

An sEMG-Controlled 3D-Printed Prosthetic Hand Using a Compact 1D-CNN Pipeline: Prototype Integration and Preliminary Bench Evaluation

Munay M. Aljighdafi^{1*}, Suad M. Abraheem², Areej N. Hamad³, Ahlam Salah Ashour⁴

^{1,2,3}Department of Biomedical Engineering, College of Medical Technology - Benghazi, Benghazi,
Libya

⁴Department of Mechanical Engineering, College of Engineering, Benghazi, Libya

Email: ¹munay.aljighdafi@gmail.com, ²souadelfaidy@gmail.com, ³areej99al@gmail.com,
⁴ahlamalfaitouri@gmail.com

Received: June 09, 2026

Accepted: September 15, 2026

Published: October 15, 2026

DOI: 10.65998/ijees.v4i2.192

Abstract: Myoelectric prosthesis research requires control chains that are reproducible on constrained hardware and transparent about evidential limits. This work presents a 3D-printed upper-limb prototype combining surface electromyography (sEMG) sensing, an InMoov hand, five tendon-driven servo actuators, an ESP32-class controller, and an archived compact one-dimensional convolutional neural network (1D-CNN) pipeline. The original student project involved one participant without upper-limb loss. For the present revision, the surviving archive was audited rather than reconstructed by imputation: a 28 × 19 feature matrix, a 28-entry binary prediction file, and a 1,620-value single-channel EMG trace were inspected together with the preserved Python and C++ appendices. The training code generated target labels from an amplitude rule applied to the same input values, so the CNN is interpreted as reproducing a threshold-derived mapping rather than as independently validated physiological intent recognition. Applying the archived training rule to the surviving feature matrix yields 14 rows per class, and the preserved predictions agree with those threshold-derived labels for all 28 rows. Replaying the documented 80/20 random split produces 22 training and 6 held-out rows; exact duplicate rows from the training set are present in 3 of the 6 held-out rows, confirming substantial leakage risk. The CNN architecture remains identifiable as two Conv1D/max-pooling stages followed by a 64-unit dense layer and sigmoid output (approximately 18,753 trainable parameters), but full CNN inference on the ESP32 cannot be verified from the surviving firmware. The contribution is therefore a transparent engineering proof-of-concept and archive audit that preserves the 1D-CNN as a documented component while separating pipeline reconstruction from claims of independent predictive or clinical performance.

Keywords: Surface electromyography; prosthetic hand; 3D printing; convolutional neural network; embedded rehabilitation robotics.

1. Introduction

Upper-limb loss can restrict manipulation, self-care, employment, and social participation. Although active prostheses can restore selected functions, device acceptance depends on more than mechanical capability. Reliability, comfort, training burden, control intuitiveness, weight, maintenance, and cost all influence whether a prosthesis is used in daily life [1]. These constraints are particularly important in settings where access to specialized fitting, replacement parts, and high-cost commercial hands is limited.

Surface electromyography provides a non-invasive route for detecting residual muscle activation and mapping it to prosthetic commands. The signal contains information about motor intent, yet its amplitude and spectral structure vary with electrode placement, contraction intensity, fatigue, limb position, skin condition, and acquisition hardware [2]. Classical pattern-recognition controllers demonstrated that short windows of myoelectric activity can support multifunction control [3]. Later datasets such as NinaPro enabled standardized evaluation across many movements and participants [4], while deep-learning studies showed that convolutional models can learn useful signal representations directly from minimally engineered inputs [5,6].

Algorithmic performance alone does not guarantee a useful prosthesis. Offline accuracy may overestimate behavior in a closed control loop, where the user observes the device and continuously adapts muscle activation. Modern evaluations therefore increasingly examine online selection time, false activations, robustness across sessions and limb positions, calibration burden, and embedded computational cost [7-9]. At the same time, open-source and additively manufactured hands demonstrate that functional prototypes can be produced from accessible components [10,11]. The engineering challenge is to connect these two directions: resource-aware mechanics and electronics on one side, and defensible intent decoding on the other.

The present study addresses the integration challenge through a binary opening-closing prototype. Its contribution is not a new CNN architecture or a clinical demonstration. Rather, it lies in documenting a complete sensing-to-actuation chain and in auditing the surviving 1D-CNN pipeline without converting archival inconsistencies into unsupported performance claims. The revised manuscript therefore distinguishes procedures carried out in the original student project from analyses performed during the present archive audit.

The study objectives were to: document the construction of an anthropomorphic hand from an open-source design; describe the archived forearm sEMG acquisition and binary control pathway; reconstruct the compact 1D-CNN architecture from preserved code; audit the surviving data files for class construction, split behavior, duplicate leakage, and prediction consistency; document embedded hand actuation; and define the experimental work required to progress from a bench prototype to a validated assistive system. This work contributes an end-to-end integration of sEMG acquisition, signal preparation, binary state mapping, embedded control, and tendon-driven actuation in a 3D-printed hand using open-source and off-the-shelf elements. It also provides a transparent evidence audit that separates documented observations, quantities recalculated from surviving files, and claims that cannot be verified. The scope is engineering integration, archival reproducibility, and evidential transparency rather than classifier novelty, cost benchmarking, or clinical generalization.

2. Related Work

2.1 Myoelectric pattern recognition

Early multifunction myoelectric controllers relied on engineered time-domain features and statistical classifiers [3]. Their appeal is computational efficiency, but their performance can change when the electrode configuration, limb posture, or session changes. Farina et al. emphasized that the clinical translation of sEMG control requires extracting stable neural information while accounting for the non-stationary measurement interface [2]. Public datasets subsequently made it possible to compare algorithms across larger movement sets and both intact-limb and amputee cohorts [4].

CNN-based approaches reduce dependence on handcrafted features by learning filters from the data. Atzori et al. demonstrated that relatively simple CNNs could classify a broad set of movements in NinaPro recordings [5]. Instantaneous sEMG image representations and multi-view networks further explored spatial structure across channels [6,12]. Ameri et al. evaluated CNN-based simultaneous control in real time [13], while Cote-Allard et al. used transfer learning to reduce the amount of user-specific training required [14]. More recent work has targeted position awareness, force estimation, user adaptation, and robustness across subjects [15-17].

2.2 Embedded and affordable prosthetic systems

Low-cost hands such as Tact and related open-source platforms demonstrate how additive manufacturing, compliant mechanisms, off-the-shelf actuators, and standardized sockets can lower fabrication barriers [10,11]. Embedded recognition introduces an additional constraint: model size, inference time, sensor throughput, actuator power, and control-loop stability must be considered together. Lightweight CNNs have been used for real-time robotic-arm control [18], while compact non-deep representations can achieve competitive performance with very small memory requirements [19]. These studies suggest that a useful low-cost contribution should report both classification metrics and system-level measurements rather than treating the classifier in isolation.

Table 1. Position of the present prototype is relative to representative prior work.

Study	Evaluation basis	Control problem	Relevance to this prototype
Atzori et al. [5]	67 intact-limb and 11 amputee participants across NinaPro subsets	CNN classification of many hand movements	Establishes the need for multi-user, independently labeled data
Tam et al. [7]	12 able-bodied participants; online trials	Six gestures with transfer learning	Reports calibration burden, online PPV, and selection latency
Williams et al. [8]	Multi-position myoelectric recordings	Position-aware recurrent CNN control	Highlights robustness to posture as a validation target
Fratti et al. [17]	Benchmark datasets including healthy and amputee participants	Multi-scale CNN with user adaptation	Shows current expectations for cross-subject evaluation
Present work	One participant without upper-limb loss; archived bench record	Binary open-close mapping using threshold-derived labels and an archived compact 1D-CNN	Demonstrates system integration and archive reconstruction, but not independent intent recognition or clinical generalization

3. Materials and Methods

3.1 Study design and evidence provenance

This study is an engineering proof-of-concept with a retrospective audit of an earlier bench prototype. Procedures performed in the original student project included mechanical fabrication, sEMG acquisition, threshold-based class construction, 1D-CNN training, servo-control programming, and bench photography. Procedures performed specifically for the present revision were limited to inspection of the surviving report, appendices, and partial CSV artifacts; recalculation of file dimensions and class balance; reproduction of the documented random split; exact-duplicate analysis; and comparison of the archived prediction file with labels generated by the preserved training rule. No new human data were collected, no missing observations were imputed or simulated, and no model retraining is presented as a recovered result.

Three partial data artifacts survive: (i) a feature matrix containing 28 rows and 19 numeric values per row; (ii) a prediction file containing 28 binary outputs; and (iii) a single-column EMG sequence containing 1,620 scalar readings. The archive also describes one 23-year-old female participant without upper-limb loss and states that 100 samples were collected for gripping and 100 for release. The surviving files do not contain trial identifiers, session identifiers, gesture markers, timestamps, window-overlap metadata, or a mapping between the stated 100/100 counts and the 28-row feature matrix. The trained model, fitted scaler, executable notebook, complete timing log, and final firmware that would permit full reproduction are not available. Accordingly, exact sample accounting at the original trial level remains unresolved and is reported as a limitation rather than harmonized retrospectively.

3.2 Mechanical and electronic architecture

The hand was based on the open-source InMoov geometry and fabricated by additive manufacturing. The surviving archive is inconsistent about whether the printed polymer was PLA or ABS, and it does not preserve the printer model, nozzle size, layer height, infill, CAD version, printing time, material consumption, or final device mass. These values are therefore not supplied retrospectively. What is documented is the articulated finger construction, braided 0.8-mm fishing-line tendons, five MG995-class servo motors, and 3 x 20-mm fasteners joining the palm and forearm shell.

The tendon paths mechanically couple the servos to the digits to produce flexion and extension. Moreover, an EMG sensor conditioned the differential muscle potential detected by three surface electrodes. The documented anatomical placement assigned one active electrode to the muscle belly, a second to the end of the muscle group, and a reference electrode to a nearby low-activity or bony site. The surviving archive does not identify the sensor manufacturer/model, amplifier gain, electrode manufacturer/type, inter-electrode distance, skin-preparation protocol, ADC resolution, or ADC voltage range. The sensor was described as using two 9-V batteries to obtain positive and negative rails, while an ESP32-class microcontroller generated pulse-width-modulated servo commands. A complete regulated actuator-supply specification, verified current budget, grounding diagram, motor-noise suppression scheme, overcurrent protection, and emergency shutdown mechanism are not preserved; consequently, electrical safety and power adequacy are not claimed as validated.



Figure 1. Reconstructed sensing-to-actuation and data path. The compact 1D-CNN is retained as an archived mapping stage; exact on-device CNN inference is not claimed.

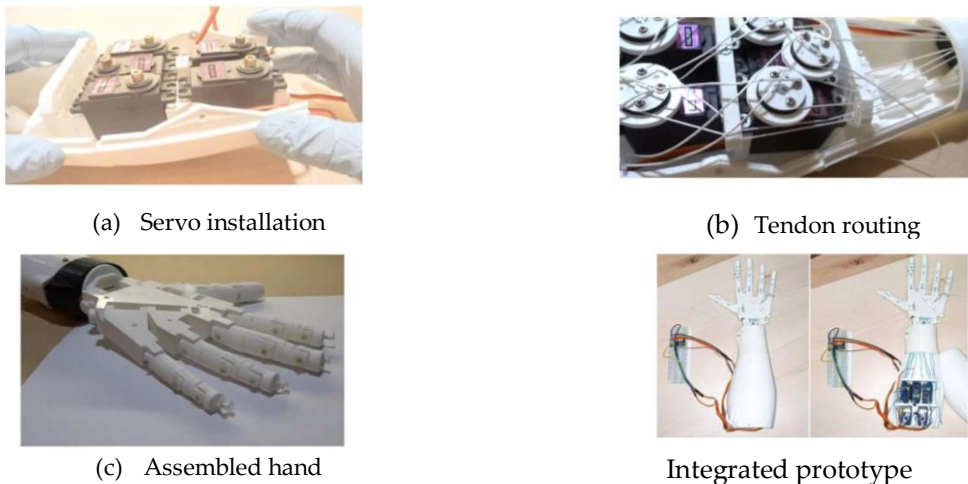


Figure 2. Mechanical assembly and integration of the 3D-printed hand. Images are reproduced from the authors' project archive.

Table 2. Principal hardware and software elements preserved in the archive.

Subsystem	Element	Recorded role
Sensing	Three-electrode EMG sensor	Condition residual muscle activity and provide an analog output
Acquisition	Arduino-compatible ADC	Read the analog EMG channel and stream integer samples
Decision	ESP32-class controller	Receive or execute a binary state decision and issue motor commands
Actuation	Five MG995-class servos	Pull artificial tendons to change the five-finger configuration
Mechanics	3D-printed InMoov hand	Provide the articulated palm, digits, joints, and forearm enclosure
Software	Python, TensorFlow/Keras, Arduino IDE	Prepare data, train the archived 1D-CNN, and program the controller

3.3 Signal acquisition and preprocessing

The preserved acquisition loop used `analogRead` on channel A0, transmitted each reading at 9600 baud, and inserted a 10-ms delay. Ignoring serial transmission and ADC-conversion overhead, that loop implies a nominal upper bound of approximately 100 samples/s rather than a measured 1000-Hz

acquisition rate. A separate Python appendix specified a fourth-order Butterworth low-pass filter with a 10-Hz cutoff and set a sampling-frequency variable to 1000 Hz, while an archived figure was labeled as a moving-average or Butterworth-bandpass result in different places. The surviving 1,620-value EMG trace contains no timestamps. Therefore, the actual measured sampling frequency, deployed filtering method, window timing, and real-time causality cannot be reconstructed reliably. The revision reports these contradictions explicitly and does not use the 1000-Hz variable as an acquisition fact.

Table 3. Surviving data artifacts, acquisition details, and preprocessing limitations.

Item	Archived evidence	Defensible interpretation
Surviving feature matrix	28 rows x 19 numeric values	Exact file shape is known; row-to-trial/session mapping and window overlap are not preserved.
Threshold-derived class balance	Applying all(values>200) gives 14 class-0 and 14 class-1 rows	This is deterministic label reconstruction, not independent ground truth.
Surviving prediction file	28 binary outputs; 14 zeros and 14 ones	Predictions agree with threshold-derived labels for all 28 rows; this is a consistency check, not independent validation.
Surviving raw EMG trace	1,620 single-channel scalar values	No timestamps, gesture boundaries, session identifiers, or trial labels are present.
Original project count	100 grip and 100 release “samples” stated in the report	The term “sample” cannot be mapped to ADC points, windows, repetitions, or trials from the surviving archive.
Acquisition loop	analogRead on A0, 9600 baud, 10-ms delay	Nominal upper bound is approximately 100 samples/s when overhead is ignored; actual measured rate is unavailable.
Python sampling-frequency variable	1000 Hz	This value conflicts with the acquisition loop and is not treated as measured sampling frequency.
Filter specification	Fourth-order Butterworth low-pass; 10-Hz cutoff	Present in archived Python code, but the exact deployed preprocessing cannot be verified.
Other preprocessing descriptions	Moving average and Butterworth bandpass also appear in the archive	The competing descriptions cannot be reconciled without the executable notebook.
ADC/sensor/electrode specifications	Resolution, voltage range, sensor model/gain, electrode type/distance and skin preparation not preserved	These specifications are explicitly reported as unavailable rather than inferred.

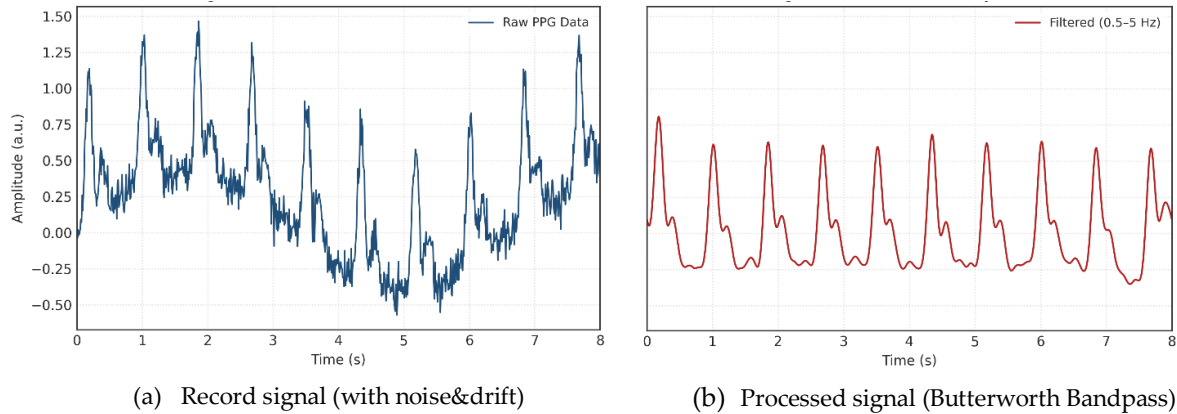


Figure 3. Archived sEMG traces before and after smoothing. The figure label and preserved Python code use inconsistent filter descriptions; the exact deployed filter and sampling rate cannot be independently verified.

3.4 Ground-truth construction, threshold controller, and risk of circular validation

The project used an amplitude threshold of 200 ADC units to construct binary states, but two different rules survive. In the Python training script, a row is assigned to class 1 only when every value in that row exceeds 200; otherwise it is assigned to class 0. Applying this rule to the surviving 28 x 19 feature matrix produces 14 rows in each class. In the threshold-controller firmware, by contrast, the presence of any value above 200 in a fixed array triggers hand closure. These rules represent different archived implementations and no surviving evidence identifies a single verified deployment rule. The threshold is therefore documented as an inherited project setting rather than a newly calibrated physiological decision boundary.

More importantly, the CNN targets were generated deterministically from the same amplitude values supplied as model inputs. The archived prediction file contains 28 binary outputs and agrees with the threshold-derived labels for all 28 surviving feature rows. This agreement verifies consistency of the archived mapping but cannot demonstrate independent physiological intent recognition. Replaying `train_test_split(test_size=0.2, random_state=42)` on the 28-row matrix yields 22 training rows and 6 held-out rows. Fourteen rows in the feature matrix belong to seven exact duplicate pairs; under the documented split, 3 of the 6 held-out rows have an identical counterpart in the training set. This confirms that a random row-level split is leakage-prone for the surviving data and precludes interpretation of the internal accuracy as a generalization estimate.

3.5 Archived 1D-CNN

The preserved Python training script standardizes each feature column using a `StandardScaler` fitted on the training subset and reshapes the input to a one-channel sequence. Because the surviving feature matrix contains 19 columns and the model input is defined programmatically as `X_train.shape[1]`, the executable training pipeline corresponds to a 19-point input sequence. The network uses two valid `Conv1D` layers with 32 and 64 filters, each followed by `max` pooling, then a 64-unit rectified-linear dense layer, 50% dropout, and a single sigmoid output. Binary cross-entropy and Adam optimization are used for 15 epochs with a batch size of 32.

For a 19-point input, valid convolution and pooling produce 17×32 , 8×32 , 6×64 , and 3×64 intermediate shapes before flattening to 192 features; the resulting model has approximately 18,753 trainable parameters. The same training script uses the 20% held-out subset as validation data during fitting and later evaluates the model on that same subset, so it does not define a separate independent test set. A separate archived testing script creates a new `StandardScaler` and applies `fit` transform to the test file rather than reusing the training scaler, which further prevents that script from being treated as a faithful independent evaluation. These are properties of the preserved code, not newly inferred performance results.

Table 4. Reconstructed CNN architecture from the code and preserved weight shapes.

Stage	Configuration	Output shape	Parameters
Input	19 samples x 1 channel	19×1	0
Conv1D-1	32 filters; kernel 3; ReLU	17×32	128
MaxPool-1	Pool size 2	8×32	0
Conv1D-2	64 filters; kernel 3; ReLU	6×64	6,208
MaxPool-2	Pool size 2	3×64	0
Dense	192 to 64; ReLU; dropout 0.5	64	12,352
Output	64 to 1; sigmoid	1	65
Total	Archived compact 1D-CNN	-	18,753

3.6 Controller-to-hand mapping

The binary state was mapped to simultaneous commands for five servos, utilizing zero degrees for extension across all fingers. For the grip state, angles were set to 180 degrees for all fingers, except for the thumb, which used 135 degrees. Because the final calibrated firmware is unavailable, neither angle pair is designated retrospectively as the verified final setting. The photographs show that the integrated assembly could change configuration in response to programmed binary states. The C++ excerpt presented as model deployment contains a small dense layer rather than the two convolutional stages reconstructed from the Python code, declares an 18-row weight array while its example input contains 19 values, and therefore cannot verify execution of the full 1D-CNN on the ESP32. The evidence supports embedded actuation following a binary decision; the CNN is retained as an archived offline pipeline component rather than claimed as verified on-device inference.



(a) Open-hand state



(b) Commanded alternative state

Figure 4. Prototype states preserved in the benchrecord. The photographs demonstrate actuator-driven configuration changes, not functional grasp performance or clinical usability.

4. Results

4.1 Bench prototype realization

The manufacturing sequence produced an integrated forearm-hand assembly containing five servos, tendon routing, printed joints, controller wiring, and an external EMG acquisition path. The project photographs document the major build stages and the completed device. At bench level, the controller generated coordinated opening and closing commands across the digits. No standardized grasp taxonomy, object-manipulation task, load test, endurance test, or validated activities-of-daily-living measure were recorded. Accordingly, the demonstrated outcome is binary state actuation rather than dexterous or multifunctional grasping.

4.2 Archive audit and classifier behaviour

The preserved training log reports training accuracy increasing from 0.4545 at epoch 1 to 1.0000 at epoch 6 and remaining at that value through epoch 15; validation accuracy is reported as 0.8333 at epoch 1 and 1.0000 from epoch 2 onward. Training loss decreases from 0.7272 to 0.2350 and validation loss from 0.6298 to 0.1155. These curves show that the compact network fitted the archived threshold-derived mapping.

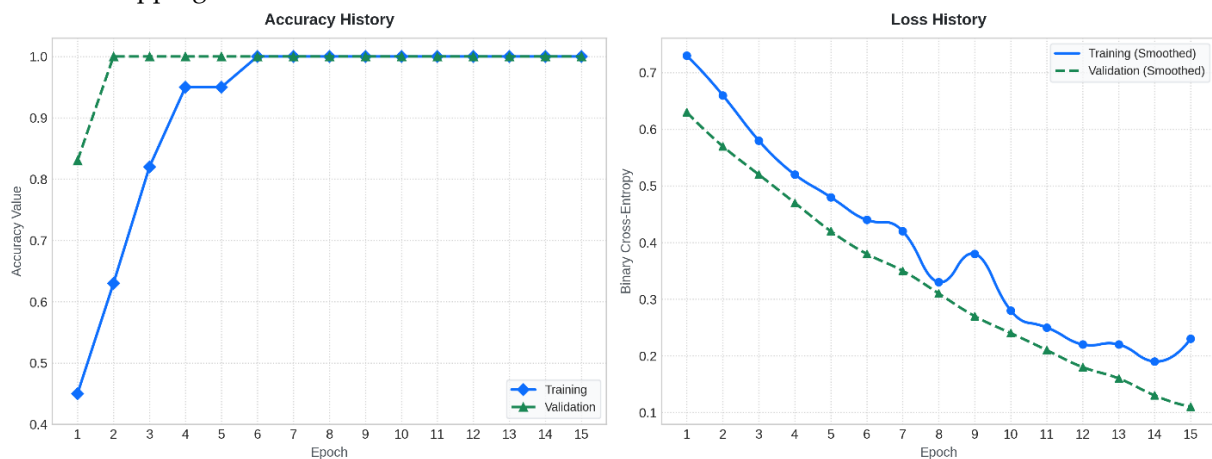


Figure 5. Learning curves reconstructed exactly from the 15-epoch values reported in the project archive.

The surviving 28×19 feature matrix yields 14 threshold-derived rows per class. The separate prediction file contains 28 binary outputs and matches these derived labels in all 28 cases; this corresponds to perfect agreement with the threshold rule, not a ground-truth confusion matrix for physiological intent. Reproduction of the documented split yields 22 training and 6 held-out rows, and exact duplicate analysis shows that 3 of the 6 held-out rows have an identical counterpart in training. A separate archived chart reports test accuracy of 0.75, but it cannot be reconciled with the code path that reuses the held-out subset for validation and evaluation, nor with the 28-entry prediction file. Accordingly, 1.00 and 0.75 are retained as discordant archival outcomes, while the 28/28 agreement is reported only as a pipeline-consistency result.

Table 5. Evidence status and defensible interpretation of the major archived outcomes and claims.

Archived item	Preserved/recalculated value	Defensible interpretation
Training log	Accuracy 0.4545 to 1.0000	Shows fitting of threshold-derived labels; not independent intent validation.
Validation log	0.8333 at epoch 1; 1.0000 thereafter	Internal 20% split used during training; not a separate independent test set.
Feature matrix	28 rows x 19 values; 14 threshold-derived rows per class	Provides partial sample accounting only; trial/session provenance is absent.
Archived predictions	28 binary outputs; 28/28 agreement with threshold-derived labels	Pipeline consistency, not physiological ground-truth accuracy.
Random-split replay	22 training rows; 6 held-out rows	Reproduces the code path for the surviving matrix.
Exact duplicates	7 duplicate pairs; 3/6 held-out rows duplicated in training	Direct evidence of leakage risk under the archived random split.
Separate test chart	Accuracy 0.75	Discordant archival result that cannot be reconciled with the surviving code/files.
Testing scaler	Separate script applies fit_transform to test data	Does not reuse the training scaler; unsuitable as evidence of faithful out-of-sample evaluation.
Bench photographs	Two commanded hand states	Physical actuation is demonstrated; grasp performance is not quantified.
1D-CNN architecture	Two Conv1D stages; ~18,753 trainable parameters	Architecture is reconstructable; full on-device execution is not verified.
Cost evidence	No preserved bill of materials, printing time, material consumption, or final mass	No quantitative low-cost claim is made in the revised manuscript.
Clinical effectiveness	No residual-limb fitting or intended-user functional testing	Not evaluated and not claimed.

5. Discussion

5.1 Meaning of the proof-of-concept.

The strongest outcome is the documented assembly of a complete resource-conscious control pathway linking surface electrodes, signal conditioning, numerical processing, a binary decision stage, microcontroller commands, five actuators, and a printed anthropomorphic hand. The surviving archive also permits reconstruction of the compact 1D-CNN architecture and a limited consistency audit of its threshold-derived mapping. This is a legitimate engineering contribution because failures often emerge at subsystem interfaces; however, the present evidence does not support claims that the CNN independently recognized motor intent or improved functional control.

5.2 Embedded feasibility versus verified deployment

The reconstructed 1D-CNN contains approximately 18.8 thousand trainable parameters, corresponding to roughly 75 kB of 32-bit floating-point weights before activation memory and runtime overhead. That scale is compatible in principle with ESP32-class inference, but model size alone does not demonstrate deployment. Exact preprocessing constants, scaler parameters, convolution kernels, tensor shapes, quantization state, memory allocation, and inference timing would need to be exported and verified using identical Python and embedded test vectors. The surviving C++ excerpt does not implement the reconstructed two-stage Conv1D model, so the revised claim is limited to embedded binary actuation while the CNN remains documented as an archived offline pipeline component.

5.3 Why the apparent perfect accuracy must be qualified.

A classifier intended to replace a threshold must be evaluated against labels that are independent of that threshold. Here, the target is a direct mathematical function of the same 19 amplitude values presented to the CNN. The threshold rule therefore achieves perfect agreement with those labels by definition. The archived CNN predictions also agree with the threshold-derived labels for all 28 surviving rows, so no incremental benefit over the threshold baseline can be established. In addition, seven exact duplicate pairs exist in the feature matrix and three of six held-out rows in the documented random split have identical training counterparts. These findings provide a concrete explanation for apparently perfect internal accuracy and show why additional LDA, logistic-regression, SVM, random-forest, or shallow-network comparisons on the same leakage-prone, threshold-labeled archive would not constitute a valid baseline study. Such comparisons should be performed only on newly acquired, independently labeled trials split by trial, session, or participant.

5.4 Hardware, power, and control implications

Simultaneous actuation of five MG995-class servos can demand substantial transient current. The surviving archive does not preserve a verified regulated actuator rail, peak-current measurements, grounding architecture, motor-noise suppression, isolation details, overcurrent protection, or emergency shutdown. The two 9-V batteries described for the EMG front end should not be interpreted as evidence that the actuator supply was adequate. Accordingly, the prototype is treated as a laboratory bench platform, not as a wearable device with demonstrated electrical safety. A future build should document separate regulated sensing and actuator supplies, common-reference strategy where appropriate, protection, noise suppression, and measured peak/average current before human-connected testing.

Timing must likewise be measured across the complete control loop rather than inferred from visible motion. Relevant intervals include ADC acquisition, window accumulation, preprocessing, CNN inference or threshold decision, PWM update, and mechanical travel. The original discussion mentioned a response below 0.5 s, but no synchronized timing trace or repeated protocol survives; the value is therefore not retained as a quantitative result. Future evaluation should report inference latency, full-loop response time, false activations, repeated transition success, servo-position error, grip force, payload, object-holding success, actuator current, battery runtime, thermal behavior, and robustness to electrode displacement, fatigue, and limb-position changes.

5.5 Clinical and translational interpretation

The prototype was tested with a participant without upper-limb loss and was not fitted to a residual limb. Muscle anatomy, available contraction patterns, skin-electrode coupling, fatigue, and compensatory strategies can differ substantially after amputation. The device should therefore not be described as clinically validated, adaptive across users, or suitable for daily wear. Translation would require ethics oversight, informed consent, socket and safety engineering, testing with intended users, and functional outcomes that go beyond classifier accuracy. The present system is better described as a laboratory platform for developing and teaching low-cost myoelectric control.

6. Limitations and Future Work

The main limitation is incomplete preservation of the original experimental record. Three partial CSV artifacts survive, but they lack the trial/session metadata, timestamps, independent gesture labels, and acquisition settings required for reproducible validation. The trained model, fitted scaler, executable notebook, final firmware, full timing records, cost records, and several mechanical/electrical specifications are also unavailable. The original experiment used one participant without upper-limb loss, two gross hand states, amplitude-derived labels, and a random row-level split. The surviving feature matrix contains exact duplicates that cross the reconstructed train/held-out split, further invalidating the internal accuracy as a generalization estimate. No cross-session, cross-position, fatigue, electrode-shift, object-grasp, endurance, or amputee testing is documented.

The next experimental phase should begin with a new, version-controlled dataset rather than attempting to manufacture missing observations. Each trial should store participant code, session, limb position, instructed gesture, repetition, synchronized timestamps, raw ADC values, electrode placement, acquisition settings, and explicit event markers. Labels should be assigned from the instructed gesture, synchronized event marker, or another independent reference. Training, validation, and testing should be separated by complete trials before window generation and preferably by session; participant-independent or leave-one-subject-out evaluation should be added when sample size permits. The threshold controller should be evaluated beside the 1D-CNN and lightweight classical baselines using confusion matrices, sensitivity, specificity, precision, recall, F1 score, balanced accuracy, confidence intervals, and repeated seeds.

For embedded validation, the fitted scaler and trained 1D-CNN should be exported together to a supported inference runtime and checked against identical Python vectors before motor integration. Model size, RAM use, numerical agreement, inference time, full-loop latency, false-activation rate, actuator current, battery runtime, thermal behavior, and repeated state-transition success should be

recorded. A complete bill of materials, unit and total costs, printing time, material use, final mass, printer settings, servo calibration, circuit diagram, and power-distribution architecture should also be documented before restoring any quantitative low-cost or wearable-safety claim.

7. Conclusion

This project demonstrates the feasibility of integrating surface EMG sensing, a binary decision pathway, embedded control, five servo actuators, and an open-source 3D-printed hand in a laboratory prototype, while preserving a compact 1D-CNN as a documented part of the archived processing pipeline. The surviving evidence supports physical assembly, commanded hand-state changes, reconstruction of the CNN architecture, and a limited 28-row consistency audit showing that the archived CNN outputs reproduce the threshold-derived labels. It does not support independent physiological intent recognition, a generalizable 100% classification claim, verified execution of the full CNN on the ESP32, a measured sub-0.5-s response time, quantified low-cost performance, electrical safety, or clinical effectiveness. The appropriate interpretation is therefore an engineering proof-of-concept and archive audit that provides a transparent foundation for a new, independently labeled and leakage-resistant validation study.

Author Contributions: Authors have contributed significantly to the development and completion of this article.

Funding: This article received no external funding.

Acknowledgments: Acknowledgments:

The authors would like to thank the Department of Biomedical Engineering, College of Medical Technology, Benghazi, Libya, and the Department of Electrical and Electronic Engineering, University of Benghazi, Benghazi, Libya, for providing laboratory facilities, necessary equipment, and academic support.

Ethics approval: The original student project involved one adult volunteer. The surviving archive does not contain an ethics approval/exemption identifier; institutional confirmation of the applicable oversight status and consent should be obtained and reported before resubmission.

Conflicts of Interest: The author(s) declare no conflict of interest.

References

- [1] E. A. Biddiss and T. T. Chau, 'Upper limb prosthesis use and abandonment: A survey of the last 25 years,' *Prosthetics and Orthotics International*, vol. 31, no. 3, pp. 236-257, 2007. doi: 10.1080/03093640600994581.
- [2] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, and O. C. Aszmann, 'The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges,' *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 797-809, 2014. doi: 10.1109/TNSRE.2014.2305111.
- [3] K. Englehart and B. Hudgins, 'A robust, real-time control scheme for multifunction myoelectric control,' *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848-854, 2003. doi: 10.1109/TBME.2003.813539.
- [4] M. Atzori et al., 'Electromyography data for non-invasive naturally controlled robotic hand prostheses,' *Scientific Data*, vol. 1, art. 140053, 2014. doi: 10.1038/sdata.2014.53.
- [5] M. Atzori, M. Cognolato, and H. Muller, 'Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands,' *Frontiers in Neurorobotics*, vol. 10, art. 9, 2016. doi: 10.3389/fnbot.2016.00009.
- [6] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, 'Gesture recognition by instantaneous surface EMG images,' *Scientific Reports*, vol. 6, art. 36571, 2016. doi: 10.1038/srep36571.
- [7] S. Tam, M. Boukadoum, A. Campeau-Lecours, and B. Gosselin, 'Intuitive real-time control strategy for high-density myoelectric hand prosthesis using deep and transfer learning,' *Scientific Reports*, vol. 11, art. 11275, 2021. doi: 10.1038/s41598-021-90688-4.

- [8] H. E. Williams, A. W. Shehata, M. R. Dawson, E. Scheme, J. S. Hebert, and P. M. Pilarski, 'Recurrent convolutional neural networks as an approach to position-aware myoelectric prosthesis control,' *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 7, pp. 2243-2255, 2022. doi: 10.1109/TBME.2022.3140269.
- [9] J. Zbinden, J. Molin, and M. Ortiz-Catalan, 'Deep learning for enhanced prosthetic control: Real-time motor intent decoding for simultaneous control of artificial limbs,' *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 32, pp. 1177-1186, 2024. doi: 10.1109/TNSRE.2024.3371896.
- [10] P. Slade, A. Akhtar, M. Nguyen, and T. Bretl, 'Tact: Design and performance of an open-source, affordable, myoelectric prosthetic hand,' in *2015 IEEE International Conference on Robotics and Automation*, 2015, pp. 6451-6456. doi: 10.1109/ICRA.2015.7140105.
- [11] A. Akhtar et al., 'A low-cost, open-source, compliant hand for enabling sensorimotor control for people with transradial amputations,' in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2016, pp. 4642-4645. doi: 10.1109/EMBC.2016.7591762.
- [12] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, and W. Geng, 'Surface-electromyography-based gesture recognition by multi-view deep learning,' *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 10, pp. 2964-2973, 2019. doi: 10.1109/TBME.2019.2899222.
- [13] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, 'Real-time, simultaneous myoelectric control using a convolutional neural network,' *PLOS ONE*, vol. 13, no. 9, e0203835, 2018. doi: 10.1371/journal.pone.0203835.
- [14] C. Côté-Allard et al., "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 4, pp. 760–771, 2019. doi: 10.1109/TNSRE.2019.2892629.
- [15] B. Fang et al., 'Simultaneous sEMG recognition of gestures and force levels for interaction with prosthetic hand,' *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 2426-2436, 2022. doi: 10.1109/TNSRE.2022.3199809.
- [16] H. Wang, Y. Zhang, C. Liu, and H. Liu, "sEMG based hand gesture recognition with deformable convolutional network," *Journal of Machine Learning and Cybernetics*, vol. 13, no. 6, pp. 1729–1738, 2022. doi: 10.1007/s13042-021-01482-7.
- [17] R. Fratti, N. Marini, M. Atzori, H. Muller, C. Tiengo, and F. Bassetto, 'A multi-scale CNN for transfer learning in sEMG-based hand gesture recognition for prosthetic devices,' *Sensors*, vol. 24, no. 22, art. 7147, 2024. doi: 10.3390/s24227147.
- [18] B. Guo, Y. Ma, J. Yang, Z. Wang, and X. Zhang, 'Lw-CNN-based myoelectric signal recognition and real-time control of robotic arm for upper-limb rehabilitation,' *Computational Intelligence and Neuroscience*, vol. 2020, art. 8846021, 2020. doi: 10.1155/2020/8846021.
- [19] R. Mobarak et al., "MyoPose: Position-limb-robust neuromechanical features for enhanced hand gesture recognition in colocated sEMG-pFMG armbands," *Journal of Neural Engineering*, vol. 22, no. 4, art. 046044, 2025. doi: 10.1088/1741-2552/adf888.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third-party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.