機械学習エンジニアコース
Sprint

－ セグメンテーション1 －

DIVE INTO CODE

ここで実現しようとしてる事

セマンティックセグメンテーションとは

画像の中の画素(pixel)単位で
どのクラスに属するか分類する

画像認識を実現するために必要なタスク

コンピュータビジョン領域の

画像認識（visual recognition）タスク：

画像分類（a）：

目的は、特定の画像内のオブジェクトの意味のカテゴリ(semantic categories)を認識すること。

物体検出（b）：

目的は、オブジェクトカテゴリを認識するだけでなく、バウンディングボックスによって各オブジェクトの位置(location)予測。

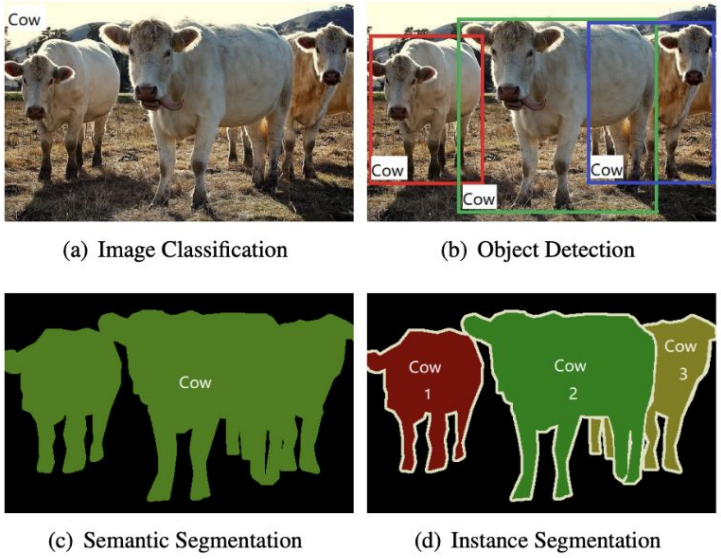
セマンティックセグメンテーション（c）：

目的は、ピクセル単位で分類子（classifiers）を予測して、特定のカテゴリラベルを各ピクセルに割り当てること。物体検出と異なり、セマンティックセグメンテーションは同じカテゴリの複数のオブジェクトを区別しない。

インスタンスセグメンテーション（d）：

物体検出の特殊な形態と見なすことができ、目的は、バウンディングボックスによってオブジェクトの位置を特定する代わりに、ピクセルレベルで位置を特定すること。

https://arxiv.org/pdf/1908.03673.pdf



(a) Image Classification (b) Object Detection

(c) Semantic Segmentation (d) Instance Segmentation

物体検出は何をしてたか

ディープラーニング前の物体検出：

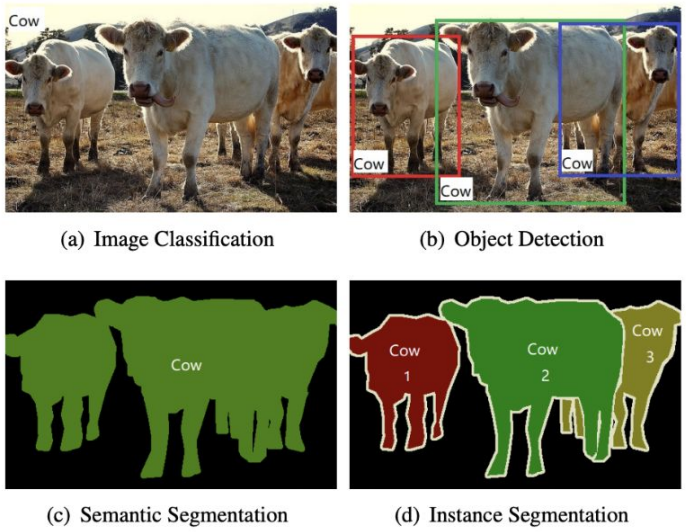
初期段階では、物体検出のパイプラインは3つのステップに分割されていた。

１提案の生成
２特徴ベクトルの抽出
３領域分類

まず最初のステップにおける提案とは、画像内の領域を検索して対象を含む可能性のある領域を見つけることである。これらの場所は、関心領域（ROI）とも呼ばれていた。
検索で行なっていることは、入力画像のサイズを異なるスケールに変更し、マルチスケールなスライディングウィンドウ（sliding windows）を使用して画像全体をスキャンすることだった。

次のステップでは、画像の各所で、スライディングウィンドウから固定長の特徴ベクトルを取得して、その領域を識別する意味（セマンティック）情報を取得していた。

最後のステップでは、分類器（一般的にはSVMを使用）を学習し、対象領域にクラスラベルを割り当てる。



(a) Image Classification (b) Object Detection

(c) Semantic Segmentation (d) Instance Segmentation

https://arxiv.org/pdf/1908.03673.pdf

最初の頃の考え

試しに、画像分類モデルで物体検出をやってみよう。

まず画像を分割します（対象の位置とは関係なく分割）。

次に、全ての領域にCNNをかけ、領域に対してクラス分類を行います。

それから検出された物体を持つオリジナル画像に戻すためにこれらの領域を結合します。

1. First, we take an image as input:

2. Then we divide the image into various regions:

https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

最初の頃の考え

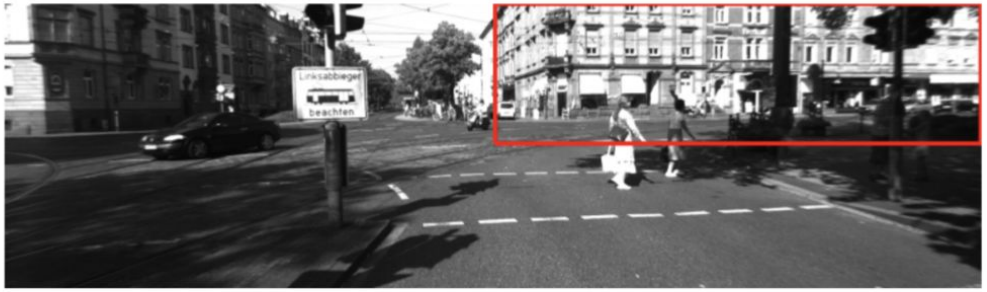
困った

問題点：

ある対象がその画面においては空間の大部分を占めている（しかも画面から切れていたり）一方で、同じ対象が異なる画面においてはほんの数パーセントを占めているだけかもしれない。

写り方によって領域ごとにその対象の形状や部分が異なるかもしれない。

3. We will then consider each region as a separate image.
4. Pass all these regions (images) to the CNN and classify them into various classes.
5. Once we have divided each region into its corresponding class, we can combine all these regions to get the original image with the detected objects:



最初の頃の考え

どうしよう

結果：

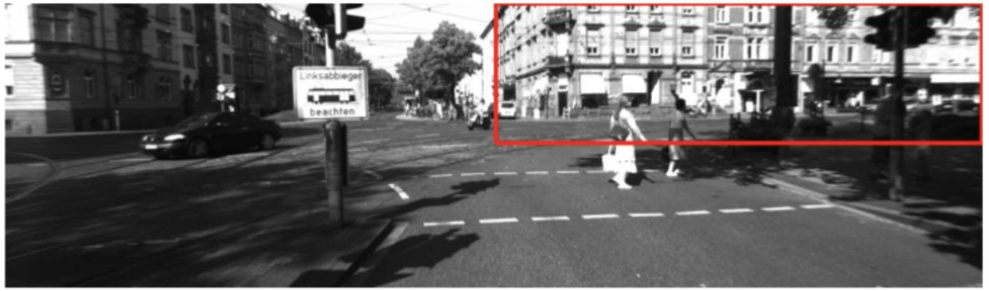
莫大な計算量を必要とする多くの領域分割が必要となる。

画像を分割して、片っ端からCNNをかけたら

すごく時間がかかりそう。。

なんとかしてこの領域を減らしたいなあ。

3. We will then consider each region as a separate image.
4. Pass all these regions (images) to the CNN and classify them into various classes.
5. Once we have divided each region into its corresponding class, we can combine all these regions to get the original image with the detected objects:



アイデア：

。。領域を選べばいいんじゃないかしら？

あ、だから領域提案というステップがあるのかー。

考えられた手法

というわけで

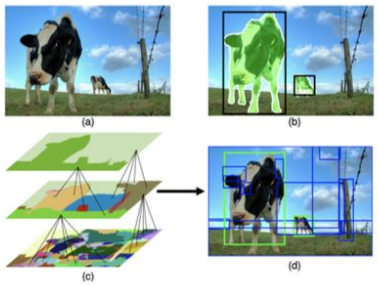
R-CNN(2014)

Region Proposals（領域提案）：

selective searchを用いた**物体認識(Object Recognition)**を行い、領域提案を実現した。

CNNアーキテクチャはAlexNet(2012)がベース。入力は227×227にする。

https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf



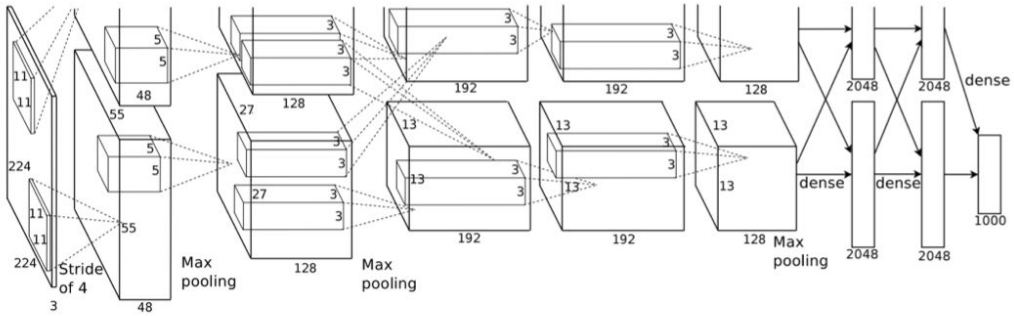


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

考えられた手法

R-CNN(2014)

Warping（画像変換）：

画像の歪みとり

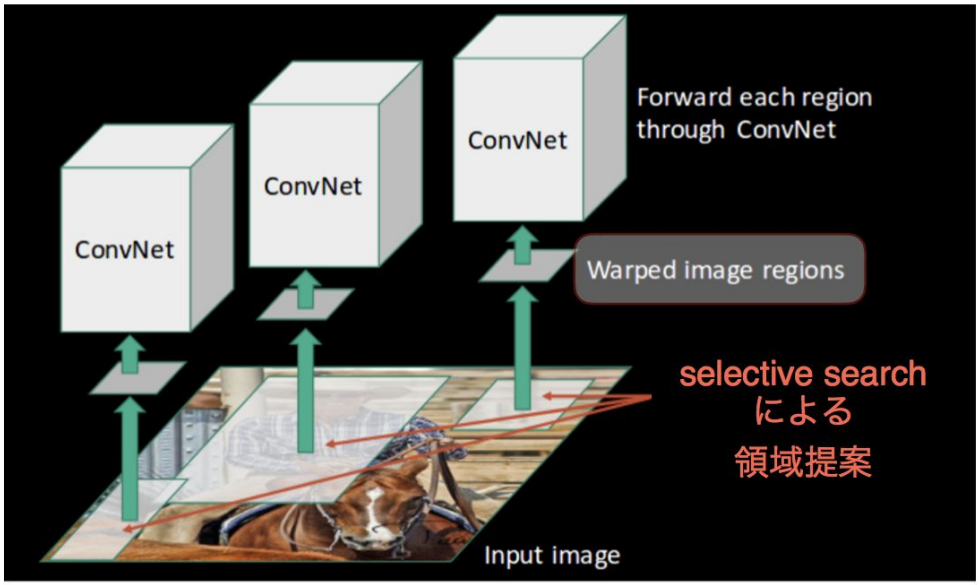
https://www.youtube.com/watch?v=6DtzCZVorSw

CNN（特徴抽出）：

提案領域が2,000あれば、画像ごとにCNNを2,000回実行する。

結果：

1画像ごとに40-50秒かかる。



考えられた手法

selective search：

膨大な数のBOXを提案して物体認識を行う。
Selective Searchはセグメンテーション（注：DCNNによるセグメンテーションとは別物）アルゴリズムを用いている。

「すべてのスケールで（対象の）**位置**を獲得するもっとも自然な方法は、階層的セグメンテーションアルゴリズム」
「初期領域から開始して、最も類似した2つの領域を繰り返しグループ化する貪欲なアルゴリズムを使用します」
「この新しい地域とその近隣地域間の類似性を計算します。」
('Segmentation as Selective Search for Object Recognition' より)

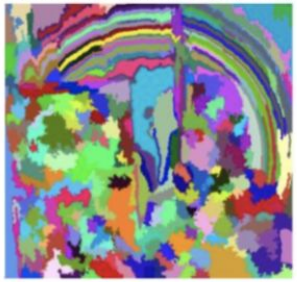
https://www.koen.me/research/pub/vandesande-iccv2011.pdf

http://www.huppelen.nl/publications/selectiveSearchDraft.pdf

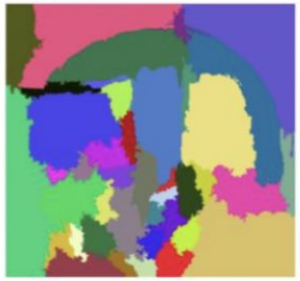
https://ivi.fnwi.uva.nl/isis/publications/bibtexbrowser.php?key=UijlingsIJCV2013&bib=all.bib

- It first takes an image as input:

- Then, it generates initial sub-segmentations so that we have multiple regions from this image:



- The technique then combines the similar regions to form a larger region (based on color similarity, texture similarity, size similarity, and shape compatibility):



- Finally, these regions then produce the final object locations (Region of Interest).

考えられた手法

Fast R-CNN(2015)

CNN（特徴抽出）：

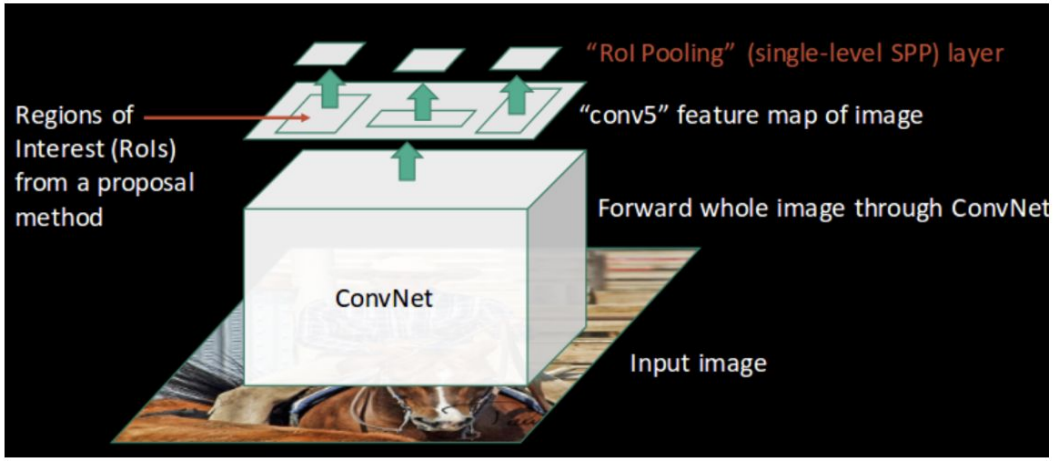
画像に対し一つの特徴マップを出力

Region Proposals（領域提案）：

特徴マップにselective searchを行う。

結果：

selective searchはやはり時間がかかる。1画像あたり2秒。



考えられた手法

Faster R-CNN(2015)

CNN（特徴抽出）：

画像に対し一つの特徴マップを出力

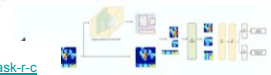
Region Proposal Network（領域提案ネットワーク）：

selective searchに代わり、k個のanchor boxesを持つ**sliding window**（1×1のCNNのこと）を用いる。物体（not クラス）かどうかの確率を返し、さらにground truthにfitするanchor box座標を回帰する。（RPN lossを使用）

https://medium.com/lsc-psd/faster-r-cnn%E3%81%AB%E3%81%8A%E3%81%91%E3%82%8Brpn%E3%81%AE%E4%B8%96%E7%95%8C%E4%B8%80%E5%88%86%E3%81%8B%E3%82%8A%E3%82%84%E3%81%99%E3%81%84%E8%A7%A3%E8%AA%AC-dfc0c293cb69

RoI pooling layer：

RPNのproposalsをトリミング（スケーリング）する

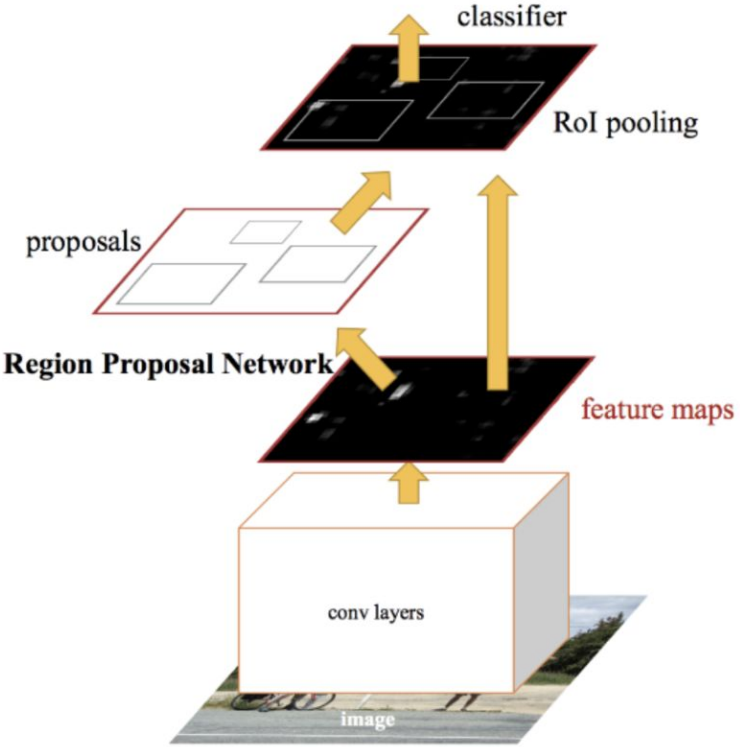


https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

結果：

selective searchよりはマシだがproposalsの計算も割と時間がかかる。1画像あたり0.2秒。複数のシステムから動いているので、他のシステムにパフォーマンスが依存する。

https://medium.com/@whatdhack/a-deeper-look-at-how-faster-rcnn-works-84081284e1cd



セマンティックセグメンテーション

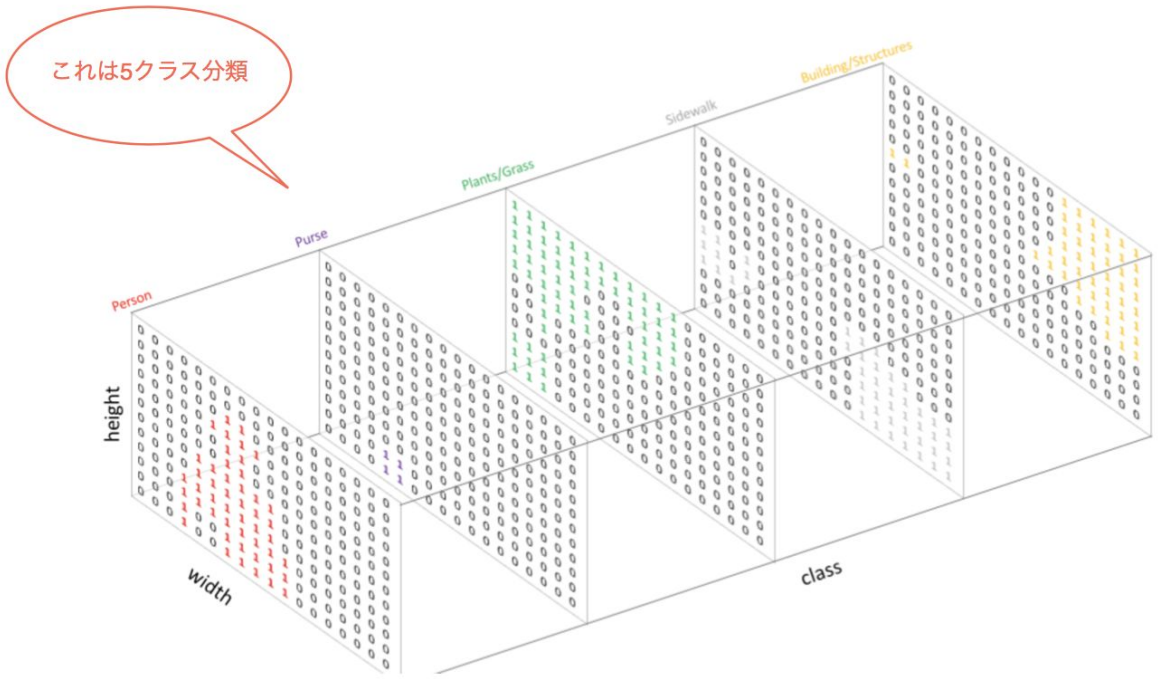
セマンティックセグメンテーション

Bboxレベルのアルゴリズムでは、四角形のボックスで対象の位置を特定（localization）していたが、マスク（＝意味のある形でピクセルをグルーピングする）レベルのアルゴリズムで対象をセグメント化するには、より正確にピクセル単位の位置においてクラスを識別（密に予測する必要性からdense predictionとも呼ばれる）しなければならない。

セマンティックセグメンテーション

最後に出力されるFeatureMapはクラス数分のチャネルを持つ

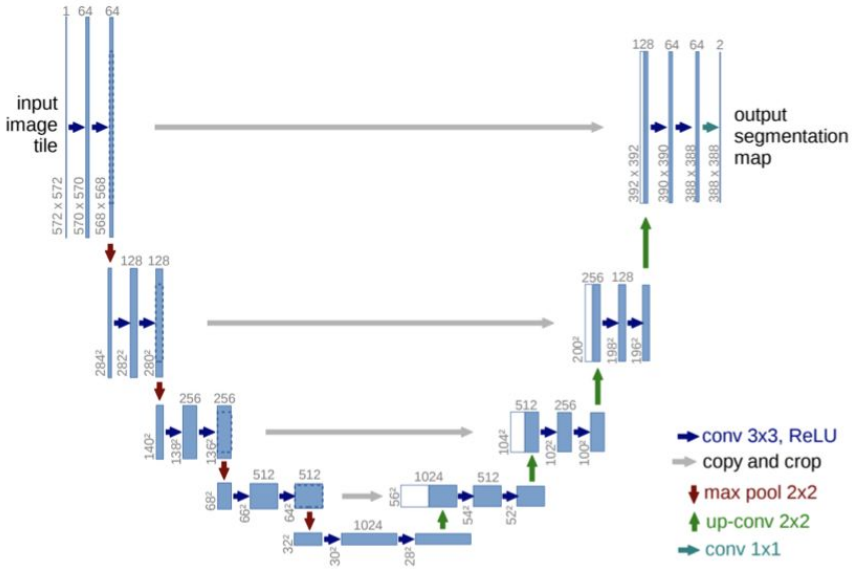
softmaxを用いてピクセルごとにチャネル方向に向かって合計が1になるように確率を出力する



セマンティックセグメンテーション

U-Net

論文を読んでコードリーディングをしよう



セグメンテーション1
完