色と距離情報を入力とした 3 次元拡張型 YOLO

○高橋 正裕（中央大学），Moro Alessandro（RITECS Inc.），
池 勇勳（中央大学），梅田 和昇（中央大学）

Extended YOLO Using Color and Depth Input

○Masahiro TAKAHASHI (Chuo Univ.), Moro ALESSANDRO (RITECS Inc.)
Yonghoon JI (Chuo Univ.), and Kazunori UMEDA (Chuo Univ.)

Abstract: Among object detectors using deep learning, those that deal with distance information have been actively studied in recent years. However, the conventional detector has a large network structure, and real-time performance has been impaired. Therefore, in this research, a stereo camera is assumed, and a 3D object detector is constructed by using a depth and a color image as input. The network architecture is based on YOLOv3 and is extended to three dimensions in part of the input layer and the intermediate layer. We verified the effectiveness of this detector using a dataset. As a result, the proposed model was able to output 3D information and the processing speed was 44.4fps.

１．緒言

物体を検出もしくは認識することや，物体の数を数えることは，近年防犯やマーケティングの分野で需要がある．しかし，人を雇ってこれらの作業を行った場合，不注意からくるヒューマンエラーや人件費といった問題が起こりうる．そこで，これらの問題を解決するために，自動物体検出を実現するカメラシステムが必要である．

近年の深層学習の目覚ましい発展により，物体検出器は飛躍的に進歩した．その中でも，特に 3 つの型のネットワーク構造が一定の成果を上げている．1 つ目は，Fast/Faster/Mask R-CNN [1, 2, 3]を代表とする R-CNN (Region-based CNN)型 [4]である．特に Mask R-CNN は FCN (Fully Convolutional Network) [5]を組み込むことで，ピクセル単位の物体検出を実現している．しかし，このネットワークは物体ごとに畳み込みネットワークが存在するため，ネットワーク構造が大きくなることで計算量が多いという欠点があった. 2 つ目は, SSD 型 [6]である．SSD は R-CNN に比べて高速であり，それぞれのスケールごとの特徴抽出結果を反映するため，多オブジェクトがシーン内に存在しても検出可能であるという利点を持つ．しかし，SSD にはスケールの選択や基本矩形のサイズ設定等，恣意的なパラメータが多い．そして最後は，YOLO (You Only Look Once)型 [7]である．YOLO は 1 つのシンプルなネットワークで構成されており，YOLOv2 [8]は SSD よりも高い精度と処理速度を達成した．しかし，アルゴリズムの制約上，シーン内に多オブジェクトが存在する場合の検出が困難であるという性質を持つ．

一方，人数カウントや自動運転などでは，物体同士のオクルージョンに対応するため，物体の距離情報を考慮した物体検出を要するが，上記のネットワークは距離の入力に対応していない．そこで最近では，3 次元点群を入力とした物体検出器が多く研究されている．しかし，これらはネットワーク構造が巨大であり，リアルタイム性を重視していない．リアルタイム性を重視した YOLO3D [9]などの検出器も存在するが，これらは距離の俯瞰画像を入力する必要があり，ステレオカメラなどの安価なセンサで検出を行うことを想定していない．

本研究では，距離情報とカラー情報を入力とし，物体の 3 次元位置を出力する物体検出器の構築を目的とする．この時，ネットワークの軽量化により，よりリアルタイム性のあるシステムの構築を目指す．これにより，従来では LiDAR (Laser imaging Detection And Ranging)のような高額な距離センサにより行われていた 3 次元位置の特定が，ステレオカメラにより簡易的かつ高速に行う事が可能である．

２．3 次元拡張型 YOLO

2.1 ネットワーク構造

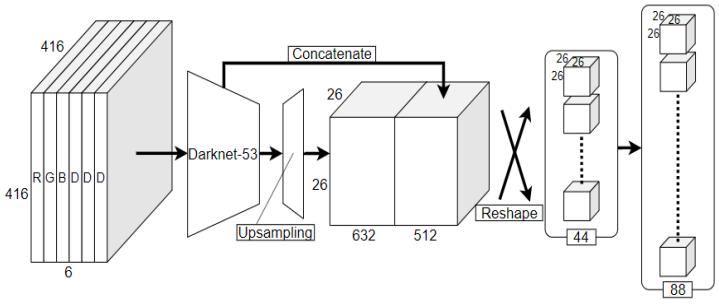

Fig. 1　ネットワークの概要図

本研究で提案するネットワークの概要図を Fig. 1 に示す．YOLOv3 [10]では，RGB の 3 チャンネルから構成されるカラー画像を入力としていた．それに対し提案モデルでは，距離画像に対応させるため，新たに距離画像のチャンネルを設け，1 つの画像として入力する．ここで，加えるチャンネル数が 1 つでは，畳み込みにおける各フィルタに対する重みの均等化を行うことが難しくなる．そこで，全く同じ距離画像 3 枚をチャンネル方向に並べた 3 チャンネルの画像を用意し，カラー画像のチャンネル方向後方にこれを加える．提案モデルでは，こうして出来上がった 6 チャンネルの画像を入力とする．

中盤までの特徴抽出を行う部分のネットワーク構造としては，YOLOv3 で用いられている Darknet-53 を採用する．理由は次のとおりである．YOLOv3 において，バウンディングボックスを予測するための特徴抽出に成功し，かつそれを ResNet (Residual Network) [11]により行った場合と比較して，精度と処理速度において高いスコアを達成している．このことから，2 次元の画像から特徴抽出を行う場合において，Darknet-53 は効率的なネットワーク構造であると言える．提案モデルでは入力に深度情報が含まれるが，これを距離画像として入力するので，カラー画像の場合と同様に特徴抽出を行うことが可能である．

続いて，ネットワーク構造の中で，Darknet-53 を用いて得られたカラー画像と距離画像の特徴から，3 次元のバウンディングボックスを出力する部分について説明する．この部分では，アップサンプリングを用いて異なるスケールで得られた 2 つの出力を連結し，畳み込みを行った後，チャンネル方向に一定範囲で分割を行うことで，テンソルの 3 次元化を行う．我々はこの部分を YOLO に共通する概念である Unified Detection を元に作成した．Unified Detection とは，バウンディングボックスの座標やクラス分類結果等をチャンネル毎に格納することで，クラス分類と領域特定を同時に出力することが可能となる概念である．この概念を元に，チャンネル毎に出力値の種類をコントロールすることが可能であるため，チャンネル範囲毎でも同じことが可能である．ここで，YOLO v 3 には FPN (Feature Pyramid Network) [12]構造が採用されているが，提案モデルでは計算コスト削減のために，中間部分のスケールのみを扱う．

以上のようなネットワーク構造により，提案モデルはカラー画像と距離画像から 3 次元のバウンディングボックスを出力する．Fig. 2 に示すように，全ての畳み込み層は 3×3 もしくは 1×1 のカーネルフィルタを用いている．また，途中で 2 次元から 3 次元への形状変更を行うため，入力画像の各チャンネルにおけるサイズは 416x416, 出力のサイズは 26x26x26 となっている．よって，提案モデルではシーンを Fig. 3 のように捉え，検出を行うことができる．

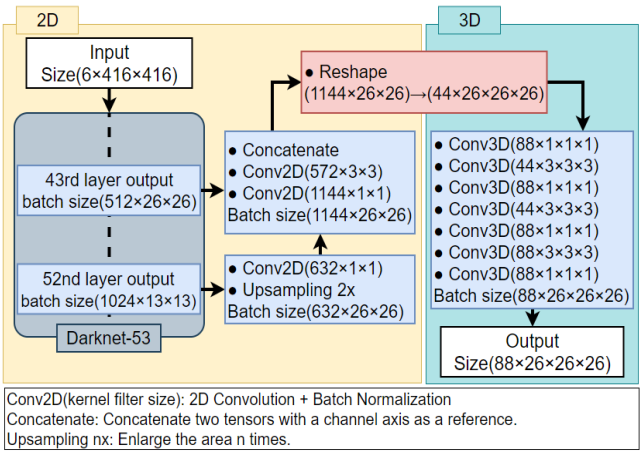

Fig. 2　　提案モデルの各パラメータ

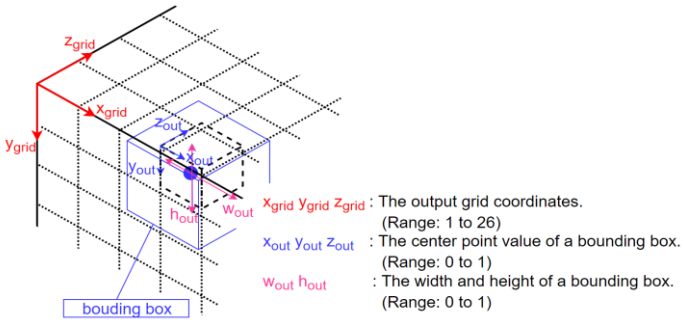

Fig. 3　　提案モデルの出力形式

2. 2 学習と推論

提案モデルでは，Darknet-53 におけるハイパーパラメータは YOLOv3 で用いられていた値に設定した．また，それ以外の値に関しては Fig. 2 に示されているように設定した．ここで出力が 88 チャンネルである理由としては，バウンディングボックスの中心座標(x, y, z)を出力するのに 3 チャンネル，バウンディングボックスの横幅，縦幅，奥行きを出力するのに 3 チャンネル，物体が存在するか否かを判定するのに 2 チャンネルであり，残りの 80 チャンネルは YOLOv3 と同じクラス数を設けたためである．

提案モデルでは，YOLOv3 に加えて，奥行き方向についても出力を行うことになる．そのため，学習のために深度情報に関する 2 乗誤差を含めた YOLO3D の誤差関数を学習に利用する．この誤差関数は，物体の中心座標の 2 乗誤差，バウンディングボックスの横幅・縦幅・奥行きの 2 乗誤差，物体が存在するか否かのクロスエントロピー誤差の合計で表される．よって，誤差関数は式(1)のようになる．

$$
\begin{aligned}
L_{ExpandableYOLO} = {} & \lambda_{coord} \sum_{i=1}^{G} l_i^{obj} \left[\left(t_x^{(i)} - \hat{t}_x^{(i)} \right)^2 + \left(t_y^{(i)} - \hat{t}_y^{(i)} \right)^2 + \left(t_z^{(i)} - \hat{t}_z^{(i)} \right)^2 \right] \\
& + \lambda_{coord} \sum_{i=1}^{G} l_i^{obj} \left[\left(t_w^{(i)} - \hat{t}_w^{(i)} \right)^2 + \left(t_h^{(i)} - \hat{t}_h^{(i)} \right)^2 + \left(t_d^{(i)} - \hat{t}_d^{(i)} \right)^2 \right] \\
& + \lambda_{coord} \sum_{i=1}^{G} l_i^{obj} \left(c_{obj}^{(i)} - \widehat{c_{obj}}^{(i)} \right)^2 + \lambda_{noobj} \sum_{i=1}^{G} l_i^{obj} \left(c_{obj}^{(i)} - \widehat{c_{obj}}^{(i)} \right)^2 \\
& + \sum_{i=1}^{G} \sum_{k=1}^{K} l_i^{obj} \left(p_k^{(i)} - \widehat{p_k}^{(i)} \right)^2
\end{aligned} \tag{1}
$$

ここで，今回任意に設定される誤差の使用度を示すλは，バウンディングボックスの中心座標とサイズの誤差に対する変数を 1 に，物体が存在しない場合の誤差に対する変数を 10 に設定した．識別子 1 は教師データに含まれるバウンディングボックスがそのセルに存在するかどうかを示すものであり，存在する場合はバウンディングボックスに対する誤差が計算され，存在しない場合は物体の信頼値部分の計算が行われる．

得られた候補から最適なバウンディングボックスを選択する代表的な手法としては，NMS (Non Maximum Suppression) がある．これは IoU (Intersection over Union) と呼ばれる領域の重なり度合いを表すスコアをもとに，同じ物体に対して推定されたバウンディングボックスを消去する方法である．ここで，3DYOLO 等の 3 次元情報を扱う物体検出器は，センサ正面方向と垂直方向の 2 方向から 2 次元に対する IoU を計算し，NMS によるバウンディングボックスの選択を行っていた．しかし，この方法では同じ 3 次元のバウンディングボックスに対して 2 回 IoU を計算していることとなり，GPU による並列計算を行う上で非効率的である．そこで，提案モデルでは，Fig. 4 のような体積の重なり度合いを表す 3DIoU を定義し，これをもとに NMS を実行する．これにより，IoU 計算は GPU 上で 1 回しか行わず，処理速度の改善が期待できる．NMS における IoU のしきい値は，YOLO 等においては 0.5 が用いられていたが，今回のように体積比の場合には，体積と面積の関係から，$0.5^{3/2}=0.35$ を用いる．

3．人物検出実験

提案モデルの有効性を確認するため，複数のデータセットを用いてモデルの精度を検証した．また，リアルタイム性についても検証するため，実装したシステムについて処理時間を計測した．

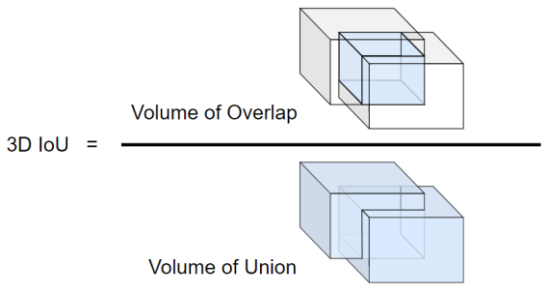

Fig. 4　3D IoU

我々が作ったデータセットを用いてモデルの学習を行い，学習に要する時間の計測と IoU スコアによるモデル精度の評価を行った．中央大学後楽園キャンパスの 2720 室にてデータを取得した．データの取得は Fig. 5 のように，屋内で人が歩いている様子を撮影した．ラベルの作成方法としては，まず Mask R-CNN による自動のラベル付けを行い，うまく検出できなかったシーンに対しては人力で行った．また，奥行き方向のラベルは 70±10 cm を奥行きの最大値 10m として正規化した値である 0.07±0.01 とした．こうして取得した 1,280 シーンのうち，1,140 枚を学習データ，100 枚をバリデーションデータとして学習を行った．学習係数は 0.001 に設定し，最適化手法には Adam (Adaptive moment estimation) を用いた．

学習の経過を示す誤差の遷移を Fig. 6 に，学習済みモデルによる検出例を Fig. 7 に示す．誤差の遷移を見てみると，誤差は一定の収束を見せており，距離画像を入力に加えても特徴抽出は可能であるということがわかる．しかし誤差は 40〜60 エポック周辺で頭打ちとなっており，以降は学習効果が薄いと考えられる．原因としては，学習データの枚数不足により，過学習が起こりかかっていると考えられる．

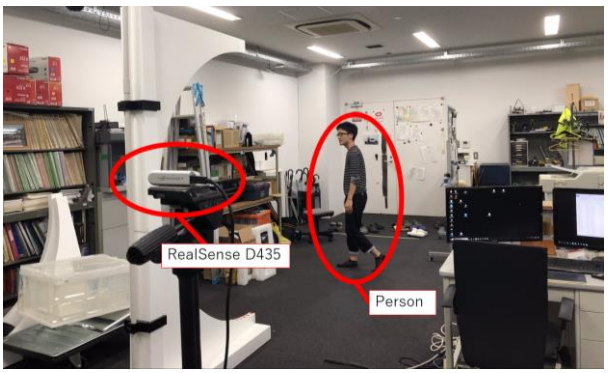

Fig. 5　データの取得環境

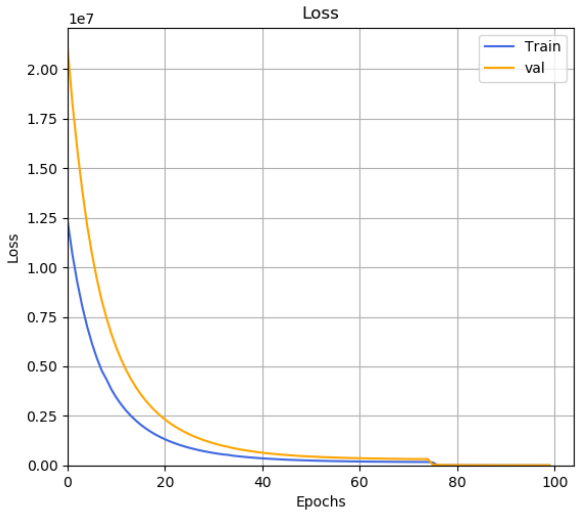

Fig. 6　誤差の遷移

今回は学習データとバリデーションデータのシーンにあまり大きな違いがないために誤差の発散は見られなかったが，カメラ自体が移動するようなシーンにおいては，バリデーションデータの誤差が発散する恐れがある．そのため，様々なシーンを含めたデータセットでの検証が必要であるほか，奥行きの特徴獲得のために特徴抽出部分の調整が必要であると考えられる．

続いて，学習済みモデルによる検出例と，YOLOv3による検出結果(Fig. 8)を比較する．これらのうち，赤色のバウンディングボックスは Ground Truth を示している．2 次元平面における検出は提案モデルでも YOLOv3 とほぼ同じように行うことができている．しかし，奥行きの推定に関しては少し安定しない．バウンディングボックスの中心座標に関しては正確に推定できているが，バウンディングボックスの奥行きを推定するのが難しい傾向にある．この原因としては，今回のデータセットはステレオカメラを用いて 1 方向から取得したものを用いたため，その物体の奥行き方向の長さの情報を距離画像から推定することが困難であるということがあげられる．この改善案としては，各物体の正確な奥行き方向の長さを正解として与えるとともに，距離画像に疑似カラーを与え，奥行きが距離ごとに分割されたものとして入力することで，奥行きを推定する際の基準をモデルに対して示すことがあげられる．

Table 1 は, 2 章で述べた IoU の平均と最大値を示す．平均の IoU を見てみると，一見 3DIoU は 2DIoU に比べて低いが，体積と面積の関係からみると，安定はしないものの，同等の精度を実現していることがわかる．しかし，スコア全体を見ると，最大値はかなり高いものの, 平均値は YOLO の論文におけるスコアである 0.8 には及ばず，十分な精度とは言えない．この原因としては, 2 次元から 3 次元へ拡張する作業を連結によって行ったことで，それまでに抽出されていた特徴が打ち消されてしまっていることが予測される．そのため，単純な連結ではなく，距離と色情報の特徴抽出を明確に分ける必要があると考えられる．

Fig. 7　　提案モデルによる検出結果

Fig. 8　　YOLOv3 による検出結果

処理速度の比較結果を Table 2 に示す．処理速度に関しては，提案モデルがシンプルなネットワークで構成されていることもあり，高速な検出を実現することができた．同じく距離情報を入力とした Complex-YOLO [14]と比較してみると, わずかに届かなかった．しかし，Complex-YOLO では GPU として Titan X を利用しており，性能で劣る GTX 1080Ti を用いてこの処理速度が出せたというのは良い結果であるといえる．処理速度が低下した原因としては, 3 次元のテンソルに変換した後, 3 次元の畳み込みを利用したことで, パラメータの総数が多くなってしまったことがあげられる．しかし，パラメータの最適化による層の削減等により改善が見込まれる．

3．結言

本研究において，私たちは，RGB-D から 3 次元のバウンディングボックスを出力可能で軽量なモデルを作成した．提案モデルでは，距離情報を画像として入力することで，通常の画像と同様の方法で距離情報の特徴抽出を行い，バウンディングボックスの奥行き方向の位置と長さを取得した．

今後の展望としては，パラメータの最適化や疑似カラーの導入等により，精度向上を図る．

Table 1　提案モデルによる IoU スコアの結果

	2D IoU	**3D IoU**	**3D IoU$^{2/3}$**
Mean	0.54	0.39	0.53
Max	0.92	0.85	0.90

Table 2　処理速度比較結果

Method	**GPU**	**FPS**
YOLOv3 [10, 13]	GTX 1080Ti	74.0
Complex-YOLO [14]	Titan X	50.4
提案モデル	GTX 1080Ti	44.4
YOLO3D [9]	Titan X	40
VoxelNet[15]	Titan X	4.3

参考文献

[1] R. Girshick, “Fast R-CNN,” in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 1440-1448, 2015.

[2] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Conference on Neural Information Processing Systems (NIPS)*, 2015.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *arXiv preprint arXiv:1703.06870*, 2017.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 580-587, 2014.

[5] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Neural Networks for Semantic Segmentation,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431-3440, 2015.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” in *arXiv preprint arXiv:1512.02325*, 2015.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016.

[8] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] W. Ali, S. Abdelkarim, M. Zahran, M. Zidan, and A. E. Sallab, “YOLO3D: End-to-end real-time 3D Oriented Object Bounding Box Detection from LiDAR Point Cloud,” *arXiv preprint arXiv:1808.02350*, 2018.

[10] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” in *arXiv preprint arXiv:1804.02767*, 2018.

[11] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *arXiv preprint arXiv:1409.1556*, 2014.

[12] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117-2125, 2017.

[13] PyTorch-YOLOv3, https://github.com/eriklindernoren/PyTorch-YOLOv3, 2019/09/30.

[14] M. Simon, S. Milz, K. Amende, and H. M. Gross, “Complex-YOLO: An Euler-Region-Proposal for Real-Time 3D Object Detection on Point Clouds,” in *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 197-209, 2018.

[15] Y. Zhou and O. Tuzel, “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.