Integration Memo v0.2.0, (6/28, Fri)

Main Changes:

- Add new detector
    a. PatchAutoencoderDetector (DeepLearning)
- Modify parameters
    a. DictionaryReconstructDetector
    b. OneClassSVMDetector
    c. IsolationForestDetector

New Algorithms:

PatchAutoencoderDetector:

How to import: `dfctdetr.image.PatchAutoencoderDetector`
Parameters:

Parameter	type	default	Computation Time Effect	Accuracy Effect	Note
train_size	int	10000	↗	↗	>= 1
random_state	int	0	-	-	
encoding_dims	Tuple[int]	(128, 64, 32)	↗	↗	each element <= the product of patch_size
epochs	int	100	↗	↗	>= 1
batch_size	int	256	↗	?	<= train_size
optimizer	str	‘adam’	-	-	fixed
loss	str	‘binary_cross entropy’	-	-	‘binary_cross entropy’ or ‘mean_squared_error’ or ...
patch_size	(int, int)	(16, 16)	↗	?	

extraction_step	int	8	↘	↘	>= 1
thresh_anomaly	float	0.0	-	?	Larger -> too small defect will be ignored
deviation_step	float	2.0	-	?	>= 0 Larger -> almost will be OK, Smaller -> almost will be NG
ignore_outer	(int, int)	(0, 0)	-	?	Ignore distance from outer edge

Old Algorithms (some parameters changed)

DictionaryReconstructDetector:

How to import `dfctdetr.image.DictionaryReconstructDetector`
Parameters:

Parameter	type	default	Computation Time Effect	Accuracy Effect	Note
reconstruct_algorithm_name	str	‘nmf’	- (no relation)	-	Choose ‘nmf’, ‘minibatch’, or ‘ksvd’ nmf is very fast
n_components	int	10	↗	↗	>= 1 n_components is too large -> almost will be OK. Too small -> almost will be NG.

train_size	int	1000	↗	↗	>= 1
patch_size	(int, int)	(8, 8)	↗	?	(8,8) or (16,16) is best
extraction_step	int	1	↘	↘	>=1 (Divisor of patch_size)
deviation_step	float	2.0	-	?	>= 0 Larger -> almost will be OK, Smaller -> almost will be NG
random_state	int	0	-	-	
max_iter	int	10	↗	-	>= 1
thresh_anomaly	float	None	-	?	Larger -> too small defect will be ignored
ignore_outer	(int, int)	(0,0)	-	?	Ignore distance from outer edge

One-class SVM:

How to import `dfctdetr.image.PatchOneClassSVMDetector`
Parameters:
Deleted:

- nu_max
- nu_min
- nu_num

Added:

- nu
- delate_anomaly

Current Parameters:

Parameter	type	default	Computation Time Effect	Accuracy Effect	Note
train_size	int	10000	↗	↗	>= 1
random_state	int	0	-	-	
patch_size	(int, int)	(4, 4)	↗	?	
extraction_step	int	2	↘	↘	>= 0
dilate_anomaly	(int, int)	(1, 1)	-	?	>=(1, 1) Larger -> small defect will be detected
thresh_anomaly	float	0.0	-	?	Larger -> too small defect will be ignored
ignore_outer	(int, int)	(0,0)	-	?	Ignore distance from outer edge
nu	float	0.01	-	?	Too big will lead overdetection

IsolationForestDetector

How to import `dfctdetr.image.IsolationForestDetector`
Parameters:
Deleted:

- None

Added:

- Delate_anomaly

Current Parameters:

Parameter	type	default	Computation Time Effect	Accuracy Effect	Note

train_size	int	10000	↗	↗	>= 1
random_state	int	0	-	-	
patch_size	(int, int)	(16, 16)	↗	?	
extraction_step	int	4	↘	↘	>= 0
dilate_anomaly	(int, int)	(1, 1)	-	?	>=(1, 1) Larger -> small defect will be detected
thresh_anomaly	float	0.0	-	?	Larger -> too small defect will be ignored
ignore_outer	(int, int)	None	-	?	Ignore distance from outer edge
n_estimators	int	100	↗	↗	>= 1
n_jobs	int	None	↘	-	Parallel num (depending on the #core of CPU)
contamination	float	0.01	-	?	0 < contamination <= 0.5 Ratio of ng in train data is preferred. Too big will lead overdetection

Note:

Next version0.3.0, supervised learning will be integrated.