筑波大学大学院博士課程

システム情報工学研究科修士論文

ポーズ情報の考慮と背景データの拡張による CNN を用いた人物画像の意味的領域分割

菊池 敬済

修士（工学）

（コンピュータサイエンス専攻）

指導教員　金森 由博

2018 年 3 月

概要

人物の画像に対して肌の領域、衣服の領域、髪の領域といった人体の部位や服の領域ごとに分割を行い、それぞれの領域が何であるかを示す意味ラベルを割り当てる人物の意味的領域分割は、画像中の人物を解析する重要な手段である。一般にこのタスクを達成する手法として、理想的に領域分割されたデータを正解画像とした教師あり学習ベースの手法が用いられている。しかし、正解となる領域分割データを作るには 1 ピクセルずつ人手で意味ラベルを割り当てる必要があるため、学習するのに十分なデータを用意するのが難しく、十分な精度が得られにくい。そこで本研究では、少数の正解データしか得られない場合でもラベル割り当ての精度を向上させるアプローチとして、以下の 2 つを提案する。一つは、畳み込みニューラルネットワーク (CNN) によるラベル割り当ての学習モデルに、CNN によるポーズ推定のモデルを結合し、ポーズ推定の学習結果をラベル割り当ての学習モデルに転移させて学習させるアプローチである。人物のポーズ情報を明示的にネットワークに組み込むことによって、様々なポーズの人物画像に対応できると考えられる。もう一つは、学習に用いるデータセットを拡張 (data augmentation) するアプローチである。新たに手動でラベルを割り当てる代わりに、既にラベルが割り当て済みの人物画像に対し、別の背景画像を合成することでデータセットを拡張する。これにより訓練データ中の背景パターンが増加し、様々な背景の人物画像に対応できると考えられる。これら 2 つの手法が有効かを検証するために公開データセットを用いた実験を行い、従来手法に比べ精度が向上することを確認した。さらに、本手法による領域分割結果を利用した応用例を作成し、実際のアプリケーションの使用に耐えうることを示した。また、実験の際にデータセットに含まれるラベルのうち、該当する領域の小さいラベルでの精度が落ちることが確認された。領域の小さいラベルでも精度を向上させるために、class weight を用いた追加実験を行い対応策について議論した。

目 次

図 目 次

第1章　序論

1.1　研究の背景

近年カメラ付きスマートフォンの普及により、撮影した写真をその場で加工することが身近なものになっている。また、ソーシャル・ネットワーキング・サービス (SNS) の普及により、撮影・加工した写真をネットワーク上で共有できるようになったことで、SNS にアップロードされた画像からトレンドや消費者の行動を分析するサービスも生まれている。そういった多量の画像が存在する中で、画像中の物体やその物体の位置といった情報を自動的に処理・分析することが重要になっている。その中で、画像中の物体をピクセル単位で抽出する意味的領域分割 (Semantic Segmentation) と呼ばれるタスクは重要であり、特にコンピュータビジョンの分野で盛んに研究されている。意味的領域分割の例を図 1.1 に示す。意味的領域分割では、図 1.1(a) のような画像を入力したときに、写真に写っている物体ごとに、ピクセル単位で領域を抽出する。そして抽出した領域に写っている物体が何であるかのラベルを付与し、最終的に図 1.1(b) の結果のような結果を得る。

そのような意味的領域分割タスクの中に、人物画像の意味的領域分割タスクが存在する。図 1.2 に、人物の意味的領域分割の例を示す。人物の意味的領域分割は入力した画像に対して物体ごとに領域を分割するのでなく、顔、腕、足、ドレスなどの身体の一部や衣服の領域に意味ラベルを割り当てるタスクである。このタスクが達成されると、仮想試着システム [1]、衣服の検索 [2] や推薦 [3,4] といった様々な応用が期待できる。

最近の人物画像の意味的領域分割手法は、畳み込みニューラルネットワーク (CNN) を用いることで著しく改善されているが、様々な人間のポーズや複雑な背景画像に対処するために十分に大きな訓練データセットが必要となる。十分な訓練データが得られない場合、人間のポーズと様々な背景に対処することが困難となり、精度が低下してしまう。単純な解決策として、訓練データを増やすために手作業でピクセル単位の正解ラベルを付与することが挙げられるが、これはクラウドソーシングを用いたとしても多くの時間的・金銭的なコストが必要となる。そのため、限定された訓練データセットを使用して如何に精度を改善できるかが課題となる。

1.2　研究の目的

本研究の目的は、限定された訓練データセットを使用して精度を改善することにある。この課題に対して、以下の 2 つのアプローチを試みる。

(a)入力画像　　(b) 意味的領域分割の結果

図 1.1: 意味的領域分割の例. (a) のような画像を入力した際に (b) のように物体ごとに領域を分割し, 対応する意味ラベルを割り当てる.

画像の出典: “*Instance-aware Semantic Segmentation via Multi-task Network Cascades*” [5]

(a) 入力画像　　(b) 意味的領域分割の結果

図 1.2: 人物の意味的領域分割の例. 人物の意味的領域分割では人物画像を入力し, (b) のように身体の部位や服ごとに領域を分割し、対応する意味ラベルを割り当てる.

一つは、様々なポーズに対処するために、人物のポーズ推定の学習結果を転移するアプローチである。ポーズ推定の場合、学習に必要なデータは関節ごとの位置情報であり、これは画素ごとに人物の意味ラベルを割り当てるよりも正解データを作ることが容易である。我々のキーアイデアは、人間のポーズ推定情報を人物の意味的領域分割の CNN に転移するために、ポーズ推定と意味的領域分割の CNN を統合することである。このアイデアは様々な手法で実現できるが、本研究では最新の CNN モデルの一つであり、比較的単純なポーズ推定モデル [6] と人物の意味的領域分割モデル [7] を利用する。従来の CNN ベースの人物の意味的領域分割はポーズ情報を明示的に考慮していないが、提案手法では明示的にこれらを統合することで精度を改善できる。

もう一つは、人物の意味的領域分割のために全自動でデータを拡張し、学習データを増加させるアプローチである。様々な背景画像を扱うために、既存のラベル付きデータの背景を、大規模な公開データセット [8] から得られた背景画像で置き換えて訓練データ数を増加させることで、精度の改善を図る。

本研究では、ポーズ推定およびデータ拡張を行っていない既存の CNN ベースの手法と定量的および定性的に比較することで、本手法の有効性を実証する。さらに、提案手法のアプリケーションとして、衣服の色変更、テクスチャの転写、およびファッション分析のための可視化の 3 つを実装することで、本手法が実応用においても有効であることを実証した。

1.3 本論文の構成

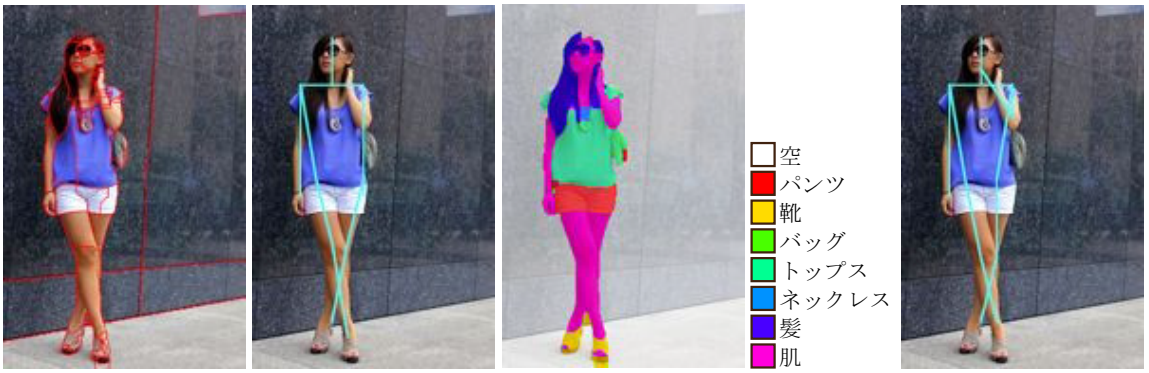
本論文は全 7 章で構成される。第 1 章では、本研究の背景と目的を述べた。第 2 章では関連研究について、本研究の位置づけを述べる。また、第 3 章では本研究においてベースとなる二つの既存手法について詳しく説明する。第 4 章では本研究の提案手法である人物のポーズ推定の転移とデータ拡張について順に説明する。第 5 章では提案手法の有効性を示すための実験とその結果、またそれによって生じた問題点について議論する。第 6 章では本手法の領域分割結果を用いたアプリケーションを例示する。第 7 章では本研究を総括し、結論を述べる。

第2章　関連研究

2.1　人物画像の意味的領域分割に関する研究

2.1.1　条件付き確率場 (CRF) による手法

人物の意味的領域分割の初期の研究は、条件付き確率場 (CRF) を用いた手法が主であった。Yamaguchi らは人物のポーズと領域分割を相互に学習することで人物の意味的領域分割を行う手法 [9] を提案した。図 2.1 に、Yamaguchi らの手法の概略を示す。Yamaguchi らの手法ではまず、図 2.1(a) のようなスーパーピクセルによる分割と、図 2.1(b) のようなポーズ推定を行う。ここでスーパーピクセルとは、類似するピクセルを集めて 1 つのピクセルとして扱われるものである。図 2.1(a) では赤線で囲まれた部分が 1 つのスーパーピクセルとなっている。スーパーピクセルの分割結果とポーズ推定結果と、画像の色情報やフィルタ処理によって得られた特徴等を基に、CRF によって図 2.1 (c) のような領域分割の結果を得ることができる。加えて図 2.1 (d) のように、領域分割の結果からさらに正確なポーズの再推定結果を計算することができる。図 2.1 (d) では、左腕の位置が修正されていることがわかる。



(a) スーパーピクセル　(b) ポーズ推定結果　(c) 領域分割結果　(d) ポーズ再推定結果

図 2.1: CRF による Yamaguchi らの手法 [9].

画像の出典: "*Parsing Clothing in Fashion Photographs*" [9]

さらに Yamaguchi らはそれを発展させ、k-近傍法により類似画像を検索し、そのタグ情報も利用して意味的領域分割を行う手法 [10] も提案した。図 2.2 に Yamaguchi らの手法のパイプラインを示す。画像を入力すると、まずシステムはデータベースから類似のスタイルを持

つ画像を検索し、検索した画像の持つタグから、入力した画像のもつタグを予測する。その後、3 つの領域分割結果を出力する。1 つ目は入力画像の特徴とタグを使用して文献 [9] により学習した領域分割結果、2 つ目は検索により得られた類似の画像のみで学習したモデルを用いて分割した結果、3 つ目は 1 つ目の尤度マップを、検索した類似画像に転写して得られる結果である。この 3 つの結果を結合して平滑化することで、最終的な結果を出力している。

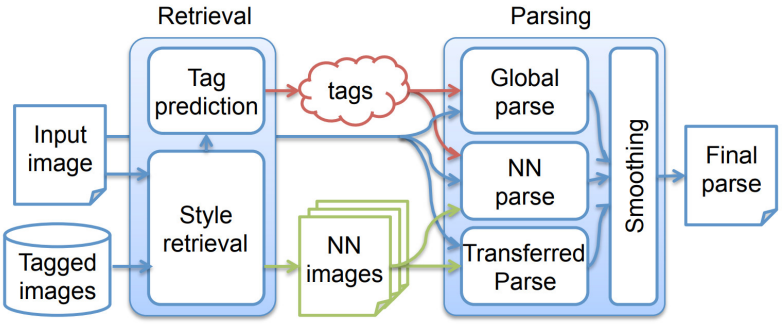


図 2.2: 画像検索を用いた Yamaguchi らの手法 [10] のパイプライン.

画像の出典: “*Retrieving Similar Styles to Parse Clothing*” [10]

Simo-Serra らはスーパーピクセルの位置と形状を考慮することで Yamaguchi らの手法 [9] を改良した [11]。Simo-Serra らの手法では、CRF のポテンシャル関数に、Yamaguchi [9] らの手法での特徴に加え、前景・背景のマスク、人体や衣服の位置、形状や服装の局所的な外観から計算された特徴量を使用している。また、スーパーピクセル同士の類似度や関節とスーパーピクセルとの接続関係といった、スーパーピクセルによる特徴量も考慮している。

CRF による手法の欠点として、精度を上げるには人手による特徴設計が必要であるという点がある。この欠点を解消する手法として、畳み込みニューラルネットワーク (CNN) ベースの手法が提案されている。

2.1.2 畳み込みニューラルネットワーク (CNN) による手法

近年では、CNN に基づいたアプローチが提案され、精度が改善されている。Liang らは人物の意味的領域分割に初めて CNN を用い、ATR フレームワーク [12] を提案した。図 2.3 に、ATR フレームワークの概略図を示す。ATR フレームワークでは、まず入力された画像に対し、人体のバウンディングボックスを見つけトリミングを行う。トリミングされた画像は、Active Template Network と Active Shape Network に入力される。Active Template Network では各ラベルの信頼度マップを、Active Shape Network では背景であるかどうかの信頼度マップを出力する。最後に、入力画像から生成されたスーパーピクセルごとに、この 2 つの信頼度マップを平滑化し、最終的な領域分割結果を得る。

Liang らは ATR ネットワークの後に、各層の出力と画像の大域特徴を組み合わせた Con-

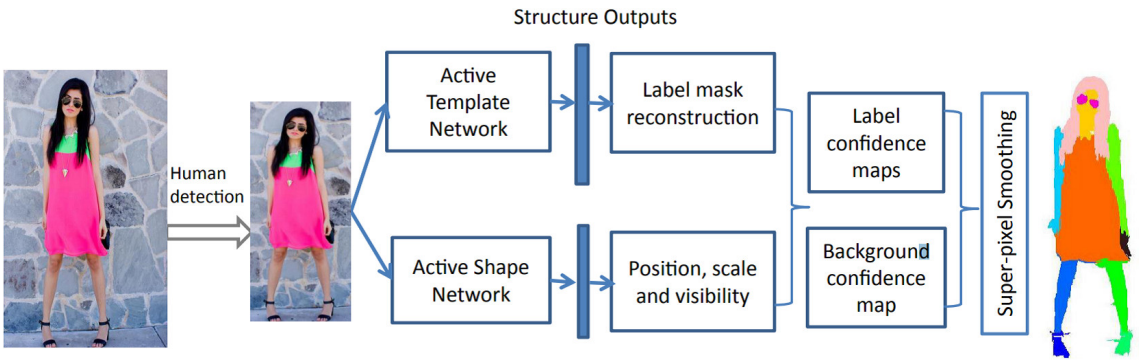


図 2.3: Liang らの ATR フレームワーク [12].

画像の出典: "*Deep Human Parsing with Active Template Regression*" [12]

textualized CNN (Co-CNN) [7] を提案した。この手法は本手法のベース手法となるため、詳細は 3 章で説明する。

Liu らは、Matching CNN と呼ばれる、$k-$ 近傍法により検索された類似画像を入力として学習させる手法 [13] を提案した。図 2.4 に、Matching CNN のフレームワークの概略図を示す。このフレームワークでは、画像を入力すると、手動で領域分割済みの類似画像がデータベースから検索される。検索された画像は、意味ラベルごとに分割され、入力画像とともに Matching CNN アーキテクチャへと入力される。Matching CNN アーキテクチャでは、類似画像ごとに、入力画像中の意味領域の信頼度マップが出力される。その出力結果をそれぞれの意味ラベルごとに組み合わせ、スーパーピクセルによる平滑化を経て、最終的な分割結果が得られる。

CNN の手法は特徴量を自動で推定するため、 CRF の手法に比べ精度が高いことが挙げられる。しかし、CNN ベースの手法では多くの正解データセットが必要であるという欠点がある。本研究では、CNN ベースの手法に ポーズ情報を明示的に組み込み、さらに背景データ拡張を行い学習データを増やすことで、少数の意味的領域分割データでも効率的な学習が行えるようになる利点がある。

2.2 意味的領域分割に関する研究

人物の意味的領域分割は一般物体画像の意味的領域分割の研究の一種であり、一般物体画像を対象とした CNN ベースの手法も多数提案されている [14–19]。この中には、異なるタスクを同時に学習することで精度の向上を図っている CNN ベースの手法が存在する。

Dai らは、1 つのネットワークで複数のタスク (オブジェクト検出、マスク抽出、および意味ラベリング) を処理する Multi-task Network Cascades (MNCs) と呼ばれるネットワーク [5] を提案した。図 2.5 に MNCs の概略図を示す。MNCs では、入力された画像に対し、畳み込み層を通して共有特徴マップが抽出される。共有特徴マップは、まず box instances のステージ

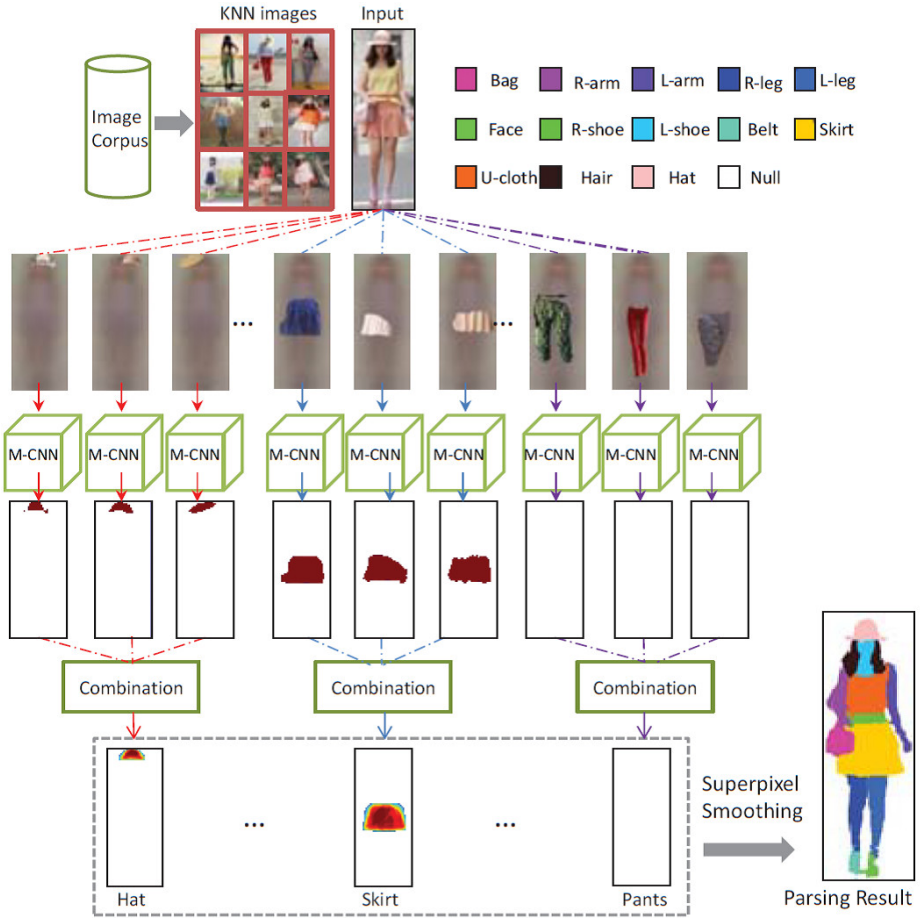


図 2.4: Liu らの Matching CNN のフレームワーク [13].

画像の出典: "*Matching-CNN Meets KNN: Quasi-Parametric Human Parsing*" [13]

によって、画像中の物体を矩形領域として抽出する。次に、共有特徴マップと box instances の出力が、mask instances ステージへと入力される。mask instances ステージでは、Region of Interest (RoI) プーリングと全結合層によって、画像中の物体のマスクが生成される。最後に、共有特徴マップと mask instances の出力を入力として、各マスクごとに意味ラベル付けを行うことで、最終的な意味的領域分割の出力としている。この手法は図 2.5 右上の図のような カスケード型のネットワークが特徴であり、1 つのタスクを複数のステージとして明示的に分離することで従来手法よりも高い精度での認識が可能であるとしている。

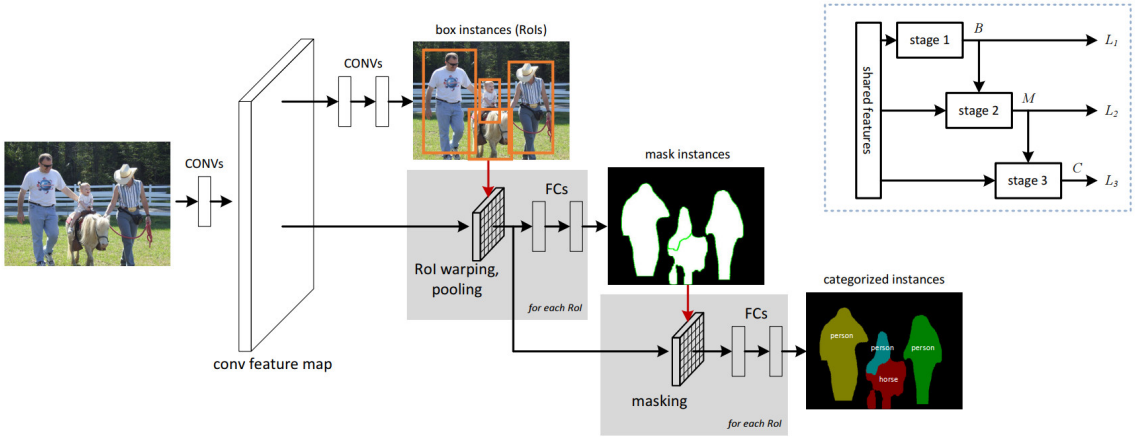


図 2.5: Dai らの Multi-task network cascades [5] の概略図.

画像の出典: "*Instance-aware Semantic Segmentation via Multi-task Network Cascades*" [5]

Hong らは、意味的領域分割と画像のクラス分類を同じネットワーク上で学習させる手法 [20] を提案した。図 2.6 に Hong らの手法のアーキテクチャを示す。画像が入力されると、まずエンコーダ f_{enc} によって特徴が抽出される。抽出された特徴は、attention モデル f_{att} により顕著性が推定される。その後、attention モデルの出力がデコーダ f_{dec} へと入力され、最終的な領域分割結果が得られる。このアーキテクチャの特徴として、学習時にクラスの学習をする分類器 f_{cls} と領域分割を行うデコーダ f_{dec} を同時に学習することが挙げられる。この 2 つのドメインを組み合わせて学習することで、精度を従来手法よりも改善させている。

Papandreou らは、矩形の領域推定、画像レベルのクラス分類といった多くの単純なタスクの訓練データを利用して、少数の訓練データしかない意味的領域分割のタスクを Expectation-Maximization (EM) 法に基づいて学習する手法を提案した [21]。図 2.7 に Papandreou らの手法のモデルの一例を示す。この手法では、図上段のピクセル単位でのラベル付けによる誤差関数の他に、図下段で示している画像単位でのクラス分類といった意味的領域分割よりも単純なタスクによる誤差関数と組み合わせることで、精度の向上を図っている。画像単位でのクラス分類の他に、バウンディングボックスの誤差関数との組み合わせによる検証も行っている。

このように異なるタスクを転移することで意味的領域分割の精度を向上させる研究が多く

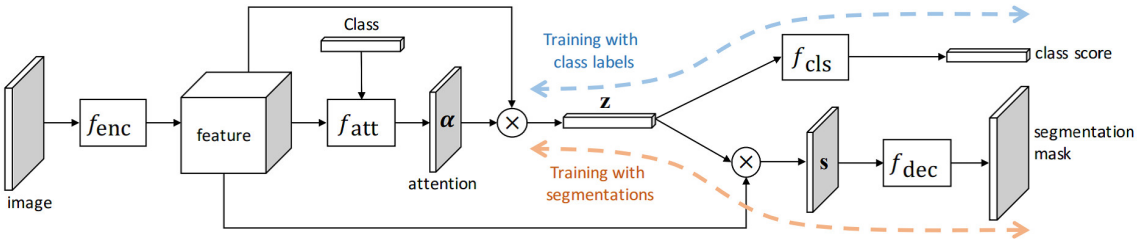


図 2.6: Hong らのアーキテクチャ [20] の概略図.

画像の出典: "*Learning Transferrable Knowledge for Semantic Segmentation with Deep Convolutional Neural Network*" [20]

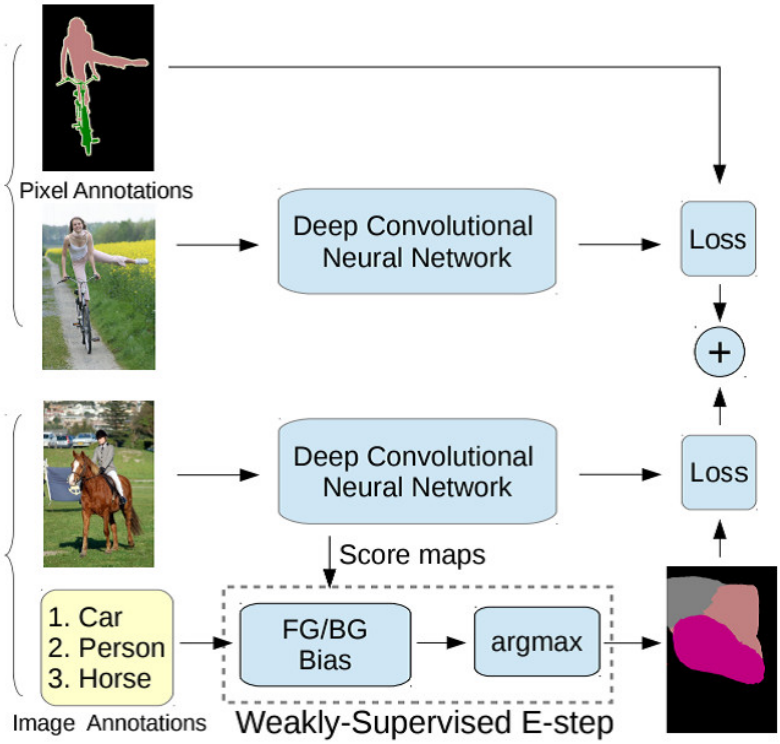


図 2.7: Papandreou らのモデル [20] の一例.

画像の出典: "*Weakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation*" [21]

存在する。本研究においても、人物の意味的領域分割の CNN モデルに明示的にポーズ情報を組み込むことで精度の向上を図っており、CNN ベースの人物の意味的領域分割において、この試みは我々の知る限り初めてである。

2.3 ポーズ推定に関する研究

2.3.1 CNN によるポーズ学習時の関節情報の表現

ポーズ推定用のデータセットでは、正解データは各関節の座標情報で与えられる。CNN でポーズを学習する際は正解に関節の座標を出力するのではなく、関節ごとに尤度を表すヒートマップを出力として学習を行う。図 2.8 に、ポーズ推定で使用されるヒートマップの例を示す。図 2.8(a) で示している赤点は、正解となる人物の関節位置を可視化した結果である。図 2.8(b) で示されるように、ヒートマップは関節ごとにガウス関数のような分布関数によって生成され、正解となる関節位置 (図 2.8(a) の各赤点) で最も尤度が高くなり、そこから遠くなるにつれて尤度が低くなるように設定される。図 2.8(b) では、尤度が高いほど赤に近い色で、尤度が低いほど青い色で可視化されている。

(a) 人物画像　　(b) 生成されるヒートマップ

図 2.8: ポーズ推定でのヒートマップの例. (a) の赤点は人物の関節の正解位置を示している.

2.3.2 CNN によるポーズ推定の研究

Yang らは、身体の幾何学的な知識を事前に CNN のフレームワークに組み込んだ手法 [22] を提案した。図 2.9 に Yang らの手法のアーキテクチャを示す。このアーキテクチャでは、人物画像を入力すると、まず画像中の特徴を抽出する CNN (図 2.9(b)) へと入力される。その後、推定された特徴は Message Passing Layers (図 2.9c) へと渡され、図 2.9(a) 中の白線や矢印で示されるような体の接続関係や変形についての制約を基にした推定を行い、最終的な出力を行う。このような人物の構造を明示的に考慮した CNN フレームワークを考慮することで、精度の向上を図っている。

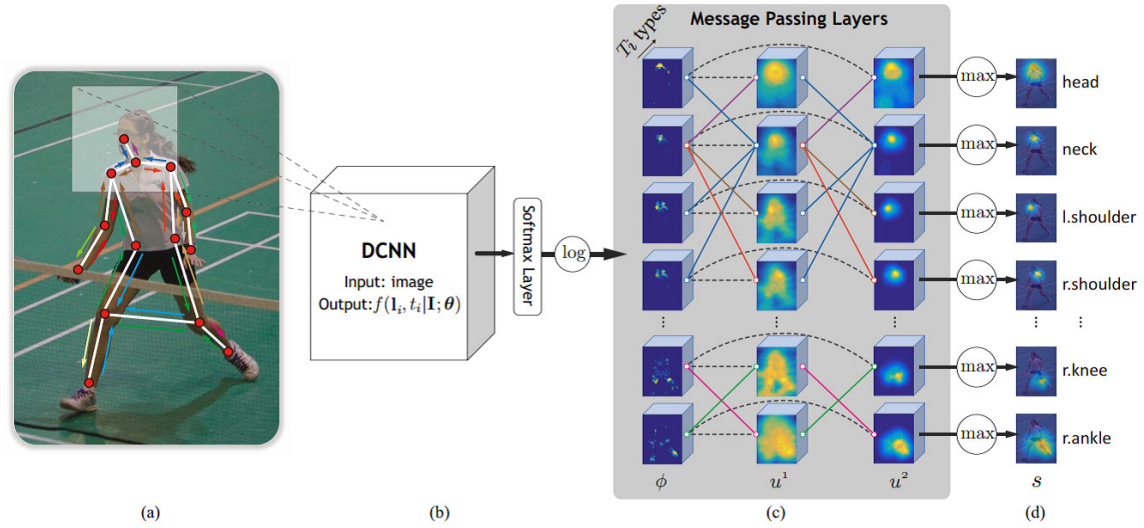


図 2.9: Yang らのフレームワーク [22] の概略図.

画像の出典: "*End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation*" [22]

Chu らは、CNN に関節間の相関関係を組み込んで学習する手法 [23] を提案した。図 2.10 に、Chu らの手法のパイプラインを示す。入力された画像は畳み込み層を通り、各関節の特徴マップ (図 2.10(1)) が出力される。その後、関節の構造特徴を学習する層 (図 2.10(2)) へと入力される。この層で関節は順方向の木 (図 2.10(2,a)) と逆方向の木の 2 つの有効グラフとして表現される。最後に、この層から各関節のヒートマップを推定し、最終的な結果を出力している。双方向木を使って end-to-end の学習を行うことで、ポーズ推定の精度を高めている。

他に、畳み込み層とプーリング層からなる単純なモデルによってポーズ推定を行う手法 [6] がある。この手法は本手法のベース手法となるため、詳細は 3 章で説明する。

本研究ではこのような CNN のポーズ推定手法を明示的に組み込むことで、人物の意味的領域分割の精度の向上を図る。

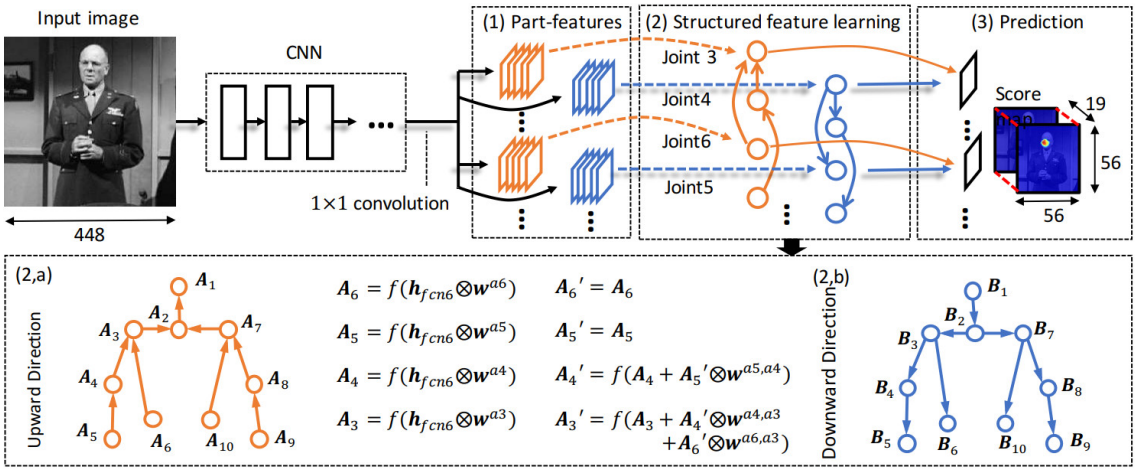


図 2.10: Chu らの手法のパイプライン [23] の概略図.

画像の出典: ‘’*Structured Feature Learning for Pose Estimation*” [23]

第3章　ベース手法

1 章で、本研究では様々なポーズに対処するために、ポーズ推定と意味的領域分割の CNN を統合すると説明した。本研究のアイデアは様々な手法で実現できるが、実証実験においては最新の CNN モデルの一つであり、比較的単純なポーズ推定モデルである Wei らの手法 [6] と Liang らの人物の意味的領域分割モデル [7] を利用している。本章では、 この 2 つの CNN モデルについて説明する。

3.1　Convolutional Pose Machines

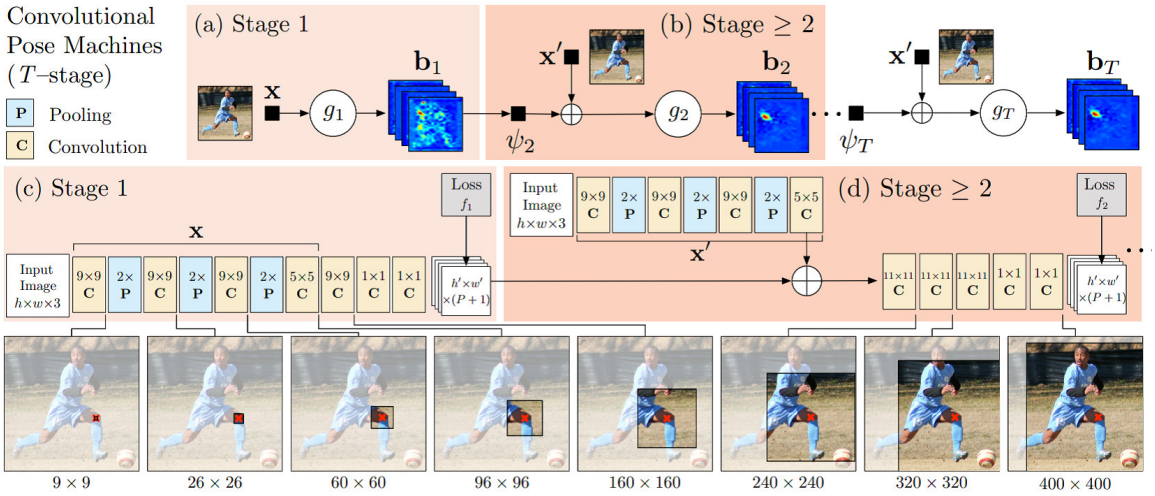


図 3.1: Convolutional pose machines [6] のモデル図.

画像の出典: “*Convolutional pose machines*” [6]

Convolutional pose machines [6] のモデルを図 3.1 に示す。Convolutional pose machines はポーズ推定を行うネットワークであり、人物画像を入力とし、各関節のヒートマップを出力するネットワークモデルである。Convolutional pose machines では、畳み込み層とプーリング層からなるネットワークをステージと定義している。図 3.1 中では、(c) と (d) に該当する。このステージを複数回繰り返すことによって、ヒートマップを改善している。ネットワークでは、ステージの繰り返しによる勾配消失問題を解消するために、各ステージの損失関数を最

小化することで学習する。この Convolutional pose machines のネットワーク構造は、単純な畳み込み層とプーリング層で構成され、end-to-end での学習が可能となるため、本研究のフレームワークに簡単に統合することができる。

3.1.1 学習

一般的なポーズ推定データセットでの正解となる各関節情報は、画像の二次元座標として記録されている。Convolutional pose machines では、学習のために 正解となる関節 p の二次元座標がピークとなるようなガウス関数によって生成されるヒートマップ $b_*^p(z)$ を正解データとして学習を行う。ここで z は画像中のピクセル位置である。学習の際は、各ステージにおいて、以下のような誤差関数が定義される。

$$f_t = \sum_{p=1}^{P+1} \sum_{z \in \mathcal{Z}} \| b_t^p(z) - b_*^p(z) \|_2^2 \tag{3.1}$$

ここで、P はジョイントの総数、$\mathcal{Z}$ は画像中のピクセルの集合、$b_t^p(z)$ は各ステージ t で出力されるヒートマップである。これを、各ステージにおいて以下のように総和をとることでモデル全体の損失関数としている。

$$F = \sum_{t=1}^{T} f_t \tag{3.2}$$

ここで、T はすべてのステージの合計である。また、ステージ 2 以降は図 3.1(d) の $\mathbf{x}'$ の部分のパラメータを共有している。

3.2 Contextualized CNN (Co-CNN)

図 3.2 に Liang らの Co-CNN [7] モデル図を示す。Co-CNN [7] は人物の意味的領域分割のネットワークである。このネットワークでは、意味的領域分割の性能を向上させるために、Cross-layer context と Global image-level context によって、局所的な特徴と大域的な特徴を学習している。Cross-layer context は、出力が同じサイズとなるダウンサンプリングの層とアップサンプリングの層の各ペアの間で、出力をそれぞれ加算することで実現している。Global image-level context は、全結合層を用いて画像全体でのラベルが予測され、予測されたラベルはアップサンプリングしている層と連結されている。

また Co-CNN では、誤差計算を行う前にスーパーピクセル層を挿入することで、学習効率を向上させている。スーパーピクセル層は 3 層あり、以下で説明する通り、スーパーピクセル内での平滑化層、隣接スーパーピクセルによる投票層、ピクセル単位での予測に変換する層がある。

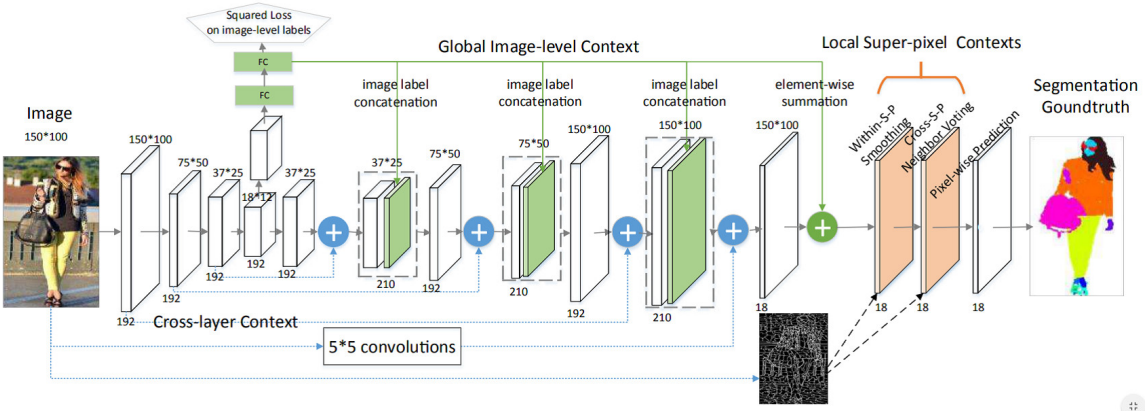


図 3.2: Liang らの Co-CNN [7] のモデル図.

画像の出典: "*Human Parsing with Contextualized Convolutional Neural Network*" [7]

3.2.1 スーパーピクセル層

スーパーピクセル内での平滑化層　平滑化層では、スーパーピクセル内での信頼度マップを平滑化する。ここで信頼度マップは、直前の層から得られた各クラスのピクセル単位での尤度を表すマップである。スーパーピクセル $\mathcal{S}$ 上のクラス c についての信頼度 $\tilde{x}_{\mathcal{S},c}$ は、以下のように表すことができる。

$$\tilde{x}_{\mathcal{S},c} = \frac{1}{|\mathcal{S}|} \sum_{s_c \in \mathcal{S}} s_c \tag{3.3}$$

ここで、$\mathcal{S}_c$ はスーパーピクセル $\mathcal{S}$ 中のあるピクセルにおけるクラス c の信頼度、$|\mathcal{S}|$ はスーパーピクセル $\mathcal{S}$ 中にあるピクセルの総数である。

隣接スーパーピクセルによる投票層　投票層では、隣接スーパーピクセルができるだけ類似したラベルを持つように、隣接するスーパーピクセル間で投票を行う。あるスーパーピクセル $\mathcal{S}$ に対するクラスの投票応答 $\bar{x}_{\mathcal{S}}$ は、以下のように表すことができる。

$$\bar{x}_{\mathcal{S}} = (1-\alpha)\tilde{x}_{\mathcal{S}} + \alpha \sum_{\mathcal{S}' \in \mathcal{D}_{\mathcal{S}}} \frac{\exp(-\|b_{\mathcal{S}} - b_{\mathcal{S}'}\|^2)}{\sum_{\hat{\mathcal{S}} \in \mathcal{D}_{\mathcal{S}}} \exp(-\|b_{\mathcal{S}} - b_{\hat{\mathcal{S}}}\|^2)} \tilde{x}_{\mathcal{S}'} \tag{3.4}$$

ここで、α は重みで、$\alpha = 0.3$ としている。$\mathcal{D}_{\mathcal{S}}$ はスーパーピクセル $\mathcal{S}$ に隣接するスーパーピクセルの集合、$b_{\mathcal{S}}$ はスーパーピクセル $\mathcal{S}$ の特徴 (RGB、Lab、HOG) である。

ピクセル単位での予測層　最後に、スーパーピクセル単位での結果をピクセル単位での結果に戻し、softmax 関数を適用することによって、各クラスのピクセルごとの信頼度マップが得られる。 スーパーピクセル内での平滑化層と隣接スーパーピクセルによる投票層は一種のプー

リング層とみなすことができるため、一般的に使用される誤差逆伝播のアプローチによってネットワークパラメータを最適化することができる。

第4章　提案手法

本章では、人物のポーズ情報を人物の意味的領域分割に転移するネットワークと、背景データ拡張のアプローチについて説明する。

4.1　ポーズ情報の転移

人物の様々なポーズに対処するために、人物の意味的領域分割の前にポーズの推定を行い、ポーズ推定結果を用いて入力画像の各ピクセルにラベルを割り当てる。図 4.1 に、本手法でのネットワークモデルを示す。まず、共有ユニットに画像が入力され、低レベルおよび中間レベルの特徴量が抽出される。この共有ユニットは、カーネルサイズ 5、ストライド 1、パディング 2、出力チャネル 128 の 4 つの畳み込み層から構成される。次に、共有ユニットで抽出された特徴量は、ポーズ推定ユニットに入力される。ポーズ推定ユニットの詳細図を図 4.2 に示す。図中の各層の意味については、図 4.4 を参照されたい。ポーズ推定ユニットの構造は、Wei らの手法 [6] のネットワークモデルに基づいて構成される。Wei らのネットワークは 3 章で説明したように、畳み込み層とプーリング層からなるネットワークを 1 ステージと定義し、このステージを複数回繰り返すことでポーズの推定を行っている。その後、ポーズ推定ユニットの出力は共有ユニットと連結される。ラベル割り当てユニットの詳細を図 4.3 に示す。連結された特徴量はラベル割り当てユニットに入力され、最終的なラベル出力となる。ラベル割り当てユニットには、Co-CNN モデル [7] を使用している。Co-CNN モデルは 3 章で説明したように、畳み込み層の後、全結合層を経てラベルの大域的な分布を出力する。一方、逆畳み込み層を介して人物の領域分割結果が計算され、最終的にスーパーピクセルを用いた平滑化により、結果が出力される。

4.1.1　学習

提案したモデルを、ポーズ推定のデータセットと、人物の意味的領域分割用のデータセットを用いて学習させる。本研究では、これら 2 つのデータセットを、片方ずつ交互に学習させる。本来であれば、ポーズ推定と意味的領域分割を同時に学習することが理想的であるが、両者の正解データを持つデータセットが存在しないため、このようなアプローチを採用した。ポーズ推定データセットで学習を行う場合、次の誤差関数を最小化することで、共有ユニッ

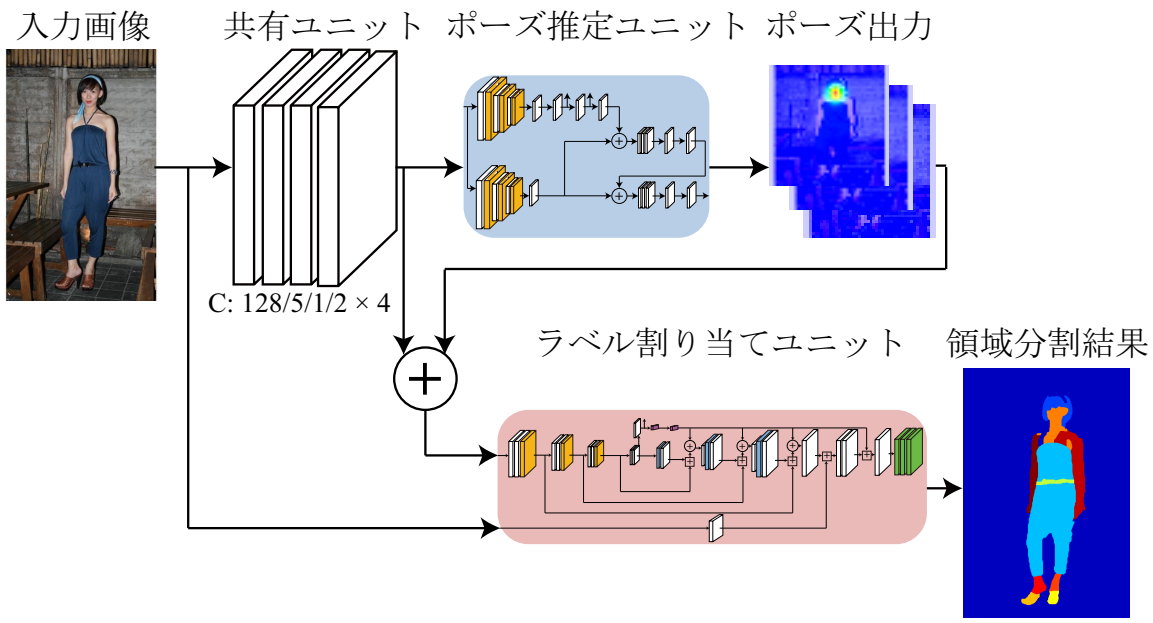


図 4.1: 本手法のネットワークモデルの簡略図. 本手法のモデルでは, 画像が与えられると, まず共有ユニットで特徴量が抽出される. 次に, 人物のポーズがポーズ推定ユニットで関節ごとにヒートマップとして推定される. 出力された推定結果は, 共有ユニットと結合されて人物の意味的領域分割ユニットへと入力される. 最後にラベル割り当てユニットによって最終的な人物の意味ラベルが出力される.

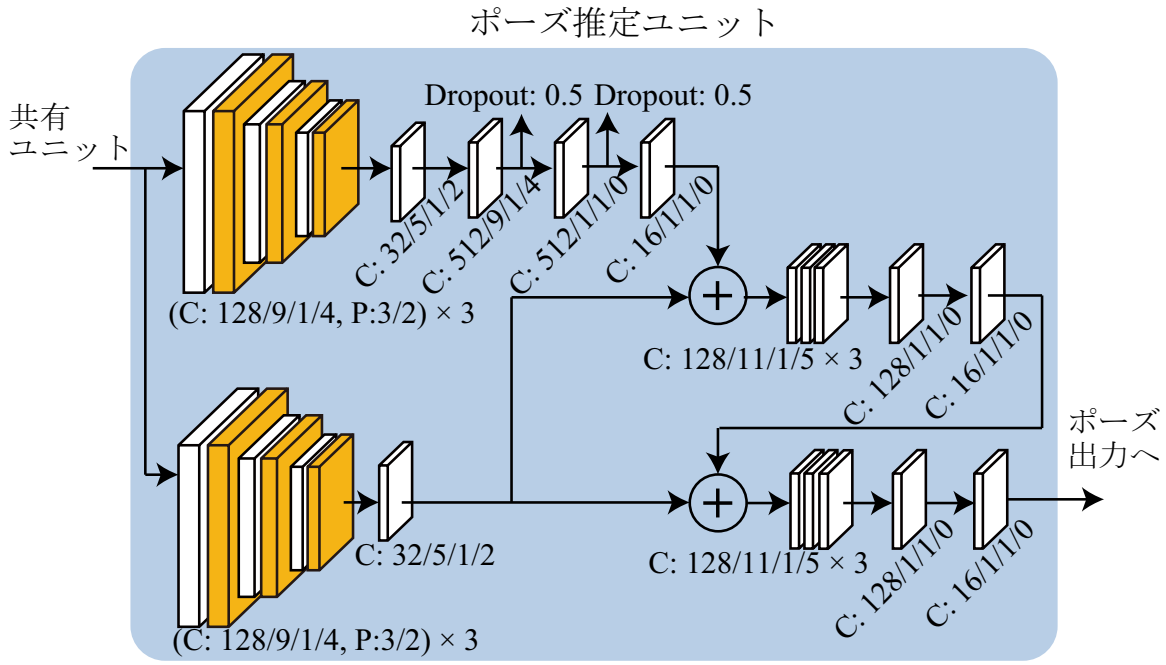


図 4.2: ポーズ推定ユニットの詳細図. 本手法では Convolutional pose machines [6] をベースにしている.

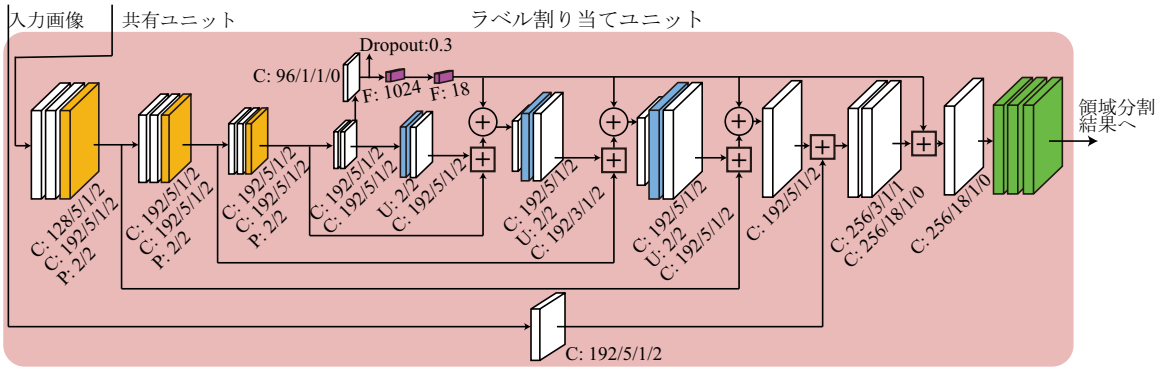


図 4.3: ラベル割り当てユニットの詳細図. 本手法では Co-CNN [7] をベースにしている.

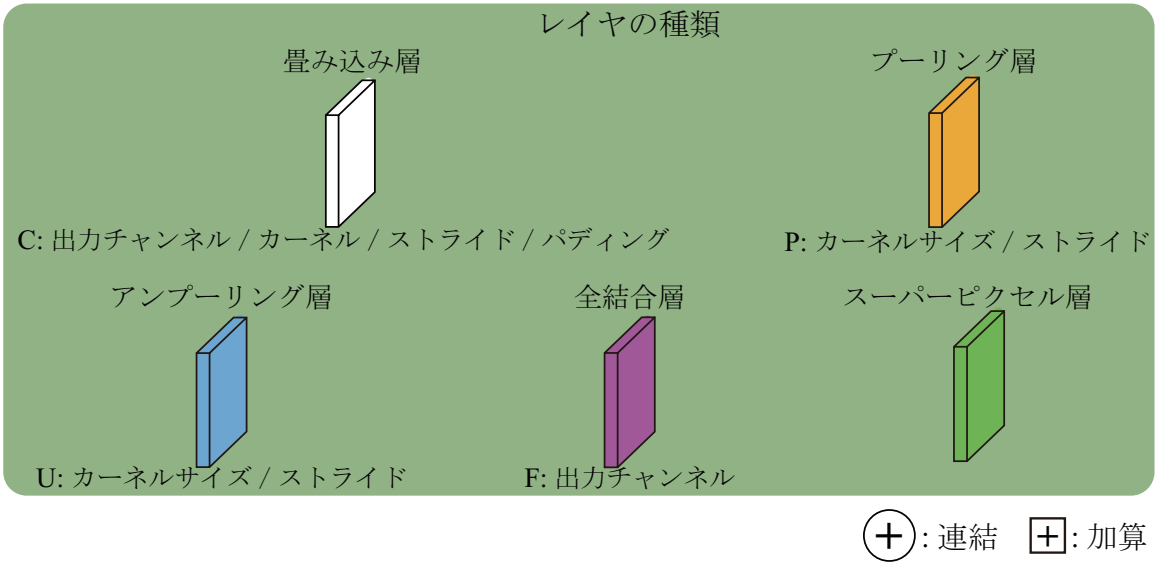


図 4.4: 図中の各レイヤの詳細.

トとポーズ推定ユニットのパラメータ θ_s と θ_p を最適化する。

$$E_p = \sum_{\{\mathbf{b}_i, \mathbf{b}_l\} \in \mathcal{B}} \sum_{t=1}^{T} \sum_{j=1}^{J} \|\mathbf{b}_l^j - \mathbf{B}_t^j(\mathbf{b}_i; \theta_s, \theta_p)\|_2^2 \tag{4.1}$$

ここで、$\mathcal{B}$ はポーズ推定データセット、$\mathbf{b}_i$ は入力画像、$\mathbf{b}_l$ は正解となる関節のヒートマップである。T は反復するステージ数、J は推定される関節の数、$\mathbf{B}$ はポーズ推定のモデルによって推定された関節のヒートマップである。正解となる関節のヒートマップは、位置 $\mathbf{x}$ に関するガウス関数 $\exp(-\|\mathbf{x} - \mu_j\|^2/\sigma^2)$ によって生成している。ここで、μ_j は j 番目の関節の位置で、$\sigma = 2$ とした。

人物の意味的領域分割用のデータセットで学習する場合、式 (4.1) の代わりに、以下の誤差関数を最小化することでネットワーク全体のパラメータ θ を最適化する。

$$\begin{aligned} E_l &= E_l^{orig} + E_l^{accel}, \\ E_l^{orig} &= -\sum_{\{\mathbf{d}_i, \mathbf{d}_l\} \in \mathcal{D}} \sum_{j}^{M} \sum_{k}^{L} \mathbf{d}_{l_{jk}} \ln(\mathbf{F}_{jk}(\mathbf{d}_i; \theta)) \; + \sum_{\{\mathbf{d}_i, \mathbf{d}_{l'}\} \in \mathcal{D}} ||\mathbf{d}_{l'} - \mathbf{H}(\mathbf{d}_i; \theta)||^2, \\ E_l^{accel} &= -\sum_{\{\mathbf{d}_i, \mathbf{d}_l\} \in \mathcal{D}} \sum_{j}^{N} \sum_{k}^{L} \mathbf{d}_{l_{jk}} \ln(\mathbf{G}_{jk}(\mathbf{d}_i; \theta)) \end{aligned} \tag{4.2}$$

E_l^{orig} は文献 [7] において使用されている誤差関数に基づいている。本研究ではさらに E_l^{accel} を追加することで、収束速度が向上することを確認している。$\mathcal{D}$ は入力画像 $\mathbf{d}_i \in \mathbf{R}^{h \times w \times c}$ と正解ラベル $\mathbf{d}_l \in R^{h \times w \times L}$ を含む人物の意味的領域分割用のデータセット、$\mathbf{d}_{l'} \in \mathbf{R}^L$ は全画

像のクラスの分布である。w と h は画像の幅と高さ、c は画像のチャネル数、M はスーパーピクセルの数、N は画像中のピクセルの数、L はクラス数 (本研究では、Liang ら [7] と同様に $L = 18$ とした)。$\mathbf{F}$ はラベル割り当てユニットの出力、$\mathbf{G}$ はラベル割り当てユニットでのスーパーピクセル処理の前の出力、$\mathbf{H}$ は全結合層の後の出力である。

なお学習時は、ポーズ推定用の全データに対して E_p に基づく最適化を行った後、意味的領域分割用の全データに対して E_l に基づく最適化を行う処理を 1 エポックとしている。最適化には Momentum SGD を用い、Liang ら [7] の手法を参考に学習率 0.001、慣性項 0.9、減衰項 0.0005 とした。

4.2 背景データの拡張

人物の意味的領域分割をより多様な背景に対して頑健にするために、訓練データセットの背景パターンを拡張し、訓練データセットを増加させる。具体的には、ラベル付けされた画像から前景の人物領域を切り出し、背景画像データセットから得られた新しい背景に貼りかえる。

データを拡張する手順を図 4.5 に示す。入力は、切り抜かれた人物画像とそれに対応するラベル画像のペアと、新しい背景画像である。ほとんどの背景画像は横長であるため、切り出した人物と背景画像の比が元の意味的領域分割データセットの比率と一致するように背景画像をトリミングする (図 4.5(c))。図 4.6 に、トリミング手順の詳細を示す。まず、人物の意味的領域分割データセットの各画像における人物領域の相対的な幅および位置の統計値 (平均値、標準偏差) を計算する。そして、統計値を基にした正規乱数によって新しい背景の幅と人物の位置を決定する。最後に、トリミングした新しい背景と切り抜かれた人物を合成し、同じ位置にラベル画像を配置して最終的な結果を得る。

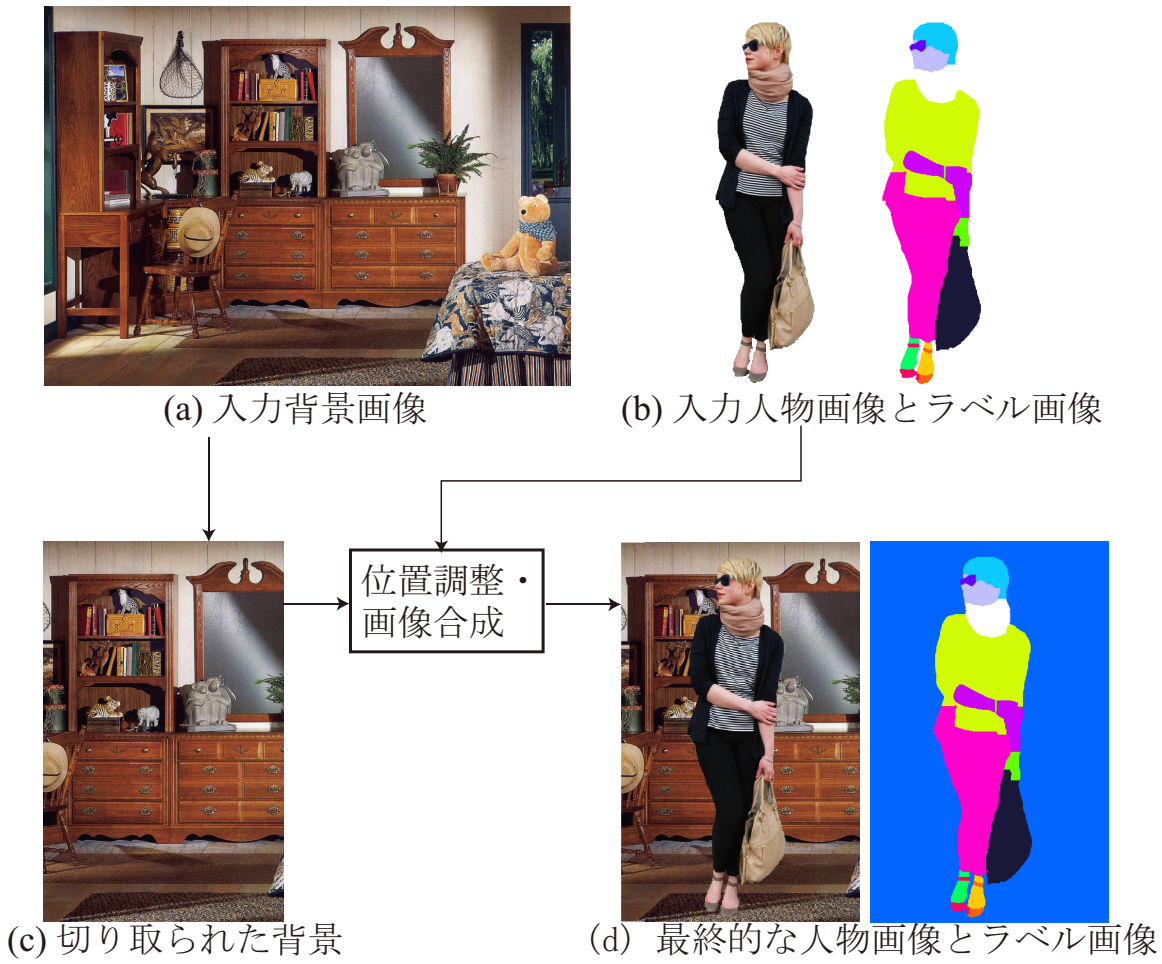


図 4.5: 背景データの拡張処理の手順. (a) 新たな背景画像と (b) 既に分割済みの人物画像とラベル画像を入力する. 背景画像は横長のものが多いため, (c) 背景をトリミングし, 位置調整・画像合成を行い最終的な人物画像とラベル画像 (d) を得る.

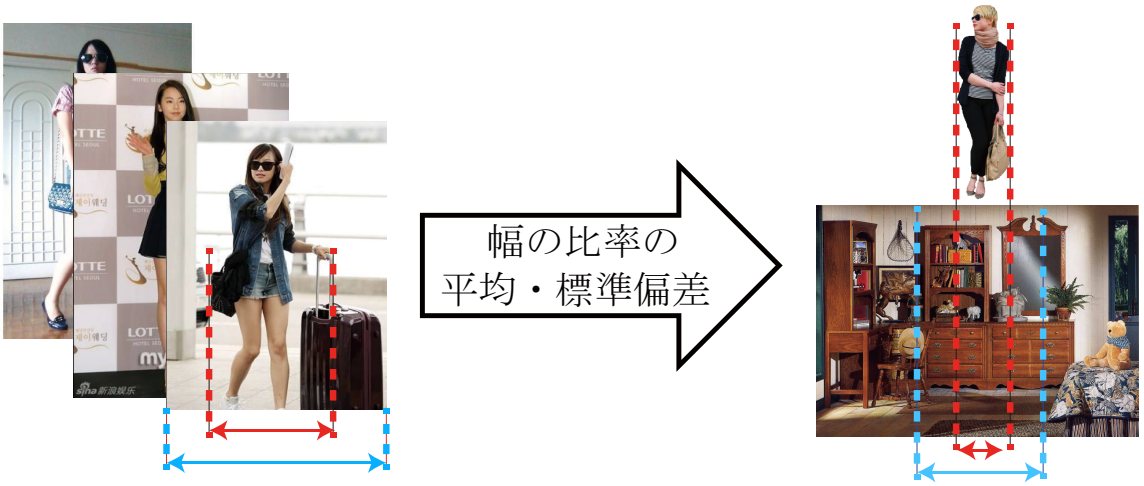


図 4.6: 背景画像のトリミング方法の詳細. (a) データセット中の画像から人物と背景の幅の比率の平均と標準偏差を取得し, (b) 正規乱数により合成する背景の幅と人物の位置を決定する.

第5章　実験

本章では、提案手法とそのベース手法である Co-CNN [7] とを比較した実験結果を示す。

5.1　実験設定

提案手法が基にしたポーズ推定手法 [6] では、反復するステージ数が 6 段階に設定されているが、本実験では計算時間と GPU メモリの使用量を削減するために 3 段階に減らした。人物の意味的領域分割ネットワーク [7] ではいくつかの特徴量を使ってスーパーピクセル間の類似度を計算しているが、一部特徴量の計算方法が明確に示されていないため、本実験では RGB 特徴のみを使用した。本手法と Co-CNN の実装には Python 言語と Chainer ライブラリを使用し、NVIDIA GeForce GTX 1080 を搭載した PC を使用してモデルを学習した。学習にかかった時間は、ポーズ推定を含むモデルで訓練データに画像 12,000 枚使用し、約 1ヶ月程度であった。本手法のモデルを用いた、学習後の推定にかかる順伝播の計算時間は、1,000 枚のテストデータについて画像 1 枚あたり平均で約 0.028 秒であった。

人物の意味的領域分割データセットについては、ATR データセット [12] を使用した。このデータセットは、7,702 枚の画像データを含み、本研究では訓練データに 6,000 枚、検証データに 702 枚、テストデータに 1,000 枚の画像を使用した。ポーズ推定データセットには MPII Human Pose Dataset [24] を使用した。このデータセットには 24,984 枚の画像が含まれており、この中で学習用の関節情報がついた、人物が一人のみ映っている 10,298 枚のデータを学習に用いた。背景データ拡張には Indoor scene recognition データセット [8] からランダムに 6,000 枚の画像を選択し、これらの背景に人物画像を合成して ATR データセットの訓練データを 2 倍にした。なお、データセットからランダムに画像を選択しているため、人物に対して不自然な画像が選択されている可能性がある。しかし、不自然な背景でも “背景 (Bg)” の意味ラベルを持つため、人物と背景の組み合わせのバリエーションを増やす、という点で有効である。

入力する画像サイズについて、ベースとなる手法 [7] とデータ拡張を適用した場合では、100×150 の画像を入力とした。ポーズ推定を含むモデルを使用した場合、入力画像を 2 のべき乗とするため、256×256 の画像を入力とした。生成後のすべての結果は、元の入力画像サイズにリサイズして精度を比較している。

実験において推定するラベルは、Liang ら [7] と同様に、背景 (Bg)、帽子 (Hat)、髪 (Hair)、サングラス (Glass)、上着 (U-Cloth)、スカート (Skirt)、パンツ (Pants)、ドレス (Dress)、ベルト (Belt)、左右の靴 (L-Shoe、R-Shoe)、顔 (Face)、左右の足 (L-leg、R-leg)、左右の腕 （L-arm、R-arm）、バッグ (Bag)、スカーフ (Scarf) の計 18 ラベルである。

5.2 実験内容

実験では、ベースとなる手法 (Co-CNN [7]) とデータ拡張した結果 (DA)、ポーズ推定を用いた結果 (PE) を比較した。評価指標として、正解率 (Accuracy)、適合率 (Precision)、再現率 (Recall)、F1 を用いた。任意のラベル l について、予測結果と正解が l である場合の総数を P_{tp}、予測結果が l で正解が l 以外の場合の総数を P_{fp}、正解が l で予測結果が l 以外の場合の総数を P_{fn}、予測結果も正解も l 以外の場合の総数を P_{tn} とすると、各指標は以下のように計算される。

$$Accuracy = \frac{P_{tp} + P_{tn}}{P_{tp} + P_{fp} + P_{fn} + P_{tn}}, \tag{5.1}$$

$$Precision = \frac{P_{tp}}{P_{tp} + P_{fp}}, \tag{5.2}$$

$$Recall = \frac{P_{tp}}{P_{tp} + P_{fn}}, \tag{5.3}$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}. \tag{5.4}$$

訓練データ量に応じた提案手法の有効性を検証するため、意味的領域分割の訓練データを 1,000 枚と 6,000 枚に変えて実験した。この際、DA は 1,000 枚と 6,000 枚の訓練データにさらにそれぞれ 1,000 枚と 6,000 枚の背景データを拡張し、PE はどちらの場合も 10,298 枚のポーズデータを追加で利用している。学習では式 (4.2) に示す誤差関数 E_l が下がらなくなるまで繰り返し、検証データに対して最も良いものを最終結果とした。なお、Co-CNN [7] は独自に実装したものであるが、以下の理由から忠実に性能を再現することは文献 [7] の著者以外ほぼ不可能である。(1) まずソースコードが公開されておらず、(2) テストデータ、訓練データの選び方が明記されておらず、(3) HoG などの特徴量の実装方法が明記されていない。本研究では独自実装した Co-CNN をベースラインとして、少ないデータセットでも精度を改善できることを示す。

5.3 結果

表 5.1 にテストデータに対する各手法の性能を示す。各手法のアルファベットの後の数字は訓練データ数を示している。データ拡張の結果から、訓練データ数が 1,000 の場合、Co-CNN の結果よりも性能が向上していることが分かる。訓練データ数が 6,000 枚の時に性能の向上は見られないが、これは背景パターンがすでに十分に存在するためと考えられる。また、ポーズ推定の結果を転移した場合では、データ数 1,000、6,000 の場合どちらでも性能の向上が見られた。さらに、表 5.2 に示すように、クラスごとの F1 に対しても同様の傾向が見られた。特に、データ拡張は訓練データが少数の場合、背景 (Bg) を含む複数のクラスに対して数値の向上が見られた。 また、ポーズ推定の結果を転移した場合では、訓練データが 6,000 の際に、scarf を除く全てのクラスで大きな数値の向上が見られた。加えて、DA と PE を組み合わせる

表 5.1: 訓練データ数 1,000, 6,000 による各手法の性能比較.

手法	Accuracy	Precition	Recall	F1
Co-CNN1000	82.07	79.14	82.07	80.19
DA1000	83.27	81.64	83.28	81.81
PE1000	84.77	83.06	84.77	83.49
DA+PE1000	**85.18**	**84.67**	**85.18**	**84.43**
Co-CNN6000	86.15	84.79	86.15	84.95
DA6000	86.16	84.78	86.16	85.15
PE6000	88.31	88.82	89.00	88.41
DA+PE6000	**89.73**	**89.46**	**89.73**	**89.37**

(DA+PE) ことで、更なる数値の向上が確認できる。図 5.1、 5.2 に示す通り、結果画像をいくつか比較すると、データ拡張によって背景と前景の分類が上手くいった例や、ポーズ推定によって人体の部位の領域をより綺麗に抽出できた例を確認できる。

5.4 議論

表 5.2 を見ると、belt や scarf といった特定のラベルで精度が低いことが確認できる。この原因として、belt や scarf といったラベルの総数が少ないことが挙げられる。このようなデータ数の少ないラベルでの検出精度も向上させるために、class weight を用いた実験を行った。class weight は、式 (4.2) の誤差関数について、ラベルごとにラベルのピクセル総数に応じた重み係数をかけることで、各ラベルの偏りを低減する方法 [25] である。本実験における i 番目のラベルの重さ w_i は以下のように設定した。

$$w_i = \frac{|\mathcal{I}|}{|\mathcal{I}_i|} \tag{5.5}$$

ここで、$|\mathcal{I}|$ は画像 $\mathcal{I}$ 中のピクセル総数、$|\mathcal{I}_i|$ は画像 $\mathcal{I}$ における i 番目のラベルの総数である。画像中にラベルが含まれないこともあるが、その場合は $w_i = 0$ としている。

実験では、5.1 節と同様の環境で実験を行い、データ拡張とポーズ推定を組み込んだモデルで、誤差関数に class weight を設定した場合と class weight を設定しない場合との比較を行った。表 5.3、表 5.4 に class weight を設定した場合と設定しない場合でのテストデータ全体での結果の比較を示す。ここで、マイクロ平均は結果をすべて合計してから性能を計算する方法、マクロ平均はそれぞれのテスト画像で性能を計算してから平均をとる方法である。マイクロ平均ではテストデータ一つ一つの性能が、マクロ平均ではテストデータ全体での性能がそれぞれ確認できる。class weight を設定した場合では、class weight を設定しない場合に比べ、性能が全体的に低下していることが確認された。また、class weight を設定した場合で、recall のマクロ平均が向上していることが分かった。これらの理由として、ネットワークが領域の小さいラベルをできるだけ検出するように学習したため、結果的に大きいラベルの検出精度

表 5.2: 各手法におけるラベルごとの F1 の比較.

手法	Bg	Hat	Hair	Glass	U-cloth	Skirt	Pants	Dress	Belt
Co-CNN1000	93.89	**4.17**	52.46	4.08	51.40	9.63	37.41	26.66	4.14
DA1000	94.76	3.11	57.70	**9.39**	55.02	9.11	32.32	**32.48**	**4.32**
PE1000	95.54	0.29	61.34	0.52	60.96	21.48	40.65	30.49	0.00
DA+PE1000	**96.18**	0.50	**63.06**	0.00	**62.88**	**36.31**	**49.50**	16.23	0.46
手法	L-shoe	R-shoe	Face	L-leg	R-leg	L-arm	R-arm	Bag	Scarf
Co-CNN1000	25.44	25.57	61.42	42.66	41.32	31.22	27.72	12.81	0.46
DA1000	30.33	30.95	64.23	47.41	46.55	33.03	34.19	15.30	**1.03**
PE1000	**38.26**	35.75	72.23	48.85	50.18	**41.94**	39.14	28.93	0.00
DA+PE1000	36.41	**38.86**	**73.22**	**54.51**	**54.64**	41.65	**43.45**	**34.54**	0.00

手法	Bg	Hat	Hair	Glass	U-cloth	Skirt	Pants	Dress	Belt
Co-CNN6000	95.73	18.15	66.37	14.04	64.09	23.83	49.39	37.26	7.05
DA6000	95.93	0.15	68.28	8.00	63.89	28.76	50.83	36.67	4.50
PE6000	97.20	40.30	74.71	18.87	69.64	41.57	**61.55**	50.75	**21.56**
DA+PE6000	**97.55**	**45.58**	**77.22**	**31.31**	**74.46**	**47.49**	61.40	**51.67**	16.73
手法	L-shoe	R-shoe	Face	L-leg	R-leg	L-arm	R-arm	Bag	Scarf
Co-CNN6000	39.77	40.59	74.08	58.13	58.12	48.27	47.39	35.90	**3.56**
DA6000	35.96	39.70	73.62	57.82	57.54	47.50	47.01	36.99	0.37
PE6000	44.85	45.09	80.54	65.39	64.31	62.16	61.70	48.58	0.03
DA+PE6000	**45.72**	**46.09**	**82.44**	**67.11**	**66.89**	**65.07**	**63.25**	**53.32**	0.10

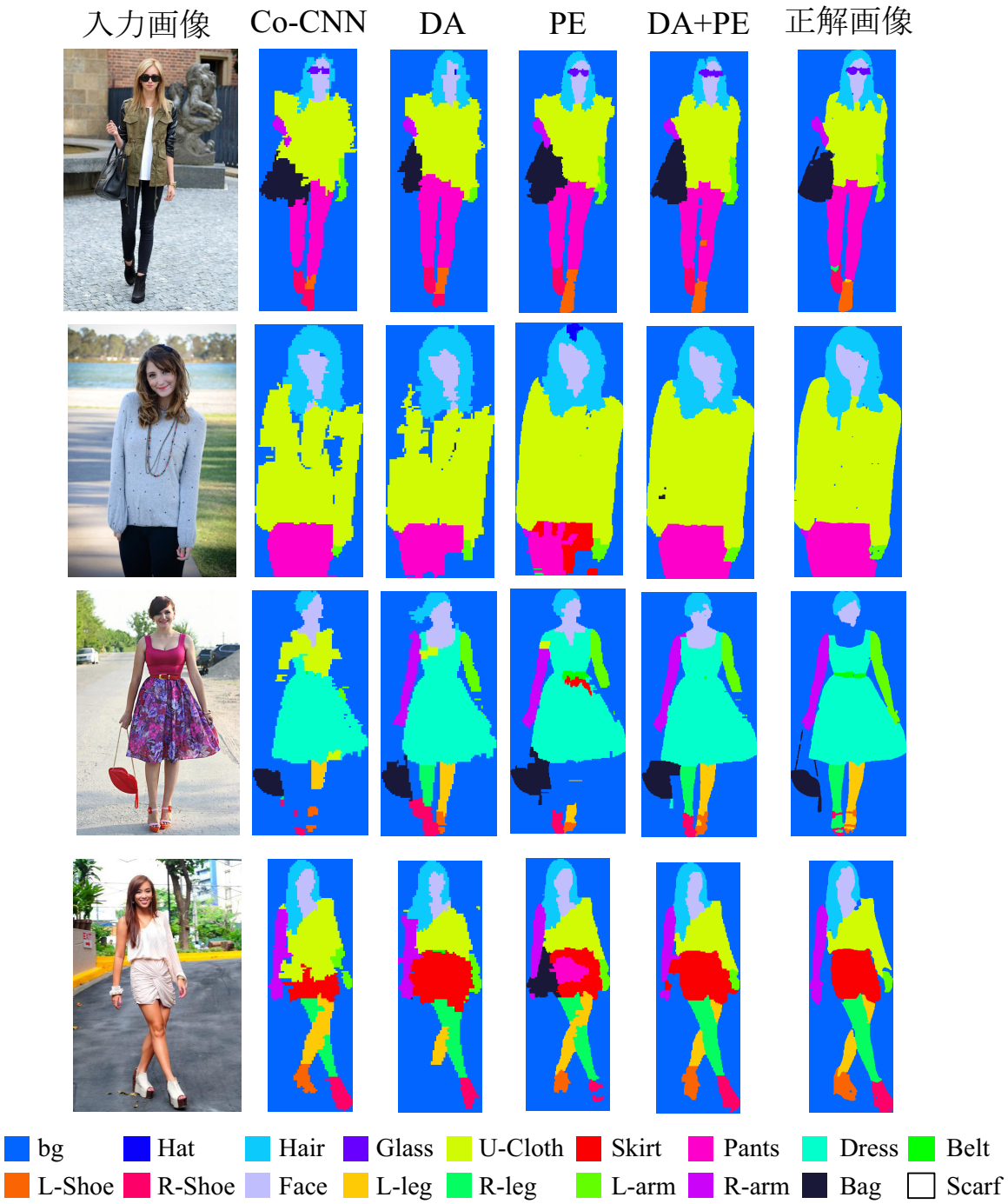


図 5.1: 各手法により得られた結果の比較 (1). 従来手法に比べ正解画像に近い結果が得られていることが分かる.

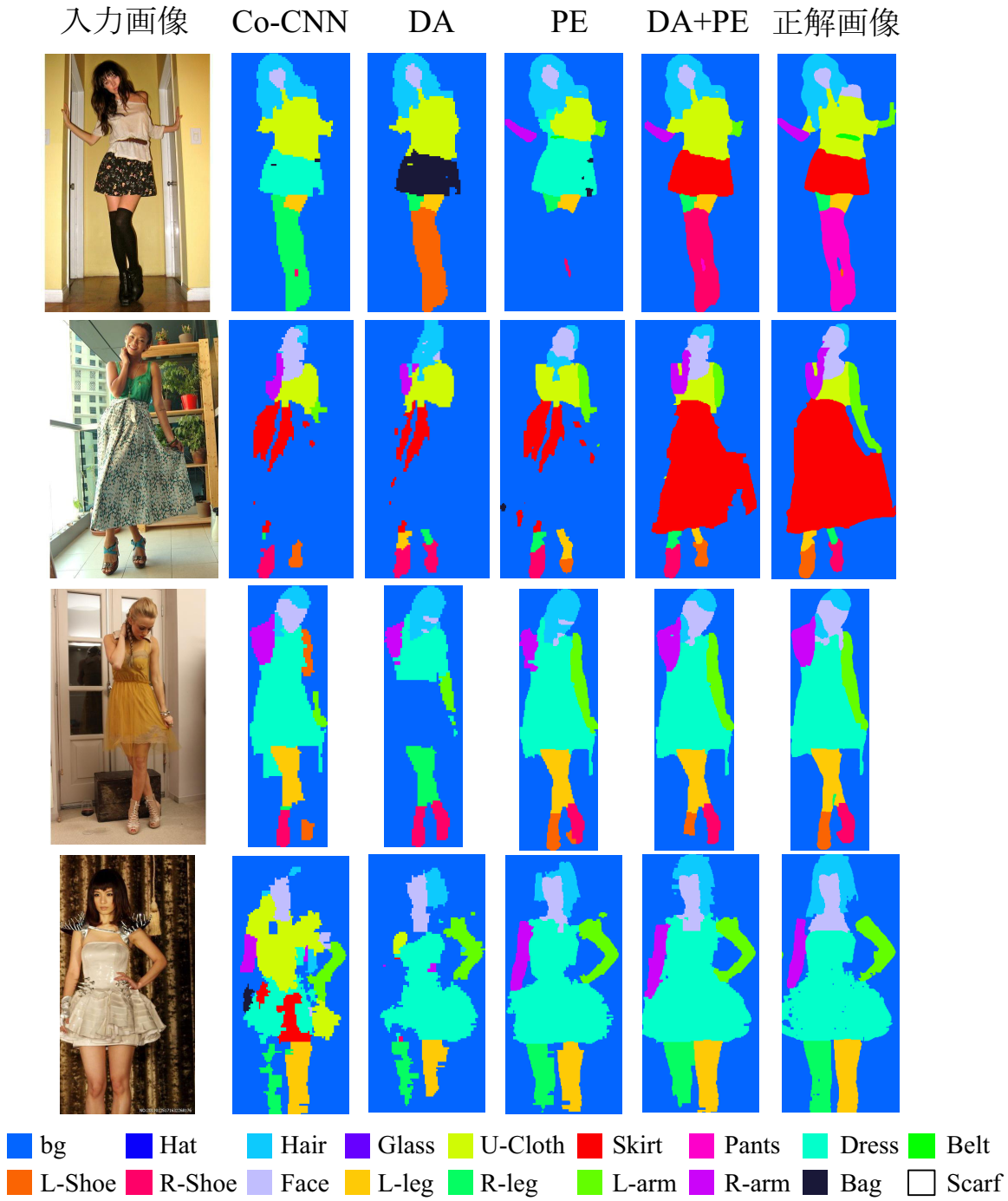


図 5.2: 各手法により得られた結果の比較 (2). 従来手法に比べ正解画像に近い結果が得られていることが分かる.

が下がり、細かいラベルの検出精度が上昇したためであると考えられる。表 5.5 に、ラベルごとの F1 の比較を示す。ピクセル総数の少ない scarf、belt、glass のクラスで精度の向上が確認できた。図 5.3、 5.4 に class weight を設定した場合としない場合での混同行列を示す。混同行列は縦軸のラベルが正解である場合に、予測された結果が横軸のラベルとなった時の確率を示した行列である。理想的には対角成分がすべて 1 で、他の成分は全て 0 となる。混同行列は、0 に近いほど青く、1 に近いほど赤く色づけされるように設定されている。予測クラスについて、glass、belt のようなラベル総数が少ない列を見ると、class weight を設定したネットワークの方が、ラベル総数の少ないラベルの検出回数が上がっていることが分かる。図 5.5 に class weight の有無による結果の比較を示す。上段や中段の画像では、class weight が無い場合には検出できていなかった glass や belt が検出できていることが確認できる。しかし、下段の画像のように背景との境界を正しく識別できなくなっている画像も確認された。そのため応用する際は用途に応じて学習したモデルを使い分けることが望ましいと考えられる。

表 5.3: class weight の有無によるマイクロ平均の比較.

手法	Accuracy	Precition	Recall	F1
class weight なし	**89.73**	**89.46**	**89.73**	**89.37**
class weight あり	84.29	87.48	84.29	85.41

表 5.4: class weight の有無によるマクロ平均の比較.

手法	Precition	Recall	F1
class weight なし	**65.91**	52.21	**55.19**
class weight あり	45.15	**60.48**	50.17

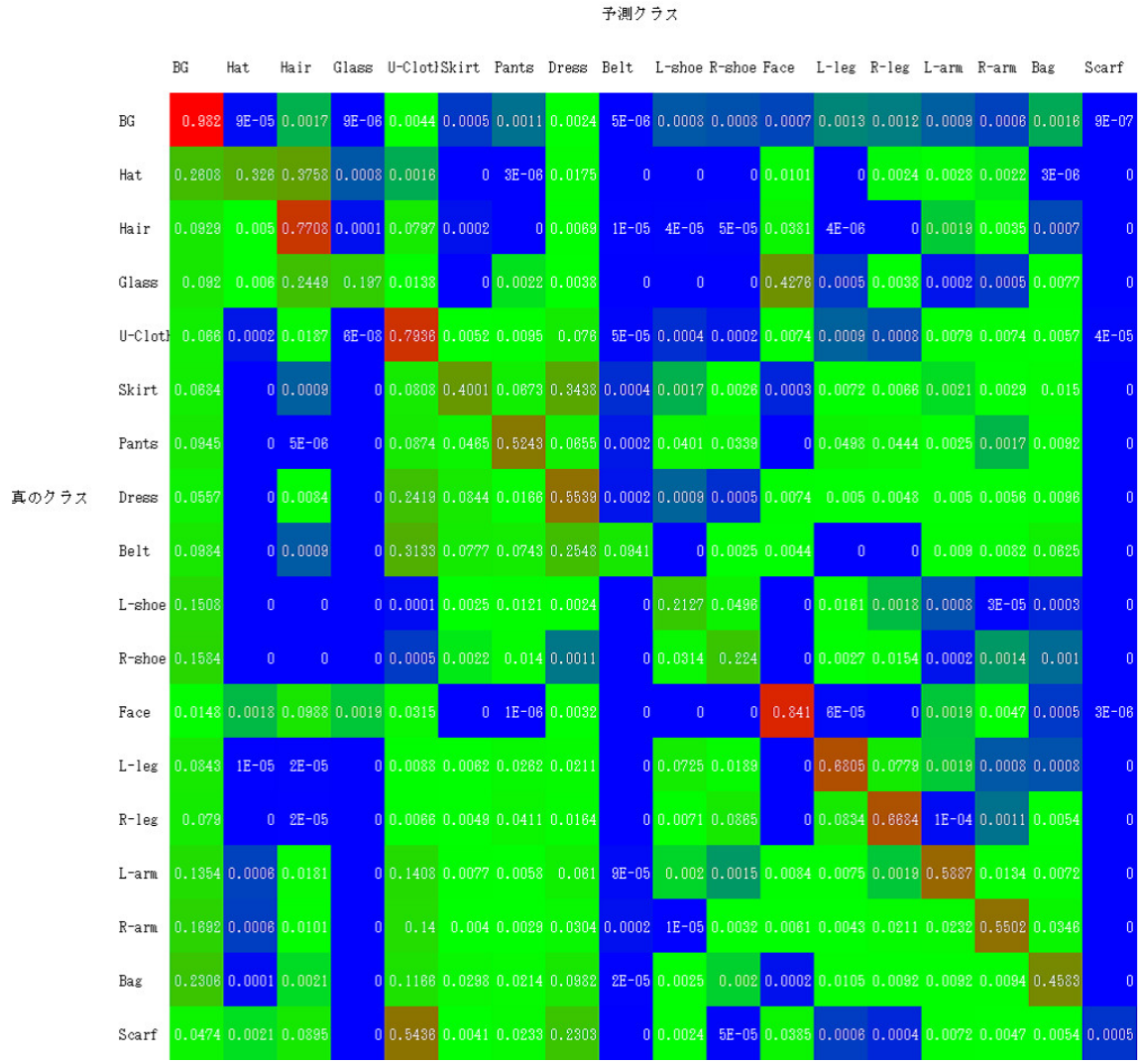


図 5.3: class weight を設定しないモデルの混同行列. 理想的には対角成分がすべて 1 で, 他の成分は全て 0 となる. 図は 0 に近いほど青く, 1 に近いほど赤く色づけされている. glass や belt のようなラベル総数が少ないラベルは, 列単位で見ると他のクラスと比べ検出頻度が低いことが分かる.

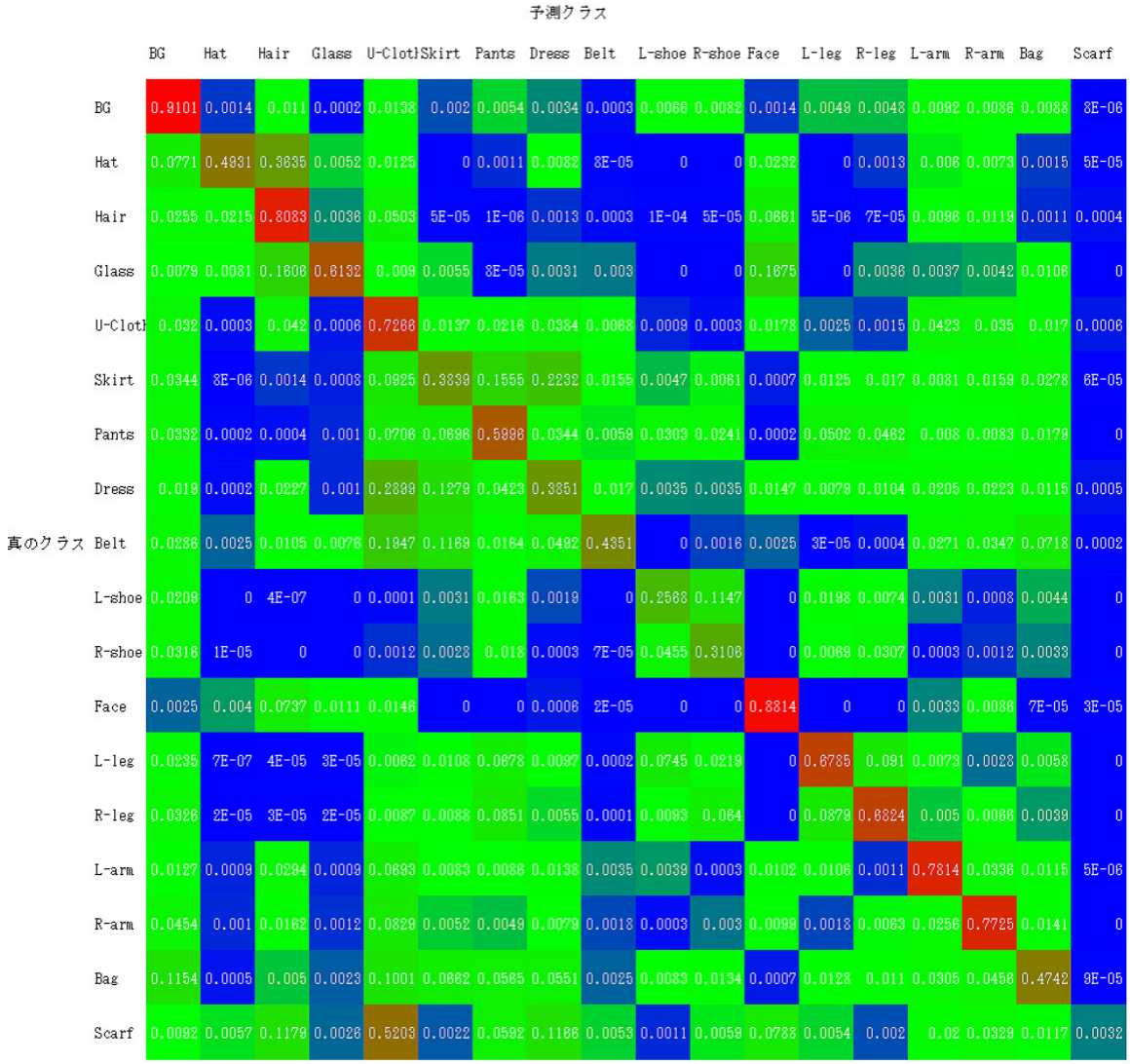


図 5.4: class weight を設定したモデルの混同行列. 図 5.3 と比べ glass や belt のラベルの検出頻度が高くなっており, 該当ラベルの正解率も向上していることが分かる.

図 5.5: class weight の有無による結果画像の比較. 上段, 中段ではラベル総数の少ない glass や belt の領域が検出できている. しかし, ラベル総数の多いラベルは推定されにくくなり, 下段のように推定に失敗する場合もある.

表 5.5: class weight の有無による各ラベルの F1 の比較.

手法	Bg	U-Cloth	Doress	Pants	Hair	Skirt
class weight なし	**97.55**	**74.46**	**51.67**	**61.40**	**77.22**	**47.49**
class weight あり	94.76	67.39	43.81	57.23	67.54	40.53
ピクセル総数	183,636,164	15,894,028	7,448,144	6,184,731	5,976,091	4,455,537
手法	Face	L-leg	R-leg	Bag	R-arm	L-arm
class weight なし	**82.44**	**67.11**	**66.89**	**53.32**	**63.25**	**65.08**
class weight あり	78.49	58.84	58.92	40.69	50.02	49.53
ピクセル総数	3,223,948	2,762,742	2,751,533	2,735,495	2,107,738	2,083,711
手法	R-shoe	L-shoe	Hat	Scarf	Belt	Glass
class weight なし	**46.09**	**45.72**	**45.58**	0.10	16.73	31.31
class weight あり	38.72	37.37	44.46	**0.62**	**26.58**	**47.56**
ピクセル総数	1,244,905	1,241,236	571,702	485,594	227,207	146,481

第6章　アプリケーション

本手法の意味的領域分割結果が実応用でも有効であることを示すために、人物の意味的領域分割の結果を活用し、衣服の色変更、テクスチャの転写、ファッション分析のための可視化などのアプリケーションを開発した。

6.1　衣服の色変更

人物の意味的領域分割により得られた特定の衣類の領域の色を、自動的に変える簡単なアプリケーションを実現した。図 6.1 に、衣服の色変更の流れを示す。まず、抽出された衣服領域を自然に転写するためトライマップを生成し、トライマップからアルファマットを生成する。ここでトライマップとは、マスク領域 (図 6.1(c) では白色)、マスク領域か否かが不明な領域 (灰色)、マスクでない領域 (黒色) の 3 つの領域からなるマップである。本手法でのトライマップは、指定した衣服領域にモルフォロジ演算を適用することで生成する。アルファマットは、マスクの領域をそのまま合成すると輪郭部分に発生してしまうジャギーを抑えるために、マスクの輪郭領域のアルファ値を調整したマップである。アルファマットの生成には、最新の手法 [26] を使用した。次に、CIE Lab 色空間の ab チャンネルをユーザが指定した色に置き換え、アルファマットの領域の色を変更する。最後に、色領域を調整するため、色の変更前の画像の隣接ピクセルを基にしたジョイントバイラテラルフィルタを使用してマットの輪郭周りの色を平滑化した。

図 6.2、6.3 に衣服の色を変更した結果を示す。ポーズ推定とデータ拡張から生成されたマスクのアルファマットと再現結果は、その他の結果よりも正解画像の結果に近いことが分かる。

6.2　衣服のテクスチャの転写

参照画像の特定の衣服領域をターゲット画像に転写するアプリケーションを実装した (図 6.4)。まず、色変更と同様の手法によりターゲット画像と参照画像のアルファマットを生成する。次に、二値化したアルファマットの輪郭から、Mean Value Coordinate [27] によりテクスチャ座標を計算する。これにより歪ませたテクスチャを、最終的にターゲット画像のアルファマットに合わせて合成する。合成の際は、オーバーレイ合成により、ターゲット画像に起因する陰影を維持している。オーバーレイ合成では、位置 (i, j) での チャンネル c のピクセルの色

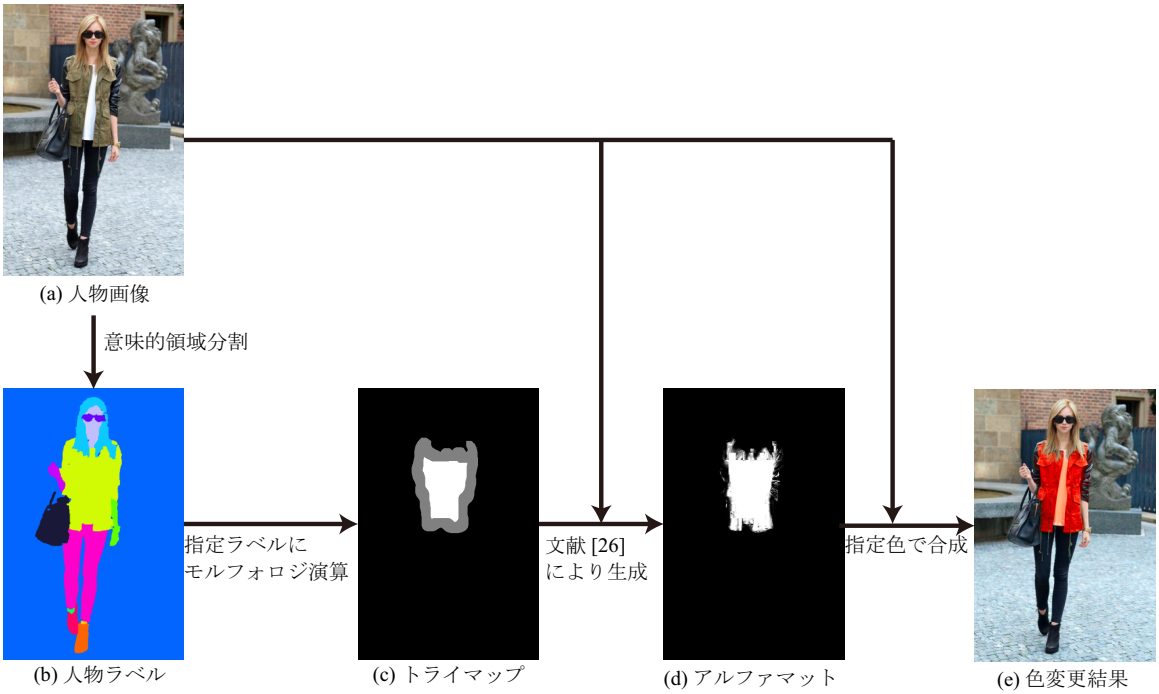


図 6.1: 衣服の色変更の流れ. (a) 入力画像から (b) 人物ラベルを推定し, 色変更したいラベルに対しモルフォロジ演算を行い, (c) トライマップを生成する. トライマップから (d) アルファマットを生成し, アルファマットで指定される領域について, 入力画像の Lab 色空間の ab を指定色で置き換えることで (e) 最終結果を得る.

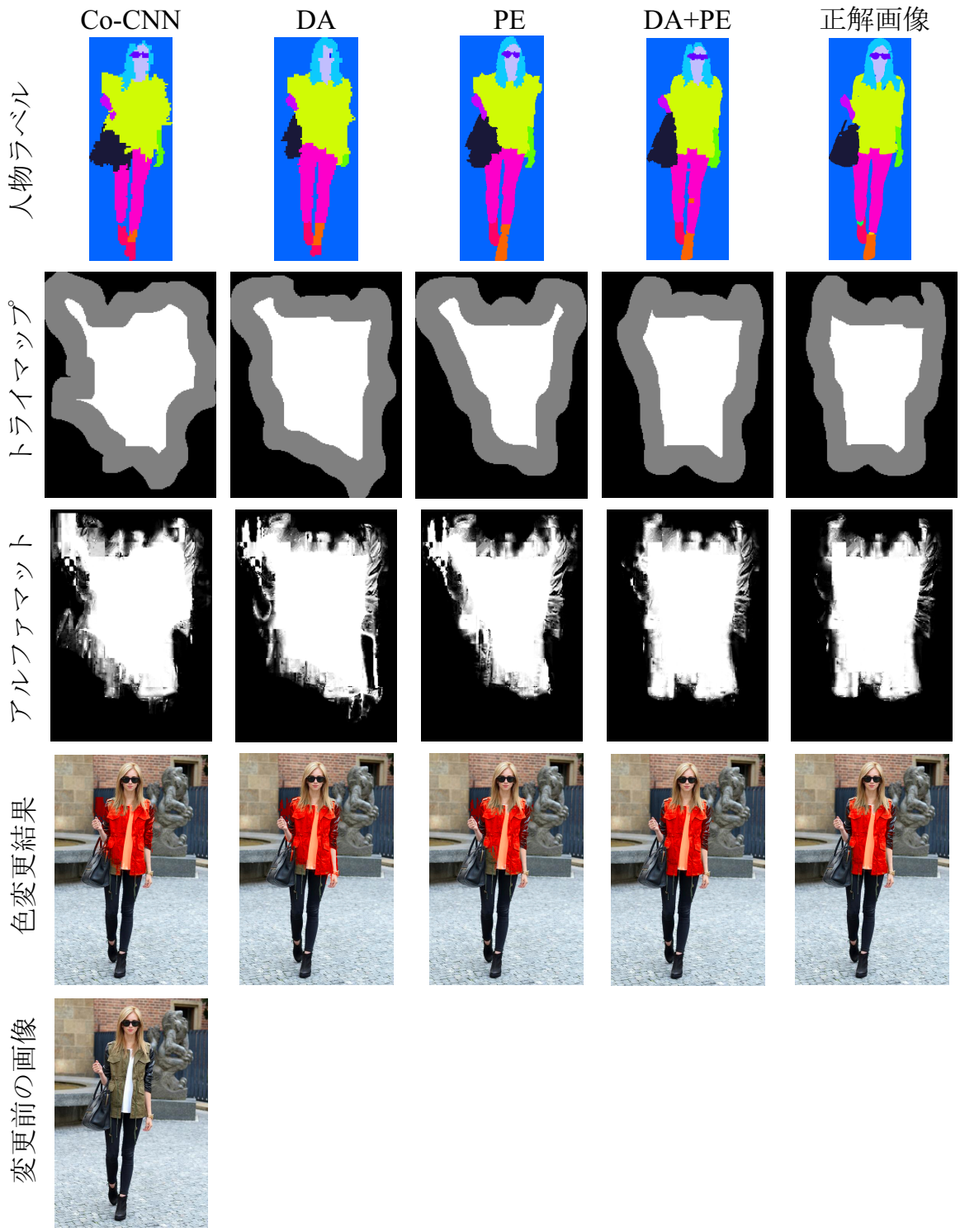


図 6.2: 上着に対する色変更の結果. ポーズ推定とデータ拡張を組み合わせた結果から得られたアルファマットは, 正解画像から得られたアルファマットに近いことが分かる.

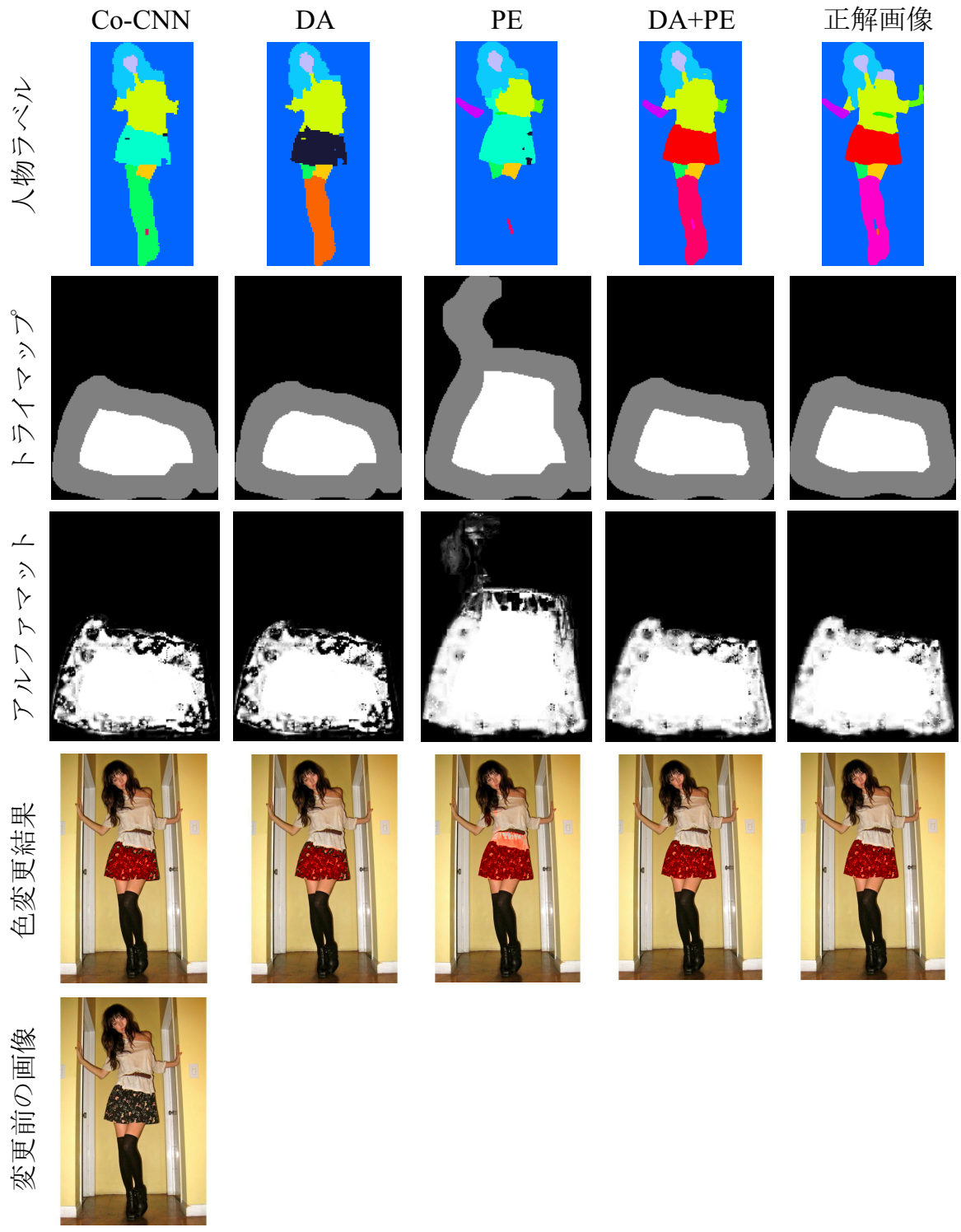


図 6.3: スカートに対する色変更の結果. ポーズ推定とデータ拡張を組み合わせた結果から得られたアルファマットは, 正解画像から得られたアルファマットに近いことが分かる.

$p_{i,j,c}$ は以下のように計算される。

$$p_{i,j,c} = \begin{cases} 2 \times p_{i,j,c}^{reference} \times p_{i,j}^{gray} & \left(\text{if } \mathrm{p}_{\mathrm{i,j}}^{\mathrm{gray}} < \tau\right) \\ 1 - 2 \times \left(1 - p_{i,j,c}^{reference}\right) \times \left(1 - p_{i,j}^{gray}\right) & (otherwise.) \end{cases} \tag{6.1}$$

ここで、$p_{i,j,c}^{reference}$ は参照画像のピクセル値、$p_{i,j}^{gray}$ はターゲット画像のグレースケール変換された値である。τ は閾値で、ピクセル値の範囲を $0 \leq p_{i,j}^{gray} \leq 1$ とした場合、本研究では $\tau = 0.5$ となるように設定している。

(a) ターゲット画像　(b) 参照画像　(c) テクスチャ転写結果

図 6.4: 衣服のテクスチャの転写結果. (a) ターゲット画像と (b) 参照画像のスカートのマスク (画像右下) を生成し, テクスチャを転移し (c) 結果画像を得る.

6.3　ファッション分析のための可視化

人物の意味的領域分割の結果を利用し、ファッション分析用の人物画像の可視化を行った。これにより、ユーザは衣服の類似性を測定し、二次元空間内に人物画像をマッピングすることでファッションの分析が可能となる。具体的には、各画像からカラーチャネルごとに 128 個のビンからなる正規化された RGB ヒストグラムを特徴量として抽出している。この処理では、複数の RGB ヒストグラムが、ユーザによって指定された K 種類の領域 ($0 \leq K \leq 17$) から個別に計算される。その後、すべての RGB ヒストグラムを連結し、各画像の $128 \times 3 \times K$ 次元のベクトルを取得する。これらの高次元特徴を二次元空間に埋め込むために、t-SNE [28] を使用した。これらの特徴は視覚的に一貫性を持つ結果を得るには十分だが、将来的にはより複雑な特徴を検討したいと考えている。

図 6.5 にテストデータの可視化結果を示す。図 6.5(a) に示されるように、画像の全領域の特徴を使用した場合 (ラベルを指定しない場合)、類似した人物は近くに位置せず、背景の色に大きく依存した配置となる。対照的に、図 6.5(b) では、人物の意味的領域分割の結果を使用した有効性を示している。この場合、背景の色に関係なく、選択された衣類 (この場合、帽子) で類似した色が互いに近い位置に配置されるように人物がマッピングされていることが分かる。さらに、ユーザは 図 6.5(c) で示されるように複数のラベルを選択できる。この場合は、ユーザはパンツ、スカート、上着の 3 つを指定している。画像では、主に衣類の種類 (例えば、パンツとスカート) と各ラベルの色によって によって分類され、同じラベルで、かつ色が類似している画像が近くに配置されていることが分かる。図中で矩形で囲まれた部分では、オレンジ色のパンツや明るい色の上着といった、複数のラベルを利用した衣服のコーディネートを分析することもできる。これらの結果は、人物の意味的領域分割がファッション可視化に有効であることを示している。

なお、Simo-Serra と Ishikawa も CNN ベースの特徴を使ってファッションの可視化を行っているが [29]、彼らのアプローチは人物の前景と背景を大まかに区別しているのに対して、本手法は人物の意味的領域分割によって得られた詳細な服の種類を考慮できる点が特徴である。

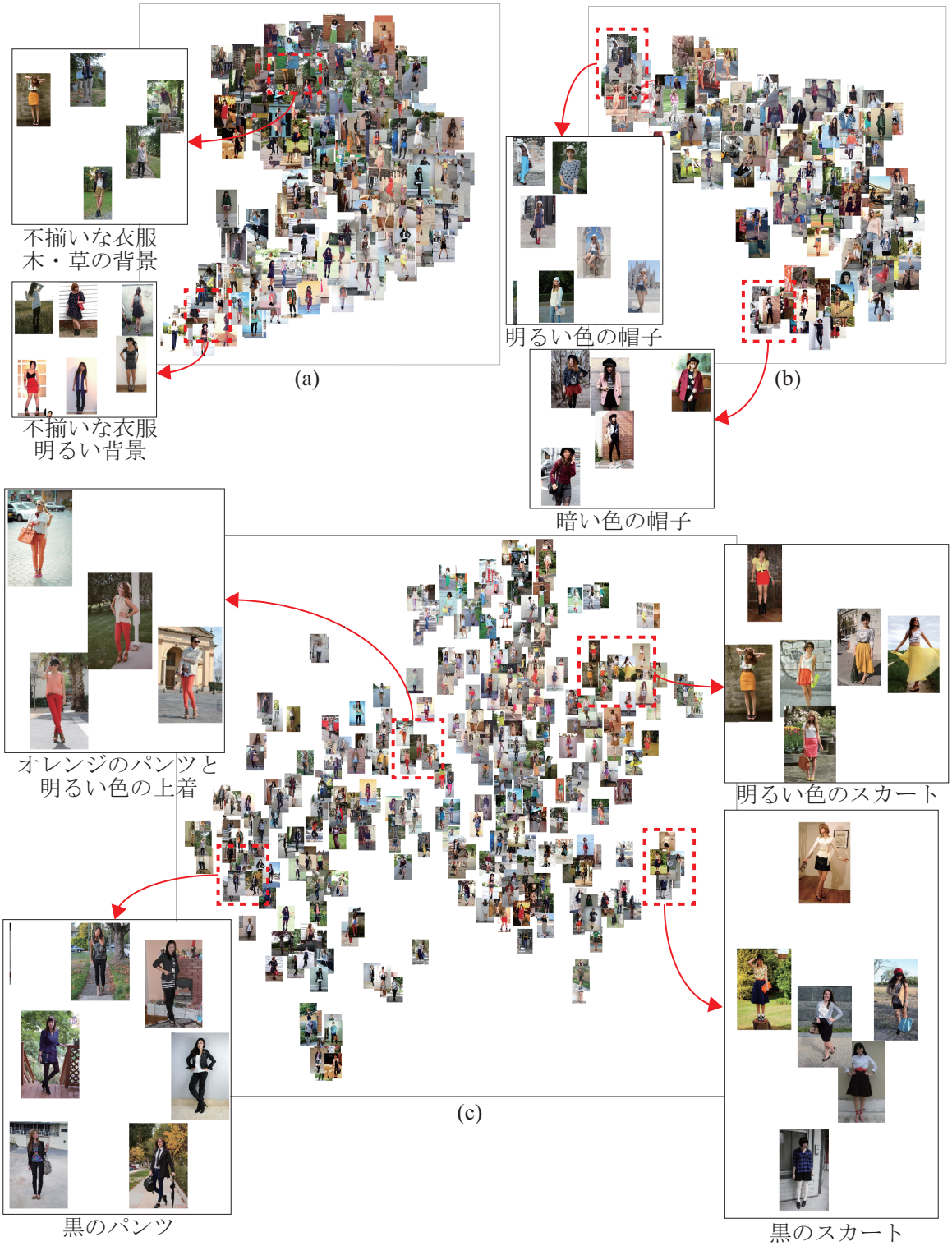


図 6.5: (a) 全領域 (ラベルの指定なし), (b) 帽子ラベル, (c) パンツ, スカート, 上着のラベル特徴に基づき, t-SNE [28] を使用してファッション分析用に人物画像の可視化を行った結果.

第7章　結論

本研究では、人物画像に対する意味的領域分割において新たなポーズ推定の学習結果を転移する手法とデータ拡張手法を提案した。ポーズ推定の転移では、人物の意味的領域分割ネットワークとポーズ推定ネットワークを組み合わせた。データ拡張では、既存の背景画像とラベル割り当て済み画像を組み合わせることで、新たな学習データを生成した。また、既存研究との比較を行い、本手法のデータ拡張とポーズ推定結果の転移が人物のラベル割り当てに対して有用な手法であることも示した。加えて、いくつかのアプリケーションを作成し、意味的領域分割の結果が応用に有効であることも示した。

7.1　今後の課題

総数の少ないラベルでも学習が行えるように class weight による学習を行い、少ないラベルの検出精度が向上したが、一方で多数のラベルの精度や全体的な精度が低下してしまった。そのため、全体の精度をできるだけ保ちつつ、少ないラベルの精度を向上させたいと考えている。具体的には、class weight を使用した場合と使用しない場合で学習した結果を組み合わせる方法や、背景と前景のマスクを学習し、その後前景に対し class weight による学習を行う方法が考えられる。

謝辞

本論文の執筆にあたり、金森由博先生、遠藤結城先生、三谷純先生には多くのご助言やご指導をいただきました。特に金森由博先生には研究方針や論文執筆面でのご指導、遠藤結城先生には深層学習に関する技術的な面でのご指導をいただき、心より感謝申し上げます。また、非数値処理アルゴリズム研究室の皆様には、日頃の生活や研究に関する意見等、様々な面でサポートをしていただきましたことを、感謝申し上げます。特に、橋本泰輔君には、実装の手伝いや論文のチェック等でサポートをしていただきましたことをここに感謝申し上げます。

参考文献

[1] Yoshihiro Kanamori, Hiroki Yamada, Masaki Hirose, Jun Mitani, and Yukio Fukui. Image-based virtual try-on system with garment reshaping and color correction. *Part of the Lecture Notes in Computer Science book series (LNCS)*, Vol. 9550, pp. 1–16, 2016.

[2] D. Wei, W. Catherine, B. Anurag, P. Robinson, and S. Neel. Style finder: Fine-grained clothing style detection and retrieval. In *Proc. of CVPR Workshops*, pp. 8–13, 2013.

[3] Y. Hu, X. Yi, and L. S. Davis. Collaborative fashion recommendation: A functional tensor factorization approach. In *Proc. of ACM international conference on Multimedia*, pp. 129–138, 2015.

[4] Y. Kalantidis, L. Kennedy, and L.-J. Li. Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proc. of ACM conference on international conference on multimedia retrieval*, pp. 105–112, 2013.

[5] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proc. of CVPR 2016*, pp. 3150–3158, 2016.

[6] S. Wei, V. Ramakrishna, T. kanade, and Y. Sheikh. Convolutional pose machines. In *Proc. of CVPR 2016*, pp. 4724–4732, 2016.

[7] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan. Human parsing with contextualized convolutional neural network. *IEEE Trans. on PAMI*, Vol. 39, No. 1, pp. 115–127, 2016.

[8] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proc. of CVPR 2009*, pp. 413–420, 2009.

[9] K. Yamaguchi, M. Kiapour, L. Ortiz, and T. Berg. Parsing clothing in fashion photographs. In *Proc. of CVPR 2012*, pp. 3570–3577, 2012.

[10] K. Yamaguchi, M. Kiapour, L. Ortiz, and T. Berg. Retrieving similar styles to parse clothing. *IEEE Trans. on PAMI*, Vol. 37, No. 5, pp. 1028–1040, 2014.

[11] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. A high performance crf model for clothes parsing. In *Proc. of ACCV 2014*, pp. 869–877, 2014.

[12] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, L. Lin J. Dong, and S. Yan. Deep human parsing with active template regression. *IEEE Trans. on PAMI*, Vol. 37, No. 12, pp. 2402–2414, 2015.

[13] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, and S. Yan. Matching-CNN meets KNN: Quasiparametric human parsing. In *Proc. of CVPR 2015*, pp. 1419–1427, 2015.

[14] B. Gedas, S. Jianbo, and T. Lorenzo. Semantic segmentation with boundary neural fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3602–3610, 2016.

[15] Golnaz Ghiasi and Charless C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *Proc. of ECCV 2016*, pp. 519–534, 2016.

[16] G. Lin, C. Shen, A. van den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proc. of CVPR 2016*, pp. 3194–3203, 2016.

[17] V. Raviteja, T. Oncel, L. Ming-Yu, and C. Rama. Gaussian conditional random field network for semantic segmentation. In *Proc. of CVPR 2016*, pp. 3224–3233, 2016.

[18] Xiaodan Liang, Xiaohui Shen, Donglai Xiang, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with local-global long short-term memory. In *Proc. of CVPR 2016*, pp. 3185–3193, 2016.

[19] Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with graph LSTM. In *Proc. of ECCV 2016*, pp. 125–143, 2016.

[20] S. Hong, J. Oh, H. Lee, and B. Han. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *Proc. of CVPR 2016*, pp. 3204–3212, 2016.

[21] G. Papandreou, L. Chen, K. P. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proc. of ICCV 2015*, pp. 1742–1750, 2015.

[22] Y. Wei, O. Wanli, L. Hongsheng, and W. Xiaogang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Proc. of CVPR 2016*, pp. 3073–3082, 2016.

[23] C. Xiao, O. Wanli, L. Hongsheng, and W. Xiaogang. Structured feature learning for pose estimation. In *Proc. of CVPR 2016*, pp. 4715–4723, 2016.

[24] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *Proc. of CVPR 2014*, pp. 3686–3693, 2014.

[25] Chao Chen and Andy Liaw. Using random forest to learn imbalanced data. Technical report, Technical report, University of California, http://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf, 2004.

[26] Y. Aksoy, T. O. Aydın , and M. Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *Proc. of CVPR 2017*, pp. 29–37, 2017.

[27] M. S. Floater. Mean value coordinates. *Journal Computer Aided Geometric Design*, Vol. 20, No. 1, pp. 19–27, 2003.

[28] L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, Vol. 9, pp. 2579–2605, 2008.

[29] Edgar Simo-Serra and Hiroshi Ishikawa. Fashion Style in 128 Floats: Joint Ranking and Classification using Weak Data for Feature Extraction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 298–307, 2016.