修士論文

漫画画像を対象とした物体検出

平成 30 年 1 月 29 日

情報理工学系研究科 電子情報学専攻

48-166414　　小川 徹

指導教員　　相澤 清晴 教授

内容梗概

本論文では漫画画像に対する物体検出に着目する．物体検出は写真などの自然画像の分野で広く研究されており, 現在は畳み込みニューラルネットワーク (convolutional neural networks, CNN) を用いた手法が高い性能を示している．一方で漫画は自然画像と比較して物体間の重なりが大きく, 自然画像を対象とした既存の物体検出手法を漫画に適用した場合, 性能の低下が起こる．本論文ではこの問題を解決するために新しい物体検出手法, SSD300-fork を提案する．この手法は既存の物体検出手法である SSD300 を改良し, ネットワークを分岐させることで重なりが大きい物体でも適切に扱えるようにしたものである．SSD300 を含む既存の物体検出手法との比較の結果, 提案手法での性能向上が確認できた．

目次

図目次

表目次

第 1 章

序論

1.1 背景

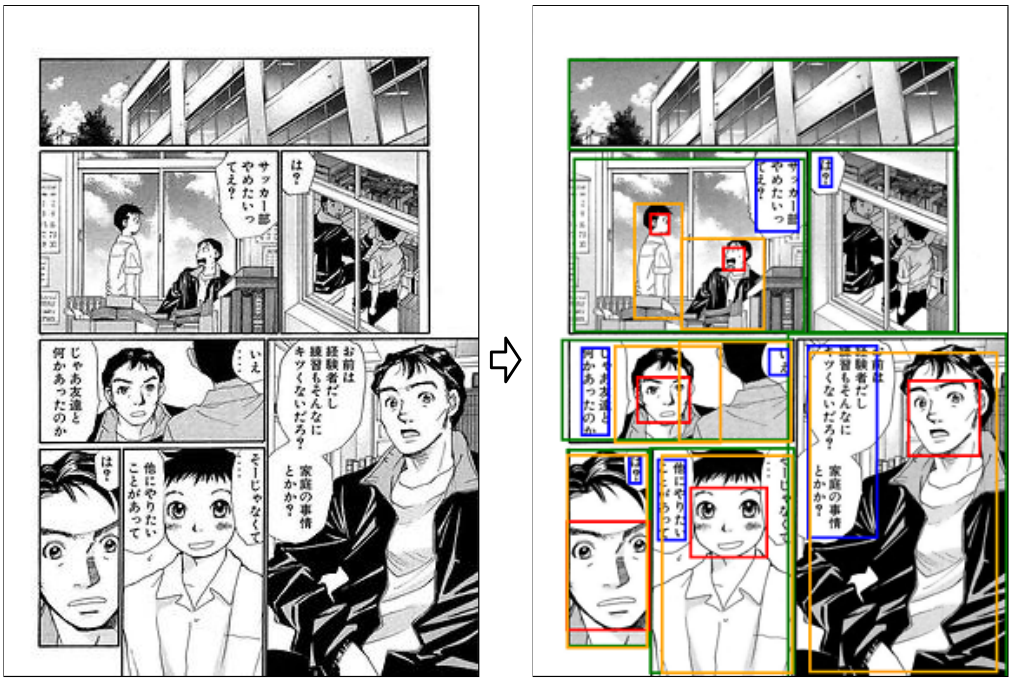
漫画は娯楽と文化の双方の面で重要な役割を担うメディアの一つである．従来の漫画は紙に印刷された白黒のものが主流であったが, 近年では電子漫画や海外輸出などの漫画市場の拡大に伴い, 漫画の形式も多様化している．同時に漫画に対する情報技術, 特に画像認識 · 処理技術の需要が高まっている．

本論文ではそのような画像認識技術の中でも基礎的な技術の一つである物体検出に着目する．漫画画像における物体検出はリターゲティング, 翻訳, 検索などの様々な分野に応用できる．リターゲティングは漫画をタブレット端末等の媒体に合わせて作り変える技術である．タブレット端末向けの漫画ではスクロールして読みやすいように, 縦方向にコマを並べるという工夫がされることがある．また縦長の画面に合わせて要素を再配置することも場合によっては必要となる．漫画の物体検出, 特にコマの検出はこの処理を行う上で重要な要素技術となる．翻訳は海外市場の拡大とともに需要が高まっている分野である．翻訳の自動化にはテキストの検出, 文字認識 (OCR), 機械翻訳が不可欠である．インターネット経由で多くの漫画が入手可能となった今日では検索の技術も重要となっている．物体検出の技術を用いることでキャラクター検索やセリフ検索などの新しい検索手法が可能となる．

1.2 目的

背景で述べたように本論文では漫画画像に対する物体検出を目的とする．これは漫画画像を入力として画像中に含まれる物体およびそのカテゴリを出力する課題である (図 1.1)．物体検出は大きく bounding box detection (object detection), semantic segmantation, instance segmentation の 3 種類に分けられるが (図 1.2), 本論文では bounding box detection を取り扱う．

Bounding box detection は物体の位置を外接矩形で表現し, それぞれについてカテゴリを出力する課題である．一般に外接矩形の回転は考えず, 位置および大きさを 4 次元のベクトル $(x_{min}, y_{min}, x_{max}, y_{max})$ で表現する．Object detection と言った場合, この課題のことを指すこ



(a) 入力 (漫画画像) [*1]　　(b) 出力 (物体の検出結果) [*1]

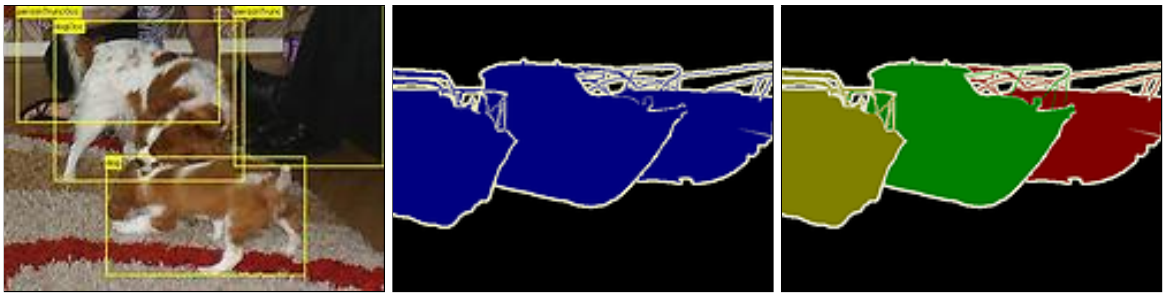
図 1.1: 本論文の目的．入力された漫画画像 (a) に含まれている物体の位置 (外接矩形の座標) およびカテゴリを出力する (b).

とが多い．この課題を取り扱った手法については 2.3 節で紹介する．Semantic segmentation はピクセルごとにどのカテゴリに属しているかを出力する課題である．Bounding box detection に比べて詳細な形状を推定できる一方で, 同じカテゴリに属する異なる物体を判別できないという欠点がある．たとえば図 1.2b では 3 隻の船が区別されず, 全て同じカテゴリの領域として指定されている．Instance segmentation は bounding box detection と semantic segmentation の両方の利点を備えており, 各物体ごとにピクセルレベルの領域を推定する．たとえば図 1.2c のように 3 隻の船がそれぞれ別の領域として区別される．

1.3　検出するカテゴリ

検出するカテゴリとしてはコマ, テキスト, 顔および全身の 4 つを採用する．これらのカテゴリは漫画を構成する基本的な要素であり, その検出技術はリターゲティング, 翻訳, 検索など様々な応用が可能である．既存の漫画研究でもこれらの要素を取り扱ったものが多い．図 1.3 に 4 カテゴリの例を示す．各矩形が物体の外接矩形を表現しており, その色がカテゴリに対応している (コマ: 緑, テキスト: 青, 顔: 赤, 全身: 橙).

[*1] “やまとの羽根” ©咲 香里



(a) bounding box detection　(b) semantic segmentation　(c) instance segmentation

図 1.2: 物体検出の種類 [1]. Bounding box detection (object detection) (a) では各物体の外接矩形およびカテゴリを推定する. Semantic segmentation (b) ではピクセルごとにどのカテゴリの領域かを推定する. Instance segmentation (c) では各物体の領域をピクセル単位で推定する.

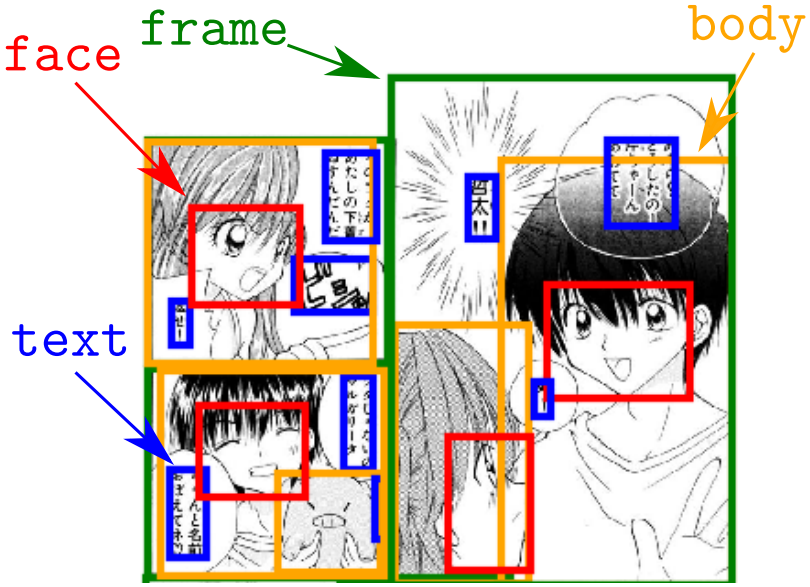


図 1.3: 検出の対象となる 4 カテゴリ*2

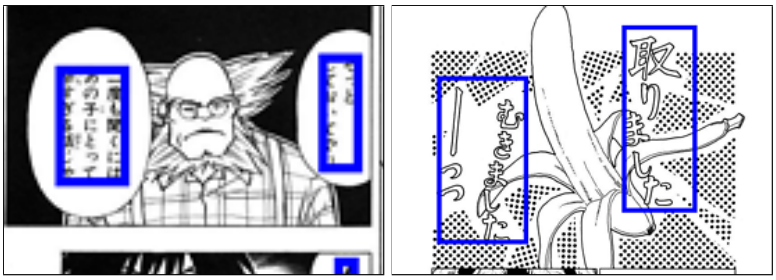
各カテゴリについて簡単に述べる. コマは一つのシーンを描写する単位であり, 通常漫画のページは複数のコマで集合として構成される (図 1.4). その他の要素 (キャラクター, 背景, テキスト等) はコマの中に描かれることが多い. 読者はこのコマを特定の順序に従って読み進めていくことでストーリーを追っていく. コマは一般的に黒い縁を持った長方形であるが, 縁がなかったり (図 1.4b) 多角形だったり (図 1.4c) とその様式は多様である.

テキストはキャラクターの発言や心理描写, ナレーションなどを含む要素である (図 1.5). 多くのテキストはフキダシと呼ばれる白く塗られた領域の中に配置される. このときフキダシの領域は

*2 “爆烈！かんふー娘” ©うえだ 美貴

(a) 一般的なコマ [*3]　(b) 枠のないコマ [*4]　(c) 矩形ではないコマ [*5]

図 1.4: コマの例. 多くのコマは黒い枠で囲まれた矩形領域である (a) が, 漫画によっては枠がなかったり (b) 矩形でなかったり (c) する.



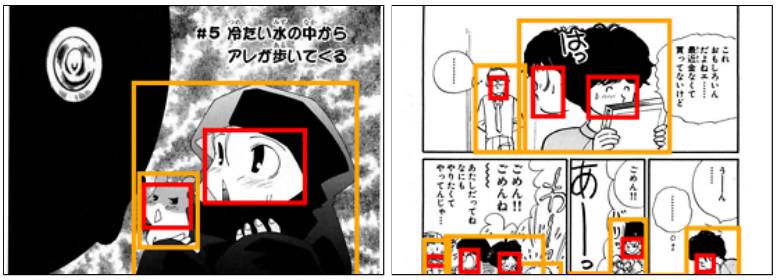
(a) フキダシとテキスト [*6]　(b) フキダシのないテキスト [*7]

図 1.5: テキストの例. 一般にテキストはフキダシと呼ばれる領域の上に配置される (a). フキダシを設けずに直接テキスト書くこともある (b). また (a) のテキストはフォントによって表現されている一方で, (b) のテキストは手書きで書かれている.

他の要素の上を覆う形になることも少なくない (図 1.5a). 一方で, フキダシを設けずに直接テキストを書き込む場合も存在する (図 1.5b). 一般にテキストの文字はフォントを使って表現される (写植と呼ばれる) が (図 1.5a), 手書きのテキストも用いられることもある (図 1.5b).

顔および全身はキャラクターを構成する要素である. 本論文では藤本ら [8] の定義に基づき, 顔を眉からアゴおよび頬を含む矩形領域, 全身を任意の身体の一部 (頭, 髪, 腕, 脚など) を含む領域とした. 顔は身体の一部であるから, 顔が存在する場合は常にその顔領域を含む全身の領域が存在することになる. ただし, 漫画によっては頭の運動を表現するために, 複数の顔を同一の全身の中に描

[*3] “サイコスタッフ” ©水上 悟志
[*4] “プラチナジャングル” ©篠原 正美
[*5] “ベルモンド Le VisiteuR” ©石岡 ショウエイ
[*6] “ありさ^2” ©八神 健
[*7] “空っぽハイスクール” ©高口 里純



(a) 顔と全身 [*8]　　(b) 複数の顔を含む全身 [*9]

図 1.6: 顔と全身の例. 顔は眉, アゴ, 頬を含む領域として定義される (a). 顔の運動を表現するときなど, 1 つの全身領域が複数の顔を含んでいることもある (b).

くことがあるため, 顔と全身の一対一対応は必ずしも成立しない (図 1.6b). イヌやネコなどの人間以外のキャラクターについては, 主要キャラクターであれば顔や全身も検出の対象とする. 主要キャラクターの基準については藤本ら [8] の定義に従う. たとえば図 1.6a ではハムスターのキャラクターの顔および全身も検出対象になっている. 顔や全身はコマの中に描かれることが多く, したがってその一部がコマの枠によって切り取られることが良くある. この場合, 外接矩形は切り取られた (描かれている) 領域について定義する.

1.4　構成

本論文の構成は次の通りである. 本章では背景, 目的および検出の対象とするカテゴリについて述べた. 2 章では関連研究, 特に漫画画像データセットおよび物体検出に関連した先行研究について紹介する. 3 章では既存手法の問題点および, それを改善するための提案手法である SSD300-fork について説明する. 4 章では漫画データセットである Manga109 および eBDtheque を利用した実験とその結果について述べる. 最後に 5 章で本論文のまとめおよび今後の展望について述べる.

[*8] “あくはむ” ©新居 さとし
[*9] “愛さずにはいられない” ©よし まさこ

第 2 章

関連研究

2.1 漫画データセット

本節では現在利用可能な漫画データセットについて紹介する (表 2.1). 一般的に流通している漫画は著者あるいは出版社によって保有されており, 著作権の問題により自由に用いることができないことが多い. したがって学術研究に利用可能なデータセットの数も限られている.

Guérin らは eBDtheque [2] というデータセットを公開している. このデータセットは 100 ページの漫画画像から構成されている (図 2.1). 各ページはフランス, アメリカおよび日本の漫画から抜粋されたものである. このデータセットでは画像に加えて, 手動でつけられたアノテーションも公開されている. このアノテーションはコマ, テキスト行, フキダシおよびキャラクターの外接多角形であり, 物体検出の評価に利用できる. テキスト行はテキストを構成する文字を行ごとにまとめたものであり, 追加のアノテーションとして文字データが付与されている. このデータセットは漫画の研究においてもっとも広く用いられているデータセットであるが, 画像の枚数が少なく, 本論文でのモデルの学習に利用するのは困難である. そのため 4.2 節の実験では別のデータセット (Manga109) を利用して学習したモデルを転用し, eBDtheque では評価のみを行う.

Mohit らは COMICS [3] という漫画のデータセットを公開している. このデータセットは漫画データセットの中で最大の規模のものであり, アメリカの漫画 3,948 冊を含んでいる (図 2.2). またコマおよびフキダシについて外接矩形のアノテーションを提供している. 大規模なデータセットの全てのページに手動でアノテーションをつけることは困難であるため, これらのアノテーションは機械学習を用いて半自動で生成されている. この手法ではまず 500 ページについて手動でコマのアノテーションを行い, そのデータを利用して既存の物体検出手法である Faster R-CNN [9] の学習を行う. そして残りのページに学習された Faster R-CNN を適用することで, 疑似的なアノテーションを生成している. フキダシについても同様に 1500 個のコマについてフキダシのアノテーションを手動で付与し, Faster R-CNN の学習を行っている. また検出されたフキダシには OCR を適用することで文字データ化も行っており, こちらもアノテーションとして提供されている. これらの機械学習により生成されたデータは形式としては本論文の目的と合致しているが, 物体検出によって自動的に生成されているため, 物体検出の学習や評価に用いるのは不適切である.

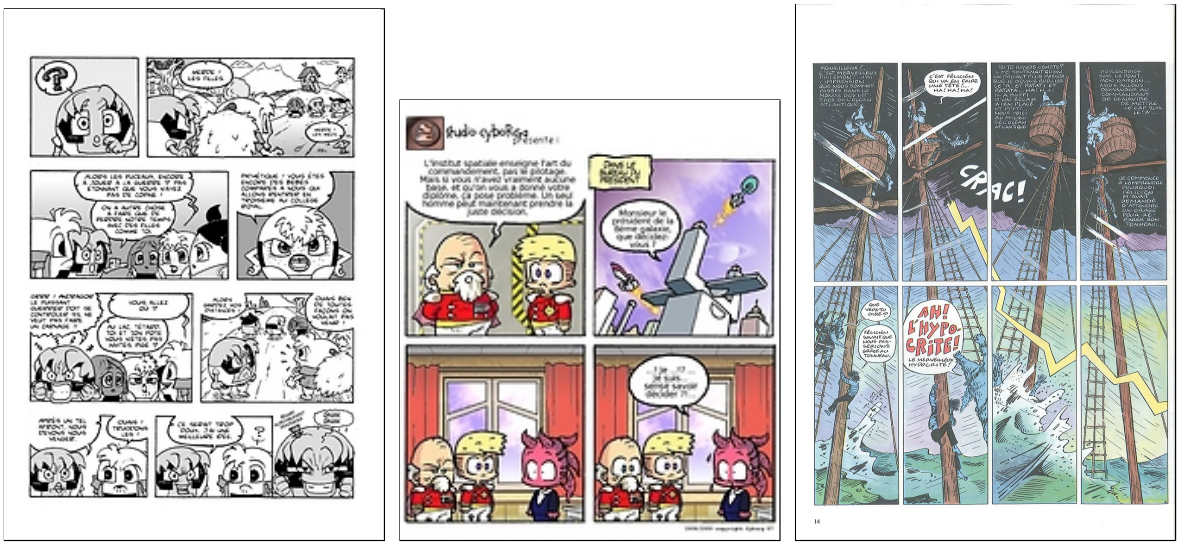


図 2.1: eBDtheque [2]

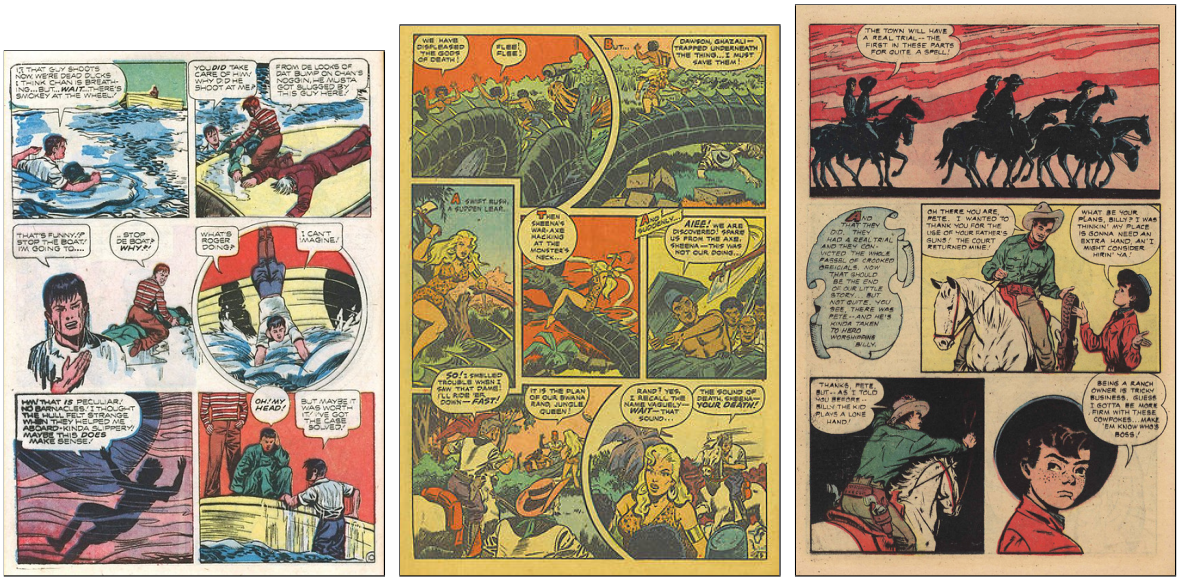
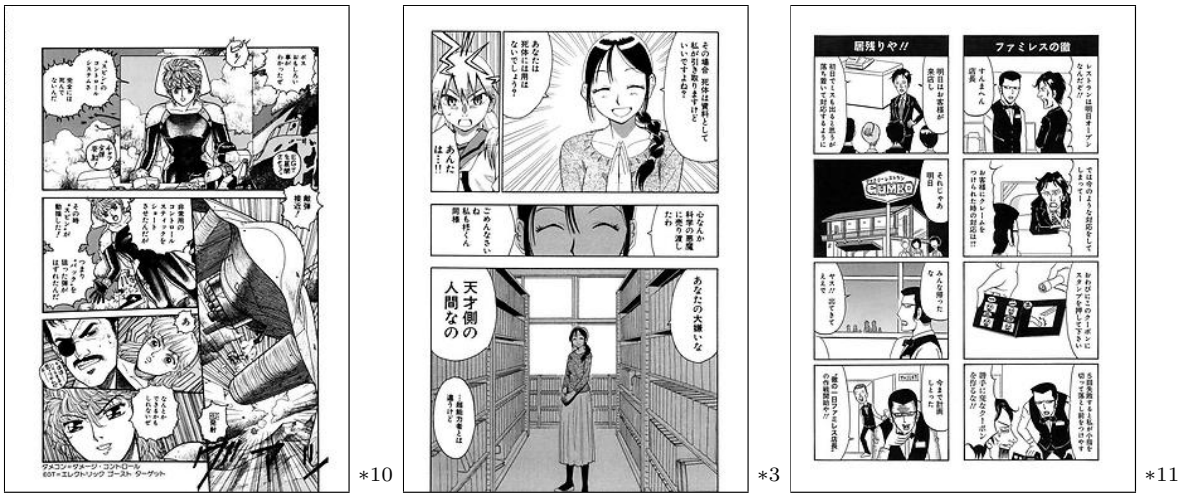


図 2.2: COMICS [3]

松井らは Manga109 [4] という日本の漫画 109 冊からなるデータセットを公開している (図 2.3). また藤本ら [8] によって Manga109 に手動アノテーションが付与されており, コマ, テキスト, 顔および全身の 4 カテゴリについて外接矩形の情報が与えられている. またテキストについてはその内容の文字データが, 顔および全身についてはキャラクター名が, それぞれ追加の情報として含まれている. 本論文ではこのデータセットを主に利用して学習や評価を行う.

*10 “ARMS” ©加藤 雅基
*11 “徹さん” ©川口 憲吾



*10 *3 *11

図 2.3: Manga109 [4]

表 2.1: 漫画データセットの比較

			アノテーション数			
データセット	冊数	ページ数	コマ	テキスト	顔	全身
eBDtheque [2]	25	100	850	1,092	-	1,550
COMICS [3]	3,948	198,657	(1,229,664) †	(2,498,657) †	-	-
Manga109 [4, 8]	109	10,130 ‡	103,900	147,918	118,715	157,152

† 疑似アノテーション (自動アノテーション). ‡ 見開きページ.

2.2 物体検出を利用した漫画研究

Chu ら [10, 11] は漫画の画風を表現するための特徴量を提案している. この手法はコマ, フキダシおよびキャラクターの位置と大きさを元に特徴量を計算している. Rigaud ら [12] は与えられたフキダシに対して対応する話者を推定する手法を提案している. この手法ではキャラクターとフキダシの検出を最初の処理として行っている. Le ら [13] 漫画画像を検索するための手法を提案している. この手法では検索に必要な漫画画像特徴量として, ページ内におけるコマの位置を利用している. 荒巻ら [14] は漫画画像中のテキストを認識する手法を提案している. ここでの認識とは漫画画像中からテキスト領域を検出し, その内容を文字データ化することである. この手法では画素の連結情報を利用してテキスト領域の検出を行っている.

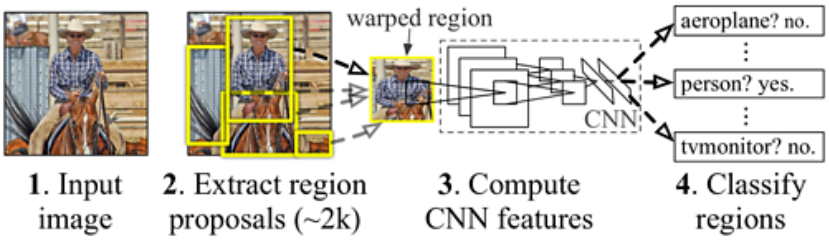


図 2.4: R-CNN の構成 [5]. 画像中から物体らしい候補領域を大量に選び, 各領域ごとにカテゴリ分類を行う.

2.3 自然画像における物体検出手法

自然画像における物体検出は PASCAL VOC [1] や MS COCO [15] などのコンテストが開催されるなど, 活発な研究がなされている分野である. 自然画像での物体検出の基礎となる手法として R-CNN [5] がある. この手法ではまず Selective Search [16] などの手法を用いて, 画像中から物体らしい候補領域を大量に選択する. そして各領域について特徴量を計算し, その領域がどの物体か (あるいは背景か) という分類問題を解くことで物体を検出している (図 2.4). また外接矩形をより正確にするために候補領域に対する真の外接矩形のズレを推定するという問題も同時に解いている.

R-CNN の後継として Fast R-CNN [17] や Faster R-CNN [9] がある. Fast R-CNN は画像全体の特徴を計算しておき, そこから候補領域を切り出すことで, 候補領域ごとに特徴量を計算する必要があるという R-CNN の欠点を改善している. Faster R-CNN は候補領域を提案する部分にも Region Proposal Network (RPN) と呼ばれるニューラルネットワークを用いることで, 候補の提案と分類, 矩形の推定という 3 つの問題を同時に解いている.

Redmon ら [6] は Faster R-CNN とは異なる検出のアプローチとして YOLO と呼ばれる anchor ベースの手法を提案している. この手法では RPN を用いず, あらかじめ決められた anchor box と呼ばれる領域ごとにクラスの分類問題および矩形の推定問題を解いている (図 2.5). 複数の物体に対応するために anchor box は位置, 大きさ, 縦横比の異なるものを複数容易し, 各物体に最も近い anchor box が反応するように学習を行う. テスト時には近傍の anchor box が物体の位置およびカテゴリを出力するので, その結果を non-maximum suppression (NMS) によって統合する. この手法では RPN が不要となり, 1 回の計算で全部の物体に対しての推論値が得られるため, より高速な物体検出が可能となる.

また YOLO の発展として. YOLOv2 [18], SSD [7] や DSSD [19], FPN [20], RetinaNet [21] などの手法が提案されている. SSD [7] は YOLO では 1 個しか使っていない特徴量マップを複数個に増やしており, 様々なスケールの物体に対応することで精度を向上させている (図 2.6). また DSSD [19], FPN [20], RetinaNet [21] では一度畳み込んだ特徴マップを拡大して利用することでグローバルな情報を検出に利用している. YOLO を含む, これらの anchor ベースの手法は精度や

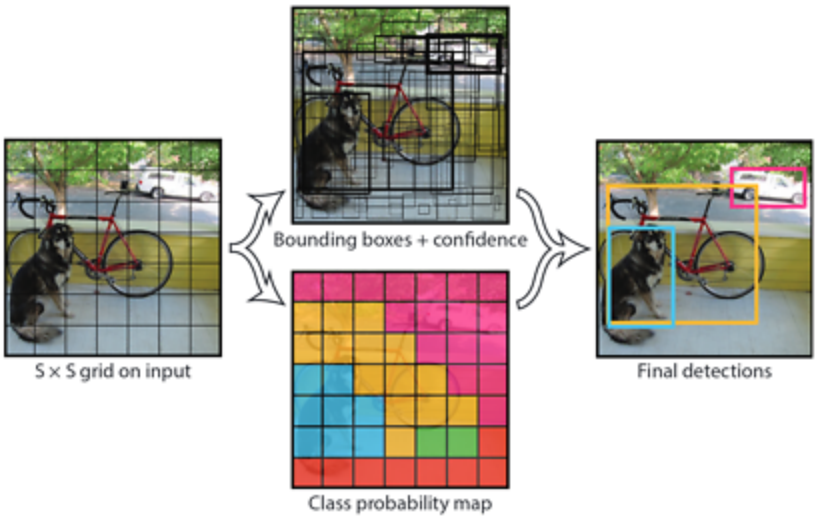


図 2.5: Anchor ベースの手法の概要 [6]. 画像を小領域に区切り, 各領域ごとに矩形推定とカテゴリ分類を行う. 最終的に物体らしさの確信度が高い矩形を NMS によって選択する.

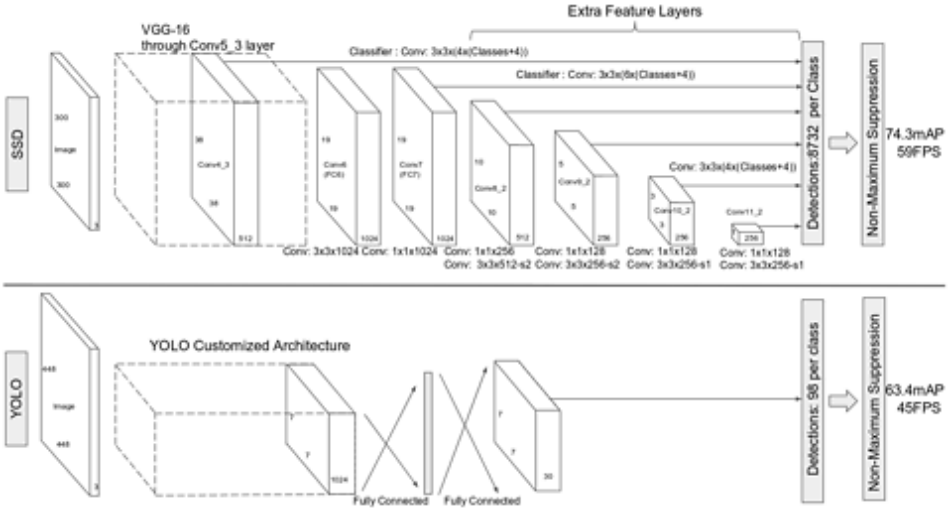


図 2.6: SSD および YOLO の構成 [7]. SSD では特徴量マップを複数利用することで様々なスケールの物体に対応している.

速度ともに高い性能を示しており, 自然画像における物体検出の主流となっている. これらの手法の詳細については 3.1 節で述べる.

第 3 章

提案手法

本章では提案手法である SSD300-fork について説明する．この手法は既存の物体検出手法を漫画に適用した際に起こる “割り当て問題” に着目したものであり, 既存の深層学習を利用した物体検出手法を元に, 出力層を分岐 (fork) することでこの問題の解決を図る．各分岐がそれぞれ 1 つのカテゴリの検出を担当することで, カテゴリ間の競合を回避することが狙いである．まず 3.1 節で現在の物体検出において主流である anchor ベースの手法について説明し, 3.2 節で本論文が注目する割り当て問題について述べる．また 3.3 節で提案手法である fork モデルについて説明し, 3.4 節で SSD300-fork についての詳細を述べる．

3.1 Anchor ベースの手法

2.3 節で述べたように, 自然画像における物体検出では anchor ベースの手法が主流となっている．図 3.1 はこの方式の可視化したものである．破線で囲われた矩形がそれぞれ 1 つの anchor box を表現しており, anchor box をまとめる実線の枠が anchor set (anchor box の集合) を表現している．各 anchor box は決まった位置, 形状, 大きさを持っている．この手法での検出 (テスト) は次のように行われる与えられた画像に対して, 各 anchor box がそれぞれ最も近い物体の位置およびカテゴリを推定する．複数の anchor box が同じ物体に対する推定をすることがあるため, non-maximum suppression (NMS) を適用することで重複する推論結果を統合する．学習時には各 anchor box が画像の特徴量からそれぞれ最も近い物体の位置およびカテゴリを返すように学習が行われる．このとき学習データに含まれる ground truth は anchor box へと割り当てられる．たとえば図 3.1 では右上の楕円が右上の横長の anchor box に割り当てられている．このとき割り当てられた anchor box は入力画像に対して右上の楕円の位置およびカテゴリ (楕円) を返すように学習される．この割り当ては物体の位置, 大きさおよび形 (縦横比) を元に決定される．

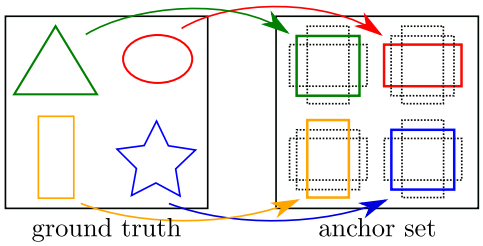


図 3.1: Anchor ベースの手法における学習時の割り当て．物体は位置や大きさに応じて適切な anchor box (破線の矩形) に割り当てられる．物体の重なりが少ない通常の画像では、全ての物体が最低 1 個の anchor box へと割り当てられる．

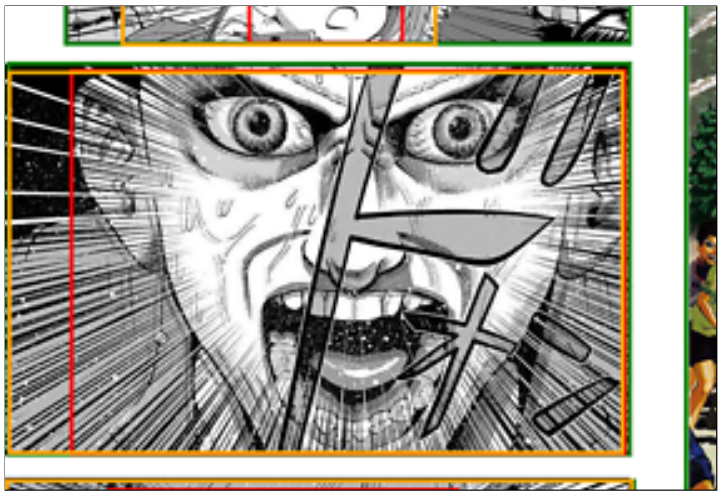


図 3.2: 漫画における重なりが大きいカテゴリの例[*12]．コマ (緑), 顔 (赤) および全身 (橙) が大きく重なっている．

3.2 割り当て問題

本節では anchor ベースの手法を重なりが大きい物体が存在する画像に適用した場合に起こる “割り当て問題” ついて述べる．これは位置や大きさが近い物体が複数存在する場合に, これらの物体を適切に anchor box へと割り当てることができないという問題である．図 3.3 では青の星形と緑の三角形がほぼ同じ場所に配置されており, その大きさや縦横比も近い．位置および大きさに従って, 両方の物体を右下の正方形の anchor box へと割り当てようとするが, 1 つの anchor box が担当できるのは最大で 1 つの物体までであるため, 結果として少なくとも一方の物体は割り当てに失敗する．このような重なりは PASCAL VOC [1] や MS COCO [15] といったデータセットでは起こりにくいが, 漫画のようなデータセットにおいてはしばしば起こる (図 3.2)．同様の問題はテスト時にも起こる．各 anchor box は最近傍の物体について出力するため, 重なりが大きい物体がある場合は一方が出力されないことがある (図 3.4)．

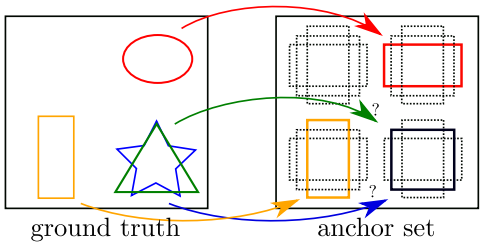


図 3.3: 学習時の “割り当て問題”. 画像が重なりが大きい物体を含む場合, 割り当てるべき anchor box が重複し, 割り当てられない物体が生じる. この場合, 青の星形と緑の三角形が競合している.

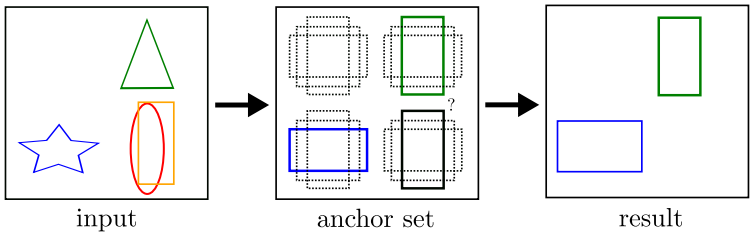


図 3.4: 推論時の “割り当て問題”. 重なりが大きい物体がある場合, 担当すべき anchor box が重複してしまい, 適切に検出ができない.

3.3 Fork モデル

本節では前述した “割り当て問題” を解決するための提案手法である fork モデルについて述べる. このモデルではベースとなる物体検出手法の anchor set を複製し, C 個の anchor set を持つ. ここで C は検出すべきカテゴリの数であり, 本論文では $C = 4$ である (コマ, テキスト, 顔, 全身). 複製された anchor set はそれぞれが 1 つのカテゴリの検出を担当する. 図 3.5 の右側の 4 つの正方形は複製された 4 つの anchor set を表しており, 外枠の色 (緑, 赤, 橙, 青) は対応するカテゴリを表している. 各 anchor set に含まれる anchor box はベースとなるモデルの anchor box (図 3.1, 3.3) と個数, 位置, 形状および大きさが全て同じである.

Fork モデルの学習では各物体はそのカテゴリに基づいて対応する anchor set 内の anchor box へと割り当てられる (図 3.5). ここで入力されているデータは図 3.3 と同じもので, 青の星形と緑の三角形が大きく重なっている. Fork モデルでは青の星形は右下の青の anchor set 内の anchor box に割り当てが行われ, 緑の三角形は左上の緑の anchor set 内の anchor box に割り当てが行われる. このとき割り当てられる anchor set 内の anchor box はベースモデルと同様に位置や形状および大きさを元に決定される. したがって anchor set の段階で振り分けが行われることを除き, ベースモデルのときと同じ anchor box へと割り当てが行われる. このように anchor set を複製す

*12 “極限サイクロン” ©高波 伸

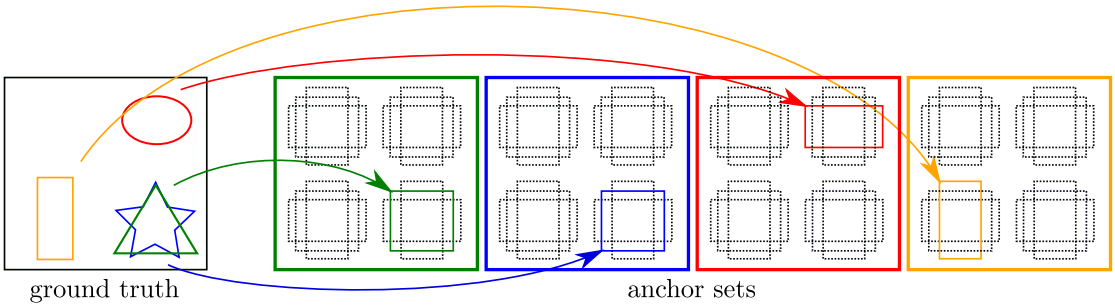


図 3.5: Fork モデルにおける割り当て．Anchor set を複製することで, 重なりの大きい物体であっても適切に割り当てを行うことができる．青の星形と緑の三角形は異なる anchor set に割り当てられている．

ることで, ベースモデルでは競合していた青の星型と緑の三角形を適切に異なる anchor box へと割り当てられる．

物体を anchor box に割り当てたのち, 各 anchor box ごとに分類問題および矩形の推定問題を学習する．分類問題については物体であるかどうかの二値分類問題を学習する．これは各 anchor box が担当するカテゴリの情報を持っているため, 通常のモデルで行われる $C+1$ カテゴリの多値分類が不要だからである．たとえば顔用の anchor set 内のある anchor box が物体を検出した場合, 検出された物体は顔だと自動的に判別が可能である．矩形の推定問題については通常のモデルと同様に anchor box に対する差分 (4 次元) の推定を行う．

推論時には各 anchor box で二値分類および矩形推定を行ったのち, 全 anchor set の結果を統合する (図 3.6)．前述のように検出された物体のカテゴリはどの anchor set 内から検出されたかによって決定される．図 3.6 の赤の楕円と橙の矩形のように大きく重なっている物体であっても, 異なる anchor set が対応することで検出することができる．

3.4 SSD300-fork

本節では既存の物体検出手法である SSD300 [7] に fork モデル適用した SSD300-fork について説明する．ベースのモデルとしては SSD300 以外のものも利用可能であるが, 既存の物体検出手法である Faster R-CNN, SSD300 および YOLOv2 の 3 手法のなかで最も高い性能を示した SSD300 を採用した (表 4.2)．

SSD300 はマルチスケール特徴抽出器と検出層から構成されている (図 3.7a)．マルチスケール特徴抽出器は VGG-16 をベースとしてたニューラルネットワークであり, 300×300 の大きさにリサイズされた画像を入力とし 6 個の特徴マップを出力する．このネットワークは入力された画像を畳み込みによって縮小していき, conv4_3 (norm4), conv_7, conv8_2, conv9_2, conv10_2, conv11_2 の合計 6 層の値を特徴マップとして利用する．特徴マップは全て正方形であり, i 番目の特徴マップの一片の大きさを g_i とすると, $g_1 = 38, g_2 = 19, g_3 = 10, g_4 = 5, g_5 = 3, g_6 = 1$ となる．

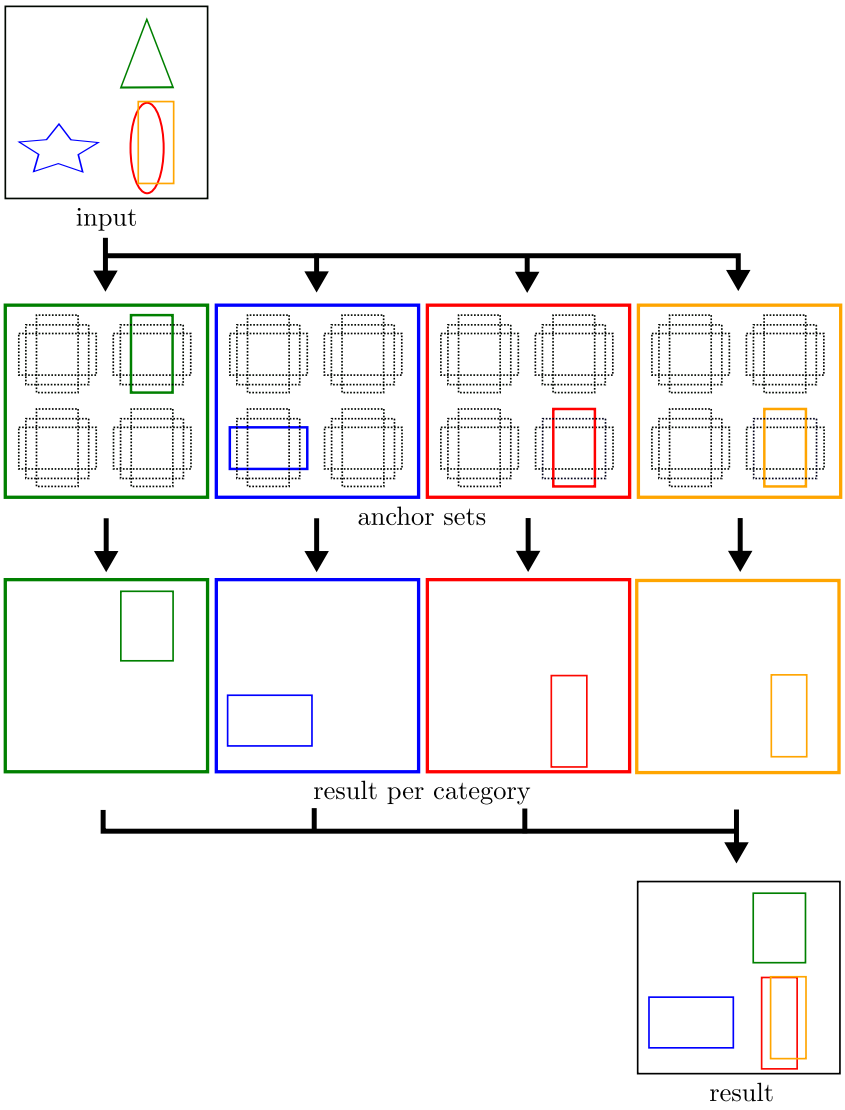


図 3.6: Fork モデルにおける推論．各 anchor set の検出結果をまとめることで全カテゴリの検出を行う．

検出層は出力された特徴マップをもとに各 anchor box ごとのカテゴリ分類および矩形推定を行うネットワークである．実際には各特徴マップにつきそれぞれ別の畳み込み層を適用することで計算が行われているが，本論文では便宜上，6 つの畳み込み層をまとめて 1 つの検出層として扱う．このとき検出層は特徴マップを入力として，2 つの多次元行列，$\mathbf{z}_{\mathrm{loc}}$ および $\mathbf{z}_{\mathrm{conf}}$ を計算するネットワークとして捉えられる．ここで $\mathbf{z}_{\mathrm{loc}} \in \mathbb{R}^{K\times 4}$ は矩形推定に対応しており，K は全ての anchor box の数である (後述するように SSD300 では $K = 8732$)．各 anchor box ごとに位置および大きさの推定のために 4 次元のベクトルが必要なため $\mathbf{z}_{\mathrm{loc}}$ は $K \times 4$ 個のスカラー値を持つ．この 4 次元は anchor box の中心から変位および anchor box の大きさとの差分 $(\Delta x_{center}, \Delta y_{center}, \Delta width, \Delta height)$ として表現される．同様に $\mathbf{z}_{\mathrm{conf}} \in \mathbb{R}^{K\times(C+1)}$ はカ

テゴリ分類に対応しており, 各 anchor box ごとに $C+1$ 次元のベクトルを出力する. ここで $c \in \{1, 2, \dots C\}$ 番目の値はそれぞれ c 番目のカテゴリに相当し, $C+1$ 番目の値は物体ではない領域 (背景) に対応する値である. 推論時には, softmax 関数によって正規化され各カテゴリの確率として利用される.

前述したように SSD300 では anchor box の数 (K) は 8732 個であるが, これは次の式で決定される.

$$K = \sum_{i=1}^{6} k_i g_i^2 \tag{3.1}$$

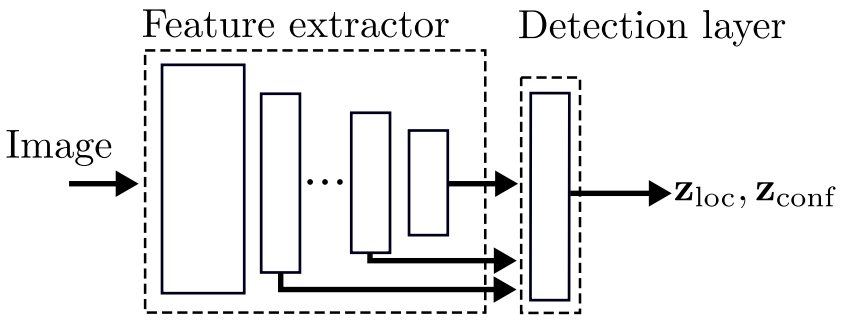
ここで k_i は i 番目の特徴マップにおいて anchor box が k_i 種類あるということを示す値である. ここでの種類は大きさおよびアスペクト比の組み合わせで定義される (位置は考慮されない). たとえば最初の特徴マップである conv4_3 (norm4) では $30 \times 30, 15 \times 60, 60 \times 15, 42 \times 42$ の 4 種類の anchor box が定義されており, $k_1 = 4$ である. なお, ここでの anchor box の大きさは入力画像サイズ 300×300 における大きさを示している. 以降の特徴マップについても同様に anchor box が定義されており, $k_2 = 6, k_3 = 6, k_4 = 6, k_5 = 4, k_6 = 4$ となっている. これの値と前述の $g_1 = 38, g_2 = 19, g_3 = 10, g_4 = 5, g_5 = 3, g_6 = 1$ を式 (3.1) に代入することで, $K = 8732$ を得る.

SSD300-fork では $C(= 4)$ 個の anchor set を持つ. SSD300 において 1 つの検出層が 1 つの anchor set に対応しており, したがって SSD300-fork は一つの共通のマルチスケール特徴抽出器と C 個の検出層で構成されることになる (図 3.7b). ここでマルチスケール特徴抽出器は SSD300 と同じものであり, 抽出された特徴マップに C 個の検出層を並列して適用することになる. c 番目の検出層は c 番目のカテゴリを担当し, 2 つの配列 $\mathbf{z}^c_{\mathrm{loc}} \in \mathbb{R}^{K \times 4}$ および $\mathbf{z}^c_{\mathrm{conf}} \in \mathbb{R}^K$ を計算する. $\mathbf{z}^c_{\mathrm{loc}}$ は SSD300 と同様に各 anchor box における矩形推定に対応しており, K 個の 4 次元ベクトル $(\Delta x_{center}, \Delta y_{center}, \Delta width, \Delta height)$ から構成されている. $\mathbf{z}^c_{\mathrm{conf}}$ は分類問題に対応した配列だが, 3.3 節で述べたように fork モデルにおいては多値分類は必要ないため, 各 anchor box につき 1 つのスカラー値を持っている. この値は推論時には sigmoid 関数によって該当する領域の物体らしさとして用いられる.

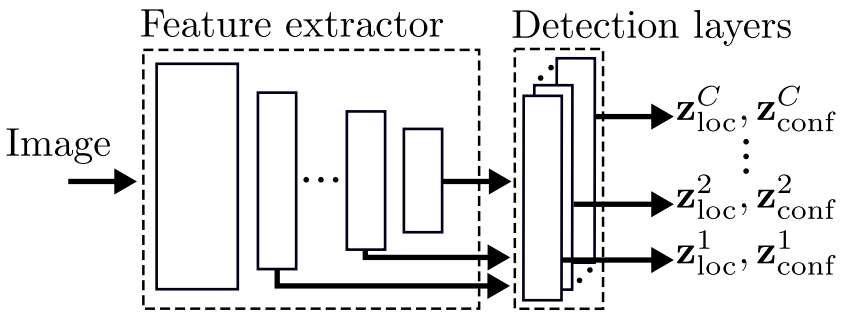
また SSD300-fork は C 個の検出層を持つため, 学習時には各検出層で計算される損失関数を統合し, 共通の特徴抽出器に伝搬させる必要がある. SSD300-fork では次式で定義される重み付き損失関数 L を利用して学習を行う.

$$L\left(\mathbf{z}^1_{\mathrm{loc}}, \mathbf{z}^1_{\mathrm{conf}}, \dots, \mathbf{z}^C_{\mathrm{loc}}, \mathbf{z}^C_{\mathrm{conf}}\right) = \sum_{c=1}^{C} w_c \frac{L_{\mathrm{loc}}\left(\mathbf{z}^c_{\mathrm{loc}}\right) + L_{\mathrm{conf}}\left(\mathbf{z}^c_{\mathrm{conf}}\right)}{N^c_+}, \tag{3.2}$$

ここで L_{loc} は矩形推定問題に対応する損失関数であり SSD300 で用いられているものと同じものである. この関数は各 anchor box について推定された矩形とその anchor box に割り当てられた ground truth の矩形との差分を huber 関数を用いて計算する. このとき ground truth が割り当てられなかった anchor box については計算に含めず, 勾配も返さない. L_{conf} はカテゴリ分類問題に対応する損失関数である. この関数は各 anchor について ground truth が割り当てられて



(a) SSD300



(b) SSD300-fork

図 3.7: ネットワークの構造．(a) SSD300 はマルチスケール特徴抽出器 (feature extractor) と 1 個の検出層 (detection layer) から構成されている．検出層は出力された特徴マップをもとに $\mathbf{z}_{\mathrm{loc}} \in \mathbb{R}^{K\times 4}$ および $\mathbf{z}_{\mathrm{conf}} \in \mathbb{R}^{K\times(C+1)}$ を計算する．ここで K は anchor box の数 ($K = 8732$) であり C はカテゴリの数 ($C = 4$) である．(b) SSD300-fork は共通のマルチスケール特徴抽出器と C 個の検出層を持つ．c 番目の検出層は c 番目のカテゴリを担当し, $\mathbf{z}^c_{\mathrm{loc}} \in \mathbb{R}^{K\times 4}$ および $\mathbf{z}^c_{\mathrm{conf}} \in \mathbb{R}^K$ を計算する．

いるか (=物体領域かどうか) を元に sigmoid cross entropy を計算する．また Liu ら [7] が学習を安定化させるために導入した, hard-negative mining についても同様に適用する．これは ground truth に割り当てられている (positive) な anchor box の数 N^c_+ に比べて, 割り当てられていない (negative) な anchor box の数 N^c_- が多くなる問題に対応するために, 計算に含める negative な anchor box の数 N^c_- を $N^c_-/N^c_+ \leq k$ となるように抑える手法である．Negative な anchor box が多すぎる場合, 損失関数が大きくなる (hard-negative) 方から anchor box を kN^c_+ 個選ぶ．損失関数が小さい negative な anchor box は L_{conf} の計算には含めず, 勾配も返さない．w_c は c 番目の検出層の重みであり, カテゴリ間の難易度のバランスを取るための係数である．実験の結果, $(w_1, w_2, w_3, w_4) = (0.2, 0.2, 0.4, 0.2)$ で最も良い性能を示した．ここで w_1, w_2, w_3 および w_4 はそれぞれコマ, テキスト, 顔および全身の検出層の重みである．

第 4 章

実験

提案手法である SSD300-fork を Manga109 [4, 8] および eBDtheque [2] で評価した.

4.1 Manga109 を用いた実験

4.1.1 既存手法との比較

本節では Manga109 を利用して SSD300-fork および既存の自然画像における物体検出手法の学習および評価を行う. 実験にあたっては全てのページを見開きページとして扱った. ここで見開きページとは左右のページを結合して一枚のページとして扱うことであり, 大きなシーンを描くために用いられる (図 4.1). このときコマやキャラクターが左右のページにまたがって描かれることがある. 藤本ら [8] はこれらの物体を適切に扱うために全てのページを見開きページとみなして, アノテーションを行っている. 本実験でもそれに倣い, 全てのページを見開きページとして扱い学習および評価を行った.

また表紙や目次などの通常のページと異なるページを除外した. これらのページは通常のページとは形式が大きく異なっていることがあり, 学習や評価に不適切であると考えられるためである. 見開きページの結合および例外的なページの除去を行った後, 学習 · テストセットの分割を行った. Manga109 に含は 109 冊の漫画が含まれるが, そのうち 99 冊を学習セット, 10 冊をテストセットとして利用した. 各セットに含まれるアノテーション数を表 4.1 に示す.

既存手法としては Faster R-CNN, SSD300 および YOLOv2 を利用した. 学習 · テストにあたっては実験の条件を揃えるために, 各手法が PASCAL VOC での実験で利用している条件をそのまま利用し, 漫画特有の調整は一切行っていない. 各手法の実験条件を次に示す.

Faster R-CNN

Ren ら [9] が PASCAL VOC での実験で利用している学習条件を利用した. モデルの初

*13 “犯罪交渉人 峰岸英太郎” ©記伊 孝
*14 “ドールガン” ©出口 竜正
*15 “平成爺メン” ©やまだ 浩一

(a) [*13] (b) [*14]

図 4.1: 見開きページ．これらのページでは左右のページを結合することで 1 つの大きな画像を表現している．(a) では中央のキャラクターが左右のページにまたがって描かれている．(b) では右ページのコマが左ページにはみ出している．

(a) 通常のページ [*15] (b) 表紙 [*15] (c) 目次 [*15]

図 4.2: 通常のページ (a) と除外するページ (b, c)．通常のページは複数のコマから構成されており, キャラクターやフキダシがコマの内部に配置されている (a)．また日本の漫画では一般的にテキストは縦書きで書かれる．除外するページ (b, c) ではコマがなく, 物体はページ全体に直接配置されている．またこれらのページでは横書きのテキストが用いられることもある．

期化は ImageNet [22] で学習された VGG-16 [23] の重みで行い, data augmentation として random flip を行った．学習時/テスト時ともに画像から平均輝度値を減じるという正規化の処理が行われるが, この平均輝度値には ImageNet の平均を利用した．ミニバッチサイズは 1 に設定し, 全部で 70000 iteration の学習を行った．学習率は 10^{-3} から開始し,

表 4.1: Manga109 の学習 · テスト分割.

			アノテーション数			
	冊数	ページ数	コマ	テキスト	顔	全身
学習	99	9,250 †	94,746	135,376	108,353	144,448
テスト	10	880 †	9,154	12,542	10,362	12,704
合計	109	10,130 †	103,900	147,918	118,715	157,152

† 見開きページ.

50000 iteration で 10^{-4} へと変更した. 実装は ChainerCV [24] の提供する再現実装を利用した.

SSD300

Liu ら [7] が PASCAL VOC での実験で利用している学習条件を利用した. モデルの初期化は ImageNet で学習された VGG-16 の重みで行い, data augmentation として color distortion, random expansion, random crop, resize および random flip を行った. Faster R-CNN と同様に平均輝度値を減じるという正規化には ImageNet の平均を利用した. また random expansion には画像の背景を平均輝度値で埋めるという処理が含まれているが, この平均輝度値にも ImageNet の平均を利用した. ミニバッチサイズは 32 に設定し, 全部で 120000 iteration (=3,791 epochs) の学習を行った. 学習率は 10^{-3} から開始し, 80000 iteration で 10^{-4} へ, 100000 iteration で 10^{-5} へと変更した. 実装は ChainerCV [24] の提供する再現実装を利用した.

YOLOv2

Redmon ら [18] が PASCAL VOC での実験で利用している学習条件を利用した. モデルの初期化は ImageNet で学習された Darknet19 の重みで行った. ミニバッチサイズは 64 に設定し, 全部で 45000 iteration の学習を行った. 学習率は 10^{-4} から開始し, 100 iteration で 10^{-3} (warming up [25]), 25000 iteration で 10^{-4}, 35000 iteration で 10^{-5} へと変更した. 実装は Redmon らによって公開されている Darknet [26] による実装を利用した.

SSD300-fork

SSD300 と同様に Liu ら [7] が PASCAL VOC2007+2012 の実験で利用している初期化, data augmentatin, 学習率スケジュールを利用した. 通常の SSD300 に比べて検出層が複製されパラメータ数が 4 倍になっているため, 検出層だけ学習率を 4 倍にするという調整を行った. 実装は Chainer [27] および ChainerCV を利用して行った.

表 4.2 に各手法の average precision (AP) およびその平均である mean average precision (mAP) を示す. 評価にあたっては PASCAL VOC と同じ IoU ≥ 0.5 をしきい値として用いた. これは自然画像の物体検出において標準的なしきい値である. 表 4.2 からわかるように SSD300-fork

表 4.2: Manga109 による比較.

		各カテゴリの AP			
手法	mAP	コマ	テキスト	顔	全身
Faster R-CNN [9]	49.9	96.1	23.8	15.7	63.9
SSD300 [7]	81.3	**97.1**	82.0	67.1	79.1
YOLOv2 [18]	59.7	90.2	64.6	37.1	46.9
SSD300-fork	**84.2**	96.9	**84.1**	**76.2**	**79.6**

は他の手法に比べて優れた検出性能を示した. 特に顔の検出においてはベースとなった SSD300 に比べて 9 % の精度向上を達成した.

図 4.3 に SSD300-fork によって重なりが大きい物体が適切に検出された例を示す. これらの例ではコマおよび全身が大きく重なっている. SSD300 では全身の検出に失敗しているが, SSD300-fork ではコマと全身の両方が検出できている.

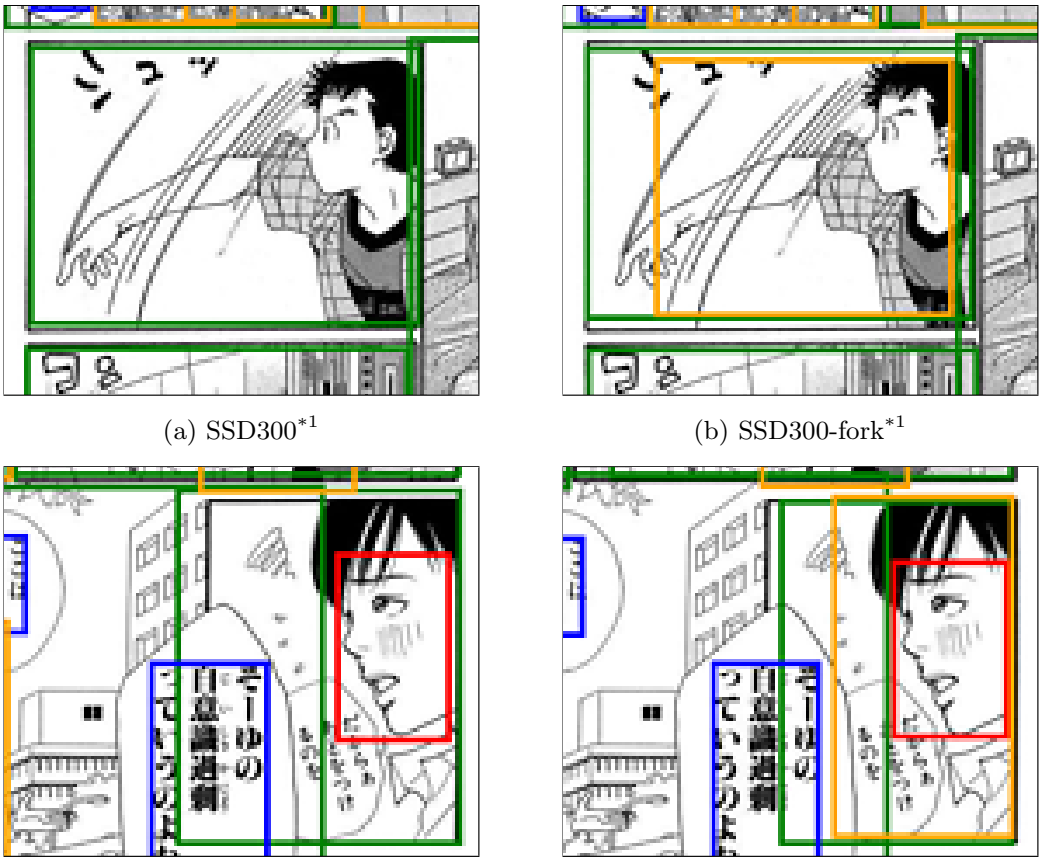
4.1.2 漫画ごとの分析

本節では漫画ごとにおける検出精度の違いについて分析を行う. 表 4.3 に SSD300-fork の各漫画における性能を示す. 表中で * の印が付いている漫画は同じ著者の作品が学習セットの中にも含まれているものである. 同じ著者の漫画はその画風も類似していると考えられるため, これらの漫画の画風はモデルの学習時に利用されており, それ以外の漫画については学習に利用されていない新規の画風であると言える. 表 4.3 から分かるように * の有無と性能の明確な関係は見られず, 新規の漫画についても同程度の性能を示せるロバストな検出器が得られたと言える.

テストセットに含まれる漫画 10 冊の全てでコマの検出が最も高い性能を示した. 特に “幼稚園ぼうえい組” ではコマの検出率が 100 % を達成した. この漫画は 4 コマ漫画と呼ばれるジャンルの漫画であり, 同じ大きさのコマが縦に 4 つ並べられているという特徴がある (図 4.4). このような単純な構造を持っているため, コマの検出が簡単になったと考えられる.

“やまとの羽根” は全身の検出精度が 10 冊の中で最も高くなっている. この漫画はスポーツ漫画であり, 動きを描写するために身体を大きく描いたコマが多用されている (図 4.5a). 一方で “花影戦記 妖魔降臨” は全身の検出結果が最も低い. この漫画では小さいキャラクターが詰め込まれたコマが多用されている. たとえば図 4.5b の上部および左下のコマでは多くのキャラクターが描かれているが, その全ての検出に失敗している.

顔の検出が最も難しい漫画は “アンバランス・トーキョー” であった. たとえば図 4.6 では 25 個の顔領域が存在するが, そのうちの 4 個しか検出されていない. このページでは写実的な表現のために, 通常の漫画に比べて顔の部位が小さく描かれており, そのことが検出に悪影響を及ぼしたと考えられる.



(a) SSD300[*1]　(b) SSD300-fork[*1]

(c) SSD300[*16]　(d) SSD300-fork[*16]

図 4.3: SSD300 (a, c) と SSD300-fork (b, d) の比較．これらのページではコマおよび全身が大きく重なっている．SSD300 では全身の検出に失敗しているが, SSD300-fork ではコマと全身の両方が検出できている．

4.2　eBDtheque を用いた実験

4.2.1　既存手法との比較

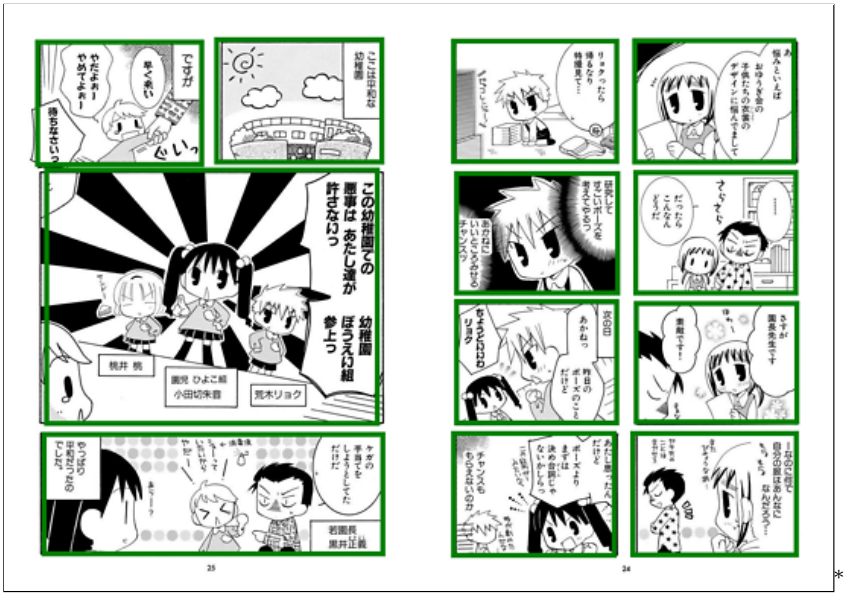
本節では eBDtheque [2] を用いた検出の実験を行う．既存の漫画物体検出の手法の多くがこのデータセットでの評価値を報告しているため, eBDtheque に対して SSD300-fork を適用することで, それらの手法との比較が可能になる．2.1 節で述べたように eBDtheque に含まれる画像は 100 ページと少ないため, eBDtheque はテストのみに利用し, モデルは Manga109 で学習済みのもの

*16 “雪の降る街” ©山田 雨月
*17 “幼稚園ぼうえい組” ©てんや
*18 “花影戦記 妖魔降臨” ©島崎 譲, 鷹 司
*19 “アンバランス・トーキョー” ©内田 美奈子

表 4.3: SSD300-fork の漫画ごとの性能.

				各カテゴリの AP			
漫画	ページ数	ジャンル	mAP	コマ	テキスト	顔	全身
うるとら☆イレブン	108	スポーツ	84.5	95.2	88.4	79.6	75.0
アンバランス・トーキョー	82	SF	79.8	98.6	83.3	62.8	74.4
ワレワレハ、オニデアル	91	ラブコメ	83.2	94.0	84.9	71.1	82.8
やまとの羽根	109	スポーツ	92.0	98.6	85.7	92.8	90.9
やさしい悪魔	89	ファンタジー	89.3	97.8	93.5	80.6	85.2
幼稚園ぼうえい組	26	4 コマ	83.4	100.0	73.6	77.6	82.5
花影戦記 妖魔降臨 *	101	ファンタジー	83.6	99.2	91.8	74.9	68.4
雪の降る街 *	93	恋愛	77.1	95.4	71.2	63.5	78.4
ゆめのかよいじ *	96	ファンタジー	84.4	94.2	77.9	81.4	84.2
ゆめ色クッキング	85	恋愛	89.0	97.7	87.0	82.4	89.1
合計	880	-	84.2	96.9	84.4	76.2	79.6

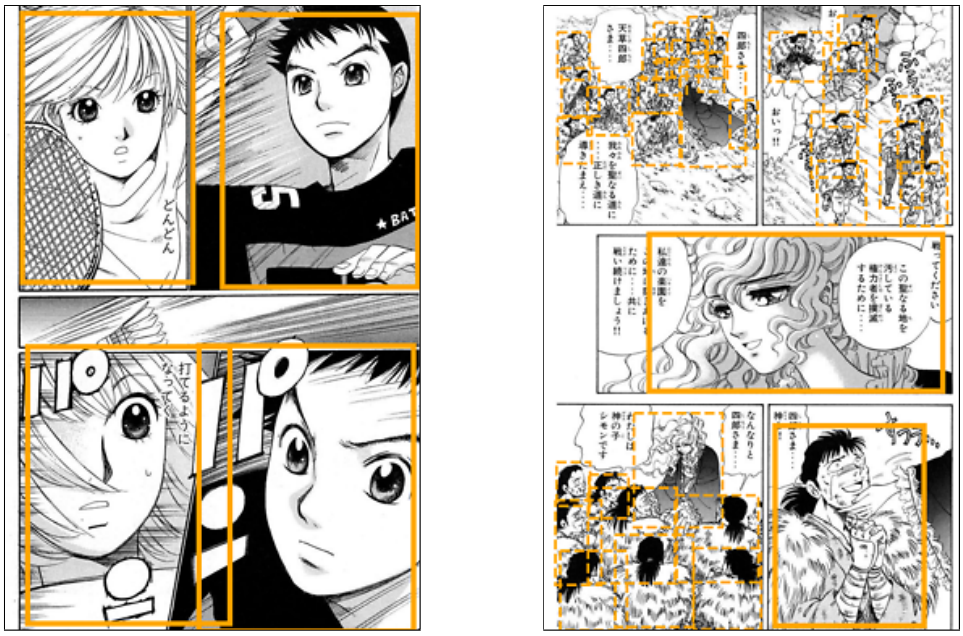
* 学習セットに同じ作者の漫画を含む.



*17

図 4.4: 4 コマ漫画におけるコマの検出. 全てのコマが正しく検出されている.

を利用した. また検出するカテゴリとしてはコマおよび全身のみとした. これは eBDtheque において提供されているアノテーションのうち本論文で対象とするものがこの 2 種類のみであるからである. たとえば eBDthque ではフキダシおよびテキスト行のアノテーションを提供しているが, こ



(a) 簡単な例 *1　　(b) 難しい例 *18

図 4.5: 全身の検出結果. 検出されなかった全身を破線で示す.

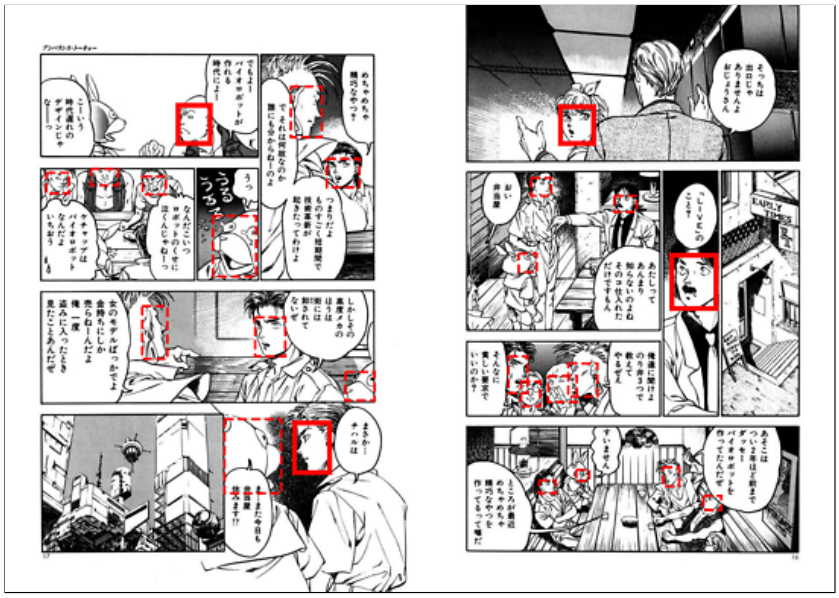


図 4.6: 顔検出*19. このページ (見開きページ) に 25 個の顔がアノテーションされているが, そのうちの 4 個しか検出されていない. 検出されなかった顔を破線で示す.

れらの定義は本論文が対象とするテキストとは異なっている.

表 4.4 に SSD300-fork および既存手法の評価値を示す. 既存手法の評価値については各手法の実装が公開されていなかったため, Rigaud ら [28] の論文の報告値を引用した. また通常の物体検

表 4.4: eBDtheque を用いた既存の漫画物体検出手法との比較．コマおよび全身について recall (R), precision (P) および F 値 (F) の値を示す．

	コマ			全身		
手法	R	P	F	R	P	F
Arai ら [29]	58.0	75.3	65.6	-	-	-
Rigaud ら [30]	78.0	73.2	75.5	-	-	-
Rigaud ら [28]	**81.2**	**86.6**	**83.8**	21.6	40.5	28.2
SSD300-fork	73.3	76.4	74.8	**42.2**	**58.0**	**48.8**

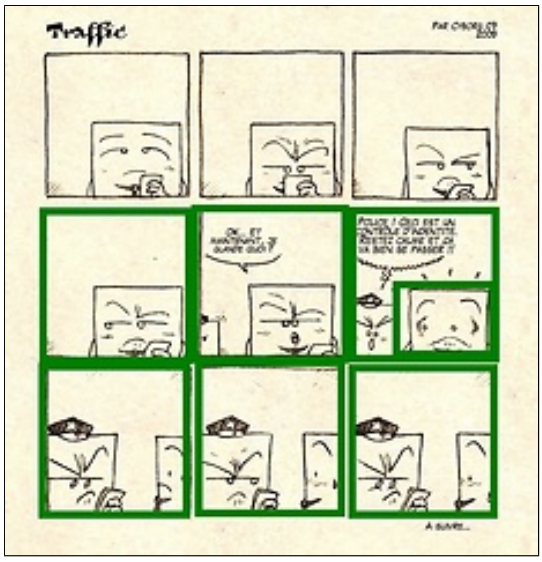
出では mAP を評価に用いることが多いが, 既存の手法が recall, precision および F 値による評価を行っていたため, 本比較ではそれに倣っている．これらの評価値の計算には確信度に対するしきい値処理が必要であるが, 提案手法の評価値については F 値が最大となるように設定した．

コマの検出については Rigaud ら [28] の方がより良い性能を示した．これは日本の漫画でしか学習していない SSD300-fork がフランスやアメリカの漫画に対して適応できていないことが原因であると考えられる．たとえば, 日本の漫画においてコマの枠線は直線で描かれるのが基本であるが, 図 4.7c に示すページのように eBDtheque にはフリーハンドで描かれたと思われる枠線が曲がっているコマが見られる．このページは日本の 4 コマ漫画のように単純なコマの配置をしており検出が容易に思われるが, SSD300-fork では上段の 3 コマが検出されていない (図 4.7a)．この原因が枠線のスタイルによるものであることを確認するために, これらのコマについて直線化の処理を行い, 再度検出を試みた．簡単のために直線化は画像編集ソフトウェアを用いて手動で行った．まず各コマの枠線を背景部分のテクスチャで塗り潰し (図 4.7d)．次に元のコマの四隅に重なるように直線で枠を描いた (図 4.7e)．この加工を検出に失敗した 3 コマについて行った上で, 再度 SSD300-fork による検出を適用したところ図 4.7b のように全てのコマが検出された．このことから枠線のスタイルがコマの検出性能に影響すること, および Manga109 で学習したモデルは日本のスタイルに過学習していることがわかる．

一方, 全身については SSD300-fork が既存手法に対して全ての評価値で約 20 % の性能向上を示した．これは Manga109 に含まれるキャラクターが高い多様性を持っていたためだと考えられる．

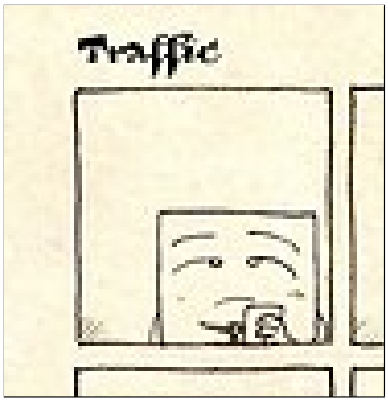
4.2.2 テキストの検出

本節では eBDtheque を用いてテキストの検出の評価を行った．4.2.1 節で述べたように, eBDtheque で提供されているフキダシおよびテキスト行の定義は本論文におけるテキストの定義とは異なり, そのまま評価に利用するのは不適切である (図 4.8a)．そこで本実験では簡単な前処理を行い, 擬似的なテキストアノテーションを生成した．この前処理では同じフキダシに含まれているテキスト行を全て結合し, 1 つの大きなテキストとして扱う (図 4.8b)．フキダシに含まれ

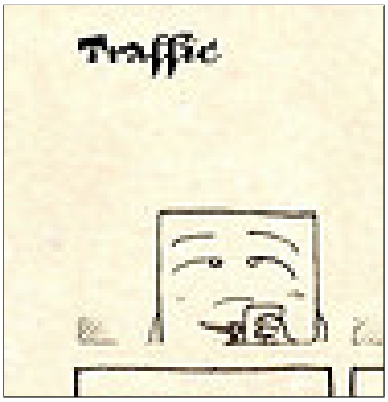


(a) 検出結果

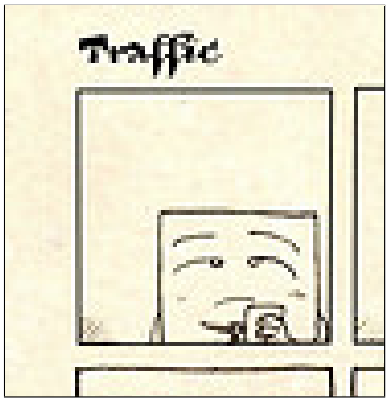
(b) 枠線の直線化を行った場合の検出結果



(c) 枠線が曲がっているコマ



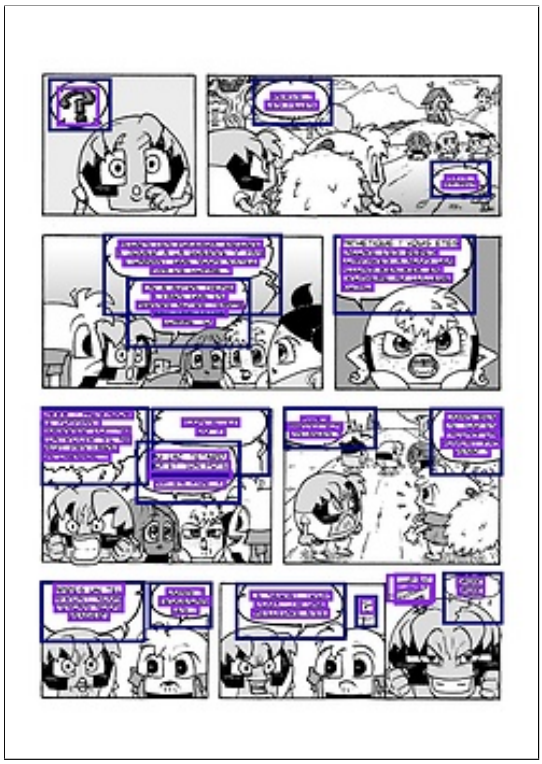
(d) 枠線の除去



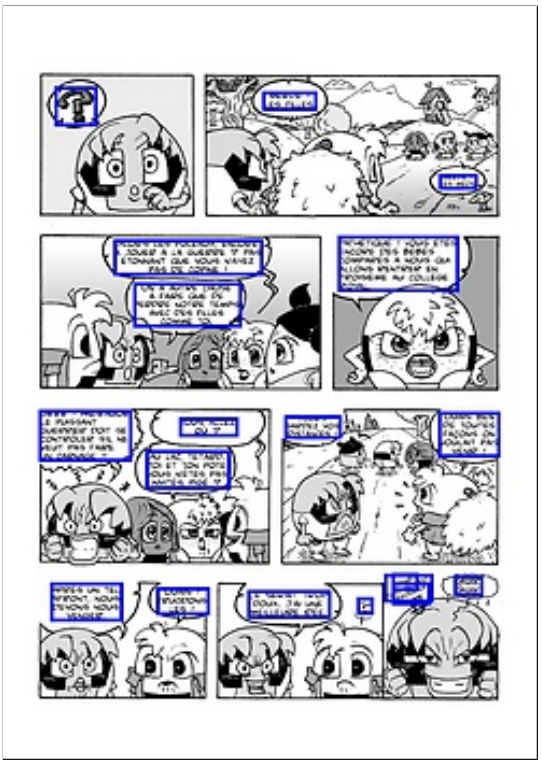
(e) 直線化されたコマ

図 4.7: eBDtheque におけるコマ検出．(a) のように単純な形状および配置のコマでも検出されない場合がある．これは (c) のようにコマの枠線が曲がっていることが原因であると考えられる．枠線を手動で一旦除去し (d), 直線で枠線を描き直す (e) 処理を行ったところ (b) のように全てのコマが検出された．

ないテキスト行についてはそのまま 1 つのテキストとして利用する．実験では 4.2.1 節と同様に, Manga109 で学習された SSD300-fork を適用した．またテキスト特有の工夫として画像を 90 度回転させた上で検出を行うという前処理 (回転) を試した．これは Manga109 においてテキストは縦書きが主流であるのに対し, eBDtheque のテキストは横書きが多いという差を埋めるための処理である．結果を表 4.5 に示す．表 4.5 からわかるように, 回転処理を加えることによって検出性能が F 値で 15 % 以上向上している．



(a) eBDtheque におけるフキダシ (紺) およびテキスト行 (紫) のアノテーション.



(b) 統合処理による擬似的なテキストのアノテーション (青).

図 4.8: eBDtheque におけるテキスト. eBDtheque では (a) のようにフキダシおよびテキスト行のアノテーションしか提供されていない. Manga109 で学習した検出器を適用するために同じフキダシに含まれるテキストを統合することで擬似的なテキストのアノテーションを生成した (b).

表 4.5: eBDtheque でのテキストの検出結果.

手法	R (%)	P (%)	F (%)
SSD300-fork	29.0	59.5	39.0
SSD300-fork (回転あり)	46.8	68.4	55.6

第 5 章

結論

5.1 まとめ

本論文では漫画における物体検出という課題に着目した．自然画像での物体検出で主流となっている深層学習による手法は漫画のような物体の重なりが大きい画像では割り当て問題により性能が劣化する．この問題を解決するために SSD300-fork という新しいネットワーク構造を提案した．このネットワークでは検出層を複製し, 各検出層がそれぞれ 1 つのカテゴリを担当することで, 割り当て問題を解決する．

Manga109 を用いた比較実験の結果, SSD300-fork が既存の物体検出手法よりも優れた検出性能を示すことがわかった．ベースとした手法である SSD300 と比較した場合, mAP で 3 % の改善が見られた．特に顔の検出においては AP が SSD300 に対して 9 % 向上した．

また eBDtheque をテストに利用して, 既存の漫画物体検出の手法との比較を行った．コマについては state-of-the-art の手法 [28] の F 値 86 % に対して SSD300-fork は 75 % の検出性能を示した．また全身の検出については既存の手法に対して約 20 % の性能向上を達成した．コマおよびテキストについては日本の漫画とフランスの漫画の違いによる性能の低下がみられた．

5.2 今後の展望

漫画物体検出の精度向上に向けたいくつかの改善案を挙げる．

物体の境界を意識した矩形改善

本論文で利用した深層学習による物体検出手法は, 矩形の位置を数値として推定するため, 物体の境界に一致しにくいという問題がある．物体の境界を適切に推定できないという欠点は画像加工をするにあたっては問題となる．漫画の物体, 特にコマは枠線という形で明確な境界を持っていることが多いため, 検出器による検出の後処理として, 枠線を利用した矩形の改善を加えることが考えられる．

より多様な漫画画像に向けた学習

4.2 節で述べたように, 日本の漫画 (Manga109) で学習を行ったモデルはフランスの漫画

(eBDtheque) で十分に性能を発揮できなかった. 単純な方法としては, 多様な漫画を用いて学習を行うという手法が挙げられるが, 大規模なアノテーションデータセットを作成する必要があるという問題がある. 別の方法として, data augmentation を工夫することで, より汎用的な検出器を学習するという手法が考えられる. たとえば 4.2.1 節で述べたコマの枠線が曲がっていると検出性能が下がるという問題については, 学習時にコマを多少曲げた画像を生成して用いるなどである.

コンテキストを利用した性能改善

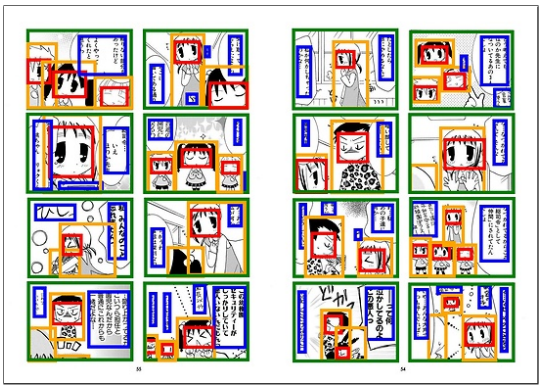
本論文では漫画を単純な画像の集合として捉えて, 通常の物体検出と同じ枠組みで検出を行った. しかし, 漫画はストーリーを表現するための連続した画像群であり, 個々のページには意味的な関連性がある. そのような情報を活用することで性能向上が期待できる.

第 6 章

付録: 検出結果

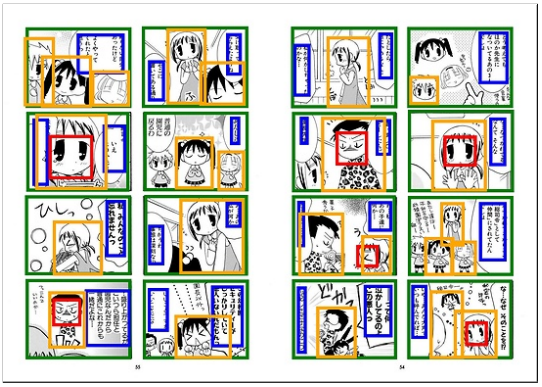
本章では 4 章での実験における検出結果を示す．図 6.1–6.3 は 4.1 節で行った, Manga109 での検出結果である．実験に利用した Faster R-CNN, SSD300, YOLOv2 および提案手法である SSD300-fork での検出結果を示す．

図 6.4, 6.5 は 4.2 節で行った, eBDtheque での検出結果である．図 6.4 は 4.2.1 節におけるコマおよび全身の検出結果であり, 図 6.5 は 4.2.2 節におけるテキストの検出結果である．



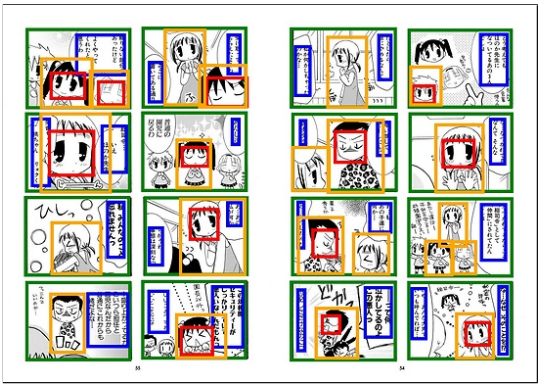
(a) Ground truth

(b) Faster R-CNN



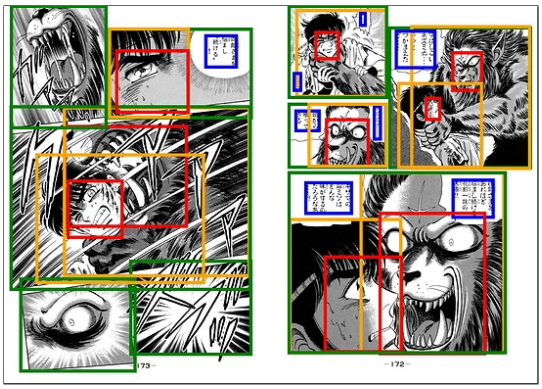
(c) SSD300

(d) YOLOv2

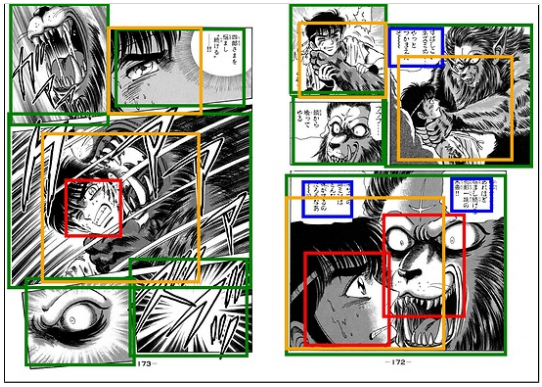


(e) SSD300-fork

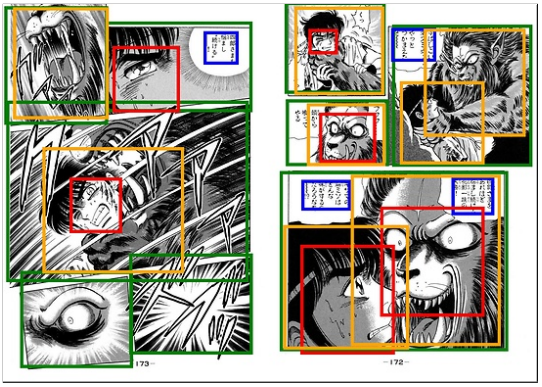
図 6.1: Manga109 での検出結果[*17] (1/3)



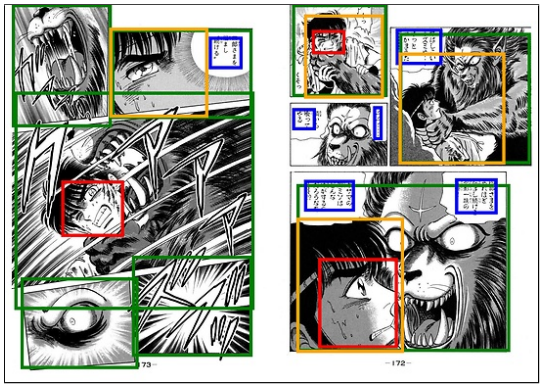
(a) Ground truth



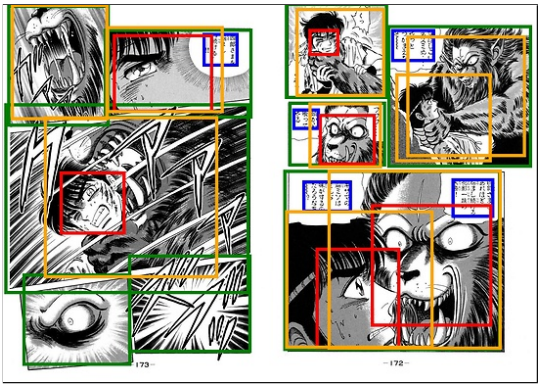
(b) Faster R-CNN



(c) SSD300

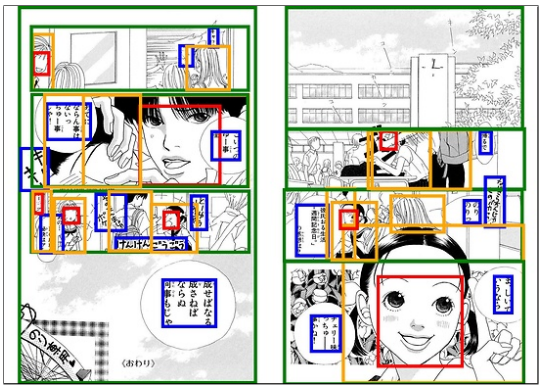


(d) YOLOv2

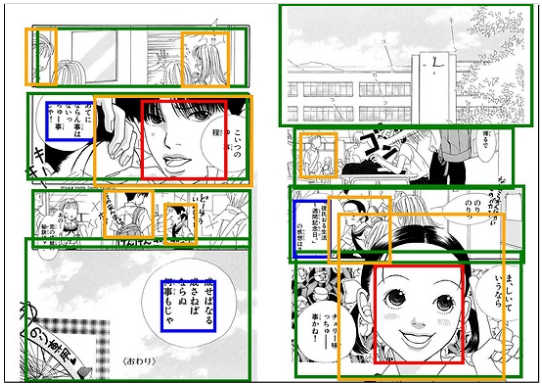


(e) SSD300-fork

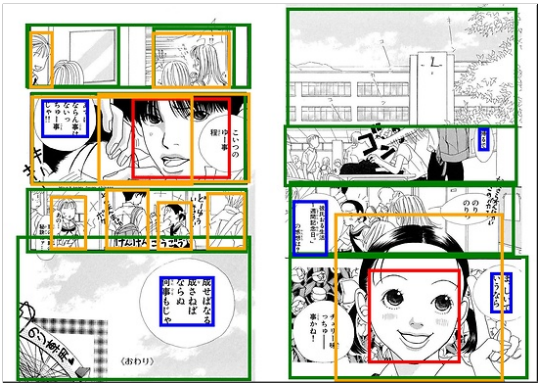
図 6.2: Manga109 での検出結果[*18] (2/3)



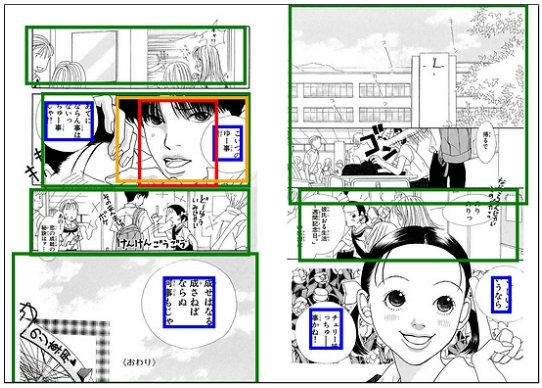
(a) Ground truth



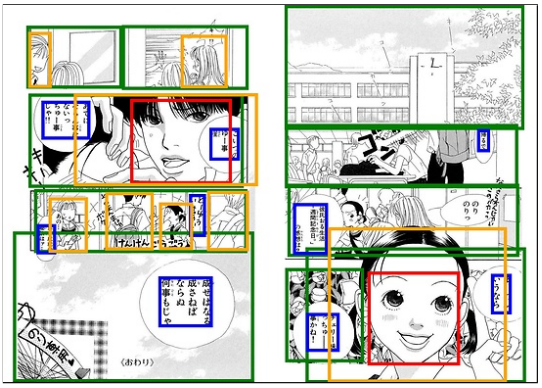
(b) Faster R-CNN



(c) SSD300

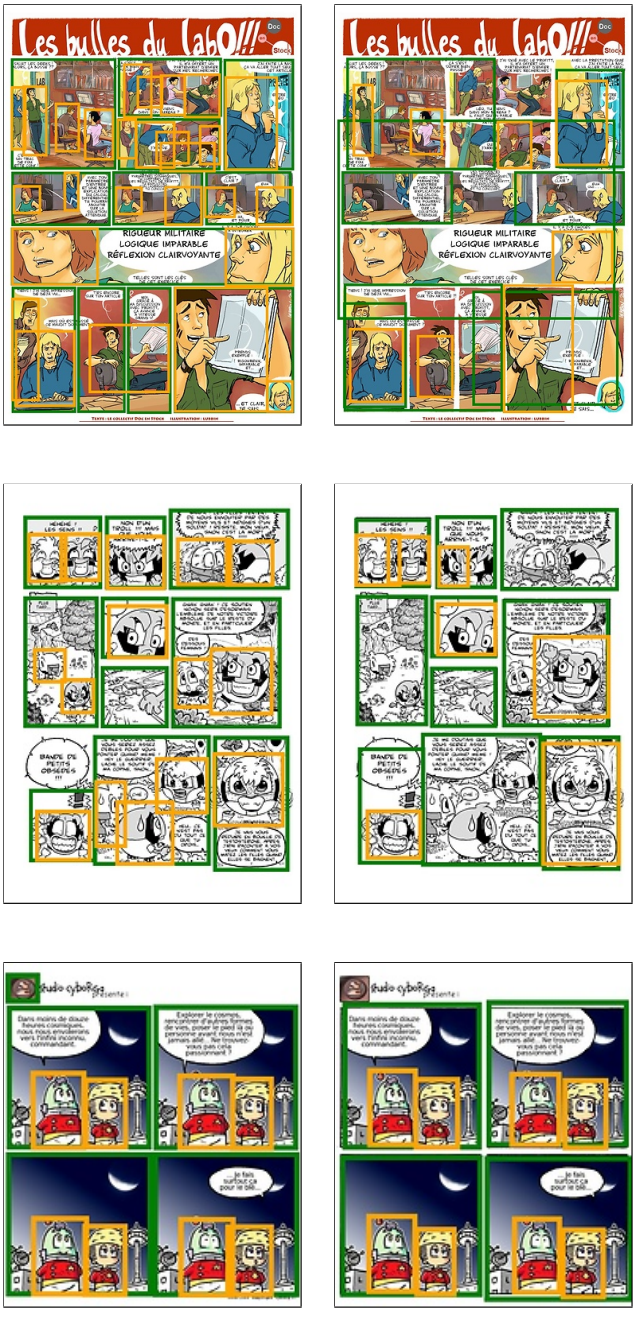


(d) YOLOv2



(e) SSD300-fork

図 6.3: Manga109 での検出結果[*16] (3/3)



Ground truth SSD300-fork

図 6.4: eBDtheque でのコマおよび全身検出の結果 (コマ: 緑, 全身: 橙)

Ground truth (疑似)　　SSD300-fork　　SSD300-fork (回転あり)

図 6.5: eBDtheque でのテキスト検出の結果

第 7 章

付録: 提案手法の拡張

本章では SSD300-fork の拡張として SSD300-x4, SSD512-fork, 階層的 SSD300-fork の 3 つを検討する.

7.1 手法

7.1.1 SSD300-x4

このモデルでは anchor set を複製する際に検出層だけではなく, 特徴抽出器もカテゴリごとに独立して持つ (図 7.1). カテゴリ間で一切のパラメータを共有しないため, 1 カテゴリの検出器を 4 個並列に利用するものと等価になる.

- 利点: 特徴抽出層を複製することにより, 各カテゴリに特化した特徴を獲得できる可能性がある.
- 欠点: SSD300 や SSD300-fork に比べてパラメータや計算時間が増大する. パラメータ数は SSD300 で 24.1 M, SSD300-fork で 25.6 M であるが SSD300-x4 では 96.0 M まで増大する. また学習およびテストには約 4 倍の時間が必要となる.

7.1.2 SSD512-fork

このモデルではベースのモデルを SSD300 から SSD512 に変更する. SSD512 は SSD300 と同様の検出手法であり, 入力画像の大きさが 512×512 に拡大されている.

- 利点: 入力画像を高解像度にすることにより, 高精度な検出が期待できる.
- 欠点: 大きな特徴マップを扱うため, 学習およびテストでの消費メモリおよび計算時間が増大する.

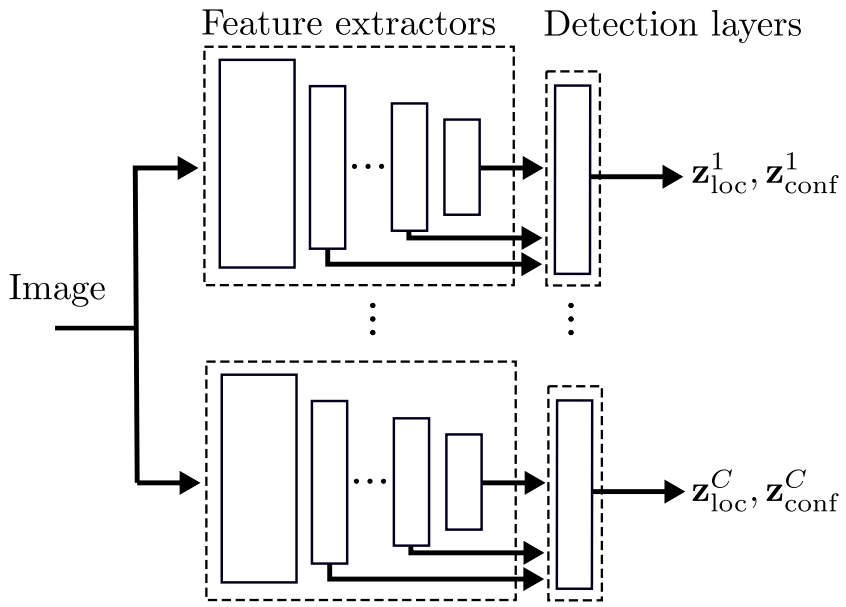


図 7.1: SSD300-x4 の構造．特徴抽出器から分岐させる．1 カテゴリの検出器を 4 個並列に使用するものと等価．

7.1.3 階層的 SSD300-fork

このモデルは漫画において多くの物体はコマの中に描かれるという性質に着目する．検出器をコマごとに再帰的に適用することで精度の向上を図る (図 7.2).

- 利点: テスト時の工夫であり, 既に学習済のモデルを転用できる．学習時のコストが増大しない．
- 欠点: コマ数に応じてテスト時の計算量が増大する．Manga109 では平均して 1 ページにつき 10 コマが含まれるため, ネットワークの計算で約 11 倍の計算量が必要となる (実際には NMS の計算量の増大により, 全体としてはさらに遅くなる).

7.2 実験

前節で挙げた SSD300-x4, SSD512-fork, 階層的 SSD300-fork の 3 つを比較する．また参考として SSD512 単体の性能も示す．SSD300-x4, SSD512, SSD512-fork の学習は SSD300, SSD300-fork と同様の条件で行った．ただし SSD300-x4 については学習を安定させるために最初の 1000 iteration だけ学習率を 10^{-4} して学習を行った (warming up [25]). また階層的 SSD300-fork のしきい値は 0.6 とした．これは Liu ら [7] が SSD300 の可視化に用いているしきい値である．表 7.1 に各手法での mAP およびカテゴリごとの AP を示す．また図 7.3–7.5 に検出結果の例を示す．全てのカテゴリにおいて SSD512-fork が最も高い性能を示した．また SSD512 と SSD512-fork の比

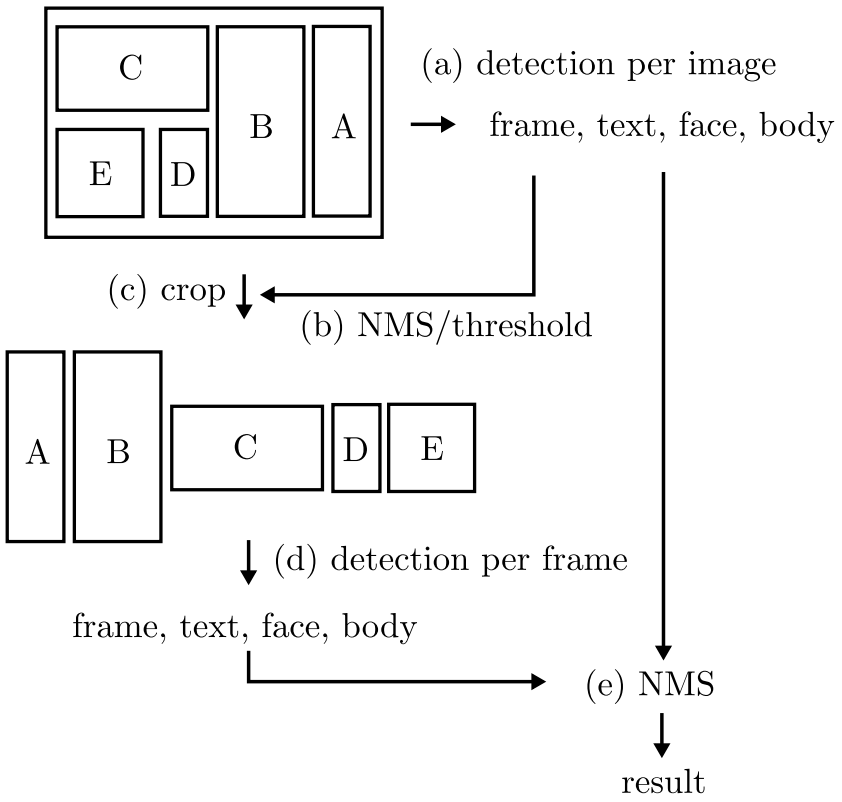
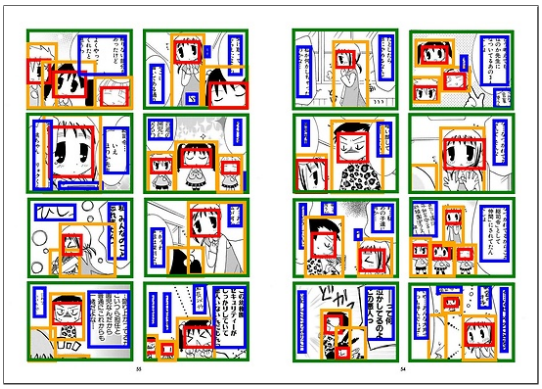


図 7.2: 階層的 SSD300-fork. まず入力画像全体に検出器を適用する (a). 検出された物体のうち充分な確信度でコマであるものを選別する (b). 選別されたコマの情報をもとに入力画像をコマ領域ごとに分割する (c). 分割された領域ごとに検出器を再度適用する (d). 画像全体で検出された物体およびコマごとに検出された物体を統合し, 最終的な検出結果とする (e).

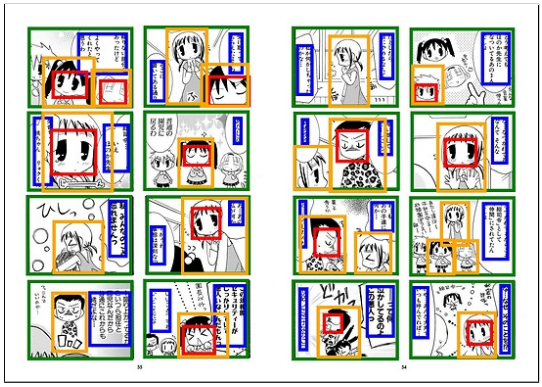
表 7.1: 拡張手法の比較.

		各カテゴリの AP			
手法	mAP	コマ	テキスト	顔	全身
SSD300 [7]	81.3	97.1	82.0	67.1	79.1
SSD300-fork	84.2	96.9	84.1	76.2	79.6
SSD300-x4	86.5	97.9	88.4	77.9	81.9
SSD512 [7]	89.3	97.9	89.5	83.0	86.7
SSD512-fork	**90.9**	**98.0**	**90.6**	**88.0**	**86.8**
階層的 SSD300-fork	87.2	96.9	84.4	85.9	81.7

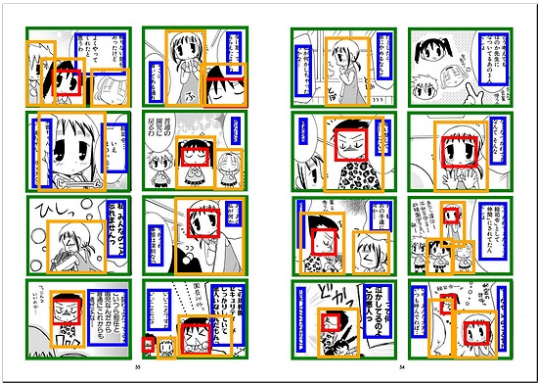
較からわかるように, ベースが SSD512 であっても fork モデルは有効であることがわかる.



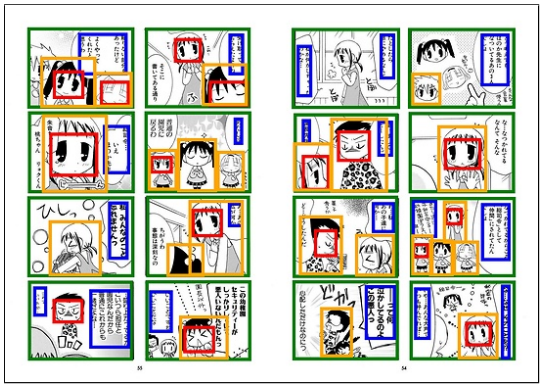
(a) Ground truth



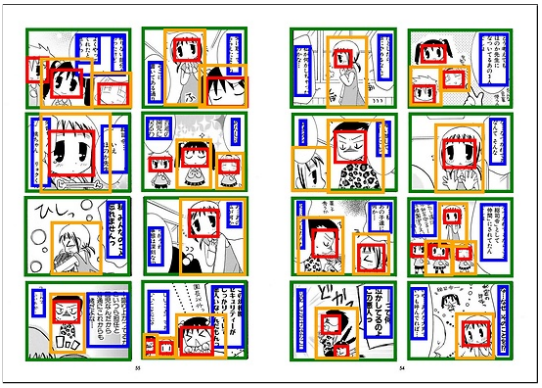
(b) SSD300-fork



(c) SSD300-x4

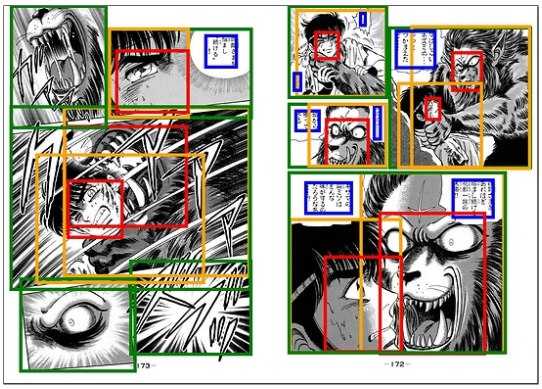


(d) SSD512-fork

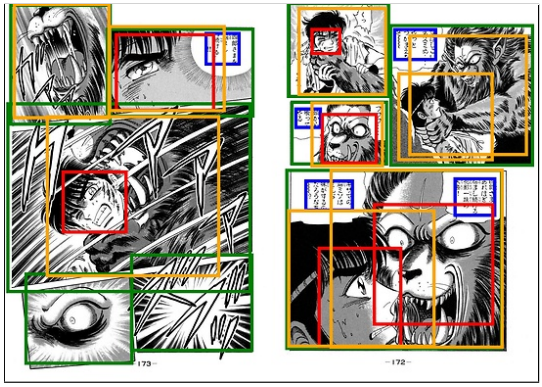


(e) 階層的 SSD300-fork

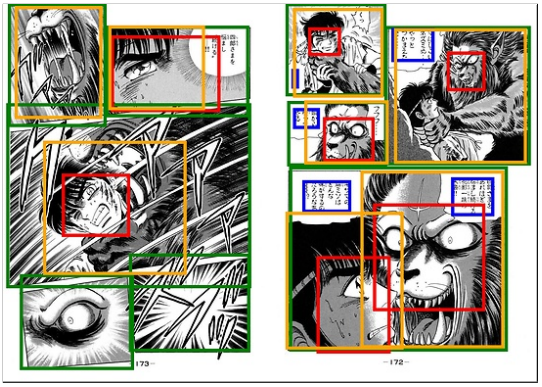
図 7.3: 拡張手法での検出結果[*17] (1/3)



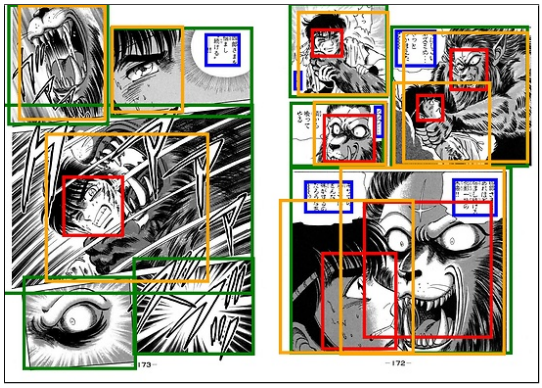
(a) Ground truth



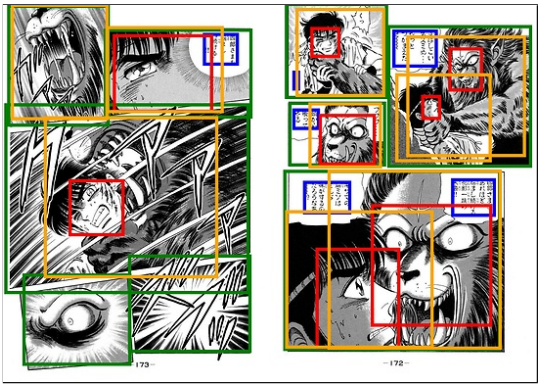
(b) SSD300-fork



(c) SSD300-x4

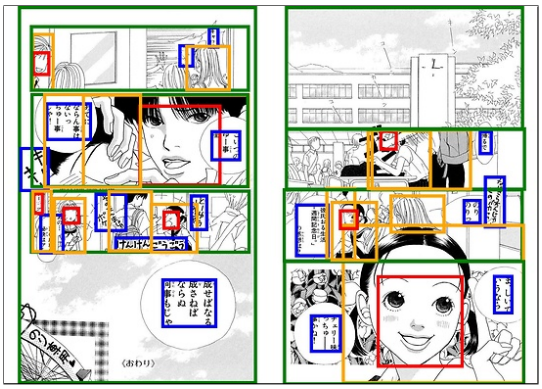


(d) SSD512-fork

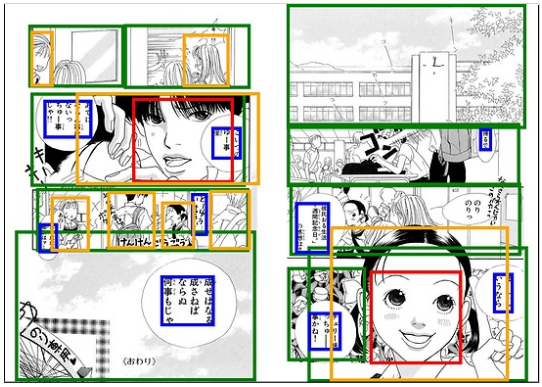


(e) 階層的 SSD300-fork

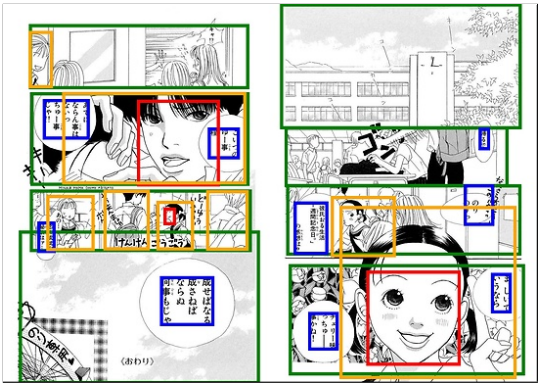
図 7.4: 拡張手法での検出結果[*18] (2/3)



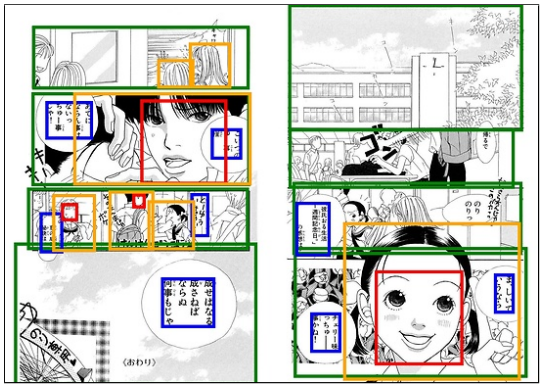
(a) Ground truth



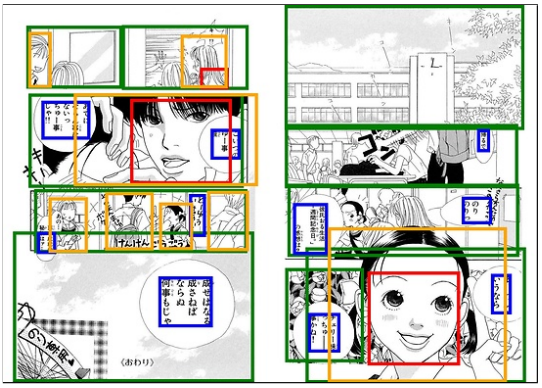
(b) SSD300-fork



(c) SSD300-x4



(d) SSD512-fork



(e) 階層的 SSD300-fork

図 7.5: 拡張手法での検出結果[*16] (3/3)

参考文献

[1] Mark. Everingham, S. M. Ali. Eslami, Luc. Van Gool, Christopher. K. I. Williams, John. Winn, and Andrew. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, Vol. 111, No. 1, pp. 98–136, January 2015.

[2] Clément Guérin, Christophe Rigaud, Antoine Mercier, Farid Ammar-Boudjelal, Karell Bertet, Alain Bouju, Jean-Christophe Burie, Georges Louis, Jean-Marc Ogier, and Arnaud Revel. eBDtheque: a representative database of comics. In *Proceedings of the 12th International Conference on Document Analysis and Recognition*, pp. 1145–1149. IEEE, 2013.

[3] Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan Boyd-Graber, Hal Daumé III, and Larry Davis. The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

[4] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, pp. 1–28, 2016.

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pp. 580–587. IEEE, 2014.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Conference on Computer Vision and Pattern Recognition*, pp. 779–788. IEEE, 2016.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, pp. 21–37. Springer, 2016.

[8] Azuma Fujimoto, Toru Ogawa, Kazuyoshi Yamamoto, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Manga109 dataset and creation of metadata. In *Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understand-*

ing, p. 2. ACM, 2016.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[10] Wei-Ta Chu and Ying-Chieh Chao. Line-based drawing style description for manga classification. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 781–784. ACM, 2014.

[11] Wei-Ta Chu and Wei-Chung Cheng. Manga-specific features and latent style model for manga style analysis. In *International Conference on Acoustics, Speech and Signal Processing*, pp. 1332–1336. IEEE, 2016.

[12] Christophe Rigaud, Nam Le Thanh, J-C Burie, J-M Ogier, Motoi Iwata, Eiki Imazu, and Koichi Kise. Speech balloon and speaker association for comics and manga understanding. In *Proceedings of the 13th International Conference on Document Analysis and Recognition*, pp. 351–355. IEEE, 2015.

[13] Thanh-Nam Le, Muhammad Muzzamil Luqman, Jean-Christophe Burie, and Jean-Marc Ogier. Content-based comic retrieval using multilayer graph representation and frequent graph mining. In *the 13th International Conference on Document Analysis and Recognition*, pp. 761–765. IEEE, 2015.

[14] Yuji Aramaki, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Text detection in manga by combining connected-component-based and region-based classifications. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pp. 2901–2905. IEEE, 2016.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

[16] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, Vol. 104, No. 2, pp. 154–171, 2013.

[17] Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision*, pp. 1440–1448. IEEE, 2015.

[18] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

[19] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. DSSD: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[24] Yusuke Niitani, Toru Ogawa, Shunta Saito, and Masaki Saito. ChainerCV: a library for deep learning in computer vision. In *ACM Multimedia*, 2017.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] Joseph Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016.

[27] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems in The Twenty-ninth Annual Conference on Neural Information Processing Systems*, 2015.

[28] Christophe Rigaud, Clément Guérin, Dimosthenis Karatzas, Jean-Christophe Burie, and Jean-Marc Ogier. Knowledge-driven understanding of images in comic books. *International Journal on Document Analysis and Recognition*, Vol. 18, No. 3, pp. 199–221, 2015.

[29] Kohei Arai and Herman Tolle. Method for real time text extraction of digital manga comic. *International Journal of Image Processing (IJIP)*, Vol. 4, No. 6, pp. 669–676, 2011.

[30] Christophe Rigaud, Norbert Tsopze, Jean-Christophe Burie, and Jean-Marc Ogier. Robust frame and text extraction from comic books. In *Graphics Recognition. New Trends and Challenges*, pp. 129–138. Springer, 2013.

関連する発表文献

国際論文誌

[1] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. “Sketch-based Manga Retrieval using Manga109 Dataset”. Multimedia Tools and Applications, Springer, 2016.

国際会議

[2] Azuma Fujimoto, Toru Ogawa, Kazuyoshi Yamamoto, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. “Manga109 Dataset and Creation of Metadata” International Conference on Pattern Recognition, Workshop MANPU (The First International Workshop on coMics ANalysis, Processing and Understanding), 2016.

[3] Yusuke Niitani, Toru Ogawa, Shunta Saito, Masaki Saito. “ChainerCV: a Library for Deep Learning in Computer Vision” ACM Multimedia, Open Source Software Competition, 2017.

[4] Toru Ogawa, Atsushi Otsubo, Rei Narita, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. “Object Detection for Comics using Manga109 Annotations” Computer Vision and Pattern Recognition (CVPR), 2018 (投稿中).

国内会議

[5] 藤本東, 小川徹, 山本和慶, 松井勇佑, 山崎俊彦, 相澤清晴. “Manga109 とそのメタデータ基盤の構築” 画像の認識・理解シンポジウム (MIRU), 2016.

[6] 小川徹, 山﨑俊彦, 相澤清晴. “漫画物体検出に向けた検出器の並列化” 第 16 回 情報科学技術フォーラム (FIT), 2017.

[7] 小川徹, 山崎俊彦, 相澤清晴. “並列化された検出器による高精度漫画物体検出” 映像情報メディア学会 メディア工学研究会, 2018.

その他の発表文献

国際会議

[1] Toru Ogawa, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. “Sketch Simplification by Classifying Strokes” International Conference on Pattern Recognition (ICPR), 2016.

国内会議

[2] 小川徹, 山﨑俊彦, 相澤清晴. “イラスト制作支援に向けたラフ画像の線画化” 画像の認識・理解シンポジウム (MIRU), 2016.

[3] 成田嶺, 小川徹, 松井勇佑, 山崎俊彦, 相澤清晴. “深層学習を用いたスケッチに基づく漫画検索” 第 31 回 人工知能学会全国大会 (JSAI), 2017.

[4] 山本和慶, 小川徹, 山崎俊彦, 相澤清晴. “データドリブンなアプローチを用いた漫画画像中の吹き出しの話者推定” 映像情報メディア学会 メディア工学研究会, 2018.

[5] 坪田亘記, 小川徹, 山﨑俊彦, 相澤清晴. “個別の漫画に特化した深層距離学習による漫画キャラクター特徴量の導出とクラスタリング” 電子情報通信学会 総合大会, 2018.

[6] 金子真也, 石見和也, 小川徹, 山崎俊彦, 相澤清晴 “深層学習による屋外利用に向けた SLAM の改善” 電子情報通信学会 総合大会, 2018.

謝辞

指導教員である相澤先生には学部時代のコンタクトグループで担当していただき, また漫画班として 3 年間お世話になりました. 修士のテーマを決められずに試行錯誤していたときも, 各テーマについて前向きなコメントをしてくださいました.

山﨑先生には特に発表やスライドついて多くのアドバイスをいただきました. 中でもよく知らない人がどう思うかという視点でのコメントは大変参考になりました. また先生が進めてくださった計算資源の拡充のおかげで, 大規模な実験も試すことができました.

学術支援職員の松林さんには学会出張など事務手続きで助けていただきました. 僕の不手際で手間を増やしてしまったことも多々あり, 申し訳ありませんでした. また漫画プロジェクトではデータの配布やアノテータへの謝金管理などの業務を引き受けてくださいました.

漫画班のメンバーである松井さん, 藤本さん, 山本君, 大坪君, 成田君, 坪田君には日頃のミーティングから外部でのデモまで大変お世話になりました. 特に松井さんは英語論文の執筆にあたって手取り足取り指導してくださいました. ご自身の研究が忙しい中, スケジュールから添削まで面倒を見ていただき, ありがとうございました. また藤本さん, 大坪君, 成田君にはアノテーションデータセットで大変お世話になりました. このデータセットは僕の修士研究において不可欠なものでした. 画像を提供していたたいだ作者の方々およびアノテーションに参加していただいた研究室の皆様, 外部のアノテータの方々にもこの場を借りて感謝を述べたいと思います.

同期の皆とは一緒に締切を乗り越えたり, 学会に参加したりしてきました. 皆でわいわいと作業することで締切前も楽しく過ごすことができました. また先輩・後輩の方々にも感謝を申し上げます. 研究の相談はもちろん, 一緒に夕食を食べに行ったり, 気分転換をしたりと様々な面で交流をしてくださりありがとうございました.

最後に修士になっても良くわからないことをしている息子を暖かく見守ってくれた両親に感謝を述べたいと思います.

2018 年 1 月 29 日

小川 徹