研究論文

AIを活用した金属製品の外観確認の自働化

木嶋　祐太*　菅家　章*　小林　豊**　大野　宏***
土屋　理規****　乙川　直仁****　長瀬　敬之****　井上　大輔****

Development of Metal Parts Visual Inspection System Using AI

KIJIMA Yuta*, KANKE Akira*, KOBAYASHI Yutaka**, OHNO Hiroshi***,
TSUCHIYA Masanori****, OTOKAWA Naohito****, NAGASE Noriyuki**** and INOUE Daisuke****

抄　録

本研究では丸棒形状の金属製品を対象としたAI（Artificial Intelligence）で欠陥の判別を行うシステムを開発した。金属製品を撮影するための環境整備，物体検出用ニューラルネットワークやCNN（Convolutional Neural Network）の検査への適用と評価，検査システムの高速化と処理時間の評価，ロボットを活用した搬送装置の構築を実施した。

1. 緒　言

AI 技術の急速な発達によって，さまざまな分野でこれを使った製品・サービスの高付加価値化や生産性向上への取組が行われている。今後到来する人口減少社会への対応や，よりよい労働環境を実現するためにも，こうした取組はますます重要になることが予想される。

本研究では金属製品を対象に AI で欠陥の判別を行うシステムを開発した。外観検査を人から AI へ置き換えることで，検査精度の向上や検査員の直接生産に寄与する付加価値の高い作業への配置転換を実現できる。

2. 検査システムの構築

2.1 撮影環境の整備

検査システムはカメラ，照明，検査対象を回転させる装置で構成される。カメラと照明の位置によって撮影できる画像は大きく変わる。今回は直接光照明で撮影する方法と間接光照明で撮影する方法の 2 種類を比較した。

図 1　検査対象

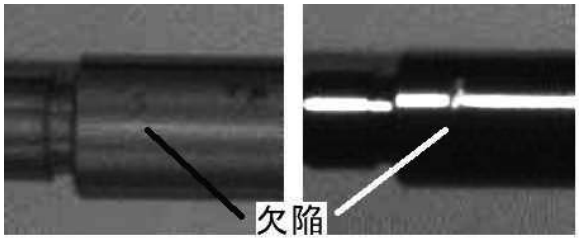

図 2　欠陥の撮影
（左：間接光照明，右：直接光照明）

図 1 に検査対象の画像を，図 2 に欠陥を撮影した画像を示す。今回検出する欠陥は金属表面についている凹凸であるため，直接光照明で撮影した時の明視野と暗視野の境界付近において，欠陥とそうでない部分の差が大きくなることが予想された。そのため，検査対象を回転させ，この境界部分で全周を検査できるようにした。

*　研究開発センター
**　下越技術支援センター
***　中越技術支援センター
****　THK 新潟（株）

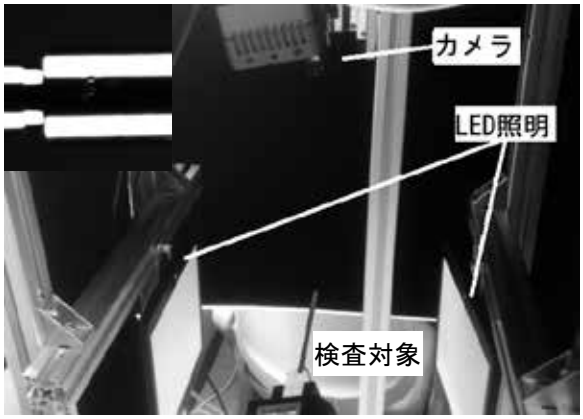

図 3　撮影環境（左上：撮影画像）

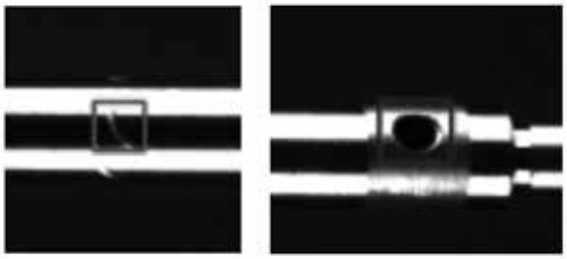

図 4　YOLO による欠陥と穴の検出

図 3 に構築した撮影環境を示す。LED 照明を両サイドから検査対象にあて，上部からカメラで撮影した。図 3 の左上が撮影した画像であり，欠陥が撮影されている。

2.2　欠陥検出方法

2.2.1　YOLO による欠陥の検出

欠陥を判別する仕組みとして，ディープラーニングを使った一般物体検出用ニューラルネットワークである YOLO を用いた。YOLO は検出した物体の分類だけでなく，物体が写っている場所も出力することができる。YOLO を学習させるためには，欠陥を撮影した画像に対し，欠陥の位置情報を付加するためのアノテーションを行う必要がある。

学習には欠陥が撮影されている画像のみを使用し，正常部の画像は使用しない。アノテーション付きの欠陥の画像を学習させて，学習に用いない画像を判定したところ欠陥が検出できた。

しかし，検査対象には穴が開いており，その部分を欠陥と誤認識する場合があった。そこで，穴についてもアノテーションを行い，穴は欠陥ではなく穴として検出するようにした。図 4 に欠陥と穴の検出結果を示す。

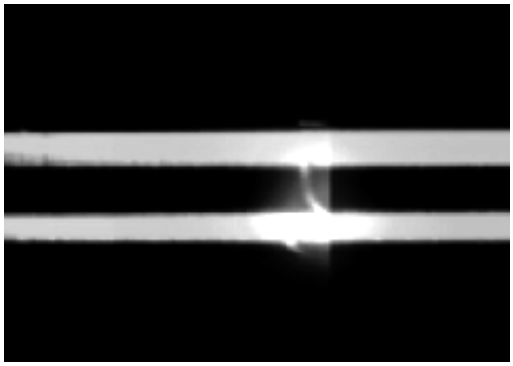

図 5　GradCAM による判断根拠の表示
（判断根拠とした部分が白くなる）

2.2.2　ResNet101 による欠陥の検出

欠陥判別の他の方法として， CNN を用いて画像を分類した。CNN の一つで，ニューラルネットワークの層を深くしても学習が進むように改良された ResNet101 を使い，画像を正常と欠陥に分類する CNN を構築した。学習には，欠陥が撮影されている画像に加えて，正常部が撮影されている画像が必要となる。YOLO で学習に使用した画像に，正常部の画像を加えたもので学習を行った。学習した ResNet101 を用いて，学習していない画像が判定できることを確認した。

YOLO では検出した欠陥の位置情報を持つため結果が妥当か判断しやすいが，ResNet101 による判定では画像全体が正常または欠陥と判定するだけで，画像のどの部分を欠陥として判定したかがわからない。そこで，GradCAM という画像の中のどの部分が判断根拠となったか可視化する手法を ResNet101 に追加した。図 5 に示すとおり，画像のどの部分をみて検査システムが判断したか，根拠を表示できるようにした。今回検証した ResNet101 では，欠陥と判定した場合は，おおむね欠陥の場所が判断根拠となっていた。

2.2.3　複数フレームの活用

人間が判定した時には欠陥とはならないが，検査システムでは欠陥と判定してしまう微小な傷が検査対象の表面に存在する場合がある。このような傷があっても人間は検査対象を傾けるなどして，照明の角度を変えた画像をえて総合的に判定できることが，人間が誤判定しにくい要因の一つである。そこで，照明の角度を変えた画像を判定に用いる方法として，複数フレームを活用する方法を考案した。検査対象を回転させて撮影するので，その前後のフレームの画像を活用すれば，照明の角度を変えた画像とすることができる（図 6）。

検査システムは ResNet101 と同様とした。ResNet101 は RGB のカラー画像を扱える。今回使用している画像はモノクロなので，カラーのデータを入れる代わりに前後のフレームを含めて 3 つのフレームを入力とすることができる。

2.3　欠陥検出の評価

2.2 の 3 種類の方法にて欠陥検出を実施し，結果を比較した。

2.3.1　データと学習の条件

21 本の検査対象のうち，15 本を学習用，1 本を検証用，5 本をテスト用に分類した。学習用と検証用の画像は，カメラの露光時間を 3 種類に変えてそれぞれ撮影し，テスト用の画像はその中間の露光時間で撮影した。カメラで撮影する画像は 1,920×1,200 であるが，ニューラルネットワークが大きくなりすぎるので，YOLO は 320×320 に，ResNet101 は 224×224 に画像をトリミングした。

学習の条件を表 1 に示す。YOLO で使用している学習用画像と検証用画像は欠陥と穴の画像のみである。ResNet101 ではそれに正常部の画像が加わる。ResNet101 の 3 フレームの画像数が少ないのは，YOLO と ResNet101 の 1 フレームでは 3 フレームのそれぞれのフレームを 1 枚の画像として使用できるからである。

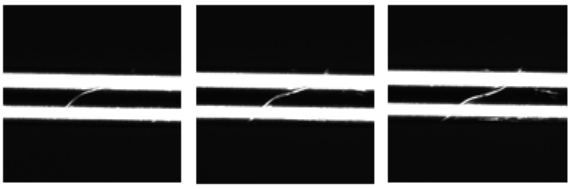

図 6　複数フレーム
（中：欠陥，左右：前後のフレーム）

表 1　学習条件

項目	YOLO	ResNet101	
		1 フレーム	3 フレーム
学習用画像数	1,092	2,522	617
検証用画像数	57	276	68
エポック数	500	50	100
データ拡張	回転，反転，ズーム等	平行移動，回転，反転，ズーム	
テスト用画像数	335	335	93

表 2　テストデータの正解率（%）

しきい値	YOLO		ResNet101			
			1 フレーム		3 フレーム	
	OK	NG	OK	NG	OK	NG
0.001	78.4	100	96.4	100	95.5	100
0.01	93.5	100	98.1	100	95.5	100
0.4	97.4	92.0	99.0	96.0	95.5	100
0.8	98.7	64.0	100	92.0	96.7	100

テスト用画像について，YOLO と ResNet101 の 1 フレームは全く同じ画像である。ResNet101 の 3 フレームは前述と同様の理由で少なくなっている。

2.3.2　テストデータの判定結果

テストデータに対する正解率を表 2 に示す。表の OK は正常部に対して正常と判定した割合で，NG は欠陥を含んだ画像に対して，欠陥と判定した割合である。NG が 100 の場合は，欠陥を含んだ画像を正常と回答することがなかったことを意味する。

検査システムは 0～1 の範囲の数値を出力するが，正常判定に近いほど 0 に，欠陥判定に近いほど 1 に近い数値を出力する。しきい値は判定の基準となる値で，しきい値以上の値を検査システムが出力した場合に欠陥判定となる。しきい値を低くすると欠陥が検知しやすくなるが，正常部を誤認識しやすくなる。

YOLO と ResNet101 に関しては，わずかに ResNet101 の正解率がよかった。YOLO は正常部に関する画像を使っていないため，その差ではないかと推測される。複数フレームを活用した結果については差異はあまりみられなかった。

2.4 検査システムの高速化

2.4.1 並列化の評価

2.3.1 の YOLO を用いると，1 フレームごとに 320×320 の画像 8 枚に対し判定処理を実行しなければならない。処理時間を高速化するためには，リソースを割り当てて並列で処理する方法があり，今回は AI で判別を行うパーソナルコンピュータに標準で搭載されている CPU（Central Processing Unit）のリソースを増やした場合と GPU（Graphics Processing Unit）のリソースを増やした場合を評価した。複数の GPU を使用する場合は，1 個目は，NVIDIA 社の TITAN RTX で，2 個目は RTX 3070 である。

表 3 が YOLO を用いて 8 枚の画像を計算した実行時間である。顕著に効果が出たのは CPU を増やした場合であった。GPU を増やす効果は，CPU を増やした時に大きくなった。GPU を増やすコストは格段に大きいため，まずは CPU の増加を検討するほうがよいと思われる。

2.4.2 アプリケーションへの適応

実際のアプリケーションでは，全周を 10°ずつ撮影することを想定し，撮影を 2 秒で終了すると仮定すると，20fps で撮影することになる。表 3 で記載した時間がかかるとするとオンタイムでは終わらないので，撮影後に計算が終了するまでの待ち時間がかかる。

表 3　8 画像の実行時間

CPU 数	GPU 1 個	GPU 2 個
1	335ms	
2	213ms	193ms
4	175ms	131ms
8	172ms	121ms

表 4　画像取得終了から計算終了までの時間

CPU 数	GPU 1 個	GPU 2 個
1	7.7s	
2	3.2s	2.9s
4	2.0s	0.7s

この計算終了までの時間を測定した結果が表 4 である。CPU や GPU を増やせば，その時間が短縮できることを確認した。ただし，撮影後に検査対象が待機位置まで戻り正常品と欠陥品を振り分けるような場合は，待機位置までの移動とこの計算は並列に動作するため，無理に短縮する必要はない。

2.5 ロボットを活用した搬送装置の構築

この検査システムを製造現場に適用するため，図 7 の搬送装置を製作した。検査対象の搬送は垂直多関節ロボットを用いて行う。

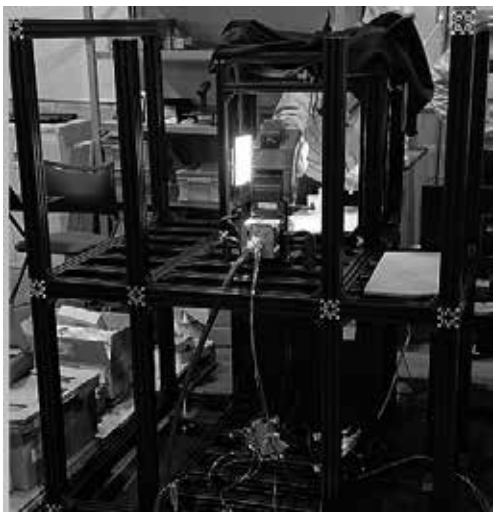

図 7　搬送装置

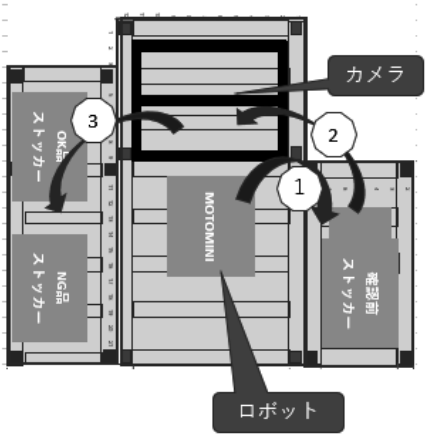

図 8　搬送装置を上から見た図

検査システムが動作する PC とロボットとの通信には産業用ネットワークプロトコルの一つである EtherNet/IP 通信を用いた。

動作としては，図 8 の①～③の工程で自働搬送と振り分けを行う。

①ストッカーより 1 本ずつ切り出された検査対象を取りに行く。

②カメラ下まで検査対象を保持，全周を撮影し検査システムで判定する。

③検査システムが判定した結果により OK 品もしくは NG 品のストッカーへ，検査対象を振り分ける。

3. 結　　言

(1) 金属製品の欠陥を検出するディープラーニングのシステムを開発した。

(2) 欠陥を検出するために，直接光照明で撮影した時の明視野と暗視野の境界付近を活用した。

(3) テストデータに対する正解率について，YOLO と ResNet101 を比較すると，わずかに ResNet101 の方がよかった。複数フレームを活用する方法では差異はあまりなかった。

(4) 検査システムの高速化方法としては，CPU を増やすことが特に効果的であった。

(5) 検査システムを製造現場に導入するため，ロボットを使用した自働化設備を製作している。