山口大学教育学部研究論叢　第74巻　163頁～171頁　2025年3月

筋電位での機械学習による リアルタイム動作識別について

伊藤正剛*・北本卓也**

On Real-Time Motion Classification by Machine Learning Using Myoelectric Potential

ITO Masataka*, KITAMOTO Takuya**

(Received September 27, 2024)

パソコンにおいてスマートフォンのように直感的な操作方法があれば、パソコン操作に不慣れな高齢者や障がい者に役立つ道具となる。具体的には、手のジェスチャーによりテキストカーソルなどを動かし、パソコン操作の補助とすることである。以前の研究で、筋電位をもとに機械学習を用い手の動作識別を行ってきた。確実な動作識別を行うことができれば、テキストカーソルの操作を精度良く行うことができ、最終目標である筋電義手作成へ応用できる。本論文では、前々回の研究[1]と前回の研究[2]を発展させ、精度の高いリアルタイム動作識別とテキストカーソル操作を目指して行ったことを報告する。さらに、人手による場合分けの困難さを解決するために、新たな 2 段階の動作識別手法も紹介する。また、使用センサを変更し、安定した信号下での開発を行うことで実用化を目指す。

1. はじめに

表面筋電位（以下、筋電位）とは、表在筋を中心とした骨格筋筋線維全体に発生する活動電位のことであり、筋電位を計測し分析することによって様々なことを行うことができる。例えば、脳波によりロボット動作の誤りを自動検知し、筋電位を用いたジェスチャーによるエラー検出に重点をおいた監視制御を目標とする研究がある[3]。医用工学の分野では、筋電義手の技術を応用して、施術者の動きと連動したロボットハンドの作成などが試みられている[4]。

筋電位を用いて動作識別を行い、多自由度の電動義手を作成し、上肢切断者の腕の代わりとして使う研究も1960 年代から盛んに行われている。また、現在では筋電位信号自体の解析も進んでおり、平均 0 のガウス分布に従うと仮定されていた筋電位信号が、非ガウス的な性質を示す場合があるという観点から新しい信号解析法の創出を行っている研究もある[5]。

本論文では、コンピュータでのジェスチャー操作、将来的には筋電義手に応用するため、手の動作識別について扱う。正確な動作識別を行うには、複数の動作意図を筋電位データから抽出する必要がある。近年、機械学習の中でも深層学習に目覚ましい発展があり、筋電位の解析にも大きく影響を与えている。

機械学習における動作識別では、吉川らがサポートベクターマシンを分類器とし、変換を行った IEMG 信号を用いて本人では約 97%と高い識別率の実現を報告している[6]。深層学習に関しては、山野井らが畳み込みニューラルネットワークを用いて時間による変性に強いモデルを作成している[7]。また、ニューラルネットワークに関して以前から辻らが、確率統計理論に基づくネットワークを提案している[8]。さらに、筋シナジーを用いて論文で提示された 10 種類の動作に対して、筋電義手の分類精

* 放送大学

** 山口大学教育学部 〒 753-8513 山口市吉田 1677-1 kitamoto@yamaguchi-u.ac.jp

度 90%以上の動作識別率を報告している。この研究では、3D プリンタを使った、低コストの筋電義手の提案も行っている[9]。

研究[1]では、深層学習を用いてオフライン 2 ヶ所の部位（橈側手根屈筋と総指伸筋の部位）で握り、反り、屈曲、動作なしの 4 つの動作の分類を行い、高い識別率を実現した。ただし、リアルタイム動作識別ではオフラインとは異なり、時系列に沿ってデータが変化する。リアルタイム動作識別において推論値の出現パターンが異なることを研究[2]で報告し、この部分に課題があることを明らかにした。

本論文では、3 章で 2 つの研究事項について報告する。1 つ目は、以前のセンサと新しく導入したセンサの比較である（計測機器としてセンサの概要は 2 章でも述べる）。2 つ目は、推論時の信号の多様な出現パターンを人手による分類ではなく、さらにその信号に対して深層学習や機械学習を用い、2 段階の推論を行う手法を提案する。なお、文脈の都合上、動作識別とカーソル操作の用語を使い分けている場所もある。動作識別が正しければ、カーソルは指定された向きに正しく動くので、動作識別の正しさとカーソル操作の正確さはほぼ同じものと考えてもらいたい。さらに、「動作識別」と「推論」は同義で用いているが、動作の識別を強調する場合は「動作識別」を使っている。

2. 方法

2.1 実験参加者

実験には健常な 4 名が参加し、全ての計測を左腕・橈側手根屈筋の部位で行った。プログラム設計の検証や信頼性を確保するため、まずは予備実験として少人数による実験とした。また、筋肉の大きさや位置に個人差があるため、電極の貼り付け部位は、大体の位置を写真で示した。それをもとに、実験参加者にセンサパッドを貼り付けてもらった。測定は、腕を机の上に置き伸ばした状態で手の動作を行った。

なお、本研究は放送大学研究倫理委員会の倫理審査の承認を得たものである（通知番号 2023-73）。実験参加者は、事前に説明文書と同意書を読んだ上で自由意思のもと同意を得た上で実験を行った。

2.2 計測機器と使用コンピュータ、使用言語

最初に、筋電位の計測機器について説明する。センサとして MyoWare 筋電センサを研究[1][2]で用いたが、MyoWare 筋電センサが故障したこと、及び、センサから得られる信号を安定させるため、別の筋電センサを探した。その中で見つかった SS 社製のセンサを新たに使用することとした。

以前用いた MyoWare 筋電センサは、筋電センサの中でも安価（2022 年 8 月時点で 4,000 円前後）で、特にプロトタイピングに優れている。ただし、電子工作の知識が必要となる。今回用いた SS(スポーツセンシング)社製の DSP ワイヤレス筋電センサ（SS-EMGW-HM）は、筋電センサ全体の中では安い部類に入る。12 g と小型軽量でありながら 4 時間の連続動作を実現している。また、SDK を購入し仕様書に沿ってプログラムを作成すれば、ARV（Average Rectified Value：整流平滑化）を利用してリアルタイム動作識別を高精度に行うことができる。ARV については後ほど説明する。

さらに、実用化する場合、筋電センサの腕への取り付け違和感、つまり付け心地がハードルとなってくるが、SS 社製のセンサでは、小型軽量のためそれほど違和感を感じない。図 1 に示すようにセンサパッドを使い電極を取り付ける湿式と異なり、乾式はセンサ本体に電極がついており、取り付けは容易である。今回は、取り付けやすさは乾式が有利であるものの信号の安定度から湿式を使用している。

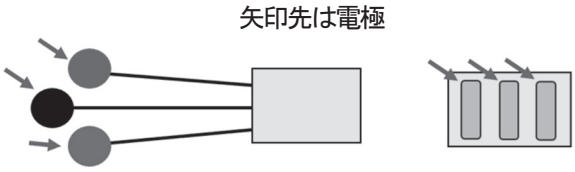

図 1 DSP ワイヤレス筋電センサ 左:湿式 右:乾式

2023 年 8 月から SS 社製センサを使用して実験を行っており、現時点（2024 年 7 月）まで、1 日のうち、どの時間に計測しても安定した信号が得られている。なお、センサの使用数は 1 つで、計測箇所は 1 ヶ所である。これは、システムの単純化や低コスト化を目指しているためである。以上で、計測機器について述べた。

次に、使用したコンピュータについて表 1 に示す。

表 1 実験コンピュータスペック

OS Windows 11 Pro 64bit
CPU Intel Core i7-8565U
(1.80GHz-4.60GHz 4 コア/8 スレッド)
RAM 16GB LPDDR3 2133MHz

このコンピュータは、研究[1][2]で使用したコンピュータと同じであり、CPU は 2024 年 7 月現在ではミドルレンジに分類される。

最後に、使用したプログラミング言語について述べる。以前の研究では Python3.9 を使用したが、作成したデータ収集プログラムで並行処理を行うとデータのロスが生じた。プログラムを修正したが、Python3.9 では解決でき

ず、Python3.11 に変更することで解決した。そこで、実験は主に Python3.11 を用いた。

2.3 動作識別の詳細

2.3.1 センサによる ARV データの取得

本研究では、ARV を用いて実験を行っている。なお、筋電位に対して絶対値をとり（整流化）、ローパスフィルターをかけたものが ARV、絶対値に対して積分したものが IEMG と区別される。しかし、波形の形状が似ているため研究によっては ARV と IEMG が厳密に区別されていないことがある[10]。研究[2]では MyoWare 筋電センサの取り扱い説明書の記載（Rectified & Integrated EMG Signal）から IEMG とした。研究[2]の IEMG と本研究で用いた ARV の動作識別精度は、ほぼ同様であり変化していない。

DSP ワイヤレス筋電センサで時刻 t における ARV は、以下の式で表される。

$$ARV(t) = \frac{1}{N} \sum_{k=t-N+1}^{t} |e_k| \qquad (1)$$

ここで、N = 100、 e は k における筋電位（単位は mV）である。また、 t は 1 から始まる自然数で、添字は以下も同様である。

2.3.2 手の動作と教師ラベル

動作識別は、手の動作を決め、機械学習や深層学習に必要な教師ラベルを設定する必要がある。今回では、動作は握り（Grasp）、反り（Extension）、動作なし（No hand motion）の 3 つとする。研究[2]で行った動作と全く同じであるため、図を引用して掲載する。

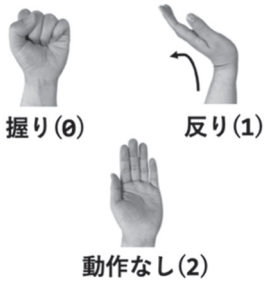

図 2 動作と教師ラベル

図 2 に示されているとおり、握りは 0、反りは 1、動作なしは 2 の教師ラベルとする。本論文では、この値を前提に話を進める。

2.3.3 データ取得と学習および推論

握り、反り、動作なしの 3 種類の動作からデータを取得する。分かりやすさのため、研究[2]の図 8 や図 9 を参照されたい。また、データ取得前や端点の問題はリアルタイム動作識別では考える必要がない。太字でベクトルや行列を表し、そうでなければスカラーである。

ここで、得られたデータを表す D 次元列ベクトルを、$\boldsymbol{z} = (z_1, z_2, \cdots, z_D)^T$ とする。T は転置を表す。$D = 4100$ とし、データを収集するプログラムによって、握りのデータ $\boldsymbol{x_1}$ 、反りのデータ $\boldsymbol{x_2}$ 、動作なしのデータ $\boldsymbol{x_3}$ を得る。

これらの各々のデータは基準を 2 標準偏差とし 10 個のデータ幅で Hampel フィルタにより処理される。

さらに、データから

$$I = \{\, i \mid 2 \le i \le D-1,\ z_i > z_{i-1} \wedge z_i > z_{i+1} \} \qquad (2)$$

によって、 $\boldsymbol{z}$ の全ての極大値のインデックス I が得られる。ただし、適切にインデックスを求めるには、スパイク状のノイズへの対処や極大値間の距離の近接を防ぐため、閾値や距離の適切な設定をする必要がある。

また、 $\boldsymbol{x_1}$、 $\boldsymbol{x_2}$、 $\boldsymbol{x_3}$ の極大値を与えるインデックスを含むベクトルを $\boldsymbol{r_1}$、 $\boldsymbol{r_2}$、 $\boldsymbol{r_3}$ とする。それぞれのベクトルは、得られる極大値数により次元数は変化するが、$\boldsymbol{r_1}$、 $\boldsymbol{r_2}$、 $\boldsymbol{r_3}$ の最大の次元数よりも使用される次元数が大きくなることはインデックス数選択の都合上ない。

筋電位データを結合し、動作データ行列

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x_1}^T \\ \boldsymbol{x_2}^T \\ \boldsymbol{x_3}^T \end{pmatrix}, \quad \boldsymbol{X} \in \boldsymbol{P}^{m \times D} \qquad (3)$$

となる。 $\boldsymbol{P} = [0, 8.3]$、$m = 3$、D = 4100 である。

しばらくはベクトルで処理しても問題はないが、プログラムでは学習に行列を用いるので、その形も示しておいた。

さて、ここで訓練データを作る必要がある。訓練データは、極大値を与えるインデックス r から、片側 d、両側 $2d$ だけ取得する。$d = 10$ として実験を行った。

握りの訓練データ、1 サンプルは、

$$\boldsymbol{x}_{\boldsymbol{train1\_1}} = (z_{r-d+1}, z_{r-d+2}, \cdots z_r, \cdots z_{r+d})^T \qquad (4)$$

となる。

これを 5 サンプル分集めた行列は、

$$\boldsymbol{X}_{\boldsymbol{train1}} = \begin{pmatrix} \boldsymbol{x}_{\boldsymbol{train1\_1}}^T \\ \boldsymbol{x}_{\boldsymbol{train1\_2}}^T \\ \vdots \\ \boldsymbol{x}_{\boldsymbol{train1\_5}}^T \end{pmatrix} \qquad (5)$$

となる。反り $\boldsymbol{X}_{\boldsymbol{train2}}$ と動作なし $\boldsymbol{X}_{\boldsymbol{train3}}$ も同様である。

これらを結合すると、

$$\boldsymbol{X_{train}} = \begin{pmatrix} \boldsymbol{X_{train1}} \\ \boldsymbol{X_{train2}} \\ \boldsymbol{X_{train3}} \end{pmatrix}, \quad \boldsymbol{X_{train}} \in \boldsymbol{P}^{\,p\times q} \qquad (6)$$

となる。p＝15、 q＝20 である。

$\boldsymbol{X_{train}}$を用い、深層学習で学習する。ハイパーパラメータ等、表 2 に示す。全結合層からなるニューラルネットワークである。

表 2 深層学習ハイパーパラメータ

隠れ層の数	2	epoch	2000
隠れ層のニューロン数	1200	バッチサイズ	8
活性化関数	隠れ層全て ReLU	損失関数	Categor-ical Cross Entropy
最適化手法	SGD (Stochastic Gradient Descent)	Dropout	隠れ層全て 0.5

使用者の負担を減らすため、短時間の事前学習で動作識別を行えるように設計してあり、データ取得の工程は D＝4100 の場合、2 分程度で終了する。深層学習による学習時間はデータ量やハイパーパラメータに依存するが、今の実験系では 1 分 30 秒程度である。3 分から 5 分程度でデータ取得、特徴抽出、学習は終了する。

k 番目の動作の確率は、

$$y_k(\boldsymbol{x_{realtime}}, \boldsymbol{w}) = \frac{exp(a_k(\boldsymbol{x_{realtime}}, \boldsymbol{w})}{\sum_j exp(a_j(\boldsymbol{x_{realtime}}, \boldsymbol{w}))} \qquad (7)$$

で表される。これはソフトマックス関数と呼ばれる。表 2 で与えられるニューラルネットワークの出力層の k 番目のニューロンへの入力を a_k とする。$\boldsymbol{w}$ はニューラルネットワークの重みを表す。$\boldsymbol{x_{realtime}}$ はリアルタイムで得られる ARV の 20 次元ベクトルである。

確率が最大のインデックスが選ばれ、

$$argmax(y_1, y_2, \ldots, y_l) \qquad (8)$$

で現在の動作が選ばれる。今回の実験では l ＝3 である。

3. 実験と新たなシステムの提案

3.1 センサ間の比較

実験で用いたMyoWare 筋電センサと DSP ワイヤレス筋電センサの主な違いを表 3 に示す。

表 3 筋電センサ間の比較

	MyoWare 筋電センサ	DSP ワイヤレス筋電センサ
信号の数値型	3 桁の int 型	float 型
無線機能	拡張すればあり	あり
信号の安定性	時変性あり	高い

表のとおり、センサ間で信号の数値型が異なる。研究[1][2]の MyoWare 筋電センサを用いた実験と今回の DSP ワイヤレス筋電センサ用いた 3 動作識別でのカーソル操作で数値型による動きの変化は見られなかった。

次に、無線機能は、MyoWare 筋電センサは自由度が高く、無線モジュールを自作すれば、目的に合わせた機器を作ることができる。一方、DSP ワイヤレス筋電センサは、スポーツでの使用を想定して、無線機能が備わっており、データ送受信機（SS-RF24TR1）で見通し 50m、ハイパワーデータ送受信機（SS-RF24TR2）で見通し 100m の通信ができる。総務省の平成 30 年 住宅・土地統計調査 住宅および世帯に関する基本集計[11]によると、総数では、1 住宅当たりの延べ面積は 92.06m^2 であるため、日本の平均的な家屋では部屋のどこにいても正常に通信することができる。これは、ベッドに寝た状態やコンピュータと離れた状態で通信ができることを意味する。

最後に、得られるセンサ値の時変性の問題であるが、筆者の環境では MyoWare 筋電センサは、起床時直後に測定すると信号は安定しているものの、時間が経つにつれ不安定になり正確な動作識別が不可能になった。一方、DSP ワイヤレス筋電センサは、一日のどの時間に測定しても安定した信号が得られるため、以前より多く実験ができ、素早く開発できる環境が得られた。

ただし、価格においては、安価なシステムを目指しているため、今後、考慮する必要がある。

3.2 リアルタイム動作識別とその信号

研究[2]で報告したとおり、リアルタイム動作識別では、動作を行った後の推論値、つまり動作識別の信号にバリエーションがある。次のページの図で、深層学習による動作識別とその特徴について述べる。

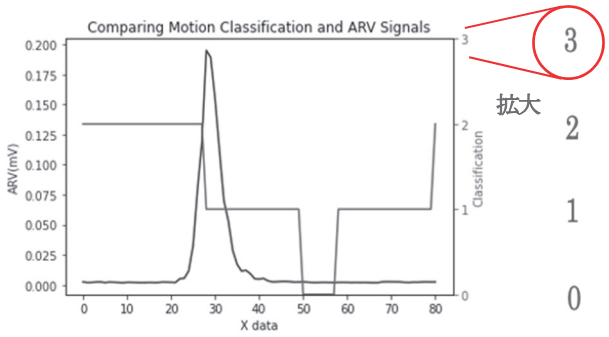

図 3 握ったときの ARV と動作識別信号

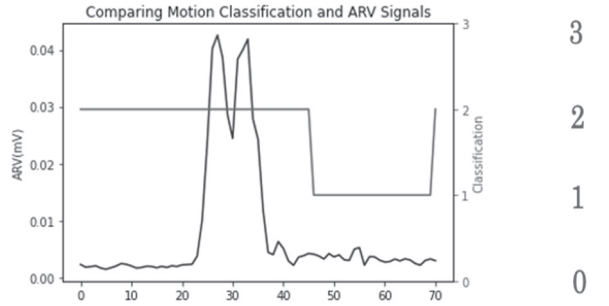

図 4 反ったときの ARV と動作識別信号

図 3 に、握ったときの ARV 信号（曲線）及び、どのように動作識別されたかを表す信号（直線）を示す。図 4 では、同様に、反ったときの ARV 信号（曲線）とどのように動作識別されたかを表す信号（直線）を示す。y方向の直線は信号の切り替わりを視覚的に分かりやすくするためで、動作識別値が連続的に出力されるわけではない。また、軸の数値が小さく分かりにくいため拡大した。グラフプロットプログラムが自動で、y軸の目盛りに 3 を出力しているが、3 動作識別で教師ラベルは 0、1、2 であるため 3 が出現することはない。

図 3 から、握ったときの動作識別信号は握りの信号 0 が、ただ 1 つ出力されるわけではなく、反りの信号 1 も含み、それらの信号はある程度持続することが分かる。動作なしの部分は 2 が出力されている。図 4 で、反ったときの信号は同様に、ある場所で反りの信号 1 が 1 つ出力されるわけではなく、1 の信号が単独で持続し出力されている。動作なしの部分は信号 2 が、図 3 同様に出力されている。このことから、握ったときの動作識別信号は、集合 $\{0, 1\}$ からなり、反ったときの信号は、$\{1\}$ からなることが分かる。ただし、プログラムの都合上、動作なしの 2 を含めている。つまり、握りの動作識別信号は $\{0, 1, 2\}$ であり、反りは $\{1, 2\}$ 、動作なしは $\{2\}$ である。

実際のプログラムでは、deque（両端キュー）に混在する信号が保存され、それが集合型へと変換される。

例えば、握りの混在する信号は、以下のようになる。図 3 と全く同一ではないが、動作識別信号の直線と見比べてほしい。なお、以下の信号は実際に実験で得られたものである。

deque([1, 1, 1, 0, 0, 0, 1, 1, 1, 2])

この混在した信号を、集合型に変換すると、

$$\{0, 1, 2\}$$

となる。

また、動作識別値が複数出現する理由として、動作時のある瞬間、波形のピークをもとに学習しているためだと考えられる。つまり、リアルタイム動作識別では、ある瞬間の値だけでなく信号の立ち上がりや減衰などの途中経過を含むため、複数の信号が出現すると考えられる。

ここで、先ほど挙げた 1 段階目の動作識別信号から、人手により結果の解釈を行うプログラムを掲載する。

Algorithm 1 人手による推論結果解釈

```
1: 入力: actionSignal
2: if 0 ∈ actionSignal and 1 ∈ actionSignal and 2 ∈ actionSignal then
3:     Print "握っています"
4: else if 1 ∈ actionSignal and 2 ∈ actionSignal then
5:     Print "反っています"
6: end if
```

actionSignal は、$\{0, 1, 2\}$ または $\{1, 2\}$ である。Algorithm1 では、そのことを素直に条件式に書いた。動作なしの信号 $\{2\}$ は、能動的に動作を行ったわけではないので、条件分岐に含めておらず、actionSignal にも含まれない。このプログラムでは、1 段階目の動作識別信号をもとに、2 段階目で筆者による手動、つまり人手で信号のパターンを場合分けしたプログラム（Algorithm1）で動作識別を行う。実際に、3 動作識別では、うまく動くものの動作数が増えると、1 段階目の信号が複雑になり人手による場合分けが困難になる可能性がある。

そこで、新たな手法である、1 段階目の推論で得られる信号をデータとして一時的に保存しておき、再度、深層学習やその他の機械学習を使って学習し推論すれば、複雑な場合分けを避けられる可能性がある。次節で、2 段階のシステムを提案する。今回は、予備実験から深層学習の安定性が高いことが分かっているので、1 段階目に深層学習を用いたものを述べる。

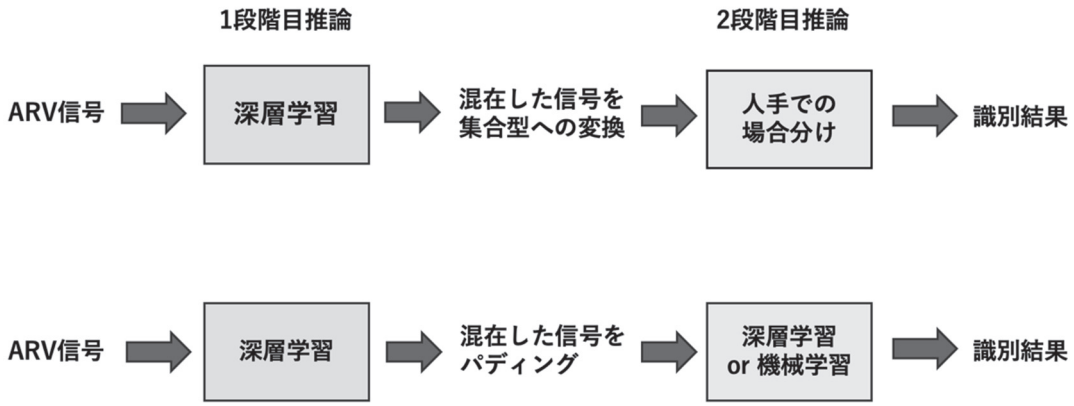

図5 2段階による推論 上:手動 下:自動

3.3 2段階システムの提案

ここで、2段階の手法を提案する。図5上は、1段階目の推論に深層学習を用い、2段階目は人手による場合分けを行っている。図5下では、1段階目の推論には同様に深層学習を用い、2段階目も深層学習あるいは機械学習で推論を行っている。

図5上では、混在した信号を集合型へと変換している点、下では混在した長さが違う信号をパディングしている点で異なる。パディングはするものの、図5下の手法では、2段階目の学習データとして、ほぼそのままの1段階目の推論値を入力している。図5下の方法は、すぐ後で述べる識別のための場合分けの簡単さだけでなく、システム自体の単純化も実現している。これは、データをそのままを入力し特徴を自動で捉えることができる深層学習に向いている。紙面の都合上、付録にAlgorithm2を掲載する。2段階での動作識別のアルゴリズムを擬似コードで示す。このコードではAlgorithm 2のdequeであるqに保存されたARVの値を用いて、13行目で1段階の推論、sに保存された1段階目の信号をパディングし、それをもとに27行目で2段階目の推論を行っている。Classification_sは推論結果である。

次に、1段階目の推論結果の解釈を行うAlgorithm3を掲載する。

Algorithm 3 2段階システム対応済みの推論結果解釈

```
1: 入力: Classification_s
2: if Classification_s = 0 then
3:     Print "握っています"
4: else if Classification_s = 1 then
5:     Print "反っています"
6: end if
```

Algorithm 1と比較すると、集合の要素を用いた場合分けは必要なく、単一の値からなる2段階目の推論結果、Classification_sだけで解釈が行われている。

3動作識別では、人手でも場合分けがそれほど難しくないため、あまり効力を感じられないが、4動作識別や5動作識別となった場合、決定境界で分離される領域は狭まり、1段階目の信号は人手による場合分けが困難になることが予想される。従って、このような2段階の手法は、筋電位の推論だけでなく、混在した信号を自動で場合分けする際に有用である。特にリアルタイムでの推論を扱う場面で活用が期待される。

3.4 2段階システムを用いた動作識別実験

今回、実験参加者4名の内3名で2段階システムを用いた動作識別が正確に行われるかを確認した。

まず、1段階目で出力される信号について共通していることを述べる。予備実験で分からなかったことであるが、1段階目の推論により出力される混在した信号には、個人差がある。この時点で、Algorithm 1が多様な信号に弱いことが分かる。また、提案した2段階のシステムを用い、1段階目の信号を学習し、2段階目の推論にかけたところ、個人間の異なる信号でもAlgorithm 3のロジックで正確に動作識別が行えた。つまり、2段階のシステムは、多様な信号で3動作識別が正確に行える。ただし、少人数での実験であるため、将来、規模を増やす必要がある。

次に、2段階目が深層学習のときに特有のことを述べる。実験参加者にカーソルの操作を行ってもらったところ、深層学習は多様な信号に対して頑健であるものの、遅延が発生する。

ただし、現在、NPU（Neural Processing Unit）に代表される深層学習や機械学習専用向けプロセッサが開発されており、今後、解消される可能性が高い。

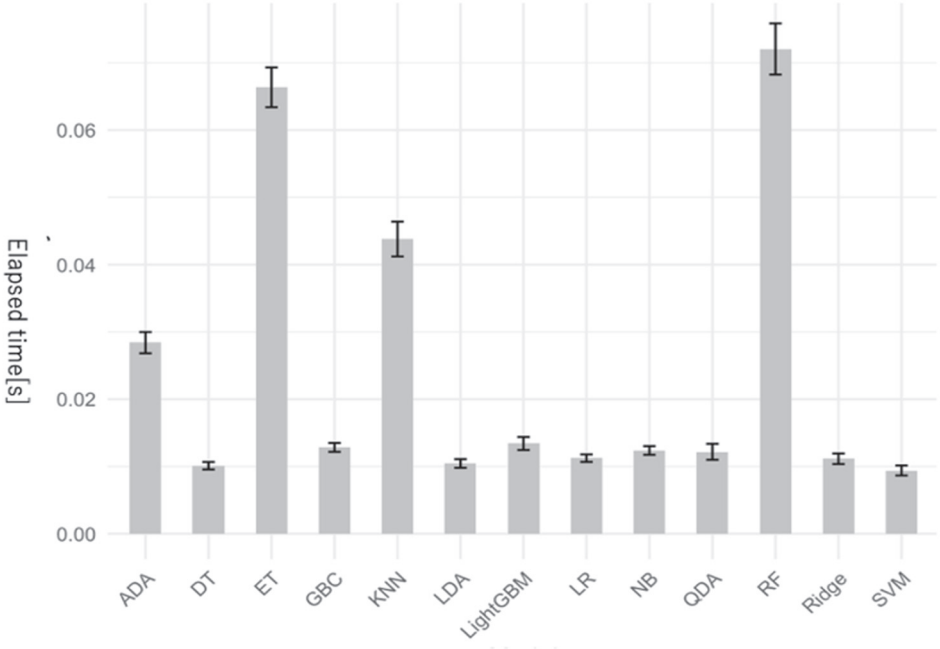

図 6 様々な機械学習による経過時間

表 4 機械学習手法一覧

ADA, Ada Boost Classifier	LR, Logistic Regression
DT, Decision Tree Classifier	NB, Naive Bayes
ET, Extra Trees Classifier	QDA, Quadratic Discriminant Analysis
GBC, Gradeint Boosting Classifier	RF, Random Forest Classifier
KNN, K Neighbors Classifier	Ridge, Ridge Classifier
LDA, Linear Discriminant Analysis	SVM, Support Vector Machine
LightGBM, Light Gradient Boosting Machine	

さらに、2 段階目に深層学習以外の機械学習を用いたことについて述べる。時間計測プログラムを用いて、1 段階目の推論直前からカーソル動作終了までの経過時間を測った。機械学習の種類による応答性の違いを示すため、図 6 に各手法の経過時間を示し、略記された機械学習手法の正式名称を表 4 に示した。また、エラーバーは標準偏差である。経過時間が短いモデルが必ずしも動作識別の精度が高いというわけではない。

今回、実験参加者に表 4 の機械学習手法からランダムに 1 つ選んでもらった。GBC（Gradeint Boosting Classifier）が選ばれ、2 段階目に用いて実験を行ったところ、多様な 1 段階目の信号に対して、遅延はあるものの正しく分類することができた。

これらの遅延も、先程同様、専用プロセッサの開発や CPU に最適化されたモジュールにより解消される可能性が高い。また、GBC 以外の動作精度についても、今後詳しく調べる必要がある。

4. 考察と将来の展望

今回、SS 社製のセンサである DSP ワイヤレスセンサを使用し、使用機器の構成を変え、安定した実験環境を構築した。安定したデータの取得は多人数の実験に向けて必須である。

次に、2 段階の動作識別手法について提案した。2 段階の動作識別手法は、1 段階目の推論結果を 2 段階目でさらに推論をかけることにより、人手による場合分けの困難さを解消し、個人間の多様な信号を自動で識別している。手の動作識別において、さらなる動作を増やした場合の問題解決策となる。また、遅延の問題は、深層学習や機械学習の専用プロセッサの最適化で、時間とともに解決される可能性も述べた。

将来の展望として、本研究ではコンピュータ操作に不慣れな高齢者や障がい者の補助ツール作成や筋電義手へ応用を目指している。カーソル操作は、動作識別を確実に行えたならば、実用上問題なく行える。ただし、現状では

システムに限界があり、動作間のデータの類似度の高さにより、誤識別することがある。特に弱い握りと反りが誤って判定をされることがある。また、動作の個人間の特徴を完全に捉えきれていない。他にも、学習時とリアルタイム動作識別時の両方である程度、動作の仕方について操作者にアドバイスをする必要がある。つまり、システムを人間近づけるか、人間をシステムに近づけるかという問題がある。これらの課題もフォローしていきたい。

最後に、機能的な側面だけでなく、センサの付け心地やデザインも大きな要素であるため、さらなる研究で解決が待たれる。

参考文献

[1] 伊藤正剛, 北本卓也 (2023)「筋電位を利用した深層学習による手の動作識別について」 山口大学教育学部研究論叢 第 72 巻 227-235

[2] 伊藤正剛, 北本卓也 (2024)「様々な機械学習を用いた手の動作識別について」 山口大学教育学部研究論 叢 第 73 巻 91-100

[3] Joseph DelPreto, Andres F.Salazar-Gomez, Stephanie Gil, Ramin Hasani, Frank H.Guenther, Daniela Rus, “Plug-and-Play supervisory control using muscle and brain signals for real-time gesture and error detection”, Autonomous Robots (2020) 44: 1303-1322

[4] 横浜国立大学 研究者ナビ 理工学部 機械・材料・海洋系学科専攻 准教授 加藤 龍
https://yokokoku-kenkyusya-navi.ynu.ac.jp/kato_ryu/
（2024 年 8 月 20 日アクセス）

[5] 広島大学 古居研究室 表面筋電位の確率的生成モデルと信号解析のページ
https://home.hiroshima-u.ac.jp/furui/project/emg-mod
（2024 年 8 月 20 日アクセス）

[6] 吉川雅博, 三河正彦, 田中和世「筋電位を利用したサポートベクターマシンによる手のリアルタイム動作識別」電子情報通信学会論文誌 D, vol.J92-D, no.1,pp.93-103, 2009.

[7] 山野井祐介 (2018 年度)『筋電義手制御のための信号特徴の時変性を考慮したビッグデータを用いた手指動作推定法』横浜国立大学大学院工学府システム統合工学専攻機械システム工学コース博士論文

[8] T.Tsuji, O.Fukuda, H.Ichinobe and M.Kaneko, “A Log-Linearized Gaussian Mixture Network and Its Application to EEG Pattern Classification”, IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, Vol. 29, No. 1, pp.60-72, February, 1999.

[9] Akira Furui, Shintaro Eto, Kosuke Nakagaki, Kyohei Shimada, and Toshio Tsuji, “A myoelectric prosthetic hand with muscle synergy–based motion determination and impedance model–based biomimetic control”, Sci. Robot 4, eaaw6339 (2019)

[10] 増田正 (2001)「〜口伝　表面筋電図篇〜 ただいま処理中の巻」 バイオメカニズム学会誌 Vol.25 No.(2001)

[11] 総務省統計局 平成 30 年住宅・土地統計調査 調査結果
https://www.stat.go.jp/data/jyutaku/2018/tyousake.htm
（2024 年 8 月 20 日アクセス）

付録

Algorithm 2 2 段階の推論

空の両端キュー q を初期化
最大長 2 の両端キュー r を初期化
空の両端キュー s を初期化
c を 0 に設定
$flag$ を 0 に設定
$Classification\_s$ を 0 に設定
while True **do**
    $ARV \leftarrow$ センサーからの ARV 値
    ARV を q の右端に追加
    if $c \geq 2\times$ 片側データ数 **then**
        q の左端要素を削除
        if c が適当な整数で割り切れる **then**
            $y\_3movements\_pred \leftarrow 1$ 段階目のモデルを使用して q で推論
            $y\_3movements\_pred$ を r の右端に追加
            if r の長さ ≥ 2 **then**
                if r の左端要素 $= 2$ かつ r の右端要素 $\neq 2$ **then**
                    $flag$ を 1 に設定
                    $y\_3movements\_pred$ を s の右端に追加
                end if
                if $flag = 1$ **then**
                    $y\_3movements\_pred$ を s の右端に追加
                end if
                if r の左端要素 $\neq 2$ かつ r の右端要素 $= 2$ **then**
                    $flag$ を 0 に設定
                    i をインクリメント
                    $padded\_s \leftarrow s$ の右端をパディング
                    $Classification\_s \leftarrow 2$ 段階目のモデルを使用して $padded\_s$ で推論
                    Improved_Classification 関数を呼び出し
                    s をクリア
                end if
            end if
        end if
    end if
end while