外部メモリの導入による歩行者検出の計算コスト削減

EP15043 神谷知紀　　指導教授：山下隆義

1. はじめに

歩行者検出を困難としている要因の一つとして，歩行者と背景の各クラスのバリエーションが多いことがあげられる．そのため，各クラスのバリエーションを網羅して学習するためにはユニット数や層数などのパラメータが膨大となる．そこで本研究では，外部メモリを取り入れた Memory Network[1] のフレームワークを歩行者検出手法へ応用する．外部メモリを用いたネットワークを構築することで，パラメータ数を削減し，同程度の歩行者検出を可能とする．

2. Memory Network

外部メモリを導入した Memory Network (MemNet) は，入力データから特徴量を抽出して外部メモリへ格納し，外部メモリを参照しながら認識結果を出力する．外部メモリを参照する際は，Soft Attention を用いて関連度のあるメモリを選択する．

3. 提案手法

提案手法のネットワーク構造を図 1 に示す．提案手法では，Faster R-CNN[2] をベースとして外部メモリを導入した歩行者検出を行う．MemNet の外部メモリには，Region Proposal Network (RPN) により得られる歩行者と誤検出の特徴マップを記憶する．

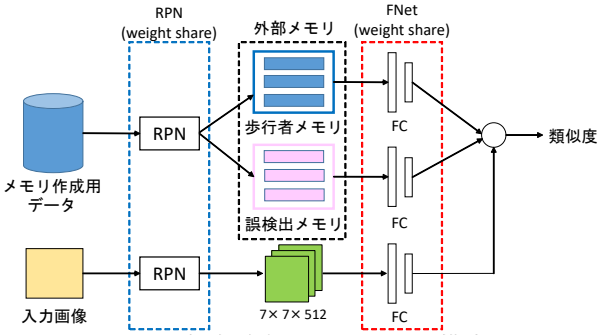

図 1：提案手法のネットワーク構造

3.1. 提案手法の流れ

まず最初に外部メモリを作成する．RPN で検出した歩行者候補領域の特徴マップを，教師ラベルから歩行者及び誤検出に分け，クラスタリングにより得られたサンプルを外部メモリに格納する．そして，Triplet loss を用いて特徴ベクトル抽出ネットワーク (FNet) の学習を行う．推論時は，テストデータの歩行者候補領域と各メモリからユークリッド距離を求めることで，類似度を計算する．算出した類似度から歩行者候補領域が歩行者か否かを識別する．

3.2. 外部メモリの作成

外部メモリは，歩行者の特徴を格納する歩行者メモリと誤検出の特徴を格納する誤検出メモリで構成されている．はじめに，歩行者検出用に事前学習した RPN を用いて，画像中の物体候補領域を検出する．検出した歩行者候補領域は歩行者と誤検出に分類され，RoI プーリングにより 7×7 の固定サイズの特徴マップにリサイズする．リサイズした特徴マップを k-means 法を用いて歩行者と誤検出それぞれ個別にクラスタリングする．最後に各クラスタのセントロイドをサンプルとして外部メモリに格納する．

3.3. 階層型クラスタリング

各メモリと入力された歩行者候補領域の類似度を計算する際，全探索を行うと計算コストが増加する．そこで，階層型クラスタリングを導入する．階層クラスタリングは，クラスタリングを数回に分けることで計算コストを削減する．階層型クラスタリングを用いた外部メモリを図 2 に示す．図 2 のように，段階的にクラスタリングを行うことで，全探索回数は各層のメモリに対する類似度計算の合計回数となる．

3.4. Triplet loss による学習

類似度は，FNet から得た特徴ベクトルから算出する．FNet の学習には Triplet loss を用いる．Triplet loss は，

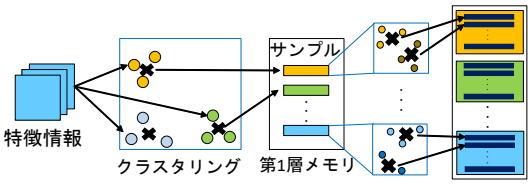

図 2：階層型のメモリ構造

anchor，positive，negative の 3 つのサンプルを入力し，anchor と positive の距離を小さく，negative の距離を大きくするように学習する．anchor は，入力画像から得た物体候補領域である．positive は，anchor と同じクラスの外部メモリ，negative は異なるクラスの外部メモリに属するサンプルである．

4. 実験

Faster R-CNN を従来手法とし，提案手法と比較して提案手法の有効性を調査する．

4.1. 実験概要

精度の比較では，Reasonable，Reasonable small (small)，Reasonable occlusion (occ)，All の 4 つの評価指標で評価する．また，計算コストの比較では，パラメータ数及びメモリの探索回数を比較する．使用するデータセットは，Cityperson Dataset である．外部メモリの作成に歩行者領域 14,660 個と誤検出領域 2,614 個，学習用に 2,975 枚，テスト用に 500 枚使用する．全探索型の各メモリの数は，歩行者メモリ 100 個と誤検出メモリ 600 個とする．階層型の各層のメモリの数は，歩行者メモリでは 1 段目と 2 段目に 10 個，誤検出メモリでは 1 段目に 5 個と 2 段目に 120 個とする．また，FC 層の 2 層目のユニット数を 4096 または 128 個に変えて実験する．

4.2. 実験結果

Faster R-CNN と提案手法の精度及びパラメータ数，メモリの探索回数を表 1 に示す．表 1 において，階層型のメモリの数の表記を歩行者メモリの場合，歩 (1 段目のメモリ数，2 段目のメモリ数) とし，誤検出メモリの場合，誤 (1 段目のメモリ数，2 段目のメモリ数) とする．表 1 より，提案手法の階層型 FC128 は評価指標 small において Faster R-CNN より精度が向上した．その他の 3 つの指標においても Faster R-CNN と同等の未検出率を得られた．またパラメータ数においては，全探索型と階層型の FC128 が Faster R-CNN より 16M 削減できた．探索回数では，階層型の方が探索回数を削減している．よって階層型 FC128 は，Faster R-CNN と同等以上の精度を持ち，計算コストを削減できたといえる．

表 1：歩行者検出の精度と計算コスト

手法			未検出率 [%] Reasonable	small	occ	All	パラメータ	探索回数
Faster R-CNN			**19.63**	49.71	**49.51**	**44.26**	119M	-
提案手法	全探索型 FC4096	歩行者 100 誤検出 600	20.04	53.86	50.38	44.90	119M	700
	全探索型 FC128	歩行者 100 誤検出 600	21.46	52.14	51.22	45.41	103M	700
	階層型 FC4096	歩 (10，10) 誤 (5，120)	21.10	52.25	53.78	46.15	119M	145
	階層型 FC128	歩 (10，10) 誤 (5，120)	19.80	**49.63**	50.60	44.66	103M	145

5. おわりに

本研究では，外部メモリを用いて歩行者検出の計算コスト削減する手法を提案し，メモリでの計算コストを削減して従来手法と同等以上の精度を得られた．今後は，メモリの階層数や各層のメモリ数を変更をして，更なる精度向上と計算コスト削減を目指す．

参考文献

[1] S. Sukhbaatar, *et al.*, “End-To-End Memory Networks”, NIPS, 2015.

[2] S. Ren, *et al.*, “Faster R-CNN:Towards Real-Time Object Detection with Region Proposal Networks”, NIPS, 2015.