

MOM: Machine Learning Aplicado a Seguridad Vial – Promoviendo Buenas Prácticas de Uso de Casco.

Jheyson F. Villavisan, Asesora: Alba Ávila

Enero de 2023, jf.villavisan@uniandes.edu.co, a-avila@uniandes.edu.co

Resumen – Los cascos para motociclistas están diseñados para evitar lesiones en la cabeza o la muerte en caso de colisión o caída. Sin embargo, usar el casco y usarlo abrochado es usualmente una decisión del motociclista, lo que reduce la efectividad de protección del casco. El uso abrochado del casco puede ser detectado mediante el sonido y el gesto de ponerse el casco y redes neuronales desplegadas sobre sistemas embebido. En este trabajo presentamos MOM: Un sistema que, para cascos integrales con broche micrométrico, detecta y retroalimenta al motociclista del buen uso del casco. Se colectó una base de datos con los patrones asociados a la acción de ponerse el casco y el sonido de abrocharlo, y se entrenaron redes neuronales que fueron desplegadas sobre un Arduino, para la retroalimentación al motociclista se diseñó un aplicativo móvil para celulares Android. El sistema alcanzó un *Positive Predictive Value* de 72%, lo que implica un alto rendimiento en términos de detección de condiciones seguras. Adicionalmente la detección fue probada por motociclistas reales y es escalable a aplicaciones que requieran la detección de condiciones seguras.

Palabras Clave – TinyML, Embedded System, Neural Network, Convolutional Neural Network, Micrometric Buckle.

1. INTRODUCCIÓN

Según el reporte de 2018 de la organización mundial para la salud (*World Health Organisation WHO*), aproximadamente 1.35 millones de personas mueren en accidentes de tráfico cada año, en donde el 28% de las muertes corresponden a usuarios de motocicleta [1]. El uso del casco de motocicleta reduce la probabilidad de muerte o lesiones en la cabeza en un 42% y 69% respectivamente [2]. Se estima que el casco integral (*Full-Face*) es el mejor en términos de protección en comparación con los cascos de tipo jet (*Open-Face*) y los cascos tipo abatibles o convertibles (*Half-coverage*) [3]. Sin embargo, se ha demostrado que el estado de fijación del casco resulta más crucial que el tipo de casco, dado que alrededor del 30% de los usuarios de motocicleta usan el casco abrochado incorrectamente y 10% no lo usan abrochado [4][5].

Para la detección del uso de casco se han implementado diferentes tecnologías, invasivas y no invasivas. Las tecnologías invasivas involucran colocar sensores dentro de los cascos e incluso incorporar señales de monitoreo fisiológico con diferentes sensores, ópticos [6], unidades de medición

inercial [7], resistencias de detección de fuerza [8] y sensores ultrasónicos [9]. Las tecnologías no invasivas se refieren a dispositivos colocados en el exterior del casco o incluso sin ningún contacto con este. Los algoritmos de visión artificial son quizás la técnica más utilizada. Su ventaja más significativa es la infraestructura disponible en sistemas de videovigilancia, en los que dichos algoritmos podrían implementarse o diseñarse para detectar si el pasajero o el motociclista lleva o no casco mediante el procesamiento de imágenes [10]. Sin embargo, tanto las soluciones invasivas y no invasivas mencionadas presentan la misma desventaja, no detectan el uso del casco abrochado.

En este artículo, presentamos MOM: Un sistema electrónico con interfaz en aplicativo móvil, el cual detecta más allá de la existencia del casco, detecta si esta y si esta abrochado (buen uso del casco), mediante técnicas de *Machine Learning* desplegadas sobre un sistema embebido o *Tiny Machine Learning* (TinyML). Algunos ejemplos de casos de uso de TinyML puede incluir: reconocimiento de imágenes [11], reconocimiento de palabras [12], procesamiento de lenguaje de señas [13], detección de rostros [14], reconocimiento de gestos con las manos [15], estimación de postura del cuerpo [16], monitoreo de vehículos autónomos [17], entre otros. MOM es la primera implementación para la seguridad de los motociclistas, dirigida a detectar el buen uso del casco.

MOM está basado en la tarjeta de desarrollo Arduino Portenta H7, en la que se desplegaron redes neuronales y redes neuronales convolucionales, entrenadas para detectar el buen uso del casco, por medio del gesto de ponerse el casco en la cabeza y patrón o sonido del broche micrométrico cuando es abrochado. Las señales de los gestos de ponerse el casco en la cabeza fueron adquiridas usando la tarjeta de desarrollo basada en el sensor de medida inercial (IMU) MPU6050. Los sonidos del broche corresponden a audios tomados de la base de datos del trabajo previo “*Detecting buckle fastening of helmets using CNN: An approach to promote motorcyclists’ safety*” [18]. El rendimiento de los modelos de las redes neuronales y redes neuronales convolucionales fueron evaluadas mediante la precisión (Accuracy) en los datos de validación, la precisión con datos de validación en el sistema embebido y se utilizó el Positive Predictive Value (PPV) como métrica del rendimiento en términos de condiciones seguras. El modelo seleccionado alcanza un rendimiento PPV de 72% sobre el sistema embebido. Los resultados muestran que la detección del buen uso del casco fue posible en condiciones reales mediante el uso

de tecnologías no invasivas para casco integral con broche micrométrico. Esta solución aquí planteada puede ser extendida a otros tipos de soluciones que impliquen la detección de condiciones seguras.

El artículo se encuentra organizado como se describe a continuación, sección 2 describe detalladamente los materiales y metodología utilizados, sección 3 presenta los principales resultados y análisis, sección 4 presenta las conclusiones.

2. MATERIALES Y METODOLOGÍA

Esta sección describe detalladamente los materiales, herramientas y equipos usados (2.1), el procedimiento para generar y almacenar la base de datos (2.2), la configuración, el entrenamiento y selección de las arquitecturas y modelos de redes neuronales (2.3), los criterios de selección y evaluación de modelos (2.4), el procedimiento para el despliegue y evaluación de los modelos sobre el sistema embebido (2.5), la descripción de la carcasa del prototipo (2.6) y finalmente el diseño del aplicativo móvil (2.7) y del manual de usuario (2.8).

2.1 Materiales

Para este estudio, se utilizó un casco *Full Face* (Casco que ofrece cobertura alrededor de la cabeza y cuello, que incluye un broche micrométrico) certificado por *US Department of Transit* (DOT). Se instalaron en el casco dos sensores acelerómetros y giróscopos (MPU6050, InvenSense [19]) en la parte superior y lateral izquierda del casco (posiciones típicas de accesorios como cámaras e intercomunicadores) respectivamente. El sensor micrófono capacitivo omnidireccional (MP34DT05, STMicroelectronics [20]) y el motor vibrador (TEK002, Pimoroni LTD [21]), fueron instalados en la parte lateral izquierda cerca al broche micrométrico. Como se observa en la figura 1.

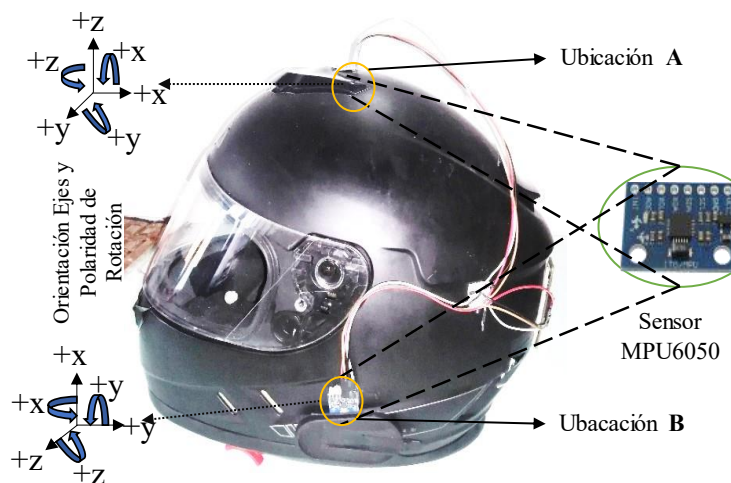


Figura 1. Casco integral con broche micrométrico. **Ubicación A**, superior: sensor acelerómetro 2 y giróscopo 2, **Ubicación B** lateral izquierda: sensor acelerómetro 1, giróscopo 1 y sensor micrófono.

Para la adquisición de datos e inferencia se usó la tarjeta de desarrollo Arduino Portenta H7 con 1 MB (MegaByte) de memoria y procesador M7 con unidad de punto flotante FPU. Los datos se transmitieron por comunicación

serial y se almacenaron sin pre-procesamiento en formato de valores separados por comas (CSV) usando el software Matlab versión R2022a [22]. El pre-procesamiento de los datos y entrenamiento de las redes neuronales se realizaron mediante la plataforma online Edgeimpulse y la librería Keras de python en la plataforma online Google Colab. El despliegue de los modelos seleccionados se realizó sobre la tarjeta de desarrollo Arduino Portenta H7 utilizando el software arduino IDE [23]. El diseño del prototipo electrónico funcional fue desarrollado utilizando el software Altium Designer versión 22.0.2 y KiCad versión 6.0. Los circuitos impresos se fabricaron en laminado epóxico reforzado con fibra de vidrio (FR4) y se ensamblaron a mano. El diseño de prototipo mecánico se desarrolló usando la versión online del software Autodesk “TinkerCad”. Las versiones del empaque se fabricaron con la impresora 3D MakerGear 3D en filamento PLA +eSUN, disponibles en la Universidad de los Andes.

El aplicativo móvil se desarrolló con el framework React Native de Java Script y fue codificado sobre el software Visual Studio Code [24], se diseñó para ser compatible con dispositivos móviles con sistema operativo Android Versión 9 o superior y fue probado sobre un teléfono, con CPU de 2GHz y memoria RAM de 64GB.

2.2 Base de datos

A continuación, se describe la base de datos usada para la detección del broche micrométrico mediante sonido y el procedimiento para la creación de la base de datos usada para la detección del buen uso del casco mediante los sensores acelerómetro y giróscopo.

2.2.1 Base de datos sonido broche micrométrico.

Se utilizó la base de datos creada en el trabajo previo “*Detecting buckle fastening of helmets using CNN: An approach to promote motorcyclists’ safety*” [18], que contiene +9000 sonidos de la acción de abrochar el casco en dos distintos ambientes, el primer ambiente corresponde a audios grabados en un estudio de música profesional y contiene audios grabados bajo dos condiciones: la primera, audios de la acción de abrochar el casco sin ruido, la segunda, la acción de abrochar el casco mezclada con ruido simulado de tráfico vehicular. El segundo ambiente contiene audios grabados en condiciones de ruido real, en el que se mezcla el sonido correspondiente a la acción de abrochar con ruido de motor de motocicleta, ruido de bocina de vehículos, ruido de motor de vehículos etc. Estos audios fueron grabados con tres micrófonos; un micrófono profesional referencia “Shure SM81” con ancho de banda de 20Hz a 20kHz, un micrófono de manos libres genérico 3.5mm, con ancho de banda de 20Hz a 16kHz y un micrófono genérico de intercomunicador Bluetooth con ancho de banda 20Hz a 8kHz referencia EJEAS-V6. Los audios fueron grabados a una frecuencia de 16kHz, ubicados a una distancia máxima de 10 cm del broche micrométrico y almacenados en formato WAV (*Waveform audio file format*).

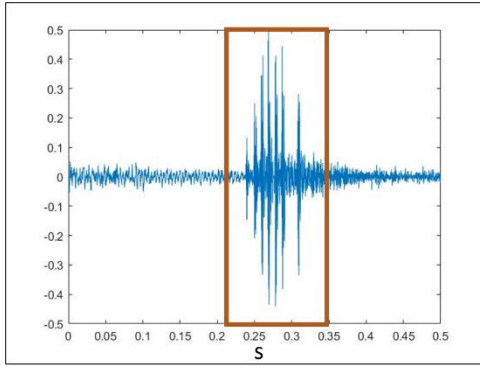


Figura 2 resultado en rojo señal de audio sin ruido, producida por acción de abrochar el cierre micrométrico. Grabado con sensor micrófono de manos libre genérico 3.5mm.

En la figura 2 se observa la muestra corresponde la señal de audio correspondiente a la acción de abrochar el cierre micrométrico. Se observan 8 picos correspondientes al sonido producido por cada uno de los 8 dientes del broche micrométrico. La acción de abrochar el cierre micrométrico dura en promedio 250ms y tiene una máxima duración de 500ms (depende del usuario).

Los audios fueron etiquetados como clases “0” y “1”, en donde clase “0” corresponde al sonido producido por la acción de abrochar el casco y clase “1” a otros sonidos. Todos los sonidos fueron preprocesados mediante 3 diferentes técnicas: Coeficientes Cepstrales en las frecuencias de Mel (MFCC), características de energía del banco de filtros de Mel (MFE Bank) y Espectrograma con el objetivo de reducir el tamaño de las señales.

2.2.2 Base de datos sensor acelerómetro y giróscopo.

Tres usuarios fueron entrenados para seguir los siguientes protocolos: 1) ponerse el casco en la cabeza y abrochar la hebilla micrométrica firmemente, 2) ponerse el casco sobre la cabeza sin abrochar la hebilla micrométrica, 3) ponerse el casco de manera incorrecta (en la cabeza sin ajustarlo) o no ponérselo. Los usuarios ejecutaron estos protocolos aleatoriamente y se colectó una base de datos balanceada con 240 diferentes señales para el sensor acelerómetro y giróscopo. Las señales correspondientes a esas acciones fueron capturadas mediante el sensor integrado MPU6050 durante 8 segundos a una frecuencia de muestreo de 125 Hz. Cada señal contiene las componentes de aceleración y ángulo de giro de los ejes “x”, “y” y “z”. El sensor transmitió los datos a la tarjeta de desarrollo Arduino Portenta H7 mediante el protocolo de comunicación *Inter-Integrated Circuit* (I2C) y posteriormente, mediante comunicación serial se transmitieron a un algoritmo desarrollado en Matlab y se almacenaron en formato CSV, clasificados con las etiquetas: clase “0” para buen uso del casco, clase “1” para “casco puesto sin abrochar” y clase “2” para “desconocido”.

La base de datos del sensor acelerómetro y giróscopo fue aumentada mediante la técnica de ventanas móviles, realizando recortes de la señal original de un tamaño de 6 segundos (750 muestras) con corrimientos de 250 ms como se observa en la figura 2 y mediante redimensionamiento se obtuvieron vectores 1-dimensionales de tamaño 1x2250. Con lo anterior se obtuvo una base de datos de 1920 señales.

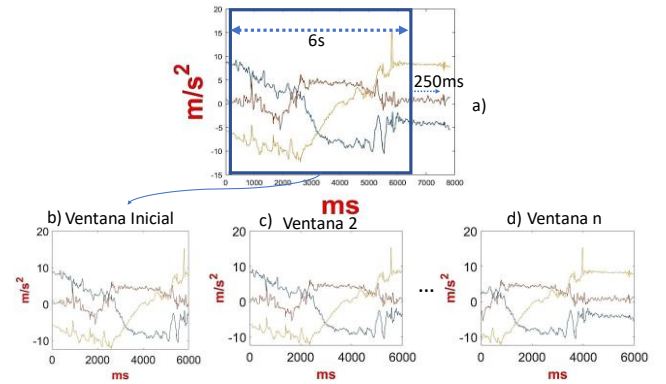


Figura 3 a) Señal acelerómetro producida por el gesto de ponerse el casco y abrocharlo durante 8 segundos. Técnica ventanas móviles, tamaño de la ventana de 6 segundos y desplazamiento cada 250ms. b) muestra generada sin desplazamiento, tamaño 6 segundos, c) muestra número 2 generada con desplazamiento de 250ms y tamaño de 6 segundos d) muestra desplazada 250*n ms.

2.3 Arquitectura, Entrenamiento y Selección de Modelo.

A continuación, se describen detalladamente las arquitecturas de las redes neuronales implementadas para la detección del sonido generado por la hebilla micrométrica cuando se realiza la acción de abrochar, seguido por la descripción de la arquitectura implementada para la detección del uso del casco mediante los sensores acelerómetro y giróscopo.

2.3.1 Sonido

La tabla 1 muestra la arquitectura de red neuronal convolucional implementada, la arquitectura seleccionada consiste en una capa de entrada, una capa de redimensionamiento, seguido por dos capas convolucionales de 1 Dimensión de tamaño 8 y 16 con función de activación ReLu (Por sus siglas en ingles de *Rectified Linear Unit*), una etapa de aplanamiento (*flattening*), una capa completamente conectada con *Dropout* de 50%, con N neuronas y función de activación ReLu y finalmente una capa de salida con 2 neuronas, correspondiente a cada clase, con función de activación SoftMax.

Tabla 1. Arquitectura de red neuronal convolucional configurada como sigue: capa de entrada de tamaño 1x2240, capa *reshape*, dos capas convolucionales de 1D con 8 y 16 neuronas respectivamente y función de activación ReLu, una etapa de *flattening*, seguido por una capa completamente conectada y con *Dropout* de % y dos neuronas en la capa de salida con función de activación SoftMax.

Layer	Number of Neurons	Activation Function Used
Input	2240	ReLu
Reshape (40 columns)	-	-
1D conv	8	ReLu
1D conv	16	ReLu
Flatten	-	-
Hidden	N	ReLu
Dropout (0.5)	-	-
Output	2	SoftMax

Los modelos se entrenaron con el 80% de la base de datos y se reservó el 20% para evaluación, cada modelo fue entrenado durante 50 épocas usando el algoritmo de propagación hacia atrás (*Backpropagation*), con descenso de gradiente estocástico

(SGD) como función de optimización a una tasa de aprendizaje (LR) de 0.005. Los parámetros de exploración fueron la cantidad de neuronas N en la capa 3 y la técnica de preprocesamiento: MFCC, MFE y Espectrograma.

2.3.2 Acelerómetro y giróscopo

La tabla 2 muestra la arquitectura de red neuronal convolucional implementada, la arquitectura consiste en una capa de entrada, dos capas escondidas y una capa de salida con 3 neuronas. Como entrada de la red neuronal, las señales sin preprocesamiento de los sensores acelerómetro o giróscopo respectivamente, etiquetadas con las clases “0” para la clase correspondiente a “buen uso”, “1” para la clase correspondiente a “casco puesto sin abrochar” y “2” para la clase “mal puesto”. Seguido a la capa de entrada se configuraron 2 capas escondidas completamente conectadas con N1 y N2 (parámetros de exploración) neuronas en cada capa, con función de activación ReLu y finalmente se configuró la capa de salida con 3 neuronas correspondiente a cada clase y función de activación SoftMax. La arquitectura fue entrenada con el 80% de la base de datos y se reservó el 20% para evaluación de los modelos, cada modelo fue entrenado durante 50 épocas usando *Backpropagation* y SGD a una tasa LR de 0.0005.

Tabla 2 Arquitectura de red neuronal configurada como sigue: capa de entrada de tamaño 1x2250 con Dropout de 50%, dos capas completamente conectadas con N1 y N2 cantidad de neuronas respectivamente y función de activación ReLu, tres neuronas en la capa de salida con función de activación SoftMax.

Layer	Number of Neurons	Activation Function Used
Input (DropOut 0.5)	2250	ReLu
Hidden	N1	ReLu
Hidden	N2	ReLu
Output	3	SoftMax

2.4 Selección y evaluación de modelos entrenados.

Las métricas usadas para evaluar y seleccionar los modelos en términos de rendimiento fueron: *Accuracy* (ACC) One vs One para los modelos entrenados con las señales del sensor de audio y *ACC One vs All* para los modelos entrenados con los sensores acelerómetro y giróscopo, la ecuación (1) define la métrica ACC.

$$ACC = \frac{TP+TN}{P+N} \quad (1)$$

Donde P es el número de casos positivos reales, N es el número de casos negativos reales, TP es el número de casos positivos reales clasificados como positivos, y TN es el número de casos negativos reales clasificados como negativos.

Los modelos se evaluaron con el conjunto de datos de validación y se seleccionaron como elegibles para el despliegue los modelos que alcanzaron un ACC igual o superior a 85%. Adicionalmente de los 1MB de memoria disponible por la tarjeta de desarrollo Arduino Portenta H7 se reservaron 500kB para el algoritmo principal, para almacenamiento de variables locales, configuración de puertos, comunicaciones etc. Por lo que adicionalmente se tuvo como parámetro de selección el tamaño del modelo en kB (kilobytes), en donde los modelos con

tamaño igual o inferior a 500kB se consideraron como elegibles.

2.5 Despliegue, evaluación y selección de modelo.

Los modelos seleccionados fueron desplegados en la tarjeta de desarrollo Arduino Portenta H7, usando el software Arduino IDE y la librería de Tensorflow lite para microcontroladores de Python. Todos los modelos elegibles se desplegaron cuantizados (los valores de los pesos de las neuronas se optimizaron y convirtieron en formato entero de 8 bits) y sin cuantizar (los valores de las neuronas en formato flotante de 32 bits). El rendimiento de los modelos se evaluó con un subconjunto de datos de validación usando el mismo criterio que el literal 2.4. La evaluación se realizó almacenando el subconjunto de datos en un buffer del tamaño de la entrada de la red neuronal y se realizó la inferencia de cada señal. El resultado de la clasificación se transmitió por comunicación serial hacia un algoritmo desarrollado en Matlab que calculó el ACC. El modelo con mayor ACC fue seleccionado como modelo final. La métrica usada para evaluar el modelo seleccionado en términos de seguridad vial fue el *Positive Predictive Value* (PPV), la ecuación (2) define la métrica PPV.

$$PPV = \frac{TP}{FP+TP} \quad (2)$$

Donde TP es el número de aciertos de clasificación de buen uso del casco y FP es el número de errores de clasificación del buen uso del casco.

2.6 Diseño de prototipo

MOM tiene 2 funciones, detectar el buen uso del casco y retroalimentar al usuario del estado del casco. El prototipo desarrollado es el encargado de sensar, clasificar, retroalimentar y transmitir el estado del casco mediante comunicación Bluetooth Low Energy (BLE), la transmisión se realiza continuamente hacia el aplicativo móvil, y allí se realiza la retroalimentación al usuario mediante retroalimentación háptica, auditiva y visual.

2.6.1 Prototipo electrónico

El prototipo electrónico se diseñó empleando el software Altium Designer 22.0.2 y Kicad V6.0. El prototipo fue diseñado para funcionar con un voltaje de trabajo de 3.3V, comunicarse mediante bus de datos I2C y modulación de señales por densidad de pulsos PDM, con los sensores acelerómetro y de audio respectivamente. Los actuadores encargados de la retroalimentación se diseñaron para funcionar mediante pines de propósito general GPIO y modulación de anchos de pulso PWM. El diseño se dividió en dos diferentes circuitos impresos: 1) El circuito impreso principal, compatible con la tarjeta de desarrollo Arduino Portenta H7, fue diseñado para adquisición de datos, preprocesamiento, inferencia, clasificación, retroalimentación háptica y comunicación BLE, 2) el circuito impreso de retroalimentación lumínica, diseñado para que mediante 3 diferentes colores y luces led retroalimente al usuario del estado del casco; iluminación verde para casco bien puesto, iluminación amarilla para casco sin abrochar e iluminación roja para casco mal puesto o casco no puesto.

La figura 4, muestra el diagrama de bloques correspondiente al prototipo funcional diseñado.

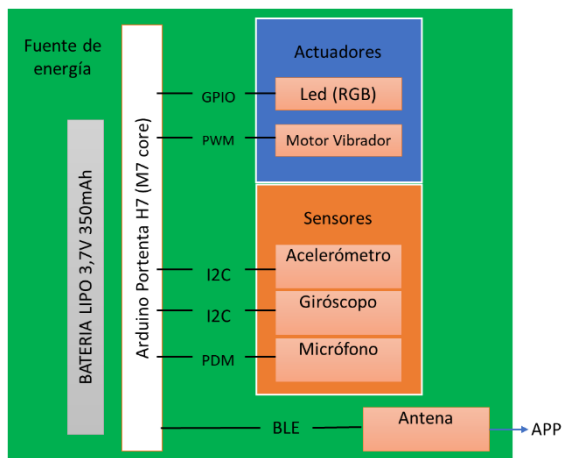


Figura 4 Diagrama de bloques diseño electrónico MOM. Bloque Azul corresponde a actuadores. Bloque Naranja corresponde a los sensores utilizados. El suministro de energía para todos los bloques se realiza con una batería LiPo de 3.7 V – 350mAh desde la tarjeta Arduino Portenta H7.

2.6.2 Prototipo Mecánico

El prototipo de la carcasa se diseñó empleando el software FreeCad y AutoDesk en su versión online TinkerCad. La figura 5 muestra el diseño compacto del prototipo de la carcasa, diseñado en una sola pieza para ubicarse en el exterior del casco al costado lateral izquierdo del casco, **ubicación b** (figura 1), y para ajustarse a la curvatura convencional de los cascos de motocicleta tipo integrados y abatibles comerciales. En la carcasa se embebieron los circuitos impresos, el motor vibrador y la batería, las dimensiones son: 71 mm de largo, 69 mm de ancho y 19.5mm de alto y peso <150g.

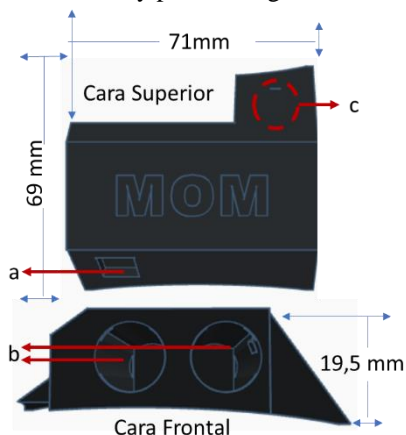


Figura 5 Diseño prototipo mecánico de MOM. a) ubicación del sensor de audioMP34DT05, b) ubicación circuito impreso de retroalimentación visual, c) ubicación del motor vibrador.

2.7 Prototipo Aplicativo móvil

El prototipo de aplicativo móvil fue desarrollado empleando el framework de JavaScript React-Native. El aplicativo móvil está diseñado para ser la interfaz entre el usuario y el dispositivo. Adicionalmente para concientizar al usuario del buen uso del casco mediante retroalimentación. Dependiendo de la clasificación reportada por el sistema embebido, el aplicativo móvil realiza 3 diferentes tipos de retroalimentación:

retroalimentación visual, háptica y sonora. Y estas a su vez presentan tres tipos de combinaciones, para la clasificación correspondiente a la clase 0, se realiza una retroalimentación sonora y visual con la cual se le confirma al usuario el buen uso del casco mediante el mensaje “Casco listo”, para el estado correspondiente a la clase 1, se realiza una retroalimentación sonora y visual con el mensaje “Casco sin abrochar” y una retroalimentación háptica suave cada segundo hasta que se detecte el cierre de la hebilla micrométrica, para el estado del casco correspondiente a la clase 2, se realiza una retroalimentación sonora y visual con el mensaje “Sin Casco” y retroalimentación háptica fuerte hasta que se detecte el buen uso del casco. La figura 6 muestra los mockups del aplicativo móvil con la retroalimentación visual de las clases: “0”, “1” y “2”.

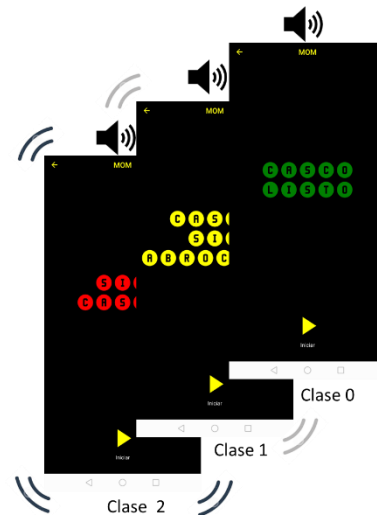


Figura 6 Diseño prototipo aplicativo móvil de MOM. Clase 0) Estado correspondiente a buen uso del casco, no hay retroalimentación háptica, la retroalimentación sonora es solo una vez y el color de la fuente se presenta en verde, Clase 1) Estado correspondiente a casco sin abrochar, la retroalimentación háptica es leve, la retroalimentación sonora es cada 1 segundo y el color de la fuente se presenta en amarillo, Clase 2) Estado correspondiente a casco mal puesto, la retroalimentación háptica es fuerte, la retroalimentación sonora es cada 1 segundo y el color de la fuente se presenta en rojo.

2.8 Diseño Manual de usuario

El manual de usuario fue diseñado en Microsoft Word y el aplicativo online Canva. El manual de usuario incluye información e instrucciones relevante para la instalación, configuración, emparejamiento con el prototipo electrónico, el uso, el mantenimiento y las características eléctricas. Ver anexo A.

3. RESULTADOS Y ANÁLISIS

La figura 7 muestra el rendimiento de los modelos. El rendimiento de los modelos con datos de validación en términos de ACC se encuentra entre 86 y 91%, 29 y 40% y 95 y 98%, para los sensores acelerómetro, giróscopo y sonido respectivamente, con excepción del modelo con N1=10 y N2=5 neuronas, del sensor acelerómetro que alcanza un 66.5%. Los modelos desplegados alcanzaron un rendimiento en términos de ACC entre 70 y 77% y 85 y 98% para los sensores acelerómetro y de sonido respectivamente. Ninguno de los modelos correspondientes al sensor giróscopo alcanzó el criterio de ACC sobre los datos de validación, por lo que no fueron desplegados, de igual manera, los modelos que superaron el tamaño de

500kB no fueron desplegados, estos se observan sobre el eje x con un ACC de 0%. Adicionalmente se puede observar que, aunque los modelos desplegados que corresponden al sensor de audio muestran mejor desempeño en términos de $ACC > 95\%$, este sensor no permite identificar el buen uso del casco por sí solo (solo identifica el sonido del broche).

un PPV de 72%, lo que, en términos de seguridad vial, implica que el sistema identifica más 7 de cada 10 condiciones seguras correctamente, y las 3 condiciones seguras etiquetadas como inseguras solo implican la repetición de la acción del buen uso del casco, lo que no resulta en un riesgo en términos de seguridad vial.

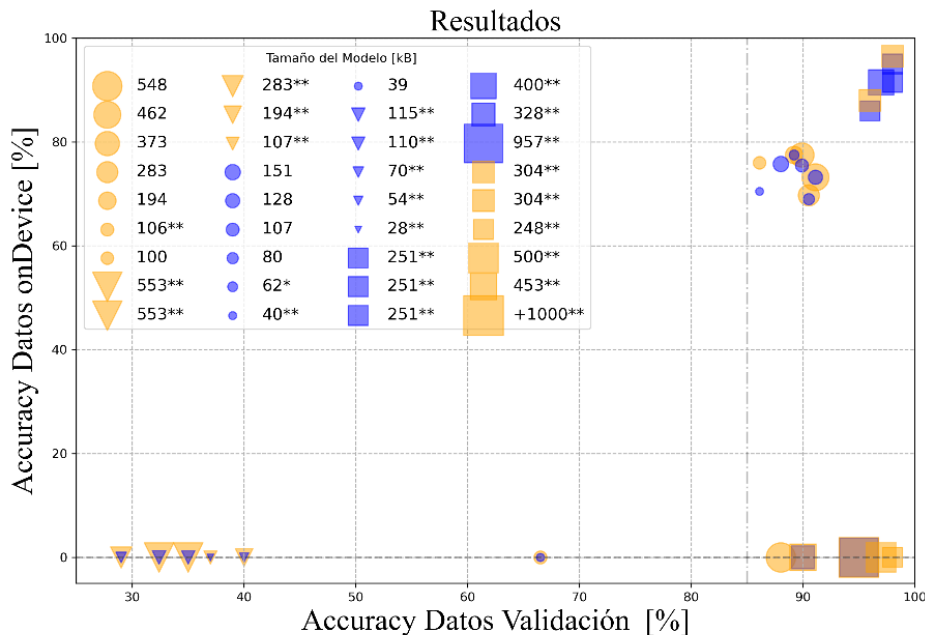


Figura 7. Rendimiento de los modelos. El eje x se representa el ACC de los modelos sobre los datos de validación, el eje y representa el ACC de los modelos desplegados sobre el sistema embebido. Los resultados del sensor acelerómetro están representados con círculos, los correspondientes al sensor giróscopo con triángulos y los correspondientes al sensor de audio con cuadrados. Los modelos sin cuantizar están representados con el color naranja y los modelos cuantizados con el color azul.

Por otra parte, los modelos desplegados que corresponden al sensor acelerómetro no supera el 77.75% de ACC, lo que conlleva al uso complementario de los dos sensores para la identificación del buen uso del casco. Una identificación del buen uso del casco con doble validación, en donde el sensor acelerómetro identifica el patrón asociado al gesto de ponerse el casco y abrocharlo, confirmado por el sonido de la acción de abrochar la hebilla micrométrica.

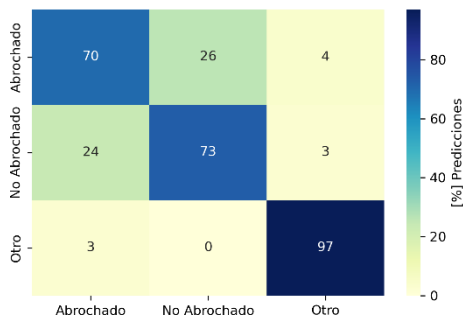


Figura 8. Matriz de confusión / matriz de error modelo seleccionado, las columnas de la matriz representan el porcentaje de aciertos en las predicciones del modelo, las filas de la matriz representan la clase real.

La figura 8 muestra la matriz de confusión del modelo seleccionado, en la cual se puede observar que modelo alcanza

4. CONCLUSIONES

Se diseña un sistema hasta ahora reportado en el estado del arte, capaz de identificar, de manera no invasiva, el buen uso del casco (casco puesto y abrochado) para los cascos tipo integral con broche micrométrico, desplegando redes neuronales sobre un sistema embebido. Se generó una base de datos con más de 1500 señales correspondientes a los patrones asociados al buen uso del casco de usuarios de motocicleta reales. Se define el *Positive Predictive Value* (PPV) como métrica para evaluar modelos de *machine learning* en términos de seguridad vial para el buen uso del casco de motociclistas. El sistema alcanza un ACC mayor al 89% con los datos de validación y un ACC de más de 77% validado sobre el sistema embebido. El modelo seleccionado cuenta con un tamaño de 373kB y un PPV de >72 % de PPV. Lo que en comparación con las soluciones existentes resulta más confiable, dado que detecta el uso del casco y el uso abrochado del casco. Finalmente, esta solución puede ser extendida a sistemas personalizados, tales como detección del uso del casco por reconocimiento de voz, a otro tipo de casco y desplegado sobre un sistema microcontrolado de menores recursos.

5. AGRADECIMIENTOS

Los autores agradecen al laboratorio de música y al laboratorio de ingeniería eléctrica y electrónica de la Universidad de los Andes por permitir el acceso a sus laboratorios para la recolección de datos, ejecución de experimentos y la fabricación de los prototipos. Los autores también brindan un especial agradecimiento al semillero de investigación nanoelectrónica por la constante retroalimentación en la preparación de la presentación durante la ejecución de este proyecto.

6. REFERENCIAS

- [1] Organisation mondiale de la santé: Global Status Report on Road Safety 2018, WHO (2018).
- [2] Liu, B. C., Ivers, R., Norton, R., Boufous, S., Blows, S., & Lo, S. K. (2008). Helmets for preventing injury in motorcycle riders. *Cochrane database of systematic reviews*, (1).
- [3] Tabary, M., Ahmadi, S., Amirzade-Iranan, M. H., Shojaei, M., Asl, M. S., Ghodsi, Z., ... & Rahimi-Movaghar, V. (2021). The effectiveness of different types of motorcycle helmets—A scoping review. *Accident Analysis & Prevention*, 154, 106065.
- [4] Ramli, R., & Oxley, J. (2016). Motorcycle helmet fixation status is more crucial than helmet type in providing protection to the head. *Injury*, 47(11), 2442-2449.
- [5] Freitas, C. A., Furtado, A. M., Petterle, R. R., Pimentel, S. K., & Carvalho, F. H. (2018). Helmet retention system types as a risk factor to properly attach the chin strap. *Traffic injury prevention*, 19(2), 173-178.
- [6] Lokeshwaran, M., Mathew, S. N., & Joshuva, A. (2020, August). Raphael—The Smart helmet. In 2020 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET) (pp. 48-51). IEEE
- [7] Li, P., Meziane, R., Otis, M. J. D., Ezzaidi, H., & Cardou, P. (2014, October). A Smart Safety Helmet using IMU and EEG sensors for worker fatigue detection. In 2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings (pp. 55-60). IEEE
- [8] Shravya, K., Mandapati, Y., Keerthi, D., Harika, K., & Senapati, R. K. (2019). Smart helmet for safe driving. In *E3S Web of Conferences* (Vol. 87, p. 01023). EDP Sciences.
- [9] elasco, A., Barragan, M., Ram, D., & McCoy, M. Safety Helmet.
- [10] Chiverton, J. (2012). Helmet presence classification with motorcycle detection and tracking. *IET Intelligent Transport Systems*, 6(3), 259-269.
- [11] Wong, A., Famouri, M., & Shafiee, M. J. (2020). Attendnets: tiny deep image recognition neural networks for the edge via visual attention condensers. *arXiv preprint arXiv:2009.14385*.
- [12] Wong, A., Famouri, M., Pavlova, M., & Surana, S. (2020). Tinspeech: Attention condensers for deep speech recognition neural networks on edge devices. *arXiv preprint arXiv:2008.04245*.
- [13] Paul, A. J., Mohan, P., & Sehgal, S. (2020, December). Rethinking generalization in american sign language prediction for edge devices with extremely low memory footprint. In 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS) (pp. 147-152). IEEE.
- [14] Giordano, M., & Magno, M. (2021, November). A Battery-Free Long-Range Wireless Smart Camera for Face Recognition. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems* (pp. 594-595).
- [15] Zhou, A., Muller, R., & Rabaey, J. (2021). Memory-Efficient, Limb Position-Aware Hand Gesture Recognition using Hyperdimensional Computing. *arXiv preprint arXiv:2103.05267*.
- [16] Vuletic, M., Mujagic, V., Milojevic, N., & Biswas, D. Edge AI Framework for Healthcare Applications.
- [17] de Prado, M., Rusci, M., Capotondi, A., Donze, R., Benini, L., & Pazos, N. (2021). Robustifying the Deployment of tinyML Models for Autonomous mini-vehicles. *Sensors*, 21(4), 1339.
- [18] Jheyson F. Villavisan, Juan P. Oviedo, Javier L. Castellanos, Zulay Villabona, Mario Varón, Alba Ávila. Detecting buckle fastening of helmets using CNN: An approach to promote motorcyclists' safety. *IEEE Transactions on Intelligent Transportation Systems*. (Under Review).
- [19] INVENSENSE. (2013, August 19). MPU-6000 and MPU-6050 Product Specification Revision 3.4. <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- [20] STMicroelectronics. (2021, June). MEMS audio sensor omnidirectional digital microphone. STMicroelectronics.
- [21] PIMORONI. (2017, January 12). Sewable Motionboard (Vibrating Motor) /TEK002. https://media.digikey.com/pdf/Data%20Sheets/Pimoroni%20PDFs/TEK002_Web.pdf
- [22] Matlab. MathWorks. (n.d.). Retrieved January 12, 2023, from <https://www.mathworks.com/products/matlab.html>
- [23] Software (n.d.). Arduino. <https://www.arduino.cc/en/software/>
- <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- [24] Microsoft. (2021, November 3). Visual studio code - code editing. redefined. RSS. Retrieved January 12, 2023, from <https://code.visualstudio.com/>

7. AUTORES

Jheyson Fabian Villavisan Buitrago, Ingeniero en Control y Especialista en Gestión de Proyectos de Ingeniería de la Universidad Distrital Francisco José de Caldas, interesado en la ingeniería aplicada al bienestar social e Inteligencia Artificial. Correo alternativo de contacto jfvillavisanb@correo.udistrital.edu.co

Alba Graciela Ávila Bernal, Profesora Asociada de la Universidad de los Andes, Ingeniera Electrónica y Magíster en Ingeniería Eléctrica de la Universidad de los Andes, PhD de la Universidad de Cambridge, interesada en investigaciones asociadas a nanotecnología, caracterización de nanomateriales y materiales avanzados.