

# リリースノート v0.2.0

## DictionaryReconstructDetector

正常画像のみから学習するモデル。訓練画像に対して、パッチに区切って、辞書学習を行い再構成誤差を最小化する。新しい画像に対して、得られた辞書による再構成を行い、その誤差の大小によって異常検知を行う。

パラメータ	変数名	型：範囲	既定値	計算時間への影響	精度への影響	備考
アルゴリズムの種類	reconstruct_algorithm_name	str : {'nmf', 'minibatch', 'ksvd'}	'nmf'	-	-	'nmf'が一番高速に動作する
辞書の数	n_components	int : [1, inf]	10	大 -> 大	大 -> ほとんど正常 小 -> ほとんど異常	辞書が多いと複雑なものでも再構成できるようになる。
訓練時のパッチ数	train_size	int : [1, inf]	1000	大 -> 大	大 -> 大	
パッチのサイズ	patch_size	(int, int) : 0~height, 0~width	(8, 8)	大 -> 大	-	(4, 4) or (8,8) or (16,16) が経験的によい。
パッチのステップ幅	extraction_step	int : [1, inf]	1	大 -> 小	大 -> 小	パッチサイズの約数が望ましい。1のとき一番計算負荷がかかるが、もっとも性能がよい
外れ値判定の閾値	deviation_step	float : [0, inf]	2.0	-	大 -> ほとんど正常	再構成誤差に対する閾

					小 -> ほとんど異常	値、1.5 ~ 3程度の値が経験的によい
乱数シード	random_state	int : any	0	-	-	再現性確保のため、固定してある。
反復回数	max_iter	Int : [1, inf]	10	大 -> 大	-	少なすぎると性能が悪くなるが、ある程度大きければ、性能はそれほど変わらない。
最終的な異常度の閾値	thresh_anomaly	float : [0, inf]	None	-	-	最終的に正常化異常かを決める閾値
無視されるべき外周からの幅	ignore_outer	(int, int) : 0~height, 0~width	(0,0)	-	-	外周からの無視する領域の幅 画像の縦, 横のピクセル数までの値にする

### One-class SVM Detector:

大量の正常画像と少量の異常画像からパッチを切り出し、One-class SVMという古典的な教師なし学習を行う。新しい画像に対しては、そのパッチが正常なのか異常なのかを学習されたモデルを使って推論することで、画像全体としての異常検知を行う。

パラメータ	変数名	型 : 範囲	既定値	計算時間への影響	精度への影響	備考
訓練時のパッチ数	train_size	int : [1, inf]	10000	大->大	大->大	

乱数シード	random_state	int : any	0	-	-	再現性確保のため、固定してある。
パッチのサイズ	patch_size	(int, int) : 0~height, 0~width	(4, 4)	大->大	-	(4, 4) or (8,8) or (16,16) が経験的によい。
パッチのステップ幅	extraction_step	int : [1, inf]	2	大 -> 小	大 -> 小	パッチサイズの約数が望ましい。1のとき一番計算負荷がかかるが、もっとも性能がよい
異常部分の拡大率	dilate_omaly	(int, int) : 0~height, 0~width	(1, 1)	-	-	大きくすると、小さい異常部分を検出できるようになる。ただし過検出も増える
最終的な異常度の閾値	thresh_omaly	Float : [0, inf]	0.0	-	-	最終的に正常化異常かを決める閾値
無視されるべき外周からの幅	ignore_outer	(int, int) : 0~height, 0~width	(0,0)	-	-	外周からの無視する領域の幅 画像の縦, 横のピクセル数までの値にする
異常とみなせるパッチの割合	nu	Float : [0, 0.5]	0.01	-	大 -> ほとんど異常	学習に使われた異常パッチの割合が望ましい

## IsolationForestDetector

大量の正常画像と少量の異常画像からパッチを切り出し、Isolation Forestという古典的な教師なし学習を行う。新しい画像に対しては、そのパッチが正常なのか異常なのかを学習されたモデルを使って推論することで、画像全体としての異常検知を行う。

**How to import** `dfcddetr.image.IsolationForestDetector`

**Parameters:**

パラメータ	変数名	型：範囲	既定値	計算時間への影響	精度への影響	備考
訓練時のパッチ数	train_size	int : [1, inf]	10000	大->大	大->大	
乱数シード	random_state	int : any	0	-	-	再現性確保のため、固定してある。
パッチのサイズ	patch_size	(int, int) : 0~height, 0~width	(4, 4)	大->大	-	(4, 4) or (8,8) or (16,16) が経験的によい。
パッチのステップ幅	extraction_step	int : [1, inf]	2	大 -> 小	大 -> 小	パッチサイズの約数が望ましい。1のとき一番計算負荷がかかるが、もっとも性能がよい。
異常部分の拡大率	dilate_ano maly	(int, int) : 0~height, 0~width	(1, 1)	-	-	大きくすると、小さい異常部分を検出できるようになる。ただし過検出も増える。

最終的な異常度の閾値	thresh_anomaly	Float : [0, inf]	0.0	-	-	最終的に正常か異常かを定める閾値
無視されるべき外周からの幅	ignore_outer	(int, int) : 0~height, 0~width	(0,0)	-	-	外周からの無視する領域の幅 画像の縦, 横のピクセル数までの値にする
決定木の数	n_estimators	Int : [1, inf]	100	大->大	大->大	正常異常の識別に使われる決定木の数
並列度	n_jobs	Int : [1, #cpu_core]	None	大->小	-	並列計算に使うコアの数, デフォルトは1コアのみ
異常とみなせるパッチの割合	contamination	Float : [0, 0.5]	0.01	-	大 -> ほとんど異常	学習に使われた異常パッチの割合が望ましい

## PatchAutoencoderDetector:

正常画像のみから学習するモデル。正常画像からパッチを切り出し、ディープニューラルネットワークを用いたオートエンコーダによる学習を行う。新しい画像に対しては、オートエンコーダによる画像の再構成を行い、その誤差の大小によって異常検知を行う。

**How to import:** `dfcddetr.image.PatchAutoencoderDetector`

**Parameters:**

パラメータ	変数名	型 : 範囲	既定値	計算時間への影響	精度への影響	備考
訓練時のパッチ数	train_size	int : [1, inf]	10000	大->大	大->大	

乱数シード	random_state	int : any	0	-	-	再現性確保のため、固定してある。
パッチのサイズ	patch_size	(int, int) : 0~height, 0~width	(4, 4)	大->大	-	(4, 4) or (8,8) or (16,16) が経験的によい。
パッチのステップ幅	extraction_step	int : [1, inf]	2	大 -> 小	大 -> 小	パッチサイズの約数が望ましい。1のとき一番計算負荷がかかるが、もっとも性能がよい。
各層の次元数	encoding_dims	(int, int, ...)	(128, 64, 32)	大->大	大->大	入力のパッチサイズより小さい値。既定値では, 入力層 -> 128 -> 64 -> 32 -> 64 -> 128 -> 出力層のように層が構成される。
学習時の反復回数	epochs	Int: [1, inf]	100	大->大	大->大	
バッチのデータ数	batch_size	int	256	大->大	-	train_size より小さい2の累乗が好ましい。
最終的な異常度の閾値	thresh_anomaly	Float : [0, inf]	0.0	-	-	最終的に正常か異常かを定める閾値。
外れ値判定の	deviation_ste	float : [0,	2.0	-	大 -> ほと	再構成誤差

閾値	p	inf]			んど正常 小 -> ほと んど異常	に対する閾 値、1.5 ~ 3程度の値 が経験的に よい。
無視される べき外周か らの幅	ignore_out er	(int, int) : 0~height, 0~width	(0,0)	-	-	外周からの 無視する領 域の幅 画像の縦, 横のピクセ ル数までの 値にする。