

オクルージョンを考慮した深層距離学習による組立作業の進捗段階の推定

○橘川拓実^{*}, 湯本茂樹^{*}, Sarthak Pathak[†], Alessandro Moro[‡], 梅田 和昇[†]

○ Takumi KITSUKAWA^{*}, Shigeki YUMOTO^{*}, Sarthak PATHAK[†],
Alessandro MORO[‡], and Kazunori UMEDA[†]

^{*}: 中央大学大学院理工学研究科, kitsukawa@sensor.mech.chuo-u.ac.jp

[†]: 中央大学理工学部 [‡]: RITECS Inc.

＜要約＞本研究ではオクルージョンを考慮した工場での製品の組立作業の可視化を目的とし、深層距離学習を用いた進捗判定手法を提案する。まず設置された定点カメラから取得した画像に対し、深層学習を用いた物体検出手法で対象の組立製品を検出する。その部分の切り出し画像に対して深層距離学習（Deep Metric Learning）を用いた提案手法により進捗ステップを判定する。

＜キーワード＞深層距離学習, 進捗推定, オクルージョン, 異常検知, 組立作業

1 序論

近年、製造業における人手不足は深刻度を増しており、現場の作業効率を上げて生産性を向上させることが求められている。解決方法としてIoTやロボットを利用した製造ラインの自動化が進められている [1][2]。しかし、顧客のニーズに合わせて製品を生産する多品種少量生産型の工場では、自動化が進んでおらず、人手による作業が大半を占めている。特に製品の組立作業は、商品の仕様が変わるたびに生産ラインを変更する必要があるため、人手による作業が行われている場合が多く、作業の進捗管理を紙などのアナログな方法で行っているという問題がある。これに対し、工場に設置した定点カメラから映像を取得し作業の進捗状況を推定することで、自動で管理するシステムの導入が有効であると考えられる。

これまでに、人物の骨格点情報を時系列的に利用し動作認識する研究 [3] や作業員視線で手の動きを見ることが進捗推定する研究 [4] が行われている。しかし大型装置の組立は、作業が数日かかるものや組立作業における一つ一つの動作が細かく決まっていないものが多く、作業員の動きや時系列データを利用した進捗推定が適していない。また、隣のステップとの間違いを考慮した手法 [5] も提案されているが、工場内では作業員や障害物によるオクルージョンにより誤判定が起きる

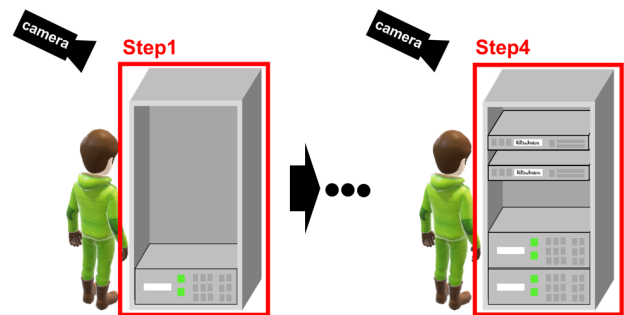


図 1 提案手法の概要図

という問題がある。

そこで本研究では、組立てる対象物に着目しオクルージョンを考慮した組立作業の進捗推定システムを構築することを目的とする。進捗推定は、事前に設定した組立進捗ステップに対し、どのステップまで作業が進んでいるのかを判定する (図 1 参照)。

2 距離学習の利用

設定したステップに対し現在どのステップに属するかを判定するシステムの構築としては、クラス分類問題として問題設定することができると考えられる。画像におけるクラス分類は、CNN (Convolutional Neural Network) を用いた深層学習手法が高い精度を誇っている。しかし今回の組立進捗推定におけるクラス分類は、クラス同士の見た目の違いが小さいことが考えられ、細かな特徴の違いを抽出することが求められる。そこで機

機械学習手法の一つである Metric Learning (距離学習)[6]の利用を検討する。

Metric Learning は、サンプル間の距離または類似度に基づいてクラスごとに分離されるように、クラス間距離が出来るだけ長くなるような特徴量とその空間を学習させる手法である。Metric Learning の目的は、同じクラスのサンプル間の距離を小さくしながら異なるクラスのサンプル間の距離を大きくすることである。特にパターンの学習の際に多層構造のニューラルネットワークを用いて特徴抽出をする Deep Metric Learning (深層距離学習)[7] も提案されている。

一般的なクラス分類は、CNN で抽出した特徴量を全結合層に通し、それをソフトマックス関数でクラス所属確率に変換する。しかしこれだと同クラスと他クラスのサンプル間の距離を考慮せずに特徴量抽出ネットワークを学習させることになる。一方 Deep Metric Learning は、意図的に他クラスのサンプル間を大きくし、同クラスのサンプル間の距離を小さくするように学習することで、他クラスサンプルとの特徴の違いを大きく表現できる特徴量を得ることができる。そのため各クラスのサンプルが少ないときや未知クラスがあるときに性能を発揮する。組立作業の進捗判定を実際の工場で実用化することを考えると、少ないデータでの学習が求められる。また、未知クラス判定により誤判定を抑えられる。これらを踏まえ、本研究では Deep Metric Learning を利用した進捗推定手法を提案する。

3 提案手法

3.1 提案手法の概要

提案手法の概要を図2に示す。事前の準備として、画像内での対象物の位置を検出できるようにするために、カスタムデータセットによる物体検出モデルの学習を行う。また、切り出し画像の進捗推定を行うステップ判定モデルを学習させる。システムの流れとしては、定点カメラから取得した画像に対し物体検出手法で対象物を検出してその位置を切り出し、Deep Metric Learning によるステップ判定モデルで進捗を推定する。物体検出手法には、高速でありながら正確な検出が可能な Faster R-CNN[8] の学習済みモデルをファインチューニングして利用する。

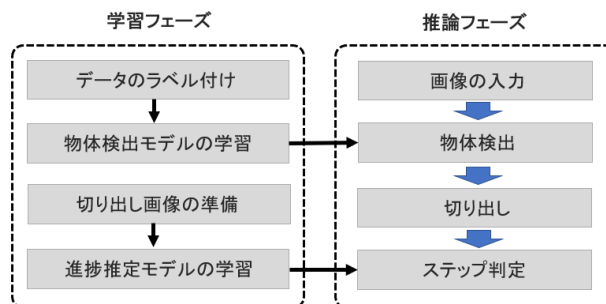


図2 提案手法の処理の流れ

3.2 進捗推定モデル

提案する進捗推定モデルは大きく学習フェーズと推論フェーズに分かれる。

まずは学習フェーズについて説明する。モデル構造を図3に示す。初めに組立進捗の中で判定したいステップを設定して学習データを用意する。方法は、組立動画から対象物部分の画像を切り出し、ステップごとにフォルダを分けて保存する方法である。学習モデルの処理の流れとしては、まず初めに用意した学習データの中からランダムに画像を一枚選び、anchor サンプルを設定する。この anchor サンプルと同ステップの画像を一枚選んだものを positive サンプル、他ステップの画像を一枚選んだものを negative サンプルとする。次にこれらの設定した三枚の画像をそれぞれ4層のCNN層に入力、さらに1層の全結合層に入力することで128次元の特徴ベクトルを取得する。この時CNNモデルの重みは共有する。そして得られた三つの特徴ベクトルのうち、anchor サンプルと positive サンプルの距離を近づけ、anchor サンプルと negative サンプルの距離を遠ざけるように損失関数を定義することで、パラメータの更新を行う。損失関数には anchor サンプル、positive サ

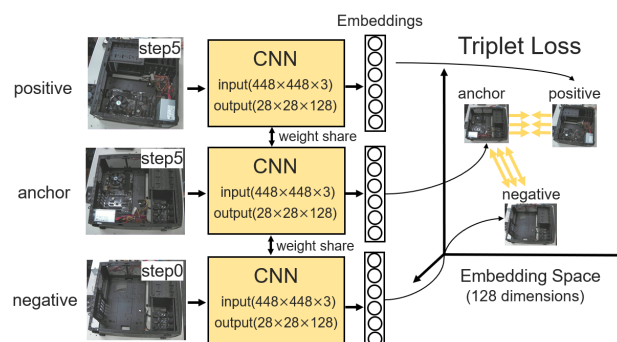


図3 学習フェーズのモデル構造

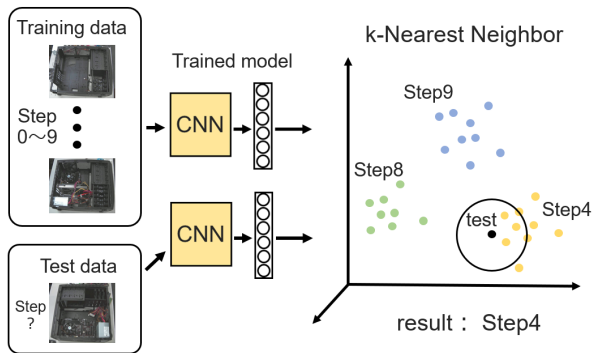


図 4 推論フェーズのモデル構造

ンプル, negative サンプルの三枚を一組のデータとして相対的距離を算出する Triplet Loss[9] を用いる. 式を (1) に示す.

$$L_{Triplet} = \max(d_p - d_n + m, 0) \quad (1)$$

d_p, d_n は, 特徴量空間における positive サンプル, negative サンプルと anchor サンプルのそれぞれの距離である. m は, 距離を遠ざける・近づける作業の動かす程度を表す定数であり, マージンと呼ぶ. 距離の計算にはユークリッド距離を利用する.

次に, 推論フェーズについて説明する. モデル構造を図 4 に示す. ここでは学習した CNN モデルを利用してステップ判定を行う. まず学習で利用した各ステップの画像をもう一度一枚ずつ学習済みモデルに入力し, 特徴量空間に埋め込む. すべての画像を特徴量ベクトルに変換したら, 推定したい入力画像を同じく学習済みモデルに入力し, 特徴量空間に埋め込む. これらの埋め込まれたデータに対し k 近傍法 (k-Nearest Neighbor algorithm)[10] を利用することで未知のテストデータがどのステップに属するかを推定する. また, 作業ミスがある画像やオクルージョン等で製品の一部分が確認しづらい画像は, 学習データに含まれていないため, 特徴量空間内でどのサンプル群からも離れた位置に配置されると考えられる. そこで k 近傍法による判定の際, 近傍 k 点との距離の平均がしきい値より大きいものをエラーと判定する. これを行うことで, 組立作業における組立ミスの検知やオクルージョンで生じる誤判定の削減をすることができる.

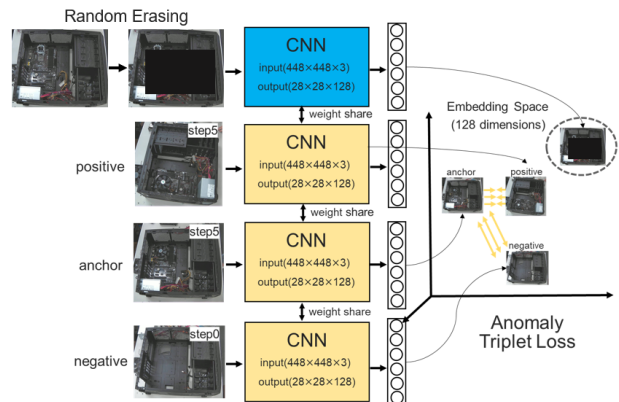


図 5 オクルージョンを考慮したモデルの構造



図 6 Random Erasing によってノイズをかけた画像

3.3 オクルージョンを考慮した学習モデル

実際の組立作業での利用を考えると作業員などの様々な障害物によるオクルージョンが発生することが考えられる. そこで, オクルージョンによって正しく推定できない画像を, 学習データに異常データを含めずに異常画像と判定できる学習モデルを提案する.

提案する学習モデルを図 5 に示す. 3.2 節で提案した手法に加えて, 正常な組立ステップにノイズを加えた画像を anomaly サンプルとして学習データに加えている. ノイズのかけ方としては, 画像の一部短冊領域をランダム画素値でマスクする Random Erasing を利用する. 実際に Random Erasing によりノイズをかけた画像を図 6 に示す. さまざまなオクルージョンに対応できるように, 各画像に対して Random Erasing をそれぞれ二回行っている. そして anchor, positive, negative の 3 種類のサンプルと同様に CNN 層に入力し 128 次元の特徴量データに変換する. 得られた, すなわち 4 つのデータの特徴量空間内での位置関係を学習させることで異常画像を考慮に入れた推定ができるようになる. 特徴量空間での位置関係を図 7 に示す. 提案する損失関数を (2) に示す.

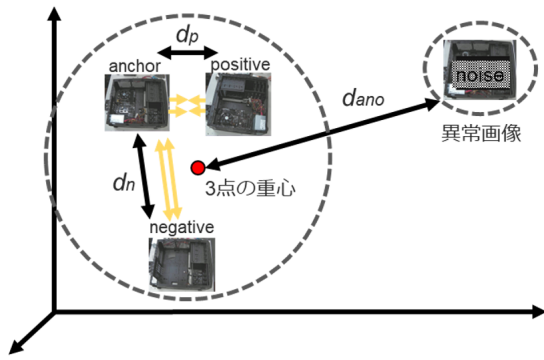


図 7 特徴量空間における各サンプルの位置関係

$$L_{TripletAno} = \max(d_p - d_n + m_\alpha, 0) + \lambda \cdot \max(d_n - d_{ano} + m_\beta, 0) \quad (2)$$

m_α, m_β はそれぞれの項のマージンを示しており, d_{ano} は anchor, positive, negative サンプルの 3 点の重心と anomaly サンプルの距離である. λ は d_p, d_n の項と d_n, d_{ano} の項の重みを調整する係数である.

4 実験

4.1 実験環境

進捗推定モデルの有効性を確認するために実験を行った. 組立製品の対象としてデスクトップ PC を使用した. カメラはアイ・オー・データ機器社のネットワークカメラ Qwatch の TSWR-LP[11] を利用し, 画素値は 1980×1080 に設定した. 実験環境を図 8 に示す. 判定するステップは, ステップ 1 から 8 の 8 つに設定した. ステップの分け方は図 9 に示す.

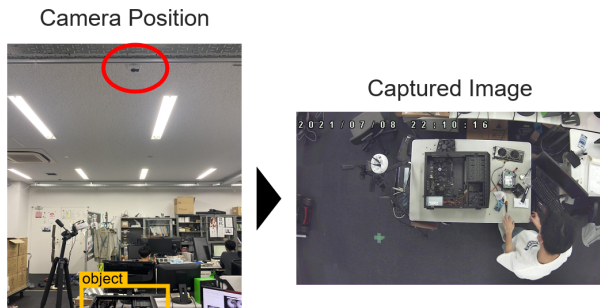


図 8 カメラ位置と取得画像

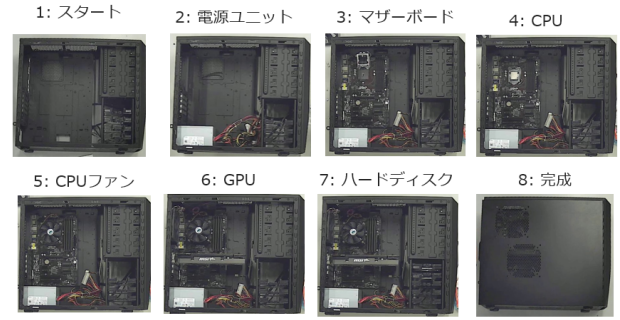


図 9 進捗ステップの分け方

4.2 提案手法の実装と学習

学習データは各ステップ 40 枚の計 320 枚, テストデータは異常画像を含めて計 1600 枚, k 近傍法は $k=4$, 学習は 100epoch で行った. また学習データが少量であるためデータに偏りが生まれないように, 入力画像に対して回転, 射影変換, 色彩変化, 一部切り落としをそれぞれ 70% の確率でランダムに加えることでデータを拡張した. 実際に行ったデータ拡張を図 10 に示す. また損失関数の中の項の重みを調節する変数 λ は 50epoch までは 0 で 50epoch から 100epoch にかけて徐々に大きくなるように設定した.

実装後の評価として, 提案モデルを 100epoch 学習させた際の正解率の推移を図 11 に示す. また, 特徴量空間内での anchor サンプルと positive サンプルの距離, anchor サンプルと negative サンプルの距離, anchor, positive, negative 3 点の重心と anomaly サンプルとの距離それぞれの学習回数における推移を図 12 に示す.

図 11 の結果から, 学習データ, 評価データ共に学習回数が増えるにつれて正解率が上がっていることが分かる. また 100epoch 付近では評価データでも 80% 以上の正解率を達成することができ, うまく学習ができていることが確認できた.

図 12 の結果から, d_n は学習回数が増えるにつれて d_p と比較して距離が大きくなっていることが分かる. また d_{ano} は, 損失関数の項の重みを持ち始める 50epoch 以降で d_p と d_n どちらと比較しても距離が大きくなっていることが分かる. これらの結果から, 提案したモデルと損失関数により特徴量空間における異常データの学習が正しくできたと言える.

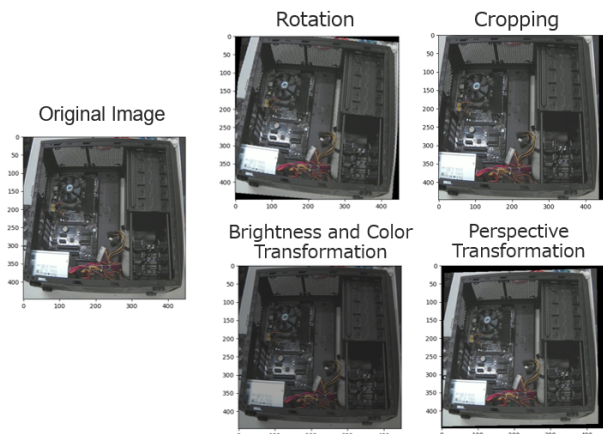


図 10 データ拡張のための画像処理

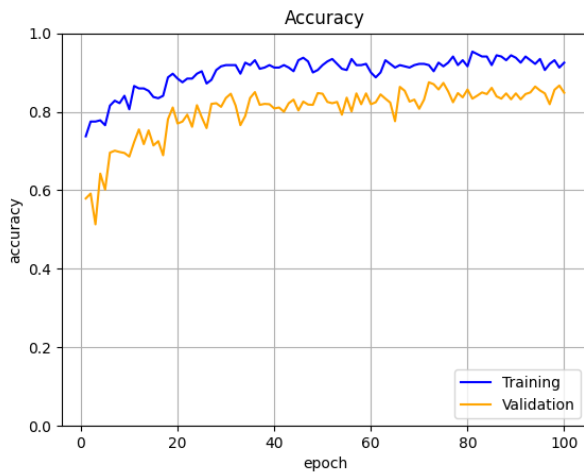


図 11 学習回数における正解率の推移

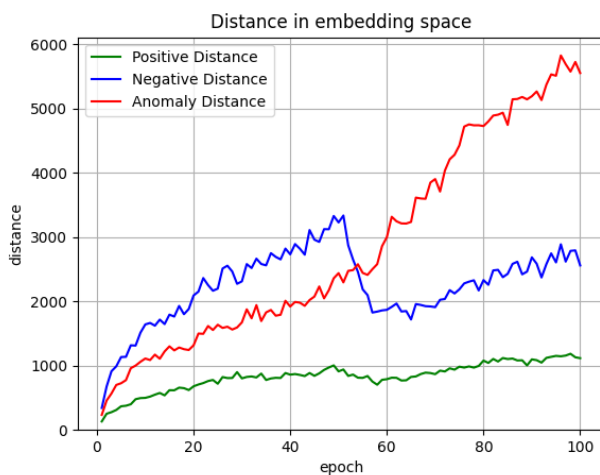


図 12 特徴量空間内の各サンプルの距離

4.3 進捗推定モデルの評価

オクルージョンを考慮した進捗推定モデルの評価を行った。

提案手法を 100epoch 学習させた際の混同行列を図 13, t-SNE[12] による特徴量空間の 2 次元可視化の結果を図 14 に示す。

また、隣のステップとの間違いを考慮した適応的なマージン手法 [5] との比較も行った。提案手法であるオクルージョンを考慮した固定マージン、適応的マージンの手法と考慮していない固定マージン、適応的マージンの手法それぞれの正解率の結果を表 1 に示す。

提案したオクルージョンを考慮に入れた適応的マージン手法で、82.9%という最も高い正解率を達成することができた。また混同行列から、提案手法で高い精度で推定に成功していることが確認できた。さらに、t-SNE による 2 次元可視化の結果から、同ステップのサンプルではまとまりができ、異常サンプルとは距離が離れていることが確認できた。これらの結果より、提案手法の有効性を検証することができた。

4.4 検出を含めた手法全体の評価

物体検出、切り出し、進捗推定の一連の流れに対する実験を行った。物体検出においては、物体追跡手法の Siam Mask[13] を用いて約 1000 枚の画像に対しラベル付けを行い、Faster R-CNN をファインチューニングした。Faster R-CNN は深層学習用ライブラリ

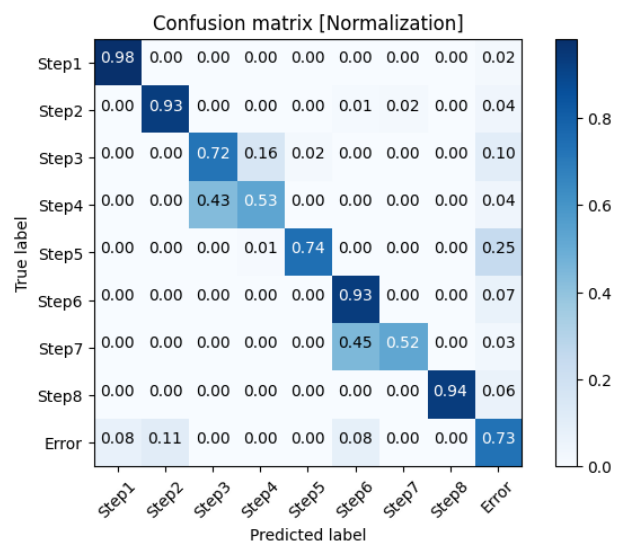


図 13 混同行列による評価

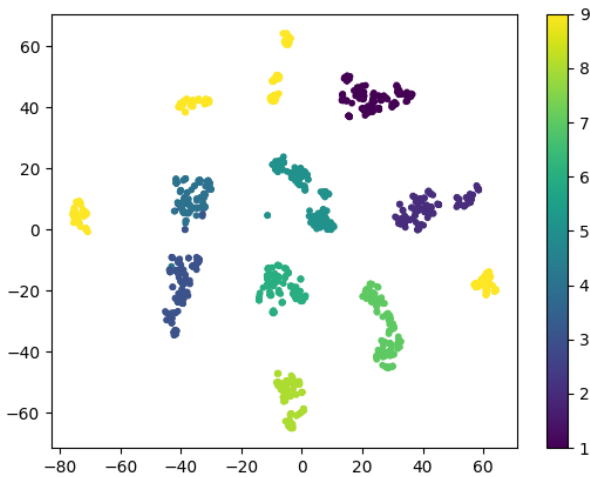


図 14 t-SNE による特徴量空間の二次元可視化

表 1 各手法との正解率の比較

Loss Function Name	Accuracy
Triplet(fixed)	0.793
Trplet(adaptive)	0.796
Triplet Ano(fixed)	0.826
Trplet Ano(adaptive)	0.829

Detectron2[14] で実装し、MS COCO データセット [15] で事前学習済みのモデルに対してファインチューニングを行った。また作業員によるオクルージョンで生じる誤判定を避けるため、20 フレーム連続で同じ判定が出たら判定ステップを確定する方法を利用した。

実験の結果、すべてのステップで正しい判定をログとして残すことができた。実際のシステムを動かした様子を図 15 に示す。黄色のパウンディングボックスが対象物の検出を表しており、右側の赤文字の数字が判定されたステップを表している。これらの結果から、物体検出を含めたシステム全体の有効性も検証できた。

5 結論

本研究では対象物に着目した組立作業の進捗推定システムの構築を行った。進捗推定では Deep Metric Learning を用いた手法を提案した。また、オクルージョンを考慮した推定手法を提案し、実験でその有効性を検証



図 15 物体検出と進捗推定に成功している様子

した。

今後の展望として、検出した対象物の時系列情報を記録するために、トラッキング手法を取り入れたシステムの構築を目指す。

参考文献

- [1] H. S. Kang, J. Y. Lee, S. S. Choi, H. Kim, J. H. Park, J. Y. Son, B. H. Kim and S. D. Noh, "Smart manufacturing: Past research, present findings, and future directions," International Journal of Precision Engineering and Manufacturing-Green Technology 3, pp.111-128, 2016.
- [2] J. Wang, Y. Ma, L. Zhang, R. X. Gao and D. Wu, "Deep learning for smart manufacturing: Methods and applications," Journal of Manufacturing Systems, vol.48, Part C, July 2018, pp.144-156.
- [3] 日立産業制御ソリューションズ: "AI による人物姿勢・動作認識ソリューション", https://info.hitachi-ics.co.jp/product/activity_evaluation/, Accessed: 2023-01-12.
- [4] M. Funk, A. Bachler, L. Bachler, T. Kosch, T. Heidenreich, A. Schmidt, "Working with Augmented Reality? A Long-Term Analysis of In-Situ Instructions at the Assembly Workplace," PETRA '17, 2017.
- [5] 橘川拓実, A. Moro, S. Pathak, 原田佳周, 西川英雄, 野口稔, 濱谷章史, 梅田和昇, "Deep Metric Learning を用いた組立作業の進捗判定", 精密工学会秋季大会学術講演会講演論文集, D21, 2022, pp300-301.

- [6] E. Xing, M. Jordan, S. J. Russell, A. Ng, “Distance Metric Learning with Application to Clustering with Side-Information,” *Neural Information Processing Systems (NIPS)*, 2002.
- [7] J. Lu, J. Hu, J. Zhou, “Deep Metric Learning for Visual Understanding: An Overview of Recent Advances,” *IEEE Signal Process. Mag.* 2017, 34, pp.76–84.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Neural Information Processing Systems (NIPS)*, 2015.
- [9] E. Hoffer, N. Ailon, “Deep metric learning using Triplet network,” *International Workshop on Similarity-Based Pattern Recognition, SIMBAD 2015*, pp.84-92.
- [10] T. Cover, P. Hart, “Nearest neighbor pattern classification”, *IEEE Transactions on Information Theory*, vol.13, 1967, pp.21-27.
- [11] IO DATA Co., Ltd.: “Network Camera Qwatch TS-WRLP,” <https://www.iodata.jp/product/lancam/lancam/ts-wrlp/index.html>, Accessed: 2023-01-12.
- [12] L. van der Maaten and G. E. Hinton, “Visualizing Data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp.2579-2605, 2008.
- [13] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, “Fast online object tracking and segmentation: A unifying approach,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [14] Facebook AI: “Detectron2,” <https://ai.facebook.com/tools/detectron2/>, Accessed: 2022-07-22.
- [15] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft COCO: Common objects in context,” *Computer Vision and Pattern Recognition (CVPR)*, 2015.