深層学習による動画像の連続フレームからの物体検出

工学研究科　情報システム工学分野
T2103288　田邉　英介
指導教員　椋木雅之

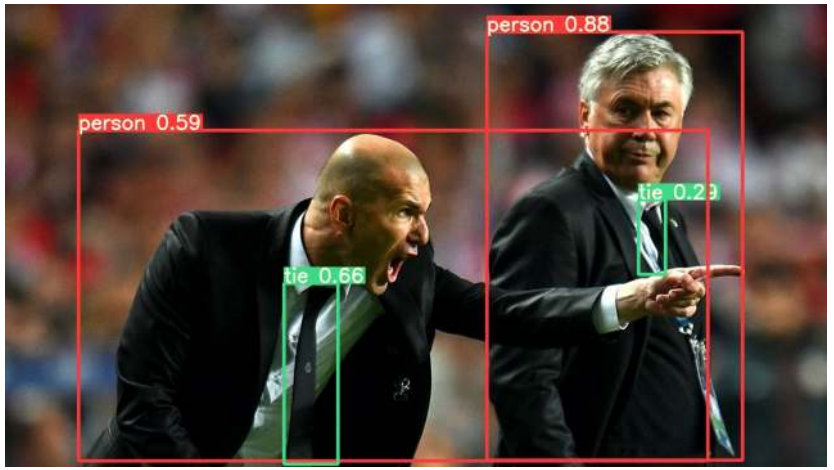
研究背景

物体検出：

動画や単一画像に含まれる特定の物体の位置と範囲を推定する技術

近年、深層学習を用いた物体検出手法が提案されている

大量のデータを学習することにより、
高精度の物体検出が可能



研究背景

物体検出は動画像にも適用されている

動画像：

連続する時刻で取得した画像（フレーム）を時間方向に並べたもの

連続するフレームでの画像内容は変化が小さい

連続での検出に失敗する問題がある

従来研究　YOLO [1]

- 深層学習を用いた物体検出の手法の１つ
- リアルタイムに処理できるが検出漏れが生じる

[1] Joseph Redmon , Santosh Divvala, Ross Girshick , Ali Farhadi “You Only Look Once: Unified, Real-Time Object Detection”, CVPR (2016)

研究目的

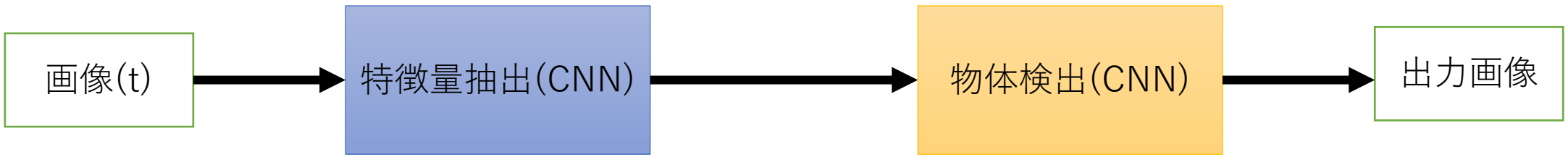
時間方向での物体検出結果を安定させる

時間方向に連続する数フレームの情報を物体検出に利用

従来手法の考え方

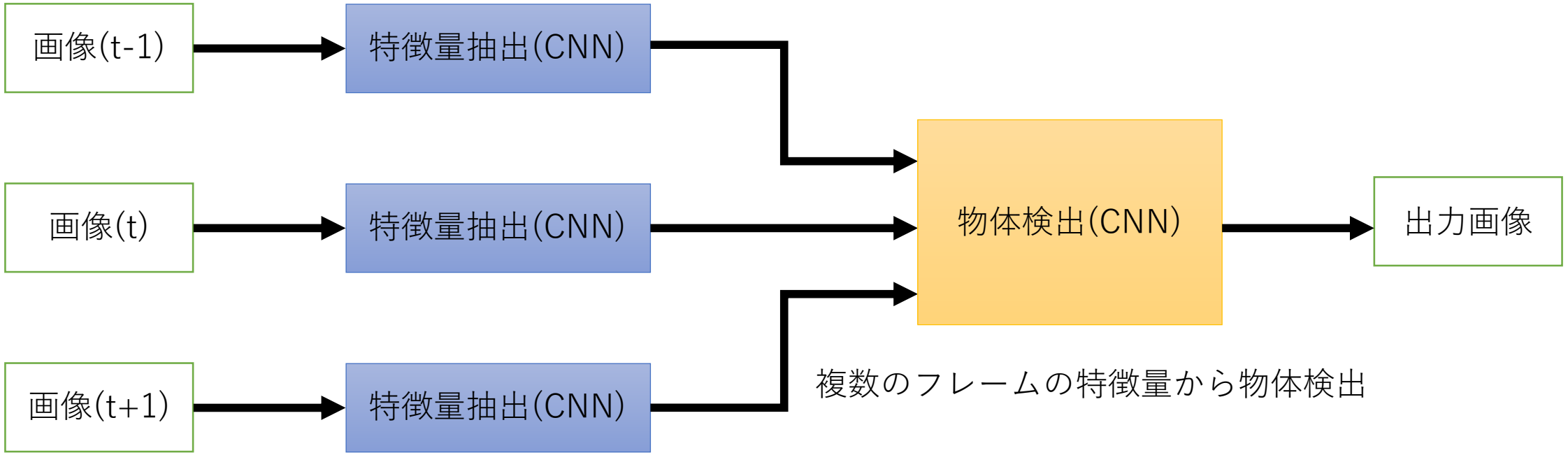
畳み込みネットワーク(CNN)と呼ばれるネットワーク構造を用いる
動画像の場合、各フレームに対して一枚ずつ物体検出処理を適用

前後のフレームの情報は物体検出に利用されない



提案手法の考え方

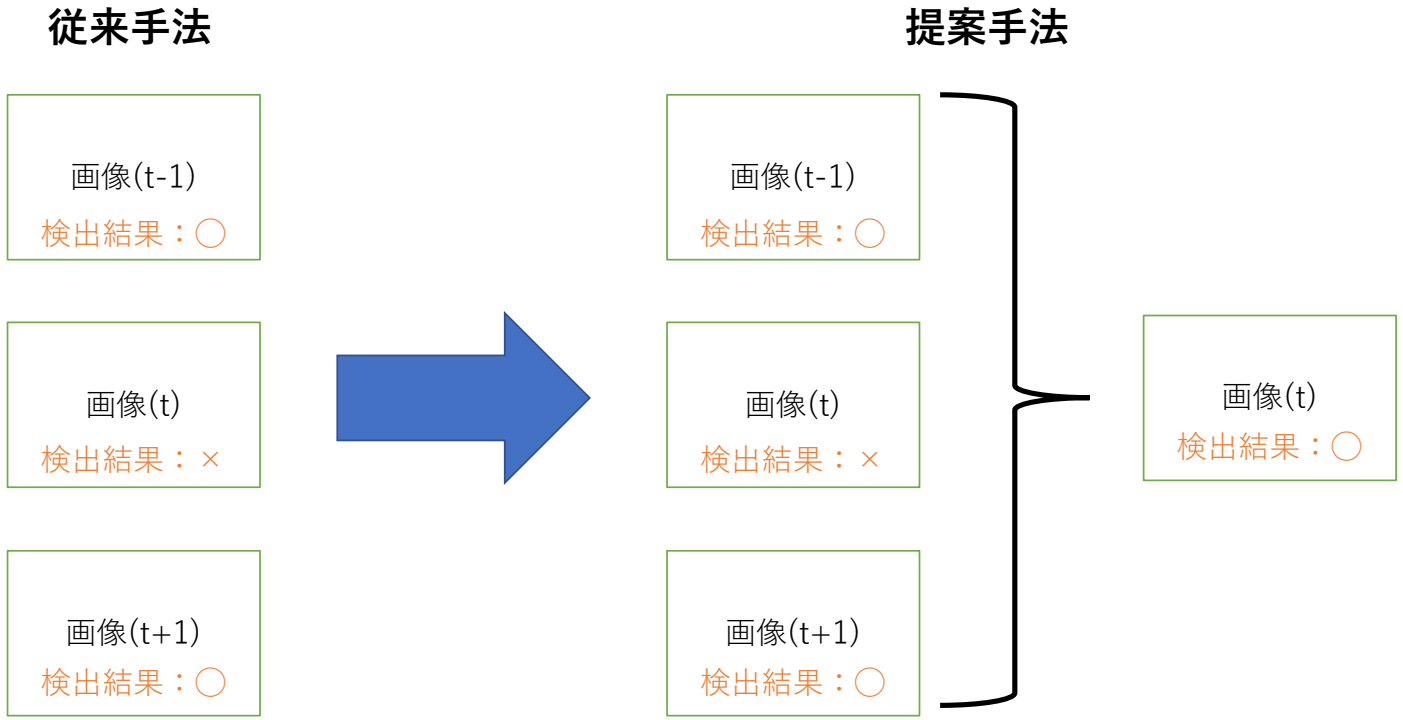
物体検出を適用したいフレームだけでなく、その前後を含めた
連続した３フレームの情報を利用



提案手法の考え方

前後のフレームの情報も物体検出に利用

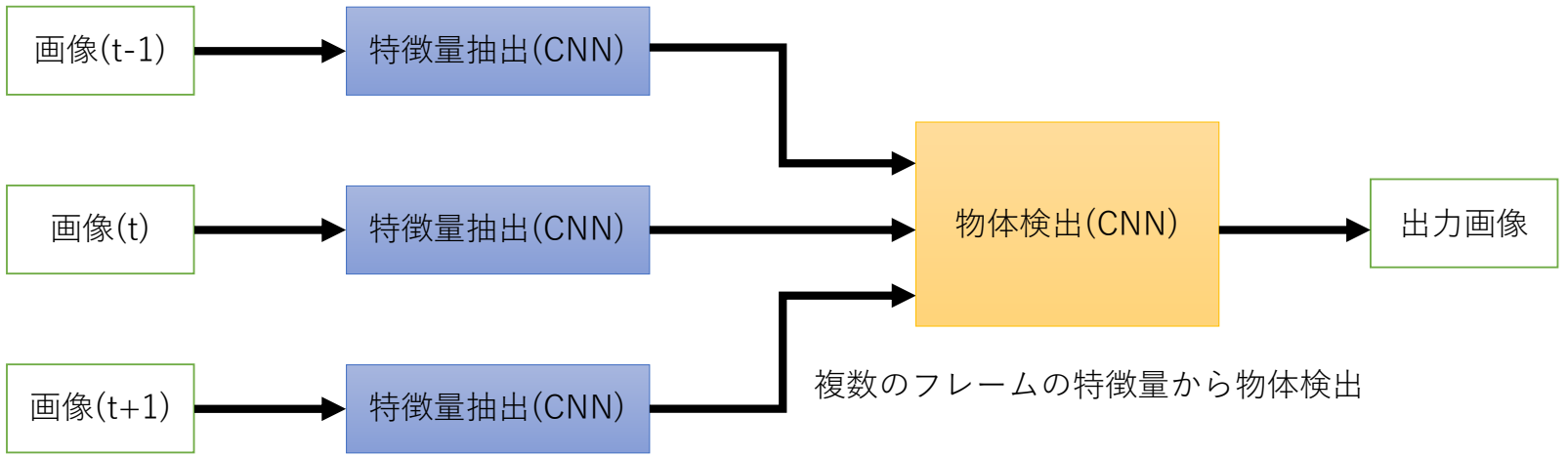
連続して物体検出ができるようになると考える



提案手法のネットワーク構造

- 特徴量抽出器は、学習済みのResNet を使用

- 物体検出ネットワークは、YOLO v3のネットワーク構造を参考
  - 結合した特徴量を扱えるようにチャネル数等を変更



特徴量抽出器　ResNet [2]

- 深層学習を用いた手法
- 層の数が多い

YOLO v3[3]：精度と処理速度を重視してDarkNet-53を採用

提案手法：精度を重視してResNet-152を採用

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [15]	74.1	91.8	7.29	1246	**171**
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

[2] Kaiming He , Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, CVPR (2016)
[3] Joseph Redmon, Ali Farhadi, “YOLOv3: An Incremental Improvement”, arXiv18.04.02767[cs.CV] (2018)

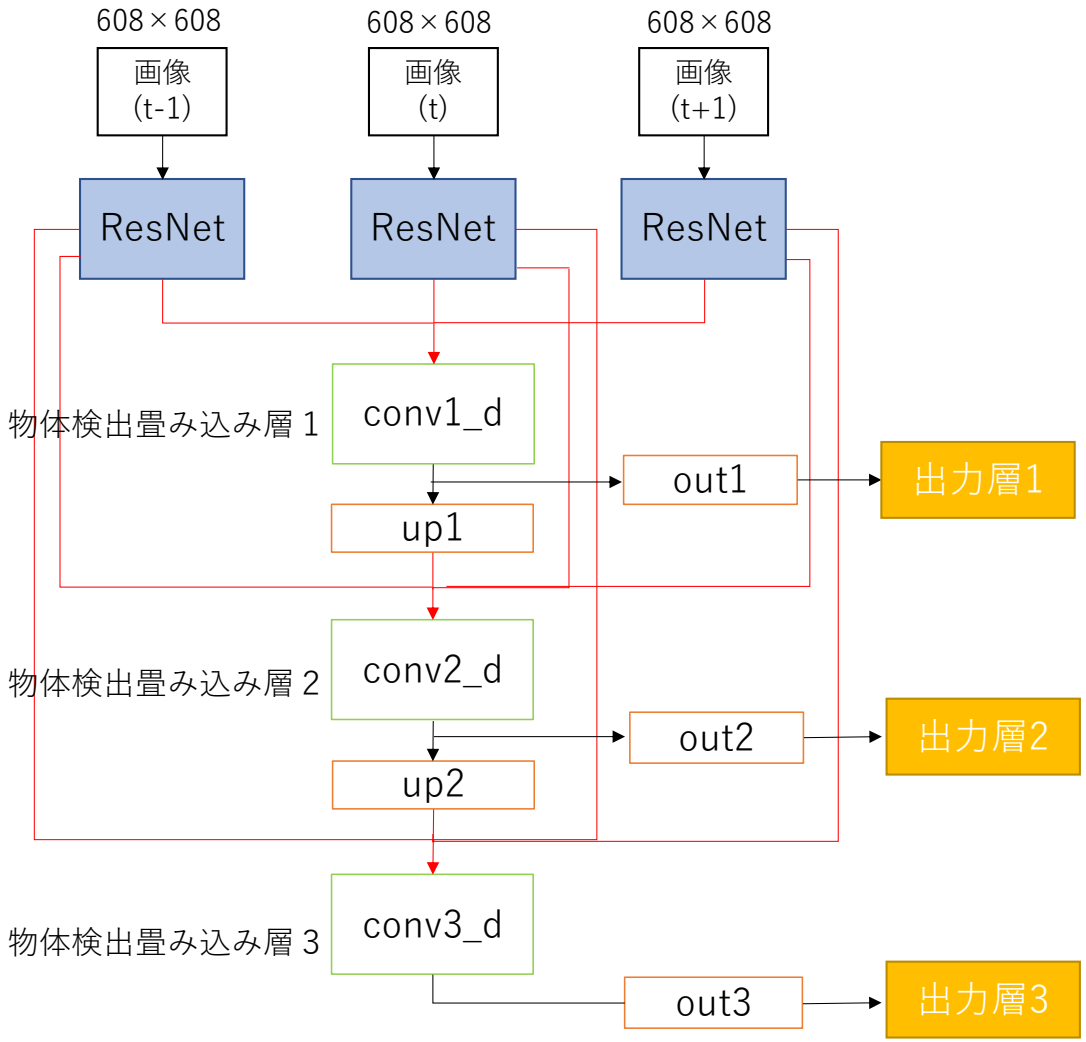
特徴量抽出器　ResNet [2]

- 様々な大きさの物体に対応するために、３段階の大きさの特徴マップを利用
- conv3_x,conv4_x,conv5_xの出力をそれぞれ3フレーム分結合したものを物体検出ネットワークの入力として使用

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

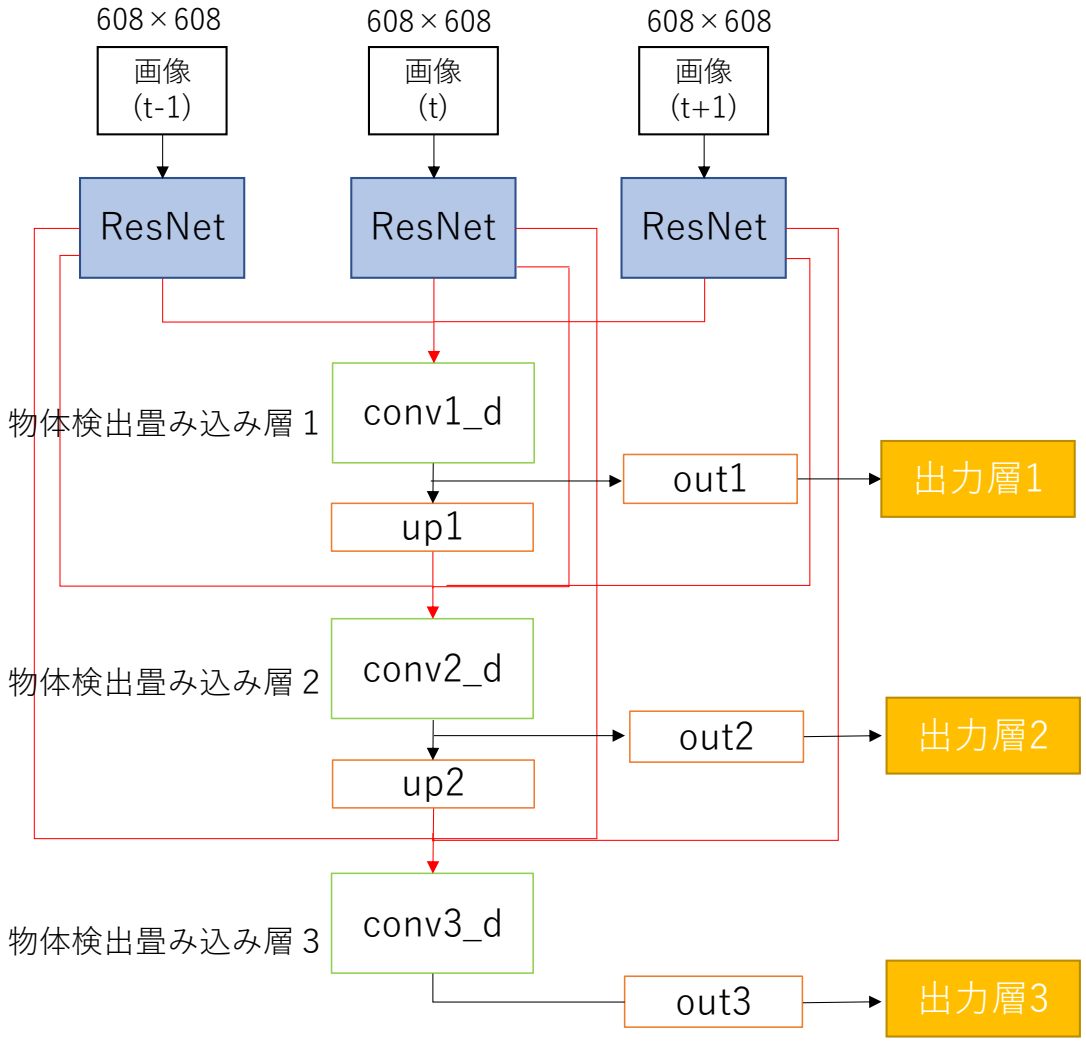
[2] Kaiming He , Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, CVPR (2016)

物体検出ネットワーク



	サイズ	チャネル数
conv1_d	19×19	6144 1024 2048 1024 2048 1024
out1	19×19	2048
up1	38×38	512
conv2_d	38×38	3584 512 1024 512 1024 512
out2	38×38	1024
up2	76×76	256
conv3_d	76×76	1792 256 512 256 512 256
out3	76×76	512

物体検出ネットワーク



３つの出力層がある

出力層では各グリッド内での物体の位置、範囲、種類、評価値を算出して出力

出力結果を統合して、物体検出結果とする

実験

提案手法が従来手法（YOLO v3）と比べて動画像での物体検出がどの程度安定するかを調査する

- 検出する対象物体は馬の１クラス
- 実験データとして35本の馬の動画像を使用

実験

- 動画像をフレームに分解し、10枚の連続フレームの画像群（クリップ）を作成
- 動画像に応じて１つの動画像から３～８個のクリップ
- 合計で1400枚（140クリップ）の画像を使用

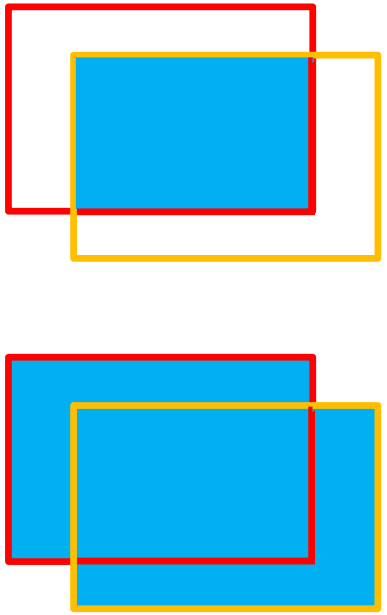
実験１

平均適合率（AP）を用いた比較を行う

AP：m 個の正解ラベルのうち、どのくらいのラベルを検出できているかを平均的にあらわしたもの

APの計算にはIntersection over Union（IoU）を使用

$$\textbf{IoU} = \frac{\text{重複領域}}{\text{全体領域}}$$



IoUの閾値：0.5

実験１

TP：正解したBBox

FP：正解でないBBox

FN：どの検出したBBoxとも紐づいていない正解の矩形

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Precision$(P(r))$とRecall(r)からAPを求める

$$\text{AP} = \int_0^1 P(r)dr$$

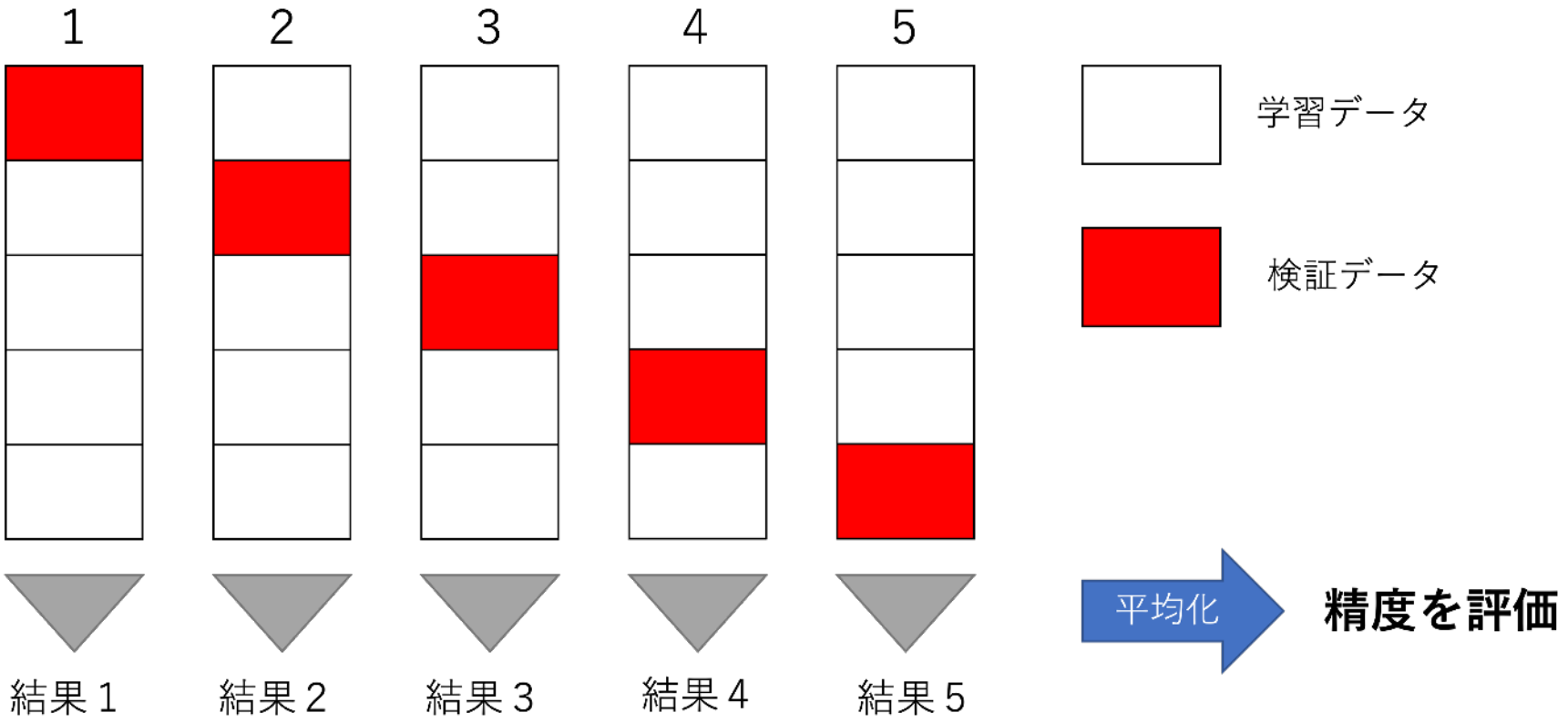
APの最大値は１となり、値が大きいほど検出精度が高い

実験１

- クロスバリデーションの手法により５つのパターンで検証
- 学習回数は10000回
- 学習の途中段階での挙動も比較するために、学習回数3000回、5000回、10000回での結果をパターン毎に比較

クロスバリデーション

35個の動画像を５つに分割し、５パターンの実験から評価



実験１の結果

		パターン１	パターン２	パターン３	パターン４	パターン５	平均
YOLO v3	3000	0.5012	0.3072	0.5746	0.6864	0.3589	0.4857
	5000	0.5873	0.3607	0.6358	0.7072	0.4454	0.5473
	10000	0.6162	0.4214	0.6838	0.6460	0.4933	0.5721
提案手法	3000	0.5366	0.2045	0.6440	0.6112	0.3407	0.4674
	5000	0.6307	0.3588	0.6702	0.6917	0.4860	0.5675
	10000	0.6362	0.4325	0.7240	0.7050	0.5533	0.6102

実験１の結果

		パターン１	パターン２	パターン３	パターン４	パターン５	平均
YOLO v3	3000	0.5012	0.3072	0.5746	0.6864	0.3589	0.4857
	5000	0.5873	0.3607	0.6358	0.7072	0.4454	0.5473
	10000	0.6162	0.4214	0.6838	0.6460	0.4933	0.5721
提案手法	3000	0.5366	0.2045	0.6440	0.6112	0.3407	0.4674
	5000	0.6307	0.3588	0.6702	0.6917	0.4860	0.5675
	10000	0.6362	0.4325	0.7240	0.7050	0.5533	0.6102

実験１の結果　補足

MSCOCOで学習を行ったYOLO v3と提案手法の比較

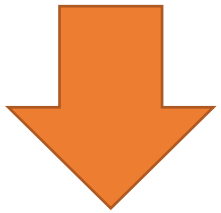
MSCOCO：馬のクラスを含む80種類のクラスの約10万枚の画像で構成されるデータセット

	パターン１	パターン２	パターン３	パターン４	パターン５	平均
YOLO v3 (MSCOCO)	**0.7571**	**0.8396**	**0.9021**	**0.9978**	**0.7026**	**0.8398**
提案手法	0.6362	0.4325	0.7240	0.7050	0.5533	0.6102

実験１の考察

- 同じ実験データで学習したYOLO v3と比べると結果の値は上回った
- MSCOCOで学習したYOLO v3と比べると大幅に値が下回った

提案手法では、１つのパターンでクリップが約110種類(動画像：28本)ほどしか学習に使われない

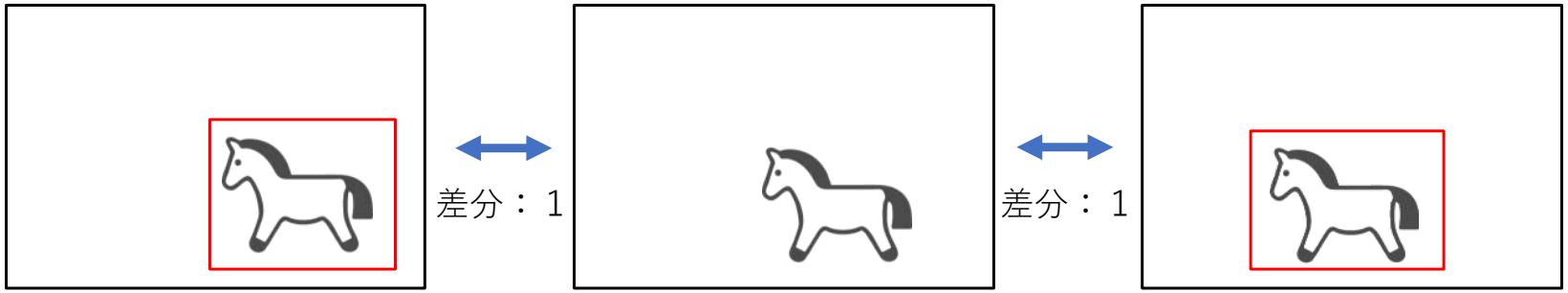


実験データの数が少なかったことが、結果の良くなかった原因の一つとして考えられる

実験２

時間方向での安定性を比較する

- あるフレームで検出されたBBoxの数とその次のフレームで検出されたBBoxの数の差分で比較
- 対応する馬ごとに140クリップ分の差分を取り、合計数で比較
- 信頼度スコアの閾値は0.4



実験２の結果

	差分の合計数
YOLO v3	251
提案手法	**133**

提案手法の方がYOLO v3よりも差分の合計数が少ない

提案手法の方が時間方向での物体検出結果が安定している

実験２　３つの例

実験データの中から３つの例を選び、10フレームで検出される矩形領域の数の比較を行う

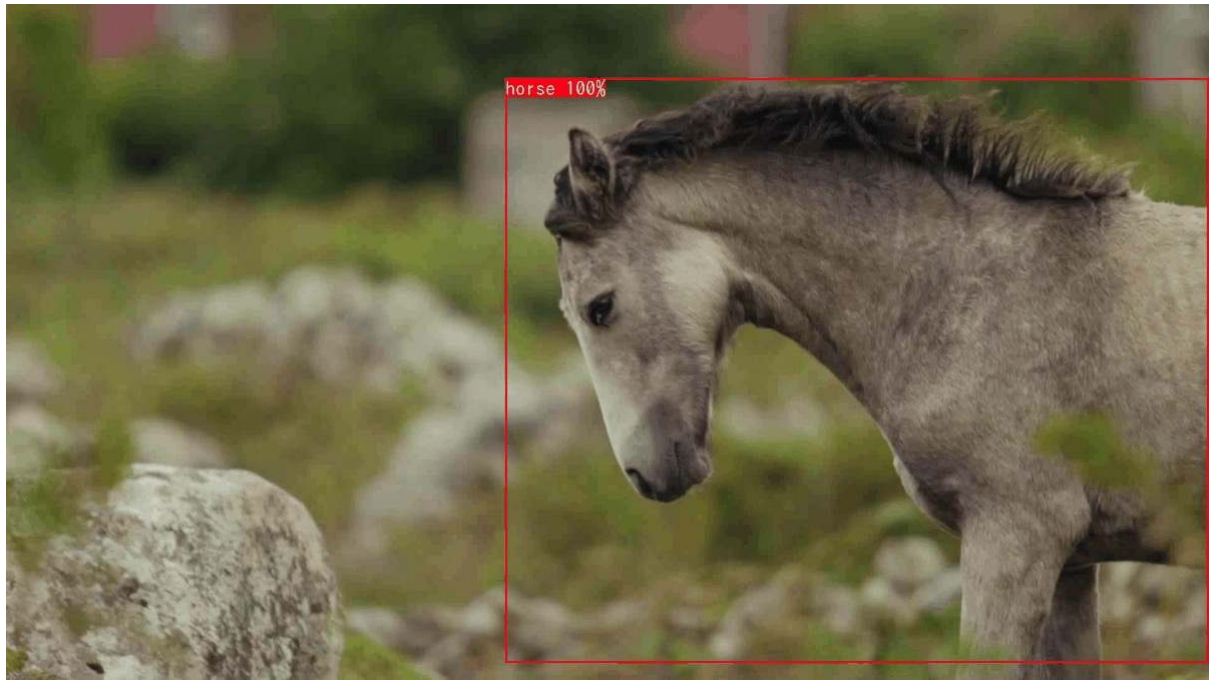
例１

例２

例３

例１の比較

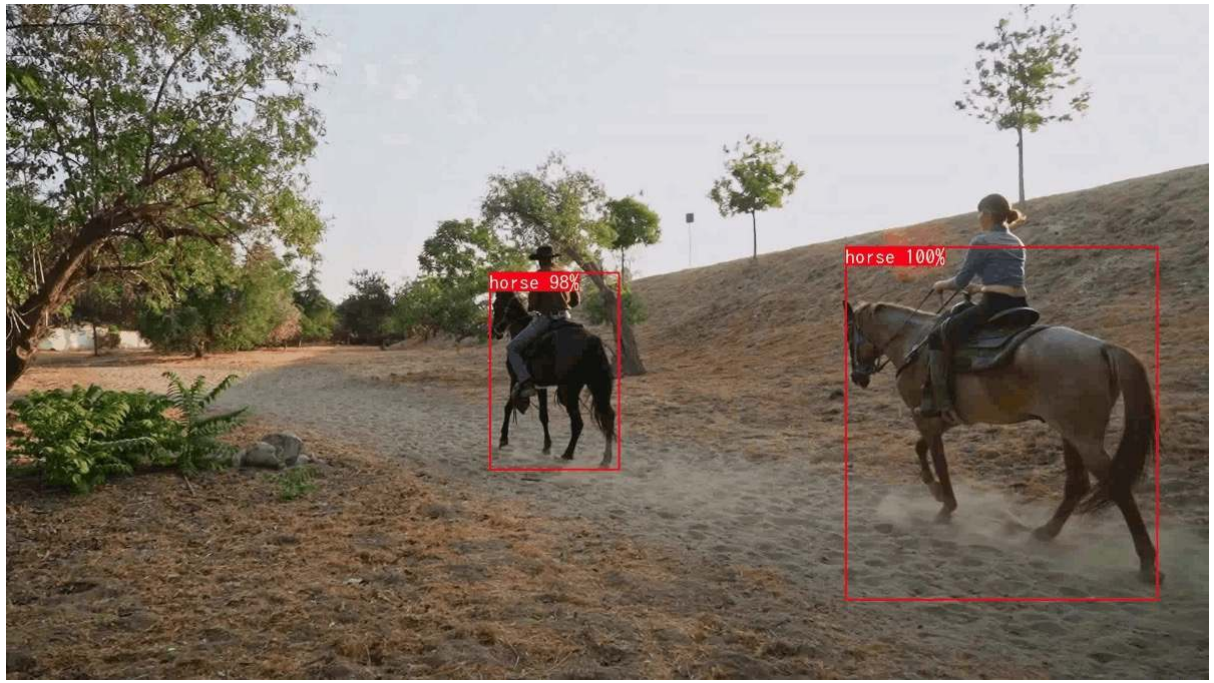
YOLO v3



提案手法

例２の比較

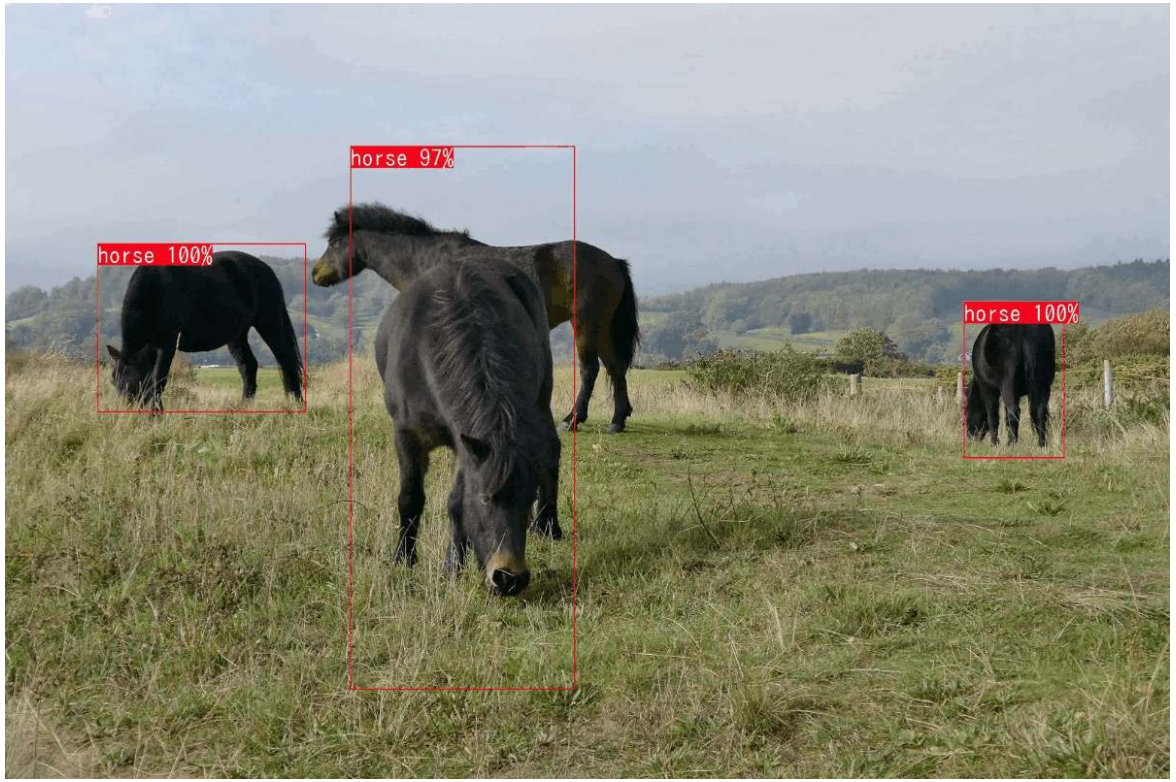
YOLO v3



提案手法

例３

YOLO v3



提案手法

まとめ

- 複数フレームを用いた物体検出手法を提案
- 提案手法と従来手法（YOLO v3）を比較

提案手法は精度、時間方向での安定性の点で優れていた

今後の課題

- 学習データを増やす
- 入力するフレームの数を増やす