Monitoring Subconcussive Impacts Through a Helmet-Embedded Device

Jahaan Patel

Part 2 - Engineering Problem and Goal

- **Repetitive head impacts (RHI)** can lead to "functional and structural changes in the brain" (Alosco & McKee, 2018). There is little technology available to access live impact records in military and athletes.
- **Military soldiers** experience multiple blast exposures, resulting in the loss of psychological function (Lippa, 2024). Impact tracking can provide military personnel pre-diagnosis on head trauma injuries. "[The] lack of reliable and valid measures for assessment of blast exposure [is a] major limitation restricting this research." (Lippa, 2024)
- Similar issue is shown in **high school contact-sport athletes** since "[a] single practice session involving head contact…can result in impairment" (D'Arcy, 2024).
- **<u>Engineering Goal</u>**: Develop a prototype that is able to provide live data on when the subject experiences an impact to the head and the severity of each occurring impact.

Part 3 - Procedures

Materials

- 6x Square Force-Sensitive Resistor (FSR) (Alpha MF02A-N-221-A01) → detects significant pressures from impacts but does not measure magnitude of impact
- 6x 10K Ohm Resistors → regulates FSR sensitivity
- 2x Adafruit LIS3DH Triple-Axis Accelerometer → measures acceleration of the head
- Adafruit ESP32 Feather V2 → manages data handling BLE communication
- Lithium Ion Polymer Battery with Short Cable - 3.7V 350mAh → powers circuit
- 6x 2-Pin Terminal Blocks → connects to FSRs via copper wires
- Printed Circuit Board (PCB) → circuit board with electronic components mounted on it
- Insulated Copper Wires (Male/Female Headers) → connects FSRs and terminal blocks
- Heat Shrink Tubing → secures FSR and copper wire connection
- Cardboard → providing backing support to FSRs
- 2x Stemma QT Cables → connects accelerometers to ESP32
- EVA Porous Closed-Cell Foam → protects PCB and accelerometers from impacts
- Breadboard → used for prototyping
- 3D Printed PLA Plastic → 3D printed cases for ESP32 and accelerometers
- Velcro Pads → attach device to the inside of the helmet

Four Phases of Procedure

- Circuit Design
- PCB Programming
- Protective Case Design
- Testing

Part 3 - Procedures - Circuit Design

- Tested circuit designs through **breadboard** prototyping with various combinations of resistor and capacitor values. To minimize PCB size, I only used **six 10K Ω resistors** to regulate FSR sensitivity.
- Wired analog-to-digital converter (ADC) pins to FSRs via **2-pin terminal blocks** to read output voltage
- Utilized Fusion 360, a CAD software, to create schematics of the finalized PCB design.
- Soldered ESP32 microcontroller, terminal blocks, and resistors to PCB
- Connected accelerometers to ESP32 via **JST cables**
- Powered by **LiPoly Battery**

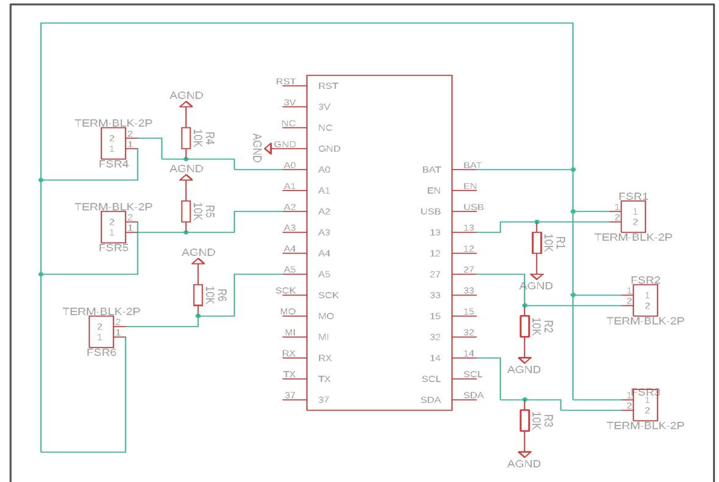


Fig 1: Electrical Schematic of Circuit Design

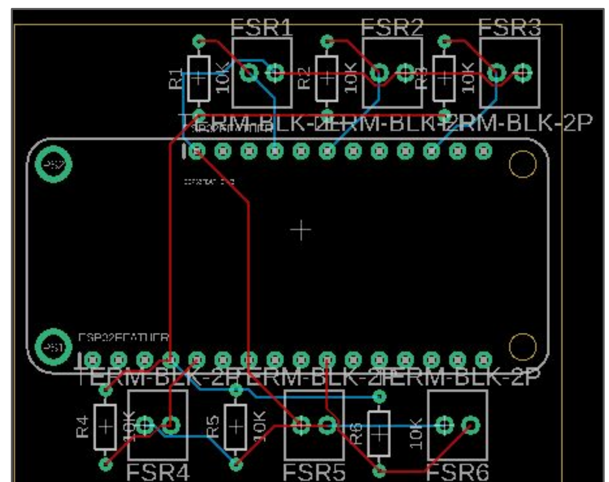


Fig 2: PCB Layout

Part 3 - Procedures - PCB Programming

- Programmed with Circuit Python and Thonny IDE for the ESP32 to transmit FSR and accelerometer data via Bluetooth Low Energy (BLE).
- Presented data in a Python-based UI software
    - Graphical analysis of acceleration and FSR spikes correlates FSR spikes with acceleration spikes to determine if a significant head impact occurred.
    - Displayed force values and time of impact in a scrollbar.
    - Implemented Newton's 2nd Law of Motion and used the product of user-inputted head mass and the change in acceleration.
    - Head mass calculations:
        - ~ linear relationship with head circumference (Ching et al., 2002)
        - ~ body to head density ratio

```python

    # Detect spikes in accelerometer data
    accel_spikeL1 = detect_spike(accelerationL1, prev_accelL1, ACCELERATION_THRESHOLD)
    accel_spikeR1 = detect_spike(accelerationR1, prev_accelR1, ACCELERATION_THRESHOLD)

    # Detect spikes in FSR data
    fsr_spikeL1 = detect_spike(fsrForceL1, prev_fsrL1, FSR_THRESHOLD)
    fsr_spikeL2 = detect_spike(fsrForceL2, prev_fsrL2, FSR_THRESHOLD)
    fsr_spikeL3 = detect_spike(fsrForceL3, prev_fsrL3, FSR_THRESHOLD)
    fsr_spikeR1 = detect_spike(fsrForceR1, prev_fsrR1, FSR_THRESHOLD)
    fsr_spikeR2 = detect_spike(fsrForceR2, prev_fsrR2, FSR_THRESHOLD)
    fsr_spikeR3 = detect_spike(fsrForceR3, prev_fsrR3, FSR_THRESHOLD)

    # Update previous values
    prev_accelL1, prev_accelR1 = accelerationL1, accelerationR1
    prev_fsrL1, prev_fsrL2, prev_fsrL3 = fsrForceL1, fsrForceL2, fsrForceL3
    prev_fsrR1, prev_fsrR2, prev_fsrR3 = fsrForceR1, fsrForceR2, fsrForceR3

    return accel_spikeL1, accel_spikeR1, fsr_spikeL1, fsr_spikeL2, fsr_spikeL3, fsr_spikeR1, fsr_spikeR2, fsr_spikeR3

def approximate_force_at_spike(acceleration, mass):
    #print(f"Debug: acceleration={acceleration}, mass={mass}")
    if not isinstance(acceleration, (int, float)) or not isinstance(mass, (int, float)):
        print(f"Error: Invalid types - acceleration: {type(acceleration)}, mass: {type(mass)}")
        return None

    force_estimation = acceleration * mass
    return force_estimation

def get_force_values():
    global spike_force, accelerationL1, accelerationR1, fsrForceL1, fsrForceL2, fsrForceL3, fsrForceR1, fsrForceR2, fsrForceR3

    accel_spikeL1, accel_spikeR1, fsr_spikeL1, fsr_spikeL2, fsr_spikeL3, fsr_spikeR1, fsr_spikeR2, fsr_spikeR3 = detect_spikes(
            accelerationL1, accelerationR1, fsrForceL1, fsrForceL2, fsrForceL3, fsrForceR1, fsrForceR2, fsrForceR3
        )
```

Fig 3: Portion of Python central computer code using Matplotlib and Tkinter

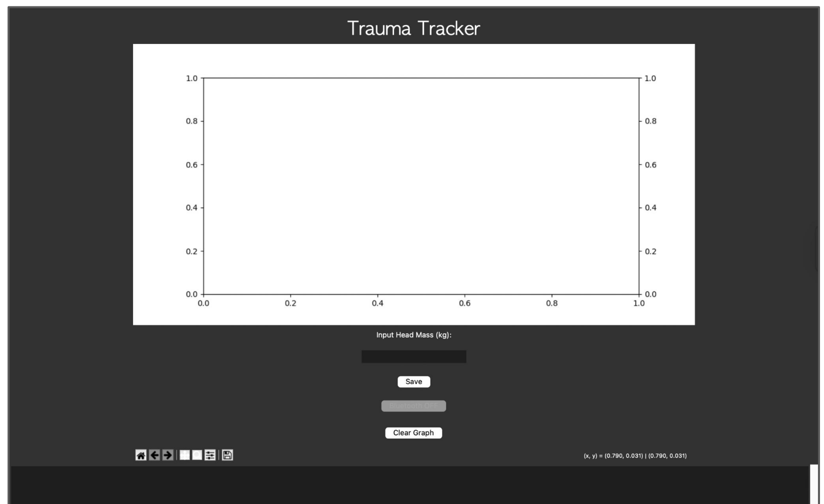


Fig 4: Python user-interface to present FSR readings, accelerometer data, and force calculations

Part 3 - Procedures - Protective Case Design

- Protected FSRs and LIS3DH accelerometers with **3D printed snap-fit cases** and padded boards with **EVA closed-cell porous foam**.
- Applied **heat shrink tubing** to secure FSRs with female wire headers.
- Backed FSRs with thin, strong **cardboard**
- Inserted PCB, accelerometers, and FSRs into the helmet with **velcro pads**.

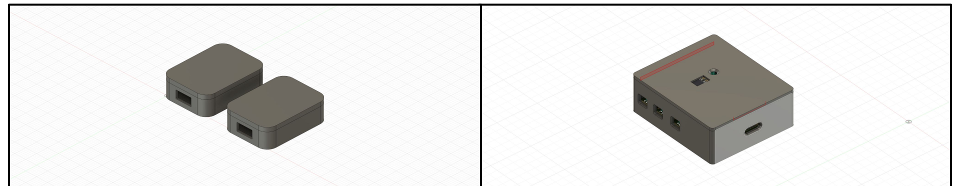


Fig 5: CAD models for LIS3DH and PCB cases

Fig 6: One finalized FSR

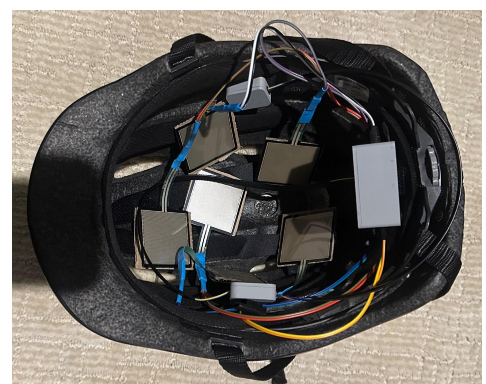


Fig 7: Example set up in a youth biking helmet

Part 3 - Procedures - Testing

- Applied impacts to device at known force values in Newtons by throwing a helmet, with the device installed, into the ground.
- Taped the FSRs to the outside of the helmet due to the lack of an opposing force into the sensor. In an authentic application, the FSRs would be placed inside the helmet and would recognize forces returned from the head of the subject.
- Used **Vernier acceleration sensors** read the acceleration of the helmet and kept the mass at a constant ~2.66 kg to get as close as I could to a realistic head mass.
- Through Newton's 2nd Law of Motion, I used the product of the Vernier acceleration values and the constant head mass as my expected force value.
- To measure the accuracy of my device, I compared the experimental force values from the device with expected force values from the Vernier acceleration sensors.

Fig 8: Testing setup with Vernier motion sensor

Part 4 - Results

- The average percent error was approximately **13.94%** during drop tests
- When attempting to increase the force by acceleration, the Vernier sensor would provide data at too large of a range to consider reliable.
- The prototype effectively recorded head impacts, but sometimes it displayed two impacts when only one occurred.

Table 1: Force values comparisons of 10 drop-test trials

Expected Force (N)	Experimental Force (N)	% Error
27.89	23.61	15.35%
26.15	22.32	14.64%
25.88	21.49	16.98%
27.15	24.32	10.43%
23.70	19.82	16.38%
23.33	25.97	11.34%
28.36	24.97	11.96%
25.41	20.89	17.80%
24.68	20.32	17.66%
26.52	29.76	12.21%
28.31	25.89	8.55%

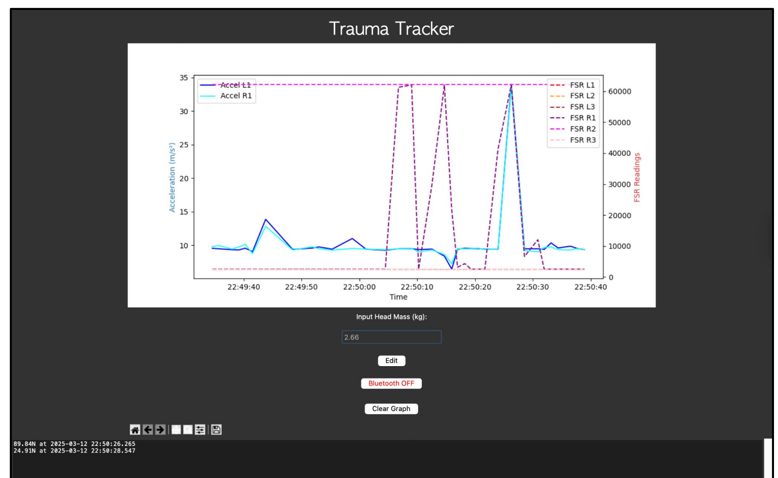
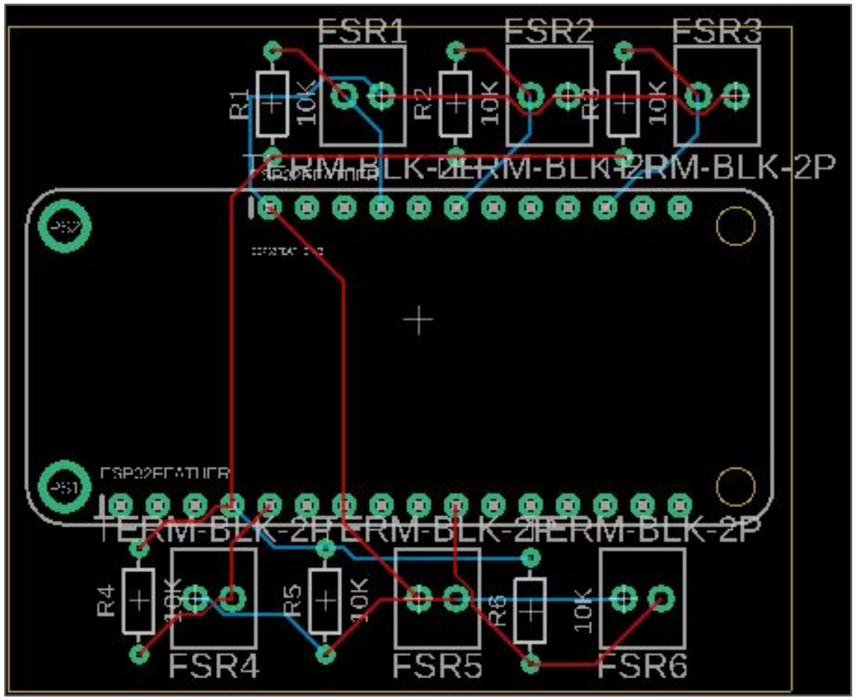
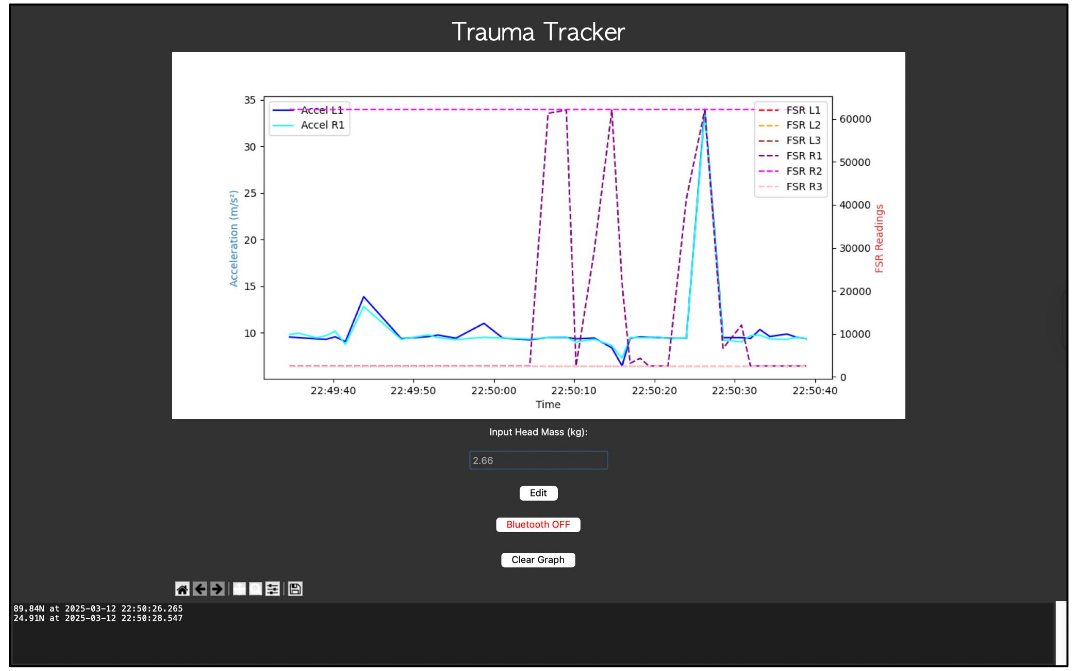
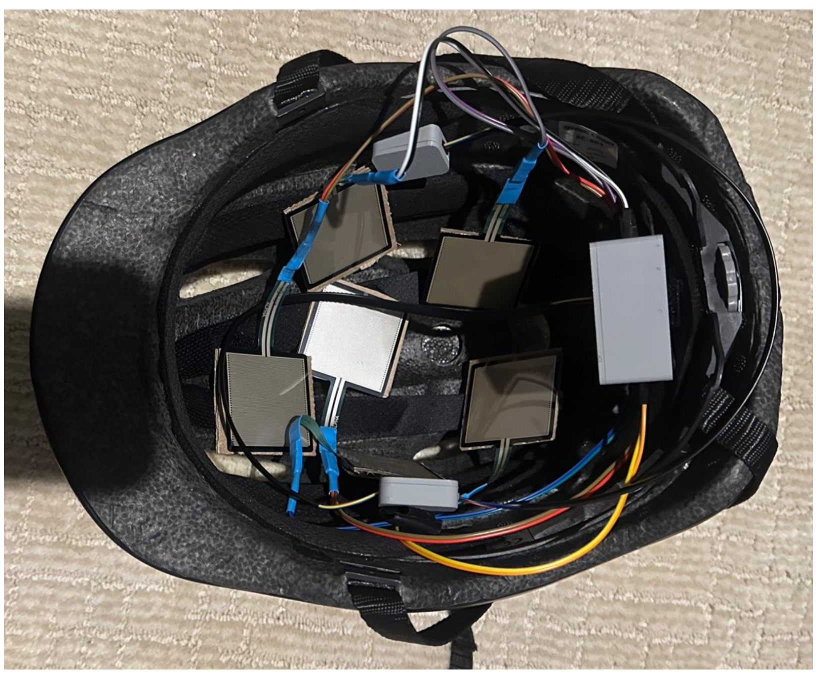
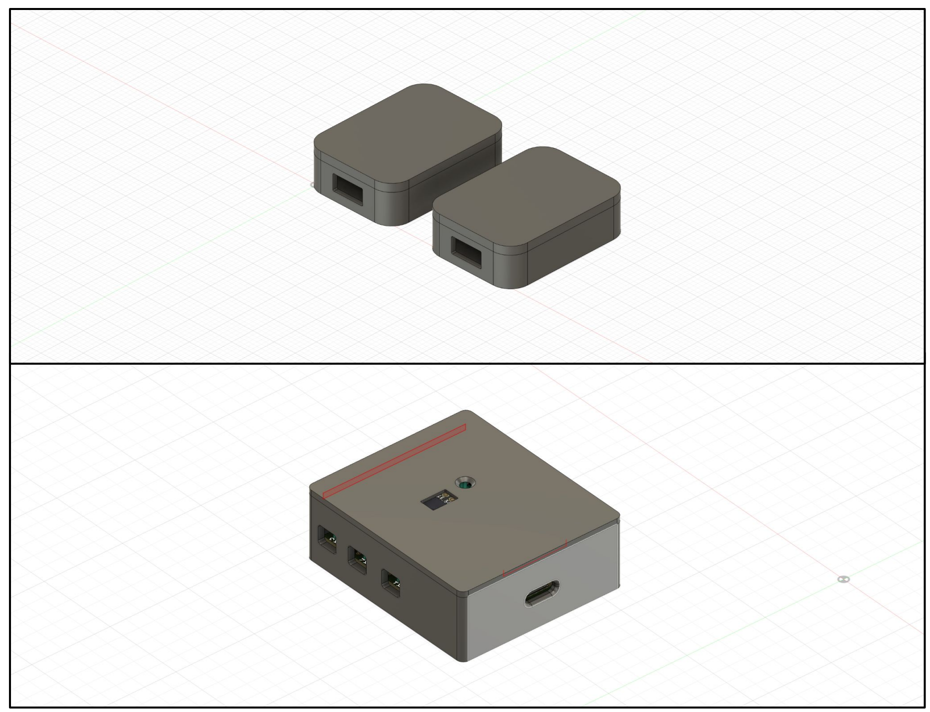


Fig 9: Graph and force calculation results


FSR1
FSR2
FSR3
R1
R2
R3
10K
TERM-BLK-2P
FSR4
FSR5
FSR6
R4
R5
R6


Trauma Tracker
Accel L1
Accel R1
FSR L1
FSR L2
FSR L3
FSR R1
FSR R2
FSR R3
Acceleration (m/s²)
FSR Readings
Time
22:49:40
22:49:50
22:50:00
22:50:10
22:50:20
22:50:30
22:50:40
Input Head Mass (kg):
2.66
Edit
Bluetooth OFF
Clear Graph
89.84N at 2025-03-12 22:50:26.265
24.91N at 2025-03-12 22:50:28.547





Part 5 - Discussion

Overall Evaluation

- The prototype worked as expected by recording head impacts and measuring the force of each impact in Newtons. It effectively displayed changes in head acceleration through a live graph, allowing medical professionals to gain insight into the RHIs of their personnel.

Possible Errors and Limitations

- Although the error percentage was slightly higher than anticipated, I lacked sufficient equipment to accurately provide expected force values.
- The PCB design and FSR layout may have caused electrical noise within the circuit, leading to inaccurate FSR readings.

Problems and Questions During Procedure

- While soldering the components to the PCB, I caused the ESP32 to malfunction, leading me to retry the soldering process
- The overall size of the device became slightly larger and more complicated than anticipated, suggesting the need for revision.

World Impact

- My prototype is an advancement in enabling medical professionals, specifically in athletics and the military, to track RHIs and avoid their patients experiencing a brain injury.
- There is little known to what specific level of force can cause brain injuries, and my project's implementation will provide a path to better understand the causes of head trauma.

Part 6 - Conclusions

Final Statements

- Developed a prototype that accurately records and measures head impacts. It maintained a relatively constant percent error at a various range of force values, demonstrating the device’s consistency in recording RHIs.
- My design fulfills the lack of insight into head trauma and allows athletes and military to avoid concussions caused by RHIs
- A recent study attempted to build machine learning algorithms only using baseline data from a previous study (Castellanos et al., 2021). This suggests the mass implementation of my device will advance machine learning algorithms to accurately predict whether a subject has a RHI-based concussion.

Extension Opportunities

- Reducing the overall size of the device will boost its comfort and versatility therefore a different microcontroller may be needed.
- Spaces between FSRs impairs the device’s ability to detect impacts, so a custom sensor is a possible solution to boost accuracy.

Part 7 - References

Castellanos, J., Phoo, C.P., Eckner, J.T. et al. Predicting Risk of Sport-Related Concussion in Collegiate Athletes and Military Cadets: A Machine Learning Approach Using Baseline Data from the CARE Consortium Study. *Sports Med 51*, 567–579 (2021). https://doi.org/10.1007/s40279-020-01390-w

Ching, R. P., Lee, M. C., & Young, D. (2002). Technical brief: Head mass and circumference correlation. *Snell Memorial Foundation*. Retrieved from https://smf.org/docs/articles/pdf/chingtechbrief.pdf

D'Arcy, R.C.N., McCarthy, D., Harrison, D. et al. An objective neurophysiological study of subconcussion in female and male high school student athletes. *Sci Rep 14*, 28929 (2024). https://doi.org/10.1038/s41598-024-80262-z

Lippa, S. M. (2024). A review of long-term outcomes of repetitive concussive and subconcussive blast exposures in the military and limitations of the literature. *The Clinical Neuropsychologist*, 1–36. https://doi.org/10.1080/13854046.2024.2441395

McKee AC, Alosco ML. Assessing Subconcussive Head Impacts in Athletes Playing Contact Sports—The Eyes Have It. *JAMA Ophthalmol*. 2019;137(3):270–271. doi:10.1001/jamaophthalmol.2018.6199