

画像処理による弾性包帯の圧迫圧推定に関する研究

Study on estimating compression pressure of elastic bandages using image processing

○戸嶋貴紀*, 高階淳子*, 南斉俊佑*, 星崎みどり*, 関健史*, 長縄明大*

○Takanori Toshima*, Junko Takagai*, Shunsuke Nansai*
Midori Hoshizaki*, Takeshi Seki*, Akihiro Naganawa*

*秋田大学

*Akita University

キーワード： リンパ浮腫 (Lymphedema), 弾性包帯 (Elastic bandage),
画像処理 (Image processing), 機械学習 (Machine learning)

連絡先： 〒 010-8502 秋田市手形学園町 1-1 秋田大学大学院理工学研究科
長縄 明大, Tel.: (018)-889-2726 E-mail: naganawa@gipc.akita-u.ac.jp

1. 緒言

リンパ系の機能不全によって引き起こされる症状である「リンパ浮腫」は、体液がリンパ管内に蓄積し、四肢や体幹部にむくみが現れることが特徴である。多くの場合、痛みはなく、感覚の鈍さや倦怠感を感じられる。一度発症すると完治は難しいことが多く、放置すると慢性化し、皮膚の変性や関節の機能障害、リンパ系の感染が発生する可能性がある。これにより日常生活に大きな影響を及ぼすことになる。日本リンパ浮腫学会によると、リンパ浮腫の治療を受けている患者はおおよそ 14 万人とされているが、未診断のケースも多いため、患者数はさらに多くなると考えられる¹⁾。

リンパ浮腫の治療法として、様々な治療法を組み合わせる「複合的理学療法」が最良の方法として挙げられており、中でも弾性包帯を用い

て患部を圧迫する圧迫療法が効果的な治療法とされている。しかし、現状ではリンパ浮腫セラピストから包帯の巻き方について指導を受ける機会が、数か月に一度の通院時となるケースが多く、時間とともに治療効果が薄れる点が問題点として挙げられている。

そこで、著者らは、看護手技をデジタル化する方法として、弾性包帯の圧迫圧を推定する方法について検討している。その目的は、弾性包帯を巻く際の圧迫圧をリアルタイムに患者へフィードバックすることにより、患者が自らリンパ浮腫セラピストと同様に、患部に対して適切な圧迫圧を与え、効果的な治療を実現するためである。

本研究では、弾性包帯に印刷された図形の形状変化をカメラで捉え、画像処理技術を用いて圧迫圧を推定する方法について検討したため、その成果を報告する。

2. 画像処理の方法

2.1 弾性包帯

弾性包帯には、ミドルストレッチ包帯である「BIFLEX 16+」(幅 8 cm, 長さ 3 m)を用いた。BIFLEX 16+には、包帯の表面に長方形(長辺 3.2 cm × 短辺 2.5 cm)と圧迫圧指標ゲージが印刷されており、これらに合わせて包帯を伸縮させることで、目視による圧迫圧の調整が可能となる。なお、BIFLEX の指標では、包帯を 30% 伸ばし、長方形の短辺 2.5 cm を長辺の長さ 3.2 cm に合わせて正方形にすることで、適切な圧迫圧で包帯を巻くことができるとしている²⁾。

本研究では、BIFLEX の図形変化をカメラ(USB C270n HD ウェブカメラ, logicool 製)で撮影し、その結果を画像処理することにより、適切な圧迫圧を実現できているかどうかの定量化を行う。

2.2 処理の流れ

Fig. 1 に本研究で検討している処理の流れを示す。左側の学習モジュールでは、包帯の図形検出に必要なデータを収集し、AI を活用した物体検出アルゴリズムである「YOLO」(You Only Look Once)を用いてモデルの構築および学習を行う。

一方、右側の検出モジュールでは、圧迫療法の様子をカメラで撮影してリアルタイムでデー

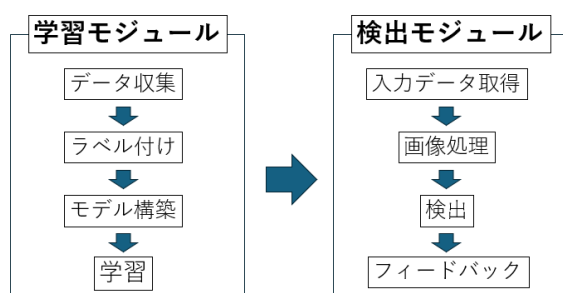


Fig. 1 処理の流れ

タを取得し、画像処理を行う。図形検出の結果に基づいて弾性包帯の圧迫圧が適切であるかを判断し、フィードバックを与えることで、使用者は正確に包帯を巻くことが可能になる。

3. 学習モジュール

3.1 学習・検証用データの作成

学習用データは、弾性包帯 BIFLEX16+に印刷されている長方形をカメラで画像を撮影した。なお、学習には、包帯に力をかけていない状態の画像と、包帯表面の長方形が正方形になっている状態の画像がそれぞれ必要である。

そこで、Fig. 3 に示す引張試験機を用い、22 cm に切断した弾性包帯を 6.6 cm 伸ばすことで、全ての辺がそれぞれ 3.2 cm となる正方形を作成し、30%の伸張状態を再現した。

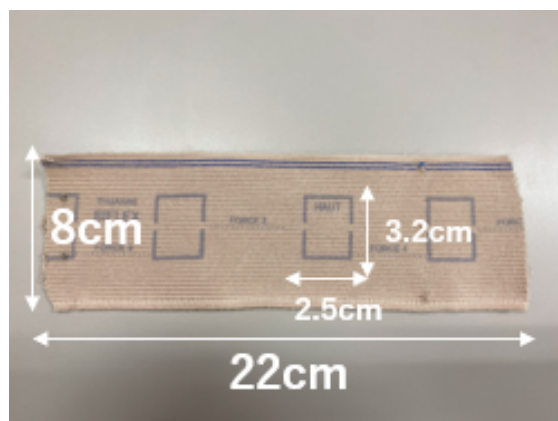


Fig. 2 使用した BIFLEX の様子



Fig. 3 引張試験機

撮影した画像について、図形が3つずつ写るようにし、包帯上の図形が長方形と正方形の状態でそれぞれ50枚ずつ撮影し、合計100枚の包帯画像を用意した。この中から、70枚の画像を学習用データとして、また、30枚の画像を検証用データとして扱う。

撮影したすべての画像について、画像アノテーションソフトウェアである LabelImg を用いて、用意した100枚の画像の包帯上の図形に対して、矩形のアノテーションを行った (Fig. 4)。なお、通常状態の長方形画像の場合には「rectangle」という名前でアノテーションを行い、正方形画像の場合には「square」という名前でアノテーションを行い、データを作成した。

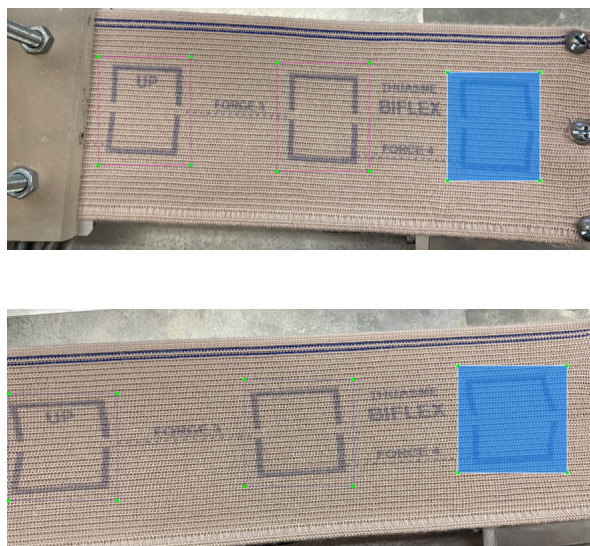


Fig. 4 LabelImg でのアノテーションの様子 (上図: rectangle 下図: square)

3.2 データ拡張

機械学習による画像認識では、大量の学習データが必要である。今回用意した100枚の画像ではデータ量が少ないと考え、データ拡張 (Data Augmentation) の手法を用意して、学習データの多様性を増やすことにした。

用意した100枚の画像に対して、反転画像と回転画像を作成した。反転画像については、画

像を X 軸・Y 軸・XY 軸中心にそれぞれ反転させ、回転画像については、画像を 90 度・180 度・270 度にそれぞれ回転した画像を作成した。反転画像と回転画像ともに、画像枚数を4倍ずつに増やし、全部で1600枚の画像データを用意し学習を行った。

3.3 YOLOv8 のモデル作成

前節で述べた学習用画像は、YOLOv8 を用いてディープラーニングで学習を行い、モデルを作成した。YOLOv8 の各モデルは、開発者が公開する github から入手し、利用した³⁾。学習は、GPU (NVIDIA GeForce GTX 1660 SUPER) を搭載した PC を利用し、Python 3.8.10 及び PyTorch 2.0.0 を用いて実行した。GPU のメモリ容量の制限のため、学習時の batch size を2とし、epoch 数を50に設定し、学習を行った。得られた結果を Fig. 5, Fig. 6, Fig. 7 に示す。

Fig. 5 に学習用データにおける損失の推移を表す。図より、ボックスの位置予測、クラス分類、分布焦点の3つの分野でともに損失を1以下に抑えることができていることから、精度は高く、学習が成功しているといえる。

Fig. 6 に検証用データにおける損失の推移を示す。図より、損失が1を大きく超える数値になっており、学習用データに比べて精度はあまり高くないといえる。原因としては、学習用に比べてデータが少ない点やモデルが複雑になってしまった点が挙げられる。

Fig. 7 にモデルパフォーマンス指標の推移を示す。図より、精度、再現率、平均適合率の3つの分野でそれぞれ値を1に近づけることができており、この点においても学習が成功し精度を向上させることができた。

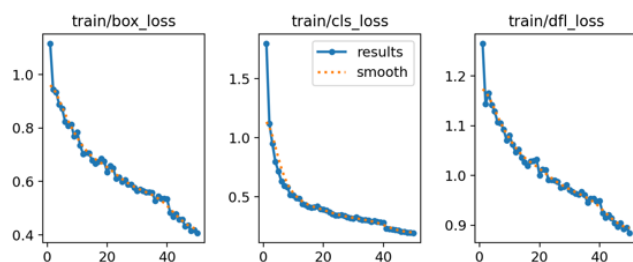


Fig. 5 学習用データにおける損失の推移
(左から、ボックス予測, クラス分類, 分布焦点)

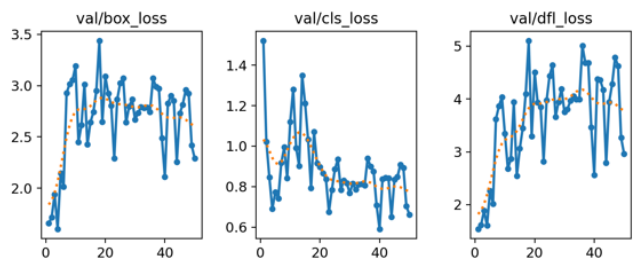


Fig. 6 検証用データにおける損失の推移
(左から、ボックス予測, クラス分類, 分布焦点)

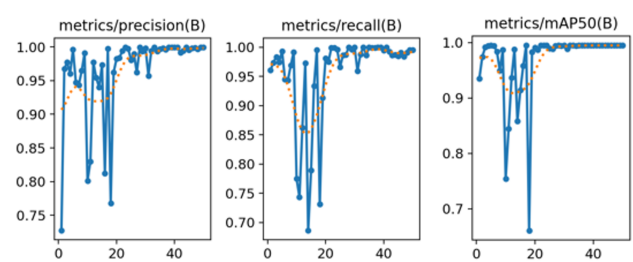


Fig. 7 モデルパフォーマンス指標の推移
(左から、精度, 再現率, 平均適合率)

4. 検出モジュール

4.1 画像での検出

作成した学習モデルを使用して、新たに用意した 10 枚の包帯画像で物体検出を行った。その際、検出信頼度の閾値 (cof 値) を 0.7 に設定し、信頼度が 0.7 未満の場合は検出を無視するように設定した。これにより、ノイズや誤検出を最小限に抑えた。

検出結果を Fig. 8 に示す。rectangle と square とともに誤検出がみられず、どちらの図形においても検出信頼度も 0.9 付近を記録しており、高い精度で検出を行うことができている。

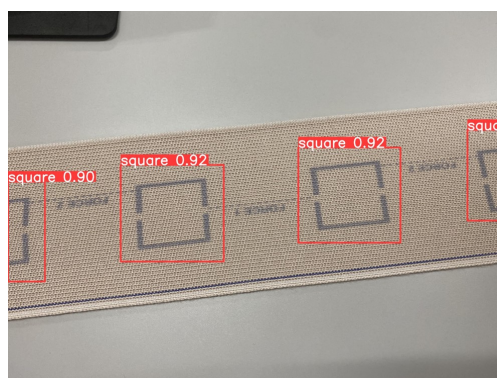
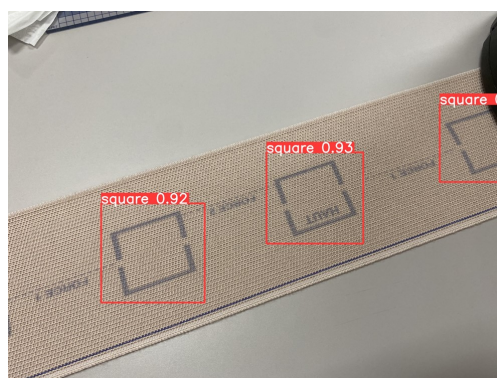
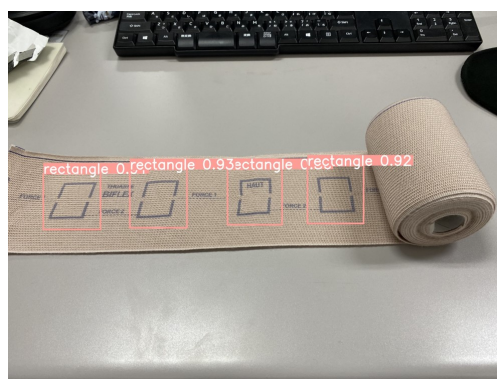
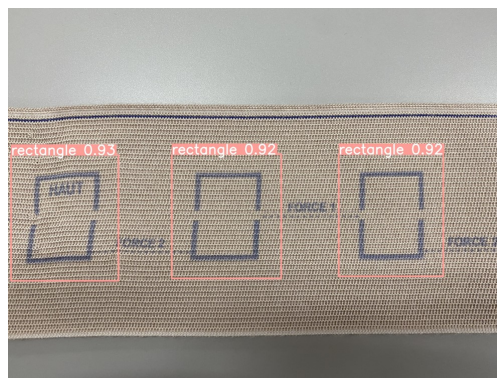


Fig. 8 画像検出の様子

4.2 リアルタイムでの検出

同じ学習モデルを使用し、リアルタイムでの検出も行った。PCに接続したWebカメラで撮影した映像をリアルタイムで処理し、実際の動作環境での性能を評価した。リアルタイム検出においても、前述の閾値設定($\text{cof}=0.7$)を維持し、誤検出を防止した。

検出結果を Fig.9 に示す。リアルタイムでの検出においても、rectangle 及び square どちらの図形に対しても検出信頼度が安定して 0.9 付近を記録しており、リアルタイム処理でもモデルの精度が保たれていることがわかる。また、映像のフレームレートに影響を与えることなく、スムーズに物体検出を行うことができた。

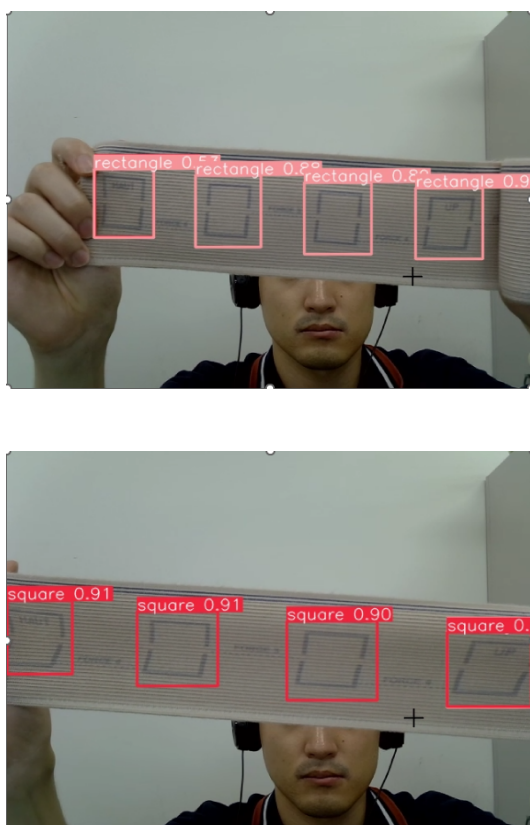


Fig. 9 リアルタイム検出の様子
(上図：rectangle 下図：square)

5. 結言

本研究では、リンパ浮腫治療で使われる弾性包帯により患部に適切な圧迫圧を与える方法として、画像処理による包帯表面上の図形の変形を活用した方法について検討した。その結果、通常状態の図形と 30%の力をかけた状態の図形ともに高い精度で検出することができた。今後は、実際に包帯を巻く場面において実験し、本手法の有効性について検証する予定である。

参考文献

- 1) 重松宏：我が国のリンパ浮腫の現状とリンパ浮腫療法士の育成，そして日本リンパ浮腫治療学会の設立，日本皮膚科学会雑誌 129 巻 9 号，pp.1871-1876(2019)
- 2) THUASNE, ”リンパ浮腫ケア総合カタログ”，https://solve-net.com/files/thu_catalogjp_2018s.pdf, (参照 2023-12-04)
- 3) lakshanthad, glenn-jocher, ”Ultralytics”, <https://github.com/ultralytics/ultralytics>, (参照 2024-05-22)
- 4) 花岡創, 武津英太郎：トドマツの樹冠画像からの球果検出に基づく物体検出アルゴリズム YOLO のモデル比較, Bulletin of FFPRI, Vol.21, No.4, pp.267-274(2023)
- 5) T.Nukui, ”YOLOv8 でオリジナルデータの物体検出をする”, <https://farml1.com/yolov8/>, (参照 2024-05-22)