1　タスクコネクション（分類→領域検出）

これまで（1）（2）のチュートリアルでは、それぞれの AI タスクについて触れてみました。MENOU の AI を最大限に活用するには、これらの AI タスクを組み合わせることが大切です。

「AI の組み合わせ」と聞いても難しく考える必要はありません。

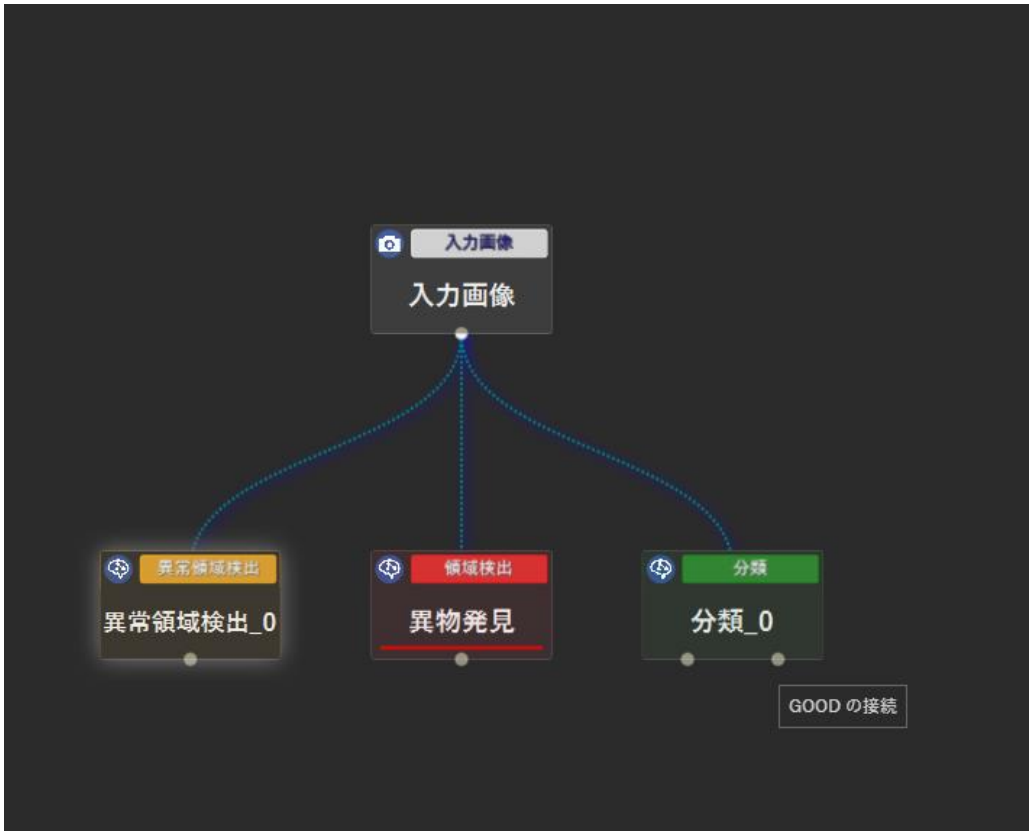
例えば、
角のキズを見つけたい　→　「角を見つけて」＋「そこのキズを見つける」
A 部品の異物を見つけたい　→　「（他の部品ではなく）A 部品を見つけて」＋「異物を見つける」
異物を種類ごとに見つけたい　→　「異物を見つけて」＋「種類に分ける」

などと考えたことをタスクに分けるだけです。
やってみましょう。

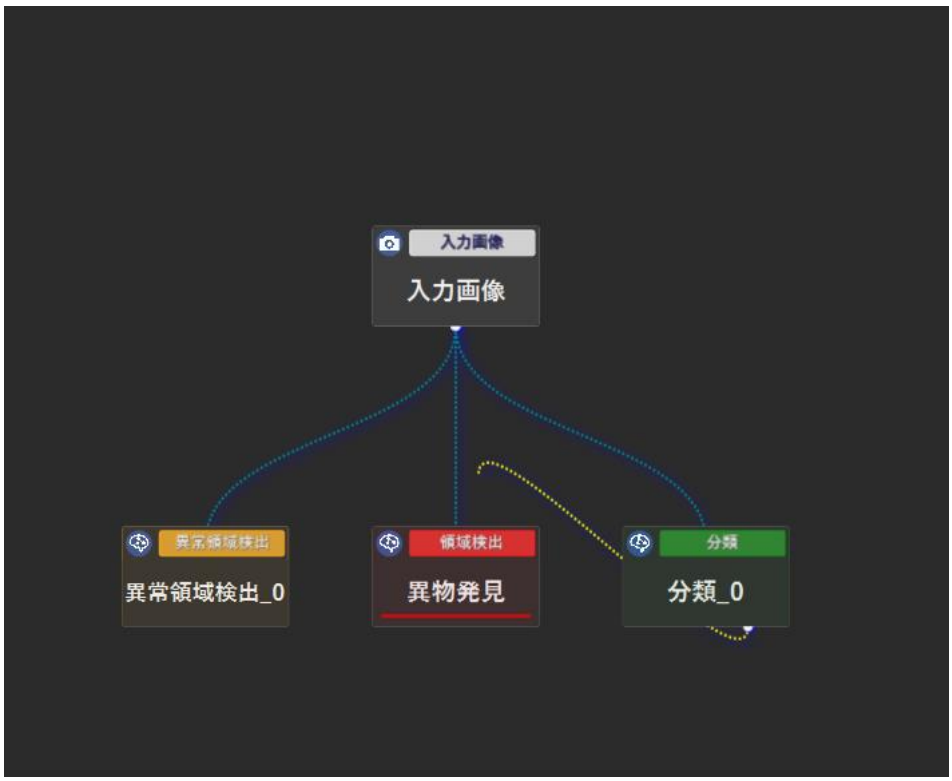
1.1　タスクコネクション

「タスクコネクション」タブをクリックし、タスクコネクションの画面にします。
今回はこのようにはみ出したり、見切れた画像は無効とし、それ以外の画像から異物を検出するという課題を考えます。

ちなみに、コンベヤー上のワークを検査するときなど、対象物がずれていることがあったり、2 つのワークが視野に入っていたりすることもあるので、最初に正しく見切れずに撮像できているかどうか判別するのは良く行われる手法です。



試しに、分類タスクの下に〇が 2 つありますが「GOOD の接続」という方を選んでみてください。



一度クリックすると線が延びるので、「異物発見」ボックスの上部まで引っ張ってください。

ボックスの上面が入力になりますので、そこでもう一度クリックすると緑の「分類」と「異物発見」が直列に繋がりました。

このように、各タスクの箱の上がタスクの入力側、下が出力側となっており、処理は上から下へと流れるフローチャートと同じだと考えてください。

分類タスクには、分類されたクラスごとに出力がありますので、どちらを接続するか注意しましょう。

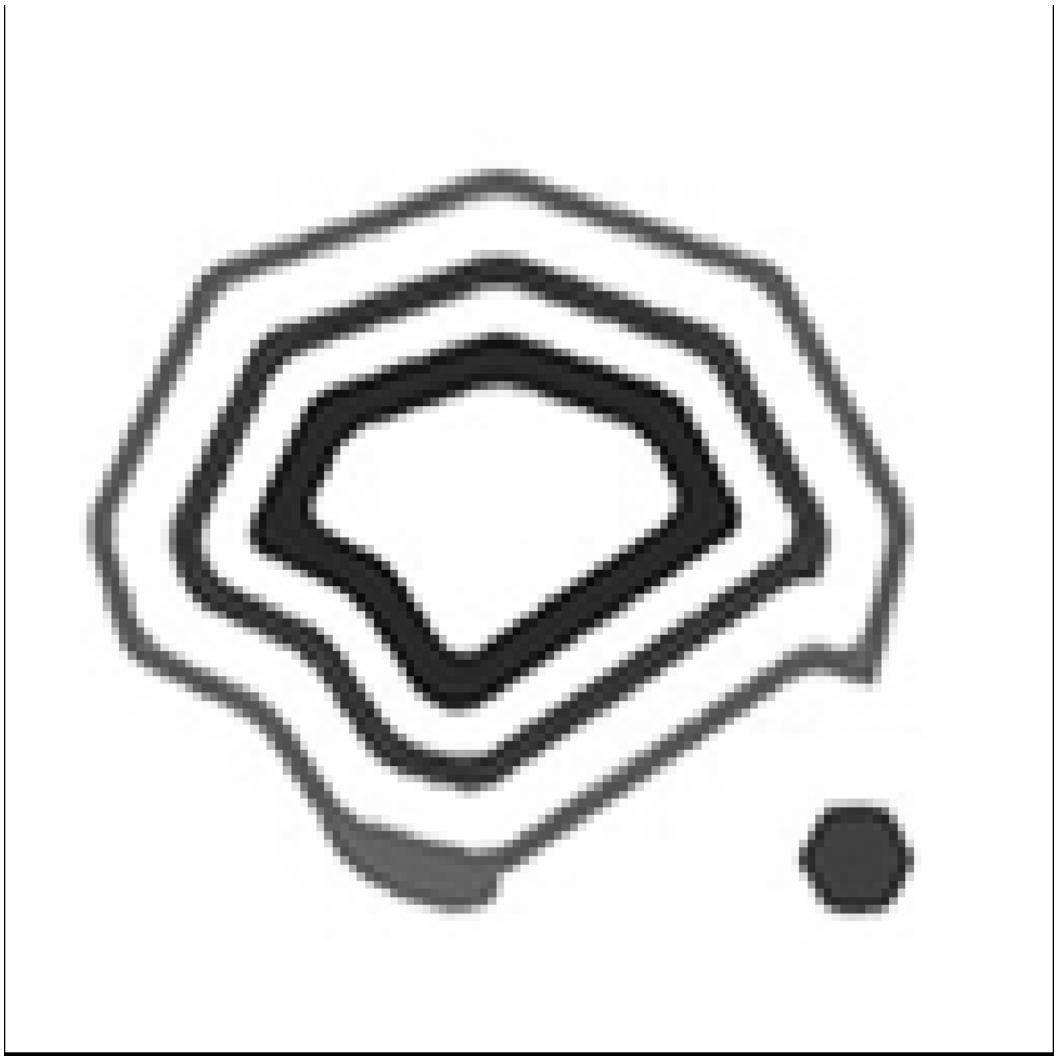
1.2　分類タスクのアノテーション（前処理）

それでは最初のステップである分類タスクを学習させます。

ただし、今回は画像が画角に収まっているかどうかだけを学習させるという「易しい」学習なので、「前処理」によって学習を単純化してみます。
前処理というのは、各タスクのアノテーションを楽にしたり、学習を早く、正確に行うための簡単な画像処理を指します。

今回は見切れた画像かどうかだけを判別したいため、色情報を無視した「グレー画像」で分類しても問題ありません。そのため、右側のタスク設定画面で「グレー画像」のチェックボックスをクリックし、有効にします。
また、画像の解像度も高い必要がありませんので、「サイズ変更」で 1/2 を選ぶことにします。
前処理設定を変更したら、忘れずに「適用」をクリックしてください。

「プレビュー」をクリックすると…

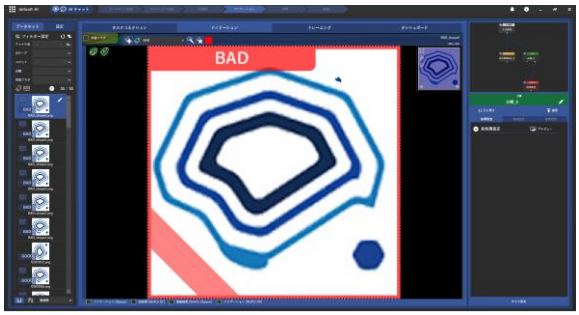


グレースケールの少し粗い画像が表示されます。
AI はこの画像を学習することになります。
見切れているかどうかとは関係のない、色や解像度などの余計な情報がなくなるので、少ない枚数で精度よく学習したり、学習速度が速くなるメリットがあります。
このように、前処理設定を使いこなすことで処理スピードを高めることができます。

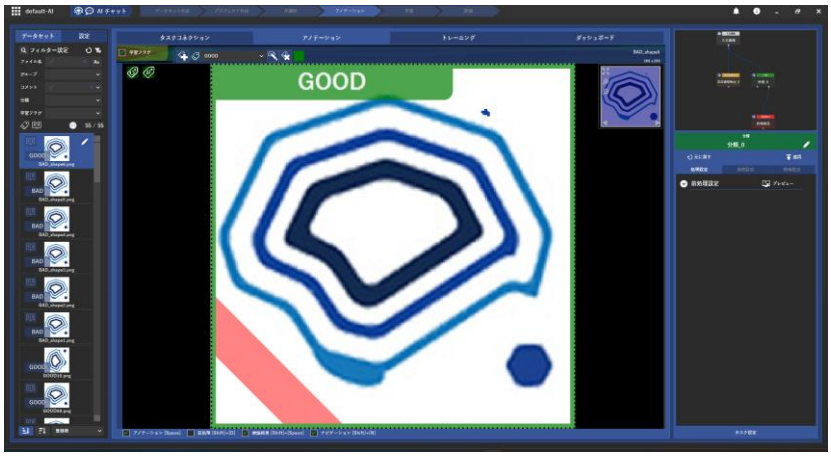
プレビューを見ると、目的を果たしそうなので、アノテーションに進みましょう。

1.3　分類タスクのアノテーション（クラス分け）

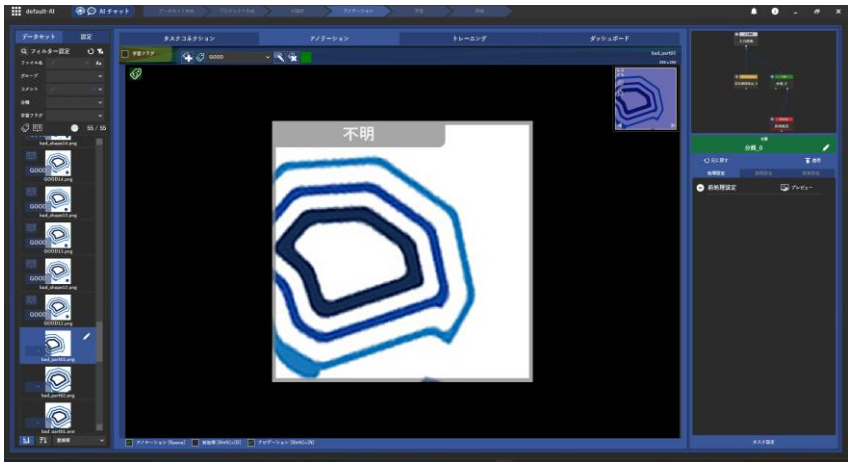
前回は、異物のある画像も、見切れているものも、すべてを「BAD」としてしまいました。
今回は、ロゴ画像が完全に収まっていれば「GOOD」とします。



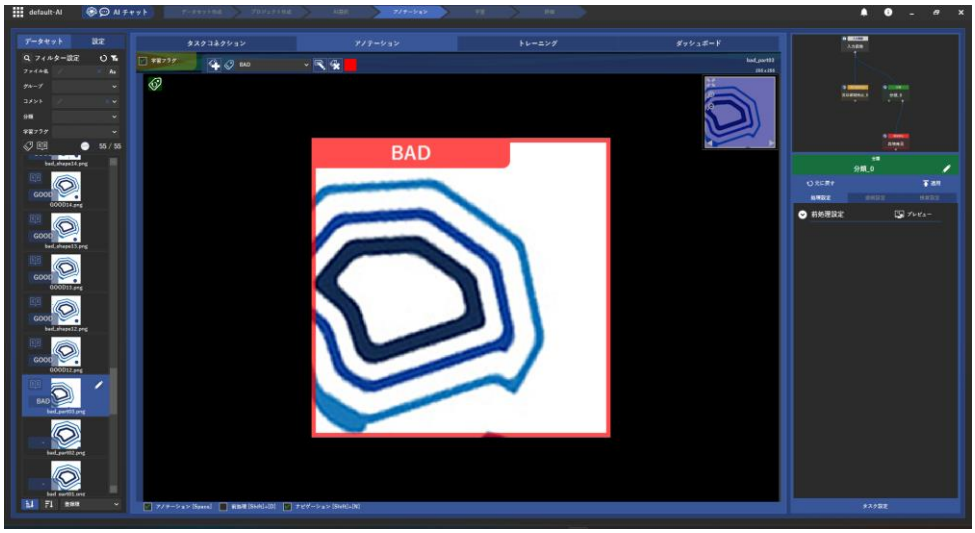
これは画角に収まっているので「GOOD」というクラスに再度割り当ててください。



元々「GOOD」なものには一切問題のない良品もありますので、それらの画像は変える必要はありません。

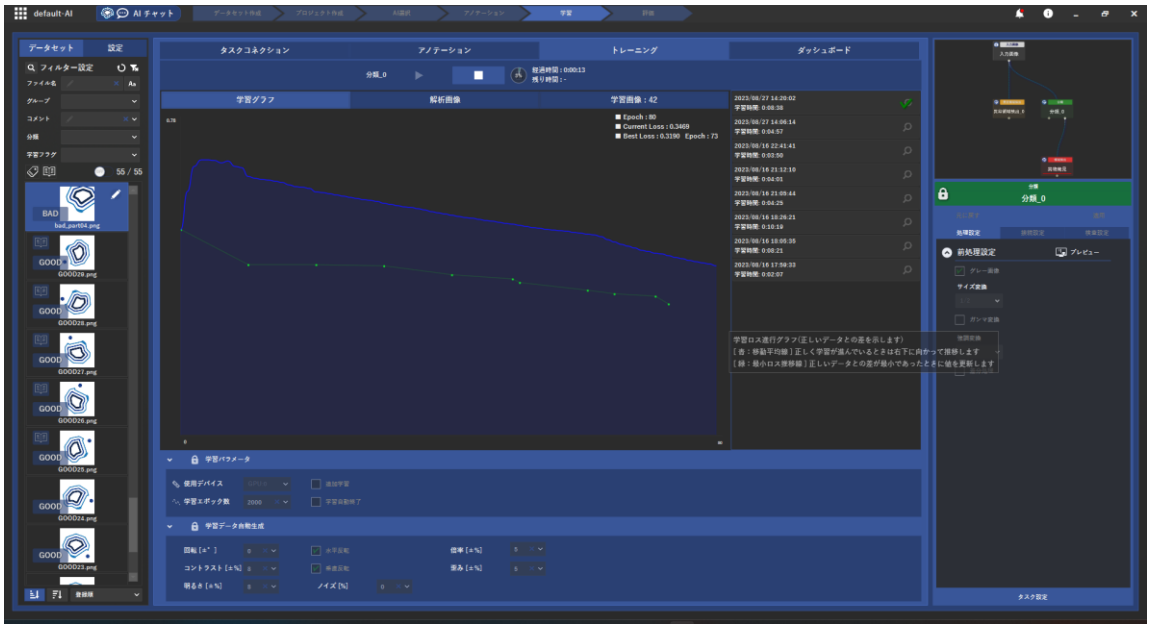


１つひとつ見ていきながら、見切れた画像は「BAD」にします。



それぞれ 7～8 割の画像は「学習フラグ」を ON にしてトレーニングに進みます。

1.4　分類タスクのトレーニング



今回は学習しなおしますので、忘れずに「追加学習」のチェックボックスを外し、学習を行います。
すると、このような結果が得られました。

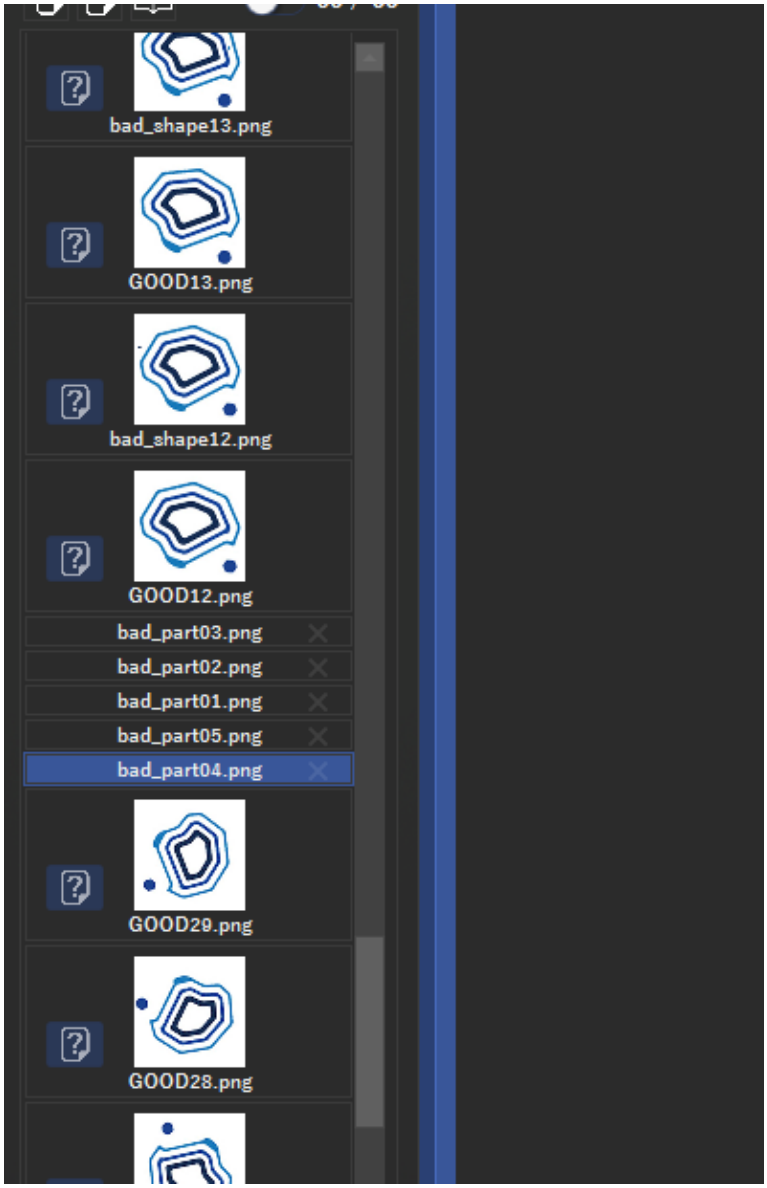
見切れた画像は「BAD」、それ以外の画像は「GOOD」として分類できるようになりました。

1.5　領域検出

つぎは、画角に収まった画像だけで、異物を見つけてみましょう。
タスクコネクションタブをクリックし、「異物発見」のタスクを選びハイライトします。

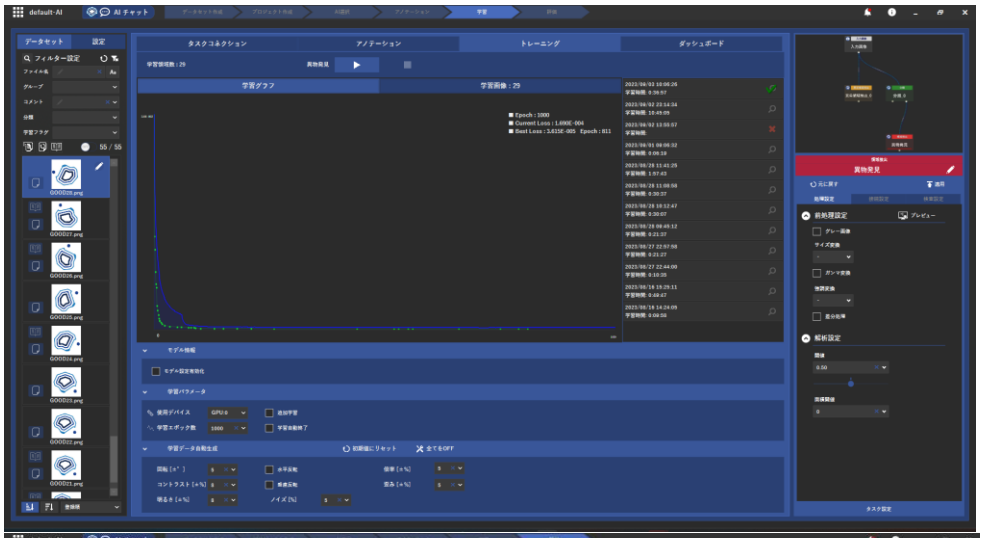
1.6　領域検出アノテーション

左側の画像プレビュー画面を見ると、表示されない画像があります。



「BAD」として分類された画像、つまり見切れいている画像は、このタスクでは処理されません。したがって、アノテーションの必要性もないのです。

あとは、いつも通り学習させます。



どのような画像を学習させると早く AI を作成することができるのか、少しコツが必要かもしれません。最初は「量が質を担保する」という原則で、なるべく多くの画像を学習させるのはとても良い考え方です。しかし、どうしても画像の枚数を稼げないときなどもあるので、どのような画像を学習させると早く AI が学ぶのかを習得すると、効率的で効果的な AI 作成ができるようになります。

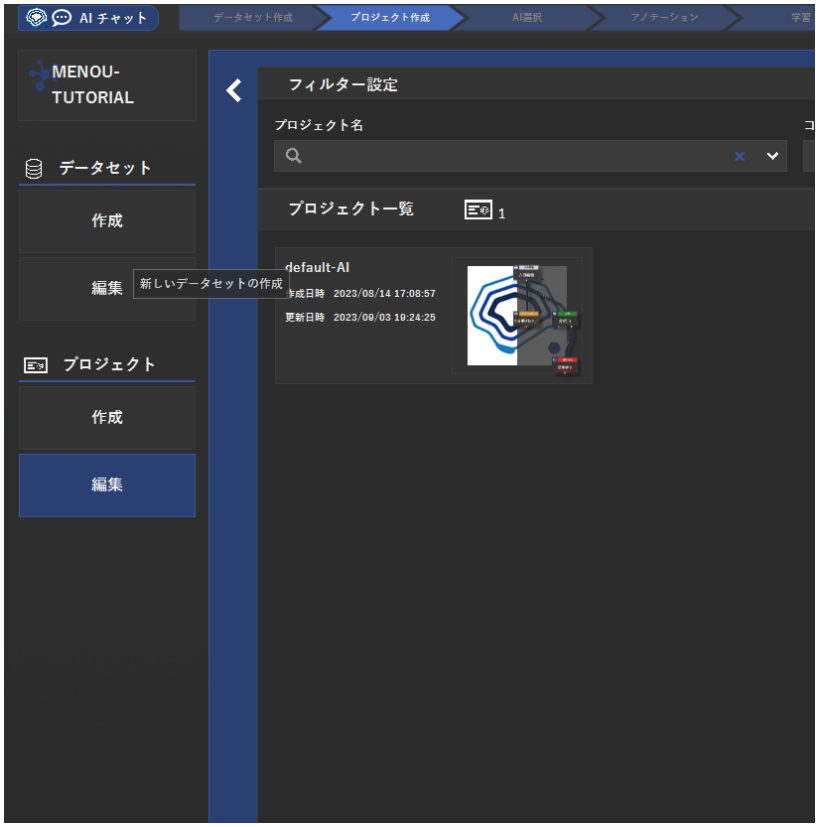
2　タスクコネクション（領域検出→分類）

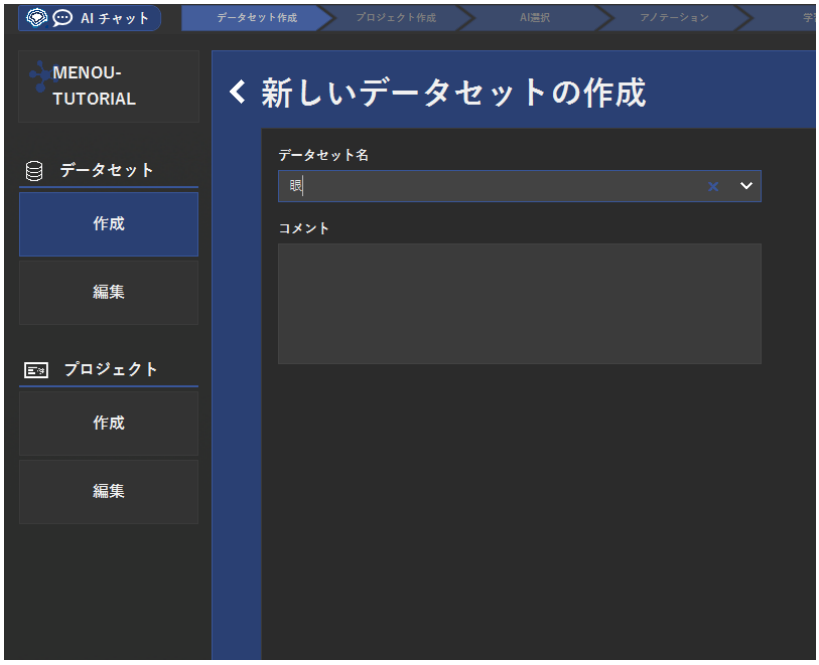
それでは、逆のパターンのタスクコネクションを行ってみましょう。
次の図のように、右下の「眼」部分に白い空隙がある画像を見つけます。

もちろん、この程度のことであれば、全体から白い部分を見つけ出すことも十分に可能ですが、今回は演習のために、右下の眼を探し、次に空隙を探し出すことを行います。

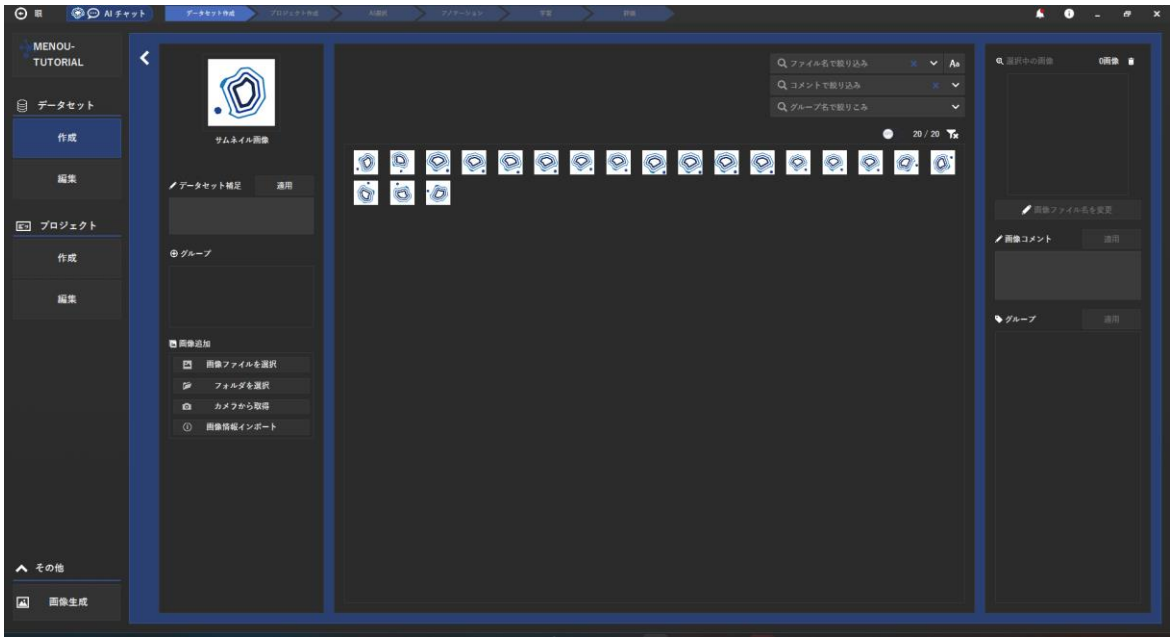
2.1　データセット作成

MENOU_LOGOS_EYE20.zip　をこちらのリンクからダウンロードして、新しいデータセットに格納してください。





データセットには「眼」などの名前を付け、作成します。
次にダウンロードした画像をドラッグアンドドロップなどし、追加してください。



また、良品の画像として以前使用した GOOD01.png ~ GOOD30.png の画像もデータセットに加えます。

2.2　プロジェクト作成

データセットを作ったら、プロジェクトも作成しましょう。

2.3　眼の検出（セグメンテーション）

タスクコネクション画面で、このように「眼の検出」というタスクを作成しましょう。

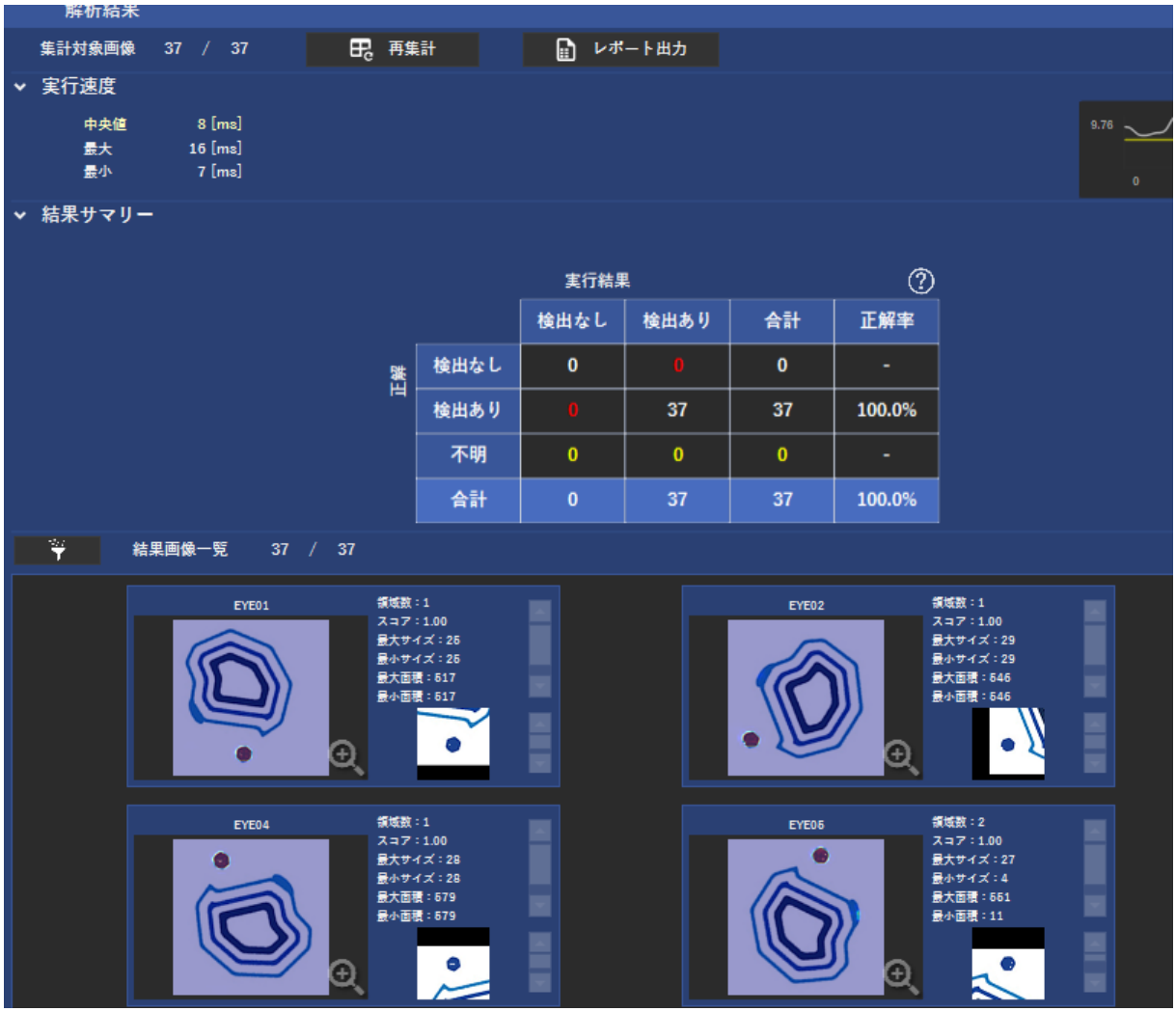
2.4　欠陥の検出（セグメンテーション）

眼というのは、非常に明確な特徴なので、MENOU が学習できる最少画像枚数である 2 枚で学べるかどうか試してみましょう。
前処理設定も前回のようにグレー画像化し、サイズも縮小してみます。


領域検出
眼の検出
元に戻す
適用
処理設定
接続設定
検査設定
前処理設定
プレビュー
グレー画像
サイズ変換
1/4
ガンマ変換
強調変換
-
差分処理
解析設定
閾値
0.50
面積閾値
0


学習グラフ
学習画像：2
Epoch：1000
Current Loss：7.495E-003
Best Loss：1.309E-004 Epoch：835
モデル情報
モデル設定有効化
学習パラメータ
使用デバイス
GPU:0
追加学習
学習エポック数
1000
学習自動終了
学習データ自動生成
初期値にリセット
全てをOFF
回転［±°］
水平反転
倍率［±%］
コントラスト［±%］
垂直反転
歪み［±%］
明るさ［±%］
ノイズ［%］

すると見事にたった 2 枚で学習ができました！

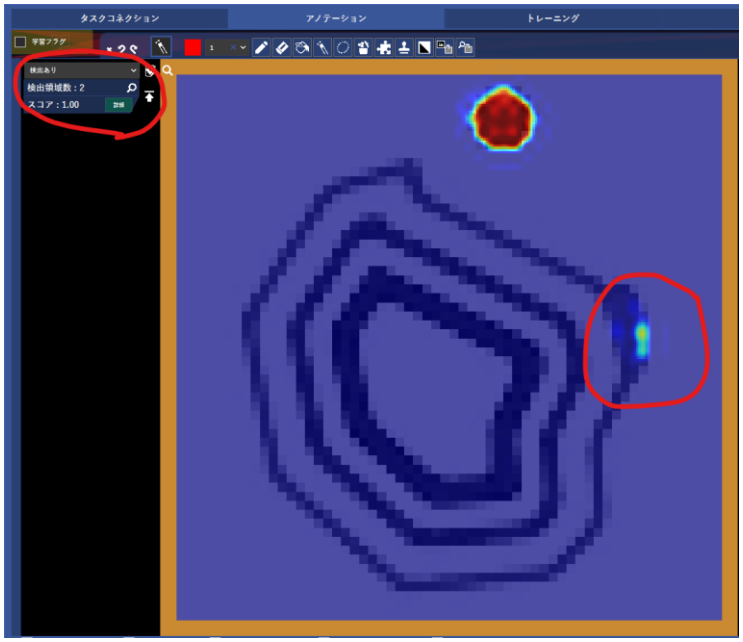


ちゃんと、眼の部分が赤く光っています。

結果画面を見ると、「最大サイズ」「最小サイズ」「最大面積」「最小面積」と表記があります。
これは、見つけ出した特徴のサイズ（長さ）と、面積を指しています。
概ね同じ大きさであり、同じ面積なので、無事に眼の部分を抽出していると言えそうです。
もちろん、一枚一枚見ていきますが、この面積も非常に便利な機能です。なぜなら、時々AI が小さな部分を検出してしまうことがあるからです。

2.5　解析設定

「眼」を探していても、この画像のように似た部分を「眼」らしいと判断してしまうことがあります。



この時、左上の「検出領域数」が 2 カ所となっていることが分かります。この場合、不要な部分が次のタスクに引き継がれてしまわないように、「解析設定」を調節します。
以前、「閾値」にて「眼らしさ」を調整する方法を行いましたが、今回は「面積閾値」を使います。
ほとんどの「眼」の面積は 600 以上だったので、ここは「500」とし、「適用」をクリックします。

すると、検出領域数も 1 つになりました。

面積が 500 未満のものは検出しないというルールが適用されたためです。

2.6　欠陥の発見（セグメンテーション）

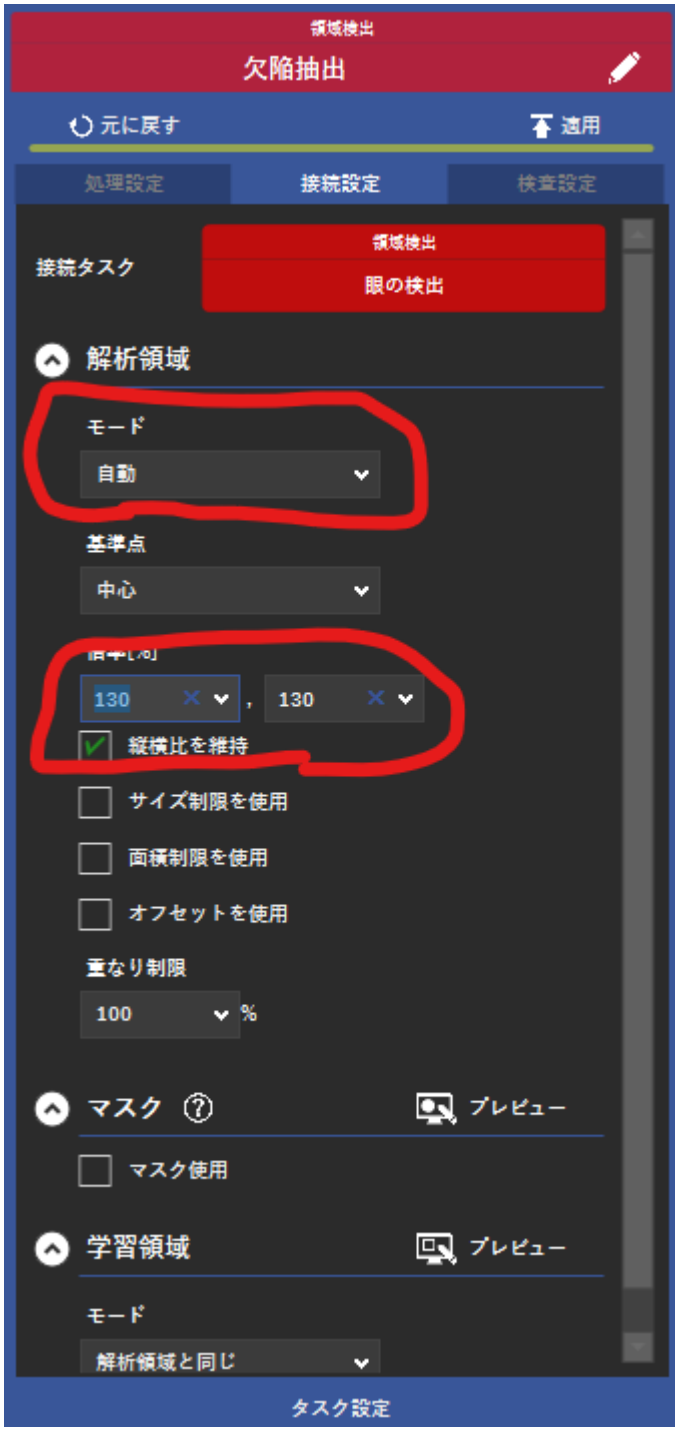
それでは、眼が見つかったのでその中を詳しく探すことにします。
タスクコネクション画面で、もう一つ領域検出タスクを追加し、「欠陥抽出」と名付けます。

そして、先ほど学習させた「眼の検出」と接続します。

2.7　接続設定

「接続設定」を行っていきます。
「接続設定」とは、前の処理（タスク）の結果をどのように引き継ぐのかを指定するパラメータです。

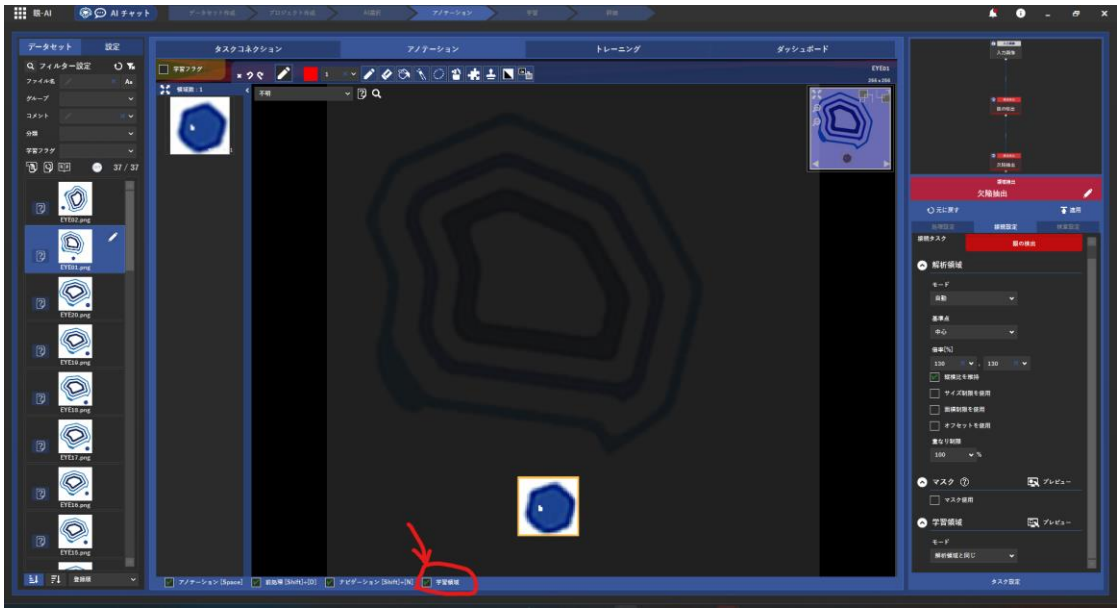
「モード」が 2 つの接続方針を決めます。
今回は、前のタスクで見つけた「眼」を利用した解析を行うため、「自動」を選びます。
（「固定」を選ぶことで、前のタスクとは関係なく決めた領域を解析し、「なし」を選ぶことで前のタスクの結果とは関係なく全体が解析対象になります。）



また「倍率」を指定することで、前のタスクで検出した領域よりも拡大したり、縮小した部分を使うことができます。
今回は、AI が「眼」を抽出した領域よりも少し広めに取ることにし、130% としました。

忘れずに「適用」をクリックします。

一体どの部分が接続されているのか不安になると思うので、「アノテーション」タブで確認することができます。
その際、下のチェックボックスで「学習領域」というチェックボックスを ON にすると、このタスクで学習される画像を確認することができます。



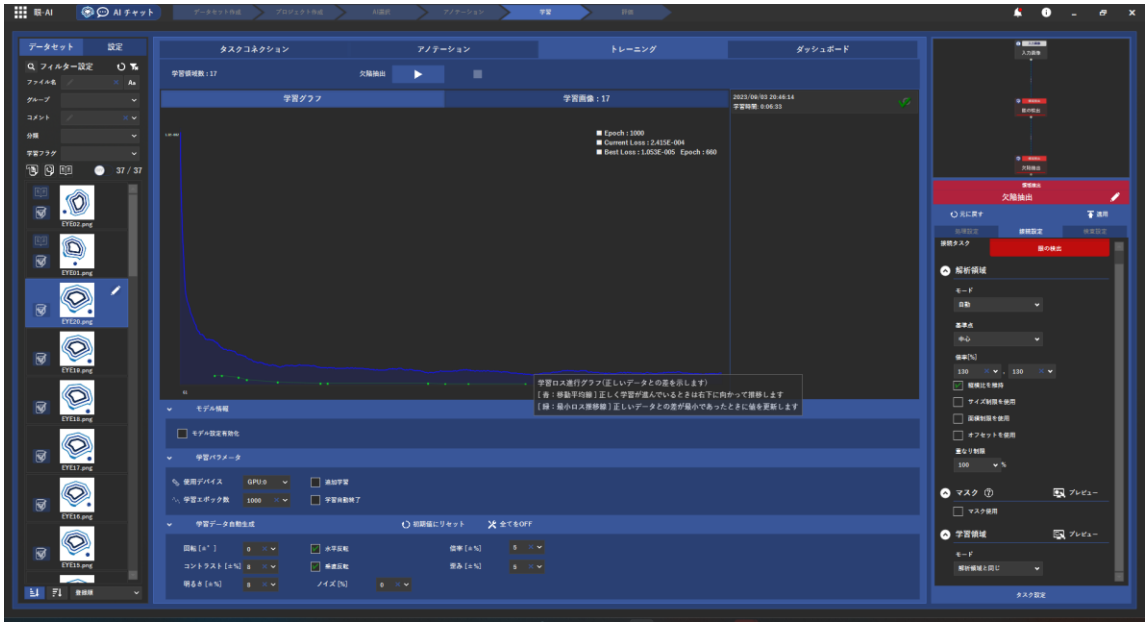
あとは、これまで通りアノテーションするだけです。

2.8　アノテーション

良品（検出なし）として GOOD01.png～GOOD10.png に学習フラグを立てます。
また、不具合品（検出あり）として EYE01.png～EYE05.png の 5 枚だけアノテーションしてみましょう。
今回も、少ない枚数で学習できるかどうか試したいと思います。

2.9　トレーニング

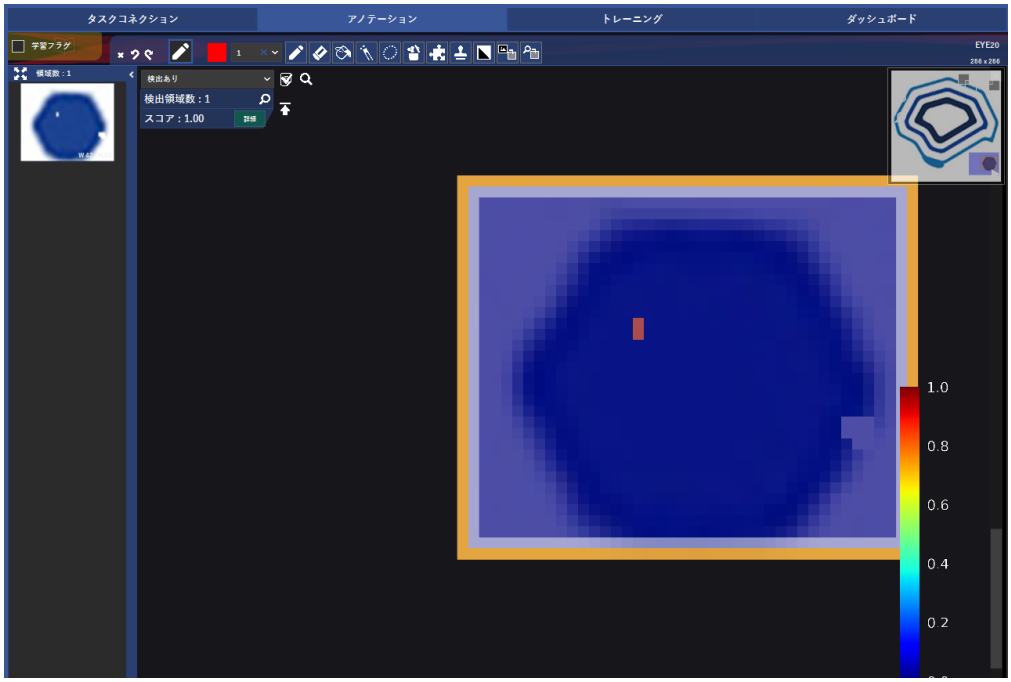
トレーニングはデフォルト設定で行います。



少し閾値設定を修正する必要があるものの、100%の精度を 5 枚だけで達成しました。

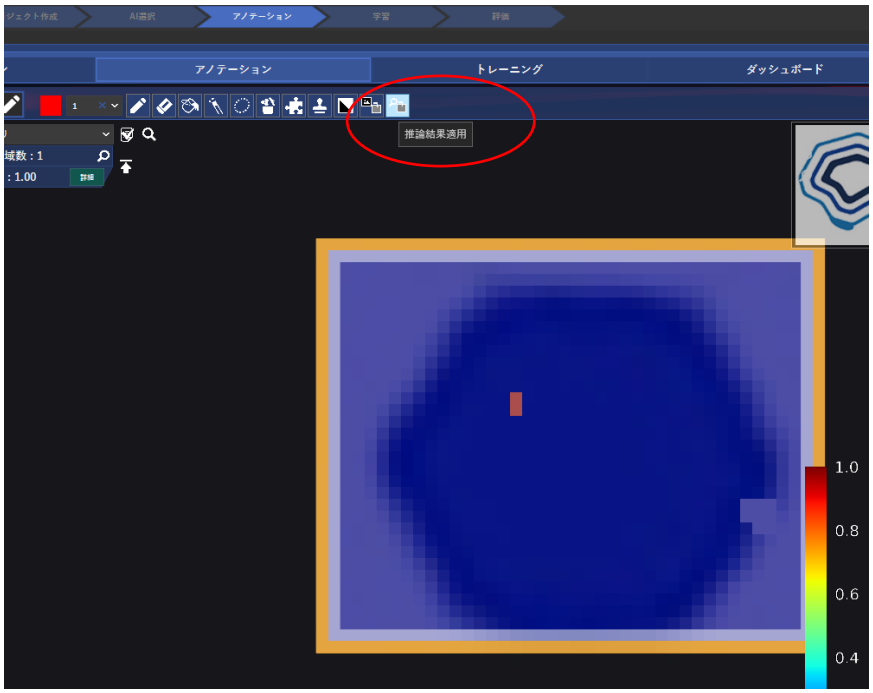
2.10 モデルの修正

精度を達成したからといって、不安は残ります。たった 5 枚だけで全体を判断してもいいのかどうか、確認したいところです。
詳しく 1 枚ずつ見ていきましょう。
ほとんどの画像で正しく検出できていますが、EYE20.png では、外形が欠けているような欠陥を見逃しています。



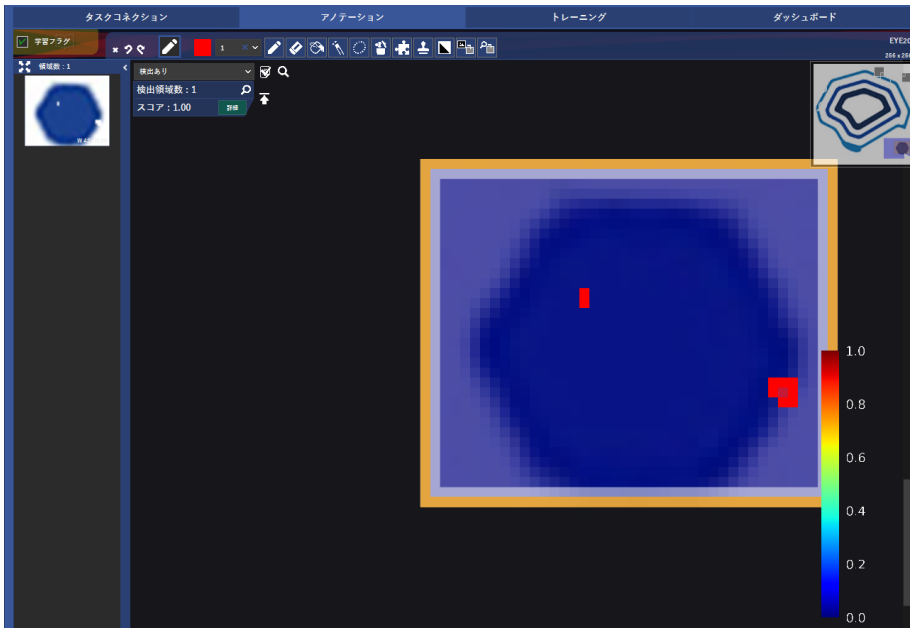
2.11 推論結果適用

この画像にアノテーションをかけます。
その際、「推論結果適用」という機能を使うことで、すでに推論できた部分をアノテーションに加えることができます。



こうすることによって、内部の隙間はアノテーションされ、周辺部分だけの作業になります。
大量の画像をアノテーションする際、段階的にアノテーションすることで大幅に効率化できる機能

ですので、利用しましょう。



EYE17.png も部分的に検出できていないところがあったので、追加のアノテーションを行い、学習フラグも立てます。
このように画像を追加した際は、追加学習をかけるとよいでしょう。

すると、このように学習ができ、スコアのヒストグラムも綺麗に分かれるようになりました。
このように、タスクコネクション機能を使って、非常に少ない枚数で欠陥を抽出することができるようになりました！ぜひ MENOU を使ってもっと難しい検査に取り組んでみてください。