Pontificia Universidad JAVERIANA Cali

Facultad de Ingeniería y Ciencias

Acta de Correcciones al Proyecto de Grado
Ingeniería Electrónica

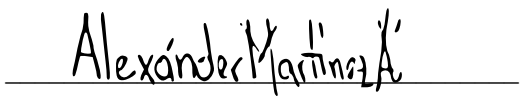
Fecha: 17/05/2023

Autores: María de los Ángeles Delgado Giraldo
Juan Felipe Penagos Angrino

Nombre del Proyecto de Grado: Aprendizaje de máquina aplicado a un proceso de clasificación para un robot colaborativo

Director : Alexánder Martínez Álvarez
Codirector : José Hernando Mosquera de la Cruz

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.



Firma de Director del Proyecto de Grado

Firma de Codirector del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado
en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero Electrónico.

Dr. HERNÁN CAMILO ROCHA NIÑO
Decano de la Facultad de Ingeniería y Ciencias

Dr. LUIS EDUARDO TOBÓN LLANO
Director Carrera Ingeniería Electrónica.

Dr. ALEXÁNDER MARTINEZ ÁLVAREZ
Director Trabajo de grado

Mg. JOSÉ HERNANDO MOSQUERA DE LA CRUZ
Codirector Trabajo de grado

Dr. HUMBERTO LOAIZA CORREA
Jurado 1

Dra. SANDRA ESPERANZA NOPE RODRÍGUEZ
Jurado 2

Santiago de Cali, abril 26 de 2023

Doctor
Luis Eduardo Tobón Llano
Director
Carrera de Ingeniería Electrónica

Asunto: Entrega final trabajo de grado "Sistema de clasificación para un robot colaborativo utilizando una arquitectura cognitiva" y "Sistema de clasificación para un robot colaborativo utilizando técnicas de aprendizaje de máquina"

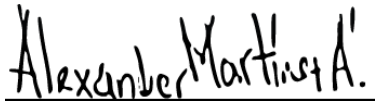
Cordial saludo.

Por medio de la presente nos permitimos presentar los trabajos de grado terminados bajo nuestra dirección: "SISTEMA DE CLASIFICACIÓN PARA UN ROBOT COLABORATIVO UTILIZANDO UNA ARQUITECTURA COGNITIVA" el cual fue desarrollado por los estudiantes de Ingeniería Electrónica Juan David Jamioy Cabrera (cod: 8943644) y Gabriel Alejandro Rodríguez Téllez (cod: 8943090) y "SISTEMA DE CLASIFICACIÓN PARA UN ROBOT COLABORATIVO UTILIZANDO TÉCNICAS DE APRENDIZAJE DE MÁQUINA" el cual fue desarrollado por los estudiantes de Ingeniería Electrónica María de los Ángeles Delgado Giraldo (cod: 8943648) y Juan Felipe Penagos Angrino (cod: 8945827).

Estos trabajos de grado fueron desarrollados bajo el marco del proyecto de investigación "APROXIMACIÓN A UNA ARQUITECTURA COGNITIVA PARA EL APRENDIZAJE Y GENERALIZACIÓN DE UN PROCESO DE CLASIFICACIÓN APLICADO EN UN ROBOT COLABORATIVO" el cual pertenece a la Convocatoria interna de proyectos "Por una universidad transformadora: Horizonte 2021-2025". Estos trabajos de grado comparten un núcleo común en lo referente a: 1) La tarea de identificación de objetos cúbicos por color; 2) La forma de adquirir la información de entrada (comandos gestuales y verbales) y 3) Las salidas del sistema (Realimentación auditiva, gráfica y control del robot UR3). Todos estos desarrollos fueron realizados de manera conjunta por los cuatro estudiantes trabajando en equipo.

Consideramos importante aclarar que la principal diferencia de los trabajos, la cual visibiliza el aporte individual de cada pareja de estudiantes, radica en el método de aprendizaje utilizado, ya que un trabajo utilizó una aproximación de la Arquitectura Cognitiva SOAR y el otro utilizó Redes Neuronales Artificiales (Perceptrón Multicapa MLP), para realizar el aprendizaje y la solución al problema de clasificación de objetos cúbicos propuesto.

Atentamente,



Alexander Martínez Álvarez, Ph.D.
Director

José Hernando Mosquera de la Cruz, M.Sc.
Codirector

Aprendizaje de máquina aplicado a un proceso de clasificación para un robot colaborativo

Maria de los Angeles Delgado-Giraldo, Juan Felipe Penagos-Angrino

***Resumen*— Se desarrolló un sistema basado en redes neuronales artificiales (RNA), el cual permite el aprendizaje y la ejecución de una tarea de clasificación de objetos cúbicos por color al robot colaborativo UR3 mediante interacción multimodal comandada por gestos y voz.**

Las pruebas se realizaron con siete sujetos de prueba a los cuales inicialmente se les realizó una encuesta para identificar el nivel de conocimiento en robótica e interacción multimodal. Posteriormente, se les explicó cuáles eran las diferentes formas para enseñarle al robot la tarea de clasificación por color de objetos cúbicos. Finalmente, se ejecutaron las pruebas en dos formatos; cuantitativas y cualitativas.

Las pruebas cuantitativas evaluaron 588 interacciones verbales, 21 comandos de voz, 252 interacciones gestuales, 63 interacciones multimodales y 63 clasificaciones por color de objetos cúbicos, obteniendo un total de acierto del 94.86% para comandos multimodales, un 96.83% para comandos de voz, un 80.95% para los comandos multimodales y 100% para las clasificaciones por color de objetos cúbicos.

Las pruebas cualitativas se realizaron mediante una encuesta de cinco preguntas evaluada con una escala *likert,* consultando a los usuarios sobre su experiencia respecto al desempeño de las interacciones verbales, gestuales, multimodales, realimentación y clasificación. Las encuestas mostraron una alta satisfacción sobre la arquitectura propuesta durante la interacción del usuario con el robot.

***Palabras Clave*—Aprendizaje de máquina, Clasificación, Comunicación multimodal, Interacción humano robot, Redes Neuronales Artificiales, Robótica colaborativa.**

I. INTRODUCCIÓN

Las industrias pertenecientes al sector productivo son las entidades que más apetecen la adquisición y desarrollo de nuevas tecnologías que puedan ser implementadas para mejorar la eficiencia y la capacidad de producción. Sin embargo, las industrias ubicadas en Sudamérica no parecen dar grandes pasos hacia la implementación de estas tecnologías cuyo propósito es dignificar el trabajo, pues en 2020 mientras que en Sudamérica solo se implementaron unos cientos de robots con inteligencia artificial, en Norteamérica, Europa y Asia hablan de miles de unidades robóticas [1].

El uso de un robot colaborativo por parte de un usuario depende en gran medida de la automatización en la programación de sus tareas que generalmente requiere ajustes específicos, tiempos prolongados de programación y reprogramación, que normalmente son actividades que un ingeniero experimentado debe realizar [2], [3]. Estos requerimientos pueden ser reducidos en gran medida si para la comunicación entre el usuario y el robot se implementa una interfaz intuitiva, capaz de ser utilizada por cualquier persona del común, que permita la asignación de tareas al robot colaborativo, las cuales puedan ser realizadas de manera independiente [4].

En la revisión bibliográfica realizada se encontraron distintas aplicaciones que utilizan algoritmos de aprendizaje de máquina con objetivos específicos: en [5] se presenta un sistema que utiliza un algoritmo con redes neuronales convolucionales (CNN) para clasificar las frutas según su madurez. [8] se hace la clasificación de terrenos desconocidos con robots móviles, utilizando múltiples modalidades de datos sensados y algoritmos de clasificación, que al combinarse crean un algoritmo de aprendizaje conjunto con un bajo consumo de tiempo. Por último, en [9] se encontró un recurso muy valioso, pues se realizaron trabajos con un robot de servicio y una fusión multimodal, proponiendo una técnica mejorada de interacción, sintetizando los datos visuales y sonoros. La comunicación multimodal es sin duda una herramienta muy útil a la hora de brindar nuevas opciones a las personas en la comunicación, ya que permite la intervención de diferentes canales para brindar un mensaje más completo y con una mayor naturalidad. Es por esto que la técnica implementada en [9] destaca por encima de los demás trabajos relacionados. Frente a los demás trabajos relacionados se encontró gran cantidad de sistemas que aplican interacciones auditivas, visuales o sensitivas que potencian su desempeño haciendo uso de la teleoperación y distintos métodos de aprendizaje.

Este trabajo tuvo como objetivo lograr la implementación de un algoritmo de clasificación de objetos cúbicos por color, para un robot colaborativo UR3, utilizando técnicas de aprendizaje de máquina y una interacción multimodal humano-robot. Se destaca la naturaleza del problema de aprendizaje al tener múltiples entradas y salidas, la posibilidad de modificar la estructura interna de la red y la facilidad de implementación en Python articulándose con el resto de algoritmos de audio y video implementados. Se decidió hacer uso de un modelo de redes neuronales artificiales y un sistema de interacción multimodal comandado por gestos y voz.

En las secciones 2 y 3 de este documento se presenta la metodología y la descripción del sistema, respectivamente. Los resultados obtenidos de las pruebas realizadas a los usuarios, así como sus respectivas tablas y porcentajes de desempeño se presentan en la sección 4, la discusión se presenta en la sección 5; y en la sección 6 se exponen las conclusiones y se proponen algunos trabajos futuros.

II. METODOLOGÍA

La metodología seguida para el desarrollo del sistema propuesto se basó en cuatro componentes principales. En primer lugar, se especificó la tarea que debía llevar a cabo el robot colaborativo UR3, la cual se definió para clasificar objetos cúbicos por color; se definieron los requerimientos y las características del espacio de trabajo y los objetos utilizados, así como también la interacción humano robot que se da por medio de gestos y comandos, ya sea de voz o multimodales. Por último, se hizo una validación del sistema, evaluando su desempeño mediante un protocolo de pruebas, usuarios con diferentes perfiles y ciertas métricas establecidas. A continuación, se describen las partes que componen la solución propuesta.

Espacio de trabajo, Tarea de clasificación, y Objetos: para definir el espacio de trabajo, primeramente se tomaron medidas generales teniendo en cuenta la ubicación fija y las limitantes de desplazamiento del robot UR3, pues por la posición apuntando hacia abajo en que se necesita la pinza de agarre, el espacio de trabajo sobre la que se puede operar es menor; el tamaño de la retícula es proporcional al volumen de trabajo del robot, por lo que para aumentar el espacio de trabajo se requiere aumentar la capacidad del robot para alcanzar zonas más lejanas, pero por la complejidad y los alcances de este proyecto de grado se limita el espacio de trabajo. Teniendo esto en cuenta, se definió el espacio de trabajo por una mesa metálica de 90cm por 50cm, sobre la cual se demarca un rectángulo como zona de trabajo de 33.2cm por 23.2cm, con sus bordes en color rojo de 1cm de grosor, se seleccionó el color rojo para obtener un mayor contraste con el color metálico de la mesa. En la siguiente figura puede apreciarse el espacio de trabajo comprendido por la mesa, el robot manipulador, el soporte de la cámara y el teaching pendant, además de la distancia de operación sobre la cual deben ubicarse los usuarios para poder interactuar con el sistema de forma segura.

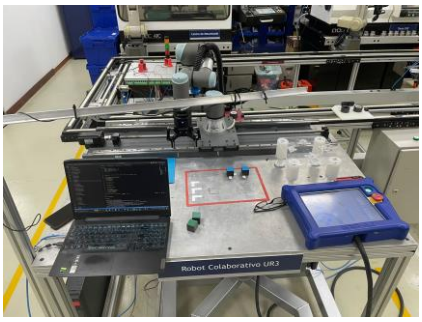


Figura 1. Espacio de trabajo

El espacio de trabajo se subdividió en una retícula de cuatro columnas y tres filas, quedando un total de doce zonas, en las cuales se organizan tres cubos verdes y tres cubos azules, de 2cm de lado, impresos en 3D con material PLA. La cantidad de zonas se definió considerando las dimensiones de los cubos, garantizando que entre sí, no se encuentren muy cercanos y además, que cuando la pinza los tome, esta no vaya a colisionar con cubos aledaños. Esta división gráfica del espacio de trabajo en doce zonas se realiza con el fin de brindarle al usuario una herramienta para ubicarse mejor en el espacio de trabajo y poder dar instrucciones verbales más específicas, al solicitar al robot el desplazamiento de cierta pieza a una zona en específico. Este tipo de clasificación no se encuentra dentro de los trabajos referenciados y se debe a que muchos de los objetos que se desean identificar dentro de las imágenes no poseen forma fija todo el tiempo, lo que dificulta la tarea de subdividir el área de trabajo para así solo contener un objeto por zona, como es el caso de esta investigación.

La tarea de clasificación está orientada a agrupar los cubos por color en la primera y última columna, como se muestra en la figura 2. La tarea de clasificación puede variar la cantidad de cubos por color de uno a tres. Esta tarea se definió teniendo en cuenta el nivel de dificultad, pues al ser un proyecto de grado e implementar inteligencia artificial, la complejidad al utilizar una tarea más compleja como apilado o ensamble, podría superar el alcance del proyecto de grado.

Se utilizó un sistema de visión artificial que se encarga de reconocer los cubos y el espacio de trabajo del que dispone el robot, haciendo uso de una cámara en espectro visible Logitech C920, con una resolución estándar de 640x480 pixeles y corrección de iluminación automática, con las condiciones de ruido e iluminación controladas que ofrece el laboratorio de experimentación utilizado para las pruebas.

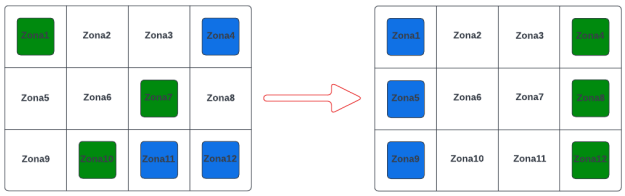


Figura 2. Resultado esperado de la tarea de clasificación.

Definición de la interacción gestual: se simula la pinza de agarre que tiene el robot para facilitar la comunicación de las instrucciones de selección o desplazamiento de un cubo dentro del espacio de trabajo. Se definió el gesto “seleccionar”, el cual se encarga de indicar al robot el objeto que debe ser tomado, por medio del sistema de visión por computador que detecta ciertos puntos específicos para determinar si la mano se encuentra abierta, o sea que las yemas de los dedos tengan una distancia considerable entre sí o si la mano se encuentra cerrada las yemas de los dedos deben estar juntas. El comando multimodal a partir de la expresión “toma este cubo y ponlo aquí” identifica que la siguiente acción del sistema es tomar la información entregada por la visión para identificar las dos ejecuciones del gesto “seleccionar”, que son el cubo y la zona objetivo. Este gesto debe realizarse en pararelo con el comando de voz, ya que el gesto solo indica cuál cubo debe desplazarse.

Definición de la interacción verbal: el módulo de audio permite el reconocimiento de la voz del usuario e identifica comandos verbales. Se definió un diccionario con un conjunto de expresiones verbales de uso cotidiano, para identificar actividades provenientes de una interacción humano robot como el saludo, aprendizaje o ejecución de una tarea. Dichas expresiones permiten definir el movimiento, la posición y el color de cada cubo, procurando evitar expresiones que puedan alterar el lenguaje. Se definió un comando de voz con una

estructura como la del siguiente ejemplo: “toma el cubo verde dos y ponlo en la zona once”, el cual permite seleccionar un cubo e indicar hacia donde se desea desplazar. Se debe aclarar que la enumeración de los cubos está programada para asignar los índices numéricos de la siguiente forma: para cada color, desde la fila inferior de la zona de trabajo hasta la fila superior, un barrido va enumerando cada cubo en orden ascendente, pero si se presentase el caso de haber dos cubos del mismo color en la misma fila, entonces el sistema nombrará los cubos de manera ascendente de izquierda a derecha.

Definición de la interacción multimodal: se definió un comando multimodal por medio de gestos y comandos de voz, permitiendo seleccionar y mover un cubo de posición dentro del espacio de trabajo. Este comando cuenta con la siguiente estructura: *“Mira, este cubo (indica con gesto el cubo a seleccionar), ponlo aquí (indica con gesto la zona a la que se quiere trasladar)”*. Esta estructura no se modifica para ninguno de los posibles comandos multimodales; sin embargo, sí pueden usarse otras expresiones verbales que pertenezcan al diccionario de palabras para así no caer en la monotonía de usar siempre las mismas palabras.

Validación del sistema: se definió un conjunto de métricas para evaluar cada una de las etapas del sistema implementado y se realizó una prueba tanto cualitativa como cuantitativa. La cualitativa se realizó por medio de encuestas, midiendo así la satisfacción del usuario frente a cada tipo de interacción presente en el sistema. La prueba cuantitativa mide el desempeño del sistema en el reconocimiento y ejecución de comandos gestuales, verbales y multimodales durante la interacción humano robot. En la fase experimental, un grupo de siete personas con diferentes perfiles, sin realizar un entrenamiento previo, ejecutaron las pruebas cuantitativas con un total de 588 expresiones verbales, 63 veces el comando de voz, 63 veces el comando multimodal y 27 clasificaciones por color utilizando el sistema completo, mientras que cualitativamente se realizaron un total de 35 preguntas evaluadas con la escala *Likert de cinco niveles* [10], para identificar la percepción de los usuarios frente al desempeño del sistema propuesto. El conteo del número de repeticiones muestra la cantidad total de veces que dicha prueba fue llevada a cabo, sumando las veces que cada usuario realizó la prueba, por lo que no existe la posibilidad de armar 588 expresiones verbales diferentes, sino que cada usuario ejecutó un total de 84 expresiones verbales, 9 comandos de voz, etc.

III. DESCRIPCIÓN DE LA ARQUITECTURA

En la figura 3 se presenta el sistema propuesto, que está constituido por dos etapas; una de procesamiento offline (*Rojo punteado*) y una de procesamiento online (*Azul punteado*).

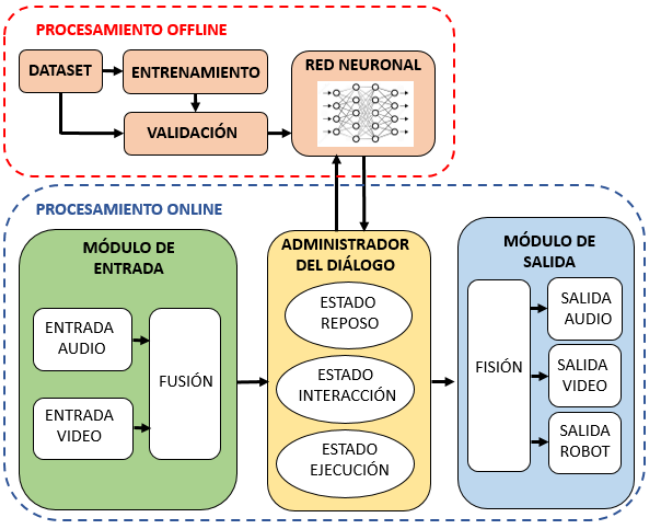


Figura 3. Arquitectura del sistema propuesto.

En la etapa de procesamiento offline se encuentra el modelo de aprendizaje. Como modelo computacional de aprendizaje se eligió el de Redes Neuronales Artificiales (RNA) de tipo perceptrón multicapa (MLP) para realizar la clasificación de los objetos cúbicos por color. Una vez entrenada esta red, se almacena para poder ejecutarse posteriormente en la etapa de procesamiento online. Debido a que el sistema presenta múltiples entradas y múltiples salidas distribuidas en vectores binarizados de doce posiciones y tiene un número finito de casos posibles; una red neuronal se ve potenciada cuando los datos para aprender tienen esa naturaleza [11], debido a que se puede variar el número de capas ocultas, neuronas por capa y algoritmos de optimización para obtener los mejores resultados posibles utilizando la estructura más simple.

La etapa de procesamiento online está compuesta por un módulo de entrada (encargado de capturar y mezclar las señales de audio y video), un administrador del diálogo (encargado de coordinar la interacción con base en tres estados: reposo, interacción y ejecución) y, finalmente, un módulo de salida (encargado de dividir y controlar la realimentación auditiva, visual y accionar el robot).

El sistema fue implementado utilizando un sistema operativo *Windows 10 Pro* de 64 bits, utilizando como IDE *Visual Studio Code* y lenguaje de programación *Python 3.9.12*; librerías propias de *Python* como *scikit-learn* [12] de aprendizaje automático gratuita, *itertools* [13], *joblib* [14], *PyAudio* [15], *Speech Recognition* [16], *OpenCV* [17], *MediaPipe Hands* [18], *Pyttsx3* [19], como también librerías que fueron creadas con base en el lenguaje de programación *URScript* [20], desarrollado por el fabricante *Universal Robots*, tales como *controlUR3* y *comunicacionUR3* [21], [22]. Todas estas librerías presentan facilidad de articular muy bien los diferentes módulos implementados para el procesamiento de cada etapa, sumando la posibilidad de exportar y guardar la red neuronal una vez entrenada para su posterior uso.

A continuación, se presenta una descripción detallada de cada uno de los módulos implementados:

A. PROCESAMIENTO OFFLINE

Esta etapa está compuesta por los módulos: *dataset*, *entrenamiento*, *validación* y *almacenamiento*; los cuales son explicados con detalle a continuación.

1) Dataset

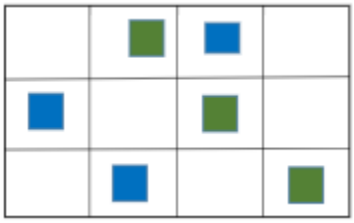
El *dataset* propuesto contiene todas las posibles posiciones que pueden tomar los cubos, que pueden ser mínimo dos (un azul y un verde) y máximo seis (tres azules y tres verdes), en este orden de ideas, para obtener todos los escenarios posibles, se utilizó la permutación sin repetición; $P_k^n = \frac{n!}{(n-k)!}$, donde n es el número total de elementos y k es el número de objetos que vamos a permutar. Para efectos de este sistema, el valor de *n=12* que corresponden al total de casillas del tablero y el valor de *k=2,3,4,5,6* representa el número de cubos que pueden estar presentes en el tablero, obteniendo un total de 893.772 posibles entradas al sistema, como se presenta en la Tabla 1.

Tabla 1. Cantidad de escenarios posibles del sistema

Permutación (n,k)	Azul	Verde	Valores (n,k)	Equivalente en %
P(12,2)	1	1	132	0.01%
P(12,3)	1	2	1.320	0.15%
P(12,4)	1	3	11.880	1.33%
P(12,3)	2	1	1.320	0.15%
P(12,4)	2	2	11.880	1.33%
P(12,5)	2	3	95.040	10.63%
P(12,4)	3	1	11.880	1.33%
P(12,5)	3	2	95.040	10.63%
P(12,6)	3	3	665.280	74.44%
Total de escenarios posibles			**893.772**	**100.00%**

Una vez definido las posibles ubicaciones de los cubos, se generaron los 893.772 datos de entrada utilizando la función *permutations( )* que viene de la librería *itertools* [13]. De igual modo, se crearon los datos de salida que serían la misma cantidad de los datos de entrada, pues estos representan la clasificación de los cubos para los 893.772 escenarios posibles, siguiendo el patrón presentado en el tablero derecho en la figura 1; los cubos azules deben quedar alineados en la primera columna y los cubos verdes en la última columna. Así se obtiene el conjunto de datos de entrada y salida necesarios para realizar el entrenamiento de la red neuronal MLP.

Inicialmente se realizó una sub-división del espacio de trabajo convirtiéndola en un tablero de 3x4 zonas de interacción, en las cuales se colocan de forma aleatoria tres cubos azules y tres cubos verdes; como los datos son representados de forma vectorial, para binarizar el tablero asignamos el número 0 a los espacios vacíos y el número 1 en los espacios que contienen un cubo de color, obteniendo un vector binario de 24 posiciones que representa la ubicación y el número de cubos, como se presenta en la figura 4.



Cubos Azules												Cubos Verdes											
0	0	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1

Figura 4. Representación binaria del espacio de trabajo.

2) Entrenamiento

El entrenamiento de la red neuronal MLP utilizó las librerías *scikit-learn*, por medio de la función *MLPClassifier* ( ) [12] de *Python*. La red neuronal propuesta tiene 24 neuronas en la capa de entrada, 10 neuronas en la capa oculta y 24 neuronas en la capa de salida. Se utiliza el algortimo de optimización *adam*, una función de activación *tanh* (tangente hiperbólica) para cada neurona y finalmente, el algoritmo se entrenó por un total de 1000 iteraciones.

Los criterios para seleccionar la arquitectura propuesta fueron los siguientes: 1) se definió la entrada y salida de 24 neuronas debido al tamaño del vector binario que representa el tablero con los cubos antes y después de ser clasificados, 2) se realizó un proceso empírico de reducción progresivo de capas y neuronas en las capas ocultas, empezando con dos capas ocultas con cien neuronas cada una y terminando con una capa con diez neuronas, ya que se obtenían los mismo resultados (100%) al clasificar los datos de validación, esto con el fin de llegar a la configuración más simple posible que obtuviera el mejor resultado, 3) se probaron los algortimos de optimización sgd, adam y lbfgs, de igual forma las funciones de activación logistic y tanh, donde los mejores resultados experimentales fueron obtenidos por el algortimo adam y la función de activación tanh, 4) se realizó un barrido para definir el numero de iteraciones empezando con 10.000 iteraciones, pero el sistema automaticamente hacia paradas tempranas para evitar el sobreentrenamiento en valores de aproximadamente 1.000 iteraciones.

3) Validación

Los porcentajes de la partición del dataset para entrenamiento y validación se ajustaron con la finalidad de definir una proporción que garantizara un buen desempeño. Inicialmente se utilizó una asignación del 40% para entrenamiento y 60% para validación obteniendo un desempeño del 100% para el conjunto de validación. Posteriormente, se realizó una disminución al 20% del conjunto de datos para entrenamiento, que corresponde a 178.754 posibles situaciones y 80% del conjunto de datos para la validación, que corresponde a 715.018 casos posibles, conservando un desempeño del 100% en el conjunto de validación. Finalmente, al disminuir al 10% el conjunto de datos de entrenamiento y 90% para validación, se obtuvo un desempeño del 97%, por está razón se conservó como mejor solución la red obtenida con el 20% - 80% para aprendizaje y validación respectivamente, teniendo en cuenta el comportamiento representado en la figura 5.

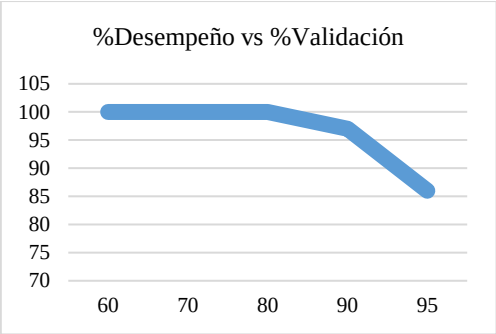


Figura 5. Desempeño respecto al porcentaje de datos para validación.

Se utilizó la función *train_test_split* ( ) [23] que permite dividir arreglos o matrices en subconjuntos de datos seleccionados de forma aleatoria.

4) Almacenamiento de la Red Neuronal

La red neuronal entrenada se almacena utilizando la librería *joblib* y la función *dump( )* [14], la cual únicamente necesita dos parámetros para funcionar: 1) El modelo para guardar y 2) el nombre como se desea almacenar la red neuronal (la cual queda con extensión .SAV).

B. PROCESAMIENTO ONLINE

Esta etapa está compuesta por los *módulos de entrada*, *administrador del diálogo* y *módulo de salida* (ver figura 3), los cuales son descritos a continuación:

1) Módulo de Entrada

Este módulo consta de un submódulo de *entrada de audio* responsable de reconocer los comandos de voz, un submódulo de *entrada de video* responsable de detectar los colores, la posición de los cubos en una cuadrícula y el gesto de "seleccionar" y "aquí", utilizados para la asignación de tareas por comandos multimodales. Y un submódulo de *fusión multimodal* que es el responsable de priorizar y fusionar señales de voz y gestos.

a) Entrada de audio

La estructura del comando de voz implementado está compuesta por un conjunto de seis acciones: tomar, objeto, color, índice, dejar y ubicación. La finalidad del comando es seleccionar un cubo indicando su color e índice y definir la zona hacia la que se desea desplazar. El comando "*toma el cubo verde dos y ponlo en la zona nueve*", es un ejemplo, que se puede construir con base en un diccionario de 28 expresiones verbales que se presentan en la tabla 2. El equipo utilizado para capturar el audio era una diadema Logitech G935, mientras que el nivel de ruido en el ambiente de trabajo se debía a conversaciones de fondo y a la puesta en marcha de los equipos de ingeniería mecánica que comparten el centro de automatización con el UR3. Sin embargo, después de realizar un gran número de pruebas, se descubrió que con desactivar el micrófono de la diadema el sonido ya no es captado como un tipo de señal por el sistema.

Tabla 2. Estructura del comando de voz.

Acciones que componen la estructura del comando de voz implementado						
Tomar	**Objeto**	**Color**	**Índice**	**Dejar**	**Ubicación**	
['toma', 'sujeta', 'coge', 'agarra']	['cubo']	['azul', 'verde']	['uno', 'dos', 'tres']	['ponlo', 'colócalo', 'déjalo', 'muévelo']	['zona']	['uno', 'dos', 'tres', 'cuatro', 'cinco', 'seis', 'siete', 'ocho', 'nueve', 'diez', 'once', 'doce']

Con respecto a la Tabla 2, existe un comando de voz que permite el paso del estado de reposo a cualquiera de los estados de aprendizaje o ejecución, y es la palabra 'hola'. Sin embargo, dado que esta palabra no se encuentra contemplada dentro de la estructura del comando de voz, no ha sido incluida en la anterior tabla. Un caso similar sucede con los verbos 'aprender´ y 'ejecutar', ya que se encuentran configurados como comandos de activación que permiten migrar de un estado del sistema a otro.

Por medio de una función de transcripción de voz a texto se obtiene una cadena de caracteres, que a su vez usa la función Text2num 2.5.0 [25] para transformar los números de texto compuesto a números en dígitos. Luego se realiza una segmentación para separar las palabras que componen la cadena de caracteres reconocidos en posiciones distintas. Se compara palabra por palabra con un diccionario de 28 palabras (tercera fila de la tabla 2) y se excluyen las palabras del vector que no están en el diccionario, y finalmente, se realiza una validación con las acciones de la estructura del comando de voz, para verificar si el vector no contiene una expresión incluida en cada categoría o no fue reconocido en el orden correcto para dar sentido al comando; en estos casos, el vector será clasificado como un comando no válido. El reconocimiento de voz independiente del hablante se realizó utilizando las bibliotecas *PyAudio 0.2.13* [15] y *Speech Recognition 3.9.0* [16], la cual hace uso del motor de reconocimiento de voz *Google Speech Recognition* [24].

b) Entrada de video

Se utilizó la librería *OpenCV-Python 4.7.0.68* [17] para el procesamiento digital de imágenes. La adquisición de las imágenes se realizó con una cámara en espectro visible Logitech C920, con una resolución estándar de 640x480 pixeles y corrección de iluminación automática en unas condiciones controladas de ruido e iluminación tipo oficina, la cual fue ubicada al costado de la mesa opuesto al robot, a 50 cm sobre el espacio de trabajo como se ilustra en la figura 6, para la cual se describen los objetos: 1) robot colaborativo UR3, 2) cámara Logitech C920, 3) objetos cúbicos 2x2x2cm impresos en 3D material PLA y 4) espacio de trabajo dimensiones 33.2x23.2 centímetros.

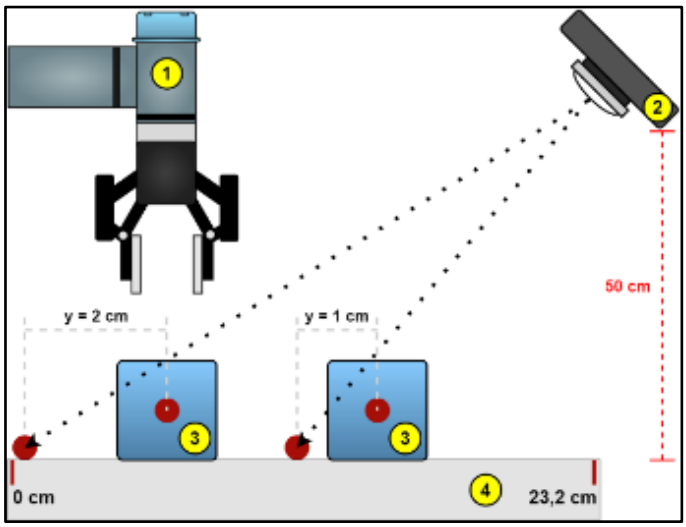


Figura 6. Ilustración del espacio de trabajo.

Para evitar oclusiones de los cubos por parte del robot, la cámara se ubicó al lado opuesto del robot y no sobre la parte superior del espacio de trabajo (figura 6). En la figura 7 se presenta el robot tomando un cubo de color verde ubicado en la zona 10 y el sistema de visión artificial por computador; al lado derecho de la pantalla del computador se observa la imagen original capturada por la cámara y al lado izquierdo el espacio de trabajo con la realimentación gráfica de la retícula.

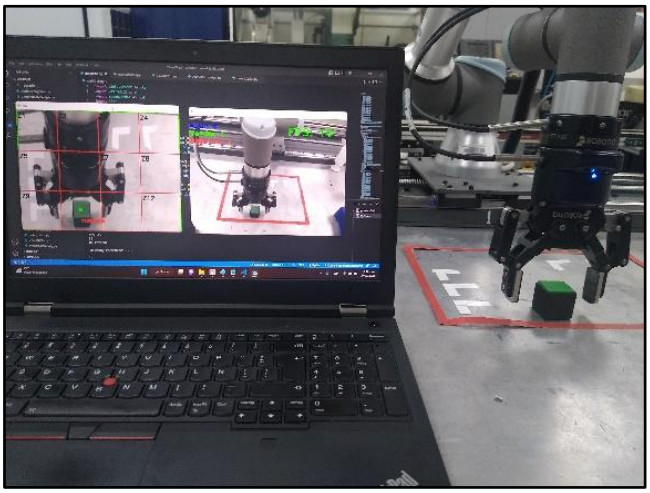


Figura 7. Espacio de trabajo y procesamiento de video.

Para la segmentación del espacio de trabajo se utilizó una binarización en el espacio de color HSV (Matiz, Saturación y Valor), que define un modelo de color en términos de sus componentes. Para este caso, se deseaban obtener dos umbrales para el color rojo, siendo H [0 – 5], S [100 – 255] y V [20 – 255] para el primero y H [175 – 179], S [100 – 255] y V [20 – 255] para el segundo, rangos de valores definidos de manera empírica. Con la imagen binaria resultante se logra localizar los pixeles que conforman el rectángulo rojo que demarca el espacio de trabajo como se presenta en la figura 8 (Derecha).

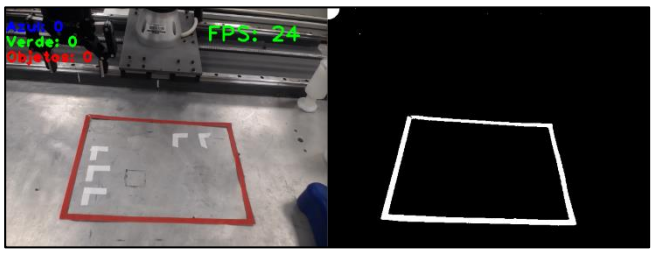


Figura 8. Filtrado por color y binarización del espacio de trabajo.

Los pixeles en blanco que se encuentran en el borde superior izquierdo de la anterior figura para la imagen del lado derecho son causados por el rango de visión de la cámara, ya que al proyectarse mas allá del espacio de trabajo establecido, puede estar realizando lecturas de tonalidades rojas que generen este efecto en la binarización.

Sobre la imagen binarizada, se realizó una extracción de los bordes acompañada de una aproximación poligonal de cuatro lados que da como resultado dos rectángulos: uno correspondiente al borde externo y otro correspondiente al borde interno de la imagen binarizada. Luego, se seleccionó el rectángulo con menor área y se tomaron los puntos de los vértices para identificar las esquinas del espacio de trabajo.

Con estos puntos ya encontrados de acuerdo con su posición en pixeles, se realiza la transformación de perspectiva para generar una imagen con resolución de 640x480 pixeles que simula el espacio de trabajo como si se observase de manera perpendicular a esta superficie, la matriz de rotación / traslación requerida para esta transformación se presenta a continuación:

$$M = \begin{bmatrix} -2{,}6163 * 10^{0} & -1{,}8080 * 10^{-1} & 1{,}4692 * 10^{3} \\ -5{,}3953 * 10^{-1} & 2{,}5701 * 10^{0} & 5{,}5277 * 10^{1} \\ 2{,}0502 * 10^{-4} & 6{,}6274 * 10^{-4} & 1{,}0000 * 10^{0} \end{bmatrix}$$

En la figura 9 (Izquierda) se presenta la captura original de prueba con la cámara, en la que se observa sobre la mesa una retícula que delimita el espacio de trabajo y dos objetos, mientras que en la figura 9 (Derecha), se observa la respectiva transformación de perspectiva del espacio de trabajo demarcado. Para reducir el impacto de la deformación presente en los objetos cúbicos de color azul y verde con el proceso de transformación de perspectiva, se optó por cubrir sus caras laterales con color negro, con el objetivo de facilitar la detección de la cara superior y asegurar que los centros encontrados correspondan únicamente a la cara superior de los cubos.

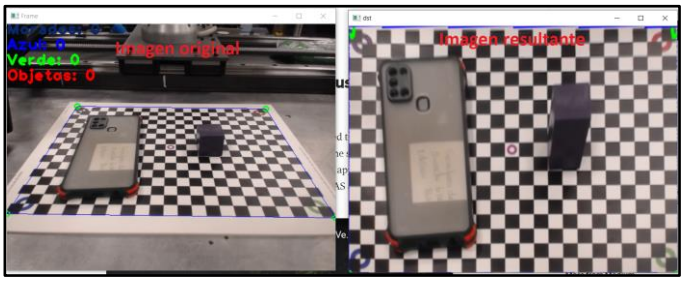


Figura 9. Transformación de perspectiva.

Una vez se tiene la imagen con transformación de perspectiva del espacio de trabajo, se utilizan dos rangos de umbrales más

para los cubos de color: al color azul corresponde H [100 – 107], S [150 – 255] y V [20 – 255], mientras que al color verde corresponde H [47 – 99], S [50 – 230] y V [20 – 230]. En la figura 9 (Izquierda) se puede apreciar la imagen binarizada utilizando los rangos de umbrales para detectar únicamente los objetos cúbicos azules. En la figura 10 (Derecha) se puede observar la imagen con la transformación de perspectiva.

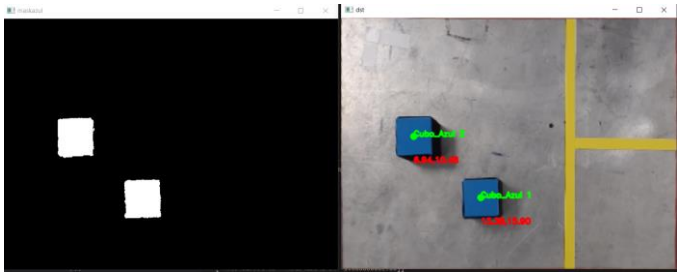


Figura 10. Binarización (izquierda) y cubos detectados (Derecha).

Sobre las imágenes binarizadas se realiza la aproximación poligonal de las aristas de la cara superior de cada cubo. Posteriormente, se identifican los centros, obteniendo así un vector final que describe la cantidad de cubos por color, las coordenadas del centro, la ubicación en las diferentes zonas de la retícula y el índice de identificación que toma el valor de 1, 2 o 3, pues este se asigna según la cantidad de cubos de cada color y la posición en que se encuentren. Con respecto al cálculo del centroide, experimentalmente se identificó una corrección geométrica del centroide de acuerdo a los cambios registrados en ciertas zonas de la imagen de acuerdo a los datos reales referentes a la identificación y ubicación del centro.

Experimentalmente se observó una variación entre la ubicación real en el espacio de trabajo para los centros de los cubos y la calculada por el algoritmo propuesto. Esto puede deberse a la proyección visual de la cámara respecto al espacio de trabajo, o principalmente a la falta de un proceso de calibración de la cámara que permita adaptar los parámetros de funcionamiento a las necesidades del entorno.

Para corregir este problema se tomaron mediciones milimétricas sobre la ubicación real y calculada, para realizar un ajuste correctivo como se presenta en la figura 11. De esta manera, dependiendo de en cuál de las 5 zonas mostradas en la figura 11 se ubicase el cubo, así era la cantidad de centímetros que se sumaban o restaban a los valores obtenidos del cálculo del centro de los cubos.

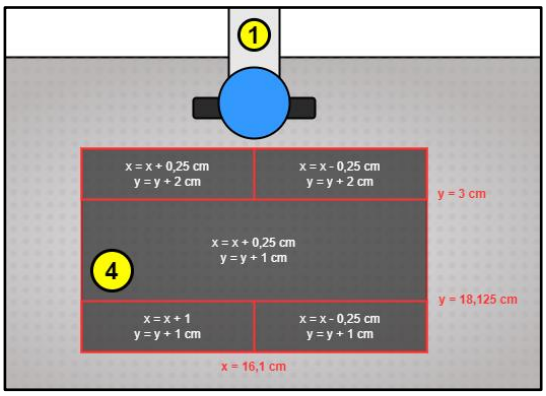


Figura 11. Ilustración de las sub-zonas con sus factores de conversión.

Es importante mencionar que una vez realizado el cálculo de los centro de los cubos desde la perspectiva de la cámara, se necesita de una transformación para que las coordenadas vistas por la cámara sean asimiladas y entendidas por el manipulador.

Se utilizó la librería *MediaPipe Hands* [18] para realizar la detección y esqueletización de la mano del usuario por medio de inteligencia artificial, esta arroja como salida un conjunto de 20 puntos que describen las articulaciones de la mano. Después de reconocer las puntas de los dedos índice y pulgar, se trazó una línea entre estos, se calculó el punto medio y se creó un umbral de distancia mínima para determinar si la postura de la mano es abierta o cerrada, estableciendo empíricamente un umbral de 30 pixeles para identificar la postura como "cerrado" (figura 12, derecha), y en el caso contrario como “abierto” (figura 12, izquierda). Así mismo, se dibuja en pantalla una línea entre ambas puntas, en la cual se muestra centralmente un punto que sirve de guía al usuario para seleccionar el cubo o la zona.

Figura 12. Identificación de la postura “abierto” y “cerrado” implementada.

Una vez identificada la postura “abierta” o “cerrada”, se implementó el gesto “seleccionar”, el cual se activa al juntar las yemas de los dedos y luego separarlas dentro de una zona atractiva para, de este modo, seleccionar el cubo que se desea mover o una zona hacia la que se desea trasladar un cubo.

c) Fusión multimodal

Este sub-módulo prioriza e integra las señales verbales y gestuales obtenidas como salidas de los sub-módulos de *Entrada de audio* y *Entrada de video*.

Se implementó el comando multimodal: *“Mira este cubo (gesto seleccionar sobre un cubo,) ponlo aquí (gesto seleccionar sobre una zona de la retícula)”*, el cual permite ejecutar la selección y movimiento de un cubo hacia una zona específica. Finalmente, la fusión multimodal calcula en paralelo la interacción verbal y gestual, de manera que no tiene sentido el comando verbal si no va acompañado por el gesto adecuado.

2) Identificador de estados

Este módulo recibe la información multimodal proveniente del *módulo de entrada* y establece una comunicación con la red neuronal MLP entrenada offline. Con base en la información recibida, se encarga de definir las acciones a realizar dependiendo del estado de interacción: *estado de reposo, estado de interacción y estado de ejecución*. Para cualquiera de los dos estados diferentes al de reposo, siempre se espera que a manera de entrada al sistema se cuente con señales de video y de audio, ya que así se hace posible la implementación de la multimodalidad.

a) Estado reposo

En la etapa de reposo se tiene el robot encendido y se activa al escuchar la palabra *"Hola"* pronunciada por el usuario. Mediante un menú de opciones selecciona la acción a realizar: *"Interactuar"* para modificar la posición de los cubos en el espacio de trabajo o *"Clasificar"* para ordenar los cubos con base en la tarea de clasificación definida (cubos azules en la primera columna y cubos verdes en la cuarta columna).

b) Estado interacción

El estado de interacción permite al usuario realizar movimiento de los cubos en la zona de trabajo y se activa al pronunciar la expresión verbal *"interactuar"* en el estado de reposo. Una vez activado el estado de interacción, el sistema calcula el número total de cubos, e identifica el índice, el color y la posición para cada uno de los cubos presentes en el espacio de trabajo. Posteriormente, los usuarios pueden realizar un total de interacciones correspondiente al total de cubos identificados, permitiendo al usuario dar instrucciones para mover cada uno de los cubos utilizando los comandos definidos. Al terminar las interacciones, el sistema realimenta al usuario por audio pronunciando *"interacción terminada"*, volviendo al estado de reposo.

c) Estado ejecución

El estado de ejecución lleva a cabo la clasificación de los cubos con base en la tarea de clasificación definida. Inicialmente el sistema calcula el número total de cubos, el índice, el color y la posición para cada uno de los cubos presentes en el espacio de trabajo, posteriormente calcula la *representación binaria* para los cubos ubicados en el espacio de trabajo, seguido a esto, envia esta a la *red neuronal* (Entrenada Offline), y esta le devuelve otra *representación binaria* que contiene la solución de la clasificación para los cubos presentes en el espacio de trabajo. Para realizar la ejecución de la solución, el sistema inicialmente verifica que ninguna de las zonas de ubicación de destino estén ocupadas; en caso de que un cubo ocupe una posición destino, el sistema procede a retirar el cubo de esa zona y llevarlo a una zona donde no cause conflicto durante la ejecución de la tarea, en este punto envia acciones de control para mover cada uno de los cubos a su posición destino de forma secuencial y ubicando un cubo a la vez, cuando complete la tarea, el sistema realimenta auditivamente al usuario con el siguiente mensaje: *"la tarea se ejecutó con éxito",* y posteriormente regresa al estado de reposo. Gracias a la forma en la cual fue desarrollado el sistema, siempre que este reconoce una entrada del usuario y al evaluarla detecta que no hay errores en su representación, se tiene un 100% de desempeño, ya que mientras no se presenten errores en la lectura del comando, el robot colaborativo está en la capacidad de ejecutarlo. También, internamente al nivel del entrenamiento y la validación, utilizando el porcentaje de validación descrito en la figura 5 se obtiene un 100% en el desempeño, lo que da a entender que puede clasificar todos los escenarios posibles.

3) Módulo de salida

Este módulo está compuesto por los siguientes sub-módulos: *fisión multimodal* encargado de desagregar y dividir las instrucciones para cada modalidad, *salida de audio* encargado de realizar la realimentación auditiva al usuario, *salida de video* responsable de la realimentación visual al usuario, y *salida del robot* que controla la cinemática y el accionamiento de la pinza una vez se han validado las condiciones de seguridad del robot.

a) Fisión Multimodal

Este sub-módulo recibe una instrucción multimodal proveniente del administrador del dialogo y la divide en un conjunto de comandos individuales que entrega a los sub-módulos: *salida de audio*, *salida de video* y *salida del robot*.

b) Salida de audio

Es el sub-módulo encargado de realizar la realimentación auditiva al usuario, ya sea para saludar, solicitar información al usuario, indicarle lo que va sucediendo en el proceso o validar una acción. El diccionario de síntesis de voz se presenta en la tabla 3, y está compuesto por 15 expresiones. Para realizar la síntesis de voz se utilizó la librería de conversión de texto a voz *Pyttsx3 2.90* [19], se seleccionó el intérprete de voz en español y para establecer una forma ordenada de reproducir las oraciones se crearon dos tipos de expresiones para reproducir, una es *solicitudes* y la otra es *respuestas.*

Tabla 3. Diccionario de síntesis de voz.

Función	Síntesis de Voz
solicitudes	'¿Quieres interactuar con el sistema o clasificar los cubos?'
respuestas	'Hola', 'Hola como estás', 'Hola, que tal', 'Es bueno volver a verte', 'Regresaste', '¿Qué tal?'
	'La tarea se ejecutó con éxito'
	'Interacción terminada'
	'La tarea no fue reconocida, por favor repita nuevamente'
	'Siguiente instrucción'
	'Error, comando de voz no identificado'

Como se puede observar en la anterior tabla, se han definido algunos mensajes que se le dicen al usuario cuando se presenta un error en el sistema o cuando se pone en operación el mismo. Con respecto al mensaje de error, este se muestra en pantalla cada vez que el sistema no reconoce un comando del usuario, no visualiza correctamente la cantidad de cubos o pierde la correcta distribución de los índices de los cubos al inicio de la ejecución. Este mensaje de error se muestra al usuario con el fin de darle otra oportunidad para ingresar de nuevo la entrada y no tener que refrescar el sistema. En cuando a las respuestas del sistema, estas se reproducen de manera aleatoria dependiendo del estado en que se encuentre el sistema.

c) Salida de video

Este sub-módulo es el responsable de brindar una realimentación visual al usuario. Durante la interacción, en la pantalla del computador utilizado (Figura 13), se puede observar el espacio de trabajo que es captado con la cámara, también se pueden observar las coordenadas del centro de cada cubo, su etiquetado, la división del espacio en 12 zonas y el índice de cada cubo.

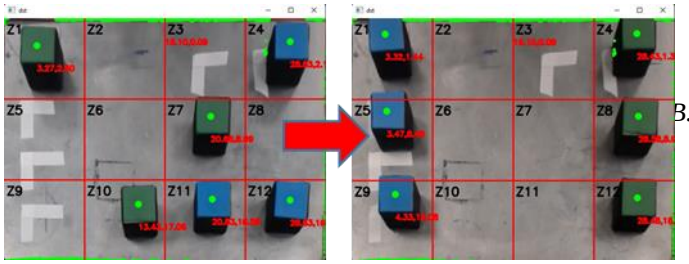


Figura 13. Realimentación gráfica del sistema.

d) Salida del robot

Este sub-módulo se encarga de controlar la cinemática y el accionamiento de la pinza una vez se han validado las condiciones de seguridad del robot. El control del movimiento y accionamiento de la pinza para el robot colaborativo UR3 se realiza mediante una comunicación TCP/IP en una red LAN, a la cual debe estar conectado también el equipo de cómputo que corre los algoritmos, estableciendo una comunicación cliente/servidor tipo *socket* con IP y puerto estáticos.

Este proyecto de grado se apoyó en los desarrollos presentados en [21], [22], los cuales proporcionan las librerías *comunicacionUR3.py* y las funciones *Gripper.activate()*, *Gripper.close()*, *Gripper.half()*, *Gripper.open()* para establecer comunicación con el robot UR3 y accionar la pinza. Finalmente, contiene la librería *controlUR3*, en la cual se encuentra la función *move()* que se encarga de desplazar el brazo robótico, el cual utiliza como parámetros las coordenadas *x,y,z* de la posición final en centímetros.

IV. PRUEBAS Y RESULTADOS

El desempeño del sistema se evaluó en tres etapas. Inicialmente se realizó una encuesta para caracterizar las experiencias previas de los usuarios con interacciones multimodales, posteriormente se realizaron cinco pruebas de tipo cuantitativo en las cuales se evaluó el desempeño del reconocimiento de cada una de las 28 expresiones verbales del diccionario; el reconocimiento de los comandos de voz; el accionamiento del gesto *"seleccionar"* en cada una de las doce zonas del espacio de trabajo; la interacción por comandos multimodales y el desempeño general del sistema para resolver el problema de clasificación de objetos cúbicos por color. En la tercera etapa se realizaron encuestas para conocer la percepción de los usuarios frente al desempeño de la arquitectura propuesta, evaluando de forma cualitativa la interacción gestual, verbal, multimodal y la realimentación audiovisual.

A. Experiencias previas con interacciones multimodales

Las pruebas fueron realizadas con un grupo de siete personas con rango de edad entre los 22 y 26 años, todos con conocimientos previos en computación y con experiencia en áreas de robótica e inteligencia artificial provenientes de su formación académica. Todos los usuarios manifestaron tener experiencia en interfaces gestuales o verbales gracias a una interacción previa con robots, sistemas a control remoto, consolas de videojuegos y asistentes virtuales. Finalmente, respecto a la experiencia con sistemas multimodales solo cuatro usuarios manifestaron tener experiencia al interactuar con un sistema mediante dos o más modalidades en simultaneo, mientras que los otros tres solo han interactuado con sistemas utilizando una sola modalidad.

B. Pruebas cuantitativas

Las pruebas cuantitativas se dividieron en: reconocimiento del diccionario de expresiones verbales, reconocimiento de comandos de voz, reconocimiento de gestos en zonas de interacción, reconocimiento de comandos multimodales y medición del desempeño de la arquitectura propuesta. El propósito de este tipo de pruebas es asignar porcentajes con el fin de evaluar cada una de las funcionalidades del sistema, considerando el 100% como el correcto cumplimiento de las funcionalidades evaluadas, y el 0% un fallo que no permite un adecuado funcionamiento del sistema. La ejecución y los resultados de estas pruebas se describen a continuación:

1) Diccionario de expresiones verbales

Cada uno de los siete usuarios realizó tres veces la pronunciación de cada una de las 28 expresiones verbales que componen el diccionario de interacción verbal para la arquitectura propuesta. Con esta prueba se busca evaluar qué tan bien el módulo de reconocimiento de voz del sistema identifica las palabras del diccionario que se le han programado, cuando diferentes usuarios las pronuncian. Los resultados son presentados en la tabla 4, donde se muestra el porcentaje de éxito del sistema en reconocer cada una de las expresiones pronunciadas por los diferentes usuarios (referidos como U1 a U7).

Tabla 4. Resultados del reconocimiento del diccionario de expresiones verbales.

Expresión	Porcentaje desempeño por usuario							Total
	U1	U2	U3	U4	U5	U6	U7	
"hola"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"interactuar"	66,7	100,0	66,7	100,0	100,0	100,0	100,0	**94,44**
"clasificar"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"zona"	100,0	100,0	66,7	100,0	100,0	100,0	100,0	**94,44**
"uno"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"dos"	100,0	100,0	66,7	100,0	66,7	100,0	100,0	**88,89**
"tres"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"cuatro"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"cinco"	66,7	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"seis"	66,7	66,7	100,0	100,0	100,0	66,7	100,0	**88,89**
"siete"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"ocho"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"nueve"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"diez"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"once"	100,0	66,7	100,0	100,0	66,7	100,0	100,0	**88,89**
"doce"	100,0	100,0	100,0	100,0	66,7	66,7	100,0	**88,89**
"cubo"	100,0	100,0	66,7	66,7	66,7	100,0	66,7	**77,78**
"toma"	66,7	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"sujeta"	66,7	100,0	66,7	100,0	100,0	100,0	100,0	**94,44**
"coge"	66,7	100,0	66,7	66,7	100,0	100,0	66,7	**83,33**
"agarra"	100,0	66,7	100,0	100,0	100,0	100,0	100,0	**94,44**
"aquí"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"ponlo"	66,7	100,0	66,7	100,0	100,0	100,0	66,7	**88,89**
"colócalo"	100,0	100,0	100,0	100,0	100,0	66,7	100,0	**94,44**
"déjalo"	100,0	100,0	66,7	66,7	100,0	66,7	100,0	**83,33**
"muévelo"	66,7	66,7	100,0	100,0	100,0	100,0	100,0	**94,44**
"azul"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
"verde"	100,0	100,0	100,0	100,0	100,0	100,0	100,0	**100,00**
Total	**90,48**	**95,24**	**90,48**	**96,43**	**95,24**	**95,24**	**96,43**	**94,84**

Se probaron un total de 588 interacciones, repartidas en 28 expresiones verbales, cada una repetida tres veces, para cada uno de los siete usuarios, lo que resultó en 84 expresiones por cada usuario. Al promediar los resultados de los siete usuarios obtenemos un porcentaje de acierto del 94.84% con una desviación estándar del ± 6.26% para el reconocimiento del

diccionario de palabras por parte del sistema, donde los usuarios cuatro y siete fueron quienes obtuvieron los mejores porcentajes de éxito en la prueba, es decir, el sistema presentó la menor cantidad de errores al momento de reconocer sus expresiones verbales, logrando en ambos casos un desempeño de (96.43% ± 10.50%); mientras que para los usuarios uno y tres fueron se presentó el caso opuesto, logrando el sistema un desempeño de (90.48% ± 15.33%). Por otro lado, aquellas expresiones verbales que al momento de ser identificadas por el sistema no presentaron un porcentaje de acierto mayor al 90% al no ser reconocidas correctamente en alguna de sus repeticiones fueron: “cubo”, “déjalo”, “coge”, “dos”, “seis”, “once”, “doce” y “ponlo”.

Durante el desarrollo de esta primera prueba cuantitativa se notó que algunos usuarios pronunciaban las palabras muy rápidamente o con un volumen de voz muy bajo, por lo que el módulo de reconocimiento de voz no reconocía algunas sílabas o simplemente captaba tanto la entrada de audio emitida por el usuario, como el ruido de fondo.

2) Comandos de voz

Cada uno de los siete usuarios realizó la pronunciación de tres comandos de voz, repitiendo tres veces cada uno de ellos. Con esta prueba se busca evaluar la efectividad del módulo de reconocimiento de voz para identificar las tareas de clasificación que diferentes usuarios asignan al sistema. Los comandos usados y sus respectivos resultados se presentan en la tabla 5.

Tabla 5. Resultados del reconocimiento de comandos de voz.

USR	Comando de Voz			Total
	"Toma el cubo azul uno y ponlo en la zona tres"	"Toma el cubo verde siete y ponlo en la zona doce"	"Toma el cubo azul cuatro y ponlo en la zona cinco"	
U1	100,00	100,00	100,00	**100,00**
U2	66,67	100,00	100,00	**88,89**
U3	100,00	100,00	100,00	**100,00**
U4	100,00	100,00	100,00	**100,00**
U5	100,00	100,00	100,00	**100,00**
U6	100,00	100,00	66,67	**88,89**
U7	100,00	100,00	100,00	**100,00**
Total	**95,24**	**100,00**	**95,24**	**96,83**

Se evaluó un total de 63 comandos de voz, con cada comando siendo evaluado 21 veces, al ser repetido tres veces por cada uno de los siete usuarios. Al promediar los resultados se observa un porcentaje de acierto del 96.83% con una desviación estándar del 5.42%, observándose que de las 63 interacciones se presentaron dos fallas, la primera en el reconocimiento por parte del sistema del primer comando emitido por el Usuario 2, y la segunda en el reconocimiento del tercer comando emitido por el Usuario 6. Esto se debe a que ambos usuarios, la primera vez que dieron dicha instrucción verbal al sistema no utilizaron las palabras exactas indicadas para la prueba, sino que usaron palabras que se encontraban por fuera del diccionario del sistema, por lo que este lanzó un mensaje de error. Al darse cuenta de la razón del error, no volvieron a utilizar palabras por fuera de las indicadas en las instrucciones verbales de la prueba y no se presentaron más errores. Es importante mencionar que el porcentaje obtenido en estas pruebas corresponde al desempeño del sistema para reconocer los comandos de voz y no a su ejecución, ya que como se explica más adelante, las redes neuronales artificiales permitieron la ejecución del 100,00% de los comandos reconocidos correctamente, gracias al entrenamiento de todos los posibles escenarios de clasificación que se pueden presentar en la zona de trabajo.

3) Interacción gestual

Se solicitó a cada usuario ejecutar el gesto “seleccionar” sobre cada una de las 12 zonas definidas en el espacio de trabajo y esto se repitió tres veces por cada zona. Con esta prueba se buscó evaluar qué tan bien el módulo de visión artificial del sistema identifica el gesto que se le ha programado, cuando diferentes usuarios lo ejecutan. Las zonas y los porcentajes de acierto del sistema en el reconocimiento de los gestos realizados por cada usuario se presentan en la tabla 6.

Tabla 6. Resultados de interacciones gestuales.

Zona	Porcentaje desempeño por usuario							Total
	U1	U2	U3	U4	U5	U6	U7	
1	66,67	100	33,33	100	33,33	66,67	66,67	**66,67**
2	66,67	66,67	100	66,67	66,67	66,67	66,67	**71,43**
3	66,67	66,67	100	66,67	66,67	66,67	66,67	**71,43**
4	66,67	33,33	33,33	66,67	100	66,67	100	**66,67**
5	66,67	66,67	66,67	100	100	33,33	100	**76,19**
6	100	100	66,67	100	100	66,67	100	**90,48**
7	100	100	100	100	66,67	100	66,67	**90,48**
8	66,67	100	66,67	66,67	66,67	100	66,67	**76,19**
9	66,67	100	66,67	66,67	66,67	100	100	**80,95**
10	100	100	100	100	66,67	100	100	**95,24**
11	66,67	100	100	100	100	100	100	**95,24**
12	100	100	66,67	100	66,67	100	100	**90,48**
Total	**77,78**	**86,11**	**75,00**	**86,11**	**75,00**	**80,56**	**86,11**	**80,95**

Se realizó la validación con siete usuarios, quienes ejecutaron el gesto “seleccionar” 36 veces cada uno, para un total de 252 interacciones gestuales, distribuidas en 12 zonas, siendo evaluadas cada una de ellas un total de 21 veces. Al promediar los resultados se obtiene un porcentaje de acierto del 80.95% con una desviación estándar del 10.93%. Durante el desarrollo de estas pruebas se logró identificar que las zonas dentro del espacio de trabajo que presentaban mayores dificultades para ser reconocidas por el módulo de visión artificial se encontraban en la fila superior del espacio de trabajo, pues las cuatro zonas que componen esta fila presentan un porcentaje de acierto igual o menor al 71,63%, que corresponde al menor porcentaje de acierto obtenido en la prueba.

Durante el desarrollo de esta tercera prueba cuantitativa se notó que algunos usuarios cometían el error de tocar el cubo que deseaban seleccionar, esto debido a que su trayectoria la realizaban mirando la zona de trabajo y no la pantalla del ordenador, como se les había indicado, donde se mostraba una realimentación visual de lo que el módulo de visión artificial podía observar. También, los usuarios en ocasiones olvidaban realizar el gesto de juntar el dedo pulgar con el índice para seleccionar el cubo, por lo que el sistema solo seleccionaba cualquiera de los cubos cercanos al dedo pulgar; y por último se descubrió que, si la cámara llegaba a detectar la mano dentro de la zona de trabajo por accidente, el cubo más cercano a esta se guardaría para la siguiente instrucción gestual que se le diera al sistema.

4) Interacción multimodal

Para este tipo de prueba se solicitó a cada usuario ejecutar tres comandos multimodales ya definidos, los cuales constan de pronunciar la frase “Mira este cubo, ponlo aquí” al mismo tiempo que se indica mediante el gesto “seleccionar” el cubo

que debe moverse y posteriormente la zona a la que se debe llevar, esto se repitió tres veces por cada comando. Los comandos multimodales utilizados y los resultados obtenidos en el reconocimiento de las interacciones efectuadas por cada usuario se presentan en la tabla 7.

Tabla 7. Resultados del reconocimiento de comandos multimodales.

USR	Comando multimodal			Total
	'Mira este cubo (gesto de seleccionar sobre cubo verde dos) ponlo aquí (gesto de seleccionar sobre la zona once)'	'Mira este cubo (gesto de seleccionar sobre cubo azul uno) ponlo aquí (gesto de seleccionar sobre la zona cuatro)'	Mira este cubo (gesto de seleccionar sobre cubo verde tres) ponlo aquí (gesto de seleccionar sobre la zona ocho)'	
U1	66,67	100,00	66,67	**77,78**
U2	100,00	100,00	66,67	**88,89**
U3	66,67	100,00	33,33	**66,67**
U4	100,00	100,00	66,67	**88,89**
U5	66,67	66,67	66,67	**66,67**
U6	100,00	100,00	66,67	**88,89**
U7	100,00	100,00	66,67	**88,89**
Total	**85,71**	**95,24**	**61,90**	**80,95**

Se evaluó un total de 63 comandos multimodales, siendo cada comando evaluado 21 veces, al ser repetido tres veces por cada uno de los siete usuarios. Se observó un desempeño total del 80.95% con una desviación estándar del 10.57%. De todas las 63 interacciones, se observa que el menor desempeño del sistema se obtuvo al reconocer las interacciones realizadas por los usuarios tres y cinco, ambos con un porcentaje de acierto del (66,67% ± 16.67%). Los resultados globales de esta prueba son muy similares a los presentados en las pruebas de la interacción gestual, lo que puede estar relacionado con el hecho de que la interacción gestual tiene mayor fuerza sobre los resultados de acierto en el reconocimiento del comando multimodal, dado que esta es la que permite definir el cubo y la zona. Además, el desempeño promedio de la interacción verbal fue superior al 95%, dejando casi toda la responsabilidad en el reconocimiento a la interacción gestual.

5) *Desempeño del sistema*

Para evaluar el desempeño del sistema se realizó una prueba con cada uno de los siete usuarios en la que se ejecuta el estado de clasificación para cada una de las posibles combinaciones que pueden tener los cubos en el espacio de trabajo. Por ejemplo: un cubo azul y un cubo verde; un cubo azul y dos cubos verdes; un cubo azul y tres cubos verdes; hasta llegar al escenario en el que estén presentes en el espacio de trabajo tres cubos azules y tres cubos verdes.

Para la etapa de ejecución de la investigación, se tomaron los resultados de esta última prueba con un total de 63 interacciones en las que cada usuario ejecutó el ejercicio una sola vez por cada una de las nueve posibles combinaciones, el sistema ejecutó correctamente la clasificación de todos los cubos verdes y azules el 100.00% de las veces. El sistema no presentó errores debido a que, una vez identificada la cantidad de cubos, con su respectiva ubicación y color, estos eran descritos mediante su *representación binaria* y esta era enviada a la *red neuronal*, la cual fue entrenada y validada con todas las posibles variaciones posibles sin presentar errores en la clasificación. Una vez obtenida la solución proveniente de la *red neuronal*, el sistema ejecutó correctamente el movimiento de todos los cubos hasta completar la tarea satisfactoriamente. Esto demuestra que el 100% de las veces que el sistema reconoce que un usuario está pidiéndole la tarea de clasificar objetos cúbicos por color, este lo realiza sin errores ya que dentro del entrenamiento de la red neuronal se han considerado todos los posibles escenarios de distribución de los cubos. En cuanto al desempeño del sistema en relación a las diferentes herramientas utilizadas para la comunicación multimodal, ya los resultados han sido expuestos en el apartado anterior, en donde mediante la interacción y práctica, se buscaba empatizar con el usuario acerca de cómo funciona el sistema, para que así los resultados revelaran el proceso de aprendizaje que los usuarios pasaban para poder operar el sistema correctamente, con la menor cantidad de errores posibles.

C. Pruebas cualitativas

Para las pruebas cualitativas se solicitó a los usuarios finalmente diligenciar una encuesta en la que se utilizó la escala *Likert de cinco niveles* [10] para evaluar la satisfacción de estos al utilizar la arquitectura propuesta para comandar al robot colaborativo UR3. De este modo, los resultados de las 7 encuestas realizadas a los sujetos de pruebas se presentan en la misma escala Likert. Las afirmaciones fueron las siguientes; 1) Según su criterio, el sistema multimodal desarrollado en la arquitectura mejora la experiencia de usuario comparado con una interacción kinestésica o tele operada, 2) Según su percepción, los comandos de voz fueron identificados correctamente por la arquitectura, 3) En su opinión, la arquitectura identificó correctamente los comandos gestuales, 4) Considera que la arquitectura identificó correctamente los comandos multimodales y 5) Desde su punto de vista, considera que la realimentación gráfica/auditiva contribuyó a tener una mejor experiencia. Los resultados obtenidos se presentan en la figura 14, en la que se puede apreciar el porcentaje de usuarios que respondió con cada uno de los niveles de la escala a cada pregunta.

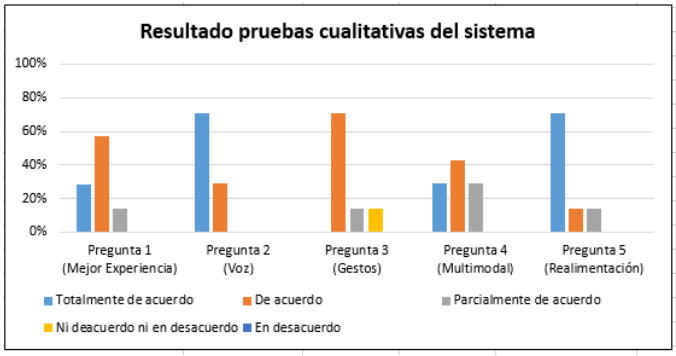


Figura 14. Encuesta del desempeño cualitativo de la arquitectura.

Para la primera pregunta el 86% de los usuarios manifestaron estar totalmente de acuerdo o de acuerdo en que el sistema desarrollado mejora la experiencia de interacción con el robot, comparado con sistemas de interacción tradicionales como la kinestésica o tele operada, los usuarios manifestaron que, aunque es algo novedoso debe mejorarse la identificación de gestos y destacaron el buen desempeño de la interacción verbal. Este punto de vista de los usuarios es válido ya que en el apartado de experiencias previas donde se preguntaba a los sujetos de prueba por los tipos de interacción con los cuales

habían tenido la oportunidad de trabajar, estos mencionaban las falencias que presentaban las modalidades de interacción kinestésicas y tele operadas, en relación a la realimentación visual, dificultad al momento de manipular al robot físicamente o dependencia de instrumentos o controles independientes para la operación. Dado que ya habían manejado este tipo de interacciones, eran conscientes al identificar las ventajas que ahora el sistema multimodal les brindaba.

En la segunda pregunta el resultado fue satisfactorio para todos los usuarios. Al preguntarles acerca de la forma en que debían de hablar al sistema, las palabras con las que contaban y el orden en que se había acomodado el formato para enseñar instrucciones al robot, se obtuvieron respuestas positivas, donde se mencionaba lo cómodo que se sentía hablar al robot y que este pudiese responderte y hacer lo que le indicas. Se menciona también que el diccionario de palabras debería de ser estudiado previamente por los usuarios para que las fallas sean menos frecuentes.

Para la tercera pregunta el 71% de los usuarios manifestaron estar de acuerdo en que los comandos gestuales fueron correctamente identificados por el sistema, pero el 29% restante no lo estuvo. Al preguntarle a los usuarios la profundidad de ese sentimiento, responde que era mucho más complicado enseñarle tareas al robot haciendo uso de comandos multimodales, en lugar de comandos verbales únicamente, por lo que la elección de modalidad para algunos usuarios era bastante clara. Es una nueva forma de comunicarse con el robot aseguraron todos los usuarios, pero debe de ser muy bien explicada, al igual que bien desarrollada por los investigadores, para que la percepción de los usuarios hacia el uso de estos comandos mejore.

Para la cuarta pregunta se obtuvieron los mismos resultados que en la pregunta anterior, los usuarios manifestaron entender que si la identificación de los gestos presentaba problemas era de esperarse que la identificación de los comandos multimodales también generase problemas, incluso mencionaron que zonas en las que más les fallaba el gesto solo, eran las mismas zonas en las que a través de la interacción multimodal también presentaban inconvenientes.

Si bien hasta este punto las preguntas realizadas a los usuarios tienen conexión con los resultados porcentuales obtenidos de las pruebas cuantitativas, el principal objetivo de las preguntas es conocer el grado de satisfacción de los usuarios, no en base a los números que arrojan las pruebas realizadas en el campo con cada usuario, sino a la experiencia de contar con múltiples maneras de comunicarse con un robot colaborativo para la asignación de tareas de clasificación de objetos.

En la quinta y última pregunta, los usuarios en su mayoría manifestaron que la realimentación gráfica era bastante útil para poder identificar las zonas y los cubos, expresan también que la realimentación auditiva, aunque útil podría nutrirse con un vocabulario más amplio. Solo un usuario manifestó que falta mayor integración entre la realimentación gráfica y auditiva dado que en la selección de cubo y zona al enseñar una tarea mediante el gesto, el usuario considera que le sería de gran ayuda que además de poder observar en pantalla el bloque y zona que se encuentra seleccionando, la arquitectura también comunique auditivamente esta información.

V. DISCUSIÓN

El sistema propuesto tiene como principales características: (i) capacidad de interpretar comandos gestuales y verbales simultáneamente; (ii) implementación de un algoritmo de aprendizaje de máquina para el aprendizaje y ejecución de tareas de clasificación; (iii) experimentación en un ambiente real utilizando el robot UR3; (iv) realimentación gráfica y auditiva para mejorar la experiencia de usuario.

En la tabla 8 se presenta una tabla comparativa con algunos desarrollos consultados de aplicaciones robóticas reales o simuladas, mediante interacciones comandadas por gestos, voz o sensores, las cuales utilizan algoritmos como *Bayesiano, SVM, CNN, KNN, SURF o Redes Neuronales Artificiales*.

Tabla 8. Tabla comparativa con trabajos relacionados

Autor	Tipo Aplicación		Tipo de interacción		Tipo de Algoritmo					Desempeño		
	Real	Simulada	Gestos	Voz	Bayes	CNN	KNN	SURF	RNA	Audio	Vision	Cumplimiento
[7]	X	X			X					N/A	N/A	96%
[9]	X		X	X		X			X	80%	72%	94%
[25]	X				X		X			N/A	N/A	95.5%
[26]		X						X		N/A	100%	100%
[27]		X					X		X	N/A	N/A	98%
Sistema Actual	X		X	X					X	96.83%	81%	94.84%

Se evidenció que el 50% de estos sistemas consultados lograron ser probados sobre aplicaciones robóticas reales, utilizando sensores como cámaras en espectro visible e infrarrojo cercano para la captura de las señales de video. Además de eso, cabe resaltar que en la mayor parte de investigaciones leídas exceptuando una, utilizan solo un método de interacción ya sea por voz, gesto o sensores, lo cual distingue a nuestro sistema de los demás, pues este utiliza un método de interacción multimodal efectivo según la percepción de los usuarios y los resultados de las pruebas realizadas. El factor de diferencia con [9], es que plantean un sistema visual y auditivo con distintas características a nuestro sistema, pues su enfoque es construir un sintetizador de estos módulos para mejorar la eficiencia en la multimodalidad aplicada. En su fase experimental probaron los distintos métodos de interacción, ya fuera solo gesto, voz o multimodal, utilizaron tres ambientes de ejecución de pruebas, en condiciones normales sin interferencias, ruido interior e interferencias de luz, siendo este otro factor de diferencia, pues en este proyecto de grado las pruebas se realizaron bajo las condiciones de ruido e iluminación controladas del laboratorio de experimentación, que vendría siendo un ambiente de tipo *indoor*. Comparando los resultados obtenidos en ambos trabajos relacionados, respecto al desempeño del 80% de su modelo de audio, es superado por el desempeño de nuestro sistema auditivo que obtuve un porcentaje de acierto de 96,83%; en su modelo visual obtienen un desempeño de 72%, el cual también superamos obteniendo un porcentaje de acierto de 80,95%, por último, en cuanto al desempeño de 94% de su modelo multimodal utilizando gestos y voz, que cabe resaltar es bastante eficiente pero no aún más que el modelo multimodal aplicado a nuestro sistema, este es sobrepasado por un 0,84%, pues se obtuvo un porcentaje de acierto del 94,84%. Además, cabe resaltar que el diccionario de expresiones se compone por distintas cantidades de expresiones verbales, estando presente en este proyecto de grado un total de 28 expresiones, siendo 22 expresiones más respecto a las presentes en [9]. En relación con la cantidad de

gestos, en [9] son utilizados cinco gestos durantes la interacción, a diferencia de este proyecto de grado que tiene un gesto. Se resalta entonces el rendimiento y eficiencia de nuestro sistema, aplicando una interacción multimodal y un sistema de audio y visión artificial, pues así lo demuestra los resultados obtenidos en la fase experimental, convirtiéndose en un gran aporte, pues presenta ventajas relevantes sobre otros modelos.

Los resultados obtenidos en el protocolo de pruebas junto con las respuestas a todas preguntas de satisfacción que se le hizo a los usuarios, permiten considerar que se ha alcanzado la meta de un sistema de clasificación de objetos cúbicos por color que mediante el aprendizaje de máquina y la comunicación multimodal pudiera permitir a un mayor número de personas la posibilidad de interactuar con robot sin la necesidad de ser expertos en programación.

VI. CONCLUSIONES

En este trabajo se desarrolló y validó un sistema que permite interactuar a una persona con un robot colaborativo UR3 mediante una comunicación multimodal por gestos y por voz para la enseñanza y ejecución de tareas de clasificación de objetos cúbicos por color, haciendo uso de algoritmos de visión artificial, reconocimiento de voz y redes neuronales artificiales.

Se evaluaron 588 interacciones verbales, 21 comandos de voz, 252 interacciones gestuales, 63 interacciones multimodales y 63 clasificaciones por color de objetos cúbicos, obteniendo un total de acierto del 94.84% para las expresiones verbales, un 96.83% para comandos de voz, un 80.95% para los comandos multimodales y 100% para las clasificaciones por color de objetos cúbicos. Esto da a entender que al menos 8 de cada 10 veces en que un usuario decidió asignar una tarea de clasificación al robot haciendo uso de cualquiera de los ítems evaluados, logró satisfactoriamente comunicarle al robot lo que en realidad deseaba, pues al asegurar solo el reconocimiento del comando utilizado, ya se cuenta con su correcta ejecución en el 100,00% de los casos.

Los sujetos de prueba concuerdan en que los comandos verbales y multimodales implementados mejoran la experiencia de interacción con el robot, debido a que además de tener experiencia con interfaces kinestésicas o tele operadas, ellos afirmaron que la multimodalidad presentada en el sistema desarrollado en este trabajo otorga más opciones, posibilidades y facilidades al momento de designar tareas. Así, al ver ahora que para hacer uso del robot los usuarios no requieren programarlo de manera convencional, el esfuerzo se minimiza y los resultados en la ejecución de tareas son satisfactorios.

Los comandos gestuales fueron destacados por los sujetos de prueba de todo el sistema, pues cinco de ellos mencionaron que era la primera vez que enseñaban tareas a un robot haciendo uso de sus dedos como selector de objetos. Sin embargo, así como fue esta funcionalidad del sistema lo que más les llamó la atención, también fue donde más errores e interrupciones encontraron a la hora de realizar las pruebas. Esto es sin duda solo un pequeño ejemplo de las grandes posibilidades que las redes neuronales y la comunicación multimodal representan en el mundo de la robótica y la inteligencia artificial. Para este trabajo solo se aplicó una tarea de clasificación, pero trabajos venideros podrían llegar a utilizar lo que aquí se ha implementado para aplicaciones como la separación de residuos, la identificación del estado de maduración de las frutas o la solución de algoritmos en la investigación de operaciones, entre otros. Las posibilidades son muchas y ahora que se ha dado la oportunidad de incluir a personas con pocos conocimientos en la programación, muchas puertas a nuevas ideas se han abierto.

Como trabajo futuro se propone desarrollar una manipulación más precisa de la pinza incluyendo su rotación, el control del porcentaje de apertura de la pinza y la velocidad de movimiento del robot. Además, integrar nuevos comandos y demostraciones al robot permitiendo el aprendizaje y ejecución de nuevas tareas de manipulación de objetos como la paletización y el ensamble de piezas. Respecto a la ejecución de nuevas tareas de clasificación, para su posible realización tendría que entrenarse nuevamente la máquina de aprendizaje de modo *offline* teniendo en cuenta los nuevos parámetros establecidos como la cantidad de cubos y sus ubicaciones en las zonas presentes en el espacio de trabajo, pues estos factores vendrían siendo la composición del vector de salida que sería el único variante, ya que el sistema conoce todas las posibles combinaciones de entrada que pueden ser presentadas por un usuario durante la interacción. Es por ello que sería posible realizar cualquier tipo de clasificación de los cubos de color utilizando nuestra arquitectura. El sistema también podría ajustarse para reconocer cubos de otros colores, incluso agregar más gestos y comandos de voz o hasta cambiar el método de aprendizaje como con arquitecturas cognitivas permitiendo así el aprendizaje *online* del sistema, todo esto de acuerdo a la necesidad del problema, no obstante el desarrollo de este proyecto de grado sirve de base para todos esos posibles trabajos futuros relacionados que puedan ser abordados, ya que cuenta con un sistema visual y auditivo eficiente y una máquina de aprendizaje con un gran potencial, pues se obtuvo un desempeño del 100%.

VII. AGRADECIMIENTOS

Este trabajo fue financiado con recursos del proyecto de investigación “Aproximación a una arquitectura cognitiva para el aprendizaje y generalización de un proceso de clasificación aplicado en un robot colaborativo” el cual pertenece a la Convocatoria interna de proyectos Por una universidad transformadora: Horizonte 2021-2025, de la Pontificia Universidad Javeriana Cali.

VIII. REFERENCIAS

[1] Á. A. Castro, “Más robots industriales operan en todo el mundo: panorama en México y Latinoamérica,” Nov. 08, 2021. https://www.mms-mexico.com/noticias/post/mas-robots-industriales-operan-en-todo-el-mundo-panorama-en-mexico-y-latinoamerica (accessed Jul. 27, 2022).

[2] B. M. Faria, L. P. Reis, N. Lau, and G. Castillo, “Machine Learning Algorithms applied to the Classification of Robotic Soccer Formations and Opponent Teams,” in *2010 IEEE International Conference on Cybernetics and Intelligent Systems*,

CIS 2010, IEEE, Jun. 2010, pp. 344–349. doi: 10.1109/ICCIS.2010.5518540.

[3] M. Haage *et al.*, "Teaching Assembly by Demonstration Using Advanced Human Robot Interaction and a Knowledge Integration Framework," *Procedia Manuf*, vol. 11, pp. 164–173, Dec. 2017, doi: 10.1016/j.promfg.2017.07.221.

[4] L. Rouhiainen, *Inteligencia artificial : 101 cosas que debes saber hoy sobre nuestro futuro*, 1st ed. Barcelona: Alienta Editorial, 2018.

[5] B. Rodrigues, R. Kansara, S. Singh, D. Save, and S. Parihar, "Ripe-unripe: Machine learning based ripeness classification," in *5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 1–5. doi: 10.1109/ICICCS51141.2021.9432349.

[6] S. Jana, S. Basak, and R. Parekh, "Automatic fruit recognition from natural images using color and texture features," in *2nd International Conference on Devices for Integrated Circuit, DevIC 2017*, Kalyani, Nadia: IEEE, Mar. 2017, pp. 620–624. doi: 10.1109/DEVIC.2017.8074025.

[7] L. Yang, Y. Dong, J. Zhuang, and J. Li, "A Recognition Algorithm for Workpieces Based on the Machine Learning," in *11th International Symposium on Computational Intelligence and Design, ISCID 2018*, Hangzhou: Institute of Electrical and Electronics Engineers Inc., Dec. 2018, pp. 371–375. doi: 10.1109/ISCID.2018.10185.

[8] A. Dutta and P. Dasgupta, "Ensemble Learning with Weak Classifiers for Fast and Reliable Unknown Terrain Classification Using Mobile Robots," *IEEE Trans Syst Man Cybern Syst*, vol. 47, no. 11, pp. 2933–2944, Nov. 2017, doi: 10.1109/TSMC.2016.2531700.

[9] H. Wang, X. Li, and X. Zhang, "Multimodal Human-robot Interaction on Service Robot," in *5th IEEE Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2021*, Chongqing, China: IEEE, Mar. 2021, pp. 2290–2295. doi: 10.1109/IAEAC50856.2021.9391068.

[10] Saul Mcleod, "Likert Scale Definition," pp. 1–4, Mar. 08, 2008. Accessed: Apr. 24, 2023. [Online]. Available: www.simplypsychology.org/likert-scale.html

[11] Caicedo Eduardo and Lopez Jesús, *Una aproximación práctica a las redes neuronales artificiales*. Programa Editorial Universidad del Valle, 2017.

[12] scikit-learn, "MLPClassifier," *Multi-layer Perceptron classifier*, 2023. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html (accessed Feb. 11, 2023).

[13] Python, "itertools - Functions," *permutations*, 2023. https://docs.python.org/3/library/itertools.html#itertools.permutations (accessed Feb. 11, 2023).

[14] joblib, "dump," 2023. https://joblib.readthedocs.io/en/latest/generated/joblib.dump.html (accessed Feb. 11, 2023).

[15] Pypi.org, "PyAudio 0.2.13," Dec. 26, 2022. https://pypi.org/project/PyAudio/ (accessed Jan. 11, 2023).

[16] Pypi.org, "Python Speech Recognition 3.9.0," 2022. https://pypi.org/project/SpeechRecognition/ (accessed Jan. 17, 2023).

[17] OpenCV Team, "Librería OpenCV," Dec. 30, 2022. https://pypi.org/project/opencv-python/ (accessed Jan. 11, 2023).

[18] Google, "MediaPipe Hands," 2023. https://google.github.io/mediapipe/solutions/hands (accessed Jan. 11, 2023).

[19] Pypi.org, "Pyttsx3 2.90," Jun. 07, 2022. https://pypi.org/project/pyttsx3/ (accessed Jan. 11, 2023).

[20] Universal Robots, "The URScript Programming Language for e- Series," 2023. https://s3-eu-west-1.amazonaws.com/ur-support-site/115824/scriptManual_SW5.11.pdf (accessed Jan. 11, 2023).

[21] S. Blandon, "Interfaz de voz humano-robot para controlar un brazo robótico UR3," Jun. 10, 2022. https://github.com/SebastianBlandon/UR3_VOICE_INTERFACE (accessed Jan. 11, 2023).

[22] J. D. Holguín, "Algoritmo de fusión de señales de audio y vídeo para el manejo de un Robot UR3," Jul. 25, 2021. https://drive.google.com/drive/folders/10noppvKpBRxzNWYBAgXmV2x_Bg3PioBy (accessed Jan. 11, 2023).

[23] scikit-learn, "train_test_split," 2023. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (accessed Feb. 11, 2023).

[24] Google LLC, "Language model selection for speech-to-text conversion," 2023. https://patents.google.com/patent/US9495127B2/en (accessed Jan. 17, 2023).

[25] Allo-Media, "text2num," Dec. 09, 2022. https://pypi.org/project/text2num/ (accessed Jan. 11, 2023).

[26] K. B. Sun and B. J. Super, "Classification of contour shapes using class segment sets," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, San Diego, CA, USA: IEEE, Jun. 2005, pp. 727–733. doi: 10.1109/CVPR.2005.98.

[27] T. Holtorf, F. J. Knoll, and S. Hussmann, "Multimodal image stitching algorithm for weed control applications in organic farming," in *2016 SAI Computing Conference, SAI 2016*, London, UK: IEEE, Jul. 2016, pp. 336–342. doi: 10.1109/SAI.2016.7556003.

[28] C.-Y. Yu and C.-C. Peng, "Machining Learning for 2D-SLAM Object Classification and Recognition," in *7th IEEE International Conference on Consumer Electronics - Taiwan, ICCE-Taiwan 2020*, Taoyuan: Institute of Electrical and Electronics Engineers Inc., Sep. 2020, p. No. doi: 10.1109/ICCE-Taiwan49838.2020.9258112.