

研究論文

畳み込みニューラルネットワークによる繊維鑑別

市毛将司^{*1}、後藤拓海^{*1}、木村和幸^{*1}、三輪幸弘^{*2}

Fiber Identification Using Convolutional Neural Networks

Masashi ICHIGE^{*1}, Takumi GOTO^{*1}, Kazuyuki KIMURA^{*1} and Yukihiro MIWA^{*2}Owari Textile Research Center^{*1*2}

畳み込みニューラルネットワーク(CNN)による画像判別を用いた繊維鑑別を提案した。データセットに用いた繊維画像は、顕微鏡で撮影を行いハフ変換による直線検出アルゴリズムを利用し適切なサイズにトリミングした。データセットを VGG16、Resnet50 V2 及び MobileNetV2 モデルで学習・検証した結果 97%以上の正解率が得られた。顕微鏡システムと共有フォルダを介してデータ連係を行い、1 フレームあたり 0.4 秒の推論速度を得た。

1. はじめに

尾張地域は古くから繊維産業が盛んであり、紡績機、撚糸機、織機など、多くの繊維機械が稼働している。そうした中、異種繊維の混入について相談を受けた際には、当センターでは繊維鑑別を依頼試験として行っている。繊維鑑別は JIS L1030-1 に従って行われるが、燃焼試験及び化学試験、顕微鏡試験、赤外吸収スペクトルの測定試験など複数の工程があり時間や経験も必要である。

AlexNet¹⁾が登場した 2012 年以降画像に対する処理技術が急速に発展し、画像のカテゴリ判別やキズ検出などの応用例が相次いでいる。そこで、これまで熟練者に頼っていた光学顕微鏡による繊維鑑別について、CNN を活用した繊維の種類判別ネットワークを構築し、分析者のサポートや簡易的な鑑別を目指すこととした。

判別対象としては、赤外吸収スペクトルの測定が繊維鑑別の実質的なファーストステップであることを考慮し、赤外分光分析だけでは鑑別困難なセルロース系繊維及びポリアミド系繊維とした。

2. 実験方法

2.1 光学顕微鏡の主要諸元と撮影

画像判別用の CNN を構築するには多くの教師画像(学習用データセット)を必用とする。画像の取得には以下の光学顕微鏡を使用した。

(本体)OLYMPUS BX53

(対物レンズ)OLYMPUS UPlanSApo 20 倍

(カメラ)OLYMPUS DP73 201 万画素 1/1.8 インチ カラー CCD

(出力画像)jpg 形式 1600×1200 画素

綿、麻、レーヨン、キュプラ及び羊毛、絹の 6 種の繊維について、JIS L0803 に用いる添付白布及び当センター保有試料を撮影した。各繊維はあらかじめ単繊維にほぐし、流動パラフィンに封入したプレパラートを作製した。1 から数本の繊維が視野に収まるように調整し、各繊維で 1000 枚程度撮影した。

2.2 試験用データセットの作成

一般的な CNN で提供されているモデルの入力画素数は 224×224 画素や 299×299 画素である。入力サイズに合わせて撮影画像を単純に縮小すると各種繊維の特徴が失われてしまうため、トリミングが有効である。

トリミングにあたっては、ルールベースの画像処理を活用し、図 1 のとおり自動化することとした。繊維には直線部分が多いことに着目し、直線検出アルゴリズムであるハフ変換を利用し、トリミングの対象を設定した。ハフ変換の前処理として画像輪郭の抽出と 2 値化を行った。ハフ変換は OpenCV4 の HoughLinesP 関数²⁾を利用した。ハフ変換により複数の直線が検出されるが、学習用画像として多様性を持たせるためある程度離れた直線を選択することが望ましい。ここでは、200 画素ごとのグリッド内では複数選択しないこととした。最終的に選択した直線の中心座標を中心とした正方形画像 400×400 画素でトリミングした。学習用データセットと検証用データセットの分割にあたっては、トリミング前の画像が双方にまたがらないようにした。

2.3 畳み込みニューラルネットワークの実験環境

CNN を利用するためのフレームワークなどの実験環境は表 1 を用いた。CNN のモデルは、最終の全結合層の前にグローバルアベレージプーリングを行った。グ

^{*1} 尾張繊維技術センター 機能加工室 ^{*2} 尾張繊維技術センター 機能加工室 (現県民生活課)

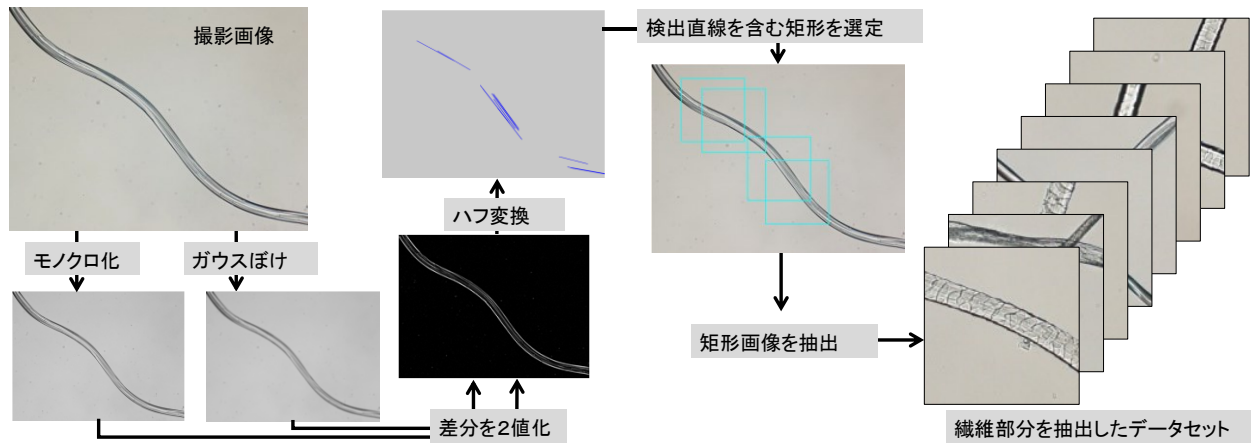


図1 撮像画像から試験用データセットの作成

ローバルアベレージプーリングを行うことにより、パラメータが減少して計算負荷が下がると共に、精度向上することが知られている³⁾。

表1 実験環境

言語、フレームワーク	Python, Tensorflow2.5
モデル	VGG16, ResNet50V2, MobileNetV2 (Image net 事前学習済み)
入力画像サイズ	224×224 画素
学習エポック数	50(全層を学習)
バッチサイズ	32
学習率	10 ⁻⁶ で開始し、10 エポック目で 10 ⁻⁴ とし、3 エポック保持した後 1 エポックごとに 0.95 倍した。
損失関数	カテゴリカルクロスエントロピー
OS	Ubuntu22.04.1 LTS
CPU	Intel Core i9-13900K
メモリ	128[GB]
GPU	NVIDIA GeForce RTX4090

2.4 注目領域の可視化

従来型ルールベースによる工業製品の画像検査などでは、寸法・形状、指定部位の色が異なる等、判断根拠を明確に述べるができるが、今回利用した畳み込みニューラルネットワークでは根拠の明確化はできない。しかし、モデルが望ましい動作をしているか否か可能な限り把握する必要がある。判別に寄与した画像内の部分を可視化するため Grad-CAM⁴⁾を各学習済みモデルの最終

畳み込み層に対して適用し、ヒートマップを対象の画像に重畳して表示した。

2.5 顕微鏡画像のオンサイトでの推論

多くの顕微鏡付属のカメラは専用の顕微鏡システムを構成しており、外部のシステムを組み込んで連携することが難しい。また、既存の顕微鏡システムの PC はスペックが低く CNN の推論プログラムの動作が困難であった。このため、図2のように、顕微鏡システムで画像をキャプチャして共有フォルダに保存し、推論用 PC から共有フォルダを監視して推論を行った。

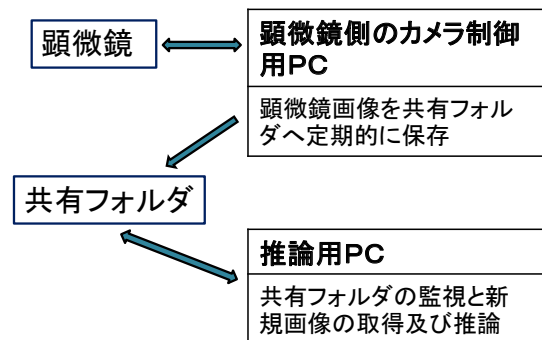


図2 顕微鏡システムと推論用 PC の連携

3. 実験結果及び考察

3.1 学習中の損失関数と正解率

図3に学習用データセットによって CNN が学習される様子を示す。CNN では学習データでの損失関数の値が小さくなるように誤差逆伝播により学習が行われる。最初の数エポックで急激に損失関数の値が下がると共に正解率が向上し、その後も少しずつ学習が進んでいくことが分かる。今回の学習条件では、VGG16 と比較すると ResNet50V2 や MobileNetV2 の方が前半でスムーズ

に学習が進行した。

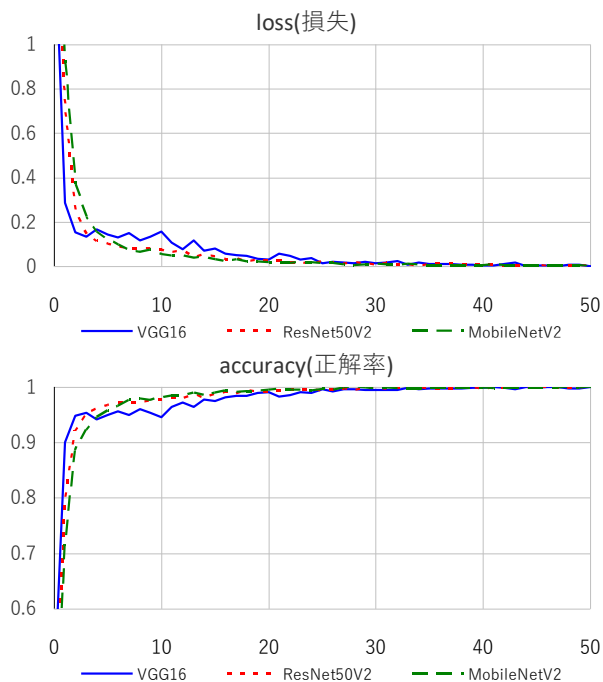


図3 学習データセットでの損失と正解率

図4は、図3の学習には使用しなかった検証用データセットを用いてモデルの性能を各エポックごとに評価したものである。MobileNetV2では5から20エポックでは検証用データでの損失が大きく、図2の学習用データでの損失は小さいことから過学習していると思われるが、

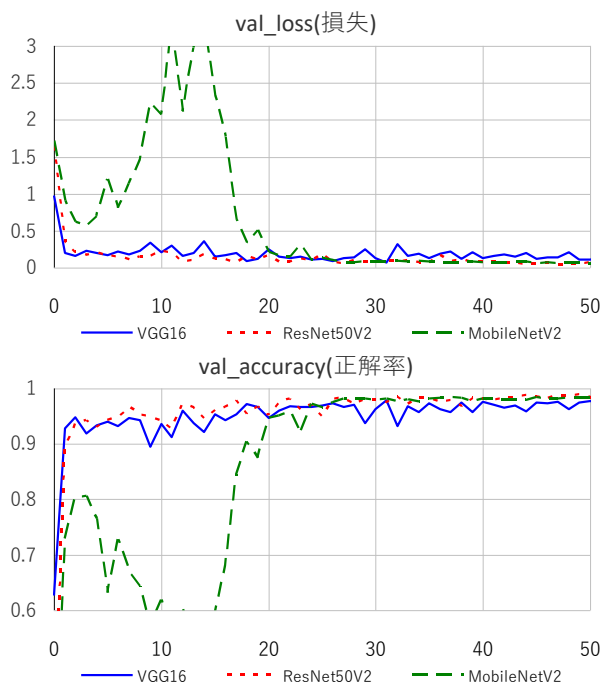


図4 検証データセットでの損失と正解率

その後は良好に学習した。いずれのモデルでも30から50エポックで学習がほぼ平衡に達しており、50エポック学習後の検証用データセットでの正解率はResNet50V2が0.9844、MobileNetV2が0.9835、VGG16が0.9770であった。また、モデル及び繊維の種類ごとの正解率は96%を上回り分析者をサポートしなからの鑑別への利用が可能と考えられた。

3.2 Grad-CAMによる注目領域の可視化

図5は羊毛の画像を学習済みモデルで推論した際のGrad-CAMによるヒートマップである。ヒートマップの青い部分は判別への寄与が少なかった領域、赤い部分は判別への寄与が大きかった領域を示している。画像の例は2本の繊維が同一視野に含まれた場合である。VGG16と比較するとResNet50V2はやや偏りがあることと、MobileNetV2ではより広い領域に注目していることが判った。

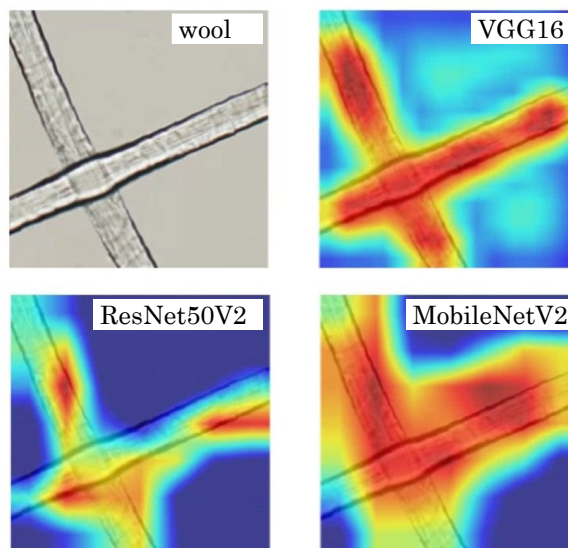


図5 Grad-CAMによる判断領域の可視化

3.3 顕微鏡システムとの連携と推論速度

図6に推論を行っている様子と、表2に推論速度を示す。推論速度の計測には46枚のデスクトップのキャプチャ画像を用い、図中に枠で例示した推論に用いられたトリミング領域は1画像あたり3から7(平均3.8)であった。CPUのみで推論を行ったケースでは、1画面あたりの処理時間はMobileNetV2が一番速く、ついでResNet50V2、VGG16であった。MobileNetV2はパラメータ数が小さくモデルが軽量なため処理速度が速かったと考えられる。CNNの推論にGPUを用いたケースでは、1画面あたりの処理時間は10から20%CPUに比べて高速になった。GPUを用いた場合に推論速度が大

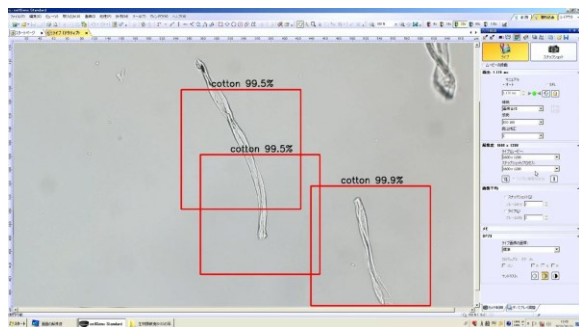


図 6 顕微鏡画像の推論

幅には速くならなかった原因としては、共有フォルダの監視と画像取り込みに一定時間要することと、トリミングが CPU 演算で行われていることが考えられる。また今回記述したコードではトリミング画像 1 枚ずつの推論であるが、複数枚バッチ処理することで高速化できると考えられる。

表 2 各モデルの推論速度

モデル	CPU 推論 (秒/フレーム)	GPU 推論 (秒/フレーム)
VGG16	0.439 ± 0.073	0.340 ± 0.037
ResNet50V2	0.414 ± 0.067	0.343 ± 0.039
MobileNetV2	0.371 ± 0.062	0.339 ± 0.037

4. 結び

本研究の結果は以下のとおりである。

- (1) 顕微鏡により 6 種の単繊維の撮像を行った。
- (2) 全体画像から繊維画像を含む部分を自動抽出する機

能を実装した。

- (3) 繊維鑑別モデルを学習し、VGG16、ResNet50V2、MobileNetV2 いずれをベースとした場合でも正解率が 96%を上回ること、繊維部分に着目されていることを確認した。
- (4) 顕微鏡観察と同時に推論するため、共有フォルダを介するシステムを構築し、1 フレームあたり 0.4 秒での推論速度を達成した。

文献

- 1) Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton: ImageNet Classification with Deep Convolutional Neural Networks, https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf, (2024/07/24)
- 2) OpenCV: OpenCV 4.6.0 Documentation, https://docs.opencv.org/4.6.0/dd/d1a/group__imgproc__feature.html#ga8618180a5948286384e3b7ca02f6feeb, (2024/05/15)
- 3) Min Lin, Qiang Chen, Shuicheng Yan: Network In Network, <https://arxiv.org/pdf/1312.4400>, (2024/05/15)
- 4) Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, <https://arxiv.org/pdf/1610.02391>, (2024/05/15)