

不可視AR

～Faster-RCNN～

ER17076 安井 理

目次

- ① Faster-RCNNについて
- ② RPN
- ③ Anchor
- ④ まとめ・課題

1. Faster-RCNN

・Faster-RCNNとは

- 特徴マップを入力として物体領域候補を生成するRPN (Resion Proposal Network)
- ①: 特徴マップを出す ⇒ ②: 「矩形」と「背景OR物体」を学習 ⇒ ③: ②が具体的に何なのかを学習

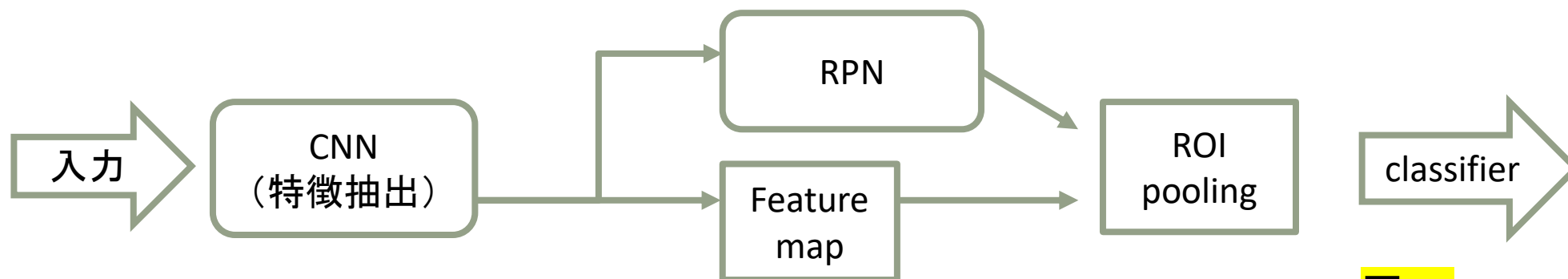


図: 1

1. Faster-RCNN

- Faster-RCNNとは

- 特徴マップを入力として物体領域候補を生成するRPN (Resion Proposal Network)

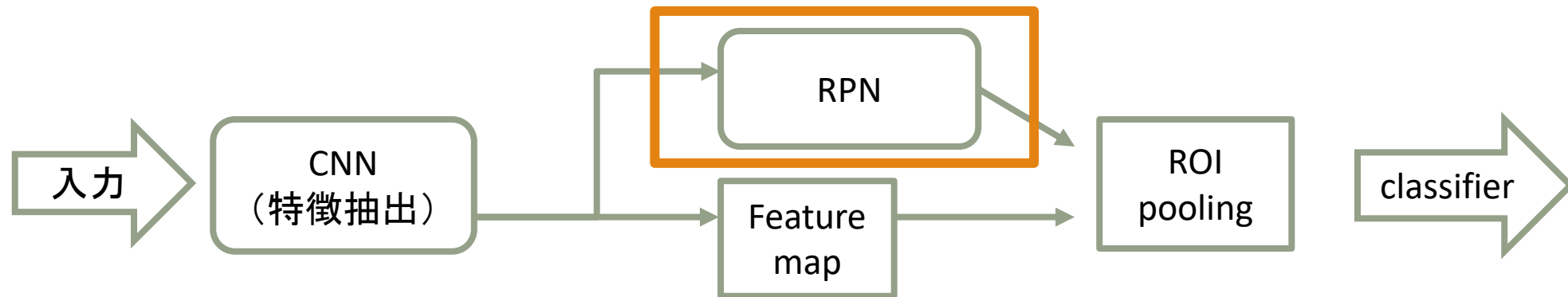


図: 1

1. Faster-RCNN

・Faster-RCNNとは

- ①: 特徴マップを出す ⇒ ②: 「矩形」と「背景OR物体」を学習 ⇒ ③: ②が具体的に何なのかを学習

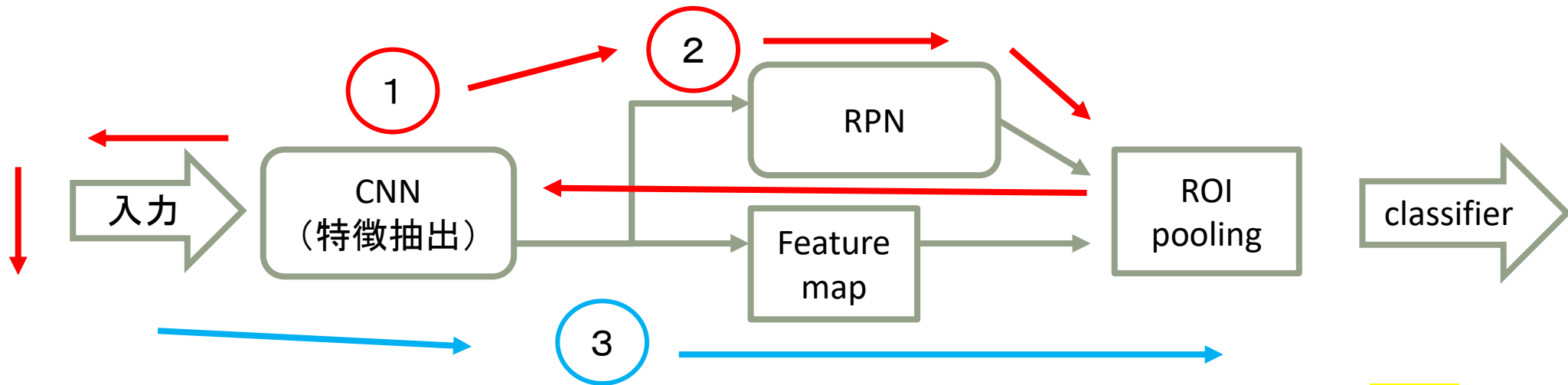


図: 1

2.RPN(Resion Proposal Network)

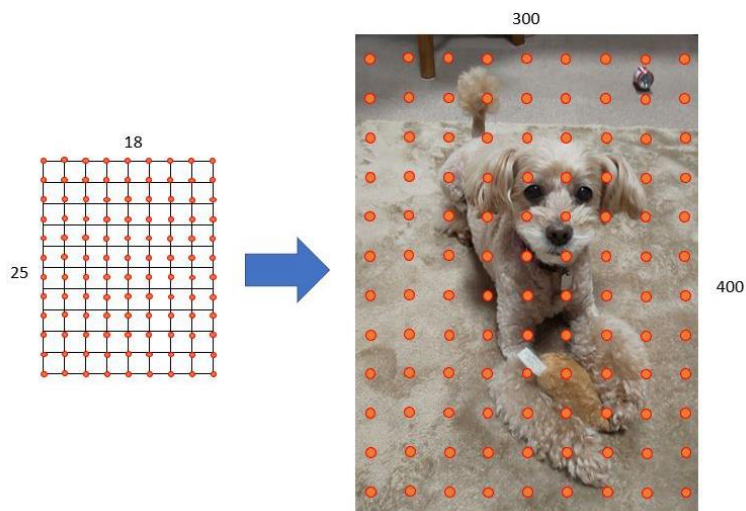
- RPNは小さなニューラルネットワーク
- 物体領域候補(「物体の場所」・「矩形の大きさ」)を学習する
- CNNで抽出した特徴マップに対しBounding BoxとObjectnessを表すスコアを出力

2.RPN(Resion Proposal Network)

- ①入力画像をVGGやZFNetなどの畳み込みを行い特徴マップを抽出
- ②1に対して 3×3 畳み込み(512チャンネル) $\rightarrow 1 \times 1$ の畳み込みで「背景OR物体」用のアウトプット「ground truthとのズレ」用のアウトプット(36チャンネル)
- ③「物体OR背景」 \rightarrow 分類問題・バイナリクロスエントロピー
「ズレ」 \rightarrow 回帰問題・Lノルム(絶対値誤差)
をベースとした誤差関数で①を入力②を出力モデルとして学習

3. Anchor

- 矩形の中心点
- 使われているCNNでは1/16のサイズになる。
- 入力画像に対し16ピクセルに一つの割合の点・まばらかつ均等



例 画像サイズ : 300×400
Anchor : $18 \times 25 = 450$ 点

参考 : <https://medium.com/lsc-psd/faster-r-cnn> におけるrpnの世界一分かりやすい解説-dfc0c293cb69

3. Anchor

- Anchor BoxはおいたAnchorを中心とした矩形(候補領域)
- 1つのAnchor(座標)に対し3(長さ) × 3(比) = 9つのBoxが用意される
 - 注意: 長さ64に対して | 1:1 ⇒ 64 × 64 (4096) | 1:2 ⇒ 45 × 91 (4096) | 2:1 ⇒ 91 × 45 (4096) |

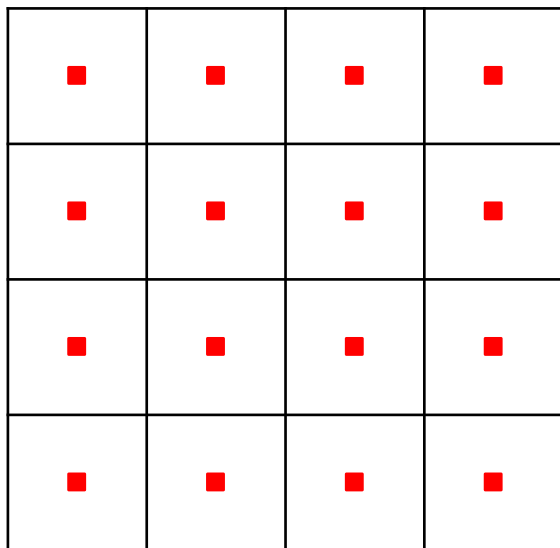
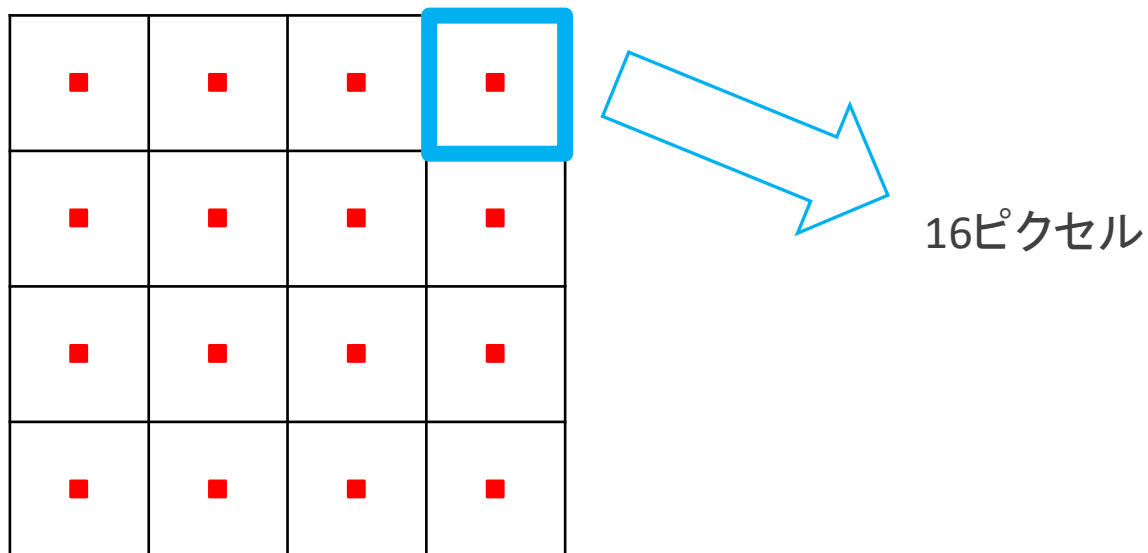


図: 2

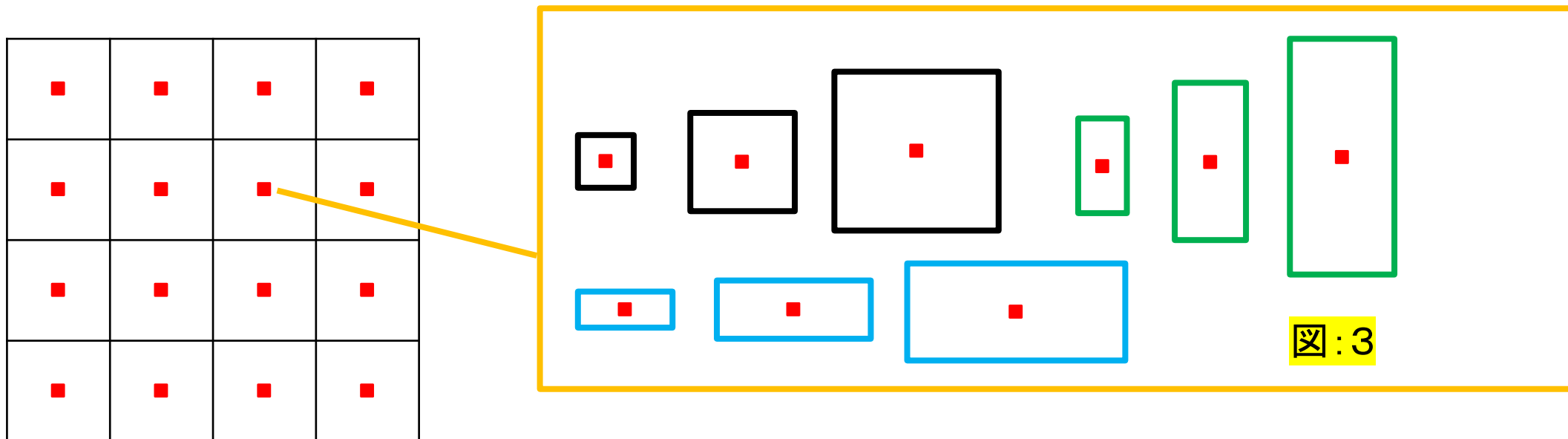
3. Anchor

- Anchor BoxはおいたAnchorを中心とした矩形(候補領域)
- 1つのAnchor(座標)に対し3(長さ) × 3(比) = 9つのBoxが用意される
 - 注意: 長さ64に対して | 1:1⇒64 × 64 (4096) | 1:2⇒45 × 91 (4096) | 2:1⇒91 × 45 (4096) |



3. Anchor

- 1つのAnchor(座標)に対し3(長さ) × 3(比) = 9つのBoxが用意される



3. Anchor

- 1つのAnchor(座標)に対し3(長さ) × 3(比) = 9つのBoxが用意される

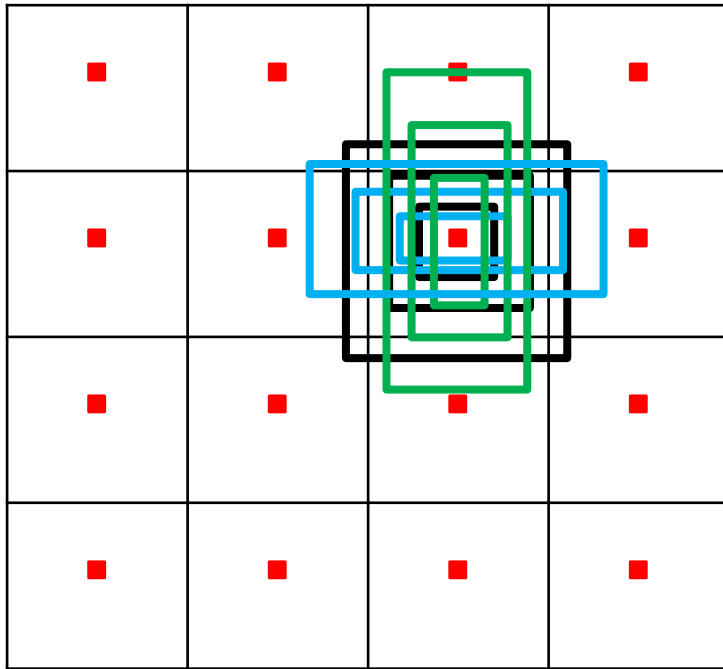


図: 4

3. Anchor

- 「Ground Truth」と「Anchor Boxes」を比べ2つの学習
- ①AnchorBoxは「物体OR背景」か学習
- ②物体の時「GroundTruthとのズレ」を学習

3. Anchor

- ①AnchorBoxは「物体OR背景」か学習
- Ground TruthとAnchor BoxesのIoUを計算
 - IoU: Intersection/Union (共通領域/領域合計)

3. Anchor

- ①AnchorBoxは「物体OR背景」か学習

☆IoU<0.3「背景」 IoU>0.7「物体」 それ以外は使わない

- 図4のように1つのAnchorに9つAnchorBoxを用意
- 1座標(Anchor) AnchorBox 9×2 「背景OR物体」 \Rightarrow 18クラス用意
- IoUが該当したものに1を出力

3. Anchor

- ②物体の時「GroundTruthとのズレ」を学習
 - ズレ = 「x座標」「y座標」「横幅」「縦幅」の4つ
 - 1座標(Anchor) AnchorBox 9×4 「ズレ」の36クラス用意
 - 物体の時 = $IoU > 0.7$ のときに対応する値を出力

4.まとめ

- RPN

- ①入力画像の畳み込みを行い特徴マップを抽出
- ②「背景OR物体」用のアウトプット
「ground truthとのズレ」用のアウトプット(36チャンネル)
- ③ ①を入力②を出力モデルとして学習

4.まとめ

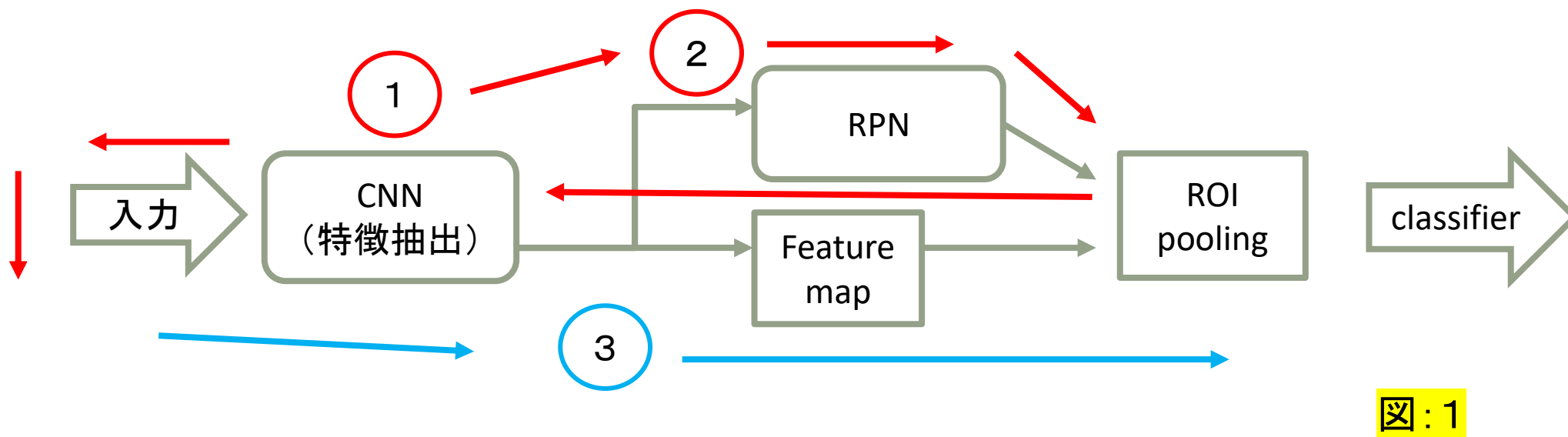
▪ Faster-RCNN

- ①入力画像の畳み込み(RPN①と同じ重み)を行い特徴マップを抽出
- ②RoI Poolingで特徴を固定長の長さにする
- ③全結合層に2回通し、クラス分類・矩形のズレ の2種類の出力
- ④ ①を入力②を出力のタスクとして学習

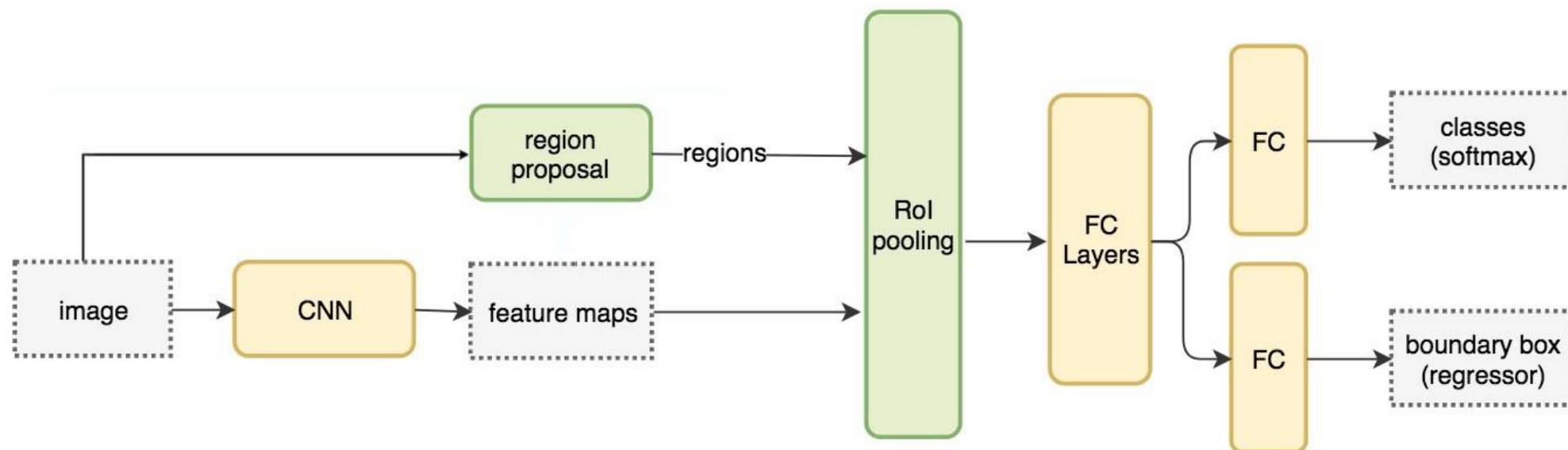
4.まとめ

▪ Faster-RCNN

- ①→②→③で1エポック...？



4.まとめ



引用: https://medium.com/@jonathan_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9

4. 課題

- Google colaboryで動かせていないので、動かせるようにする
- AnchorBoxは正方形だけのほうが良いのではないか？
- 理解できていないことが多い

4. 参考

- [illegible]