物体クラスを考慮したFM-NMSによる物体検出の知識蒸留

○劉　履壮†，　平川　翼†，　山下　隆義†，　藤吉　弘亘†
†: 中部大学
liulyuzhuang@mprg.cs.chubu.ac.jp

概要：物体検出モデルの精度と速度の両立が重要である.組み込み端末に実装する場合，軽量化したモデルを用いることで，検出速度は高速化できるが，精度は低下するという問題点がある．そこで本研究では，教師モデルにYOLOv4，生徒モデルにYOLOv4-tinyを用いて，One-stageの物体検出に対して知識蒸留を適用する．また，物体検出で行う一般的なNMS処理の代わりに特徴マップにNMS処理を行うFM-NMSを導入して知識蒸留する．このとき，物体クラスごとに着目サイズを変えることで誤検出を抑制，精度向上を図る．評価実験の結果，提案手法で知識蒸留したYOLOv4-tinyモデルの検出精度が向上することを確認した．
＜キーワード＞ 物体検出，知識蒸留，NMS処理，One-stage

1. はじめに

物体検出モデルの精度と速度の両立が重要である. Grishickらは深層学習を物体検出分野に応用し，Two-stage物体検出モデルであるR-CNN [1] を提案した. そして, RoIプーリング層を導入して高速化したFast R-CNN [2] を提案した．一方，Renらはこれまで物体候補領域の選択に用いられたSelective Searchの代わりにCNNで構成されたRegion Priposal Network(RPN)を用いる高速なFaster R-CNN [3] を提案した．Two-stage物体検出モデルは精度は高いものの，検出速度が遅いという欠点がある．この欠点を解決するために，Redmonらは One-stage の物体検出モデルであるYou Only Look Once(YOLO) [4] を提案した．さらに，AlexeyらはYOLOの特徴抽出層に，計算量を大幅削減できるCross Stage Partial Network(CSPNet)構造と，解像度が異なる空間的な特徴量を集約する Path Aggregation Network(PANet)構造を用いたYOLOv4 [5]を提案した．これらの物体検出モデルのネットワーク構造は複雑かつ大規模であるため，組込み端末に実装することは難しい．そこで，大規模なモデルに近い精度にしつつモデル圧縮が可能な，知識蒸留が注目されている．

2015年にHintonらが知識蒸留 [6] を提案して以降，画像分類分野において知識蒸留の研究が活発に行われている．しかし，物体検出分野における知識蒸留に取り組んでいる研究は少ない．Chen [7] はFaster R-CNNを検出ネットワークとし，特徴抽出層，分類損失，回帰損失に対して同時に知識蒸留する手法を提案した．しかし，Two-stage物体検出モデルは One-stage 物体検出モデルより処理時間がかかるため，組込み端末に実装するのが難しい．Mehta[8]らは One-stage 物体検出モデルYOLOv2 を検出ネットワークとして，物体検出で行うNon Maximum Surpression(NMS)処理を特徴マップに適用するFM-NMSを提案した．Mehtaらは FM-NMS処理の出力をSoft Targetとして知識蒸留を行い, YOLOv2の精度を向上した. しかし, Mehtaらが提案したFM-NMSでは，クラスごとに適した着目サイズを考慮してないという問題点がある．

本研究では，One-stageの物体検出モデルに対して知識蒸留を適用する．この時，YOLOv4 を教師モデルとし，YOLOv4 を軽量化した YOLOv4-tiny[9] モデルを生徒モデルとして知識蒸留する．これより，処理時間の高速化を図る．また，Mehtaらと同様に特徴マップに対して NMS を行う FM-NMS[8]を用いる．このとき，物体クラスごとにFM-NMSで着目するサイズを変えることで高精度化を図る．

2. 関連研究

本章では，知識蒸留と物体検出分野での知識蒸留の従来手法について述べる．

2.1. 知識蒸留

精度が高いネットワークを学習するためには，畳込み層のカーネル数や全結合層のユニット数，さらにはとネットワークの層数を増やす必要がある．これにより，ネットワークのパラメータ数が増加するため，高精度かつ高速なネットワークを学習することは困難である．

Hinton らは大きなネットワーク(教師モデル)を学習した後，教師モデルの出力情報を Soft Target として，小さなネットワーク (生徒モデル) を学習することで，高精度かつネットワーク構造が小さなモデルを学習する知識蒸留を提案した．例えば，図 1 のように 5 クラス分類のネットワークを学習するとき．正解ラベル(Hard Target)は dog が 1 であり，それ以外のクラスが 0 である．教師モデルの出力は dog : 0.82，cat : 0.08，cow : 0.07，car : 0.02，person : 0.01 になり，この中の不正解クラスのうち，cat と person の出力はそれぞれ 0.08 と 0.01 である．これは dog が cat に似ているが，person に似ていないことを示している．このような不正解クラス間の情報は Hard Target から学習できない．Hinton らはこのような教師モデルの出力から得られる情報を活用した知識蒸留を実現している．

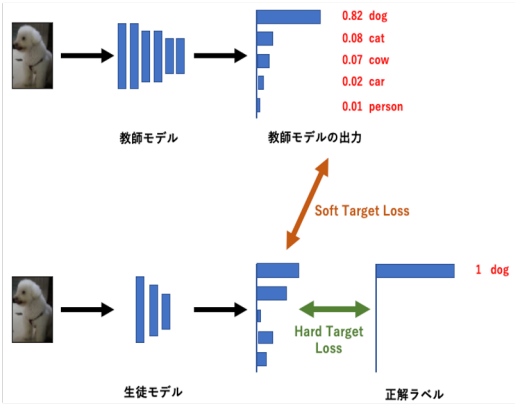


図 1 知識蒸留の例

2.2. Two-stage 物体検出モデルにおける知識蒸留

Chen らは Two-stage 物体検出モデル Faster R-CNN を用いた知識蒸留手法を提案した [7] ．教師モデルと生徒モデルは同構造の検出ネットワークを利用しているが，生徒モデルの特徴抽出ネットワーク (Backbone) は軽量化したネットワークとしている．図 2 のように，画像を教師モデルと生徒モデルに入力し，Backbone (hint) の出力，クラス分類(classification)の出力と回帰 (regression) の出力に対してそれぞれの損失を計算して Soft Target Loss (L_{soft}) として求める．正解ラベルとの損失を Hard Target Loss (L_{hard}) とし，Hard Target Loss と Soft Target Loss から式 (1) に示す検出損失 (l_{loss})を算出する．

$$L_{loss} = L_{hard}(s,T) + \lambda L_{soft}(s,t) \qquad (1)$$

ここで，s は生徒モデル出力，T は正解ラベル，t は教師モデル出力，λは重みである．誤差逆伝播法で生徒モデルの重みパラメータを更新して，損失値を最小になるように学習し，生徒モデルの出力を教師モデルの出力分布に近づける．

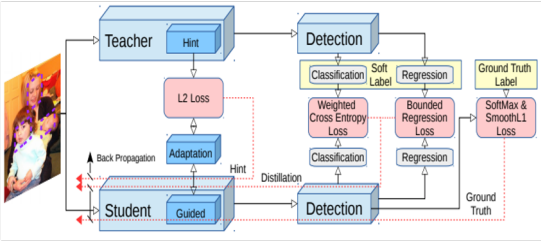


図 2 Two-stage 物体検出モデルの知識蒸留処理の流れ[9]

2.3. FM-NMS

Mehta [8] らは One-stage 物体検出モデルを知識蒸留する時に，教師モデルの出力に大量な重複した物体検出情報があるという問題を解決するため，教師モデルが出力する特徴マップに対して行う Non Maximum Suppression (NMS) 処理を改善した手法である FM-NMS を提案した. FM-NMS は教師モデルから出力される特徴マップに対して 3×3 の隣接するグリッドセル内の最大点を探索する．同じクラスでスコアが最も高い点以外の位置のスコアを 0 にし，重複している検出結果を除去することができる．しかし，この手法は全てのクラスに対して同一のサイズで比較するため，誤って大きな物体の検出結果も除去されてしまうという問題点がある．

3. 提案手法

本研究では，Mehta らが提案した知識蒸留手法に着目する．この手法は，全てのクラスに対して同じサイズで NMS を行なっている．そこで，クラスごとにサイズを変える新たな FM-NMS を提案する．また，YOLOv4 モデルを教師モデルとし，教師モデルから蒸留した知識を用いて生徒モデルの YOLOv4-tiny モデルを学習し，検出精度を向上させる．

3.1. クラスごとの FM-NMS 処理

クラスごとの FM-NMS 処理は大きなクラスと小さなクラスに対して異なる大きさで着目して NMS 処理を行う．処理流れを図4に示す．手順は下記の 3 つになる．

手順 1　各クラスの着目領域の大きさを決定する．

手順 2　各クラスの着目領域の大きさで FM-NMS

処理をする.

手順 3　手順 2 で生成した FM-NMS した特徴マップを Soft Target として生徒モデルと損失を計算する.

手順 1 では，実際に使用するデータセットに応じて各クラスの着目サイズを決定する．物体の大きさが大きなクラスの着目サイズは大きく設定し，物体の大きさが小さなクラスの着目領域は小さく設定する.

手順 2 では，手順 1 で決めた着目サイズの中で，この着目サイズに対応するクラスのスコアが最も高い点に対応するグリッドセルを抽出する.

手順 3 では，手順 2 で抽出した特徴マップを Soft Target として生徒モデルの信頼度 (Confidence) 出力，分類 (Classification) 出力と回帰 (Regression) 出力について，それぞれの損失を計算する.

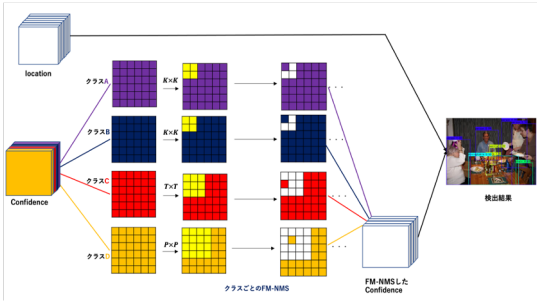


図 4 クラスごとの FM-NMS 処理図

3.2.　蒸留損失

図 5 のように YOLOv4 が出力特徴マップ出力のうち最も小さなサイズは19×19である．この特徴マップサイズにおいて，各グリッドセルの中に5つのバウンディングボックス情報が含まれるとすると，$19 \times 19 \times 5 = 1805$個の検出結果となる．この中には目標物体だけではなく，背景情報が含まれているという問題点があるため，背景領域を考慮した損失関数にする必要がある.

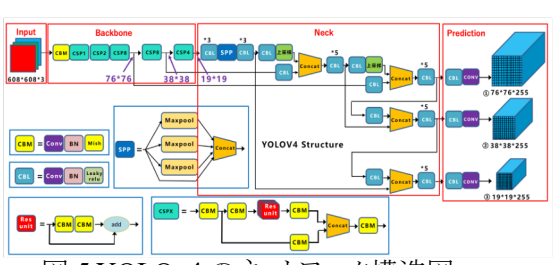


図 5 YOLOv4 のネットワーク構造図

そこで，本研究では教師モデルの信頼度 (Confidence) 出力を利用して，蒸留の損失関数を設計する．背景の信頼度出力は目標物体の信頼度出力より小さいため，教師モデルと生徒モデルの分類 (Classification) 出力と回帰 (Regression) 出力のそれぞれの損失計算式の係数とすると，蒸留した知識から背景の影響を抑制できる．蒸留中の損失関数L_{loss}を式 (2), (3), (4) に示す.

$$L_{loss} = L_{hard}(s,T) + \alpha L_{soft}(s,t) \tag{2}$$

$$L_{hard} = L_{conf}(s,T) + L_{cls}(s,T) + L_{reg}(s,T) \tag{3}$$

$$\begin{aligned} L_{soft} = {} & L_{con_{kd}}(s,t) \\ & + conf_t L_{cls_{kd}}(s,t) \\ & + conf_t L_{reg\_kd}(s,t) \end{aligned} \tag{4}$$

式 (2) において，L_{hard}は生徒モデル出力と正解ラベルとの損失であり，L_{soft}は生徒モデル出力と教師モデル出力の損失である．L_{hard}と L_{soft}を混合する割合をパラメータαによって調整する．本研究ではαを 1 とする.

式 (3) において，L_{conf}は生徒モデルと正解ラベルの信頼度損失，L_{cls}は生徒モデルと正解ラベルの分類損失，L_{reg}は生徒モデルと正解ラベルのバウンディングボックス回帰の損失である.

式 (4) の中，L_{conf_kd}は生徒モデルと教師モデルの信頼度損失，L_{cls_kd}は生徒モデルと教師モデルの分類損失，L_{reg_kd}は生徒モデルと教師モデルのバウンディングボックス回帰損失である．また，$conf_t$は教師モデルの信頼度出力である．ここで，L_{conf_kd}， L_{cls_kd}とL_{reg_kd}は全て平均二乗誤差(MSELoss)で算出する．平均二乗誤差の計算式を式 (5) に示す.

$$MSELoss(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{5}$$

3.3.　学習の流れ

提案手法を用いて学習する流れを図 6 に示す．詳細手順は下記となる.

Step1. 入力画像を学習済みの教師モデルと生徒モデルに入力する.

Step2. 教師モデル出力に対してクラスごと FM-NMS 処理を行い，出力値を Soft Target として使用する.

Step3. 生徒モデルの出力と Hard Target の誤差 L_{hard} を算出する.

Step4. 式 (4) より，生徒モデルの出力と Soft Target

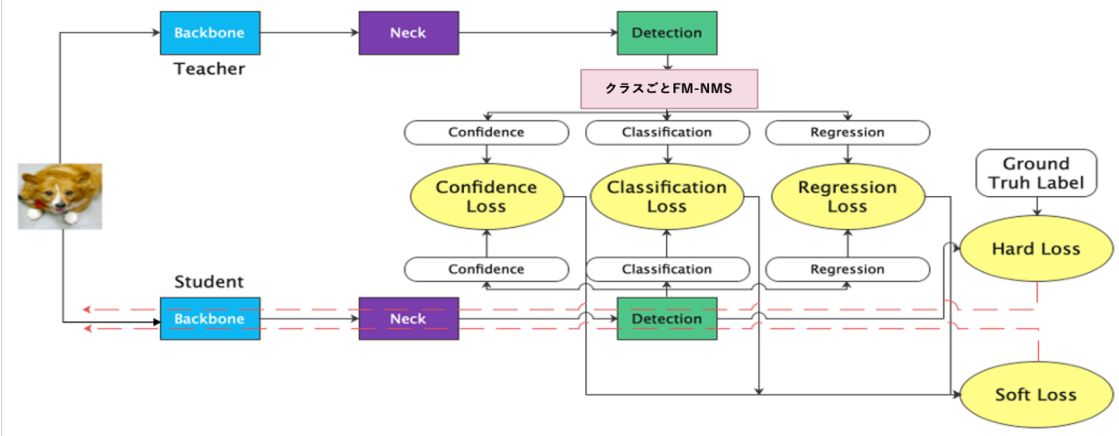


図 6 提案手法の学習流れ

から L_{soft} を算出する.

Step5. 式 (2) より, L_{hard} と L_{soft} から L_{loss}を算出する.

Step6. 誤差逆伝播法によって生徒モデルの重みパラメータを更新する.

4. 評価実験

提案手法の有効性を確認するため，YOLOv4 モデルを教師モデル，YOLOv4-tiny モデルを生徒モデルとして学習する．VOC データセット[10]を用いて，蒸留せずに学習した YOLOv4-tiny モデルとの比較実験をする．評価指標には mean Average Precision(mAP)を用いる．また，1 秒間処理可能なフレーム数(FPS)を処理速度として比較する．本実験では, CPU に Inter Xeon Gold 5122, GPU に Quadro RTX 8000 を用いる.

4.1. 実験概要

VOC2012 および VOC2007 の学習用と検証用データセットを合わせて本実験の学習用データセットとして使用する．また，VOC2007 のテスト用データセットを本実験のテスト用データセットとして用いる. データセットには 20 クラスの物体が含まれており，学習用およびテスト用データセットの画像枚数はそれぞれ 16,551 枚と 4,952 枚である.

評価指標である mAP の計算式は式(6)に示す. ここで, I_{AP}は精度であり, Cはクラスの種類であり, Nはクラスの数である.

$$I_{mAP} = \frac{1}{N}\left(\sum I_{AP,C}\right) \tag{6}$$

学習時，まず学習済みの YOLOv4 モデルを初期化とし，3.3 節で述べた流れに基づいて学習する. クラスごとの FM-NMS を用いる着目サイズ，各クラスの平均面積を求め，その大きさから決める. 面積が小さなクラスは2×2のサイズで FM-NMS 処理を行う. 対象クラスは bottle, pottedplant, chair, tvmonitor，sheep，boat である．面積が大きなクラスは4×4の比較サイズで FM-NMS 処理を行う. 対象クラスは train，aeroplane，bus，motorbike，diningtable，horse である．その他のクラスは3×3の着目サイズで FM-NMS 処理を行う.

ミニバッチサイズは 8，Instersection over Union(IoU)は 0.5 とする．最適化手法には SGD を用い，学習率範囲は$[10^{-6}, 10^{-4}]$とする.

4.2. 定量的評価結果

実験結果を表 2 に示す．提案手法で学習した YOLOv4-tiny モデルは蒸留なしの YOLOv4-tiny モデルより mAP が 12.5 ポイント向上した． 提案手法で学習したモデルは Mehta らの FM-NMS で学習したモデルと比較すると mAP が 1 ポイント向上した．精度向上したクラスは， potteplant，chair，boat，car，person，bird，horse，bus，diningtable，motorbike，train，dog，sofa である．一方で，本実験では計 20 クラスに対して，3 つの着目サイズで FM-NMS 処理をしたため，着目サイズの設定はまた不十分だと考える．精度が向上していないクラスについては，適切な着目サイズを設定すると，精度が向上できると考える.

4.3. 定性的評価結果

図 7 に各手法の検出結果例を示す．図 7 の 1 行目から 3 行目を見ると，従来手法（YOLOv4-tiny）では検出できてないクラスに対して提案手法では検出できている．また，図 7 の 4 行目と 5 行目を見ると，従来手法では重複して検出しているが，提案手法では正しく検出できている.

表 1 Pascal VOC における各手法の mAP 比較

methods	mAP	AP																			
		Aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
YOLOv4	86.3	94	92	86	79	80	92	96	91	74	91	77	90	92	92	90	62	87	82	92	87
YOLOv4-tiny	46.4	57	62	45	40	59	42	69	21	46	65	17	30	48	52	66	34	64	24	27	59
Mehta	57.9	**69**	**75**	53	45	**59**	57	81	**50**	46	**68**	32	**51**	66	68	75	**40**	**65**	40	51	**67**
Ours	58.9	66	72	**54**	**47**	**59**	**63**	**82**	47	**50**	66	**36**	**51**	**70**	**70**	**76**	**40**	**65**	**43**	**56**	64

これより，提案手法は教師モデルの精度に近づいていることが確認できる．

4.4. 検出速度の比較

提案手法で学習したモデルと YOLOv4-tiny モデルを用いて，30 枚のテスト画像を検出する際の処理速度を比較する．ただし，画像の前処理と可視化処理は計測時間から除く．比較結果を表 3 に示す．

表 2 処理速度の比較

methods	処理速度(fps)
YOLOv4-tiny	23.4
Ours	49

提案手法で学習したモデルと YOLOv4-tiny モデルのネットワーク構造は同じであり，モデルの推論時間自体は変わらないものの, 通常の NMS と比べて本手法ので導入したクラスごとの FM-NMS 処理が高速であることから，表 3 より，従来手法と比べ処理速度が改善されていることが確認できる．

5. おわりに

本研究では，知識蒸留を用いて One-Stage 物体検出モデルの学習方法を提案した．提案手法を用いて YOLOv4-tiny モデルの検出精度を向上することを確認できた．また，提案手法により YOLOv4-tiny モデルの検出速度を向上した．今後は，クラスごとの FM-NMS 処理をさらに最適化し，そして，サイズごとに適用することで精度改善を図る．また，教師モデルの中間層の特徴を知識蒸留する方法についても検討する．

参考文献

[1] R. Girshick, *et al*., “Richfeature hierarchies for accurate object detection and semantic segmentation”, IEEE, 580-587, 2014.

[2] R. Girshick, “Fast R-CNN”, IEEE, 1440-1448, 2015.

[3] S. Ren, *et al*., “Faster-rcnn: towards real-time object detection with region proposal networks”, NeurIPS, 91-99, 2015.

[4] J. Redmon, *et al*., “You Only Look Once: Unified, Real-Time Object Detection”, IEEE, 779-788, 2016.

[5] A. Bochkovskiy, *et al*., “YOLOv4: Optimal Speed and Accuracy of Object Detection”, arXiv, 2004.10934, 2020.

[6] G. Hinton, *et al*., “Distilling the knowledge in a neural network”, stat, Vol. 1050, P.9, 2015.

[7] G. Chen, *et al*., “Learning Efficient Object Detection Models with Knowledge Distillation”, IEEE, 2017.

[8] R.Mehta, *et al*., “Object detection at 200 Frames Per Second”, ECCV, 2018.

[9] A. Bochkovskiy. Darknet:Open Source Neural Networks in C. 2020. Available online: http://github.com/AlexeyAB/darknet(accessed on 2 November 2020).

[10] M.Everingham, *et al*., “The PASCAL visual object classes (VOC) challenge”, A Retrospective[J]. International Journal of Computer Vision, 2015, 111(1): 98-136.

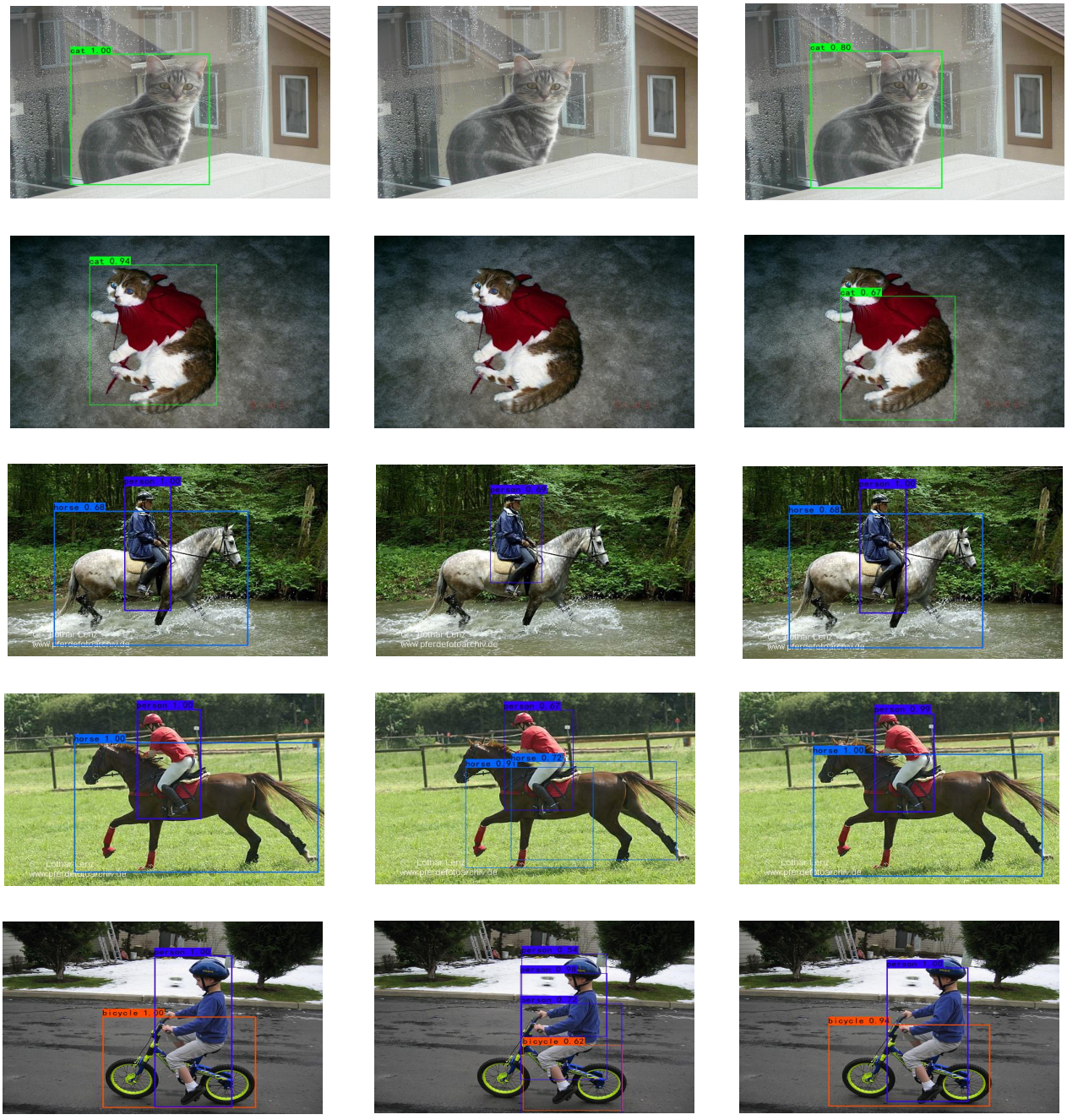
劉履壮 : 2021 年中部大学情報工学科卒業，現在中部大学大学院修士課程在学中，自動運転の実現に向けた物体検出モデルの研究に従事.

平川翼 : 2013 年広島大学院博士課程前期修了，2014 年広島大学大学院博士課程後期入学，2017 年中部大学研究員(~2019), 2017 年広島大学大学院博士後期課程修了. 2019 年中部大学特任助教. コンピュータビジョン，パターン認識，医用画像処理の研究に従事.

山下隆義 : 2002 年奈良先端科学技術大学大学院博士前期課程修了. 2002 年オムロン株式会社入社，2009 年中部大学大学院博士後期課程修了(社会人ドクター)，2014 年中部大学講師，2021 年より同大学教授. 人の理解に向けた動画像処理，パターン認識・機械学習の研究に従事.

藤吉弘亘 : 1997 年中部大学大学院博士後期課程修了. 1997~2000 年米カーネギーメロン大学ロボット工学研究所

– Postdoctoral Fellow．2000 年中部大学講師．2004 年より同大学教授．2005~2006 年米カーネギーメロン大学ロボット工学研究所客員研究員，計算機視覚，動画像処理，パターン認識・理解の研究に従事．



(a)　YOLOv4　　(b)　YOLOv4-tiny　　(c)　提案手法で学習したモデル

図 7　各手法の検出結果例