

detection_utils.utils.object
_detection_evaluation_test.Weighted
PascalEvaluationTest.test_returns
_correct_metric_values

detection_utils.utils.object
_detection_evaluation_test.Precision
AtRecallEvaluationTest.test_returns
_correct_metric_values

detection_utils.utils.object
_detection_evaluation_test.Weighted
PascalEvaluationTest.test_returns
_correct_metric_values_with_difficult_list

detection_utils.utils.object
_detection_evaluation_test.Precision
AtRecallEvaluationTest.test_returns
_correct_metric_values_with_difficult_list

detection_utils.utils.object
_detection_evaluation_test.Precision
AtRecallEvaluationTest.add_common_detected

```
graph LR; A["detection_utils.utils.object  
_detection_evaluation_test.Weighted  
PascalEvaluationTest.test_returns  
_correct_metric_values"] --> D["detection_utils.utils.object  
_detection_evaluation_test.Precision  
AtRecallEvaluationTest.add_common_detected"]; B["detection_utils.utils.object  
_detection_evaluation_test.Precision  
AtRecallEvaluationTest.test_returns  
_correct_metric_values"] --> D; C["detection_utils.utils.object  
_detection_evaluation_test.Weighted  
PascalEvaluationTest.test_returns  
_correct_metric_values_with_difficult_list"] --> D; E["detection_utils.utils.object  
_detection_evaluation_test.Precision  
AtRecallEvaluationTest.test_returns  
_correct_metric_values_with_difficult_list"] --> D;
```