre=["corpus"+str(captura)+'Ipca1'+'.jpg']
            input_image, display_image, output_scale = posenet.read_cap(
                cap, scale_factor=args.scale_factor, output_stride=output_stride)

            heatmaps_result, offsets_result, displacement_fwd_result, displacement_bwd_result = sess.run(
                model_outputs,
                feed_dict={'image:0': input_image}
            )

            pose_scores, keypoint_scores, keypoint_coords = posenet.decode_multi.decode_multiple_poses(
                heatmaps_result.squeeze(axis=0),
                offsets_result.squeeze(axis=0),
                displacement_fwd_result.squeeze(axis=0),
                displacement_bwd_result.squeeze(axis=0),
                output_stride=output_stride,
                max_pose_detections=10,
                min_pose_score=0.15)

            keypoint_coords *= output_scale
```

Figura 22 *Algoritmo para la generación de corpus*

En la **Figura 22** se presenta una parte del algoritmo para la obtención del corpus correspondiente a las posturas, se puede apreciar el proceso donde se inicia el modelo de PoseNet para la estimación de puntos clave, luego con la cámara del computador, se capturan imágenes y se adquieren sus características como mapas de calor, desplazamientos etc. Y finalmente se genera un listado con los puntos clave de las personas existentes en la imagen.

2.2. Entrenamiento

Una vez generado el corpus se procede al entrenamiento del modelo de red convolucional para lo cual se dividen las imágenes en dos partes, train y test; se eligió un porcentaje de 80% para la proporción de imágenes de entrenamiento y 20% para las de prueba. Las imágenes son redimensionadas a un alto y ancho de WxHmientras más grandes sean estos valores, se necesitará una mayor cantidad de memoria de procesamiento, por lo cual se realizaron pruebas con dimensiones de 64x64 y 128x128, obteniendo mejores resultados con la segunda opción.

Teniendo en cuenta que las imágenes del corpus se forman a partir de líneas que unen los puntos clave de la postura, se facilita el procesamiento, y simplemente se procede a la carga de imágenes de donde se obtienen dos arreglos, x_{train} en el cual cada fila contiene una imagen de la base de datos y y_{train}el cual contiene las etiquetas correspondientes a cada imagen.

Los valores de cada píxel varían entre 0 y 255 en escala de grises, para el entrenamiento de la red es necesario que estos valores se encuentren normalizados por lo cual se divide para 255, y se obtienen valores entre cero y uno. Luego de procesar las imágenes para que puedan ingresar a la fase de entrenamiento, se debe tener en cuenta que el volumen de información es demasiado grande y no es posible cargar todas las imágenes, éstas se ingresan en proporciones o lotes(batch), se ha seleccionado un valor de batch de 150, mientras mayor sea el valor del batch se obtendrán mejores resultados, sin embargo se requiere un mayor procesamiento, esto se puede realizar con ordenadores que cuenten con aceleradores gráficos, GPU, que trabajen en paralelo con la CPU para disminuir su carga, en este caso solo se cuenta con el hardware descrito en la **Tabla *3***. Si se realiza el entrenamiento con un valor muy alto se generan errores correspondientes a la capacidad de memoria necesaria que requiere Tensorflow para poder llevar a cabo dicho entrenamiento.

El entrenamiento de la red convolucional, para obtener el modelo con los pesos y bias adecuados, hace uso de la API de Keras, cuya estructura central o Core, permite organizar las capas según el modelo que se pretende entrenar, el tipo más popular es el modelo de tipo secuencial.

Las capas del modelo convolucional se pueden apreciar en la siguiente tabla:

LAYER (TYPE)	OUTPUT	PARAM #
conv2d_1 (Conv2D)	(60, 60, 64)	1664
max_pooling2d_1 (MaxPooling2)	(28, 28, 64)	0
conv2d_2 (Conv2D)	(26, 26, 64)	36928
conv2d_3 (Conv2D)	(24, 24, 64)	0
average_pooling2d_1 (Average)	(11, 11, 64)	0
conv2d_4 (Conv2D)	(9, 9, 128)	71856
conv2d_5 (Conv2D)	(7, 7, 128)	147584

average_pooling2d_2 (Average)	(3, 3, 128)	0
flatten_1 (Flatten)	(1152)	0
dense_1 (Dense)	(1024)	1180672
dropout_1 (Dropout)	(1024)	0
dense_2 (Dense)	(1024)	1049600
dropout_2 (Dropout)	(1024)	0
dense_3 (Dense)	(6)	6150

Tabla 2 *Capas del modelo convolucional*

Para el entrenamiento de la red convolucional se realiza un procesamiento de las imágenes, de tal manera que puedan ingresar al modelo secuencial de entrenamiento.

Se definen los parámetros necesarios para las capas de convolución y Max pooling, tales como: clases, tamaño de lotes, numero de épocas, densidades y las métricas utilizadas para evaluar el entrenamiento del modelo. Finalmente, el modelo es compilado y se guarda en un archivo .JSON, mientras que los pesos se guardan con una extensión .h5 para que luego puedan se invocados desde otro algoritmo.

```
nb_classes = 6
targets = y_val.reshape(-1)
y_val = np.eye(nb_classes)[targets]

model = Sequential()

model.add(Conv2D(64, (5, 5), activation='relu', input_shape=(IMG_SIZE, IMG_SIZE,1)))
model.add(MaxPooling2D(pool_size=(5,5), strides=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

model.add(Flatten())

model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))

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

gen = ImageDataGenerator()
train_generator = gen.flow(x_train, y_train, batch_size=batch_size)
model.summary()

model.compile(loss='categorical_crossentropy'
    , optimizer=keras.optimizers.Adam()
    , metrics=['accuracy']
)

model.fit_generator(train_generator, steps_per_epoch=batch_size, epochs=epochs, callbacks=[tensorboard])

score = model.evaluate(x_val, y_val)
```

Figura 23 *Algoritmo para el entrenamiento*

En la **Figura *23*** se tiene la parte del algoritmo en la cual se definen los parámetros para el entrenamiento de la red neuronal convolucional, el modelo de entrenamiento es de tipo secuencial y ha sido configurado de la siguiente manera:

Las capas están organizadas como se puede apreciar en la **Tabla *2***, para determinar las pérdidas se elige una función "categorical_crossentrophy" la cual calcula la pérdida entre un tensor de salida y un tensor objetivo, para ello las clases deben estar ordenadas de forma categórica. Para la optimización se utiliza un algoritmo de gradiente estocástico ADAM, y finalmente se definen las métricas de evaluación del modelo en este caso "accuracy".

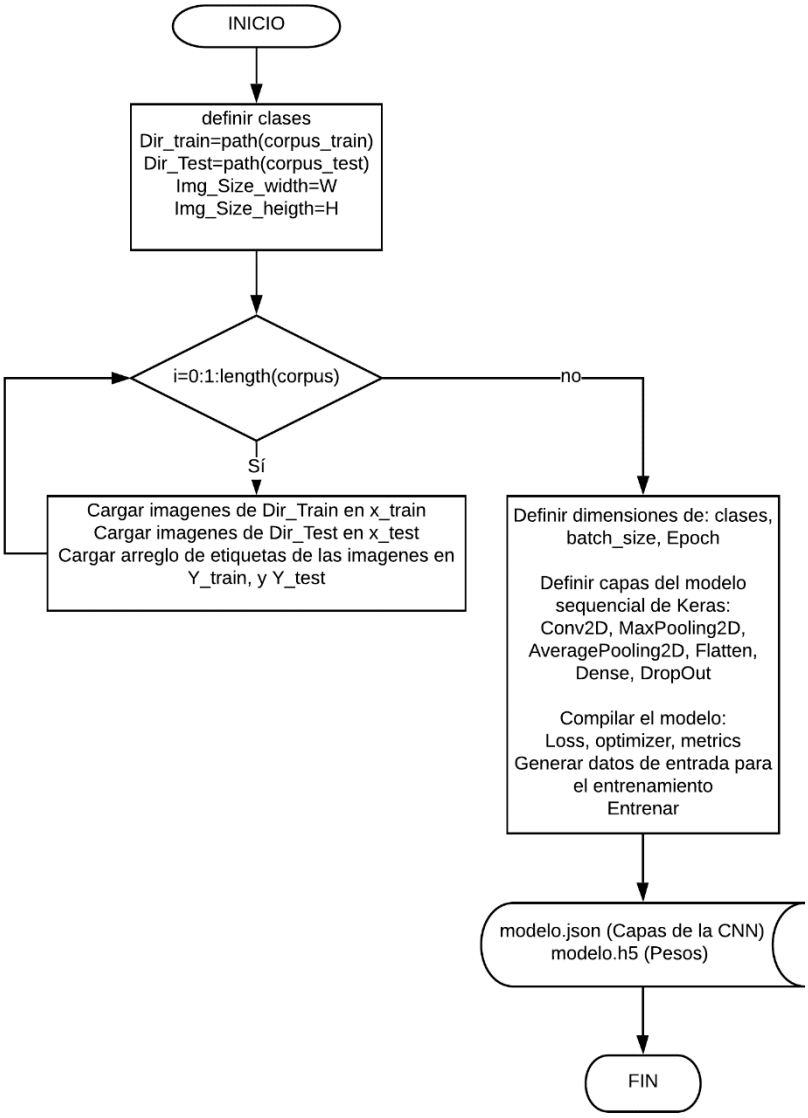


Figura 24 *Entrenamiento de la red CNN*

En la **Figura *24*** se tiene el diagrama correspondiente al entrenamiento de la red convolucional para la estimación de posturas, en el cual inicialmente se definen las clases (Posturas), la dirección o path en donde se encuentra la base de datos con las imágenes para el entrenamiento y las dimensiones de las imágenes. Luego, estas imágenes son procesadas para que puedan entrar a la fase de entrenamiento donde se definen los parámetros del modelo que va a ser entrenado, y finalmente se guardan sus

resultados en dos tipos de archivos, un modelo .JSON el cual contiene el modelo con sus capas y un archivo llamado modelo .H5 el cual contiene los pesos obtenidos luego del entrenamiento.

2.3. Aplicación

Finalmente, una vez entrenado el modelo, se procede a ejecutar la estimación, para lo que se obtienen capturas y se les da un procesamiento similar al que se realizó para obtener los esqueletos, luego se realiza una inferencia y el algoritmo se encarga de clasificar las imágenes.

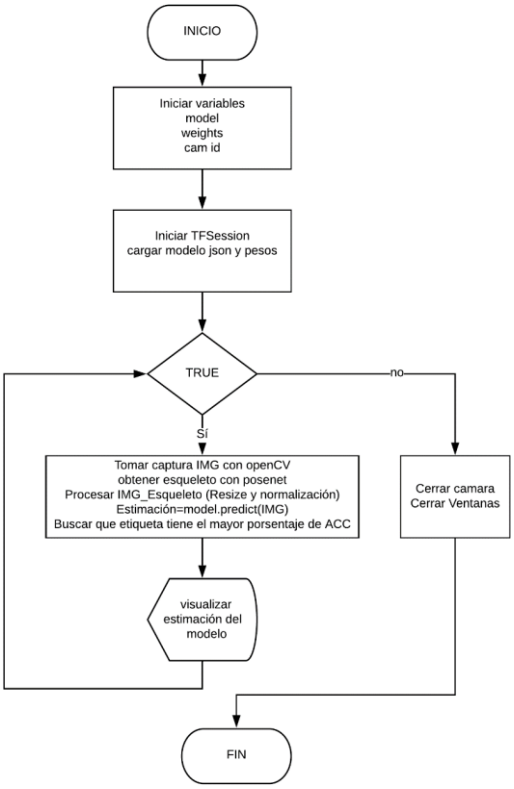


Figura 25 *Ejecución de la estimación con la red CNN entrenada*

En el diagrama de flujo presentado en la Figura 25 inicialmente se realiza un proceso similar al de esqueletización para adquirir un esqueleto de una persona en tiempo real con PoseNet, luego se carga el modelo y pesos obtenidos en el entrenamiento, y se inicia el bucle en el cual la cámara del ordenador va a tomar capturas continuamente. PoseNet se encarga de obtener el esqueleto y el modelo entrenado estima en que postura se

encuentra la persona presente en la imagen y entrega la que tenga mayor porcentaje de precisión, finalmente los resultados son presentados en una ventana que contiene la captura y la estimación de postura junto con el porcentaje.

```
config = tf.ConfigProto()
config.gpu_options.allow_growth=True
model = model_from_json(open("modelo.json", "r").read())
model.load_weights('modelo.h5') #load weights

posicion_detectada = cv2.resize(gray, (IMG_SIZE, IMG_SIZE))
img_pixels = image.img_to_array(posicion_detectada)
img_pixels = np.expand_dims(img_pixels, axis = 0)
predictions = model.predict(img_pixels)
max_index = np.argmax(predictions[0])
emotion = labels[max_index]
#print(emotion, np.max(predictions[0])*100)
#print(predictions)
#print(emotion)
text = "{}: {:.3f}".format(emotion, np.max(predictions[0])*100)
cv2.putText(overlay_image, text, (20, 20), cv2.FONT_HERSHEY_SIMPLEX,1, (0,255,0), 2)
overlay_image=cv2.resize(overlay_image,(600, 600))
cv2.imshow('Reconocimiento CNN-OPENCV',overlay_image)
```

Figura 26 *Algoritmo para la estimación de posturas*

Finalizado el entrenamiento se puede comprobar su funcionamiento con el algoritmo presentado en la **Figura *26***, donde luego de cargar el modelo y los pesos, se obtienen las estimaciones de la red CNN entrenada y se presenta la imagen adquirida por la cámara junto con el porcentaje de la estimación correspondiente a la persona en la imagen.

3. CAPITULO 3: EXPERIMENTACION Y RESULTADOS.

3.1. Experimentación.

Se llevaron a cabo varias pruebas con 5 diferentes niños los cuales tienen las siguientes condiciones especiales:

1. Multidiscapacidad
2. Parálisis cerebral infantil
3. Deficiencia cognitiva
4. Deficiencia sensorial
5. Hiperactividad
6. Discapacidad degenerativa

Para la fase de prueba del modelo, se realizó en el IPCA, con ayuda de los terapeutas y profesores de los niños, para cada una de las 6 posturas, la obtención de 8 estimaciones por cada niño, de donde se obtienen los resultados presentados a continuación.

Para la postura de caballero se obtuvieron las estimaciones presentadas en la Figura 27 donde se puede apreciar que existe un buen porcentaje de estimación con un promedio de 99.75% de precisión.

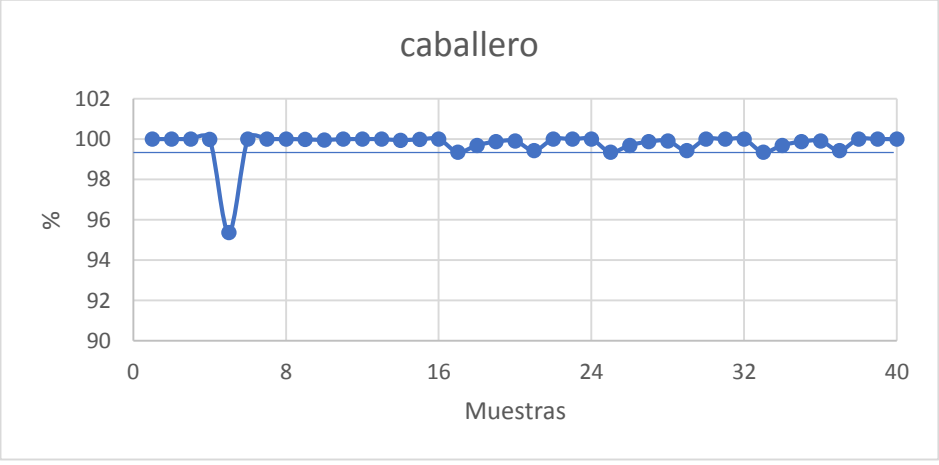


Figura 27 *Estimaciones obtenidas para la postura caballero*

En la **Figura 28**, se muestra el resultado de estimar la postura de Caballero, el sujeto de prueba presenta algo de dolor en las articulaciones de su cadera al realizar esta postura, sin embargo, lo realiza correctamente y la red CNN estima una similitud de $\hat{y}_i = 99.97\%$ con la posición real en la que se encuentra.

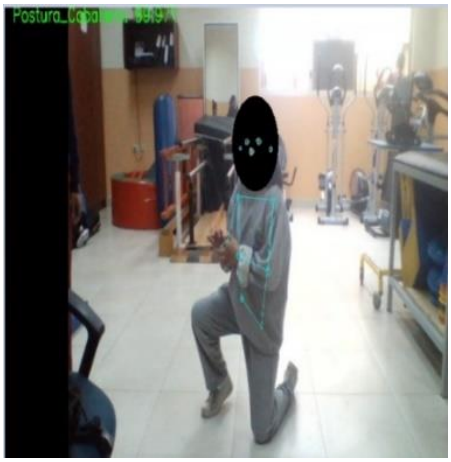


Figura 28 *Postura de caballero*

Para la postura de inhibición, se obtuvieron las estimaciones presentadas en la ***Figura 29***, en este caso se tienen algunas equivocaciones en las estimaciones debido a que se confunde con otras posturas, esto se debe a la dificultad que tienen los niños al flexionar sus extremidades y en ocasiones no logran completar la postura, el promedio de las estimaciones es de 91.88%.

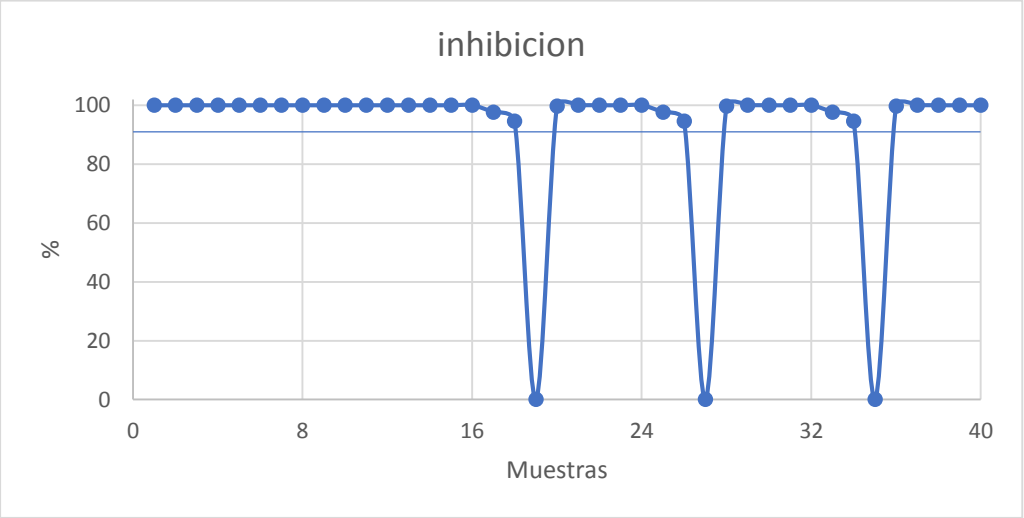


Figura 29 *Estimaciones obtenidas para la postura inhibición*

En la ***Figura 30***, el sujeto de prueba realiza la postura de Inhibición, en ocasiones la CNN, se confunde y nos devuelve una estimación errónea de la postura original, como se puede observar en la imagen de la derecha, donde aproxima la postura Inhibición a postura de caballero, esto se puede mejorar ampliando la base de datos de esta postura.

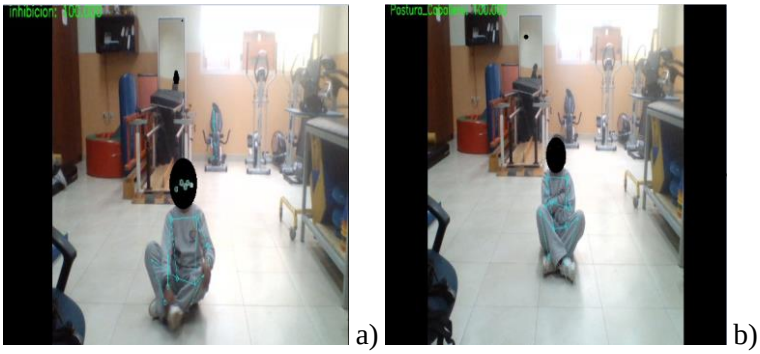


Figura 30 *Postura inhibición*

En la postura reposo de pie las estimaciones son casi perfectas como se puede apreciar en la ***Figura 31*** con un promedio de 100% en las estimaciones obtenidas.

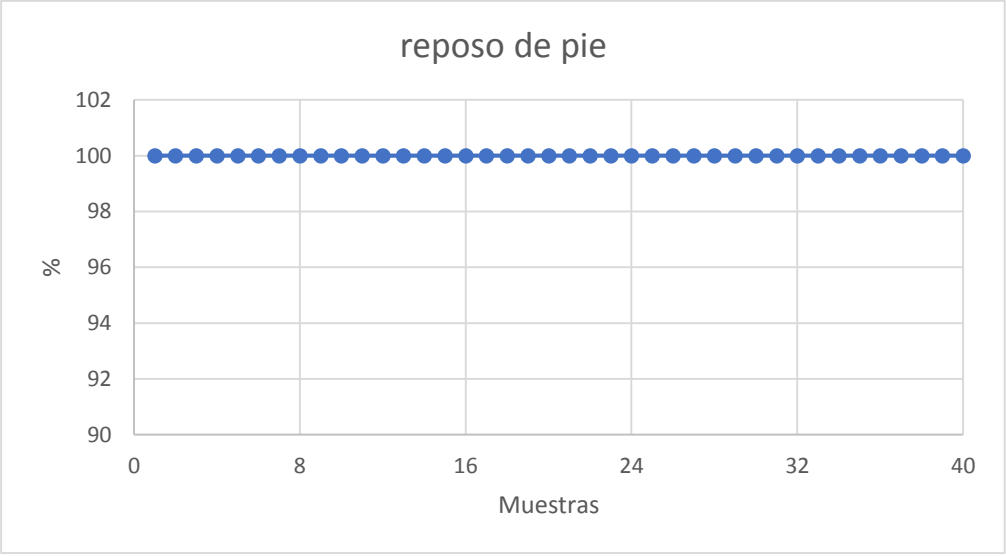


Figura 31 *Estimaciones obtenidas para la postura reposo de pie*

En la siguiente imagen el sujeto de prueba realiza la postura reposo de pie, en este caso no puede permanecer en esta posición por mucho tiempo, debido a la presencia de dolor en la planta de sus pies y las articulaciones de sus tobillos, por lo que siempre trata de estar en movimiento y necesita la ayuda de su terapeuta, quien le enseña cómo realizar la posición de forma correcta. Se puede apreciar como el algoritmo detecta la existencia de varias personas en la imagen por lo que nos devuelve dos sets de "punto clave", de los cuales se selecciona el primero en ser detectado para estimar su postura.

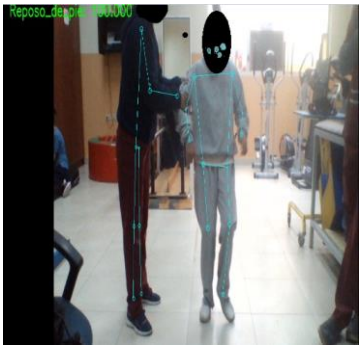


Figura 32 *Postura reposo de pie*

Al realizar pruebas con la postura de sentado es necesario el uso de una silla, al realizar pruebas no se generan dificultades, obteniendo los resultados que se pueden apreciar en la ***Figura 33*** con un promedio de 99.61%.

Figura 33 *Estimaciones obtenidas para la postura sentado*

En la ***Figura 34*** se realiza la postura Sentado, inicialmente con el modelo que fue entrenado con una dimensión de las imágenes de 64x64 tendía a confundir las estimaciones con las posturas de caballero e inhibición, al entrenarlo con una dimensión de 128x128 se obtuvo una mejora significativa en las estimaciones.

Figura 34 *Postura sentado*

Para la postura de gateo se tiene un promedio de 98.62% las estimaciones obtenidas se pueden apreciar en la ***Figura 35***.

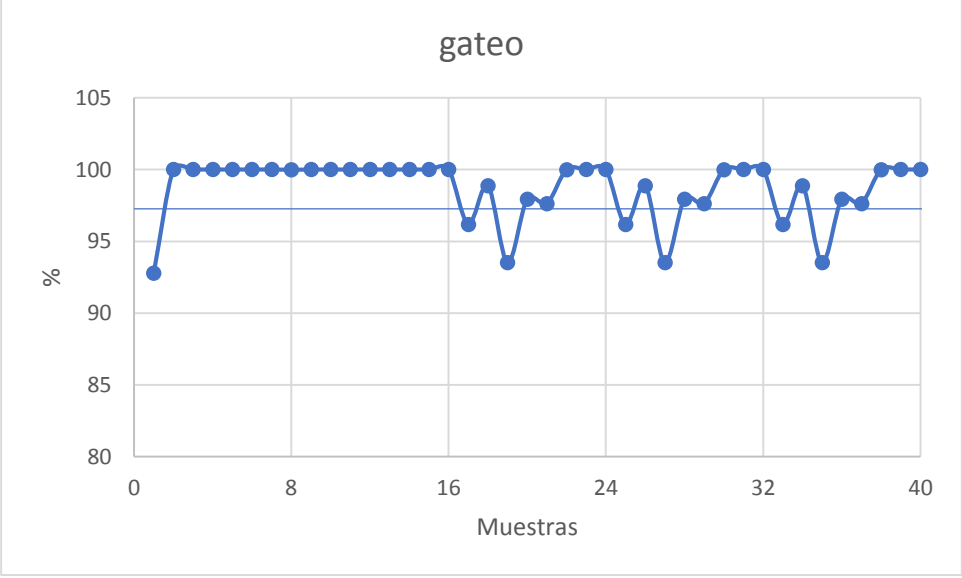


Figura 35 *Estimaciones obtenidas para la postura gateo*

En la ***Figura 36*** se tiene un niño realizando la postura de gateo, no se presentan dificultades más que distracción en ocasiones, y la mayoría de las muestras son aceptables.

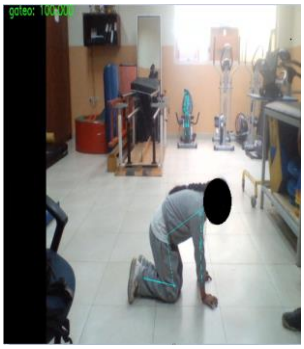


Figura 36 *Postura gateo*

Por último, se presentan los resultados de las estimaciones obtenidas al realizar pruebas con la postura Levantar Peso, todas las predicciones son acertadas, pero el porcentaje varía mucho, esto se debe a que esta postura consiste en una secuencia de movimientos en donde inicialmente los niños se inclinan al piso para recoger un objeto y luego lo levantan hasta llegar a una postura reposo de pie, se obtiene un porcentaje promedio de 89.04%.

Figura 37 *Postura levantar peso*

En la ***Figura 38*** se puede apreciar un niño ejecutando la postura levantar peso, la estimación para esta muestra es del 100% de similitud.

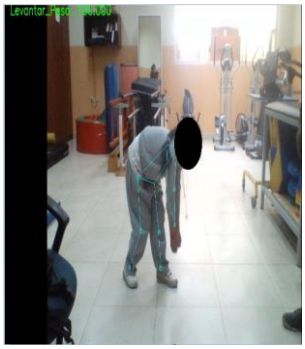


Figura 38 *Postura Levantar Peso*

3.2. Resultados.

PoseNet de TensorFlow facilita el procesamiento de las imágenes para la obtención de las posturas, de esta manera se evita la necesidad de utilizar una GPU, o una TPU los cuales pueden llegar a requerir de una gran inversión económica. Si bien es cierto aceleran los procesos y mejoran el rendimiento de los algoritmos, al trabajar en conjunto con la CPU, sin embargo, en este caso no es necesario una gran cantidad de procesamiento por lo que se utiliza simplemente un ordenador con las siguientes características:

Procesador	Intel(R) Core (TM) i7-4720HQ CPU 2.60GHz
Memoria RAM	8GB
Disco Duro	1 TB
Sistema operativo	Windows 10 Home Edition

Tabla 3 *Descripción del Hardware*

A pesar de no contar con una tarjeta gráfica, la CPU por si misma es capaz de procesar instrucciones al usar AVX2 (Advanced Vector Extensions), que se trata de un conjunto de instrucciones de 256 bits, el cual provee nuevas características e instrucciones proporcionando un esquema de compilación que trabaja con un vector de extensión SIMD de 256 bits para conseguir paralelismo a nivel de datos y poder trabajar con operaciones de punto flotante intensivo. Mejorando el rendimiento en las nuevas aplicaciones, y algunas existentes, mediante el manejo de paquetes de datos vectoriales

más grandes, y el uso de más hilos y núcleos del procesador, (Lemmetti, 2016, September)

Finalizado el entrenamiento, la API de Keras, nos permite obtener un registro de los valores de Exactitud (Accuracy) y valores de perdida (Val Loss) en épocas sucesivas. Con la herramienta de TensorBoard se obtienen los resultados correspondientes a la exactitud y las pérdidas del modelo convolucional durante el entrenamiento, las gráficas obtenidas se pueden apreciar en Figura 39 y Figura 40

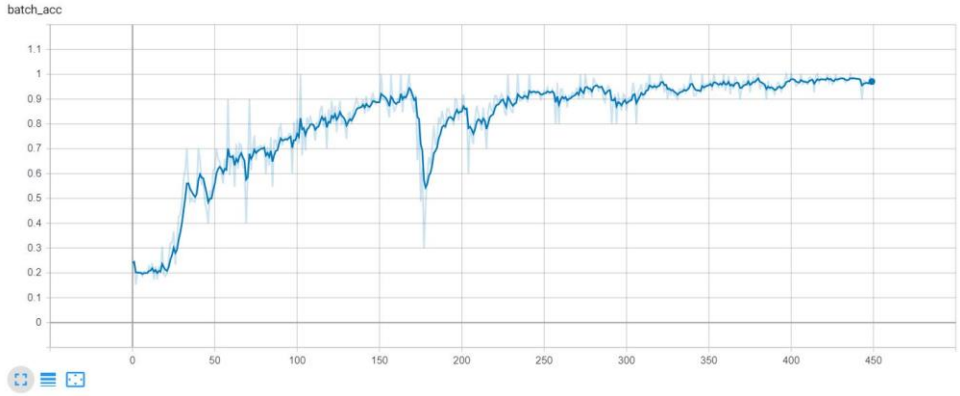


Figura 39 *Exactitud del modelo convolucional*

En la ***Figura 39*** se puede observar los resultados de precisión del entrenamiento por lotes llevado a cabo, donde aproximadamente en la muestra numero 350 los pesos ya se han ajustado y el modelo es capaz de clasificar las imágenes en las categorías propuestas.

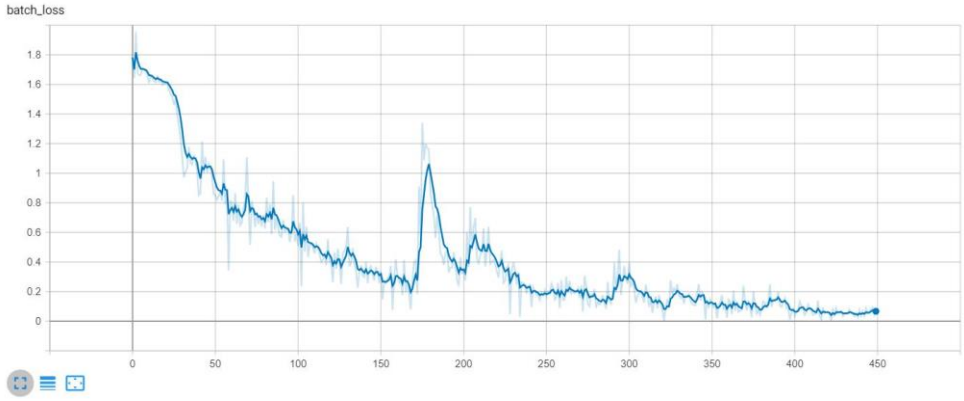


Figura 40 *Pérdidas durante el entrenamiento.*

En la ***Figura 40*** se presenta la gráfica correspondiente a las perdidas, cuanto menor sea la perdida, mejor será el modelo, a diferencia de la precisión, la pérdida no es un porcentaje, sino que más bien representa un resumen de los errores cometidos para cada muestra en conjuntos de entrenamiento.

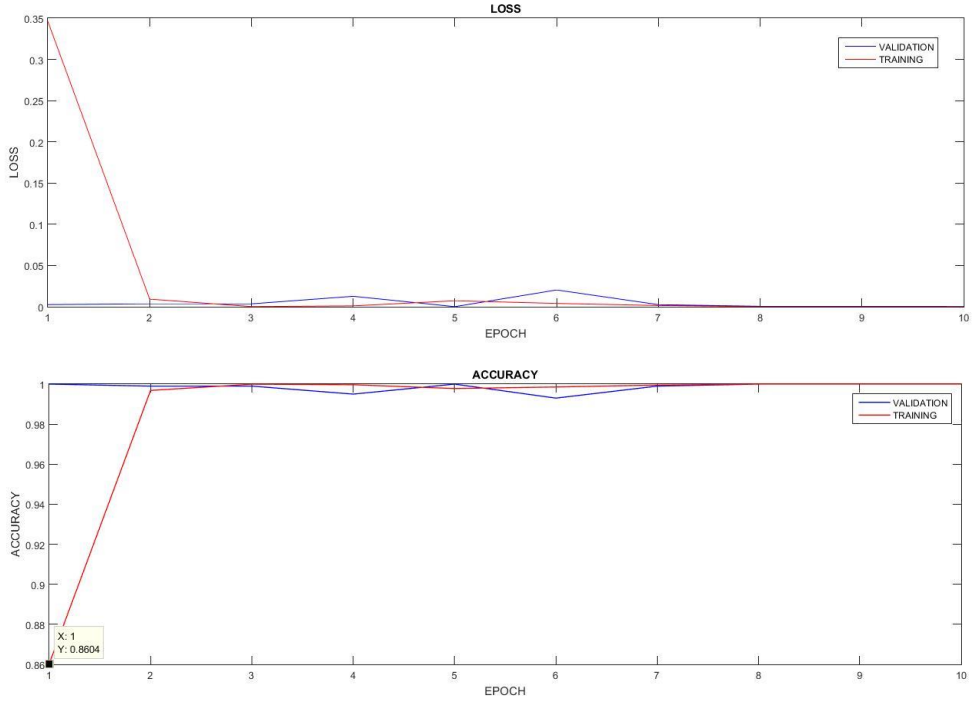


Figura 41 *Perdidas y exactitud en la fase de entrenamiento y validación*

En la ***Figura 41*** se observan las pérdidas y exactitud frente a las épocas, tanto en entrenamiento como en la fase de validación, con una base de datos de 6000 imágenes, donde se muestra que el punto ideal es en la época 3, ya que en dicho punto el modelo ya converge.

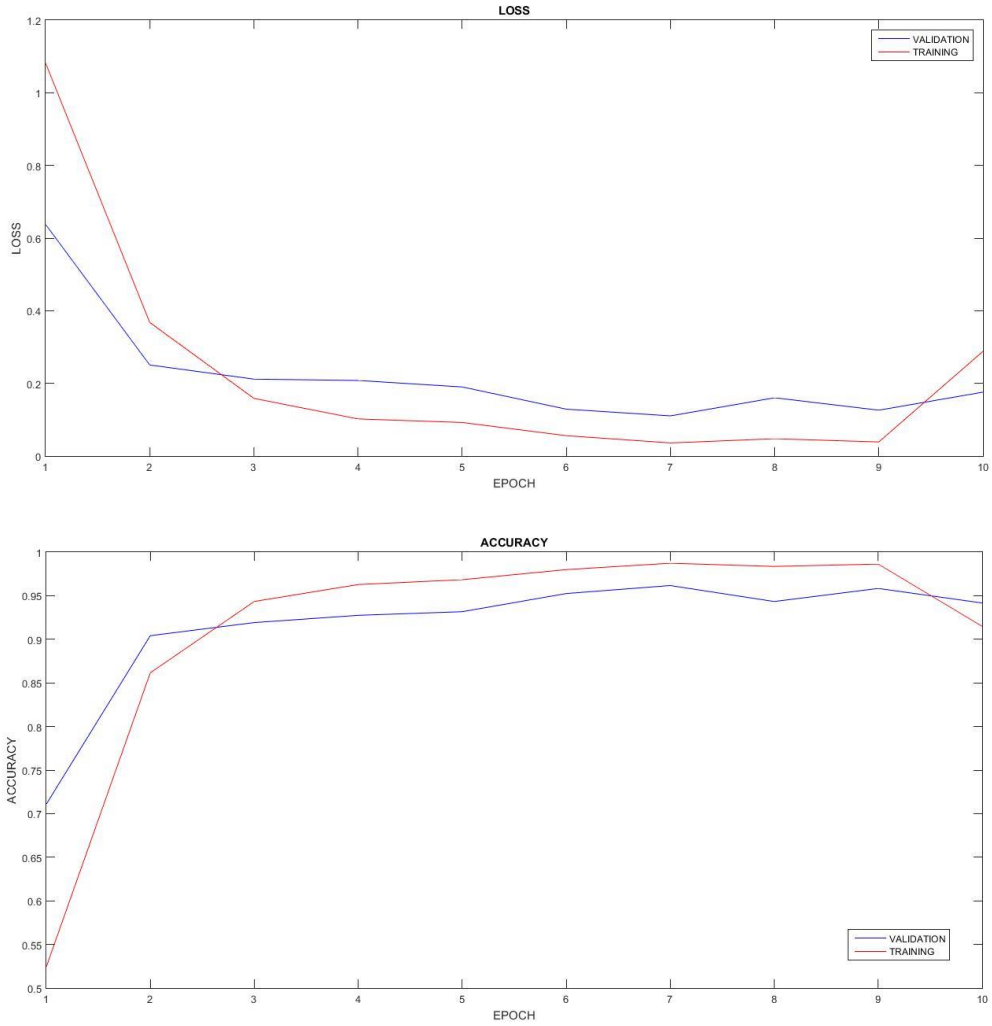


Figura 42 Perdidas y exactitud en la fase de entrenamiento y validación

En la ***Figura 42*** se observan las pérdidas y exactitud frente a las épocas, tanto en entrenamiento como en la fase de validación, con una base de datos de 3000 imágenes, donde se muestra que el punto ideal es en la época 7, ya que en dicho punto el modelo ya converge.

Al ajustar los parámetros de las imágenes para el entrenamiento de la red CNN tanto de dimensiones de largo y ancho de las imágenes como tamaño de lote, varían significativamente los resultados del modelo como se puede ver en la siguiente tabla:

Dimensión WxH	Batch_Size	Acc %	Test Loss
64x64	300	86.956	0.5152
	150	75.386	0.625
128x128	300	99.99	0.1500
	150	92.950	0.3500

Tabla 4 *Resultados del entrenamiento con diferentes parámetros*

Con dimensiones de 64x64 y tamaño del lote de 300 se obtiene una precisión de 87% aproximadamente, sin embargo, al realizar pruebas en tiempo real el modelo tiende a confundir las posturas Sentado, Caballero, Gateo e inhibición. Al aumentar las dimensiones de las imágenes, mejoran los resultados llegando a una precisión de 92.95% con imágenes de 128x128 y un tamaño del lote de 150 donde el modelo puede clasificar perfectamente las imágenes.

En el caso de aumentar demasiado el tamaño del lote como se puede apreciar en la tabla anterior el modelo se sobreentrena, es decir, para las imágenes del set de entrenamiento las estimaciones mejoran llegando a una precisión del 99% mientras que para muestras nuevas diferentes a las del entrenamiento va empeorando la estimación de postura deseada.

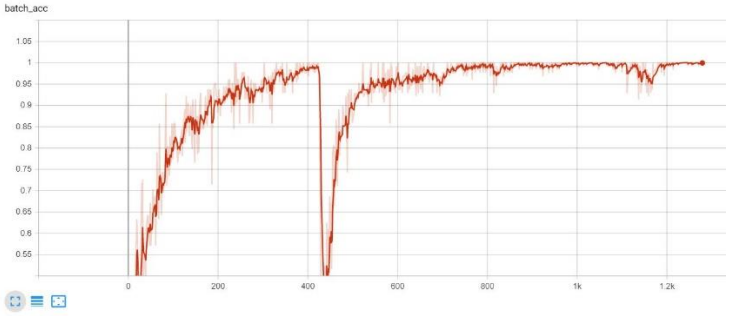


Figura 43 *Precisión obtenida con un tamaño de lote de 300 (Overfitting)*

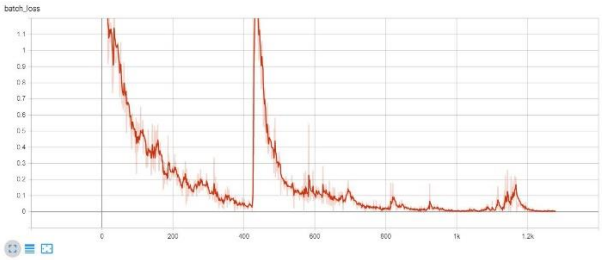


Figura 44 *Perdida obtenida con un tamaño de lote de 300 (Overfitting)*

3.2.1. Matriz de confusión.

Para la obtención de la matriz de las pruebas realizadas con los 5 niños, donde cada uno realiza las posturas indicadas por el terapeuta, y el modelo estima cual es la postura estimada del niño se obtiene la siguiente tabla:

a) De pie
b) Caballero
c) Gateo
d) Inhibición
e) Alzar peso
f) Sentado

	a					b					c					d					e					f				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
a	✓	✓	✓	✓	✓																									
b						✓	✓		✓	✓																✓				
c											✓	✓	✓	✓	✓															
d								✓	✓							✓	✓													✓
e																					✓	✓	✓	✓	✓					
f													✓								✓						✓		✓	✓

Tabla 5 *Matriz de confusión del modelo convolucional*

Las filas de la matriz representan los valores reales de cada medición, mientras que en las columnas se tienen los valores estimados por el clasificador, se puede ver como para las posturas De Pie, Gateo y Levantar Peso existen mayor cantidad de t_p, para las posturas Caballero y sentado este valor disminuye y el peor caso es para la postura de Inhibición la cual se confunde con las posturas caballero y sentado; esto se

debe a la similitud existente entre estas tres posturas, sin embargo el error es mínimo y se tiene una probabilidad aceptable de clasificación de las posturas.

Representando la matriz con valores numéricos se puede apreciar de mejor manera la cuantización de los valores correspondientes a t_p, f_p, f_n

	De Pie	Caballero	Gateo	Inhibición	Levantar Peso	Sentado
De pie	5	0	0	0	0	0
Caballero	0	4	0	0	0	1
Gateo	0	0	5	0	0	0
Inhibición	0	2	0	2	0	1
Levantar Peso	0	0	0	0	5	0
Sentado	0	0	1	0	1	3

Tabla 6 *Matriz de confusión representada numéricamente*

3.2.2. Precisión, Recall, F1-Score

Con estos valores se obtienen los resultados de las métricas Precisión, Recall, F1-Score, con un soporte de 5 unidades por clase.

	PRECISIÓN	**RECALL**	**F1-SCORE**	**SUPPORT**
DE PIE	1	1	1	5
CABALLERO	0.67	0.80	0.73	5
GATEO	0.83	1	0.91	5
INHIBICIÓN	1	0.40	0.57	5
LEVANTAR PESO	0.83	1	0.91	5
SENTADO	0.60	0.60	0.60	5

Tabla 7 *Resultados de las métricas para evaluar el modelo*

En la ***Tabla 7*** se puede apreciar la capacidad del módulo de clasificar las imágenes por cada clase, tanto los valores de precisión, recall y F1-Score; son aceptables, excepto para la postura inhibición donde el algoritmo confunde dicha postura con sentado o postura de caballero.

Para la postura Inhibicion como se puede apreciar en la ***Tabla 7*** , a pesar de tener una precisión de 1 su valor de Recall es muy pequeño (0.40), esto se debe a la similitud entre las posturas, y la dificultad que representa para los niños que no están acostumbrados a doblar sus articulaciones tanto de las rodillas como de las caderas.

Finalizadas las pruebas de validación, se lleva a cabo el desarrollo de una interfaz gráfica cuyo objetivo es motivar a los niños a mejorar su desempeño tanto en destreza física, de aprendizaje y cognitiva. La interfaz fue creada con la herramienta TK-inter de Python.

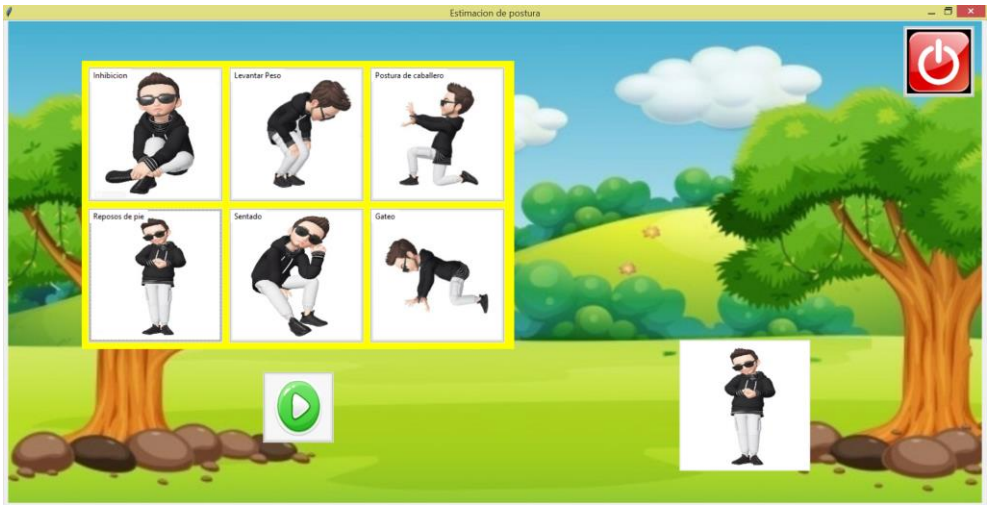


Figura 45 *Interfaz grafico del módulo de aprendizaje*

La reacción de los niños al utilizar el módulo es positiva, ya que les llama mucho la atención al observar la interfaz de la aplicación y les motiva a realizar las actividades organizadas por el profesor o terapeuta que se encuentra trabajando con los niños.

Al momento de realizar las actividades se debe tener en cuenta que los niños deben tener la capacidad de realizar los movimientos, tengan facilidad de desplazar sus extremidades, y puedan corregir su postura de manera voluntaria, además de su habilidad para prestar atención a las órdenes del terapeuta que está trabajando con ellos y que no se distraigan fácilmente por las condiciones del entorno en el que se encuentran.

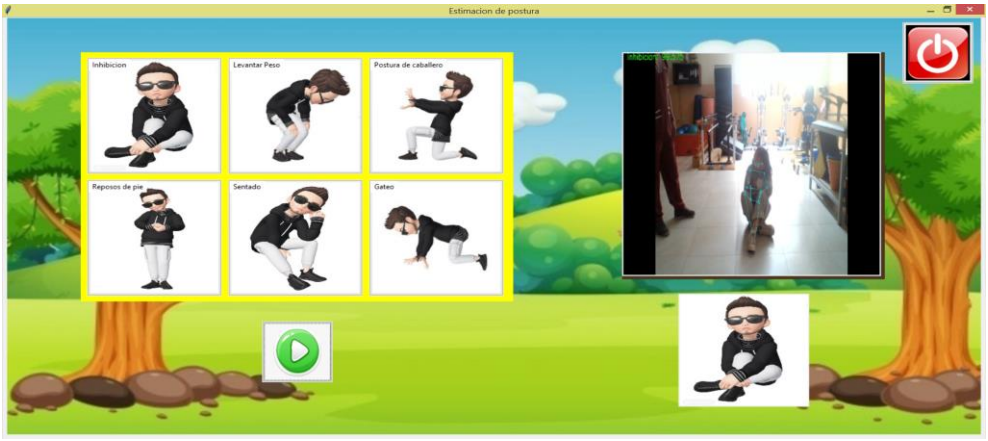


Figura 46 *Resultado del módulo al realizar postura de inhibición*

Se presentan dificultades al momento de doblar mucho las articulaciones de las extremidades, sobre todo en las rodillas y caderas, ya que algunos niños no están acostumbrados a realizar estas posiciones y sus articulaciones no les permiten rotar sus extremidades de manera normal, sin embargo, intentan realizarlo voluntariamente hasta que el programa les muestre que lo han realizado de manera correcta.

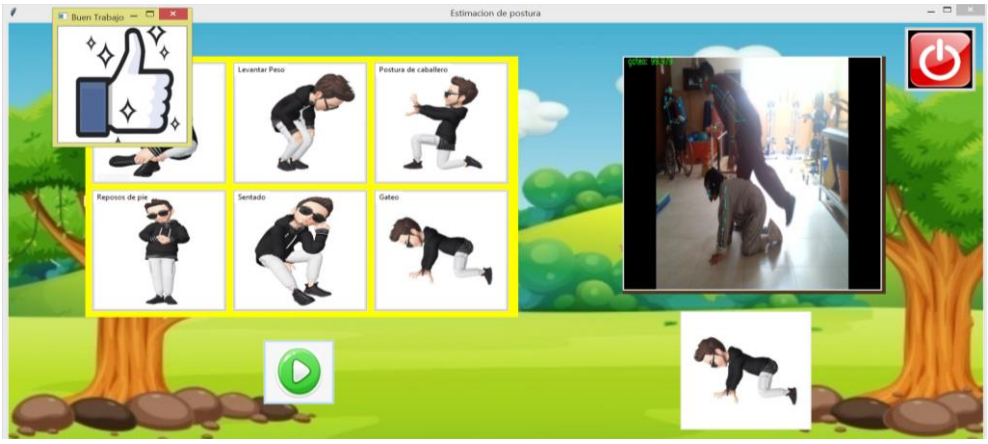


Figura 47 *Resultado del módulo al realizar postura de gateo*

En algunas ocasiones, es necesaria la ayuda del docente para realizar esta postura debido a que los niños pierden la atención y empiezan a moverse alrededor del entorno.

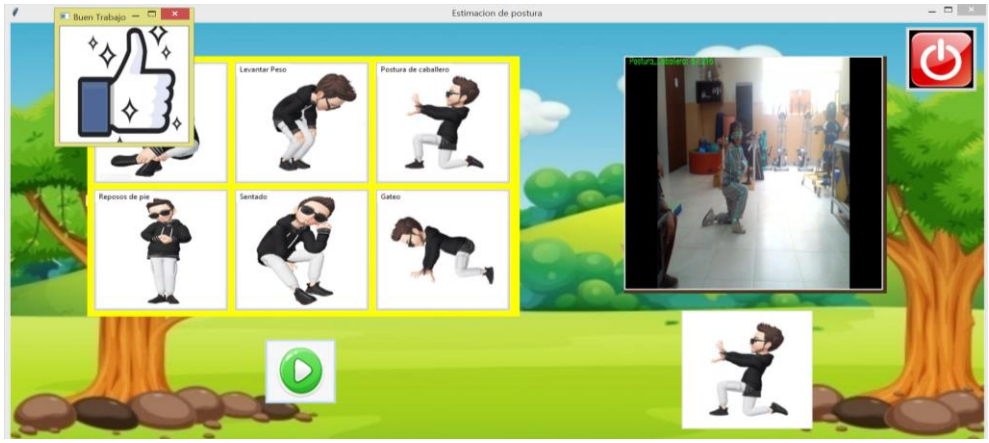


Figura 48 *Resultado del módulo al realizar postura de caballero*

La postura caballero, no representa mucha dificultad para los niños que pueden desplazar sus extremidades; pero para niños que no tienen mucha movilidad en sus articulaciones les toma un poco más de tiempo realizarla.

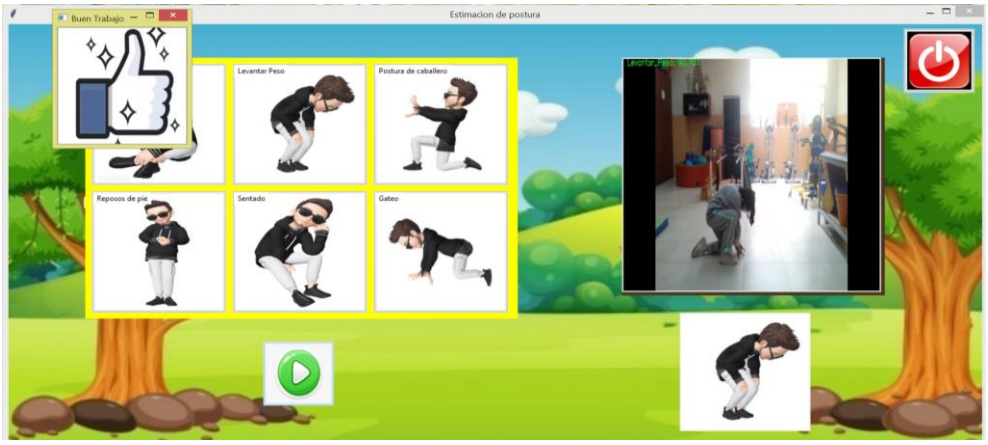


Figura 49 *Resultado del módulo al realizar postura de levantar peso*

La posición de levantar peso, conlleva una serie de movimientos que exigen a los niños forzar sus articulaciones y músculos de sus extremidades inferiores, por lo cual en ocasiones se les hace difícil mantener el equilibrio de su cuerpo.

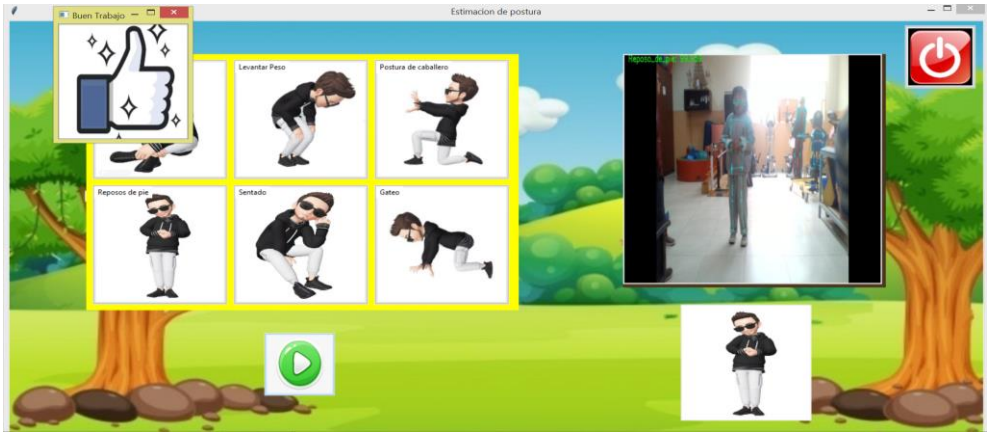


Figura 50 *Resultado del módulo al realizar postura Reposo de pie*

En reposo de pie, para la mayoría de los niños no presenta dificultad, sin embargo, en algunos no se logra mantener la posición correcta debido a que generan demasiada tensión ya sea en la planta del pie, o de su talón con lo cual aumenta la probabilidad de sufrir lesiones y les obliga a mantenerse en movimiento.

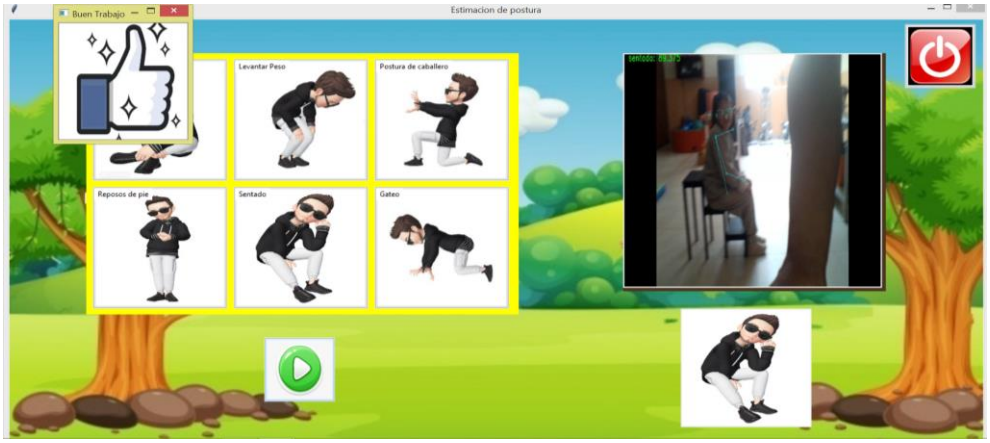


Figura 51 *Resultado del módulo al realizar postura sentado*

Para finalizar, se presenta la postura sentado donde los niños se encuentran en estado de relajación, todos intentan realizar las posturas voluntariamente facilitando de esta manera el trabajo del terapeuta o profesor que se encuentra trabajando con ellos.

4. CAPITULO 4: CONCLUSIONES Y RECOMENDACIONES:

- Los resultados al realizar entrenamientos con imágenes de dimensiones de ancho y alto de 64x64 con un tamaño de lote de 300 nos entregan una perdida $Test\_Loss = 0.5152$ y una precisión de $Acc = 86.956\%$, que al parecer son buenas; pero al realizar la evaluación en tiempo real se confunde demasiado. Al aumentar las dimensiones a 128x128 y disminuir el tamaño de lote a 150 se mejoran los resultados, obteniendo una pérdida de $Test\_Loss = 0.38$ y una precisión de $Acc = 92.35$, de esta forma el modelo es capaz de estimar perfectamente todas las posturas al realizar pruebas en un ambiente controlado.
- El modelo de estimación de postura de PoseNet, tiene una gran precisión en sus estimaciones. Además de no verse afectado por factores como distancia, iluminación, entorno en el que se encuentra etc. Solo basta que los puntos clave de la persona sean visibles en la imagen, para detectar fácilmente las coordenadas en las que se encuentran dichos puntos y obtener una estimación de una postura especifica.
- La precisión global del modelo convolucional obtenido es del 80% al realizar las respectivas mediciones con las métricas mencionadas en el capítulo 3, donde solo una de las seis posturas presentó mayor confusión para el modelo, esto se debe a la similitud entre las posturas y puede ser mejorado ampliando el corpus de imágenes de dichas posturas y seleccionando de manera más exhaustiva cada una de estas imágenes.
- Al utilizar el método propuesto en este documento, se disminuye el costo computacional gracias al uso de modelos pre-entrenados los cuales son desarrollados a partir de corpus gigantes como Coco dataset, Mnist, Caltech101 entre otras, por lo cual se ahorra la inversión en equipos que pueden llegar a ser demasiado costosos, de esta forma con un ordenador normal sin tarjeta gráfica se puede llevar a cabo el entrenamiento de la red CNN con la ayuda de AVX2 el cual permite ejecutar procesos en paralelo sin necesidad de una tarjeta gráfica.
- El uso del módulo presentado en este trabajo es de gran ayuda al momento de trabajar con niños que pueden desplazar sus extremidades de forma voluntaria, aumenta el interés en ellos al realizar las actividades propuestas por los terapeutas y profesores del IPCA y les motiva a seguir mejorando en su rehabilitación física.
- Para la estimación de un numero de posturas mayor al presentado en este trabajo se recomendaría usar equipos que cuenten con una tarjeta gráfica la cual acelerara los procesos al trabajar en paralelo con el CPU.

Trabajos futuros:

- El modelo propuesto puede ser reentrenado con un mayor número de posturas para poder trabajar con diferentes tipos de personas y ayudarlos en su rehabilitación,
- También se podría implementar como herramienta para trabajar con adultos mayores y detectar si tienen algún tipo de deformidad en su estructura ósea que les impida realizar sus actividades cotidianas.

BIBLIOGRAFIA

Fundación Wikimedia, Inc. (16 de Octubre de 2019). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/TensorFlow

Arista-Jalife, A., Calderón-Azua, G., Fierro-Radilla, A., & Nakano, M. (2017). Clasificación de Imágenes Urbanas Aéreas: Comparación. *Información Tecnológica*, 213,214.

Belagiannis, V., & Zisserman, A. (2017). Recurrent Human Pose Estimation. *12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017).* Washington.

Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., & Slobodan, L. (2016). 3D Pictorial Structures Revisited: Multiple Human Pose Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Benito Gorron, D. (2018). *Detección de voz y música en un corpus a gran escala de eventos de audio.* Madrid.

Bièvre, D. (2009). The 2007 International Vocabulary of Metrology (VIM). *Clinical biochemistry*, 43-49.

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., & Black, M. J. (2016). Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. *ECCV*, 1-18.

Bonenfant, M., Laurendeau, D., Fortin, A., Cardou, P., Gosselin, C., Faure, C., . . . Bouyer, L. (2017). A Computer Vision System for Virtual Rehabilitation. *Conference: 2017 14th Conference on Computer and Robot Vision (CRV).* Edmonton.

Cao, Z., Tomas, S., Shih-En, W., & Yaser, S. (2017). Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *CVPR 2017*, (págs. 7291-7299). Honolulu.

COCO. (5 de 11 de 2019). *Coco dataset*. Obtenido de Coco dataset: http://cocodataset.org/#home

Damle, R., Gurjar, A., Joshi, A., & Nagre, K. (2015). Human Body Skeleton Detection And Tracking. *International Journal of Technical Research and Application*, 222-225.

Dushyant, M., Helge, R., Dan , C., Pascal, F., Oleksandr, S., Weipeng, X., & Theobalt, C. (2017). Monocular 3D Human Pose Estimation In The Wild. *International Conference on 3D Vision.* Qingdao.

Flach, P. &. (2015). Precision-recall-gain curves: PR analysis done right. . *In Advances in neural information processing systems* , 838-846.

Gamino del Rio, I., & Sanchez, J. (2018). *Detector de elementos para el videojuego Dark Souls.* Madrid.

Gimenez Palomarez, F., Monsoriu, J., & Alemany-Martinez, E. (2016). Aplicacion de la convolucion de matrices al filtrado de imagenes. *Modelling in Science Education and Learning*, 97-108.

Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Hartwig, A. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv preprint*.

Inc., A. (10 de 9 de 2019). *Anaconda*. Obtenido de Anaconda Documentation : https://docs.anaconda.com/

IPCA. (12 de 10 de 2019). *Instituto de paralisis cerebral del Azuay IPCA*. Obtenido de Instituto de paralisis cerebral del Azuay IPCA: http://ipca.catedraunescoinclusion.org/nosotros/

Lemmetti, A. K. (2016, September). AVX2-optimized Kvazaar HEVC intra encoder. . *In 2016 IEEE International Conference on Image Processing (ICIP).*, (pp. 549-553).

Malik, J., Elhayek, A., & Stricker, D. (2017). Simultaneous Hand Pose and Skeleton Bone-Lengths Estimation from a Single Depth Image. *3DV-2017.* Qingdao.

Noren, A. (16 de Octubre de 2019). *Sitiobigdata.com*. Obtenido de http://sitiobigdata.com/2019/06/22/relu-funciones-activacion/#

OMS. (2013). Disability, Induding Prevention, Management and Rehabilitation. *Fifty-Eight WHO Assembly.*

Papandreou, G. Z. (2018). Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. . *In Proceedings of the European Conference on Computer*, 7-9.

Pfister, T., Charles, J., & Zisserman, A. (2015). Flowing ConvNets for Human Pose Estimation in Videos. *2015 IEEE International Conference on Computer Vision (ICCV).* Santiago.

Phyton. (10 de 9 de 2019). *Phyton*. Obtenido de About Phyton: https://www.python.org/about/

Song, S., Lan, C., Xing, J., Zeng, W., & Liu, J. (2017). An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, (pp. 4263-4270). San Francisco.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. . *Remote sensing of Environment*, 77-89.

Tompson, J., Jain, A., LeCun, Y., & Bregler, C. (2014). Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. *CVPR*.

Toshev, A., & Szegedy, C. (2014). DeepPose: Human Pose Estimation via Deep Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition.* Columbus.

Valle Barrio, A. (2018). *Aplicacion de Tensorflow en Deep Learning.* Madrid.

Villena, V., Fuster, A., Saval, M., & Azorin, J. (2017). An Iterative Method for 3D Body Registration Using a Single RGB-D Sensor. *International Journal of Computer Vision and Image Processing*, 26-39.

Wang, Z., Guoliang, L., & Guohui, T. (2017). Human Skeleton Tracking Using Information Weighted Consensus Filter in Distributed Camera Networks. *Chinese Automation Congress (CAC).* Jinan.

Xiaolong, L., Pengyuan, L., Xiang, B., & MIng-MIng, C. (2017). Fusing Image and Segmentation Cues for Skeleton Extraction in the Wild. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*.

Xiaoqiang, L., Zhang, Y., & Liao, D. (2017). Mining Key Skeleton Poses with Latent SVM for Action Recognition. *Hindawi Publishing Corporation*.

Xiu, Y., Li, J., Wang, H., Fang, Y., & Lu, C. (2018). Pose Flow: Efficient Online Pose Tracking. *British Machine Vision Conference (BMVC).* Newcastle.

Yang, W., Ouyang, W., Wang, X., Ren, J., Li, H., & Wang, X. (2018). 3D Human Pose Estimation in the Wild by Adversarial Learning. *CVF*, 5255-5264.

UNIVERSIDAD POLITECNICA SALESIANA
UNIDAD DE POSGRADOS

MAESTRIA EN CONTROL Y AUTOMATIZACION INDUSTRIALES

Autor: Lenin German Aguilar Sigüenza

Dirigido por: Vladimir Espartaco Robles Bykbaev

DISEÑO Y DESARROLLO DE UN MÓDULO PARA DETERMINAR LA POSTURA HUMANA EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL Y RECONOCIMIENTO DE PATRONES COMO HERRAMIENTA DE SOPORTE EN EL DESARROLLO DE LA MOTRICIDAD GRUESA DE NIÑOS CON DISCAPACIDAD.

Este trabajo presenta los resultados de exactitud y precisión de un módulo que realiza la estimación de la postura humana, empleando un algoritmo basado en redes neuronales con técnicas de visión artificial, el cual infiere la postura de los usuarios con la finalidad de usarlo como una herramienta para el aprendizaje de niños con parálisis cerebral. Este algoritmo realiza un procesamiento de un banco de aproximadamente mil imágenes por cada postura, las cuales pasan por un proceso de normalización, aplanamiento y segmentación.