

最先端のビジュアル品質検査

インテルにより最適化されたエンドツーエンドのモジュール式
リファレンス・キットを使用した効率的な異常検出

Pratool Bharti インテル コーポレーション AI フレームワーク・エンジニア
Vrushabh Sanghavi インテル コーポレーション AI フレームワーク・エンジニア

異常検出は、時間と労力がかかるため、一般的な産業環境の大量データへの適用は制限されます。人工知能（AI）は、生産性の向上、優れた洞察、ダウンタイムの短縮、優れた製品品質を可能にすることで、産業用のモノのインターネット（IIoT）を変革しています。[異常検出リファレンス・キット](#)（英語）は、AI を活用した高解像度画像のビジュアル品質検査を提供し、生産ラインで生じる不良品のように、まれに発生する異常事象を特定します。概念的には、通常の数値ポイントと統計的に異なる外れ値を検出するため、金融取引の不正検出、製造業の品質監視、スパムメールのフィルタリング、ネットワークの侵入検知などのユースケースにも適用できます。このリファレンス・キットは「そのまま」使用することも、アプリケーションの要件に合わせて変更することもできます。

異常検出にはいくつかの課題があります。生データから表現を抽出するには、特徴量エンジニアリングが必要です。従来のマシンラーニング手法は手動の特徴量エンジニアリングに依存しており、異なる設定に必ずしも一般化できるとは限りません。ラベル付きのトレーニング・データも必要になるため、データ収集とアノテーションのオーバーヘッドが増加します。異常はまれにしか発生しないため、適切なバランスのトレーニング・データを用意することは不可能です。つまり、標準的な分類手法は適していません。異常の性質は偶発的な可能性があり、欠陥はさまざまな予測不能な理由で発生する可能性があるため、異常の種類を予測することが不可能な場合があります。

優れたパフォーマンスを達成しつつこれらの課題を克服するため、異常検出向けの、教師なし、混合方式のエンドツーエンドの調整および推論リファレンス・キットを利用することを提案します。このキットは、教師なし方式で正常なデータから正規性のモデルを学習し、モデルからの逸脱に対し異常としてフラグを立てます。このリファレンス・キットは、[インテルにより最適化されたソフトウェア](#)（英語）によって高速化され、使いやすい[インテル® Transfer Learning Tool](#)（英語）のAPIを使用して構築されています。

MVTec AD – 教師なし異常検出用の包括的な実際のデータセット

[MVTec AD](#)（英語）は、工業検査アプリケーションの異常検出に広く利用されているベンチマーク・データセットです。ボトル、タイル、木材など、15 の異なる物体カテゴリーの 5,000 枚を超える高解像度カラー画像から構成されており、すべてのカテゴリーで合計 60 種類の異常が含まれています（[表 1](#)）。各カテゴリーで、データセットは通常の画像のトレーニング・セットと 6 つの異なる種類の異常を含む画像のテストセットを提供します。異常は、物体にさまざまな変化（切れ目、傷、汚れなど）を加えることにより手動で導入されます。このデータセットは、現実の世界の異常を模倣した高難易度かつ現実的な画像セットを提供するため、異常検出手法のパフォーマンスを評価するのに役立ちます。

	Category	# Train	# Test (good)	# Test (defective)	# Defect groups	# Defect regions	Image side length
Textures	Carpet	280	28	89	5	97	1024
	Grid	264	21	57	5	170	1024
	Leather	245	32	92	5	99	1024
	Tile	230	33	84	5	86	840
	Wood	247	19	60	5	168	1024
Objects	Bottle	209	20	63	3	68	900
	Cable	224	58	92	8	151	1024
	Capsule	219	23	109	5	114	1000
	Hazelnut	391	40	70	4	136	1024
	Metal Nut	220	22	93	4	132	700
	Pill	267	26	141	7	245	800
	Screw	320	41	119	5	135	1024
	Toothbrush	60	12	30	1	66	1024
	Transistor	213	60	40	4	44	1024
	Zipper	240	32	119	7	177	1024
	Total	3629	467	1258	73	1888	-

表 1. MVTec AD データセットの概要。カテゴリーごとに、トレーニング・イメージとテストイメージの数、テストイメージに含まれる欠陥に関する情報が示される（[出典](#)（英語））。

ビジュアル品質検査のパイプライン

このリファレンスは、分布外検出アプローチである[深層特徴モデリング](#)（英語）を使用しています。モデルは、トレーニング・データと異なる入力を検出すると、その入力を異常としてマークします。ビジュアル品質検査パイプラインには、データの前処理、特徴抽出、良画像のトレーニング、良および不良テスト画像の推論、評価メトリックの計算が含まれます。

データの前処理と特徴抽出

MVTec AD はトレーニング・データセットとテスト・データセットにすでに分割されているため、画像を正規化してサイズを変更し、特徴抽出モデルと互換性があるようにすることから始めます。すべての画像のサイズを [224, 224, 3] に変更し、ImageNet データセットの平均と標準偏差を使用して正規化します。特徴抽出では、畳み込みニューラル・ネットワークまたはその他のディープラーニング・アーキテクチャーを使用して、入力画像から高レベルの特徴を抽出します。

異常検出のトレーニング

このステップでは、抽出した特徴の正常パターンと異常パターンを区別する方法を学習して、異常を特定するようにモデルをトレーニングします（**図 1**）。ビジョンサブタスクのモデリングには 3 つのオプションがあります。

1. 事前トレーニング済みバックボーンは、ImageNet のような大規模なビジュアル・データセットでトレーニングされた ResNet-51 v1.5 を使用します。
2. シyam・ネットワーク・ベースの対照学習手法は、ラベルを使用しないでデータセットの意味のある表現を学習します。[SimSiam](#)（英語）自己教師あり学習には、1 つの基礎となる画像から 2 つの異なる拡張画像を生成できるデータローダーが必要です。目標は、両方の画像で同じ特徴を生成するようにネットワークをトレーニングすることです。ResNet モデルをバックボーンとして使用し、拡張データセットで調整して、より近い特徴の埋め込みにします。
3. SimSiam に似た、[CutPaste](#)（英語）と呼ばれる対照学習手法は、トレーニング中に使用する拡張が異なります。ResNet モデルをバックボーンとして使用し、画像パッチを切り取って大きな画像のランダムな場所に貼り付けるデータ拡張方法を適用した後、調整することにより、異常データを必要とすることなく、ハイパフォーマンスな異常検出のモデルを構築します。

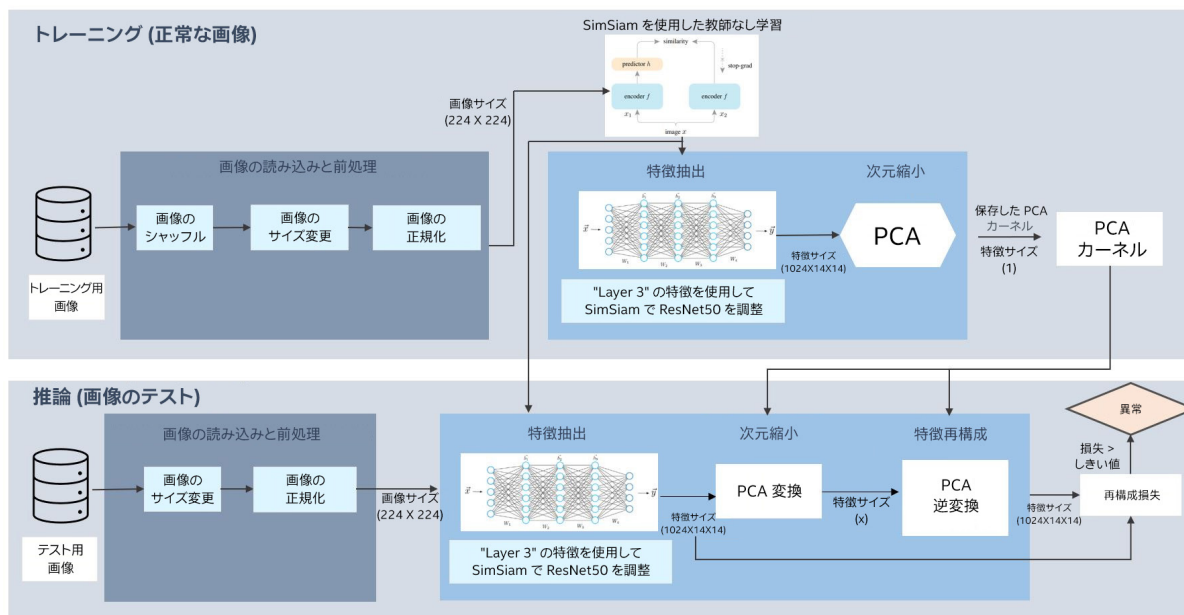


図 1. ビジュアル品質検査のパイプライン。この図は、SimSiam の自己教師ありトレーニングの例。

トレーニング段階では正常なデータのみ使用されることに注意してください。画像はロード後に、シャッフルされ、サイズが変更され、正規化されます。次に、3 つの転移学習手法のうち 1 つを使用してモデルを調整し、中間層から識別特徴を抽出します。主成分分析 (PCA) は、99% の分散を維持して次元を削減するために使用されます。行列の特異点やランク不足の発生を防ぐには、この DNN の中間特徴の前処理が必要です。

推論と評価のメトリック

推論中、テスト画像からの特徴は以前と同じネットワークを通じて生成されます。次に、以前にトレーニングした PCA カーネルを使用して PCA を実行し、逆変換を適用してオリジナルの特徴を再作成し、オリジナルの特徴ベクトルと対応する縮小埋め込みの事前画像との差のノルムである特徴再構成誤差スコアを生成します。異常を含む画像では、特徴が正常なトレーニング・セットの分布外にあるため、オリジナルの特徴を再構成する際に大きな誤差が発生します。異常な画像と正常な画像を区別する際のこれらのスコアの有効性は、受信者動作特性 (ROC) 曲線をプロットすることにより評価されます。この曲線は、分類スコアのしきい値を変化させた場合の、分類器の真陽性率と偽陽性率のプロットです。ROC 曲線下の面積 (AUROC) メトリックは、この曲線を 0 から 1 の間で要約します。1 は完璧な分類を示します。

結果

このビジュアル品質検査リファレンス・コードを実行した場合のサンプル出力を表 2 に示します。

Category	Test set (Image count)	AUROC	Accuracy (%)
BOTTLE	83	99.92	98.8
CABLE	150	94.36	88.67
CAPSULE	132	95.33	87.12
CARPET	117	91.65	83.76
GRID	78	86.3	82.05
HAZELNUT	110	99.25	97.27
LEATHER	124	99.9	98.39
METAL_NUT	115	93.3	90.43
PILL	167	96.02	86.83
SCREW	160	83.3	81.88
TILE	117	98.81	99.15
TOOTHBRUSH	42	96.11	88.1
TRANSISTOR	100	96.42	91.0
WOOD	79	99.3	97.47
ZIPPER	151	97.16	90.07

表 2. 異常検出リファレンス・ケースの精度

システム構成 : シングルノード、デュアルソケットの第 4 世代 Intel® Xeon® Platinum 8480+ プロセッサ、ソケット当たり 56 コア、Intel® ターボブースト・テクノロジー有効、Intel® ハイパースレッディング・テクノロジー有効、1024GB メモリー (16x64GB)、メモリー速度 = 4800MT/s、Intel® SSD DC S3710 (SSDSC2BA012T4)、CentOS® 8、BIOS=EGSDCRB.86B.WD.64.2022.29.7.13.1329、CPU Governor=performance、intel-extension-for-pytorch v2.0.0、torch 2.0.0、scikit-learn-intelx v2023.1.1、pandas 2.0.1。精度 = bfloat16、バッチサイズ = 32、事前トレーニング済み Resnet50 v1.50 モデルから特徴を抽出。

まとめ

この記事では、深層特徴抽出と分布外検出を使用する異常検出リファレンス・キットについて説明しました。このキットは、モデルの調整と特徴の抽出に、調整可能な、モジュール式ワークフローを使用します。モデルの調整と特徴の抽出には、Intel® Transfer Learning Tool を使用します。Intel® アーキテクチャーで最適なパフォーマンスを得るため、スクリプトは、PyTorch* 向け Intel® エクステンション、scikit-learn* 向け Intel® エクステンション、および Intel® アドバンスド・マトリクス・エクステンション (Intel® AMX) を使用して第 4 世代 Intel® Xeon® スケーラブル・プロセッサで bfloat16 を実行するオプションが有効になっています。

このアーキテクチャーは、分布外工業異常サンプルに混合方式の、マルチモデル予測を実行する [リファレンス・ユースケース](#) (英語) で構成されています (図 2)。調整モジュールは、Intel® Transfer Learning Tool を使用する、個別の構成可能なモジュール式 [ワークフロー](#) (英語) として利用可能です。モジュール式アーキテクチャーにより生産性が向上し、解決までの時間が短縮されます。転移学習手法は、Intel® Transfer Learning Tool の API をシームレスに使用する、独立したワークフローにより利用可能になります。設定ファイルを使用すると、ユーザーは、大幅なコード変更を行うことなく、パラメーターと設定を変更できます。特徴抽出の事前トレーニング済みモデルと中間層は柔軟に選択できます。リファレンス・キットは、デフォルトの状態ですべてのパフォーマンスを提供する、Intel® により最適化された基本ツールを活用しています。

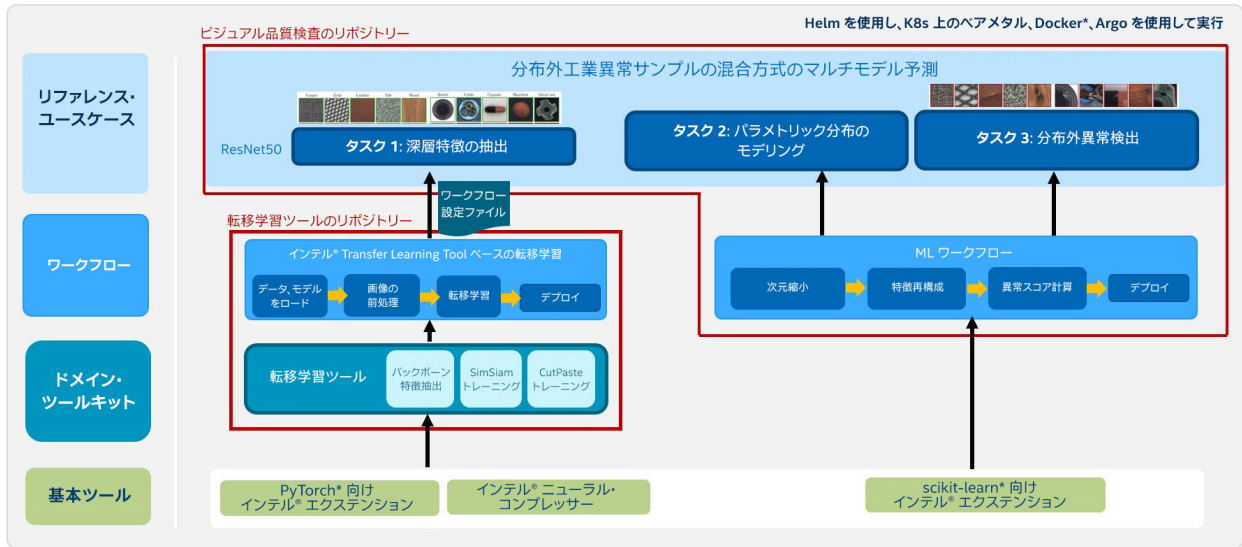
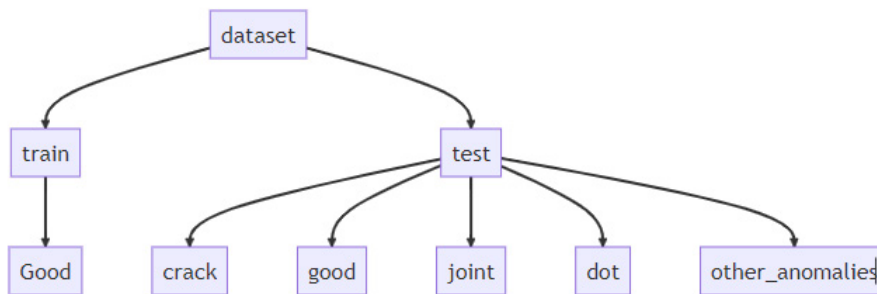


図 2. リファレンス・キットのアーキテクチャー

このリファレンス・キットをカスタマイズするための、調整可能な構成とパラメーターが公開されています。ユーザーは、YAML 設定ファイルを使用して、コードを変更または検索することなく、モデルのトレーニングのハイパーパラメーター、データ型、パス、データセット設定を変更できます。リファレンス・キットは、トレーニングとテスト用の画像をフォルダー構造に配置するだけで、データセットに簡単に導入できます。



このアプローチでは、正常な画像のみトレーニングに使用されることに注意してください。例えば、Kaggle の [大理石表面の異常検出データセット](#) (英語) に対して実行するには、データセットをダウンロードした後、Good フォルダのみ含まれるように train フォルダを更新します。train フォルダ内の異常な画像を含むサブフォルダは、対応する test フォルダに移動するか、削除します。

独自の事前トレーニング済みのカスタマイズされたモデルをプラグインすることもできます。詳細は、[readme](#) (英語) を参照してください。