UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE CAÍDAS EN EL ADULTO MAYOR MEDIANTE VISIÓN ARTIFICIAL UTILIZANDO REDES NEURONALES (DEEP LEARNING)

AUTOR: ORBE CISNEROS, EDWIN ALEXANDER

DIRECTOR: PhD. VARGAS VALLEJO, VANESSA CAROLINA

SANGOLQUÍ, AGOSTO 2022

Tabla de Contenidos

01 Introducción

Antecedentes, Motivación, Justificación, Alcance y Objetivos del Proyecto

Antecedentes

OMS

684 000 Muertes debido a caídas

Personas Mayores y sus problemas de salud

INEC

37.4% encuestados sufrieron caídas

Motivación del proyecto

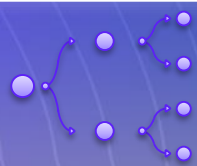
DISMINUIR LAS CONSECUENCIAS DE CAÍDAS EN PERSONAS ADULTAS SOLAS AL ALERTAR INMEDIATAMENTE DE UN POSIBLE SUCESO UTILIZANDO DISPOSITIVOS Y TECNOLOGÍA ACCESIBLE A BAJO COSTO

Tecnologías con sensores

Información relacionada con el movimiento y entorno del ser humano

Visión artificial

El uso de algoritmos de análisis estadístico en imágenes



Aprendizaje Profundo

Redes neuronales artificiales

Justificación e Importancia

Caídas en adultos mayores

Detección de caídas

Identificación de escenas con eventos de caídas

Sistema de visión artificial

Acontecimiento frecuente y riesgoso

Minimizar el nivel de riesgo

Análisis de imágenes

Solventar dificultades técnicas y económicas

Alcance del proyecto

Entrenamiento de la CNN

Detección de caídas en tiempo real mediante cámara

Notificación de alerta de caída mediante mensaje vía WhatsApp

Sistema de detección de caídas en adultos mayores

Objetivos

Objetivo General

Diseñar e implementar un **sistema de detección de caídas** en el adulto mayor mediante visión artificial utilizando redes convolucionales (Deep Learning)

Objetivos específicos

Seleccionar el entorno de desarrollo, el lenguaje de programación, **el set de datos** y las librerías para desarrollar el sistema de detección de caídas

Configurar un **sistema de visión artificial** utilizando deep learning basado en **redes convolucionales**

Seleccionar e implementar **algoritmos** para la detección de caídas en adultos mayores.

Realizar la **prueba** de concepto del Sistema mediante fotogramas en **tiempo real**

02 Fundamentos Teóricos

Procesamiento de imágenes, visión artificial y Deep learning

Inteligencia Artificial vs Machine Learning vs Deep Learning

INTELIGENCIA ARTIFICIAL

Imitar el comportamiento humano por máquinas o computadores

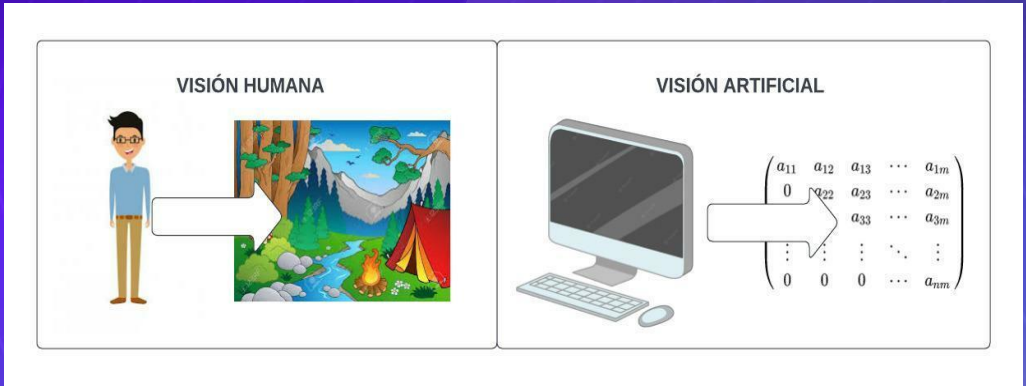
MACHINE LEARNING

Algoritmos para enseñar al computador como reconocer un objeto, una orden, un evento y como actuar frente a ello

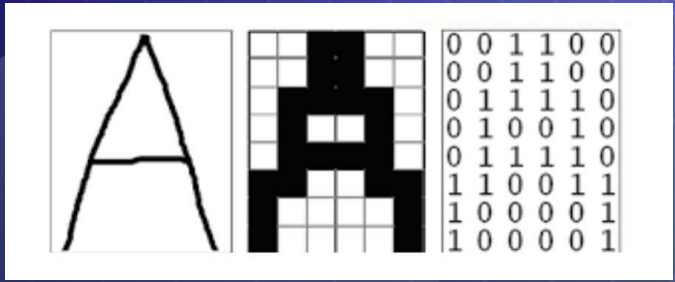
DEEP LEARNING

Algoritmos basados en redes neuronales a los que se les entrega suficiente información con la finalidad de que aprenda y reconozca por sí solo un evento.

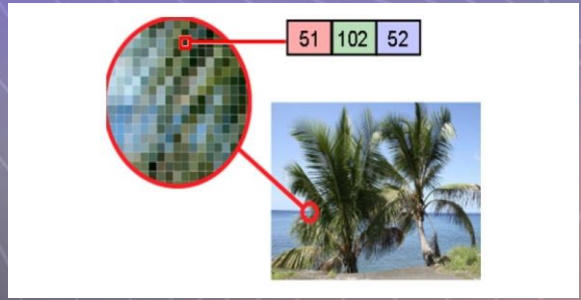
Visión humana vs Visión artificial



Interpretación de imágenes en Visión artificial



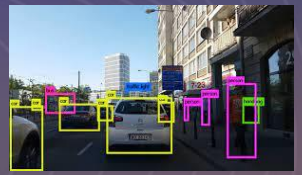
Interpretación imagen blanco/negro



Interpretación imagen color

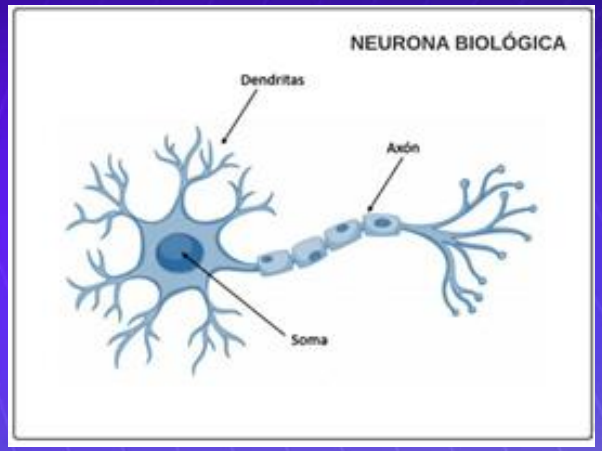
Etapas de un sistema de visión artificial

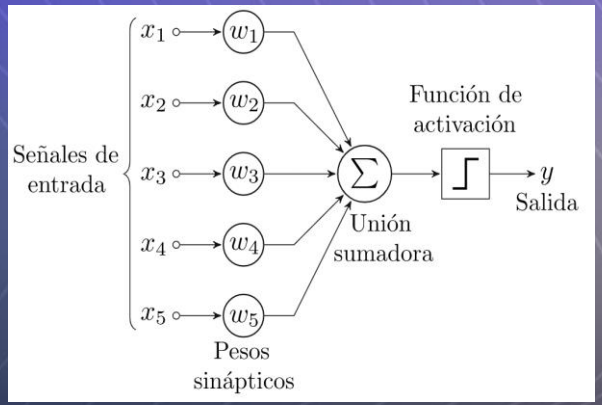
Escenario a analizar → Adquisición y digitalización → Procesamiento previo → Obtención de características → Reconocimiento e interpretación de información → Aplicación

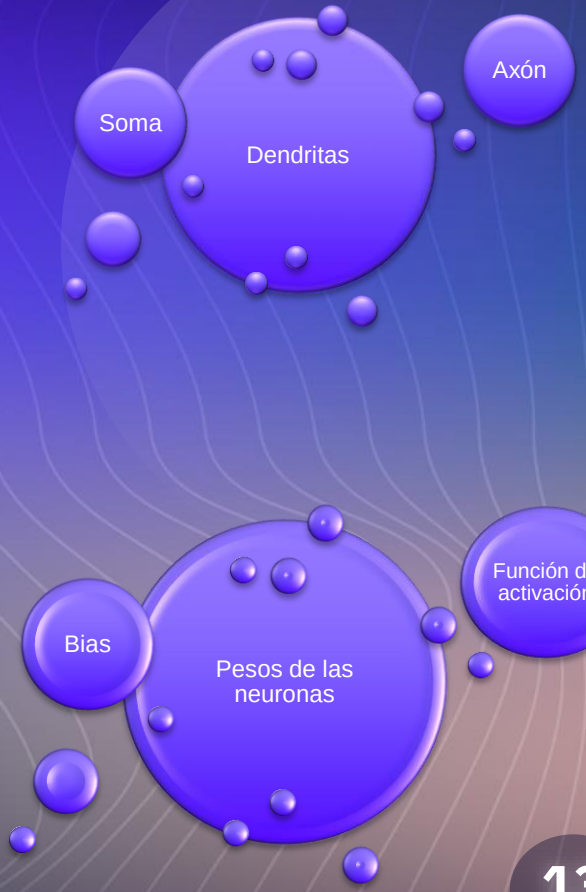


DEEP LEARNING

Redes Neuronales Artificiales





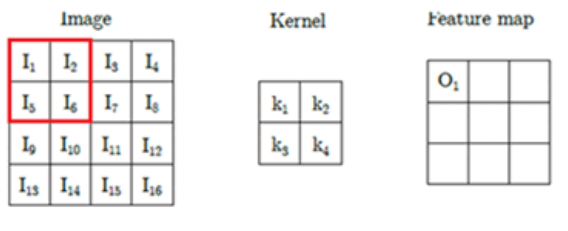


Redes Neuronales convolucionales - CNN

Capa Convolucional

Núcleo de CNN

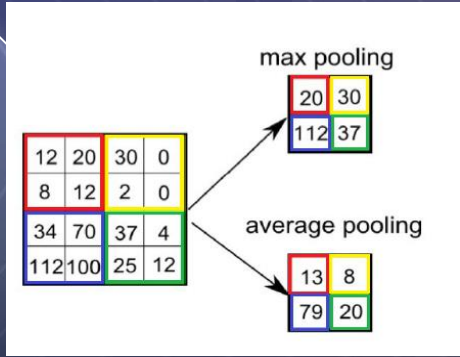
Operación Convolución → extracción de características



Capa de Pooling

Reducir dimensionalidad

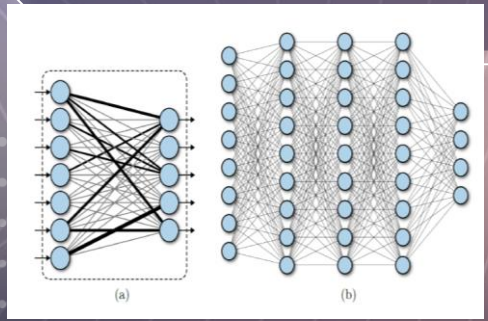
Max Pooling
Average Pooling

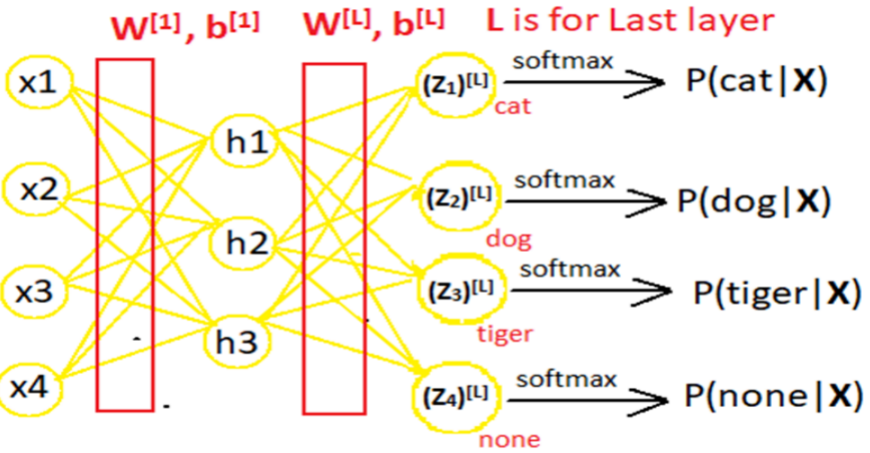


Capa Fully Connected

Información entre capa a capa

Propagación de información entre capas permite reducir errores





Capas de entrada

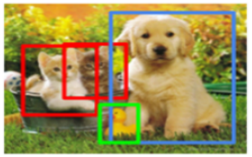
Recolección de datos

Capas ocultas

Extracción de características

Capas de salida

Clasificación y detección de datos



GATO , PERRO , PATO

Arquitectura de una CNN

Set de datos para entrenamiento

03 Desarrollo del sistema

Plataforma de desarrollo, transferencia de aprendizaje, entrenamiento y predicciones de la red neuronal

Plataforma de desarrollo

Hardware

- AMD Ryzen 7 4800H con Radeon Graphics 2.90 GHz.
- RAM 12 GB.
- Windows 11 Home Single Language

Computador portátil HP Pavilion Gaming

- NVIDIA GeForce GTX 1650 Ti
- Memoria total disponible para gráficos 9914 MB
- Memoria de vídeo dedicada 4096 MB

Unidad de procesamiento gráfico GPU

- Integrada al computador

Cámara

Software

- Etiquetado de imágenes

LabelImg

- Suite de código abierto

Anaconda

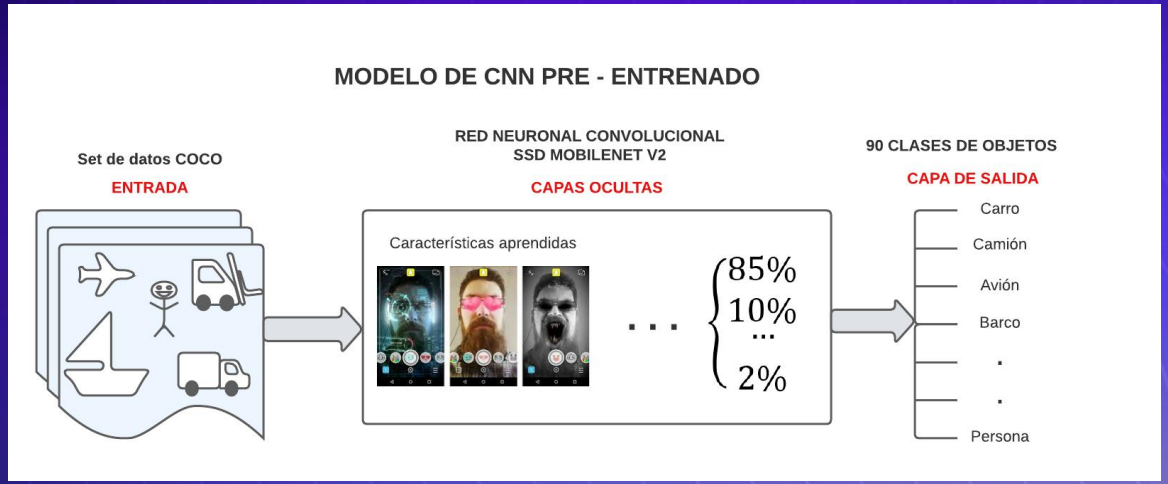
- Entorno de desarrollo de código abierto

Jupyter Notebook

- Herramienta para configurar modelos de CNN

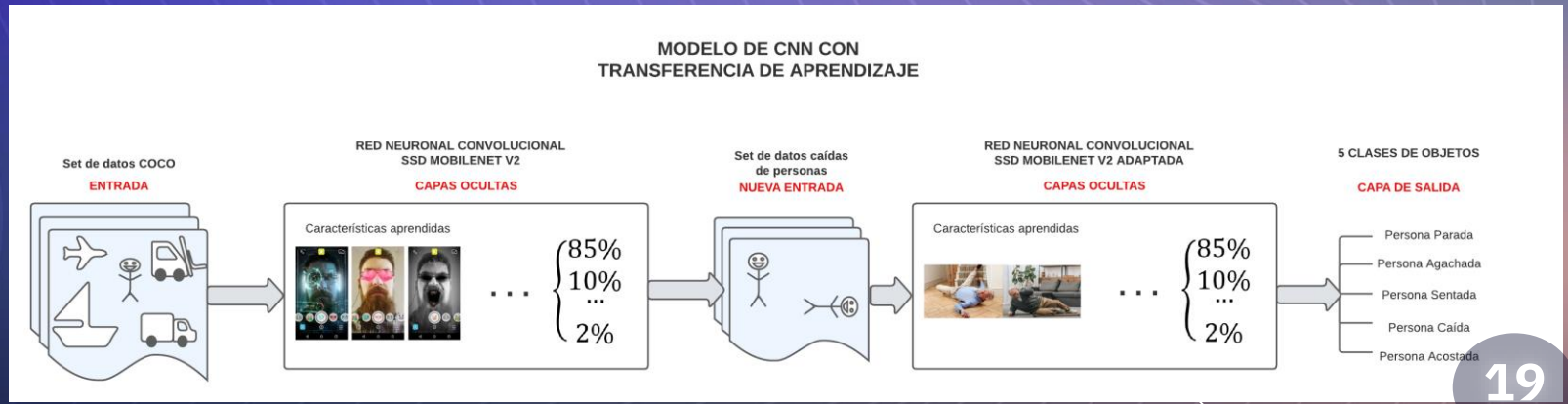
API de detección de objetos de Tensorflow

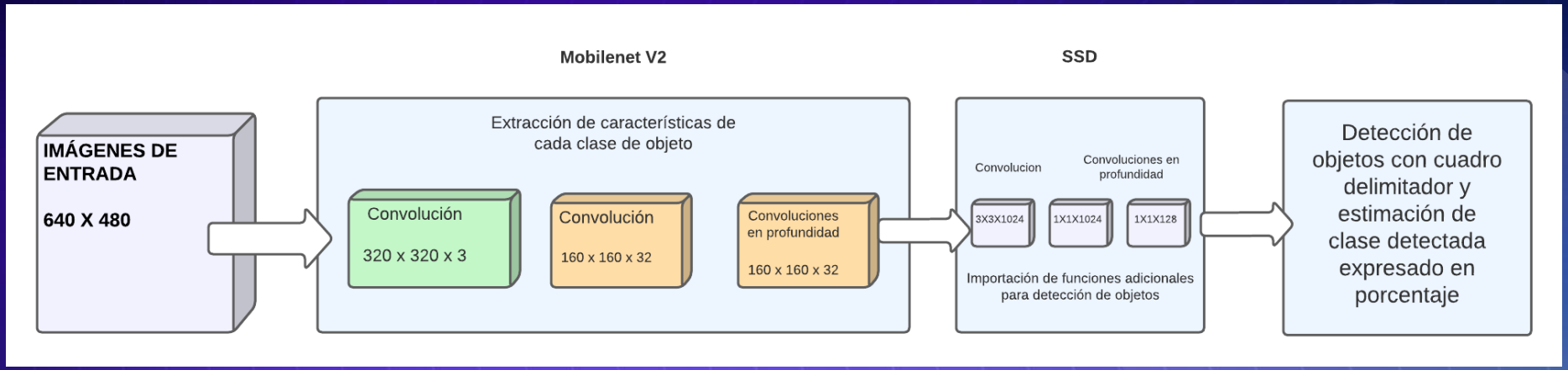
Pre – entrenamiento de la CNN con transferencia de aprendizaje



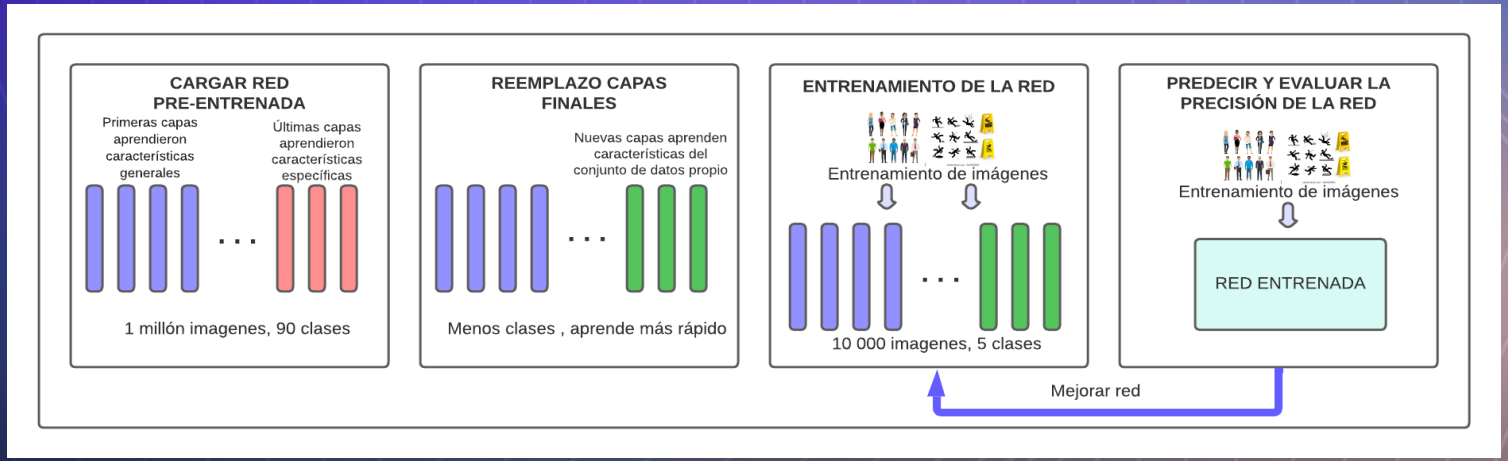
Entrenamiento desde cero

Entrenamiento con transferencia de aprendizaje



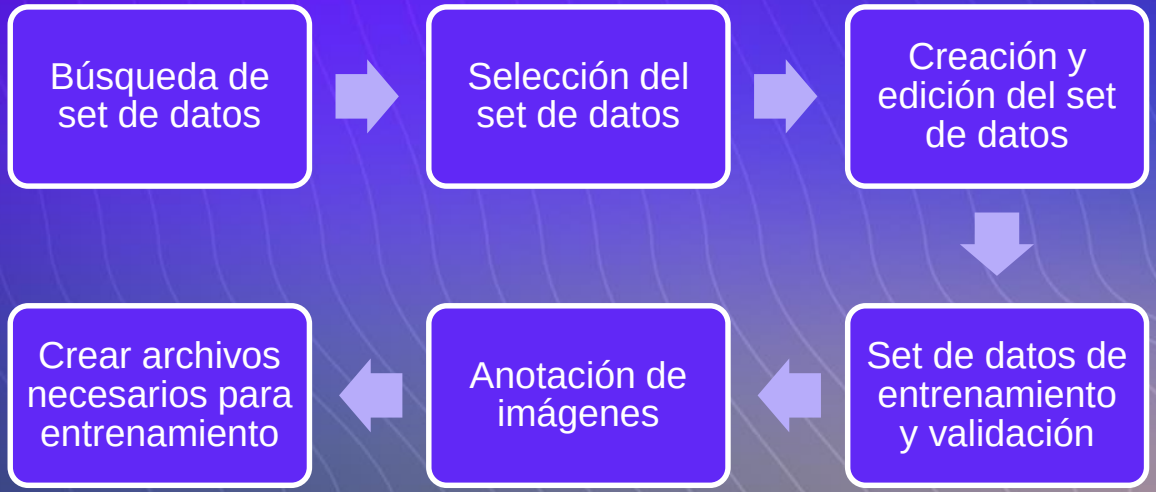


ARQUITECTURA DE LA CNN SSD MOBILENET V2



METODOLOGÍA DE TRANSFERENCIA DE APRENDIZAJE APLICADA

Metodología para el desarrollo del set de datos



Etiquetado de imágenes

LabelImg

```
<annotation>
	<folder>Alexander Dataset</folder>
	<filename>142.png</filename>
	<path>D:\alexo\ALEXO ESPE\TESIS\Programas y Dataset\labelImg\data\Alexander Dataset\142.png</path>
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>640</width>
		<height>480</height>
		<depth>3</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>Persona Agachada</name>
		<pose>Unspecified</pose>
		<truncated>1</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>181</xmin>
			<ymin>111</ymin>
			<xmax>519</xmax>
			<ymax>480</ymax>
		</bndbox>
	</object>
</annotation>
```

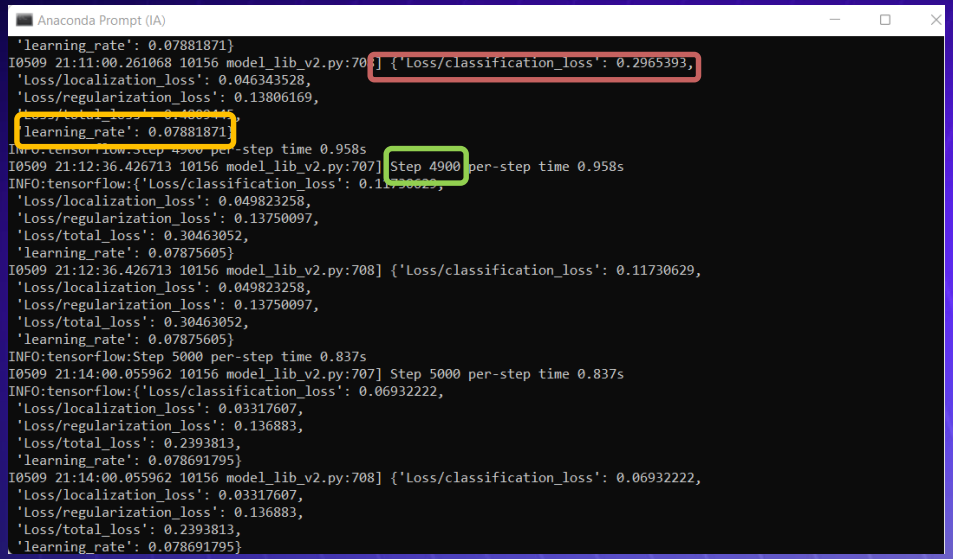
Dimensiones de la imagen

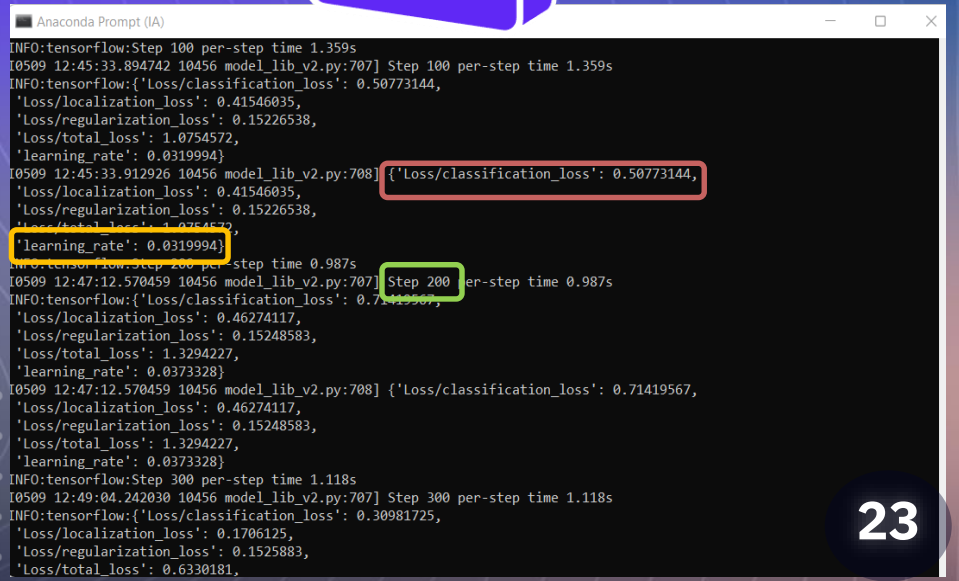
Instancia de clase

Ubicación del objeto

Entrenamiento

Parámetros



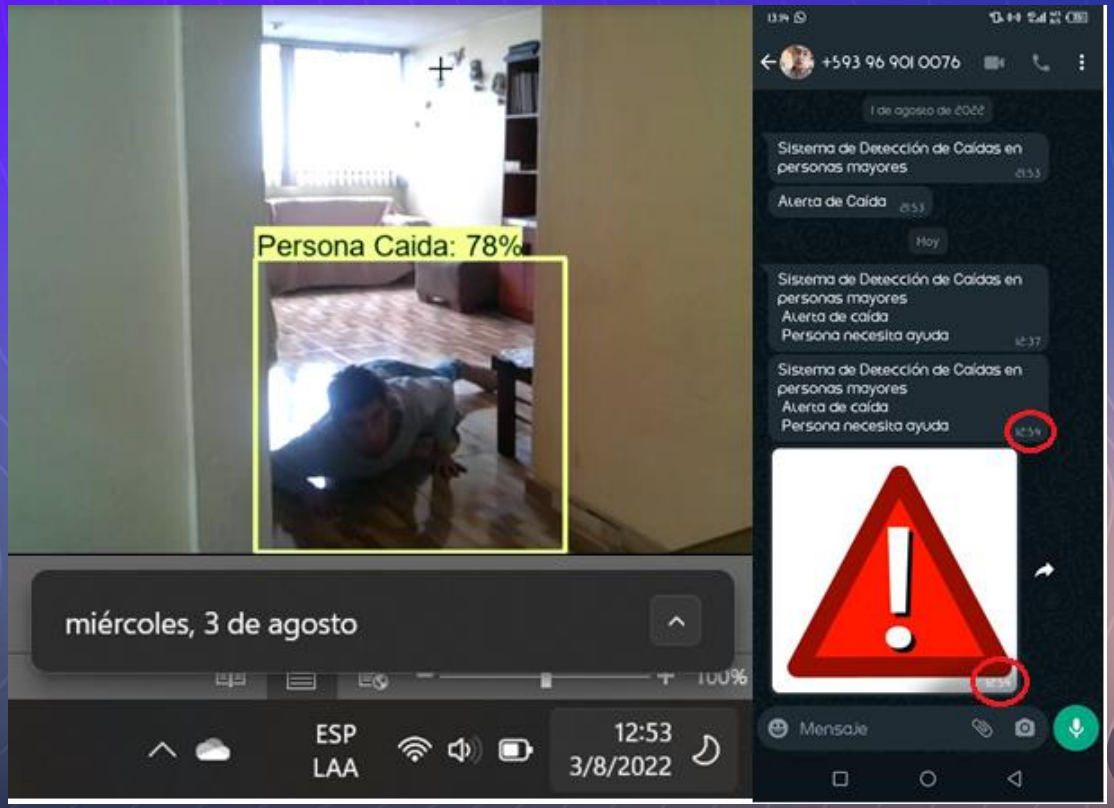


Notificación de caída

Envío de mensaje por WhatsApp

pywhatkit

- Librería de Python para envío de mensajes a whatsapp

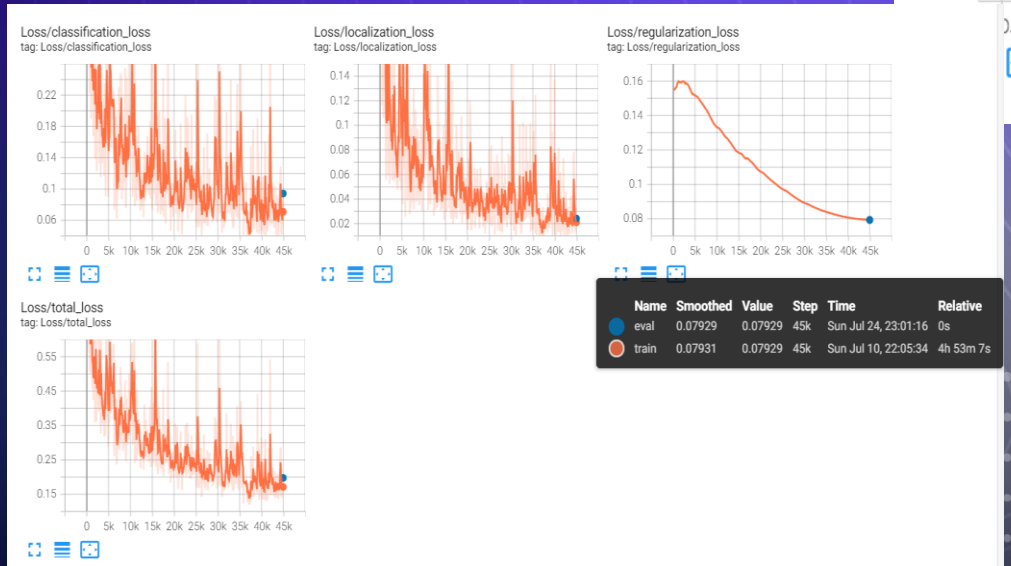


04 Experimentación y resultados

Tensorboard y parámetros de la red neuronal

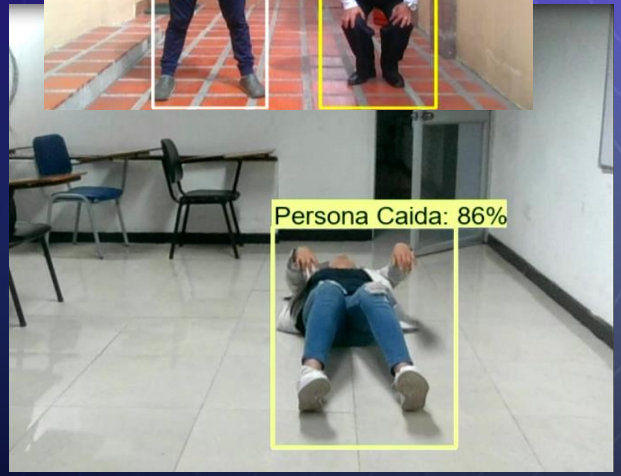
Tensorboard

Parámetros de evaluación – Función de pérdida



Ejecución del modelo de CNN

Datos en tiempo real



05

Conclusiones y recomendaciones

Conclusiones, recomendaciones y trabajos futuros

Conclusiones

Se diseñó e implementó un **sistema de detección de caídas en adultos mayores** aplicando técnicas de aprendizaje profundo con una precisión superior al 80% utilizando para ello un **modelo de redes neuronales convolucionales.**

Para lograr un buen funcionamiento del sistema fue esencial la creación de **un set de datos con imágenes propias** que facilitaron el aprendizaje del modelo de CNN

La elección del modelo **SSD Mobilenet V2** como CNN permitió el ahorro de recursos computacionales comparado con otros modelos que tienen un mayor costo de memoria.

La evaluación del sistema mediante herramientas de análisis de redes como **Tensorboard** permitió obtener la métrica función de pérdida en un valor, **aproximado de 0.079** el cuál siendo cercano a cero se lo considera como aceptable.

La utilidad del **sistema de detección de caídas** se evidencia ante la **notificación de alerta mediante mensajería instantánea**. Esto permite dar auxilio inmediato a la **víctima** reduciendo el impacto del accidente

Recomendaciones

Es importante asegurar la compatibilidad entre las **versiones de software** y **librerías de inteligencia artificial utilizadas.**

Cabe tomar en cuenta que las **últimas versiones de software** se encuentran generalmente en procesos de prueba, por lo que se sugiere **seleccionar versiones anteriores de software** que resultan más estables para el desarrollo de los proyectos.

Como se indicó anteriormente, **el set de datos es parte fundamental de la CNN**. Motivo por el cual para **mejorar el aprendizaje de la red neuronal** se debe incrementar el número de imágenes.

Si bien es cierto en **la transferencia de aprendizaje** se encuentran varios modelos de CNN pre – entrenados. Pero antes de seleccionar un modelo de trabajo, se deben **analizar las métricas tamaño vs precisión**

06 Bibliografía

Bibliografía

Birodkar, V., Joglekar, S., & Rathod, V. (7 de mayo de 2021). *Github*. Obtenido de TensorFlow 2 Detection Model Zoo: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

Casilari, E., & SantoyoRamón, J. (2018). Fall Detection Dataset (Universidad de Malaga). Obtenido de https://figshare.com/articles/dataset/UMA_ADL_FALL_Dataset_zip/4214283

Kępski, M., & Kwolek, B. (Diciembre de 2014). UR Fall Detection Dataset. Obtenido de Human fall detection on embedded platform using depth maps and wireless accelerometer, Computer Methods and Programs in Biomedicine: http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html

Lin, T.-Y., Patterson, G., Ronchi, M., Cui , Y., Maire, M., Belongie, S., . . . Dollár, P. (2017). COCO. Obtenido de Common Objects in Context: https://cocodataset.org/#home

Tzutalin. (2015). *LabelImg*. Obtenido de Git code: https://github.com/tzutalin/labelImg

Vladimirov, L. (24 de Marzo de 2022). Tutorial de la API de detección de objetos de TensorFlow. Obtenido de https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/install.html

¡GRACIAS!

¿PREGUNTAS?

Enlace Vídeo:

https://drive.google.com/file/d/13wLnHg4Wn9l31JM17BHLuCUdqL8HB5ZT/view?usp=sharing

Contacto:

Alexander Orbe
eaorbe@espe.edu.ec
0969010076