Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

Evaluación en línea del grado de involucración del usuario en actividades de interacción humano-robot

Autor:

Castañeda González, Mario

Tutores:

Gómez García-Bermejo, Jaime
Ingeniería de Sistemas y Automática

Duque Domingo, Jaime
Ingeniería de Sistemas y Automática

Valladolid, Septiembre de 2022.

Agradecimientos

Quiero mostrar mi agradecimiento, en primer lugar, a mis tutores Jaime Gómez García-Bermejo y Jaime Duque Domingo por el tiempo que me han dedicado y los conocimientos que me han transmitido.

Agradecer a toda mi familia y mi pareja por confiar en mí, y apoyarme en los momentos difíciles.

Y, por último, a mis amigos y compañeros por acompañarme en este proceso.

Resumen

Para las personas es imprescindible comprender la atención, el estado de ánimo y el contexto de una interacción con otra persona para que esta se desarrolle de una manera correcta. Sin embargo, para las máquinas esta capacidad de comprensión no es natural y hay que dotarles de esta funcionalidad.

Este trabajo trata de resolver esa falta de comprensión de las máquinas y hacerlas más “empáticas”. Tras una investigación bibliográfica, y la búsqueda de diferentes herramientas que ayuden a realizar estas tareas, se desarrolla un sistema capaz de detectar las emociones y el nivel de interés de las personas con los equipos mediante lenguaje no verbal.

Analizando las emociones y el nivel de interés, una máquina puede decidir si esa interacción supone un refuerzo positivo o debe realizar algún cambio en la manera de relacionarse con la persona. Estas capacidades ayudarán a las máquinas y los ordenadores a mejorar la comunicación con los humanos.

Palabra clave: HRI, Machine Learning, empatía, visión artificial, inteligencia artificial.

Abstract

For people, it is essential to understand the attention, mood and context of an interaction with another person in order for it to develop properly. For machines, however, this understanding does not come naturally, and they need to be given this functionality.

This work tries to solve this lack of understanding of machines and make them more "empathetic". After bibliographic research, and the search for different tools that help to carry out these tasks, a system capable of detecting the emotions and the level of interest of people with computers through non-verbal language is developed.

By analysing the emotions and level of interest, a machine can decide whether that interaction is a positive reinforcement or whether it should make a change in the way it interacts with the person. These capabilities will help machines and computers to improve communication with humans.

Keywords: HRI, Machine Learning, empathy, artificial vision, artificial intelligence.

Índice

Índice de Figuras

Índice de tablas

Abreviaturas

CPU - Central Processing Unit (Unidad Central de Procesamiento)

CNN - Convolutional Neural Network (Redes Neuronales Convolucionales)

DL - Deep Learning (Aprendizaje Profundo)

FPS - Frame Per Second (Fotogramas Por Segundo)

GPU - Graphics Processing Unit (Unidad de Procesamiento de Gráficos)

HCI - Human-Computer Interaction (Interacción Humano-Computador)

HMI - Human-Machine Interface (Interfaz Humano-Máquina)

HRE - Human-Robot Engagement (Participación Humano-Robot)

HRI - Human-Robot Interaction (Interacción Humano-Robot)

IA - Inteligencia Artificial

IDE - Integrated Development Environment (Entorno de Desarrollo Integrado)

LDA - Linear Discrimiant Analysis (Análisis Discriminante Lineal)

LBP - Local Binary Patterns (Patrones Binarios Locales)

LBPH - Local Binary Patterns Histogram (Histograma de Patrones Binarios Locales)

ML - Machine Learning (Aprendizaje Automático)

NN - Neural Networks (Redes Neuronales)

ROI - Region Of Interest (Región De Interés)

SVM - Support Vector Machine (Máquinas de Vector Soporte)

1. Introducción y objetivos

En este capítulo se manifiestan las razones por las cuales es necesario la realización del presente TFG. Además de exponer dichos motivos, se han declarado una serie de objetivos que se deben cumplir durante el desarrollo, y la estructura que se ha seguido durante la construcción del trabajo.

1.1. Introducción

El término animal social es utilizado para definir un organismo de origen animal que interactúa con miembros de su misma especie formando distintas y variadas sociedades (humanos, hormigas, abejas, ...). Para Aristóteles (384-322 a.C.) los seres humanos son animales sociales. Esta afirmación está basada en que estos se agrupan en familias, comunidades y Estados. La diferencia fundamental entre los animales y los humanos es que estos últimos son seres que presentan la capacidad de hablar.

Dentro de estas agrupaciones, los humanos necesitan establecer vínculos y relacionarse con el resto de los seres de su misma especie. No solo transmitiendo conocimiento y formando culturas como grupo, sino de manera individual con su desarrollo y la búsqueda de la felicidad. Para todo ello utilizan, en gran medida, la empatía.

La empatía es la capacidad que presenta una entidad para reconocer o comprender el estado de ánimo y/o la emoción de otra (Nummenmaa et al., 2008). Ésta ayuda a que los humanos participen en dichas interacciones sociales donde se mantiene a su vez a las personas motivadas y mejorando su estado anímico. La empatía no solo permite comprender la perspectiva de otra persona para alcanzar a entender cómo piensa o cómo se siente, también ayuda a averiguar las intenciones de las demás personas y adelantarnos a sus reacciones y comportamientos.

Para las personas, la empatía es una habilidad natural con la que logran una mejor convivencia y una buena interacción con su entorno. Pero para las máquinas esto es muy diferente, esta capacidad de comprender a una persona y adelantarse a sus reacciones no es natural. A través de algoritmos y programas, los ordenadores pueden lograr percibir ciertos rasgos y patrones, y realizar acciones en consecuencia. Al no tener la capacidad de pensar de manera autónoma, hay que programarlos para detectar cada uno de los rasgos y patrones que se deben identificar para una buena comunicación, y para determinar qué deben hacer en cada caso. Para estas labores, en el presente TFG se ha recurrido a técnicas de Visión por Computador y de Inteligencia Artificial (Artificial Intelligence, AI), concretamente Aprendizaje Máquina (Machine Learning, ML).

Dotar de estas capacidades a las máquinas tiene múltiples campos de aplicación que pueden ir desde ingeniería, donde los humanos pueden trabajar mano a mano con robots especializados, hasta medicina, donde un ordenador puede generar una alerta para avisar a los especialistas de algo que considera inusual debido a que se le ha programado para detectarlo como extraño o sospechoso en una prueba. También puede aplicarse en un ámbito social, donde un ordenador trata de empatizar con las personas para mantener conversaciones e interacciones más complejas.

Hoy en día los seres humanos, dentro de este ámbito social, plantean ciertas actividades para fomentar la socialización de las personas de avanzada edad con el resto de las personas dentro de las residencias. La intención de estas actividades es mantener el cuerpo en buena forma, mejorar el estado de ánimo, y, sobre todo, prima el ejercitar la creatividad y la memoria. Estas personas no siempre reciben suficientes estímulos, lo que suele conllevar una aceleración en el proceso de degradación anímico y cognitivo.

Para mejorar el desarrollo de esas capacidades físicas y cognitivas, el centro tecnológico CARTIF aborda el proyecto AIROSO. Se trata de un proyecto de investigación en el que se desarrolla un robot llamado “Pepper” capaz de interactuar con las personas y su entorno de forma autónoma. Este proyecto cuenta con varios objetivos, entre los que destaca mejorar la interacción humano-robot. Esta información ha sido obtenida directamente de su página web “Proyectos CARTIF AIROSO”.

Dicho objetivo es indispensable para cualquier robot u ordenador que quiera interactuar fluidamente con humanos. Entendiendo como mejora de la interacción el dar a las máquinas una herramienta para apoyar su control autónomo, y en ningún caso para sustituir profesionales de sus puestos laborales, tales como podrían ser auxiliares, psicólogos, acompañantes, etc.

Aquí es donde se crea el propósito primordial de este TFG. Crear una herramienta que mida el grado de compromiso en las interacciones humano-máquina reconociendo el estado de ánimo y de atención de los participantes. En este caso, ambas cuestiones estarán centradas en la comunicación no verbal de los individuos.

El alcance de este TFG es estudiar las diferentes posibilidades que se pueden encontrar para realizar la medición del grado de atención de una persona, mostrando qué ventajas e inconvenientes tiene cada una de esas opciones. Tras una búsqueda bibliográfica inicial, se establecen diferentes formas de medir el grado de atención y las emociones, y los requisitos que requiere cada una de ellas.

1.2. Objetivos

El objetivo principal de este trabajo se centra en el desarrollo de un sistema cuya finalidad es dotar a las máquinas de la capacidad de percibir cómo está desarrollándose una interacción entre humanos y máquinas. Este objetivo servirá de mejora a robots como "Pepper" del proyecto AIROSO y cualquier otro robot u ordenador que necesite un sistema de apoyo a la autonomía para mejorar su comunicación con las personas.

Para poder lograr el objetivo principal de crear este sistema se han establecido varios objetivos secundarios:

- Elegir qué métodos ayudan a recibir información relevante y alcanzar el objetivo principal.
- Encontrar un lenguaje de programación que permita una implementación sencilla en los equipos, y con variedad de librerías y documentación sobre la que poder trabajar.
- Evaluar los recursos mínimos necesarios sobre los que funciona este sistema.
- Evaluación y comparación de diferentes métodos para reconocer caras y emociones.
- Evaluación y comparación de diferentes métodos para localizar zonas del cuerpo con las que poder trabajar.
- Exposición de los resultados del programa obtenidos mediante gráficos o datos para que puedan ser interpretados por una persona o un equipo computarizado.

1.3. Estructura del proyecto

Este TFG posee seis capítulos donde se introduce, explica y justifica todo el proyecto. Además, consta de un capítulo bibliográfico y un capítulo de anexos al final del trabajo.

- *Capítulo 1: Introducción y objetivos*

Se hace una introducción sobre la necesidad de dotar a las máquinas de herramientas para medir el nivel de interacción de las personas con ellas, y evaluar la calidad de esa interacción. También se declaran los objetivos principales de este trabajo.

- *Capítulo 2: Fundamentos teóricos previos sobre Inteligencia Artificial*

Resumen de los conocimientos previos al desarrollo sobre los diferentes campos de IA para comprender qué hace el sistema creado y cómo funciona.

- *Capítulo 3: Proyectos y publicaciones más relevantes*

Recopilación de diferentes trabajos de la bibliografía donde se resume cómo han ejecutado sus proyectos, y porqué se hace necesario el desarrollo de este trabajo.

- *Capítulo 4: Evaluación en línea del grado de involucración del usuario en actividades de interacción humano-robot.*

Análisis comparativo de diferentes puntos sobre los que se apoya el proyecto y justificación del camino que ha seguido el trabajo.

- *Capítulo 5: Resultados*

Exposición y comparación de los diferentes resultados que se han obtenido tras el desarrollo de este trabajo en diferentes contextos. Y discusión de la necesidad forzar al sistema a funcionar dentro de alguno de esos contextos.

- *Capítulo 6: Conclusiones y líneas futuras*

Explicación de las conclusiones extraídas tras la realización del trabajo, y posibles líneas de trabajo a futuro.

- *Bibliografía y Webgrafía*

Referencias a toda la documentación usada en la elaboración del proyecto.

- *Anexos*

Incluye la descripción de las características de hardware sobre el que se ha implementado el sistema y el código en Python del propio sistema.

2. Fundamentos teóricos

A lo largo de la explicación de los proyectos y publicaciones más relevantes de interacción entre humanos y máquinas, y el desarrollo del proyecto, se manejarán una serie de conceptos que conviene tener claros para la correcta comprensión del presente trabajo.

2.1. Inteligencia Artificial (IA)

No hay una definición exacta de IA, pero puede entenderse como un grupo de herramientas orientadas a desarrollar las capacidades tales como el razonamiento y el aprendizaje autónomo en máquinas. Estas herramientas se dividen en diferentes ramas, entre las que se encuentran las que se han usado en el presente TFG como Visión por computador y Machine Learning (ML).

2.1.1. Visión por computador

La visión por computador, o visión artificial, es una rama de la IA, que consiste en un grupo de herramientas que incluyen métodos para obtener y procesar imágenes. El ordenador obtiene datos numéricos o simbólicos de estas herramientas y es capaz de reconocer información de ellos como personas, animales, plantas, objetos, etc.

Los IA's más básicas detectan patrones constantes y su precisión es muy limitada si se sale de las características programadas, pero las más elaborados son capaces de realizar clasificaciones más precisas cuando los patrones son aleatorios. Estos métodos son equivalentes al uso que hacen los humanos y los animales con sus ojos y cerebros para percibir el entorno que les rodea.

2.1.1.1. LBPH

A lo largo del desarrollo del sistema para la estimación del nivel de atención de un humano se ha planteado la posibilidad de que aparezca más de una persona dentro de una imagen. Esto hace necesario diferenciar a unas personas de otras para guardar los registros de manera independiente.

El rasgo más significativo que se ha encontrado para diferenciar a las personas es la cara, y por ello se ha decidido utilizar un método de reconocimiento facial liviano para el sistema como LBPH. Éste es un método de visión artificial muy usado en el reconocimiento facial basado en el operador Local Binary Patterns (Patrones Binarios Locales, LBP) para obtener características de una imagen en escala de grises (Esparza C.H., 2015).

Este operador originalmente consistía en una ventana de 3x3 pixeles de imagen donde el píxel central actuaba como el umbral de los 8 píxeles circundantes, tal como se aprecia en la Figura 1. Si el valor del píxel central era mayor que el del píxel con el que se comparaba, se sustituía ese valor con un 0. Si por el contrario el número era inferior se sustituía con un 1.

Al sustituir los 8 números vecinos por unos y ceros se obtenía una cadena binaria de 8 dígitos. Empezando por la celda superior izquierda de la matriz y siguiendo sentido horario se transformaba ese número de binario a decimal. Como la cadena binaria estaba formada por 8 dígitos, el rango de ese número binario era de 0 a 255 en decimal, como se puede ver en la Ecuación 1 y la Ecuación 2.

$$00000000_2 = 0_{10} \qquad \textit{Ecuación 1}$$

$$11111111_2 = 255_{10} \qquad \textit{Ecuación 2}$$

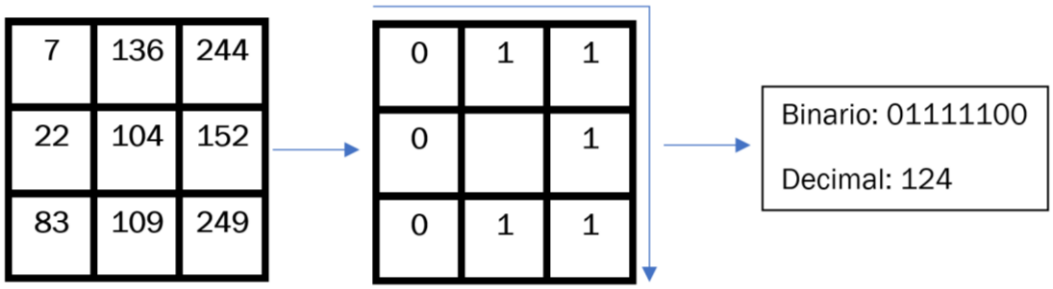

Figura 1 - Funcionamiento operador LBP con vecindad cuadrada

Versiones mejoradas de este operador permitían dimensiones superiores a 3x3 y, en vez de utilizar una vecindad cuadrada usaban una circular, como se muestra en la Figura 2. Toda esta información aparece reflejada en "Introducción al principio de LBP y la implementación del algoritmo" (Programmerclick).

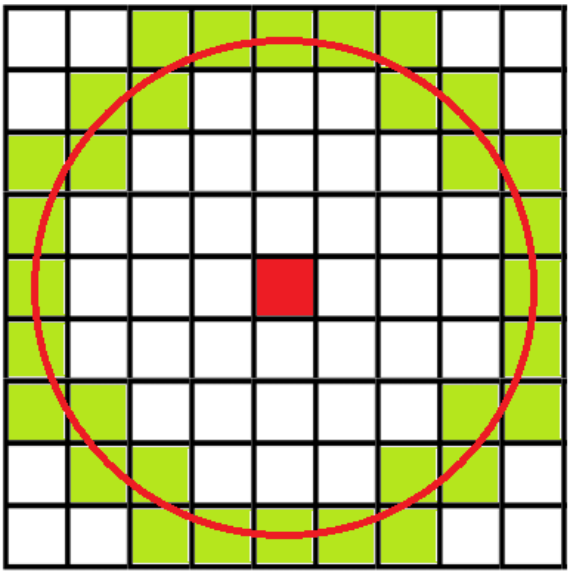

Figura 2 - Operador LBP con vecindad circular

Al aplicar LBP a toda la imagen se obtiene un valor entre 0 y 255 en cada posición de la imagen, y se crea su histograma de características. Después se divide en regiones más pequeñas, y se vuelve a obtener el mismo rango de valores en cada posición de cada una de esas regiones, y se crean los histogramas de características de dichas zonas.

Comparando estos gráficos entre diferentes imágenes se puede obtener su similitud. A este proceso en particular que trabaja con histogramas se le llama LBPH. Un histograma es un gráfico de barras que representa las distribuciones de frecuencias, y ayudan a ver un conjunto de datos fácilmente, tal como se observa en la Figura 3.

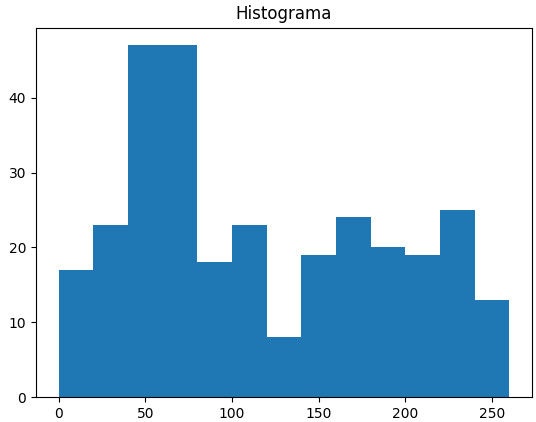

Figura 3 - Ejemplo de histograma

Si las figuras de estos gráficos son semejantes dentro de una tolerancia se considera que las imágenes son la misma, y se devuelve un valor de proximidad entre ellas.

2.1.1.2. FisherFaces

Al igual que el método LBPH, FisherFaces también se ha analizado como técnica para el reconocimiento facial. Este procedimiento de visión artificial está basado en el Análisis Discriminante Lineal (LDA - Linear Discrimiant Analysis), que es otro método muy empleado en estadística que se utiliza para encontrar una combinación lineal de predictores que separan dos o más clases de objetos.

FisherFaces le aporta a LDA los predictores que debe usar, y obtiene como resultado la probabilidad de pertenecer a una de esas clases analizadas mediante el teorema de Bayes. Este teorema calcula la probabilidad de un suceso en función de una información obtenida previamente, cuya fórmula estadística se encuentra visible en la Ecuación 3.

$$P[A_n/B] = \frac{P[B/A_n] \cdot P[A_n]}{\Sigma P[B/A_i] \cdot P[A_i]}$$

Ecuación 3 - Teorema de Bayes

Trabajos como el de comparar tres métodos convencionales de reconocimiento facial y confrontar el de mejor resultado frente a una computación cognitiva (Ospina A., 2017) concluyen que el método FisherFaces, al que nombran “en cascada”, es igual de rápido que LBPH, y además posee una tasa mayor de aciertos.

2.1.2. Machine Learning (ML)

Existen otras ramas de IA como el ML que aporta a los equipos la competencia de identificar patrones y elaborar predicciones. Existen diferentes algoritmos, pero los más apropiados para el desarrollo de este trabajo han sido neural networks (NN), deep learning (DL), SVM y el detector-rastreador (detector-tracker).

Estas técnicas de IA requieren un proceso de entrenamiento con un conjunto de muestras que sirven para ajustar los parámetros que requiera cada algoritmo. Después del entrenamiento se necesita un proceso de validación con otro conjunto de muestras, el cual proporciona una valoración imparcial del

entrenamiento anterior. Y finalmente se realiza un testeo después de la validación, que consiste en utilizar otro conjunto de prueba que se utiliza para obtener la tasa de aciertos de los modelos obtenidos con unos casos semejantes a los reales.

A continuación, se explican los métodos de ML utilizados para el desarrollo de tareas como detectar y rastrear las caras localizadas en las imágenes, y la evaluación del grado de interés de una persona localizada en la imagen.

2.1.2.1. Neural Networks (NN) y Deep Learning (DL)

Las técnicas de NN son un subconjunto de las técnicas de ML, las cuales se han utilizado en este trabajo para evaluar el grado de interés de las personas que aparecen en las imágenes. Su característica principal reside en que su funcionamiento está inspirado en las neuronas del cerebro, lo cual induce que, de la utilización de esta en tareas influenciadas por comportamientos humanos, se obtengan mejores resultados. Aunque en el sistema desarrollado en este TFG no se usa GPU, este método alcanza su máximo rendimiento haciendo uso de este elemento.

El algoritmo está formado por capas de nodos que presentan una capa de entrada (círculos rojos), una de salida (círculos verdes) y una o varias capas ocultas entre medias (círculos azules). Estos nodos consisten en patrones que sirven como máscaras en las imágenes, quienes permiten obtener la semejanza de esa imagen con otra.

Cada nodo, o neurona, se conecta a los nodos de las capas siguientes, con un peso y un umbral. Si un nodo está por encima de ese umbral, envía los datos a la siguiente capa, y continúa ese proceso hasta llegar al final, tal como se ve en la Figura 4.

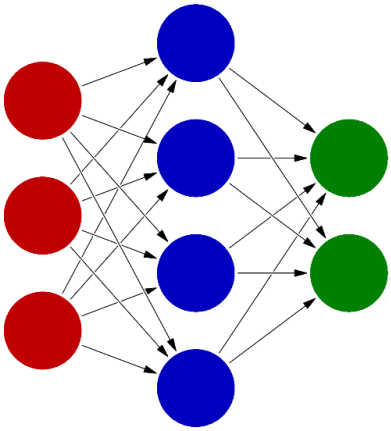

Figura 4 - Esquema de red neuronal (rojo - capa de entrada, azul - capa/s oculta/s, verde - capa de salida)

DL en esencia es igual a NN, pero utiliza una gran cantidad de capas intermedias (azules), es decir, trabaja a más profundidad. De ahí la palabra “deep” (profundo). Como se ha mencionado anteriormente, las capas intermedias (azules) pueden estar formadas por una única capa o por varias.

Las primeras capas suelen estar formadas por patrones más reconocibles como líneas y circunferencias. La ventaja que se obtiene de utilizar muchas capas intermedias es que los patrones en las capas más profundas son más aleatorios. De esta manera se consiguen máscaras más elaboradas para identificar determinados rasgos y con las que diferenciar a unas personas de otras.

2.1.2.2. Support Vector Machine (SVM)

A fin de comparar los resultados de las NN’s para evaluar el grado de interés de una persona y una máquina, también se ha analizado SVM. Este método consiste en un conjunto de algoritmos de aprendizaje supervisado que, dado un conjunto de muestras (datos de entrenamiento) etiquetados en alguna clase, construyen un modelo capaz de devolver una estimación de la clase a la que pertenece.

El funcionamiento de esta técnica consiste en seleccionar dos categorías pertenecientes a todo el conjunto de clases, y situar todos los elementos en un espacio vectorial igual al número de componentes que tienen. Como explica Joaquín Amat en su escrito sobre máquinas de vector soporte (Amat J., 2017), si los elementos se pueden separar mediante un hiperplano, se obtiene directamente la ecuación que les caracteriza. Cuando se use el modelo para determinar a qué clase pertenece, se aplica dicha ecuación y se evalúa a qué lado del hiperplano cae un dato (Gráfico 1 -Figura 5).

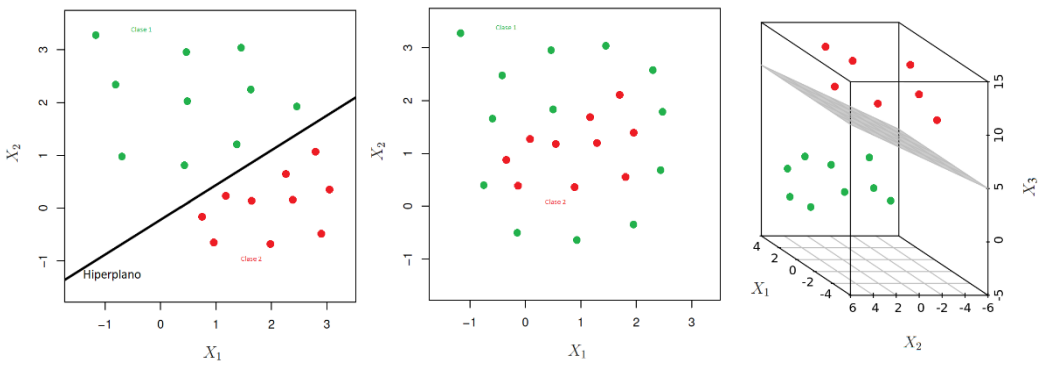

Figura 5 - Ejemplo de separación de clases con dos componentes sin kernel y con kernel.

Si los datos no se pueden separar, entran en juego los kernels. Los kernels realizan cálculos entre las componentes que determinan una clase generando componentes nuevas. Al generar una nueva componente, aumenta en una dimensión el espacio vectorial de manera que se pueda separar cada clase con un nuevo hiperplano (Gráficos 2 y 3 - Figura 5).

Una vez comparadas todas las clases involucradas en el entrenamiento dos a dos, se selecciona el resultado más repetido, tal y como se muestra en la Tabla 1.

Comparación clases	Resultado
Clase 1 vs Clase 2	Clase 1
Clase 1 vs Clase 3	Clase 1
Clase 2 vs Clase 3	Clase 3

Tabla 1 - Ejemplo de comparación multiclase SVM

En este ejemplo el resultado más repetido sería la Clase 1, por lo que el algoritmo estima que ese es el resultado correcto.

2.1.2.3. Detector-rastreador

El algoritmo detector-rastreador en un primer paso detecta una región de interés (ROI – Region Of Interest) en una imagen, y después mediante DL, el rastreador averigua los puntos característicos en ese ROI y les hace un seguimiento. En este trabajo las ROI's serían las personas dentro de la imagen, y los puntos característicos serían las articulaciones del cuerpo y zonas características de la cara.

2.2. Solve Perspective n Points (solvePnP)

SolvePnP es un algoritmo desarrollado en la librería de OpenCV que resuelve el problema de estimar la *pose* (posición) de un cuerpo humano en una imagen obtenida a través de una cámara calibrada. Para ello necesita como mínimo 3 o más puntos del cuerpo con información en 2D y 3D con los que se realiza el cálculo de la matriz rotación | traslación. La operación con la que se realiza dicho cálculo se muestra en la Ecuación 4.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Ecuación 4

Tal como señalan en "Descripción de método solvePnP de OpenCV" (Delftstack.), el primer vector $[u \quad v \quad 1]^T$ corresponde con la posición 2D de un punto y el vector $[x \quad y \quad z \quad 1]^T$ corresponde con la posición en 3D del mismo punto anterior.

$\begin{bmatrix} \mathrm{f_x} & 0 & \mathrm{u_0} \\ 0 & \mathrm{f_y} & \mathrm{v_0} \\ 0 & 0 & 1 \end{bmatrix}$ es la matriz de la calibración de la cámara, y $\begin{bmatrix} \mathrm{r_{11}} & \mathrm{r_{12}} & \mathrm{r_{13}} & \mathrm{t_1} \\ \mathrm{r_{21}} & \mathrm{r_{22}} & \mathrm{r_{23}} & \mathrm{t_2} \\ \mathrm{r_{31}} & \mathrm{r_{32}} & \mathrm{r_{33}} & \mathrm{t_3} \end{bmatrix}$ es la matriz $[r|t]$ (rotación | traslación).

Esta ecuación itera sobre todos los puntos hasta que se encuentra una matriz $[r|t]$ que se cumpla en todas las ecuaciones. Por tanto, el dato que esta función devuelve es la matriz de rotación y traslación en los 3 ejes como una lista de vectores (de Python).

2.3. RQDecompo3x3

Es una función de OpenCV que realiza la transformación de una matriz de rotación en ángulos de Euler. Siendo las rotaciones sobre los ejes X, Y y Z, los ángulos ϕ, θ y ψ, tal y como se muestra en la Ecuación 5, Ecuación 6 y Ecuación 7.

$$r_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,\phi & -sin\,\phi \\ 0 & sin\,\phi & cos\,\phi \end{bmatrix}$$

Ecuación 5 - Matriz de rotación sobre eje X

$$r_Y = \begin{bmatrix} cos\,\theta & 0 & sin\,\theta \\ 0 & 1 & 0 \\ -sin\,\theta & 0 & cos\,\theta \end{bmatrix}$$

Ecuación 6 - Matriz de rotación sobre eje Y

$$r_Z = \begin{bmatrix} cos\,\psi & -sin\,\psi & 0 \\ sin\,\psi & cos\,\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 7 - Matriz de rotación sobre eje Z

2.4. Perpendicular de la proyección de un segmento sobre un plano

Cuando no se poseen 3 puntos para aplicar solvePnP se puede recurrir a las proyecciones. En este proyecto, se ha utilizado la proyección sobre el plano XZ (Figura 6) para estimar la orientación sobre el eje Y del torso.

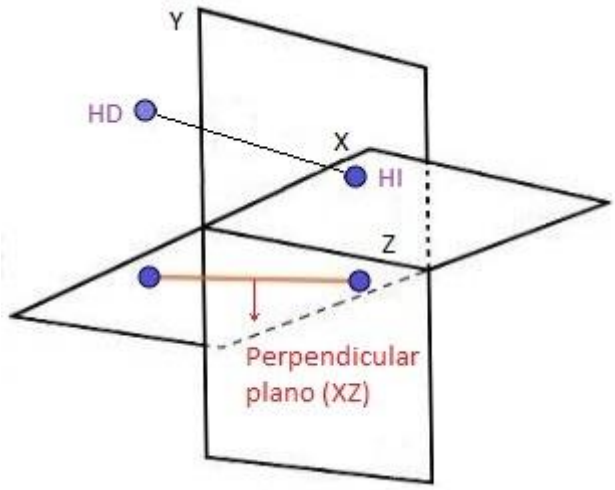

Figura 6 - Representación de la proyección de un segmento sobre plano XZ. HD (Hombro Derecho) y HI (Hombro Izquierdo)

Una vez proyectado el segmento que une el hombro derecho (HD) y el hombro izquierdo (HI) sobre el plano XZ se ha buscado el punto medio y se ha trazado la perpendicular. El ángulo de ese segmento respecto al eje Z, da el ángulo de rotación sobre el eje Y.

3. Proyectos y publicaciones más relevantes de interacción entre humanos y máquinas

En la actualidad existen diversos trabajos relacionados con la interacción humano-computador (HCI), interacción humano-máquina (HMI) y con la participación del robot con los humanos (HRE). Los estudios de estos temas abarcan distintos sectores desde humanidades (para el estudio de la conducta humana), pasando por medicina (para la mejora de los tratamientos de los pacientes), llegando hasta otros como ciencias sociales, ingeniería, etc (Zulkifli, 2018).

3.1. Empatía en máquinas

Algunas publicaciones afirman que la empatía es un desencadenante emocional básico para los seres humanos que influye en el comportamiento y las relaciones sociales (Vallverdu, 2020). Partiendo de esta premisa, se busca reforzar esa HMI en las máquinas haciendo que los humanos podamos empatizar más con ellas. Y, por otro lado, que estos equipos sean capaces de analizar mejor a los humanos. En esta misma publicación obtienen como conclusión que los humanos muestran más empatía hacia el robot cuanto más físicamente humano parece, y cuanto más reconoce el afecto humano y responde de manera correcta (Vallverdu, 2020).

Figura 7 - Dos personas realizando gesto de compromiso. (Obtenida de Wikimedia - Bajo licencia Creative Commons)

Coincidiendo con dicha conclusión (Vallverdu, 2020), en otra investigación se realiza un experimento que indica que los humanos eran más participativos cuando el robot con el que se estaban comunicando realizaba gestos (Sidner C.L., 2005). Estas señas del robot son llamadas “gestos de compromiso“ como los de la Figura 7, y su función es hacer que los humanos desarrollen más empatía hacia la máquina. En dicha figura se observa cómo, el gesto haciendo referencia a dos unidades, refuerza la comunicación verbal con lenguaje no verbal, y la persona que recibe la información realiza un gesto reflejo imitando a su interlocutor.

3.2. Uso del HRI en la ayuda a la atención de personas con necesidades especiales

Estas investigaciones se realizan a menudo para ayudar a personas que necesitan mayor atención de la que se le suele dar, por ejemplo, niños con autismo (Amir, 2010). En este trabajo explican que las personas con autismo, frecuentemente, tienen problemas de imitación motora lo cual les limita la realización de esos gestos de compromiso. Pero enseñándoles a reproducir los movimientos mejoran su capacidad de respuesta social porque la imitación y el mimetismo son indicadores del compromiso humano. El estudio sobre el que trabajan es HRI, puesto que la función del robot es hacer que los niños copien los gestos mientras se asegura de que lo hacen correctamente. Para ello el robot analiza en una imagen la posición de zonas características del cuerpo. Posteriormente hace un seguimiento de la posición a lo largo del tiempo, diferenciando de esa manera unos gestos de otros. Cabe destacar que, aunque este equipo trabaja de forma autónoma, siempre realizan los ejercicios delante de un terapeuta o un miembro de la familia.

En este trabajo obtienen datos de la persona a la que está observando (HRI), pero en ningún caso se obtienen datos que supongan un entendimiento de esa persona. Además, la respuesta de la máquina no modifica la interacción basándose en la empatía (HRE), solo evalúa si es correcto o incorrecto. Tampoco se analizan los movimientos de la persona como lenguaje no verbal por parte del equipo, sino como gestos que deben ser repetidos dentro de un contexto.

Por otro lado, la obtención de la sonrisa como medio para evaluar la respuesta a una interacción puede ser un importante indicador de una interacción adecuada (Zulkifli, 2018). En esta investigación se presenta a un robot como una cría de foca llamada “PARO” para que seis personas en rehabilitación por depresión tras un accidente cerebrovascular empaticen más profundamente con PARO. Con el análisis de la sonrisa basado en OpenCV-Python se tiene un indicador de emociones positivas. Los resultados se compararon con

evaluaciones psicológicas y mostraron que, cuando estas personas sostenían a PARO, el número de sonrisas aumentaba, lo cual ayudaba a los pacientes a controlar su angustia psicológica. La detección de un indicador como la sonrisa, que es una respuesta emocional positiva, sí puede considerarse como empatía, y puede reconocerse como un indicador de HRE. Pero en este proyecto solo utilizan esos datos como un punto donde obtener información y no está previsto un cambio en la interacción con la persona por parte de la máquina si dejan de recibirse sonrisas.

3.3. HRI y HRE

Para reconocer el afecto de la persona, las máquinas se basan en gran medida en la comunicación no verbal (gestos involuntarios de la cara, posición del cuerpo y las extremidades, dirección de la mirada, etc.). Como señala Jaime Duque-Domingo en su trabajo sobre el control de la mirada de una cabeza robótica para una interacción realista con humanos, cuando dicha máquina interactúa con varias personas, el control de la mirada es muy importante (HRI) (Duque-Domingo J., 2020). En su artículo presentan un método para decidir a quién prestar atención al interactuar con varias personas al mismo tiempo a través de la mirada. Para ello han construido la cabeza de un robot integrada en un modelo de arquitectura ROS (Robot Operating System). A este autómata le introducen en una conversación con diferentes participantes y teniendo en cuenta las miradas de los participantes, replica el comportamiento humano. Este trabajo sí que modifica la manera de relacionarse en cada interacción autónomamente y percibiendo la participación de las personas (HRE), pero en ningún caso percibe cómo se pueden sentir, ni qué grado de atención tienen sobre el robot. Solo modifica su orientación en función de la gente que hay y quién orienta su mirada hacia él (Duque-Domingo J., 2020).

Algunos autores se refieren a la dirección de la mirada como la trayectoria hacia donde está orientada la cara mediante reconstrucciones 3D que nos ofrecen algunas aplicaciones. Una referencia de estas reconstrucciones en 3D es el Trabajo de Fin de Máster de la UPV (Universidad Politécnica de València), donde se entrenan NN para clasificar puntos del cuerpo humano en 3D (Pérez F., 2021). El objetivo de este trabajo era clasificar y corregir las posturas del cuerpo humano, y para ello, localizaban diferentes puntos del cuerpo y analizaban las posturas comparándolas con una base de datos antropométrica creada por ellos mismos. Esta base de datos utilizaba imágenes de personas realizando los diferentes movimientos a analizar. A diferencia de otros trabajos, hay que mencionar que este es llamativo entre otras cosas porque se desarrolló para una app de móvil, dejando claro que programar en equipos con menos recursos es una vía razonable (Pérez F., 2021).

Al igual que en otros trabajos consultados para el buen desarrollo del presente TFG, el sistema de este TFM analiza datos y emite una alerta para que la persona realice un cambio. Pero se basa en estudiar estrictamente lo que ve, no profundiza en sentimientos, ni en la capacidad de comprender si la interacción está siendo correcta.

Otros autores se refieren a la dirección de la mirada estrictamente como la trayectoria hacia donde apuntan los ojos (Rico C., 2021). El objetivo principal de ese trabajo era la creación de un entorno experimental donde predecir la mirada a través de la webcam. Para ello entrenaron una NN alimentada con información del análisis de la posición de las pupilas y el iris en tiempos previos y posteriores a una interacción. La acción consistía en hacer clic en un lugar de la pantalla y guardar la posición del puntero del ratón. De esta manera la IA aprendía dónde estaba mirando en función de la información que recibía y el lugar donde se estimaba que estaba fijando la vista. La ventaja de este trabajo es que explica claramente cómo usaba una IA para obtener automáticamente la dirección de la mirada sin tener que realizar tediosos cálculos. Pero como el resto, solo se basa en obtener la trayectoria hacia donde están dirigidos los ojos, no evalúa de ningún otro modo un nivel de interacción en función de la mirada.

Existe un trabajo diferente a los anteriores que estudia la interacción humano-robot desde el punto de vista de la frustración (Kapoor A., 2007). Lo interesante de este artículo es que usan los mismos sistemas de visión artificial que en otros artículos, pero además usan sensores de presión tanto en el ratón como en la silla. Con estos canales de entrada de información consiguen evaluar la frustración de los alumnos (niños) evitando la falta de motivación y el abandono al encontrar signos de dificultad. Este trabajo corresponde a HRE porque valora de alguna manera el estado de ánimo de los niños, pero en ningún caso se utilizan estos datos para modificar la conducta de un robot u ordenador.

Finalmente existe un último trabajo relacionado con HRE el cual plantea la predicción automática de la participación de la persona basándose en características no verbales (Oertel C., 2020). Estas características son: la velocidad de respuesta, atención visual y postural, vacilación conversacional, entre otras. Además, también incide en reacciones fisiológicas tales como la frecuencia cardiaca y la actividad electrodérmica. Estas evaluaciones se llevan a cabo en entornos muy controlados y con muchos sensores. Aunque este trabajo sí que plantea como medir la participación de la persona con el robot, requiere sensores externos que impiden que el autómata tenga libertad para moverse.

Todos los trabajos que aquí se plantean aportan nociones sobre cómo obtener algunos datos (emociones, posiciones del cuerpo, …), o cómo utilizar esa

información para generar una respuesta más empática. Pero ningunos de ellos profundiza en la calidad ni en la magnitud de dicha interacción, excepto en el último proyecto analizado que sí que lo hacía, pero asociaba el funcionamiento de la máquina a un lugar concreto de trabajo. Este es uno de motivos fundamentales por los que se ha desarrollado este TFG, para buscar cómo evaluar el grado en el que las personas se involucran con el equipo, y aparte, si las interacciones se están realizando de una manera correcta, sin limitar la movilidad de los equipos.

4. Evaluación en línea del grado de involucración del usuario en actividades de interacción humano-robot

A continuación, se ha realizado una comparación de todas las posibles vías de implementación de este sistema, y se han seleccionado y justificado los motivos por los que se han utilizado ciertos elementos y otros han sido descartados.

En el Diagrama 1 se muestra el flujo de funcionamiento general del sistema. En la imagen se puede observar la necesidad de crear ciertas funciones que sirvan para detectar diferentes puntos del cuerpo y obtener información de ellos. Así como la necesidad de utilizar técnicas rápidas y efectivas que mejoren el rendimiento de todo el sistema.

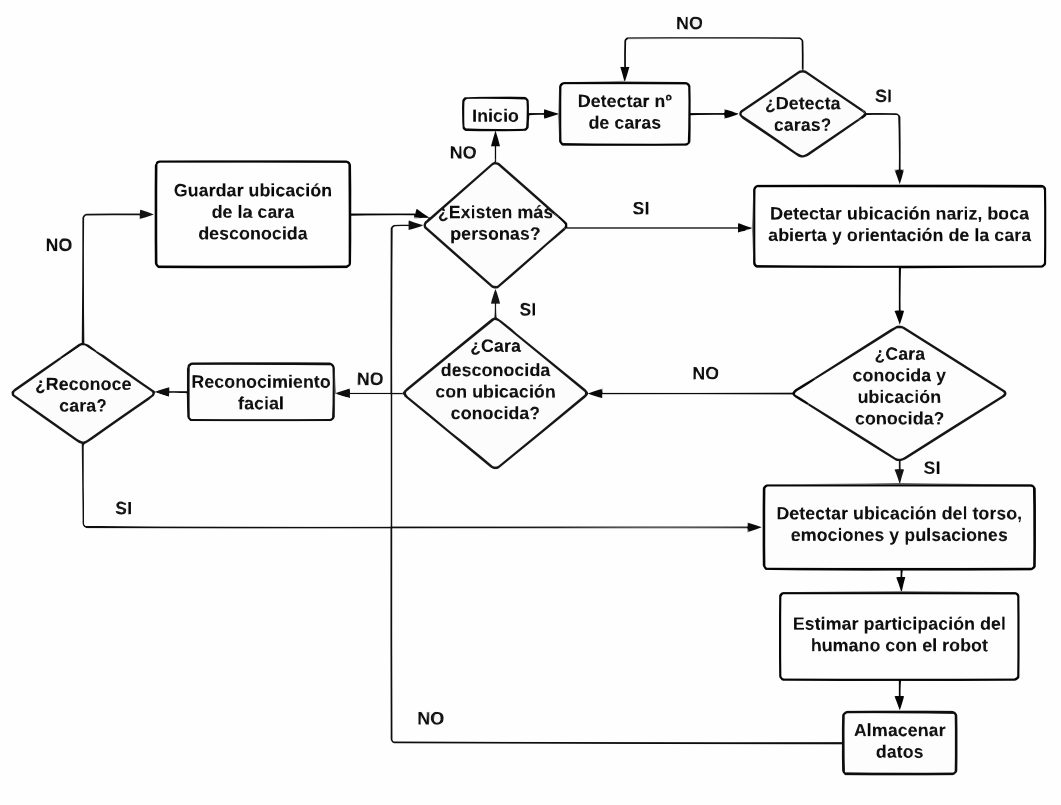

Diagrama 1 - Bucle de funcionamiento del sistema para recibir información

4.1. Canales de entrada de información

Durante la investigación bibliográfica se recopilaron diferentes métodos para obtener información a partir de varios canales de entrada como los de Duque-Domingo J. (2020), Oertel C. (2020) y Kapoor A. (2007). Los canales de entrada de información que se han barajado para el sistema de este trabajo presentan

diferentes ventajas y características. A continuación, se describe cuáles han sido sus fortalezas, y qué inconvenientes han provocado que se rechazara su uso en este trabajo.

4.1.1. Sensores en la zona de trabajo de la máquina

A partir del trabajo de Kapoor A. (2007) se planteó la idea de colocar sensores en una zona de trabajo controlada como son el respaldo, reposabrazos y asiento de una silla. Estos sensores, ofrecían la posibilidad de estimar de una manera precisa en qué posición se encuentra un cuerpo. Una figura humana bien sentada, con una posición erguida o con el torso adelantado podían ser indicadores de un gran nivel de atención, por el contrario, un cuerpo mal sentado indicaría distracción y que no existe interés por la interacción.

Este método nos hubiera dado una idea aproximada de cómo se encuentra una figura en un momento puntual, pero planteaba una serie de desventajas que fueron motivo de rechazo. En primer lugar, necesitaba un entorno controlado asociado a un asiento o una mesa, lo que le restaba movilidad al equipo en el que pudiese ser instalado. En segundo lugar, se hubiera requerido de una inversión y una instalación previa con algún controlador para recibir y transmitir esa información. Además, existen otros medios con los que recibir dichos datos, aunque sean menos precisos.

4.1.2. Cámara principal de la máquina

Una cámara integrada en un ordenador o robot limita el campo visual. La ventaja es que, por la distancia del ROI se pueden captar los mismos detalles con una cámara de poca calidad con el correspondiente ahorro económico. Y, además, al estar asociado al equipo donde está implementado no le resta movilidad. Tal y como demuestra Duque-Domingo J. (2020) en su trabajo, las técnicas de Visión Artificial ofrecen una gran cantidad de posibilidades para obtener información que se pueden aprovechar en este proyecto.

Esta particularidad lo hace versátil y por eso se ha implementado en este TFG. Pero hay que evaluar qué características mínimas debe tener dicha cámara para funcionar en el mayor número de escenarios posibles.

4.1.3. Cámaras en diferentes perspectivas

Complementando a la idea anterior se puede obtener un gran campo de visión del entorno si, además de la cámara de la máquina, se colocaran varias cámaras en la habitación donde se disponga a trabajar el robot u ordenador.

Con este método ganamos en capacidad visual, aunque requiere una programación más elaborada. Además, supone un coste bastante elevado en función del número de cámaras usadas, y también se estaría asociando el equipo a un lugar de trabajo. Teniendo otras opciones que nos permiten más movilidad, limitar el equipo a una zona de trabajo es el motivo de rechazo de esta idea.

4.1.4. Periféricos de entrada como teclado y pantalla táctil

Con el trabajo de Kapoor A. (2007) se pudo ver la utilidad de los periféricos de entrada como teclados, ratones y pantallas táctiles de los que disponen la mayoría de los equipos. Estas pueden servir para saber si se está interactuando con el equipo, es decir, si se le está prestando atención tal y como se aprecia en la Figura 8.

Figura 8 - Persona interactuando con los periféricos de entrada con un nivel de atención alto sobre el equipo. (Obtenida de Pxhere - Bajo licencia Creative Commons)

Aunque no todas las máquinas disponen de este hardware, su ausencia no limita la movilidad del propio equipo, ni interfiere con la obtención de otros resultados, solo provoca que no se tenga en cuenta esa información. Como la posible ausencia de este elemento no interfiere con el resto del sistema, no ha supuesto un rechazo de la idea. Por este motivo se ha implementado en el proyecto.

4.2. Plataformas de trabajo

Habitualmente los robots sociales incluyen una tableta para mejorar la comunicación con los humanos. Estas tabletas normalmente son equipos con menor potencia de cálculo que los ordenadores. Las computadoras, aunque tengan más potencia de cálculo suelen tener otras funciones ya integradas como el control del cuerpo de un robot y la gestión de las comunicaciones del equipo. El sistema de evaluación del nivel de participación desarrollado debe integrarse en alguno de estos lugares para no requerir una instalación más costosa, es decir, tener que comprar un hardware específico para instalar este sistema.

Por esto se propone como un objetivo secundario que el sistema sea suficientemente liviano como para funcionar correctamente en equipos con menos prestaciones, o al que se van a destinar menos recursos para ejecutar este sistema. Para poder implementar este proyecto en equipos de bajos recursos como tabletas u ordenadores de poca potencia se ha propuesto utilizar Android Studio o Python.

Programar en Android Studio supone aprender lenguajes de programación muy estrictos como Java o Kotlin. Existen librerías como TensorFlow Lite, OpenCV o Mediapipe para crear aplicaciones sencillas en Android. Estas librerías a menudo suelen ser incompatibles entre ellas porque requieren una serie de complementos con unas versiones diferentes. Y aunque se ha conseguido una aplicación básica para detectar y mostrar zonas características del cuerpo con TensorFlow Lite, no se han conseguido utilizar las funcionalidades de otras herramientas en la misma aplicación.

Como uno de los objetivos de este trabajo es evaluar diferentes maneras de obtener el nivel de interacción de una persona y sus emociones, y la dificultad de este problema se encuentra en el entorno de programación, se ha decidido seguir otro camino más versátil. En este caso, se continúa mediante Python, que ofrece una programación y un uso de diferentes librerías más directa. Python se puede instalar sobre diferentes sistemas operativos, lo cual no quiere decir que el proyecto pueda funcionar en cualquiera de esos sistemas.

El proyecto se ha desarrollado en Python sobre Windows 10, y al probar Python sobre Android se ha puesto de manifiesto la posibilidad de su instalación, pero las librerías son las que limitan su uso. Por ejemplo, la herramienta OpenCV se puede instalar en Python sobre Android, pero no accede a recursos como la cámara o ventanas para mostrar imágenes.

Cabe mencionar que se ha pensado en plataformas comerciales como Unity, que permite utilizar estas librerías, programar en C# y exportar la aplicación a

diferentes plataformas. Pero al ser comerciales se descartaron para evitar gastos en la implementación.

4.3. GPU vs CPU

Las librerías que utilizan IA mejoran su rendimiento con el uso de las GPU's gracias a que pueden realizar en paralelo cientos de cálculos. Pero hay que evaluar la necesidad de este elemento en función del rendimiento, el coste de adquirir este hardware y la facilidad de implementación.

Para evaluar la necesidad de la GPU en función del rendimiento hay que tener en cuenta si existe algún otro elemento que lo pueda reemplazar, en este caso la CPU. Haciendo uso de los equipos con las características definidas en el Anexo I, el rendimiento del sistema puede llegar a alcanzar 1,8 FPS. Este valor es bastante ajustado, la fluidez de la captura de imágenes deja de ser fluida y se podría considerar como el límite desde el cual no debería disminuir.

Si este sistema se implantara en equipos con menos recursos, el rendimiento sería aún menor y la captación de las imágenes estaría muy distanciada en el tiempo. En esos intervalos se pueden producir eventos breves que no quedarían registrados y se perdería dicha información. Esto es debido a que alguna biblioteca como la de reconocimiento facial requiere gran cantidad de capacidad de cálculo. En principio, esto serviría para forzar el uso de la GPU, pero se ha conseguido realizar un seguimiento de las caras reconocidas para hacer uso de esta librería únicamente cuando se requiere.

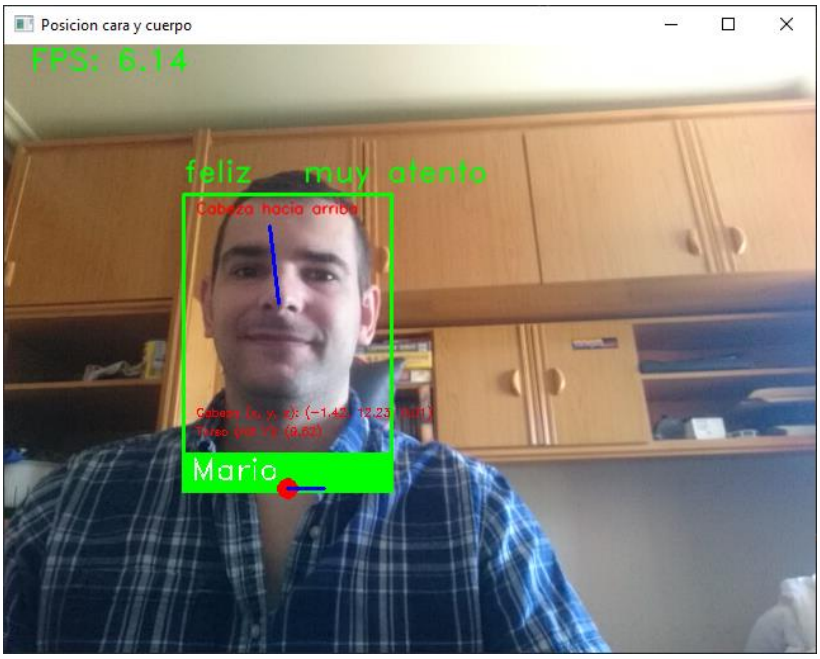

Figura 9 - Rendimiento de 6 FPS con seguimiento de la nariz.

Tal y como se muestra en la Figura 9, con este seguimiento se alcanzan valores en torno a 6 FPS de media, y nos aporta un margen razonable para mantener la fluidez del sistema en situaciones donde se pueda tener menos potencia de cálculo en los equipos. Y, por tanto, haciendo que la GPU no sea un elemento indispensable.

Se ha planteado hacer uso de alguna nube como Google Colab, que permite el uso de GPU's y no supone una inversión tan grande. Pero se ha observado que el tiempo que gana mejorando el rendimiento se pierde por la latencia de enviar y devolver imágenes a través de internet.

Para trabajar en local, aparte de suponer una inversión mucho mayor, la implementación para que Python haga uso de la GPU es más compleja. Requiere de varios programas con unas versiones específicas, que únicamente se pueden utilizar en unas series de tarjetas gráficas determinadas. De esta manera, se estaría obligando a realizar una inversión muy concreta en un elemento con un coste bastante elevado y ya no se podría utilizar dicho sistema en equipos con pocos recursos.

Estas razones llevan a rechazar su uso en este proyecto, tanto en local como en una nube, pero se sigue considerando una vía muy razonable para otros proyectos que exijan más potencia de cálculo y estén dispuestos a realizar una inversión mayor.

4.4. Posición de la cara, el cuerpo y la mirada

Como el dispositivo principal para recibir la información de entrada es la cámara, se utiliza IA para obtener información útil para el sistema. En este caso, es relevante descubrir cuál es la orientación de la cara, el cuerpo y la mirada. Y para ello se utilizan librerías que estiman puntos en 3D a partir de puntos 2D de una imagen mediante ML (detector-tracker).

A partir de esos puntos se puede resolver la proyección de la cara, el cuerpo y la mirada para saber hacia dónde están orientados. Mediante lenguaje no verbal y en función de esas trayectorias, una posibilidad consiste en asignar manualmente un valor a cada una de esas proyecciones dándole más peso a la mirada que a la cara, y a la cara que al cuerpo. Sin embargo, en el presente TFG se ha optado por hacer uso de IA, concretamente NN's y SVM. De esta manera se consigue que la IA entrene con los datos que nos ofrece el sistema desarrollado, para que se puedan tener en cuenta relaciones no descritas que mediante programación pueden pasar desapercibidas.

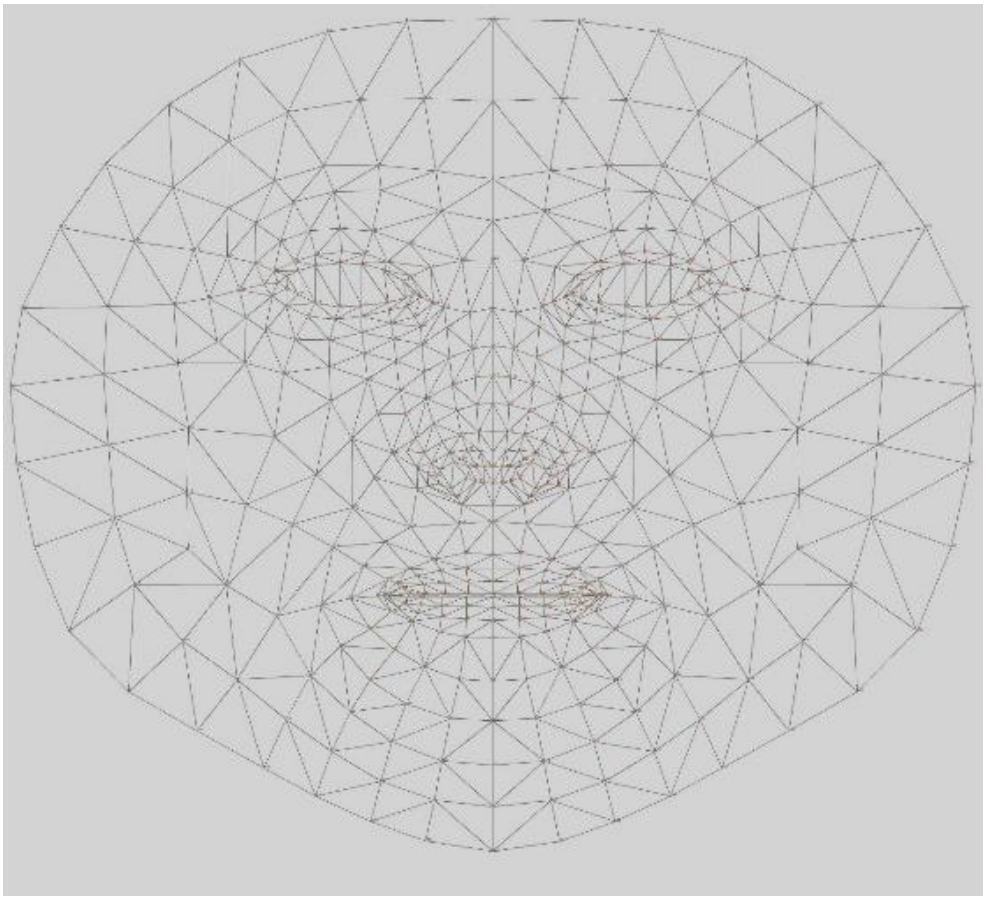

Figura 10 - Puntos de referencia Mediapipe FaceMesh. (Imagen obtenida de la web)

Con puntos significativos de la cara como la nariz, boca, cejas y barbilla (Figura 10) se puede aplicar el algoritmo solvePnP desarrollado en OpenCV para descubrir la orientación de la cara en función de la rotación de cada eje. Cada localización está numerada con un índice para poder trabajar con ella, tal y como se observa en la Figura 11. Como la rotación sobre los ejes aporta una información más intuitiva con la que poder trabajar, se realiza una transformación mediante la función RQDecomp3x3 de OpenCV.

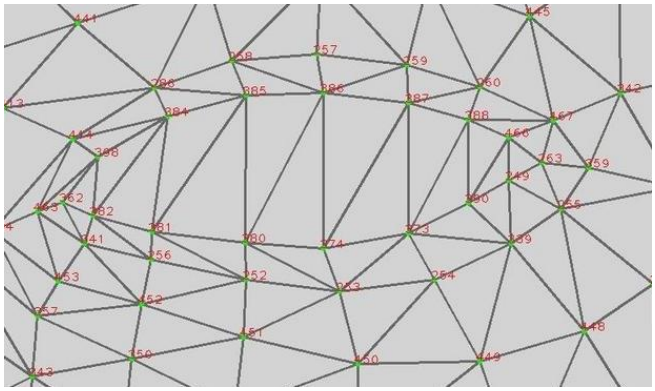

Figura 11 - Ampliación de los puntos de referencia del ojo izquierdo Mediapipe FaceMesh. (Imagen obtenida de la web)

Como se explicó anteriormente, esta función transforma la matriz de rotación obtenida a ángulos de Euler. Como solvePnP devuelve una lista de vectores (de Python) en vez de una matriz, se necesita realizar un paso intermedio con la función "Rodrigues" de OpenCV que convierte esa lista en una matriz de rotación. La necesidad del uso se esta función junto con solvePnP para poder trabajar con ángulos de Euler aparece explicado en el video "*Head Pose Estimation with MediaPipe and OpenCV in Python*" (Nielsen N., 2021).

La detección de la pose del cuerpo se realiza de una manera similar, pero con un inconveniente ya que, debido a la distancia entre la cámara y la persona con la que interactúa, únicamente se ven en la imagen dos puntos del torso, los hombros. Estos puntos corresponden a los puntos número 11 (hombro izquierdo) y 12 (hombro derecho) de la Figura 12. Ahí se aplica la proyección del segmento que une esos puntos sobre el plano XZ (horizontal), y se obtiene el ángulo de rotación sobre el eje Y.

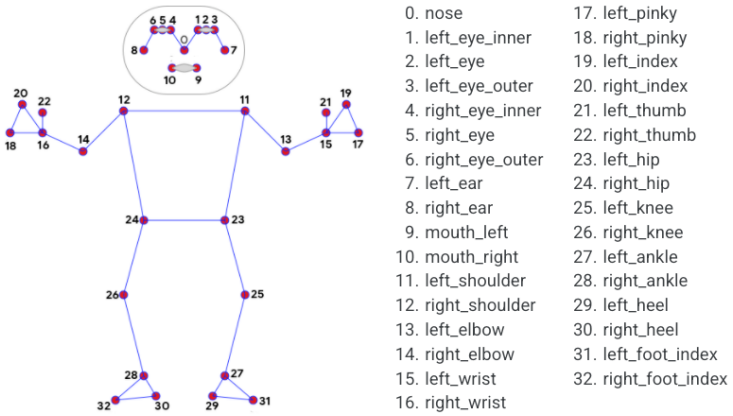

Figura 12 - Puntos de referencia Mediapipe Pose. (Imagen obtenida de la web)

En cuanto a la orientación de la mirada, se ha planteado usar los puntos del iris y la pupila, pero la calidad de la cámara y la iluminación no siempre permiten estimar correctamente estas posiciones del cuerpo. Por la pequeña distancia entre los puntos dentro de la imagen, una mínima variación supone un gran cambio en los ángulos de la proyección. Y como añadido, el uso de gafas distorsiona ligeramente la imagen de los ojos captada por la cámara provocando aún más error. Debido a la inestabilidad en la obtención de información mediante la mirada se decide no contar con este parámetro.

4.5. Mediapipe, TensorFlow y Dlib

Además de OpenCV, existen múltiples librerías para realizar tareas como detectar caras y obtener diferentes puntos del rostro y el cuerpo. A

continuación, se describen OpenCV y otras 3 librerías que se han usado durante la creación del sistema.

OpenCV es una librería dedicada al tratamiento de imágenes, aunque posee alguna función para entrenar alguna IA. Es cierto que ofrece varias herramientas para localizar caras y tiene funciones para trabajar con ellas, pero no obtiene tanta información como otras librerías. Como el punto fuerte de esta librería es el uso de herramientas para manejar diferentes canales de imagen y modificarlas, se suele complementar con otras librerías más especializadas y se utiliza OpenCV únicamente para editar las imágenes.

Mediapipe es una librería de ML de código abierto que admite el uso de funciones que permiten detectar objetos y personas. La ventaja de esta librería es que posee una gran cantidad de funciones. Tal como se indica en la web de los desarrolladores (Descripción de herramientas de Mediapipe. Google Github) esta librería es capaz de reconocer diferentes partes del cuerpo como la malla de la cara, puntos de las articulaciones, puntos de las manos, etc.

El inconveniente principal es que no posee herramientas más allá de detectar esos puntos característicos de la cara y el cuerpo. Cualquier información que se quiera obtener de esos puntos hay que procesarla posteriormente mediante programación.

TensorFlow también es una librería de IA, pero su campo es mucho más extenso que la identificación de puntos característicos. Se puede usar esta librería para entrenar cualquier conjunto de datos de entrada con una gran variedad de herramientas. Es decir, se puede entrenar una IA para hacer todo lo que hacía Mediapipe, pero habría que hacerlo desde el principio, obteniendo una base de datos, entrenando la IA, eligiendo el modelo más conveniente e implementándolo.

En cuanto a la localización de la información del cuerpo es mucho mejor Mediapipe, ya que posee modelos previamente entrenados por personas que han dedicado tiempo a que se obtenga el mejor resultado, y, además, su programación es mucho más sencilla. TensorFlow ofrece una gran cantidad de herramientas de IA, pero su uso se ha destinado a otras funciones como la evaluación de HRE que se comentará más adelante.

Dlib también permite localizar puntos de la cara y utilizar funciones para entrenar una IA, pero no reconoce puntos del cuerpo como realiza Mediapipe, y no posee tanta variedad de funciones de IA como TensorFlow. Estas razones podrían haber hecho que se descartara esta librería, pero existen otras como face-recognition, las cuales aportan unas herramientas potentes y de un uso sencillo para el reconocimiento de caras, que están construidas sobre Dlib y su utilización es indirecta.

Cada una de estas librerías tiene campos que se pueden aprovechar en este proyecto, pero no hay ninguna que realice todas las funciones que se buscan. En principio con TensorFlow se podrían entrenar diferentes IA para generar la misma información que las otras librerías, pero no se obtiene beneficio creando algo nuevo cuando puedes complementar unas herramientas con otras. Por esta razón, se ha decidido aprovechar las fortalezas de cada una de estas herramientas en el proyecto y se han utilizado bajo el mismo sistema.

Como TensorFlow tendría que entrenarse y buscar un alto grado de eficiencia para detectar puntos de la cara y el cuerpo, mientras que Mediapipe ya tiene modelos eficientes previamente entrenados para este propósito, se ha elegido Mediapipe para obtener información relacionada con las poses humanas.

Por otro lado, como TensorFlow destaca en el entrenamiento de IA se ha utilizado únicamente para la evaluación de HRE.

Por último, la librería Dlib no se ha usado directamente en el proyecto debido a que es una herramienta con menos utilidades que Mediapipe y TensorFlow. Pero se empleará de manera indirecta con otras librerías basadas en Dlib que están más desarrolladas, que sirven para reconocer caras y emociones, mencionadas anteriormente.

4.6. Métodos para reconocer caras y emociones

Se han planteado diferentes librerías para reconocer caras. Cada una de ellas exponen diferentes ventajas e inconvenientes que se describen a continuación.

4.6.1. Face-recognition

En primer lugar, se ha utilizado una librería especializada en este ámbito llamada face-recognition cuyo único propósito es el reconocimiento facial.

Es una herramienta entrenada mediante DL. Como ya se ha comentado, DL está inspirado en las neuronas del cerebro, al igual que NN, pero con muchas capas intermedias. Al principio estas capas intermedias generan máscaras con patrones muy reconocibles tales como líneas rectas, rectas ligeramente curvadas, círculos, etc. Pero a medida que se profundiza, estas máscaras son más arbitrarias y sirven para discriminar rasgos de las caras que pertenecen a personas concretas.

Gracias a este método, se reconocen las caras a partir de una sola fotografía con un alto índice de aciertos, y su uso es sencillo. El inconveniente que tiene esta librería es que, en el equipo 1 (características detalladas en el Anexo I)

donde se han realizado las pruebas, tarda 0,5 segundos en reconocer cada cara.

Con un reconocimiento de las caras tan lento, en el peor de los casos en el que ha habido más de una cara, ha disminuido el rendimiento del sistema lo suficiente como para inutilizarlo. Pero esto ha sido resuelto realizando un seguimiento sencillo de la cara a partir de la nariz.

De esta forma, cuando se procesa una imagen y se localiza la posición de la nariz, se guarda su ubicación. En la imagen posterior, se calcula la posición de la nariz de la cara detectada y, si está dentro de un margen de una posición ya detectada, presupone que es la misma cara.

4.6.2. LBPH y FisherFaces

También se han analizado otros métodos situados dentro del campo de la visión artificial menos exigentes para el equipo como LBPH y FisherFaces.

Para aprovechar al máximo la rapidez de estos métodos, ambos se han utilizado en el sistema para reconocimiento facial y reconocimiento de las emociones.

Tras su uso, se ha comprobado un aumento significativo en la velocidad de todo el sistema. El primer inconveniente de ambos métodos es que requieren una base de datos de imágenes mucho mayor y en diversos escenarios, la cual es difícil de obtener. El segundo inconveniente es que son métodos muy antiguos, y sus resultados no son precisos ni en reconocimiento facial ni en reconocimiento de emociones, además de ser muy susceptibles a cambios de iluminación.

Como no realizan correctamente la función para la que han sido desarrollados han sido descartados del proyecto.

4.6.3. DeepFace

Aparte, se ha utilizado otra librería más moderna llamada DeepFace que también reconoce caras, y además estima emociones y edad, entre otras cosas.

La particularidad de esta librería es que, para comparar caras usa modelos entrenados previamente de diferentes librerías y compañías conocidas tales como: Google FaceNet, OpenFace, Facebook DeepFace, DeepID y DlibSFace.

La velocidad de esta librería es similar a face-recognition, en torno a los 0,5 segundos para detectar cada cara, y su precisión también presenta un alto índice de aciertos.

Todo ello es debido a que usa modelos de Dlib, al igual que face-recognition, y aunque los desarrolladores garantizan que algunos de sus modelos tienen mayor índice de aciertos, experimentalmente se ha comprobado que el modelo de Dlib era el más acertado.

En cuanto al reconocimiento de las emociones, no siempre se realiza correctamente si las condiciones no son adecuadas. Necesita detectar rasgos faciales con mucho más detalle y es susceptible a cambios en la iluminación, contrastes y calidad de la cámara.

4.6.4. Elección del método utilizado

Finalmente, tras haber descartado anteriormente LBPH y FisherFaces, se ha procedido a elegir qué librería utilizar. En DeepFace, su uso en Python no es costoso, pero no es tan sencillo como face-recognition, por esta razón se decide seguir usando face-recognition para el reconocimiento facial.

De igual manera, se aprovecha DeepFace para estimar las emociones por ser la librería que mejores resultados ha dado. Aunque como ya se ha indicado, necesita unas condiciones mínimas para funcionar correctamente las cuales se explican en el apartado de Resultados.

4.7. Evaluación HRE (SVM vs NN)

Para obtener la información anterior sobre rostros y emociones existían librerías con modelos entrenados previamente, pero para medir el nivel de participación no. Esta parte entra dentro de los objetivos de este proyecto.

Parametrizar la información recibida de manera manual, indicando hasta que nivel la persona está atenta según una serie de combinaciones de esos datos, puede no ser del todo exacto. Ésta es la razón por la que se decide entrenar dos IA's con algoritmos diferentes, para que encuentren relaciones entre esos parámetros que los humanos no detectan a simple vista, y después las comparen. Para entrenar ambas IA's se recogen datos de entrenamiento y de validación, y después se testea directamente sobre el sistema implementado en diferentes escenarios (muy atento, atento, despistado, ...).

Cada una de estas situaciones se ha registrado en video, y al mismo tiempo se ha guardado información tal como la orientación sobre los ejes X e Y de la cabeza, la orientación sobre el eje Y del torso, la apertura de la boca, y las

pulsaciones del teclado durante la sesión de grabación de datos. Dicha información ha sido guardada en diferentes archivos Excel que han servido como base para entrenar los modelos de aprendizaje automático. En las figuras que se muestran a continuación se ha descrito qué se ha tenido en cuenta para asignar a cada video dentro de una categoría.

En la Figura 13 se observa a una persona con una orientación sobre los ejes X e Y de la cara centrados en la imagen, y del mismo modo, el torso también aparece centrado en la misma, haciendo que parezca que se muestra un gran interés hacia lo que se tiene en frente. Y aunque no se pueda percibir en la imagen, durante la interacción se ha grabado la apertura de la boca con un cambio frecuente en la amplitud, imitando una conversación fluida, y realizando pulsaciones sobre el teclado, imitando el contacto con cualquier periférico de entrada.

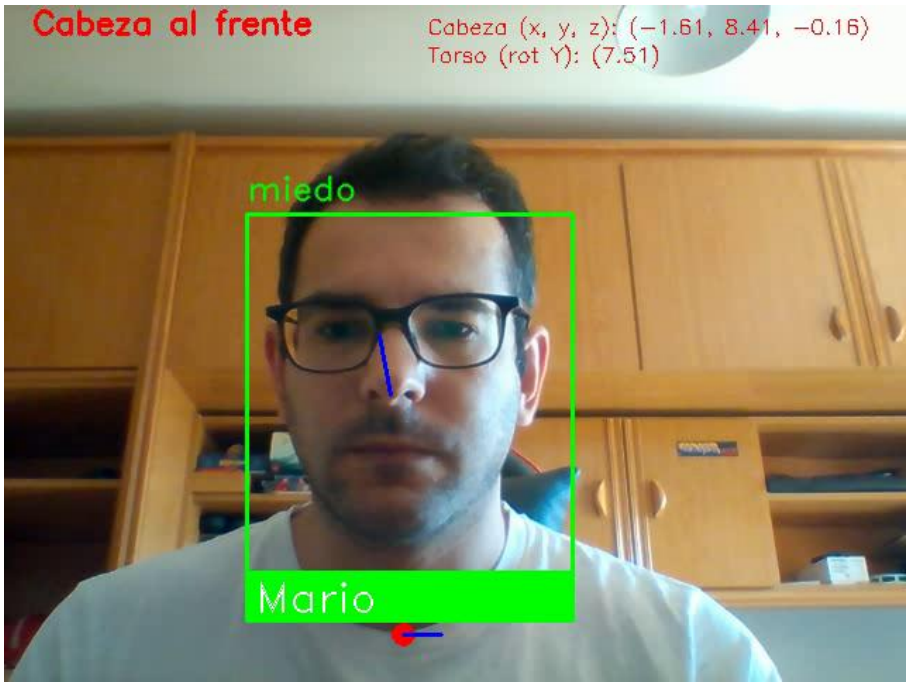

Figura 13 - Datos de entrenamiento "muy atento".

En la Figura 14 se observa una figura parecida a la anterior, pero se puede apreciar que el torso está hacia un lado, y se ha reducido la frecuencia con la que se mueve la boca, y con las que se realizan las pulsaciones del teclado. Esta situación imita a una persona que sigue mostrando un gran nivel de atención, pero que no está concentrado al máximo en la interacción.

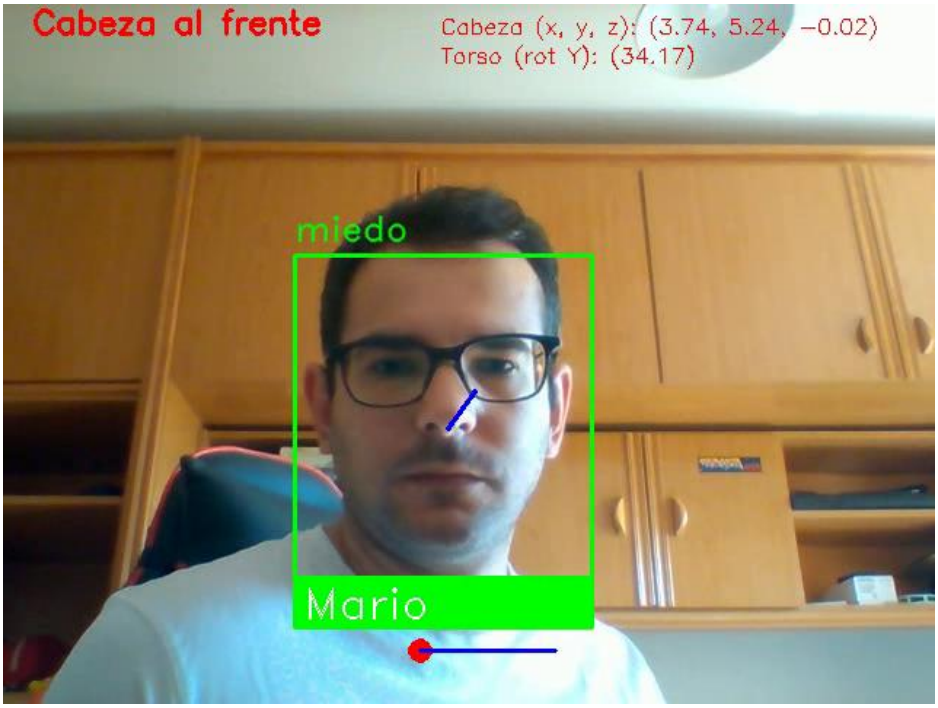

Figura 14 - Datos de entrenamiento "atento".

El contexto de la Figura 15 muestra cómo las orientaciones del cuerpo y la cabeza están separadas del centro de la imagen, y la frecuencia de las pulsaciones del teclado y el movimiento de la boca son parecidas a la situación de la figura anterior. Con este escenario se imita a una persona que está cerca de perder la atención, pero continúa mostrando interés.

Figura 15 - Datos de entrenamiento "poco atento".

En la Figura 16 se puede observar cómo las orientaciones de la cara y el cuerpo están también alejadas del centro de la imagen, y, además, ya casi no se realizan pulsaciones sobre los periféricos de entrada, ni se mueve la boca

frecuentemente como se hacía en las anteriores situaciones. Este contexto pretende imitar un nivel muy bajo de concentración sobre la máquina, donde no se la presta mucha atención, pero aun así no llega a alcanzar el mínimo.

Figura 16 - Datos de entrenamiento "despistado".

Por último, en la Figura 17 se muestra una posición completamente apartada del centro de la imagen, donde no se realizan pulsaciones en el teclado, representando una completa falta de interés del humano hacia la máquina.

Figura 17 - Datos de entrenamiento "muy despistado".

Se debe aclarar el hecho de admitir que una situación interpretada como “atento”, sea evaluada por otras personas como “muy atento” o “ligeramente atento”. Por esta razón, se evalúa este modelo según la precisión y validez de la respuesta que devuelve.

4.7.1. Redes Neuronales (Neural Network - NN)

Una de las IA’s entrenadas fue una NN mediante la librería Keras de TensorFlow, que está especializada en este campo. Keras ofrece una gran magnitud de opciones para entrenar un modelo, pero no todas ellas son válidas para entrenar el modelo buscado.

Para localizar qué parámetros funcionan mejor en la preparación del modelo, lo primero que se ha hecho son una serie de pequeños entrenamientos y validaciones para ver qué parámetros respondían mejor a estos datos. Aunque este método permite evaluar muchas opciones rápidamente, no ha sido muy concluyente. Dada la naturaleza aleatoria de los entrenamientos del método, con pequeños aprendizajes, algunos parámetros daban buen resultado por encima de otros que deberían funcionar mejor.

En este caso, la mejor solución es entrenar y contrastar los resultados de cada grupo de opciones preparándolos el tiempo suficiente como para que los datos sean relevantes. Pero esta opción se descarta debido a la gran cantidad de combinaciones que existen y el tiempo que tardaría en realizar cada una de ellas.

Finalmente se ha optado por elegir los parámetros de una forma razonada. Entre estos parámetros se ha seleccionado el número de capas densas y el tipo, la tasa de abandono de neuronas (dropout), y la métrica. Empezando por el más sencillo, la métrica es la función con la que se mide el rendimiento de cada modelo e indica cuál ha sido el mejor. Existen diferentes métricas disponibles en función del objetivo buscado tales como Accuracy, Probabilistic, Regression, True/False, Image segmentation y Máximum-margin classification.

Para cuantificar cuál ha sido el mejor modelo se ha propuesto usar un método como Accuracy, que permite obtener el mejor modelo de entrenamiento encontrado basándose en cuál de ellos devuelve unos resultados más exactos a los esperados. El resto de las agrupaciones sirven para otras tareas como devolver las probabilidades de que algo ocurra, la relación entre las variables de entrada, obtener resultados verdaderos o falsos, etc.

Dentro de la categoría Accuracy existen diferentes subgrupos tales como los que se pueden observar en la Tabla 2. Para el uso en este trabajo donde las etiquetas no son binarias y se busca únicamente con qué frecuencia los

resultados son correctos, se ha hecho uso de la clase con el mismo nombre que la categoría a la que pertenece (Accuracy).

Clase	Descripción
Accuracy	Esta clase sirve para indicar con qué frecuencia las predicciones equivalen a alguna etiqueta.
BinaryAccuracy	El resultado que devuelve muestra con qué frecuencia las predicciones equivalen a etiquetas binarias.
CategoricalAccuracy	Parecido a Accuracy, pero el resultado devuelve un vector con ceros en sus valores y con un uno en la predicción.
SparseCategoricalAccuracy	También es semejante a Accuracy, pero el resultado en este caso es la posición indexada de la clase que considera verdadera.

Tabla 2 - Opciones de la métrica Accuracy

Respecto a las capas de la NN se han usado diferentes grupos de capas densas con diferentes características. Una capa es densa cuando todas las neuronas de dicha capa están conectadas con todas las neuronas de la siguiente, provocando que se tengan más datos con los que trabajar. Por el contrario, también produce que sea más lento a la hora de procesar datos.

En la programación se han usado 3 bloques de este tipo, tal y como se observa en Figura 18. La primera ha sido una capa profunda de 256 niveles, y 5 entradas. Cada nodo de entrada corresponde a cada una de las clases que se manejan en los datos de entrenamiento (muy atento, atento, etc). La segunda ha usado 128 niveles, con una sola entrada alimentada por la capa anterior, reduciendo así el número de combinaciones y acelerando el entrenamiento.

Entre la segunda y la última capa densa se ha utilizado un bloque de abandono (*dropout*), cuya utilidad es descartar aleatoriamente algunas conexiones de la red neuronal y evaluar si se obtienen mejoras en los resultados. En caso de no haber ninguna mejora, se deshace el *dropout* y se mantiene la conexión.

Por último, se ha utilizado una capa densa de tan solo 5 niveles, pero con una característica a mayores, el tipo de activación es 'softmax'. Este parámetro es usado frecuentemente porque convierte un vector de resultados en una distribución de probabilidad de la que se obtiene la predicción más aproximada.

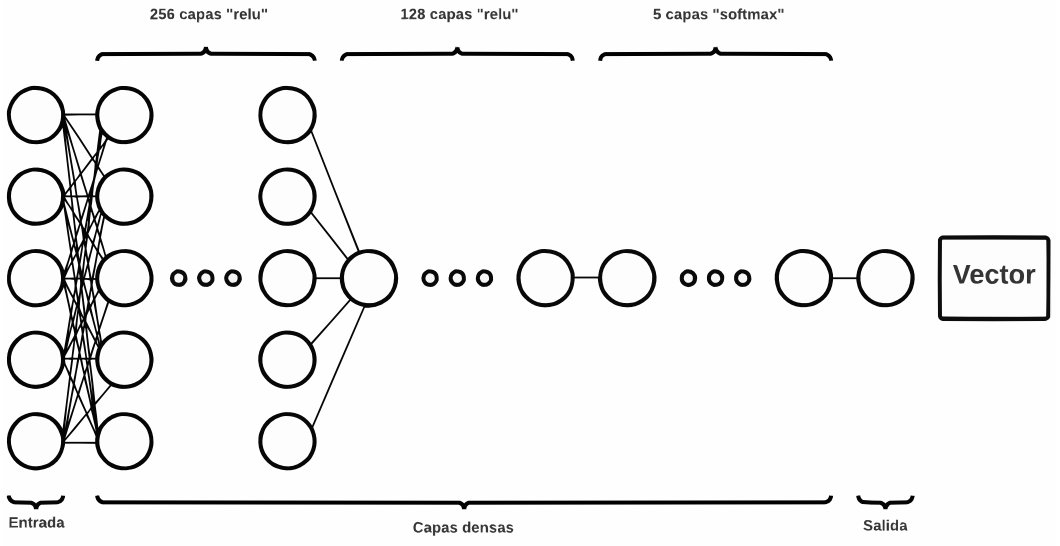

Figura 18 - Diagrama entrenamiento NN

En cuanto a la obtención del modelo, haciendo un aprendizaje muy profundo, tal y como se ha explicado, iterado 20.000 veces, es decir con 20.000 épocas, se ha obtenido un 75,12 % de resultados precisos y un 96,08 % de resultados válidos, tal y como se puede observar en la Figura 19. En el contexto de la IA se entiende por época el número de veces que se han recorrido las capas hasta alcanzar los pesos óptimos en cada nodo para que las predicciones sean cercanas a los resultados esperados.

```
Resultados validos: 96.08
Resultados precisos: 75.12
14/14 [==============================] - 0s 2ms/step - loss: 1.9385 - accuracy: 0.7512
test loss, test acc: [1.9384970664978027, 0.7511520981788635]
```

Figura 19 - Resultado del entrenamiento de NN (Keras)

Este entrenamiento se ha realizado en aproximadamente 40 minutos, un periodo de tiempo mucho menor que SVM, sin haber hecho falta entrenarlo con un gran número de combinaciones. Por este motivo, se demuestra que las NN son herramientas más potentes para entrenar con datos relacionados con la conducta humana.

Cabe destacar que no se habrían localizado resultados significativamente mejores, aunque se hubiera entrenado durante más tiempo. Tal y como se ve en la Figura 20, la precisión de los resultados ha alcanzado un estacionario, y para poder mejorarlos habría que probar con parámetros diferentes. Se ha

ensayado con otra serie de valores durante el entrenamiento, pero estos fueron los que obtuvieron mejores resultados.

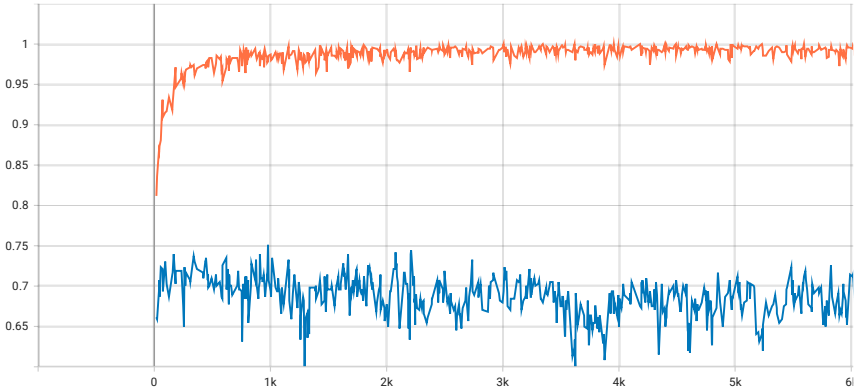

Figura 20 - Precisión en cada época del entrenamiento. Naranja (train), Azul (validation)

Tal y como se observa en la Figura 21, el mapa de calor de las NN da como resultado un comportamiento mucho más acertado que SVM. Siguiendo el criterio explicado anteriormente, en esta imagen se aprecian valores más precisos y mucho más válidos que en la Figura 23.

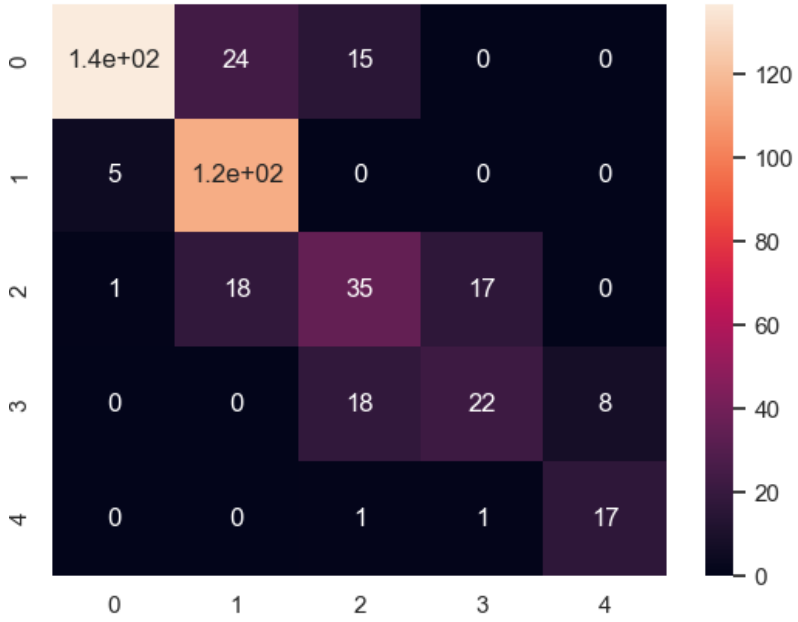

Figura 21 - Mapa de calor de respuestas precisas y válidas del modelo NN (Creado con Seaborn).

Aunque existen valores erróneos, en la programación del sistema se ha expuesto el valor más repetido de los 10 últimos. De esta manera se aíslan estos errores, se mantiene un resultado más cercano a la realidad y no se producen cambios constantes en el nivel de atención.

Como se mencionó con anterioridad, para el valor esperado “atento” es válido recibir “muy atento” o “ligeramente atento”. Como se ha podido observar en la

Figura 21, los resultados precisos pueden parecer bajos, pero hay que comprender la naturaleza subjetiva de los datos con los que se entrena. A partir de la información que prepara a la IA, si esta es interpretada por diferentes personas, los resultados pueden variar ligeramente entre ellas.

4.7.2. Máquinas de Vectores de Soporte (Support-Vector Machines - SVM)

Por otro lado, también se ha entrenado y validado una IA mediante SVM perteneciente a OpenCV. Se pueden crear modelos de esta IA utilizando diferentes parámetros que se describen en la Tabla 3.

Tipos de SVM

C_SVC	C-Support Vector Classification. La clasificación de la clase n, siendo n ≥ 2, tolera la división irregular de clases y posee un coeficiente C que sirve como multiplicador de penalización para los valores singulares.
NU_SVC	μ-Support Vector Classification. La clasificación de la clase n, también consiente la división irregular de clases. Para este tipo de SVM, se emplea el parámetro μ con un rango entre 0 y 1. Cuanto mayor sea dicho valor, más amplio será el límite de decisión.
ONE_CLASS	Estimación de distribución. Este tipo sirve para entrenar datos que pertenezcan a una única clase. En este caso, SVM crea un hiperplano que aísla la clase del resto del espacio.
EPS_SVR	Regresión vectorial de soporte ϵ. Este tipo limita la distancia entre los vectores de características del conjunto de entrenamiento, y el hiperplano de ajuste mediante el coeficiente p. Al igual, que en C_SVC, se hace uso del multiplicador de penalización C para valores atípicos.
NU_SVR	υ-Support Regresión vectorial. Este tipo también se parece a EPS_SVR, pero en este caso se utiliza υ en lugar de p.

Tabla 3 - Tipos de SVM

Para ejercitar esta IA se ha descartado usar algunos tipos como ONE_CLASS, que solo sirve para clasificar una clase en los datos de entrenamiento y en este caso hay varias, concretamente cinco. No se ha utilizado NU_SVC porque no interesa descartar datos de entrenamiento. Tampoco se han empleado ni EPS_SVR, ni NU_SVR, debido a que es necesario especificar límites entre los

vectores de características y el hiperplano. Por estas razones se ha decidido utilizar C_SVC que aprovecha todos los datos de entrenamiento.

Tipo de Kernel

También existen múltiples opciones para seleccionar el tipo de Kernel, aunque en este caso se ha seleccionado SVM_RBF. Entre los diversos ejemplos que existen, en la web de OpenCV aseguran que Radial Basis Function (RBF) es una buena opción en la mayoría de los casos.

Resultados del entrenamiento

Como la validación inicial de los datos de entrenamiento era demasiado baja (en torno al 40 %) se ha decidido crear un modelo de esta IA combinando los datos de diferentes maneras hasta dar con la mejor combinación de datos encontrados para trabajar. Hay que resaltar que la mejor combinación de datos encontrada no es la mejor posible. Solo realizando todas las composiciones se podría afirmar que se ha encontrado la mejor, pero eso llevaría una cantidad de tiempo muy alta.

Para este caso, se ha ejercitado SVM con un proceso que tarda 3 días en entrenar una gran variedad de combinaciones y se ha logrado un 68,2 % de resultados precisos, tal como muestra la Figura 22.

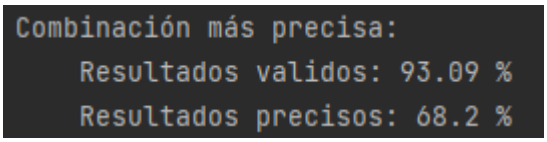

Figura 22 - Resultado del entrenamiento de SVM (OpenCV)

Como se ha explicado anteriormente, también se ha validado el programa obteniendo como resultados positivos un valor cercano al que se esperaba, dando como resultados “válidos” para SVM un 93,09 %.

La relación entre los resultados esperados y los obtenidos se localiza en un mapa de calor en la Figura 23. Tal y como se observa, el número de aciertos exactos y válidos es mucho más alto que los erróneos, aunque dichos resultados no son tan buenos como en NN (Figura 21).

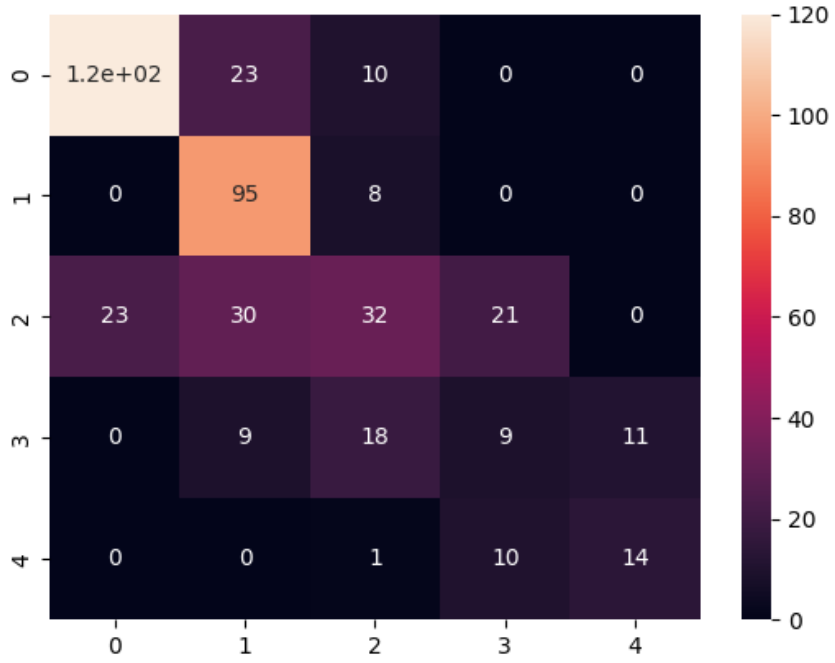

Figura 23 - Mapa de calor de respuestas precisas y válidas del modelo SVM (Creado con Seaborn)

Para alcanzar estos resultados han hecho falta 3 días de entrenamiento, pero no se ha planteado realizar más entrenamientos de SVM para mejorar los resultados anteriores ya que, el siguiente bloque de combinaciones para crear el modelo de esta IA tardaría alrededor de 30 días en completarse (aproximadamente 710 horas), tal como se puede observar en la Figura 24. Otra elección habría sido combinar los diferentes tipos de parámetros que se han seleccionado realizando entrenamientos suficientemente largos como para poder evaluar qué método es mejor, pero el tiempo necesario para ello es mucho mayor que la opción anterior.

```
Combinación [0, 1, 2, 3, 5, 7, 9, 4, 8, 6] testeada
Tiempo previsto para finalizar: 955 horas, 28 minutos y 31 segundos
Combinación [0, 1, 2, 3, 5, 7, 9, 6, 4, 8] testeada
Tiempo previsto para finalizar: 677 horas, 20 minutos y 57 segundos
Combinación [0, 1, 2, 3, 5, 7, 9, 6, 8, 4] testeada
Tiempo previsto para finalizar: 710 horas, 35 minutos y 9 segundos
Traceback (most recent call last):
  File "C:\Users\mario\Desktop\model_trainning\trainning_model_result.py", line 134, in <module>
    svm.trainAuto(descriptores_comb, cv2.ml.ROW_SAMPLE, trainY_comb)
KeyboardInterrupt

Process finished with exit code -1073741510 (0xC000013A: interrupted by Ctrl+C)
```

Figura 24 - Captura del tiempo que tardaría en completarse se dividieran los datos en 10 bloques.

4.7.3. Comparación entre NN y SVM

Aunque los resultados indican que las NN son las más indicadas para realizar esta función, se han incorporado los dos métodos en el sistema para poder observar si existe alguna diferencia en su funcionamiento.

Como se ha explicado con anterioridad, dentro de la programación de este sistema se elige el valor más repetido de las 10 últimas imágenes analizadas para poder obtener un resultado más estable. El grado de interés no cambia repentinamente en cada instante, y a una media de 3 FPS esas 10 imágenes abarcarían un pequeño intervalo de aproximadamente 3,33 segundos.

Experimentalmente se aprecia que ambas funciones emiten el mismo resultado, tal y como se aprecia en la Figura 25. En esta figura la línea del método con NN (verde) se corresponde con la del método SVM (roja).

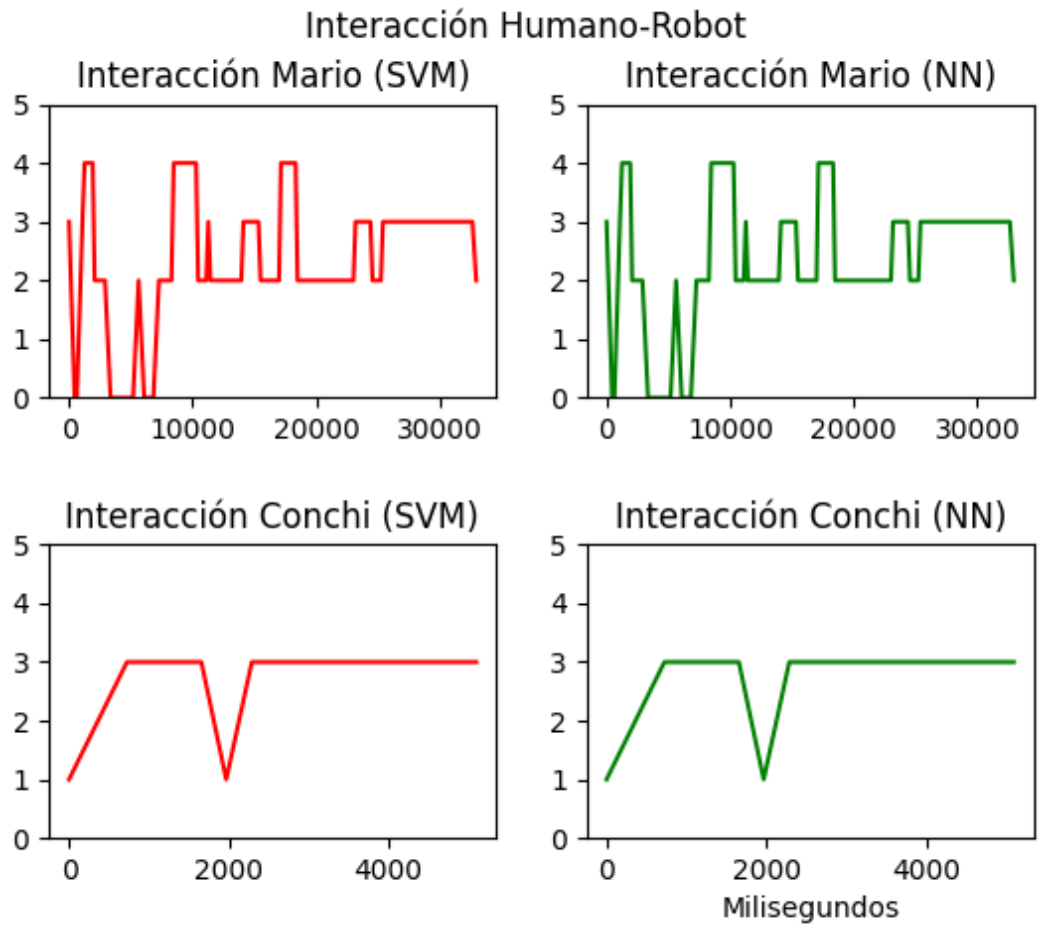

Figura 25 - Gráfico HRE con resultados superpuestos NN (verde) sobre SVM (rojo-no visible).

En relación con las IA's en general, y a las NN's en particular, hay que destacar que el uso de la GPU habría agilizado mucho el proceso de entrenamiento de

los modelos de NN. Estos componentes son capaces de acelerar enormes operaciones matriciales y realizar cálculos complejos en una simple operación ahorrando mucho tiempo de procesamiento. Pero tal y como se ha comentado en la Figura 20, con más tiempo de entrenamiento en NN no se habrían obtenido resultados significativamente mejores, la diferencia es que se podría haber obtenido el modelo en unos pocos minutos o incluso segundos.

5. Resultados

5.1. Condiciones para los ensayos

Existen varios factores que afectan al rendimiento del programa como son: la iluminación, el número de personas en la imagen, el equipo donde se encuentra implementado y la calidad de la cámara.

A continuación, se muestran una serie de pruebas realizadas en diferentes condiciones en las que se evalúan la precisión de los resultados y el rendimiento del sistema. Las características de los equipos utilizados se pueden encontrar en el Anexo I.

5.1.1. Equipo 1 utilizando cámara 1 (webcam integrada)

La cámara 1 es un dispositivo integrado en un portátil de mala calidad, y el equipo 1 es un ordenador portátil que presenta un rendimiento medio-bajo. Seguidamente, se exponen los resultados obtenidos tras haber realizado diferentes pruebas con ellos.

5.1.1.1. Variación de la iluminación

Se han hecho diferentes pruebas para observar cómo afecta la iluminación al procesamiento de la imagen. En la Figura 26 se puede observar claramente que, con una cámara de poca calidad, pero con una iluminación natural fuerte y uniforme, todos los elementos del sistema recogen los datos correctamente.

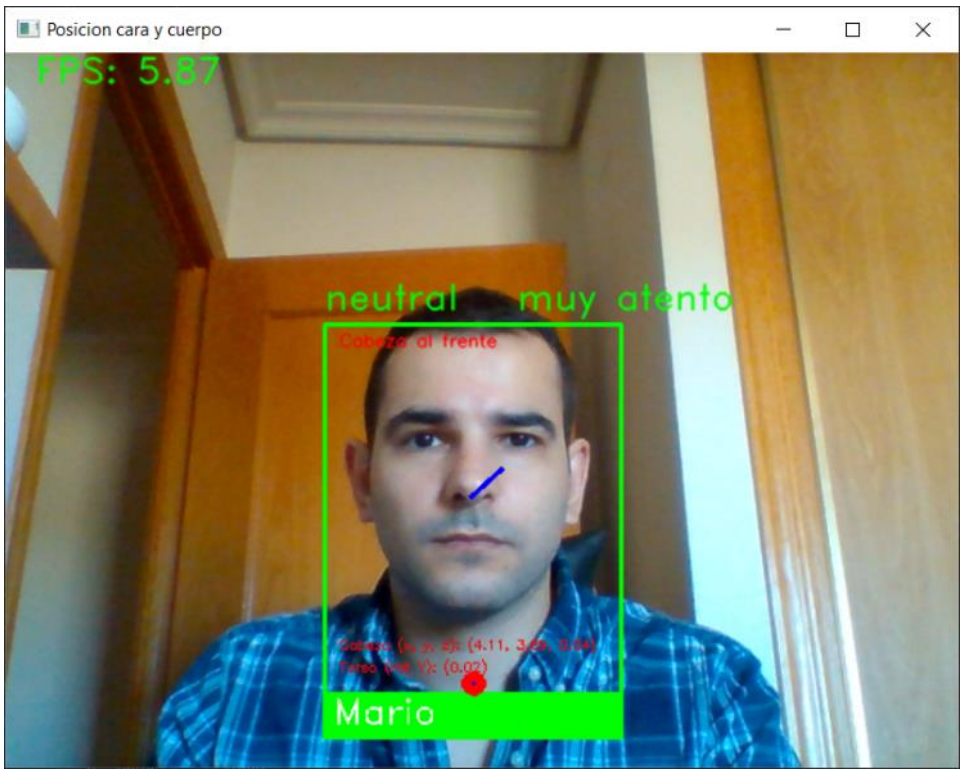

Figura 26 - Imagen del Equipo 1 con la cámara 1 - luz de frente.

En dicha figura aparece “Mario”, con una actitud neutral y con una posición completamente enfrentada a la cámara, lo que da a entender que tiene un alto grado de atención hacia el equipo que tiene en frente. En la propia imagen se puede apreciar cómo el sistema identifica correctamente el nombre y la emoción, y en la zona superior derecha del cuadro verde aparece el nivel de atención como “muy atento”, el grado más alto.

Aunque en esta situación el sistema funciona tal y como se espera, hay que dejar claro que esta es una situación casi ideal donde la luz solar está repartida uniformemente por la escena, y no crea grandes contrastes en la cara u otras zonas del cuerpo.

A continuación, se comparan otras situaciones posibles donde los escenarios no son tan favorables, como, por ejemplo, que la luz natural incida por el lateral de la imagen, tal y como se aprecia en la Figura 27.

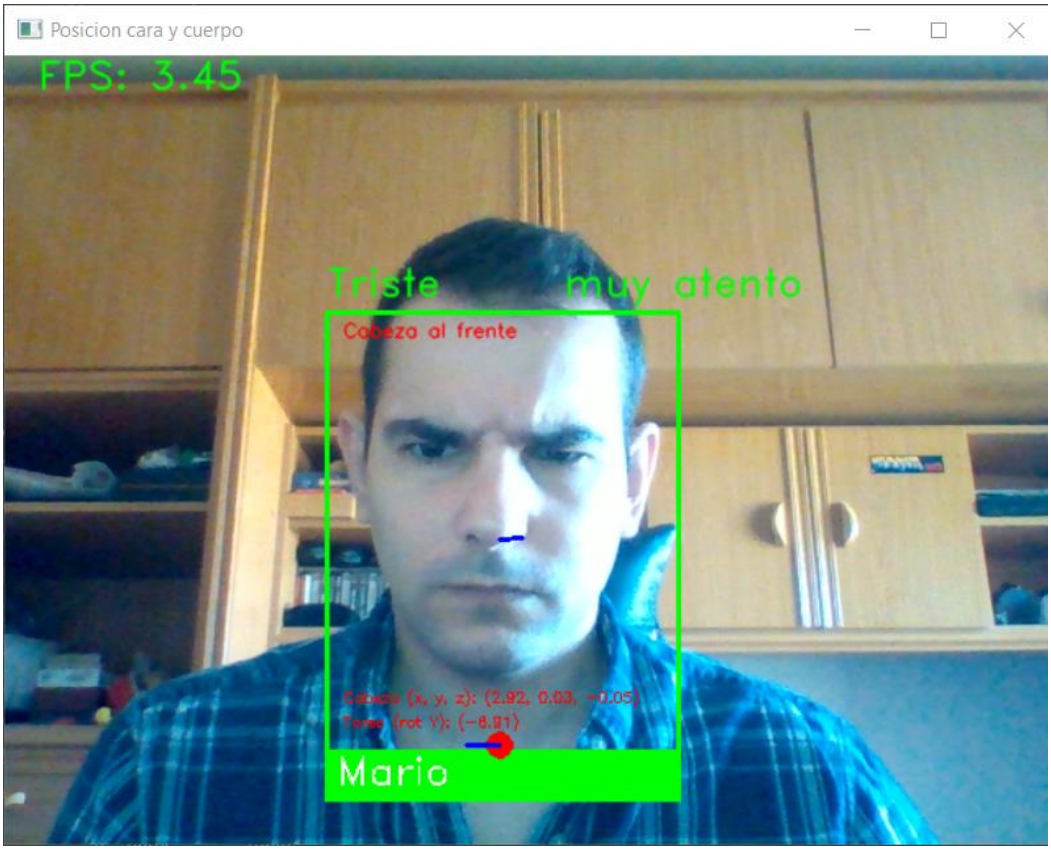

Figura 27 - Imagen del Equipo 1 con la cámara 1 – luz lateral

En este caso la cara tiene un gran contraste. Aunque la identificación de “Mario” sigue siendo correcta, se observa que el gesto de la cara es

"enfadado", mientras que el resultado de la estimación de la emoción es "triste". Esto hace comprender que la herramienta DeepFace encargada de evaluar el sentimiento que expresa la cara es sensible a los contrastes provocados por la luz. Por otro lado, el reconocimiento facial (face-recognition) es resistente a la presencia de contrastes en las imágenes.

En cuanto al nivel de interacción, parece que recoge bien las proyecciones del cuerpo y los modelos entrenados consiguen generar una respuesta válida ("muy atento").

Para poner a prueba la firmeza de face-recognition se prueba con un caso en el que una luz artificial genera sombras en la cara (luz por encima de la cabeza), tal y como se puede observar en la Figura 28.

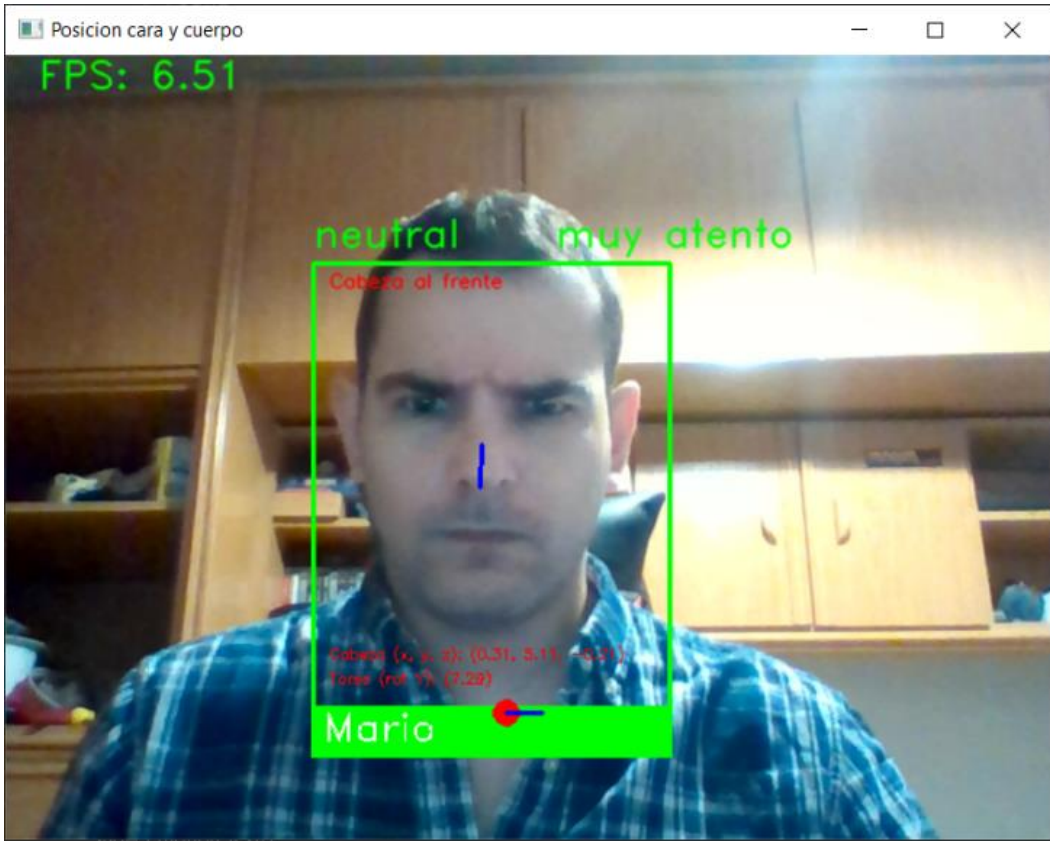

Figura 28 - Imagen del Equipo 1 con la cámara 1 – luz encima de la cabeza.

En este caso, el nombre sigue siendo correcto y la estimación del nivel de interacción también, por tanto, se puede afirmar que, tanto face-recognition como los modelos encargados de evaluar el nivel de la interacción han superado la prueba de cambios en la luminosidad y contrastes.

Por otro lado, DeepFace sigue teniendo problemas en entornos con luminosidad irregular. En este caso, el rostro muestra un gesto “enfadado” y la emoción que predice es “neutral”. En este punto, no se ha descartado el uso de esta herramienta porque estas son las peores situaciones en las que se puede encontrar, y aún existen otros escenarios donde evaluar su funcionamiento.

Para poder evaluar si el problema del reconocimiento de las emociones son los contrastes o el nivel de iluminación se realiza un experimento con un nivel de luminosidad muy baja (habitación a oscuras). La única luz que existe es la de la propia pantalla del ordenador que ilumina mínimamente el rostro tal y como se puede observar en la Figura 29.

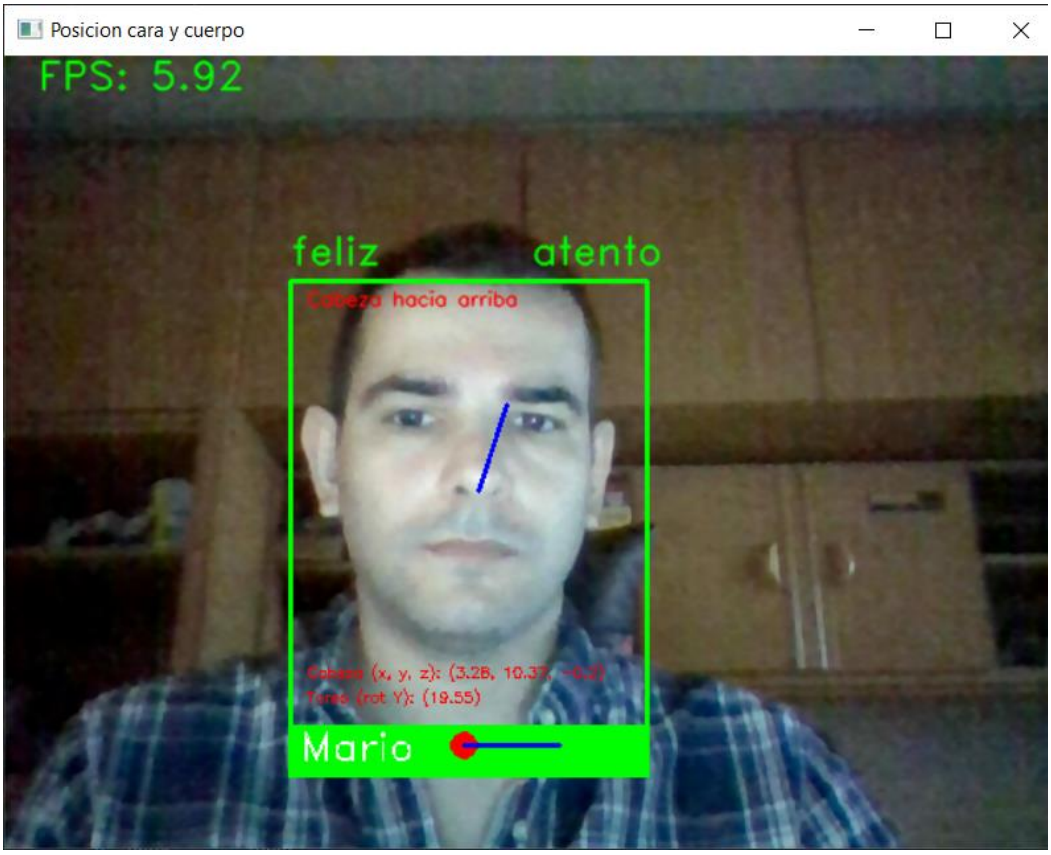

Figura 29 - Imagen del Equipo 1 con la cámara 1 – Iluminación muy baja, con luz residual de la pantalla.

Esta iluminación no produce contrastes sobre el rostro, aunque la iluminación es muy baja. En la Figura 29 se puede observar cómo la imagen posee una cantidad de ruido considerable y, aunque el gesto de la cara es “neutral”, el resultado es “feliz”. De este experimento se obtiene como resultado que, la

incorrecta iluminación afecta negativamente al sistema incluso cuando no existen zonas con contrastes o zonas con mucho brillo que borren información de la imagen.

Como la iluminación es un elemento crítico en este y otros sistemas que utilizan visión artificial, se ha intentado solventar dicho problema con un foco que contiene una bombilla led y una pantalla traslúcida, imitando a los estudios de fotografía. Esta pantalla se encarga de difuminar uniformemente por la escena los rayos de luz generados por el foco dando un nivel mínimo de luminosidad y reduciendo los contrastes.

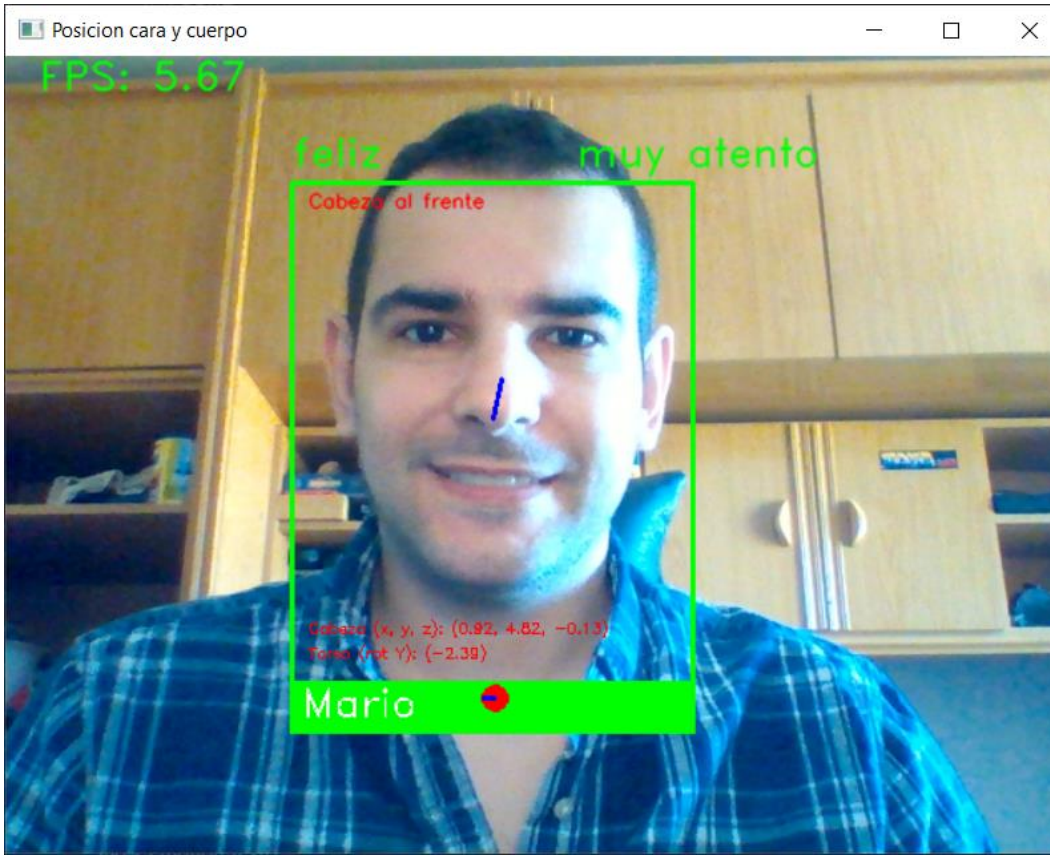

Figura 30 - Imagen del Equipo 1 con la cámara 1 – Luz lateral con foco difuminado

En la Figura 30 las facciones de la cara se suavizan y no hay tanto contraste como en la Figura 27. Esta luz es intensa, pero suavizando las zonas más oscuras se reduce el contraste y mejora la imagen lo suficiente como para que deepFace funcione correctamente.

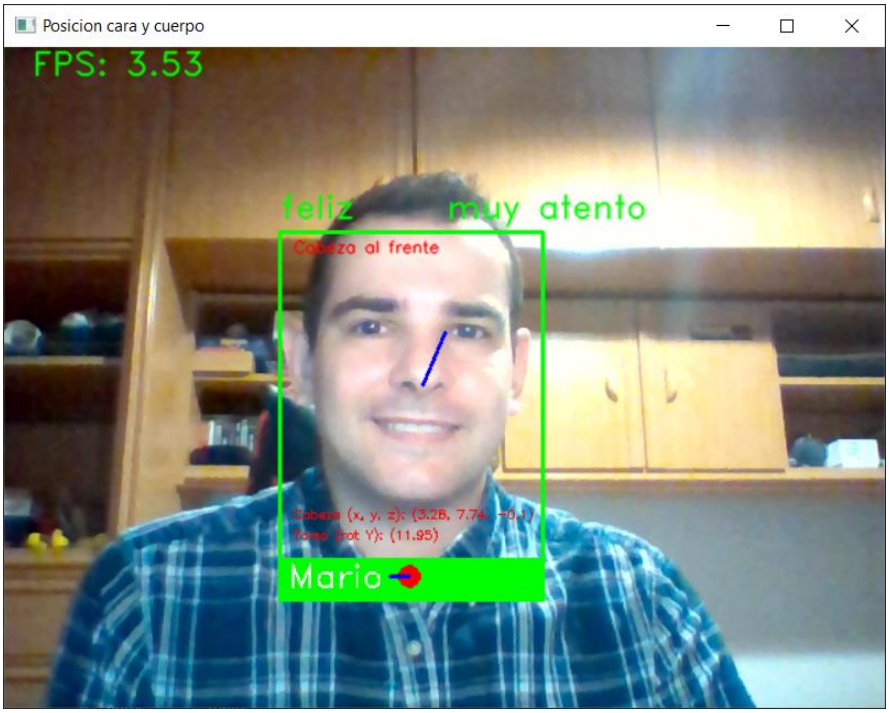

Figura 31 - Imagen del Equipo 1 con la cámara 1 – Luz encima de la cabeza con foco difuminado.

De igual manera, como se puede observar en la Figura 31, al añadir un foco difuminado, los contrastes producidos por la luz detrás de la cabeza disminuyen, y se obtiene un resultado correcto del detector de emociones.

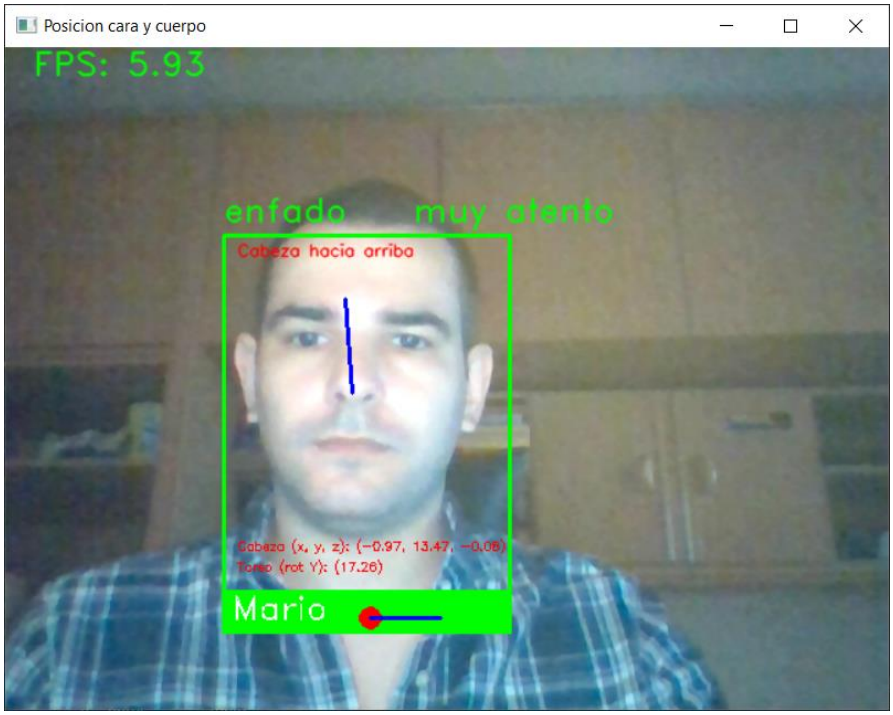

Figura 32 - Imagen del Equipo 1 con la cámara 1 – Iluminación muy baja con foco difuminado

En la Figura 32 se observa cómo, haciendo uso del foco en las mismas condiciones de iluminación que la Figura 29, el resultado sigue siendo equivocado. En la imagen se sigue apreciando ruido, aunque en menor proporción, y en unas condiciones con tan poca luz, un foco alumbrando directamente la cara puede provocar resultados igualmente negativos.

Esto hace evidente que sea necesaria una luz ambiental mínima para que la herramienta DeepFace funcione correctamente cuando la calidad de la cámara es baja. El foco ayuda a mejorar dicha luz reduciendo contrastes, pero no sirve como única fuente de iluminación.

5.1.1.2. Variación del rendimiento según el número de personas

En la Figura 28 se puede observar que los FPS se alcanzan el valor de 6,51. Esto quiere decir que puede llegar a procesar 6,51 imágenes por segundo cuando aparece una sola persona.

Para contrastar si el sistema de seguimiento es efectivo se ha introducido a otra persona en la imagen. De esta manera, se ha comparado cuánto afecta al rendimiento el procesamiento de la información de dos personas teniendo en cuenta que no solo es afectado por el reconocimiento facial, sino por todos los procesos como cálculos y predicción del HRE de cada una de ellas.

Esta comparación se puede establecer entre la Figura 33, donde aparecen dos personas “Conchi” y “Mario” y el rendimiento del sistema cae a 3,33 FPS, y la Figura 28, en la que únicamente se encuentra “Mario”.

Esto se produce porque el sistema tiene que analizar a dos personas con toda la información que obtiene de ellas en vez de a una sola. Y, además, en este caso donde se exponen los datos, también debe escribir el doble de parámetros. Por esta razón queda demostrado que el número de personas a analizar en la imagen es un parámetro que se debe tener en cuenta a la hora de utilizar este sistema.

Tal y como se ha mencionado anteriormente, 2 imágenes por segundo es igual a decir que se procesa una imagen cada 0,5 segundos. En un intervalo de tiempo tan amplio, un gesto rápido con la cabeza puede provocar que el sistema de seguimiento de las personas que se han reconocido se pierda. En ese caso, el sistema tendría que volver a reconocer quién es esa persona, con su correspondiente tiempo de procesamiento, y su pérdida de rendimiento. Además, existe una situación más crítica, donde el puntero de seguimiento de una persona sea localizado por el umbral de otra y se mezclen los resultados.

Esta situación solo duraría hasta que se volviera a reconocer a la persona con la identificación real, pero es un límite de mal funcionamiento que hay que tener en cuenta.

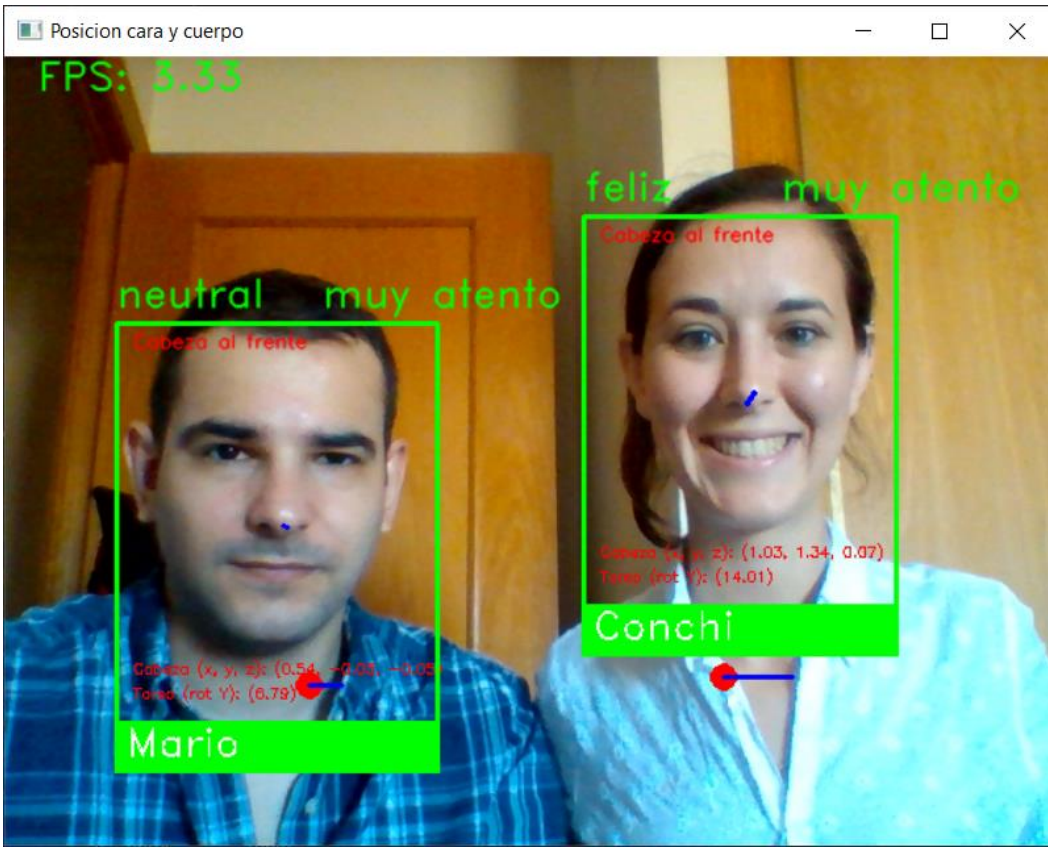

Figura 33 - Imagen del Equipo 1 con la cámara 1 – Participación de dos personas en la misma imagen.

De vuelta a la Figura 33, hay que tener cuidado con alcanzar ese límite inferior de mal funcionamiento, pero a 3,33 FPS aún se percibe una imagen fluida. Esto significa que todavía hay un margen hasta alcanzar ese umbral.

5.1.2. Equipo 1 utilizando cámara 2

Para saber si DeepFace puede funcionar en entornos con variaciones de luminosidad, se han realizado pruebas con una cámara de mayor calidad (de móvil) donde la lente de la cámara está más preparada para diferentes escenarios, y el procesamiento de la imagen es mejor que el que se consigue con una webcam de mala calidad.

De este modo, compensaremos la falta de luz y los contrastes con resoluciones más altas y con un procesado de la imagen mejor por parte de la cámara. Además, también se ha tenido en cuenta en este análisis si un aumento del tamaño de las imágenes afecta al rendimiento del sistema.

5.1.2.1. Variación de la iluminación

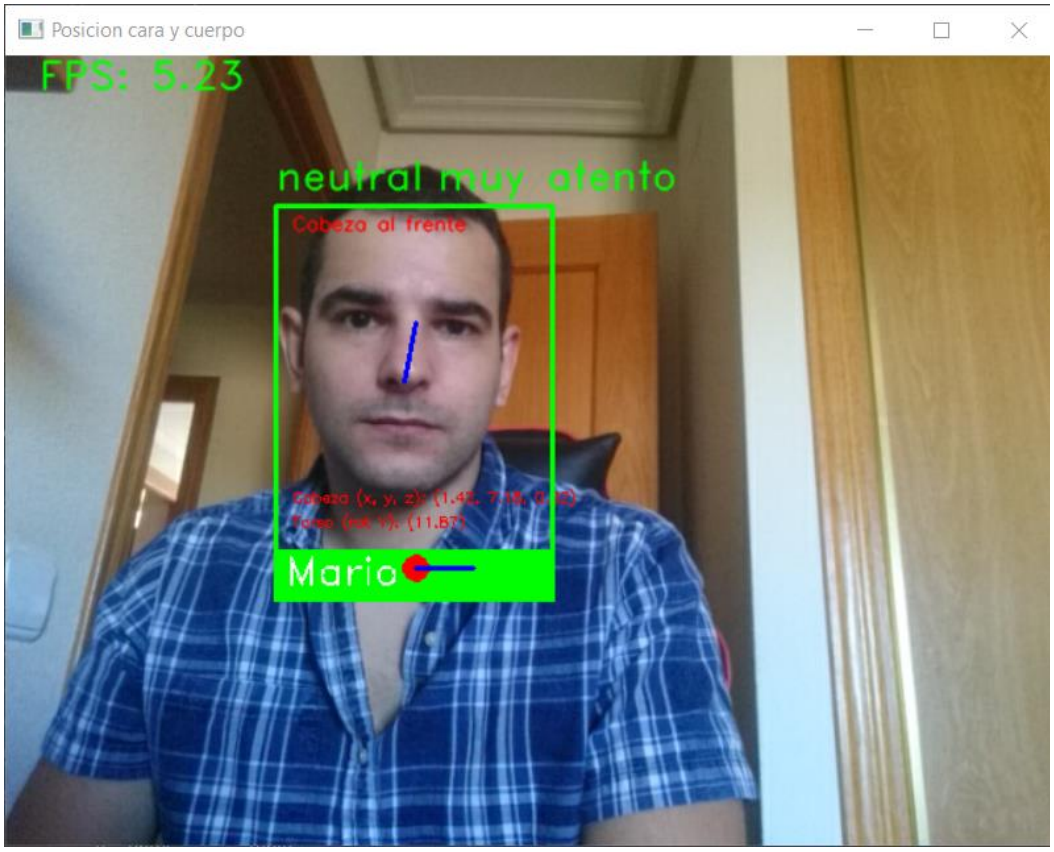

Figura 34 - Imagen del Equipo 1 con la cámara 2 – luz de frente.

En la Figura 34 se sigue observando el correcto funcionamiento del reconocimiento de caras y la estimación del nivel de la interacción, que dan como resultados “Mario” y gesto “neutral”. Como estas condiciones son aún más ideales que las de la Figura 26, DeepFace vuelve a realizar una buena valoración.

Tal y como se esperaba, el análisis de la participación del humano con la máquina devuelve “muy atento”, resultado que sigue siendo correcto debido a que las condiciones del entorno han mejorado.

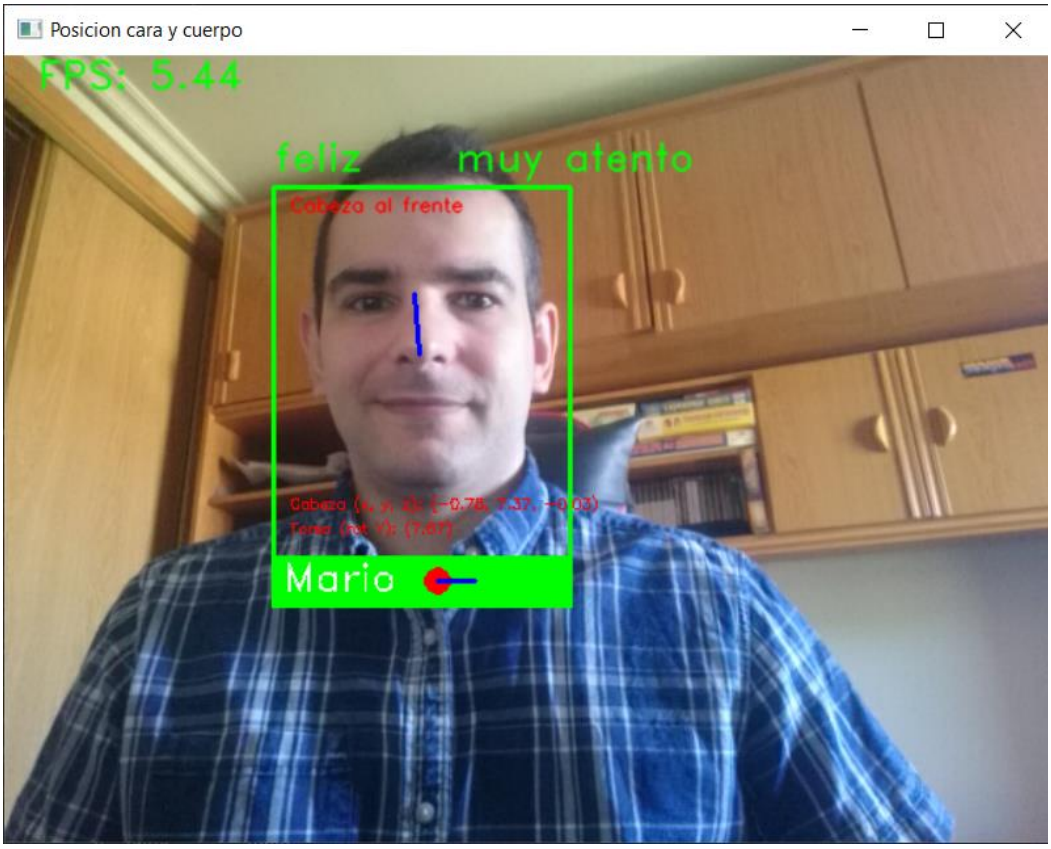

Figura 35 - Imagen del Equipo 1 con la cámara 2 – luz lateral.

En la Figura 35 se percibe cómo el aumento en la calidad de la cámara ha supuesto una mejora en los resultados de DeepFace para analizar las emociones. En este caso la valoración de “feliz” es acertada.

A diferencia de la Figura 27, una mejora en la calidad de la cámara ha corregido notablemente los contrastes en la cara lo suficiente como para que DeepFace dé buenos resultados. En este caso también se produce un funcionamiento correcto del resto de resultados.

En cuanto a la valoración de la Figura 36 respecto al cambio de luminosidad, se reconoce que el resultado de haber mejorado la calidad de la cámara no ha sido significativo en condiciones de luminosidad bajas. Al igual que la Figura 28, ésta mantiene unos resultados de los diferentes parámetros de salida erróneos.

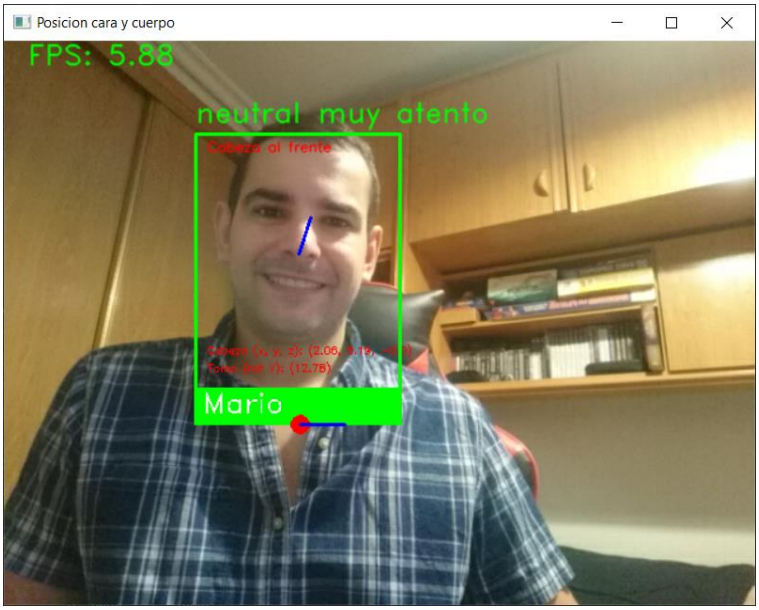

Figura 36 - Imagen del Equipo 1 con la cámara 2 – luz encima de la cabeza.

En dicha figura se observan las sombras irregulares que pueden deformar el rostro con las que se asocian estos resultados incorrectos. Un mejor procesamiento de la imagen con una cámara de mayor calidad podría solucionar este problema, pero no se dispone de una cámara de una calidad superior con la que hacer pruebas.

Finalmente, en la Figura 37 se puede observar una situación completamente a oscuras donde el resultado del reconocimiento facial sigue siendo erróneo, aunque el ruido de la imagen mejora respecto a la Figura 29.

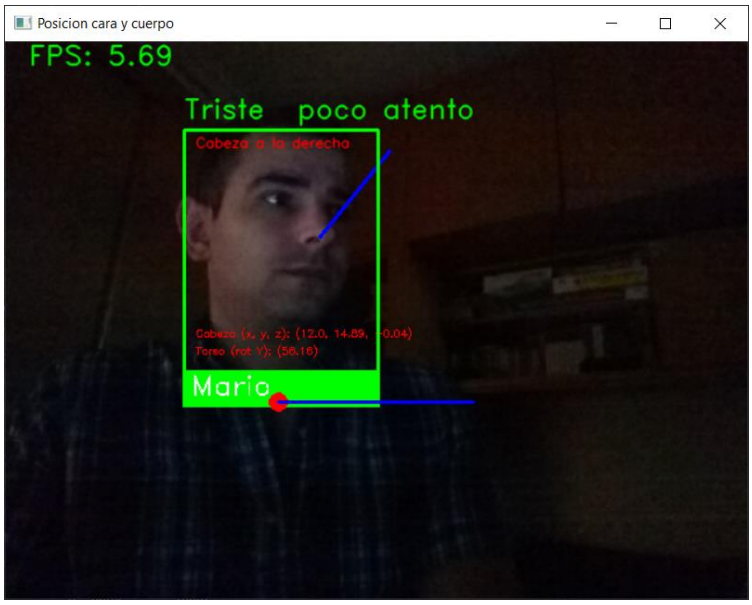

Figura 37 - Imagen del Equipo 1 con la cámara 2 – Iluminación muy baja, con luz residual de la pantalla.

5.1.2.2. Combinación de mejora en cámara e iluminación

En la Figura 30 se ha mostrado cómo una mejora en la iluminación corregía el funcionamiento de DeepFace y mostraba los sentimientos correctamente. A continuación, en la Figura 38 se observa cómo los resultados se mantienen correctos al combinar una mejora en la calidad de la cámara y la iluminación.

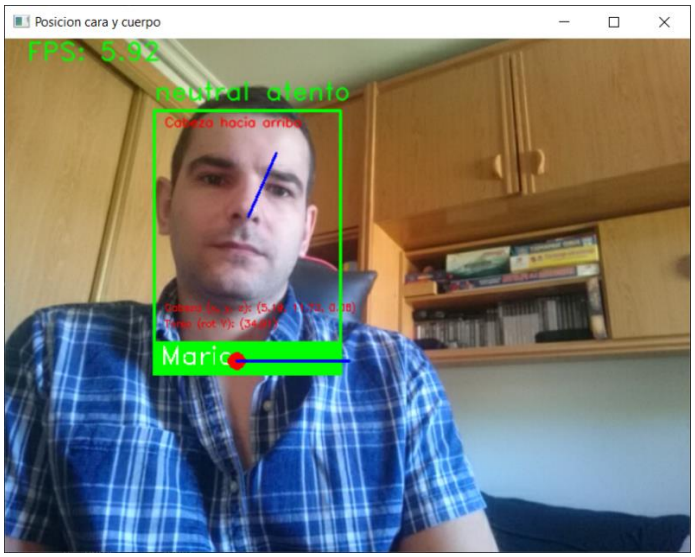

Figura 38 - Imagen del Equipo 1 con la cámara 2 – luz lateral con foco difuminado.

En este caso, la obtención de todos los resultados es correcta. Una posición del torso ligeramente más lateral y con una orientación de la cabeza por encima del centro de la cámara dan como resultado “atento”, un grado inferior al máximo “muy atento”.

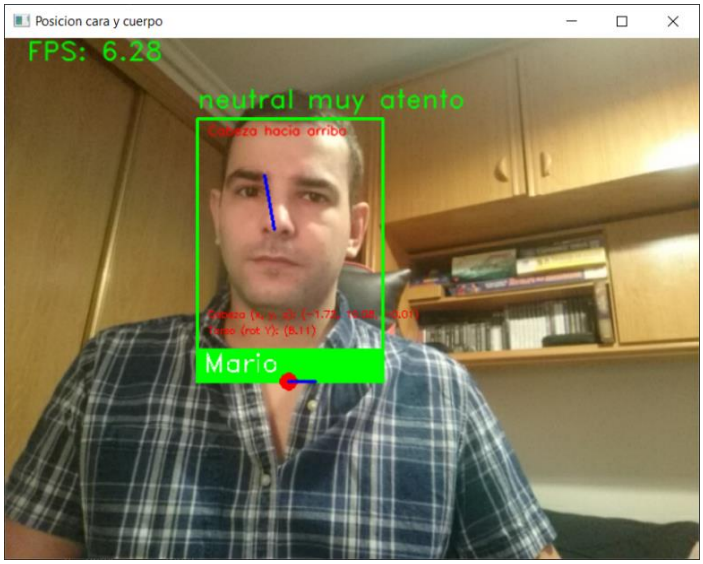

Figura 39 - Imagen del Equipo 1 con la cámara 2 – luz encima de la cabeza con foco difuminado.

Al igual que en la Figura 31, en la Figura 39 se puede observar cómo la mejora de la cámara y la ayuda de un foco que difumina la luz influyen en el resultado del reconocimiento de emociones.

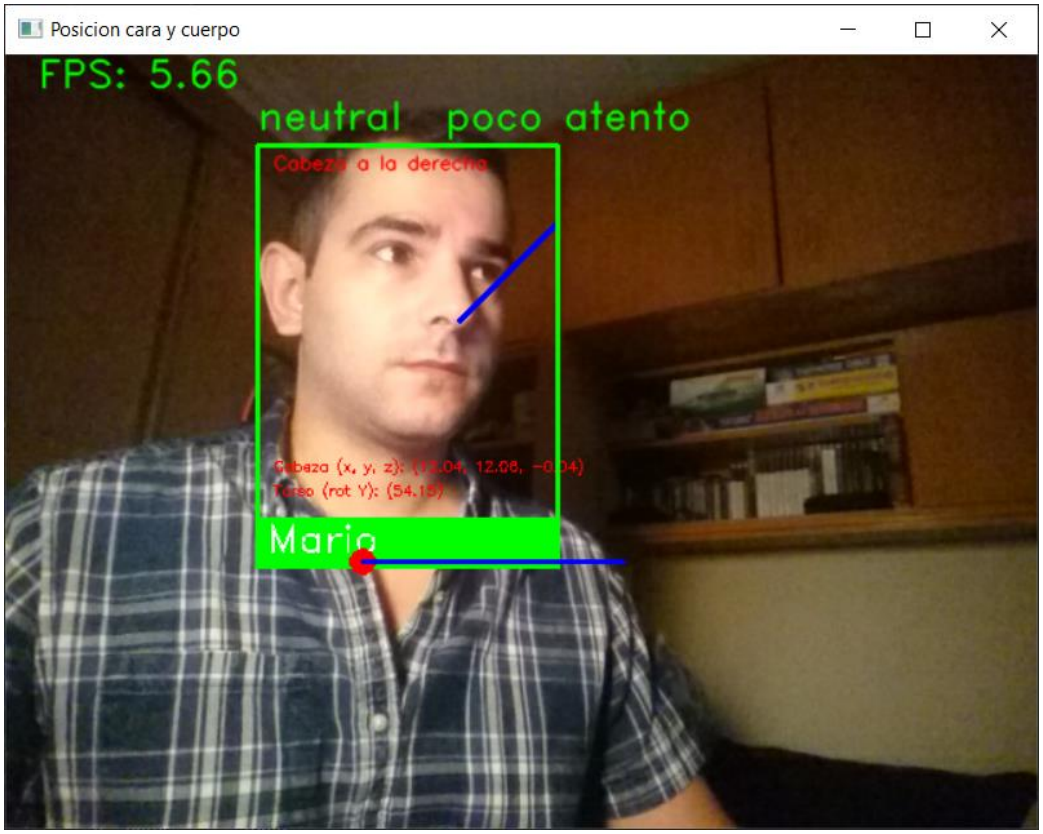

Figura 40 - Imagen del Equipo 1 con la cámara 2 – Iluminación muy baja con foco difuminado

La Figura 40 es la última prueba de iluminación donde se combina la mejora de una cámara y la presencia del foco de luz difuminada en el peor escenario posible, a oscuras. A diferencia de la Figura 32 donde la mejora de la iluminación no ha sido relevante para la corrección del funcionamiento del sistema, aquí sí se han conseguido detectar correctamente las emociones. Esto demuestra que una combinación del cambio en la calidad de diferentes factores puede dar como resultado un buen funcionamiento del sistema ante situaciones adversas.

5.1.2.3. Variación del rendimiento en función de la cámara

Si comparamos la Figura 26, la Figura 28 y la Figura 34, la única diferencia que hay entre ellas es la cámara. Tienen la misma iluminación, el mismo número de personas y el resto del hardware es el mismo. En la Figura 26 se obtiene un valor de 5,87 FPS. Como el tamaño de la imagen de la cámara 2 es más grande, y, por tanto, se debe analizar más información, los resultados son ligeramente inferiores, 5,23 FPS.

Esta disminución del rendimiento no es preocupante, pero indica que la resolución de la cámara es un dato que se debe tener en cuenta a la hora de mejorar nuestro sistema.

La calidad de las cámaras no se mide únicamente por su resolución, también se tienen en cuenta otros parámetros como calidad de la lente u objetivo, así como tiempo y calidad de procesamiento de la imagen por parte de la cámara. Estos últimos son más difíciles de comparar entre diferentes cámaras, pero hay que dejar claro que mejorar la calidad no tiene porqué suponer un aumento en el tamaño de la imagen obtenida, y por tanto una pérdida de rendimiento.

5.1.2.4. Variación del rendimiento según el número de personas

Para poder comparar el rendimiento en función del número de personas se han debido utilizar los mismos elementos en todas las fotos. En este caso, una o dos personas, la cámara 2 y los diferentes equipos usando el mismo sistema.

En la Figura 41 se puede observar cómo, cuando aparecen dos personas, se utiliza la cámara 2 y se utiliza el equipo portátil, el rendimiento es de 3,0 FPS.

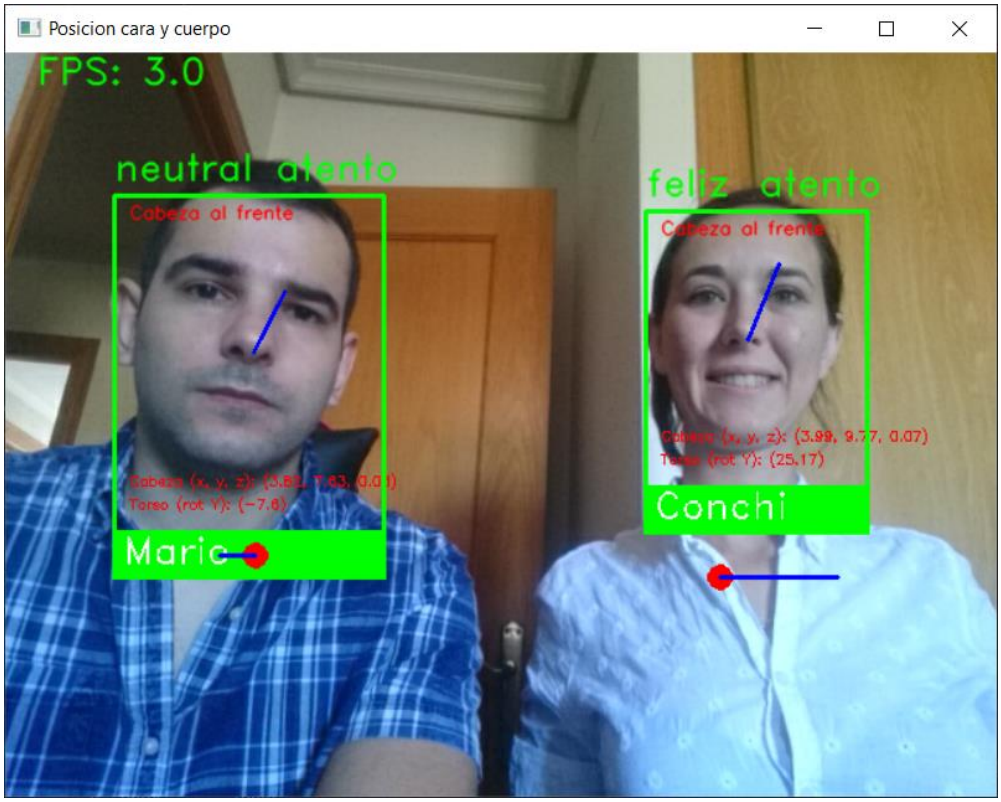

Figura 41 - Imagen del Equipo 1 con la cámara 2 – Participación de dos personas en la misma imagen.

5.1.3. Equipo 2 utilizando cámara 2

Se ha estudiado si una mejora en el resto de las prestaciones del equipo supondría un aumento en el rendimiento del sistema. El equipo 2 tiene más potencia de CPU a su disposición, unas tarjetas RAM de mayor capacidad y que funcionan a más frecuencia y un disco duro más rápido que el equipo 1.

5.1.3.1. Variación del rendimiento en función del hardware

A continuación, se muestra la Figura 42 que corresponde con las mismas condiciones ambientales que la Figura 35, pero con la imagen procesada por un equipo con más potencia de cálculo.

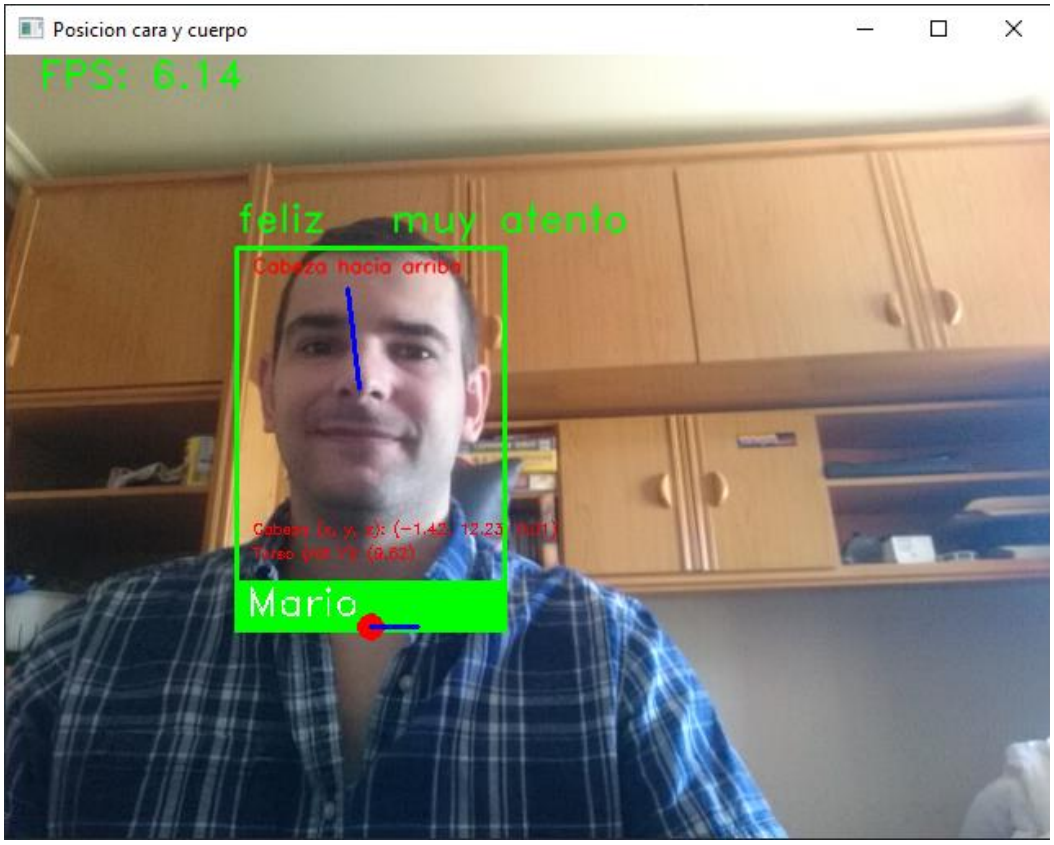

Figura 42 - Imagen del Equipo 2 con la cámara 2 – luz de lateral.

Las imágenes procesadas a través de un equipo con mayores prestaciones han aumentado el rendimiento a 6,14 FPS en comparación con los 5,44 FPS que había antes. Este comportamiento es normal teniendo en cuenta que el equipo 2 destina más recursos que el equipo 1 al procesamiento de las imágenes.

5.1.3.2. *Variación del rendimiento en función del número de personas*

En la Figura 43 se puede observar una situación parecida a la Figura 33, pero en este caso la diferencia es que se utiliza el equipo 2 de mayores prestaciones y la cámara 2.

Esta figura se ha utilizado en el resumen de resultados para realizar una comparación conjunta del rendimiento del sistema en diferentes escenarios.

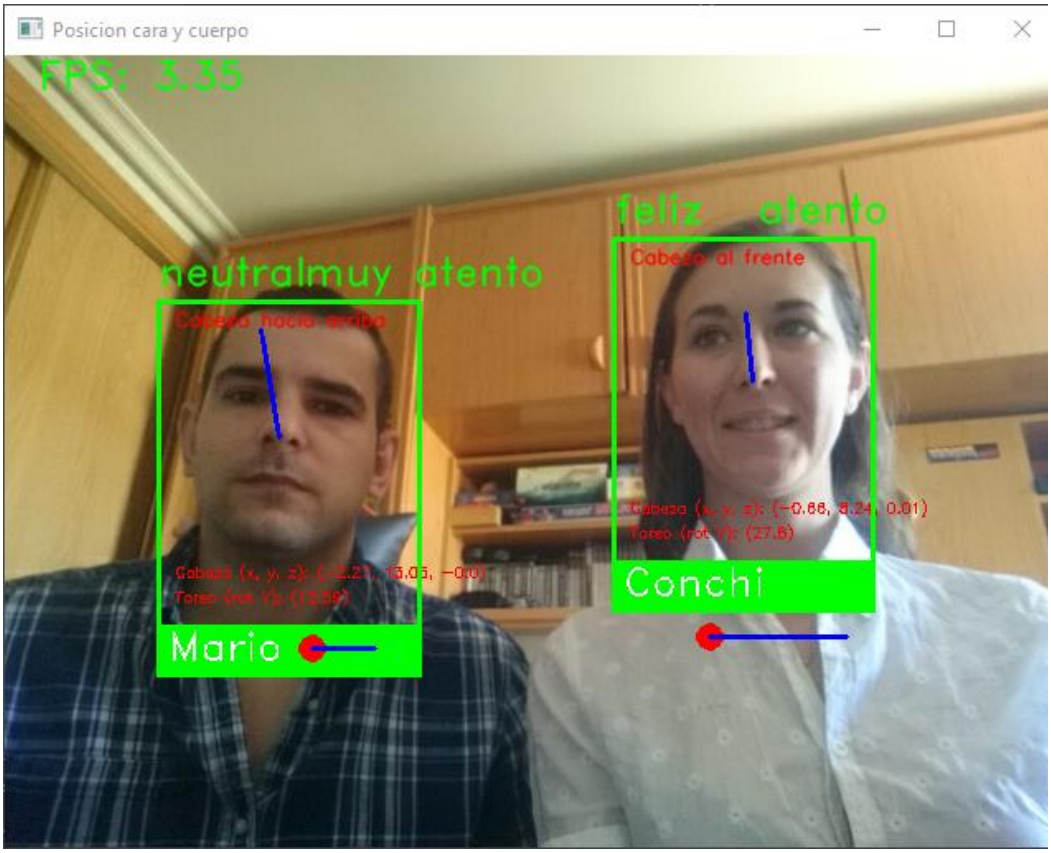

Figura 43 - Imagen del Equipo 2 con la cámara 2 – Participación de dos personas en la misma imagen.

5.1.4. Equipo 3 utilizando cámara 2

También se han realizado pruebas sobre un equipo con menos potencia que el equipo 1, para así buscar el límite de características que pueden reunir los equipos destinados a la implementación de este sistema.

5.1.4.1. *Variación de la iluminación*

Debido a la propia ubicación del equipo 3, el nivel de iluminación natural es bajo, tal y como se observa en la Figura 44. Esta iluminación es incluso menor que en la Figura 28, donde la librería DeepFace tuvo un bajo índice de aciertos.

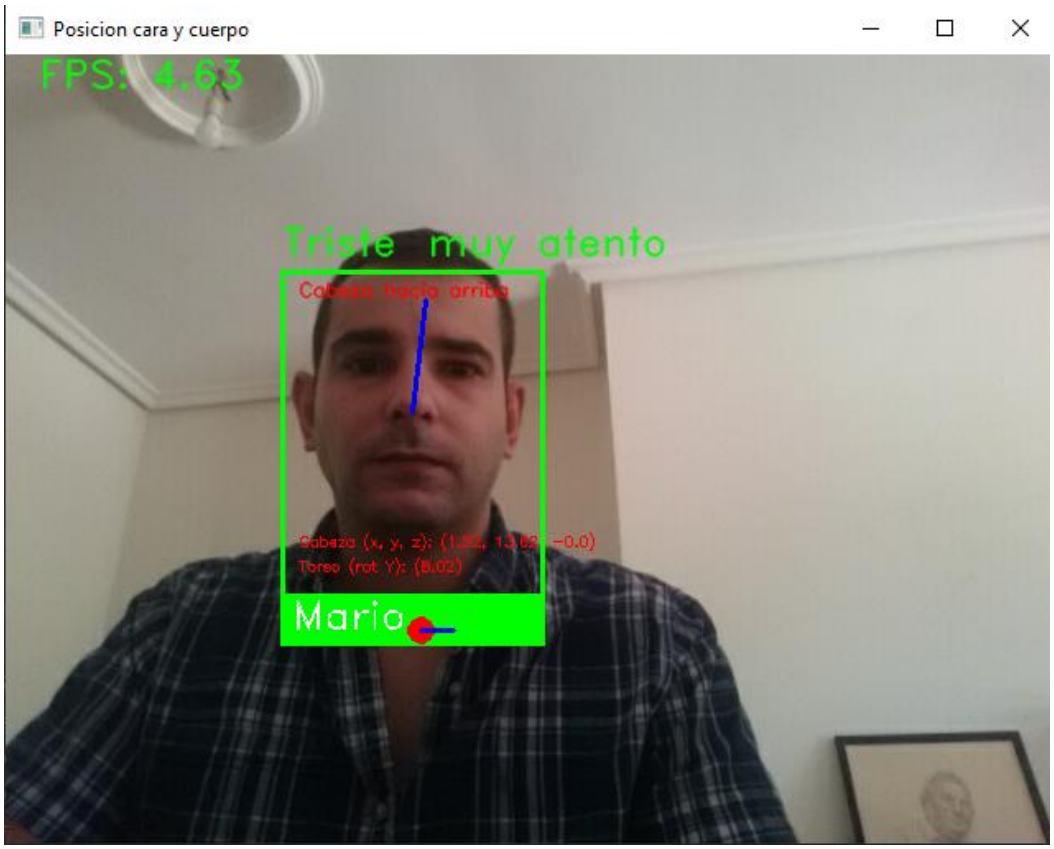

Figura 44 - Imagen del Equipo 3 con la cámara 2 – luz de frente.

La Figura 44 representa la situación más ideal que se puede dar en ese contexto. Aun así, la iluminación es baja, cuesta ver claramente la mitad del rostro, y solo aparece una persona dentro de la imagen.

Como se puede observar, la imagen es procesada en un equipo con menos recursos que los anteriores, y, aun así, obtiene un rendimiento de 4’63 FPS. Esta cifra más baja no sorprende, pero sigue siendo lo suficientemente buena como para realizar un seguimiento fluido de la escena.

En cuanto a las herramientas hay que destacar que la iluminación ha afectado demasiado a la librería DeepFace y no se registran correctamente los datos de las emociones, que por otro lado se encuentran en una situación muy desfavorable en cuanto a iluminación y contraste.

5.1.4.2. *Variación del rendimiento en función del número de personas*

Cuando se aumenta el número de personas también varía el rendimiento del sistema, igual que en la Figura 33. En la Figura 45 se aprecia esa disminución de rendimiento llegando a cruzar el umbral de 2 FPS.

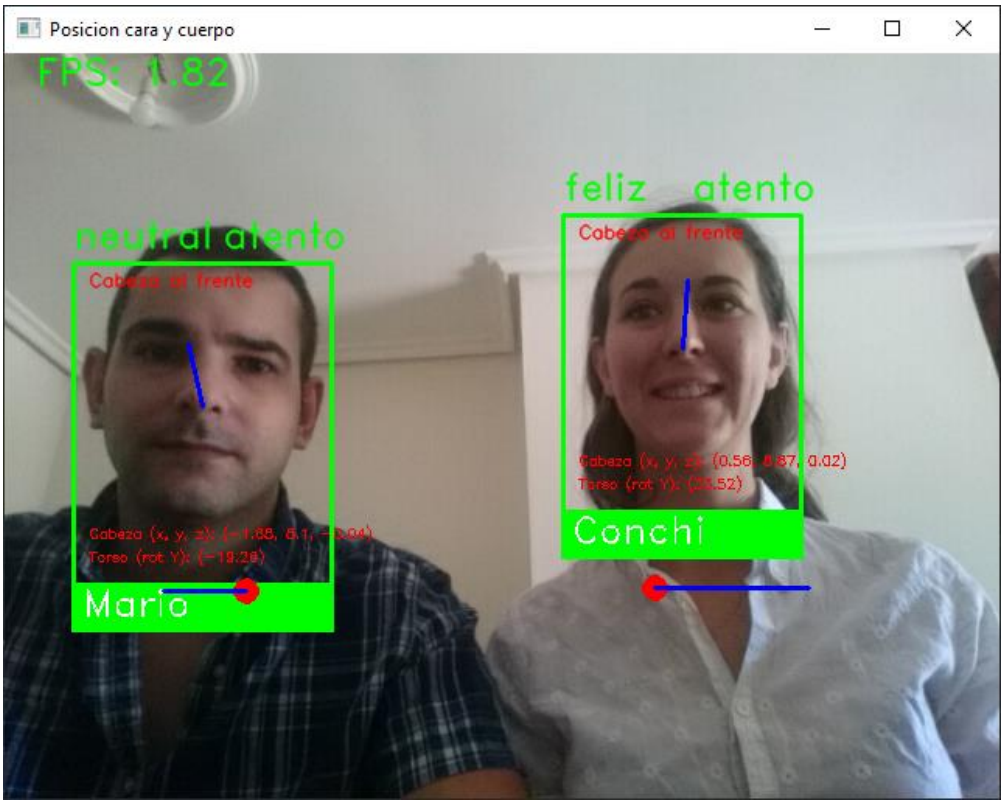

Figura 45 - Imagen del Equipo 3 con la cámara 2 – Participación de dos personas en la misma imagen.

La reducción de valor tiene sentido teniendo en cuenta que es un equipo con menos recursos que los anteriores, y, además, tiene que analizar el doble de información. En cuanto a la fluidez de las imágenes se percibe falta de continuidad entre ellas, pero sin desestabilizar el funcionamiento del sistema ni producir errores críticos.

Aunque en la Figura 45 los resultados de las emociones son correctos, experimentalmente se observa que no es así en todas las imágenes procesadas.

5.2. Resumen de resultados

Durante los ensayos se han planteado 20 escenarios diferentes donde se ha puesto a prueba el sistema desarrollado. En las siguientes tablas se exponen los resultados de dichas pruebas para mostrar visualmente todo lo comentado anteriormente.

Resultado tareas / Funciones probadas	Funcionamiento correcto	Funcionamiento incorrecto
Reconocimiento facial	20	0
Reconocimiento de las emociones	12	8
Reconocimiento de la participación	20	0

Tabla 4 - Exposición de los resultados de las funciones del sistema desarrollado

Por un lado, en la Tabla 4 se presenta el correcto desarrollo de las funciones de reconocimiento facial y de participación de la interacción humano-máquina. Por otro lado, el reconocimiento de las emociones no funciona en cualquier escenario, haciendo necesario acotar en qué situaciones esta librería funciona correctamente.

A continuación, se muestra la relación de resultados positivos y negativos del reconocimiento de las emociones en función de la iluminación y las cámaras utilizadas.

Iluminación / Cámara	Iluminación natural frontal	Iluminación natural lateral + foco difuminado	Iluminación natural lateral	Luz lámpara + foco difuminado	Luz lámpara	Oscuras + foco difuminado	Oscuras
Cámara 1	✓	✓	✗	✓	✗	✗	✗
Cámara 2	✓	✓	✓	✓	✗	✓	✗

Tabla 5 - Resultados del reconocimiento facial en función de la calidad de la cámara y la iluminación

En la Tabla 5 se puede observar claramente cómo el aumento de la calidad de la cámara mejora los resultados del reconocimiento de los rostros cuando la luz incide lateralmente sin necesidad de otra ayuda. Y, por otro lado, se consiguen perfeccionar los resultados de dicho reconocimiento cuando existe un nivel de iluminación más bajo con ayuda de un foco, a diferencia de una cámara de baja calidad donde ni con esta ayuda funciona correctamente.

Para poder realizar una comparación bajo el mismo contexto, se ha implementado el mismo sistema y cámara en diferentes equipos mostrando un número diferente de personas en cada imagen.

Nº personas / Rendimiento	1 persona	2 personas
Rendimiento equipo 1	5,9 FPS	3 FPS
Rendimiento equipo 2	6,14 FPS	3,35 FPS
Rendimiento equipo 3	4,63	1,82 FPS

Tabla 6 - Rendimiento en función del número de persona y el equipo.

En la Tabla 6 se puede observar la relación directa en cuanto al número de personas y la potencia de los equipos. Como se ha explicado anteriormente, cuantas más personas aparecen en la imagen menor rendimiento. Esto es debido a que el proceso de reconocimiento de rostros, de emociones y de interacción se debe realizar más veces por cada imagen, reduciendo de esa manera la tasa de refresco. Por otro lado, se puede apreciar qué, cuanto más potente es un equipo, más rápido realiza las tareas mencionadas, y más rápido analiza cada imagen.

A raíz de los resultados obtenidos y expuestos en este capítulo, se ha registrado que face-recognition es una herramienta sólida con una tasa de error muy baja para el reconocimiento facial.

	Resultados precisos	Resultados válidos
NN	75,12	96,08
SVM	68,20	93,09

Tabla 7 - Resumen de resultados de los modelos de NN y SVM

Por otro lado, tal y como se observa en la Tabla 7 y la Figura 25, la NN y SVM entrenadas para evaluar el nivel de participación también han cumplido su función, y los resultados han sido acordes a las situaciones de cada momento. Aunque la NN ha obtenido mejores resultados que SVM durante la validación, se han utilizado los modelos de ambos métodos en la exposición de resultados finales, y experimentalmente se ha observado que se obtienen las mismas predicciones.

La herramienta DeepFace no ha cumplido correctamente su función en todos los escenarios planteados. En el apartado "Discusión" se valorará la necesidad de esta librería en este sistema y qué alternativas se ofrecen.

En cuanto al rendimiento, ha quedado claro que la potencia del equipo que sostiene el sistema, el número de personas y la resolución de la cámara afectan directamente en el rendimiento, por lo que hay que buscar un equilibrio entre ellos para un funcionamiento correcto. El límite inferior en cuanto a potencia de equipo ha sido el equipo 3, el cual presenta menos recursos, pero el que ha logrado mantener el rendimiento lo suficiente como para que el sistema mantenga un funcionamiento correcto cuando hay 2 personas en la escena y una cámara con una calidad aceptable.

El elemento de la cámara ha sido clave para que la librería DeepFace realice mejor su trabajo. Cuando se ha utilizado una cámara de mejor calidad bajo el mismo contexto que cuando se utilizaba la de menor calidad, se apreciaba notablemente un mejor funcionamiento en la detección de emociones. También se ha notado alguna mejoría en cuanto al aumento de la iluminación en la escena.

5.3. Discusión

Las IA's basadas en SVM y NN's para realizar la evaluación del nivel de interacción entre humanos y máquinas han sido correctas en ambos casos. Como se ha explicado en otros capítulos de este TFG, los datos de entrenamiento y validación fueron creados bajo el sesgo de la persona que creó la información. Y los resultados obtenidos de estas IA's pueden haber estado influenciados bajo esa subjetividad.

Face-recognition también tiene una tasa de aciertos muy alta en cualquier situación. Como su funcionamiento es correcto y cumple todo lo que se espera de ella, se considera oportuno mantener esta librería.

El rendimiento de la librería DeepFace está muy vinculado a la iluminación, contrastes, y calidad de la cámara. Para contrarrestar los problemas de la iluminación se plantean alternativas como el uso de una luz asociada a la

cámara que ilumine correctamente la escena, lo cual puede no ser una solución práctica en cualquier situación donde se quiera implantar el sistema. Un foco de luz, aunque esté difuminado, puede causar molestias en los ojos del interlocutor y afectar negativamente al HRI.

También se propone la mejora de la calidad de la cámara en situaciones que planteen problemas de luminosidad. Se pueden mejorar muchas características de una cámara como sensibilidad del sensor, resolución, calidad del objetivo, calidad y tiempo de procesado de la imagen, etc, pero siempre asumiendo que estas mejoras afectan a la inversión que se quiera realizar para implantar este sistema.

Por último, se plantea la posibilidad de usar una librería diferente a DeepFace. En internet existen un gran número de alternativas, pero hay que comprender que esta librería usa DL, y es uno de los mejores métodos para entrenar con datos ligados a la conducta humana, en este caso las emociones mediante lenguaje no verbal. Además, posee diferentes modelos de empresas comerciales y programas conocidos como Facebook, Google, etc, para detectar las emociones, pero no ofrecen resultados mejores.

Para terminar, se establece que el límite de personas que pueden aparecer en equipos con pocos recursos es 2. Por encima de ese valor se perdería demasiado rendimiento y generaría muchos errores en el funcionamiento en equipos con pocos recursos. Por otro lado, aunque puede parecer un límite muy bajo, por la distancia a la que se encuentra la cámara de la escena que graba, tampoco se podrían introducir en esa escena más personas sin amontonarse.

6. Conclusiones y líneas futuras

6.1. Conclusiones

Con la finalización de este trabajo se ha podido concluir que se ha cumplido tanto con los objetivos principales como con los secundarios que se plantearon en un principio.

Tras una búsqueda bibliográfica inicial se han evaluado diferentes formas de obtener información de una persona para poder estimar su grado de atención hacia un robot u ordenador, y se han utilizado aquellas que, además de añadir valor al sistema, no han reducido la movilidad de los equipos donde pueda ser implementado. Mediante la detección de emociones también se ha evaluado si la interacción está siendo positiva o debe modificarse la conducta de la máquina, y de esta forma aumentar su autonomía para interactuar con las personas.

A lo largo de todo el trabajo ha habido que enfrentar tareas que no solo implicaban investigación, sino también un desarrollo con una visión completa de todos los objetivos para que las funciones de este sistema se realizasen de una manera correcta.

El mayor reto que se ha afrontado ha sido la pérdida de rendimiento por la librería de reconocimiento facial. El hecho de no desestimar una solución que da buenos resultados, y complementar su funcionamiento con el sistema para que se ayuden mutuamente, evitando así la reducción del rendimiento, ha sido muy didáctico.

Se ha conseguido encontrar un lenguaje de programación con un uso muy directo en los equipos. Basta con instalar Python, las librerías con el código de este trabajo, y una cámara a elección de la persona que lo quiera usar, para empezar a utilizar el sistema desarrollado. A partir de ahí, se puede ejecutar el código y se empieza a obtener tanto el grado de participación de un humano en las interacciones con el equipo, como si esta interacción se está realizando correctamente.

Se han tenido en cuenta los requisitos de hardware que deben tener las máquinas para implementar este sistema. La ventaja de este proyecto es que puede ser instalado en ordenadores y robots con pocos recursos, lo cual no limita que pueda instalarse en equipos de mayores prestaciones.

Inicialmente se habían focalizado los resultados del rendimiento según la potencia de los equipos. Pero también hay que tener en cuenta si en dichos equipos se realizan otras tareas como controlar los movimientos de un robot, o si tiene tareas en segundo plano con las que se deban compartir dichos recursos, tanto hardware como software. En este trabajo se han realizado los

experimentos bajo las mismas condiciones, en los equipos solo se ejecutaba el programa desarrollado bajo el sistema operativo Windows 10 completamente actualizado.

Por último, se han comparado gran cantidad de librerías y algoritmos tanto para evaluar caras, como para evaluar emociones, y gracias a estas comparaciones se ha podido aprender a estimar para qué pueden ser útiles algunas de ellas.

Se han conseguido utilizar herramientas para identificar la cara y la emoción que expresan las personas, pero una barrera que aún no se ha solucionado es la de detectar las caras falsas. Una persona que comprenda que el equipo evalúa exactamente lo que ve la cámara, donde la máquina no entiende que el lenguaje no verbal está siendo modificado, puede engañar constantemente a este sistema.

En cuanto a los resultados, experimentalmente se obtienen resultados acordes con los esperados, siempre que se cumplan unos mínimos de iluminación y de calidad de los equipos y cámaras.

Debido al presupuesto limitado, no se ha podido probar este sistema con cámaras de muy alta calidad. Por esta razón, aunque no está previsto que el sistema sea usado en condiciones con muchos problemas de iluminación, no se descarta que exista una combinación de potencia del equipo y cámara de alta calidad donde este sistema funcione en condiciones de iluminación baja.

Una de las dificultades de este trabajo ha sido entrenar un sistema con unos datos de entrenamiento definidos bajo el sesgo de una persona. La información ha sido obtenida directamente de un humano, por ello NN y SVM han conseguido generar modelos de los que se obtienen predicciones adecuadas a cada situación. Pero estos resultados nunca van a considerarse como el resultado exacto por unanimidad debido a que las personas puede percibir la misma experiencia de manera diferente entre ellas.

Finalmente, para que las personas puedan interpretar fácilmente los datos, al cerrar la interacción se representa el nivel de atención con un gráfico, aunque este sistema está previsto para ser usado por máquinas.

6.2. Líneas futuras

Una posible línea futura es poder diferenciar qué equipos sí poseen elementos como GPU y realizar el mismo trabajo con más potencia, además de dar prioridad a este sistema sobre otros del equipo menos necesarios, para que se le asignen más recursos y mejorar su rendimiento. Actualmente solo tiene en

cuenta la CPU y no se gestiona la prioridad de la ejecución de este sistema sobre otros.

Otra posible línea futura es que un robot o un ordenador gestionen las acciones que realizan en función de las respuestas que recibe de este sistema.

Por último, se podría reducir la subjetividad de la base datos de entrenamiento mediante un proceso de obtención de la información de las personas desde un campo más cercano a la psicología. Un psicólogo conductual tiene más conocimiento sobre este campo, y además disponen de métodos y estrategias para realizar los experimentos con los que obtener una base de datos más objetiva.

Bibliografía

Ahmad M. I. Mubin O. & Orlando J. (2017). Adaptive Social Robot for Sustaining Social Engagement during Long-Term Children–Robot Interaction. https://www.tandfonline.com/doi/abs/10.1080/10447318.2017.1300750?journalCode=hihc20

Amir, A & Tapus, A. (2010). *Gestures Imitation with a Mobile Robot in the Context of Human-Robot Interaction (HRI) for Children with Autism*. https://www.researchgate.net/publication/266448568_Gestures_Imitation_with_a_Mobile_Robot_in_the_Context_of_Human-Robot_Interaction_HRI_for_Children_with_Autism

Bueno F. (2019). Redes Neuronales: Entrenamiento y Comportamiento. https://eprints.ucm.es/id/eprint/64564/1/BUENOPASCUALFERNANDO.pdf

Duque-Domingo J., Gómez-García-Bermejo J. & Zalama E. (2020*) Gaze Control of a Robotic Head for Realistic Interaction With Humans*. https://www.frontiersin.org/articles/10.3389/fnbot.2020.00034/full

Esparza C.H., Tarazona C., Sanabria E.E., Velazco D.A. (2015). Reconocimiento Facial Basado en Eigenfaces, LBPH y Fisherfaces en la beagleboard-xM. https://1library.co/document/zpnjkp24-reconocimiento-facial-basado-eigenfaces-lbhp-fisherfaces-beagleboard-xm.html

Kapoor A., Burlesonc W. & Picardb R.W. (2007). *Automatic prediction of frustration*. https://www.sciencedirect.com/science/article/abs/pii/S1071581907000377

Moon G, Chang J. Y. & Lee K. M. (2019). Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image. https://arxiv.org/abs/1907.11346

Nummenmaa, L., Hirvonen, J., Parkkola, R. & Hietanen, J. K. (2008). Is Emotional Contagion Special? An Fmri Study on Neural Systems for Affective and Cognitive Empathy. NeuroImage. https://pubmed.ncbi.nlm.nih.gov/18790065/#:~:text=The%20results%20suggest%20that%20emotional,bodily%20states%20than%20cognitive%20empathy.

Oertel C., Castellano G., Chetouani M., Nasir J., Obaid M., Pelachaud C & Peters C. (2020). *Engagement in Human-Agent Interaction: An Overview*. https://www.frontiersin.org/articles/10.3389/frobt.2020.00092/full#:

~:text=Engagement%20is%20a%20concept%20of,capable%20of%20adapting%20to%20users.

Ospina A., Pulido M. (2017). Comparar tres métodos convencionales de reconocimiento facial y confrontar el de mejor resultado frente a computación cognitiva. https://repositorio.itm.edu.co/handle/20.500.12622/1706

Pérez F. (2021). Clasificación y corrección de posturas humanas en dispositivos móviles. https://m.riunet.upv.es/handle/10251/175049

Ren J., Ding R., Li S., Zhang M. Wei D., Feng C. L., Xu P. & Luo W. (2022). *Features and Extra-Striate Body Area Representations of Diagnostic Body Parts in Anger and Fear Perception*. https://pubmed.ncbi.nlm.nih.gov/35447997/

Rico C. (2021). Estudio de algoritmos de seguimiento de mirada a través de videos. https://repositorio.uam.es/handle/10486/700141

Rogers S. & Pennington B. F. (1991). *A theoretical approach to the deficits in infantile autism*. https://www.researchgate.net/publication/333184571_A_theoretical_approach_to_the_deficits_in_infantile_autism

Sidner C.L., Lee C., Kidd C. D., Lesh N. & Rich C. (2005). *Explorations in engagement for humans and robots.* https://www.sciencedirect.com/science/article/pii/S0004370205000512

Vallverdu, J., Nishida, T., Ohmoto, Y., Moran, S. & Lázare Boix, S. (2020). *Fake Empathy and Human-Robot Interaction (HRI): A Preliminary Study*. https://www.researchgate.net/publication/320487257_Fake_Empathy_and_Human-Robot_Interaction_HRI_A_Preliminary_Study

Zhao Z., Zheng P., Xu S. & Wu X. (2019). *Object Detection With Deep Learning: A Review*. https://www.researchgate.net/publication/330708842_Object_Detection_With_Deep_Learning_A_Review

Zulkifli, W., Shamsuddin, S., Jafar, F., Ahmad, R., Abdul, A., Abdulsalam, A. & Lim, T. H. (2018). *Smile Detection Tool using OpenCV-Python to Measure Response in Human-Robot Interaction with Animal Robot PARO. International Journal of Advanced Computer Science and Applications*. https://www.researchgate.net/publication/329420821_Smile_Detection_Tool_using_OpenCV-Python_to_Measure_Response_in_Human-Robot_Interaction_with_Animal_Robot_PARO

Webgrafía

Amat J. (Abril de 2017). Máquinas de Vector Soporte (Support Vector Machines, SVMs). Cienciadedatos.net.

https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines

Descripción de herramientas de Mediapipe. Google Github. (https://google.github.io/mediapipe/solutions/solutions.html).

Descripción de método solvePnP de OpenCV. Delftstack. (https://www.delftstack.com/es/howto/python/opencv-solvepnp/).

Introducción al principio de LBP y la implementación del algoritmo. Programmerclick. (https://programmerclick.com/article/15601557767/).

Nielsen N. (21 de diciembre de 2021). *Head Pose Estimation with MediaPipe and OpenCV in Python.* Youtube. https://www.youtube.com/watch?v=-toNMaS4SeQ&ab_channel=NicolaiNielsen-ComputerVision%26AI

Proyectos CARTIF AIROSO. CARTIF. (https://www.cartif.es/airoso/)

Web OpenCV. cv::ml::SVM Class Reference. docs.opencv.org. https://docs.opencv.org/4.x/d1/d2d/classcv_1_1ml_1_1SVM.html#aad7f1aaccced3c33bb256640910a0e56

Imágenes licencia Creative Commons

Figura 7 - Dos personas realizando gesto de compromiso. (Obtenida de Wikimedia - Bajo licencia Creative Commons):

https://commons.wikimedia.org/wiki/File:Understanding_in_Xian_0546.jpg

Figura 8 - Persona interactuando con los periféricos de entrada con un nivel de atención alto sobre el equipo. (Bajo licencia Creative Commons):

https://pxhere.com/es/photo/824718

Anexo I Características de los equipos usados

Equipo 1 (portátil):

- Sistema operativo: Windows 10 Pro x64
- Procesador: Intel Core i5-6300HQ CPU 2,30GHz (4 núcleos)
- RAM: 8 GB
- Tipo de disco duro: SSD Sata
- **Webcam integrada (Cámara 1):** VGA (640 x 360)

Equipo 2 (Sobremesa):

- Sistema operativo: Windows 10 Pro x64
- Procesador: Intel Core i7-5820K CPU 3,30GHz (6 núcleos)
- RAM: 16 GB
- Tipo de disco duro: SSD M.2

Equipo 3 (Sobremesa):

- Sistema operativo: Windows 10 Pro x64
- Procesador: Intel Pentium CPU G2120 3.10GHz
- RAM: 4 GB
- Tipo de disco duro: HDD

Cámara 2:

- **Cámara trasera Sony Xperia Z3 compact:** 20.7 MP (1/2.3")

Anexo II Detalles de implementación

Para realizar esta implementación, se ha utilizado un entorno de desarrollo integrado (IDE) para Python llamado Pycharm. Este IDE ha permitido el uso sencillo de Python sobre el sistema operativo, y la instalación de librerías de una manera gráfica. Se han escrito los comandos de instalación y los requisitos para instalar cada librería por si se requiere en alguna otra implantación.

Después de instalar Pycharm Community Edition (2022.1.3) se ha creado un entorno virtualizado con Python 3.9, y se ha abierto la carpeta del proyecto llamada "HRIEngagement". No se ha descartado su uso en versiones posteriores de Python, pero su uso no ha sido probado.

Después se ha abierto el intérprete de Python y se ha comenzado la instalación de las siguientes librerías:

- **OpenCV-contrib-Python**

Esta librería ofrece el repositorio completo de herramientas de OpenCV, incluidos módulos extra aportados por la gente. Está dedicada al tratamiento de imágenes y posee alguna función para entrenar IA's, pero existen herramientas más desarrolladas para esta labor como Keras. No necesita cumplirse ningún requisito previo para su instalación.

Última versión comprobada: 4.6.0.66

Comando de instalación:

pip install opencv-contrib-python

- **Mediapipe**

Es una librería que ofrece soluciones de ML. En este proyecto es utilizada para detectar la malla de la cara y puntos característicos del cuerpo gracias a sus herramientas pre-entrenadas para esta labor. No necesita cumplirse ningún requisito previo para su instalación.

Última versión comprobada: 0.8.10.1

Comando de instalación:

pip install mediapipe

- NumPy

Es una librería diseñada para trabajar con grandes volúmenes de datos para cálculo numérico y análisis de esos datos. Frecuentemente es utilizada por otras librerías que trabajan con IA. Se instala automáticamente con OpenCV. No necesita cumplirse ningún requisito previo para su instalación.

Última versión comprobada: 1.23.1

Comando de instalación:

pip install numpy

- CMake

Esta librería gestiona el proceso de compilación dentro de un sistema operativo independientemente del compilador. No necesita cumplirse ningún requisito previo para su instalación.

Última versión comprobada: 3.22.5

Comando de instalación:

pip install cmake

- Face-recognition y Dlib

Es biblioteca de reconocimiento facial que permite reconocer y manipular caras en Python. Esta librería sí tiene requisitos previos. Está basada en Dlib, y se instala automáticamente con el mismo comando, pero necesita tener previamente instalado CMake y "Desarrollo para el escritorio con C++" del IDE Visual Studio. Esto sucede porque Dlib trabaja originalmente en C++ y se utilizan estos complementos para poder utilizarlo en Python.

Dlib es un conjunto de herramientas para ML y análisis de datos.

Última versión comprobada Face-recognition: 1.3.0

Última versión comprobada Dlib: 19.24.0

Comando de instalación:

pip install face-recognition

pip install Dlib (por si no se instalara automáticamente)

- **DeepFace**

Librería basada en Keras especializada en análisis de la cara como reconocimiento facial, de emociones, de género, de edad, etc.

Última versión comprobada DeepFace: 0.0.75

Última versión comprobada Keras: 2.9.0

Comando de instalación:

pip install deepface

pip install keras

- **Pynput**

Librería con funciones para monitorizar y controlar dispositivos de entrada. No necesita cumplirse ningún requisito previo para su instalación.

Última versión comprobada: 1.7.6

Comando de instalación:

pip install pynput

Tras la instalación de las librerías y la instalación de una cámara si no está integrada en el equipo, ya se puede ejecutar el sistema desarrollado.