

# 折り図の自動生成のための折り線推定とその評価

加藤 眸<sup>1,a)</sup> 加藤 大貴<sup>1,b)</sup> 平山高嗣<sup>2,c)</sup> 駒水 孝裕<sup>1,d)</sup> 井手 一郎<sup>1,e)</sup>

## 概要

折り紙は、紙を折ることで様々な対象を模した形状を作る遊びである。折り紙作品の共有や折り方の伝達のためには、折り図や折り方動画が広く用いられている。しかし、折り図の作成には多大な時間と労力を要するため、本研究では折り方動画から折り図を自動生成することを試みる。本報告では、その中でも折り操作の前後に撮影された画像対から折り線を推定する「折り線推定」という部分問題に焦点を当てる。独自に構築したデータセットを用いた評価実験の結果、5層の畳み込みニューラルネットワーク (Convolutional Neural Network; CNN) をバックボーンとするモデルが最も高い性能を示し、その推定誤差は距離のずれが画像の1辺の約6%、角度のずれが約14°であった。

## 1. はじめに

折り紙は、紙を折ることで様々な対象を模した形状を作る遊びである。折り紙は世界中で親しまれており、教育や医療分野への応用も期待されている。

折り紙の折り方を伝達する主な手法に、折り図と折り方動画がある。折り図は、折り方の手順を段階的に示す図であり、多くの書籍で採用されている。一方、折り方動画は、手と紙の動きを撮影することで折り手順を記録し、連続的な視覚情報を通じて直感的な理解を促すものである。

折り図は、読者にとって理解しやすいように各手順の折り紙の状態を描画することで、折り方を正確に伝達できる利点があるが、描画ソフトウェアを用いて手作業で手順毎の図を描画する必要があるため、作成に多大な労力と時間を要する。一方、折り方動画には、画角や照明などの影響により折り紙の状態が正確に伝わらないという欠点がある。例えば、手の動きが視界を遮ったり、光の反射の影響で折りの目印となる折り線が見えなくなったりすることで、折り紙の状態を認識することは困難になる。このよう

な背景から、折り紙の折り手順をより効果的に伝達するためには、両形式を相補的に併用することが望ましい。近年では、スマートフォンの普及により折り方動画の作成が容易になったことから、折り方動画から折り図を自動生成する技術の実現が期待されている。

コンピュータ上で折り紙をモデル化し、折り手順を自動生成する研究はいくつか存在する [1, 2] が、折り方動画から直接折り図を生成する研究は存在しない。

本研究の最終的な目標は、任意の折り方動画から折り図を自動生成するシステムを開発することである。我々はこの目的の達成に向けて、折り図生成過程を5つの部分問題に整理し、システム実現に向けた第一歩として、その部分問題の1つである折り線推定に焦点を当てた手法を提案した [3]。具体的には、以下の条件に問題設定を限定した折り線推定に取り組んだ。

- 入力 は 折り状態画像対である。ここで、「折り状態画像対」とは、1つの折り操作の前後の折り紙の画像の組を指す。
- 出力 は、折り状態画像対間で実行された折り操作を表す線分である。
- 折り操作は、1つの谷折り線で表現可能な「直線折り」に限定し、1回の折り操作で追加される折り線は1本のみとする。

本発表では、我々が以前発表提案した手法 [3] についてより精緻で詳細な評価・分析を行なうため、一般的な画像処理タスクに用いられる複数のモデルをバックボーンに用いた実験を行ない、折り線推定手法を詳細に評価する。

## 2. 折り線推定器

先行研究 [3] で提案した、折り状態画像対  $(I_{i1}, I_{i2})$  が与えられたときに、折り線  $l_i$  を推定する手法について述べる。ここで、 $I_{i1}$  は折り操作前、 $I_{i2}$  は折り操作後の画像である。折り線推定器  $f$  は、折り線  $l_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2}) \in \mathbb{R}^4$  を出力する。この4次元ベクトルは、 $l_i$  の両端点の  $I_{i1}$  上での座標  $(x_{i1}, y_{i1})$  および  $(x_{i2}, y_{i2})$  を表す。

学習段階では、折り線推定器が折り状態画像対とそれに対応する折り線との関係を学習する。推定段階では、学習段階で得られた折り線推定器  $f$  に折り状態画像対を入力す

<sup>1</sup> 名古屋大学

<sup>2</sup> 人間環境大学

a) katoht@cs.is.i.nagoya-u.ac.jp

b) katoh@cs.is.i.nagoya-u.ac.jp

c) t-hirayama@uhe.ac.jp

d) taka-coma@acm.org

e) ide@i.nagoya-u.ac.jp

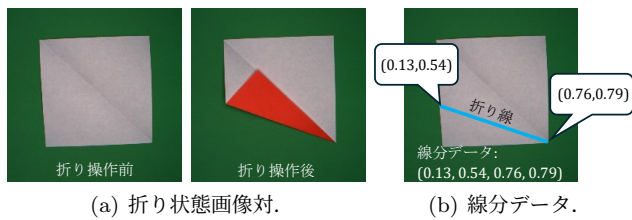


図 1: 折り状態画像対に対する線分アノテーションの例。座標は浮動小数点の組として  $[0.00, 1.00]$  の範囲で表現され、画像の左端の  $x$  座標を 0, 右端を 1, 上端の  $y$  座標を 0, 下端を 1 とする。

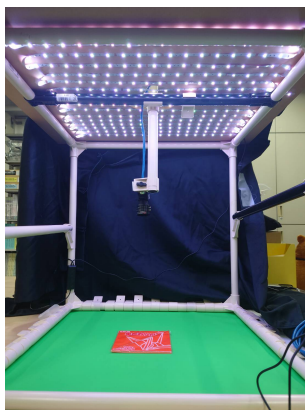


図 2: 撮影環境。天井に白色 LED テープライト (SwitchBot 製の白色 LED テープライト, 全長 10 m) を設置し、外光の影響を排除するために全体を暗幕で覆っている。

ることで折り線推定を行なう。

### 3. 直線折りデータセット

折り状態画像対と、それに対応する折り線を表す線分データからなる公開データセットは存在しないため、我々は 49 種類の折り方に対して、学習用・検証用・評価用のデータセットを独自に作成した。これら 3 つのセットに含まれる折り方はすべて同一である。49 種類の折り方を 3 回通り撮影後、折り状態画像対に対して線分アノテーションを付与し、49 組のデータを 3 セット構成した。図 1 に折り状態画像対に対する線分アノテーションの例を、図 2 に撮影環境をそれぞれ示す。

## 4. 評価実験

### 4.1 折り線推定器の実装

折り状態画像対と折り線を表す線分アノテーションとの関係を学習するために、バックボーンと全結合層からなる単純な回帰モデルを構築した。モデルの概要を図 3 に示す。折り操作の前後の各画像に対して、バックボーンによりそれぞれの特徴を抽出し、それらを連結して全結合層に入力し、最終的に折り線を表す 4 次元ベクトルを出力する。本実験では、折り線推定性能の比較のため、以下の 6 種

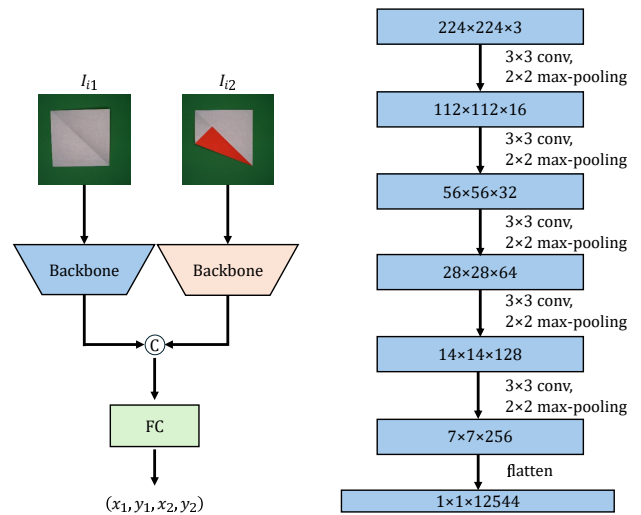


図 3: 折り線推定器の構造。

図 4: CNN バックボーン。

類のバックボーンを用意した。

- CNN: 5 層構成の単純な畳み込みニューラルネットワーク (Convolutional Neural Network; CNN) (図 4 に構造を示す)
- RN18: 事前学習なしの ResNet18 [5]
- RN18+: ImageNet1k [4] で事前学習された ResNet18
- RN34: 事前学習なしの ResNet34 [5]
- RN34+: ImageNet1k で事前学習された ResNet34
- ViT+: ImageNet1k で事前学習された Vision Transformer (ViT) -B/16 [6]\*

損失関数には平均二乗誤差 (Mean Squared Error; MSE) を用いた。学習エポック数は早期終了により決定し、検証誤差が 10 エポック連続で最小値を更新しなかった場合、学習は収束したとみなして終了した。

### 4.2 評価方法と指標

3 分割交差検証により折り線推定器の性能を評価した。折り線推定器の学習時には、回転および平行移動によるデータ拡張を行なった。その際、折り操作中に制作者が折り紙の位置や向きをわずかに変える可能性を考慮し、このような変動に対する頑健性を高めるため、 $I_{i1}$  および  $I_{i2}$  の回転角をそれぞれ約  $10^\circ$  以内、平行移動量を約 10 画素 (画像の 1 辺の長さの 5% 程度) 以内の範囲で無作為に変化させた。データ拡張の倍率は、10 倍 (490 組), 100 倍 (4,900 組), 300 倍 (14,700 組), 1,000 倍 (49,000 組) の 4 段階で設定した。

折り線推定器が出力した線分データの精度を評価するために、先行研究 [3] と同様に Zhan らによる線分検出に関する研究 [7] で定義された EA-Score を参考にした 2 つの評価指標  $E_d$  および  $E_\theta$  を定義する。

まず、線分間の相対的な位置ずれを評価する指標として

\* ViT をはじめてから学習するにはデータ量が不十分であると判断し、事前学習済みモデルのみを使用した。

$E_d$  を導入する. 具体的には, 線分間の距離をそれぞれの中点間の Euclidean 距離と定義し,  $N$  個のテストデータに対する出力線分と正解線分の距離の平均をとる. この際に, 画像の 1 辺の長さを 1 に正規化した相対値として距離を表現する. 座標は  $[0.00, 1.00]$  の範囲の浮動小数点数の組で表されるため,  $i$  番目のテストデータ ( $i = 1, 2, \dots, N$ ) に対して, 正解線分  $l_i$  の中点の座標を  $(x_i, y_i)$ , 推定線分の中点の座標を  $(\hat{x}_i, \hat{y}_i)$  とすると,  $E_d$  ( $0 \leq E_d \leq \sqrt{2}$ ) は以下のように定義される.

$$E_d = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (1)$$

ここで,  $(x_i, y_i) = ((x_{i1} + x_{i2})/2, (y_{i1} + y_{i2})/2)$ ,  $(\hat{x}_i, \hat{y}_i) = ((\hat{x}_{i1} + \hat{x}_{i2})/2, (\hat{y}_{i1} + \hat{y}_{i2})/2)$  である.

$E_d$  は距離のずれを評価するが, 角度のずれは評価できない. 折り図においては, 折り線の角度のずれも小さいことが望ましいため, 次に  $E_\theta$  を定義する. これは, 出力線分と正解線分間の角度の平均を表す指標である.  $i$  番目のテストデータに対して, 正解線分  $l_i$  と推定線分  $\hat{l}_i$  のなす角を  $\theta(l_i, \hat{l}_i)$  とすると,  $E_\theta$  ( $0^\circ \leq E_\theta \leq 90^\circ$ ) は以下のように定義される.

$$E_\theta = \frac{1}{N} \sum_{i=1}^N \theta(l_i, \hat{l}_i) \quad (2)$$

### 4.3 折り線推定性能の評価実験

表 1 および表 2 に, 各バックボーンおよびデータ拡張倍率に対する距離のずれ  $E_d$  および角度のずれ  $E_\theta$  の値を示す. また, 画像  $I_{i1}$  および  $I_{i2}$  が与えられたときに CNN をバックボーンに用いたモデルにより推定された線分データ  $\hat{l}_i$  の例を図 5 に可視化して示す.

バックボーン間で評価指標を比較すると, 多くの場合において CNN が  $E_d$  および  $E_\theta$  の両方で最良の性能を示した. 特に, データ拡張倍率が 1,000 のとき,  $E_d$  は最小値 0.059 を記録した. これは, 距離のずれが画像の 1 辺の長さの約 6% に相当することを意味する.

ResNet モデルにおける事前学習の有無による比較では, いずれの指標においても顕著な差は見られなかった. これは, ImageNet [4] のような一般画像による事前学習が, 本タスクにおいてはドメインギャップのために有効に働かなかったことが原因と考えられる.

加えて, ResNet 系のバックボーン性能が総じて良くなかったことから, モデルの層が深くなることで, 折り線に関する位置情報が失われた可能性も考えられる.

### 4.4 失敗事例の分析

提案した折り線推定器の特性をさらに検討するため, 4.3 節の実験において CNN をバックボーンに用いたモデルが折り線の推定に失敗した画像を分析する. 図 6 に失敗

表 1: データ拡張倍率およびバックボーンごとの距離のずれ  $E_d$ .

拡張倍率	データ数	CNN	RN18	RN18+	RN34	RN34+	ViT+
10	490	0.177	0.159	0.156	0.151	0.151	0.153
100	4,900	0.137	0.149	0.148	0.145	0.142	0.153
300	14,700	0.110	0.149	0.134	0.124	0.126	0.123
1,000	49,000	0.059	0.121	0.118	0.120	0.105	0.056

表 2: データ拡張倍率およびバックボーンごとの角度のずれ  $E_\theta$  [ $^\circ$ ].

拡張倍率	データ数	CNN	RN18	RN18+	RN34	RN34+	ViT+
10	490	39.93	40.86	40.99	40.72	40.69	41.11
100	4,900	25.90	40.46	41.07	39.51	40.59	36.57
300	14,700	24.88	38.37	40.01	35.46	39.98	31.19
1,000	49,000	13.96	31.60	34.66	33.22	36.26	17.40

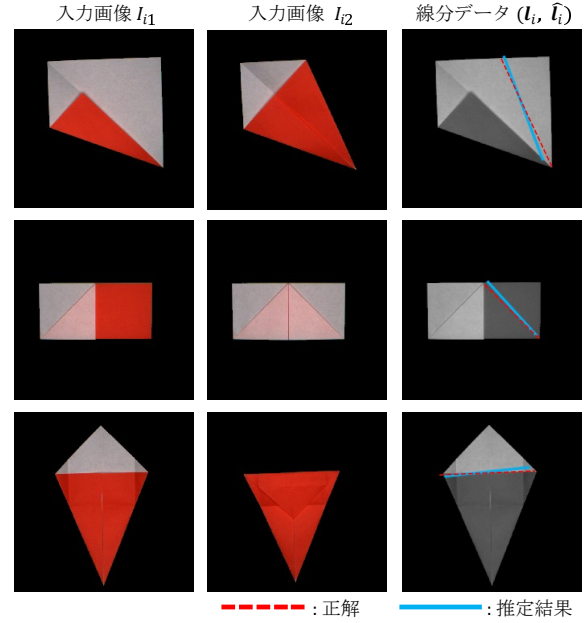


図 5: 入力画像と出力線分データの例.

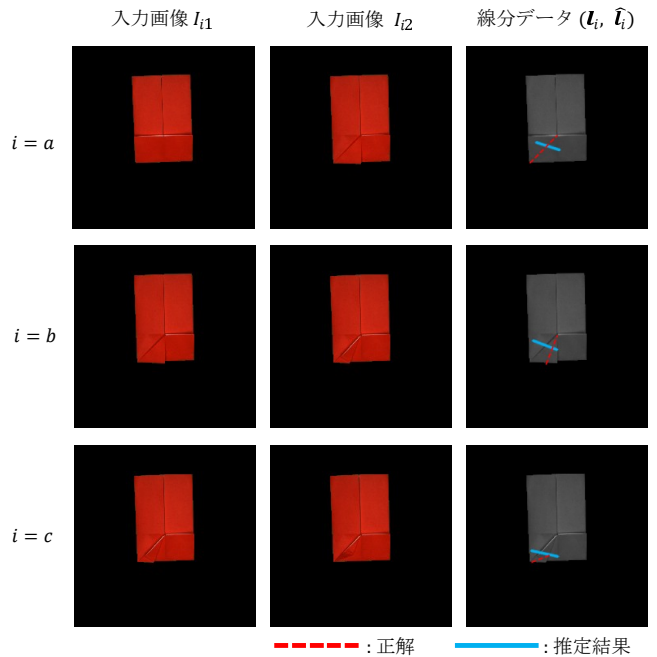


図 6: 折り線推定に失敗した例.

表 3: 各データ拡張倍率における距離のずれ  $E_d$  および角度のずれ  $E_\theta$  [°].

拡張倍率	データ数	$E_d$	$E_\theta$
10	490	0.133	59.26
100	4,900	0.103	17.76
300	14,700	0.063	10.69
1,000	49,000	0.055	5.82

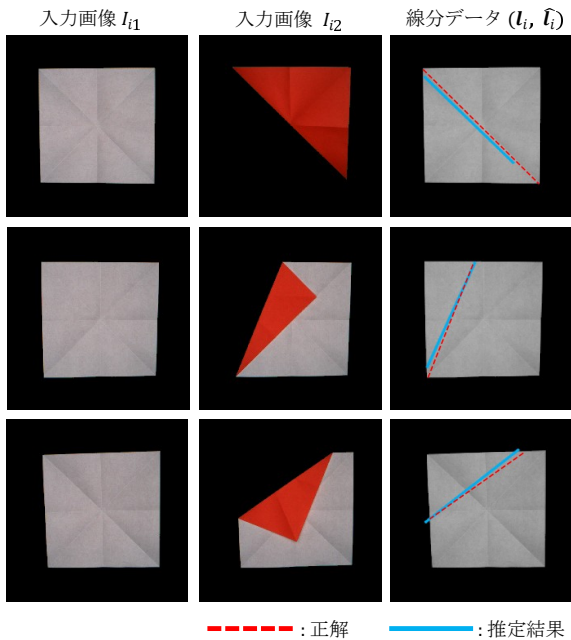


図 7: 追加実験における入力画像と出力線分データの例.

例を 3 つ示す. これらの入力折り状態画像対を, それぞれ  $(I_{a1}, I_{a2})$ ,  $(I_{b1}, I_{b2})$ ,  $(I_{c1}, I_{c2})$  とする.  $I_{a1}$ ,  $I_{b1}$ ,  $I_{c1}$  および  $I_{a2}$ ,  $I_{b2}$ ,  $I_{c2}$  は, いずれもシルエットや色が類似しており, 折り操作の前後で折り紙のシルエットに変化が見られない. また, これら 3 組に対して推定された折り線は, いずれも類似した位置にあり, 推定器がこれらの画像対を区別できていない可能性がある.

#### 4.5 類似した折り方に対する追加分析

4.4 節の実験では, 推定器が一部の折り状態画像対を区別できない事例が確認された. そこで本節では, 推定器が識別可能な折り操作の種類を調査するため, 追加実験を行った. 具体的には, 同じ折り紙の状態から始まり, シルエットが変化するような折り操作を含む  $N = 10$  個の新しいテスト用データを用意し, 推定器が正しく折り線を予測できるかを検証した.

まず, 折り操作前の画像  $I_{i1}$  ( $i = 1, 2, \dots, 10$ ) における折り紙は, 縦・横・対角方向に 4 本の折り線が入った正方形の折り紙である. 次に, 折り操作後の画像  $I_{i2}$  における折り紙は, この正方形状態から何らかの 1 本の折り線に沿って折られている. これらの画像を, 4.3 節で学習した CNN をバックボーンとする折り線推定器に入力し, 評価

指標  $E_d$  および  $E_\theta$  を用いて性能を評価した.

データ拡張倍率に対する  $E_d$  および  $E_\theta$  の関係を表 3 に示し, 画像  $I_{i1}$  および  $I_{i2}$  が与えられたときに推定された線分データ  $\hat{I}_i$  の例を図 7 に示す. これらの結果から, 両評価指標の値は 4.3 節の実験結果と同程度であること, 各データに対して異なる線分データが出力されていることが確認できる. また, 本実験で用いたデータには学習データに含まれていない折り方も含まれていたが, 推定器はある程度の性能を示した.

図 6 に示した結果では, シルエットや色が類似した画像対に対して類似した線分データが出力されていたのに対し, 本追加実験では各データに対して異なる線分データが出力されている. これは, 推定器が折り操作前後でシルエットが変化するような折り方の各データの違いを識別し, 入力画像と線分データの関係性を正しく学習していたことを示唆している. さらに, 学習データに含まれていない折りパターンに対してもある程度の性能を示したことから, 折り操作前後でシルエットが変化するような折り方に対して, 推定器は一定の汎化能力を有していると考えられる.

4.4 節および本節の議論から, 直線折りの中にも推定が比較的容易な折り方と困難な折り方が存在しており, これらが混在することで, 推定が困難な折り方に対しては, 特徴が十分に学習されず推定に失敗している可能性がある. この問題を解決するためには, 直線折りの種類ごとに異なる折り線推定器を学習させることが考えられる.

## 5. 結論

本研究では, 折り方動画から折り図を自動生成するシステムの開発を目指し, 直線折り操作における折り線推定に取り組んだ. そのために, 折り線推定器の学習および評価用のデータセットを構築した. 評価においては, 5 層の CNN, ResNet [5], および ViT [6] をバックボーンとする 6 種類の回帰モデルを実装し, 折り線推定性能を比較した. その結果, 5 層の CNN を用いたモデルが最も高い性能を示し, 距離のずれは画像の 1 辺の約 6%, 角度のずれは約  $14^\circ$  であった. これらの結果は, 学習ベースの折り線推定器の構築が実現可能であることを示唆している.

今後の課題としては, 折り線推定器の推定能力の向上, 直線折り以外の折り操作への対応が挙げられる. また, 折り図の自動生成に必要な他の部分問題, 特に折り操作の分類や折り紙状態図の生成も, 既存手法の適用可能性や新たな手法の必要性などの検討が必要である.

**謝辞** 本研究にあたり, 貴重なご助言をいただいた名古屋大学の松平茅隼さん, 池田昂太郎さんに深く感謝申し上げます.

参考文献

- [1] H. Akitaya, M. Ribeiro, C. Koike, and J. Ralha, “Development of an intuitive algorithm for diagramming and 3D animated tutorial for folding crease patterns,” in *Origami 5: 5th Int. Meeting of Origami Science, Mathematics, and Education*, pp. 347–369, AK Peters / CRC Press, Boca Raton, FL, USA, 2011. <http://dx.doi.org/10.1201/b10971-31>
- [2] J. Fastag, “eGami: Virtual paperfolding and diagramming software,” in *Origami 4: 4th Int. Meeting of Origami Science, Mathematics, and Education*, pp. 273–283, AK Peters / CRC Press, Boca Raton, FL, USA, 2009. <http://dx.doi.org/10.1201/b10653-29>
- [3] 加藤暉, 加藤大貴, 平山高嗣, 駒水孝裕, 井手一郎, “折り図の自動生成に向けた折り紙画像からの折り線推定,” 信学技報, MVE2024-98, 2025.
- [4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. 2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, 2009. <http://dx.doi.org/10.1109/CVPR.2009.5206848>
- [5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. 2016 IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, 2016. <http://dx.doi.org/10.13140/RG.2.2.33865.52329>
- [6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. 9th Int. Conf. on Learning Representations*, 16p., Virtual, 2021. <http://dx.doi.org/10.48550/arXiv.2010.11929>
- [7] K. Zhao, Q. Han, C.-B. Zhang, J. Xu, and M.-M. Cheng, “Deep Hough transform for semantic line detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4793–4806, 2021. <https://doi.org/10.1109/TPAMI.2021.3077129>