

飲食店における天井のカメラを用いた空きジョッキ検出

七五三掛 豊和[†] 中島 克人[‡]

東京電機大学[‡]

1 はじめに

国内の飲食店業界では人手不足が深刻な問題となっており、少ない人手で効率良い店舗運営が望まれている。中でもアルコール飲料を提供する居酒屋などでは酒類等の追加注文をタイムリーに受ける事は、顧客満足度と回転率の向上に役立つと考えられる。そこで本研究では、店舗の天井カメラで顧客のテーブル上の空の食器類を検出して店員に通知する事により、ホール店員の無駄な巡回を削減し、追加注文や空き食器の片付けの迅速化を図る事を目的とするシステムの開発を目指す。今回はビールなどのジョッキに焦点を当て、一般物体検出器に基づくジョッキ検出とそのジョッキが空きか否かの判別を行う2段階式の空きジョッキ検出器を試作・評価した。

2 関連研究

近年、深層畳み込みニューラルネットワーク(DCNN)による画像認識の技術とその応用が急速に広まっている。一般物体検出器ではその認識精度と速度のバランスからYOLOv3[1]は良く利用される。画像内の物体のクラスを判別するための畳み込みニューラルネットワーク(CNN)では、深い層で画像特徴の記述力を高めたVGGNet[2]、深層ながら残差構造で学習を容易化したResNet[3]、幅方向に広くすることで記述力を高めるInceptionV3[4]などがある。

3 提案手法

3.1 2段階式空きジョッキ検出器

シーン画像内の物体検出にはシーン内の物体の位置検出とその物体の種類分類という2つのタスクが必要であるが、前者には公開された性能の良い学習済みのモデルを用いることができるため、我々は後者の分類(空きジョッキ判別)のみを学習し、2段階式の空きジョッキ検出器を構成する(図1)。なお、2段階方式は他の食器の空き検出を機能追加する際にも柔軟性に優れると考えられる。

3.2 ジョッキ検出器

前段で用いる一般物体検出器にはCOCO datasetで事前学習されたYOLOv3[1]を使用する。モデルには、空きジョッキを[cup], [bottle], または[wineglass]として最も良く検出したYOLOv3-SPPを選択した。

3.3 空きジョッキ判別器

前段で検出され切り取られた画像を、後段の空きジョッキ判別器に入力する。空きジョッキ判別器用

Empty Mug Detection using Ceiling Cameras in Restaurants

[†] Toyokazu Shimekake · Tokyo Denki University

[‡] Katsuto Nakajima · Tokyo Denki University

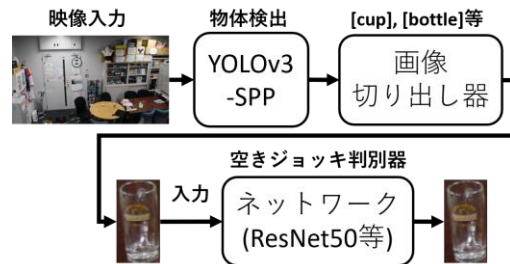


図1 空きジョッキ検出器の全体図

にはVGGNet, ResNet, 及びInceptionV3の3種類の畳み込みネットワーク(以後, CNN)を学習し, その判別精度を比較する。

3.4 ジョッキデータセット

ジョッキが1個以上映るシーン画像をYOLOv3-SPPに入力し, 出力された検出矩形に沿って切り取り, それを「空きジョッキ」と「中身在りジョッキ」の2つにラベル付けする。今回, 次節の条件下で撮影した288枚のシーン画像に対して, 空きジョッキ672枚, 中身在りジョッキ635枚の計1,307枚のデータセットを作成した。図2にデータ作成処理の流れを, 図3に切り出したジョッキ画像例を示す。

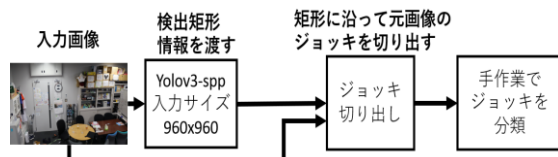


図2 元画像からジョッキ画像生成の流れ



(a) 空きジョッキの例 (b) 中身在りジョッキの例

図3 切り取ったジョッキ画像の例

3.5 データセットの撮影条件

データセットのためのシーン画像の撮影位置を図4に示す。一般的な店舗では天井までの高さが240cm程度までであり, 机の高さは70cmが多い。そのため, 今回は撮影時の高さは机の高さ(70cm)より136~172cm高い状況を想定し図4の配置とした。撮影時の照明条件は表1に示す。

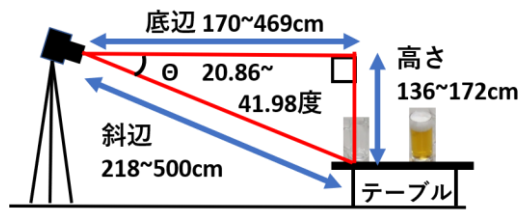


図4 撮影位置の略図

表1 照明条件 (Lux)

名称	範囲
蛍光灯	118~214, 432~629
エジソンランプ	19.5~94.1
照明なし	3.1~12.2

4 実験と結果

4.1 空きジョッキ判別器の学習

作成したジョッキデータセットを 7:1.5:1.5 で分割し、訓練データ 914 枚、検証データ 196 枚、テストデータ 196 枚とした。ResNet-50, VGG-19, InceptionV3 の 3 つの CNN を用いて空きジョッキ判別器を学習し、その精度評価を行った。

4.2 学習環境と学習パラメータ

各 CNN の学習では、CPU に Intel(R) Xeon(R) の 1 コアを、GPU に NVIDIA Tesla K80 を使用した。入力は 224×224 pixel の 3 チャンネルで、オプティマイザは Adam[5] とした。CNN の学習パラメータを表 2 に示す。

表2 各空きジョッキ判別器のパラメータ

CNN 名	Batch 数	Epoch 数	学習率
ResNet-50	64	30	初期値は 全て 0.0001
VGG-19	64	50	
InceptionV3	4	20	

4.3 空きジョッキ判別器の学習結果

学習後の空きジョッキ判別器にテストデータを与えた際の正解率、適合率、再現率を表 3 に示す。表 3 より InceptionV3 がすべて正解する結果となり、他のモデルも高い正答率を達成している。VGG-19 の再現率が最も低い値になったが、すべての値が 96% を超える、非常に高い結果となった。

4.4 2 段階式空きジョッキ検出器としての精度

表3 空きジョッキ判別器の精度比較 (%)

CNN 名	正解率	適合率	再現率
ResNet-50	97.44	98.05	97.11
VGG-19	97.95	100.00	96.11
InceptionV3	100.00	100.00	100.00

YOLOv3-SPP によるジョッキ検出と空きジョッキ判別器を組合せた 2 段階構成による空きジョッキ検出器の精度評価を行った。今回は 30fps で撮影された

10 秒の図 4 と同様のカメラ配置による食事シーン動画を 1 フレームごとに YOLOv3-SPP へ入力した。

空きジョッキ検出の正解率、適合率、再現率を表 4 示す。表 3 の空きジョッキ判別器の精度に比して全体的に低い事から、YOLOv3-SPP によるジョッキ検出の精度に問題があることがわかる。

表4 YOLOv3-SPP と各空きジョッキ判別機の組合せによる正解率、適合率、再現率 (%)

CNN 名	正解率	適合率	再現率
ResNet-50	43.33	14.94	17.51
VGG-19	44.28	1.66	1.36
InceptionV3	32.06	12.97	20.74

5 考察

YOLOv3-SPP によるジョッキ検出の精度が悪いのは、[cup], [bottle], または[wineglass]クラスでの検出であることと、使用した食事のシーン動画ではジョッキと他の食器の重なりが多かったからと考えられる。このことから、空きジョッキの検出精度向上には今回用いた YOLOv3-SPP のモデルに対し、部分隠れも含めたジョッキ画像で増分学習することや、後段の空きジョッキ判別器の学習データセットに、実際の居酒屋で撮影されたジョッキ画像を含めることが考えられる。また、汎用性は乏しくなるが、空きジョッキのだけを学習する end-to-end の 1 段階式の空きジョッキ検出器も検討する必要があるだろう。

6 まとめ

今回我々は前段として YOLOv3-SPP によるジョッキ検出器を、後段として独自に学習した空きジョッキ判別器を組合せて 2 段階式の空きジョッキ検出器を作成した。前段では COCO データセットで学習済みの YOLOv3-SPP を用いてジョッキと思われる食器を検出し、その検出矩形画像を後段の空きジョッキ判別器で「空きジョッキ」と「中身在りジョッキ」に判別した。

後段の空きジョッキ判別器は ResNet50 と VGG-19 と InceptionV3 の 3 種類に対して、YOLOv3-SPP で切り取ったジョッキ画像で学習を行い、高い判別精度を得たが、前段の検出精度の低さにより、後段を組合せたジョッキ検出器全体の検出精度には課題が残った。前段の精度向上、もしくは、1 段階式の空きジョッキ検出器の実装が今後の課題である。

参考文献

- [1] J. Redmon, et al., "YOLOv3: An Incremental Improvement", In *arXiv:1804.02767*, 2018.
- [2] K. Simonyan, et al., "Very Deep Convolutional Networks for Large-Scale Image Recognition," In *arXiv:1409.1556*, 2014.
- [3] K. He, et al., "Deep Residual Learning for Image Recognition," In *CVPR*, 2016.
- [4] C. Szegedy, et al., "Rethinking the Inception Architecture for Computer Vision," In *CVPR*, 2015.
- [5] D. P. Kingma, et al., "Adam: A method for stochastic optimization," In *arXiv:1412.6980*, 2014.