

Neural Network Console クラウド版

スターターガイド

-物体検出編-

ソニーネットワークコミュニケーションズ株式会社

概要

本ドキュメントではNeural Network Console(NNC)で物体検出モデルを作成する一連の流れをまとめました。

サンプルプロジェクトを利用してモデル作成を行う流れになっていますので、Deep Learningモデルの設計ノウハウが無い方でも取り組み易い構成になっています。

また、ウェブにも物体検出の解説動画([実践Deep Learning : 物体検出](#))を準備しておりますので、モデルのネットワーク設計の詳細を理解されたい場合には、こちらも併せてご覧ください。

目次

1 Deep Learningを用いた物体検出とは

2 アカウントサインイン

3 データセット作成

4 物体検出モデル作成

5 物体検出モデル利用

Deep Learningモデルを作るとは

Deep Learningモデルとは分類や予測などを行うためのアルゴリズムです。モデルはネットワークとパラメータに分解できます。よくDeep Learningは脳の神経構造に例えられますが、ネットワークとは回路図で、パラメータとはその上の抵抗値のようなものです。

モデル作成とは、目的に合わせたネットワークを構築し、準備したデータセットを用いてパラメータを最適化する作業です。データセットによるパラメータの最適化を学習と呼び、学習をしてできたモデルを学習済みモデルと呼びます。

Deep Learningの学習



Deep Learningを用いた物体検出

Deep Learningを用いた物体検出では、画像の中にある複数の物体を認識し、おおよその位置がわかります。Deep Learningを用いた画像認識技術には様々なものがあり、より正確な位置や形状を知る場合には領域抽出が適切ですし、画像そのものを分類する場合には画像分類で十分です。様々な画像認識技術の中から解決したい問題やモデル作成の期間・工数に応じて適切な手法を選択する必要があります。

Deep Learningを用いた画像認識技術

画像分類

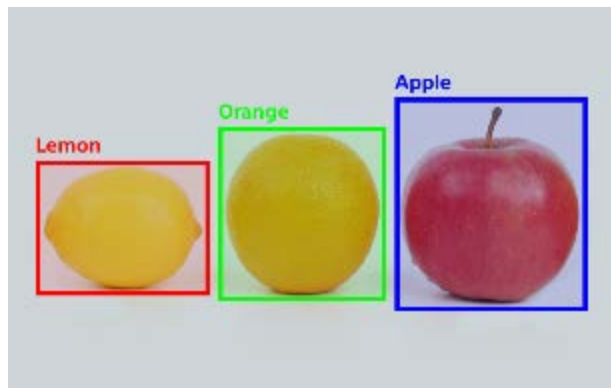
入力画像の種類を予測する



イヌ

物体検出

入力画像の中にある物体の種類とその位置を予測する



領域抽出

ピクセル単位で入力画像を物体の種類を予測する



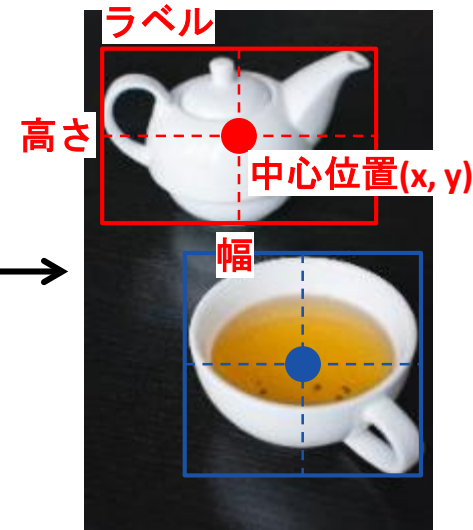
物体検出の基本的な入出力

物体検出では画像データを入力として、検出対象である物体のラベルと物体を囲う長方形(バウンディングボックス)を出力します。バウンディングボックスを定義するために、その中心位置と高さ、幅を記載することが一般的です。

入力画像



出力データ(赤字)



目次

1

Deep Learningを用いた物体検出とは

2

アカウントサインイン

3

データセット作成

4

物体検出モデル作成

5

物体検出モデル利用

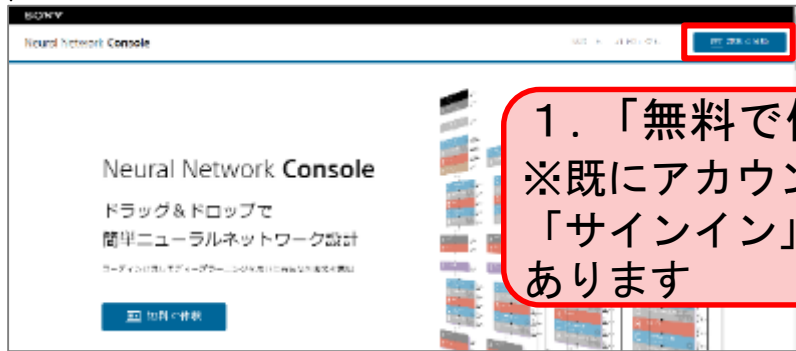
サインインページへの移動

Chromeを利用して、<https://dl.sony.com/ja>に移動します。

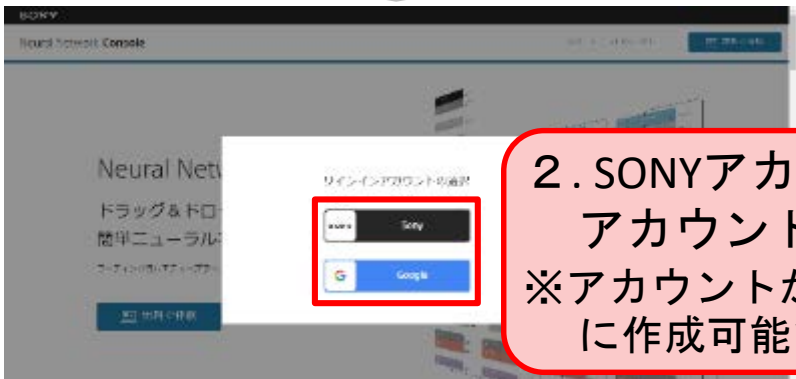
ページの右上にある「無料で体験」をクリックします。

サインインするためのアカウントをSONYアカウントまたはGoogleアカウントから選択します。どちらを選択してもこのドキュメントの内容は進めることが可能です。

サインインページへの移動方法



1. 「無料で体験」をクリック※
※既にアカウント作成済みの場合は「サインイン」と表示される場合があります



2. SONYアカウントまたはGoogleアカウントのどちらかを選択
※アカウントがない場合には選択後に作成可能です

GoogleアカウントとSonyアカウントの違い

Sony アカウント

- ✓ Sonyアカウントを利用しているその他のサービスとのアカウント連携が可能
- ✓ Sonyアカウントを既にお持ちの場合は、アカウント作成不要でNNCの利用が可能

Google アカウント

- ✓ Googleアカウントを利用しているその他のサービスとのアカウント連携が可能
- ✓ Googleアカウントを既にお持ちの場合は、アカウント作成不要でNNCの利用が可能

SONYアカウントでのサインイン

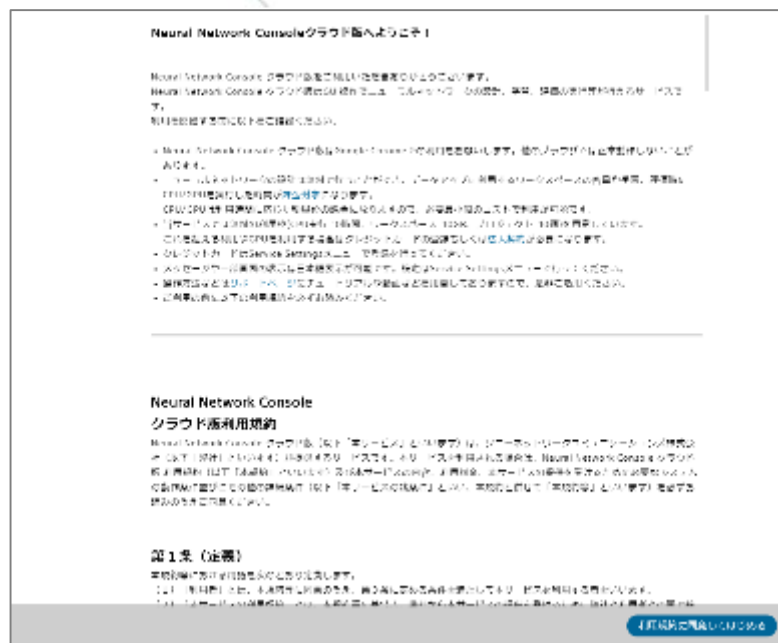
SONYアカウントに登録しているメールアドレス・パスワードを入力し、ログインを行います。
※アカウントをお持ちでない方は、「新しいアカウントの作成」から新規作成を行ってください。
詳細は、Appendixの[SONYアカウントの取得方法](#)に記載があります。

1. メールアドレスの入力



The screenshot shows the Sony login interface. At the top is the Sony logo. Below it, the text 'サインイン' (Sign In) is centered. Underneath, there's a link '一つのアカウントで、Sonyグループの複数サービスへアクセス もっと詳しく' (Access multiple Sony Group services with one account. Learn more). The 'サインインID' (Sign-in ID) field contains 'youraddress@example.com'. Below it is a checkbox 'サインインIDを記憶する' (Remember sign-in ID). The 'パスワード' (Password) field is masked with dots. A blue 'サインイン' (Sign In) button is at the bottom. Below the button is a link 'サインインでお困りですか?' (Having trouble signing in?). At the very bottom is a button '新しいアカウントの作成' (Create new account).

2. 利用規約への同意



The screenshot shows the 'Neural Network Console クラウド版へようこそ！' (Welcome to Neural Network Console Cloud Edition!) page. It contains a welcome message and a list of terms of use. The terms of use are numbered 1 through 5. At the bottom, there is a blue button labeled '利用規約に同意して利用する' (I agree to the terms of use and use the service).

Neural Network Console クラウド版へようこそ！

Neural Network Console クラウド版は、最新のAI技術を提供するためのサービスです。
Neural Network Console クラウド版は、お客様のAI開発をサポートするためのサービスです。
本サービスは、お客様のAI開発をサポートするためのサービスです。

- 1. 本サービスは、お客様のAI開発をサポートするためのサービスです。
- 2. 本サービスは、お客様のAI開発をサポートするためのサービスです。
- 3. 本サービスは、お客様のAI開発をサポートするためのサービスです。
- 4. 本サービスは、お客様のAI開発をサポートするためのサービスです。
- 5. 本サービスは、お客様のAI開発をサポートするためのサービスです。

Neural Network Console
クラウド版利用規約

Neural Network Console クラウド版は、最新のAI技術を提供するためのサービスです。
Neural Network Console クラウド版は、お客様のAI開発をサポートするためのサービスです。
本サービスは、お客様のAI開発をサポートするためのサービスです。

第1条 (定義)

本規約において、次の用語は、次のように定義されます。

1. 「本サービス」とは、本規約に定める範囲で、お客様のAI開発をサポートするためのサービスです。

2. 「お客様のAI開発」とは、お客様のAI開発をサポートするためのサービスです。

4. 利用規約に同意して利用する

Googleアカウントでのサインイン

Googleのメールアドレス・パスワードを入力し、ログインを行います。

1. メールアドレスの入力

 Google にログイン

ログイン

| [sony.com](#) | に移動

メールアドレスまたは電話番号

xxxxx@gmail.com

[メールアドレスを忘れた場合](#)



音声または画面上のテキストを入力

続行するにあたり、Google はあなたの名前、メールアドレス、言語設定、プロフィール写真を [sony.com](#) と共有します。このアプリを使用する前に、[sony.com](#) の [プライバシーポリシー](#) と [利用規約](#) をご確認ください。

アカウントを作成

次へ

2. パスワードの入力



Google にログイン

ようこそ

パスワードを入力

続行するにあたり、Google はあなたの名前、メールアドレス、言語設定、プロフィール写真を [sony.com](#) と共有します。このアプリを使用する前に、[sony.com](#) の[プライバシー ポリシー](#)と[利用規約](#)をご確認ください。

パスワードをお忘れの場合

次へ

3. 利用規約への同意

[illegible]

目次

1 Deep Learningを用いた物体検出とは

2 アカウントサインイン

3 データセット作成

4 物体検出モデル作成

5 物体検出モデル利用

データの有無と本パートの進め方について

本ドキュメントではデータをお持ちの方とそうでない方の両方を読者として想定しています。データがない方はNNCのサンプルデータをご利用いただくか、オープンデータをご利用ください。以下でそれぞれにあった進め方を確認してください。

1. データをお持ちの方

アノテーション(物体のラベルや位置の入力)やデータセット分割などの作業が必要ですので、次頁以降を読み進めてください。ご準備いただいたデータの画像サイズは事前に統一していただく必要はありません。

2. データがなく、サンプルデータを利用される方

データをお持ちでない場合でも、NNCのサンプルデータを用いて本ドキュメントを進めることが可能です。NNCのサンプルデータの詳細は[サンプルデータの説明](#)に記載がありますので、ここから読み進めることも可能です。

3. データがなく、オープンデータを利用される方

データをお持ちでない場合でも、[UCI Machine Learning Repository](#)や[Kaggle](#)等で公開されているウェブ上のオープンデータを利用することで、データセット作成から一連のモデル作成を体験できます。データをダウンロードのうえ、「1. データをすでにお持ちの方」と同様に、次頁以降を読み進めてください。ダウンロードしたデータの画像サイズは事前に統一していただく必要はありません。

データセット作成のステップ

画像データに物体のラベルとバウンディングボックスの対応付けをしたデータセットを作成し、本ドキュメントで用いるモデル構築手法を利用するためのフォーマット変換を行います。

データセットは学習用と検証用に分割したのちに、それぞれNNCにアップロードします。

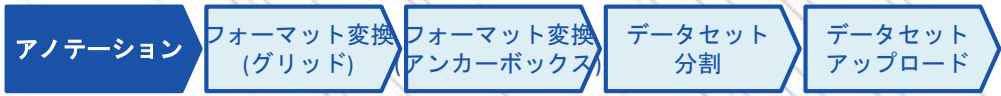
ステップの2～4はNeural Network Libraries(NNabla)のコマンドラインを用いることで、一度に実施することができます。

データセット作成のステップ

ステップ		概要	利用ツール
1	アノテーション作業	画像データと物体のラベル・バウンディングボックスの対応付け(アノテーション)作業を行います	アノテーションツール※
2	フォーマット変換 (グリッドの導入)	グリッドという学習効率化と速度向上のための手法を用いるため、フォーマット変換をします	NNablaのコマンドライン(NNabla_CLI)
3	フォーマット変換 (アンカーボックスの導入)	アンカーボックスという学習効率化と精度向上のための手法を用いるため、フォーマット変換をします	
4	データセット分割	データセットを学習用と検証用の2つに分割します	
5	データセットアップロード	2つのデータセットをNNCにアップロードします	NNC専用のアップローダ

※ アノテーション作業は特化したツールをご利用ください。詳細は[アノテーション作業](#)に記載

アノテーション作業



画像データと物体のラベル・バウンディングボックスの対応付けには、アノテーションに特化したツールをご利用ください。

代表的なアノテーションツールは下表になります。ここでは、次のステップで変換が必要なため [YOLO フォーマット](#) で出力可能なツールのみを記載しています。

データセットの保存はYOLOフォーマットをご利用ください※1。

アノテーションツールのイメージ

アノテーションツールではGUIを通して、バウンディングボックスとラベルを設定することができます



ラベル一覧

- 1. ティーカップ
- 2. マグカップ
- 3. お皿
- 4. ポット
- 5. フォーク
- 6. スプーン

YOLOフォーマットで出力が可能な 代表的なアノテーションツールの例

ツール名	URL
VoTT V1※2	https://github.com/Microsoft/VoTT
labelImg	https://github.com/tzutalin/labelImg

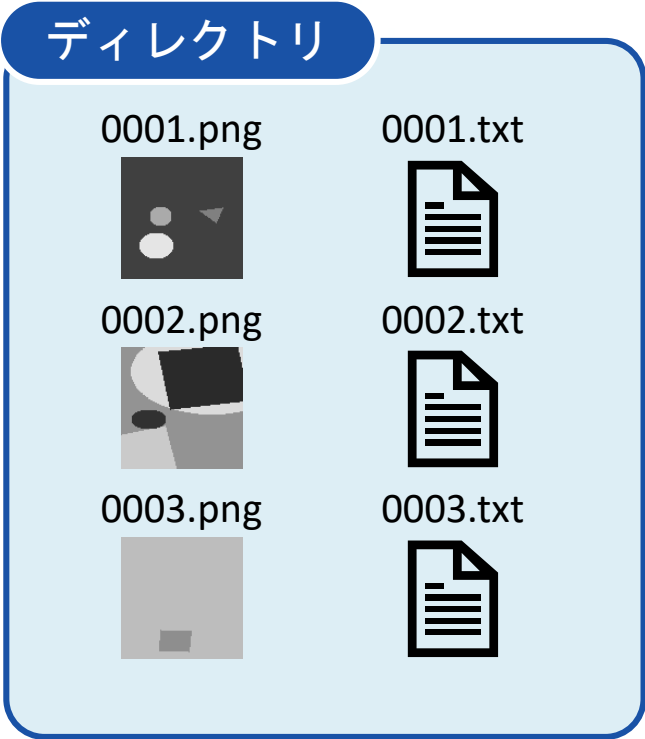
※ 1 YOLOフォーマット以外で出力されるツールをご利用の場合には、プログラミングなどでフォーマットを変換いただく必要があります。

※ 2 VoTT V2ではYOLOフォーマットでの出力機能がないのでご注意ください。

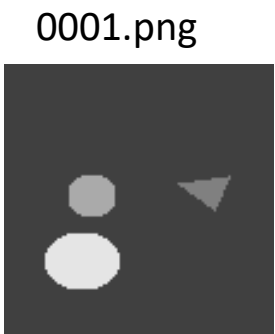
(補足) YOLOフォーマット

YOLOフォーマットでは画像と同じディレクトリに、画像ファイル名と同じ名称のテキストデータを作成し、そこに物体のラベルとバウンディングボックスの情報を保存します。

ディレクトリ構造の例



画像とアノテーションデータの対応



0001.txt

0	0.333	0.488	0.170	0.151
0	0.289	0.730	0.269	0.205
1	0.741	0.477	0.183	0.129

ラベル 中心位置の横、縦 バウンディングボックスの幅、高さ

※それぞれの大きさは画像サイズで規格化

※ 1 ファイル名は連番である必要はありませんが、NNCでは日本語などの2バイト文字に対応していませんので、ご注意ください。
※ 2 NNCではpng、jpg、jpeg、gif、bmp、tifの画像フォーマットに対応しています。

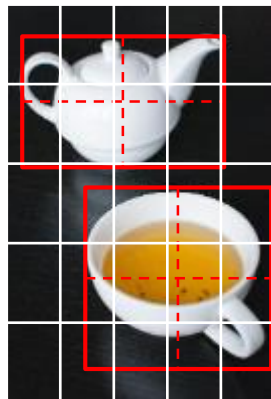
フォーマット変換: グリッド

画像を格子状に分割したグリッドを用いて、物体のおおよその位置を予測することで学習効率とモデル精度を向上できます。

グリッドを用いることでフォーマットが2次元の表データとなり、バウンディングボックスの中心が含まれるセルにグリッド内での相対位置(詳細は[Appendix](#)参照)とバウンディングボックスのサイズが記載されます。

また、NNabla_CLIの変換では、相対位置はグリッドサイズで規格化をし、バウンディングボックスのサイズはグリッドサイズで規格化した後に対数に変換します。

画像サンプル



YOLOフォーマット

ラベル	0	0.44	0.30	0.68	0.76
	1	0.61	0.66	0.70	0.50
	中心位置の横、縦			バウンディングボックスの幅、高さ	

※YOLOフォーマットではそれぞれの大きさは画像サイズで規格化

グリッドによる変換後フォーマット

グリッドごとのラベル	グリッド内での相対位置(横)	グリッド内での相対位置(縦)	バウンディングボックスの幅	バウンディングボックスの高さ
-1 -1 -1 -1 -1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
-1 -1 0 -1 -1	0 0 0.2 0 0	0 0 0.1 0 0	0 0 0.53 0 0	0 0 0.58 0 0
-1 -1 -1 -1 -1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
-1 -1 -1 1 -1	0 0 0 0.05 0	0 0 0 0.3 0	0 0 0 0.54 0	0 0 0 0.40 0
-1 -1 -1 -1 -1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0

※NNabla_CLIによる変換では、物体の中心が含むセルにのみ値が出力され、含まないセルにはラベルは-1、それ以外は0で埋められます。

フォーマット変換: アンカーボックス

アノテーション

フォーマット変換
(グリッド)

フォーマット変換
アンカーボックス

データセット
分割

データセット
アップロード

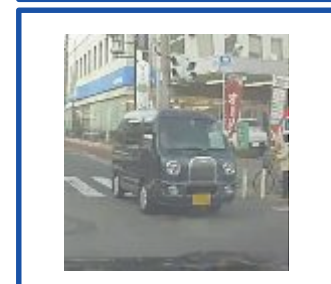
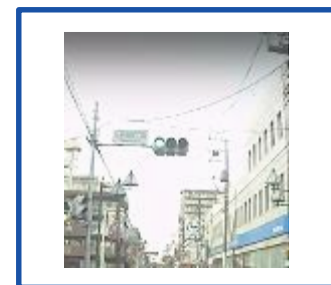
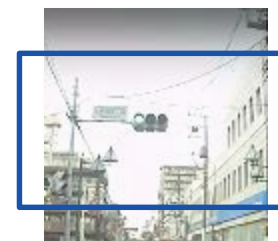
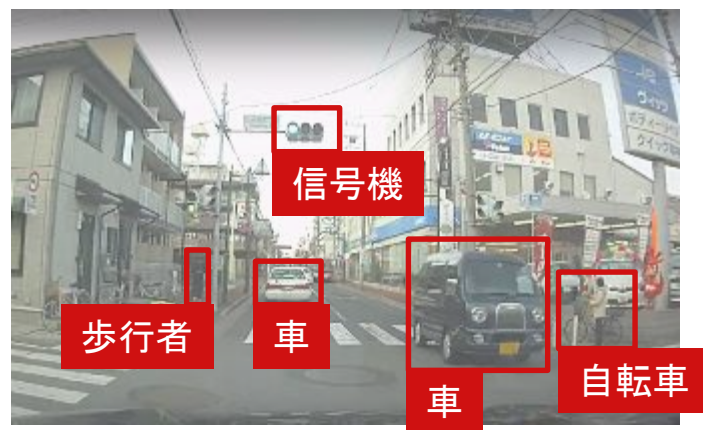
物体検出では大小さまざまなサイズの物体を認識する必要があります。

検出するバウンディングボックスの大きさやアスペクト比に応じて、認識するネットワーク部分を複数用意することで、学習効率や検出速度が向上します。

アンカーボックスと呼ばれる代表的な形状のバウンディングボックスを複数用意し、認識する物体をそれぞれのアンカーボックスに割り当て学習を行います。

アンカーボックスの形状やデータセットの割り振りは、NNabla_CLIの変換では学習データのクラスタリングにより、自動的行われます。

最適なアンカーボックス(例: アンカーボックスを5個準備する場合)



小



大

データセット分割

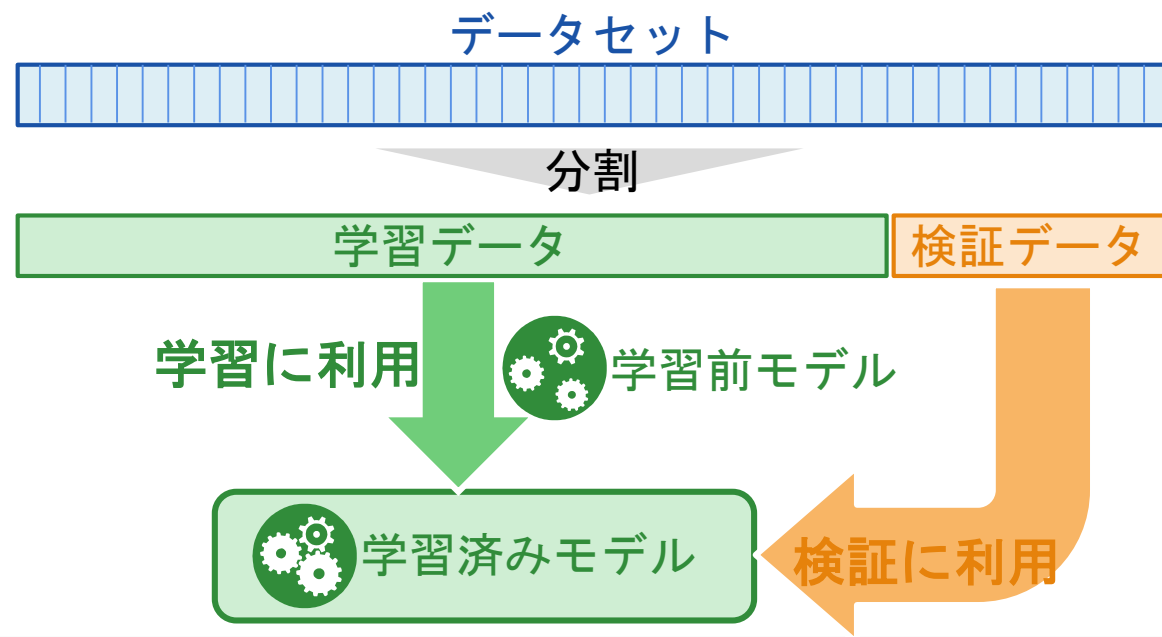
モデル作成には、学習に利用するデータセット（学習データ、Training Data）と、モデル精度を検証するデータセット（検証データ、Validation Data）の2つが必要になります。

作成したモデル精度を正しく検証するためには、学習に利用していないデータを準備する必要があるため、あらかじめデータセットを学習データと検証データに分割しておきます。

このとき、学習データと検証データの分割割合は7:3や8:2が一般的です。

NNabla_CLIの変換では引数にそれぞれの割合を指定することができます。

データセットの件数については、Deep Learningモデルはデータ数が多ければ多いほど、精度が高くなる傾向があります。(参考: [データの重要性](#))



NNabla_CLIの実行

NNabla_CLIを利用することで、グリッドとアンカーボックスを用いるためのフォーマット変換、入力画像サイズの統一とデータセット分割を一度に実行することができます※1。

コマンドライン実行後の出力の詳細は、次頁で実行例とともに記載をしております。

《NNabla_CLIの実行方法》

```
nnabla_cli create_object_detection_dataset -i データのディレクトリ -o 出力ディレクトリ -n 判別クラス数 -c モノクロ(1)/カラー(3) -w 入力画像の幅 -g 入力画像の高さ -a アンカー数 -d グリッドサイズ[16※2] -m 画像修正方法(resize/trimming/padding) -s ランダム分割(true/false) -f1 学習データファイル名 -r1 学習データ割合 -f2 出力検証データ名 -r2 検証データ割合
```

黄字が変更箇所になります

※1 NNabla_CLIはNNablaに付随するコマンドラインです。NNablaのインストールについては、以下のドキュメントをご参照ください。

<http://nnabla.readthedocs.io/en/latest/python/installation.html>

※2 本ドキュメントの通りモデルを作成される場合にはグリッドサイズは16に指定してください。

16から変更される場合には、ご自身でグリッド数を計算し、ネットワーク構造を修正する必要があります。

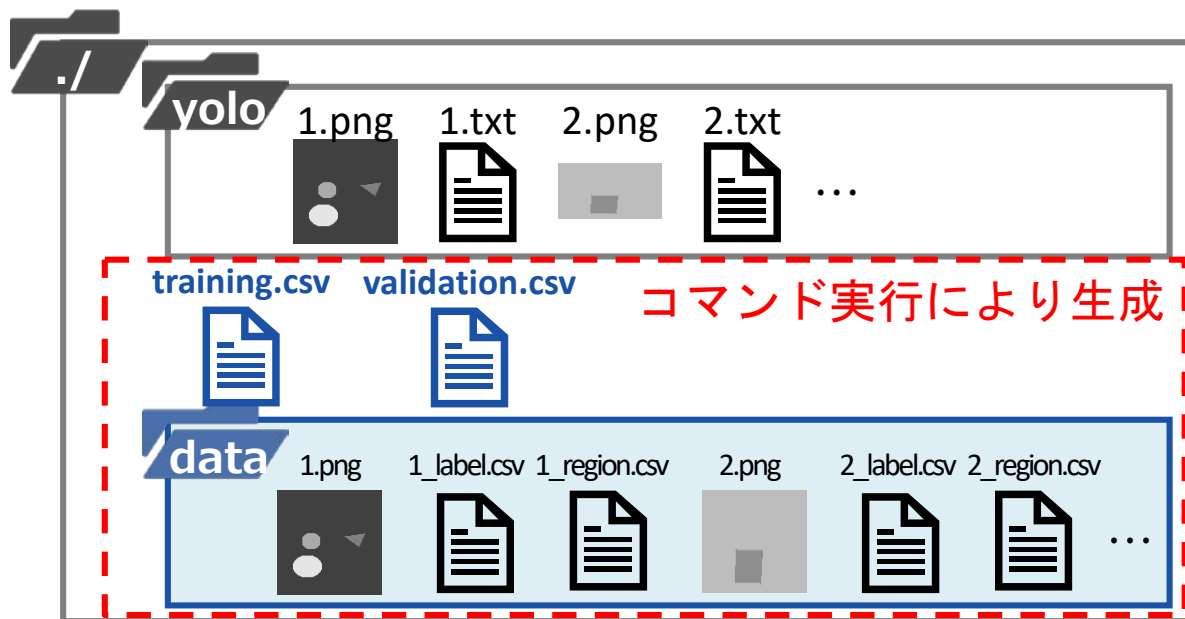
NNabla_CLIの実行例

コマンドラインの実行後、出力ディレクトリに学習用と検証用のデータセットを取りまとめたCSVファイルが出力されます。また、サイズ統一をした入力画像と出力データを記載したCSVファイルがdataディレクトリの中に出力されます。

《コマンドラインの実行例》

```
nnabla_cli create_object_detection_dataset -i ./yolo -o ./ -n 4 -c 1 -w 112 -g 112 -a 5 -d 16 -m resize -s true -f1 training.csv -r1 80 -f2 validation.csv -r2 20
```

《出力ディレクトリの例》



《出力ファイルの説明》

	出力ファイル	説明
1	training.csv	学習用データセットの入出力の対応関係を取りまとめたCSVファイル
2	validation.csv	検証用データセットの入出力の対応関係を取りまとめたCSVファイル
3	data/[画像ファイル名].png	元画像を入力画像サイズにサイズ統一した画像
4	data/[画像ファイル名]_label.csv	画像にある物体のラベル情報を示したcsvファイル
5	data/[画像ファイル名]_region.csv	画像にある物体のバウンディングボックスの情報を示したcsvファイル

データセットのアップロード

アノテーション

フォーマット変換
(グリッド)

フォーマット変換
アンカーボックス

データセット
分割

データセット
アップロード

NNCはクラウドサービスのため、専用のアップローダを利用し、モデルを作成するために必要なデータセットをあらかじめクラウドにアップロードする必要があります。
アップローダ上で準備したCSVファイルを指定し、データをアップロードします。
アップロードは学習用と検証用の2回行う必要があります。

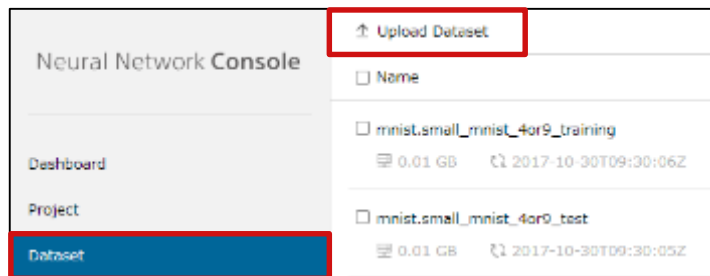
1. アップローダの取得

- ✓ 以下のリンクからアップローダをダウンロード

<https://support.dl.sony.com/docs-ja/データセットアップロードツールのダウンロード/>

2. アップロードキーの取得

- ✓ NNCにログインし、Datasetタブの中のUpload Datasetをクリック

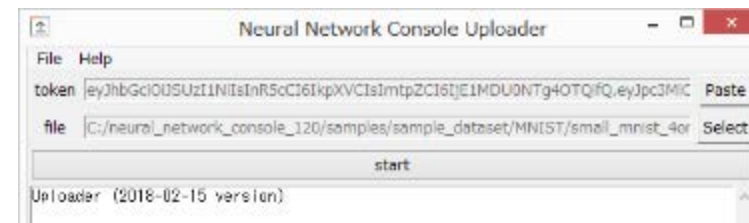


- ✓ ポップアップ画面に表示されるアップロードキーをコピー



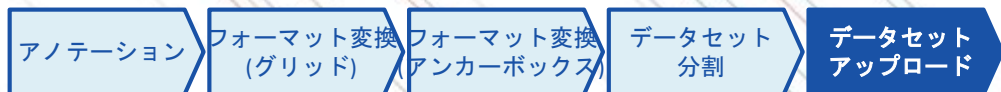
3. アップローダの実行

- ✓ 1で取得したアップローダを起動
- ✓ tokenに2で取得したアップロードキーを貼り付け
- ✓ fileに作成したCSVファイルを指定
※ここでのファイル名がデータセット名になります
- ✓ Startをクリックし、アップロードを実行



※アップロードには時間がかかる場合があります

アップロード先のデータセット確認



アップロード後はDatasetタブの一覧にデータセットが追加されます。

アップロード時のCSVファイルのファイル名がデータ名として一覧に表示され、選択することで中身を確認することができます。表示の際に、画像などはサムネイルの形で確認できます。

データセット一覧に追加されたデータセット

※labelやregionが時系列表示されますが、表データとして取り扱えますので意識する必要はありません。

【参考】NNC上のサンプルデータ(synthetic_image.object_detection)の説明

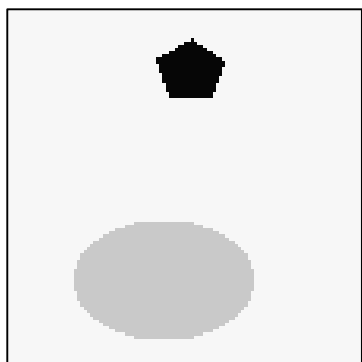
正方形の背景の上に楕円、三角形、四角形、五角形を色、個数、形状をランダムに配置し、それらを検出するサンプルデータです。

入力データが画像データ(image)となり、出力データがアンカーボックスとグリッドごとのラベル情報(label)とバウンディングボックスの情報(region)になります。

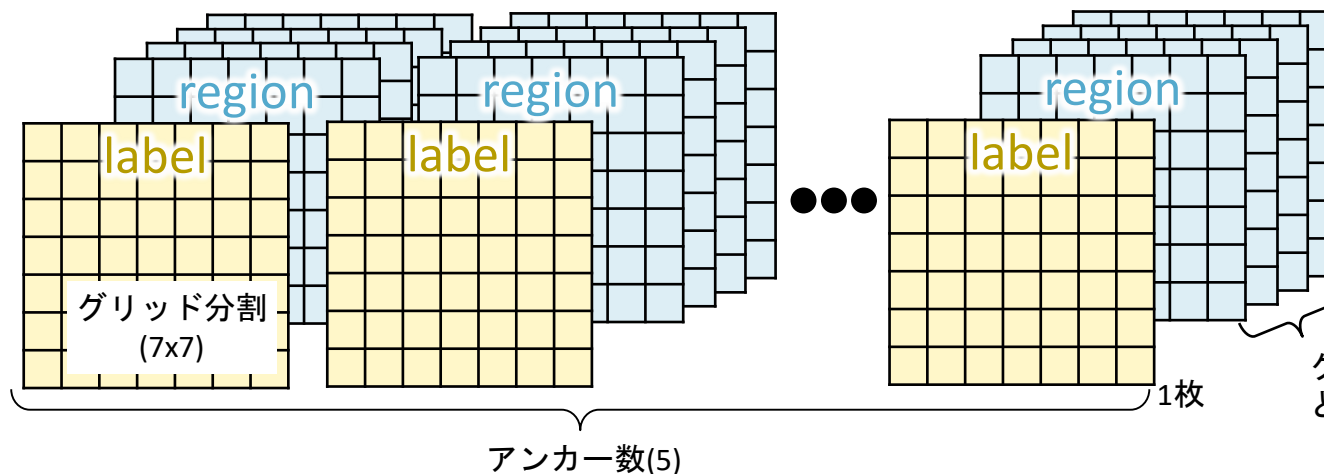
バウンディングボックスの中心が含まれる場所にデータが記載されています。中心が含まれない箇所はcategoryは-1、regionは0で穴埋めがされています。

サンプルデータの入出力

入力: 画像(image)



出力: 物体のラベル情報(label)とバウンディングボックス情報(region)



カテゴリマスタ

index	カテゴリ
0	楕円
1	三角形
2	四角形
3	五角形

グリッド内での相対位置(横、縦)と幅、高さの4枚

目次

1

Deep Learningを用いた物体検出とは

2

アカウントサインイン

3

データセット作成

4

物体検出モデル作成

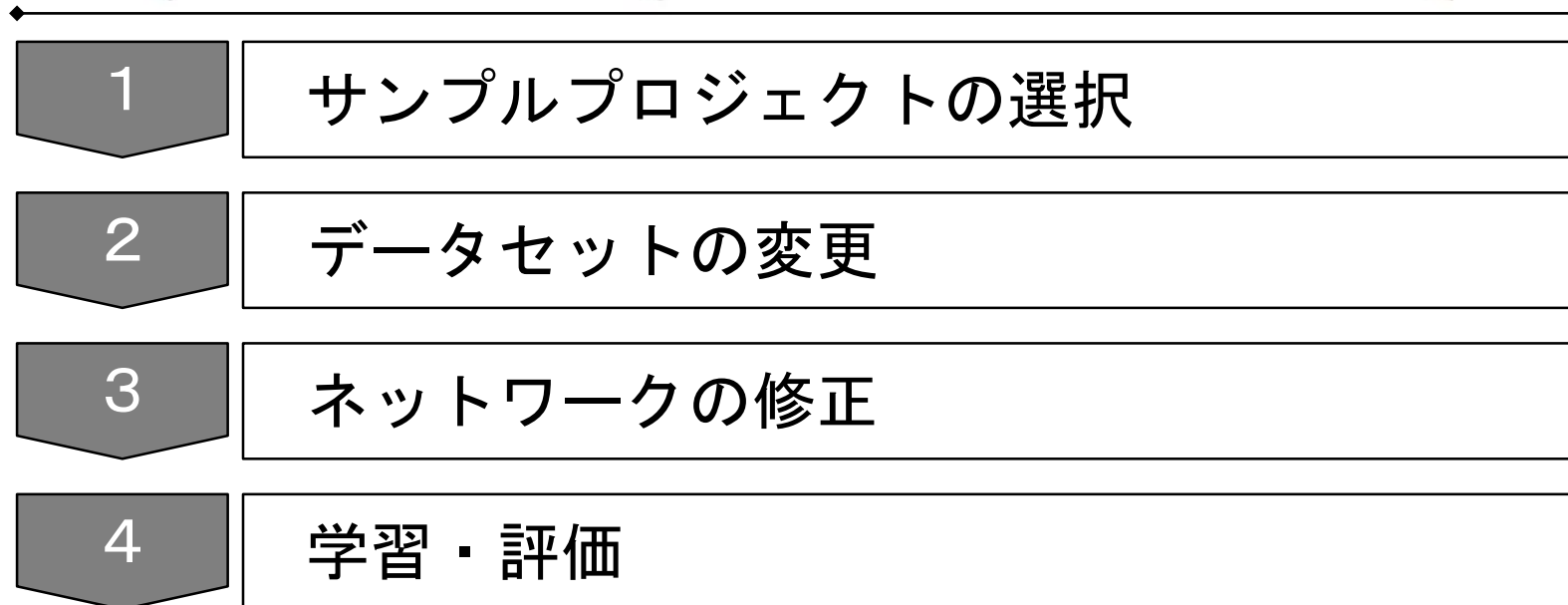
5

物体検出モデル利用

物体検出モデル作成のステップ

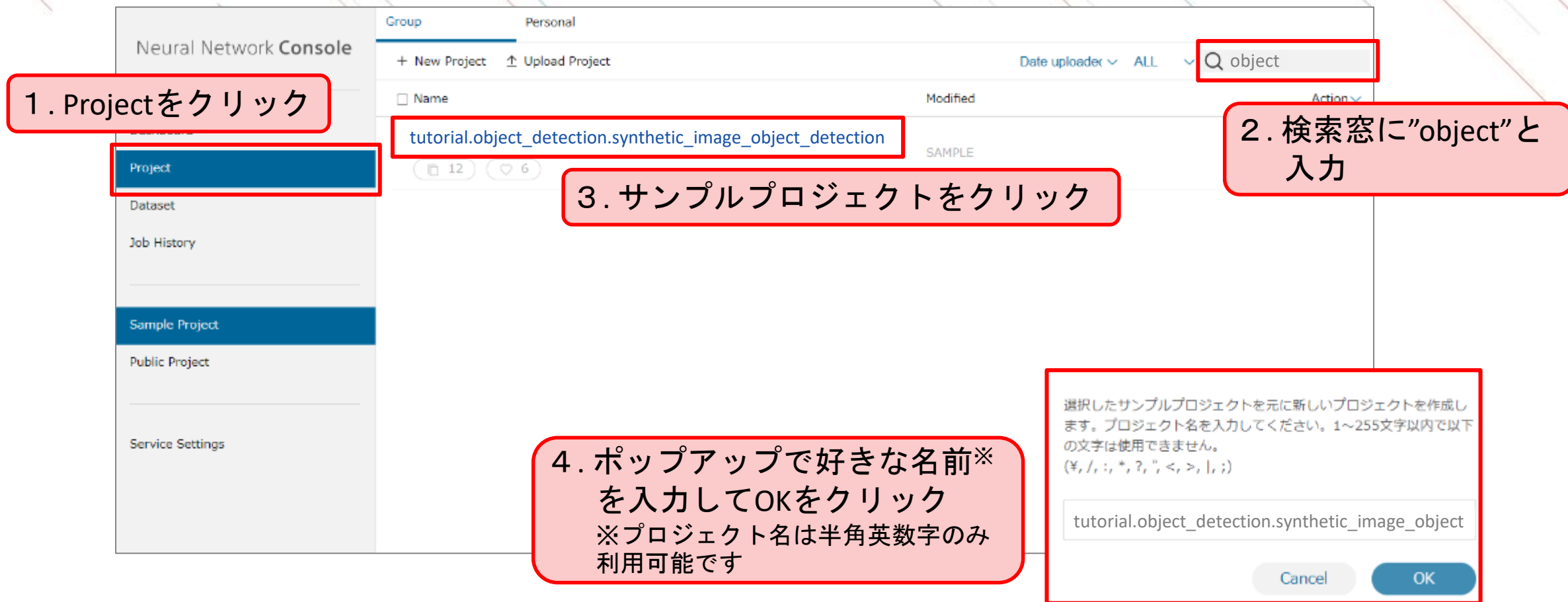
NNCでは新規にモデルを作成することも可能ですが、このドキュメントでは、物体検出モデルのサンプルプロジェクトをベースにすることで、比較的容易にモデル作成を可能にします。

物体検出モデル作成のステップ



サンプルプロジェクトのコピー

サンプルプロジェクト(tutorial.object_detection.synthetic_image_object_detection)を検索し、コピーします。



※学習結果やモデルも含めコピーを行うため、処理に時間がかかる場合があります。

データセットの変更

サンプルプロ
ジェクト選択

データセット
変更

ネットワーク
修正

学習・評価

DATASETタブからTrainingとValidationのデータセットをそれぞれ変更します。
(サンプルデータを利用する場合には変更は不要です)

1. Datasetタブをクリック



2. Trainingをクリック



3. Link Datasetの横のハイパーリンクをクリック
サンプルでセットされている既存のデータセット名が記載されています。
Training: synthetic_image.object_detection.training
Validation: synthetic_image.object_detection.validation

6. Validationをクリックし、 3～5と同様の手順を実施

4. 一覧から学習に用いるデータセットを選択
データセットのアップロード時に指定したファイル名を選択してください。



選択したデータセット名

object_detection_training

500 Rows 2 Cols 20

5. リンクマークをクリック※

※ブラウザの拡大率によって表示されないことがあります。
表示されない場合は表示の縮小をお試しください。

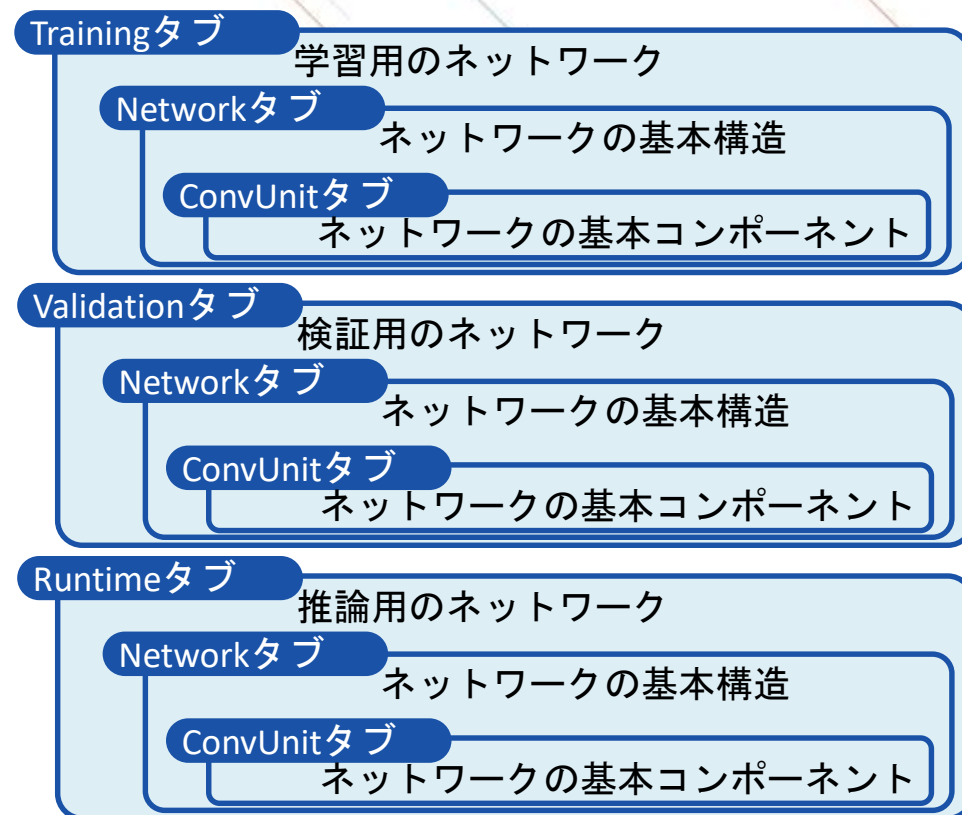
ネットワークの構成

本サンプルでは学習用、検証用、推論用のネットワークが入出力や評価指数などを個別に設定するため、Trainingタブ、Validationタブ、Runtimeタブと別々に記載しています※。
各ネットワークでは共通のNetworkタブが引用されており、NetworkタブはConvUnitタブを引用されています。

ネットワーク画面の説明

The screenshot shows the Neural Network Console interface with the 'Network' tab selected. The 'Components' panel on the left lists various layers and parameters. The main area displays a hierarchical tree of the network structure, including 'NetworkInput', 'MulScalar', 'Unit', 'Unit_2', 'Unit_3', 'Unit_4', 'Unit_5', 'Unit_6', and 'WorkingMemory'. Red boxes and arrows highlight specific elements: a box around the 'Network' tab in the top bar, a box around 'Unit_3' in the tree, and a box around the 'Unit' layer details. A red callout box points to the 'Unit' layer details, stating: 'Network、Training、Validation、Runtime、ConvUnitの5つのネットワークタブが存在' (There are 5 network tabs: Network, Training, Validation, Runtime, and ConvUnit). Another red callout box points to the 'Unit_3' layer, stating: '修正が必要になる回数が多い変数はパラメータとして設定' (Variables that require many corrections are set as parameters). A third red callout box at the bottom left states: 'Unitレイヤーとは他のネットワークタブを引用できるレイヤーです。Networkタブの中ではConvUnitタブのネットワークを引用してます。' (The Unit layer is a layer that can reference other network tabs. In the Network tab, it references the network in the ConvUnit tab).

ネットワークタブの引用関係



※それぞれのネットワークはCONFIGタブの設定画面で学習用、検証用、推論用に対応づけられています

ネットワークの修正

本サンプルではデータセットが変更された際に調整するパラメータを引数レイヤー(Argument)として抜粋しています。データセット作成時の入力画像・分類数・アンカー数をもとにパラメータを修正してください。(サンプルデータを利用する場合には変更は不要です)

ネットワーク修正の操作説明

The screenshot shows the Neural Network Console interface with the following steps highlighted:

1. Trainingタブをクリック (Click the Training tab)
2. NumClassをクリック (Click NumClass)
3. Valueの値を修正 (Modify the Value)
4. 他パラメータも2、3の手順で修正 (Also modify other parameters using steps 2 and 3)
5. Validation、Runtimeタブも2～4の手順で修正※ (Also modify Validation and Runtime tabs using steps 2-4 ※)

The interface shows the 'Training' tab selected, with the 'NumClass' parameter set to 4. The 'Layer Property' panel shows the 'Argument' layer with 'Name: NumClass' and 'Value: 4'.

修正が必要なパラメータの説明

パラメータ	説明
NumClass	分類数
NumAnchor	アンカー数
InputCh	入力画像のモノクロ/カラーを設定 1:モノクロ、3:カラー
InputWidth	入力画像の幅
InputHeight	入力画像の高さ
ScoreThreshold	推論時に物体を検出する閾値 0～1の値で、値を大きくするにつれ判断基準が厳しくなる (Runtimeタブのみ設定)

※ 引用元のConvUnit、Networkタブについては、Training、Validation、Runtimeタブでパラメータを設定することで、自動でパラメータ情報が反映されます。

学習の実行

新規プロジェクト作成

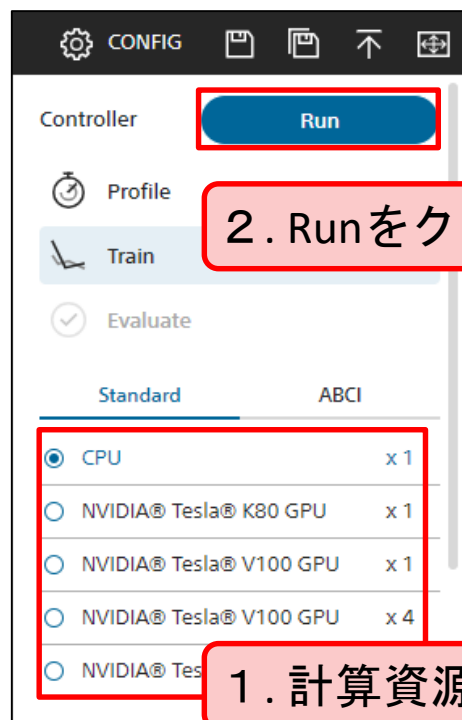
データセット変更

ネットワーク修正

学習・評価

EDITページのRunボタンをクリックすることで学習が実行されます。
GPUを選択すると、高速に学習を行うことができます※。（参考: [学習環境と処理時間](#)）
GPU等有料のメニューを利用する場合は事前にクレジットカード登録もしくは法人契約が必要になります。
(法人契約: <https://dl.sony.com/ja/business/>)

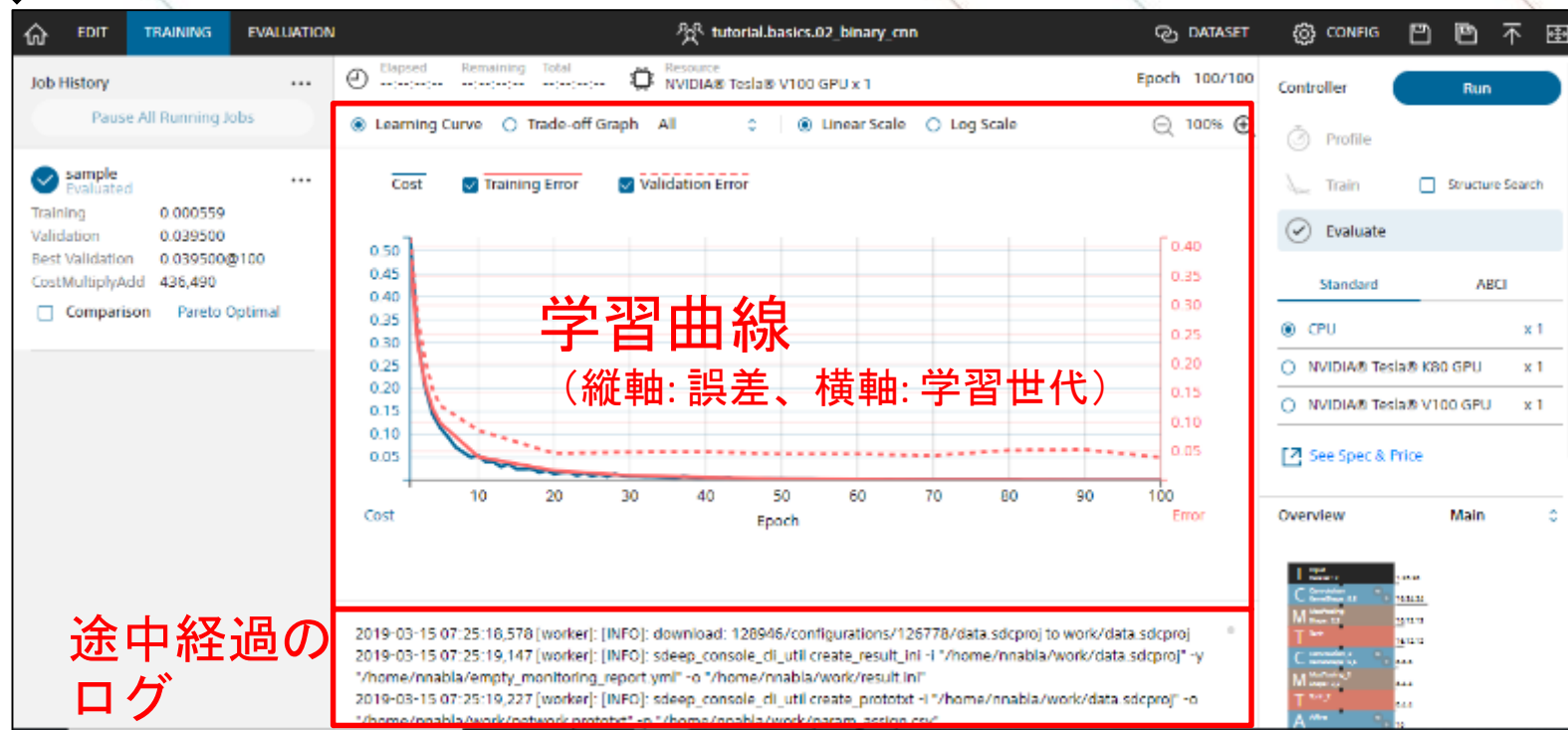
学習実行の方法



2. Runをクリック

1. 計算資源を選択

TRAININGページの概要



※ サンプルプロジェクトのネットワークは複雑なため、CPU実行の場合にはGPU実行と比較して学習時間が長時間になります。
計算途中にウェブブラウザを閉じてても計算が止まることはありませんので、長時間に及ぶ場合にはあらためて結果をご確認ください。

学習曲線の読み取り方

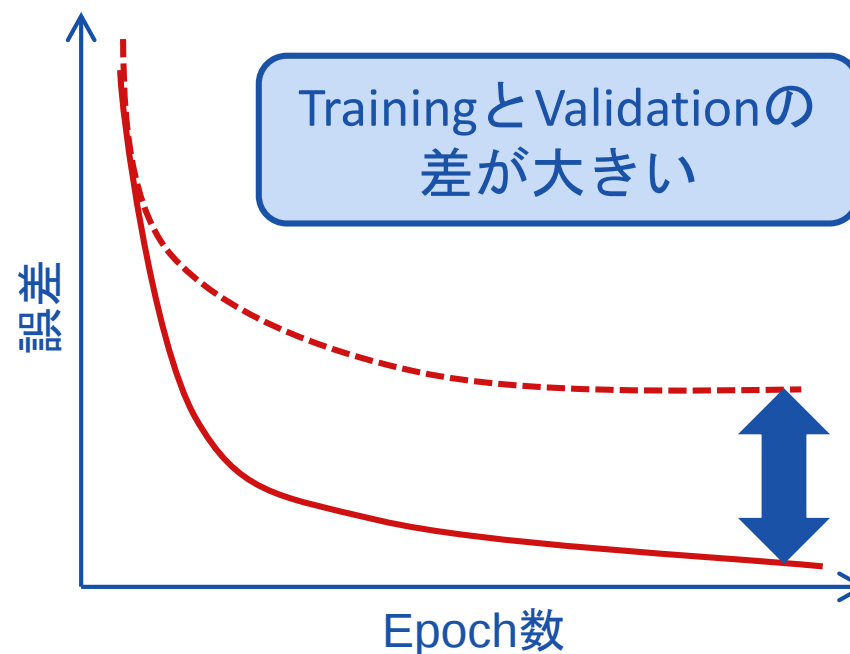
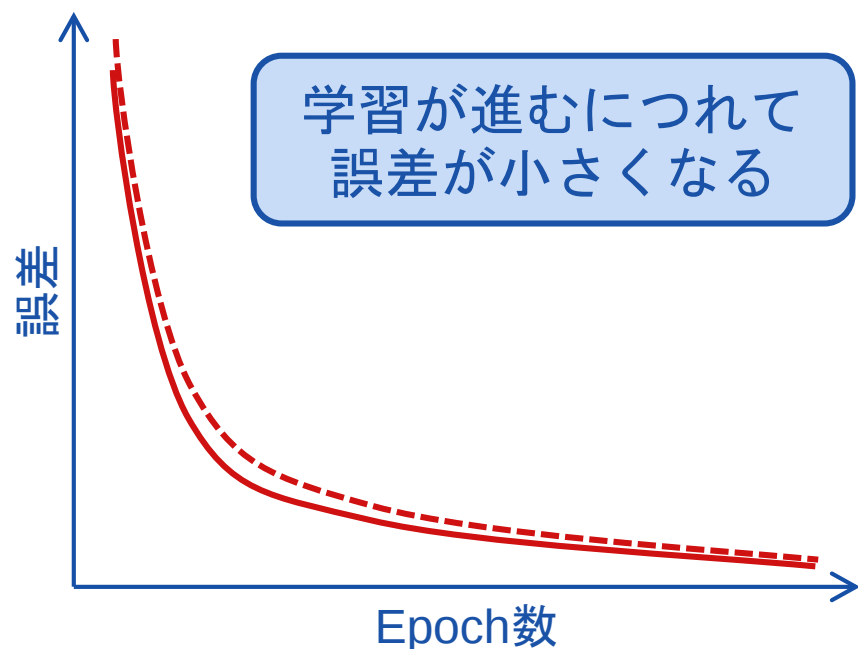
学習結果の良し悪しは、まずは学習曲線から判断をします。

TrainingとValidationの差が大きい場合(過学習)は、モデルがTraining Dataに特化し過ぎた状態(教科書を丸暗記した場合)に
応用問題が解けないのと似た状態)です。

未知のデータの予測精度が低いため、学習データ量を増やすなどの改善が必要です。

良いモデルの学習曲線

悪いモデルの学習曲線（過学習）



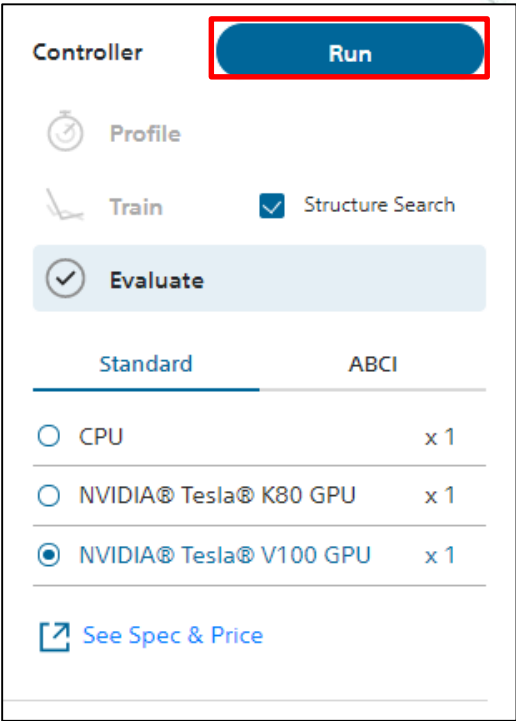
— Training Data
(学習に使用したデータ)
- - - Validation Data
(学習に使用しないデータ)

評価の実行

TRAININGページのRunをクリックするとEVALUTIONページに遷移し、詳細な判定結果を確認できます。
分類問題の場合には、各データに対するモデルの判定結果や統計的な精度や指数、混同行列などを確認できます。

評価実行の方法

表示可能なグラフの概要



評価グラフ	内容	問題
Output Result	各データの1つ1つの判定結果	すべて
Confusion Matrix	データセット全体の統計的な指標と混同行列(分類ラベルごとに結果を集計した表)	分類
Classification Result	各データの判定確率上位3カテゴリーの確率	分類
Classification Matrix	カテゴリーごとのモデルの判定傾向	分類
Likelihood Graph	判定確率と正答率の傾向	分類

物体検出は対象外のため、表示されない

Output Resultの見方

サンプルプロ
ジェクト選択

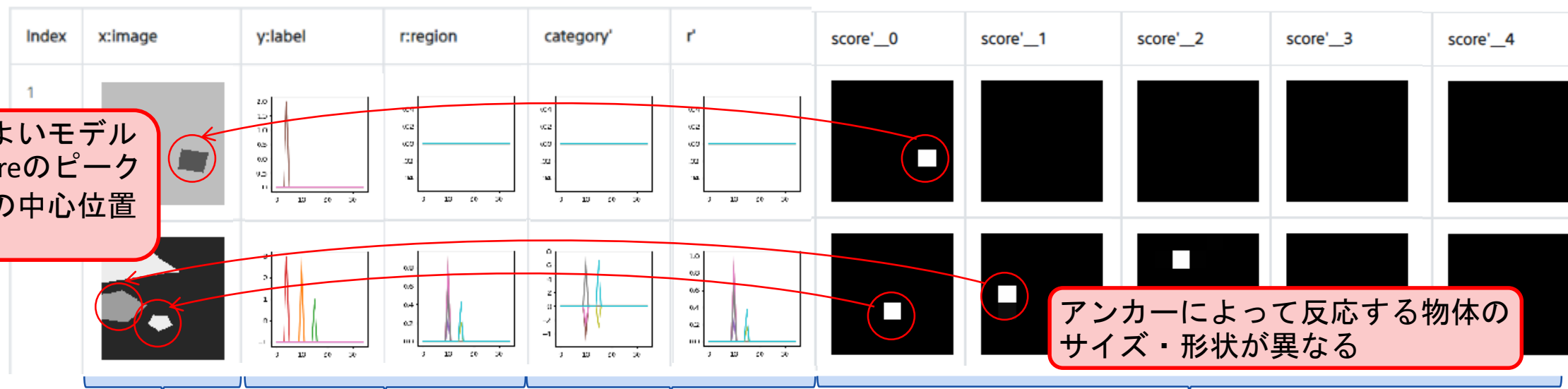
データセット
指定

ネットワーク
修正

学習・評価

Output Resultでは検証データの1つ1つの判定結果を確認できます。
予想結果のScoreのピークが物体中心にあるかどうかで、モデル精度がよいかを確認できます。
その他の出力の説明は次頁をご覧ください。

Output Resultの概要



精度がよいモデル
は、scoreのピーク
が物体の中心位置
と一致

アンカーによって反応する物体の
サイズ・形状が異なる

入力画像

入力データのラベル情
報とバウンディング
ボックスの詳細位置

カテゴリとバウンディ
ングボックスの位置

アンカーごとにグリッド内に物体中心が含まれる確率

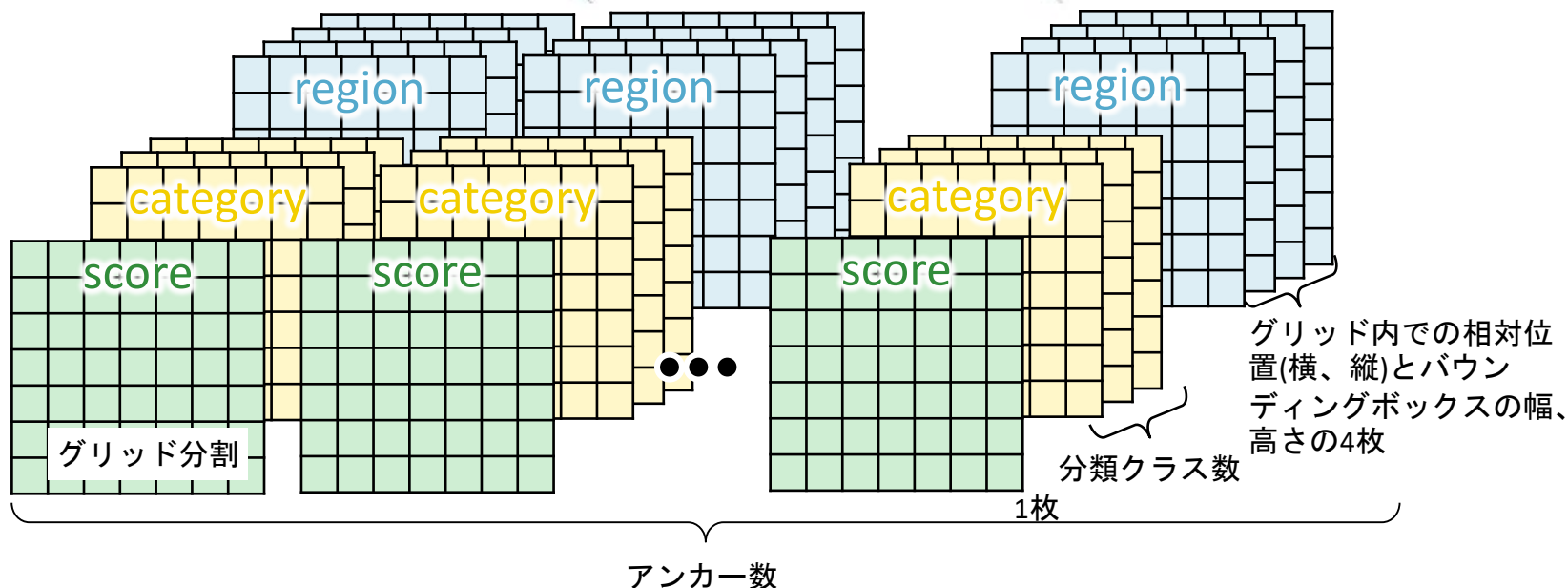
※label、region、category'、r'が時系列表示されますが、実際のデータは表データ(CSVファイル)として保存されています。

サンプルプロジェクトの出力

サンプルプロジェクトの出力はグリッド内に物体中心を含む確率(score)とグリッド内に含まれる物体中心のラベルのスコア(category)とバウンディングボックスの情報(region)の3種類になります。

出力データのイメージ※1

出力ラベルの説明



ラベル	説明
score	分類クラスによらずグリッド内に物体中心が含まれる確率
category	分類クラスごとのグリッド内に物体中心が含まれるスコア※2
region	グリッド内に物体中心があるバウンディングボックスのグリッド内の相対位置(横、縦)とバウンディングボックスの幅と高さ※3

※1 実際の出力データはscoreがアンカー数分の確率値を示したモノクロ画像、categoryとregionが2次元のCSVファイルになります
詳細は[サンプルプロジェクトの出力ファイル](#)を参照

※2 softmax変換により確率値に変換できます

※3 グリッドサイズにより規格化されており、バウンディングボックスのサイズはさらに対数変換した値です

目次

1 Deep Learningを用いた物体検出とは

2 アカウントサインイン

3 データセット作成

4 物体検出モデル作成

5 物体検出モデル利用

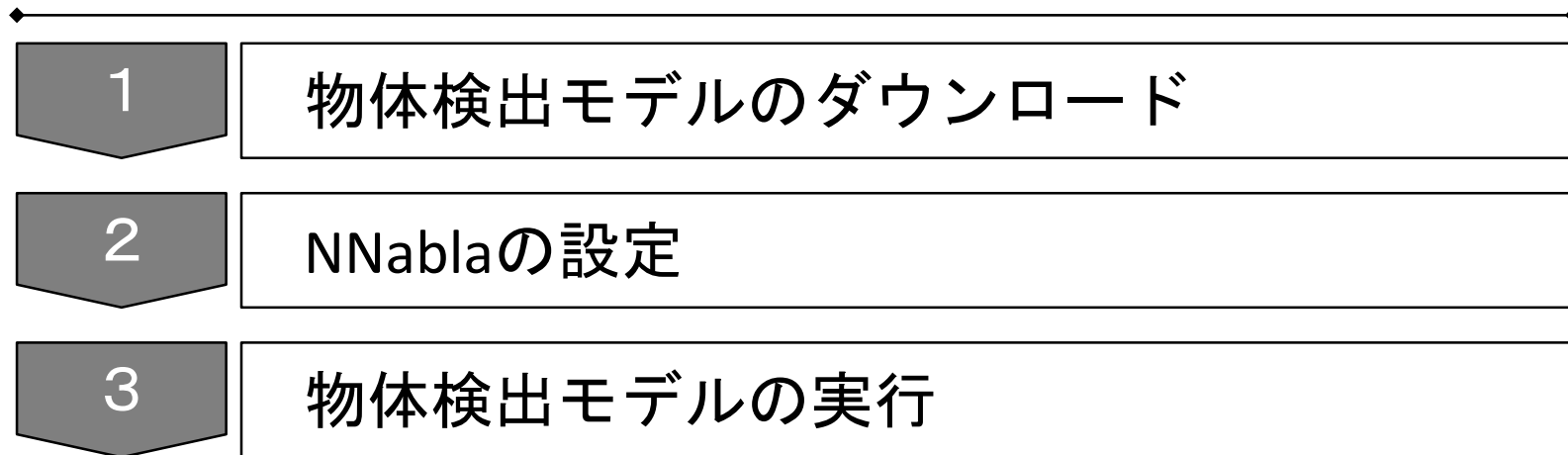
物体検出モデル利用のステップ

NNCから分類モデルをダウンロードすることで、お客様の環境で自由にモデル利用ができます。

モデルを実行するためには、NNablaが必要になります。

NNablaを用いることでコマンドラインやPythonなど様々な方法で作成した物体検出モデルが実行可能となります。

物体検出モデル利用のステップ



物体検出モデルのダウンロード

モデルの
ダウンロード

NNablaの設定

モデルの実行

学習が完了し最適なモデルを作成した後は、「Job History」の中から該当のモデルを右クリックし、選択肢の中の「Download」をクリックすることでモデルをダウンロードできます。

NNP、NNB、ONNXは作成したネットワークと学習済みパラメータの値が含まれたファイルで、実行方法に応じて使い分けをします(詳細は[モデルの実行方法](#))。また、html betaは学習結果などの内容をhtml形式で出力したものです。

作成したモデルの権利は作成者に帰属し、自由にDeep Learningモデルを利用することができます。

The screenshot shows the 'Job History' tab in the Neural Network Console. A context menu is open for the job '20190409_051421'. The 'Download' option is highlighted, and a sub-menu is displayed showing the following options:

- NNP(Neural Network Libraries file format)
- NNB(NNabla C Runtime file format)
- ONNX
- html beta

NNablaの設定

モデルの
ダウンロード

NNablaの設定

モデルの実行

任意のPCにNNablaをインストールします。

NNablaのインストールについては、以下のドキュメントをご参照ください。

<http://nnabla.readthedocs.io/en/latest/python/installation.html>

モデルの実行方法

モデルの
ダウンロード

NNablaの設定

モデルの実行

NNablaを用いてモデルを実行する方法は、使用する言語に応じて様々な方法があります。
また、ONNXを利用することで、他のDeep Learningのフレームワークを利用することも可能です。
次頁以降では、コマンドラインとPythonで実行する方法を解説いたします。

	実行方法	GPU 利用	特徴	ダウンロード ファイル	参考URL
1	コマンドライン	可能	最も簡単に利用可能	NNPファイル	https://support.dl.sony.com/docs-ja/チュートリアル：neural-network-consoleによる学習済みニューラ/
2	Python	可能	比較的容易に利用可能	NNPファイル	https://support.dl.sony.com/docs-ja/チュートリアル：neural-network-consoleによる学習済みニューラ/
3	C++	可能	推論環境にPythonのインストールが不要	NNPファイル	https://github.com/sony/nnabla/tree/master/examples/cpp/mnist_runtime
4	C	不可	非常にコンパクトであり、組み込み利用向き	NNBファイル	https://github.com/sony/nnabla-c-runtime
5	他Deep Learning フレームワーク	環境 依存	環境依存	ONNXファイル	https://nnabla.readthedocs.io/en/latest/python/file_format_converter/file_format_converter.html
6	TensorFlow	環境 依存	環境依存	TensorFlow frozen graphファイル	TensorFlowのウェブページをご覧ください

次頁に
解説あり

コマンドラインでの推論実行

モデルの
ダウンロード

NNablaの設定

モデルの実行

NNablaのインストールされたPython環境で、コマンドラインから以下を実行します。

出力はグリッドごとのscore、category、regionになりますので(参照:[サンプルプロジェクトの出力](#)、[サンプルプロジェクトの出力ファイル](#))、バウンディングボックスの描画には、scoreの高いグリッドのcategoryとregionのデータを取り出し、計算していただく必要があります。

```
nnabla_cli forward ¥  
-c [ダウンロードしたネットワークモデルファイル(*.nnp)] ¥  
-d [推論をするデータセットを取りまとめたCSVファイル] ¥  
-o [推論結果の出力ディレクトリ]
```

※ NNCのEVALUATIONタブでの推論実行時に同様のコマンドを使用しているため、ログの出力ウインドウに同様のものが出力されています。

```
2017-10-24 05:54:28,942 [worker]: [INFO]: nnabla_cli forward -c  
/home/nnabla/results/results_current_100.nnp -d ccbf15a0-bcb6-4ba6-b10e-  
27fc877c4348/1002/index.csv -o /home/nnabla/results
```

Pythonでの実行方法

ダウンロードしたネットワークファイルをPythonで読み込んで利用します。
以下のサンプルでは入力画像を読み込み、モデルを実行し、バウンディングボックスを描画するまでの一連の流れを記載しています。

```
# 必要なmodulesのインストール
import nnabla as nn
from nnabla.utils import nnp_graph
from nnabla.utils.image_utils import imread
import numpy as np
import math
import matplotlib.pyplot as plt
import matplotlib.patches as patches
# ファイル名などの設定
nnp_file = "./result_object_detection.nnp"
input_figure = "./input.png"
output_figure = "./output.png"
colorlist = ["r", "g", "b", "c"]
classlist = ["Ellipse", "Triangle", "Rectangle", "Pentagon"]
threshold = 0.6
num_anchor = 5
num_class = 4
img_height = 112
img_width = 112
figscale = 3
```

1/4

ここの部分を問題に応じて
変更する必要があります

Pythonでの実行方法

2/4

```
# 画像の読み込みと縦横サイズの取得
img = imread(input_figure, grayscale=False, size=(img_width, img_height))
# モデルがモノクロの場合には下記
# img = imread(input_figure, grayscale=True, size=(img_width, img_height))
# nnpファイルの読み込み
nnpFile = nnp_graph.NnpLoader(nnp_file)
networkModel = nnpFile.get_network('Runtime', batch_size=1)
# 入出力レイヤーの設定
nnval_x = networkModel.inputs['Input']
nnval_y_score = networkModel.outputs["score"]
nnval_y_category = networkModel.outputs["category"]
nnval_y_r = networkModel.outputs["r"]
# 入力レイヤーに画像をセットし、推論を実行
nnval_x.d = np.array(img).reshape(nnval_x.d.shape)
nn.forward_all([nnval_y_score, nnval_y_category, nnval_y_r], clear_buffer=True)
# 数値データの取り出し
y_score = nnval_y_score.d[0]
y_category = nnval_y_category.d[0]
y_r = nnval_y_r.d[0]
# グリッドサイズの計算
_, grid_height_num, grid_width_num = y_score.shape
grid_height = img_height/grid_height_num
grid_width = img_width/grid_width_num
```

画像データを読み込み、推論実行をしている箇所。
ここより後方の部分はバウンディングボックスの計算と描画部分である。

※入力画像がモノクロの場合には、ソースコードの一部をコメントアウト部分と変更してください。

Pythonでの実行方法

3/4

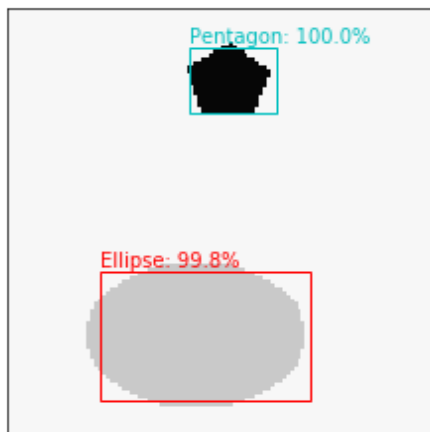
```
# スコアが閾値を超えたグリッドのインデックスを取得
indices_anchor, indices_grid_y, indices_grid_x = np.where(y_score>threshhold)
# Softmax関数を定義
def softmax(x):
    u = np.sum(np.exp(x))
    return(np.exp(x)/u)
# 描画のための推論結果の取り出し
bboxes, texts = [], []
for idx_anchor, idx_grid_x, idx_grid_y in zip(indices_anchor, indices_grid_x, indices_grid_y):
    # category情報の取り出し
    prob = y_category[idx_anchor*grid_height_num+idx_grid_y][num_class*idx_grid_x:num_class*idx_grid_x+num_class]
    prob = softmax(prob)
    label = prob.argmax()
    texts.append("{0}: {1:.1f}%".format(classlist[label], 100*prob[label]))
    # region情報の取り出し
    # dx, dyはGridの左上からのグリッドサイズで規格化したもの
    # ratio_width, ratio_heightはグリッドサイズで規格化し、対数変換したもの
    ratio_dx, ratio_dy, ratio_width, ratio_height = y_r[idx_anchor*grid_height_num+idx_grid_y][4*idx_grid_x:4*idx_grid_x+4]
    cx = (idx_grid_x+ratio_dx)*grid_width
    cy = (idx_grid_y+ratio_dy)*grid_height
    width = math.exp(ratio_width)*grid_width
    height = math.exp(ratio_height)*grid_height
    bboxes.append(patches.Rectangle(xy=(cx-0.5*width, cy-0.5*height), width=width, height=height, ec=colorlist[label], fill=False))
```

Pythonでの実行方法

4/4

```
# バウンディングボックスの描画
fig = plt.figure(figsize=(figscale*img_width/img_height, figscale))
ax = plt.axes()
ax = fig.add_axes([0, 0, 1, 1])
ax.imshow(img)
# モデルがモノクロの場合には下記
# ax.imshow(img, cmap='gray', vmin=0, vmax=255)
for (r, text) in zip(bboxes, texts):
    ax.add_patch(r)
    ax.text(r.xy[0], r.xy[1], text, ha='left', va='bottom', transform=ax.transData, color=r.get_ec() )
ax.tick_params(labelbottom=False, labelleft=False, labelright=False, labeltop=False)
ax.tick_params(bottom=False, left=False, right=False, top=False)
plt.savefig(output_figure)
```

保存画像の例



※入力画像がモノクロの場合には、ソースコードの一部をコメントアウト部分と変更してください。



Appendix

SONYアカウントの取得

アカウント作成ページに移動し、メールアドレスやパスワードなどを設定します。

1. 作成ページへの移動 1

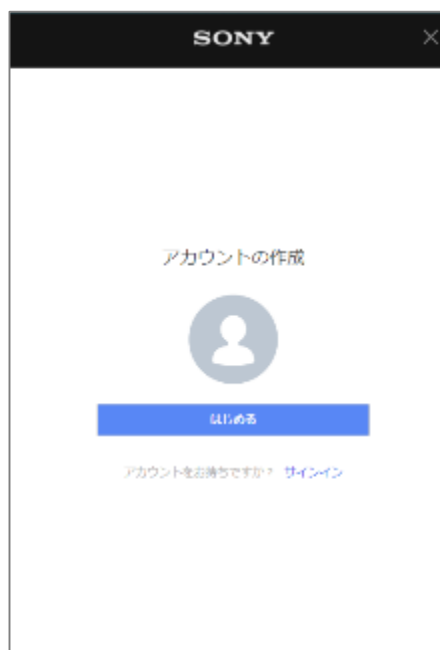
✓「新しいアカウントの作成」を押下



The screenshot shows the Sony login page. At the top, there's a 'SONY' header with a close button. Below it, the title 'サインイン' (Sign In) is centered. A link 'このアカウントで、Sonyグループの様々なサービスへアクセス
もっと詳しく' is present. The 'サインインID' (Sign In ID) section has an 'Eメールアドレス' (Email address) input field and a link 'サインインIDを記憶する' (Remember Sign In ID). The 'パスワード' (Password) section has a password input field with an eye icon to toggle visibility. A blue 'サインイン' (Sign In) button is at the bottom. Below the button, there's a link 'サインインでお困りですか?' (Having trouble signing in?) and a link '新しいアカウントの作成' (Create new account).

2. 作成ページへの移動 2

✓「はじめる」を押下



The screenshot shows the Sony account creation page. At the top, there's a 'SONY' header with a close button. Below it, the title 'アカウントの作成' (Account creation) is centered. A large blue circle with a white person icon is in the center. Below the icon is a blue 'はじめる' (Get started) button. At the bottom, there's a link 'アカウントをお持ちですか? サインイン' (Do you have an account? Sign in).

3. メールアドレス等の入力

✓登録するメールアドレスとパスワードを入力



The screenshot shows the Sony account creation page with input fields. At the top, there's a 'SONY' header with a close button. Below it, the title 'アカウントの作成' (Account creation) is centered. A progress indicator shows the first step is active. The 'サインインID' (Sign In ID) section has an input field with 'youraddress@example.com'. The 'パスワード' (Password) section has a password input field with an eye icon to toggle visibility. Below it, there's a 'パスワードの再入力' (Re-enter password) section with another password input field. A green progress bar indicates the password strength. At the bottom, there's a link 'このアカウントで、Sonyグループの様々なサービスへアクセス
もっと詳しく' and two buttons: '戻る' (Back) and '次へ' (Next).

SONYアカウントの取得

生年月日などを入力し、利用規約などの確認を行います。

4. 生年月日の入力

✓国/地域、言語、生年月日を入力



5. 利用規約への同意

✓メール配信の有無を選択
✓利用規約・アカウントポリシーの確認



6. セキュリティ認証

✓「私はロボットではありません」を押下
※画像選択が表示された場合には指示に従う



SONYアカウントの取得

確認メールを受信し、アカウントの有効化を行います。

7. 確認メールの送付

- ✓登録したメールアドレス宛に確認メールが送付される



8. 確認メールの確認

- ✓確認メールを開き、「確認する」を押下



学習環境と処理時間

一般的にDeep LearningはGPUを用いることにより、CPUと比べ高速に学習処理を行うことが可能です。

学習実行環境と処理時間・ご利用料金

	学習実行環境	学習処理時間	1時間当たりのご利用料金	ご利用料金目安
1	CPU	1,209,600秒 (336時間)	85円	約28,560円
2	NVIDIA® TESLA® K80 GPU	14,976秒 (4.16時間)	210円	約874円
3	NVIDIA® TESLA® V100 GPU	3,960秒 (1.1時間)	560円	約616円

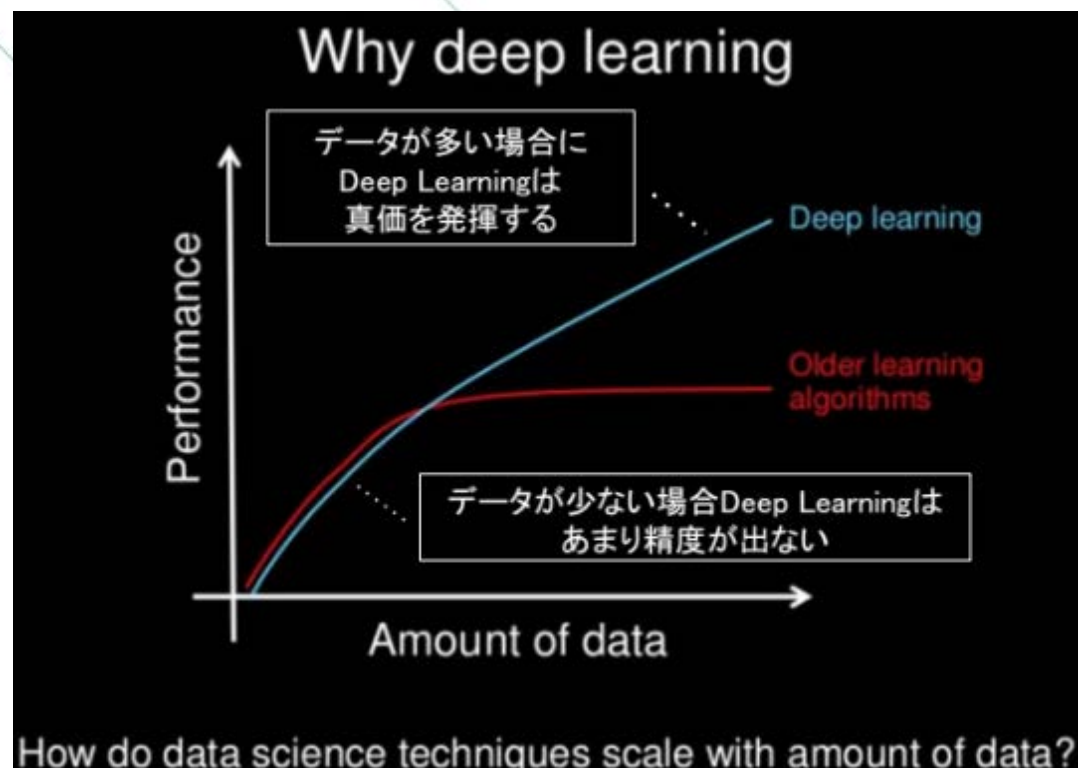
【検証環境】

- ✓ データセット : CIFAR 10
- ✓ ネットワーク : ResNet-101
- ✓ epoch : 300

データ量の重要性

Deep Learningで高い精度を得るにはデータ量が重要になります。Deep Learningではデータを増やせば増やすだけ精度が向上する傾向にあります。

一方でデータ量が少ない場合には、Deep Learning以前の従来型の機械学習に比べても精度が劣ることもあります。



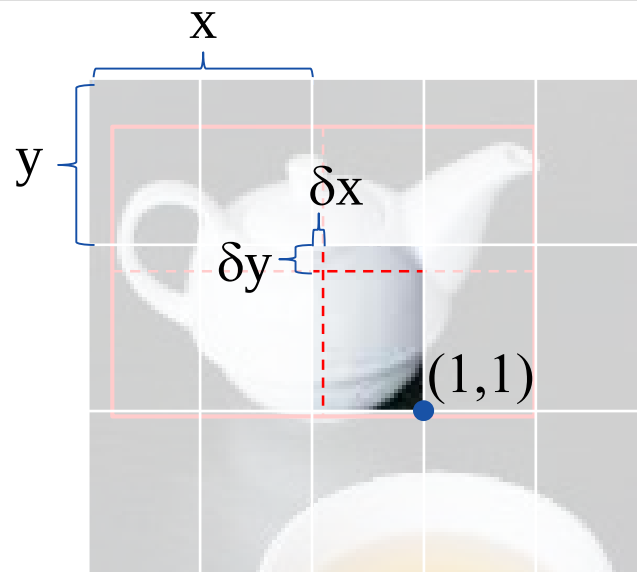
出典: <https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu>

グリッド内での相対位置

バウンディングボックスの中心が含まれるグリッドがわかれば、おおよその中心位置は決まります。グリッド内での相対位置(δx , δy)をさらに予測することで、詳細にバウンディングボックスの中心位置を決定することができます。

δx , δy は学習効率化のため、グリッドサイズで規格化されており、0~1の値をとります。

グリッド内での相対位置とバウンディングボックスの中心位置



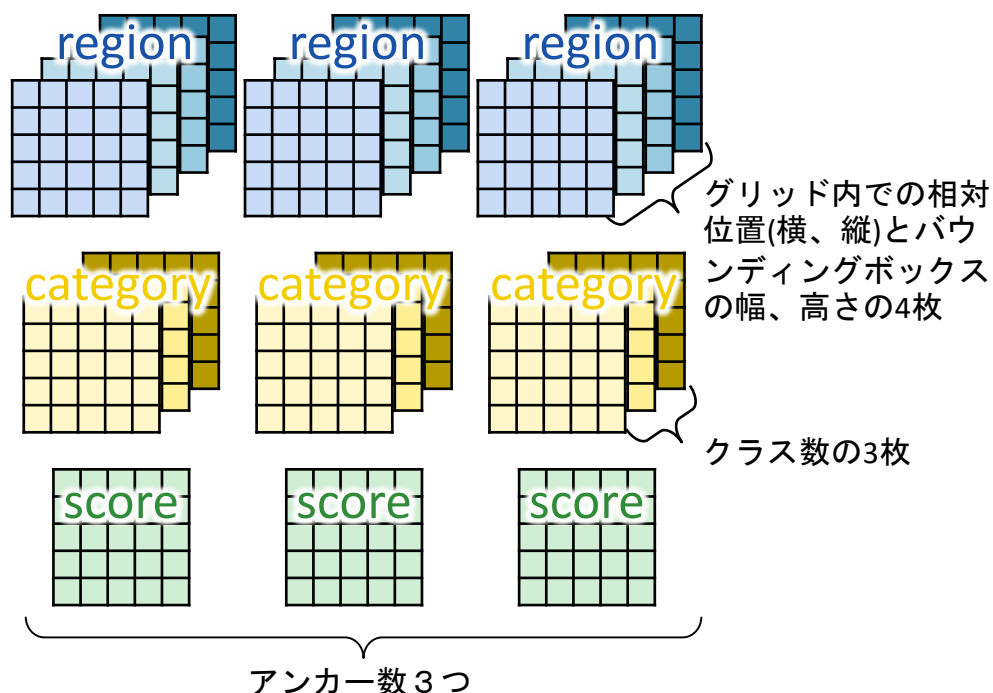
実際のバウンディングボックスの中心位置(X , Y)
 $= (x + [\text{グリッドの幅}] \times \delta x, y + [\text{グリッドの高さ}] \times \delta y)$

サンプルプロジェクトの出力ファイル

scoreは2次元データがアンカー数分あるため、確率値を表すモノクロ画像がアンカー数分出力します。
categoryとregionは3次元データがアンカー数分あるため、x方向に分類クラスの数やバウンディングボックスの情報を、y方向にアンカー数分の情報をそれぞれ並べて、2次元のCSVファイルを出力します。

出力データの変換(グリッド:5x5、クラス数:3、アンカー数:3の場合)

出力配列



出力ファイル

