横浜国立大学　大学院環境情報学府
博士学位論文

マンガ画像からの情報抽出とその応用に関する研究

A Study on Information Extraction from Manga Images and Its Applications

情報環境専攻　情報学プログラム

村上　聡
Satoshi MURAKAMI

請求学位　　博士(情報学)
責任指導教官　長尾　智晴　教授

提出年月日　令和６年1月12日
請求年度　令和5年度3月修了

あらまし

マンガは，今まで，ドメスティックな文化として進歩を遂げてきたが，今や外務省の HP でもアニメや漫画をはじめとする日本のポップカルチャは，日本国内のみならず海外においても，若い世代を中心に人気を集めています[外務省　2016]，と記載されているように，インターナショナル・カルチャの一つと捉えられている．昨今の日本においてマンガは，むしろ，若い世代だけではなく，子供から大人までのすべての年齢層に親しまれており，ハリウッド映画作品の原作としての採用事例を代表とする日本マンガの海外輸出や，マンガ原作のTVアニメやドラマ作品の増加，キャラクタ商品の増加など，より身近に感じられるようになったため，マンガに関する文化的・社会的・経済的な重要性に関する意識は急速に高まりつつある．

2022 年のコミック市場全体（コミックス+コミック誌+電子コミック）の推定販売金額は 6,770 億円，そのうち紙のコミックス単行本とコミック誌を合わせた販売金額が 2,291 億円で電子コミックが 4,479 億円，出版市場全体に占めるコミック（紙と電子の合計）のシェアは 41.5% [出版科学研究所 2023]，しかも，電子コミックの市場規模は年々増加傾向にある．

小説は，挿絵として画像を使った補助表現も使うが，マンガと違い，基本的には文章だけを使って物語を散文形式で表現した，ある程度の長さを持った読み物で，人物の容姿やしぐさ，背景，天候，時間の経過など詳細に文章によって記述する．それに対し，マンガは，ページ内を 複数の　〝コマ〟と呼ばれる小領域に分割したうえで，それらのコマ内に絵を描き，その絵とコマ内のセリフ文字の連続で物語や時間の経過を表現した読み物である，小説のような文章による詳細な情景などの記述はコマ内の画像とセリフ文字で表現されるので，マンガの方が作者の意図がより読者に直感的に伝わりやすい．

近年，深層学習をはじめとした機械学習をつかった画像の分類，認識，画像に隠れた情報の掘り起こしなどに関して，様々な分野で研究が活発に行われており．中でも深層学習モデルの一つである Convolutional neural network (CNN) は高精度な手法として知られている．

そこで，CNN を使ってマンガの画像の中に埋め込められているすべての情報を抽出し，検索エンジンで利用可能な形式で蓄積できる仕組みの構築を長期的な研究目標とした．

最初に，本稿では画像そのものの価値を向上するために，CNN ベースの超解像拡大システム及びその拡大結果について記述する．本超解像拡大システムは，マンガに代表される画像形式の電子書籍にはさまざまなスタイルのページ画像が存在するため，入力された画像スタイルを分類し，その画像スタイルに対する最適な超解像 CNN のパラメータを選択して高品質の拡大を行う．

次に，本稿ではマンガのページ画像からコマを抽出するために，CNN でコマ領域の推定を行い，その領域からルールベースの画像処理を使ってコマを抽出する仕組みを構築した．既存研究ではすべてのコマを長方形として扱っているため，物体が存在するコマ位置を間違えて認識する可能性があったが本稿の方式では，正しいオブジェクトの存在位置に基づいた，より正確なマンガ理解を可能とした．

最後に本稿では，前述のコマ抽出の CNN を利用し，マンガ書籍の各ページからコマを抽出し，それらの各コマ内の不適切オブジェクト（露出した胸に限定）を検出するシステムについて記述する．本システムを実際に電子書籍の制作現場で運用し，運用開始前後で作業負荷が大きく減少したことを示す．

さらに，マンガの中に含まれる文字情報をテキストにするために，CNN を用いて文字情報のみを選択的に抽出し，シンプルな名刺のような白地に黒の奇麗な文字画像を抽出し，既存の公開されている日本語 OCR でテキスト化を行ったが，マンガ画像特有の問題だけではなく，日本語に対する OCR 性能も充分ではない事が判明し，カスタマイズ可能な OCR を新たに構築する必要性も検討中である．

Abstract

Manga has been developed as a domestic culture, but now it is considered as one of the international cultures, as the Ministry of Foreign Affairs of Japan states on its website, “Japanese pop culture, including anime and manga, is gaining popularity not only in Japan but also abroad, especially among the younger generation. In recent years, manga has become more familiar to all age groups in Japan, from children to adults, rather than just the younger generation. People's awareness of the cultural, social, and economic importance of manga is growing rapidly, as it has become more familiar due to the fact that Japanese manga is being exported overseas, as represented by its use as the basis for Holly-wood movies, the increasing number of TV animation and drama productions based on manga, and the increase in character products. In 2022, the estimated sales value of the entire comic market (comics, comic magazines and e-comics) is 677 billion yen, of which the combined sales value of paper comics and comic magazines is 229.1 billion yen and electronic comics is 447.9 billion yen.

The total sales of comics (paper comics and electronic comics) in the total publishing market are 41.5%, and the market size of electronic comics is increasing year by year.

Unlike manga, a novel is basically a book of a certain length that expresses a story in prose form using only text, although it also uses images as illustrations for supplementary expressions.
In contrast, a manga is a reading material in which the page is divided into several small areas called “koma” (frames), pictures are drawn within the koma, and the story and the passage of time are expressed through a sequence of pictures and dialog characters within the frames.

Manga is more intuitive in showing the author's intentions to the reader, since detailed descriptions of scenes and other details are expressed only through images and dialogue text within the koma, as is the case with novels.

In recent years, machine learning, including deep learning, has been actively studied in various fields of image classification, recognition, and uncovering information buried in images. Convolutional neural network (CNN), one of the deep learning models, is known as a highly accurate method. Therefore, I set my long-term research goal as the construction of a system that can extract all information embedded in comic images using CNN and store it in a format that can be used by search engines.

In this paper, at first, we describe a CNN-based super-resolution enlargement system and its enlargement results in order to enhance the value of the images themselves. Since there are many different styles of page images in electronic books in image format, such as manga, this super-resolution enlargement system classifies the input image style and selects the most optimal parameters of the super-resolution CNN for that image style to perform high-quality enlargement.

Next, this paper describes a system for extracting frames, the basic components of a manga book, from a page image using CNN to estimate the frame area, and then extracting frames from the area using rule-based image processing. In previous studies, all frames were treated as rectangles, so there was a possibility of misidentifying the position of the frame where an object existed.

Finally, this paper describes a detection system for inappropriate objects (limited to exposed breasts) in each frame of a manga book by extracting frames from each page of the book using the above-mentioned CNN for frame extraction. The system is actually operated at an e-book production site, and we show that the workload was significantly reduced before and after the system was installed.

In addition, in order to convert textual characters contained in manga into text, we selectively extracted only textual characters using a CNN, extracted a clean black on white image of a simple name card, and converted it into text using existing publicly available Japanese OCR. However, we found that not only the problems specific to manga images, but also the OCR performance for Japanese is not sufficient, and we are considering the need to build a new CNN-based OCR that can be customized.

目次

図目次

表目次

付録目次

第1章　　序論

1.1　研究背景

近年，マンガはポップカルチャに分類されており，一般大衆に広く愛好される文化を指すが ，その内容によっては，マイナーで独自性の強いサブカルチャに分類されることもある．絵画や純文学や古典演劇などのハイカルチャーに比べると，その歴史は短いものの，日本国内だけでなく海外からも人気を集めており，合わせてその情報発信の本場である日本という国に対する文化的な興味を高める要因にもなっている．

出版科学研究所によると，2021 年のコミック市場全体（コミックス+コミック誌+電子コミック）の推定販売金額（読者が支払った金額の合計）は，6,759 億円（コロナ禍の巣ごもり需要増のため前年 6,126 億円に対し 10.3％増）で，そのうち紙のコミックス単行本とコミック誌を合わせた推定販売金額は 2,645 億円（前年 2,706 億円，前年比 2.3％減）で，電子コミックは 4,114 億円（前年 3,420 億円，前年比 20.3％増）だった．2020 年に続き，90 年代の紙のみの市場のピーク（95 年＝5,864 億円）を 2 年続けて更新し，過去最大の市場規模に達した[出版科学研究所 2022]．図 1－1 にコミック市場推移の推移をグラフで示す．[出版科学研究所 HP 2023]

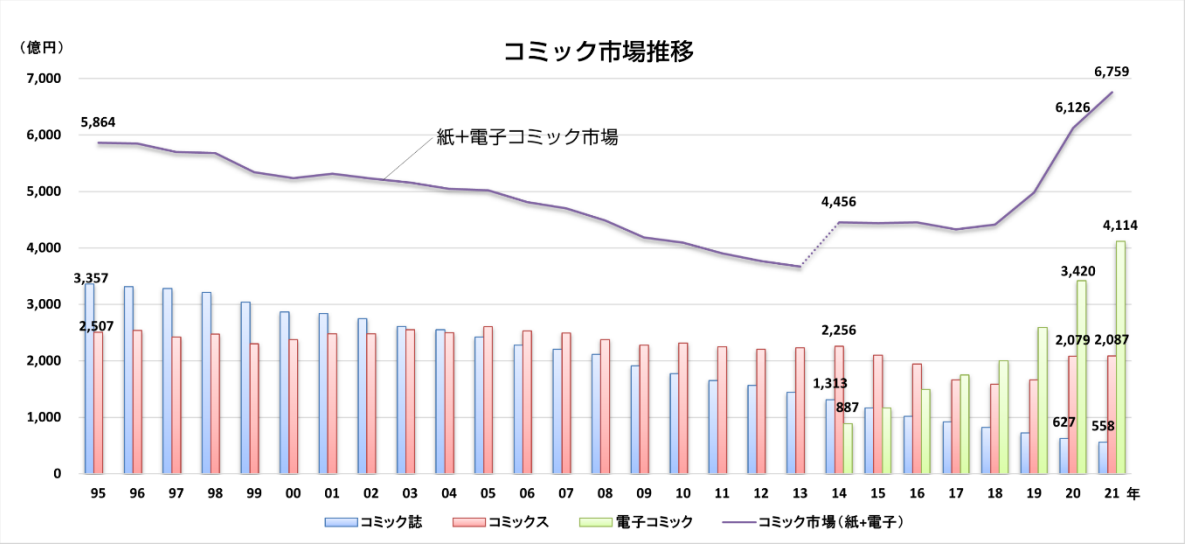


図 1-1. コミック市場推移 （出典『出版指標 年報 2022 年版』）

さらに，2022 年のコミック市場全体（コミックス+コミック誌+電子コミック）の推定販売金額（読者が支払った金額の合計）は，6,770 億円（22 年比 0.2％微増），紙のコミックス単行本とコミック誌を合わせた販売金額が 2,291 億円（コロナ禍の巣ごもり需要が収束のため 22 年比 13.4％減），電子コミックが 4, 479 億円（22 年比 8.9％増）で，出版市場全体に占めるコミック（紙と電子の合計）のシェアは 41.5％（22 年比 1.1 ポイント増）にも達し，今や出版される書籍の半分近くに迫る勢いである[出版科学研究所 2022]．

さらに，マンガは，その馴染みやすいキャラクタの一般広告への活用，キャラクタそのもののぬいぐるみ，キーホルダ，クリアファイルなどの関連商品への利用，アニメ化，ドラマ化，映画化またはドラマや映画原作としての利用など，その利用範囲及びその市場規模は，膨大な金額になると思われ，

文化的・社会的・経済的な重要性に関する意識は急速に高まりつつある．いまや，単なる娯楽の一種としてのみならず，言語や映像とはまた異なる，独自の形式を有するメディアあるいはコミュニケーション手段としても，社会のきわめて広い範囲にわたって浸透しつつあり，文化的な産業として新たな日本ブランドが確立されることも期待されている．

また，世界のマンガ市場の規模は，GrandView Research, Inc.の最新レポートによると，2030 年までに 422 億米ドルに達し，2023 年～2030 年までの年平均成長率：CAGR（Compound Average Growth Rate）は 17.4％で拡大すると予測されており，これからも引き続き，大きな発展が期待される．[グローバルインフォメーション 2023]

出版市場において紙ベースのマンガが主体だった時期から，そのメディアの主体は次第に紙から電子にその比率が移ってきたが，その背景には，通信環境の高速化と並行して，表示装置であるパソコンの表示画面の高精細化，高精細な表示画面を持つ高性能な携帯電話の普及が考えられる．

電子コミックを読む環境も大きく変わってきた．初期の電子コミックは，ほとんどがパソコン向けの市場だったが，次第に携帯電話の市場が立ち上がり，ページ画像ベースのサービスの他に，コマ画像ベースのサービスが立ち上がり，それぞれの市場が成長してきたが，2007 年に Apple 社のスマートフォンが米国に登場し，日本でも翌年に販売されてから，携帯電話の解像度が徐々に大きくなり，最初は 3.5 型で 320×480pix であったが，2010 年のモデルでは同じ 3.5 型だったが，640×960pix と 2 倍の解像度になり，一気に 4 倍のピクセル数に劇的に大きなものになり，Retina ディスプレイという名称が初めて使われ，高精細の概念が一般にも広がった．それ以来携帯電話の市場は高解像度モデルが主流になり，さらに高解像度化は進んでいる．市場が高解像度のスマートフォンが主流になると，パソコン向けのサービスが徐々に減少し，同じページ画像ベースのサービスではあるが対象機種はスマートフォンが主流となった．画面のサイズは 3.5 型から 6.5 型を越えるものも市場には出てきたが，持ち運びを考えると大きさは制限される．そのような背景から，従来の紙ベーすのページ画像を配信していたサービスに対し，対角線の長さが 5 インチ程度という物理的に小さなサイズの画面でマンガを楽しむ事を主体にした，縦スクロールでしかも小さな画面サイズに合わせて新しく描かれたマンガの配信という新しいサービスも生まれている．

さらに，身近なテレビ放送の放送方式の進歩もあり，1953 年に始まった NTSC 方式のアナログ放送は，走査線数 525 本，毎秒 30 フレームであった．1980 年代から衛星放送が始まり，90 年代には走査線 1,125 本でアスペクト比が 16:9 のアナログハイビジョン放送が始まり，2000 年には衛星放送がデジタル化（2K，FHD，1.920×1,080）し，さらに，2003 年からは2K で地上デジタル放送が始まり，ついに 2011 年にはすべてがデジタル放送に変わった．さらに高解像度での放送として，2014 年から4K スーパーハイビジョン（3,840×2,160），2016 年から8K スーパーハイビジョン（7,680×4,320）の試験放送も始まっている．

一方，パソコンの表示装置の技術進歩も激しく，デバイスも CRT から LCD に急速に変化し，またその解像度は標準化が進み，1993 年までは VGA(4：3，640×480)が主流だったが，1994 年には SVGA（4:3，800×600）が，1996 年には XGA（4:3，1024×768）が，1998 年には，SXGA（1280×1024）が，1999 年には UXGA（1600×1200）と高精細化が進んだ．

上記のような，表示装置・環境の技術進歩があり，そこに表示されるコンテンツの解像度も同様に高精細なものが求められるようになっている．

マンガと比べて，小説は，挿絵としての画像を使った補助表現も使うが，基本的には文章だけで物語を散文形式で表現した，ある程度の長さを持った読み物で，人物の容姿やしぐさ，背景，天候，

時間の経過などは詳細に文章によって記述されるが，読者のある程度の読解力や想像力を必要とするため，読者によっては十分に作者の意図が十分に伝わらない場合もある．

一方，マンガは主にページ内の複数のコマ内に情景とセリフ文字を表示し，それらのコマの連続で物語を表現した読み物で，小説のような文章による詳細な情景の記述はく，それらの情報のほとんどはコマ内に描写した画像に含まれているため，『百聞は一見にしかず』とも言われるように，作者の意図した情報は主に画像の形で読者にストレートに伝わりやすい．すなわち，マンガというメディアは，小説や映画などの表現とは大きく異なり，作者の自由な発想による表現効果が施された，コマ割りに紐づく独特な画像表現を有し，多少の曖昧さも残しながらも，その表現力の深さは他に類を見ない高度に知的なコンテンツである．

マンガ書籍を読者に提供するサービスにおいて，正確な使い勝手の良いサービスとして提供するためには，各マンガ書籍の要素を抽出し，分析理解することが必要である．これらの要素間の関係をさらに調査・理解することで，コンピュータによるマンガの理解を支援することができ，読者が画像中に含まれる情報を正確に検索するのに役立つ．マンガのストーリは，コマ，吹出し内のセリフテキスト，背景画像中の説明テキスト，表音文字，特殊効果などの描画手法，また，登場人物，背景画像，およびそれらの関係（例えば，前に説明された，～が言った，～が考えた，誰に向けて）といったさまざまな要素が複雑に絡み合って構成されている．そのために，今後は，コンピュータによるマンガの自動理解を目指したこれらの要素を自動的に分析する技術は，さらに重要になると思われえる．

紙ベースのマンガ書籍は手書きのモノクロの線画像が主体のコンテンツである．さらに，コマというページ内に配置された複数の小領域内での画像表現，及び，吹き出し領域内の文字を AI を含むコンピュータビジョン技術を用いて解析することで，従来は得ることが困難だったマンガコンテンツに関する新しい情報を掘り起こし，抽出してそれらの結果を検索できる形式で蓄積・保管することを長期的な研究目標とした．

AI による言語処理技術の進歩は，日本語に関しては欧米の言語に比べてかなり遅れていると言えるが，とてつもない速度で進化していることは紛れもない．ところが，コンピュータにテキストとして読み込ませることできなければせっかくの最新のAI技術があってもその恩恵を受けることはできない．そのため画像の情報と画像化された活字，手書き文字，擬音などはテキスト形式で収集する必要があるが，今まで出版されたマンガ書籍のデータを人間が読み取ってテキスト化するのは，現実的とはいえない．機械によって自動的にこれらの情報を日本語のテキストとして抽出できるような仕組みの研究開発が望まれている．

1.2　マンガの特徴

本稿において，直接的な研究対象として扱うマンガについてその特徴を述べる．
マンガ書籍に類似した，グラフィックス・ノベルと呼ばれる書籍は世界中で発行されており，なかでも日本・アメリカ．フランスでは独自の表示形式を特徴とする書籍市場が存在する．米国ではコミック（**comics**）ブックと呼ばれ，代表的なものは，1 タイトルのみの 32 ページ程度の薄いフルカラー印刷の刊行物として発行され，人気のエピソードのみをペーパーバックの形で 1 冊の書籍にまとめて販売される．過去には各種のジャンル（西部劇，恋愛，戦争，恐怖，犯罪，動物など）が存在していたが，1950 年代の「有害な」子供向け漫画に対する一連の排除規制により，ほとんどが淘汰され，現在ではカラーが主体のスーパーヒーローものが代表的なジャンルである．フランスやベルギーなどのフランス語圏ではバンド・デシネ（**bande dessinée**）と呼ばれ，「第 9 の芸術」と呼ばれることもあり，カラーでエンターテイメント性の高いものが中心だが，芸術性の高いものも存在する．そのほか中国，台湾，香港，韓国，フィリピン，マレーシア，インドなど世界各国にも同様のマンガが存在する．

日本のマンガは，おもにモノクロの手書きの線画として描かれ，独特のデフォルメされた絵柄や表現，ダイナミックなコマの配置も多く使われている．さらに，マンガの読書対象は広く老若男女すべてと言える．また，ストーリが長いのも特徴で単行本にして 100 冊を越えるシリーズも多く存在する．また日本以外の文章の向きは横書きが主流だが，日本の場合はほとんどが縦書きであるが，まれに横書きが混在することもある．

なお，マンガは漫画や，コミックなどと記述されることもあるが，これ以降，特に断りが無い場合，本稿では日本独自のマンガを意味する“マンガ”という記述に統一する．

マンガの画像の特徴は，1ページを一般的には長方形の複数のコマという小領域に分割したうえで，その個々のコマに別々の絵を描き，その絵とコマ内のセリフ文字の連続で物語や時間の経過を表現した読み物である．小説のような文章による説明が無いかわりにそれらの説明が直感的にとらえられる画像として表現されている，

これらのコマは，右上から左下の方向に沿って順に読むのが漠然としたルールであり慣習であるが，その順序に関してはコマ内外に読む順序の記述がないため，読者の裁量に任される部分もあり，逆に作者がそのあいまいな順序であることを利用して特殊な効果を狙うことある．

また，コマの配置や形状，大きさなどは作家のアイデアで自由に設定でき，コマ内のオブジェクトは一般的にはコマ領域をはみ出さないように描画されるが，様々な効果を狙って，しばしば領域を越えたオブジェクトの描画も存在するため，境界線はところどころ消失している事がある．

読む順序が確定している一般的な，小説，新聞・雑誌の記事などのメディアとの違いの一つでもあり，コンピュータで内容を把握する場合に困難になる一因ともなっている．

コマの中には，吹き出しという風船の形状で囲まれた領域があり，その中に活字で（まれに手書きの場合もある）文章が記述される．この文章は表示されているキャラクタのセリフであったり，心情の表現であったり，その情景の描写であったり，あるいは補足説明などが表示される．さらに，物語の状況の補足のために背景画像の中に吹き出しが無く，埋もれた状態で状況などの説明文などが表示されることもある，また，現実では実在しないが，効果音や状況を表現するための擬音などを画像化した文字や模様が描画され，物語の臨場感をより高めるための効果に使われることがある．

このように，マンガは画像と文字などの複雑な要素が自由な発想で配置され，高度な表現力を持ったメディアであり，コンピュータで分析，理解しそれらの情報もうまく検索に使えるようにするためには，コンピュータサイエンスのさまざまな分野，マルチメディア，画像処理，AI，ヒューマンマシンインターフェースなどの専門知識，研究が必要だと思われる．

1.3 研究目的

近年，デジタル化した書籍（e-book）の流通が当たり前になった．その配信データの形式は小説に代表されるテキストを主体とする形式と，マンガや写真集に代表される画像を主体とする形式が存在する．電子書籍の配信方式の概要に関して，図 1-2 に示すが，画像を主体とする配信データは，言うまでもなく，デジタル方式の静止画像である．

電子制作されたマンガ書籍の画像をデジタル形式で直接納品される場合以外のほとんどの電子書籍データは，紙に印刷されたマンガ書籍からスキャナを使ってデジタルデータに変換して取得する．そのため，画像の解像度はこのスキャン時の解像度で決定され，このスキャン時の解像度よりも精細なデジタルデータは存在しないことになる．ここにデジタル資産として数十万冊を超えるマンガデータが存在している．

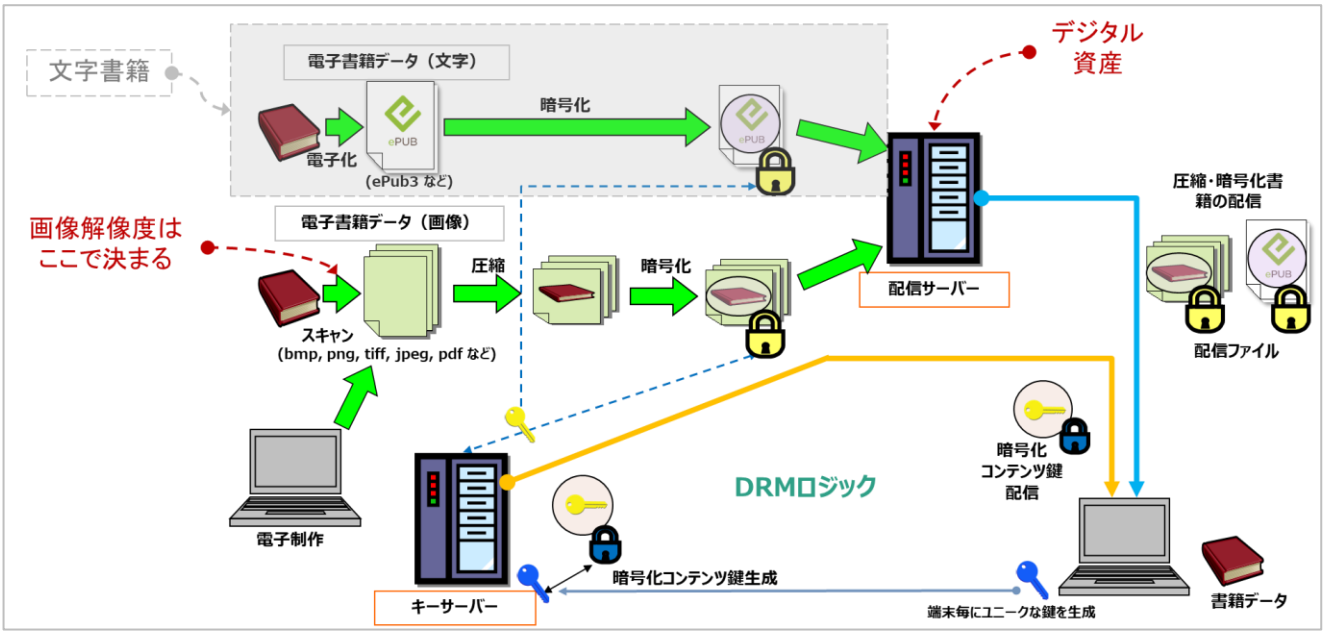


図 1-2．電子書籍制作･配信の概要

デジタルデータは劣化しないという大きな利点はあるが，それはあくまでもそのデータの解像度のままで利用する場合であり，任意のサイズに拡大すると画像が荒れてしまうという，重大な欠点が存在する．

デジタル方式の画像は，アナログ方式と違って画素という最小構成要素が存在し，画像はその画素の集合で構成されている．この画素は離散値なので，本来の解像度以上に拡大する場合には，元々は存在していない画素と画素の間の位置に新たな画素が必要になる．そのためには何らかの（推測アルゴリズムなどを使用した）補間を行って，元々の画像データとしては存在していなかった位置の新しい画素の情報を生成する必要がある．

表示器の高精細化が進み，過去に取得した静止画像に関しては，再度スキャンからやり直すのが最善策だが，短時間での対応は現実的ではないので，暫定的な対応策として高品質な低解像度画像から高解像度画像への変換策が必要となる．

また，前述の通りマンガは画像と文字などの複雑な要素が自由な発想で配置され，高度な表現力を持ったメディアであるが，コンピュータを使って自動でマンガを解析し，画像の中に込められている

コンテンツの内容にかかわる情報を抽出するための各種手法の開発が期待されており，従来利用できなかった新しい検索用の情報として活用できることも強く望まれている．

以上を踏まえ，本研究では以下の視点から検証をすることを目標とする．

1. 電子マンガ資産（デジタル静止画像）価値の維持
2. マンガ画像解析から得られる新しい情報の取得

マンガの画像に関しては，写真などの自然画像とは大きく異なり，一般的な画像 AI の恩恵を生かすことができる部分と，全く異なる部分があると思われ，さらに文字に関しても日本語が他国のどこの言語と比べても特殊であるということがあり，それぞれの場面に応じて，実際に実験を行いながら実用的な解決方法を思案していく必要があると思われる．

そのような背景の中で，著者の身近にあった電子書籍用の画像データの価値を高めるという視点から，以下のテーマに関して，実際の画像を使って順に深層学習などの画像処理技術を用いて実験を行い，その結果から，次のステップを考えていく研究計画を立案し，長期的にはコンピュータでマンガを読む AI を生涯目標の一つと考える．

(1) シングルフレーム超解像拡大処理
    (ア) 画像種別（スタイル）の分類
    (イ) 画像圧縮処理で生じる人工ノイズの削減
(2) マンガのコマの抽出
    (ア) コマの正確な抽出
(3) マンガ画像中の不適切オブジェクトの検出
    (ア) コマ内の不適切オブジェクトの認識
(4) マンガ画像中の文字画像領域の抽出とテキスト化
    (ア) 吹き出しの文字領域の抽出
    (イ) 背景画像に埋もれた，画像化活字文字領域の抽出
    (ウ) 文字領域内のノイズの削減
    (エ) 抽出文字画像セグメントのテキスト化（OCR）
        ① 既存の OCR による実験
        ② マンガ用OCRの要件
(5) マンガ画像中の顔検出と 作家別画像特徴量の抽出
(6) コマの読み順の推定と話者の推定・ストーリの要約
(7) オノマトペの抽出
(8) マンガのコマ画像からテキストへの変換

上記の（5）以降は，本大学院の在籍中には対応できなかったため，大学院終了後の目標としてとらえている．

なお，本研究で各種 CNN の学習等に使用したマンガ画像のデータセットは，著者が以前勤務していた株式会社イーブックイニシアティブジャパン（E 社）が保有していたマンガ画像から E 社の許諾の下で著者が抽出した，また，アノテーションデータも同様に筆者が付加したもので，両方ともすべて新規に自前で用意したデータセットであり，非公開である．

それに対し，公開されているマンガの画像データ及び各種アノテーションに関しては，Manga109データセット[Manga 15]が知られているが，このデータセットは，そのマンガコンテンツの制作時期が古く最新ではないこと及び，1 ページの画像が見開き単位の 2 ページ分になっており，1 ページ単位で扱っていた E 社の電子書籍データとは異なっており，さらにシングルステージのオブジェクト検出手法に基づく長方形ベースのアノテーションのため，本稿の目的とは異なり，本稿のオリジナルの研究開発（超解像拡大，マンガのコマの抽出，マンガ中の不適切画像検出）において CNN の学習用データセットとしては，いずれも使えなかった．

1.4 本稿の構成

第 2 章は，本研究に関する研究・関連研究，先行研究に関して述べる．

第 3 章は，保有しているマンガ画像を 2 倍に拡大する超解像拡大処理CNNについて述べる．拡大処理の CNN の入力画像がすべての種類の電子書籍ページ画像に対して最適に動作するか否かを検証し，電子書籍ページ画像すべてで活用できるような方法を検証する．また，超解像拡大時に，入力として使用する画像データがJPEGなどの画像データ圧縮を施したデータだった場合に，データ伸張時に発生する圧縮ノイズが，拡大と同時に 2 倍の大きさのノイズとして顕在化することの解決のための手法について記述する．

第 4 章は，マンガ画像中のコマの形状を長方形補間に固定せずに，本来のコマの正確な形状のコマとして抽出するためのセマンティックセグメンテーション CNN 及びコマ抽出処理について述べる．

第 5 章は，第 4 章のCNNを使って正しい形状で抽出したコマごとに，不適切なオブジェクトの存在の有無の判定をするための認識器について記述する．また 4 章のコマ抽出器と，不適切オブジェクト認識器の CNN を組み合わせ，高速演算が可能な GPU を実装した 1 台のサーバ上で 2 つのCNNを実行し，複数作業者が同時に RPC で接続した個別の PC からバッチ処理で複数冊のマンガ書籍内の不適切オブジェクトを検出するシステムを構築した．また，実用的な運用が可能であることを実際の作業現場で検証し，システム投入前後の作業効率の推移を確認したことに関して述べる．

第 6 章は CNN を使ってマンガ画像中の吹き出しの中の文字領域および背景画像に埋もれた文字領域の両方を抽出し，ノイズを除いた手札状のセグメント画像として出力する処理を検証し，その出力のセグメント画像を既存の OCR に入力し，テキストに変換を検証した，その場合に，日本語のマンガならではの問題点・改善点に関して記述する．

第 7 章は，本稿のまとめ及び今後の課題について述べる．

第2章　本研究に関する研究・関連研究

本章では，特に本研究と関わりのある研究領域について述べる．

2.1　研究領域

著者が本学の博士課程後期に在籍していたのと並行して所属していた会社は電子書籍の配信事業を営んでいる会社で，運営の基本資産として，電子書籍用の大量のデジタル化された書籍データを製造・保有していた．幸いなことに著者は，それらのデータの作成のための技術開発を行っていた関連もあり，それらの電子書籍データには，ほぼ自由にアクセスすることができた．そのため大量の高品位のマンガ等の静止画像データを AI の研究のため自由に取り扱えたため，研究の対象領域を主に“電子書籍用の静止画像”に焦点を当てて取り扱うこととした．そのなかでも，マンガ画像は，画像と文字が複雑に混在した独特なメディアであり，その画像を分析・解析するためには，複数の分野にわたる研究を同時に行う必要があると思われる．

マンガは一般的な写真画像，イラスト，CGなどの他のデジタル画像にはない特徴を有していると同時に，マンガを構成する要素に分けて考えれば，ビジネス文書，技術文書や新聞，雑誌，イラストや手書きの線画像など，身近にある画像にきわめて近い特徴も併せて持っていると思われるが，それらが単体としてではなくすべて同一の画面の中に存在しているので，既存の一般的な写真などの画像に対する研究手法がそのまま適用できるかどうかは，実際に適応してみなければ判断できないこともあると思われ，各種の実験は必須だと考えた．

そのような意味で，マンガ画像は類を見ない独特の画像スタイルであるがゆえに，解析手法やその結果など，得られた成果の活用分野は広いと思われる．

さらに，近年は，マンガは日本が世界に誇れる独自のポップカルチャとして大きく注目されているのにもかかわらず，小説とは異なり，しばらく前までは「単なる子供の時間つぶしの娯楽であり，成人した大人が読むようなたぐいのものではない」といった昔ながらの風潮もあり，先行した研究事例はそれほど多くはなく，”マンガ“そのものを研究対象として，技術的に解析し，理解するための研究はこれからも興味の尽きない分野であると考えられる．

2.2　超解像画像拡大

画像を拡大するという命題は古くからあり，デジタル画像であるという事を意識せずに，日常生活のなかでアナログの光学レンズを使った紙面上の文字や画像の拡大が可能であるという経験から，それと同様に，デジタル画像も簡単に拡大可能であると錯覚してしまい，いとも簡単に，しかも奇麗に拡大できると思いがちであるが，実際にはそのようにはいかない．

デジタル画像は，画像を構成するピクセル（pixel）という最小単位の点の一つずつに色と輝度の情報が付与されており，このピクセルを規則的に縦横に並べた配列として構成され，「1,024×768」のように「縦×横」のピクセル数の積の形で表現する．

例えば，この「1,024×768」のピクセルの配列を縦，横共に 2 倍の大きさにするには「2,048×1,536」のピクセルの配列にすれば良いことがわかる．そのためには，元々のピクセルとピクセルの間に，新しい何らかの値を有するピクセルを生成し配置する必要がある，

この新しいピクセルの生成方法は複数あり，単純な方法では，隣のピクセルと同じピクセルをコピーする方法から，両隣のピクセル値の平均値を新しいピクセルの値とするような方法などがあり，また，隣だけではなく周りの多数のピクセル値を用いた複雑な補間方法を用いて，より滑らかなピクセル値を生成し奇麗に見えるような方法が研究されてきたが，いずれの線形補間を使った方法でも複雑な画像に対しては，どうしてもぼけた画像になってしまうという欠点があった．**図 2-1**，**図 2-2** 参照．

図 2-1. Lena の各種フィルタ処理による拡大画像．（バストアップ部分）

図 2-1 の各種結果画像は，lena のオリジナル画像を OpenCV の cv2.INTER_AREA で 1/2 に縮小した画像をベースとして，ImageMagic の 20 数種類のフィルタを適用して拡大して得られた結果の画像から代表的なフィルタを使用して拡大した画像および，超解像と赤で記載している本稿の 3 章の超解像拡大システムを使って拡大した画像である．

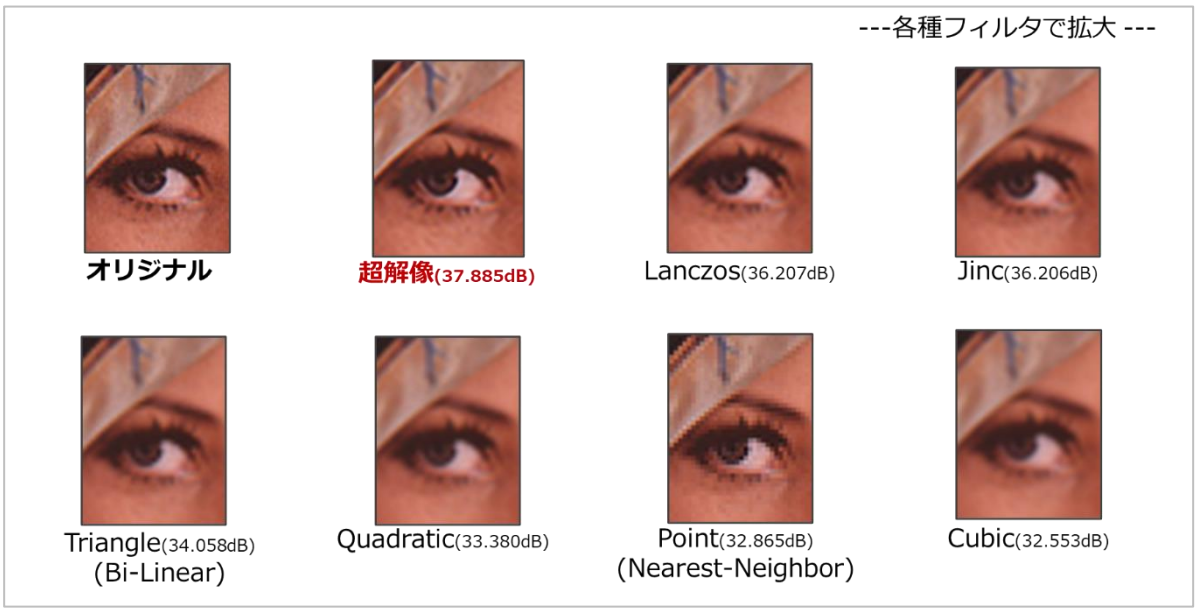


図 2-2. Lena の目元部分の各種フィルタ処理による拡大画像．

表 2-1 線形フィルタによる画像評価結果（抜粋）

lena の画像を OpenCV の cv2.INTER_AREA で 1/2 に縮小し，ImageMagic の数種類のフィルタで
2 倍に拡大しオリジナルと比較．cv2.INTER_AREA は，画像の縮小時にモアレを発生させない最適な補間方法

拡大フィルタ名	RMSE	PSNR	SSIM
超解像 Photo モード	3.253	37.885	0.949
Lanczos	3.946	36.207	0.939
Jinc	4.983	36.206	0.918
Triangle (Bi-Linear)	5.054	34.058	0.919
Quadratic	5.465	33.380	0.909
Point (Nearest-Neighbor)	5.798	32.865	0.914
Cubic	6.010	32.553	0.895

これらの線形フィルタによる各種の拡大処理に対し，1枚あるいは複数枚の低解像度画像から1枚の高解像度画像を推定し作り出す超解像と言われる技術があり，従来の高周波成分が欠落し，ぼけた画像になる補間アルゴリズム（線形補間）とは区別され，さまざまな手法が提案されてきた．代表的な超解像の手法には学習型と再構成型がある．

学習型の超解像処理の手法は，高解像度のパターンと対となる劣化の状態をシミュレートした低解像度の画像のペアを大量に生成し，高解像度のパターンと低解像度のパターンのペアからなるデータベースを作成しておき，入力された低解像パターンに対する劣化前の高解像度パターンを選択するといういわゆる辞書方式である．この方式は膨大なパターンをデータベース化する必要があり，また，データベースにないパターンの低解像画像ではうまく機能しないことが欠点である．

また，再構成型の超解像処理の手法は，同一のオブジェクトの複数（複数フレーム）枚の低解像度画像からサブピクセルシフト（1 ピクセルより細かい仮想的な単位を用いて処理すること）を用いて高解像度の画像を生成する技術で，結果画像に破綻が生ずる可能性が低いことが特徴で，この方式の応用としては，主にテレビなどの動画像の超解像処理として，オブジェクトの精密な位置合わせの技術と組み合わせて利用されることが多い．静止画像の超解像においては，利用可能な画像が動画像とは異なり，シングルフレームの画像となるため，この再構成型の超解像手法の適用は困難である．

近年の機械学習の発展に伴い，CNN（Convolutional Neural Network）を使った手法として，Dong らは，従来の学習型といわれる辞書ベースに基づいた SRCNN という超解像手法[Dong 14]，[Dong 15]を提案し高精度化を実現した．この研究では CNN がいかにして，「縮小方法が既知の画像から，なるべく元の画像に限りなく等しい高画質な画像を推定するか」という辞書方式超解像の命題を解くために用いられている．

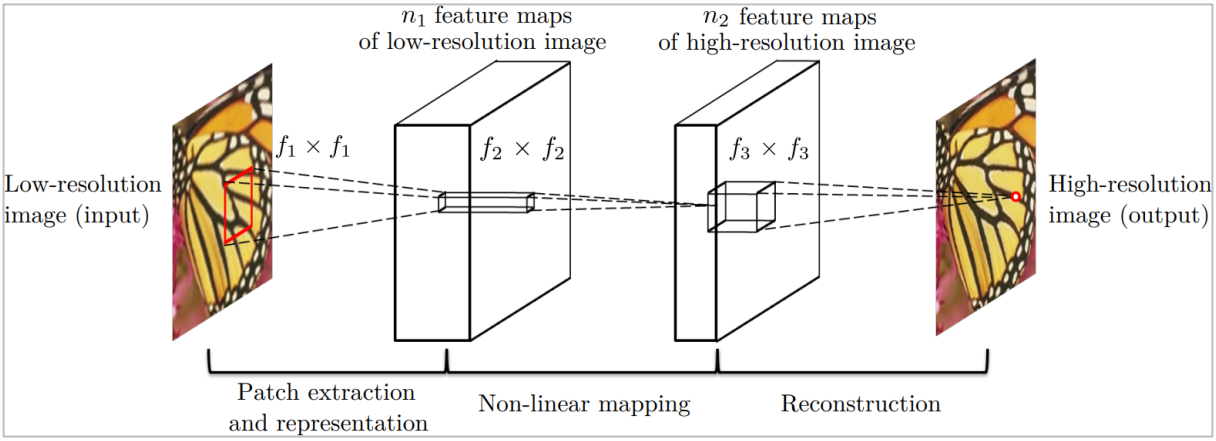


図 2-3. An overview of the SRCNN network. [Dong 14], [Dong 15]

SRCNN は, **図 2-3** に示すように全体が 3 層の CNN 構造になっており,

1 層目が 9×9 の畳み込み層(conv 9x9, 64 filters)で,
2 層目が 1×1 の畳み込み層(ReLU -> conv 1x1, 32 filters)で,
3 層目が 5×5 の畳み込み層(ReLU -> conv 5x5, 1 filter)から 構成されている.

拡大処理は CNN の内部では行わず, CNN の直前の外部で Bicubic 法で事前に拡大処理したものを CNN で refine するという構造になっていて, 損失は, CNN の結果と Ground-Truth 画像との平均二乗誤差を用いた. この方式は CNN を超解像に応用した最初の提案で, SRCNN と呼ばれ, 当時の最高手法を凌駕し画期的なものではあったが. 以下の 2 つの問題点を内包していた.

(1) CNN の層数が少ないため表現力に乏しいが, 性能向上のために 3 層以上に層数を深くすると不安定になってしまい, うまく学習ができない.
(2) 外部の拡大器の計算コストが高い

そこで, この SRCNN の(1)の層数を増やして表現力の向上ができないという問題点を解消することを目的に, 単純に層を増加させても学習が不安定になってしまうところを Residual Learning という手法[Kim 16]を使って防ぐ VDSR という方式を Kim ら[Kim 16]が提案し, SRCNN を越える性能となった. VDSR ネットワークの概要を**図** 2-4 に示す. これにより, CNN の層を 3 層以上に深くしても, 性能向上が可能となる根拠となった.

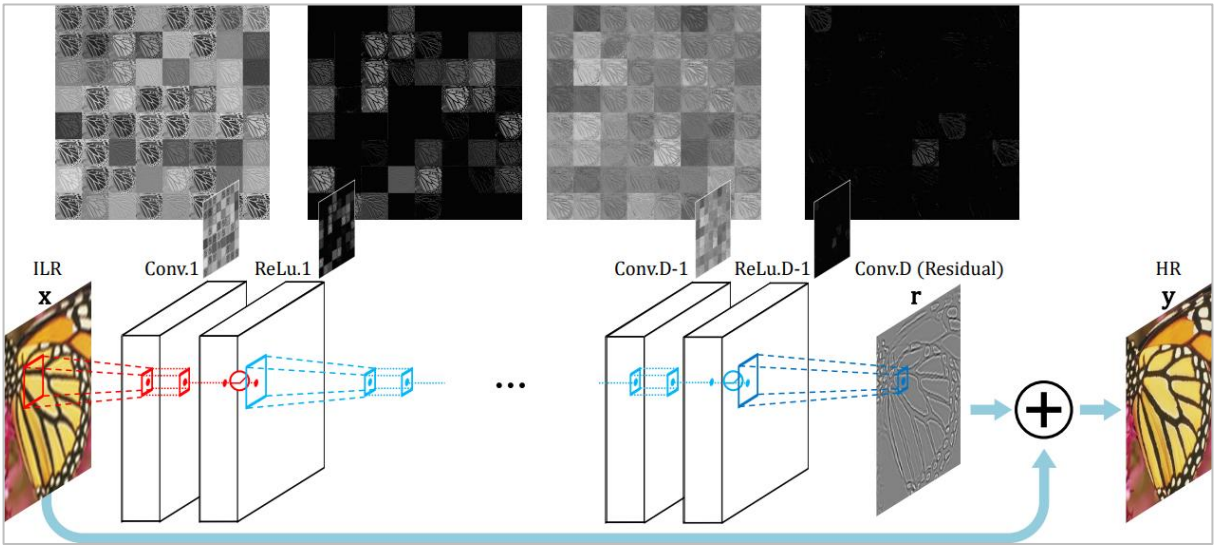


図 2-4. An overview of the VDSR network. [Kim 16]

さらに，(2)の問題点を解消することを目的に，SRCNN を提案したチームが，CNN の直前にある外部の Bicubic 法の拡大処理の計算コストを削減することを目的に，最終段の CNN の内部で拡大をおこなうための deconvolution（transposed convolution とも呼ばれる）構造を取り入れた FSRCNN（Fast SRCNN）を提案し，拡大性能は低下させず高速化を図ることが可能となった．FSRCNN のネットワーク構造の概要に関して，**図 2-5** に示す．

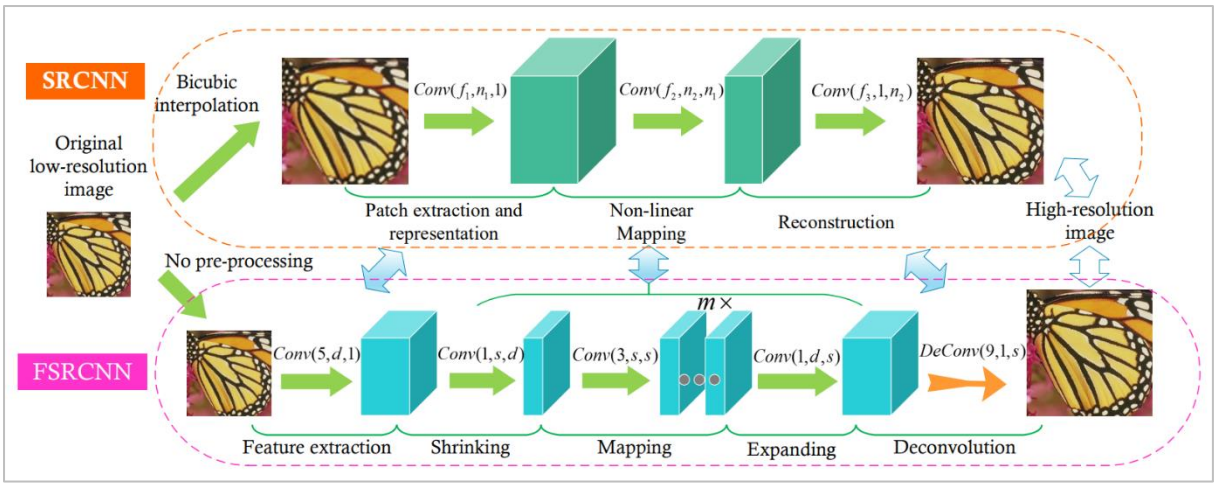


図 2-5. An overview of the FSRCNN network. [Dong 16]

これらの研究によって，3 層以上に深い CNN を使用し，高速に超解像拡大が可能なことが解明されたが，これらの研究ではその学習および評価用の画像セットは，Set14 に含まれる”Powew Point2002”の表示のある画像の 1 枚を除いて，すべてが自然画像を扱っており，マンガもしくはマンガに類似する画像を扱っているものは皆無である．

そのため，手書きのモノクロの線画像および活字文字の混在したマンガ画像にこれらの手法の適用が可能であるか，また，適用した場合に問題はないかなど検証する必要があり，文字画像，写真画像，図表などが挿入された雑誌や辞典などの画像などのデジタル配信用に作成された画像に適用した場合の問題点についても同様に検証が必要である．

2.3 コマの認識

マンガのコマに関しては，その形状，大きさ，配置が自由で，さらには，コマ境界線がたびたび描画オブジェクトなどにより遮られていることがあり，境界線の一部が欠損していることが多いなど，その描画の自由度が多いゆえにコマの抽出の問題は難問で，深層学習が興隆する以前にもいくつものコマの認識方法が提案されている．

抽出しようとしているコマの形状の多くは長方形または（4 隅の角のいずれかに直角ではないものが含まれる）四角形だが，境界線の状態により種々のケースがあり，以下のように整理される．

(1) いずれの辺の境界線も消失している部分が無い長方形のコマ（最も一般的），
(2) 1辺以上の辺が水平または垂直に対し傾斜している四角形の形状のコマ．四角形以外の多角形のコマ，
(3) オブジェクト，吹き出し，オノマトペなどの表音文字などがガター（コマの境界線と隣のコマの境界線，または紙面の端との間の空白の領域）まではみ出し，コマ境界線の一部が消失しているコマ，
(4) オブジェクト，吹き出し，オノマトペなどの表音文字画像などがガターを越えて隣のコマの中まではみ出して描画（ぶちぬき）され，2 つの隣り合うコマの境界線の一部が両方とも消失しているコマ，
(5) コマの一辺以上が裁ち落とし（境界線が存在せず紙面の端まで画像が存在）になっているコマ，
(6) 大きなコマの中に，別のコマの一部または全部が入り込んで境界線はあるがガター部分が存在しないコマ，
(7) 境界線のない（明示的なコマの境界がない）コマ，

上記の形状の各種コマを織り交ぜて使うことで，マンガの表現力を効果的に高めることができる．一方，マンガのページからコマを抽出する場合には，(3)～(4)のように，オブジェクトに遮られ描画されていない複数の箇所の境界線や，(5)のように，元々存在しない境界線や，(6)のガターが無い場合の境界線や，(7)のようにそもそも境界線が存在していない場合を正しく推定する必要がある．

さらに，上記の複数の状態が同時に該当している場合もあり，すべてのケースでコマ境界線を正しく抽出することは，単なる線分の検出処理だけでは極めて難しい．

線分の検出では古くから知られている古典的な直線の検出手法である Hough 変換[Hough 60] を利用した手法[Duda 72], [Ballard 79]や line segment detector [von Gioi 12]などの利用も考えられるが，いずれも，欠損のある境界線の推定は極めて難しい．

2007 年に Apple 社の iPhone が発表され，その後に大画面の携帯電話が市場を席巻することになる以前の 800×600pix よりも小さな表示画面サイズの携帯電話向けの電子マンガ配信では，コマ画像ベースの配信方式が主流であった．コマ画像ベースの配信は表示画素数が少なく，その表示領域が限られた小さな表示器を持つ機種が対象で，マンガ画像を小さな画面にフィットさせて読書体

験の向上を計った. それを実現するために, 膨大なマンパワーを使ってページ画像から手動で1コマずつ切り出して配信されていた. その作業の合理化のためにマンガの各ページ画像からコマを自動的に切り出すことが重要な技術として着目され研究された.

そのころのディープラーニングの隆盛以前に発表されたコマの抽出に関する研究としては, 再帰的 X-Y カット(ギロチンカットとも呼ばれる)アルゴリズム [Han 07], 網羅的探索 [Chan 07], あるいは密度勾配 [Tanaka 07] を用いて, 隣接するコマ間の境界線を検出する方法, ガターに注目して境界線からコマの検出を行う方法[石井 07], 「GT-Scan」と名付けられた濃度勾配などの画像処理技術を使った方法[野中 09]などが提案された. しかし, X-Y カットに基づく方法[Han 07]は, ノイズ等を含む長方形形状以外のコマをロバストに分割することができず, また, これらの方法はいずれもコマ境界線に欠損のあるコマや明示的な境界線のないコマを扱うことは困難であった. さらに, コマを分割するために, 主に連結成分ラベリング(CCL:connected component labeling)アルゴリズム [Arai 10]に基づく方法, あるいはページの背景マスク [Pang 14] に基づく方法が提案されたが, 白い背景ときれいなガター(2 つの隣り合うコマ境界線の間の細い空白部分)に依存しており, 長方形形状以外の不規則な形状のコマを個々の成分として識別することはできるが, オノマトペや吹出しも含む複数のオブジェクトによって境界線が消失し, 見かけ上結合しているコマを分離することは困難である. 結合したコマを処理するために, CCL マスク上で N 回の縮小と拡張 [Ho 11] のシーケンスを繰返して結合要素を分割するが, 1 辺以上が裁ち落としになっていてコマ境界線が完全に無い場合の CCL マスクには, 断片化された境界領域のグループが個々の構成要素として含まれることがあり, 侵食処理から完全なコマの形状を得ることは困難である. 連結成分のバウンディングボックスのクラスタリング [Rigaud 13]や, 検出された境界線の候補とコーナーに長方形を当てはめる[Stommel 12]ことで, 裁ち落としのあるコマ境界線の形状を回復できる場合もあるが, それらは規則的な形状のコマにのみうまく適用できた.

ほとんどの手法ではコマの領域の推定にはヒューリスティックな特徴量を使用するので, 想定とは異なるレイアウトの場合には必ず失敗する. 前述のレイアウトの複雑さや内容の多様性のため, これらの手法では依然として充分な抽出精度は得られないことが判明した.

ディープラーニングを用いた先行事例として, 物体検出モデルの SSD300 を利用した小川ら[Ogawa18]の研究や, 物体検出モデルの YOLO を利用した Arpita [Arpita19]らの研究がある. どちらの手法も物体検出のための長方形の提案領域を利用するため, 抽出するコマの形状は長方形に限定される. マンガの内容の正確な分析・理解のためには, 正確なコマ情報の抽出が望まれる.

2.4 不適切画像の自動検出

Web 上に氾濫する, 不適切な画像に関しての先行研究としては, Jay Mahadeokar らの研究[Mahadeokar 16]があるが, 1 ページの写真画像やイラストが対象で, 不特定形状のコマ内に描かれたマンガ画像は検出対象から除外されている.

近年, マンガが海外, 特に米国で新しいエンターテインメントメディアとして受け入れられつつあるが, 少年・少女向けのコンテンツに関するレーティングが規定されているため, 日本国内では問題にならなかった少年・少女向けマンガの中に含まれている女性の胸の露出などの不適切なシーンが含まれている場合には, 日本で出版されたマンガの文字部分だけを翻訳しただけでは, 受け入れられない. 従来の紙ベースでのマンガ書籍の海外販売の場合は流通と在庫のリスクがあり, 販売されるコンテンツ数も限られていたため, 極小的なマーケットに留まっていたが, 電子書籍の市場の場合は, 紙ベースの書籍販売とは大きく異なり, 流通と在庫のリスクがほとんどなく, デジタル通信網を使って

どこでもいつでも品切れがなく，手軽にデータをダウンロードできるため，PC はもとより，最近では携帯電話で大いに市場が活性化している．また，電子書籍の場合は，現物の商品を輸送する必要がないため，簡単に国外にも電子マンガを販売することが可能なので，文字部分が翻訳できれば，大きなマーケットが期待できる可能性がある．特に，全世界に電子コンテンツの販売網を有する既存のプラットフォームを利用したコンテンツ販売を行う場合は，（必要であれば，文字の翻訳などをした）コンテンツを準備できれば，そのまま海外向けに販売することも可能である．

従来，電子書籍の制作・配信手順は，**図 1-2** に概要を図示しているように，大部分の電子書籍はこれまで発行された紙ベースのマンガ書籍をそのまま電子化して販売してきた．少年少女向けの紙に印刷された既存の販売コンテンツであっても，女性の胸の露出に関しては国内では規制の対象ではなかったため，それ以外の不適切シーンのみを目視でチェックしたうえで販売してきた．

紙ベースの書籍から，既に，電子書籍の配信形式に変換してしまった数十万冊を超える電子書籍データ，および，毎月新規に制作する電子書籍データに対し，すべてを目視で確認するのは，作業量を考えると大変な作業である．何らかの作業量の軽減手法が渇望されていた．

2.5　文字領域の抽出

マンガの中に含まれる文字画像は，レンダリングされた活字文字の場合がほとんどで，一部に手書き文字が含まれる．会話などのセリフや頭の中で考えた事柄などのほとんどの文章は吹き出しの中に存在し，状況説明などは背景画像の中に埋もれた状態で存在する．また，その頻度は低いが，それ以外に手書き文字として存在する事もある．

現状は，すべて人が手作業で抽出しているが，マンガの内容理解や，海外販売のための翻訳のためのオリジナルデータとして，これらの文字情報を自動で抽出するという需要は多い．

CNNを利用して，文字領域だけを抽出し，既存のOCRを適用する事で文字画像のテキスト化まで自動でおこなえることが期待できる．ただし，既存のOCRは，ビジネス文書画像や，帳票画像，Webの文書画像などを主要なターゲットにしているため，マンガ独特の難しさの存在も想定される．

実際にマンガ画像を使用して，既存の技術でどの程度まで対応できるかについて，検証が必要である．

第3章　電子書籍画像の超解像拡大処理

3.1 はじめに

画像ベースの電子書籍のなかには様々な特徴をもつページ画像が混在している. そのため, 全てのページにおいて高品質な拡大を実現するためには, 単一の超解像処理を用いるだけではすべての画像に対しては充分な性能を得ることが不十分であることが予想できる.

また, 低解像度で保存されている画像は膨大に存在するが, その多くに画像圧縮ノイズを含んでいるため, 圧縮ノイズ成分は削減しながら, これらの多様な低解像度画像を高品質の高解像度画像に自動的に変換できるシステムが必要だと思われる.

そこで, 自動的に入力されたページ画像を複数のグループに分け, それぞれのページ画像に対して最適な特性の CNN（Convolutional Neural Network）に基づく超解像処理を適用し, 同時に圧縮ノイズの低減処理を行うことが可能な電子書籍用の超解像処理システムが求められる.

入力されたページ画像の特徴に最適な拡大条件を維持しながら適用可能な画像のパターンをできるだけ広くするために, 学習データセットをグループ化してディープラーニングする方法, および人手を介さずに拡大作業を自動実行するための手法に関して検証する.

電子書籍用のデータに限らず, 低解像度でアーカイブされている画像は膨大に存在するため, 自動で多種・大量の低解像度静止画像を高品質な高解像度画像に変換するための需要は大きく, 本稿のシステムはその用途に有益である.

本研究では主に, 紙に印刷された画像をスキャンして取得した静止画像に関して拡大性能のチューニングを行うとともに, 画像の拡大時に圧縮ノイズを含んでいる画像から人工的な圧縮ノイズを同時に低減する方法に関しても検証し実装した. 本システム及び実現手法は実用上有用であると思われる.

3.2 背景

電子書籍サービスにおいて一般的な画像配信方式の書籍データは, 紙に印刷された書籍ページの全ページをイメージスキャナなどによってページ毎にスキャンし, そのスキャンされた画像に対し, 位置, 色調, 輝度, ノイズなどの画像補正をおこない, 規定サイズに縮小した画像を圧縮して暗号化し, 1 冊分にまとめたデータとして作成し, ビューワアプリにはこの 1 冊分にまとまったデータ単位で配信する. ビューワアプリではサーバから配信された書籍データを受信し, 復号・伸張し, クライアント側の表示器の解像度に合わせて, 縮小または拡大して表示する.

従来, 配信画像データはクライアントの表示器の解像度よりも充分高い解像度を有していたため, ビューワでは画像データを縮小して表示していたので, 意図的に部分拡大をする場合以外は拡大する必要が無く, 拡大に伴うボケなどの画像劣化の問題が深刻化することはなかった.

高解像度の表示機器に低解像度の画像を入力する際は, 何らかの解像度変換（拡大）処理が必要となるが, 従来の線形フィルタによる拡大補間法では, 拡大後の画像のエッジ（線の両サイドなどの急峻な画像変化部分）のボケ, ジャギ（線や輪郭に現れる階段状のギザギザ）やリンギング（画像のコントラストの高い部分に不自然な輪郭が発生する）といった画像の劣化が生じてしまい, 表示機器が本来有する高解像表示能力を充分生かすことができない.

近年，ディジタルカメラ，スマートフォン，携帯情報端末，4K/8K テレビなど，身近な映像表示機器の高解像度化が驚くほどの速さで浸透している．それに伴い，電子書籍の表示機器の解像度も高解像度になり，配信画像の解像度を超えてしまうと単純にページ画像を表示する場合でさえ必ず拡大表示が必要になり，前述の拡大処理に伴う不自然なボケなどが発生し，画質劣化に起因する配信画像データの商品価値を大きく損なう結果となる．特に 2005 年以前に取得した画像データは低解像度で保存されているものがほとんどで，スキャンをするための原本となる古本などの印刷媒体の入手も困難なため再スキャンができないものが多数存在する．これら低解像度でしか存在しない大量の書籍の配信画像データをそのまま高品質な拡大画像に変換を可能にすることで，画像の商品価値を大きく損ねることなく再活用可能な価値のあるデータへと変換することができる．

従来から，拡大画像を得る方式として，各種の線形補間方式（nearest neighbor，Bilinear，Bicubic，lanczos，etc.）等が提案され性能改善が続けられてきたが，いずれの方式も拡大後の画像の線の両サイドなどの急峻な輝度変化部分は目視で充分判別できるレベルのボケ，ジャギやリンギングといった画像の劣化が生じるため電子書籍用途には適さない．

前述の各種の線形補間による拡大法に対し，Dong らは Bicubic 法によって仮拡大した画像を，深層学習した畳み込みニューラルネットワーク（CNN: Convolutional Neural Network）を使用して高品質な高解像度画像を生成する超解像処理（SRCNN）を提案し[Dong 14][Dong 15]，驚異的にその画質を向上させた．この Dong らの論文以降，CNN をベースにした多数の超解像処理に関する論文[Simonyan 14] など多数が発表されている．本システムでも超解像処理に深層学習を使った CNN を採用しているが，電子書籍のように様々な特性を持ったページ画像が混在している場合，1 種類の特性の CNN だけではすべてのページに対して充分な性能を維持できない．さらに圧縮ノイズを含んでいる低解像度画像を超解像処理すると画像全体が均等に拡大されるため極小の圧縮ノイズであったものが肉眼でも容易に認識しやすくなり，超解像処理をしたはずなのにかえって画質は悪くなったような悪印象を与えてしまうので，超解像拡大処理時に同時に圧縮ノイズは拡大せずに削減する機能が必須になる．本稿では電子書籍向けに開発した，特性の異なる複数の CNN を用いて多様なページ画像に対応可能な全自動超解像処理システムについて述べる．

3.2.1 従来の画像拡大技術

拡大画像を得る方式として，古くから知られている最も簡単な方法は，ニアレスト ネイバ（nearest neighbor）法という一番近い画素値を予測画素値にする画素の四角が単純に大きくなっていくだけの方法があるが，得られる拡大画像は極めて低品質である．

3.2.2 補間方式

バイリニア（Bilinear）法という予測画素周辺の 2×2 の画素とその距離に応じた割合で混合した値を予測画素値とする手法や，バイキュービック（Bicubic）法という予測画素周辺の 4×4 の画素との距離に応じた割合で混合した値を予測画素値とする手法などに代表される各種の線形補間アルゴリズムが知られている．Bilinear 法も Bicubic 法も，nearest neighbor 法よりなめらかにはなるが，画像はどうしても不明瞭（ボケ）になり，同様に高画質であるとは言い難い．

できるだけ．明瞭な拡大画像が望ましいが，線をシャープに表現するには，拡大画像側にも本来あるはずの高周波成分が必要になる．この高周波成分をなるべく損なわずに，元画像が保有している周波数成分の情報を最大限に活用できる方法が望ましい．解決策の一つの代表として，1 点の予

測画素を推定するために，元画像に含まれている高周波成分をできる限り活用し，現実的な計算量で実行する方法として lanczos 法が考案され，この方法に類する手法が従来手法による実用的な線形補間アルゴリズムとしては，ほぼ理論的な限界であると言われている．実際に拡大してみると，かなり綺麗には見えるがこの方法を使って拡大変換された画像においてもまだ不明瞭であり，まだまだ充分な品質であるとは言い難い．

3.2.3 機械学習の応用

前述の手法に対し，Dong らはニューラルネットワークを用い，Bicubic 法によって仮拡大した低周波成分画像から，畳み込みニューラルネットワーク（CNN）を用いて対応する高周波成分画像を生成する超解像（SRCNN）を提案し[Dong 14]［Dong 15]，驚異的に画質を向上させた．この論文以降，CNN をベースにした多数の超解像拡大に関する論文が出されている．[Kim 16]

機械学習による画像の拡大方式は，見方を変えると，低画質画像から拡大画素を予測する従来の方式に対し，低画質画像に対応する高画質の画像を教師画像として学習させる方式なので，高画質画像を破壊変換して得られた低画質画像を使って元の画像を復元・生成するという問題と捉えることができる．本稿では，この CNN をベースとした超解像拡大手法の成果を応用し，電子書籍画像用に専用化するとともに，自動で多種大量の静止画像の超解像拡大を行なうための機能を追加し実現した，高速で実用的な画像拡大システムの構築およびその評価について記述する．

3.3 開発目標

システムを開発するにあたり，以下の目標を設定した．

(1) 拡大画像の PSNR が従来の補間方式（lanczos）及び，SRCNN の品質を超える
超解像処理システムの入力画像は電子書籍の配信用に加工済みの画像データを想定し，様々な特性のページ画像全てで，その拡大画像の PSNR が従来の lanczos で拡大した場合の品質を大きく超え，さらに SRCNN の PSNR を 2dB 以上超えることを第一の目標とした．

(2) 画像の超解像拡大時に，画像圧縮ノイズは拡大せずに削減
配信用画像データは複数のページ画像それぞれを圧縮し，その圧縮データを暗号化している．そのため配信用画像データから取得したページ画像には，画像圧縮アルゴリズムに起因する人工的なノイズが含まれており，画像を拡大することで物理的にその圧縮ノイズも同時に拡大してしまうことで，綺麗な超解像処理をしたにもかかわらず，逆にノイズが際立ってしまうため，超解像処理と同時に圧縮ノイズ削減を行うことを第二の目標とした．

(3) 拡大処理は，全自動で実行
一般に電子書籍を出版する企業には多くの書籍の画像データが蓄積されている．例えば著者が在籍していた会社は漫画書籍を主体に 50 数万冊以上（2022 年 10 月時）の画像型の書籍を有していた．それらの書籍画像の特徴に最適な条件で高解像化するための条件設定などの人手による余分な作業コストを抑えるため，システムに入力する画像は人間の判断なしに自動で超解像拡大できる必要があり，これを第三の目標とした．

3.4 システムの必要条件・仕様

3.4.1 超解像処理方式について

超解像処理は CNN を採用するが，この CNN では拡大と同時に圧縮ノイズの削減もできなければならない．そのため，SRCNN で採用している CNN 直前の外部に仮拡大器を設置し，2 倍に拡大された画像を入力する構成ではなく，入力画像の微妙な圧縮ノイズ情報も直接 CNN に入力し CNN 内部の最終段の deconvolution で拡大を行なう waifu2x[Nagadomi 19]のシステムの upconv7 を採用した．

waifu2x ではそれぞれ vgg7，upconv7，resnet14l と名付けられた 3 種類の超解像拡大のネットワークを公開している．vgg7 と upconv7 は，それぞれ 7 層のコンボリューション層から構成され，resnet14l は 14 層で構成されている．それらのネットワークの構成は実用的な速度で拡大ができるようにそれほど深くせずに，×2 倍の拡大に限定して良好な結果を得られるようにバランスを考えて構築したオリジナル構成になっている．

オリジナルの waifu2x に関する性能評価結果に関しては，github[Nagadomi 19]の appendix に公開されている．公開版の waifu2x ではイラスト用と写真用の 2 つのジャンルに専用化した学習済みモデルを公開している．

ところが，電子書籍には，マンガや小説や雑誌など，明らかにイラストと写真以外の画像バリエーションが存在する．システムを単純にするためには，なるべく少ない種類のパラメータセットを使ってあらゆる種類の画像を高品質に拡大できるのが望ましいが，この 2 種類の構成だけでは十分に奇麗な拡大性能を実現できない．

3.4.2 様々なスタイルの画像の拡大品質の維持の必要性

電子書籍の画像には多様なスタイルのページ画像が混在する. カラーイラスト画像だけで学習した学習済みの upconv7 でカラーイラスト, 写真, モノクロマンガ, 文字主体の4種類の画像を拡大したところ, カラーイラストでは良好な PSNR であったが, 写真, モノクロマンガ, 文字主体の画像ではイラストと同程度の PSNR を得ることができなかった. 超解像 CNN の拡大品質が学習に使用した画像セットのスタイルに強く依存するためである.

そこで, 電子書籍の多様な画像スタイルの全てで良好な拡大品質を維持するために, 入力のページ画像のスタイルに対応した特性の異なる複数の CNN で補間することで多様な画像スタイルの全てに良好な品質を担保できる複数 CNN による補間方式を採用する.

CNN の学習では高周波成分を含む高解像度画像を教師画像とし, その縮小画像を入力として, end-to-end で CNN を訓練し, 縮小によって喪失した高周波成分の復元を学習する過程とも考えられる. したがって高品質な拡大結果を得るためには, 低周波成分の画像パターンに対する最適な高周波成分の画像パターンを効率よく推定することが鍵となる.

そこで, 同じような特徴をもつ画像をいくつかのグループ(以降画像スタイルと呼ぶ)に分け, その画像スタイルごとに学習データを準備して学習した場合とグループ分けしない教師データで学習した場合とを比較し, どちらが優れているかを検証した.

同一の販売ジャンルの書籍はそのページの組版(紙面上のレイアウト)や構成などが類似した作りになっていることが多いので, 同一の特徴をもつ画像セットを効率よくグループに分けるために, 電子書籍の販売属性ジャンルを活用した.

大雑把に 5 種類の画像スタイル(漫画, 小説, 写真集, 雑誌の 4 分類, さらに漫画はカラーとモノクロに分け全部を 5 分類)にグループ化した. (

表 3-1)

それぞれの画像スタイルごとに, 教師用画像 8,000 枚, 検証用画像 400 枚を収集して学習データセットを構築した.

表 3-1. 画像スタイル分類一覧

グループ	画像 スタイル	内容
漫画モノクロ	Manga_gray	モノクロ漫画のみからなるページ
漫画カラー	Manga_rgb	カラー漫画のみからなるページ
小説モノクロ	Novel_gray	主にモノクロの文字のみからなるページ
写真カラー	Photo_all	主にカラーの写真からなるページ
その他カラー	Misc_all	主に雑誌等の文字・挿絵・写真等がページ内に混在するページ

カラー漫画の場合はカラーのイラストに近い特性をもっていると考えられるが, モノクロ漫画の場合は, ほとんどが太さやトーンが異なる線, 吹き出し, 文字のみで構成されており, 色に関する情報が無いうえに, 描画されている線にはほとんど規則性は無いと考えられること, および, スクリーントーンという漫画独自の装飾表現が多く含まれるため, 一般的な写真やイラスト画像, 小説などと比較するとその特徴は大きく異なっているので, 別分野として取り扱うことにした.

表 3-2. 書籍ジャンル分類による拡大結果（×2 倍）

データセット	評価画像セット	lanczos PSNR	混合モデル PSNR	専用モデル PSNR
漫画モノクロ	BMP	19.997	23.797	23.871
	TIFF	23.346	27.699	28.049
漫画カラー	BMP	23.945	29.304	29.568
	TIFF	24.790	30.639	30.880
小説	BMP	22.517	29.979	31.339
	TIFF	27.072	38.034	38.790
写真	BMP	37.566	40.215	40.691
	TIFF	36.449	38.696	38.704
雑誌その他	BMP	26.491	32.443	32.973
	TIFF	29.033	35.877	36.515

画像スタイル分類されたそれぞれの画像セットで学習した結果と，全画像スタイルを均等に混合した画像セットで学習した結果の PSNR(Peak signal-to-noise ratio)を表 3-2 に示す．lanczos 法による拡大結果は従来技術と比較するための参考値である．（単位は dB，評価画像セットの BMP は縦方向画像サイズが 1024 以下の画像，TIFF は，縦方向画像サイズが 1024 を超える画像）

電子書籍の販売ジャンルによる大雑把なスタイル分類による学習データセットでの実験結果では，予想通り，すべてのケースにおいて，スタイル分類をした画像データで構成された学習データセットで学習した拡大結果の方が，全てをミックスした画像データで構成された学習データセットを使った学習結果よりも良好であった．

3.4.3 圧縮ノイズ削減機能

拡大時に顕在化してしまう圧縮ノイズの問題を解決するためには，超解像処理時に圧縮ノイズだけを削減し，それ以外を高品質に拡大できることが必須である．

拡大前の画像で使用される圧縮アルゴリズムは複数存在しており，アルゴリズムによりその圧縮ノイズの特性が異なる．これに対応するために画像スタイルと同様に，入力画像のノイズの特性に応じて複数の特性の CNN を切り替えて対応する．

3.4.4 自動変換機能

オペレータの判断なしに入力画像の特性の違いを判断し前述の複数の特性の CNN から最適な特性の CNN を自動選択するための機構の実装が必須となる．そのため，画像スタイルの分類器および画像に含まれている圧縮ノイズ特性量分類器の 2 種類の分類器が必須となる．

3.4.5 学習データセットの専門性をより高める

販売ジャンルによる分類による学習データセットには，書籍単位で同一のタグを付けたため，1 冊の書籍に含まれるすべてのページ画像には同じ画像スタイルとしてタグ付けをしていた．ところが，1 冊分の画像をページ単位で見直してみると，ほとんど全ての書籍で販売ジャンルとは異なる別のス

タイルにタグ付けすべき画像が混在していた．例えば，漫画の単行本の場合，表紙，目次，扉，作者の経歴のページ，あとがき，あるいは，解説のページなど，明らかにコマ割りされて描画されているマンガ本文の画像スタイルではないページが一定量含まれている．

より高品質な結果を得るためには，より専門性を高めた学習データセットを使った方が良いので，書籍単位の分類を見直し，厳密にページ単位の分類に変更して新しい学習画像データセットを構築し直して，実験・検証を行った．42,000（5 ジャンル×(8,000＋400)）枚のすべての画像のスタイルを目視で再確認するのはとても骨の折れる作業であること，さらに，大量に高速に超解像変換を実施する為には，作業オペレータが介在して画像ページごとのスタイルの確認及び指定をするわけにはいかないことなどの理由から，画像スタイルを機械的に分類する機構を開発することにした．

画像スタイル分類器も CNN を使った機械学習で開発することにした．ネットワーク構造は画像分類では定番ともいえる ILSVRC（ImageNet Large Scale Visual Recognition Challenge）2014 で Oxford Visual Geometry Group が使用した VGG Net[Simonyan 14]と呼ばれるモデルを参考にして，なるべく高速実行ができるように層数をスケールダウンした．

画像スタイルの分類は，商品の販売ジャンルを使って画像スタイルをグループ化したのに倣って，漫画モノクロ，漫画カラー，小説，写真，雑誌その他 の 5 ジャンルとしたが，分類器での漫画の扱いは，モノクロ・カラー共通とし，分類後に色情報を判断して画像を分けることとした．スタイル分類器の分類正解率を**表 3-3** に示す．平均エラー率は 0.53%となった．

表 3-3．画像スタイル分類器の分類正解率

画像スタイル	モデル正解率
漫画	99.15%
小説	100.0%
写真	99.17%
雑誌その他	99.56%
スタイル分類平均	99.47%

このスタイル分類器を使って学習画像データセットを自動分類し，再構築したデータセットを使用して，改めて CNN の学習を行った．また，このスタイル分類器は，入力画像を超解像拡大する場合に最適なパラメータを自動選択するための前処理器としても使用した．

スタイル分類器を使って改めて学習データセットを構築し直して再学習を行ったが，予想通りページ画像単位でスタイル分類をして作成した学習データセットを使って学習したほうが，書籍単位の販売ジャンルをもとにスタイル分類をして作成した学習データセットで学習した結果よりも優れた値となった．**表 3-5** に一連の実験結果をまとめて記述する．

3.4.6 高速化と画像圧縮ノイズ

画像データは，文字等のデータに比べてその容量が極めて大きくなるため，その取り扱いを容易にするために，データ圧縮をしてデータ容量を小さくすることが多い．データの圧縮方式は複数存在するが，イメージスキャナでスキャンされた画像データを取り込む場合なども含め，画像に関しては jpeg フォーマットが使われることが多い．

jpeg フォーマットはデータ量が小さくなり簡単に扱えるため利便性が高く一般的に用いられているが，非可逆でデータ圧縮をするため，画像圧縮コーデック特有のアルゴリズムに由来する人工的な

圧縮ノイズ（以下，単に圧縮ノイズと記述する）が生成され，圧縮率および元画像の複雑さの程度によって生成されるノイズ量は一様ではない．

一般的に圧縮率がそれほど高くない場合には，その圧縮ノイズは肉眼ではわからないか，ほとんど気にならない大きさだが，圧縮率が高い場合には肉眼でも画像の劣化がはっきりわかるようになる．一般的に，画像圧縮をしてデータ容量を大きく削減したい場合には，非可逆圧縮を採用することが多く，jpeg 以外の画像圧縮コーデックでも同様の傾向の問題を内包している．

通常，画像を拡大する場合は画像の全体が均等に拡大されるので，圧縮ノイズを含んでいた場合は，このノイズ自体も同様の拡大率で拡大されるため，拡大されたことによって物理的に認識できる大きさになり，拡大前よりも一層ノイズが目立つという好ましくない現象が顕在化することになる．この好ましくない現象を解決するには，画像を拡大する場合に圧縮ノイズだけは削減しながら画像全体を均等に拡大する必要がある．

waifu2x では vgg7 の処理の高速化を狙い，CNN の最終段を deconvolution に置き換えた upconv7 を公開している．upconv7 では vgg7 では画像入力の直前で必要だった仮の拡大器（最近傍補間法による拡大器）を使わない構造となっている．本システムでは，膨大な量の低解像度画像を処理するために少しでも高速で処理しなければならないため vgg7 だけではなく，upconv7 も取り入れることとした．upconv7 の構成図を**図 3-1** に示す．

upconv7 の学習データセットは，vgg7 の学習時と同じものを使用し，更に拡大の学習と同じ教師画像に対し，圧縮ノイズを含む新しい入力画像を追加で用意し，拡大の学習と同様に，圧縮ノイズを含む縮小された低画質画像を高画質画像に復元する学習を行うことにより，拡大と同時に圧縮ノイズの削減が可能であることを確認した．

学習器のネットワーク構造を upconv7 に変更したことで，vgg7 方式に比べ CNN への入力画像の pixel 数が 1/4 になりそれに伴い計算量が大きく削減できた．

実行速度に関しての比較結果を**表 3-4** に示すが，拡大処理の時間は約 1/3 になり，大幅な処理時間を短縮することができた．なお，検証データは「書籍ジャンル分類による拡大結果」で使用した漫画モノクロの 400 枚の検証用画像である．

表 3-4. ネットワーク改良後の実行速度

データセット	画像セット	vgg7 処理時間 (sec)	upconv7 処理時間 (sec)
漫画 モノクロ	BMP	164.49	59.15
	TIFF	1180.95	369.98

upconv7 では CNN の入力直前にあった仮の拡大器を削減したため，圧縮ノイズが拡大されない生の状態で CNN の入力になり，ネットワーク内で拡大と同時に効率的にノイズの特徴の評価もできるようになり，拡大と同時に行うノイズ削減の学習効率が良くなった．さらに vgg7 では拡大をネットワーク外の仮の拡大器で行っていたため，拡大に関与するパラメータが CNN の学習には全く反映できなかったものが反映できるようになった等の理由により，圧縮ノイズの削減と同時に拡大の性能も向上したと考えられる．

また，学習データセットをスタイル分類せずに混合した場合とスタイル分類した場合の結果の差がそれほど大きくないので upconv7 を使って追加実験を行った．混合モデルの学習データセットは，画像スタイル分類のための学習データからランダムに 8,000 枚の画像を抽出し学習データとした．

合計の学習データ数は各専用モデルと同じになっているが，データ内に存在する各スタイルのデータ数は専用モデルの 1/5 になっており，混合モデルの結果の方が，各グループ分類された結果に比べ不利な条件になっている可能性を否定できなかったため，スタイル分類の学習データのすべての画像データ，各分類 8,000 枚ずつ合わせて合計のデータ数としては各専用モデルの 5 倍の 40,000 枚の画像を使った学習も行い比較を行った．

これを混合モデル 40K，8,000 枚のサンプリングで学習したモデルを混合モデル 8K として結果を**表 3-5** に併記する．単位は dB．

混合モデル 40K は混合モデル 8K よりも確かに良い結果になったが，自動分類による専用モデルの方がさらに良い結果となった．これらの結果から，予想通り，より専門性の高い学習データセットを使った学習結果の方がより高品質の拡大が可能であるということが確認できた．また，画像のスタイルを指定せずに混合した画像セットを使って拡大を行なった場合の比較結果を**表 3-6** に示す．使用する専用モデルはスタイルの自動分類により選択する．この結果でも自動分類による専用モデルを使って拡大した場合が一番良い結果となった．

表中の lanczos は，一般的なフィルタでの実行例として ImageMagick で拡大した場合の参考値である．

本システムの超解像エンジンは，公開されている waifu2x とネットワーク構造は同一だが，学習に使用したデータセットは電子書籍用の画像をスタイル分類して専用化したデータセットなので，その特性は公開版とは異なっている．

表 3-5. 画像スタイル分類後の拡大結果(x2:upconv7)

スタイル 分類データセット	評価画像 セット	混合モデル 8K	混合モデル 40K	専用モデル upconv7
漫画モノクロ	BMP	25.466	25.505	25.592
	TIFF	30.716	30.820	31.194
漫画カラー	BMP	28.179	28.318	28.472
	TIFF	31.896	32.177	32.337
小説	BMP	33.424	33.629	34.727
	TIFF	39.119	39.392	39.537
写真	BMP	39.191	39.506	39.896
	TIFF	37.528	37.673	37.988
雑誌	BMP	31.854	31.961	32.305
	TIFF	37.317	37.395	37.766

表 3-6. 混合データの評価(PSNR) (x2)

データセット	lanczos	混合モデル 8K	混合モデル 40K	自動分類による 専用モデル
混合 BMP	26.145	31.623	31.784	32.199
混合 TIFF	29.098	35.315	35.491	35.764

3.5　提案する全自動超解像処理システム

3.5.1　　システムの概要

3 種類の CNN，圧縮ノイズの削減と同時に超解像処理を行う CNN（SR+NR-CNN：Super Resolution + Noise Reduction CNN），および様々な入力画像のスタイル分類を行う CNN（S-CNN: Style Classifier CNN），およびノイズ特性量分類を行う CNN（N-CNN：Noise Classifier CNN）を開発しそれらを組み合わせた．これらの CNN を深層学習する際の学習用の画像データは，配信用の電子書籍データに加工する前のアーカイブ画像（非公開）を後述の手法により分類して用いた．

SR+NR-CNN はあらかじめ S-CNN と N-CNN のそれぞれの分類器の結果の全ての組合せを満たす学習用画像データを用意して学習する．

画像の拡大時は入力画像を S-CNN と N-CNN で分類し，その結果の組み合わせからその画像に最適な SR+NR-CNN を選択することで，電子書籍画像の全てのスタイル画像に対し良好な拡大性能を維持する．

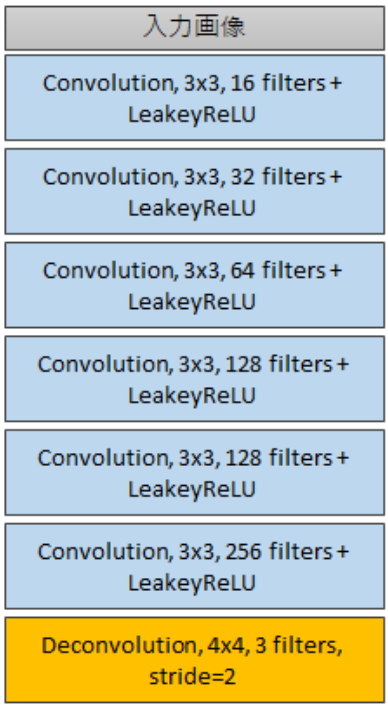


図 3-1．upconv7(deconvolution model) のネットワーク構成

3.5.2　　SR+NR-CNN の構造

本システムで採用した upconv7 の構成を**図 3-1** に示す．CNN の層数はそれほど深くせず，2 倍の拡大率（面積は 4 倍）に限定する事で実用的な速度で高品質な超解像処理ができるように永冨氏が開発したオリジナル構成である[Nagadomi 19].

電子書籍サービスでユーザに表示される画像はユーザ側の読書ビューワが表示デバイスの物理解像度に合わせて縮小して表示するため，配信側の画像サイズはユーザのデバイスの解像度より充分大きく品質が良ければ部分拡大しても実用上大きな問題にはならない．むしろ CNN が複数拡大率に対応することによる画像品質の低下や学習データの準備および学習時間の増大等のデメリ

ットを勘案し，2 倍以外の任意の拡大率は CNN 単体では対応せず，必要に応じて超解像処理エンジンの外部で対応することにした．

3.5.3 超解像処理の学習データ

低画質画像に対する高画質の画像を教師画像として学習させる方式の CNN による超解像処理で高品質の拡大性能を得るためには，大量の品質の良い学習用画像セットが必要となる．本システムでは配信用の電子書籍画像の制作のために蓄積していた大量の高解像度画像（非公開）を利用し，同じような画像スタイルをもつ書籍を電子書籍の販売属性ジャンルデータベースを活用して 5 つのグループに分類した．

同一の販売ジャンルの書籍はそのページの紙面上のレイアウトや構成などが似た作りになっていることが多く，同一の画像スタイルを有すると思われる販売ジャンルごとに約 100 冊を抽出し目視確認を行って同様な画像スタイルをもつと思われる次の 4 グループ（1：漫画，2：小説，3：写真，4：それ以外）に分類した．さらに各グループを，例えば漫画は少年／少女／女性…など，詳細なジャンルに分類した．

モノクロ漫画にはスクリーントーンというカラー漫画には無いモノクロ独自の強調や塗りつぶしの表現が多数含まれており，カラー漫画とはその特徴量が大きく異なるためモノクロ漫画とカラー漫画を別々のジャンルとして扱うこととした．

また，小説にはカラーの文字はほとんど無いためモノクロ画像として，写真はほとんどがカラーなのでカラー画像として，その他は雑誌などの種々雑多な画像や文字，グラフが入り混じっているのでカラー画像として扱うことを決定し， **表 3-7** に示す漫画モノクロ，漫画カラー，小説モノクロ，写真カラー，その他カラーの 5 種類にグループ化した．

E 社（著者が以前勤務していた電子書籍の製造販売会社）では，画像の縦サイズが 1024，1200，1600pix のいずれかの大きさに正規化処理の加工後に分類されていた保存画像と，スキャン後のサイズ補正が加えられていない縦サイズが 2400pix 以上のまちまちの大きさの未加工画像として分類されていた保存画像があったので，両方の保存画像からそれぞれ同数ずつ抽出して使用することとした．できるだけ多様な特徴を学習できるように，なるべく多くの書籍のランダムなページ位置から均等に数ページ分の画像を抽出し，まず，グループ毎におよそ 10,000 枚を学習用の画像の候補として抽出した

.

表 3-7. 画像スタイル分類 一覧

Table 3-7 Set of Image Style classification

グループ	画像スタイル	内容
漫画モノクロ	Manga_gray	モノクロ漫画のみからなるページ
漫画カラー	Manga_rgb	カラー漫画のみからなるページ
小説モノクロ	Novel_gray	主にモノクロの文字のみからなるページ
写真カラー	Photo_all	主にカラーの写真からなるページ
その他カラー	Misc_all	主に雑誌等の文字・挿絵・写真などが 1 ページ内に混在するページ

抽出した画像セットは書籍単位で分類してグループ化を行ったため，1 冊の書籍から抽出された全てのページ画像には同じスタイルにタグ付けしていたが，全ての書籍の画像で，抽出したグループの特徴とは異なる別のグループにタグ付けすべきページ画像が混在していることが判明したので，

さらに高品質な拡大結果を得るために全ページ画像をさらに目視で確認し，厳密に分類し精度を高めた画像セットとして，グループ毎におよそ 10,000 枚の画像で基本となる画像セット（以降，ベース画像セットと記述）を再構成した．

前述のグループ分類と区別するため，以降は画像スタイル分類の名称を同様の意味の英語表記である Manga_gray，Manga_rgb，Novel_gray，Photo_all，Misc_all. と記述する．このベース画像セットの画像スタイル毎にランダムに学習時の訓練用画像に 8,000 枚，訓練時の検証用に 400 枚，学習後の性能検証用に 400 枚として合計で 8,800 枚を学習用画像として抽出した．

CNN の学習時は教師画像とする高解像度画像を 1/2 に縮小した低解像画像を入力するが，特定の縮小方式に偏らないように複数の縮小方式（box, bicubic, lanczos）をランダムに選択して画像を作成した．

前述の性能検証用の 400 枚の画像セットを lanczos で 1/2 に縮小した画像を入力画像として，CNN を使用しない従来手法の lanczos，および SRCNN，upconv7 で 2 倍の拡大画像を作成した結果の PSNR を**表 3-8** に示す．全ての画像スタイルで，400 枚の結果の最高値，最低値側からそれぞれ 10 枚を除いた 380 枚で平均を取った．（以降の性能検証では 400 枚のセットの結果の最高と最低側から 5%削除した 380 枚で平均を計算．） SRCNN の結果は論文をベースに著者が学習したものを使用した．以降の表中の記述で，スタイル分類列の表記は画像スタイルを省略して表記している，それぞれ MangaG: Manga_gray, MangaR: Manga_rgb, NovelG: Novel_gray, PhotoA: Photo_all, MiscA: Misc_all を意味する．PSNR の単位は dB である．

表 3-8. lanczos, SRCNN, upconv7（ours）で比較（×2 倍拡大）

スタイル分類	lanczos		SRCNN		upconv7	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MangaG	35.639	0.864	36.287	0.891	38.041	0.938
MangaR	36.957	0.895	37.619	0.908	39.832	0.954
NovelG	26.224	0.961	28.101	0.975	36.577	0.997
PhotoA	36.808	0.970	37.388	0.977	39.198	0.978
MiscA	27.128	0.944	27.239	0.954	33.322	0.976

3.5.4 ノイズ削減の学習

画像データの圧縮方式は複数存在するが，データ容量を大きく削減する場合には非可逆圧縮を採用することが多く，その圧縮・伸長の工程が lossy なので伸張しても 100%完全に元通りのデータには戻らない．そのため伸張時に人工的なノイズ（以下，圧縮ノイズと記述）が生成される．圧縮率がそれほど高くない場合にはその圧縮ノイズは肉眼ではほとんど気にならないが，圧縮率が高い場合には肉眼でも画像の劣化がはっきりわかるようになる．

入力画像に含まれる圧縮ノイズ量は画像毎に様々であり，また，目視で入力画像に含まれる圧縮ノイズ量の多少を相対的に判定することは可能だが，その絶対量を判定することはきわめて難しい．そこで同一の高解像度教師画像に対し，ノイズ含有量が異なる学習用低解像度画像のグループを複数用意し学習をおこなうことで，学習時のデータセットの圧縮ノイズ含有量のグループに応じた段階的なノイズ削減を学習する方式とした．

深層学習による超解像処理では，圧縮ノイズを挿入して得られた低画質画像を入力に用いた場合でも元のノイズのない高画質画像の復元・生成もできることが期待できる．E 社では，配信用画像データの保存時の圧縮には jpeg および VQ(Vector Quantization)に基づく独自の非公開アルゴリズ

ムを使用している. jpeg の場合は圧縮率を制御する品質パラメータが存在しているように, VQ 圧縮を使用した独自方式でも同様に zav という誤差に関するパラメータを使って圧縮率を制御することができる.

そこで超解像処理の学習で使用した 5 種類の画像スタイル毎に, 圧縮ノイズを含まない低解像度の入力画像を元画像として使用し, 圧縮アルゴリズムを 2 種類(jpeg・VQ), 圧縮画像の品質が(Low・High)の 2 種類の計 4 種類の組み合わせと, 両方の圧縮アルゴリズムと圧縮画像品質をミックスした場合の合計 5 種類のノイズが付加された低解像画像の学習画像セットを新たに作成し, 画像スタイル毎にノイズを含まない学習セットおよびノイズを含む 5 種類の学習セットの計 6 グループを作成し, 合計で 30 種類の低解像度の学習セットを作成した. これらの画像セットの分類を表 3-9 に記述する. ノイズ種別の列の Src はノイズ削減を伴わない超解像処理のみの画像セットを意味する.
これらのノイズを含む低解像画像セットと高解像画像のペアで学習することで, 圧縮ノイズが付加された低解像度画像を使って圧縮ノイズの無い高品質の超解像処理画像を再生する SR+NR-CNN の学習を行った.

表 3-9. ノイズ種別分類

ノイズ種別	内　　容
Src	圧縮ノイズを含まないまたは検出できないクリアな画像
Jp_Lo	JPEG 品質 85~95(一様乱数)で一度圧縮し伸長した画像
Jp_Hi	JPEG 品質 65~85(一様乱数)で一度圧縮し伸長した画像
Vq_Lo	VQ ZAV=100 で一度圧縮し伸長した画像
Vq_Hi	VQ ZAV=500 で一度圧縮し伸長した画像
mix	JPEG 品質 65~85 で圧縮した画像を伸長し, さらに VQ ZAV=500 または ZAV=100 で圧縮し伸長した画像

ノイズ量分類器のための, ノイズ含有データセットは, ノイズが含まれていない元画像を実際に圧縮してすぐに伸長した画像を使用した. ノイズの含有量を調整するために, 圧縮時に設定する圧縮パラメータの値を用いたが, このパラメータの数値だけでは圧縮ノイズの含有量は定量的には決まらず, 元画像の複雑さの方が大きく影響し, 圧縮パラメータの数値が支配的とは言えないため, 使用していた 2 種類のコーデックに対して, それぞれのノイズ含有量を分別するのに, 明らかにノイズが多いか/少ないかを目視でわかる範囲で区分し, HIGH/LOW の 2 分類とした.

また, 画像データは特に意識せずとも JPEG などのコーデックを使って保存されることが一般的なため, それらの画像を超解像拡大する場合には必ず圧縮ノイズの問題が発生するのでマンガ固有の問題ではなく, 画像一般の問題であると言える.

3.5.5　　画像スタイル分類器

画像スタイル分類器 S-CNN の構成を**図 3-2** に示す．構成は画像分類では定番の VGG Net[Simonyan 14] と呼ばれるモデルを参考に精度をそれほど落とさずに少しでも高速実行ができるように層数を調整した．VGG Net と同様に，フィルタは，3×3 のフィルタのみを使用している．

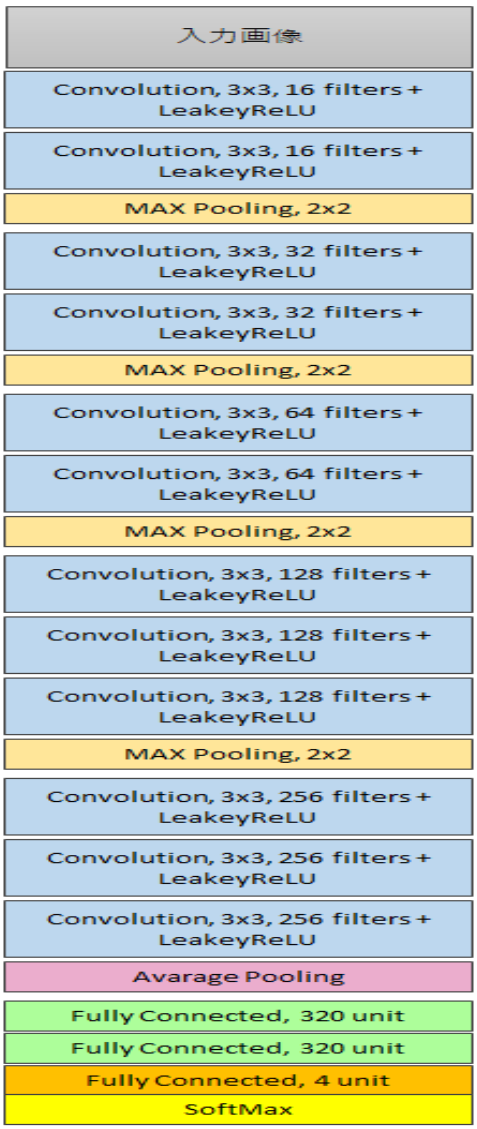


図 3-2. 画像スタイル分類器（S-CNN）のネットワーク構成

5 種類の画像スタイルは，Manga_gray，Manga_rgb，Novel_gray，Photo_all，Misc_all だが，スタイル分類器内部ではモノクロ・カラー共通で Manga とだけ判定する 4 分類の構成になっており，Manga の分類が確定した後に別途色情報の有無を判別して Manga_gray と Manga_rgb に再分類する構成にした．S-CNN は前述のページごとに画像スタイル分類をしたベース画像セットから画像スタイル毎にランダムに抽出した 2000 枚の画像で学習した． S-CNN のスタイル分類の正解率を**表 3-10** に示す．

表 3-10. S-CNN のスタイル分類の正解率

画像スタイル	正解率
Manga（gray or rgb）	99.15%
Novel_gray	100.0%
Photo_all	99.17%
Misc_all	99.56%
Average	99.47%

超解像処理時にこのスタイル分類器で入力画像の画像スタイルを自動選択するための前処理器として使用し，その結果に応じて最適な CNN のパラメータに切り替えることができるような画像スタイルの自動選択機構を実現する．

3.4.6 ノイズ特性量分類器

目視で入力画像に含まれる様々な圧縮ノイズ量の多少をある程度判定することは可能だがその絶対量の判定や圧縮アルゴリズムの違いまでを判定することは極めて難しく，入力画像ファイル情報が示す圧縮フォーマットは，ファイル化時点の圧縮フォーマットを意味し，その画像に含まれている圧縮ノイズ情報をかならずしも反映していない．入力された画像そのものから潜在的に含まれている表 3-9 の 6 種類を分類できるノイズ特性量分類器 N-CNN を開発し最適な SR-NR-CNN を選択する機構が必須である．に N-CNN のネットワーク構成図を示す．

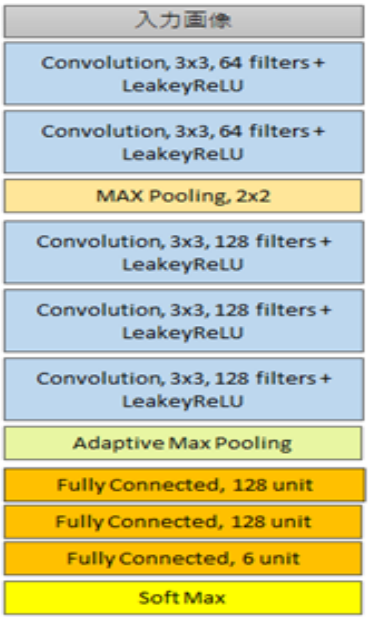


図 3-3. ノイズ特性量分類器（N-CNN）のネットワーク構成

N-CNN で画像内の圧縮ノイズ特性量の評価をする場合，圧縮ノイズが少ない部分を評価しても全く意味がないので，画像内で圧縮ノイズが多く含まれていると思われる領域を評価する必要がある．jpeg および VQ ベースのアルゴリズムの両方とも，圧縮ノイズは元画像中の輝度値の変化の大き

な領域で多く発生することがわかっているため，ページ画像全体を 96×96pix のサイズの領域に分割し，その分割した各領域内の輝度の標準偏差を求め，その値の上位 16 か所を選択し，そのページ画像のノイズ特徴量を判定する方式とした．

領域分割サイズは 32×32, 64×64, 96×96 の 3 種類の候補で実験を行い，実行時間は 20msec 程度遅かったが，分類性能が一番良かった 96×96 に決定した．それぞれのスタイル（Manga_gray, Manga_rgb, Novel_gray, Photo_all, Misc_all）におけるノイズ特性量分類器の正解率を**表 3-11** に示す．

表 3-11. それぞれのスタイルにおけるノイズ特性量分類器の正解率

（Manga_gray, Manga_rgb, Novel_gray, Photo_all, Misc_all）

・Manga_gray

ノイズクラス	Src に分類	Jp_Lo に分類	Jp_ Hi に分類	Vq_Lo に分類	Vq_Hi に分類	Mix に分類	正解率(%)
Src	363	2	4	0	0	0	98.37
Jp_Lo	0	363	6	0	0	0	98.37
Jp_Hi	0	12	357	0	0	0	96.75
Vq_Lo	0	1	0	364	0	4	98.65
Vq_Hi	0	0	0	0	365	4	98.92
mix	0	0	0	1	7	478	98.35

・Manga_rgb

ノイズクラス	Src に分類	Jp_Lo に分類	Jp_ Hi に分類	Vq_Lo に分類	Vq_Hi に分類	Mix に分類	正解率(%)
Src	343	1	0	0	0	0	99.71
Jp_Lo	3	334	7	0	0	0	97.09
Jp_Hi	2	12	330	0	0	0	95.93
Vq_Lo	0	0	0	344	0	0	100.00
Vq_Hi	0	0	0	0	344	0	100.00
mix	0	0	0	32	52	308	78.57

・Novel_gray

ノイズクラス	Src に分類	Jp_Lo に分類	Jp_ Hi に分類	Vq_Lo に分類	Vq_Hi に分類	Mix に分類	正解率(%)
Src	389	0	0	0	0	0	100.00
Jp_Lo	0	388	1	0	0	0	99.74
Jp_Hi	1	3	305	0	0	0	99.97
Vq_Lo	0	0	0	388	0	1	99.74
Vq_Hi	0	0	0	0	388	1	99.74
mix	0	0	0	2	8	468	97.91

・**Photo_all**

ノイズクラス	Src に分類	Jp_Lo に分類	Jp_ Hi に分類	Vq_Lo に分類	Vq_Hi に分類	Mix に分類	正解率(%)
Src	289	10	1	0	0	0	96.33
Jp_Lo	1	292	7	0	0	0	97.33
Jp_Hi	0	9	291	0	0	0	97.00
Vq_Lo	0	0	0	298	0	2	99.33
Vq_Hi	0	0	0	0	299	1	99.67
mix	0	0	0	12	72	278	76.80

・**Misc_all**

ノイズクラス	Src に分類	Jp_Lo に分類	Jp_ Hi に分類	Vq_Lo に分類	Vq_Hi に分類	Mix に分類	正解率(%)
Src	346	1	0	0	0	0	99.71
Jp_Lo	0	341	6	0	0	0	98.27
Jp_Hi	0	17	330	0	0	0	95.10
Vq_Lo	0	0	0	347	0	0	100.00
Vq_Hi	0	0	0	1	344	2	99.14
mix	0	0	2	2	28	380	92.23

3.6　全自動超解像処理システムの性能評価

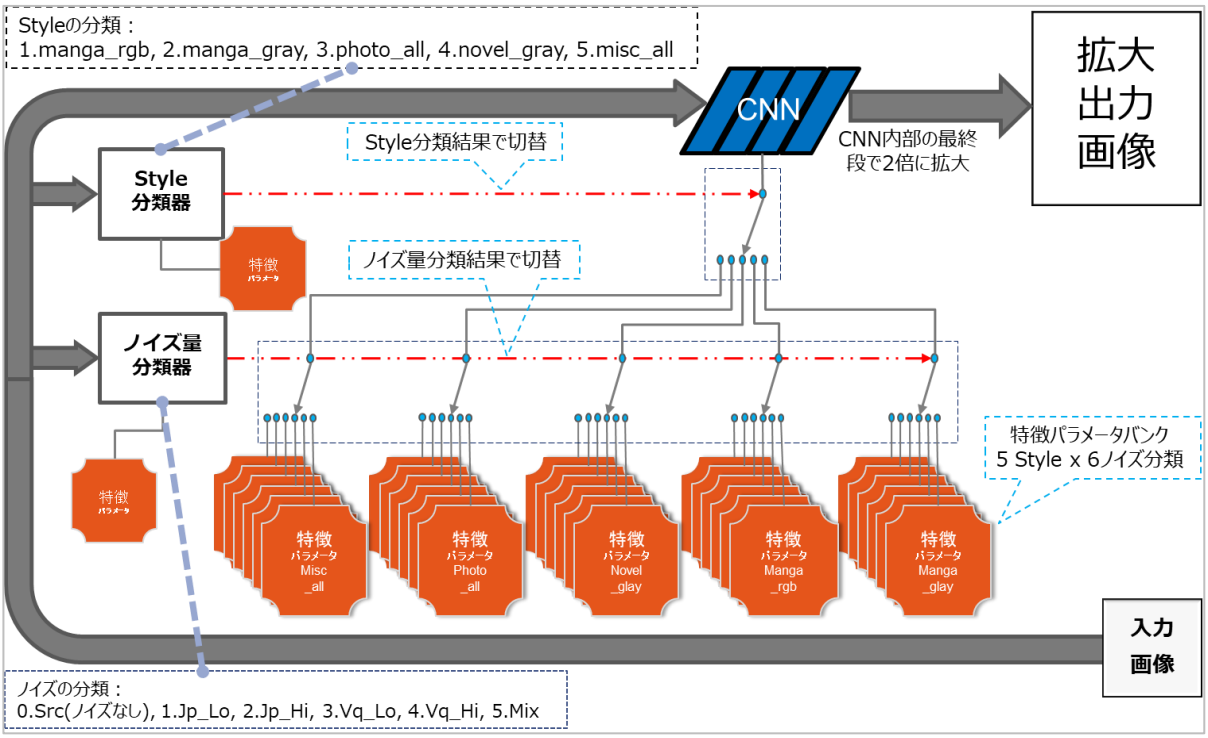


図 3-4. 全自動超解像拡大システム(概念図)

3.6.1 性能評価の概要

総合性能を確認するために 5 つの画像スタイル全てに，それぞれ6つのノイズ種別(ノイズを含まない画像，Jp_Lo，JP_Hi，Vq_Lo，Vq_Hi，Mix)の圧縮ノイズを含む画像を均等に 380 枚ずつ抽出した合計 11,400 枚の評価画像セットを用意した．

全画像を自動スタイル分類器及び自動ノイズ特性量分類器の結果に基づき超解像処理を行った．

図 3-4 示す超解像拡大システム全体を用いた(Our System) 及び lanczos および SRCNN で拡大した結果を**表 3-12** に示す．Our System が PSNR, SSIM 共に優れている．

表 3-12. 図 3-4 示す超解像拡大システム全体を用いた結果の正解率

スタイル分類	Lanczos		SRCNN		Our System (upconv7)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
全て MIX	25.697	0.910	27.109	0.922	**30.561**	**0.956**

3.6.2　　ノイズを含まないデータの超解像ノイズ削減

前述の非公開の評価画像セットから，各画像スタイルでノイズを含まないテスト画像を 380 枚抽出し，画像スタイル分類とノイズ分類を固定しすべての組み合わせで超解像処理を行った．**表 3-9** に示す 6 つのノイズ種別それぞれの圧縮ノイズ削減を学習した超解像 CNN の入力データとしてノイズのない画像を使用した場合の 2 倍拡大超解像の結果の PSNR（dB）を**表 3-13** に示す．いずれの画像スタイルでもノイズ削減せずに超解像を行った場合，表中で『Src：なし』の列が一番良い PSNR 値になった．

表 3-13. 表 3-9 に示す6つのノイズ種別それぞれの圧縮ノイズ削減を学習した超解像 CNN の入力データとしてノイズのない画像を使用した場合の 2 倍拡大超解像の PSNR(dB)

画像スタイル	Jp_Lo	Jp_Hi	Vq_Lo	Vq_Hi	Mix	Src：なし
MangaG	27.064	26.877	27.087	26.734	26.33	**38.041**
MangaR	30.502	30.246	30.386	30.108	29.92	**39.832**
NovelG	36.369	35.922	26.641	26.314	35.54	**36.577**
PhotoA	39.051	38.701	38.952	38.471	38.29	**39.198**
MiscA	32.822	32.505	32.862	32.448	32.54	**33.322**

3.6.3　　ノイズを含んだデータの超解像ノイズ削減

画像スタイル毎に抽出された性能検証用の 380 枚の画像から生成したノイズを含まない低解像画像セット，および，その低解像画像セットをベースにそれぞれ5種のノイズ区分 Jp_Lo, JP_Hi, Vq_Lo, Vq_Hi, Mix 相当のノイズを挿入した低解像テスト画像セットを生成し，それぞれの画像スタイルにおいて，手動で SR+NR-CNN に対し前記 5 種のノイズ区分のノイズ削減量をそれぞれ設定して超解像処理と同時のノイズ削減を行った結果の PSNR 値（単位は dB）をテスト画像セット別に**表 3-14** に示す．また，それぞれのテスト画像セットの表中に，超解像処理のみを行いノイズ削減を行わなかった場合をノイズ削減なしの意味で『なし』と表記した列に示す．

いずれの行でも，Jp_Lo, JP_Hi, Vq_Lo, Vq_Hi, Mix の各ノイズ特性量に合致したノイズ特性量の CNN を使用して超解像処理を行なった場合（太字表記の列）が最大の PSNR 値になっている．また Δ の表記の列は各性能評価用画像セットの表中の太字の列と「なし」の列の値の差を示してあり，ノイズ削減の効果を数値化したものと考えることができる．

また，**表 3-13** のノイズを含まないテスト画像の場合のノイズ削減なしの列の値はノイズを含まない画像の超解像結果なので，ノイズ削減がほぼ完ぺきにできた場合の PSNR の目標値と考えることもでき，ノイズ削減の効果にはまだ改善の余地が残っていると言える．

表 3-14. 圧縮ノイズ削減の性能(x2)(dB)

(1)Jpeg_Low(Jp_Lo)相当のノイズを付加したデータ

スタイル	Jp_Lo	Jp_Hi	Vq_Lo	Vq_Hi	Mix	なし	Δ
MangaG	**26.665**	26.602	26.583	26.461	26.54	26.408	0.257
MangaR	**29.911**	29.838	29.734	29.730	29.68	29.507	0.405
NovelG	**35.063**	34.817	34.704	34.481	34.84	33.579	1.484
PhotoA	**38.022**	37.765	37.885	37.750	37.75	37.751	0.271
MiscA	**32.389**	32.280	32.204	32.255	32.16	31.782	0.607

(2)Jpeg_High(Jp_Hi)相当のノイズを付加したデータ

スタイル	Jp_Lo	Jp_Hi	Vq_Lo	Vq_Hi	Mix	なし	Δ
MangaG	25.509	**25.824**	25.227	25.470	25.80	25.026	0.799
MangaR	28.444	**28.863**	28.033	28.439	28.83	27.716	1.147
NovelG	32.203	**33.118**	31.019	32.285	32.87	30.344	2.774
PhotoA	36.434	**36.603**	35.698	36.369	36.58	36.027	0.575
MiscA	30.513	**31.072**	29.904	30.586	31.03	29.465	1.607

(3)VQ_Low(Vq_Lo)相当のノイズを付加したデータ

スタイル	Jp_Lo	Jp_Hi	Vq_Lo	Vq_Hi	Mix	なし	Δ
MangaG	26.888	26.734	**26.956**	26.712	26.69	26.809	0.146
MangaR	29.996	29.853	**30.170**	30.021	29.80	29.507	0.664
NovelG	35.705	35.311	**36.013**	34.798	35.70	35.125	0.888
PhotoA	37.984	37.697	**38.179**	37.971	37.82	37.837	0.342
MiscA	32.706	32.435	**32.862**	32.599	32.38	32.516	0.347

(4)VQ_High(Vq_Hi)相当のノイズを付加したデータ

スタイル	Jp_Lo	Jp_Hi	Vq_Lo	Vq_Hi	Mix	なし	Δ
MangaG	26.151	26.170	26.180	**26.428**	26.33	25.926	0.502
MangaR	29.164	29.198	29.224	**29.424**	29.29	28.712	0.712
NovelG	32.400	33.118	32.735	**34.492**	34.23	32.089	2.402
PhotoA	36.371	36.313	36.151	**36.721**	36.55	36.151	0.570
MiscA	31.544	31.543	31.543	**31.825**	31.03	31.117	0.708

(5)Mix 相当のノイズを付加したデータ

スタイル	Jp_Lo	Jp_Hi	Vq_Lo	Vq_Hi	Mix	なし	Δ
MangaG	25.74	25.92	25.65	25.87	**26.04**	25.45	0.59
MangaR	28.63	28.90	28.50	28.81	**29.01**	28.06	0.95
NovelG	31.68	32.60	31.30	32.73	**33.38**	30.86	2.52
PhotoA	36.03	36.10	36.03	36.19	**36.39**	35.73	0.66
MiscA	30.29	30.49	30.04	30.41	**30.61**	29.71	0.90

3.7　全自動超解像処理システムの構成

全自動超解像処理システムの構成図を**図 3-4** に示す．角が内側に丸く削られている同じ大きさの 30 個（重なっている 6 個×5 ブロック）の四角形は，全て異なる条件の画像セットを使って学習した SR+NR-CNN のパラメータセットを示す． ノイズ特性量分類器の結果によって 30 個のパラメータセットに接続されている 5 連のスイッチが切替わり，Style 分類器の結果によって SR+NR-CNN に近いスイッチが切り替わる仕組みになっている．右下の小さな四角形部分に低解像度画像を入力すると，画像スタイル分類器，ノイズ特性量分類器の結果に応じて選択された最適なパラメータセットで SR-NR-CNN が動作し，右上の大きな四角形部分に超解像処理と同時に圧縮ノイズが削減された高品質な 2 倍の拡大画像が出力される．公開可能な Manga109 [Manga 15], [Matsui 16] , [Ogawa18] の中の漫画画像の拡大結果を**図 3-5**, **図 3-6**, **図 3-7**, に示す．

本システムは NVIDIA の GPU を搭載した linux 上の Torch7 ベースの機械学習環境で深層学習を行い，得られた CNN のパラメータを NVIDIA の GPU 搭載の windows 10 PC 上で動作する caffe-windows に移植し windows 上で実行環境を構築した．参考例として 760×1200pix のカラー漫画画像を upconv7 で 1520×2400 に拡大した場合，1 枚当たり画像スタイル分類と圧縮ノイズ分類処理で平均 32msec，超解像処理が平均 401msec（204 枚の画像の 3 回平均）だった．実行環境は CPU: lntel(R) Core(TM) i7-7700 (3.60GHz)，GPU: NVIDIA GeForce GTX 1070 (1.84 GHz)．8192 Mbyte Memory である．

3.7.1　　公開データを使用したベンチマーク結果

超解像に関する複数の論文で引用され，且つ公開されている写真画像データセット，Set5，Set14 , BSD100，Urban100 を使って本システムの **Photo_All** モードで 2 倍拡大時の性能を計測し，そのベンチマーク結果を**表 3-15** の **up7_OURs_Photo_All** 行にまとめた．

PSNR，MSSIM の計算における端数処理などの扱い方が異なることによる誤解を避けるため，公開されている拡大結果画像セット[Huang 15]（Set5，Set14，BSD100，Urban100）内の SRCNN と bicubic の結果画像を使い，同一の計算プログラムを使用し，著者が再計算した．

表 3-15．公開データ（Set5, Set14, BSD100, Urban100）を使用したベンチマーク結果

・Set5：(x2)

Model	MSE	PSNR（dB）	MSSIM
up7_公開版_Photo	15.390	37.009	0.9569
up7_OURs_Photo_All	17.068	36.682	0.9551
SRCNN	19.208	36.142	0.9513
Mgk_lanczos	35.255	34.282	0.9366
bicubic	40.863	33.647	0.9304

・Set14：(x2)

Model	MSE	PSNR（dB）	MSSIM
up7_公開版_Photo	50.569	32.471	0.9131
up7_OURs_Photo	52.948	32.223	0.9097
SRCNN	57.109	31.809	0.9076
Mgk_lanczos	74.450	30.528	0.8857
bicubic	81.776	30.074	0.8761

・**BSD100：(x2)**

Model	MSE	PSNR（dB）	MSSIM
up7_公開版_Photo	65.234	31.627	0.8896
up7_OURs_Photo	67.995	31.389	0.8855
SRCNN	72.068	31.059	0.8840
Mgk_lanczos	90.159	29.917	0.8554
bicubic	97.085	29.554	0.8438

・**Urban100：(x2)**

Model	MSE	PSNR（dB）	MSSIM
up7_公開版_Photo	87.489	30.262	0.9166
up7_OURs_Photo	97.123	29.633	0.9069
SRCNN	118.962	28.592	0.8930
Mgk_lanczos	163.150	27.059	0.8562
bicubic	178.081	26.650	0.8442

PSNR の計算は MATLAB と互換の以下の計算式で RGB から Y に変換した値を採用した.

Y = 0.25679R + 0.50413G + 0.097906B + 16

Mgk_lanczosはImageMagickによるlanczosフィルタを使用して拡大した結果，up7_公開版_Photoは公開版 waifu2x の写真モード，up7_OURs_Photo は本システムの upconv7 の写真スタイル(Photo_All)でのそれぞれの拡大結果を示す. この結果から，本稿のシステムの土台となる超解像の性能は SRCNN と同等かそれ以上の性能を有することがわかった.

本システムにおける写真ジャンルでは電子書籍用の写真画像を画像スタイルの一分野として学習データセットを生成し新たに学習したが，オリジナルの waifu2x は公開された写真データを使って学習し，その学習済みモデルデータを公開している.

電子書籍用の画像データの多くは印刷された画像をスキャナなどで読み取ってデジタイズされたデータなので，たとえ同じ“写真”というジャンルの画像であっても，印刷された画像をスキャナ等によって取り込んでデジタイズした画像データと，カメラで撮影した生の画像データとでは異なった特性をもっているはずである.

この写真画像の学習データの差が CNN の特性の差となり，拡大時の PSNR の差として表れていると推測され，CNN の特徴抽出の感度は極めて高いと推察できる.

本システムでは電子書籍などで多く用いられる，スキャンされた画像を拡大することに限定して開発したため，あらゆるジャンルの画像を拡大するシステムを構築する場合には，学習データセットの収集に関しては充分考慮する必要があると考えられる.

3.8 考察

3.8.1 特徴量抽出の感度

本システムでは様々な種類の画像を拡大するために，複数の画像スタイルに分けて学習することで，システムに入力するだけで，自動で最適な超解像拡大をすることが可能になり，当初の目的は達成できた．

しかし，パラメータの種類が非常に多くなってしまったことにより，複雑性が増したことはデメリットとも言える．しかも，より高品質が望まれる用途に使用する場合は，将来，万能のパラメータが使えるCNNの構成が発明されるまでは，もっと多くの画像スタイルに細分類して学習し直すか，あるいは特定のジャンル画像のみの別の詳細なグループを作るなどのさらなる工夫が必要と考えられる．

3.8.2 プロダクトの不安定さとその管理

本システムのようなDeep learningによって得られたライブラリを利用して構築されたシステムを実業で使用する場合に考慮すべき問題について記載する．

本稿のように画像を拡大するという作業に限って言えば，特異な結果が得られたとしても，殆どの場合は全く問題がないかあっても軽微なものと想定される．現在はまだDeep learningの技術が充分に成熟していないこともあり，学習したシステムが内包している（かもしれない）特異点に対する不確定な挙動に関して完全には解析できていない．

そのため，特異な結果が得られた場合に備えて，その時の現象を後から解析できるような仕組みを同時に組み込んでおくことが望ましいと考えられる．

個人が本人の楽しみのために画像を拡大する場合は，その結果が直接的に，使用者の生命や財産等に関わることはなく全く問題にはならない．あくまで仮定だが，不鮮明な画像を拡大して得られた結果をもとに，何らかの病気の診断に使ったり，あるいは，作物や生産品の出荷検査に使われたりすることがあるかもしれない．そのような拡大された画像を使って判断する場面で，全く予想しない拡大画像になってしまった場合は，重大な問題を引き起こすと思われる．

電子書籍事業で利用する場合，極端な話ではあるが，システムが低解像画像から全く予期しない拡大画像を生成し，結果として全く新しい別の画像が得られた場合は，最悪の場合には会社の信用を著しく毀損してしまう可能性も無いとは言えない．そのようなケースに備えて，少なくともその拡大変換時に使われたパラメータだけは log などに記録として残しておき，後から解析ができるような最低限の仕組みは用意しておくべきである．また，このシステムはソフトウエアプロダクトであるので実行環境のオペレーティングシステムや開発環境のバージョンに依存したプロダクトである．

一般的な，バージョン管理では，実行環境（OS 及び関連基本システム）のバージョン，プロダクトのバージョン，必要なライブラリ（dl 等）のバージョンなどが管理対象であるが，AI を使ったプロダクトの場合は，その AI ネットワークモデルのバージョン及び，学習によって生成されたパラメータ群に関して，適切に管理できる仕組みを必要とする．特に，本システムではパラメータの種類が非常に多くなってしまったため無視できない問題である．これはかなり厄介な話であり，特に開発中の AI のエンジンに関しては，同一の学習データセットを使っていても，学習を行うたびにそのパラメータは新しいものが生成されるため，これが組み合わせになるとさらに複雑になる．

そこで我々は，ネットワークパラメータセット群及び特徴データ群を階層構造化し，最下層のパラメータから順番にハッシュ化し，最後に最上層でシステム全体のハッシュを作成しリリース時にはこの最上層のハッシュをシステムの代表ハッシュとするようなバージョン管理をしている．

パラメータセット群の構成を逆解析する場合は，最上層から順に最下層に向かって階層構造を下っていき，最下層の特徴データまでを順に特定できる構造にした．

3.9 まとめ

先行例の多くでは低解像度画像のサンプルに人物，風景，建築物，動物，植物，乗り物等の写真画像を使っており，手書きの線画が主体で活字文字が多く混在している漫画画像，文字主体の画像や雑誌のように全てが混在している画像などを拡大した場合の性能については言及されていない．様々な画像スタイルを分類し，複数の画像セットを使って CNN を深層学習することで，漫画，文字主体の画像や雑誌のような人工画像であっても高品質な超解像処理画像が得られる仕組みを構築できた．

さらに，ページ画像毎に圧縮ノイズ量と画像スタイルが共に異なっている書籍も画像スタイル分類器による画像スタイルと，圧縮ノイズ特性量分類器によるノイズ特性量を自動分類した結果で最適な特性の SR+NR-CNN を選択する方式にすることで電子書籍の様々な画像全体に対して自動で超解像処理と同時のノイズ削減が可能になり画像品質の目的値を達成することができた．

本システムの基本原理は，辞書方式の超解像拡大の辞書部分を CNN に置き換え，低解像度の画像入力に対応した最適な高解像度画像を推定する構造と考えることができる．従って，辞書の推定品質の向上のために，現在のシステムで使用している upconv7 よりも広範囲の表現力を持った CNN 構造に置き換えることと，学習に使用した低解像度と高解像度の画像ペアの種類をさらに増やし，画像のバリエーションを増やすことで，さらなる精度の向上は期待できると思われる．

本システムでは，超解像処理の実現に upconv7 を使って構築したが，より高品質な CNN の使用が可能な場合でも，同様の学習画像セットをグループ化して別々に学習し分類器で最適な CNN に切り替えるという本システム構築の手法は有効であり，その場合には，さらに総合的な品質の向上が期待できると考えられる．

ARMS　@加藤 雅基

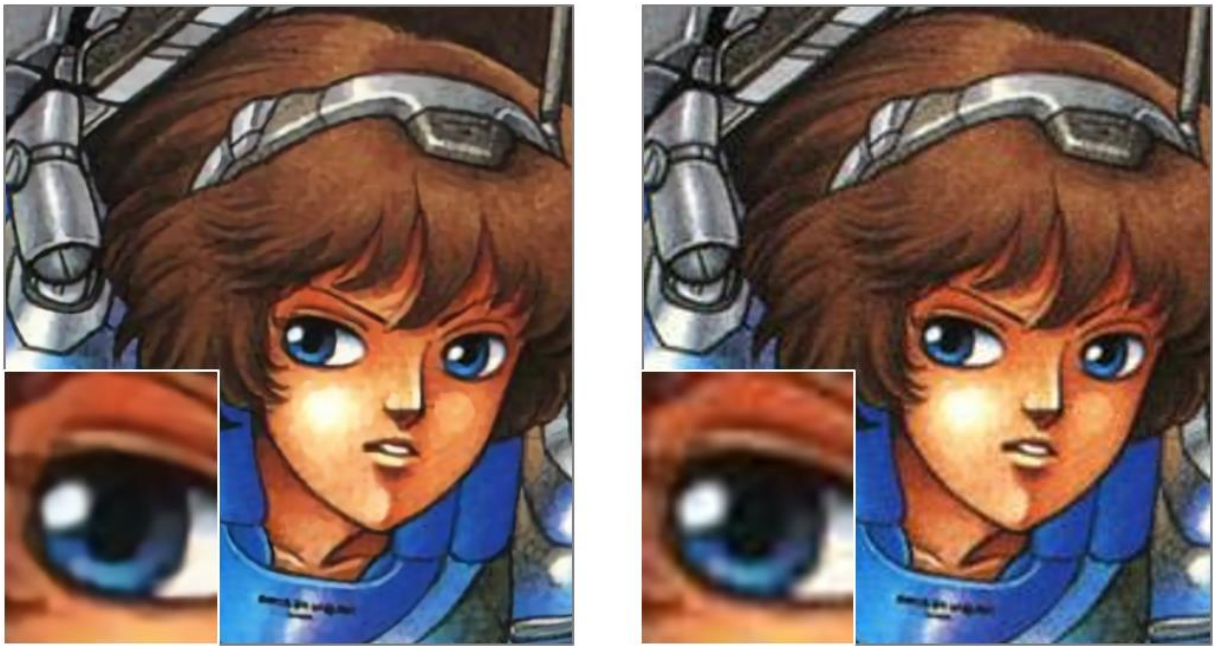


図 3-5.（左）ノイズ削減機能を備えた超解像の 2 倍拡大画像，（右）lanczos による 2 倍拡大画像，(manga_rgb)

MAD STONE©正木秀尚

図 3-6.（左）ノイズ削減機能を備えた超解像の 2 倍拡大画像，（右）lanczos による 2 倍拡大画像(manga_gray)

MAD STONE©正木秀尚

図 3-7.（左）ノイズ削減機能を備えた超解像の 2 倍拡大画像,（右）lanczos による 2 倍拡大画像(novel_gray)

ゴマブックス版, ショパン名作曲楽譜シリーズ7　スケルツォ第 1 番　ロ短調　Op.20（部分）

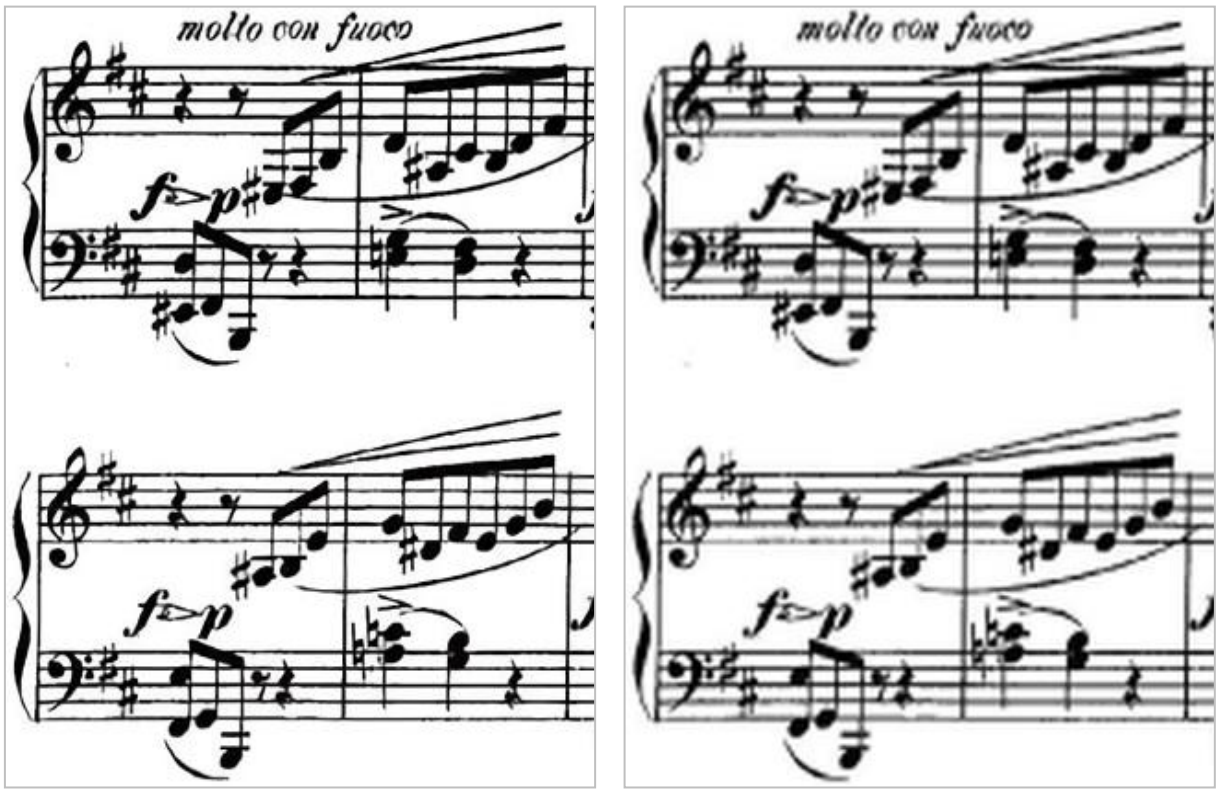


図 3-8.（左）ノイズ削減機能を備えた超解像の 2 倍拡大画像,（右）lanczos による 2 倍拡大画像（ misc_all ）

付録 3-1. 超解像学習データ例

（1）カラーマンガ（manga_rgb-train-tiff）

（2）カラーマンガ（manga_rgb-train-bmp）

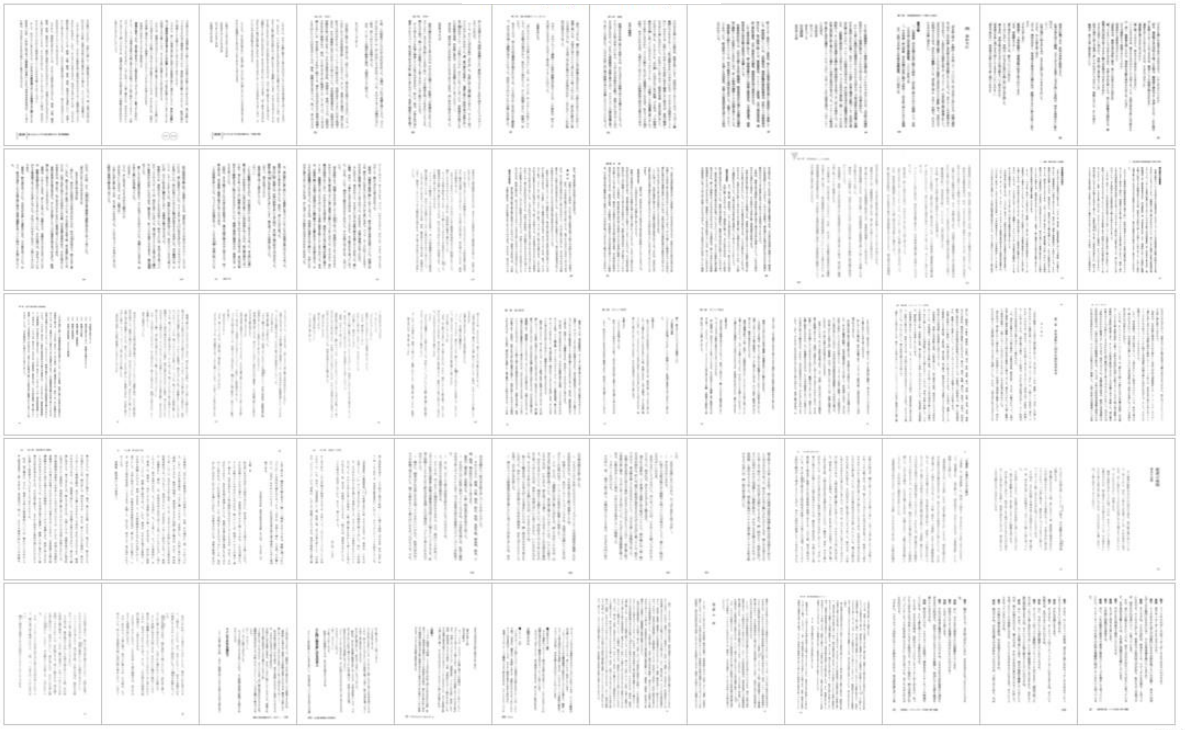
(3) モノクロマンガ (manga_gray-train-tiff)

(4) モノクロマンガ (manga_gray-train-bmp)

(5)小説(novel-train-tiff)

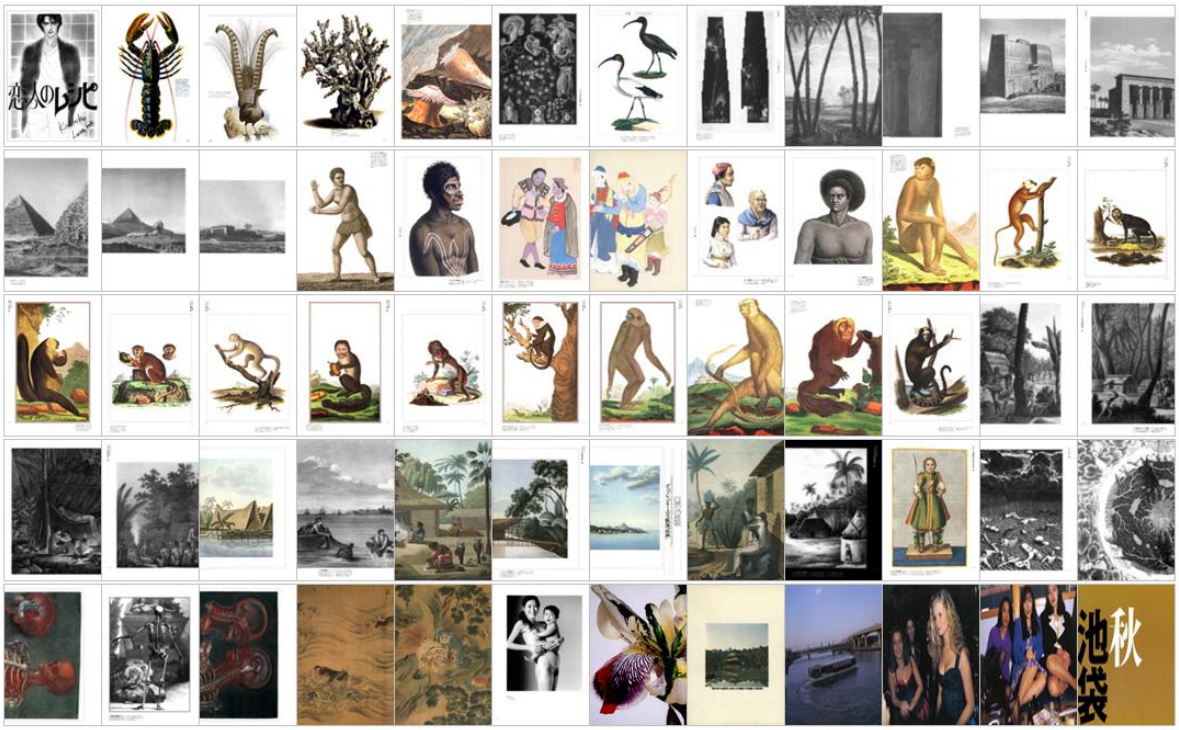
1/67

(6)小説(novel-train-bmp)



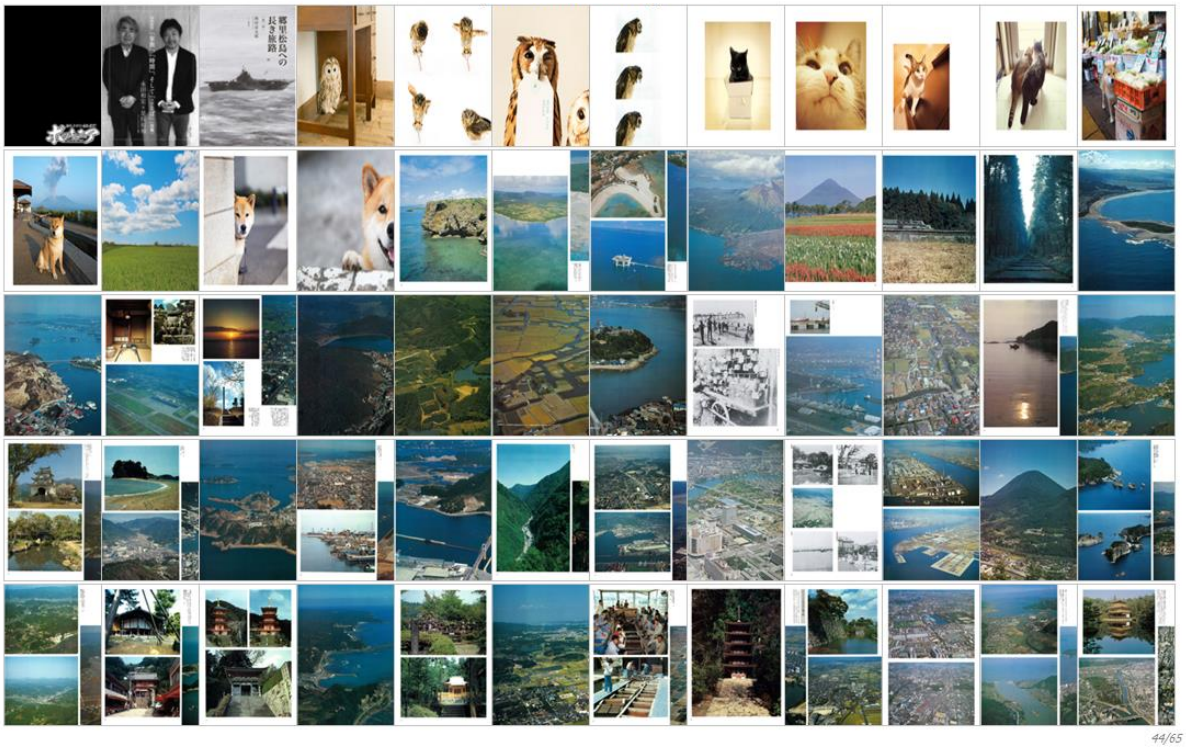
1/67

（7）**写真**（photo-train-tiff）



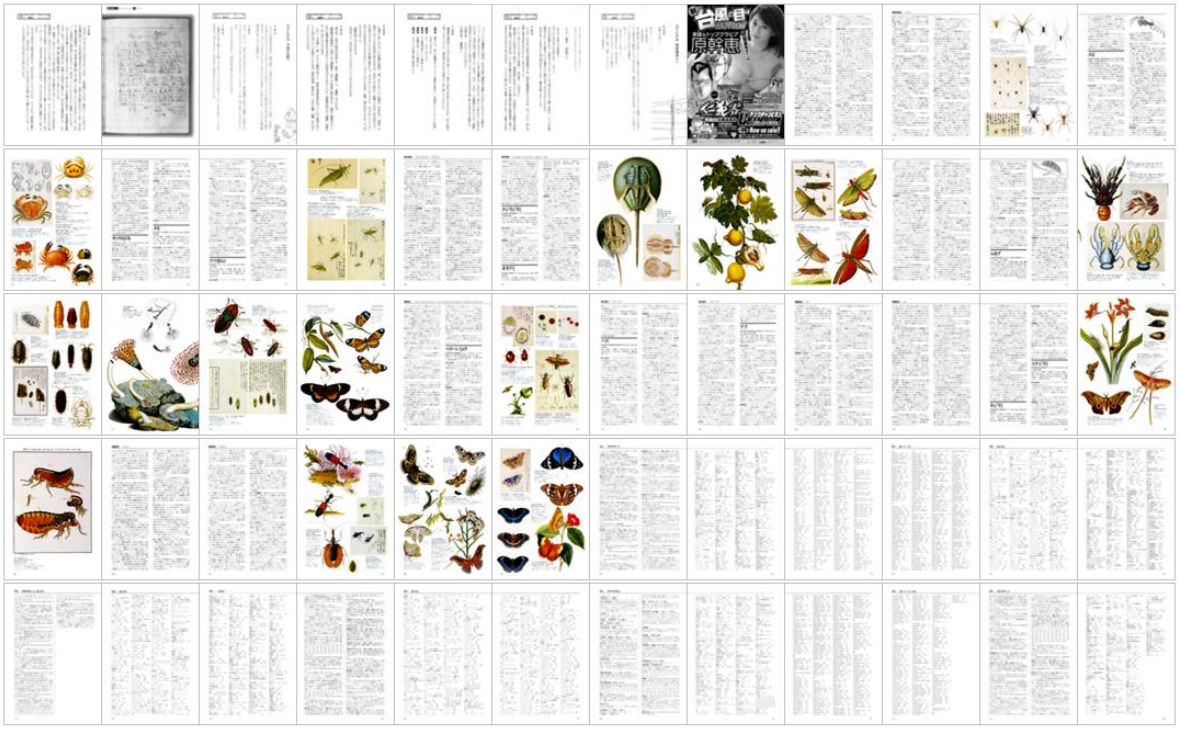
1/63

（8）**写真**（photo-train-bmp）



44/65

(9) **雑誌, 図鑑, その他** (misc-train-tiff)



1/67

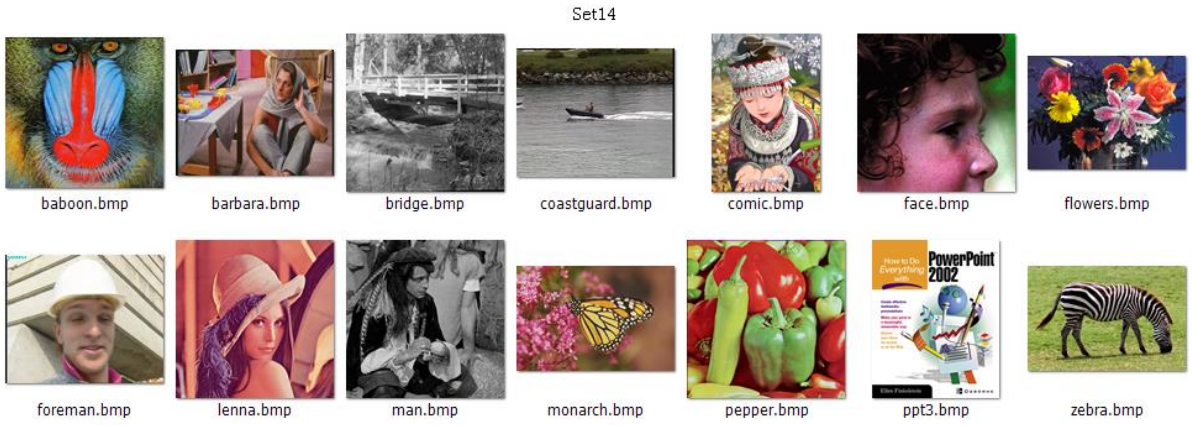
(10) **雑誌, 図鑑, その他** (misc-train-bmp)

1/67

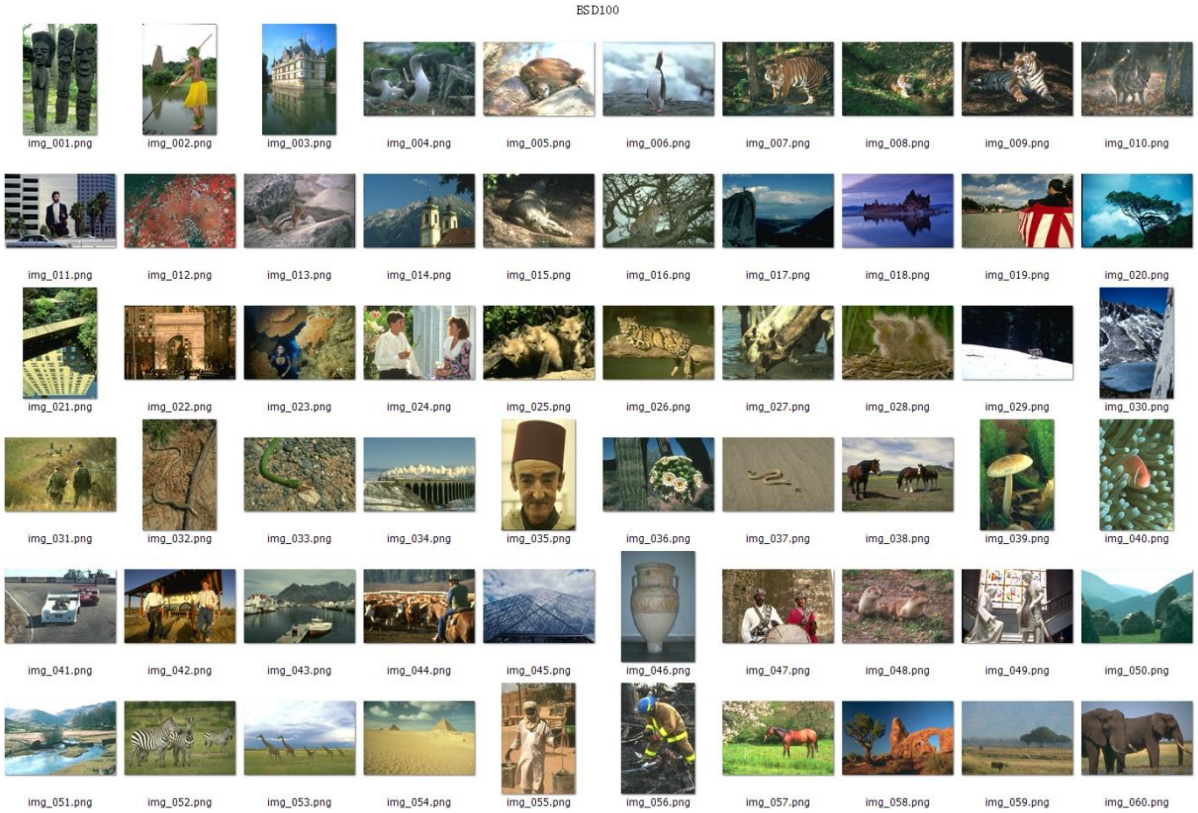
付録 3-2. ベンチマークで使用した公開データ

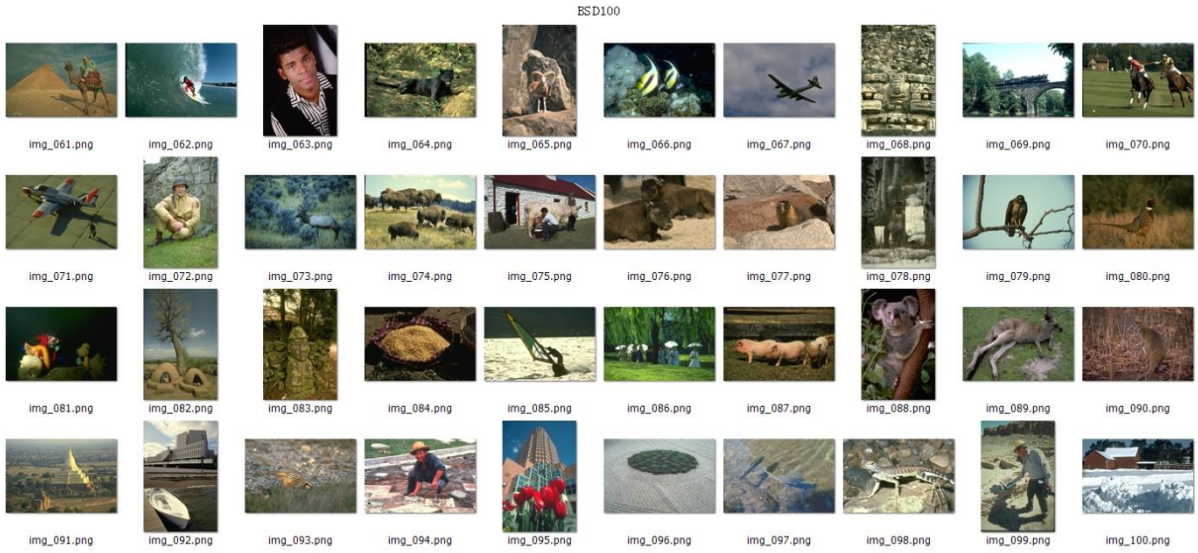
(1) Set5

(2) Set14

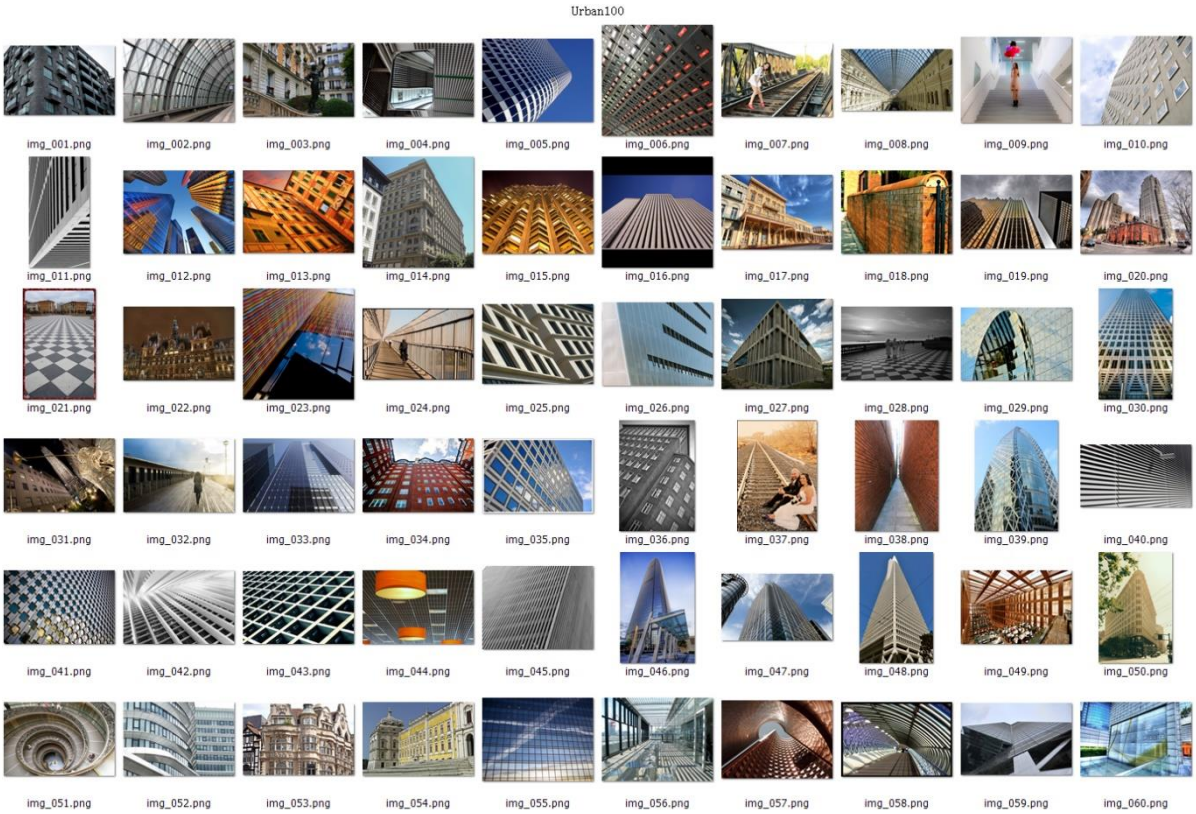


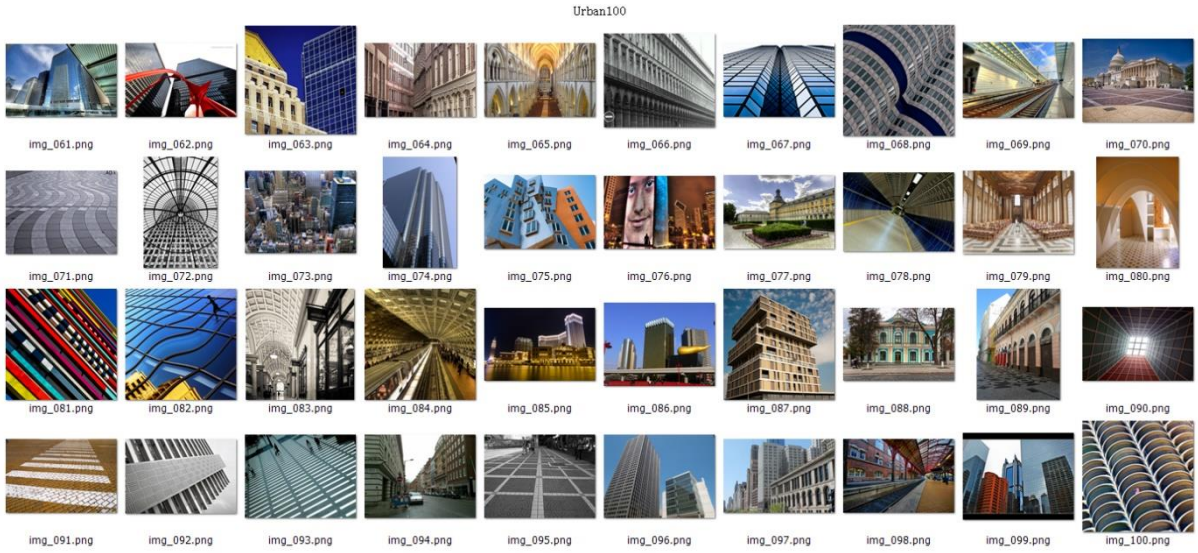
(3) BSD100


BSD100
img_001.png
img_002.png
img_003.png
img_004.png
img_005.png
img_006.png
img_007.png
img_008.png
img_009.png
img_010.png
img_011.png
img_012.png
img_013.png
img_014.png
img_015.png
img_016.png
img_017.png
img_018.png
img_019.png
img_020.png
img_021.png
img_022.png
img_023.png
img_024.png
img_025.png
img_026.png
img_027.png
img_028.png
img_029.png
img_030.png
img_031.png
img_032.png
img_033.png
img_034.png
img_035.png
img_036.png
img_037.png
img_038.png
img_039.png
img_040.png
img_041.png
img_042.png
img_043.png
img_044.png
img_045.png
img_046.png
img_047.png
img_048.png
img_049.png
img_050.png
img_051.png
img_052.png
img_053.png
img_054.png
img_055.png
img_056.png
img_057.png
img_058.png
img_059.png
img_060.png


BSD100
img_061.png
img_062.png
img_063.png
img_064.png
img_065.png
img_066.png
img_067.png
img_068.png
img_069.png
img_070.png
img_071.png
img_072.png
img_073.png
img_074.png
img_075.png
img_076.png
img_077.png
img_078.png
img_079.png
img_080.png
img_081.png
img_082.png
img_083.png
img_084.png
img_085.png
img_086.png
img_087.png
img_088.png
img_089.png
img_090.png
img_091.png
img_092.png
img_093.png
img_094.png
img_095.png
img_096.png
img_097.png
img_098.png
img_099.png
img_100.png

(4) Urban100


Urban100
img_001.png
img_002.png
img_003.png
img_004.png
img_005.png
img_006.png
img_007.png
img_008.png
img_009.png
img_010.png
img_011.png
img_012.png
img_013.png
img_014.png
img_015.png
img_016.png
img_017.png
img_018.png
img_019.png
img_020.png
img_021.png
img_022.png
img_023.png
img_024.png
img_025.png
img_026.png
img_027.png
img_028.png
img_029.png
img_030.png
img_031.png
img_032.png
img_033.png
img_034.png
img_035.png
img_036.png
img_037.png
img_038.png
img_039.png
img_040.png
img_041.png
img_042.png
img_043.png
img_044.png
img_045.png
img_046.png
img_047.png
img_048.png
img_049.png
img_050.png
img_051.png
img_052.png
img_053.png
img_054.png
img_055.png
img_056.png
img_057.png
img_058.png
img_059.png
img_060.png


Urban100
img_061.png
img_062.png
img_063.png
img_064.png
img_065.png
img_066.png
img_067.png
img_068.png
img_069.png
img_070.png
img_071.png
img_072.png
img_073.png
img_074.png
img_075.png
img_076.png
img_077.png
img_078.png
img_079.png
img_080.png
img_081.png
img_082.png
img_083.png
img_084.png
img_085.png
img_086.png
img_087.png
img_088.png
img_089.png
img_090.png
img_091.png
img_092.png
img_093.png
img_094.png
img_095.png
img_096.png
img_097.png
img_098.png
img_099.png
img_100.png

第4章　マンガ画像中のコマ抽出

4.1 はじめに

マンガは，ほぼ日本の国内市場で販売されることを前提に制作され，出版されてきたドメスティックな文化である．文化的に隔離されてきたことで独特のマンガ文化を発展させてきたが，近年日本独自のポップカルチャとして注目を浴びている．

小説は．挿絵などの画像が多数入っているライトノベルと呼ばれるジャンルの作品もあるが，基本的には挿絵画像が全くなくても文字だけで人間や社会などの情景や時間の経過を表現するが，マンガは，主に絵によって人物や背景，音を表現し，人物のセリフは吹き出し内に書かれ，絵と文字の両方によって物語や時間の経過を表現する．

また，映画やテレビなどのコンテンツは固定の長方形形状で表現するのとは異なり，マンガはコマと呼ばれるページ内の複数の小領域ごとに異なる情景を描画し，そのコマの連続で物語の展開や時間の流れを表現する．コマと呼ばれる小領域の形状の多くは四角形だが，その形状や大きさや配置に規定はなく，マンガは作画者が最良と思うアイデア次第でいかようにも描画が可能な大変自由度の高いコンテンツである．デフォルメなどの独特な表現もあるが，ダイナミックなコマ割りや，ドラマティックでスピーディなストーリ展開を有するマンガの中には，その高いエンタテインメント性から，アニメ化され，海外でも広く受け入れられている作品もある．

電子書籍市場の中でも，特にマンガコンテンツの増加に伴う成長はめざましく，より大きな発展が期待されており，さらに多数の読者に読んでもらえるようにするためにも，これからはマンガコンテンツの書誌データ（タイトル，原作者，作画者，出版社，出版日，あらすじなど）だけの検索では実現できない物語の内容に関連した高度な検索のニーズは高まると予想される．例えば特定のエピソードの検索，主人公が活躍するシーン，主人公が争うシーン，特定の人物が登場しているシーン，物語の中の名場面のシーン，登場人物の決めセリフのシーンなどその例を挙げれば枚挙にきりがない．これらの検索を実現するためには検索可能な，登場人物の特徴などに関しての詳細な文章による記述が必要であると思われる．一般的に，小説には画像が無い替わりにそれらの特徴に関する文章による記述が存在するのに対し，マンガにはそのような情報の多くは画像の中に描写された絵として存在しているので，文章で書かれた記述は存在しない．検索を行うためにはそのように絵の中に描かれた特徴（例えば，主人公はひげを生やして，眉毛の濃い，顔の形は四角く，目鼻立ちがはっきりしていて，がっしりとした体格で，長身であるなど）を画像から文章などの形で抽出しておく必要があるが，それが簡単にできる仕組みは存在しないため，そのための研究開発はこれから大いに期待され，必要となる分野である．

マンガはコマを基本要素として，オブジェクト画像，背景画像，吹き出しの中のセリフ，背景画像に埋もれた文章，画像化した文字で表現されるオノマトペ，効果線などの複数の構成要素で組み立てられており，コマごとに構成する要素はそれぞれ異なっている．そのため，マンガの内容を正しく分析・理解するためにも，最初に解決しなければならない基本的な技術要素の一つがコマの正確な抽出であると思われる．

前述のとおり，コマの形状や大きさや配置には明確な規定がなく，さらに，コマの境界線は吹き出しや描画されたオブジェクトなどで頻繁に遮られ消失するため，機械学習を使わない先行研究ではロバストで正確な抽出が見込めず，また CNN を用いた先行研究では，コマ形状を一律に長方形として扱うため正確なコマ情報を扱えず，物体の位置情報が不正確になる可能性があった．本稿では

マンガのページ画像からセグメンテーション CNN を用いてコマ形状を正確に抽出し，CNN を用いた先行研究と同程度の検出精度で抽出できる新たな方法を提案する．

また，このコマ抽出器の応用として，抽出されたコマ画像ごとに，不適切画像が含まれているかどうかを判定する，不適切画像検出について概要を記述する．

4.2 ページ画像からコマの抽出

4.2.1 コマの形状の多様性

マンガは 1 ページの領域を形状と大きさが不定の複数の小領域に分割したコマと呼ばれる領域内に異なる情景を描画し，その連続で物語の展開や時間の流れを表現する独特なメディアである．一般的なコマの形状は四角形が多いが，必ずしも四角形と決められているわけではなく，作画者のアイデアでそれ以外の多角形でも構わない．

一般的には画像オブジェクトはコマ領域をはみ出さずにコマの内部に描画されることが多いが，コマ境界線を跨ぐ画像オブジェクトや，吹出しというセリフ文字を記述した風船型のオブジェクトが，コマ境界線を跨いで描画されることもある．さらに，複数個所のコマの境界線を跨いだ巨大な画像オブジェクトが描画されている例もあり，その描画表現は多様で複雑である．コマの配置は自由で，その形状の最も一般的な形状は長方形（四隅がすべて直角）または，長方形ではない（四隅の 1 つ以上が直角以外の）四角形だが，三角形や，四角形以外の多角形のコマでも構わない．コマ境界線の状態について，以下のように整理できる．

(1) いずれの辺の境界線も消失している部分が無い長方形のコマ（最も一般的），
(2) 1辺以上の辺が垂直線または水平線に対し斜線している四角形の形状のコマ．四角形以外の多角形のコマ，
(3) オブジェクト，吹き出し，オノマトペなどの表音文字などがガター（コマの境界線と隣のコマの境界線，または紙面の端との間の空白の領域）まではみ出し，コマ境界線の一部が消失しているコマ，
(4) オブジェクト，吹き出し，オノマトペなどの表音文字画像などがガターを越えて隣のコマの中まではみ出して描画され，2 つの隣り合うコマの境界線の一部が両方とも消失しているコマ，
(5) コマの一辺以上が裁ち落とし（境界線が存在せず紙面の端まで画像が存在）になっているコマ，
(6) 大きなコマの中に，別のコマの一部または全部が入り込んで境界線はあるがガター部分が存在しないコマ，
(7) 境界線のない（明示的なコマの境界がない）コマ，

などであり．上記の状態が複数，同時に該当している場合もある．

これらの形状のコマを織り交ぜて使うことで，マンガの表現力を効果的に高めることができる．一方，マンガのページからコマを抽出する場合には，(3)～(4)のように，オブジェクトに遮られ描画されていない複数の箇所の境界線や，(5)のように，元々存在しない境界線や，(6)のガターが無い場合の境界線や，(7)のように元々存在していない境界線を正しく推定する必要があり，単なる線分の検出処理だけでは，コマの境界線を正しく推定することは極めて難しい．

4.2.2　　ディープラーニングの隆盛以前

線分の検出では古くから知られている古典的な直線の検出手法である Hough 変換[Hough 60] を利用した手法[Duda 72] [Ballard 79]や line segment detector [Rafael 12]などの利用が考えられるが, 欠損のある境界線の推定は極めて難しい. **図 4-1** は, line segment detector を使って, 線分を抽出したときの実験画像である. コマ境界線の遮られた箇所は復元できず, 吹き出し線やオブジェクトの一部の垂直または水平の線分に近い部分などのコマ境界線以外の箇所に誤反応している.

2007 年に Apple 社の iPhone が発表され, その後に大画面の携帯電話が市場を席巻することになる以前の 800×600pix よりも小さな表示画面サイズの携帯電話向けの電子マンガ配信では, コマ画像ベースの配信方式であった. コマ画像ベースの配信は表示領域の限られた小さな表示器を持つ機種が対象で, マンガ画像を小さな画面にフィットさせて読書体験の向上を計った. そのために, 膨大なマンパワーを使ってページ画像から手動で1コマずつ切り出して配信していた. そのため, マンガの各ページ画像からコマを自動的に切り出すことが重要な技術として着目され研究された.

そのころのディープラーニングの隆盛以前に発表されたコマの抽出に関する研究としては, 再帰的 X-Y カット(ギロチンカットとも呼ばれる)アルゴリズム [Han 07], 網羅的探索 [Chan 07], あるいは密度勾配 [Tanaka 07] を用いて, 隣接するコマ間の境界線を検出する方法, ガターに注目して境界線からコマの検出を行う方法[Ishii 07], 「GT-Scan」と名付けられた濃度勾配などの画像処理技術を使った方法[Nonaka 09]等が提案された. しかし, X-Y カットに基づく方法[Han 07]は, ノイズ等を含む長方形形状以外のコマをロバストに分割することができず, また, これらの方法はいずれもコマ境界線に欠損のあるコマや明示的な境界線のないコマを扱うことは困難であった. さらに, コマを分割するために, 主に連結成分ラベリング(CCL:connected component labeling)アルゴリズム [Arai 10]に基づく方法, あるいはページの背景マスク [Pang 14] に基づく方法が提案されたが, 白い背景ときれいなガター(2 つの隣り合うコマ境界線の間の細い空白部分)に依存しており, 長方形形状以外の不規則な形状のコマを個々の成分として識別することはできるが, オノマトペや吹出しも含む複数のオブジェクトによって境界線が消失し, 見かけ上結合しているコマを分離することは困難である. 結合したコマを処理するために, CCL マスク上で N 回の縮小と拡張 [Ho 11] のシーケンスを繰返して結合要素を分割するが, 1 辺以上が裁ち落としになっていてコマ境界線が完全に無い場合の CCL マスクには, 断片化された境界領域のグループが個々の構成要素として含まれることがあり, 侵食処理から完全なコマの形状を得ることは困難である. 連結成分のバウンディングボックスのクラスタリング[Rigaud 13]や, 検出された境界線の候補とコーナーに長方形を当てはめる[Stommel 12]ことで, 裁ち落としのあるコマ境界線の形状を回復できる場合もあるが, それらは規則的な形状のコマにのみうまく適用できた.

これらの手法ではコマの領域の推定にはヒューリスティックな特徴量を使用するので, 想定とは異なるレイアウトの場合には必ず失敗する. 前述のレイアウトの複雑さや内容の多様性のため, これらの手法では依然として充分な抽出精度は得られない.

狐のお嫁ちゃん(4)©Batta

図 4-1. オリジナル画像(左)　と　line segment detector の実験結果(右)

4.2.3　　ディープラーニングを利用した手法

ディープラーニングを用いた先行事例として，物体検出モデルの YOLO を利用した Arpita [Arpita19]らの研究や，物体検出モデルの SSD300 を利用した小川ら[Ogawa18]の研究がある．どちらの手法も物体検出のための長方形の提案領域を利用するため，抽出するコマの形状は長方形に限定される．

小川らは SSD300 を拡張した SSD300-fork という手法および，新しいアノテーションデータセットとして Manga109-Annotations を提案しており，ロバスト性が高く，物体がコマの境界線を遮った場合でも非常に良いコマ検出精度を達成している．提案された Manga109 マンガ書籍データセットに対するコマの領域のアノテーションデータも，物体検出の手法に合わせて長方形になっている．

ただし，Manga109 に含まれているマンガ書籍は出版された時期が最新とは言えないため，全体的にコマの形状は長方形のコマが多く，コマのアノテーションおよび長方形形状のみのコマ推定であっても精度はそれほど悪くならないと思われる．

ただし，マンガを分析・理解する過程で，コマ領域を長方形だけで近似すると，近似した長方形の領域内部に隣接するコマのオブジェクト全体またはその一部が混入してしまい，オブジェクトの位置を誤ってしまう可能性があり，長方形以外の形状のコマも正確な形状で抽出できることが，より望ましいと言える．

本稿のコマ推定器は，セマンティックセグメンテーションを行う畳み込みニューラルネットワーク(CNN)を用いて，前述の複雑なコマ境界線の状態によらず，コマの内側と外側の領域を画素単位で推定する領域分類を学習する．CNN にはマンガの1ページの画像を入力し，目的画像にはページ画像の各ピクセルをコマの内側に属するピクセルとコマの外側に属するピクセルに 2 値分類したマスク画像を与え，その出力が目的のマスク画像に近づくように学習する．コマは CNN が推定したマスク画像からルールベースの画像処理を使って推定された連続領域を多角形ポリゴンに近似して抽出する．

4.2.4　　CNN の学習用データセット

セグメンテーション CNN は，1ページのマンガ画像と，コマ領域の内側を“白”，コマ領域の外側を“黒”とする 1 チャンネルのマスク画像が目的のマスク画像を入力し，CNN の出力と目的画像を比較し Binary Cross Entropy の損失を最小化するように学習する．

CNNの学習では入力となるページ画像と，その画像とペアになる長方形形状以外のコマも正しく反映しているマスク画像（目的画像）が大量に必要だが，利用可能なデータセットが存在していなかったため，電子書籍配信用のマンガ書籍画像データを使って自作した．そのためこの実験のデータセットは非公開である．

データセットを作成するにあたり，作家ごとに画像を分類して学習データを準備した方が，良い学習結果が得られるのではないか，との疑問がある．

それに対し，同一の作品内であれば，そのコマ割りには作家ごとの依存性（特徴）はあるものの，同一の作家であっても，表現するコンテンツのジャンル（恋愛もの，学園もの，スポーツもの，時代物，SF，英雄もの，怪奇もの，その他等複数あり）や作成した年代の流行などによって，全く異なったコマ割りをすることが多いと言われている．そのため，特に作家にはこだわらず，なるべく多くの種類の画像をランダムに抽出して，学習用の画像を準備した．

複数のマンガ書籍から任意に抽出選択した 4,041 枚（使えなかった画像もあるため結果としての数字で，この数字自体には特に意味はない）のページ画像を選択し，これらのページ画像とペアになるアノテーションデータ（ページ画像内のすべてのコマを 4 頂点ポリゴンで囲う）をすべて手作業で作成した．マスク画像は，プログラムでアノテーションデータから生成した．そのうち約7%の 282 ペアをテスト用に分割した．さらに残りの 3,759 枚のページの画像と対応する生成済みのマスク画像のペアから 2 組をランダムに選択し，それぞれのページ画像とマスク画像を横方向にプログラムで連結し，新しいページ画像とマスク画像のペアの 17,317 ペアを生成して追加し 21,076 ペアを学習用とし，テスト用と合わせて合計で 21,358 ペアのデータセットを構築した．

4.2.5 CNN の入力画像

CNN の入力単位の 1 ページの画像は，書籍あるいはページによってそのサイズと色が異なるので，すべてのページ画像はグレイスケールに変換し，256×256px にリサイズする．**図 4-1** の左のオリジナル画像例をリサイズし，**図 4-2** の左の画像を得る．ただし，**図 4-2** の表示スケールは本稿の表示領域の都合で不正確で，アスペクト比のみ正しい．**図 4-2** の左側の入力画像の例において左側の 3 つのコマの左辺と，左下のコマの下辺側コマ境界線が無く，裁ち落としになっているのでページの端までが意味のあるコマ領域であると考え，画像の上下左右の四辺で構成する長方形をページのサイズとみなす．

なお，**図 4-2** の左側の入力画像の例の外側の青線と及び**図 4-3** の外側の青線は著者がページ画像のサイズを明確に表現するために追加した補助線である．

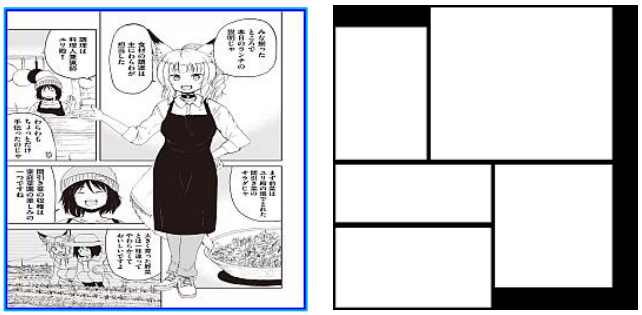


図 4-2．256x256 にリサイズした入力画像（左）と目的画像（右）

狐のお嫁ちゃん（4）©Batta

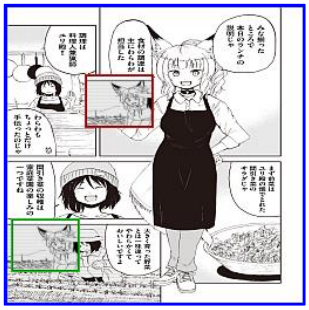


図 4-3．Random Erasing の例

4.2.6 CNNの目的画像

CNNの目的画像は，対応するページ画像のアノテーションデータからプログラムで生成した図 4-2 の右側に例を示したように，コマ領域の内部を“白”，それ以外は“黒”として描画したマスク画像である．コマの境界は 3px の黒線で，画像の端は 2px の黒線で描画し，白ピクセルが連続したそれぞれの領域が1つのコマ領域となるように目的マスク画像を描画する．
学習に用いた目的画像の例を図 4-14 に示します．

目的画像は，アノテーションを行ったページ画像とアノテーションデータのペアから以下の手順のプログラム処理で生成する．

(1) 最後に外周に追加する黒パディング（四辺各 2px）分を考慮し，252×252px サイズの黒い正方形画像を作成する．
(2) アノテーションデータの各四角形コマ領域の 4 頂点の座標データを，アノテーションを行ったマンガページ画像を 252×252px にリサイズした縮小率に合わせて換算する．
(3) 各四角形領域を面積で降順ソートし，面積が大きい方から順にコマの白領域と境界の黒線を描画し，重なっているコマの小さい方の境界線が上書きされずに残るようにする．
(4) 画像の周りに 2px の黒パディングを追加して 256×256px サイズの目的画像を得る．

4.2.7 セグメンテーションネットワークの構造

コマ検出器のネットワークは，**図 4-12** に示す U-Net で提案されたスキップ接続を持つ砂時計型の Encoder-Decoder 構造を有する CUNet と名付けた2つの CNN を直列に接続した**図 4-13** に示す構造である．U-Net が高精度なセマンティックセグメンテーションにおける密な推定タスクで高精度な画素レベルのクラス識別ができる事を期待した．2つの CUNet それぞれの出力は，入力画像と目的画像と比較し Binary Cross Entropy の損失を最小化するように学習する．一段目の CUNet1は入力画像からマスク画像に変換し，二段目の CUNet2は CUNet1の出力を入力として実行し，CUNet1の出力と目的画像との差分を出力して一段目の出力の修正を行い，CUNet1と CUNet2の損失を共に最小化するように構成する．CUNet2の出力と入力(CUNet1の出力)を足し合わせて最終出力とする．

4.2.8 Data Augmentation

CNNの学習時に，入力画像と目的画像に以下の順の変形を行いデータの増強を行う．

(1) 上下左右に 16px のパディングを追加し，256x256 でランダムに切り取る（Random Crop）．
(2) 画像内の切り取りと張り付けを使用し，ランダムな大きさの領域を消す（Random Erasing）．
(3) ランダムに±45 度の範囲で回転（Random Rotation）し，中心を 256x256 で切り取る
(4) ランダムに左右反転（Random Flip）する．

入力画像と目的画像のペアはその位置があっていなければならないので（1），（3），（4）は入力画像と目的画像の両方に同じ変換を適用する．
（1）のパディングはランダムな位置（上下左右±16px）から切り取るときの余白の確保のために行う．入力画像であるマンガのページ画像が，常に紙面の中央にあるわけではない事を学習させるために行う．

(2)は，入力画像にのみ (a) ランダムなサイズの矩形（一辺が 8〜64px）をランダムな位置からコピーし（**図 4-3**. の右側の図中の左下の緑色の矩形部分），(b) ランダムな位置の（a)と同じサイズの領域へ（a)でコピーした矩形を張り付ける（**図 4-3** の右側の図中の左上の赤い矩形部分）ランダムクロッピングという変形を行う. 吹き出しや人物などのオブジェクトがコマ境界線に重なって，コマ境界線が遮られ描画されていない場合であっても CNN でコマ境界線が本来あるべきところを推定したいので，入力画像のコマ境界線の上に他の領域をランダムに貼り付けて，コマ境界線が何らかのオブジェクトに遮られて表示されていない場合を想定し，コマ境界線が本来あるべき状態を推定できれば損失は小さくなり，できなければ損失は大きくなるように学習する.

4.2.9 コマの抽出

マンガのページ画像から CNN が推測したグレイスケールの出力画像から，以下の手順のルールベースの画像処理で複数の頂点からなるポリゴンとして各コマを抽出する.

(1) グレイスケールの CNN の出力画像を閾値（通常 128;調整可能）で2値化したマスク画像を生成する

(2) ラベリング処理で複数の非ゼロのピクセル(=コマの内側の領域)が連続する白い領域を抽出する

(3) 検出された複数の連続する白い領域に対して，左上から右下に向かって順に以下の処理を行う.

(ア) 選択した領域のマスク画像を抽出し，その輪郭線を抽出する.

(イ) 近似精度を減らしながら抽出した輪郭線を直線近似して，選択した領域のマスク画像をなるべく少ない頂点数のポリゴンに近似し，4 頂点になったらそれをコマの頂点座標として選択する.

(ウ) 選択した領域のマスク画像を 4 頂点ポリゴンに近似することができない場合は続いて以下の処理を行う.

i) モルフォロジー変換（Morphological Transformations）で細い部分の分離を行い，ひょうたん型の領域のマスクは 2 つの領域に分割する.

ii) 分割された領域に対し再度輪郭線抽出を行い，頂点数 4 という制限は付けずに分割された領域をなるべく少ない頂点数のポリゴンに近似してそれをコマの頂点座標として選択する.

(4) 最後に(3)で抽出したコマを元画像のサイズにスケーリングして，入力画像から抽出したコマとする.

狐のお嫁ちゃん(1)©Batta

図 4-4. 入力画像

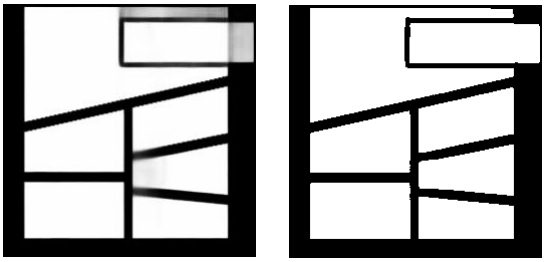


図 4-5. 256x256 に変換後の CNN の出力，
(左)グレイスケール，(右)二値化処理後

図 4-6. 抽出したコマ
領域ごとに着色

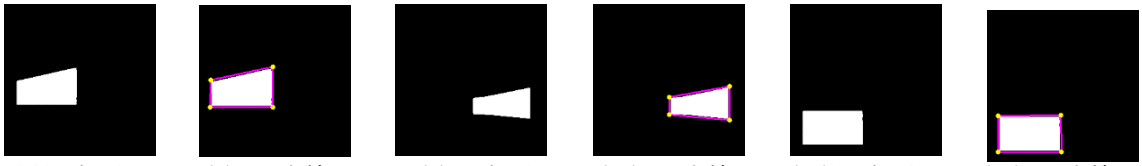
図 4-4 の入力画像の実際のマンガのページ画像(外側のブルーの線は著者が追加したページ画像の外形を示す補助線)を使った例を示す．**図 4-5** は，256x256 に変換後の CNN の出力，(左)グレイスケール，(右)CNN の出力を二値化処理のページ画像を CNN の入力画像とする．この入力画像の例では長方形コマは2つあり，左下は 4.2.1-(1)に該当し境界線の消失は無く，右上は 4.2.1-(5)，(6)に該当し，右辺が裁ち落としであり，3辺の境界線が 1 重で，大きな四角形のコマの中に，このコマが入り込んでいるため，大きな四角形のコマは結果として 8 角形になっている．

4.2.1-(2)に該当する長方形ではない四角形コマが 4 個あり 1 個は左側の真ん中にあり境界線の消失は無く，残り 3 個は 4.2.1-(4)に該当し，右側の真ん中から下に連続して存在し，1 個が 2 つの境界線の一部(上下の辺)が，2 個が 1 つの境界線(上または下の辺)の一部がオノマトペのために消失している，上部に 4.2.1-(5)に該当する 8 角形コマが 1 つ存在し，上辺は裁ち落としになっている．

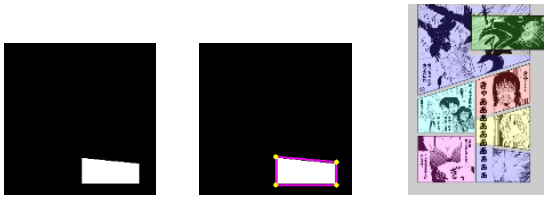
CNN の出力画像としてグレイスケールのマスク画像(**図 4-5** 左)を得て，それを閾値(通常 128; 調整可能)で二値化した画像(**図 4-5** 右)を得る．次に，二値化したマスク画像中の連続する白い領域を左上から右下方向の順に抽出し，その領域ごとに輪郭線の抽出を行い，各領域を囲うなるべく少ない頂点数のポリゴンを求める処理を繰り返す．得られたすべてのポリゴンを元画像のサイズに戻してスケーリングしてページ画像中のコマ領域の頂点座標を得ることができる．**図 4-7** に得られたすべてのコマ領域ごとに着色し，元のページ画像に重ねた結果の画像を示す．

図 4-4 の入力画像から順にコマを抽出する様子を**図 4-7** に示す．二値化したマスク画像中の連続する白い領域を左上から右下方向の順に抽出し，その領域ごとに輪郭線の抽出および各領域の頂点の補間処理(領域をなるべく少ない頂点数のポリゴンに補間)を繰り返し，すべての．連続する白い領域の処理が終わるまで繰り返し，最後に，得られたコマ領域ごとに着色して表示している．

(1)領域選択　(2)8 頂点補間　(3)領域選択　(4)4頂点補間　(5)領域選択　(6)4頂点補間



(7)領域選択　(8)4 頂点補間　(9)領域選択　(10)4頂点補間　(11)領域選択　(12)4頂点補間



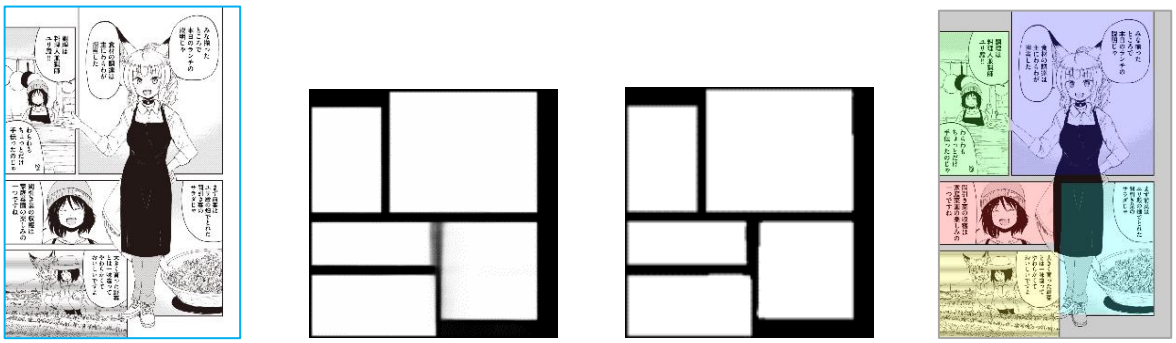
(13)領域選択　(14)4 頂点補間　(15)コマ領域ごとに着色

図 4-7. 入力画像(図 4-4)から順に白が連続する領域を選択し，コマ領域を抽出する流れ

また，コマの形状が長方形以外も含んでいる特徴的な各種ページ画像から**図 4-5** と同様にコマ抽出を行い，256x256px に変換後の CNN のグレイスケールの出力，閾値で二値化した画像，抽出した領域ごとに着色した各種画像の例を**図 4-8** に示したように，長方形，長方形以外の四角形，三角形，多角形の各種コマ形状に対応しており，また，4.2.1 で記述したコマ形状の多様性の(1)～(6)のコマ境界線の形状に対応できていることがわかる．

(i)全てが長方形

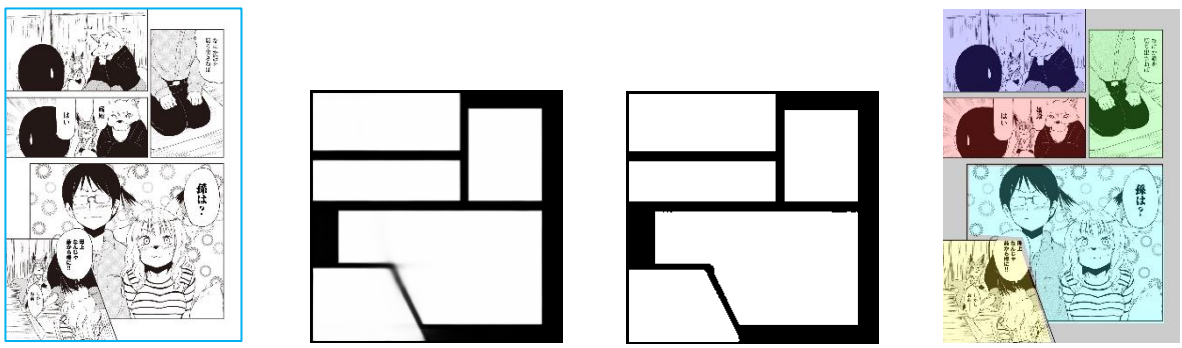
狐のお嫁ちゃん(4)©Batta



入力画像　256x256 に変換後の CNN の出力，(左)グレイスケール，(右)二値化　抽出したコマ領域ごとに着色

(ii) 長方形の他に，6 角形を含む

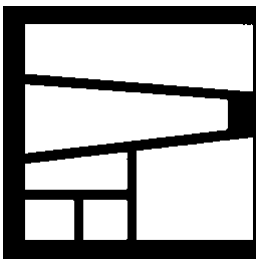
狐のお嫁ちゃん(2)©Batta



(iii) 長方形の他に上辺または下辺の一方または両方が水平に対して斜めになっている四角形

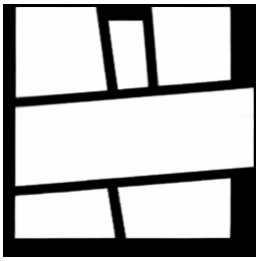
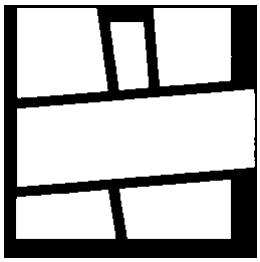
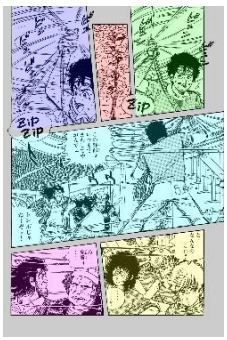
狐のお嫁ちゃん(2)©Batta





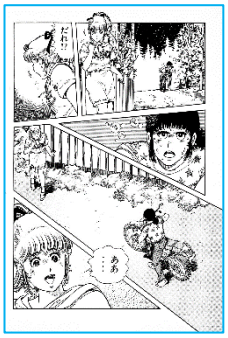

(iv) すべてが, 長方形ではない斜め線の四角形

ココナッツ AVE. 1©三浦みつる

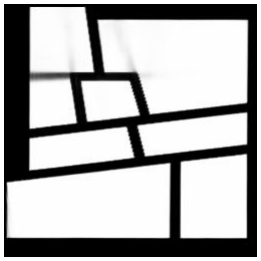
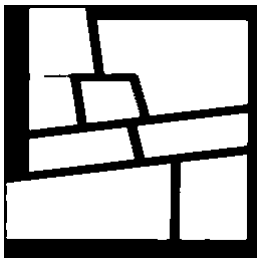
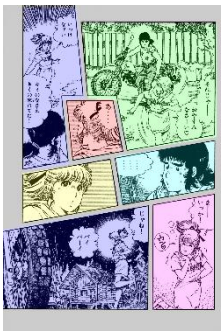
(v) 長方形ではない四角形の他に, ,三角形, 6 角形を含む

ココナッツ AVE. 5©三浦みつる




（vi） 長方形ではない四角形の他に 6 角形を含む

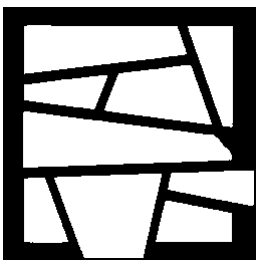
ココナッツ AVE. 5©三浦みつる

（vii） 長方形ではない四角形が多数

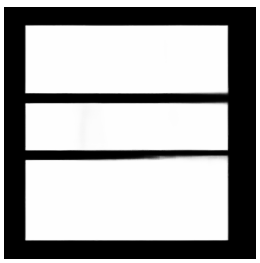
恋はグーチョキパー©御堂カズヒコ





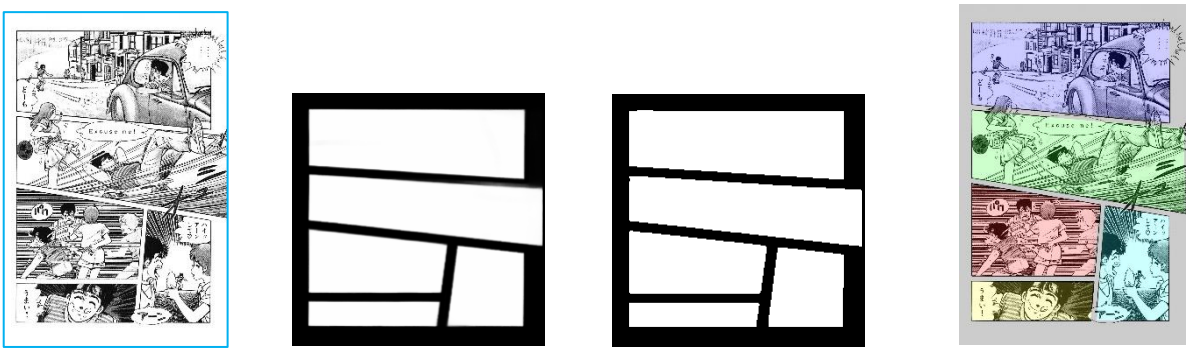

（viii） 全てが大きな長方形

狐のお嫁ちゃん(1)©Batta

（ix） 長方形ではない四角形が多数

ココナッツ AVE. 1©三浦みつる



（x） 四角形，六角形

ココナッツ AVE. 1©三浦みつる

図 4-8. 特徴的なコマを有する画像例（i）～（x）
左から順に，入力画像（左端，外枠の青線は著者が追加），CNN の出力（グレースケール）画像，
CNN出力の二値化画像，コマ領域ごとに着色した画像（右端）

4.3 コマ抽出精度の比較

CNN を用いた先行事例として，小川ら[Ogawa 18]らは物体検出モデル SSD300 を改良した SSD300-fork という手法を提案している．この手法によるコマ抽出はロバスト性が高く，物体がコマの境界線を遮った場合でも非常に良いコマ検出精度を達成している．このような CNN ベースの手法は，大規模なアノテーション付きのデータセットを必要とするため，既存のマンガ画像データセット Manga109 にアノテーションを施し，Manga109-annotations と名付けられたアノテーションデータセットを作成して SSD300-fork を学習した．

この物体検出モデルの方式上の制約から，Manga109-annotations のコマのアノテーションは，本来の形状にかかわらず，すべてのコマは長方形に近似したアノテーションを施しており，CNN は複数の長方形の提案領域の中から，そのアノテーションとの IOU が一番大きい領域を選択し出力するように学習する．従って，推定されるコマ形状は必ず長方形に限定されることになり，本来のコマ形状が長方形以外の多角形（三角形，四角形，．．．，N角形）の頂点座標は正確に取得できないとい

う制約となり，場合によってはコマ内に存在しているオブジェクトの位置を，別のコマ内であると誤認識してしまう可能性があり，正しいマンガ理解の手法としては好ましくない．

たとえば，**図 4-9** のように 2 つのコマがページ内のいずれかの位置で，左側のコマ A（淡い青色の領域）と，右側のコマ B(淡いオレンジ色の格子のハッチング部分も含む淡いオレンジ色)が斜めの境界線で接していると仮定する．

この配置でコマ A の領域を長方形として推定する場合，淡い青色の台形の上底と下底は，青色の破線の線分が赤い長方形の中で左右に移動できる範囲内のいずれかの位置として推定される．

この場合，オリジナルのコマ A は，上底が a で下底がb(a < b)，高さがhの台形だったが，**図 4-9** に示すように青色の破線の位置から左側の領域の長方形として推定され，コマ A は高さがhで，幅が x (a ≦ x ≦ b) の長方形になり，実際のコマ領域には含まれていなかった隣のコマの領域（オレンジの格子でハッチングされた三角形の領域）を含むことになる．この領域には人型のオブジェクトが描かれているが，本来はコマAの中ではなくコマBの中に存在している．つまり，この三角形の領域に存在するオブジェクトや吹き出しなどを含むコマの位置情報は正しくない．

この例のように，長方形でコマを補間した場合には，その長方形の中に隣接するコマの物体が混じってしまう可能性があり望ましくない．本方式では斜め線で囲われたコマであってもより実体に近いより正確な形状で推定できるので，物体が存在しているコマ位置の誤認識は起こりにくいという大きなメリットがある．

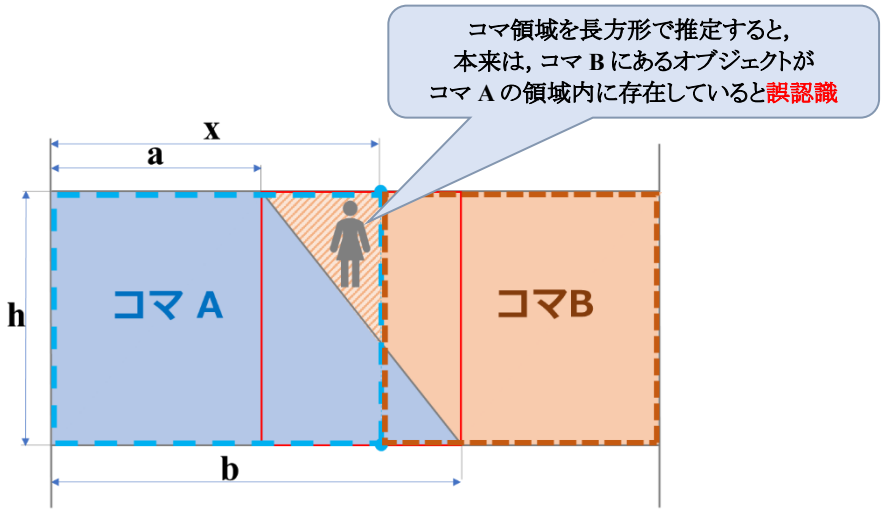


図 4-9．2 つのコマが斜線で接しているコマ配置の推定

本稿のコマ抽出の精度が既存方式と比べどの程度であるかの比較を試みるにあたり，正しいコマの形状でアノテーションされた公開データセットが無いので，公開された Manga109-Annotation に含まれるコマのアノテーションを使用し．比較データの取得時は，できるだけ同等の条件で比較できるように本方式側を修正して対応するものとした．

このアノテーションは，前述の通り，紙ベースの書籍の左右の 2 ページを連結した見開き画像を 1 ページ単位として扱い，コマのアノテーションは元々のコマ形状をすべてコマに外接する長方形に近似して扱っている．

それに対し，本稿のCNNは紙ベースの書籍の 1 ページをそのまま 1 ページの入力単位として扱い，抽出するコマの形状は長方形ではなく，元々のコマ形状として抽出する仕様なので，CNN の入

力に見開き画像を扱えるように修正した．また，IOU の計算では本稿のコマ抽出器が抽出したコマに外接する長方形を求め，それを本稿のコマ抽出器が抽出したコマとして扱い，小川らの論文中に記載されている Manga109 の書籍データセット 10 冊のコマ検出精度を求めた．小川らの論文から引用したコマの抽出精度は，frame の AP の数値を引用した．比較結果を表1に示す．2 行目の項目名の下に括弧書きしている名称は，小川らの論文内で記載されている項目名を記述した．

それぞれの CNN の学習に使った画像データセットは全く異なり学習環境が全く異なるため，直接的なコマ抽出 CNN の性能比較にはならないと言えるが，同じマンガのページ画像に対する精度を比較することによって，目安としての性能を検証した．**表 4-1** に，小川らの評価で使用した Manga109 内の 10 冊の書籍を使った比較実験の結果を示す．

表 4-1. コマの抽出精度の比較

No.	小川らの論文から引用				本方式	
	コンテンツ名 (Volume)	ページ数 (#page)	ジャンル (Genre)	コマ精度% (frame AP)	コマ精度%	差分
1	UltraEleven	108	sports	95.2	96.7	1.5
2	UnbalanceTokyo	82	science fiction	98.6	98.0	−0.6
3	WarewareHaOniDearu	91	romantic comedy	94.0	97.3	3.3
4	YamatoNoHane	109	sports	98.6	96.2	−2.4
5	YasasiiAkuma	89	fantasy	97.8	95.7	−2.1
6	YouchienBoueigumi	26	four-frame	100.0	100.0	0.0
7	YoumaKourin	101	fantasy	99.2	97.0	−2.2
8	YukiNoFuruMachi	93	love romance	95.4	96.7	1.3
9	YumeNoKayoiji	96	fantasy	94.2	94.3	0.1
10	YumeiroCooking	85	love romance	97.7	96.3	−1.4
合計／平均		880		96.8	96.6	-0.3

小川らの方式および本稿の方式で共に 100%になっている 6 番の書籍は 4 コママンガ（four-frame）という長方形形状の 4 つのコマで 1 つのテーマを表現する独特のジャンルのマンガの描画方式で，同じ大きさの 4 つの長方形が縦に並んだ単純なコマ割りになっており，コマの推定が容易なので，両方式とも 100%の高い精度になっている．10 冊の平均精度は小川らが 96.8%で本方式が 96.6%となっており，本方式の方がおよそ 0.3 ポイント低い結果になっているが，前述の通り前提条件が異なっているため，ほぼ同等の性能であると考えた．むしろ，今後のマンガ画像の解析などの応用を考えた場合には，正確なコマの形状をベースとする必要があるため，コマの抽出方式としては本方式がより有効と判断した．

4.4 コマの抽出の応用

海外においても日本のマンガに対する認識が深まりつつある．ところが，日本のマンガを海外の基準で判定すると，児童や少年，少女，ヤングアダルト向けのマンガであっても，その中に暴力的な言葉・表現や，性的な内容などの不適切な表現が含まれている場合があり，特に，世界一の市場である米国では，日本では許容されている性表現が全く認められないことも多く，そのまま販売すると問

題になることがある. そこで, 本稿のコマ抽出器の応用として, マンガの 1 ページを入力し, ページごとにコマを抽出し, それらのコマごとに不適切画像, その中でも描画頻度の高い露出した女性の胸, を検出する仕組みを構築した.

前述の通り, マンガは異なった大きさと形状のコマごとに異なったスケールで異なった視点からの画像を描画しているので, 不適切画像の検出は, ページ画像中のコマ画像ごとに判定を行う必要がある.

本稿で述べた, コマ検出器を使って構築した不適切オブジェクトの検出器の結果表示例を**図 4-10**(右から 2 つ目)及び**図 4-11**(右から 2 つ目)に示す. 不適切オブジェクトを含むコマとして検出されたコマはオレンジ色に着色している. 入力されたページ画像は, **図 4-10**(左)及び**図 4-11**(左)に示す. また, **図 4-10**(左から 2 つ目)及び**図 4-11**(左から 2 つ目)に検出されたコマのそれぞれに着色した画像を示す. 不適切オブジェクトの認識は, resnet14 をベースに SPP(Spatial Pyramid Pooling)構造[He 15]を取り入れ, 不適切画像を含む／含まないの 2 値分類モデルの CNN を使った.

CNN の学習には複数のマンガ書籍から抽出した露出した女性の胸の画像を含むコマ(およそ 34,000 コマ)と含まないコマ(およそ 232,000 コマ)を使った. **図 4-10**(右端の上・下)及び**図 4-11**(右端)は, 検出した不適切画像を含むそれぞれのコマを拡大して表示した. 小さなコマ内に露出した女性の胸が存在しているのがわかる.

ココナッツ AVE.(1)©三浦みつる

図 4-10. オリジナルページ画像(左端), 抽出コマごとに着色, 不適切画像のみ着色, 部分拡大図(右端上・下)

恋はグーチョキパー(1)©御童カズヒコ

図 4-11. オリジナルページ画像(左端), 抽出コマごとに着色, 不適切画像のみ着色, 部分拡大図(右端)

4.5 まとめ

本稿では，CNN を用いてコマの境界線に欠損があるコマ領域の推定を行い，後処理にルールベースの画像処理を適用することで，マンガ書籍のページ画像からコマ領域を抽出する方式について記述した．機械学習が多数の問題解決に利用できるとは言っても，すべてを機械学習だけで解決するのではなく，既存の優れた技術と組み合わせることで，合理的な規模の CNN を使って問題解決ができることを示した．また，本技術を利用して抽出したコマ内の不適切オブジェクトの存在の有無を確認するシステムの例を述べたが，こちらは単一の機能の CNN を2つ組み合わせて使うことで，1 つの CNN で実現することにこだわらずに目的とする機能を実現でき，実用的な問題解決が可能であることを実施例として示した．

マンガのように文字と画像が複雑に入り混じったコンテンツを機械で正しく読み解くためには，これから解決しなければならない課題は多いと思われる．次の課題の一例として，吹き出し中の画像化された活字文字や背景画像に埋もれた画像化された活字文字のテキスト化，コマ内の人物及び人物の表情の認識，コマの読みくだし順序の解析，オノマトペと言われる擬音の画像化文字の認識と解析，背景画像の解析など多数が考えられるが，いずれもコマの位置や大きさなどのコマ座標を基本として解析を行うため，本技術はマンガの解析及び内容理解のための基本技術として，極めて重要であると思われる．

将来，機械によりマンガを解析し，その内容を理解することで，マンガの内容に深く関連した詳細なデータベースの構築が可能になると思われる．それを利用することで，読者からの多くのリクエストに対して正確に案内が可能なマンガのコンシェルジュのシステムの構築も可能であろうし，また，ドメスティックに限定されているマンガコンテンツをグローバルなコンテンツとして新たに展開できる可能性を大きく広げることが期待される．

謝辞

本稿を記述するにあたり,『ココナッツ AVE』の作家の三浦みつる氏,『恋はグーチョキパー』の作家の御童カズヒコ氏, 及び『狐のお嫁ちゃん』の作家の Batta 氏, からは快く画像掲載の許諾をいただき, 画像を文中に挿入させていただいた. この場を借りて深く御礼申し上げます.

また CNN の学習用の画像は, 著者が株式会社イーブックイニシアティブジャパン在職中に販売していたマンガの電子書籍画像からランダムに選択して収集して使わせていただいた. 株式会社イーブックイニシアティブジャパンの経営陣からは, これらの学習用画像の利用および本稿の発表の許諾をいただいた. この場を借りて深く感謝いたします.

さらに, システム構築及び CNN の学習時には数々のアイデアを提供いただき, また, そのための実験を支援していただいた永冨一也氏には深く感謝いたします.

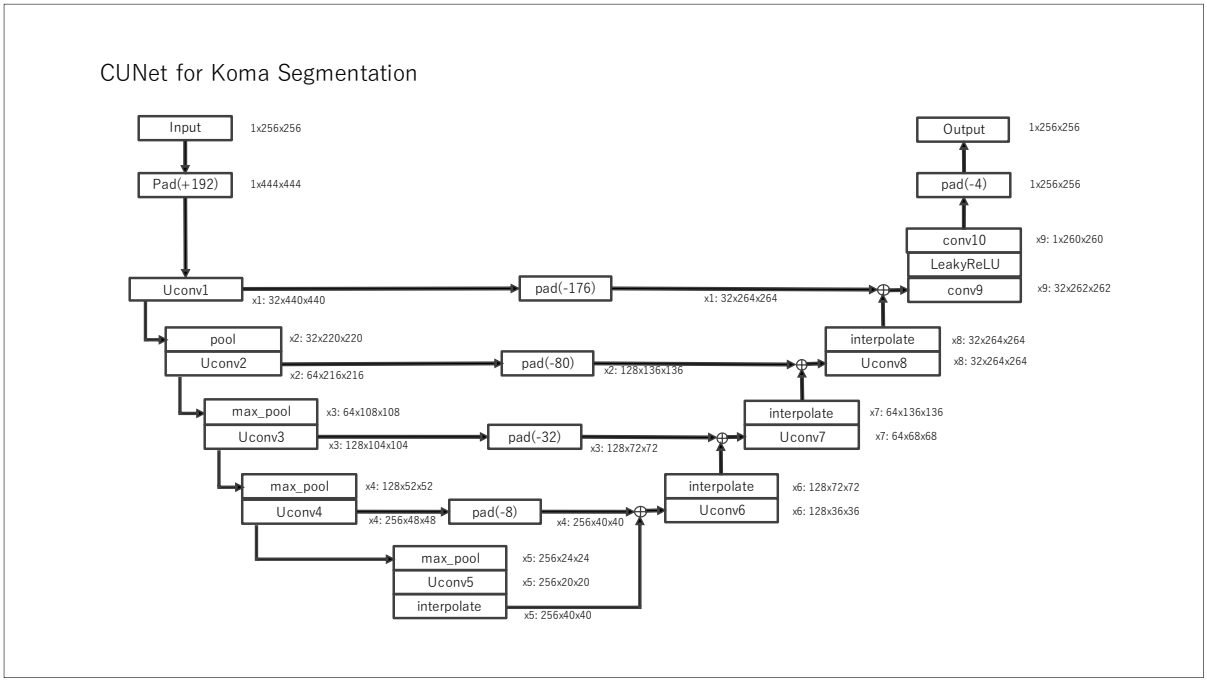


図 4-12. CUNet 構造

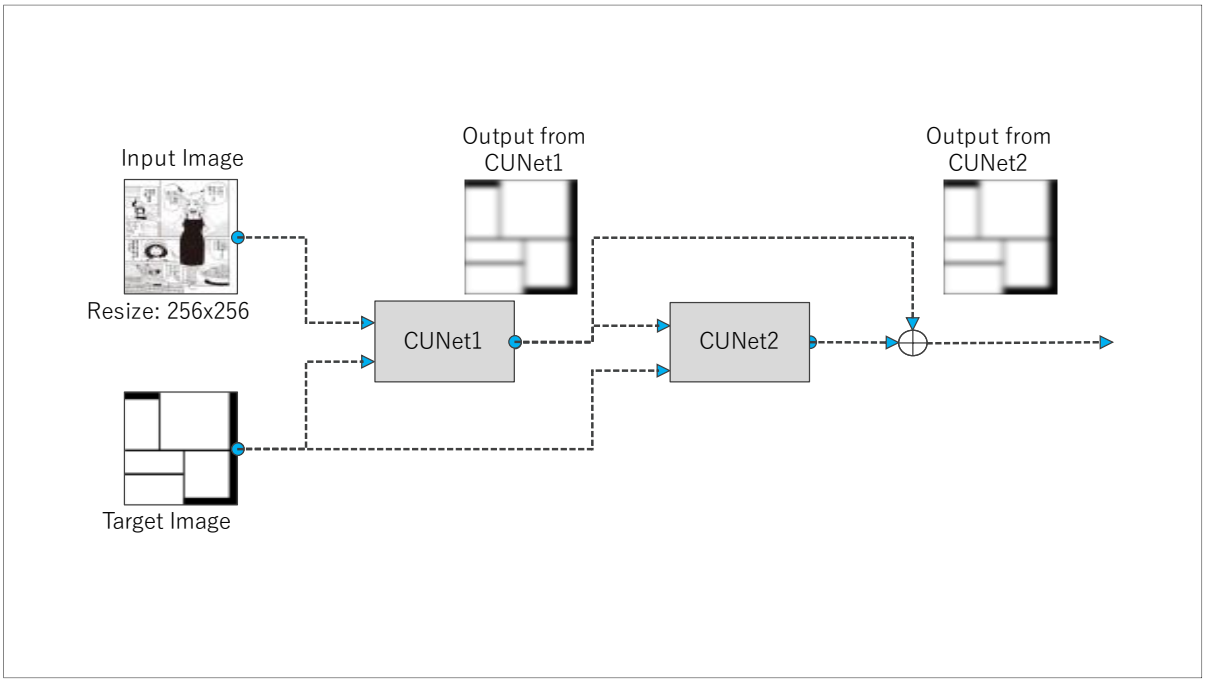


図 4-13. コマ検出 CNN の構造

図 4-14. 目的画像の例

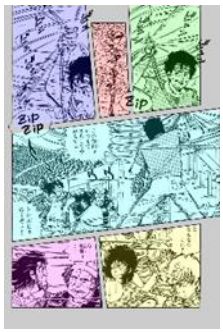
付録 4-1. 抽出した領域ごとに着色した画像一覧(i)～(x)

(i),(ii),(iii),(viii) 狐のお嫁ちゃん©Batta
(iv),(v),(vi),(ix),(x)ココナッツ AVE©三浦みつる
(vii) 恋はグーチョキパー©御堂カズヒコ

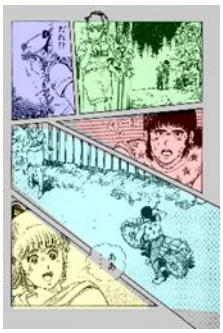

(i)長方形


(ii)長方形・四角形・六角形


(iii)長方形・四角形



(iv)四角形(斜め)


(v)三角・四角・六角形


(vi) 四角・六角形


(vii) 四角形・複雑


(viii)長方形


(ix)四角形


(x) 四角・六角形

抽出した領域ごとに着色した画像一覧(i)～(x)

第5章　マンガ書籍中の不適切な画像検出システム

5.1　はじめに

日本でマンガコンテンツの電子書籍が商業的に普及し始めてから約 20 年，電子書籍配信用データの多くは，紙に印刷されたマンガ画像をスキャンし，画像に手を加えずに商品として画質保証された電子データに変換した上で課金・配信されてきた．近年では，電子書籍は一般的になり，幅広い年齢層の一般ユーザが手軽にマンガ画像を閲覧することができるようになった．一方，マンガ書籍は，マンガ画像の表現に明確な統一されたレーティングが無く，各出版社内の自主ルールに基づき自由に制作する，という独特の文化の中で出版されてきた．そのため，既に販売された膨大な数のマンガ書籍中には，成人読者向けとして限定していない若年者を含む一般の少年少女読者を対象としている出版物であるにもかかわらず，前述のような独自の歴史的な出版事情により，マンガ書籍中のいたるところに不適切と思われる描写が散見している．

マンガが日本発のユニークでクールなポップカルチャとして国際的に認知されるにつれ，マンガの内容描写については，日本独自の出版社に委ねられた曖昧な判断基準ではなく，世界的に認められた法令や規制に基づいて評価され，暗黙も含む国際的な常識に従わなければ世界的な戦略商品として認められない傾向が強まってきており，もしもこのまま，国際的な常識に従わない場合，世界市場向けの商品としては認められなくなり，その結果，商品価値が低下し，今後のマンガコンテンツ販売事業の展開において大きなデメリットになる可能性がある．

例えば米国では，児童ポルノ保護法で，『児童または児童に見える者による性的行為の写実的描写の配布行為を禁止』してる．以前，少年・少女向けのマンガであるドラえもんの中にある複数個所の「しずかちゃんの入浴のシーンの描写」に関して，日本においては全く問題が無かったにもかかわらず，米国では問題となり，修正をして販売したところ商業的に成功し，結果として多くのユーザに愛されるコンテンツになった．実際にはしずかちゃんは実在する児童ではないので完全に違法であるとは言えないのだが，米国では特に児童ポルノには過剰に反応する傾向があると言えるのかもしれない．逆に，正しく対応できれば世界的な戦略商品として通用することが期待できる．特に海外の大手販売プラットフォーマの Web サイト経由で電子書籍を販売する場合，そのプラットフォーマが定めるレーティングという，いくつかの販売対象とする年齢ゾーンごとに規定された判断基準を満たす必要がある．

日本では，表現の自由を根拠に出版物の表現に対する明確な法令や規制がなく，出版社ごとにそれぞれ異なった自主判断で出版・販売され，市場に長く受け入れられてきたという歴史的な背景があり，特に少年・少女や若い青年向けのマンガでさえ，販売促進の含みの強い「お色気シーン」と言われる女性の入浴シーンや，下着姿のシーン，さらに裸の胸の描写などは許容され，販売されてきた．それに対し米国の少年・少女や若い青年向けのマンガの市場ではこれらの「お色気シーン」の描写は許容されない．このように，現状の日本の出版業界の判断基準が大きくずれていることによるコンテンツの表現に関する問題でプラットフォーマから指摘を受けることもあり，レーティングにうまく対応できない好ましくないコンテンツ及びコンテンツプロバイダーと認識される可能性は否定できない．

一般的に，不適切と言われる表現には，性的表現，暴力的表現，反社会的行為に関する表現，言葉や思想に関する表現などがある．なかでも性的表現には，キス，抱擁，下着の露出，性行為，裸体，性的示唆に富む表現，不倫，排泄，性風俗業，水着・コスチュームなどと細かく分類されてい

る. 本稿では不適切とされる様々な性的表現のうち, 特に女性の胸が露出している画像を不適切画像として検出するシステムについて記述する. 女性の裸の胸を描画する場面は, 性行為, ヌードや性暴力など, 不適切とされる他の性的表現と同時に描かれることが多く, ある意味, 不適切な画像を検出するためのマーカーであるとも言える. さらに, 少年少女・青年向けマンガの場合, 読者の興味を引くための「販売促進サービス」として, 様々な場面で女性の裸の胸が描画され規制されずに販売されてきた. そのため, マンガ書籍中における不適切な描写の画像全体に対して女性の裸の胸の画像の出現率はきわめて高く, 目視でも発見しやすいため, 機械学習の教師画像として抽出・収集する際に有利である.

一方, 過去に販売されたすべての書籍中にこのような不適切な画像が含まれていないかを改めて確認する必要がある要因にもなっている. 特に, 従来の方法で数十万冊を超える漫画書籍から目視で女性の胸の露出画像を確認するためには, 膨大な作業時間と根気を必要とする. しかも作業は長時間の単純作業を連続して行うことになるため, 集中力の低下によるミスが多発することは, 容易に予想される. そのため, 作業時間の短縮, 効率化と検出品質の確保は重要な課題となっている.

また, Web 上に氾濫する, 不適切な画像, イラストに関しての先行研究として, Jay Mahadeokar らの研究[Mahadeokar 16] などがあるが, 1 ページの写真画像やイラストが対象で, 不特定形状のコマ内に描かれたマンガ画像は検出対象から除外されているため, マンガ画像に特化した専用の仕組みとして, コマの抽出機能が必要である.

本稿では, 電子書籍配信用に加工されたマンガ書籍に含まれる露出した女性の胸の画像の有無を検出し, 検出したページ内のコマ画像を結果として出力する不適切画像の検出システムについて記述する. なお, 以降, 特に断りが無い場合を除き, 女性の裸の胸の露出画像を単に不適切画像と記述する.

5.2 システムの目標

不適切画像の検出システムを構築するのにあたり, コマ検出器に関しては, 4 章で記述したコマ検出 CNN を利用し, 以下の目標を定めた.

(1) 不適切画像の有無の判定は, コンテンツの販売書籍単位（書籍 1 冊または 1 話単位）で行う.
(2) 書籍内の不適切画像の判定は, 1 ページの画像から抽出したコマ単位で行う.
(3) コマ単位の不適切ではない画像の検出精度は 99%以上, 平均検出精度は 98%以上を目標とする.
(4) 不適切画像の最終判定は作業者が目視で確認する事を運用ルールとする.
(5) 不適切画像の検出結果のコマ画像は, スコア順に表示する UI を構築する.
(6) 複数の作業者が, 作業を分担し同時に, 並行して目視確認できる構成とする.
(7) システム全体の導入コストを抑える.

例えば, 1 冊が 200 ページで 2,000 コマの書籍において, 不適切画像が 2%含まれていると仮定した場合, 不適切ではない画像の検出精度が 99%のシステムでは 1 冊あたり 2,000×98%×1%=19.6 コマで適切画像を不適切画像と誤検出する. つまり, およそ 10.2 ページに 1 コマの割合で, 適切画像を不適切画像と誤検出する. また, 不適切画像の検出精度が 97%だった場合, 1 冊あたり 2,000×2%×3%=1.2 コマで不適切画像を適切画像と誤検出する. つまり, 166.7 ページに 1 コマの

割合で不適切画像を適切画像と誤検出する．長時間の作業では誤検出は煩わしく，作業者の集中力を低下させ作業の円滑性を損なう．つまり適切画像の検出精度がより高い方が誤検出するコマの出現頻度が低くなり，全体的な作業性は高いと言える．

5.3　システムの構成

本システムは，AI 推論サーバと複数の GUI クライアント間を RPC（Remote Procedure Call）で接続したサーバ・クライアント構成である（**図 5-1**）．目視確認を共同で行う複数の作業者の GUI クライアントを，同時に 1 台の同じ推論サーバに接続して作業を行う．AI 推論サーバは高速な計算リソースを有する GPU を搭載し，マンガのコマの抽出 AI と不適切画像検出 AI が搭載されており，GUI クライアントからの指示に従って，複数の書籍データを取得し，搭載された GPU を使って高速に推論演算を実行するが，書籍データの取得から全ページの画像データの復元，コマの抽出，コマ内の不適切画像判定まで 1 冊あたり数十秒掛かるので，リアルタイムではなくバッチ処理で実行する．全体の処理の流れは，以下のようになる．

まず作業者が GUI クライアントから推論サーバに対し書籍の ID リストを伝達し実行を指示する．マンガ書籍データの本体は本システムとは別のマンガ書籍サーバに 1 冊単位のデータとして保存されている．AI 推論サーバでは作業者に指定されたマンガ書籍を電子書籍配信サーバから取得し，その書籍データに含まれる全ページの画像を復元する．ページ画像は 1 ページごとにコマ抽出 AI でコマ抽出を行い，引き続きコマ単位で不適切画像の認識を行い，コマ単位の認識結果は AI 推論サーバ内の DB に書籍ごとに保存される．AI 推論サーバは結果が揃ったら GUI クライアントに実行完了を通知する．GUI クライアントは AI 推論サーバから結果を取得して表示する．

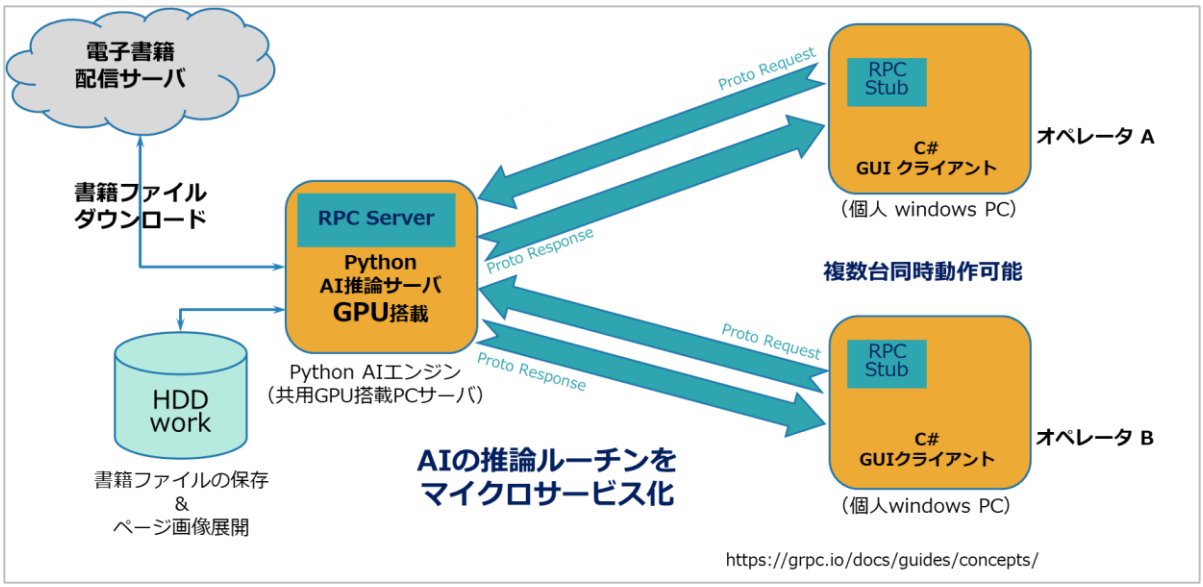


図 5-1. RPC(Remote Procedure Call)を使った Server-Client 構成

5.3.1　　GUI クライアント

GUI クライアントでは推論の演算は行わず，作業者とのインターフェースを司る．作業者は適切・不適切の判定を行う必要のある書籍のリストを作成し，GUI クライアントから AI 推論サーバにその書

籍リストを伝達し，不適切画像の検出結果を蓄積するよう指示する．GUI クライアントは AI 推論サーバに蓄積された検出結果を要求し，サーバから送出された結果に基づいてサムネイル画像表示し，作業者が結果の目視判定を行う環境を提供する．

GUI クライアントは全てのマンマシンインターフェースを担当し，推論サーバと RPC を介して高速に通信を行う．

5.3.2 AI 推論サーバ

AI 推論サーバは，GUI クライアントから RPC 通信を介して指示された書籍リストに基づき電子書籍データ配信サーバから書籍データをダウンロードし，次にダウンロードされた暗号化書籍データから，すべてのページ画像を抽出する．推論サーバには GPU が搭載されており，抽出されたページ画像は，**コマ推定 AI** により，1 ページ毎にコマを抽出する．抽出されたコマ画像は 1 コマごとに，**不適切画像検出 AI** によって不適切なオブジェクトの有無を高速に判定し，その結果を内部に蓄積する．蓄積した結果は GUI クライアントからの適切な指示に従い GUI クライアントに送出する．

5.3.3 コマ推定 AI

本システムで使用しているコマ推定 AI は，本稿の 4 章でその詳細を記述しているが，この5章の組み立て上必要な部分についてはその内容が一部重複するが，読みやすさを考慮し一部この章にも記述する．

マンガはその形状と大きさが不定の複数のコマと呼ばれる多角形（多くは四角形）領域に分割されたページ画像の集合体で，各コマのそれぞれを 1 枚のキャンバスに見立てて情景が描画される．マンガはこのコマに描かれた画像と吹き出しの中のセリフ，擬音，画像の背景及び画像に埋もれた説明文の連続で物語を表現する．

コマには連続する物語の中の 1 シーンを切り取ったような情景が描画されるが，映画やテレビのような固定の形状とサイズの描画領域とは異なり，作者の自由なアイデア次第でどのような形状や大きさでも構わない．コマ単位で 1 枚の絵として完結しているので，通常はコマ毎に描画テーマは異なっており，描画対象，描画位置，描画スケール（倍率），描画アングル（視野角，視点），背景，効果など描画に関する条件は異なっている．これらのマンガの描画手法を考慮すると，不適切画像オブジェクトの検出はページ画像単位ではなくコマ画像単位で検出を行うことが望ましい．そのためページ画像から不規則に分割されたコマを自動抽出する仕組みが必須となる．

一般的なコマ形状は四角形だが必ずしも四角形と決められているわけではなく，それ以外の多角形の場合もある．基本の描画手法に基づくと，画像オブジェクトはコマをはみ出さずにコマ内部に描画される．ただし，コマ境界線を跨ぐ画像オブジェクトや，吹出しというセリフ文字を記述した風船型のオブジェクトがコマ境界線を跨いで描画されることもある．さらに 2 つ以上のコマを跨ぐ大きな画像オブジェクトが描画される場合もあり，それらの場合，コマの境界線はいたるところで欠損が生じるが，コマの推定ではこのような連続して欠損している境界線も正しく推定する必要がある．

特に最近のマンガでは，キャラクタを強調したい場面や，読後の印象を強くしたい場面など，ストーリにメリハリをつけたい場合の表現として，複数個所のコマの境界線を跨いだ画像オブジェクトが描画されている例も多くみられ，境界線の複数の箇所がオブジェクトに遮られ描画されていないため

, 単なる線分の検出だけでは, コマの境界線を正しく推定することは極めて難しい. 線分の検出では古くから知られている古典的な直線の検出手法である Hough 変換[Ballard 79]や line segment detector [von Gioi 12]などの利用が考えられるが, 欠損のある境界線の推定は極めて難しい.

また, ディープラーニングの隆盛以前に発表された既存研究として, 野中ら[野中 09]の「GT-Scan」と名付けた濃度勾配などの画像処理技術を使った手法の発表, および, 石井ら[石井 07]のコマ間の空白に注目して分割線からコマの検出を行う発表などがあるが, これらの先行研究では充分な性能とは言えない.

ディープラーニングを用いた先行事例として, 小川ら[Ogawa18]は SSD300 という物体検出モデルを拡張した, SSD300-fork という手法を提案した. この手法はロバスト性が高く, 物体がコマの境界線を遮った場合でも非常に良いコマ検出精度を達成することができた. ただし, その形状はコマを囲む長方形に限定される.

また, 小川らが調査・評価に用いた「Manga109」のマンガ書籍データセットは[Matsui 16]に紹介があるが, 発売された時期がやや古いため, 古典的なコマ配置で全体的に長方形のコマが多いので, 長方形のみを使用したコマ推定であっても精度はそれほど悪くはならない. ただし, コマに外接する長方形でコマ領域を近似した場合に, 近似した長方形の領域内部に隣接するコマのオブジェクト全体またはその一部が混入することで, 誤った推定をする可能性があるため, 長方形以外の形状のコマも正確な形状で抽出できることが望ましい.

本システムにおけるコマ推定 AI は, 畳み込みニューラルネットワーク（CNN）を用いて, コマの内側と外側の領域を画素単位で決定する領域分割を学習する方式である. マンガのページ画像を入力とし, CNN で得られた領域推定結果のマスク画像からコマ領域及びコマ境界線を推定する.

コマ推定 AI で使用したネットワークモデルは UNet[Ronmeberger 15]で採用されているスキップコネクションを有する同一のネットワークを直列に 2 段連結した構造になっており, それぞれの出力は目的のマスク画像と比較して Binary Cross Entropy の損失を最小化するように学習する.

図 5-4 に複数の境界線を跨いでいるオブジェクトを含むコマ配置の実例と, **図 5-5** に, **図 5-4** のコマの境界線を跨ぐ大きなオブジェクトがあるページ画像に対応するマスク画像の例を示す. マスク画像は, コマの境界線を跨ぐオブジェクトによって遮られた境界線を推定して復元した形状をしている.

1 段目のネットワークは入力されたマンガ画像と正解として与えられたマスク画像を使い, マスク画像を推定する. 2 段目のネットワークは 1 段目が出力したマスク画像を入力として動作し, 正解のマスク画像との差分を出力することで補正し, 1 段目と 2 段目の両方の損失を最小にするように学習する. コマを抽出するには, ネットワークの推論結果として得られたマスク画像に内接する多角形の近似を行う. コマの形状の多くは四角形なので最初に四角形で近似を試み, うまくいかない場合に四角形以外の多角形近似を行う. 結果として最適なコマ多角形の頂点座標が得られ,多角形のコマが抽出される.

コマ抽出 AI モデルの学習のデータは, 1 ページ単位でかつ長方形だけではない四角形形状のアノテーションデータが必要なため, 電子書籍配信で販売中の複数のマンガ書籍から, 学習用に 4,323 ページ（そのうち検証用に 282 ページ）の画像を用意し, すべてのページにコマ四角形のアノテーション作業を行い, このアノテーションデータからマスク画像を生成し教師データとした.

コマ抽出モデルの入力画像の仕様が異なっているので, 直接的な性能比較にはならないが, 相対的なコマ抽出性能を確認するために, 小川らの論文に記載されている Manga109 データセット中の 10 冊の書籍についてコマ抽出精度を求め. 表 5-1 にコマ抽出精度の比較結果をまとめた. その際, 本システムのコマ抽出 CNN の入力のページ画像形状は 1 ページ単位を想定しているが, 小川

らの論文でのコマ抽出条件に合わせて，見開き(2 ページ分)を 1 ページとして入力するように変更し，見開きページの画像にも対応できるように，重複しないようにランダムに 2 ページ抽出して，左右に連結した疑似見開き画像セットを 17,317 セット水増し生成して学習を行った．さらに，本コマ推定 AI で得られたコマ形状をコマに外接する長方形に変換してコマの抽出精度を計算したので，少し不利な入力条件になっている．

表 5-1. コマの抽出精度の比較

No.	小川らの論文から引用			本方式	
	コンテンツ名	ページ数	コマ精度%	コマ精度%	差分
1	UltraEleven	108	95.2	96.7	1.5
2	UnbalanceTokyo	82	98.6	98.0	−0.6
3	WarewareHaOniDearu	91	94.0	97.3	3.3
4	YamatoNoHane	109	98.6	96.2	−2.4
5	YasasiiAkuma	89	97.8	95.7	−2.1
6	YouchienBoueigumi	26	100.0	100.0	0.0
7	YoumaKourin	101	99.2	97.0	−2.2
8	YukiNoFuruMachi	93	95.4	96.7	1.3
9	YumeNoKayoiji	96	94.2	94.3	0.1
10	YumeiroCooking	85	97.7	96.3	−1.4
合計／平均		880	96.8	96.6	-0.3

小川らの手法のコマ抽出精度の平均が 96.8%であるのに対し，同じ条件に換算した我々の手法では 96.6%を達成した．本手法は小川らの手法と，ほぼ同等のコマ抽出性能を有すると判断した．また，小川らの方式は物体検出のための複数の長方形の提案領域の中で最良のものを選択する方式のため，長方形以外の四角形の頂点座標は正確に取得できないという制約がある．

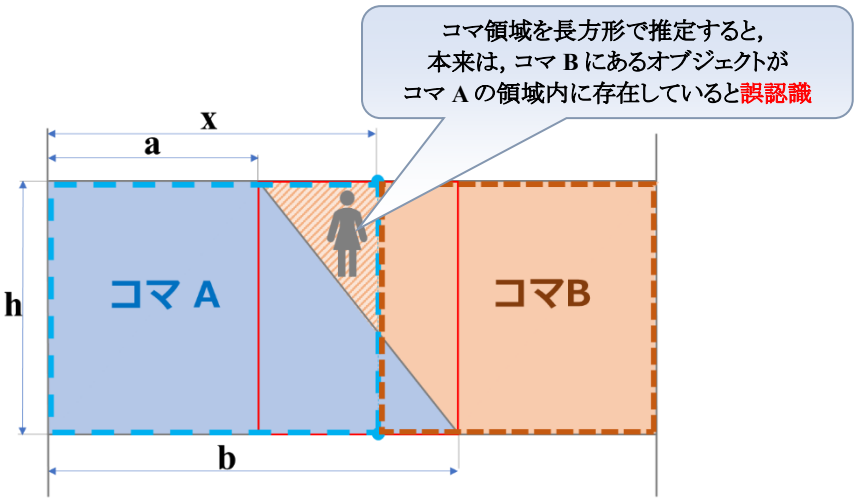


図 5-2. 2 つのコマが斜線で接しているコマ配置の例

たとえば，図 5-2 のようなコマの配置を仮定する．ページ内のいずれかの位置で，左側の淡い青色の領域のコマ A と，右側の淡いオレンジ色のコマ B(淡いオレンジ色の格子のハッチング部分も含む)が斜めの境界線で接しているとする．この配置でコマ A の領域を長方形として推定した場合，淡い青色の台形の上底と下底は，青色の破線の線分が赤い長方形の中で左右に移動できる範囲内のいずれかの位置として推定される．

この場合，オリジナルのコマ A は，上底が a で下底が b(a < b)，高さが h の台形だったが，**図 5-2** に示すように青色の破線の位置から左側の領域の長方形として推定され，コマ A は高さが h で，幅が x (b ≧ x ≧ a) の長方形になり，実際のコマ領域には含まれていなかった隣のコマの領域（オレンジの格子でハッチングされた三角形の領域）を含むことになる．**図 5-2** では人型のオブジェクトが描かれているが，本来はコマAの中ではなくコマBの中に存在している．つまり，この三角形の領域に存在するオブジェクトや吹き出しなどを含むコマの位置情報は正しくない．

マンガ画像の認識や解析では，コマの位置を基準とするので，コマに含まれているキャラクタ，吹き出し，説明文，背景，オノマトペ，などのマンガの構成要素が存在する正確なコマの位置座標は必須になると判断できるので，コマの形状がより正確で長方形に限定されない本手法を採用した．

5.3.4 不適切画像検出 AI

コマ内に描画された検出対象オブジェクトのサイズはさまざまで，コマの大きさにも依存する．コマ全体のギリギリまで使って描画する場合の少し極端な例では，1 ページ全体を 1 コマにして画面の数十分の一程度の大きさのオブジェクトを描画する場合もある．そのため，不適切画像検出 AI ではスケール差の大きな入力オブジェクトを検出する必要があり，各コマ内のオブジェクト画像のサイズに幅広く反応するために， SPP(Spatial Pyramid Pooling)構造[He 15]を取り入れて，オブジェクトの検出感度を上げた sppnet2 と名付けた分類モデル（不適切画像を含む／含まない）を構成し，**図 5-8** に示した．

学習用に収集したコマ画像のサイズはまちまちなので，ミニバッチを可能にして学習効率を上げるため，全ての画像サイズを 256x256pix に統一したが，画像のアスペクト比を変えると，描画されたオブジェクトの表現が変わってしまい，オブジェクトの検出が不可能になるため，コマをタイル状に貼り付けた画像から 1 つ以上のコマが完全に含まれる正方形を抽出する新方式を適用した．

図 5-3 に，縦に細いコマ画像をアスペクト比は維持せず正方形に変形した場合と，アスペクト比は維持し描画表現を変えずにタイル状に貼り付けて正方形にパックして変形した例を示す．

以下の手順で整形を行い，すべての画像を 256×256pix の固定サイズの正方形に加工した．

(1) 学習用に収集するコマ画像は，コマの頂点に外接する長方形画像としてページ画像から切り出す．
(2) 切り出した長方形領域画像から本来のコマ領域の外側の領域を背景色に塗りつぶした長方形（以下単に，長方形画像と記述）を作成する．
(3) 短辺を伸ばす方向に長方形の画像をタイル状に並べ，長辺の長さを超えるか，または等しくなるまで画像を並べる．
(4) 長辺の長さを超えた部分を切り捨て，長辺を 1 辺とする正方形を生成する．
(5) 全体を 256×256pix の正方形に変形する．

恋はグーチョキパー©御童カズヒコ

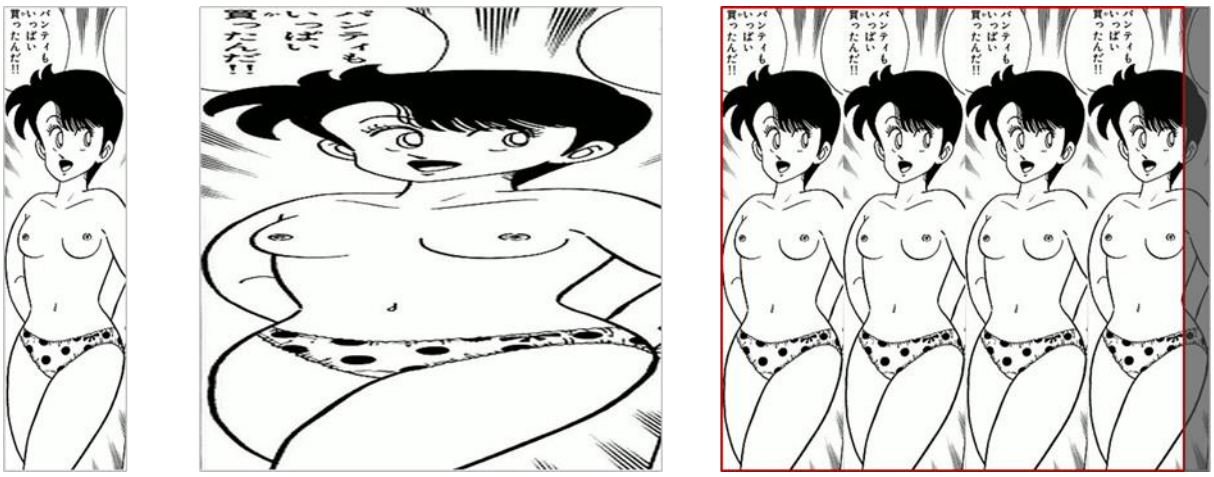


図 5-3 コマ画像をアスペクト比は維持したまま正方形に変形する例

（左）オリジナルコマ，（中）アスペクト比を変えて正方形に変形，（右）タイル状に貼り付けて正方形に変形

また，マンガ画像におけるオブジェクトは一般的な写真画像と異なり，あらゆる視点からの描画が可能なので，X, Y 両方の軸での反転および回転角度に制限を付けないランダム回転を使った変形も可能なので上記の 3）の後で回転を盛り込み，回転後に中心から 256×256pix の正方形を切り出した．

さらに，不適切なオブジェクトがコマ領域の大きさに比べて小さく描かれているために，学習時に不適切なオブジェクトをうまく検出できない可能性のあるコマ画像も含まれている．不適切な画像の割合が少ないため，そのようなコマ画像も含めすべてのコマを学習で正しく認識できるように，ハンドメイドでコマの内部に描画されている胸の中心領域（乳首を含む四角形領域）の座標を追加のアノテーションデータとして用意した．このアノテーションデータを参照して，不適切オブジェクトおよびその周辺をランダムにクロップし，不適切オブジェクトを画像の中心付近に拡大表示した水増し画像を生成した．不適切オブジェクトが常に画像に含まれることで，不適切という画像ラベルは変えずに，不適切なオブジェクトを含む新しい画像を「より検出しやすいサイズ」の画像として水増しした．

学習では，不適切オブジェクトを含むコマ画像は 33,821 コマ（そのうち追加アノテーションを作成したのは，およそ半数の 15,649 コマ），不適切ではないコマ画像は 231,859 コマを用意した．学習用及び評価用の両方を合わせて，1038 人の作家の 2965 冊の書籍から不適切オブジェクトを含むコマと含まないコマ画像を抽出した．学習用には 996 人の作家の 2782 冊の書籍から，評価用には 80 人の作家の 414 冊の書籍から抽出し，その中の 38 人の作家の 231 冊の書籍は学習用と評価用で共通に使用した．また，不適切オブジェクトを含むコマに限定すると，学習用には 762 人の作家 1764 冊の書籍から，評価用には，45 人の作家の 271 冊から抽出し，その中の 28 人の作家の 219 冊の書籍は学習用と評価用で共通に使用した．**エラー! 参照元が見つかりません。**に学習用の不適切オブジェクトを含むコマの作家 762 人の参照数の TOP30 のリスト，付録 5-2 に評価用の不適切オブジェクトを含むコマの作家 45 人の参照数の TOP30 のリストを示す．

不適切オブジェクトを含むコマ画像は出現頻度が低く収集が困難だったため，適切なコマ画像の数に比べてかなり少ない．この比率のままで学習を進めると，不適切画像の出現頻度が 1/7 程度となるため不適切画像の特徴抽出が進みにくい．そこで，学習時に不適切オブジェクトを含む画像の出現頻度を適切な画像と同程度の頻度になるようにサンプリング頻度を調整して Cross Entropy Loss を最小化するように学習を進めた．

表 5-2. アノテーションを使った場合と，使わないで学習した場合の各モデルの Accuracy（%）

Annotations	データ種別	train				test			
		resnet14	sppnet1_0	sppnet1_1	sppnet2	resnet14	sppnet1_0	sppnet1_1	sppnet2
With Annotations	適切	95.2	93.4	98.0	**97.8**	97.7	95.8	99.2	**99.2**
	不適切	93.0	95.8	96.5	**96.7**	94.2	97.0	97.1	**97.3**
	平均	94.1	94.6	97.3	**97.3**	95.9	96.2	98.1	**98.2**
Without Annotations	適切	82.9	85.0	89.3	**92.7**	98.1	97.2	**99.3**	99.2
	不適切	93.0	923	**96.4**	96.1	80.8	81.0	84.2	**84.7**
	平均	88.9	88.6	92.8	**94.4**	89.5	89.1	91.7	**92.0**

表 5-2 に前述の不適切オブジェクトの中心領域座標（胸の露出に伴う乳首を含む四角形領域の座標）のアノテーションを使って学習した場合と，使わないで学習した場合を resnet14（[He 16]で提案された resnet のモデル），sppnet1_0, sppnet1_1，sppnet2 の各モデルで学習した結果を示す．

さらに，**図 5-7** に resnet14, sppnet1_0, sppnet1_1　及び**図 5-8** に sppnet2 の各モデルの概略の構成を示す．

sppnet1_0 は resnet14 をベースに，中段と終段の間に SPP 層を挿入した構造である．sppnet1_1 は resnet14 の初段の先頭の AvgPool で入力画像を 1/2 のサイズにダウンサンプルする処理を行わずに，そのまま入力するように変更した以外は sppnet1_0 と同じ構成の SPP 層を挿入した構造となっている．sppnet1_0, sppnet1_1 に挿入した spp 層は，中段の特徴量抽出層からの出力（1）と，そこから繋がった 2 つ目の resblock の出力（2）と，さらに繋がった 2 つ目の resblock の出力（3）の 3 か所から，解像度の異なる特徴量を抽出し，conv＋AdaptiveMaxPool を使ってそれぞれ 256×1×1 の特徴量に変換し，さらにそれらを結合して 768 個の固定長の特徴量に変換して出力する．続く終段の全結合層を使った分類層では spp 層からの特徴量を分類し，オブジェクトの判定を行う．

sppnet2 は sppnet1_0 及び sppnet1_1 に挿入した spp 層を両方並列に挿入した構造で，2 系統の spp 層のそれぞれ 3 か所の抽出箇所からの特徴量を同一レベルの抽出箇所ごとにそれぞれ加算し

たものを結合して 768 個の固定長の特徴量として出力し，終段の分類層に接続する．終段の全結合層を使った分類層は．sppnet1_0, sppnet1_1 と同一の構造である．**表 5-2** に示した結果から，以下が確認できた．

(1) 追加のアノテーションの有無の結果から，全てのモデルにおいて，学習に追加のアノテーションを使った方が，使わないよりも精度が高い．
(2) resnet14 と sppnet1_x との比較から，spp 層があった方がより精度が高い．
(3) sppnet1_0 と sppnet1_1 との比較から，初段の先頭の AvgPool によるダウンサンプル層は無い方の精度が高い．
(4) sppnet2 と sppnet1_1 との比較から，1 系列の spp 層よりも 2 系列の spp 層を両方使った方の精度が高い．

上記の結果から，本システムの運用時には，sppnet2 を採用した．

5.4 不適切画像の目視判定

画像検出 AI の性能は，学習時に使用した画像セットの特徴に依存し，その画像セットに含まれていなかった（例えば，その時点で発見されていなかった，または，存在していなかった）特徴を持つ画像に対する応答は検証できない．特にマンガ画像の場合は，現実の人物や物体では決して実現できないポーズや形状の変形などの非現実的な描画も可能で，さらにその画像の状況説明のために実在しない線や物体などを効果として付加することも可能である．そのため，それらの多様なポーズや形状の変形までをあらかじめ予測した学習用の不適切画像のセットをあらかじめ用意できないことは明白であり，この点は，潜在的な問題である．

不適切画像の検出ツールは，完全に自動であることが望ましいが，前述の問題もあり，「この不適切画像検出 AI では 100％完璧に検出できる」とは断言できないので，当面は，オペレータが最終判定を目視で行うという運用ルールを設け，この作業を効率化するための補助ツールとして運用することにした．そのため，本システムではあらかじめ AI ツールで処理した大量のコンテンツの最終目視判定を，複数の作業者が分担して同時に行えるような構成にした．

不適切画像の検出結果 3 冊の表示例を**図 5-9** に示した．書籍 1 冊ごとの情報は簡潔に1ブロックにまとめ，複数冊の結果を一画面上で一覧表示した．1 ブロック内には 12 個のサムネイルの表示領域を用意し，先頭位置は書籍内容を表現していると思われる表紙のサムネイルを常に固定表示し，残りの 11 個の領域に検出された不適切オブジェクトの候補を含むコマのサムネイル画像を検出スコアの高い順に表示した．スコアは 0～1.0 までの範囲の数値で，1.0 に近いほど不適切である可能性が高いと推定できる．本システムでは 0.5 を超えた場合に不適切オブジェクトを含むコマ画像であると判定している．スコアが 0.5 未満のコマはサムネイルとしては表示せず，スコアが 0.5 以上の不適切オブジェクト含むコマが 12 個以上存在する場合は上位 11 個のサムネイル表示で打ち切った．上位 12 番目以降の画像を確認する場合は，ページ画像を表示して確認する．サムネイル表示領域を 12 個としたのは，実際に判定作業を行ったメンバーからのフィードバックで，11 コマ(表紙を入れると 12 コマ)のサムネイルの確認だけで作業上の問題は無く，書籍の判定は可能だったこと，また後述の確認用に選択した書籍セットでの数値で 10 コマ目までで 99.4%は確定できたこと，および，目視作業に使用するディスプレイの物理的な画像解像度からレイアウトを考慮して割り当てた．通常の目視判定はサムネイル画像のみで行うが，表示が小さく判定が難しい場合は，サムネイルを拡大表示して精査する．

5.4.1　　システム導入効率の向上

コマ抽出および不適切画像検出 CNN の演算を高速に効率よく行うためには GPU が搭載されているマシンが必須だが，検出結果を目視判定するためだけであれば GPU は不要である．本システムでは，2 種類の CNN の実行および結果の蓄積は GPU を搭載した推論サーバ内部で行い，その作業指示および結果の判定は RPC 通信で接続された GPU が搭載されていない作業者用のマシン上で実行する．複数の作業者が同時にサーバーマシンに接続して検出結果の確認ができるため，システムの導入に際し作業者用に新たに GPU を搭載したマシンを準備する必要はない．そのため追加投資は最小限に抑えることができ導入のためのハードルを低くすることができた．

1 冊の書籍としての不適切判定は，書籍中に 1 コマ以上の不適切画像が検出されたか否かであり，不適切画像を検出した時点でその書籍は不適切と判定する．検出スコア順に表示されたサムネイル画像を目視確認し，不適切画像と判定した時点でその書籍はそれ以降の残りのコマ画像の確認は不要となる．従来の目視確認の手順では，すべての書籍に対し，先頭のページから巻末までを不適切オブジェクトを含むコマ画像が見つかるまで順に確認していたが，本システムを使用する場合は，検出スコア順に表示された最も不適切である可能性の高いコマ画像のサムネイルから順に目視確認するので作業効率が良い．

不適切オブジェクトを含む可能性のあるコマのサムネイルが表示されない書籍の場合は，その書籍の確認は不要となり大幅に作業時間を低減できる．

1 冊の書籍中の不適切画像オブジェクトの出現頻度は低いので，従来の目視確認作業は不適切ではない画像が連続する単調作業となってしまうことが多く，長時間に亘る同一の単調な作業を継続するので，特に疲労から集中力を維持できなくなり，注意力の低下による誤判定や，見落としの要因となっていた．

本システムでは，事前にバッチ処理で多数の書籍の判定指示を行い，サムネイルを検出スコア順に一覧表示させて結果を目視確認する運用なので，従来と比べて目視確認の作業量は大きく減少し，見落としの発生もほとんどなくなった．

不適切画像の検出性能の確認のために用意した，不適切画像 1,452 冊を含む 2,196 冊の書籍セットは，不適切画像を含む書籍の割合が多い書籍セットになっているので，書籍セットの一般例とは言えないが，参考のための一例として検証した結果を以下に記載する．

この書籍セットにおいて，全書籍 (適切画像と不適切画像が混在) 中で 1 コマ目の目視確認のみで不適切画像を含む書籍であると判定できた書籍は 58. 6%，3 コマ目までの目視確認で判定できた書籍は 63.7%，5 コマ目までの目視確認で確定できたのは 64.8%，さらに，10 コマ目までの目視確認で確定できたのは 66.7%であった．

同様に，この書籍セットにおいて，不適切オブジェクトを含む書籍中では，1 コマ目の目視確認のみで不適切画像を含む書籍であると判定できた書籍は 88.6%，3 コマ目までの目視確認で判定できた書籍は 96.3%，5 コマ目までの目視確認で確定できたのは 98.1%，さらに，10 コマ目までの目視確認で確定できたのは 99.4%であった．また，10 コマ目までで判定できなかった書籍が 9 冊あったのでそれらはすべてのページを目視で判定した．

図 5-6 に，1 か月毎の不適切画像の確認のために要した 100 ページ当たりの作業時間（秒/100 ページ）と確認した書籍の総ページ数を集計し，それぞれ導入月の数値を 100％として表示したグラフを示す．導入後 4 か月目には導入前の作業工数の 20％以下に大きく減少し，作業の効率化・省力化に大きく貢献できたことがわかる．

5.5 おわりに

インターネット上にはきわめて多くの写真画像やユーザ生成画像コンテンツが溢れており. これらの画像を自動的に不適切:NSFW(Not Safe For Work)と適切:SFW(Safe For Work)に自動分類するシステムの重要性は増している. このシステムの先行例として, Mahadeokarら[Mahadeokar 16]はCNNを使用し, NSFW画像をより正確に自動的に分類できる方法をopen sourceとして提案している. ただし, このシステムの入力画像としては1ページの写真やイラストなどを想定していて, マンガ画像は対象外となっている. 本稿では入力画像をマンガ画像に限定し, まず1ページ中の複数のコマ領域に描画されている画像をそれぞれのコマ単位に分割し, 次にそのコマ単位で不適切画像の自動検出を行う方式を提案し, 構築した.

写真のように実在する物体を写し撮った画像と比較して, マンガ画像は, 紙に印刷して販売された従来スタイルのマンガ書籍は主にモノクロで手書きの線画像が主体なので, 元々の画像に含まれている情報量が少ないうえ, 吹出し内や画像の背景などに文字画像が高い割合で混在している. このような特殊な画像であってもCNNを使用することで高精度に不適切画像を検出できることを示した. いままでマンガの電子書籍の制作工程では, すべてのページの画像を先頭から順に目視確認する方法以外に不適切画像を判定する方法はなかったが, 本稿の不適切画像の検出システムを取り入れることにより, 従来, 作業者が行なわなければいけない単純な作業の膨大な繰り返しを削減し, 作業時間の削減及び結果の品質向上などが期待できることを示した.

不適切画像判定のCNNの学習に使用した画像セットは複数の販売中のマンガ書籍から目視によって抽出したが, 不適切画像の出現割合は低い. さらに検出性能向上のために追加で付加した不適切オブジェクトの中心座標のアノテーション付け作業もすべて手作業である. より精度の高い検出性能を得るためのmodel構造の決定や学習パラメータのチューニングと同様に, 品質の良い学習用のデータセットを確保する事は独自のCNNを学習して活用していくうえで特に重要な作業と言える.

本システムは, 多種ある不適切と思われる性表現の分類中で, 特に女性の胸の露出のみを不適切画像として検出することを目的として開発したので, それ以外の不適切表現には対応できない. 今後は, それ以外の不適切表現にも対応できることも求められているがそれらのいずれの不適切表現の場合も, 実在するマンガ書籍中における出現頻度は極めて低いと思われるため, 学習用の大量の不適切画像を確保するのは本システムでの同様の作業に比べて極めて困難なので, 画像の自動生成などの別の手法とも組み合わせ, 効率的に学習用の画像を用意する必要があると思われる.

謝辞

本章を記述するにあたり,『ココナッツAVE』の作家の三浦みつる氏,『恋はグーチョキパー』の作家の御童カズヒコ氏, 及び『狐のお嫁ちゃん』の作家のBatta氏からは快く画像掲載の許諾をいただき, 文中に挿入させていただいた. この場を借りて深く御礼申し上げます.

また, CNNの学習用の画像は, 著者が株式会社イーブックイニシアティブジャパン在職中に販売していたマンガの電子書籍画像からランダムに選択して収集した. なお, 株式会社イーブックイニシアティブジャパンの経営陣からは, これらの学習用画像の利用および, システムの導入効果データ取得の協力および本稿の発表の許諾をいただいた. この場を借りて深く感謝いたします.

さらに, システム構築及びCNNの学習時には数々のアイデアを提供いただき, また, そのための実験を支援していただいた永冨一也氏には深く感謝いたします.

図表

狐のお嫁ちゃん(4)©Batta

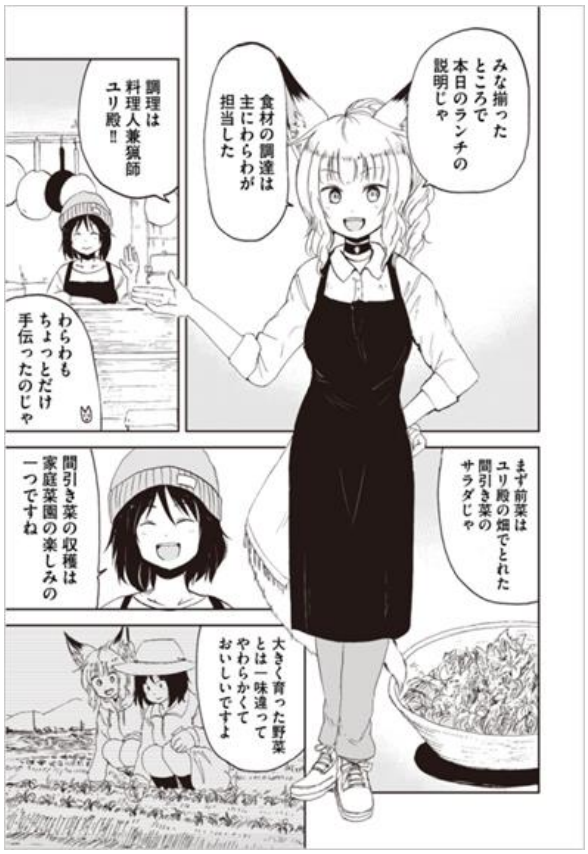


図 5-4. コマの境界線を跨いでいるオブジェクトの例

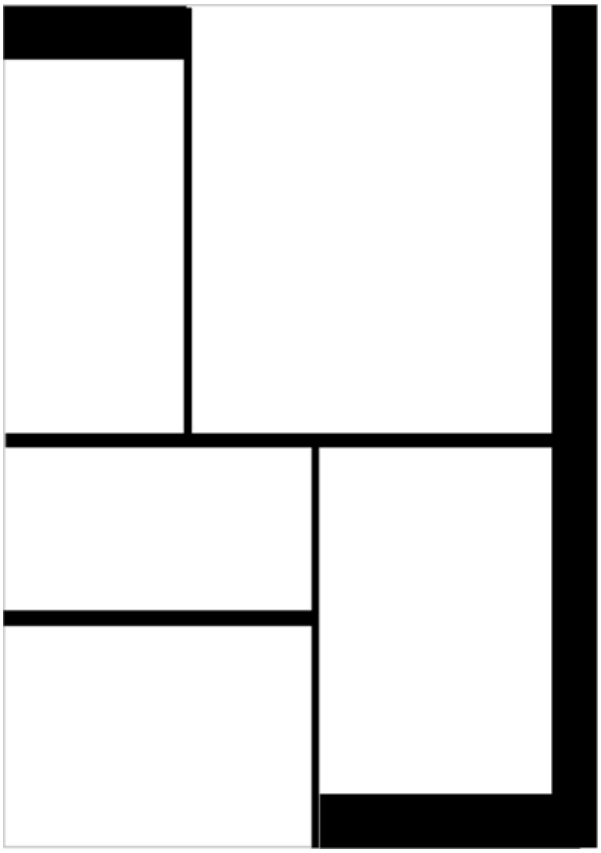


図 5-5. コマ領域推定のための学習用マスク画像例

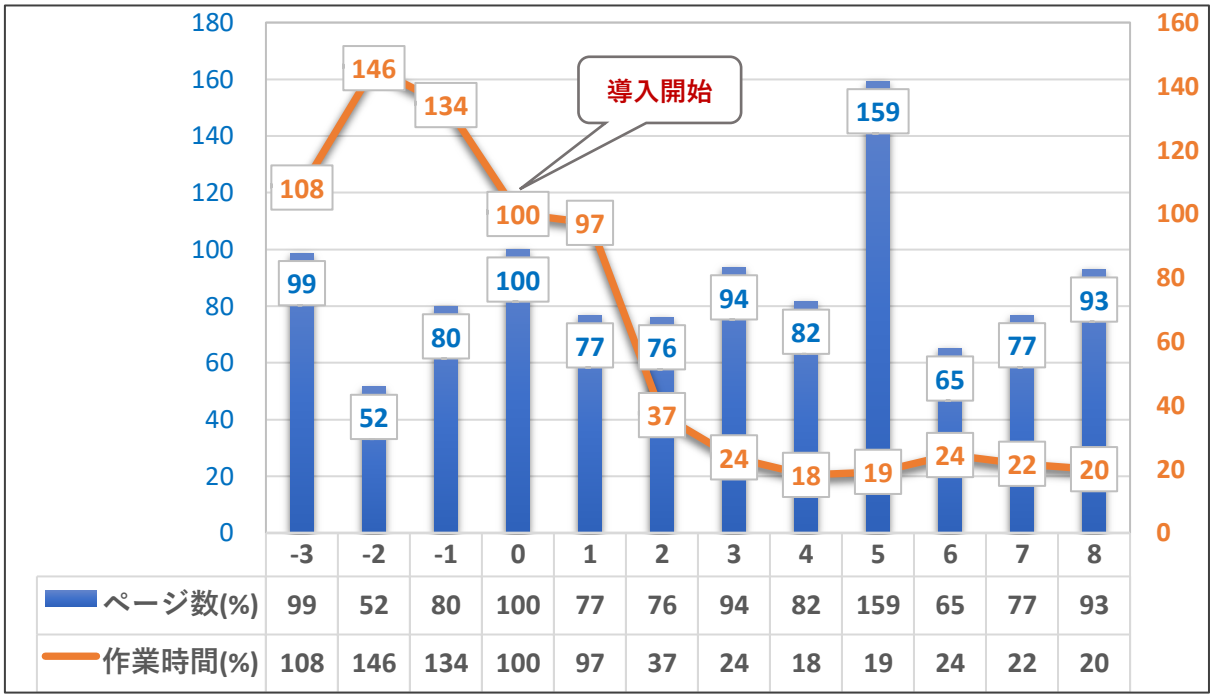


図 5-6. システム導入前後の作業量(総ページ数)と 100 ページ当たりの確認作業時間
(導入開始月を 100 とした相対値)

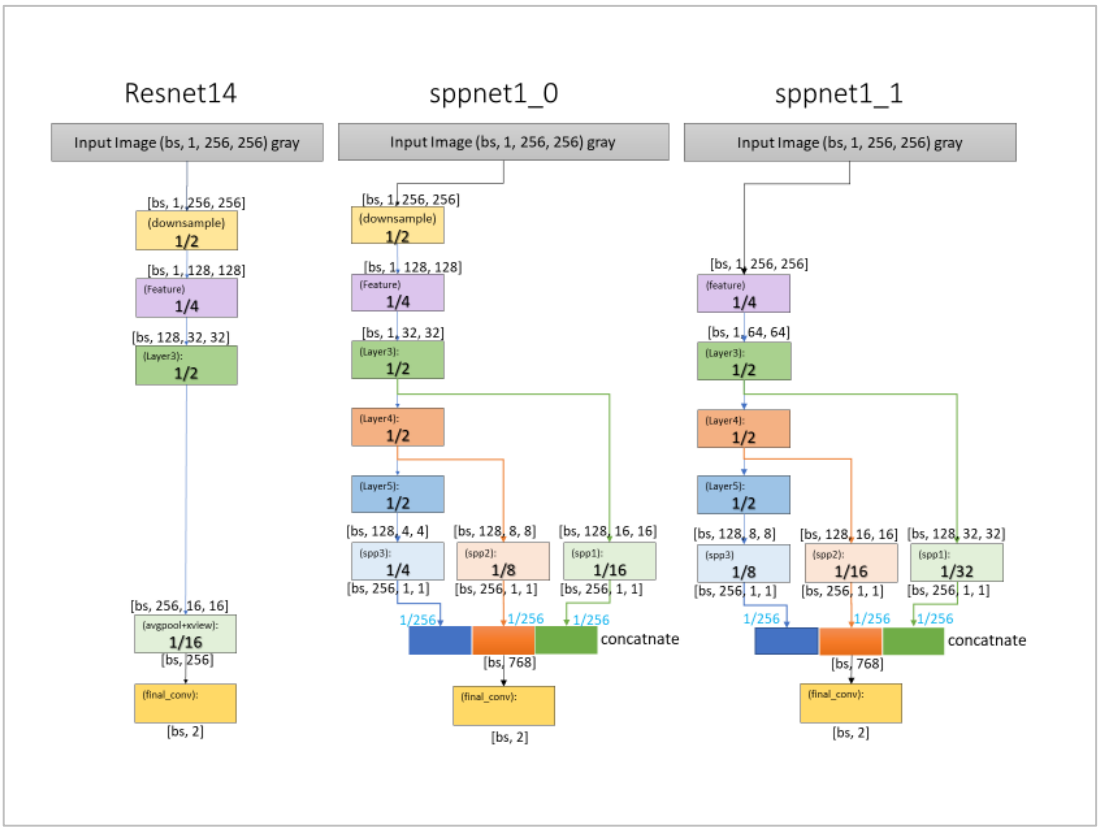


図 5-7. resnet14, sppnet1_0, sppnet1_1 の概略構成

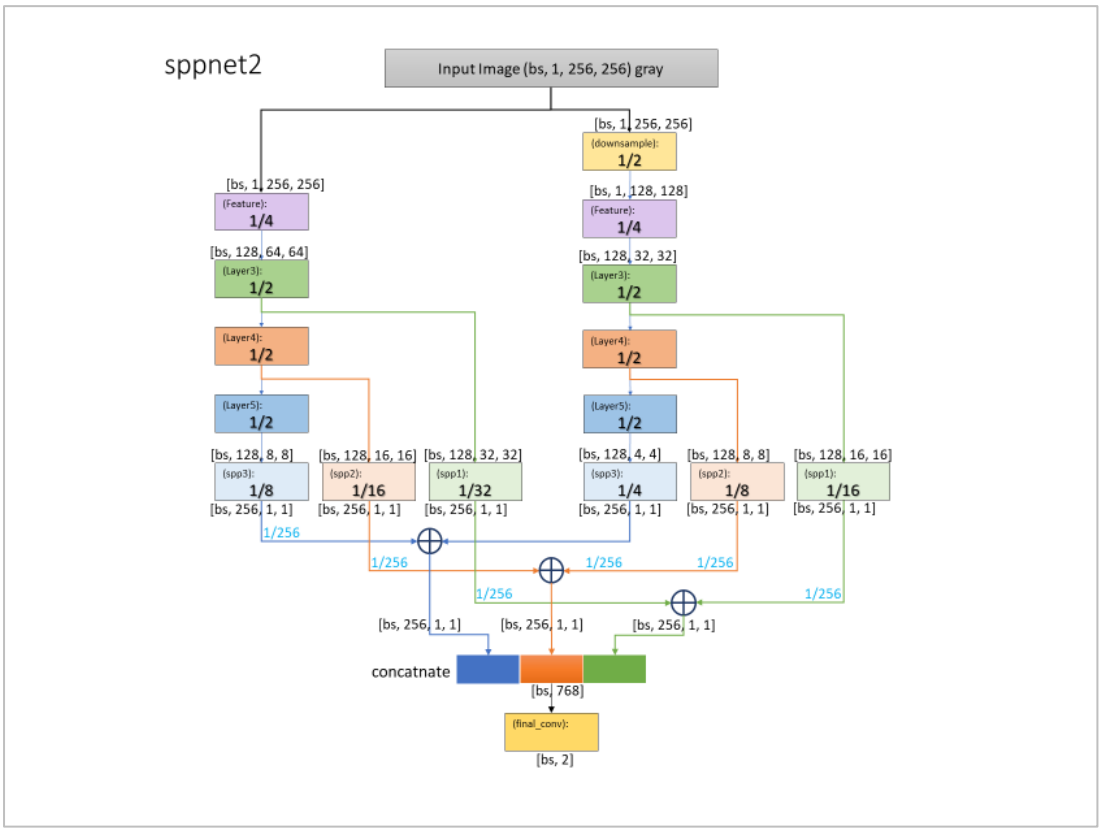


図 5-8. sppnet2 の概略構成

恋はグーチョキパー©御堂カズヒコ
ココナッツ AVE. ©三浦みつる
狐のお嫁ちゃん©Batta

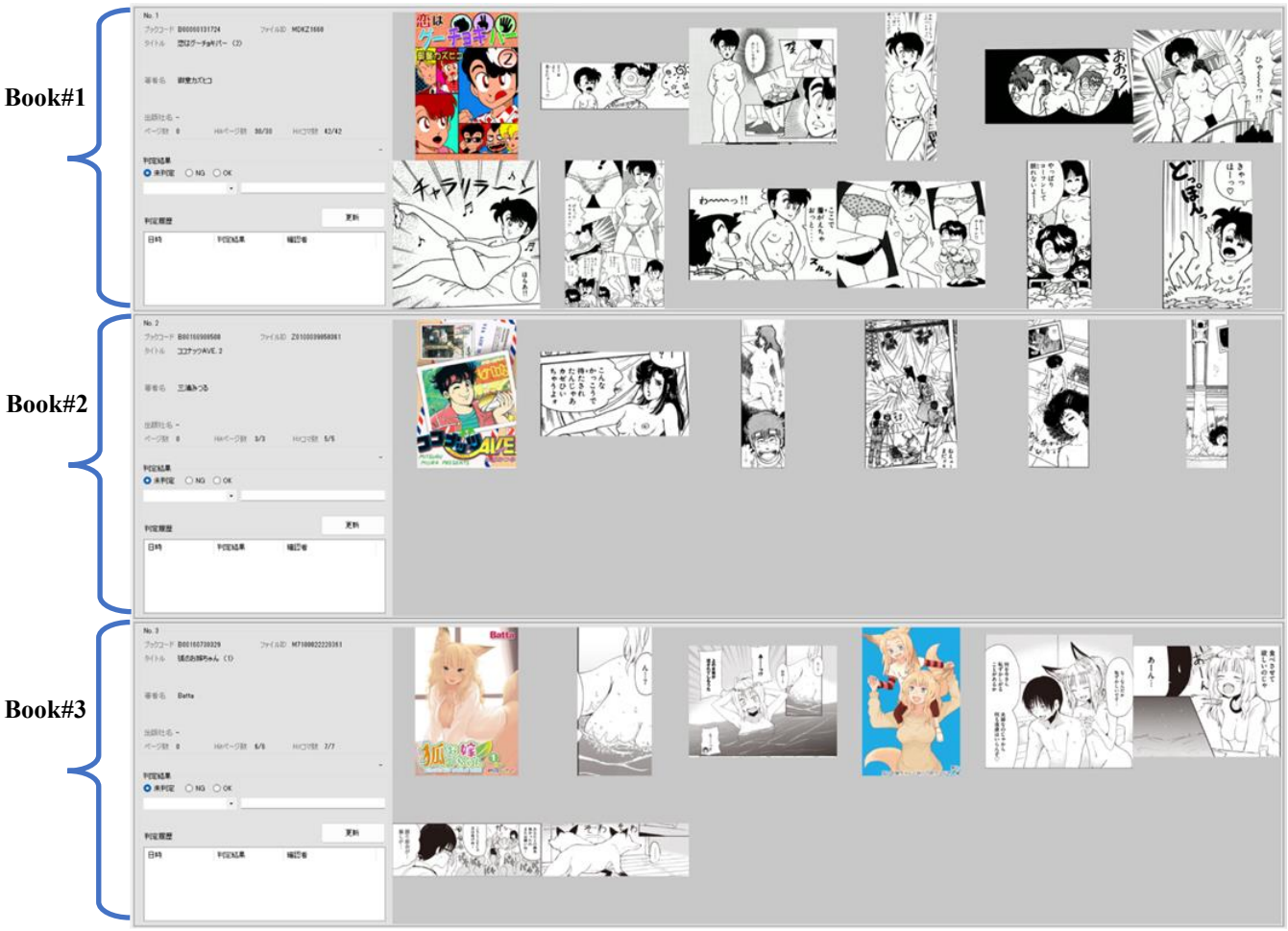


図 5-9. 不適切画像候補のサムネイル表示例

付録 5-1　学習用の不適切オブジェクトを含むコマの作家 762 人の書籍の参照数のリスト TOP30

No.	作　家	書籍 参照数 （冊）
1	克・亜樹	77
2	弓月光	31
3	村生ミオ	23
4	小幡文生	22
5	きづきあきら＋サトウナンキ	21
6	こしばてつや	20
7	作画：山口正人／原作：川辺優	18
8	弘兼憲史	16
9	高橋留美子	14
10	高橋のぼる	14
11	井上淳哉	12
12	池田ユキオ	12
13	本名ワコウ	12
14	北川みゆき	12
15	春輝	11
16	石川サブロウ	11
17	荻野真	11
18	作画：石井さだよし／原作：星野茂樹	10
19	平松伸二	10
20	画：Tetsu／作：Yoshi	10
21	山本英夫	10
22	米原秀幸	9
23	画：井上紀良／原作：小池一夫	9
24	三浦建太郎	9
25	のりつけ雅春	9
26	大和正樹	9
27	葉月京	8
28	大海とむ	8
29	松本光司	8
30	漫画：佐藤健悦／原作：吉野弘幸	8

付録 5-2. 評価用の不適切オブジェクトを含むコマの作家 45 人の参照数のリスト TOP30

No.	作 家	書籍 参照数 （冊）
1	克・亜樹	77
2	弓月光	31
3	村生ミオ	22
4	春輝	11
5	本名ワコウ	11
6	暁	8
7	Boichi	8
8	画：井上紀良／原作：小池一夫	8
9	山口貴由	6
10	葉月京	6
11	作画：カマキリ／原作：小池一夫	6
12	仙道ますみ	6
13	作画：黒瀬浩介／原作：蝸牛くも（GA 文庫／SBクリエイティブ刊）／キャラクタ原案：神奈月昇	5
14	遊人	5
15	稲光伸二	4
16	きただりょうま	4
17	めいびい	4
18	漫画：宵野コタロー／原作：LINK	4
19	黒澤 R	3
20	作画：唯浦史／原作：明月千里（GA 文庫／SB クリエイティブ刊）／構成：渡辺樹／キャラクタ原案：春日歩	3
21	漫画：氷樹一世／原作：蘇我捨恥（ヒーロー文庫／主婦の友社）／キャラクタ原案：四季童子	3
22	漫画・キャラクタ原案：雪月佳／原作：木野裕喜	3
23	作画：七波のろ／原作：みやすのんき	3
24	林崎文博	2
25	art：小畑健／story：大場つぐみ	2
26	永井豪	2
27	クール教信者	2
28	もんでんあきこ	2
29	劇画：横山まさみち／原作：沢田一矢	2
30	漫画：ラサハン／原作：kt60	2

第6章　マンガ画像中の文字画像領域の抽出

6.1　はじめに

マンガは，コマの中の画像と，コマの中に記述されたセリフ文字，および，背景画像中に埋められた文字によって，物語のストーリを紡いでいく．マンガの内容を理解するために，ストーリを形成する重要な要素の1つとして，マンガ中に表示されている文字をすべて抽出し，テキスト情報として取り出すことを検討する．

マンガのテキストの大部分は吹き出しの中およびキャプションの中に在するため，これらの要素の位置を特定することが OCR の前提条件となる．しかし，これらは異なる機能を果たすため，異なるクラスとして分類する必要がある：キャプションは通常，物語を説明する目的で使用されるが，吹き出しには通常，マンガの登場人物の直接的な会話や思考が含まれる．

文字画像のままでは活用できないので，文字画像を認識しテキスト化する技術である，OCR（Optical Character Reader）を使ってマンガの中に表示されている画像化された文字情報をテキストの形で取り出すことを考える．

そこで，まんがの画像中に混在する文字を含むセグメント領域を認識・抽出し，そのセグメント領域を，OCR が認識しやすい状態の画像で取り出すことを考える．

OCR に関しては，入手が容易なオープンソースの Tesseract-OCR の利用を前提とする．ただし，元々の OCR はビジネス文書や，小説，新聞など一定の方向で，同一のフォントで，一定のレイアウトで表示されている文章を対象としているため，マンガのセリフ文章のような短い口語表現で，任意のレイアウトで表現され，画像と文字が混在している場合に対しては，その認識性能が充分ではないことも考えられるので，その場合はできる範囲で Tesseract-OCR のチューニングを行い，読み取り精度の向上をはかり，その問題点などを考察する．

なお，各種のマンガ画像から文字を含むセグメント領域を抽出し，ノイズ削減した結果画像は，著作権の兼ね合いで本章には掲載できないため文章だけの記述になっているのをご了承願いたい．

6.2　背景

Tesseract は元々，1985 年から 1994 年の間，イギリスのヒューレット・パッカード研究所（Hewlett-Packard Laboratories Bristol UK）とアメリカのコロラド州グリーリーのヒューレット・パッカード社（Hewlett-Packard Co, Greeley Colorado USA）で開発された．2005 年，Tesseract は HP によってオープンソース化され，2006 年から 2018 年 11 月までは Google によって開発された．2021 年にリリースされたメジャーバージョン 5 が現在の安定版で，今もなお開発が進められている．

Tesseract-OCR は，日本語，中国語を含む 2byte コードの unicode 文字にも対応し，100 以上の複数の言語環境にも対応しており，国内でも商業的にも利用実績がある，オープンソースの OCR として広く知られている．Tesseract-OCR は，行認識に焦点を当てたニューラルネット(LSTM)ベースの OCR エンジンが追加されているため，ある程度はチューニングが可能である．

また，より良い OCR 結果を得るには，Tesseract に与える画像の品質を向上させる必要がある．HP には，以下のように複数の細かな記述がある．

(1)　明るい背景に暗いテキスト（白地に黒がベター）であり，背景の輝度は一定であること．

(2) DPI が 300dpi 以上の画像で最適に動作する設計であること.
(3) ある程度のノイズは, Tesseract が内部で除去するが, できるだけノイズのない鮮明な画像であること.
(4) 太字や細い文字は, 細部の認識に影響を与え, 認識精度を低下させるので, 一定の太さの文字であること.
(5) ラインセグメンテーション品質が低下するので文章の向きに傾きやゆがみが無いように水平な文章であること.
(6) 余分な文字として誤認識される可能性があるため, 文字領域の周りに余分なグラデーションや境界線は無いこと.
(7) Tesseract が認識に役立てるために使用する単語リストに専門的な単語を追加すること.

上記の記述を満たすように, マンガ中の文字領域を抽出する仕組みを前処理として構築する.

6.3 実験 phase＃1

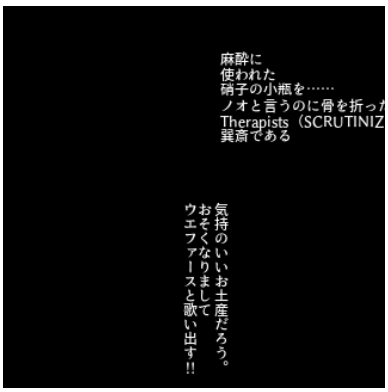
画像中の文字領域を抽出する手法に関しては, 3 章の超解像の研究で判明したように, 文字を主体とする小説のような画像とそれ以外の画像はその特徴量が違うため, CNNで分離できることは予想できる. さらに超解像ではCNNでノイズ削減が可能だった事も併せて考えると, 画像領域を抽出した場合に, 領域内に残る背景画像の断片などのノイズも同時に分離できそうなことは推測できる.

実験 phase#1 では, 既存の OCR を利用する事を大前提として, OCR への入力画像の条件が最良になるように, 入力画像中からピクセルレベルでテキスト領域を求めるセグメンテーションをベースとした CNN を学習し, それにより文字領域を抽出し, OCR でテキスト化するという方向で検討しその性能を検証する.

CNN は, 既存の OCR の変換精度が最良になるであろうと思われる, 鮮明な文字画像の抽出及び文字以外のノイズを除去したセグメント画像を取り出すための処理をする. CNN の学習用に, 以下のデータセットを作成し, 学習を行う. CNN は, 入力画像(x) から 文字画像(y1) の文字のピクセルのみを抽出するセグメンテーションを学習し, 同時に補助タスクとして入力画像(x) から 文字領域画像(y2) の変換も学習し, 文字が無い領域に発生する余分なノイズを抑制する.

6.3.1 phase＃1のデータセット

実験 phase＃1のデータセットとして, 以下の 4 種類の画像セットを準備した.



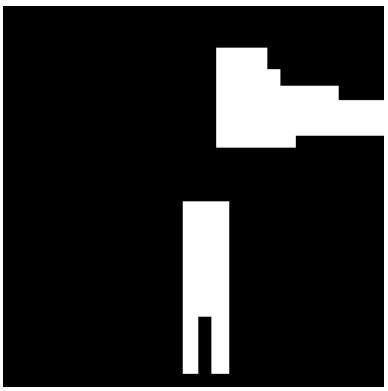


図 6-1. Phase#1 のデータセット
入力画像(X),　文字画像(Y1, Y3:Y1×2 倍),　文字領域画像(Y2)

(1) 入力画像(x): 文字を含むマンガページ中の一部領域を模した画像
(2) 文字画像(y1): 入力画像中の文字のピクセルを白，それ以外を黒とする画像
(3) 文字領域画像(y2): 入力画像中の文字列を囲う領域のピクセルを白，それ以外を黒とする画像
(4) 超解像文字画像(y3): y1 の 2 倍の解像度の画像

入力画像(x)の背景に使っている画像は，複数のマンガからランダムに抽出した背景画像として利用できそうな文字が含まれていない領域の画像で，Train 用に 3,545 カット，Test 用に 73 カット 用意した．データ作成時には，ストックされた画像からランダムに抽出し，各種変形をしながら使いまわした．画像中の任意の位置に描きこまれている文字列は，青空文庫及び ePub の複数の読み物からおよそ 100 冊程度抽出し，その中の文章から，ランダムな長さを切り出してきて，背景画像の任意の位置に貼り付け，マンガページ中の一部領域をシミュレートした画像を生成する．

文字列の表示は，ランダムに選択した吹き出しの内部，および，吹き出しがなく画像の中に埋まった状態の画像を用意した．

文字画像(y1)は，文字のピクセルを白で，それ以外を黒で表示した画像で，入力画像に表示した文字列を表示したのと同じ位置に表示した画像である．また，同時に 2 倍のサイズの文字画像 (y3)を用意し，入力文字画像の超解像 2 倍拡大を学習する場合に使用する．実作業としては，こちらの 2 倍のサイズの画像を作成してから 1/2 サイズにした文字画像(y1)を作成する．

文字領域画像(y2)は，入力画像中の文字列を囲う領域のピクセルを白，それ以外を黒とする，文字の外側を囲う四角形領域画像である．

6.3.2 文字画像抽出 CNN のネットワーク構造

文字領域を抽出するために，マンガのページから文字のピクセルのみを抽出するセグメンテーションを学習する．2 つの U-Net を直列に並べた**図 6-2** に示す CNN を使用する．先頭の U-Net では上記の入力画像(x) を入力として 文字画像(y1) 及び 文字領域画像(y2)に対応する 2 チャンネル(z1，z2)の出力を行う．

2 つ目の U-Net では，先頭の U-Net の出力(z1，z2) を入力とし，文字画像(y1)に対応する 1 チャンネル(z3)の出力を行う．

学習時は，(z1 + z3， z2)の BCE（Binary Cross Entropy）損失を最小化する．（z1+z3）が文字画像(y1)に対する出力(ただし，z3 は z1 に対する差分修正)で，z2 が 文字領域画像(y2) に対する出力である．先頭の U-Net で文字画像(y1) 及び 文字領域画像(y2)を出力できるように学習しながら，その情報を使って 2 つめの U-Net でより精度の高い文字画像(y1) に対応する出力ができるように学習する．

文字領域画像(y2)を使うことで，文字領域以外の領域に発生するノイズを抑えることができる．

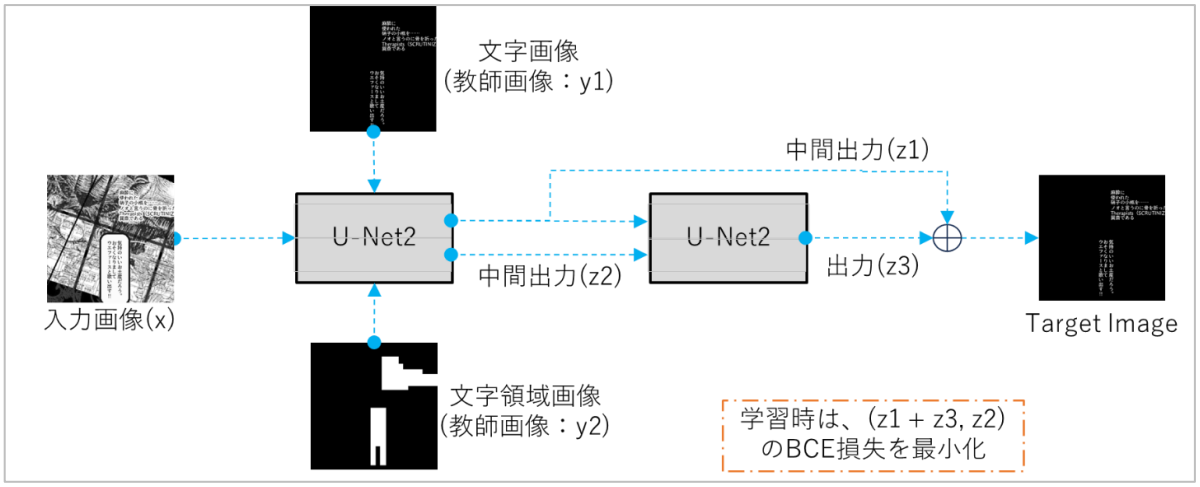


図 6-2. 文字領域の抽出ネットワークの構造

6.3.3 実験 phaase＃1のデータ(x, y1, y2)の生成

データの生成は，以下の順で行う．

1） ストックの背景画像からランダムに選択し，変形を加えてベースとなる背景画像を生成する
2） 描画文字列を生成(日本語のセリフ，改行位置選択，漢数字，カナの置換など)
3） 描画フォントと サイズ，レイアウト(縦書き，横書き，吹き出し，縁取り文字など)を選択．
4） 1)の背景画像上に 2)の設定で 3)を描画 (x)
5） 描画する文字列を(x)と同じ設定で，黒背景に対して教師データ y1，y2 を描画
6） 2 つ目の文字列・描画位置を選択し，1 つ目と重なっていなければ 2 つ目を描画

(1) 背景画像の生成(BackgroundGenerator) 及び 背景画像の変形

背景画像は，以下の 3 パターンで生成する．

a） 用意された背景画像から 1 枚選択して変形を加えて生成する．
b） 1 の方法で変形を行った 2 枚の背景画像を生成して合成する．
c） 1 の方法で変形を行った 3 枚の背景画像を生成して合成する．

合成は，画像Aの上に，黒を不透明で白を透明として画像Bをアルファブレンドする．白を背景色に黒い線画を重ねるイメージになる． 2 枚の合成を 2 回行うことで，3 枚の背景画像の合成が可能となる．

背景画像の変形 a） は，以下の手順で行う．

(i) 画素値の最小最大が 0-255 になるように正規化する．
(ii) 画像の縁に境界線を描画(太さ,色はランダムに選択)する．
(iii) 画像を 2 倍のサイズになるように reflection padding (画像の端から外に向けて，内側へ向かうパターンを反転させて展開)する．
(iv) ランダムに回転・リサイズ・左右反転を行う．
(v) ランダムな位置から使用する規定のサイズ(320x320)に切り抜く．

(2) 描画文字列の生成（TextGenerator）

描画する文字列は，青空文庫から抽出したセリフの文字列，および 複数の ePub の書籍から抽出したセリフの文字列， および Ubuntu 標準の英語単語リストからアポストロフィを含む単語を除いた英単語のリストから以下の手順で生成する．

I. マンガの吹き出しを模して使用する文の長さや改行位置を選択する．
II. 青空文庫は英語単語や数字の出現頻度が少ないと思われるので，漢字を適当な英単語や数字の列に置換する．
III. マンガは!?…などの記号がよく使用され，またその記号の抽出精度がよくないので，ランダムに文末に記号を追加する．

I. ～III. で生成した文字列に改行の情報までを，描画する文字列とする．

(3) 使用するフォントの生成（FontGenerator）

使用するフォント・サイズを選択して，描画用の ImageFont.truetype オブジェクトを生成する． 同じフォント・サイズが何度も使用されることになるので，生成結果はキャッシュして選択結果が同じだった場合はキャッシュを返すようにする．

描画するフォントの種類は，マンガでよく使用されるアンチック体(漢字がゴシック体でひらがなが明朝体)の FONT_ANTIQ， ゴシック体の FONT_GOTHIC， 明朝体の FONT_MINCH，ゴシック体の FONT_MPLUS など複数のフォント（ライセンスが不要）からランダムに選択する．

フォント・サイズは，それぞれのフォントごとに小・中・大の 3 つサイズの範囲を定義しておき，定義した確率によって選択する．選択する 確率はマンガのページや結果を観測して調節した．

アンチック体が多めで，縦 1200 のページ画像の場合にはフォント・サイズ 26 くらいが吹き出し内でよく使われていると想定して調節した．フォントのベースサイズはマンガによっても異なるので，ある程度ランダムに対応する．

(4) 教師データ(x,y1,y2)の生成（TextImageGenrator）

以下の手順で，教師データを生成する

1） BackgroundGenerator で背景画像を生成
2） TextGenerator で描画する文字列を生成
3） FontGenerator でフォントを生成
4） レイアウト(縦書き横書き,吹き出しありなし)，フォントカラー，描画位置，字間を選択して背景画像の上に文字列を描画(x)
5） 描画する文字列を x と同じ設定で黒背景に対して y1,y2 を描画
6） 2 つ目の文字列・描画位置を選択し，1 つ目と重なっていなければ 2 つ目を描画

6.4 実験 phase＃2

実験 phase#2 では，phase#1 で行った文字領域の抽出精度を改善するために，方式を根本的に見直して，シーンテキスト認識で使用されている，CRAFT（Character Region Awareness for Text Detection）[Youngmin 19]という文字の検出方式を採用する．CRAFTは，文字画像のセグメンテーションベースから，1 文字ごとの文字領域を局所化し，検出された文字をテキストインスタンスに結びつける方式で，画像内の個々の文字を特定するための文字領域スコアと，各文字を 1 つのインスタンスにグループ化するために使用する親和性スコア画像（文字と文字をつなぎ合わせる領域）を推定する CNN で構成され，この両スコア画像を二値化してコネクテッドコンポーネント（OpenCV の connectedComponentsWithStats()） 関数のようなラベリング処理を適用することで，連続する小領域に分割されたラベリング画像が得られ，同時にラベル付けされた各小領域の位置，幅，高さ，面積，重心のステータス情報が得られる．それらの情報により，各文字の領域と文字間の繋がりの情報（行の情報）が得られるので，その行の情報を使って文字画像領域を抽出し，既存のOCRでテキストを得て，その性能を検証する．

6.4.1 phase＃2 のデータセット

実験 phase＃2 のデータセットとして，以下の 5 種類の画像セットを準備した．

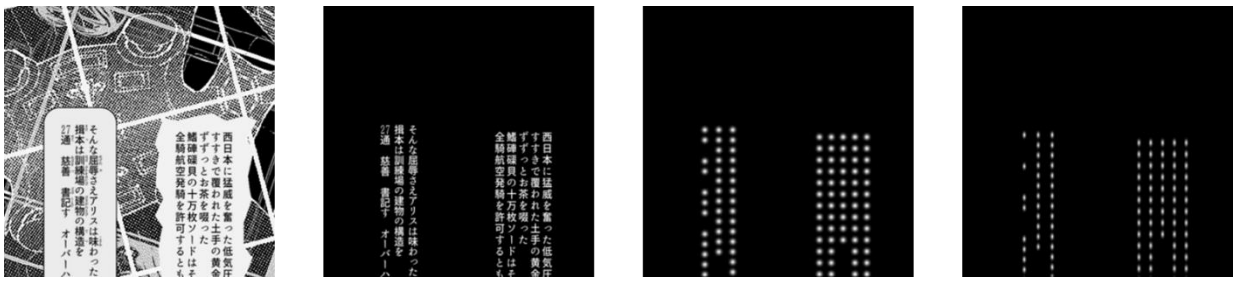


図 6-3. Phase#2 のデータセット
入力画像（X） 文字画像（Y1,Y4:Y1×2 倍） テキストスコア(Y2) リンクスコア(Y3)

(1) 入力画像(x): 文字を含むマンガページ中の一部領域を模した画像
(2) 文字画像(y1): 入力画像中の文字のピクセルを白，それ以外を黒とする画像
(3) テキストスコア画像(y2): 各文字の中心位置に 2 次元ガウシアンを描画した画像
(4) リンクスコア画像(y3): 各文字の繋がり位置（文字と文字の間）に 2 次元ガウシアンを描画した画像
(5) 超解像文字画像(y4): y2 の解像度 2 倍版(先にこちらを作ってから 1/2 縮小版を y1 に)

CNN は，入力画像(x) から 文字画像(y1) の文字のピクセルのみを抽出するセグメンテーションを学習し，同時に補助タスクとして，テキストスコア画像(y2) 及び リンクスコア画像(y3) を出力するセグメンテーションを学習する．

テキストスコア画像(y2) 及び リンクスコア画像(y3) を閾値で二値化処理した画像を作り，OpenCV の connectedComponentsWithStats() 関数などを適用することで，ラベリング画像（連続する小領域に分割），さらにラベル付けされた各小領域の位置，幅，高さ，面積，重心のステータス情報

が得られる. それにより, 各文字の領域と文字間の繋がりの情報(行の情報)を得る. これらの情報を使って, 文字領域を抽出する.

6.4.2 phaase＃2 のデータ(x,y1, y2, y3,y4)の生成

データの生成は, 以下の順で行う.

1) ストックの背景画像からランダムに選択し, 変形を加えてベースとなる背景画像を生成する
2) 描画文字列を生成(日本語のセリフ, 改行位置選択, 漢数字, カナの置換など)
3) 描画フォントと サイズ, レイアウト(縦書き, 横書き, 吹き出し, 縁取り文字など)を選択.
4) 1) の背景画像上に 2)の設定で 3)を描画 (x,y2,y3,y4 を描画)

描画の形式は, 吹き出し, 吹き出しなし(縁取り文字), 目次のいずれかから選択する. このデータでは, 縦書き横書きが混在する.

またノイズとしてルビも描画する. ルビは, y1,y2,y3,y4 には含まれず, 除去(無視)する目的で描画する.

(1) 背景画像の生成(BackgroundGenerator) 及び 背景画像の変形

背景画像を生成します. 6.3.3 の(1)と同じ

(2) 描画文字列の生成(TextGenerator)

描画する文字列を生成する.
読み込んだテキストファイルから 1～10 行をランダムに選択する.

(3) 使用するフォントの生成(FontGenerator)

使用するフォント・サイズを選択して, 描画用の ImageFont.truetype オブジェクトを生成する. 同じフォント・サイズが何度も使用されることになるので, 生成結果はキャッシュして選択結果が同じだった場合はキャッシュを返すようにする.

描画するフォントの種類は, マンガでよく使用されるアンチック体(漢字がゴシック体でひらがなが明朝体)の FONT_ANTIQ, ゴシック体の FONT_GOTHIC, 明朝体の FONT_MINCH, ゴシック体の FONT_MPLUS など複数のフォント(ライセンスが不要)からランダムに選択する.

(4) 教師データ(x,y1,y2,y3,y4)の生成(TextImageGenrator)

以下の手順で教師データ(x,y1,y2,y3,y4)を生成する

1) BackgroundGenerator で背景画像を生成
2) TextGenerator で描画する文字列を生成
3) FontGenerator でフォント設定を生成
4) レイアウト(縦書き横書き,吹き出しありなし), フォントカラー, 描画位置, 字間を選択して背景画像の上に文字列を描画(x)
5) x と同じ設定で y1,y2,y3,y4 を描画
6) 2 つ目の文字列・描画位置を選択し, 1 つ目と重なっていなければ 2 つ目を描画

6.5　CRNN ベースの OCR

実験 phase#2では，画像中から抽出する文字領域を，デノイズ（背景除去），文字画像の超解像，CRAFT ベースの文字及び行の情報抽出を行って，文字領域画像を抽出し，既存の Tesseract-OCR を用いてテキスト化するという処理でそのテキスト化性能を検証した．

CRAFT により『吾輩は猫である（改行）名前はまだない…』という文字列を抽出する場合の概念図を図 6-4 に示す．

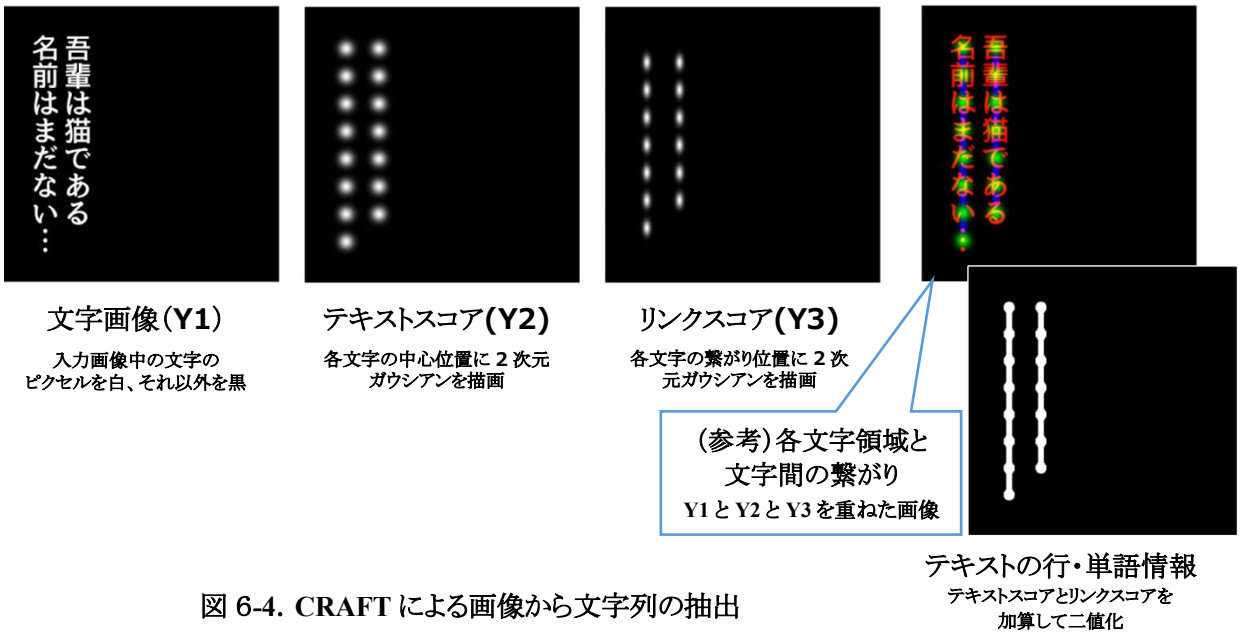


図 6-4. CRAFT による画像から文字列の抽出

Tesseract-OCR は，日本語を含む 50 言語以上の複数言語に対応しているとはいえ，その開発主体が西欧及び米国である．そのうえ，開発主体の人々は左から右に流れる横書き言語を日常使用しているに対し，おそらく開発主体の人々はほとんど知らないであろうと思われる，日本語という全くその言語構造が異なっているうえに使用されている文字の種類も，漢字，ひらかな，カタカナ，英数字，記号文字，絵文字その他など，その数も膨大にあり，さらに，文章は縦書きが主体（マンガの吹き出しの中はほとんどが縦書き）で右上から左下に流れる，日本語での OCR 実験を行ったことになる．

実験を開始する前から．日本語ではある程度の問題がある可能性も想定していたが，おそらくそれらはチューニンで改善が可能だと想定していて，マンガ用にも充分な性能であろうと期待していたが，それだけでは改善ができない部分が多数存在していることがわかった．

実験の phase#2 では phase#1 に比べて，前処理部分の性能向上はでき，さらに Tesseract-OCR の LSTM 部分も含むチューニングを行ってみたが，結果としては大きな改善は見られず，これ以上の精度向上は困難であると思えた．

そこで，テキスト化のアプローチを根本的に見直し，既存の Tesseract-OCR を使わずに，新たに CRNN (Convolutional recurrent neural network) [Shi 15]と呼ばれる構造をベースに OCR 部分をより自由度の高い CRNN（CNN＋GRU）ベースの新しい OCR を開発し，phase#2 で行った前処理部分のデノイズと超解像拡大処理を分離せずに新アーキテクチャの OCR の内部に吸収し，CRAFT ベースの文字・行情報の抽出処理のみを前処理で行うという構成に変更する．これにより，phase#1 および phase#2 の前処理の途中で失っていた（と思われる）入力画像中にあった（はずの）文字に関する

詳細情報を，OCR 内部の CNN で直接扱うことができるようになるため，それによる性能向上が期待されると考えた．

また，CRNN-OCR については，将来容易に追加が可能な訳語彙（文字）数の問題よりも，文字単位の情報が正しく OCR の入力として伝達できるかに主眼を置いて評価した．

6.6 OCR 精度の比較

実際のマンガ書籍 22 冊を選択し，書籍に含まれる目次や扉などの，コマを使わないページの画像を分類のための CNN によって除外し，マンガ本文に分類されたページの文字画像のみを使った評価実験を行った．マンガ書籍ページの分類を**表 6-1** に示す．

OCR 変換の結果について**表 6-3** に示す．コマを使わないページを除外するのは，ページによってはデザイン性の高いレイアウトで，特殊なレイアウトや文字フォントなどが使われることも多いため，基本的な OCR 性能を評価するために除外した．評価のために使用したマンガ書籍の一覧表を**表 6-2** に示す．なお，前処理結果についての画像に関しては，著作権の問題があり，掲載できないことを断っておくが，ほとんどのケースで，背景画像と文字の分離は当初の目標どおり，白地の背景に黒字の文字が描かれた，カルタの読み札のような形状での抽出が可能であったが，一部，濁点をノイズと誤認識するケースが見られた．

残念ながら，ユーザサービスにすぐに使えるレベルの実用的な精度で，マンガ画像中の文字画像のすべてをテキスト化する仕組みの構築までは実現できなかった．

表 6-1. マンガ書籍中のページ種別分類

分類	クラス番号	内容
本文	**0**	**マンガの本文**
アート_表紙	1	書籍の表紙
アート_扉	2	本文の最初にある扉ページ,タイトル等が表示されるページ
アート_カバー	3	カバーをスキャンした画像、折返しは除く
アート_あとがき	4	あとがきの中にある絵がメインの画像
カバー_折返し	5	カバーの折返しをスキャンした画像
目次	6	目次が含まれる画像
解説	7	人物紹介・地図等の解説ページ
解説_予告	8	次巻の予告等の解説ページ
小説_あとがき	9	小説または文字の割合が多いあとがきページ
奥付	10	書籍の作者・発行元等が記述してある奥付ページ
奥付 2	11	配信元のロゴ・定型文等のページ
その他	12	白紙・ロゴなど内容を持っていないページ

表 6-3 から推測すると，より OCR の精度を向上するためには OCR に入力する画像の前処理の部分も OCR の内部に取り込んだ方が，変換精度を向上できるように思える．実際には，前処理をそのまま取り込むのではなく，OCR の CNN の構造の中に前処理で行っている機能を取り込み，直接，前処理に入力されている元々の画像に含まれている情報を使って OCR を行うことで，例えば，ノイズ削減処理で本来は必要な濁点をノイズと誤認識して削減してしまうことなどで，認識率は変わってしまうことなどを味する．

表 6-2. 精度測定に使用したマンガ書籍の一覧

注） 書籍名の後の括弧()は，その書籍シリーズの巻数を示す.

No.	書籍名	著者
1	フラジャイル （10） 病理医岸京一郎の所見	漫画:恵三朗 原作:草水敏
2	世界の果てにも風は吹く （1）	ハズミツカサ
3	ブラック・ジャック （10）	手塚 治虫
4	エンジェル・ハート （10）	北条 司
5	シティーハンター （1）	北条司
6	シティーハンター （10）	北条司
7	賭博黙示録カイジ （10）	福本伸行
8	はじめの一歩 （10）	森川ジョージ
9	蒼天航路 （10）	漫画:王欣太 原案:李學仁
10	修羅の門 （12）	川原正敏
11	樹魔・伝説 （2）	水樹和佳子
12	サラリーマン金太郎 （10）	本宮ひろ志
13	ゼロ THE MAN OF THE CREATION （76）	原作:愛英史 漫画:里見桂
14	鬼滅の刃 （10）	吾峠呼世晴
15	この素晴らしい世界に祝福を！(1)	作画:渡真仁 キャラクター原案:三嶋くろね 原作:暁なつめ
16	ONE PIECE モノクロ版 （10）	尾田栄一郎
17	三国志 （10）	横山 光輝
18	海の闇、月の影 （16）	篠原千絵
19	ギャラリーフェイク （32）	細野不二彦
20	名探偵コナン （10）	青山剛昌
21	最上の命医 （10）	漫画:橋口たかし 取材・原作:入江謙三 医療監修:岩中督
22	ゴルゴ 13 （182）	さいとう・たかを

表 6-3. OCR の性能比較において，コンテンツ情報の OCR 対象文字数は OCR 対象となる文字の数で 1 文字をごとにカウントしている．また，フラグメント数は一塊として検出される文字ブロックで，多くは，吹き出し 1 つを1フラグメントとしてカウントしているが，ひょうたん型の横に長い吹き出しなどは，文章の塊を 1 つとしてカウントするため，フラグメント数は 2 になっている場合がある．

Tesseract-OCR を使った場合と，CRNN-OCR を使用した場合について，それぞれ以下の条件で，OCR 結果の精度を計測した．

(1) Tesseract-OCR（v501）をそのまま使い，入力画像は CNN（phase#1 のデータで学習）による文字抽出及びノイズ削減の前処理を行った文字領域セグメントを入力
(2) Tesseract-OCR（v501）に対し，チューニング及び LSTM の再学習を行い，CNN（phase#2 のデータで学習）による文字抽出及びノイズ削減の前処理を行った文字領域セグメントを入力し，Tesseract-OCR が出力した文字列の行判定に CRAFT のデータを使用．
(3) CRNN ベースの新 OCR を使い，(2)と同じ CNN（phase#2 のデータで学習）による文字抽出及びノイズ削減の前処理を行った文字領域セグメントを入力し，CRNN-OCR が出力した文字列の行判定に CRAFT のデータを使用．
(4) CRNN ベースの新 OCR の内部に文字抽出及びノイズ削減の処理を統合し，CRNN-OCR が出力した文字列の行判定に CRAFT のデータを使用．

表 6-3. OCR の性能比較

No.は表 6-2. 精度測定に使用したマンガ書籍の一覧実験に使用したマンガ書籍の一覧の No.に対応.

No.	CRNN（CNN＋GRU）-OCR ベース				Tesseract ベース				コンテンツ情報				
	(4) CRNN-OCR +前処理を内部吸収 +行検出 CRAFT		(3) CRNN-OCR +前処理 phase#2 +行検出 CRAFT		(2) Tesseract +LSTM など調整 +前処理 phase#2		(1) Teeseract オリジナル +前処理 phase#1		OCR 対象		ページ画像情報		
	char (%)	frag (%)	char (%)	frag (%)	char (%)	frag (%)	char (%)	frag (%)	文字数	フラグメント数	ページ数	幅 pix	高さ pix
1	0.41	2.19	1.14	5.99	4.62	21.03	6.99	30.83	11,306	1,051	198	1125	1600
2	0.63	3.49	0.93	3.88	3.22	14.29	6.75	42.37	19,691	1,546	248	1350	1920
3	0.19	2.01	0.45	4.09	2.90	24.75	5.75	50.27	32,433	1,689	239	1432	2048
4	0.66	3.97	0.65	4.55	6.74	39.53	11.53	63.18	17,127	1,032	205	720	1024
5	0.49	3.77	1.00	8.25	5.05	30.61	12.22	77.37	18,923	1,140	181	648	1024
6	0.94	5.35	2.15	8.93	6.69	35.32	11.20	69.31	22,377	1,512	183	648	1024
7	2.33	9.30	4.92	14.97	14.72	51.34	20.75	67.65	15,998	1,496	221	1365	2048
8	0.52	2.67	0.76	3.92	5.22	28.05	8.26	43.63	13,649	1,123	186	1042	1600
9	0.50	3.64	1.00	6.69	4.94	34.81	11.16	66.37	17,341	1,017	250	816	1200
10	0.64	6.37	0.82	7.80	5.12	48.79	8.21	65.49	21,741	910	201	784	1024
11	0.89	5.85	1.19	9.52	4.24	28.44	6.40	48.80	33,690	1,881	189	816	1200
12	0.55	4.50	1.59	10.64	6.84	44.93	11.83	64.28	15,772	977	222	1112	1600
13	0.66	4.70	1.46	10.66	4.68	37.14	6.76	55.11	27,920	1,341	198	800	1200
14	0.58	3.91	1.28	7.20	7.93	38.38	12.13	57.32	9,933	792	207	764	1200
15	0.27	2.30	1.08	8.39	5.21	41.36	6.75	55.04	18,477	1,001	182	1128	1600
16	1.12	6.61	3.59	17.40	14.25	58.05	19.49	78.79	13,983	1,075	194	760	1200
17	0.18	1.01	0.20	1.89	2.04	21.44	4.67	44.89	16,005	793	210	1063	1600
18	0.29	3.42	0.35	3.80	2.49	16.60	5.56	39.80	10,832	789	186	656	1024
19	0.79	4.95	1.25	9.48	5.10	41.94	8.40	60.52	57,849	3,176	320	832	1200
20	0.18	1.93	0.65	6.71	3.65	26.89	7.80	55.20	31,204	1,759	185	752	1200
21	0.27	1.67	0.52	3.72	3.33	35.77	4.39	44.81	29,531	1,560	192	768	1200
22	0.17	1.50	0.52	5.20	5.29	45.01	8.84	65.74	28,471	1,404	256	844	1200
Ave.	**0.589**	**3.974**	**1.170**	**7.653**	**5.260**	**34.895**	**8.774**	**56.789**	**484,253**	**29,064**	**4,653**	–	

OCR 出力の平均の精度比較の char err（文字単位のエラー）および fragment err（吹き出しまたは文字ブロック単位のエラー）を抜き出した結果を**表 6-4** に表示する.

表 6-4. OCR 出力の平均の精度比較

OCR の実験方式（表 6-3 の番号）	(4)	(3)	(2)	(1)
Average character error (%)	0.589	1.170	5.260	8.774
Average fragment error (%)	3.974	7.653	34.895	56.789

上記の表からわかるように，あくまで，選択した 22 冊の書籍の場合の平均だが，(1)の方式では，57%割の吹き出しの文字に誤りを含んでいたのが，(4)の方式では，およそ 4%にまで，改善できた．また，文字単位では，(1)の方式の場合は，およそ 9%(1,000 文字に 88 文字)のエラーを含んでいたのが，(4)の方式では 0.6%(1,000 文字に 6 文字)の性能まで改善することができた．

表 6-3 の中で，7 番及び 16 番の書籍の認識エラーが低くならないのは，この書籍に使用しているフォントのグリフが学習に使用したグリフの形状と異なっている部分が多いことが原因と思われる．

また，OCR 対象のフラグメント数に対して文字数が多い，3，11，20，21，22は，解説などの説明などの比率がおおく，19 番はセリフが多いため，文字数も多くなっている．さらに，10，12，14，17，18 番は，セリフの数は少ないが，長めのセリフが多いマンガと言える．

6.7 まとめ

マンガという特定のジャンルの画像中の文字を OCR するための実験を繰り返す課程で，マンガに特化した原因とする問題と，そもそもの日本語の OCR に関係した問題があると思われるので次に整理する．

6.7.1 マンガの特徴に起因すると思われる問題点

一般的な，文書における日本語文字画像のテキスト化においては，その使用頻度が低いため，あまり問題にならないと思われるが，マンガでは高い頻度で発生するケースがある，特に以下の表現の文字化に対しては，改めて何らかの対応が必要だと思われる．

(1) 漢字の読みかなとしてのルビの表現．ルビの文字の大きさが不定
現在は，ルビをノイズとして削除して OCR によるテキスト化をしたが，将来は漢字に付けられたひらかな・カタカナのルビも正しくテキスト化する必要がある．
一般的な小説などにも，ルビは出現するが，ルビが付いた文字画像

(2) 縦中横と呼ばれる，縦書きの 1 文字分の領域に，英数字記号の 2～3 文字を横に並べた表現
Dr.， Jr., Mr., Ms., Mrs., Sir., Sig., m^2, m^3, cm, cm^2, cm^3, Kg, AD, BC, AM, PM, VS., vs., No., O_2, H_2, Na, Ca, Zn, H_2O, CO_2, !!!, ???, !!?, !?, など
これらの一部の単位記号文字は，マンガ以外でも使用されているが，マンガの吹き出しなどでは表示領域が狭いため，特に高い頻度で使用される．
また，同様に，2 桁～3 桁の縦中横に組まれた文字．10～99，100～999．

(3) 主人公の名前等に使用される，名前用の漢字．JIS X 0213 以外の漢字．冴羽獠の”獠”

(4) 同一文章ブロック中に存在する，大きさが大きく異なる文字，漢字

(5) 同一ページ中に複数の字体の異なる文字，漢字

(6) 50 音表には存在しない，濁点・半濁点が付いたひらかな，カタカナ，”あ゛”，”ぬ゛”，

(7) 文末の長い長音記号 ”—”

(8) 文末の長い”．．．”

(9) ハート型などの記号文字，絵文字

(10) 表示スペースの関係で，文字間隔が特に狭い文章がある

6.7.2　　日本語に起因すると思われる問題点

日本語という極めて複雑な言語において，何よりも，大きな問題は，日常使われる表現の文章の中に漢字，ひらかな，カタカナ，アルファベット，アラビア数字などが必要な部分に混在している．そのため，英語のように文字の種類が 26 文字しかない場合の OCR を構築するのと比べて，はるかに複雑さが増すことは明白である．

一般的に使われている漢字としては，常用漢字と人名用漢字を合わせて約 3,000 字あり，漢字検定の 1 級の対象範囲の JIS 第一水準と第二水準（JIS X 0208）は，約 6,355 字あり，JIS 第三水準と第四水準（JIS X 0213）には 5,801 字の漢字がある．さらに，大修館書店で出版されている大漢和辞典では，約 50,000 字の漢字が記載されている．

また，日本語独自の問題として，以下の類型のフォントの問題があると思われる．見た目は同じような文字でも，検索の場合は全く異なるので，日常生活ではあまり意識されずに脳内で正しく認識されていることが多いので，問題にはならない場合が多いが，検索で使用する場合は大きな問題となる．ただし，これらの問題は日本語 OCR などの日本語処理などに携わっている研究者には既知の問題として捉えられていると思われるが，以下は実際に，マンガ画像で実験したときに，見つかった項目である．これ以外にも．日本語独自の問題は複数ありそうである．

(1) 複数のデザインされた漢字フォントが存在する．
(2) アルファニューメリック文字，記号文字に，半角表現と全角表現があり，文章中に混在している括弧，記号などで 半角と全角がある．{}，()，！，？，＜＞，{}，()，!，?，<>など
(3) カタカナの ロ と 漢字の口（くち）
(4) カタカナの リ と ひらかなのり
(5) カタカナの ニ と 漢数字の二（2）
(6) カタカナの カ と 漢字の 力（ちから）
(7) 促音（そくおん）の っ（小さい文字） と 標準の つ
(8) 拗音（ようおん）の ゃゅょ（小さい文字）と 標準の やゆよ
(9) 長音記号 ー と ハイフン ― または 漢数字の一（いち，1）
(10) ○（漢数字ゼロ） と ○(white circle)
(11) 丸数字 ①，②，．． 括弧数字（1），（2），．．
(12) 人（ひと） と 入（はいる）
(13) グリフのデザインで，ひらかなの濁点，半濁点の位置が，線の上が標準に対し線の下もある．げ，ざ，ず，だ，ぢ，で，ど，ば，ぼ，ぱ，ぽ，など

さらに，日本語文字フォントのデザインは多数あり，また，英文が混在した場合の英文用のフォントも多数あり，そのうえ両フォントには等幅フォントとプローポーショナルフォントが存在している．
このように膨大にある全てのデザインのフォントを CNN で学習することは難しいので，フォントのバリエーションを自動生成するなどの，新しい学習用データの生成技術が必要であると思われる．

OCR のプロトタイプの開発実験の課程で見つかった問題点を述べてきて，欧米諸国に比べて日本の文書の電子化が思うように進まない原因の一つに，日本語 OCR の性能の低さがあるのではないかと思われた．

今まで，紙に印刷された文書を保存するためにマイクロフィルム撮影や，ドキュメントスキャナーでデジタル画像化するなどして，保管場所の容積を削減することはしてきたが，残念ながらそれらの保存された画像内の文章を表示し，目視での確認は可能だがそのままでは検索に使うことができない．つまり，膨大にある過去のドキュメントの多くが，文書データの電子化ができないため，死蔵され活用できないことを意味し，アーカイブに大きなコストをかけたことが充分生かされずに，同時に，日本のデジタル化が大きく遅れていることを意味する．

一般的な技術として認識されているOCRは，まだ日本語の環境では十分な性能ではないことを，改めて認識し，このままでは，ますます非漢字文化圏である西洋諸国に対し，デジタル化が大きく遅れてしまうような危機感を感じた．

第7章　結論

マンガの画像を主たる研究対象の画像データとして，各種CNNをDeep Learningを使ってそれぞれの目的用に学習し検証した．各章で得られた成果は以下の通りである．

7.1　電子書籍用画像の超解像拡大

シングルフレーム超解像拡大は古くからある手法で，低解像度の画像パッチから辞書を使って高解像度のパッチを求めて高解像度の画像を生成していた．CNN を使用した超解像拡大はこの辞書方式の超解像拡大の辞書部分を CNN に置き換え，学習済の CNN が低解像度の画像から高解像度画像を推測していると考えられる．

CNN は，低解像度画像とそれに対応した高解像度画像のペアを使い，低解像度画像を入力した CNN の出力が限りなくペアの高解像度画像と等しくなるように学習する．そのため，学習済みの CNN に対し，学習に使用した低解像度画像に類似した画像特徴量を持つ低解像度画像を入力した場合には，学習時に使用したペアの高解像度画像と類似の特徴を持つ高解像度画像を出力することが期待できる，ということは容易に予測できる．逆に，学習に使用した低解像度画像の特徴量とは大きく異なる画像を学習済みの CNN に入力した場合には，期待した精度の高解像度画像は出力できないであろうことも類推できる．これに対し，マンガ画像以外のデータセットを使った学習済みの超解像拡大の CNN では対象とする画像の特徴量が大きく異なるため，マンガの画像は精度よく拡大できないことは実験からも確認できた．

従って，現時点においては，実用的な時間と規模で電子書籍のさまざまな画像スタイル（画像特徴量のバリエーションが広い）の低解像度画像を高精度に超解像拡大を行うためには，入力となる低解像度画像をグループ化し，それぞれのグループごとに低解像度と高解像度の画像ペアでCNNを学習し，入力画像の特徴に一番近い画像グループの学習済 CNN を選択して，超解像拡大を行う仕組みの構築は合理的だと思われ，同様の複数のスタイルを有する画像の超解像など CNN を利用したシステムを構成する場合には特に有用な構成である．

さらに，計算時間が問題にならずにより高精度の CNN の利用が可能であれば，システムの構造は変更せずに同様の仕組みを使って，CNN 部分の入れ替えのみで精度の向上を計ることは容易だと思われる．

近い将来は，GPU や CPU の性能向上がさらに進展することはほぼ確実なので，GAN を代表とする生成系の CNN などを利用し，さまざまな画像スタイルの教師画像の自動生成などを活用して学習することで，さまざまな画像スタイルの画像に対応できる広い表現力を持った CNN を 1 つだけ使用し，長時間を掛けて大量の画像データセットで学習することで，入力画像のスタイルにかかわらず，実用的な時間で高精度な超解像拡大を行うことが可能になるかもしれない．

7.2　マンガのコマの抽出

複雑な表現力を持つマンガを解析・理解するうえで，マンガのコマの 1 つ 1 つの画像及び吹き出し内の文字を正しく理解するためには，コマを正しい形状で抽出することはとても重要である．

コンピュータを使用してマンガのページ画像中のコマ領域を抽出する問題を，ページ画像中のマンガのオブジェクト，吹き出しや文字などはすべてノイズであると仮定すると，ノイズがとても多い画像の中から，複数の小さな多角形領域を認識し，抽出する問題と捉えることも可能である．

従来のルールベースのイメージ処理では，どうしてもページ画像中の複数の小さな多角形領域をうまく分離することができなかったが， CNN を使用してページ画像中のコマ領域の各 pixel が，コマの内部の pixel であるか？あるいはコマの外部の pixel であるか？のセグメンテーションを CNN で推測し，ページ画像内に存在する連続した小領域に分割することで，コマ抽出のためのマスクを得ることができた．得られた小領域のマスク画像からは，従来のルールベースのイメージ処理技術を用い，多角形ポリゴンに補間して，コマ領域を抽出した．

マンガを構成する要素の基本であるコマを正確な形状で抽出することができ，マンガを分析・理解していく過程において，その基礎部分を正しく扱える環境を整えることが可能となった．

7.3 マンガ画像中の不適切画像の検出

前章で記述したコマ抽出 CNN を活用して，マンガのページ内にある複数のコマを正確に抽出し，順次，抽出したコマの中に不適切画像（露出した女性の胸）が含まれているか否かを検証する仕組みを構築した．多くのマンガ画像はモノクロの線画像が主体なので，写真のような画像に比べ情報量の乏しい画像を認識し区別できるかに関して懐疑的だったので，あらかじめ実験をしてうまく機能することを確認してからシステムを構築した．写真とは大きく特徴が異なるマンガ画像であっても，CNN を適切に学習することで，マンガのようなモノクロの線画像であっても認識する仕組みの構築が可能なことを実証することができた．

不適切画像を検出する仕組みは，大量の不適切画像を含むコマ画像と不適切画像を含まないコマ画像を使ってコマの中に不適切画像が存在するか否かを 2 値分類する CNN を学習し，コマ抽出 CNN から得られる正確なコマ位置と，コマ内の不適切画像の有無から，不適切画像の検出システムを構築した．

本システムは，強力な演算リソースを必要とするコマ抽出CNN，及び不適切画像の認識CNNの演算は GPU を 搭載したサーバ上で実行し，その実行の指示と結果の確認を RPC で接続した複数のクライアント PC から実行する構成とした．従来は目視でしか確認する方法がなかったため，作業者には長時間の単純な作業の繰り返しを強いていたが，この不適切画像検出システムを導入したことで，より良い作業環境が構築でき，目視に掛けていた作業時間も従来の 1/5 程度に大幅に削減でき，作業の効率化が図れた．ただし，CNN の特性上，学習時には意図していなかった特徴量を持つ画像が入力された場合の CNN の応答は予測できないため，分類間違いが起こる可能性を否定できないので，現時点では CNN の結果を 100%信用することはできないことから，あくまで，目視検査の合理化ツールとしての運用に留めている．この予測できない部分を残したままの CNN では産業分野に広く応用していく上での，足かせになってしまう．何らかの方法で，予測できない部分を解明できなければ，厳しい条件の産業分野での CNN 応用は厳しいと思う．この分野に関しては，さらなる研究に期待したい．

本システムのようにサーバにのみ高速演算可能な GPU を搭載し，その GPU を使った AI の推論エンジン部分をマイクロサービス化するようなサーバ・クライアント構成のシステムでは，サーバ側の AI にかかわる部分は機械学習の処理の記述が得意な Python で記述し，クライアント側を GUI 処理の記述が得意な C#や JAVA など，それぞれの用途に合った言語を使って記述することが可能なため，CNN を使った同様のタスクの処理にも応用が可能である．

7.4　マンガ画像中の文字画像領域の抽出

マンガ画像中にはいたるところに画像化した活字文字が存在しており，その大部分は吹き出しの中または背景画像に埋もれた形で存在する．文字画像は OCR を利用してテキスト化が可能だが，文字領域周辺を切り取った断片画像を OCR の入力画像とした場合に，多くの場合は文字領域の周りに吹き出しや背景画像の断片がノイズとして残っていて，OCR の文字認識がうまく機能しないことが経験的にわかっている．

そこで，文字領域周辺を切り取った断片画像には，吹き出しや背景画像の断片が残らないようにする仕組みが必要なので，CNN を使って，文字領域とそれ以外の領域を分離して文字領域を抽出し，文字画像以外をノイズとして削減し，短冊状の白地の背景に黒文字で表示された断片画像を抽出することを目標に，Phase#1 は文字を含む領域に着目し，Phase#2 は，文字そのものと文字の並びの構造に着目し，文字の抽出手法を変えて出力画像を得る実験を行った．

初期の期待どおりに奇麗な断片画像（ノイズを削減した白地に黒の文字画像）の抽出ができることは判明したので，既存の Tesseract-OCR を可能な限りマンガの活字文字画像用にチューニングしたうえで文字化の実験を行ったが，それほど大きな性能向上はできなかった．Tesseract-OCR が充分に日本語に対応できていないことと，OCR の前処理に相当する奇麗な断片画像の抽出処理と，OCR の基本処理である画像からテキスト化の処理を別々に行っているために，十分な性能向上ができないと考えて，双方向 GRU を使った新しい構造の CRNN-OCR を開発し，OCR 処理（画像→テキストへの変換）の内部に，断片画像の抽出処理で行っている処理を取り込み，生の入力画像が持っているすべての情報を使って OCR 処理を行った方が結果的には，精度の高い OCR（画像文字のテキスト化）が可能であろうと考えた．任意に選択した 22 冊の書籍での変換結果を比較してみる限り，この仮定はほぼ正しい，と結論できる．

マンガという特別な画像の中の文字画像の OCR という限定された領域をはみ出してもっと本質的な問題に立ち返って考えてみると，そもそも，日本語という類を見ない複雑な言語を OCR するための画像処理技術に関して，十分こなれた実用レベルに達するには，まだまだ，研究の余地が残っており，我が国のデジタル化の進行状態が，西欧諸国のデジタル化に対してかなり出遅れているという現状も容易に推測できる．長期保存のために画像化した過去にアーカイブされた大量の文書を活用するためのデジタル化を推進するためにも，OCR 技術のさらなる高性能化は必須であると思われ，早急な開発推進の必要があると思われることを僭越ながら付け加えておきたい．

また，本研究では認識が容易と思われた，形式が決まっていて文字ごとに形状の揺らぎが無い活字の文字画像にその研究対象を限定して考えたが，これを手書き文字の認識にまで研究対象を広げたならば，とてつもなく難易度が高くなるであろうことは容易に想像できる．これからのAIを活用した手書き文字認識の研究の発展に大いに期待するところである．

今後，これらの開発に何らかの形で関わっていくことができたのならば，本学で学ばせてもらった身としても幸いである．

謝辞

社会人の身分で博士課程後期進学の機会をいただき，また，著者の至らぬ点を，経験と心の厚みで受け入れて下さりつつ，研究におけるご助言を明晰に，しかも迅速にご指導下さいました，長尾智晴教授に心よりお礼申し上げます．

また，本稿をまとめるにあたり貴重なご指導とご助言をいただきました，白川真一先生，富井尚志先生，原下秀士先生，森辰則先生，に感謝申し上げます．

本研究に際してご支援いただきました長尾研究室の皆様にも感謝申し上げます．研究活動を通じて，さまざまなサポートをいただきました長尾研究室の秘書様方，大変お世話になりました．

さらに，社会人の身分のまま，仕事と研究の両立に関して多大なる理解を示していただき，CNN の研究開発に不可欠な大量の画像データの自由利用を認めてくださった，株式会社イーブックイニシアティブジャパンの元社長の小出斉氏に心から感謝申し上げます．

こうして，本稿を書き上げることができたのは，株式会社イーブックイニシアティブジャパンを退社時に，高橋将峰氏，阿部逸人氏から，退社後の論文出稿を快く許諾していただいたおかげです．本当にありがとうございました．

また，CNN の学習用の画像の選択や，システム運用時の効果測定など，こまごまと協力していただいた，株式会社イーブックイニシアティブジャパンの電子書籍制作チームの三浦氏，木村氏，藤田氏をリーダーとする制作チームのメンバー，さらに，システム構築及び CNN の学習時に有効と思われるアイデアの提供，解決策に関する助言をしていただき，実験を支援していただいた永冨一也氏には深く感謝いたします．

そしていつも見守り支えて貰っている妻にはいくら感謝しても足りません．皆様のお力添えによって何とか本稿をまとめることができました．本当にありがとうございました．

前の職場の身の回りに存在していた，日本が世界に誇れるポップカルチャであるマンガ画像の認識や解析を通して，画像と文字が混在する Mixed ドキュメントを CNN を用いて個人レベルの PC でも認識することができるようになり．その結果，今まではできなかったサービスの創出や作業の効率化が可能になり，それらから得られるメリットは果てしない可能性を有しています．CNN を使った機械学習はさまざまな非効率の解消やこれまでできなかったことを可能にすることができる，次世代を切り開くための新しい手段の一つであると信じています．

本研究で培ったこの力を活用して，これからの世の中を少しでも良くしていくことに貢献できる可能性があると信じ，さらに精進していきたいと思います．

引用文献

[Arai 10] K. Arai and T. Herman.: Method for automatic e-comic scene frame extraction for reading comic on mobile devices. In Proc. ITNG, pages 370–375, 2010.

[Augereau 18] Augereau, O., Iwata, M., Kise, K.: A survey of comics research in computer science. Journal of Imaging 4 (04 2018)

[Ballard 79] Dana H. Ballard, : Generalizing the Hough Transform to Detect Arbitrary Shapes, Technical Report, UR CSD / TR55, University of Rochester. Computer Science Department, 1979, http://hdl.handle.net/1802/13802.

[Chan 07] C. Chan, H. Leung, and T. Komura,: Automatic panel extraction of color comic images. In Proc. PCM, pages 775–784, 2007.

[Dong 14] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang.: Learning a Deep Convolutional Network for Image Super-Resolution, in Proceedings of European Conference on Computer Vision (ECCV), 2014

[Dong 15] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang.: Image Super-Resolution Using Deep Convolutional Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Preprint, 2015

[Dong 16] Chao Dong, Chen Change Loy, Xiaoou Tang,: Accelerating the Super-Resolution Convolutional Neural Network, ECCV 2016

[Duda 72] 6. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM 15, 11–15, 1972

[Dutta 19] Arpita Dutta, Samit Biswas: CNN Based Extraction of Panels/Characters from Bengali Comic Book Page Images, Published in International Conference on 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), 1 September 2019.

[外務省 2016] 外務省(2016), "わかる！国際情勢 Vol.138 ポップカルチャーで日本の魅力を発信！", https://www.mofa.go.jp/mofaj/press/pr/wakaru/topics/vol138/index.html,（参照 2024-01-10）

[グローバルインフォメーション 2023] 株式会社グローバルインフォメーション, ”マンガ市場の成長と向”, https://www.gii.co.jp/report/grvi1300998-manga-market-size-share-trends-analysis-report-by.html , （参照 2023-9-21）

[Han 07] E. Han, K. Kim, H. Yang, and K. Jung: Frame segmentation used mlp-based x-y recursive for mobile cartoon content. In LNCS: Human-Computer Interaction, volume 4552, pages 872–881, 2007.

[He 15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, : Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 37, Issue: 9, Sept. 1 2015.

[He16] Kaiming. He, X. Zhang, S. Ren, and J. Sun,: Deep residual learning for image recognition, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[Ho 11] A. Ho, J. Burie, and J. Ogier,: Comics page structure analysis based on automatic panel extraction. In Proc. IAPR GREC, 2011.

[Hough 60] Method and means for recognizing complex patterns, US patent US3069654A, 1960

[Huang 15] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja: Single Image Super-resolution from Transformed Self-Exemplars, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5197-5206, 2015.
https://github.com/jbhuang0604/SelfExSR/tree/master/data

[Hubert 21] Chieh Hubert Lin, Hsin-Ying Lee, Yen-Chi Cheng, Sergey Tulyakov, Ming-Hsuan Yang: InfinityGAN: Towards Infinite-Resolution Image Synthesis, https://arxiv.org/abs/2104.03963, 2021

[石井 07] 石井大祐, 河村 圭, 渡辺 裕, : コミックのコマ分割処 理に関する一検討, 電子情報通信学会論文誌. D, 情報・システム = The IEICE transactions on information and systems (Japanese edition), Vol. 90, No. 7, pp. 1667–1670 (2007).

[Kim 16] Jiwon Kim, Jung Kwon Lee and Kyoung Mu Lee, et. all, :Accurate Image Super-Resolution Using Very Deep Convolutional Networks", CVPR（Computer Vision and Pattern Recognition）2016 Oral

[Mahadeokar 16] Jay Mahadeokar, Gerry Pesavento, : Open Sourcing a Deep Learning Solution for Detecting NSFW Images, https://yahooeng.tumblr.com/post/151148689421/open-sourcing-a-deep-learning-solution-for, Sep. 30, 2016,
https://github.com/yahoo/open_nsfw

[Manga 15] “Manga109 dataset”, http://www.manga109.org , 2015

[Matsui 16] Y.Matsui, K.Ito, Y.Aramaki, A.Fujimoto, T.Ogawa, T.Yamasaki, K.Aizawa, : Sketch-based Manga Retrieval using Manga109 Dataset, Multimedia Tools and Applications, Springer, 2016

[Nagadomi 19] K.Nagadomi, https://github.com/nagadomi/waifu2x

[野中 09] 野中俊一郎, 沢野哲也, 羽田典久, : コミックスキャン画像からの自動コマ検出を可能とする画像処理技術「gt-scan」の開発, Fuji Film research & development, Vol. 57, pp. 46–49. (2009)

[Ogawa18] Toru Ogawa, Atsushi Otsubo, Rei Narita, Yusuke Matsui, Toshihiko Yamasaki, Kiyoharu Aizawa, : Object Detection for Comics using Manga109 Annotations, arXiv 1803.08670, 26 Mar, 2018

[Pang 14] Pang, X., Cao, Y., Lau, R.W., Chan, A.B.: A robust panel extraction method for manga. In: Proceedings of the 22nd ACM International Conference on Multimedia. pp. 1125–1128. MM ’14, ACM, New York, NY, USA (2014)

[Redmon 16] Joseph. Redmon, S. Divvala, R. Girshick, and A. Farhadi: You only look once: Unified, real-time object detection," in CVPR, pp. 779–788, 2016.

[Rigaud 13] C. Rigaud, N. Tsopze, J. Burie, and J. Ogier: Robust frame and text extraction from comic books. In Proc. IAPR GREC, pages 129–138, 2013.

[Ronneberger 15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox,: U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv:1505.04597, 18 May 2015.

[Shi 15] Baoguang Shi, Xiang Bai and Cong Yao: An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition, 2015

[Simonyan 14] K. Simonyan, A. Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv technical report, 2014

[Stommel 12] M. Stommel, L. Merhej, and M. Muller.: Segmentation-free detection of comic panels. In LNCS: Computer Vision and Graphics, volume 7594, pages 633–640. 2012.

[出版科学研究所 2022] 全国出版協会・出版科学研究所(2022), "ニュースリリース 2022 年 2 月 25 日", https://shuppankagaku.com/wp/wp-content/uploads/2022/02/ニュースリリース 2202.pdf , (参照 2023-01-10)

[出版科学研究所 2023] 全国出版協会・出版科学研究所(2023), "ニュースリリース 2023 年 2 月 24 日", https://shuppankagaku.com/wp/wp-content/uploads/2023/02/ニュースリリース 2302.pdf , (参照 2023-01-10)

[出版科学研究所 HP 2023]全国出版協会・出版科学研究所(2023), "日本の出版統計 コミック販売額", https://shuppankagaku.com/statistics/comic/, (参照 2023-10-3)

[Tanaka 07] T. Tanaka, K. Shoji, F. Toyama, and J. Miyamichi.: Layout analysis of tree-structured scene frames in comic images. In Proc. IJCAI, pages 2885–2890, 2007.

[von Gioi 12] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall, : LSD: a Line Segment Detector, Image Processing On Line, 2 (2012), pp. 35–55. https://doi.org/10.5201/ipol.2012.gjmr-lsd

[Youngmin 19] Baek, Youngmin, et al.: Character region awareness for text detection. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.
https://arxiv.org/abs/1904.01941

研究業績リスト

論文誌

- **村上聡**, 永冨一也, 長尾智晴:“電子書籍のための全自動シングルフレーム超解像システム” 電子情報通信学会論文誌, VOL. J103-D, No.7, JULY 2020.

- **村上聡**, 長尾智晴:“マンガ書籍中の不適切な画像検出システム” 人工知能学会論文誌, 2024 年 39 巻 1 号 p. A-N76_1-11, 2024,
  DOI: https://doi.org/10.1527/tjsai.39-1_A-N76

国際会議発表

- **S.Murakami**, T.Nagao: “Manga Frame Extraction and its Application to Detecting Inappropriate Images in Manga Books”, in proceedings of International Workshop on Advanced Image Technology (IWAIT 2024), Langkawi, Malaysia, Jan. 7-8, 2024.

国内学会発表

- **村上聡**:“機械学習で eBookJapan を加速できるか？” JSAI2017, 2017 年度 人工知能学会全国大会（第 31 回）, コミック工学と AI（OS-4）, 3H2-OS-04b-4, 2017 年 5 月 25 日.

- **村上聡**:“機械学習で eBookJapan を加速できるか？Part2” 電子情報通信学会ヒューマンコミュニケーショングループ, コミック工学研究会 キックオッフイベント 2019 年 7 月 27 日.

- **村上聡**:“マンガ画像中の不適切画像の検出システム” 電子情報通信学会ヒューマンコミュニケーショングループ第二種時限研究会, 第6回コミック工学研究会 2021 年 11 月 20 日.