Neural Network Console クラウド版
ネットワーク解説
-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)にある物体検出のサンプルプロジェクト(tutorial.object_detection.synthetic_image_object_detection)のネットワークを解説します。

サンプルプロジェクトをベースに自らネットワークを変更して精度改善を図るために、現状のサンプルプロジェクトのネットワーク構造を理解したい方を読者と想定しております。
これから物体検出を始めようという方は、物体検出の一連の流れを解説したスターターガイド-物体解説編-をまずはご確認ください。

本ドキュメントでは各レイヤー※1がネットワーク全体でどのような役割を担っているかに焦点を当て説明していますので、ネットワークの中で用いられているレイヤーの具体的な機能はレイヤーリファレンスをご確認ください。

※1 レイヤーとはDeep Learningでネットワークを作成するための関数で、NNCに限らず一般的なもののため、それぞれの詳細な仕組みなどは入門書などでも確認することができます。

目次

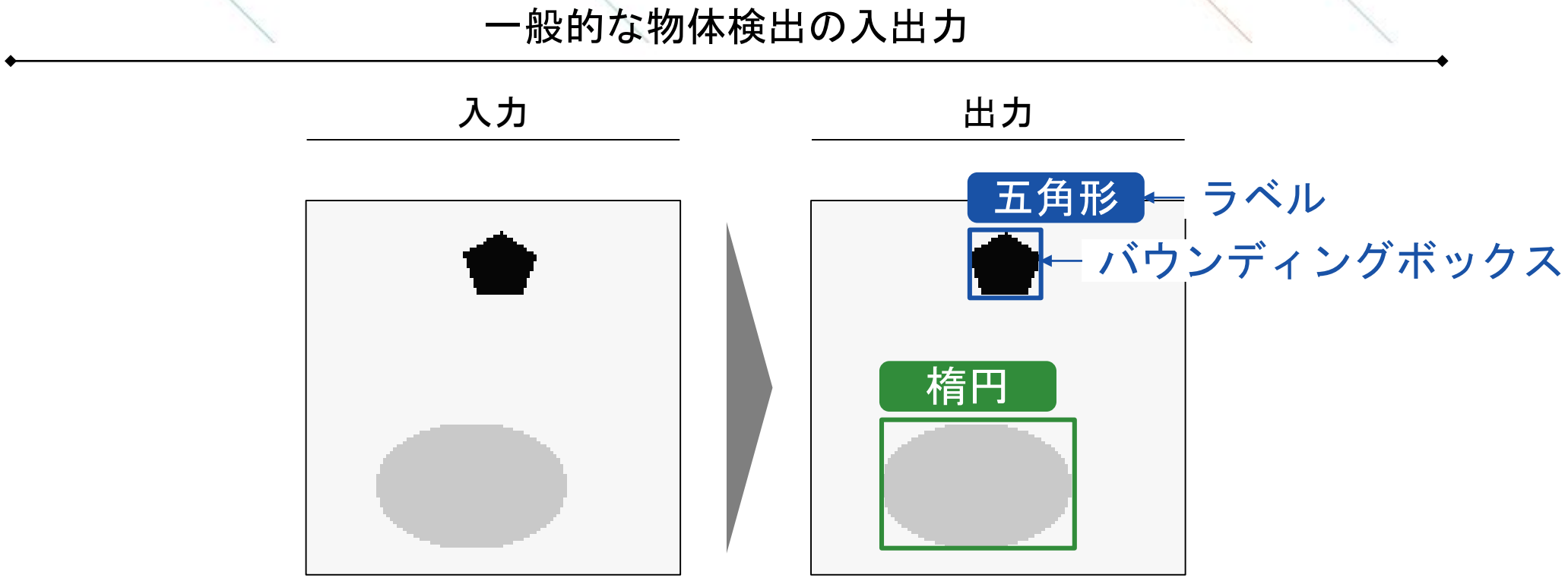
物体検出のサンプルプロジェクトにはネットワークとデータセットが含まれています。

ネットワークの全体像を理解するうえで、データセットとネットワークの入出力の理解が必須であるため、まずはこれらを解説します。そのあとにネットワークの詳細な構造を解説していきます。

一般的な物体検出の入出力

一般的な物体検出では画像に映る検出対象のラベルと、それを囲う四角形(バウンディングボックス)を検出します。

本サンプルプロジェクトには、正方形の背景の上に色、個数、形状がランダムな物体(楕円、三角形、四角形、五角形)を配置したデータセットがすでに登録されています。



データセットの説明

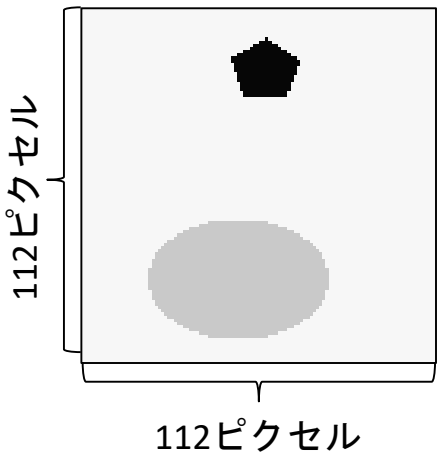
データセットには入力として画像データ(image)が、出力としてアンカーボックスとグリッドごとのラベル(label)とバウンディングボックス(region)が含まれています。

labelとregionはバウンディングボックスの中心が含まれるグリッドに値が記載されています。中心が含まれない箇所はlabelは-1、regionは0で穴埋めがされています。

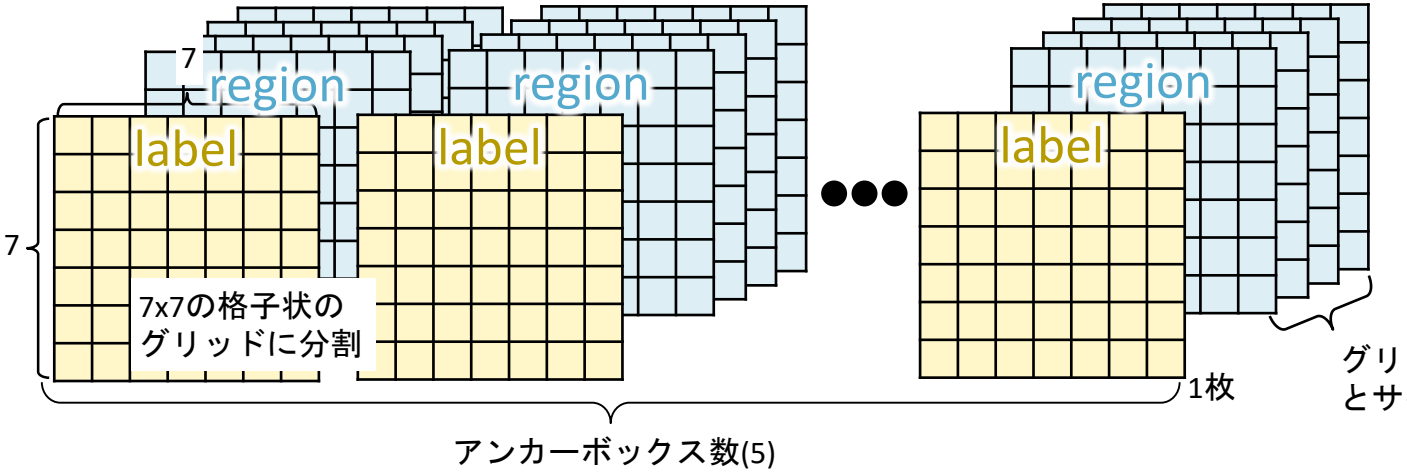
アンカーボックスやグリッドは学習効率化のための手法で、詳細はスターガーガイドで解説していますので、ご参照ください。

データセットの入出力

入力: 画像(image)



出力: グリッドごとのラベル(label)とバウンディングボックス (region)



グリッド内での相対位置(横、縦)とサイズ(幅、高さ)※の4枚

ラベルと物体の対応

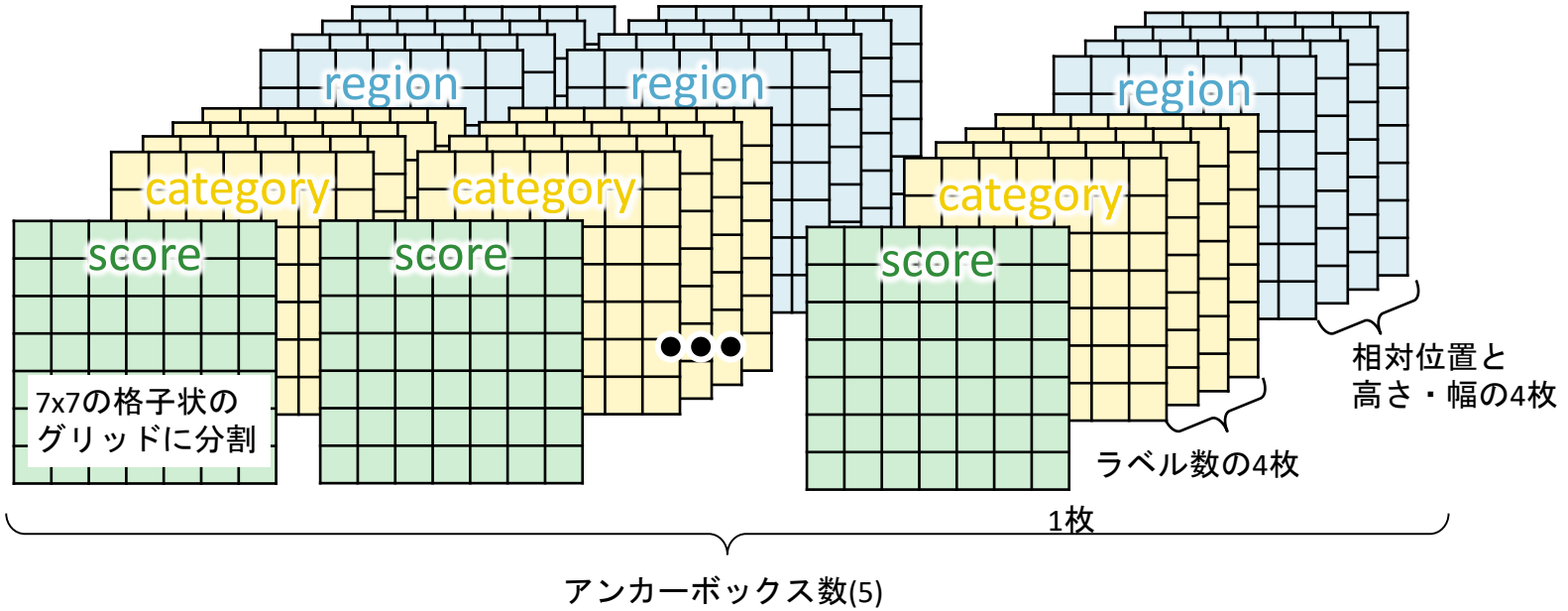
ラベル	物体
0	楕円
1	三角形
2	四角形
3	五角形

※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をした後に対数変換をしています

ネットワークの出力情報

実際のネットワークではデータセットの出力のlabelを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分類ラベルの指数※1(category)の２つに分けて予測をします。
検出物体の分類ラベル予測にあわせて、検出物体の有無(score)を予測することで効率的な学習ができます。
したがってネットワークの出力はscore、category、regionの３つになります。

ネットワークの出力情報



出力情報の説明

出力情報	説明
score	ラベルによらずグリッド内に物体中心を含む確率
category	グリッド内に含まれる物体中心の分類ラベルの指数※1
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での相対位置(横、縦)とサイズ(幅、高さ)※2

※1 softmax変換により確率値に変換できます
※2 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

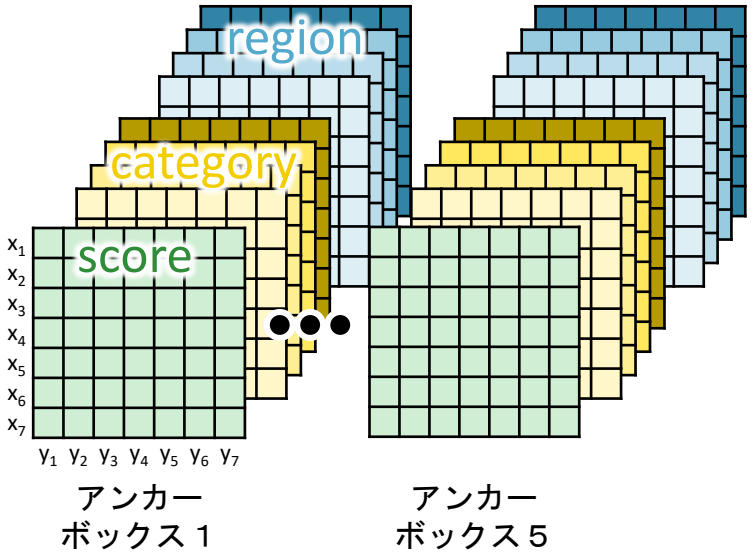
ネットワークの出力ファイル

scoreは確率値を表すモノクロ画像がアンカーボックス数分出力されます。

categoryとregionは１つのCSVファイルとして出力するために、x方向に分類クラスの確率やバウンディングボックスの情報を、y方向にアンカーボックス数分の情報をそれぞれ並べ、2次元の情報に変換します。

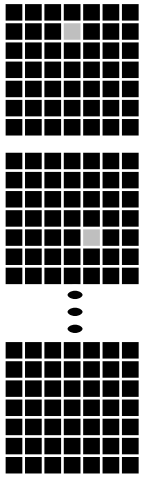
出力情報の形状変換

出力情報



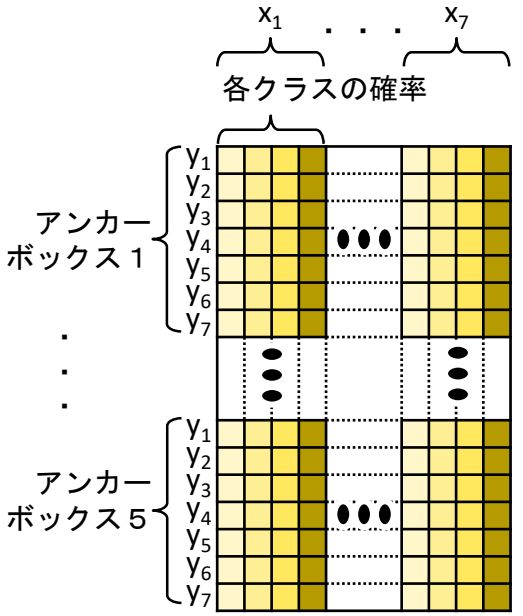
出力ファイル

score_0.png
～score_5.png

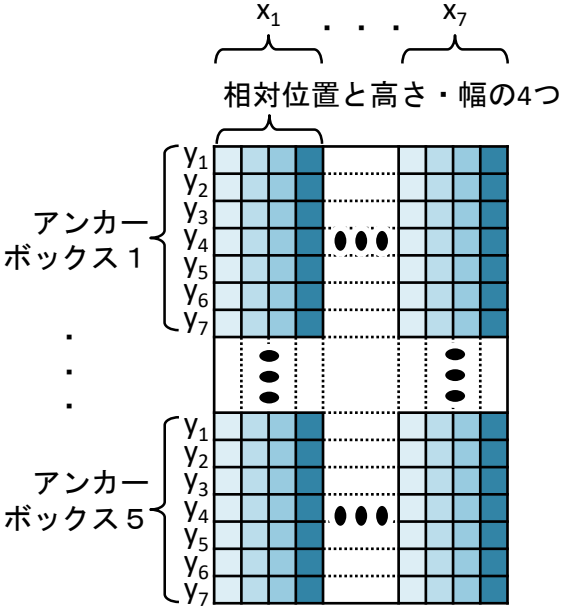


各アンカーボックスのscoreを0(黒)～1(白)の画像データで保存

category.csv



region.csv



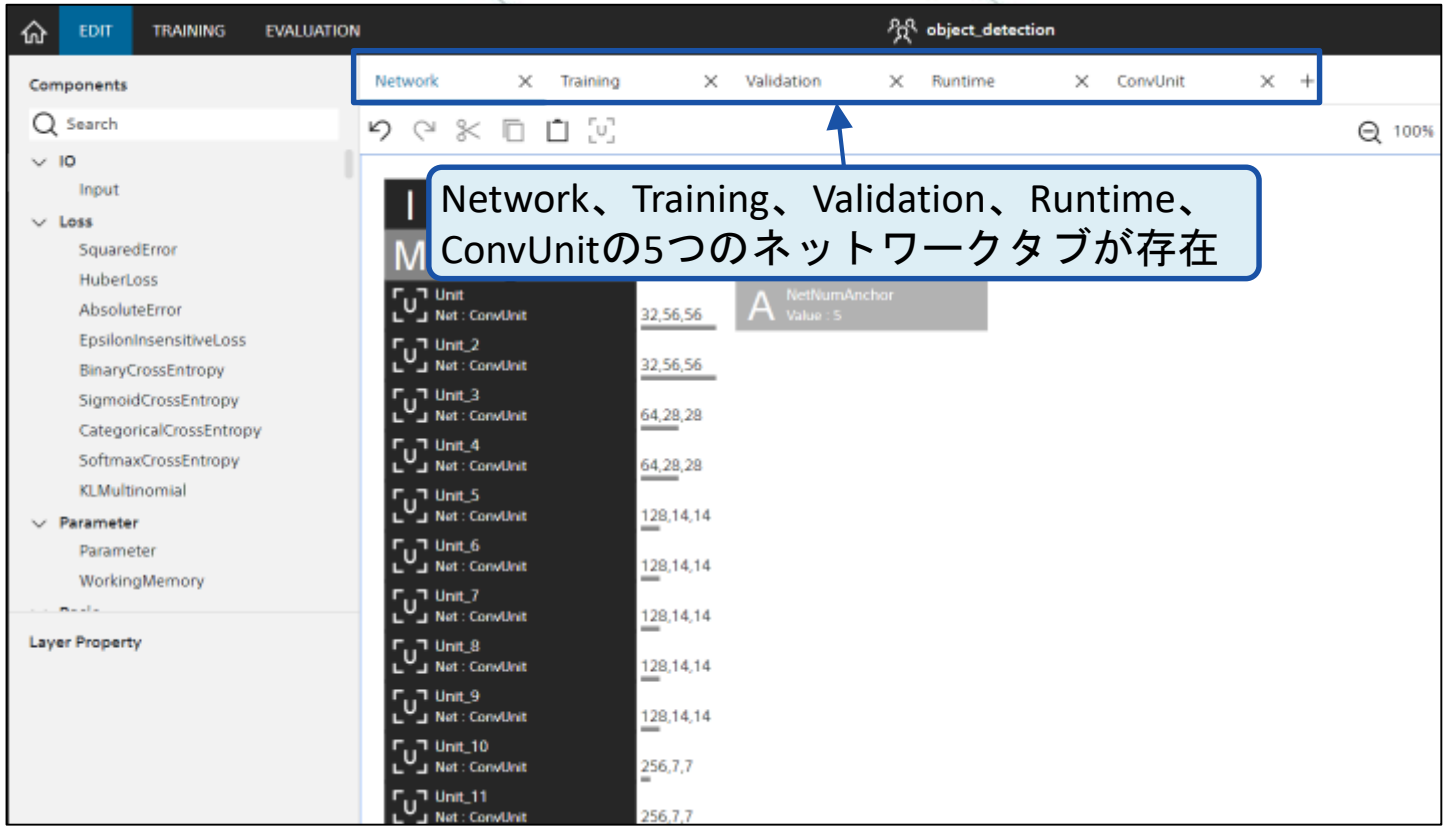
目次

ここまでネットワークの全体像を理解いただくために、データセットとネットワークの入出力を解説をしてきました。ここからはネットワークの構造を解説していきます。

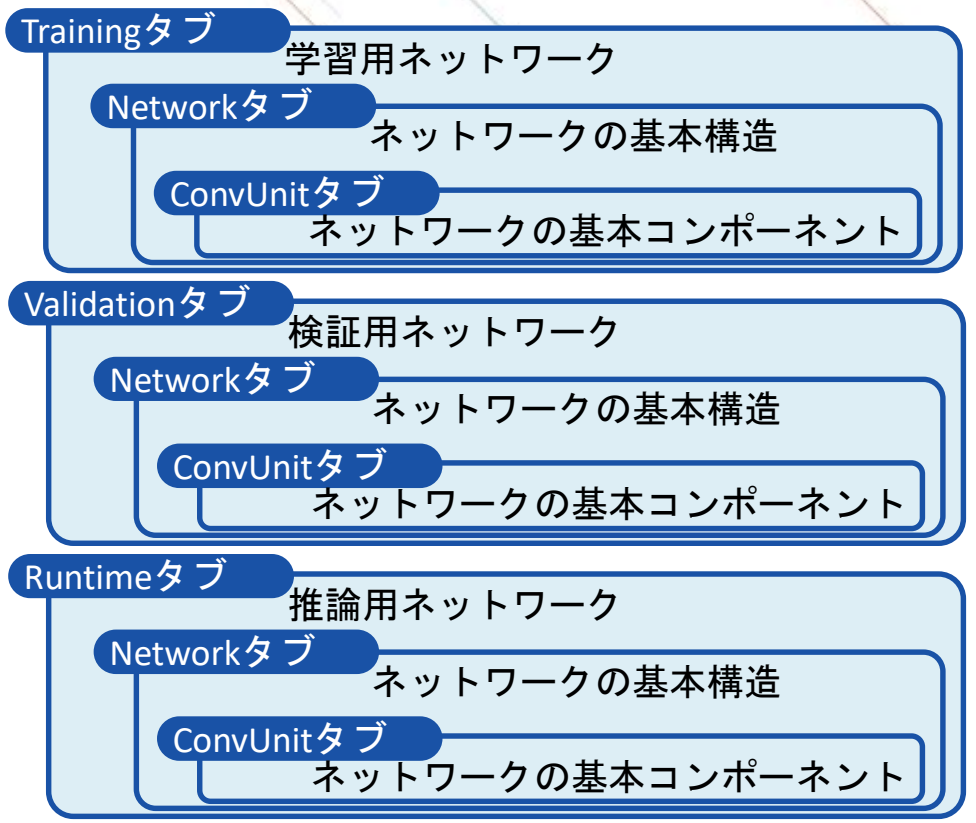
ネットワークの全体構成

本サンプルプロジェクトでは学習用、検証用、推論用のネットワークの入出力や誤差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タブを引用しており、NetworkタブはConvUnitタブを引用しています。

ネットワーク画面の説明



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

NetworkタブとConvUnitタブの概要

Networkタブでは画像処理で一般的なCNN(**C**onvolutional **N**eural **N**etwork)型のネットワークが記述されています。
Unitレイヤーで引用しているConvUnitタブがCNNの基本構造を示しています。

Networkタブ

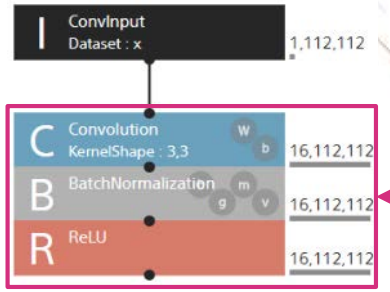
入力画像を取得

入力画像のピクセル値を0～1に変換

ConvUnitタブの引用で基本構造を連続し、CNNを形成

score、category、regionを出力※1

ConvUnitタブ



Convolution層を含むCNNの基本構造

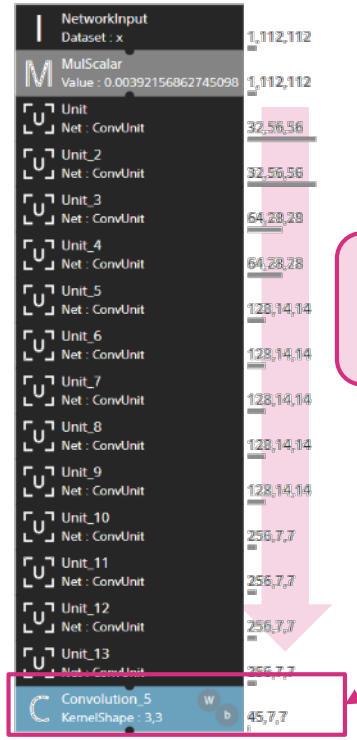
※1 サンプルプロジェクトの場合、1次元目は(score1枚, category4枚, region4枚)がアンカーボックス数5セットで45、2,3次元目は格子状のグリッドで7, 7にサイズ調整

【参考】精度改善のためのNetworkタブの変更方法

Networkタブの Unitレイヤーの増減やUnitレイヤーのパラメータ変更で、モデルの精度改善が期待できます。

変更の際には最終層のサイズをscore、category、regionのサイズに合わせることに注意してください。
各層のサイズはUnitレイヤーの詳細設定で決まります。

Networkタブの設計ポイント



特徴量マップのサイズ(2次元目x 3次元目)を小さくしながら、枚数(1次元目)を増やしていく

score、category、regionのサイズに合うように調整※1

Unitレイヤーの詳細設定の例

ConvMapsで1次元目のサイズが決定し、ConvStrideで2,3次元目のサイズが決定します。ConvStrideが1の場合には入力と同サイズ、2の場合には入力の1/2のサイズになります。
詳細設定では以下のようにパラメータを用いた入力も可能です。

Unit

Name	Unit
Input	1,112,112
Network	ConvUnit
ParameterScope	Unit
ConvBS	False
ConvMaps	32
ConvStride	2
Convolution.W.File	
Convolution.W.Initial	1

Unit_2

Name	Unit_2
Input	32,56,56
Network	ConvUnit
ParameterScope	Unit_2
ConvBS	False
ConvMaps	Input[0]
ConvStride	1
Convolution.W.File	
Convolution.W.Initial	1

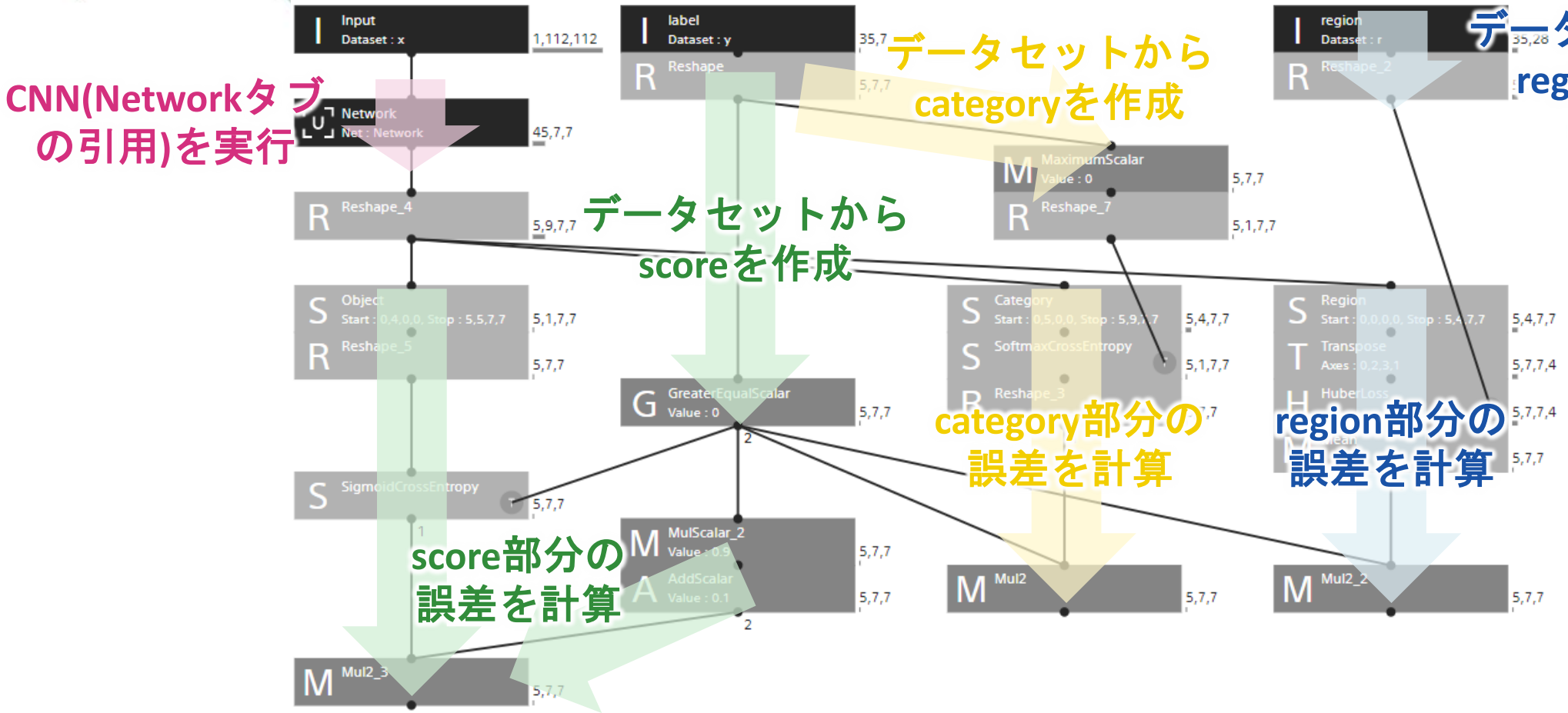
Unit_3

Name	Unit_3
Input	32,56,56
Network	ConvUnit
ParameterScope	Unit_3
ConvBS	False
ConvMaps	Input[0]*2
ConvStride	2
Convolution.W.File	
Convolution.W.Initial	1

※1 サンプルプロジェクトの場合、1次元目は(score1枚, category4枚, region4枚)がアンカーボックス数5セットで45、2,3次元目は格子状のグリッドで7, 7にサイズ調整

Trainingタブ、Validationタブの概要※1

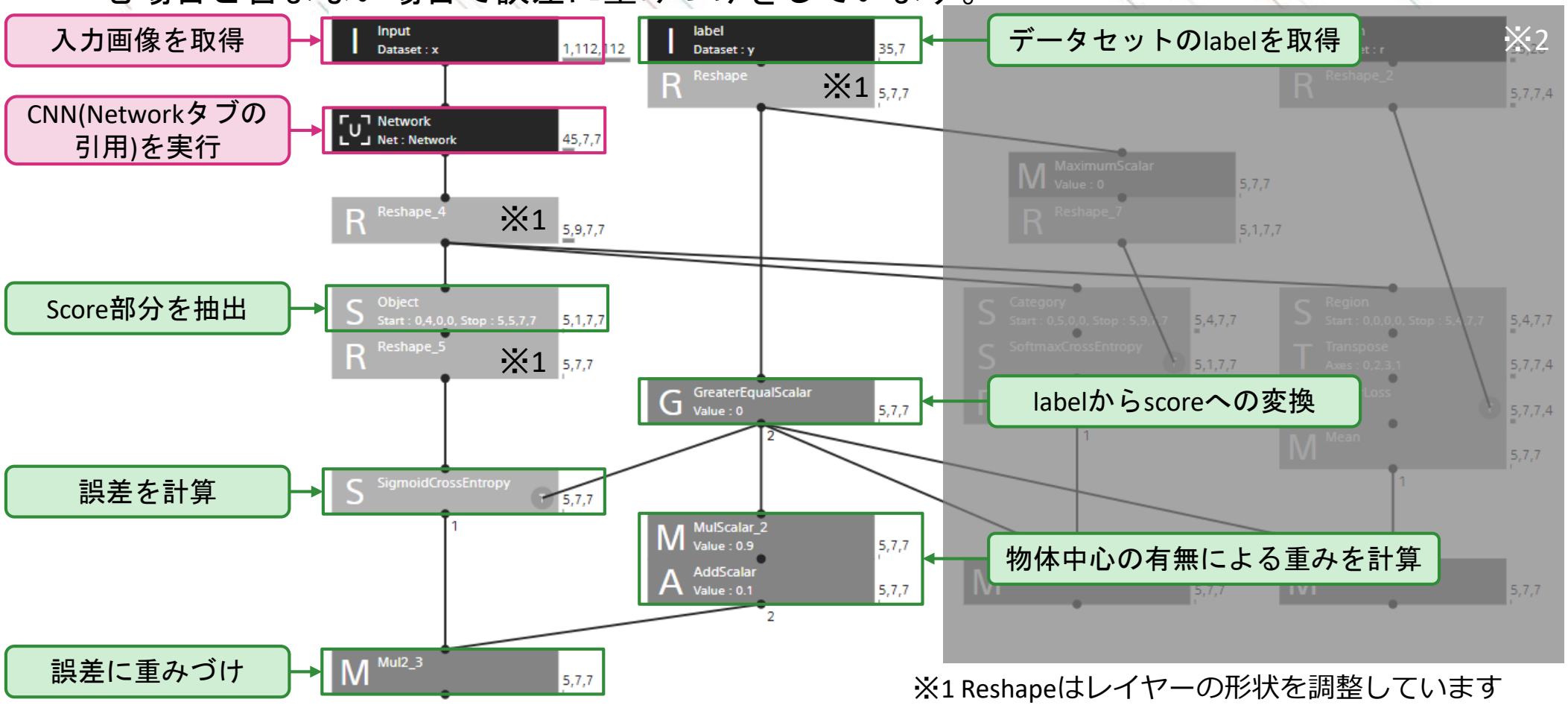
CNN(Networkタブの引用)からの出力をscore、category、regionに分割し、データセットから作成したそれぞれの値との誤差を計算をしています。誤差は合計され、学習時や学習曲線の描画で利用されます。



※1 TrainingタブとValidationタブのネットワーク構造は同様のため、合わせて説明をしています。

Trainingタブ、Validationタブのscore部分の解説

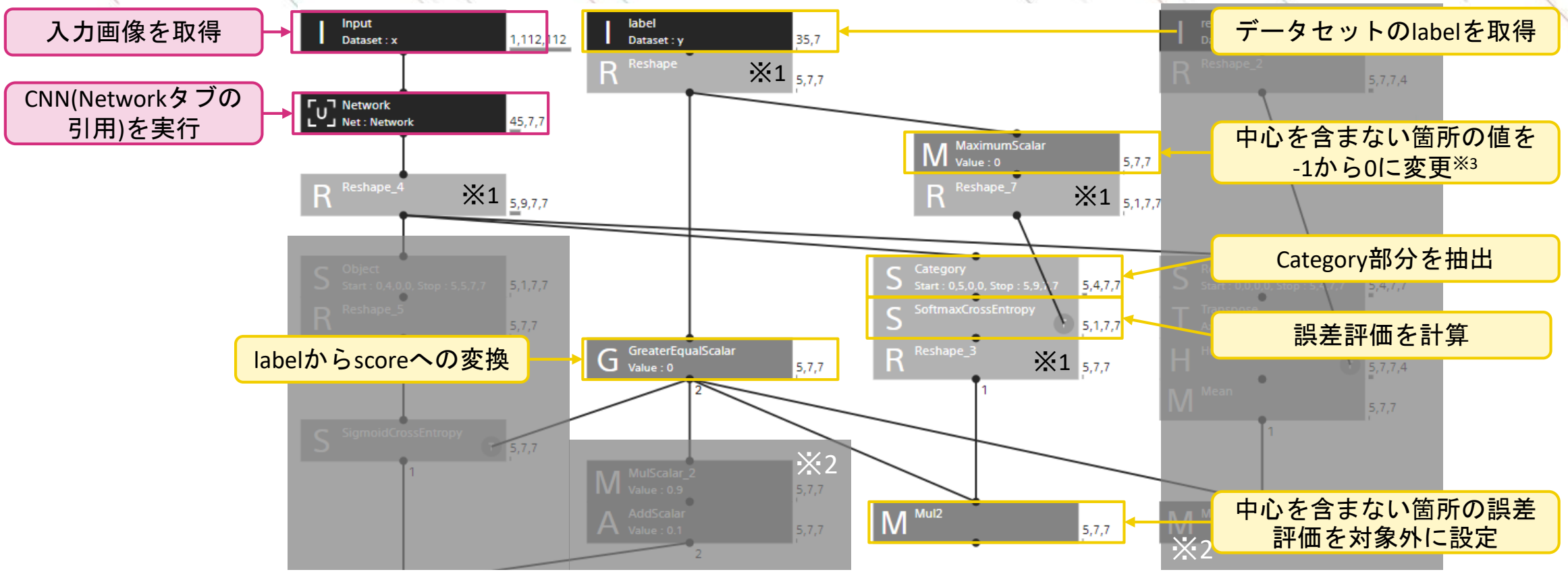
入力画像からCNN(Networkタブの引用)を通して得られたscoreと、データセットのlabelから作り出したscoreとの誤差を計算しています。scoreは物体を含まないことが多いため、学習のテクニックとして最後に物体を含む場合と含まない場合で誤差に重みづけをしています。



※1 Reshapeはレイヤーの形状を調整しています
※2 グレーの部分はcategory, regionに関連するため、他頁で説明をしています

Trainingタブ、Validationタブのcategory部分の解説

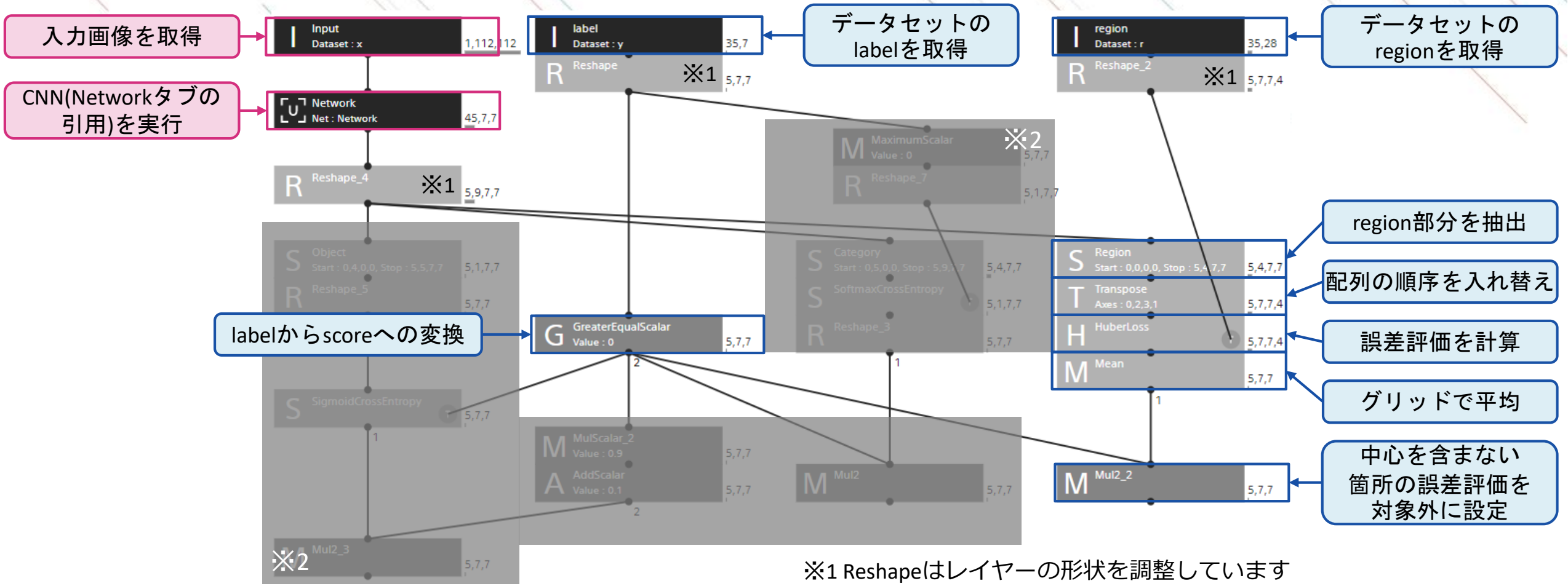
入力画像からCNN(Networkタブの引用)を通して得られたcategoryと、データセットのlabelとの誤差を計算しています。最初に全てのグリッドの誤差を求め、最後に物体中心を含まない箇所を対象外にしています。



※1 Reshapeはレイヤーの形状を調整しています
※2 グレーの部分はscore, regionに関連するため、他頁で説明をしています
※3 誤差評価時に-1ではエラーとなることの回避策です。最後に中心を含まない箇所を誤差評価の対象外にしているため、学習への影響はありません

Trainingタブ、Validationタブのregion部分の解説

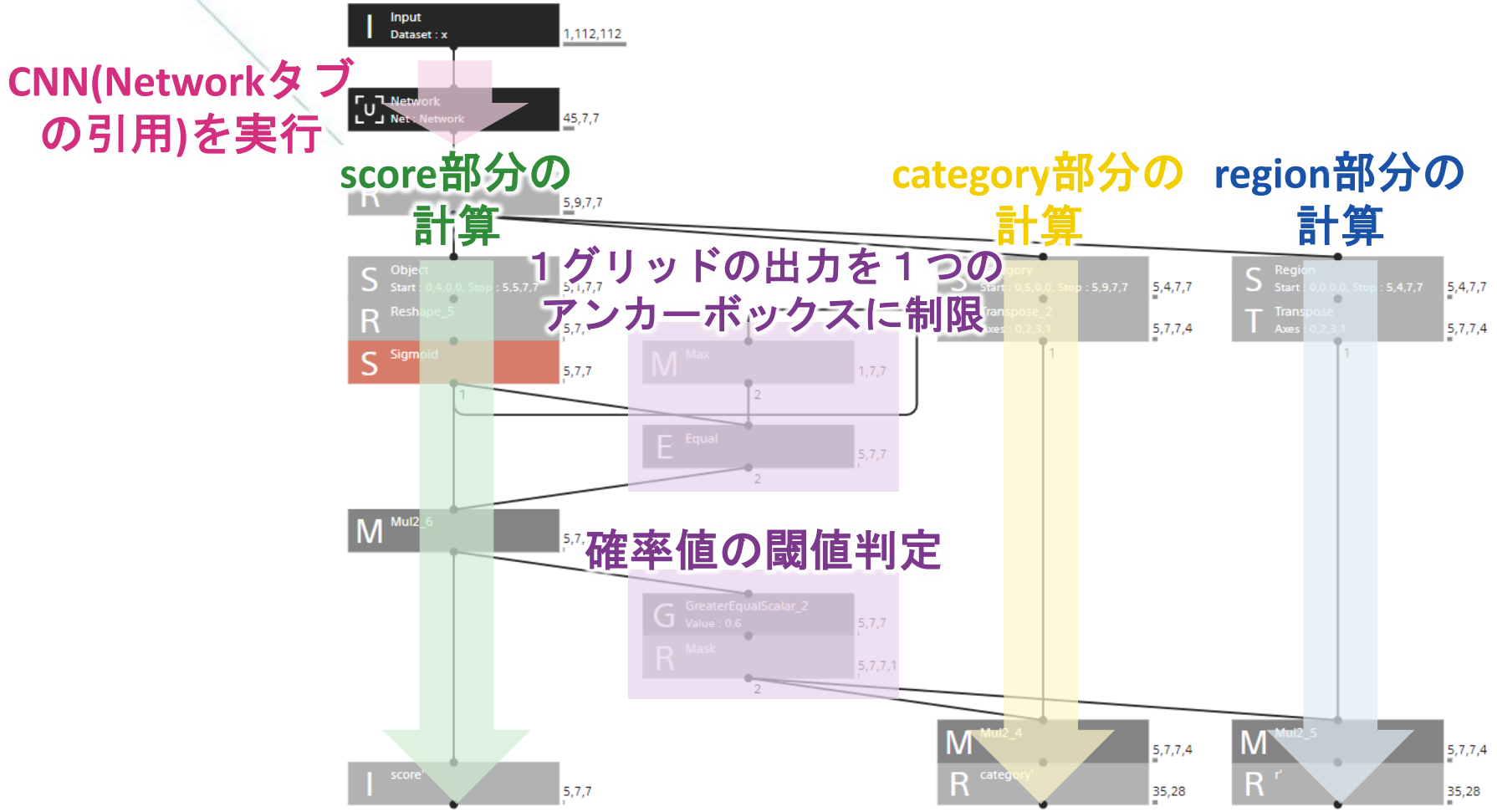
入力画像からCNN(Networkタブの引用)を通して得られたregionと、データセットのregionとの誤差を計算しています。最初に全てのグリッドの誤差を求め、最後に物体中心を含まない部分を対象外にしています。



※1 Reshapeはレイヤーの形状を調整しています
※2 グレーの部分はscore, labelに関連するため、他頁で説明をしています

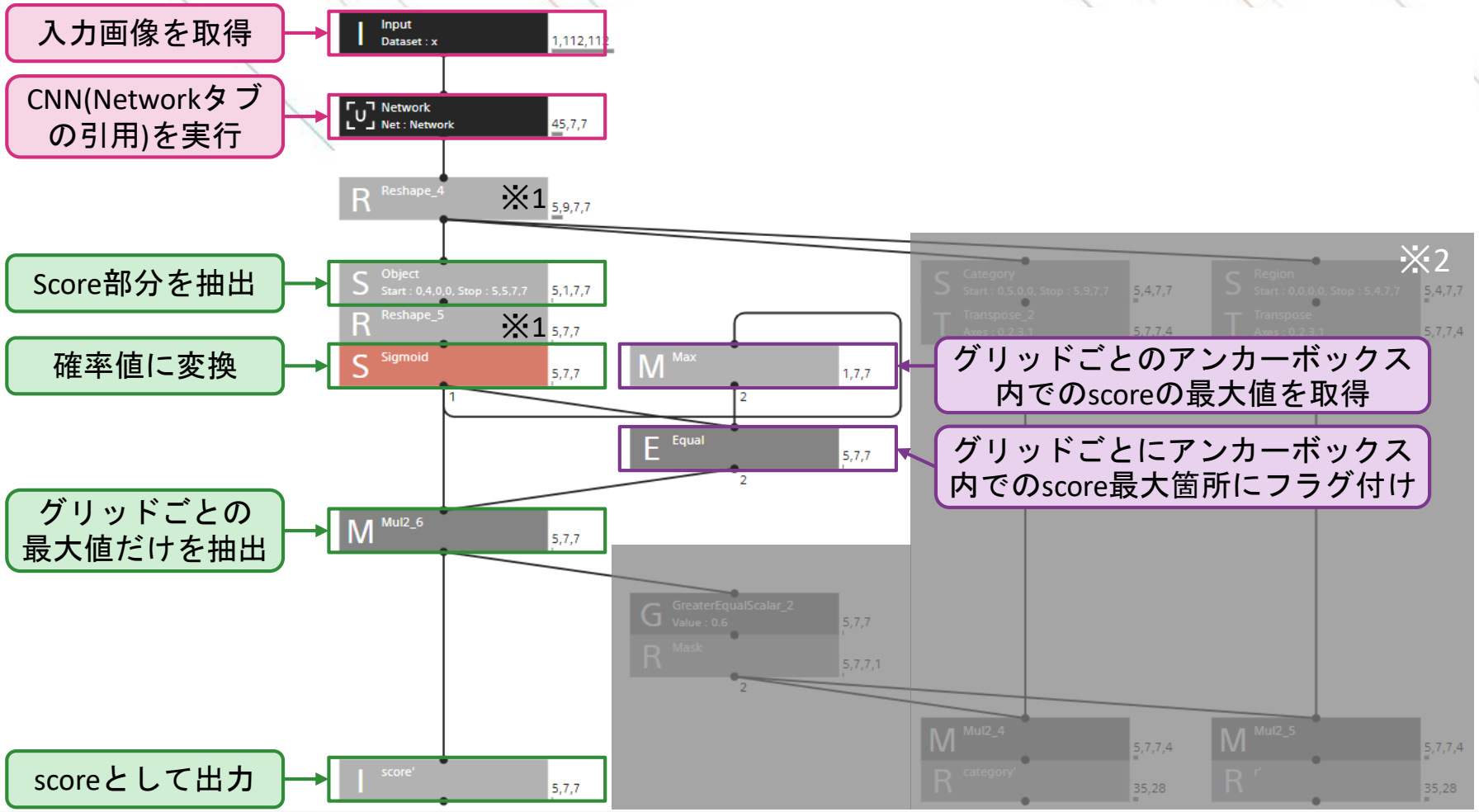
Runtimeタブの概要

CNN(Networkタブの引用)からの出力をscore、category、regionに分割し、それぞれ出力しています。
同一物体の複数のアンカーボックスでの検出を避けるため、１グリッドの出力を１つのアンカーボックスに制限しています。また、確率値の閾値判定を行い、確度が高いもののみを出力しています。



Runtimeタブのscore部分の解説

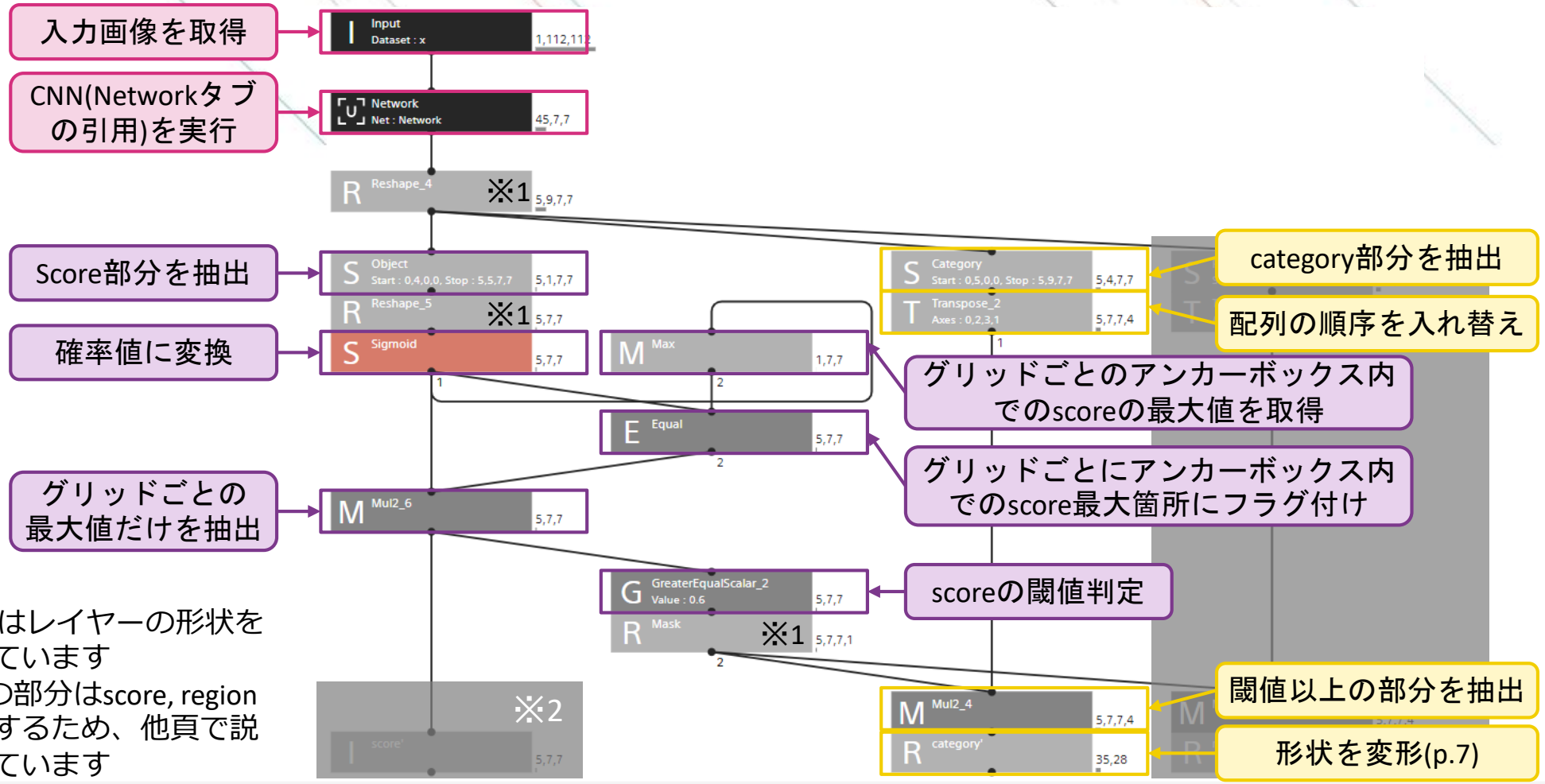
入力画像からCNN(Networkタブの引用)を通して得られたscoreを出力しています。
各グリッドで確率値の最も高いアンカーボックスの確率値だけを出力しています。



※1 Reshapeはレイヤーの形状を調整しています
※2 グレーの部分は,category regionに関連するため、他頁で説明をしています

Runtimeタブのcategory部分の解説

入力画像からCNN(Networkタブの引用)を通して得られたcategoryを出力しています。
各グリッドでアンカーボックス間のscoreの最大値が閾値を超えた箇所のcategoryだけを出力しています。



※1 Reshapeはレイヤーの形状を調整しています
※2 グレーの部分はscore, regionに関連するため、他頁で説明をしています

Runtimeタブのregion部分の解説

入力画像からCNN(Networkタブの引用)を通して得られたregionを出力しています。
各グリッドでアンカーボックス間のscoreの最大値が閾値を超えた箇所のcategoryだけを出力しています。