

平成30年度
中部大学大学院工学研究科情報工学専攻

博士学位論文

Attention機構による深層学習の高精度化と
判断根拠の理解に関する研究

福井 宏

論文要旨

画像認識の分野では、2012年に開催された大規模画像認識コンペティションにおいて Deep Convolutional Neural Network (CNN) をベースにした手法が登場して以降、様々な画像認識タスクにおいて高い性能を達成している。しかしながら、CNN は精度向上のためには複雑なネットワーク構造を構築する必要があり、以下の3つの問題が生じる。1つ目は、ネットワークのパラメータ数の増加に伴い、計算コストが大幅に増加する。State-of-the-art の CNN では精度向上を目的に多層のネットワークが用いられるが、これらのネットワークのパラメータ数が膨大に増加する。パラメータ増加は、組み込み機器や携帯端末等の小型機器に導入する際に問題となる。そのため、認識精度が高く、計算コストの低い深層学習のアプローチが必要とされている。2つ目は、サンプル数の割合や学習誤差のばらつき等の、アンバランスな学習に対して脆弱である。この問題は、一つのサンプルに対して複数のタスクの応答値を出力する Multi-task Learning において発生する。サンプルが持つ複数タスクのラベルは、タスクにおける各クラスのサンプル数の割合に偏りが生じる。また、複数のタスクのラベルから学習誤差関数を割り振ることでネットワークを学習するため、各タスクの学習誤差にばらつきが発生する。3つ目は、推論時の出力結果における判断根拠が不明確である。CNN はネットワークの構造が複雑なため、認識結果に対する判断根拠の解析が困難である。そのため、どのような判断根拠で認識したのかを容易に解析できるネットワークが必要である。

本研究では、はじめに認識精度とパラメータ数のトレードオフな関係について解消する。この問題を解決するために、Dropout を推論時に用いる手法を提案する。従来の Dropout は学習時において、ランダムに選択した特定のユニットにつながる結合重みを 0 にする処理である。提案手法では、推論時に Dropout を施した全結合層を複数構築する。これにより、低コストに擬似的なネットワークを複数生成しており、CNN の汎化性能を向上できる。

次に、Multi-task Learning におけるアンバランスな学習における問題を解決する。1つ目は、各タスクにおける学習誤差のばらつきを抑制するために、重み付き誤差関数を導入する。重み付き誤差関数の導入により、各タスクの学習誤差のばらつきを抑制して学習を安定化させることで、Multi-task Learning の性能を向上させる。2つ目は、サンプル数の割合の偏りを調節するために、稀少度を導入する。サンプル数の割合から算出した稀少度を基準に CNN へ入力するサンプルを決定することで、異なるサンプル数の割合のクラスを調整しながら CNN を学習できる。稀少度の導入により、サンプル数の割合が少ないクラスの認識精度を向上できる。

最後に、3つ目の問題に対して本研究では、CNN の推論時における判断根拠を解析するアプローチを提案する。従来の視覚的説明モデルは Attention map により推論時の注視領域を可視化するが、注視領域を獲得するためにネットワークの構造を修正する必要がある。この修正と再学習は、特定のモデルとタスクに対して性能の低下を招きやすい。本研究では、獲得した視覚的説明の Attention map を Attention 機構へ応用することで、視覚的解析の役割を保ちつつ、画像認識の性能を向上する手法を提案する。また、提案した Attention 機構を深層強化学習によるエージェントの制御に導入することで、これまで困難な課題であった深層強化学習における判断根拠の解析を可能にした。

目次

第 1 章	序論	1
1.1	研究の背景	2
1.2	研究目的	3
1.3	本論文の構成	5
第 2 章	画像認識分野における深層学習の関連研究	6
2.1	Deep Convolutional Neural Network	6
2.1.1	画像分類における CNN のモデル	7
2.1.2	CNN の汎化性能を向上させる手法	11
2.2	物体検出	16
2.3	歩行者検出	17
2.3.1	歩行者検出法の歴史	18
2.3.2	歩行者検出のデータセットと評価指標	20
2.3.3	HOG 特徴量と CNN の特徴表現	23
2.3.4	2 段階構造による歩行者検出	24
2.3.5	Region Proposal ベースの歩行者検出法	29
2.4	Multi-task Learning	32
2.5	まとめ	33
第 3 章	Random Dropout と Ensemble Inference Networks による歩行者検出と標識認識	35
3.1	関連研究	36
3.2	提案手法	37
3.2.1	Random Dropout	37
3.2.2	Ensemble Inference Networks	37
3.2.3	応答値の統合	38
3.3	提案手法による歩行者検出	39
3.4	評価実験	39
3.4.1	Random Dropout の性能評価	41
3.4.2	Ensemble Inference Networks の性能評価	41
3.4.3	従来法の性能比較	43

3.4.4	考察	45
3.5	まとめ	46
第 4 章	Multi-task Learning による複数属性の認識	48
4.1	関連研究	49
4.1.1	重み付き誤差関数	49
4.1.2	不均衡なデータセットに対するアプローチ	49
4.2	重み付き誤差関数を導入した Heterogeneous Learning による顔属性認識	50
4.2.1	提案手法	51
4.2.2	評価実験	53
4.3	稀少度を用いた歩行者属性認識	60
4.3.1	提案手法	61
4.3.2	評価実験	65
4.4	まとめ	70
第 5 章	Attention Branch Network による物体認識	71
5.1	関連研究	73
5.1.1	画像認識分野における可視化	73
5.1.2	Attention 機構	74
5.2	Attention Branch Network	74
5.2.1	Attention branch	75
5.2.2	Perception branch	76
5.2.3	ネットワークの学習方法	77
5.2.4	その他の画像認識タスクへの応用	77
5.3	評価実験	79
5.3.1	画像分類の実験概要	79
5.3.2	画像分類	80
5.3.3	他の画像認識タスクへの応用	84
5.4	まとめ	88
第 6 章	Attention 機構による深層強化学習の視覚的解析	91
6.1	関連研究	92
6.1.1	深層強化学習	92
6.1.2	強化学習における可視化の関連研究	93
6.2	提案手法	94
6.2.1	提案手法のネットワーク構造	95
6.2.2	Feature extractor	95
6.2.3	Value branch	96

6.2.4	Policy branch	96
6.2.5	A3C への応用	96
6.3	評価実験	97
6.3.1	性能比較	97
6.3.2	Attention map の可視化	99
6.4	まとめ	102
第 7 章	結論と展望	104
7.1	結論	104
7.2	展望	105
謝 辞		106
参考文献		107
研究業績一覧		119

目 次

1.1	本論文の構成.	5
2.1	画像認識タスクの種類	7
2.2	CNN の処理の流れ	8
2.3	画像分類におけるデータセット	9
2.4	ResNet の派生系ネットワークの構造	10
2.5	活性化関数の例	11
2.6	Batch Normalization の発展系の手法 (文献 [1] から引用)	14
2.7	Learning rate drop の例	15
2.8	CNN ベースの物体検出法	16
2.9	歩行者検出性能の遷移	18
2.10	Caltech Pedestrian Dataset における歩行者検出法の推移	19
2.11	歩行者検出のデータセット例	21
2.12	2 段階の検出構造の効果	24
2.13	ICF 特徴量の抽出 (文献 [2] 引用)	25
2.14	Filtered Channel Feature のフィルタパターン (文献 [3] 引用)	26
2.15	Faster R-CNN による歩行者検出結果	30
2.16	Heterogeneous Learning を導入した CNN	32
3.1	Dropout の発展系の手法	36
3.2	Random Dropout のアルゴリズム	37
3.3	EIN のアルゴリズム	38
3.4	2 段階処理による歩行者検出	39
3.5	Dropout と関連する手法と Random Dropout の性能比較	40
3.6	生成する全結合層数の変化による精度の比較	41
3.7	歩行者検出における提案手法と従来法の比較	42
3.8	歩行者検出結果の比較 (Daimler Mono Pedestrian Dataset)	43
3.9	歩行者検出結果の比較 (Caltech Pedestrian Dataset)	44
3.10	GTSRB Dataset における提案手法と従来の CNN の上位 3 つのスコアの比較	45
3.11	全結合層の生成数による識別時の処理時間	47

4.1	Heterogeneous Learning を導入した CNN の学習誤差と顔器官点検出結果	51
4.2	基準タスクの選択	52
4.3	従来法と提案手法の学習誤差の比較	54
4.4	従来法と提案手法の認識精度の比較	55
4.5	認識結果の可視化	56
4.6	誤差関数統一による推定精度の比較	57
4.7	しきい値を変化させた際の分類精度	58
4.8	カーネルと特徴マップの可視化結果	59
4.9	Heterogeneous Learning を導入した CNN で認識する歩行者属性	60
4.10	各歩行者属性における学習サンプル数と認識精度の関係	61
4.11	従来の Heterogeneous Learning による歩行者属性認識	62
4.12	稀少度の定義	62
4.13	稀少度に従った Data Augmentation	64
4.14	ミニバッチの候補の生成	65
4.15	各パラメータにおける提案手法の認識精度	68
4.16	提案手法と従来法における 1epoch 中に使用した学習サンプル数	69
4.17	従来法と提案手法の認識精度の比較	69
4.18	VGGNet による認識精度の比較	70
4.19	従来法と提案手法の認識結果の可視化	70
5.1	Class Activation Mapping と Attention Branch Network のネットワーク構造	72
5.2	Gradients weight-CAM の構造	73
5.3	Attention Branch Network の構造	75
5.4	Attention Branch Network を導入した Faster R-CNN の構造	77
5.5	Multi-task Learning を導入した Attention Branch Network の構造	79
5.6	CIFAR100 における学習と評価の top-1 error と学習誤差. Left 評価サンプルに対する Top-1 error. Middle 学習サンプルに対する Top-1 error. Right 学習誤差の推移.	80
5.7	CAM, Grad-CAM, Attention Branch Network による Attention map の比較結果	83
5.8	Attention branch の出力結果に対する Attention map の例	84
5.9	ImageNet Dataset における ABN の Attention map の例	85
5.10	CompCars Dataset における ABN の Attention map の例	86
5.11	CityPerson Dataset における ABN の Attention map の例	87
5.12	自動車制御における Attention map の例	88
5.13	CelebA Dataset における ABN の Attention map の例	89
6.1	従来の Actor-Critic 法と提案手法の Actor-Critic 法の構造	93
6.2	Deep Q-Network と Actor-Critic 法の関係	94
6.3	提案手法のネットワーク構造	95

6.4	提案手法の Attention map と Saliency map [4] の比較例	100
6.5	シューティングゲームにおける Attention map の例	101
6.6	Breakout と MsPacman の Attention map の例	102
6.7	DemonAttack における従来の A3C と提案手法の比較	103

表 目 次

2.1	各データセットの詳細	20
2.2	CNN ベースの歩行者検出法の分類	23
2.3	2 段階による歩行者検出と Region Proposal による歩行者検出の性能比較	26
2.4	2 段階による歩行者検出と Region Proposal による歩行者検出の性能比較	29
3.1	各データセットにおける CNN の構造	39
3.2	GTSRB Dataset における提案手法と従来法の比較	42
3.3	学習サンプル数に対する Miss rate の推移	45
3.4	全結合層の層数に対する Miss rate の推移	46
3.5	全結合層のユニット数に対する Miss rate の推移	46
3.6	学習時の処理時間	46
4.1	Heterogeneous Learning を導入した CNN の構造	53
4.2	実験に使用する CNN の構造	67
5.1	Attention 機構による精度の変化	80
5.2	CIFAR10, CIFAR100, SVHN, ImageNet Dataset における Top-1 error の比較	81
5.3	ImageNet Dataset における SENet との比較	82
5.4	CompCars Dataset における識別率の比較	84
5.5	CityPerson Dataset における検出精度の比較	86
5.6	Mean squared error on vehicle control	87
5.7	CelebA Dataset における認識精度の比較	89
6.1	従来法のネットワーク構造	98
6.2	提案手法のネットワーク構造	98
6.3	評価時におけるゲームスコアの比較	99

第1章

序論

本章では，本研究の背景及び目的，本論文の構成について述べる．

1.1 研究の背景

画像中に存在する物体のカテゴリや位置等を認識する画像認識技術は、Advanced Driver Assistance System (ADAS) や防犯システム、SNS やスマートフォンのアプリケーション等で幅広く応用されている。ADAS では、車載カメラで取得した走行画像から歩行者や自動車を検出することで、自動ブレーキシステム等の事故を防止するための技術に用いられる。防犯システムでは、人検出や顔検出、顔認識等が防犯カメラに映った不審者を検出、特定するために用いられる。SNS やアプリケーション等のレジャーな分野においても、顔の形状を画像等から取得して自動で編集するアプリケーションへ応用されたり、システムロック等で顔照合技術が用いられる。このように、画像認識技術が身近に取り入れられるようになった要因の一つとして、深層学習の発展により認識性能が向上したことが深く関係している。画像認識分野における深層学習では、Deep Convolutional Neural Network (CNN) [5, 6] をベースにした手法が一般的に用いられている。CNN が一般的に用いられる前の特徴量は、これまで研究者の知識や経験則から設計された特徴量が用いられていた [7, 8]。CNN は、複数のカーネルで構築される畳み込み層を主体に構築されたニューラルネットワークであり、学習によりカーネルを更新することで画像認識に有効な特徴量を獲得する。CNN が学習により獲得する特徴量は、様々な認識タスクに応用が可能であり、現在の画像認識における様々なベンチマークで CNN をベースとした手法が数多く提案されている。

様々な用途に応用できる CNN であるが、以下の3つの問題を抱えている。まず、CNN のパラメータ数とネットワークの精度がトレードオフな点である。CNN の精度向上には、ネットワークを構築する際に大量の層を積み重ねたり、複数のネットワークを導入するアプローチが一般的に用いられる [9, 10, 11]。しかし、CNN に積み重ねる層数や導入するネットワーク数が増えるに連れて、ネットワークのパラメータ数は大幅に増大する。そのため、携帯機器や組み込み機器等の計算リソースの小さな機器で処理する場合、認識精度を維持しつつ、パラメータ数がより少ないネットワークが必要となる。

2つ目は、CNN を学習する際にアンバランスな学習条件に脆弱な点である。この問題は、1つのサンプルに付与されている複数タスクのラベルを用いて学習する Multi-task Learning で発生しやすい。Multi-task Learning は、サンプルに付与されている複数タスクのラベルを用いて、各タスクに応じた誤差関数を設計することでネットワークを学習する。そのため、算出される各タスクの学習誤差にばらつきが発生しやすく、ネットワークの性能低下を招きやすい。加えて、1つのサンプルに複数タスクのラベルが付与されているため、各タスクのクラスにおけるサンプル数の割合に偏りが発生しやすい。学習サンプル数の多いクラスは学習時において頻繁に CNN へ入力されるため、高い性能を得ることができる。一方、学習サンプル数の少ないクラスは、CNN に入力される回数が非常に少ないため、高い性能を得ることが難しい。このように、Multi-task Learning はアンバランスな学習条件が発生しやすいため、アンバランスな学習条件においても安定した学習が可能なアプローチが必要となる。

3つ目は、CNN は非常に高い認識精度を実現している反面、推論時における判断根拠が不明確という点である。CNN に含まれるパラメータ数は非常に膨大であり、どのような根拠で認識結果を出

力したかを解析するのが非常に困難である。これは、CNN を ADAS 等の実アプリケーションへ適用する際に重要な課題である。例として、CNN の誤認識によりユーザへ何かしらの危害が与えられた場合、どのような理由で誤認識をしたのかを解析し、ユーザへ説明する必要がある。そのため、ネットワークがどのような判断根拠で認識結果を出力したのかを、ユーザーへ伝えることが必要とされている。

1.2 研究目的

本研究では、以下の 3 つの項目について取り組む。

1. パラメータ数の少ないネットワークによる歩行者検出と標識認識の高精度化。
2. アンバランス性の解消による CNN の汎化性能向上。
3. CNN における判断根拠の理解。

1 つ目は、ADAS における組み込み機器への導入を想定し、パラメータ数の増加を抑制しつつ高精度な物体認識を実現する。この問題では、ADAS において代表的な認識タスクである、歩行者検出と標識認識を対象とする。2 つ目は、Multi-task Learning におけるアンバランス性による性能低下を解決する手法を提案する。この問題では、アンバランスな学習条件が発生しやすい歩行者属性認識と顔属性認識を対象とする。歩行者属性は、天候や視点の変化により認識する歩行者の属性が変化しやすく、特定のクラスにおけるサンプル数の割合が偏りやすい。一方、顔属性認識は回帰で扱う顔属性 (顔器官点検出、年齢推定等) と認識で扱う顔属性 (性別、人種等) が混在しやすく、学習誤差のばらつきが発生しやすい。3 つ目の深層学習における判断根拠の理解では、視覚的解析モデルを応用することで実現する。以下に、三項目における本研究の目的について述べる。

歩行者検出と標識認識の高精度化 ADAS における歩行者検出と標識認識は事故防止のために、高い精度の認識を必要とする上で、計算リソースの少ない組み込み機器による認識が必要となる。しかし、CNN を用いたアプローチは精度向上のために層を深くしたり、他の識別器と併用するアプローチが多い。これらのアプローチはネットワークの構造が複雑になる傾向がある。本研究では、1 つのネットワークで高い性能の歩行者検出と標識認識を実現するため、Dropout [12] のアルゴリズムを用いた Random Dropout と Ensemble Inference Networks を提案する。従来の Dropout は、学習時の各更新処理において一定の割合で選択したユニットの応答値を 0 にしている。提案する Random Dropout は、ランダムに決定した割合で選択したユニットの応答値を 0 にすることで Dropout よりも汎化性能を向上させる。Ensemble Inference Networks は、評価時に構造の異なる全結合層を複数生成し、各全結合層の応答値を統合することで、最終的な出力を得る。提案手法では、Random Dropout により、歩行者や標識の隠れや姿勢変動に頑健であるネットワークを構築し、Ensemble Inference Networks により誤検出と誤認識を削減ことが可能となる。

Multi-task Learning の学習におけるアンバランス性の解消 Multi-task Learning が扱うサンプルは、1 つのデータと複数タスクのラベルが付与されており、これらのラベルを用いて誤差関数を設計してネットワークを学習するため、アンバランスな学習条件が発生しやすい。本研究では、学習誤差

のばらつきの問題と学習サンプル数の割合の偏りの問題を対象とし、それぞれの問題を解決するための2つのアプローチを提案する。Multi-task Learning における学習誤差のばらつきは、重み付き誤差関数を導入する。重み付き誤差関数を導入することで、Multi-task Learning の学習を安定化させ、顔属性認識の精度を向上することが期待できる。学習サンプルの割合の偏りは、学習サンプルに対して稀少度を定義し、稀少度を用いた学習により解決する。稀少度は学習サンプルが属しているクラスのサンプル数の割合から算出する。提案手法は算出した稀少度から CNN の学習に使用するサンプルを選択することで、学習サンプル数の割合の偏りを解消しながら CNN にサンプルを入力できる。これにより、学習サンプルの少ないクラスに属している学習サンプルを適度に CNN へ入力でき、学習サンプル数の少ないクラスの認識精度を向上できる。

深層学習における判断根拠の理解 ニューラルネットワークがどのような判断根拠で認識したのかを理解するために、本研究では視覚的説明モデル [13] を用いる。しかし、従来の視覚的説明モデルは、注視した領域を示す Attention map を出力するために、ネットワークを再学習および構造を修正する必要がある。この再学習とネットワークの修正は、特定のネットワークや認識タスクにおいて性能低下を引き起こしやすい。この問題を解決するために、本研究では Attention 機構に着目する。視覚的説明により獲得した Attention map は画像認識において出力結果を求める際に着目した領域であり、Attention 機構に応用することで画像認識の性能を向上させつつ視覚的説明を同時に行うことができる。本研究では、視覚的解析の Attention map を Attention 機構に応用できる Attention Branch Network を提案する。提案する Attention Branch Network は、画像分類をはじめ、歩行者検出や顔属性認識、自動車制御等の様々な画像認識タスクへ応用できる。

さらに、本アプローチを深層強化学習へ応用する。深層強化学習は獲得した報酬を用いてネットワークを学習するため、教師あり学習でネットワークを学習する Attention Branch Network を直接応用することができない。そこで、強化学習法の一つである Actor-Critic 法のネットワーク構造に着目し、Attention Branch Network を導入する。Actor-Critic 法は、1つのネットワークから方策と状態価値を出力する構造である。そのため、方策と行動価値をそれぞれ出力するブランチを構築することで、Attention Branch Network を容易に Actor-Critic 法へ応用できる。Atari ゲームによる実験では、Actor-Critic 法における判断根拠を、提案手法から出力される Attention map から解析できることを示す。また、Attention Branch Network をベースにした Actor-Critic 法の導入により、特定の Atari ゲームにおけるエージェントの制御の精度を向上できる。

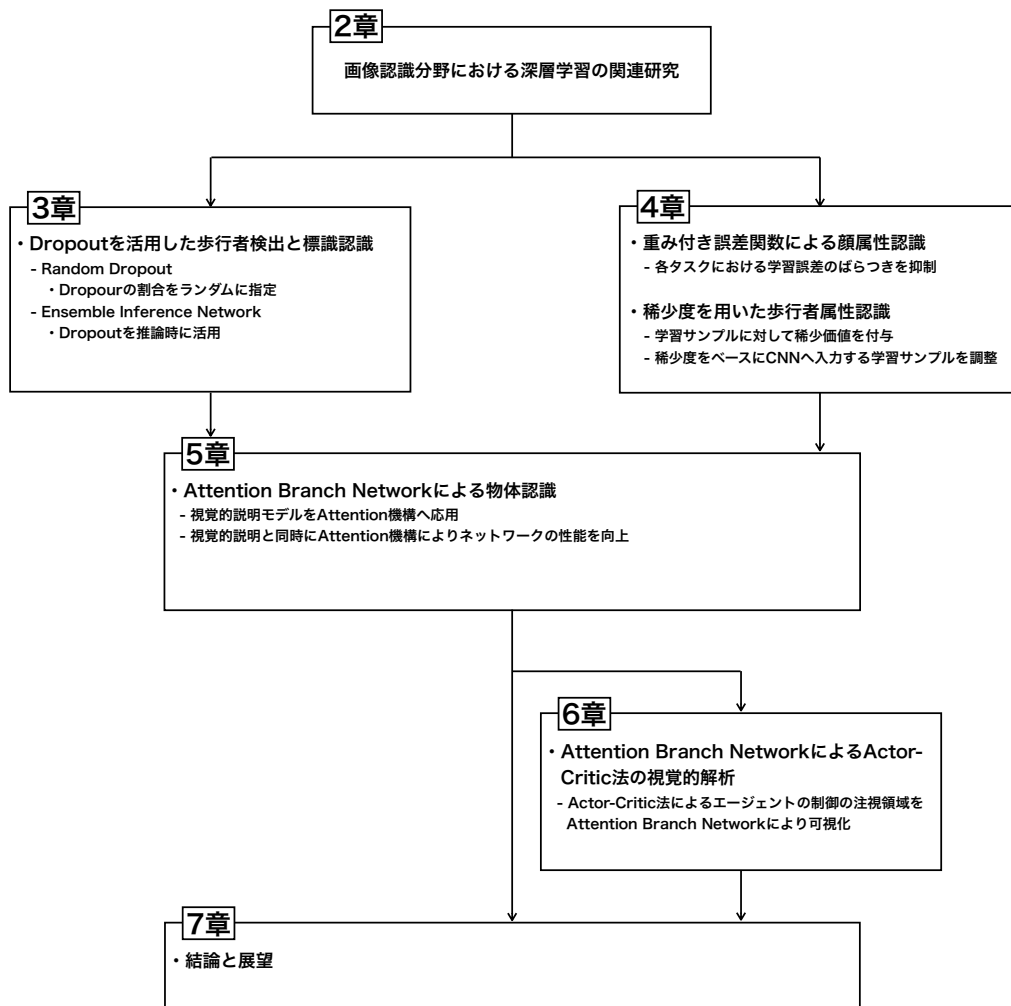


図 1.1: 本論文の構成.

1.3 本論文の構成

本論文は、図 1.1 のような章構成となっている。2 章では、本研究のベースとなる CNN の関連研究についてまとめる。3 章では、Random Dropout と Ensemble Inference Networks による歩行者検出と標識認識について提案する。4 章では、マルチタスク学習を対象に、学習サンプル数の偏りと学習誤差のばらつきを解消する手法についてそれぞれ提案する。5 章では、新たな視覚的解析モデルである Attention Branch Network の提案し、3 章と 4 章で取り組んだ内容に対して判断根拠の解析を行う。そして、6 章では 5 章で提案した Attention Branch Network を深層強化学習へ応用する。7 章では、本論文の結論と展望について述べる。

第2章

画像認識分野における深層学習の関連研究

Deep Convolutional Neural Network (CNN) は、画像分類 [6, 10] をはじめとした物体検出 [14] や属性認識 [15]、セグメンテーション [16, 17] 等、様々なベンチマークにおいて高い性能を達成している。画像認識タスクは、大まかに図 2.1 のような 5 つのタスクに分類できる。画像分類では、入力画像に対して何の物体が写っているのかを認識する技術である。物体検出は、入力画像中に存在する物体の位置とカテゴリを推定する技術である。属性認識は、特定の物体に対する属性 (対象が人の場合は性別等) を認識する技術である。セマンティックセグメンテーションは、画像中の物体位置と形状を 1 ピクセル単位で認識する技術である。インスタンスセグメンテーションは、画像中の物体位置と形状を、1 ピクセル単位かつ個別の物体ごとに認識する技術である。

本章では、本稿で最も関わりの深い画像分類と物体検出、属性認識の 3 つの認識タスクにおける関連研究を述べる。3 章では画像分類と物体検出が関係しており、4 章では属性認識が関係している。5 章は、画像分類と物体検出、属性認識の 3 つのタスクが関係している。本章の構成は、はじめに CNN の大まかな概要について述べる。CNN の概要を述べた後に、CNN の汎化性能を向上させるための技術について述べる。そして、画像分類で用いられる技術について述べた後、画像分類の応用である物体検出と Multi-task Learning について述べる。

2.1 Deep Convolutional Neural Network

CNN [5, 6] は、多層パーセプトロン的一种であり、畳み込み層とプーリング層を主体に構築されたニューラルネットワークである。CNN の構造は、Hubel と Wiesel の研究 [18] にルーツがある。Hubel らは、猫の初期視覚野に特定の傾きを持つ線分に反応する単純細胞と、特定の傾きを持つ線分を移動させても反応する複雑細胞の存在を確認した。その後、Fukushima らが数理モデルとなる Neocognitron という視覚パターン認識に関する階層型神経回路モデルを提案した [19]。Neocognitron は、同一の結合重みを持つユニットを複数並べ、上位層でユニットの出力を集積するプーリングを配置することで、幾何学的変化に対する不変性を実現している。CNN は、Neocognitron に誤差逆伝播法を導入したモデルである。

CNN の学習は、教師付き学習を前提としており、誤差逆伝播法 [20] を用いた勾配降下最適化法で学習する。CNN の構造を、図 2.2(a) に示す。CNN は複数の畳み込み層と全結合層を積み重ねた後に、

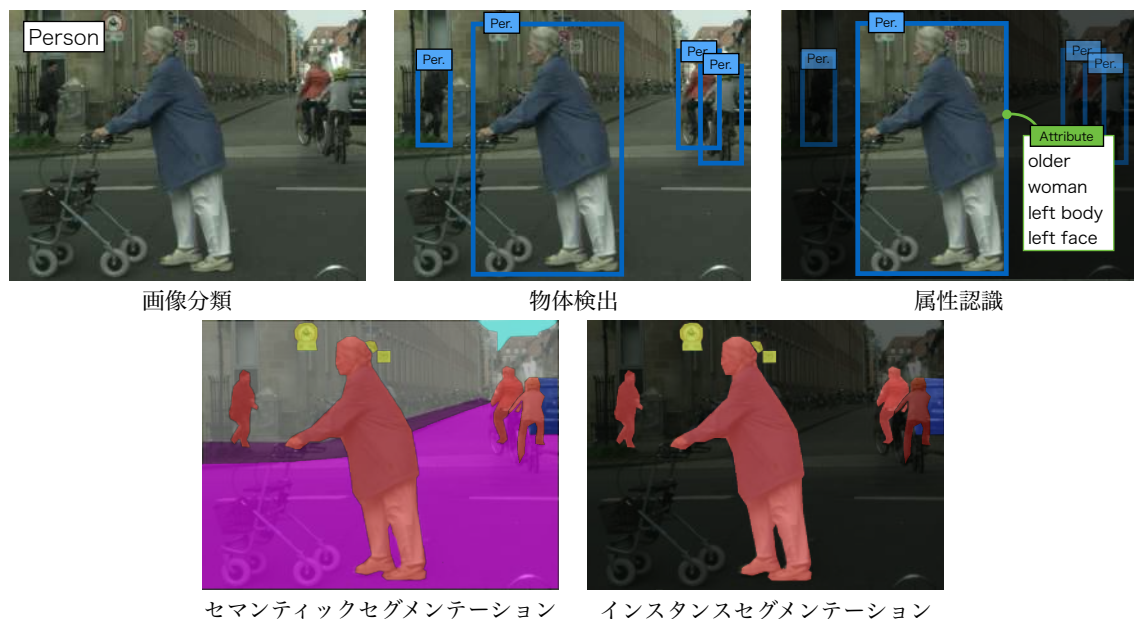


図 2.1: 画像認識タスクの種類

全結合層を配置した構造が一般的なモデルである。CNN の入力は、一般的には画像が入力される。画像が入力された後、図 2.2(b) のように畳み込み層で複数のカーネルを畳み込み、複数のマップ状の出力値が得られる。このマップ状の出力値は特徴マップと呼ばれる。プーリング層では、図 2.2(c) のように特徴マップの局所領域から 1 つの値を出力する。CNN では、繰り返し畳み込み層とプーリング層を配置することで、認識に有効な特徴を獲得できる。得られた特徴マップは、全結合層へと入力され、画像分類では各クラスにおける確率値を出力する。

2.1.1 画像分類における CNN のモデル

CNN ベースの物体検出法や Multi-task Learning, セマンティックセグメンテーション, インスタンスセグメンテーションで用いられるモデルは、画像分類で提案されたモデルをベースにネットワークを構築している例が多い [14, 21, 22, 23, 24, 17]。CNN ベースの画像分類モデルは、様々なモデルが提案されている。CNN の初代の代表的なモデルとして、AlexNet [6] がある。AlexNet は、大規模な画像認識コンペティションである ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [25] で初めて CNN のモデルでトップとなった手法である。AlexNet の構造は、5 層の畳み込み層と 2 層の全結合層から構築された CNN である。AlexNet の登場により、従来の Fisher vector と SVM をベースとした手法 [26] を大きく上回る性能を実現している。AlexNet の提案後、AlexNet をベースとした ZFNet [27] が提案された。その後、より深いネットワークを構築することで、高精度な画像分類を実現するアプローチが提案された。深いネットワークは、複雑な特徴を学習により抽出しやすくなるため、高い精度で認識ができる。初期におけるこのアプローチの代表的なモデルとして、VGGNet [28]

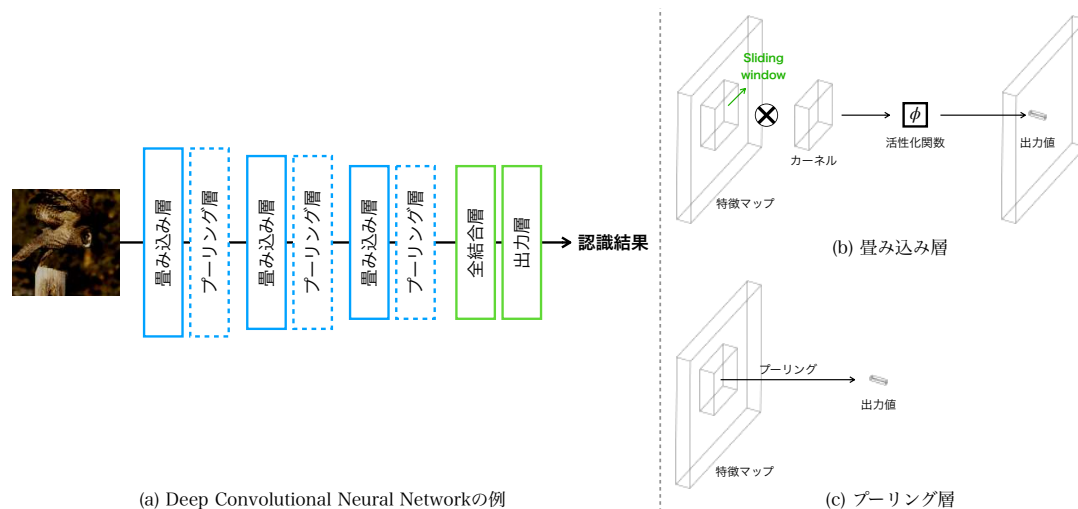


図 2.2: CNN の処理の流れ

や GoogLeNet [29] が挙げられる。VGGNet は最大で 19 層の CNN を構築し、GoogLeNet は最大で 22 層の CNN を構築している。この 2 つの CNN は、2015 年度の ILSVRC において上位の性能を達成している。その後、Batch Normalization [30] や He の正規化 [31] 等、深いネットワークの学習を安定させるためのアプローチが提案された。2016 年には、He らが Residual Learning を導入した ResNet を提案し、100 層以上の CNN を安定して学習できるようになり、CNN ベースの画像分類の精度がさらに向上した [10]。ResNet が提案された後は、ResNet をベースにした様々なモデルが提案されている [32, 33, 34]。これらの画像分類モデルの進化は、物体検出や属性認識、セグメンテーションにも大きな影響を与えている。物体検出やセグメンテーションで用いられるネットワークは画像分類のネットワークモデルをベースに構築するため、画像分類の性能向上に伴い他の認識タスクの性能も向上できる。そのため、画像分類は他の画像認識タスクにおいても重要な立ち位置に属している。

画像分類モデルの進化の背景には、CNN の学習に使用するデータセットにも大きく関係している。画像分類の評価で一般的に用いられる代表的なデータセットは、MNIST や図 2.3(a) と図 2.3(b) の CIFAR10/100、図 2.3(c) の Street View House Number (SVHN)、図 2.3(d) の ImageNet が挙げられる。MNIST と CIFAR10/100 Dataset の学習サンプル数は 50,000 枚であり、SVHN (train : 73,257+extra. : 531,131) や ImageNet (1,281,167 枚) と比べてサンプル数は少ない。CNN は大量の学習サンプルを用いて学習することで、ネットワークの汎化性能を向上できる。また、ImageNet 等の大量に学習したネットワークのパラメータを初期値とし、他の学習へ転移することも可能である。本章では、画像分類で用いられるモデルについて調査し、まとめる。画像分類モデルについて述べた後に、ネットワークの汎化性能を向上させる手法について述べる。

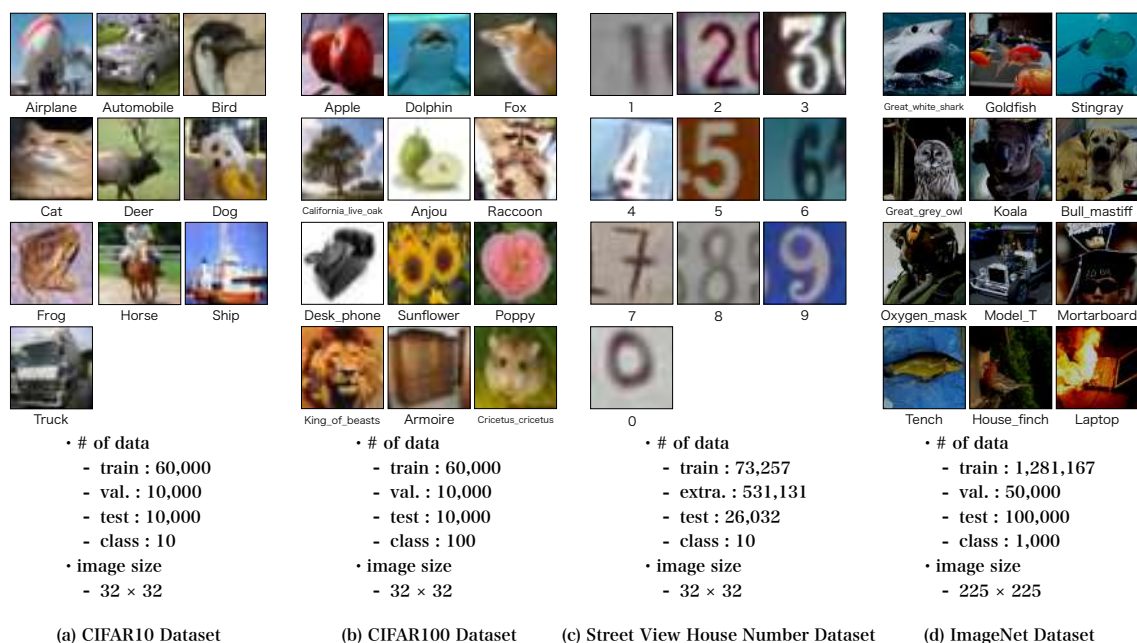


図 2.3: 画像分類におけるデータセット

AlexNet [6] AlexNet は、5 層の畳み込み層と 2 層の全結合層から構築されており、Local Response Normalization (LRN) や Dropout, Rectified Linear Units (ReLU) 等の大量のデータで深いネットワークを学習するためのテクニックを導入することで汎化性能を向上している。特に ReLU は、従来のシグモイド関数や tanh 関数とは異なり学習時の勾配消失問題等を防ぐことができる活性化関数である。そのため、ReLU は様々な派生手法が提案されており、現在のモデルにおいても幅広く用いられている。

Network In Network [35] Network In Network (NIN) は、Multi Layer Perception Convolution (MLP-Conv) と Global Average Pooling (GAP) または Global Max Pooling (GMP) を取り入れたネットワークである。MLP Conv は、CNN の活性化関数の代わりに導入され、畳み込みで得られた応答値を多層パーセプトロンに入力することで活性化関数も学習する。GAP は、特定の局所領域を平均する平均プーリングを、特定の 1 枚の特徴マップ全体に施す処理である。NIN では、出力層でクラス数枚の特徴マップを出力し、GAP と softmax 関数を施すことで各クラスにおける確率を出力している。また、GMP は平均プーリングではなく、最大プーリングを用いている。GAP は、ネットワークを畳み込み層のみで構築できるため、パラメータ数を大幅に削減することができる。

VGGNet [28] VGGNet は、 3×3 のカーネルを持つ畳み込み層を主体に構築されたネットワークである。AlexNet では、 5×5 のカーネルを持つ畳み込み層を主体にネットワークを構築している。VGGNet は、 5×5 のカーネルの受容野が、2 枚分の 3×3 のカーネルの受容野と同等である点に着目し、より深い CNN を構築している。この 3×3 のカーネルを持つ畳み込み層をベースに構築したネットワークは VGGNet が提案された後、ResNet をはじめとした様々なネットワークにおいて一般的な畳み込み層のパラメータとして使われている。

$\mathbf{X}$: 入力 : 畳み込み層 : 全結合層

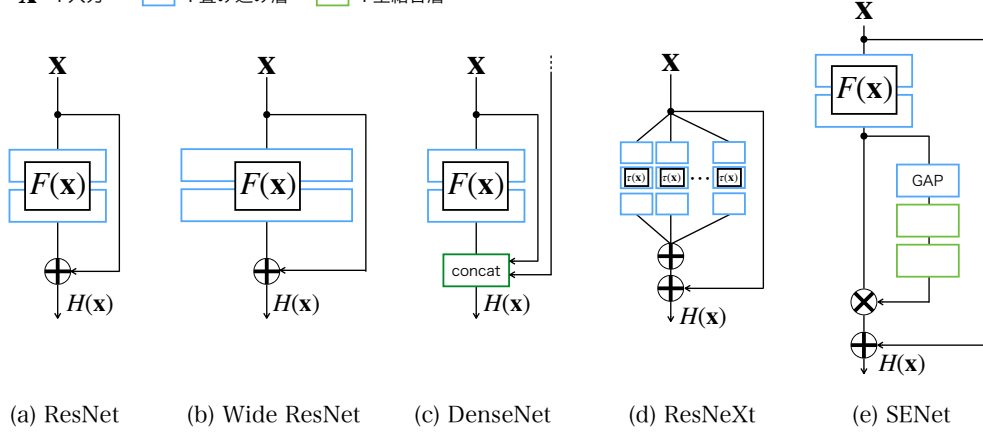


図 2.4: ResNet の派生系ネットワークの構造

ResNet [10] ResNet は, Residual Learning を取り入れたネットワークであり, 100 層以上のネットワークを学習可能にしている. Residual Learning はバイパス構造を取り入れており, 図 2.4(a) のような Residual Learning を取り入れた Residual unit を構築している. 入力 $\mathbf{x}$ に対して複数の畳み込み層から得られる出力を $F(\mathbf{x})$ と定義した際に, Residual unit の出力は $H(\mathbf{x}) = F(\mathbf{x}) + \mathbf{x}$ となる. ResNet の場合, 最適な $H(\mathbf{x})$ を学習するのではなく, 最適な $F(\mathbf{x})$ を学習する. この構造は, 入力 $\mathbf{x}$ が最適な特徴の場合に $F(\mathbf{x})$ が 0 となるまた, $\mathbf{x}$ と $H(\mathbf{x})$ の特徴が似ている場合に $\mathbf{x}$ と $H(\mathbf{x})$ の差分を学習しやすくなる. 逆伝播の際も, 入力 $\mathbf{x}$ のバイパスにより勾配の消失が発生しない構造となっている. これらの工夫により, 非常に深いネットワークを構築することができ, 当時最も深い GoogLeNet (22 層) の 5 倍以上の深さを持つネットワークを実現した. そして, 2016 年度の ILSVRC においてトップの性能を実現している.

Wide ResNet [32] ResNet は層を深く構築することで画像分類の性能を向上されている. Wide Residual Network (WideResNet) では, 図 2.4(b) のように畳み込み層のチャンネル数を増幅することで画像分類の精度を向上させている. WideResNet は, 約 1,000 層の ResNet に対して 16 層の WideResNet を構築し, 同等の精度とパラメータ数を実現している. また, Residual unit に Dropout を導入することで, 汎化性能をより向上させている.

DenseNet [33] DenseNet は, ResNet をベースに構築した Dense block を導入し, 性能向上と計算コスト削減を可能にしたネットワークである. Residual block では, 図 2.4(a) のような入力 $\mathbf{x}$ と畳み込み層の出力 $F(\mathbf{x})$ を加算することで出力 $H(\mathbf{x})$ を求めている. Dense block は, 図 2.4(c) のように入力 $\mathbf{x}$ と畳み込み層を施した $F(\mathbf{x})$ を concatenation することで出力 $H(\mathbf{x})$ を得る. また, もう一つ異なる点として, Residual block ではバイパスを入力の特徴マップ $\mathbf{x}$ のみで使用していたが, Dense block では様々な Dense block の出力値をバイパスとして使用する. Dense block の出力 $H(\mathbf{x})$ は, concatenation によるチャンネル数の増加を抑制するために 3×3 と 1×1 のカーネルを持つ畳み込み層を組み合わせ, 計算効率を向上させている.

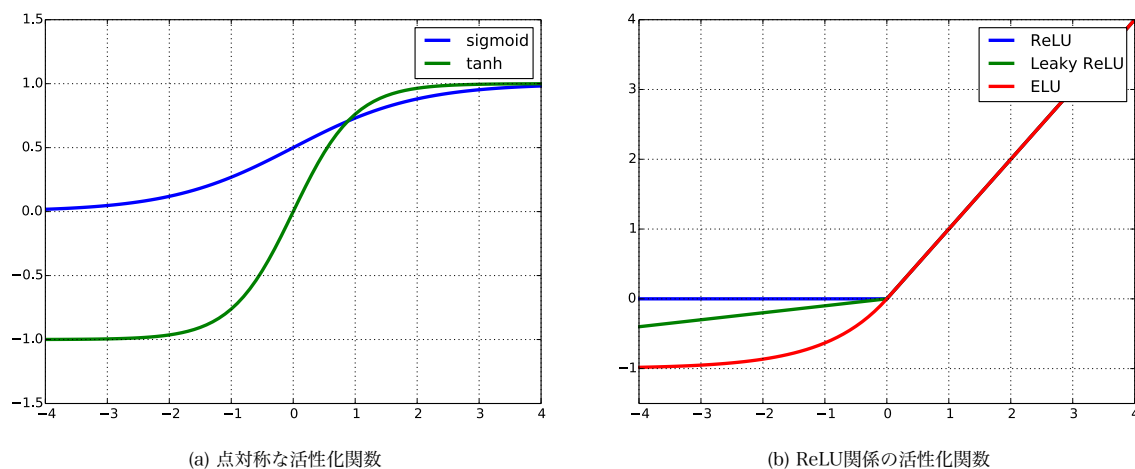


図 2.5: 活性化関数の例

ResNeXt [34] ResNeXt は、図 2.4(d) のように $F(\mathbf{x})$ を複数の変換を与える畳み込み層 $\tau(\mathbf{x})$ で構築することでブロックを構築する。ResNeXt の構造は、入力に対してチャンネル数の少ない畳み込みモジュールを複数用意し、それぞれの畳み込み層 $F(\mathbf{x})$ に入力する。そして、それぞれの畳み込み層 $F(\mathbf{x})$ で得られた特徴マップは加算 (もしくは concatenation) により統合される。ResNeXt は、チャンネル数の少ない畳み込みモジュールを複数用意するため、チャンネル方向のフィルタが従来の ResNet よりも少ない。そのため、パラメータ数の削減を実現しつつ、ネットワークの汎化性能を向上させることができる。

Squeeze-and-Excitation Network [36] Squeeze-and-Excitation Network (SENet) は、特徴マップのチャンネルに対して Attention 機構を導入したネットワークである。SENet の Attention 機構の重みは、図 2.4(e) のように獲得した特徴マップに対して GAP を施し、チャンネル数分のユニットを持つ全結合層を施すことで算出する。特徴マップのチャンネルに対して Attention 機構を施すことで、認識に有効なチャンネルを重み付けできる。さらに、SENet は VGGNet や ResNet 等の様々なネットワークに導入でき、認識精度を改善できる。2017 年の ILSVRC において、SENet はトップの画像分類精度を達成している。

2.1.2 CNN の汎化性能を向上させる手法

大量のデータを用いて深いネットワークを安定して学習するために、様々なテクニックが導入されている。これらのテクニックは、主に活性化関数や初期値、正規化、最適化方法等である。これらのテクニックを導入することで、学習の収束を早めたり、学習時における勾配の発散等を防ぐことができる。以下に、それぞれのテクニックについて述べる。

■ 活性化関数

CNN で最も一般的に用いられる活性化関数として、Rectified Linear Unit (ReLU) がある。ReLU は、AlexNet が提案された際に ImageNet Dataset を効率的に学習するために導入された活性化関数であり、式 (2.1) のように 0 以上の入力値を線形に出力する恒等関数である。ReLU が提案される前のニューラルネットワークの活性化関数では、図 2.5(a) のシグモイド関数や tanh 関数等の点対称な活性化関数が用いられてきた。しかし、点対称な活性化関数は入力値 x が大きな値であるほど勾配が消失しやすく、学習が困難になる。一方で、ReLU は入力値 x が 0 以上の場合に微分値が一定で 1 となるため、勾配が消失しないというメリットがある。また、0 以下の入力値 x を 0 にすることができるため、負の内積値や相関を取り除く効果もある。加えて、Attention 機構や視覚的説明モデルにおける注視領域を獲得する際に、非注視領域の値を取り除くためにも用いられる [13, 37, 36]。

$$\phi(x) = \max(0, x) \quad (2.1)$$

ReLU から派生された活性化関数は、図 2.5(b) のようにいくつか提案されている [38, 39]。これらの手法は、ReLU が 0 以上の正の内積値のみを出力している点に着目しており、負の内積値を考慮することでネットワークの性能を向上させている。一方で、ReLU をより一般化した Maxout という活性化関数も提案されている [40]。Maxout の本来の目的は、後述する Dropout の性能を引き上げることであるが、関数の近似能力は ReLU よりも高く、ネットワークの性能をより向上させやすい。Maxout で最大値を取るユニット数を q とした時、式 (2.2) のように表現できる。

$$\phi(x) = \max_{j \in [1, q]} \mathbf{x}_j \quad (2.2)$$

■ 初期値

ネットワークのパラメータの初期値は、認識性能を左右する要因の一つであり、適切に初期値を設定することで高い精度を発揮しやすくなる。パラメータの初期値は基本的に乱数で決定するが、特定の範囲から乱数を生成して初期値を決定する。AlexNet や VGGNet が提案された当時は、式 (2.3) のようにパラメータの初期値を中間層のユニット数から乱数の範囲を決定する Glorot Uniform (Xavier の初期化) [41] が用いられていた。ここで、 U は一様分布を示しており、 l が層のインデックス、 n がユニット数を示している。しかしながら、Xavier の初期化を CNN へ応用する際に 2 つの問題が発生する。1 つ目は、活性化関数として tanh 関数等の点対称な関数を用いることを前提としており、ReLU のような活性化関数を用いることを考慮していない。2 つ目は、全結合層のネットワークを対象とした初期値の決定方法であり、局所的な結合で構築されている畳み込み層を考慮していない。

$$\mathbf{w} \sim U \left(-\sqrt{\frac{6}{n^l + n^{l+1}}}, \sqrt{\frac{6}{n^l + n^{l+1}}} \right) \quad (2.3)$$

この問題を解決するために、He らは Xavier の初期化をベースに式 (2.4) のような初期化方法を提案している [31]。ここで、 N は正規分布を示しており、He らが提案した初期化方法を“He の初期化”と呼ぶ。He の初期化は、畳み込み層と ReLU を考慮することで、より深いネットワークを構築しても安定した学習が可能となった。Xavier の初期化では約 20 層のネットワークが学習の限界であったが、He の初期化により、直列に積み重ねられた 30 層の CNN でも学習が可能となった。

$$\mathbf{w} \sim N\left(0, \sqrt{\frac{2}{n^l}}\right) \quad (2.4)$$

■ 正規化

CNN はカーネルと入力画像または特徴マップの局所領域から内積値を求め、活性化関数を施すことで新たな特徴マップを出力する。しかし、場合によってはノイズやアウトライヤーなデータが原因で応答値のばらつきが発生し、認識性能を低下させる原因となる。この問題を解決するために、一般的には畳み込み層で得られる特徴マップや全結合層の特徴ベクトル、または活性化関数を施した応答値に対して正規化する。AlexNet では、Local Response Normalization (LRN) [6] が用いられている。LRN は、特定の特徴マップの位置を正規化する際に、近傍の位置の特徴マップを用いることで正規化を行う。LRN は AlexNet に対しては精度が向上するが、他の特定のネットワークに対しては精度が向上しなかったり、Batch Normalization [30] が提案されたことで、現在は一般的に用いられていない。

現在の CNN の正規化では、Batch Normalization が一般的な正規化方法として使われている。Batch Normalization は、式 (2.5) のように特定のチャンネルをミニバッチ単位で正規化し、平均を 0 と分散を 1 にする。ここで、 μ と σ はそれぞれ式 (2.6) と式 (2.7) のように求める。また、ミニバッチ単位で特定のチャンネルを正規化することで、内部共変量シフトが大幅に変動するのを防いでいる。Batch Normalization は、ResNet 等の様々なネットワークに導入されている。

$$\mathbf{x}' = \frac{\mathbf{x} - \mu}{\sigma} \quad (2.5)$$

$$\mu = \frac{1}{M} \sum_i \mathbf{x}_i \quad (2.6)$$

$$\sigma = \sqrt{\delta + \frac{1}{M} \sum_i (\mathbf{x} - \mu)_i^2} \quad (2.7)$$

しかし、Batch Normalization は内部共変量シフトを獲得するために、ミニバッチのサイズを 16 以上にする必要がある。この問題を解決するために、図 2.6 のような Layer Normalization [42] や Instance Normalization [43]、Group Normalization [1] が提案されている。これらの正規化の違いは、主に正規化する領域 $\mathbf{x}$ が図 2.6 のように異なる。Layer Normalization は特定のサンプルの特徴マップから全てのチャンネルを正規化する。Instance Normalization は特定のサンプルの特定のチャンネルの特徴マッ

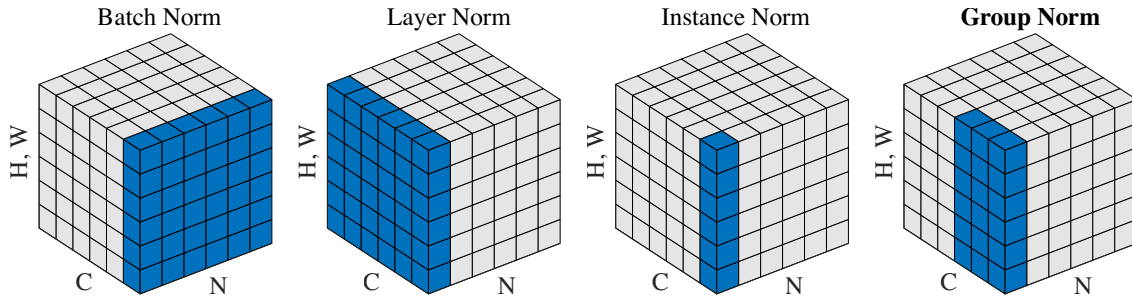


図 2.6: Batch Normalization の発展系の手法 (文献 [1] から引用)

プのみ正規化する． Group Normalization は， Layer Normalization と Instance Normalization の中間に位置しており，数枚のチャンネルのみ用いて正規化する．特に， Group Normalization はミニバッチサイズに依存しないため，少量のミニバッチサイズでも従来の Batch Normalization と同等の精度を得ることができる．

■ 学習方法

確率的勾配降下法 (stochastic gradient descent; SGD) によりネットワークを学習する場合，学習率 η を学習の過程で変化させる Learning rate schedule が一般的に導入されている．ここで，SGD は特定の更新回数 t における誤差の勾配 $\nabla E(\mathbf{w}^t)$ から，ネットワークの重み $\mathbf{w}^t$ を式 (2.8) により更新する．そして，式 (2.8) における $\alpha \Delta \mathbf{w}^t$ は， $\mathbf{w}^t$ の差分を追加する Momentum 項である． Momentum 項には，勾配の振動を抑制する効果があり，学習を安定化できる．

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \nabla E(\mathbf{w}^t) + \alpha \Delta \mathbf{w}^t \quad (2.8)$$

ネットワークの学習における学習率 η は，パラメータの更新量を制御する係数である．学習率 η を大きく設定した場合は学習の収束が早くなるが，勾配が発散しやすい．一方で，学習率 η を小さく設定した場合は最適解を獲得しやすいが，学習の収束が遅くなる．この問題を解決するために，学習の過程で学習率 η を変更することで，最適解を獲得しやすくしている． Learning rate schedule は，一般的には学習率 η を減衰させる Learning rate drop が一般的に用いられる． SGD で Learning rate drop を用いる場合，図 2.7(a) のように指定した更新回数に達した際に学習率 η を定数倍する．学習率 η を特定の更新回数で下げることで，図 2.7(a) のように算出される学習誤差をより下げることができる．学習誤差が下がった際の精度を確認した時，図 2.7(b) の Top-1 error が約 10% 削減できている．そのため，SGD における Learning rate drop はネットワークの学習において重要なテクニックとされている．

一方で，図 2.7 のように学習率 η を下げるタイミングを手動で決定するのではなく，学習の過程で自動に決定する方法も提案されている [44, 45]． Adaptive Gradient (AdaGrad) [46] は，ネットワーク

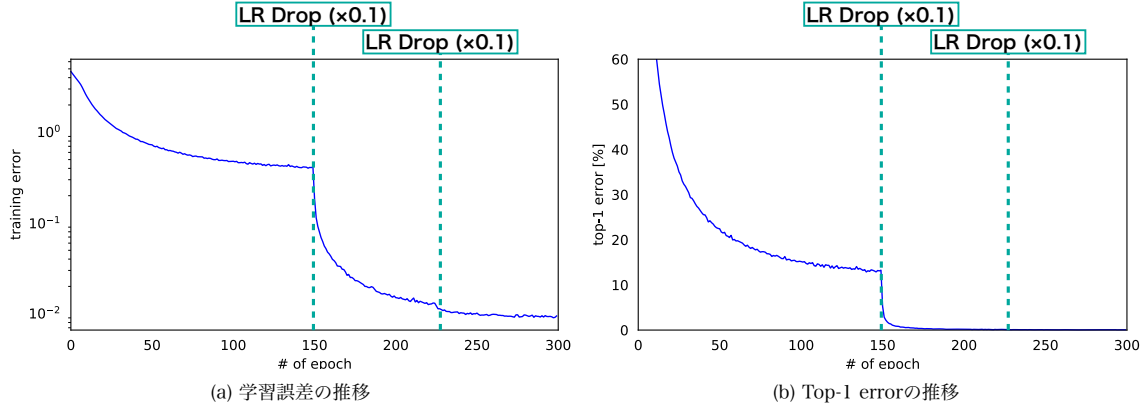


図 2.7: Learning rate drop の例

の各パラメータに対して学習率を設計し、自動で調整しながら学習できる。AdaGrad は、式 (2.9) から式 (2.11) によりネットワークを更新する。ここで、 h_t は t 回目の更新における勾配の二乗和を示している。 h_t を用いることで、 t 回目の学習率 η_t を算出する。

$$h_t = h_{t-1} + \nabla E(\mathbf{w}^t)^2 \quad (2.9)$$

$$\eta_t = \frac{\eta_0}{\sqrt{h_t}} \quad (2.10)$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t \nabla E(\mathbf{w}^t) \quad (2.11)$$

AdaGrad をベースとした学習法は、数多く提案される [45, 44, 47]。代表的な学習法として、Adaptive Moment Estimation (Adam) [47] がある。Adam は、勾配の 1 次 (平均) と 2 次 (分散) のモーメント推定を行うことで、ネットワークのパラメータを更新する。それぞれのモーメントは、式 (2.12) と式 (2.13) のように指数移動平均という指数関数的に重みを減少させる方法により算出する。算出した一次のモーメント m_t と二次のモーメント v_t は、式 (2.14) と式 (2.15) により発生するバイアスを補正する。そして、補正した 2 つのモーメント $\hat{m}_t$ と $\hat{v}_t$ を用いて式 (2.16) のようにパラメータを更新する。

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla E(\mathbf{w}^t) \quad (2.12)$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2) \nabla E(\mathbf{w}^t)^2 \quad (2.13)$$

$$\hat{m} = \frac{m_{t+1}}{1 - \beta_1^t} \quad (2.14)$$

$$\hat{v} = \frac{v_{t+1}}{1 - \beta_2^t} \quad (2.15)$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \frac{\hat{m}}{\sqrt{\hat{v} + \epsilon}} \quad (2.16)$$

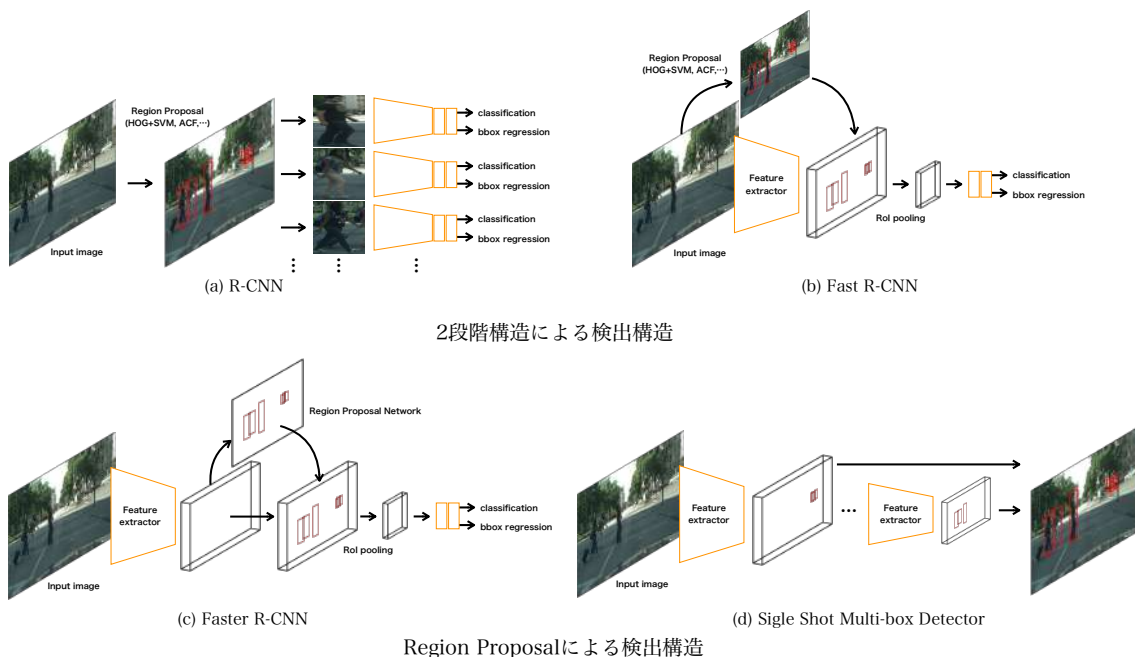


図 2.8: CNN ベースの物体検出法

2.2 物体検出

CNN ベースの画像分類法の発展に伴い、CNN をベースとした物体検出法も大きく発展している。物体検出は、図 2.1(b) のように画像中の物体の位置とそのカテゴリを推定する技術である。CNN ベースの物体検出法は、R-CNN [48] をベースに進化を遂げている。R-CNN が提案された後、Fast R-CNN [49]、Faster R-CNN [14] が提案され、物体検出の精度と計算コストが大幅に改善された。また、R-CNN のような物体候補領域を検出せずに、直接物体領域を検出できる Single Shot Multi-box Detector (SSD) [21] も提案されている。本章では、物体検出法として一般的に用いられる R-CNN と Fast R-CNN、Faster R-CNN、SSD について説明する。

R-CNN [48] R-CNN は、Girshick らが提案した初期の物体検出法であり、Fast R-CNN や Faster R-CNN の基礎となる手法である。R-CNN では、図 2.8(a) のように物体の候補領域を検出し、検出した候補領域を CNN で認識する手法である。R-CNN が提案された当時は、前段の識別器に Selective search [50] を使用し、後段の物体識別には AlexNet を使用する。後段の AlexNet は、候補領域に対するカテゴリ (背景を含む) に加え、バウンディングボックスの補正値を回帰により推定する。ここで、Girshick らは後段の識別器に AlexNet をそのまま配置するのではなく、AlexNet から獲得できる特徴ベクトルを SVM に入力して学習している。これにより、物体検出の精度を向上している。

Fast R-CNN [49] R-CNN の欠点は、検出した物体候補領域の数だけ繰り返し CNN による識別処理を行う点である。そのため、検出される物体候補領域の数だけ計算コストが増幅する。Fast R-CNN では、図 2.8(b) のように入力画像全体の特徴マップをあらかじめ算出し、検出領域の特徴マップのみ

切り出して後段の小さなネットワークで各検出領域を認識する。検出した物体候補領域を後段のネットワークに入力する際は、特徴マップを一定のサイズへリサイズする **Region of Interest pooling (RoI pooling)** を施す。これらの処理により、後段の CNN による識別を繰り返す必要がなくなり、高速に物体検出できる。

Faster R-CNN [14] Fast R-CNN の提案により物体検出の計算コストが削減されたが、物体候補領域の検出は別の識別器で行う必要がある。R-CNN と Fast R-CNN で用いられる **Selective search** は計算コストが高いため、リアルタイムな物体検出を困難にする。加えて、物体候補領域のみ検出するため、検出精度が前段の識別器に依存する。この問題を解決するために、Faster R-CNN では Fast R-CNN に対して **Region Proposal Network (RPN)** を図 2.8(c) のように導入している。RPN は、抽出した特徴マップを用いて物体候補領域を検出するネットワークである。RPN による物体候補領域の検出では、アンカーという複数の形状を持つ検出枠を用いる。特徴マップ上のアンカー領域は RPN に入力され、物体候補領域の中心座標とバウンディングボックスの幅を出力する。アンカーは特徴マップ上をスライディングウィンドウし、検出処理を行う。検出した物体候補領域は **RoI pooling** により特徴マップを切り出し、Fast R-CNN に入力されてカテゴリとバウンディングボックスの修正値を推定する。Faster R-CNN は、物体検出の一連の処理を **End-to-End** に学習できるため、高精度かつ高速に物体検出できる。

Single Shot Multi-box Detector [21] SSD は、R-CNN ベースの物体検出法とは異なり、物体候補領域の検出を行わずに検出結果を得ることができる。SSD の構造は、図 2.8(d) のように CNN の順伝播の過程で物体領域を検出する構造となっている。そして、各階層で検出する物体を物体のスケールにより決定している。入力層に近い検出層ではスケールの小さな物体を検出し、出力層に近い検出層ではスケールの大きな物体を検出する。SSD は、物体候補領域を検出する過程を必要としないため、高速に物体検出できる。

2.3 歩行者検出

歩行者検出は物体検出の中でも困難なタスクであり、広大な自動車の走行シーンから歩行者を検出する必要がある。歩行者検出では、歩行者の向きや姿勢の変化、服装、スケールに頑健な特徴量を設計する必要がある。2012 年までの歩行者検出は、**Histogram of Oriented Gradient (HOG)** 特徴量 [8] が用いられており、歩行者の勾配方向ヒストグラムを特徴量として扱うことで、歩行者の向きや姿勢、服装などの見えの変化に頑健な検出を実現している。HOG 特徴量の代表的な発展手法として、**Latent SVM** をベースにパーツベースの歩行者検出を可能にした **Deformable Part Models (DPM)** [51] や、2 つの局所領域から得られる色ヒストグラムの類似度を算出する **Color Self Similarity HOG** [52] が挙げられる。HOG 特徴量が一般的に用いられた後にも、色や勾配等の様々な特徴量を画像のチャンネルのように組み合わせたチャンネル特徴量を用いて歩行者検出をする **Integral Channeled Feature (ICF)** [2] が提案されている。ICF は、抽出したチャンネル特徴量を **Boosted tree** に入力する。チャンネル特徴は勾配や色、エッジ等の様々な特徴を抽出し、**Boosted tree** で主要な特徴量を取捨選択できるため、HOG 特徴量とは異なり、勾配以外の色やエッジを効率的に用いることで高精度に歩行者を検出できる。ま

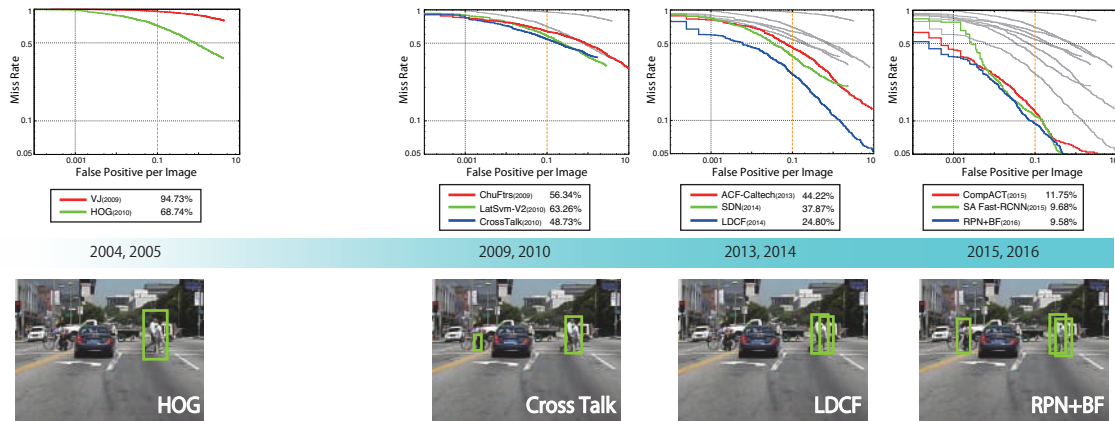


図 2.9: 歩行者検出性能の遷移

た，HOG 特徴量の正規化処理やヒストグラム化が必要ないため計算効率が非常に良く，高速に処理できる．これらの理由から，ICF をベースにした歩行者検出法も数多く提案されている [53, 54, 55]．

2012 年に開催された，ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [25] の一般物体認識コンテストにおいて Deep Convolutional Neural Network (CNN) を用いた方法がトップになって以降 [6]，CNN を歩行者検出に応用した手法が数多く提案されている [56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]．CNN をベースにした歩行者検出法は，CNN ベースの物体検出法と大きな関わりを持っている．図 2.10 に，CNN ベースの物体検出法と歩行者検出法の関係をまとめたマップを示す．Girshick らにより R-CNN を提案された後，Hosang らが歩行者検出に有効な R-CNN の設計を提案したことで，CNN ベースの歩行者検出法の性能が大幅に向上した [59]．同様に，Fast R-CNN と Faster R-CNN が提案された後，Zhang らが歩行者検出に有効な Faster R-CNN の設計を提案し，歩行者検出の性能を大幅に向上させた [69]．これらの歩行者を検出するためのノウハウを用いた手法が数多く提案され，歩行者検出の精度が大幅に向上した．加えて，2018 年の歩行者検出の研究では物体検出法の Faster R-CNN をベースとした手法が根強く提案されており，セマンティックセグメンテーションや深度情報等を取り入れた手法や，Generative Adversarial Network (GAN) [70]，Attention 機構を取り入れた手法等，様々なアプローチの手法が提案されている．

2.3.1 歩行者検出法の歴史

歩行者検出の初期の研究では，切り出した歩行者画像と背景画像の認識問題として扱われ，Wavelet 特徴量と SVM の組み合わせにより歩行者認識を行っていた [71]．そして，1 枚のフレームから歩行者の位置と大きさを推定する検出問題へ遷移し，2004 年に提案された Viola らの顔検出の研究が歩行者検出に応用された [72]．しかし，画像の濃淡のみ用いる Haar-like 特徴量は，顔検出の分野では高い性能を実現しているが，歩行者検出では十分な性能を得られなかった．そこで，Dalal らは画像の濃淡ではなく歩行者の勾配に着目して特徴量とする HOG 特徴量を提案し，Support Vector Machine (SVM)

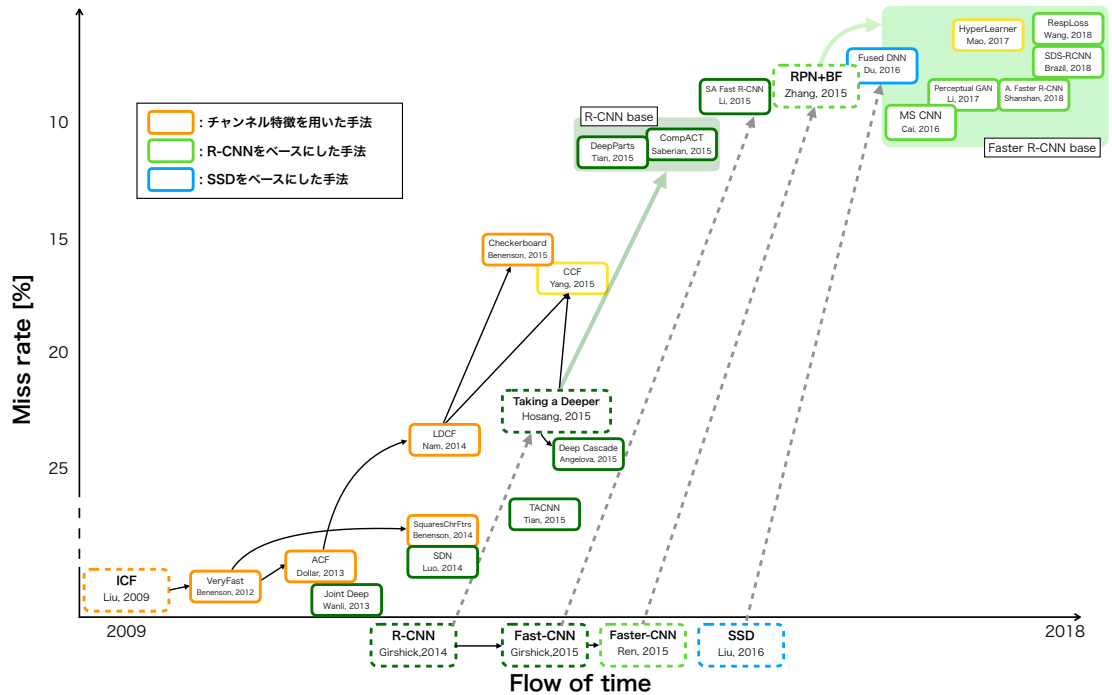


図 2.10: Caltech Pedestrian Dataset における歩行者検出法の推移

と組み合わせることで、歩行者検出の性能を大幅に向上させた [8]。HOG 特徴量をベースとした歩行者検出法は数多く提案されている。代表的な手法として、DPM がある。DPM は、解像度の異なる HOG 特徴量を用いて、Latent SVM によりグラフベースに歩行者のパーツの位置を予測しつつ歩行者検出の精度を向上させる手法である。

2009 年には HOG 特徴量のような歩行者の勾配だけを特徴量として採用するのではなく、特定の勾配方向や色、エッジ情報を特徴量として扱うチャンネル特徴量をベースとした手法が提案された。チャンネル特徴量を用いた手法は、勾配や色、エッジを抽出した後に Boosted tree で検出に有効な特徴量を学習により選択することで、歩行者を検出する。また、チャンネル特徴量を用いた ICF [2] 等の手法は、HOG 特徴量とは異なりブロック毎の正規化やヒストグラム化等を行わないため、非常に高速に処理できる。歩行者の検出精度もチャンネル特徴をベースとした手法が HOG 特徴量の性能を上回っていることもあり、2010 年から 2015 年のチャンネル特徴量をベースとした手法が数多く提案されている [54, 3, 73]。

2012 年に一般物体認識のコンペティションである ILSVRC [25] で、Krizhevsky らの CNN をベースにした手法 [6] がトップとなり、歩行者検出においても CNN をベースとした手法が数多く提案されている [56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 69, 68]。CNN による歩行者検出の先駆けとなる Joint Deep Learning [57] は、入力画像を CNN に入力することで特徴マップを獲得し、パーツ領域ごとに後段の Restricted Boltzmann Machine (RBM) [57] へ入力することで歩行者を検出する。その後、Luo らは歩行者検出に有効なパーツ領域を学習により選択する Switchable Deep Network (SDN) [58] を提案した。これらの手法は R-CNN の影響を強く受けており、2 段階の検出構造で実現されている。2 段階

表 2.1: 各データセットの詳細

	# of Ped.	学習のフレーム数	評価のフレーム数	撮影環境	撮影時間	オクルージョン情報	追加情報
Caltech [74]	学習: 192,000 評価: 155,000	33,171	4,024	走行シーン	3 時間	あり	
ETH [76]	学習: - 評価: 12,000	-	1,101	市街地	5 分	あり	ステレオ情報 トラッキング
Daimler [77]	学習: 15,600 評価: 56,500	Ped.: 15,560 Bkg.: 6,744	21,790	走行シーン	27 分 (評価のみ)	あり	ステレオ情報 セマンティックセグメンテーション
KITTI [75]	25,000	80,000		走行シーン	6 時間	あり	ステレオカメラ LIDAR 地図情報 物体の向き トラッキング
CityPerson [78]	学習: 192,000 評価: 155,000	33,171	4,024	走行シーン	3 時間	あり	セマンティックセグメンテーション インスタンスセグメンテーション ステレオ情報

の検出は、はじめに他の識別器で歩行者候補領域を検出し、後段の CNN で検出した歩行者候補領域に対して歩行者か否かを識別する。2 段階の検出構造における初期の研究では、歩行者のパーツを学習させる CNN を構築し、歩行者検出の精度を向上させた。その後、歩行者のパーツ情報ではなく、歩行者と背景の様々な属性を学習することで、歩行者検出性能を向上させた Task Assistance-CNN (TA-CNN) が提案された。2015 年には、複数の識別器をカスケード状に配置した Deep cascade や、特徴抽出においてカスケード構造を導入した Complexity-Aware Cascade Training (CompACT) [64] が提案され、CNN による歩行者検出の計算コスト削減に関する研究も取り組まれ始めた。

また、Fast R-CNN [49] や Faster R-CNN [14]、Single Shot Detector (SSD) [21] をベースにした歩行者検出法も提案されている [65, 64, 14, 68]。これら歩行者検出法は、スケールの小さい歩行者に対応するために、CNN の各層でスケールの異なる歩行者を検出したり、アンカーの走査で使用する特徴マップの解像度を変更するなどの工夫を施している。これまでに述べた、2004 年から 2016 年における歩行者検出の性能の遷移を図 2.9 に示す。図 2.9 の検出精度は、Caltech Pedestrian Benchmark [74] で比較している。図 2.9 より、HOG+SVM の検出精度が 68.74% で Fused DNN の検出精度が 8.2% であり、2004 年から 2016 年の間で 59.16% 検出精度が向上していることがわかる。

2.3.2 歩行者検出のデータセットと評価指標

歩行者検出法を同じ基準で評価するために、表 2.1 のような様々な歩行者検出のデータセットが公開されている。Dalal らは、HOG 特徴量と SVM の組み合わせを評価するために、INRIA Person Dataset [8] を構築した。INRIA Person Dataset は、約 2,200 枚程度の小規模なデータセットであり、デジタルカメラで撮影された画像から構築されている。しかし、ICF や CNN など学習サンプルを多く必要とする統計的学習法が用いられるようになり、Caltech Pedestrian Dataset [74] や KITTI Dataset [75] のような大規模なデータセットが用いられるようになった。本章では、歩行者検出の評価で用いられる各データセットと、評価指標について述べる。



図 2.11: 歩行者検出のデータセット例

■ 歩行者検出における主要なデータセット

Daimler Pedestrian Benchmark Data Sets [77] Daimler Pedestrian Benchmark Data Sets は Daimler 社が作成した初期の歩行者検出データセットであり、車載カメラで撮影した動画をベースにデータセットを構築されている。Daimler Pedestrian Benchmark Data Sets は、グレースケール画像で構築されており、切り出して正規化された歩行者画像と背景画像からデータセットが構築されている。撮影場所は、市街地や人工物の少ない道路や公園、建物内の駐車場など様々な箇所を撮影している。データセットの規模は後ほど説明する Caltech Pedestrian Dataset と同程度であり、学習サンプルと評価サンプルで約 36,000 枚の歩行者が存在している。2010 年以降に Daimler Pedestrian Data Sets は、ステレオカメラから取得した視差マップや、シーンセグメンテーションのデータも公開している。

INRIA Person Dataset [8] INRIA Person Dataset は、デジタルカメラで市街地の歩行者を撮影したシーンから構成されている歩行者検出のデータセットである。INRIA Person Dataset は他のデータセットと比べて、人の領域が大きく解像度が高いため、他のデータセットと比べて検出の難易度が低い。INRIA Person Dataset の特徴は、図 2.11(b) のような道路上に存在する歩行者だけでなく、スポーツやショッピングをしている人物等の様々なバリエーションの歩行者がデータセット中に存在する。

ETH Dataset [76] ETH Dataset は、人混みが多い市街地を撮影した動画から構成されている。また、RGB 画像のみでなく、ステレオカメラで撮影した距離画像も公開されており、歩行者の 3 次元位置とトラッキング情報も教師ラベルとして用意されている。ETH Dataset の規模は約 1,000 枚程度であるが、図 2.11(c) のような群衆のシーンが多いため検出の難易度が比較的に高い。また、ETH Dataset には学習用画像が用意されておらず、歩行者の見えが似ている INRIA Person Dataset [8] が学習サンプルとして一般的に用いられる。

Caltech Pedestrian Dataset [74] Caltech Pedestrian Dataset は、車載カメラで撮影した走行データから構成されているデータセットである。Caltech Pedestrian Dataset は、表 2.1 のように他のデータ

セットと比べて規模が大きいデータセットであり、一般道や高速、トンネルなどの様々な場所で撮影された約 6 時間程の走行映像から構築されている。データセット中には、歩行者が約 35 万枚存在している。また、歩行者にはオクルージョンが発生しているか否かのラベルも付与されている。Caltech Pedestrian Dataset の評価では、このオクルージョンの有無と歩行者のスケールで評価の難易度を変更することができる。Caltech Pedestrian Dataset は車載カメラで撮影されたカラー画像を用いて構築された初めてのデータセットであり、データ数やデータに対するラベル情報、評価ツールが充実しているため、主要な歩行者検出のデータセットの 1 つとして用いられている。

また、Caltech Pedestrian Dataset ではラベル付けのミスが報告されており、歩行者のラベルを付け直したデータが公開されている [67]。加えて、文献 [67] では歩行者検出における人の検出精度を示しており、現在の歩行者検出法ではまだ人の検出性能を上回っていない事を示している。

KITTI Vision Benchmark Suite [75] KITTI Vision Benchmark Suite は、歩行者検出や車両検出、道路認識等の Intelligent Transport Systems (ITS) 関係の研究向けに作成されたデータセットである。表 2.1 に示すように、KITTI Vision Benchmark Suite の学習サンプル数の規模は他のデータセットと比べて非常に大きい。KITTI Vision Benchmark Suite は数多くの都市を長時間撮影することで、多様な走行環境に対応したデータセットの構築を目的としている。また、RGB 画像だけでなく、ステレオ画像や LIDAR の 3 次元点群データ、GPS の地図情報などのデータも公開されている。これらのデータには、2 次元上の歩行者位置だけでなく、自動車の位置やこれらの 3 次元上の位置情報、及び地図上の位置のラベルも付与されている。このように、KITTI Vision Benchmark Suite はデータ数が非常に多く、ステレオ画像や LIDAR の 3 次元点群の情報など様々なデータが公開されていることから、2012 年以降は歩行者検出だけでなく車両検出やトラッキング等の ITS に関わる認識系のベンチマークとして幅広く活用されている。

CityPerson Dataset [78] CityPerson Dataset は、セグメンテーションの評価で用いられる Cityscape Dataset を歩行者検出用のデータセットに修正したデータセットである。CityPerson Dataset は、他の一般的であったデータセットと比べて様々な都市のデータを大量に含んでおり、画像に対するセマンティックセグメンテーションのラベルも高精細に施されていることから一般的に用いられている。CityPerson Dataset は、Caltech Pedestrian Dataset のアノテーション方法を参考に歩行者位置のラベル付けされており、歩行者のオクルージョン領域は Cityscape Dataset のインスタンスセグメンテーションを使ってラベル付けされている。また、Cityscape Dataset には視差データや GPS の情報等も付与されており、CityPerson Dataset で歩行者検出をする際に Cityscape Dataset の付与情報を併用することも可能である。

■ 評価指標

歩行者検出の評価では、Daimler や Caltech、CityPerson Dataset のように 1 枚の画像から歩行者の位置を検出ようになったため、Miss rate vs. False Positive Per Image (FPPI) における Receiver Operating Characteristic (ROC) を用いた評価と、Recall precision curve による評価が一般的に用いられる。FPPI では、1 枚の画像あたりの誤検出の割合を示している。FPPI による評価は、Caltech Pedestrian

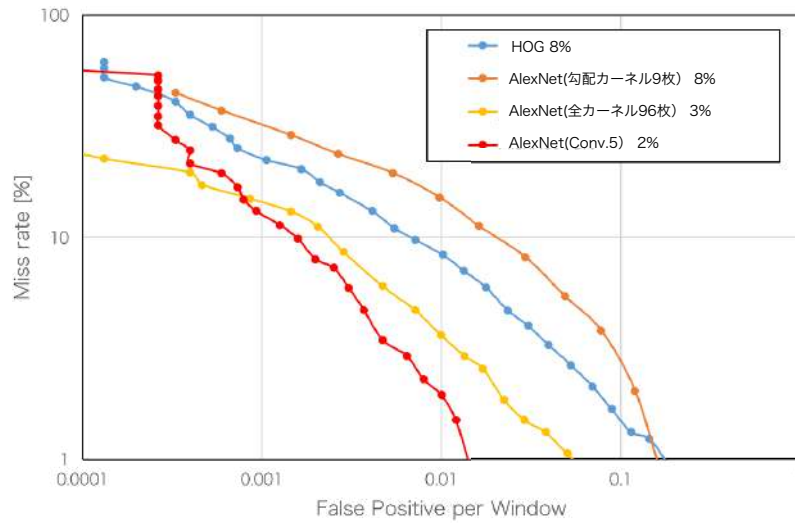


表 2.2: CNN ベースの歩行者検出法の分類

Dataset を筆頭に Daimler Mono, INRIA, ETH, CityPerson Dataset の評価指標として採用されている。KITTI Dataset では, Recall precision curve が評価指標として採用されている。基準となる評価値には平均の Miss rate が用いられ, FPPI が 0.01 から 0.1 までの 9 区間の Miss rate の平均が用いられる。ここで, Caltech Pedestrian Dataset の修正された際のラベルを用いて評価する場合, 9 区間の Miss rate の平均を用いて評価するのではなく, 0.01 と 0.0001 の 2 つの Miss rate を用いて評価するのが一般的である。

また, これらの歩行者検出では検出する歩行者の難易度を考慮した評価方法が導入されている。代表的な例として, Caltech Pedestrian Dataset では歩行者のスケール, 発生しているオクルージョンの割合を加味した評価指標が用意されている。Caltech Pedestrian Dataset の場合, 歩行者のスケールで難易度を分けた Near Pedestrian, Medium Pedestrian, Far Pedestrian が用意されており, オクルージョンに対する難易度は No Occlusion, Partial Occlusion, Heavy Occlusion が用意されている。これらの評価指標は, Daimler, INRIA, ETH Dataset においても使用することができる。CityPerson Dataset では, 小さな歩行者を評価する Reasonable small, オクルージョンの強い歩行者を評価する Reasonable Occlusion, データセット中の全ての歩行者を評価する All の評価指標が用意されている。

2.3.3 HOG 特徴量と CNN の特徴表現

2015 年以降の歩行者検出法は, CNN をベースにした歩行者検出法が活発に提案されている。ここで, 2004 年から 2009 年の歩行者検出で一般的に用いられていた HOG 特徴量と AlexNet の特徴表現の性能をトイプロBLEM で比較する。トイプロBLEM には, INRIA Person Dataset を使用し, False Positive Per Window (FPPW) により評価する。HOG 特徴量または AlexNet から獲得した特徴マップを SVM で学習することで検証する。AlexNet からは, 3 種類の特徴量を抽出し, 歩行者検出性能を

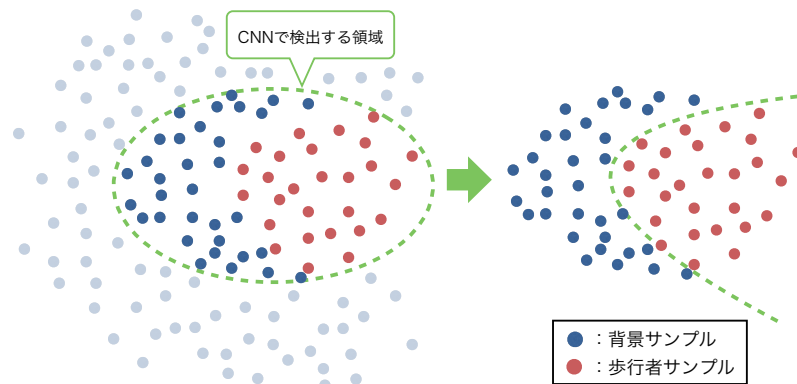


図 2.12: 2 段階の検出構造の効果

比較する．1 つ目は，1 層目の畳み込み層のカーネルから HOG 特徴量と同様に 9 方向の勾配に反応するカーネルを 9 枚選択して特徴マップを獲得し，ブロック正規化とヒストグラム化を行った特徴量である．ここで，9 枚のカーネルは目視により選択する．2 つ目は，1 層目の畳み込み層のカーネルを全て選択して特徴マップを獲得し，ブロック正規化とヒストグラム化を行った特徴量である．3 つ目は，5 層目の畳み込み層の特徴マップを特徴量として使用する．ここで，使用する AlexNet は ImageNet Dataset で学習したモデルを使用する [6]．

図 2.2 に，AlexNet と HOG 特徴量の検出精度を示す．9 枚の勾配方向カーネルを用いた AlexNet と HOG 特徴量の精度を比較した時，AlexNet の検出精度は HOG 特徴量より精度が低下している．一方で，一層目の 96 枚のカーネルを用いた場合，HOG 特徴量の検出精度より Miss rate を約 5% 削減している．この結果より，従来の HOG 特徴量より AlexNet の特徴表現の方が高性能であることが確認できる．また，AlexNet の 5 層目の特徴マップを用いた場合，Miss rate を約 2% まで削減できた．

2.3.4 2 段階構造による歩行者検出

CNN を歩行者検出に応用する場合，ICF 等の識別器と CNN を 2 段階に配置して歩行者を検出する．2 段階の検出構造による歩行者検出法は，候補領域を検出する識別器が必要である図 2.8(a) の R-CNN と図 2.8(b) の Fast R-CNN が該当する．2 段階の検出構造による歩行者検出法を使用する場合，前段の識別処理で歩行者の未検出を防ぐために検出時のしきい値を下げて未検出を防ぐ等の工夫が必要となる．

2 段階の検出構造は，図 2.12 のようなサンプルの分布が存在する場合，前段の識別器でサンプルの分布から CNN で識別する範囲を限定し，後段の CNN で詳細な識別を行っている．図 2.12 のように，後段の CNN が識別する背景パターンを削減する効果があるため，誤検出を削減できる．

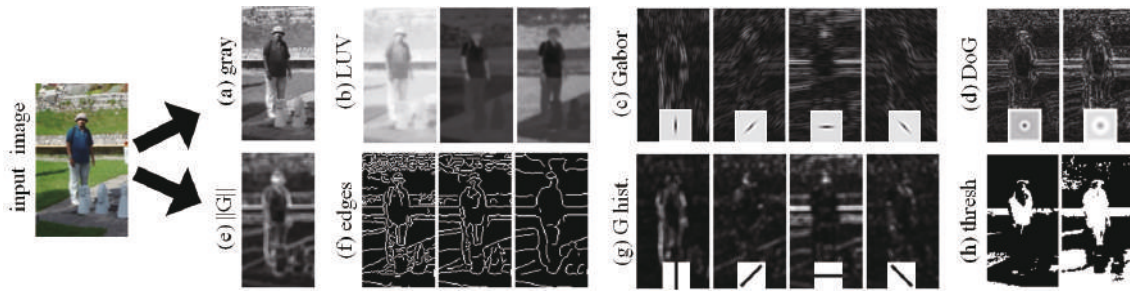


図 2.13: ICF 特徴量の抽出 (文献 [2] 引用)

■ 前段に用いる歩行者検出法

2 段階の検出構造において、前段で使用する識別器の選択は非常に重要である [59]。もし前段の識別器で歩行者を取りこぼした場合、後段の CNN に入力されずに未検出となる。また、計算コストを考慮する場合、高速に処理が可能な識別器を用いる必要がある。2014 年に提案された Joint Deep Learning と SDN では、前段の識別器に CSS+HOG 特徴量 [52] を用いている。また、2015 年以降は、ACF-Caltech を前段の検出法として用いられている。本節では、前段の識別器として用いられている手法について詳細に述べる。

Color Self Similarity HOG [52] CSS+HOG 特徴量は、勾配情報である HOG 特徴量に Color Self-Similarity (CSS) 特徴量の色情報を組み合わせた特徴量である。色情報は、歩行者の肌や服装、輝度の変化、人工物等の色の変化が発生しやすいことから、歩行者検出に有効な特徴抽出が期待できないとされていた。しかし、CSS 特徴量は画像全体の色情報を用いるのではなく、画像内の 2 つの局所領域の類似度を特徴量として扱うことで、歩行者検出に有効な特徴量となる。

Aggregate Channeled Feature [55] Aggregate Channeled Feature (ACF) は、ICF をベースにチャンネル特徴量を抽出した後に集約処理を施して特徴ピラミットを導入した手法である。ACF で用いるチャンネル特徴量は ICF と同様であり、図 2.13 の (a) 入力画像の輝度画像、(b) 表色系を変換した画像、(c) ガボールフィルタを畳み込んだ勾配画像、(d) Difference of Gaussian (DoG) フィルタを畳み込んだ画像、(e) 勾配強度画像、(f) エッジ画像、(g) 特定方向のガボールフィルタ、(h) 2 値画像の 16 チャンネルを特徴量を用いる。チャンネル特徴量を抽出した後、集約処理を行う。ACF の集約処理では、各チャンネル特徴量の局所領域内を総和し、新たなチャンネル特徴量を生成する。抽出したチャンネル特徴量は、特徴ベクトルに変換され、Boosted tree に入力される。Boosted tree に抽出したチャンネル特徴量を入力することで、膨大な特徴量群から歩行者検出に有効な特徴量を選択できる。

チャンネル特徴量をベースとした手法は、チャンネル特徴量の生成と Boosted tree による特徴量の選択が基本的な構造となっている。特に、チャンネル特徴量の生成において、バリエーションが豊富なチャンネル特徴量を生成することで、高精度化に繋ぐことができる。ACF の場合、集約処理により 1 枚あたりのチャンネル特徴量は縮小されているが、ピラミッドを作成することでチャンネル特徴量のバリエーションを増やしている。ACF をベースとした手法で、Locally Decorrelated Channel Features (LDCF) [54] と、SquaresChrFtrs [3]、Checkerboard [73] がある。これらの手法は、ACF の集

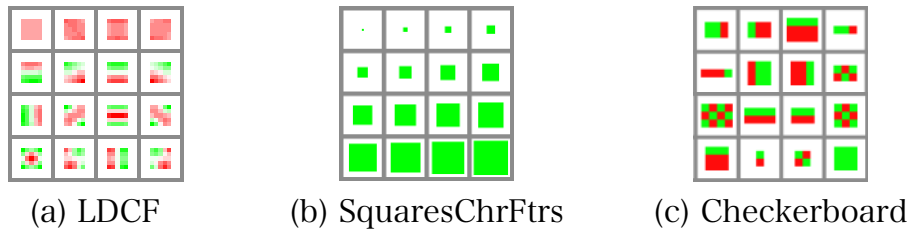


図 2.14: Filtered Channel Feature のフィルタパターン (文献 [3] 引用)

表 2.3: 2 段階による歩行者検出と Region Proposal による歩行者検出の性能比較

手法	年代	Miss rate [%]	速度 [fps]	オクルージョン パート	群衆 対応	CNN 特徴量 + 識別器	スケール対応	ベースモデル
Joint Deep [57]	2013	39.3		✓	✓	✓		CNN+RBM
SDN [58]	2014	37.87	0.7	✓	✓			CNN+RBM
TACNN [60]	2015	20.86 _(18.8)	1					AlexNet
CCF [61]	2015	17.32				✓		VGG
Deep Cascade [62]	2015	26.21	15					VGG
DeepParts [63]	2015	11.89 _(12.9)		✓				GoogLeNet
CompACT [64]	2015	11.75 _(9.2)				✓		CNN, VGG
SA-Fast R-CNN [65]	2015	9.68	2				✓	Fast R-CNN
RPN+BF [69]	2016	9.58 _(7.3)	2			✓	✓	RPN (VGG)
MS-CNN [66]	2016	9.95	2.5				✓	RPN (VGG)
Fused DNN+SS [68]	2016	8.18	0.5		✓			SSD
Perceptual GAN [80]	2017	9.48					✓	Faster R-CNN (VGG)
HyperLearner [81]	2017	— _(5.5)	4		✓			Faster R-CNN (VGG)
Adap. Faster R-CNN [78]	2018	9.18 _(5.1)					✓	Faster R-CNN (VGG)
SDS-RCNN [82]	2018	7.63	5		✓			Faster R-CNN (VGG)
RespLoss [83]	2018	— _(5.0)			✓			Faster R-CNN (ResNet)

約処理において局所領域内の総和を求めるのではなく、独自のフィルタパターンを畳み込むことで、チャンネル特徴量のバリエーションを増幅し、より高精度な歩行者検出を実現している。

Locally Decorrelated Channel Features [54] LDCF は、図 2.14(a) のような ACF の集約処理に学習サンプルの無相関フィルタを畳み込む処理を追加した手法である。集約処理で使用する無相関フィルタは、Linear Discriminant Analysis (LDA) [79] により求めている。これにより、抽出したチャンネル特徴量の相関を取り除くことができ、高精度化を実現することができる。

SquaresChrFtrs [3] 従来の ACF では、一定サイズのみで集約処理を行っていた。SquareChrFtrs は、図 2.14(b) のような複数のサイズで集約処理を行っている。これにより、集約処理で発生していたスケールの小さな歩行者の特徴量が消失する問題が半減され、スケールの小さな歩行者を高精度化に検出できる。

Checkerboard [73] Checkerboard は、ACF の集約処理に図 2.14(c) のチェッカーボードパターンのフィルタを畳み込む手法である。チェッカーボードパターンのフィルタは全部で 132 パターン使用しており、各フィルタを畳み込むことでチャンネル特徴量のバリエーションを増やし、高精度化を実現している。

■ 後段に用いる CNN ベースの歩行者検出法

後段に用いられている CNN ベースの歩行者検出法は、表 2.3 のようなグループ分けができる。表 2.3 の上半分は 2 段階の歩行者検出法を示しており、下半分が Region Proposal による歩行者検出法を示している。ここで、“Miss rate”に記されている数値は Caltech Pedestrian Dataset のラベルを用いて、FPPI が 0.1 から 0.001 までの Miss rate の平均を示した値である。カッコ内の数値は、Zhang らによって作成された新しいアノテーションデータを用いて評価した際の Miss rate を示している [67]。本節では、1) パーツベースの歩行者検出法と 2) オクルージョンと群衆に対応した歩行者検出法、3) CNN 特徴量を用いた歩行者検出法、4) スケールに対応した歩行者検出法、5) どのグループにも属さない歩行者検出法について述べる。

パーツベースの歩行者検出法 CNN ベースの歩行者のパーツ情報を用いた手法に、Joint Deep Learning [57] がある。Joint Deep Learning は、CNN で歩行者に対する各パーツのスコアを出力し、RBM で抽出した特徴量とスコアを入力して人か背景かを識別する。入力画像を Joint Deep Learning へ入力した際に、頭部や腕等の歩行者の各パーツに対するスコアを出力する。ここで、出力層の前層には Deformation 層があり、前層の特徴マップから不要なパーツの応答値を取り除いている。そして、Deformation 層で取得した特徴マップを用いて、歩行者の各パーツに対するスコアを得る。RBM には、CNN の畳み込み層で得られる特徴マップと各パーツに対するスコアを入力する。入力する CNN の特徴マップは、特定の領域にマスクをかけた状態で入力する。1 段目の CNN で得られた歩行者の各パーツに対するスコアは、RBM の各層の重みとして用いられる。Joint Deep Learning は、歩行者のパーツ領域の特徴量で RBM を最適化しているため、姿勢の変化に頑健な歩行者検出を実現している。

Joint Deep Learning が提案された後に、歩行者の頭部や腕等の局所的なパーツではなく、歩行者の上半身や腰、下半身のパーツを用いた Switchable Deep Network (SDN) [58] が提案されている。SDN は、歩行者の上半身や腰、下半身のパーツ領域から識別に有効なパーツを選択しながら学習することで、高精度な歩行者検出を実現している。歩行者のパーツの選択は、Switchable 層によって行われる。Switchable 層は、畳み込み処理とプーリング処理により特徴マップを Spatial Pooling によりパーツ毎に分割し、Switchable Restricted Boltzmann Machine (SRBN) により歩行者のパーツを選択している。

また、Joint Deep Learning や SDN のように歩行者の各パーツの特徴量を用いて検出するのではなく、学習時に歩行者の特定のパーツに対してペナルティを与える DeepParts [63] も提案されている。DeepParts は、前段の検出位置とラベル位置の重なり位置から複数の CNN を構築し、前段の検出位置とラベル位置が重なっていないパーツに対してペナルティを与える。DeepParts の学習では、前段の検出位置とラベル位置のそれぞれで学習した CNN と、これらを統合した CNN の 3 つを使用する。このとき、統合した CNN の規模は統合前の CNN より大きく、前段の検出位置とラベル位置でズレが発生しているパーツに対してペナルティが与えられ、最終的な出力が得られる。

CNN 特徴量を用いた歩行者検出法 CNN から抽出した特徴量と他の統計的学習法を組み合わせた代表的な手法として、Convolutional Channel Feature (CCF) [61] がある。CCF は、ACF と CNN を組み合わせた歩行者検出法である。従来の ACF では、LUV 画像や勾配画像等の 10 チャンネルのチャンネル特徴量を Boosted tree に入力していた。CCF では、CNN から得られる特徴マップをチャンネル特

微量として入力する。CCF で使用するネットワークモデルは、AlexNet や VGGNet, GoogLeNet [29] で検証されているが、16 層の VGGNet が最も良いとされている。CCF は、16 層の VGGNet から得られる多量の特徴マップから、歩行者検出に有効な特徴量を Boosted tree で選択することができるため、高精度な歩行者検出を実現している。

Complexity-Aware Cascade Training (CompACT) [64] は、CNN から得られる特徴量や Self-similarity (SS) [84], Checkerboard, LDA を特徴量とし、AdaBoost で弱識別器として特徴量を選択する歩行者検出法である。SS や CB, LDA, CNN の特徴量を AdaBoost で学習した場合、精度を基準に特徴量を選択するため、CNN の特徴量が数多く選択される。そのため、SS や ACF などの比較的高速に特徴量を取得できる特徴量が選択されず、計算量が膨大になり、識別速度が大幅に上昇する。CompACT は、学習時に特徴量の計算コストも考慮させることで、SS や ACF などの比較的高速に特徴量をカスケード型 AdaBoost で選択されるようにする。これにより、性能を維持しつつ計算量を削減できるため、高性能かつ高速な歩行者検出を実現している。

スケールに対応した歩行者検出法 2 段階の歩行者検出法において歩行者のスケールに対処した代表的な手法として、Scale Aware Fast R-CNN (SA-Fast R-CNN) がある。SA-Fast R-CNN は、Fast R-CNN の識別結果を出力するネットワーク (サブネットワーク) をスケールの大きい歩行者とスケールの小さい歩行者を識別するようにそれぞれ 2 つ構築する。検出時は、2 つのサブネットワークから検出スコアを出力し、歩行者のスケールをベースに重み付けすることで最終的な検出スコアを決定する。これにより、歩行者のスケールに頑健な歩行者検出が可能であり、Caltech Pedestrian Dataset のベンチマークにおいて Miss rate を 10% 未満の性能を達成している。

その他の歩行者検出法 Task Assistance-CNN [63] (TA-CNN) は、Multi-task Learning により歩行者と背景の属性を学習させることで高性能な歩行者検出を実現している歩行者検出法である。使用した歩行者の属性は、向きや自転車に乗っているか、性別などの 9 種類である。また、背景の属性は 4 つのセグメンテーションのデータセットから付与されており、空や木、建物など 8 種類ある。背景の属性は Semantic Tasks という方法により付与される。Semantic Tasks は、セグメンテーションのデータセットから Hard Negative を検出し、セグメンテーションのラベルと Hard Negative の検出位置を照らし合わせることで属性を付与している。学習時は、Task- Constrained Deep Convolutional Network (TCDCN) による学習誤差の重み付けと、Task-wise early stopping による過学習の抑制を行っている [85]。TCDCN と Task-wise early stopping は、メインタスクの学習を補助するようにサブタスクを学習できるため、精度向上を可能にしている。

Deep Cascade [62] は、カスケード構造を取り入れることでリアルタイムで処理可能な歩行者検出法を実現している。Deep Cascade は、2 段の検出構造で最も高速な歩行者検出法である。ここで、Deep Cascade の前段には VeryFast が採用されている。Deep Cascade では、大小の規模が異なるネットワークモデルを 2 つ用い、カスケード状に配置している。ここで、小規模なネットワーク (Tiny CNN) は VeryFast の後段に使用し、大規模なネットワークは小規模なネットワークの後段に配置する。最も後段にある大規模なネットワークは、VeryFast と Tiny CNN が歩行者と判定した場合のみ使用されるため、高速に歩行者を検出できる。

表 2.4: 2 段階による歩行者検出と Region Proposal による歩行者検出の性能比較

	Reasonable	Reasonable small	Reasonable occlusion	All
2 段階の検出 (LDCF+VGG)	33.15	58.42	81.55	68.07
Region Proposal の検出 (Faster R-CNN)	22.87	50.84	54.95	47.53

2.3.5 Region Proposal ベースの歩行者検出法

Region Proposal ベースの歩行者検出法は、歩行者候補領域の検出から最終的な検出結果の獲得までを一つのネットワークで処理する。Faster R-CNN や SSD 等の、1 つのネットワークで検出結果を出力できる物体検出法が該当する。Region Proposal ベースの手法の利点は、歩行者検出の一連の処理を単一のネットワークで処理できるため、高速に歩行者検出が可能である。そして、これらの一連の処理を単一のネットワークで End-to-End に学習するため、2 段階の歩行者検出法より高精度に歩行者を検出できる。

2 段階の検出構造における歩行者検出と Region Proposal ベースの手法の検出精度を比較する。2 段階の検出構造における歩行者検出法には、前段に LDCF を使用し、後段に VGGNet の 16 層モデル (以下、VGG16) を使用する。Region Proposal ベースの手法では、VGG16 をベースに構築した Faster R-CNN を使用する。データセットには CityPerson Dataset を用いて、Reasonable, Reasonable small, Reasonable occlusion, All の 4 つの評価指標で評価する。表 2.4 に 2 段階の検出構造における歩行者検出法と Region Proposal ベースの歩行者検出法の検出結果を示す。表 2.4 の結果から、Region Proposal ベースの歩行者検出法は 2 段階の歩行者検出法より全ての評価指標において Miss rate が低いことがわかる。図 2.15 に、Faster R-CNN による歩行者検出結果を示す。図 2.15 のように、遠方の小さな歩行者や群衆に対して頑健に検出できている。Region Proposal ベースの手法は精度が良く、比較的計算コストが低い。また、R-CNN は前段の識別器との学習の調整も必要であり、後段の CNN の学習が非常に難しい。これらの理由から、2016 年以降の歩行者検出法では Region Proposal ベースの手法が一般的に用いられている。

スケールに対応した歩行者検出法 歩行者検出において検出精度を大幅に低下させる要因の一つとして、歩行者のスケールの問題がある。歩行者検出では、車載カメラにより獲得した走行画像から歩行者を検出する必要がある。車載カメラから歩行者までの位置が遠方に位置する程、画像中の歩行者のスケールは小さくなる。また、自動車はブレークから完全に停止するまである程度距離を必要とするため、遠方の歩行者検出は必要不可欠な技術である。

Region Proposal ベースの物体検出法は、特徴抽出の過程でダウンサンプリングされた獲得した特徴マップを用いて検出する。そのため、Region Proposal ベースの識別器を遠方の歩行者検出に応用するのが困難である。歩行者のスケール問題を解決するために、Zhang らはベースラインで使用する RPN が遠方の歩行者も検出できるように様々なネットワークの改良を行っている [14]。Zhang らが行った改良は主に、検出する特徴マップの解像度を上げることである。VGGNet をベースに構築される RPN は VGGNet の最後の畳み込み層から物体候補領域を検出しているが、Zhang らの RPN は



図 2.15: Faster R-CNN による歩行者検出結果

最後のプーリング層を取り除いた際の特徴マップを用いて物体候補領域を検出する．これにより，検出を行う特徴マップの解像度が上がり，小さな歩行者も検出できるようにしている．また，歩行者の形状に酷似した Hard Negative をより高精度に検出できるように，Boosted Forest を導入するテクニックも導入している．このようなネットワークの調整によるアプローチは，Shanshan らの研究でも取り組まれている．Shanshan らは，入力画像のサイズを 2 倍にすることで，獲得できる特徴マップの解像度を向上させている [78]．また，CityPerson Dataset で学習した Faster R-CNN をプレトレーニングすることで，Caltech Pedestrian Dataset の歩行者検出精度を向上させている．

歩行者のスケールに対応した別の歩行者検出法として，Multi-scale CNN と Perceptual Generative Adversarial Network (Perceptual GAN) がある．MS-CNN では，一般的な Faster R-CNN では特定の 1 つの層の特徴マップから RPN と Fast R-CNN により検出を行っていたが，様々な層の特徴マップを用いて検出を行うアプローチである．このアプローチは SSD と似ており，様々な層の特徴マップから検出を行うことでスケールに頑健な歩行者検出を可能にしている．Perceptual GAN は，スケールの小さい物体の特徴マップをスケールの大きい物体の特徴マップに Generative Adversarial Network (GAN) [70] で近似する手法である [80]．これにより，小さな物体の検出をより頑健に検出することが可能である．

群衆およびオクルージョンに対応した歩行者検出法 自動車の走行シーンにおいてオクルージョンは歩行者や自動車等の位置関係から発生しやすい．オクルージョンは，歩行者検出の精度を低下させる要因の一つである．歩行者のオクルージョンには，自動車等の人工物による隠れと，歩行者同士の群衆による隠れがある．人工物によるオクルージョンでは，主にセマンティックセグメンテーションを用いた方法が数多く提案されている．

Fused DNN は，SSD の歩行者検出結果とセマンティックセグメンテーションの結果を Soft-rejection based Network Fusion (SNF) により統合することで，高精度な歩行者検出を実現している [68]．Fused DNN では，はじめに SSD による歩行者候補領域の検出とセマンティックセグメンテーションを行う．ここで，セマンティックセグメンテーションには Fully Convolutional Network (FCN) [86] をベースに構築した CNN を使用している．検出した歩行者候補領域とその領域のセマンティックセグメンテーションの結果は，SNF に入力されて最終的な歩行者検出結果を出力する．SNF は，検出した歩行者候補領域に対して 0 と 1 のラベルで学習するのではなく，SSD の検出スコアとセマンティックセグメンテーションの面積からラベルをスケールリングしている．異なるアプローチとして，セマンティックセグメンテーションだけでなく，エッジ情報やオブティカルフロー，視差 (距離情報) 等を抽出し，チャンネル特徴のように組み合わせることで歩行者検出の精度を向上させる HyperLearner も提案されている [81]．セマンティックセグメンテーションはラベル付けにコストがかかるため，セマンティックセグメンテーションのラベルを用いずに歩行者の領域を弱教師あり学習で獲得し，歩行者検出の性能を向上させる歩行者検出法も提案されている．

歩行者同士の重なりが発生しやすい群衆に頑健な歩行者検出法として，Wang らの Repulsion Loss がある [83]．Repulsion Loss は，歩行者のバウンディングボックスを学習する際に，GT のバウンディングボックスと出力したバウンディングボックスを引き寄せる誤差関数と，GT のバウンディングボックスと出力したバウンディングボックスを反発させる誤差関数の 2 つの誤差を示している．Repulsion Loss の導入により，歩行者の間に発生するバウンディングボックスが抑制させ，群衆に頑健な歩行

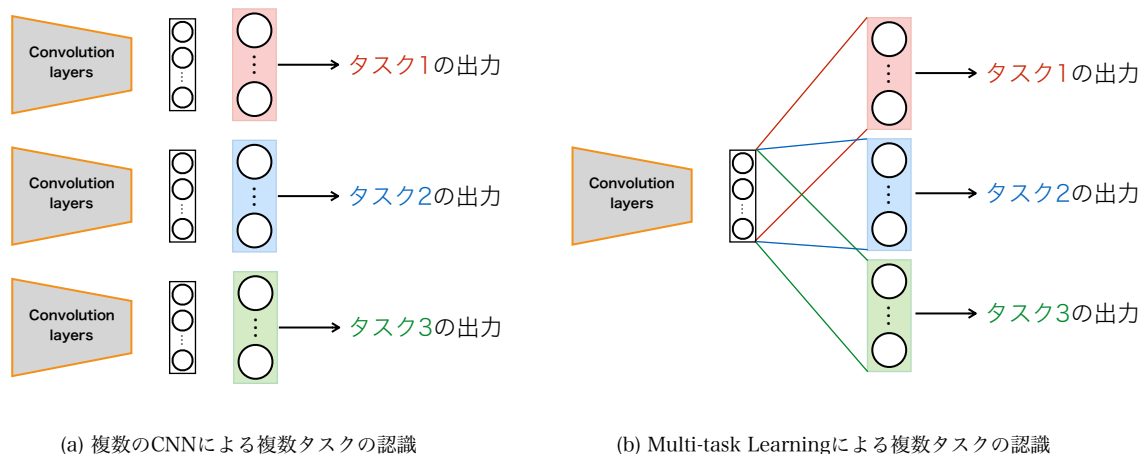


図 2.16: Heterogeneous Learning を導入した CNN

者検出を実現している。Wang らの Repulsion Loss は、2018 年における Caltech Pedestrian Dataset のベンチマークにおいてトップの性能を実現している。

2.4 Multi-task Learning

複数のタスクを 1 つのネットワークで学習、認識するアプローチとして、Multi-task Learning がある。複数のタスクを認識する場合、図 2.16 のように 1 つのタスクに対して 1 つのネットワークを用意し、認識するアプローチが取られていた。しかし、これらのアプローチは認識するタスク数が増加した際に計算コストも比例して増加する。一方で、Multi-task Learning は図 2.16 のようにネットワークの出力層以外の層を全タスクで共通化し、各タスクを認識する。これにより、1 つのネットワークで複数のタスクを認識できる。そして、各出力層から得られた認識結果からそれぞれのタスクで誤差を算出し、ネットワークを学習する。Multi-task Learning の一種として、Heterogeneous Learning がある。Heterogeneous Learning は、各タスクで学習誤差を算出する際に、異なる複数の学習誤差関数で誤差を算出するアプローチである。Heterogeneous Learning の特徴は、分類タスクや回帰タスクを単一のネットワークで学習、認識できる点である。

ミニバッチによる学習の場合、はじめに学習データセットから M 個の学習サンプルを選択し、ミニバッチを作成する。 M 個の学習サンプルをネットワークに入力することで学習誤差 L を求め、CNN のカーネルや結合重みのパラメータ θ を 1 回更新する。この時、Multi-task Learning を CNN に導入する場合、特定のタスク t に対して誤差関数を割り当てて学習誤差 L を求める。Multi-task Learning は、出力層の各ユニットが、各タスクに対するスコアを出力している。そのため、タスク t が回帰推定か認識によって誤差関数を設計する必要がある。回帰推定の場合は式 (2.17) の L2 ノルム 2 乗誤差を用い、認識の場合は式 (2.18) のクロスエントロピー誤差を用いる。そして、各タスクの学習誤差 L_t から、式 (2.19) の誤差逆伝播法を用いて CNN のカーネルや結合重みを更新する。

$$L_t^{Regression} = \frac{1}{M} \sum_{m=1}^M \| \mathbf{y} - \mathbf{o} \|_2^2 \quad (2.17)$$

$$L_t^{Recognition} = \frac{1}{M} \sum_{m=1}^M -\mathbf{y} \log \mathbf{o} \quad (2.18)$$

$$\begin{aligned} \boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} + \Delta \boldsymbol{\theta} \\ &= \boldsymbol{\theta} - \eta \frac{\partial \sum_{t=1}^T L_t}{\partial \boldsymbol{\theta}} \end{aligned} \quad (2.19)$$

CNN による顔の属性を認識する技術では、個別の顔属性を単一の CNN で推定する手法が数多く提案されていた [87, 88, 89] これは、歩行者属性認識の問題においても同様である [90]. 一方で、単一の CNN により複数の顔属性及び、歩行者属性をマルチタスク学習及び **Heterogeneous Learning** により推定する手法も提案されている [85, 63]. その際に、複数のタスクを効率的に学習するためのアプローチとして、重み付き誤差関数の設計や、少量サンプルの効率的な学習方法も導入している手法も存在する.

2.5 まとめ

本章では、画像認識分野で用いられる CNN の基礎的な構造や学習法を述べた後に、関連研究についてまとめた. 2.1 章では、CNN を構築する層の構造や処理、ネットワークの学習法等について述べた. また、CNN をベースに構築された AlexNet や VGGNet, ResNet 等の画像分類で一般的に用いられるモデルについて説明した後に、CNN の汎化性能を向上させる技術について述べた. 2.2 章では、CNN ベースの物体検出法と歩行者検出法についてのサーベイを行い、研究動向や評価指標、一般的に用いられるデータセットについての調査を行った. 2.4 章では、**Multi-task Learning** のアルゴリズムの説明と、重み付き誤差関数と少量サンプルに対する関連手法について調査し、まとめた.

以降の章では、3 章と 4 章, 5 章, 6 章について述べる. 3 章では、認識精度とネットワークが持つパラメータ数がトレードオフの関係にある問題に取り組む. この問題を解決するために、本稿では **Random Dropout** と **Ensemble Inference Networks** を提案し、パラメータの増加を抑制しながらネットワークの性能を向上させる. 4 章では、2 つの問題について取り組む. 1 つ目は、重み付き誤差関数によるアプローチは特定のタスクの精度のみを向上させるアプローチが用いられる点である. 提案手法では、全タスクにおける学習誤差のばらつきを考慮して誤差関数の重みを求めることで、異なる複数のタスクの精度を向上させる. 2 つ目は、サンプル数の割合が低いクラスの精度が低下しやすい点である. 提案手法では、各クラスにおけるサンプルの割合から適切に CNN へ入力するサンプルを選択することで、少量サンプルのクラスの精度を向上させる. 5 章では、深層学習の判断根拠を解析するために、ネットワークが注視した領域をマップで表現した **Attention map** を用いて解析する視

覚的解析モデルを応用する。従来の視覚的説明モデルが性能低下を引き起こしやすい問題を解決するために、視覚的解析モデルから得られる Attention map を Attention 機構へ応用する。Attention 機構は特定の領域の特徴を強調させることでネットワークの汎化性能を向上させる手法であり、認識において高く貢献した領域を可視化する視覚的説明手法と相性が非常に良い。本稿では、視覚的説明モデルを拡張して Attention 機構を導入した Attention Branch Network を提案する。視覚的説明モデルを Attention 機構へ応用することで、視覚的説明モデルの性能を向上しつつ、Attention map による注視領域の獲得を行うことができる。6 章では、Attention Branch Network を強化学習に応用する。Attention Branch Network を強化学習へ応用することで、エージェントを制御する際に注視した領域を可視化できるようにする。

第3章

Random DropoutとEnsemble Inference Networksによる歩行者検出と標識認識

夜間など危険な状況下でのドライバの判断を支援するADASでは、歩行者検出や標識認識を高精度に検出、識別する必要がある。また、ADASにおけるこれらの認識技術は、小型の組み込み機により処理をする必要があるため、計算コストの低い識別器を構築する必要がある。歩行者検出と標識認識は、車載カメラ映像の単一フレームを用いる方法と複数フレームを用いる方法に分類される。歩行者検出と標識認識は、車載カメラ映像の単一フレームを用いる方法と複数フレームを用いる方法に分類される。複数フレームを用いる手法は、コンテキストや動き情報を利用できるため、高精度な検出や認識が可能となる。しかしながら、コンテキストや動き情報は、計算コストが非常に高い。それに対して、単一フレームによる処理では、実時間で検出が可能であるが、走行環境や天候の変化が生じて1枚の画像から高精度に検出、認識する必要がある。本稿では、車載カメラによる運転支援システムの普及を目的とするため、単一フレームによる方法で識別を行う。

歩行者検出と標識認識では、CNNを用いたアプローチが数多く提案されている[11, 58]。しかしながら、CNNを用いたアプローチは高い精度を達成しているが、複数のネットワークを導入したり、層を大量に追加することで汎化性能を向上させている。これらの高精度化のための手段は、ネットワークのパラメータ数を大幅に増加させる。パラメータ数の増加は計算コストを大幅に増加させるため、ADASのような組み込み機器による処理を必要とする場合に大きな問題となる。

CNNのパラメータ数を抑制しつつ高い性能を実現するために、汎化性能を向上させるために用いられるDropoutを利用する。Dropoutは、学習時に一定の割合でランダムにユニットを選択し、応答値を0にする。ランダムに選択するユニットは、各更新処理で異なる。我々はDropoutの以下の2点に着目する。1つ目は、学習処理において、ユニットの応答値を0にする割合を固定している点、2つ目は、学習時への導入に留まっており、識別時に同様の処理は行われていない点である。

本稿では、Dropoutをベースとした2つの手法を提案する。1つ目は、学習時に用いるRandom Dropout、2つ目は、識別時に用いるEnsemble Inference Networks (EIN)である。Random Dropoutは、各更新回数で応答値を0にする割合をランダムに指定する。これにより、歩行者や標識の隠れや照明変化に対して頑健なネットワークを学習し、汎化性能を向上させる。EINは、学習した1つのネットワークからランダムに選択したユニットの応答値を0にする全結合層を複数生成し、最終的な出力は特定の統合方法を用いることで識別する。構造の異なる複数の全結合層のネットワークを用いて検出、識別をすることで、誤検出と誤識別を削減する。

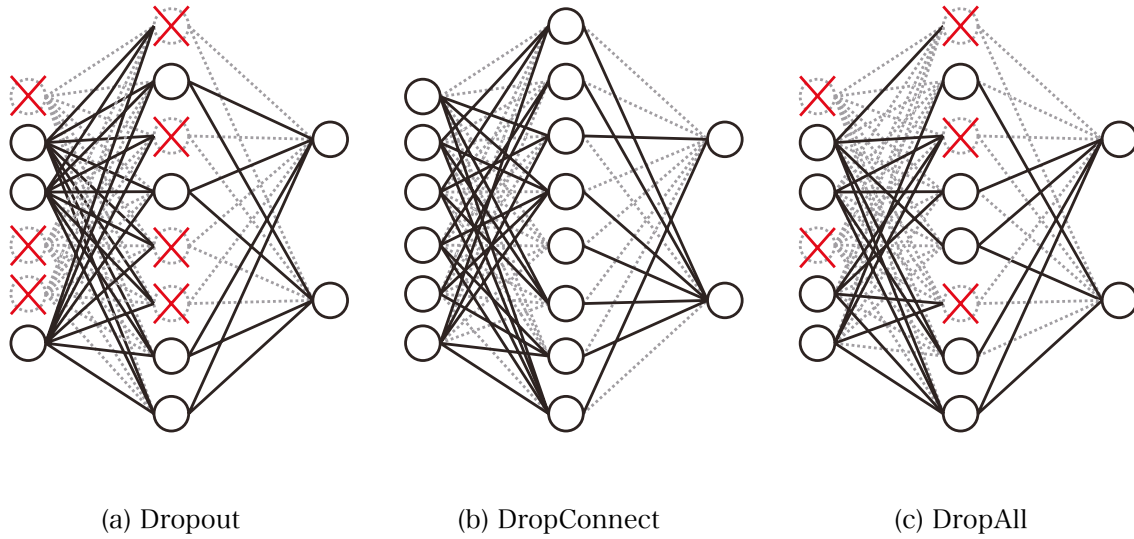


図 3.1: Dropout の発展系の手法

3.1 関連研究

ニューラルネットワークの学習において、汎化性能を向上させる正則化法として Dropout [12] がある。Dropout は、図 3.1(a) のように全結合層のユニットを指定した割合だけランダムに選択し、その応答値を 0 にして学習する手法である。式 (3.1) のように、ユニットの応答値を 0 にする場合は m_j^l を 0、ユニットの応答値を伝播する場合は m_j^l を 1 にすることで、 l 層目のユニット j の応答値を制御している。Dropout は、各更新処理で異なるユニットを選択することで、異なる一部の結合が取り除かれた場合においても識別できるようにネットワーク全体のパラメータが学習される。この処理により、ネットワークの過学習を抑制し、汎化性能を向上できる。識別時は、学習時に指定したユニットの応答値を 0 にする割合を全ユニットの応答値に乗算する。

$$x_j = \phi(\mathbf{w}\mathbf{x} + \mathbf{b}) \cdot m_j \quad (3.1)$$

Dropout をもとに DropConnect [91], DropAll [92], Adaptive dropout [93] が提案されている。DropConnect は、図 3.1(b) のように結合重みをランダムに選択し、その結合重みを 0 にする手法である。DropAll は、図 3.1(c) のようにユニットと結合重みを両方選択し、選択したユニットに繋がる結合重みと選択した結合重みの両方を 0 にする手法である。Adaptive dropout は、学習によりユニットの応答値を 0 にする割合を求める方法である。Adaptive dropout は、応答値を 0 にする割合を求める際に、Dropout を行うネットワークと行わないネットワークの 2 つを学習する必要がある。そのため、学習コストが従来の Dropout と比べて 2 倍必要になる。

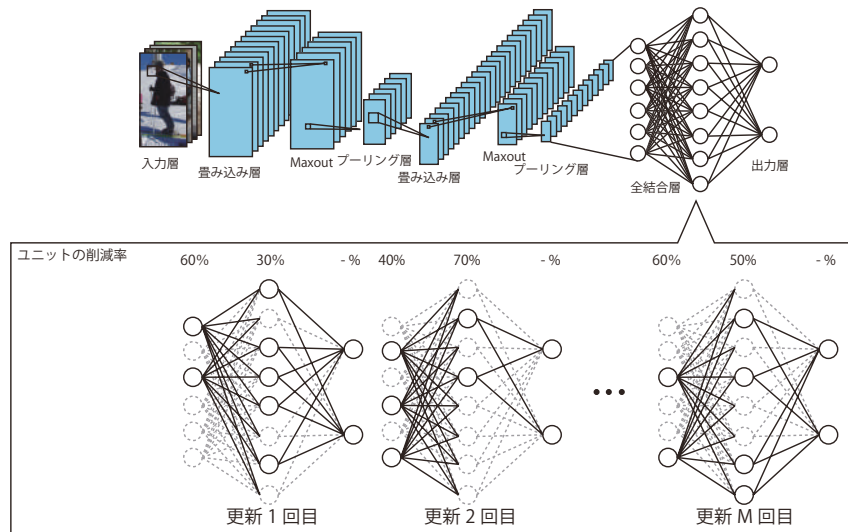


図 3.2: Random Dropout のアルゴリズム

3.2 提案手法

本稿では、Dropout をベースとした 2 つの手法を提案する．1 つ目は、学習時に用いる Random Dropout であり、2 つ目は、識別時に用いる EIN である．以下に 2 つの提案手法の詳細について述べる．

3.2.1 Random Dropout

従来の Dropout では、応答値を 0 にするユニットの割合は、各更新処理で一定である．提案する Random Dropout は、応答値を 0 にするユニットの割合を各更新処理でランダムに変化させる．図 3.2 に Random Dropout のアルゴリズム例を示す．Random Dropout は、更新 1 回目のとき、各層のユニットの削減率を 60% と 30% としている．そして、更新 2 回目では、40% と 70% としており各更新処理で異なっている．これにより、ユニット数が異なっても同じような結果を得られるようにパラメータが更新されるため、汎化性能を向上させることができる．ここで、削減率の範囲は、あらかじめ指定するパラメータとし、全ユニットが削減されないようにする．

3.2.2 Ensemble Inference Networks

EIN は、学習した 1 つのネットワークから構成が異なる全結合層を複数生成し、それらの出力を統合することで、最終的な出力を得る．EIN による識別処理は、図 3.3 のように 1) 畳み込み層およびプーリング層による特徴マップの生成、2) 複数の全結合層による識別、3) 応答値の統合の 3 つから構成されている．

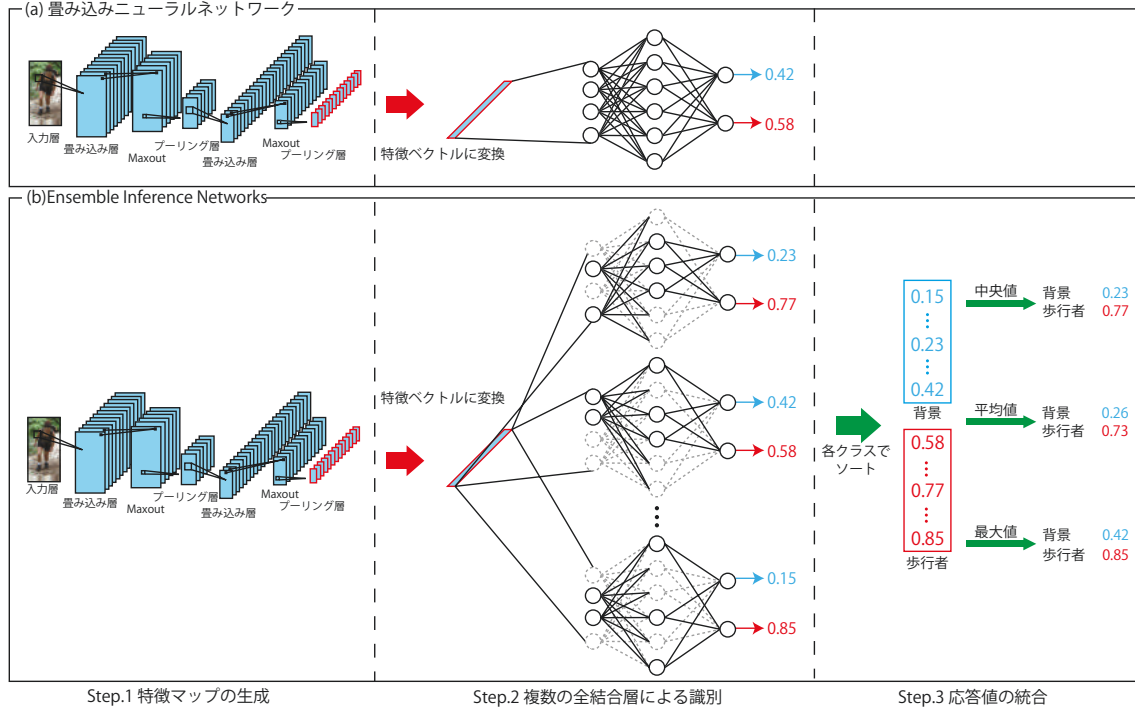


図 3.3: EIN のアルゴリズム

畳み込み層およびプーリング層 EIN の畳み込み層およびプーリング層は、従来の CNN と同じ処理である。EIN は、異なる構成の全結合層を複数生成するが、畳み込み層とプーリング層は共通である。そこで、畳み込み層とプーリング層を通した後の結果を特徴ベクトルとして保持する。これを各生成した全結合層に入力することで、畳み込み層およびプーリング層の処理を繰り返す行う必要がなく、計算コストを削減することができる。

全結合層の生成 従来の CNN による識別時は、図 3.3(a) のように特徴ベクトルを全結合層に入力し、各クラスに対するスコアを取得する。このとき、生成する全結合層は 1 つである。EIN は、図 3.3(b) のように学習したネットワークの全結合層をもとにランダムに選択したユニットの応答値を 0 にする全結合層を N 個用意する。そして、特徴ベクトルを生成した全結合層に入力して各クラスのスコアを求める。これを N 回行うことで、構造の異なる全結合層を通して得られた N 個の各クラスのスコアを求めることができる。

3.2.3 応答値の統合

前節で求めた各全結合層におけるクラス k の確率 $O_{n,k}$ を用いて最終的な出力を決める。まず、各全結合層の確率をクラスごとに格納する。このとき、各クラスに対する応答値の集合を $\mathbf{S}_k$ とする。EIN の各クラスに対する最終的な出力 $\mathbf{S}_k$ は、中央値 $\mathbf{S}_k^{Median}$ や平均値 $\mathbf{S}_k^{Mean}$ 、最大値 $\mathbf{S}_k^{Max}$ から求める。ここでは、どの値の算出方法が適しているかは、問題設定ごとに決めることができるよう

表 3.1: 各データセットにおける CNN の構造

データセット	入力層	1 層目			2 層目			3 層目			4 層目	5 層目	6 層目	出力層
		畳み込み	Maxout	最大値 プーリング	畳み込み	Maxout	最大値 プーリング	畳み込み	Maxout	最大値 プーリング	ユニット数	ユニット数	ユニット数	ソフトマックス
Caltech	108x36x3	40,9x5	2	2x2	64,5x3	2	2x2	32,3x3	2	2x2	1,000	500	100	2
Daimler	96x48x1	100,5x3	2	2x2	80,5x4	2	2x2	70,5x4	2	2x2	1,000	500	100	2
GTSRB	64x64x3	60,5x5	2	2x2	80,5x5	2	2x2	120,4x4	2	2x2	2,000	1,000	300	43

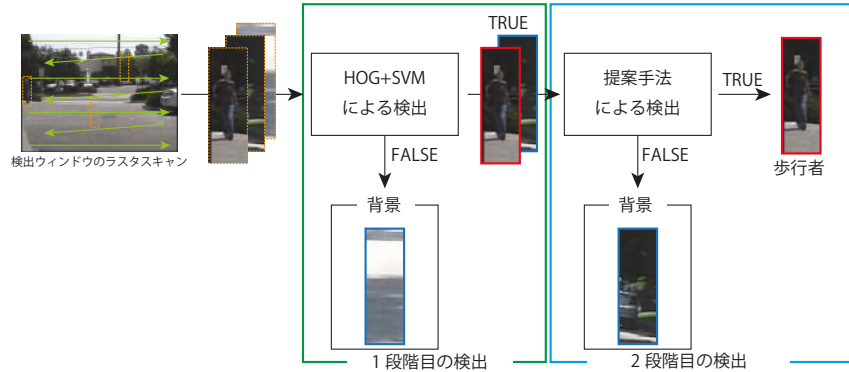


図 3.4: 2 段階処理による歩行者検出

に一般化している。

3.3 提案手法による歩行者検出

本手法では、図 3.4 のように 2 段階の検出構造を用いることで歩行者を検出する。1 段階目で HOG+SVM を用いて歩行者の候補領域の絞り込みを行う。そして、2 段階目で絞り込んだ領域に対して提案手法により最終的な識別を行う。

3.4 評価実験

本稿では、Caltech Pedestrian Dataset[74], Daimler Mono Pedestrian Benchmark Dataset[77], German Traffic Sign Recognition Benchmark (GTSRB) Dataset[94] を用いて、提案手法の性能を評価する。提案手法の評価では、Random Dropout と EIN の効果について検証する。また、性能を評価するために既存手法との比較を行う。

Random Dropout の性能評価では、Dropout の削減率および Random Dropout の削減率の範囲を変化させて比較する。Dropout または Dropout の発展形の手法の削減率は、40% から 90% まで変化させる。Random Dropout の削減率は、削減率の下限値を 0% とし、上限値を 40% から 90% まで変化させる。EIN の性能評価では、生成する全結合層の数を 1 から 33 まで変化させて性能を比較する。各比較実験の結果から、Random Dropout と EIN のパラメータを決定し、従来の歩行者

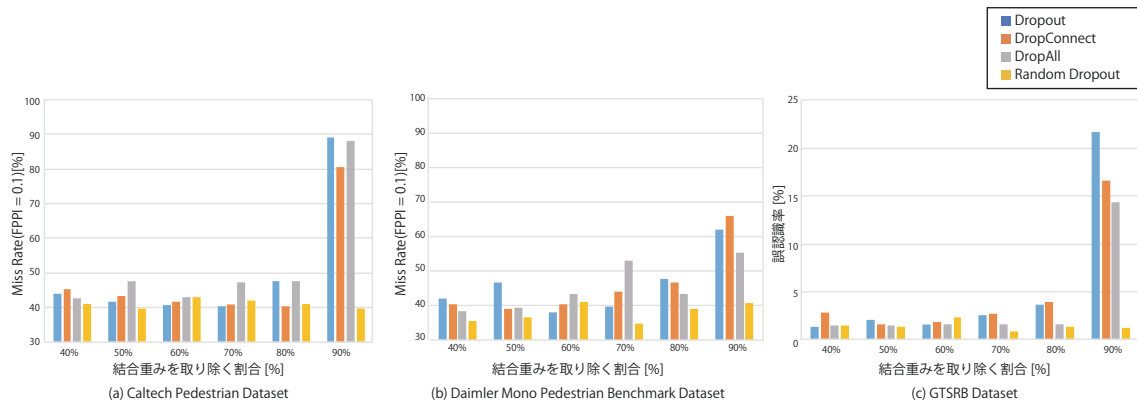


図 3.5: Dropout と関連する手法と Random Dropout の性能比較

検出法と標識認識法を比較することで最終的な性能を評価する． Caltech Pedestrian Dataset では， CNN， HOG[8]， HogLbp[95]， LatSvm-V2[51]， VJ[96]， DBN-Isol[97]， ACF[55]， ACF-Caltech[55]， Pls[98]， FPDW[99]， ChuFtrs[2]， CrossTalk[100]， RandomForest[101]， MultiResC[102]， Roerei[103]， MOCO[104]， Joint Deep[57]， Switchable Deep Network[58] と比較する． Daimler Mono Pedestrian Benchmark Dataset では， CNN， HOG[8]， Shapelet[105]， HogLbp[95]， LatSvm-V2[51]， VJ[96]， RandomForest[101]， MultiFtr[106]， MLS[107] と比較する． また， 標識認識では， Random Forests， Human Performance[108]， Multi-Scale CNNs[109]， Multi-Column Deep Neural Network[11] と比較する．

各データセットに対して使用するネットワークの構造を表 3.1 に示す． 学習パラメータは， ネットワークの更新回数を 50 万回， 学習係数は 0.01 とし， Caltech Pedestrian Dataset と Daimler Mono Pedestrian Benchmark Dataset のミニバッチサイズは 5， GTSRB Dataset のミニバッチサイズは 40 とする． 各実験は， 5 回試行している．

Caltech Pedestrian Dataset の評価は， 学習データセットに対して HOG+SVM で検出した歩行者領域約 4,000 枚と誤検出領域およびランダムに切り出した背景サンプル約 20 万枚を学習に用い， 評価画像 8,723 枚を評価した． 歩行者サンプルは平行移動とスケーリングをランダムに加えて Data Augmentation したサンプル計 10 万枚を学習サンプルとして用いる． Daimler Mono Pedestrian Benchmark Dataset は， 歩行者サンプル約 3 万枚に対して平行移動とスケーリングをランダムに加えて Data Augmentation したサンプル計 13 万枚と， 学習データセットに対して HOG+SVM で検出した際に発生した誤検出領域およびランダムに切り出した背景サンプル約 25 万枚を学習に用い， 評価画像 21,790 枚を評価した．

GTSRB Dataset では， 学習サンプルに 39,000 枚の標識画像と， 評価サンプルに 12,631 枚の標識画像を用いる． GTSRB Dataset では， Data Augmentation による画像の増幅は行わない．

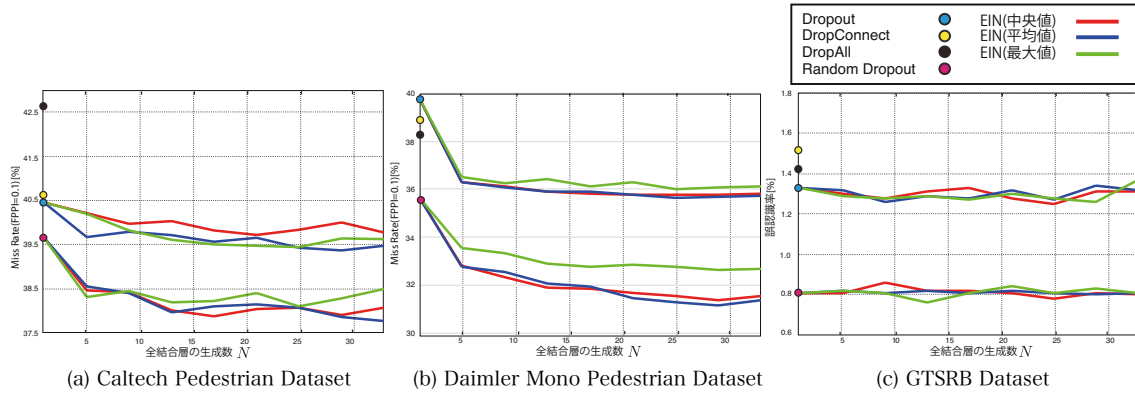


図 3.6: 生成する全結合層数の変化による精度の比較

3.4.1 Random Dropout の性能評価

Random Dropout の効果について、Dropout, DropConnect, DropAll と比較する。各データセットに対して、各手法で学習した CNN の Miss rate および誤認識率を図 3.5(a), (b), (c) に示す。ここで、図 3.5(a), (b) の Miss rate は、ROC カーブを用いて評価した際に、FPPI が 0.1 の時の Miss rate を示している。

図 3.5(a) より、Caltech Pedestrian Dataset において、Dropout は削減率が 70% の時 Miss rate が 40.45% である。一方、DropConnect は削減率が 80% の時に Miss rate が 40.52% となっている。提案手法である Random Dropout は削減率の範囲が 0% から 70% の時に 39.65% の Miss rate であり、提案手法の Miss rate が最も低いことが分かる。図 3.5(b) の Daimler Mono Pedestrian Benchmark Dataset では、Dropout は削減率が 70% の時 Miss rate が 39.81% である。一方、DropAll は削減率が 40% の時に 38.22% となっている。提案手法である Random Dropout は削減率の範囲が 0% から 70% の時に Miss rate が 35.79% であり、提案手法が最も Miss rate が低いことが分かる。

図 3.5(c) の GTSRB Dataset では、Dropout は削減率が 40% の時 1.33% の誤認識率である。一方、DropAll は削減率が 50% の時に 1.39% となっている。提案手法である Random Dropout は削減率の範囲が 0% から 70% の時に 0.81% の誤認識率であり、提案手法の誤認識率が最も低いことが分かる。

3.4.2 Ensemble Inference Networks の性能評価

EIN の全結合層の生成数、最終出力の決定方法を各評価において比較する。図 3.6(a), 図 3.6(b) に歩行者検出データセットにおける、全結合層の生成数に対する Miss rate の変化、図 3.6(c) に標識認識データセットにおける全結合層の生成数に対する誤認識率の変化を示す。なお、図 3.6(a), 図 3.6(b) は FPPI が 0.1 の時の Miss rate である。ここで、EIN の出力値の統合方法を中央値・平均値・最大値の 3 パターンで比較する。全結合層の生成数が 1 の場合は、EIN を用いていない場合であり、図 3.6(a), 図 3.6(b), 図 3.6(c) の縦軸上の各点に相当する。前節での実験結果と同様に、Random Dropout が最も良い性能となっている。

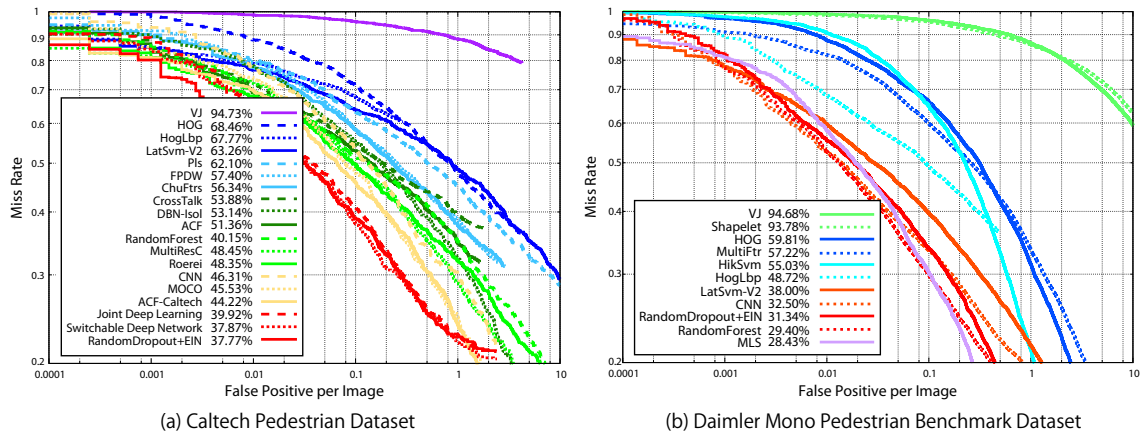


図 3.7: 歩行者検出における提案手法と従来法の比較

表 3.2: GTSRB Dataset における提案手法と従来法の比較

手法	認識率 [%]
Random Dropout + EIN	99.24
Multi-Column Deep Neural Network	99.46
Human Performance	98.84
Multi-Scale CNN	98.31
Random Forests	96.14
LDA on HOG	95.68

次に EIN を用いた場合、図 3.6(a) から、Caltech Pedestrian Dataset では Random Dropout を導入し、EIN の統合方法に平均値を使用して、生成数が 33 のとき Miss rate が 37.77% で最も精度が良い。EIN を用いていない時の Random Dropout より最大で Miss Rate が 1.88% 減少していることがわかる。図 3.6(b) から、Daimler Mono Pedestrian Benchmark Dataset では Random Dropout を導入し、EIN の統合方法に中央値を使用して、生成数が 33 のとき Miss Rate が 31.34% で最も未検出が低い。同じように、EIN を用いていない時の Random Dropout より最大で Miss Rate が 4.45% 減少していることがわかる。図 3.6(a)、図 3.6(b) より、歩行者検出において、EIN の最終出力の方法に中央値か平均値を用いることで、検出性能を向上させることができる。

標識認識のベンチマークである、GTSRB Dataset の結果を図 3.6(c) に示す。図 3.6(c) より、Random Dropout を導入し、EIN の統合方法に最大値を使用して、生成数が 17 のとき Miss Rate が 0.76% で最も良い。また、EIN を導入することで 0.1% 認識性能が向上していることが確認できる。図 3.6(c) より、多クラス認識では最大値を最終出力として用いるのが良い。

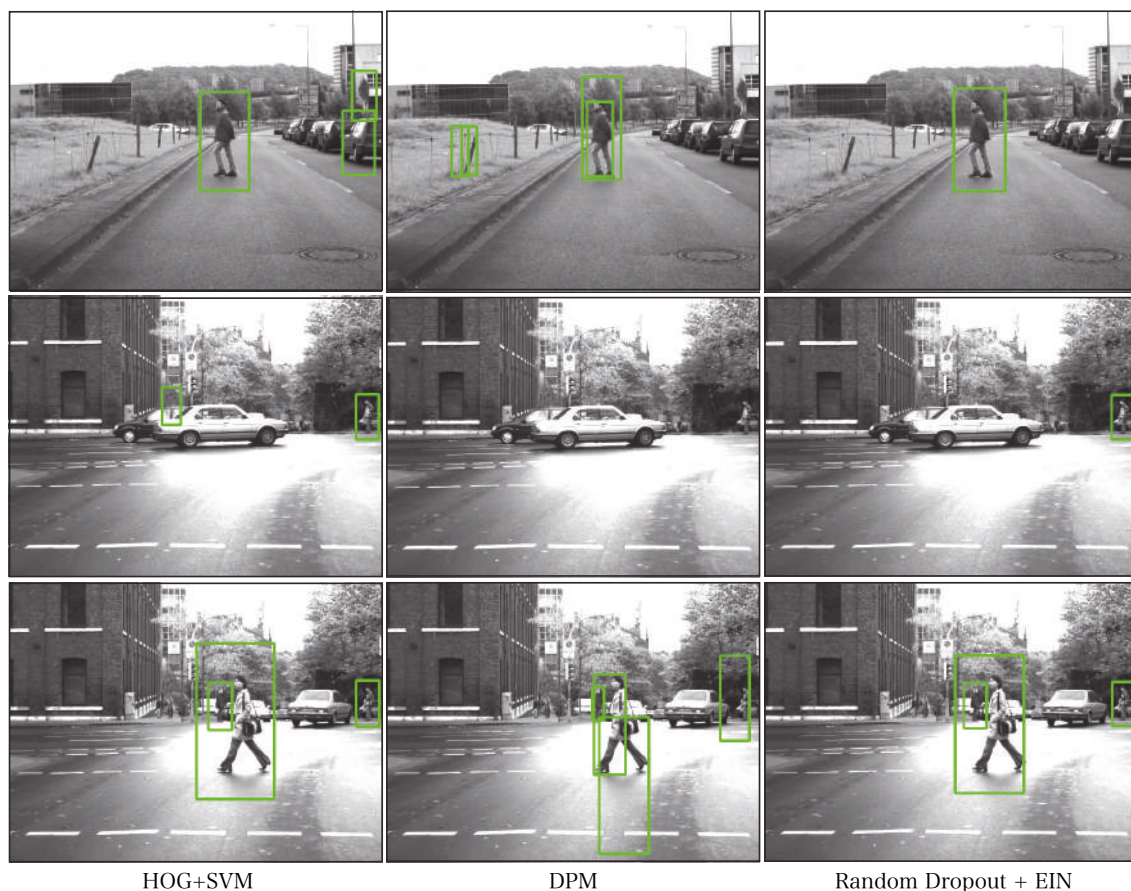


図 3.8: 歩行者検出結果の比較 (Daimler Mono Pedestrian Dataset)

3.4.3 従来法の性能比較

各データセットにおける従来法と提案手法の比較を行う。図 3.7(a) に Caltech Pedestrian Dataset, 図 3.7(b) に Daimler Mono Pedestrian Benchmark Dataset の比較結果を示す。Caltech Pedestrian Dataset では, FPPI が 0.1 のとき従来の CNN に対して提案手法は, Miss rate を 8.54% 改善することができている。Deep Learning による手法において最も高精度な歩行者検出方法である Switchable Deep Learning の Miss rate は 37.87% であり, 提案手法の Miss rate は 37.77% であることから, Deep Learning による手法で最も性能が良いことがわかる。同様に, 図 3.7(b) の Daimler Mono Pedestrian Benchmark Dataset においても, FPPI が 0.1 のとき従来の CNN と提案手法を比べて Miss rate が 1.16% 減少していることが確認できる。

表 3.2 に GTSRB Dataset の比較結果を示す。人間の認識性能が 98.84% であるのに対して, 提案手法では 99.24% であり, 人間の認識性能を上回っていることが確認できる。CNN ベースの手法である Multi-Scale CNN と Multi-Column Deep Neural Network の性能を比較したとき, それぞれ 98.31% と 99.46% であることから, 提案手法はこれらの手法と同等の性能を実現している。

図 3.8 と図 3.9 に Caltech Pedestrian Dataset と Daimler Mono Pedestrian Benchmark Detection の歩



図 3.9: 歩行者検出結果の比較 (Caltech Pedestrian Dataset)

行者検出例を示す。1 列目の歩行者検出例は従来の歩行者検出法である HOG+SVM の検出例であり，2 列目の歩行者検出例は DPM の検出例，3 列目の歩行者検出例は提案手法の検出例である．従来の歩行者検出法では，遠方に存在する歩行者や体型の変化が大きい歩行者に対して検出が困難であり，多くの誤検出が発生している．それに対して，提案手法を導入することで従来の歩行者検出法で検出できなかった歩行者を検出し，誤検出を削減できていることが確認できる．

図 3.10 に，標識認識データセットにおける Dropout を導入した CNN と提案手法の認識結果を示す．ここで，各グラフは上位 3 位までの結果である．また，赤いグラフはターゲットクラスの尤度を示しており，青いグラフは他のクラスの尤度を示している．図 3.10 より，従来の CNN では照明変動や標識の向き，解像度の低下，オクルージョンの発生に対して誤認識が発生している．それに対して，提案手法では照明変動や標識の向き，解像度の低下，オクルージョンの発生が生じた場合においても認識が可能であることが確認できる．

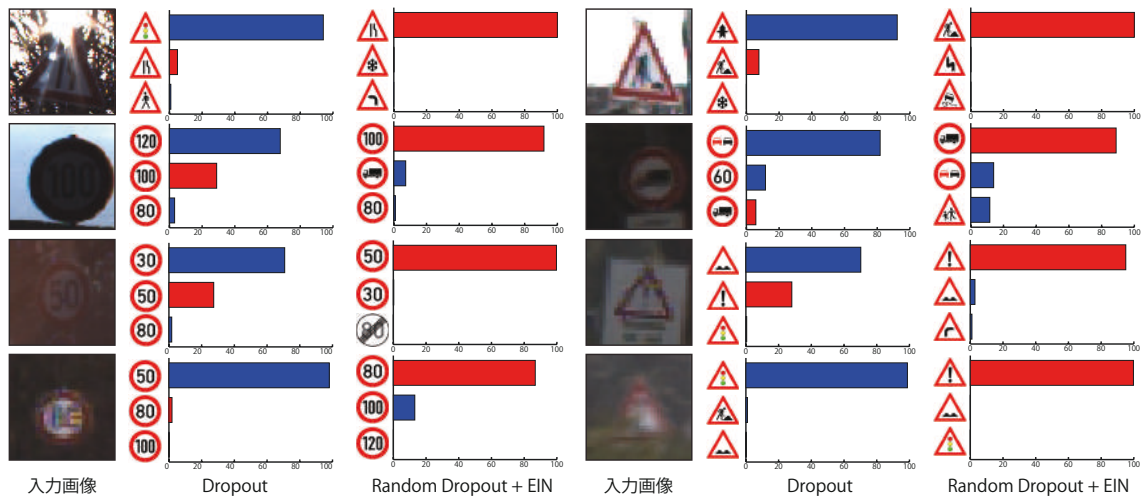


図 3.10: GTSRB Dataset における提案手法と従来の CNN の上位 3 つのスコアの比較

表 3.3: 学習サンプル数に対する Miss rate の推移

学習サンプル数	Dropout	提案手法
3 万枚	47.73%	46.17%
6 万枚	43.13%	43.12%
10 万枚	40.45%	37.77%

3.4.4 考察

学習サンプル数やネットワーク構成による性能を比較する。また、EIN による全結合層の生成数に対する計算コストを比較する。本実験では Caltech Pedestrian Dataset を使用し、Random Dropout の範囲を 0% から 90%, EIN による生成数を 33 とする。また、応答値の統合方法に平均値を使用する。

表 3.3 に学習サンプルに対する提案手法の Miss rate を示す。学習サンプルは、3 万枚から 10 万枚まで変化させている。これより、学習サンプル数を増やすことで Miss rate が低下しており、Data Augmentation により学習サンプル数を増やすことが重要であることがわかる。また、各学習サンプル数において、Dropout を用いて学習した場合よりも Miss rate を低下させることができています。

次に、ネットワーク構成による比較実験を行う。ベースとなるネットワーク構成は、表 3.1 で示した構成であり、Random Dropout と EIN の処理に関する全結合層の層数とユニット数を変化させて比較する。表 3.4 は、全結合層の層の数による性能比較、表 3.5 はユニット数による性能比較の結果である。表 3.4 より、全結合層の層数を 3 層とした場合が最も性能が良い。一方、表 3.5 よりユニット数を少なくすると性能が低下する。ユニット数を各層それぞれ 1000, 500, 100 とした場合が最も性能が良い。

表 3.6 に学習時の処理時間を示す。学習は、GPU: GeForce GTX 980 で行っている。表 3.6 より、提

表 3.4: 全結合層の層数に対する Miss rate の推移

層数	Dropout	提案手法
1 層	45.47%	51.05%
2 層	37.96%	45.52%
3 層	40.45%	37.77%

表 3.5: 全結合層のユニット数に対する Miss rate の推移

各層のユニット数	Dropout	提案手法
500, 250, 50	40.45%	42.66%
1,000, 500, 100	40.49%	37.77%
1,500, 750, 150	40.12%	38.33%

案手法の学習時に用いるネットワーク構造は Dropout とほぼ同じであるため、学習の処理時間に大きな差がないことが確認できる。また、図 3.11 に全結合層の生成数による識別時の処理時間を示す。識別処理は、CPU: Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz で行っている。図 3.11 より、EIN を用いない場合 ($N=1$)、処理時間は 11ms、EIN による生成数を 33 にした場合の処理時間は 37ms である。全結合層の生成数は 33 倍になっているが、畳み込み層とプーリング層の共有化と全結合ユニットの削減により 3 倍程度となっている。これより、学習サンプルおよびネットワーク構成を変えた場合でも提案手法が有効であることがわかる。

3.5 まとめ

本稿では、汎化性能の向上を目的とした 2 つの手法を提案した。Dropout の割合を学習の更新回数ごとにランダムで決定する Random Dropout を学習に導入することで、汎化性能を向上させることができた。EIN による識別処理では、識別過程でランダムに選択した応答値を 0 にし、複数の全結合層を生成する。そして、生成した全結合層の応答値から最終的な出力を算出することで高精度な識別を実現した。今後の課題として、リアルタイムで歩行者検出を実現するために CNN の高速化が挙

表 3.6: 学習時の処理時間

	Dropout	提案手法
学習時間	2 時間 53 分	3 時間 9 分

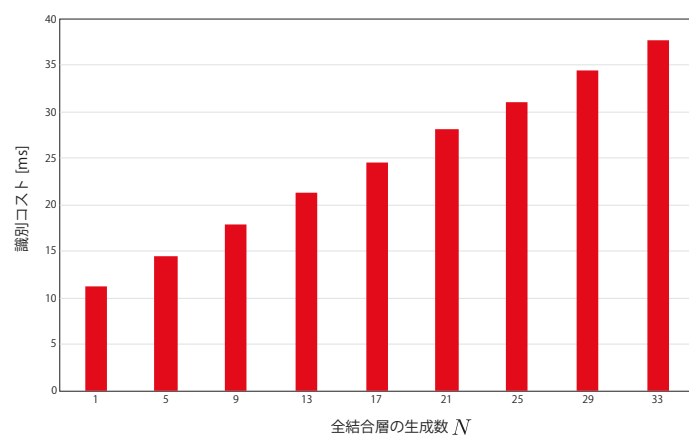


図 3.11: 全結合層の生成数による識別時の処理時間

げられる。

第4章

Multi-task Learning による複数属性の認識

Multi-task Learning は、一つのネットワークで複数のタスクを学習する手法である。Multi-task Learning の学習では、複数タスクのラベルを持つサンプルを使い、複数のタスクの誤差関数を設計することで、ネットワークを学習する。そのため、アンバランスな学習条件が生じやすく、一般的な画像分類モデルより学習が難しい。本章では、Multi-task Learning の学習を困難にする二つの要因について取り組む。

一つ目は、学習時の各タスクにおける学習誤差のばらつきである。Multi-task Learning の学習時は、認識する各タスクに対して誤差関数を割り振る。そのため、学習時における各タスクの学習誤差にばらつきが生じやすい。各タスクにおける学習誤差のばらつきを抑制するために、本研究では重み付き誤差関数を取り入れた Heterogeneous Learning を提案する。重み付き誤差関数を導入した Heterogeneous Learning により、各タスクにおける学習誤差のばらつきを抑制し、学習を安定させる。これにより、顔属性認識の認識精度を向上させる。

二つ目は、学習サンプル数の割合がクラスで偏りやすい点である。学習サンプルの割合が低いクラスは CNN に入力される頻度が少ないため、学習サンプル数の割合が少ないクラスは性能が低下しやすい。また、Multi-task Learning で用いるサンプルには複数のタスクに対してラベルが付与されるため、各タスクにおける学習サンプル数の割合を均等にしながら CNN を学習するのが困難である。この問題を解決するために、本研究では稀少度を導入した Multi-task Learning を提案する。提案手法では、学習データセットの各クラスのサンプル数の割合を用いて稀少度を算出する。そして、算出した稀少度を用いて CNN に入力する学習サンプルを選択することで、サンプル数の割合が少ないクラスの学習サンプルを均等に入力することができる。これにより、サンプル数の少ないクラスの認識精度を向上させる。

はじめに、Multi-task Learning の関連研究として、重み付き誤差関数によるアプローチと、少量の学習サンプルにおける対処について述べる。そして、重み付き誤差関数を導入した Heterogeneous Learning について述べた後に、稀少度を用いた歩行者属性認識について述べる。

4.1 関連研究

4.1.1 重み付き誤差関数

Zhang らの手法は、Multi-task Learning により 1 枚の顔画像にある顔器官点や性別、顔の向きなどの複数の顔属性を単一の CNN で推定している [85]. Zhang らのアプローチは、メインタスクである顔器官点検出の精度を向上させるためにサブタスクの学習誤差関数に対して重み係数を付与する。また、指定した条件に達したサブタスクの学習を停止する Task-wisely Early Stopping を導入しており、メインタスクの学習に寄与しないサブタスクの学習を停止する。重み付き誤差関数は、Multi-task Learning に頻繁に導入されている。Tian らは、歩行者と背景画像に対して属性情報を付与することで、高性能な歩行者検出を実現する Task Assistance CNN (TA-CNN) [60] を提案している。TA-CNN においても、Zhang らの重み付き誤差関数を導入しており、歩行者属性認識を歩行者検出の精度向上へ役立てている。Zhang らの提案する重み付き誤差関数は認識する全てのタスクの精度を向上するためでなく、特定のメインのタスクの精度を向上させるアプローチである。Zhang らの場合は顔器官点検出の精度のみ向上させ、Tian らの場合は歩行者検出の精度のみ向上させる。

他にも、Li らは顔や腕等の人体のパーツと関節点を Heterogeneous Learning により検出する際に、重み付き誤差関数を導入して検出精度を向上させている [110]. Kendall らは、セマンティックセグメンテーションとインスタンスセグメンテーション、深度推定を単一のネットワークで同時に学習する際に、重み付き誤差関数を導入することで全てのタスクの精度を向上させている [111]. Kendall らの重みは、セグメンテーションのクロスエントロピー誤差と深度推定の L1 ノルム誤差の誤差の差分を緩和させるような重みを導入している。

4.1.2 不均衡なデータセットに対するアプローチ

Multi-task Learning におけるデータセット内のサンプル数の割合が不均衡なデータセットを改善する研究は活発に行われている [112, 113, 114, 115]. 一般的なアプローチとして、重み付き誤差関数を用いたアプローチが提案されている [113, 114, 115]. これらの研究では、サンプル数の割合を誤差の重み値として採用したり、少量のサンプルのクラスと過剰のサンプルのクラスを定義して重み値を算出している。一方で、Huang らのアプローチは Triplet Loss を拡張した Quintuplet Learning を提案しており、サンプル数の多いクラスと少量のサンプル数のクラスをペアで学習することで、サンプル数に依存しない学習方法を提案している [112].

4.2 重み付き誤差関数を導入した Heterogeneous Learning による顔属性認識

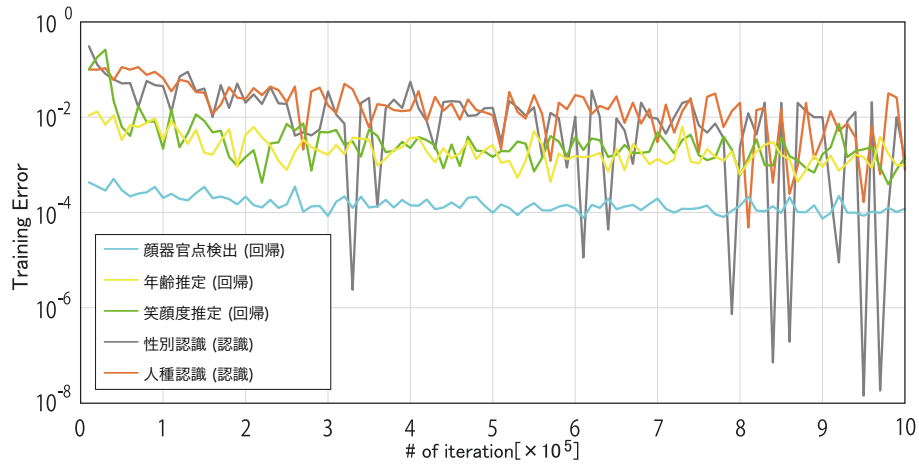
顔画像に含まれる顔器官点や性別、年齢などの顔属性情報は、マーケティング戦略や顧客へのサービスの質を向上させるために応用されている。マーケティング戦略では、画像から顧客の年齢や性別を取得することで、その顧客の世代や性別に合った商品を宣伝するシステム等に応用されている。特に、顔属性認識では顔を認識した後に顔を編集するアプリケーションやサービス等に、活発に応用されている。

一般的に、顔器官点や性別、年齢などの顔属性を識別するには、各属性に対して識別器を構築する必要がある。顔器官点検出には、Active Appearance Model (AAM) [116] や Conditional Regression Forest (CRF) [117] が用いられる。また、年齢推定や性別認識では、検出した顔器官点の位置や context, Local Binary Pattern (LBP) 特徴量を入力し、SVM や決定木で学習、識別する [118, 119]。また、CNN を一般物体認識に適応し [6]、高い性能を実現したことから、顔器官点検出 [87] や年齢推定 [88]、性別認識 [89] などにおいても CNN を用いたアプローチが提案されている。

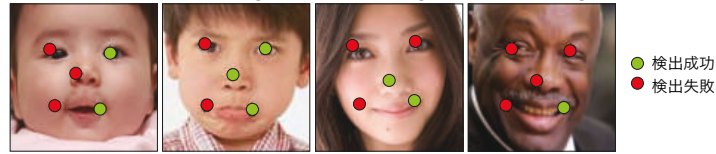
顔器官点検出や年齢推定、性別認識などの複数の属性を推定する場合、推定する属性の数だけ識別器を構築する必要がある。推定する顔属性の数に伴い、学習、識別コストが増加することから、実用化の面で非現実的である。複数の属性を単一の識別器で学習、識別するアプローチとして、Heterogeneous Learning がある。Heterogeneous Learning は Multi-task Learning の一種であり、複数の回帰推定を行うタスク (回帰推定タスク) と認識を行うタスク (認識タスク) を単一のネットワークで学習、識別する方法である。Heterogeneous Learning は、出力層の各ユニットに各タスクに対する識別結果を出力するように学習することで、複数タスクの識別を実現している。本稿では、Heterogeneous Learning を導入した単一の CNN を用いて、顔器官点検出、性別認識、年齢推定、人種推定、笑顔度推定の 5 タスクを識別する。

Heterogeneous Learning を CNN に導入して学習したとき、各タスクの学習誤差のばらつきが発生し、学習に悪影響を与える場合がある。Heterogeneous Learning で学習する場合に、一般的に回帰推定タスクでは L2 ノルム 2 乗誤差を使用し、認識タスクではクロスエントロピー誤差関数を使用する。L2 ノルム 2 乗誤差とクロスエントロピー誤差関数は、取り得る誤差の範囲が大きく異なる。そのため、認識タスクの誤差関数を L2 ノルム 2 乗誤差関数に変更することで、回帰推定タスクと認識タスクで取り得る誤差の範囲が同一になる。図 4.1(a) に、誤差関数を L2 ノルム 2 乗誤差に統一し、Heterogeneous Learning を導入した CNN で 5 タスクを学習した際の学習誤差を示す。回帰推定タスクのラベル値は 0 から 1 の連続値であるのに対し、認識タスクのラベル値は 0 か 1 の離散値である。そのため、認識タスクでは回帰推定タスクに比べ、学習誤差が大きくなるため、図 4.1(a) のように学習誤差の収束値に大きな差が生じる。図 4.1(a) の学習誤差のばらつきが原因で、図 4.1(b) のように学習誤差の収束値が最も低い顔器官点検出の性能が低下しやすくなる。

そこで、本研究では重み付き Heterogeneous Learning を導入した CNN を提案する。まず、複数存在する識別タスクから、基準となるタスク (基準タスク) を選択する。このとき、基準タスク以外のタスクをサブタスクと定義する。そして、学習時のサブタスクの学習誤差に対して重みを与える。各



(a) 従来のHeterogeneous Learningによる学習誤差



(b) 顔器官点検出の検出結果

図 4.1: Heterogeneous Learning を導入した CNN の学習誤差と顔器官点検出結果

サブタスクに対して重みを与えることで、学習誤差の収束値を調節し、突発的に変化する学習誤差を抑制する。重み付き誤差関数により学習を安定することで、複数の顔属性認識の性能を向上させる。

4.2.1 提案手法

従来の Heterogeneous Learning は、各タスクに対して平等に学習誤差を算出する。そのため、回帰推定タスクと認識タスクのラベル値の違い等が原因で、学習誤差のばらつきが生じる。提案手法では、基準タスク以外のサブタスクに対し、学習誤差 L_t に重みを与える。各サブタスクの誤差関数に重みを与えることで、各サブタスクの学習誤差のばらつきを抑制し、学習を安定させる。

■ 単一タスクの CNN の学習

初めに、図 4.2 のように基準タスクを選択するために、各タスクを個別の CNN で学習する。個別で学習した CNN の学習誤差は、Heterogeneous Learning の学習誤差と異なり、タスク間の学習誤差が干渉していない。獲得した学習誤差は、基準タスクを選択する際の基準として使用する。

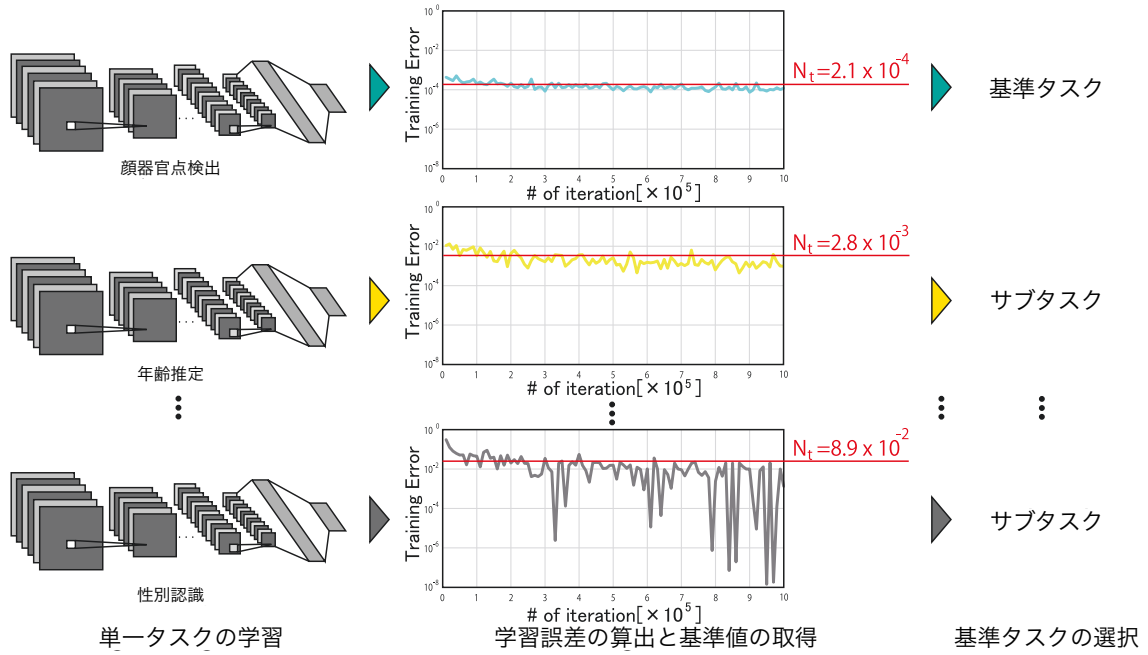


図 4.2: 基準タスクの選択

■ 誤差関数に与える重みの算出

そして、各タスクを個別の CNN で学習した際の学習誤差から、正規分布を用いて各タスク t の学習誤差の基準値 N_t を式 (4.1) から求める。正規分布におけるタスク t の学習誤差は、平均 μ と標準偏差 σ の 3 倍の和の区間に 99.7% 含まれる。この区間外は、突発的に変動した学習誤差であり、この区間外を無視して基準値 N_t を求めることで、突発的な学習誤差の影響を受けにくい基準値を求めることができる。そのため、式 (4.1) のように平均 μ と標準偏差 σ の 3 倍の和を基準値 N_t として用いる。基準値 N_t は、認識する全てのタスクに対して求める。

$$N_t = \mu + 3\sigma \quad (4.1)$$

基準値を求めた後に、基準タスクを選択する。本稿では、基準値 N_t が最も低いタスク t を基準タスクとして選択する。そのため、5 タスクの中で最も基準値の低い、顔器官点検出タスクを基準タスクとし、それ以外のタスクをサブタスクとして取り扱う。基準タスクとサブタスクを選択した後に、各サブタスクに与える重み α_t を求める。各サブタスクに与える重み α_t は、顔器官点検出の基準値 N_f と各サブタスクの基準値 N_t を用いて、式 (4.1) のように求める。

$$\alpha_t = \frac{N_f}{N_t} \quad (4.2)$$

表 4.1: Heterogeneous Learning を導入した CNN の構造

Input	画像サイズ	100×100
1 層目	カーネルサイズ	$9 \times 9 \times 16$
	Maxout	2
	最大プーリング	2×2
2 層目	カーネルサイズ	$9 \times 9 \times 32$
	Maxout	2
	最大プーリング	2×2
3 層目	カーネルサイズ	$9 \times 9 \times 64$
	Maxout	2
	最大プーリング	2×2
4 層目	シグモイド関数	200 (Dropout:50%)
出力層		17

■ 重み付き誤差関数を導入した Heterogeneous Learning の学習

算出した各サブタスクの重みを，式 (4.3) のように与える．各タスクでは L2 ノルム 2 乗誤差を用い，左の項がメインタスクの学習誤差関数，右の項がサブタスクの学習誤差関数を示している．

$$L = \frac{1}{M} \sum_{m=1}^M \left(\| \mathbf{y}_{fd,m} - \mathbf{o}_{fd,m} \|_2^2 + \sum_{t=1, t \neq f}^{T-1} \alpha_t \| \mathbf{y}_{t,m} - \mathbf{o}_{t,m} \|_2^2 \right) \quad (4.3)$$

ここで， $\mathbf{y}_{fd,m}$ と $\mathbf{o}_{fd,m}$ は，顔器官点検出のラベル値と出力値を示している．また，各サブタスク t の学習誤差に与えられる重み α_t は，各更新回数で一定である．

4.2.2 評価実験

提案手法の有効性を示すために，各識別タスクを単一タスクの DCNN と従来の Heterogeneous Learning，提案手法の識別精度を比較する．本実験では，顔器官点検出，性別認識，年齢推定，人種認識，笑顔度推定を行う．顔器官点検出では右目，左目，鼻，口の右端，口の左端の計 5 点の位置を回帰により推定し，笑顔度推定は 0 から 0.99 までの値を回帰により推定する．また，人種認識では，アジア人，白人，黒人の 3 種類を判別する．実験では，表 4.1 の畳み込み層が 3 層，全結合層が 1 層の CNN を使用する．学習パラメータは，学習係数を 0.1，ミニバッチのサイズを 10，学習の更新回数を 100 万回とする．データセットには，Web から収集した 53,663 枚の顔画像に対して 5 つの顔属性のラベルを付与する．ここで，学習サンプル 42,663 枚，評価サンプル 11,000 枚とする．入力するサンプルは 100×100 画素のグレースケール画像である．笑顔度のラベルは，複数人の目視による笑顔度の評価値を平均したものを，正解ラベルとする．年齢は，本データセットに 0 歳から 66 歳までの人物の顔画像が存在しており，その人物の年齢をラベルとして使用する．

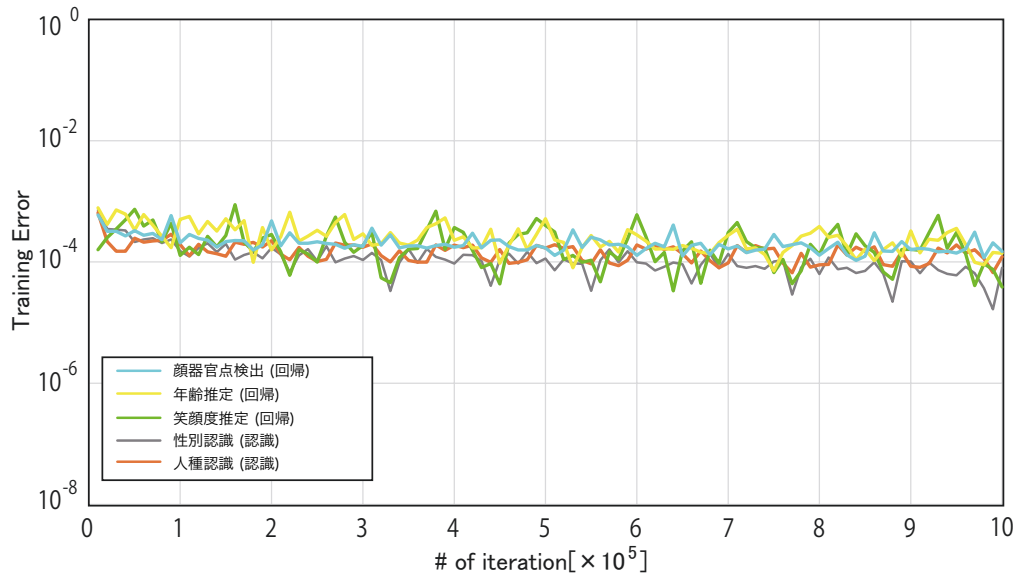


図 4.3: 従来法と提案手法の学習誤差の比較

実験で使用する従来の Heterogeneous Learning と提案手法は，表 4.1 と同様である．また，各タスクを個別の CNN で学習する場合，畳み込み層及び全結合層の構造は同様であり，出力層のみタスクに対応したユニット数となっている．

■ 学習誤差の比較

図 4.3 に，提案手法の各タスクの学習誤差を示す．従来法の Heterogeneous Learning の各タスクにおける学習誤差は，図 4.1 のように各タスクの収束値が異なっており，認識タスクの学習誤差は突発的に学習誤差が変動している．一方，提案手法の学習誤差は，各タスクの学習誤差の収束値が統一されており，突発的に学習誤差が変動していないことが確認できる．これらの結果から，各タスクの学習誤差を算出する際に重みを付与することで，学習誤差の収束値を調節し，突発的な学習誤差を抑制することで，安定した学習をしていることがわかる．また，提案手法の各タスクの学習誤差は，全体的な学習誤差の収束値が従来の Heterogeneous Learning より低いことがわかる．

■ 提案手法の精度の比較

図 4.4 に，単一タスクで学習した DCNN，従来の Heterogeneous Learning，提案手法の識別精度を示す．従来の Heterogeneous Learning は，単一タスクで識別した DCNN と比較したときに，回帰推定タスクの識別精度が大きく低下している．特に，顔器官点検出の精度が大きく低下している．これは，顔器官点検出の学習誤差の収束値が最も小さいため，他のタスクの学習誤差による影響を受けやすくなり，顔器官点検出の精度が大きく低下したと考えられる．次に，提案手法と従来の Heterogeneous Learning を比較したとき，全体の識別精度が約 5% 向上し，単一タスクで識別した CNN と同等以上

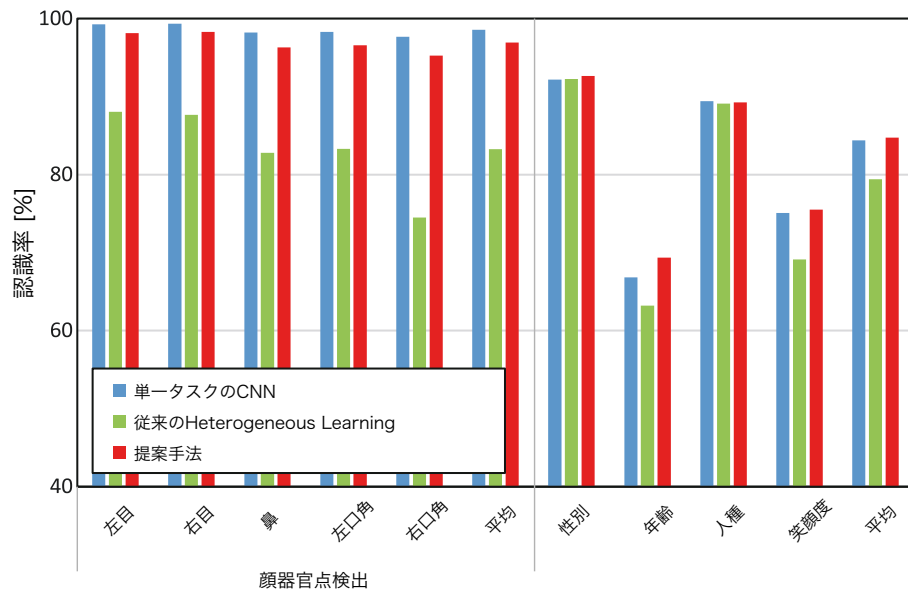


図 4.4: 従来法と提案手法の認識精度の比較

の精度であることが確認できる．特に，顔器官点検出において，約 14%精度が向上した．提案手法の導入により，Heterogeneous Learning の学習が安定したことで，顔器官点検出に有効な特徴量を獲得しやすくなったため，性能が向上したと考えられる．

図 4.5 に，従来の Heterogeneous Learning と提案手法の識別結果を示す．ここで，1，3 列目は従来の Heterogeneous Learning の識別結果を示しており，2，4 列目は提案手法の識別結果を示している．図 4.5 より，従来の Heterogeneous Learning は回帰推定タスクの性能が低いことがわかる．しかし，提案手法では顔の向きや表情の変化に対して，頑健に識別が可能であることが確認できる．また，1 枚あたりの処理時間は，Intel Core i7-4790(3.60GHz)，メモリ 8GB の環境で行った場合，提案手法は約 22ms で 5 タスクの識別ができた．そして，画像の右に記しているのは，従来の Heterogeneous Learning 及び提案手法の認識結果であり，上から性別，年齢，人種，笑顔度である．認識結果で，赤字で記してある部分は，誤認識している結果である．画像上にある緑点は，検出した顔器官点である．

■ 学習誤差関数の統一の効果

従来の Heterogeneous Learning では，各タスクに対して学習誤差関数を設計する際に，回帰推定タスクに L2 ノルム 2 乗関数を用い，認識タスクにクロスエントロピー誤差関数を用いる．しかし，L2 ノルム 2 乗誤差関数とクロスエントロピー誤差関数では，取り得る誤差の範囲が大きく異なる．そのため，クロスエントロピー誤差関数の大きな誤差が回帰推定タスクの学習に影響を与え，性能を下げってしまう．本稿では，この問題に対処するために，回帰推定タスクと認識タスクで使用する学習

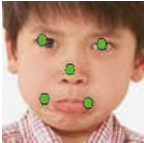
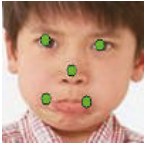
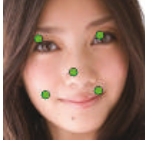
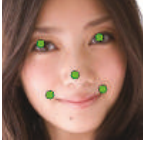
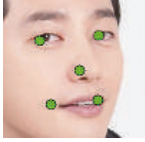
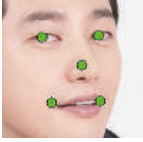
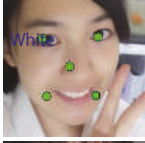
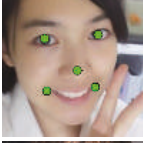
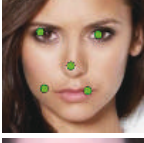
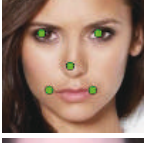
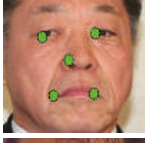
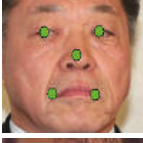
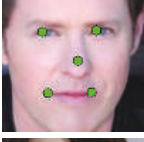
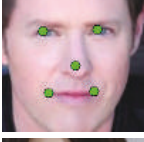
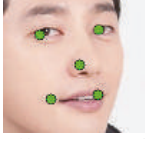
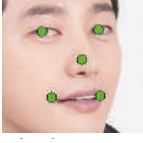
	Male 12 year Asian 4%		Male 10 year Asian 1%		Male 21 year Asian 2%		Male 8 year Asian 7%
	Male 59 year Black 91%		Male 59 year Black 91%		Female 27 year Asian 90%		Female 22 year Asian 95%
	Female 40 year Asian 33%		Female 40 year Asian 25%		Male 45 year Asian 42%		Male 37 year Asian 69%
	Male 49 year Asian 5%		Male 50 year Asian 7%		Female 19 year Asian 82%		Female 22 year Asian 86%
	Male 5 year Asian 21%		Male 1 year Asian 15%		Male 18 year Asian 18%		Male 22 year Asian 29%
	Male 54 year White 2%		Male 60 year White 7%		Female 30 year Asian 27%		Female 25 year Asian 83%
	Male 38 year Asian 59%		Male 31 year Asian 63%		Male 27 year Asian 20%		Male 33 year Asian 17%
	Male 50 year Asian 15%		Male 41 year Asian 30%		Female 30 year White 79%		Female 30 year White 79%
	Female 13 year White 6%		Female 20 year White 6%		Male 35 year Asian 4%		Male 41 year Asian 1%
	Male 32 year White 4%		Male 37 year White 5%		Male 35 year Asian 66%		Male 29 year Asian 84%
	Female 25 year Asian 45%		Female 25 year Asian 65%		Male 18 year Asian 18%		Male 22 year Asian 29%
Heterogeneous Learning		提案手法		Heterogeneous Learning		提案手法	

図 4.5: 認識結果の可視化

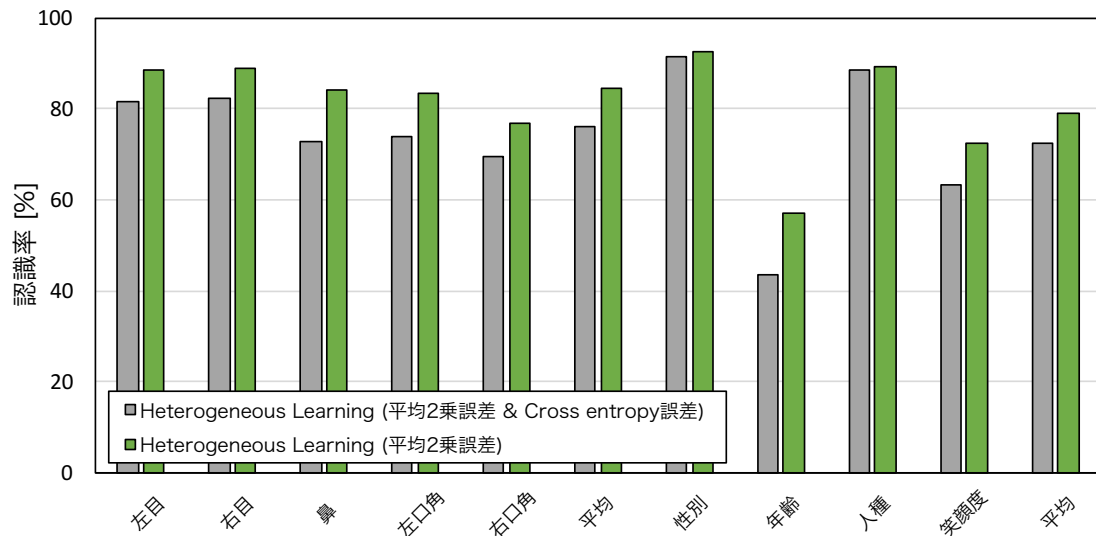


図 4.6: 誤差関数統一による推定精度の比較

誤差関数を L2 ノルム 2 乗誤差関数に統一している．本章では，学習誤差関数の統一による効果を，評価実験により評価する．

従来の Heterogeneous Learning と誤差関数を統一した Heterogeneous Learning の学習誤差を比較した際の結果を，図 4.6 に示す．図 4.6 より，学習誤差関数の統一による誤差の範囲の統一を行うことで，回帰推定タスクと認識タスク間の誤差の差を抑制でき，識別精度が向上したと考えられる．

■ しきい値を変更した際の回帰タスクの精度比較

回帰タスクの評価実験ではしきい値により，ラベルと出力値の誤差がしきい値以下の場合に識別成功，しきい値以上の場合に識別失敗とし，最終的な識別精度を算出していた．しかし，しきい値は固定して評価しているため，推定結果とラベルとの差がどの程度近いのかまでは評価できていない．そこで，評価時のしきい値を変化させることで，推定値とラベルの距離を評価し，各手法の性能を評価する．

図 4.7(a) に，顔器官点検出における閾値を変化させた際の結果を示す．図 4.7(a) より，単一タスクよりは性能は劣っているが，従来の Heterogeneous Learning より大きく性能が向上していることが確認できる．図 4.7(b) に笑顔度推定のしきい値を変化させた際の結果，図 4.7(c) に年齢推定のしきい値を変化させた際の結果を示す．図 4.7(b)，図 4.7(c) より，サブタスクにおける回帰推定タスクでは，提案手法が最も性能が高いことが確認できる．笑顔度推定と年齢推定の認識精度が単一の CNN より向上した原因として，提案手法で顔器官点検出の性能が大きく向上したことが考えられる．Heterogeneous Learning の顔器官点検出の精度を向上させたことで，顔のパーツに着目しやすくなり，推定に有効な特徴量を学習により獲得しやすくなったため，認識精度が向上したと考えられる．

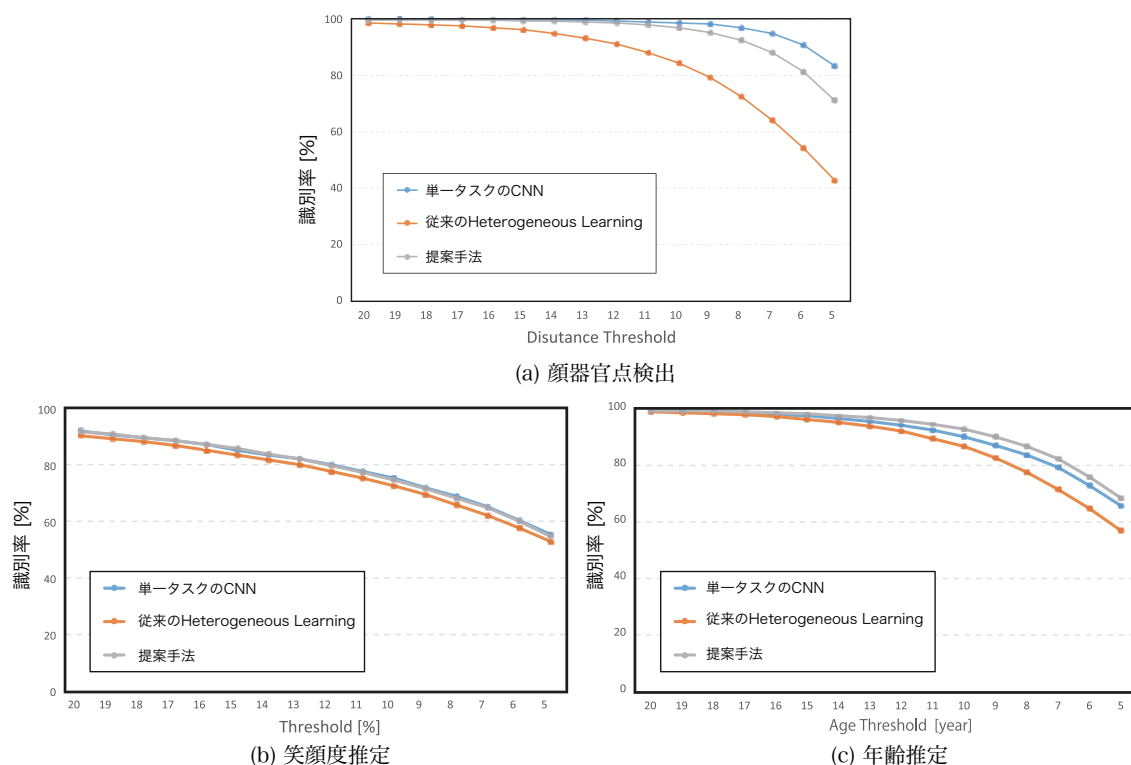


図 4.7: しきい値を変化させた際の分類精度

■ カーネルと特徴マップの可視化結果

CNN のカーネル及び、顔画像を入力した際に得られる特徴マップを可視化することで、CNN の挙動の変化を確認する。ここで、可視化をするのは CNN の畳み込み層 1 層目である。図 4.5(a) に従来の Heterogeneous Learning の可視化結果、図 4.5(b) に提案手法の可視化結果を示す。図 4.5(a) の従来の Heterogeneous Learning のカーネルは、ノイズが少なくコントラストがはっきりしたフィルタが得られている。しかし、特徴マップを見た際に、目や口等の器官のエッジが所々の特徴マップで弱くなっていることがわかる。それに対して、図 4.5(a) の提案手法の重みフィルタはややノイズが入っているが、目や口等の器官が強く反応していることがわかる。目や口等の器官が強く反応したことで、顔器官点検出や笑顔度推定、年齢推定の性能が向上したのではないかとと思われる。

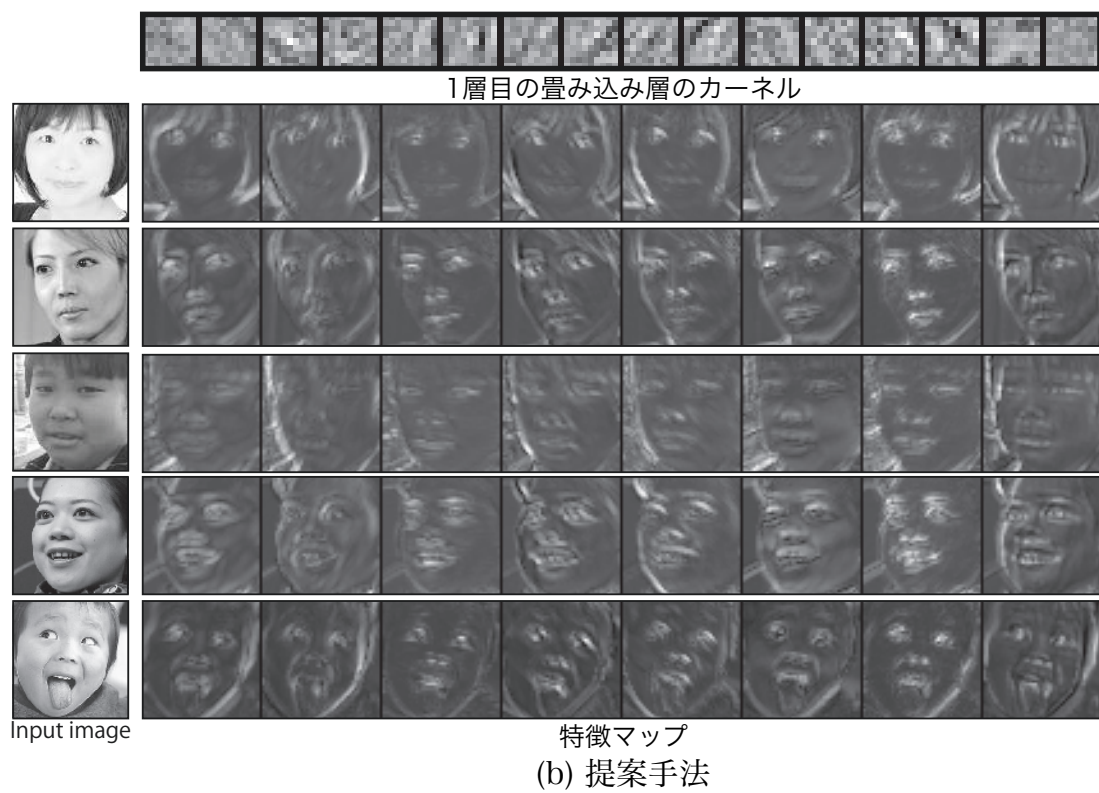
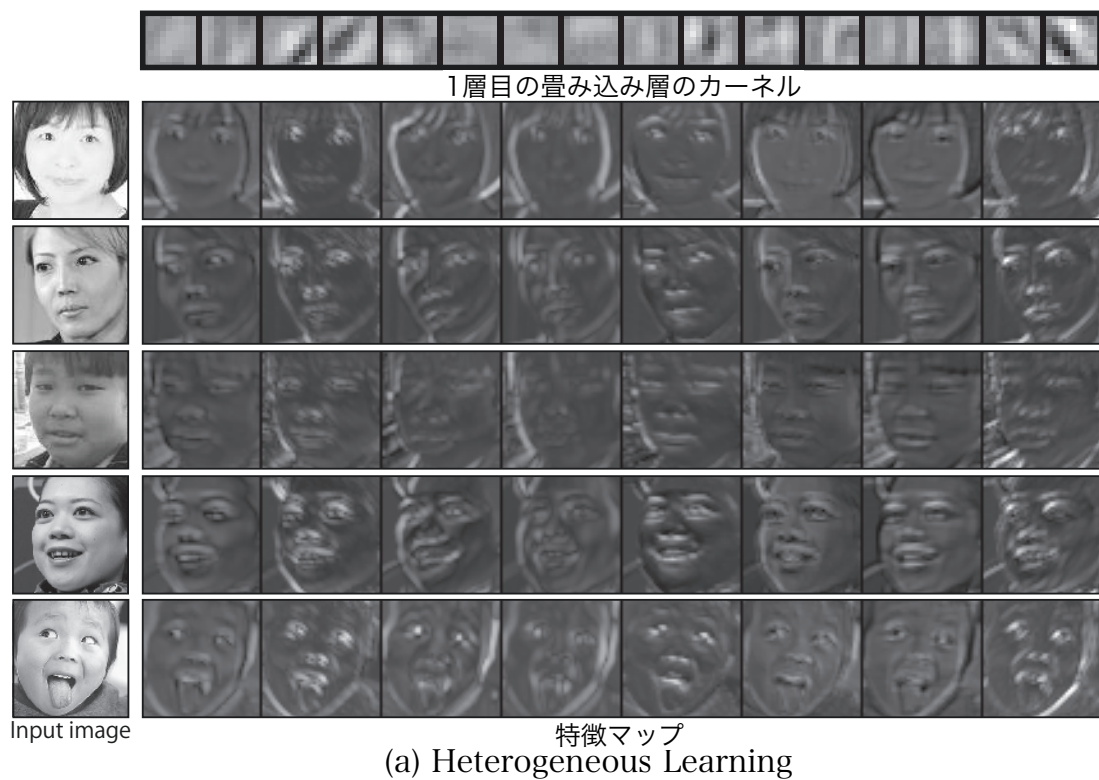


図 4.8: カーネルと特徴マップの可視化結果

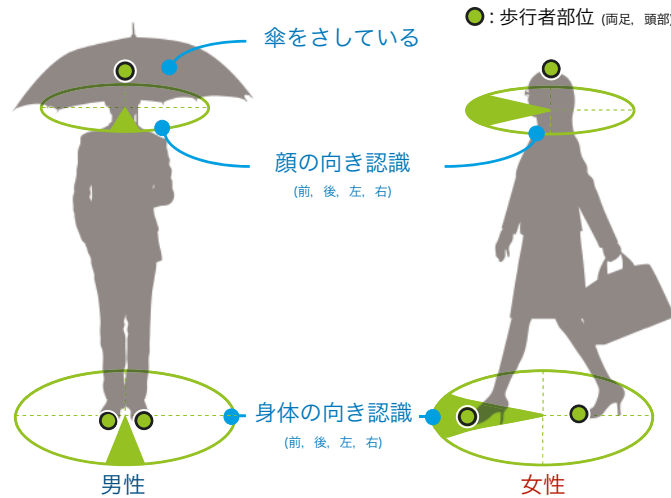


図 4.9: Heterogeneous Learning を導入した CNN で認識する歩行者属性

4.3 稀少度を用いた歩行者属性認識

ADAS の質を向上するために、歩行者属性認識技術が用いられている。一般的に用いられている歩行者の属性情報として、歩行者の進行方向や視線の向きなどが挙げられる。歩行者の進行方向や視線の向きを認識することで、歩行者の行動を予測し、道路への飛び出しを即座に認知できる。また、歩行者が傘をさした際に視界が妨げられるため、雨天時の歩行者と自動車の接触事故率が上昇する。これらの交通事故の原因を抑制するために、歩行者の属性認識が重要となる。本稿では、Heterogeneous Learning を導入した CNN を用いて、図 4.9 に示す歩行者部位検出、身体の向き、顔の向き、性別、傘をさしているかの 5 つのタスクを識別する。

歩行者の向きや部位などの複数タスクを識別する場合、識別するタスク数だけ識別器を構築する必要がある。そのため、識別するタスク数に伴って学習、識別コストが増加することから、実用化の面で非現実的である。複数タスクを識別する方法として、Heterogeneous Learning [120] がある。Heterogeneous Learning は、ニューラルネットワークの学習法の 1 つで、出力層の各ユニットに各タスクに対する識別結果を出力するように学習することで複数タスクを識別する。

Heterogeneous Learning で用いるサンプルは、1 つのサンプルに複数のタスクのラベルが付与されているため、各クラスの学習サンプル数の割合に偏りが発生しやすい。ここで、各タスクにおける各クラスの学習サンプル数の割合と、Heterogeneous Learning を導入した CNN による各タスクの識別精度を図 4.14 に示す。図 4.14 より、認識率が低いクラスは学習サンプル数の割合が低いことがわかる。原因として、サンプル数が少ないクラスに属している学習サンプルは、学習時に CNN へ入力される確率が低いため、識別性能が低下しやすいと考えられる。ここで、本研究ではサンプル数が少ないクラスに属している学習サンプルをレアサンプル、サンプル数が多いクラスに属している学習サンプルをコモンサンプルとする。

これらの課題を解決するために、従来の Heterogeneous Learning を導入した CNN が学習過程において、ランダムに選択した学習サンプルを入力している点に着目する。従来の Heterogeneous Learning

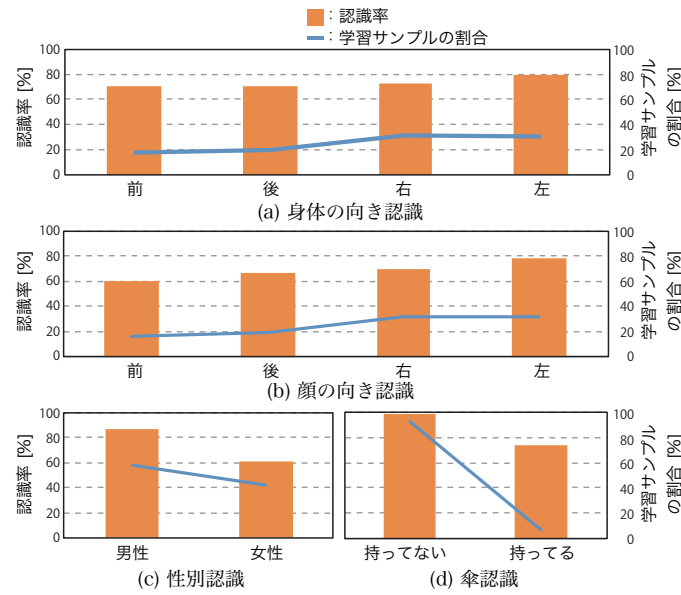


図 4.10: 各歩行者属性における学習サンプル数と認識精度の関係

を導入した CNN では、学習データセットに対して Data Augmentation により学習サンプルを増幅する。そして、増幅した学習サンプルから複数のミニバッチを作成し、CNN へ入力する。一般的な学習では、全てのサンプルを均等に CNN へ入力されるため、サンプル数の割合が低いクラスに属している学習サンプルが CNN へ入力されない。本稿では、この問題を解決するために、学習サンプルに対して稀少度を付与する。稀少度を付与することで、レアサンプルを適度に含んだミニバッチを作成する。ここで、学習サンプルに付与する稀少度は、サンプルが属しているクラスのサンプル数の割合から算出する。稀少度を付与した後に、レアサンプルを Data Augmentation により学習サンプルを数多く増幅し、コモンサンプルには Data Augmentation による学習サンプルの増幅を抑えるように学習サンプルを増幅する。そして、増幅した学習サンプル群から CNN へ入力するミニバッチを決定する。学習で使用するミニバッチを決定するために、学習サンプル群から使用するミニバッチの候補を複数作成し、ミニバッチの候補からレアサンプルを適度に含んでいるミニバッチを学習で使用する。ミニバッチを作成する際に、学習サンプルに対して稀少度を付与し、レアサンプルを適度に含んだミニバッチを作成して CNN へ入力する。これらの一連の処理により、学習サンプルの少ないクラスの識別精度を向上させる。

4.3.1 提案手法

学習サンプルが少ないクラスの識別精度を向上するために、提案手法では学習サンプルに対して稀少度を付与し、レアサンプルを適度に含んだミニバッチを作成して CNN へ入力する。従来の CNN のミニバッチ学習法は、図 4.11 のように学習サンプルをランダムに選択して CNN に直接入力するため、レアサンプルが CNN へ入力される回数が低い。そのため、学習サンプル数の少ないクラスの

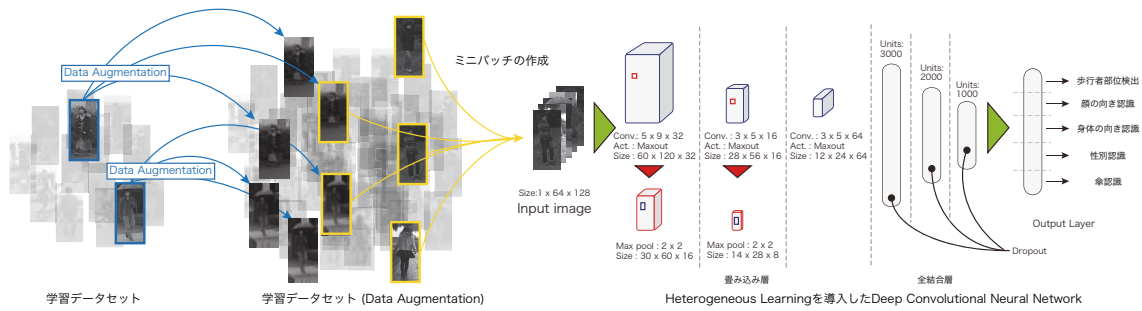


図 4.11: 従来の Heterogeneous Learning による歩行者属性認識

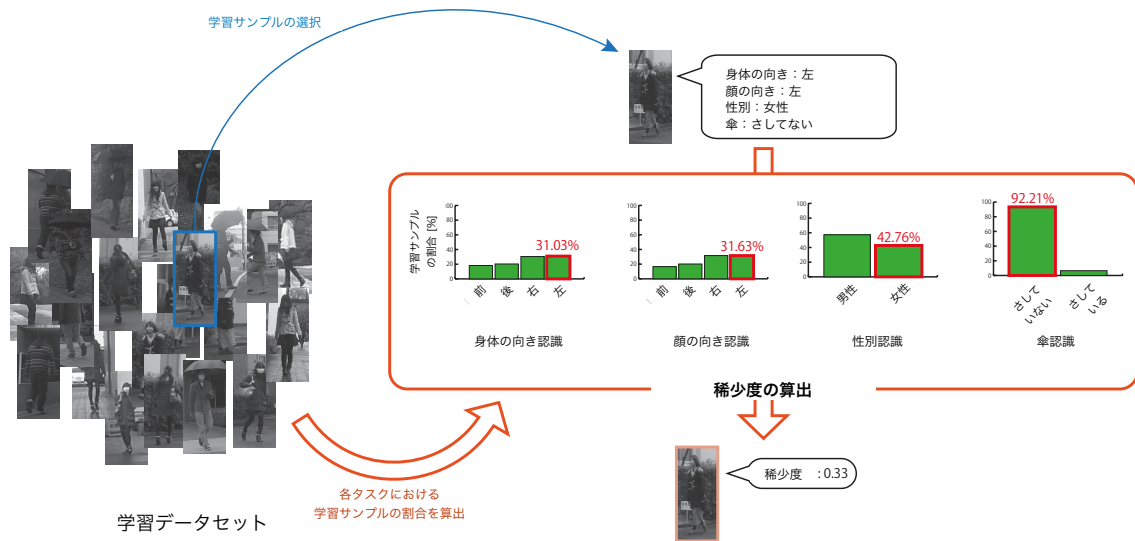


図 4.12: 稀少度の定義

識別精度が低下する。提案手法ではレアサンプルを CNN へ入力する回数を増やすことで、学習サンプルの少ないクラスの識別精度を向上させる。

はじめに、学習サンプル $\{n|1, \dots, N\}$ に対して稀少度を付与する。次に、学習サンプル n を Data Augmentation により増幅する。このとき、学習サンプル n の稀少度が高い場合は Data Augmentation により学習サンプル n を数多く増幅し、学習サンプル n の稀少度が低い場合は Data Augmentation による学習サンプルの増幅を抑制する。Data Augmentation により学習サンプルを増幅した後に、複数のミニバッチの候補を作成し、レアサンプルを適度に含んだミニバッチを CNN へ入力する。以下に、詳細なアルゴリズムについて述べる。

■ 稀少度の定義

学習サンプル n に対し、各タスクで属しているクラスの稀少価値を定量的に示す稀少度 R_n を算出する。学習サンプル n に稀少度を付与する際の処理を、図 4.12 に示す。はじめに、学習データセットにおける各クラスの学習サンプル数の割合を求める。学習サンプルに付与する稀少度 R_n は、各タ

スク t で属しているクラスの学習サンプルの割合を用いて式 (4.4) のように算出する。

$$R_n = \sum_{t=1}^{T-1} (1 - p_{n,t}) \cdot \sqrt{1 - \frac{p_t^{\min}}{p_t^{\max}}} / T \quad (4.4)$$

ここで、 p は学習サンプルの割合を示しており、 $p_{n,t}$ は学習サンプル n がタスク t で属しているクラスのサンプル数の割合、 $p_t^{\max}$ 、 $p_t^{\min}$ はタスク t におけるクラスのサンプル数の最大と最小の割合を示している。式 (4.4) の第 1 項は、タスク t で属しているクラスの稀少度を示しており、第 2 項はタスク t のクラス間における学習サンプル数の偏りを示している。

第 1 項の稀少度は、学習サンプル n がタスク t で学習サンプル数の割合が低いクラスに属している場合に第 1 項の稀少度は高くなり、学習サンプル数の割合が高いクラスに属している場合に低い稀少度となる。第 2 項のタスク t における学習サンプルの偏りは、タスク t の各クラスのサンプル数の割合から最も高い割合と低い割合を用いて求める。ここで、タスク t で最もサンプル数の多いクラスの割合が 1 になるように正規化する。

■ 稀少度に従った Data Augmentation

レアサンプルは学習データセット中で少ない割合のサンプルしか存在していないが、コモンサンプルは学習データセット中に大量に存在している。そのため、複数の学習サンプルを選択してサブセットを作成する際に、レアサンプルが選択される確率は非常に低い。Data Augmentation により全学習サンプルを増幅した場合においても、レアサンプルが選択される確率が低い状態で学習サンプルが増幅される。提案手法では、算出した稀少度により、Data Augmentation で増幅させる学習サンプル数を稀少度により変化する。これにより、図 4.13 のようにレアサンプルを Data Augmentation により数多く増幅させ、コモンサンプルは Data Augmentation による学習サンプルの増幅を抑制して新たな学習データセットを構築する。学習サンプル n の増幅させる学習サンプル数 S_n は、式 (4.5) のように求める。

$$S_n = R_n \cdot A + 1 \quad (4.5)$$

ここで、 A は Data Augmentation により増幅させる最大の学習サンプル数を示している。稀少度に応じて増幅する学習サンプル数を変えることで、コモンサンプルの無駄な学習サンプルの増幅を抑制している。コモンサンプルの増幅を抑えることで、学習サンプルを選択してミニバッチを作成する際にレアサンプルが選択される確率を向上している。また、本実験で学習サンプルに与える幾何変化は、平行移動と拡大、縮小である。

■ ミニバッチの候補の選択

Heterogeneous Learning を導入した CNN に入力するミニバッチを決定するために、複数のミニバッチの候補を作成する。そして、レアサンプルを適度に含んだミニバッチを候補から選択する。まず、

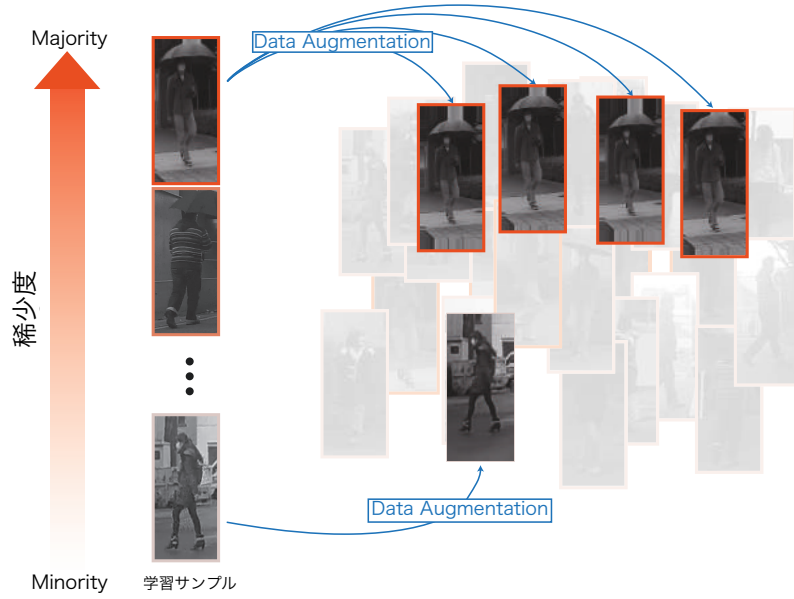


図 4.13: 稀少度に従った Data Augmentation

前節の Data Augmentation により増幅した学習サンプル群から指定したバッチサイズだけ学習サンプルを選択し、ミニバッチを作成する．ここで、作成したミニバッチに対する稀少度 R_k^B を、式 (4.6) のようにミニバッチ内の学習サンプル m に付与されている稀少度 R_m を総和して求める．そして、この処理をミニバッチの候補数 K だけ繰り返す．

$$R_k^B = \sum_{m=1}^M R_m \quad (4.6)$$

各ミニバッチ k に対する稀少度 R_k^B を付与した後に、ミニバッチに対する稀少度 R_k^B を基準にソートし、稀少度 R_m^B が中央値のミニバッチを学習に使用する．中央値を持つミニバッチは、稀少度の高い学習サンプルと、稀少度の低いサンプルが適度に分配されている．そのため、レアサンプルを適度を選択してミニバッチを作成できる．

■ 提案手法の学習アルゴリズム

提案手法の学習アルゴリズムを Algorithm 1 に示す．はじめに、学習データセット $\mathbf{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ 中の学習サンプルに対して稀少度 R_n を付与する．そして、学習サンプル n に付与した稀少度 R_n に応じて増幅する学習サンプル数 S_n を決定し、Data Augmentation により学習サンプルを増幅する．これらの処理により、Data Augmentation により増幅した新たな学習データセット $\mathbf{D}' = \{(\mathbf{x}_{n'}, R_{n'}, \mathbf{y}_{n'})\}_{n'=1}^{N'}$ を作成する．学習データセット $\mathbf{D}'$ を作成した後に、CNN を学習する．提案手法では、学習データセット $\mathbf{D}'$ から学習サンプルを選択してミニバッチを作成し、ミニバッチに対する稀少度 R^B を求める．ミニバッチの作成と稀少度 R^B の付与は K 回行う．そして、生成したミニバッチの候補 $\mathbf{R}^B$ か

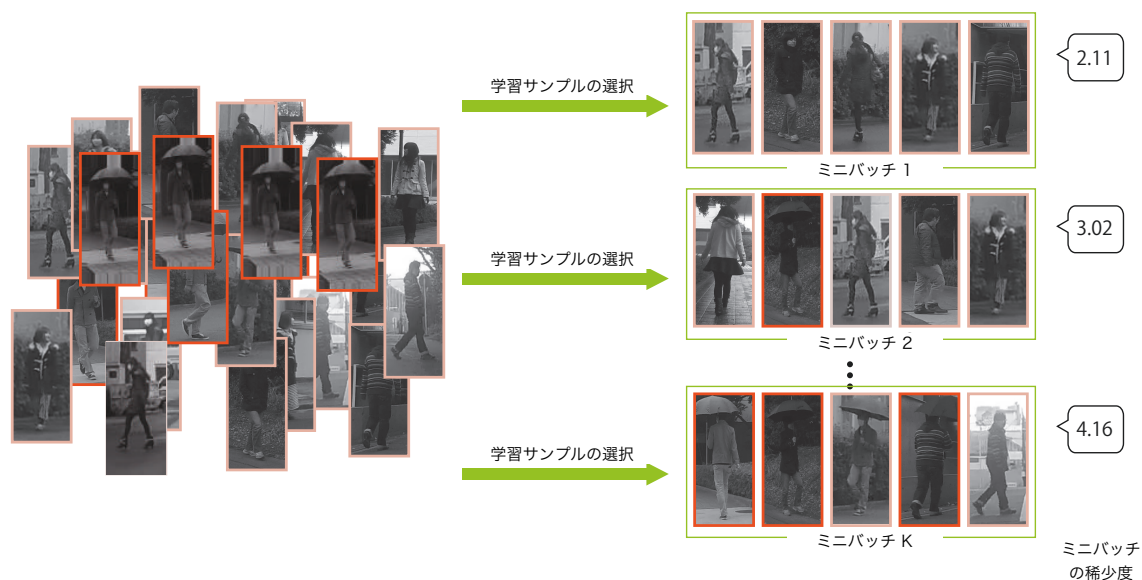


図 4.14: ミニバッチの候補の生成

ら CNN の学習で使用するミニバッチを選択する．このとき，学習で使用するミニバッチの選択方法は，ミニバッチに対する稀少度をソートした際に中央値を持つミニバッチを学習に使用する．学習方法は，従来の一般的な Heterogeneous Learning を導入した CNN と同様である．

4.3.2 評価実験

各タスクにおいて各クラスの学習サンプル数が不均衡なデータセットに対し，提案手法の有効性を評価実験により示す．まず，提案手法のパラメータを変化させることで学習過程の挙動の変化を確認し，従来手法と比較する際の提案手法のパラメータを決定する．提案手法の変化させるパラメータは，Data Augmentation により増幅させる最大の学習サンプル数 A とミニバッチの候補数 K である．Data Augmentation により増幅させる最大の学習サンプル数 A を変更することで，稀少度に応じた Data Augmentation による学習サンプルの増幅の有効性について述べる．そして，ミニバッチの候補数 K を変更することで，ミニバッチの候補から学習に使用するミニバッチを選択する有効性を示す．次に，提案手法の最適なパラメータを決定した後に，従来手法との識別性能を比較する．従来法として比較する手法を以下に示す．

- それぞれのタスクを学習した単一の CNN
- 全てのタスクを学習した Heterogeneous Learning を導入した CNN
- 提案手法により学習した Heterogeneous Learning を導入した CNN
- 提案手法により学習した Heterogeneous Learning を導入した VGG16 [28]

Algorithm 1 提案手法の学習アルゴリズム

```
1: Data: 学習データセット  $\mathbf{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ 
2: // 学習サンプルに対して稀少度を付与
3: for 0 to  $N$  do
4:    $R_n = \sum_{t=1}^{T-1} (1 - p_{n,t}) \cdot \sqrt{1 - \frac{p_t^{min}}{p_t^{max}}} / T$ 
5:    $S_n = R_n \cdot A + 1$ 
6:   for 0 to  $S_n$  do
7:     Data Augmentation による学習サンプルの増幅
8:   end for
9: end for
10: Data: Data Augmentation 後の学習データセット
     $\mathbf{D}' = \{(\mathbf{x}_{n'}, R_{n'}, \mathbf{y}_{n'})\}_{n'=1}^{N'}$ 
11: while not stopping criterion do
12:   // ミニバッチの稀少度を算出  $R^B$ 
13:   for 0 to  $K$  do
14:     学習データセット  $\mathbf{D}'$  から  $M$  個の学習サンプルを選択してミニバッチを作成.
15:      $R_k^B = \sum_{m=1}^M R_m$ 
16:   end for
17:   ミニバッチの稀少度  $R_k^B$  が中央値となるミニバッチを選択.
18:   選択したミニバッチを入力して学習誤差を算出.
19:   誤差逆伝播法によりネットワークのパラメータを更新.
20: end while
```

また、従来手法と提案手法の識別性能の比較では、ミニバッチのバッチサイズを $\{5, 10, 20\}$ と変化させて識別性能を比較する。使用するデータセットは、車載カメラで撮影した 46,956 枚の歩行者画像を用いる。使用するデータセットの歩行者には、図 4.9 の歩行者の部位位置、身体の向き、顔の向き、性別、傘をさしているかの 5 タスクのラベルが付与されている。また、各タスクの学習サンプル数の割合は図 4.14 のようになっており、最も学習サンプル数が少ないクラスは“傘をさしている”クラスであることが確認できる。歩行者部位検出以外の各タスクの評価方法は、各タスクにおける各クラスの認識率を算出して評価する。歩行者部位検出は、検出結果とラベルの距離を算出し、しきい値より距離が近ければ検出成功としている。ここで、しきい値は頭部と両足の距離の 10% とし、検出成功としたサンプルから検出率を算出して評価する。

本実験では、図 2.16 の畳み込み層 3 層と全結合層 3 層から構築されている CNN を用いる。ここで、CNN の学習係数は 0.001 であり、学習の更新回数は 50 万回である。本実験では各実験に対して 5 回試行し、識別精度のばらつきを標準偏差により評価する。また、入力する学習サンプルの違いによる識別精度のばらつきを評価するために、CNN の重みフィルタや結合重みの初期値は全て統一している。

表 4.2: 実験に使用する CNN の構造

層	詳細	サイズ
入力	グレースケール画像	$1 \times 64 \times 128$
conv.1	$5 \times 9 \times 32$	$60 \times 120 \times 32$
maxout 1	2	$60 \times 120 \times 16$
pool.1	2×2	$30 \times 60 \times 16$
conv.2	$3 \times 5 \times 16$	$28 \times 56 \times 16$
maxout 2	2	$28 \times 56 \times 8$
pool.2	2×2	$14 \times 28 \times 8$
conv.3	$3 \times 5 \times 64$	$12 \times 24 \times 64$
maxout 3	2	$12 \times 24 \times 32$
fc.1	sigmoid	3000
fc.2	sigmoid	2000
fc.3	sigmoid	1000
Output	heterogeneous	18

■ パラメータの変更による精度の変化

Data Augmentation で増幅する最大の学習サンプル数 A とミニバッチの候補数 K を変化させた際の学習誤差と各タスクの識別性能を比較する. A と K を変化させた際の識別性能を, 図 4.15(a), 図 4.15(b) に示す. ここで, 図 4.15(a), 図 4.15(b) の学習誤差は 5 回試行した際の各更新回数の学習誤差を平均した値であり, 認識率のエラーバーは 5 回試行した際の標準偏差を示している. 図 4.15(a) では, A を $\{5, 10, 25\}$ と変化させており, 1 つの学習サンプル当たりの増幅数を稀少度により変化させる場合と変化させない場合で比較している. 図 4.15(a) より, A が 25 のときに全体の識別性能が最も良く, 最も学習サンプル数の少ない“傘をさしている”クラスの認識率が約 4.6% 向上していることが確認できる. しかし, “身体の向きの正面”と“顔の向きの正面”, “性別の女性”の認識率は従来法より認識率が低下している. これらのクラスは, “身体の向き”と“顔の向き”, “性別”のタスクにおいて最も学習サンプルの少ないクラスである. 図 4.16 に, 従来法と提案手法が 1 epoch 中に使用した学習サンプル数を示す. 図 4.16 より, $A = 25$ の時に従来法では学習に約 110 万枚の学習サンプルを使用しているのに対して, 提案手法では約 40 万枚学習サンプルを使用している. 提案手法では, Data Augmentation でコモンサンプルの増幅を抑制しているため, 使用する学習サンプル数を約 1/2 以下まで抑えていることが確認できる.

図 4.15(b) では, K を $\{1, 5, 15, 25\}$ と変化している. ここで, 図 4.15(b) の実験では, A を 25 で固定して実験している. 図 4.15(b) より, ミニバッチの候補数 K を増やすことで認識率が向上し, K が 15 のときに 74.18% で最も認識率が良い. また, ミニバッチの候補数 K を増やすことにより, 複数回試行した際の標準偏差が約 2 分の 1 程度まで削減できていることが確認できる. さらに, 図 4.15(a) で“身体の向きの正面”と“顔の向きの正面”, “性別の女性”の認識率が従来より低下していたのに対

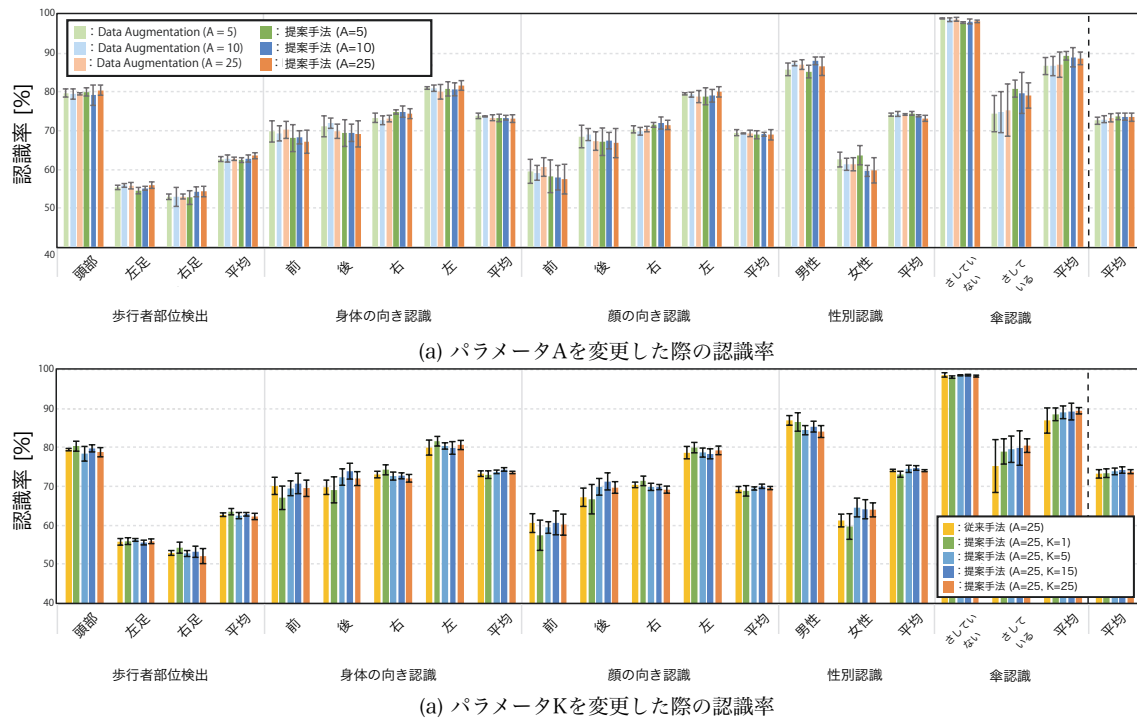


図 4.15: 各パラメータにおける提案手法の認識精度

して、 K を増加させることで従来と同等以上の認識率となっている。

■ 従来法との比較

図 4.19 に、従来のミニバッチと提案手法で識別精度を比較する。ここで、従来法と提案手法でミニバッチのサイズ M を $\{5, 10, 15\}$ に変更して性能を比較する。図 4.19 から、単一の CNN で歩行者部位検出した性能より Heterogeneous Learning を導入した CNN の歩行者部位検出のほうが高精度に検出できていることがわかる。単一の CNN で歩行者の部位を検出した場合、両足の検出性能が Heterogeneous Learning を導入した CNN より大幅に低下している。歩行者部位検出の精度が大幅に向上した原因として、他の歩行者属性を同時に認識することで両足の位置を捉えやすくなっていると考えられる。さらに、Heterogeneous Learning を導入した CNN に提案手法を適応することで、学習サンプル数の少ないクラスの認識性能を向上できていることが確認できる。また、バッチサイズ M を変化した際に、Heterogeneous Learning を導入した CNN はバッチサイズ M を 20 にしたとき最も性能が良く、提案手法ではバッチサイズ M を 5 にした際に最も性能が良い。

図 4.19 に認識結果を示す。歩行者の頭部と腰部分に存在する円がそれぞれ顔と身体の向きを示しており、歩行者にの頭部と両足にある緑の点は検出した歩行者部位を示している。また、外側の Bounding box の色は性別を示しており、青が男性、赤が女性である。そして、傘をさしていると認識された場合は右下に傘のアイコンが付与している。図 4.19 より、Heterogeneous Learning を導入した CNN により様々な姿勢の歩行者に対して複数タスクの識別が可能であることが確認できる。しか

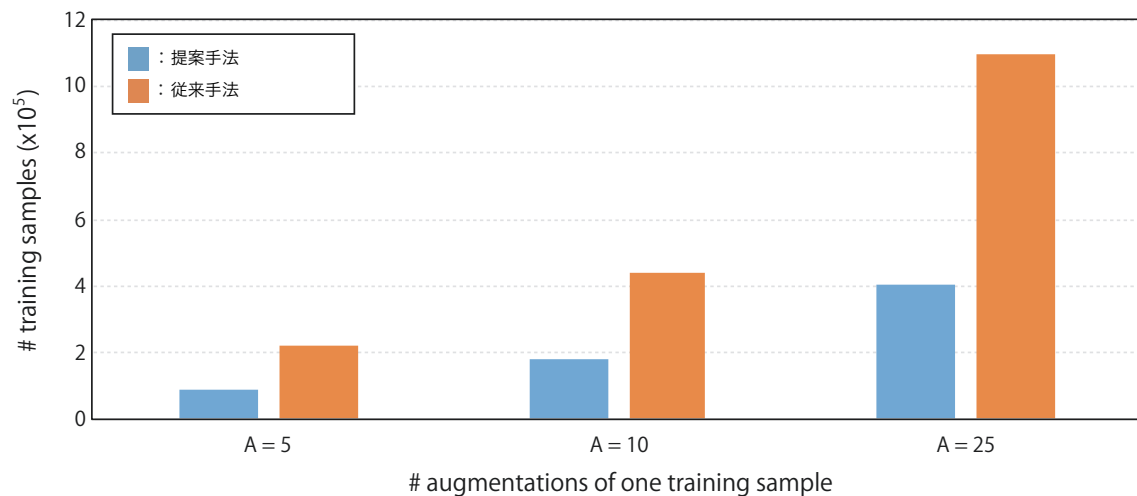


図 4.16: 提案手法と従来法における 1epoch 中に使用した学習サンプル数

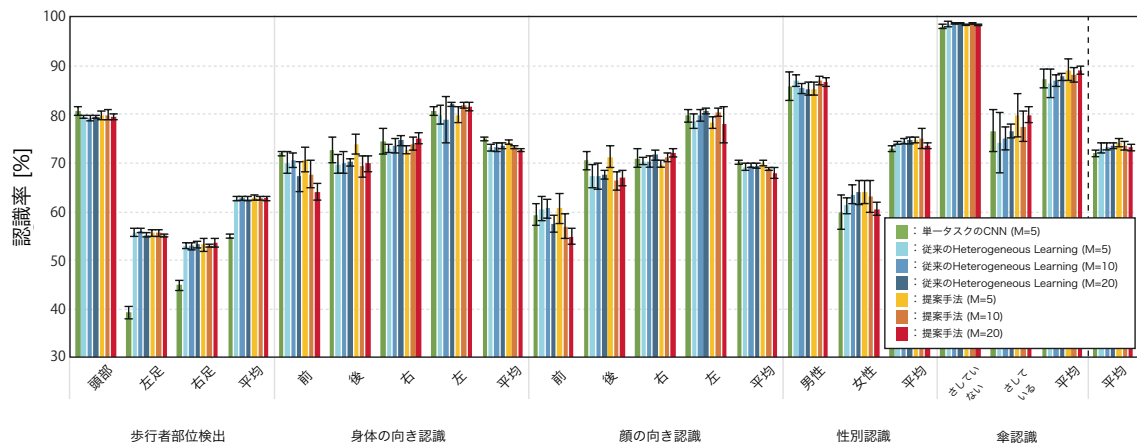


図 4.17: 従来法と提案手法の認識精度の比較

し、従来法では学習サンプルの少ないクラスである“身体が正面を向いている歩行者”と“顔が正面を向いている歩行者”、“傘をさしている歩行者”の認識が困難であるのに対し、提案手法では認識が可能であることが確認できる。

図 4.18 に VGGNet との認識結果の比較を示す。VGGNet では、表 4.2 の CNN よりパラメータ数が膨大であるため、学習サンプル数の割合の影響を受けやすい。そのため、最も学習サンプル数の割合が低い“傘をさしている歩行者”の認識精度が大幅に低下している。一方で、提案手法では“傘をさしている歩行者”の精度が大幅に精度が向上している。他のタスクにおいても提案手法の導入により性能が向上しており、全体の認識精度が約 3% 向上している。

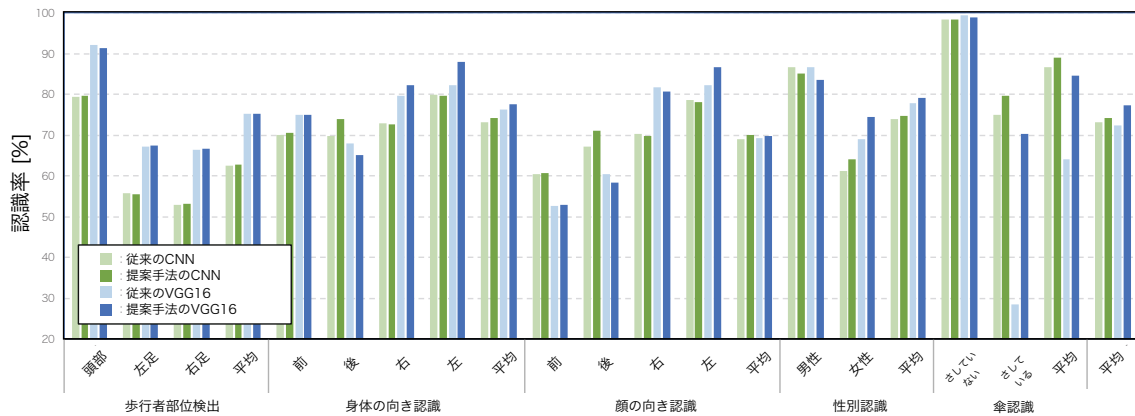


図 4.18: VGGNet による認識精度の比較



図 4.19: 従来法と提案手法の認識結果の可視化

4.4 まとめ

本研究では、重み付き誤差関数を導入した DCNN による Heterogeneous Learning と、稀少度を用いた歩行者属性認識を提案した。重み付き誤差関数を導入した Heterogeneous Learning は、各タスク間における学習誤差のばらつきを重み付けにより抑制できる。各タスクの学習誤差のばらつきを抑制することで、ネットワークの学習を安定化でき、複数タスクの認識精度を向上できる。今後は、Heterogeneous Learning をどう導入した CNN の学習を、より安定に行えるアプローチの考案を行う。

稀少度を用いた歩行者属性認識では、複数タスクの識別において学習サンプル数の少ないクラスの識別性能を向上させる手法を提案した。学習サンプルに対して稀少度を付与し、付与した稀少度に応じて Data Augmentation により学習サンプルを増幅し、レアサンプルを適度に含んだミニバッチを用いて学習することで、学習サンプル数の少ないクラスの認識性能を向上できた。今後の課題として、歩行者属性を用いた歩行者検出の高精度化が挙げられる。

第5章

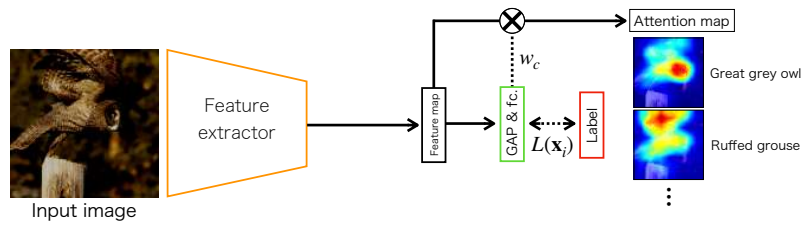
Attention Branch Network による物体認識

CNN は画像分類をはじめとする物体検出や顔属性認識等で高い性能を達成している [14, 10, 121] が、深層学習は精度向上を目的にネットワークを複雑に構築する傾向がある。そのため、ネットワークが有する膨大なパラメータが原因で、認識結果に対する判断根拠が不明確になりやすい。このような、深層学習における判断根拠を説明する取り組みは活発に取り組まれており、様々な解析手法が提案されている [13, 67, 122, 37]。

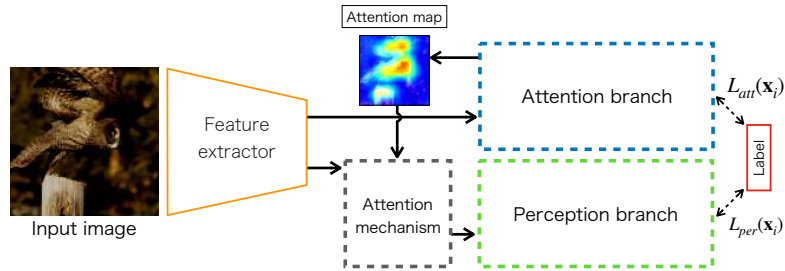
深層学習における判断根拠の解析では、ネットワークが推論の過程で注視した領域をマップで表現した Attention map を用いて判断根拠を解析する“視覚的説明”が一般的に用いられる。視覚的説明の手法は2種類の方法にカテゴリ分けでき、Bottom-up の手法と Top-down の手法に分類できる。Bottom-up の手法は、一般的に勾配情報とノイズやクラス情報を補助情報として併用することで注視領域を獲得する。Bottom-up の手法は再学習やネットワークの修正を行わずに CNN の注視領域を獲得できるが、注視領域の獲得時に勾配情報を必要とするため逆伝播を行う必要がある。対照的に、Top-down な手法は一度の順伝播から CNN の注視領域を獲得できる。Top-down に各クラスにおける注視領域を畳み込み層の応答値から獲得する Class Activation Mapping (CAM) [13] が提案されてる。CAM は、図 5.1 (a) のように全結合層の代わりに、畳み込み層と GAP を配置する。しかしながら、全結合層を取り除いてネットワークを構築する必要があり、CNN の汎化性能を低下させる場合がある。

この問題を避けるために、CNN の解析手法には Bottom-up な手法が一般的に用いられてきた。視覚的説明で獲得できる Attention map は、画像認識の推論時において高く貢献した領域を示している。本研究では、順伝播の過程で Attention map を獲得できる Top-down の手法を用いるために、視覚的説明における Top-down の手法を Attention 機構へ応用する。視覚的説明における Attention map を Attention 機構へ応用することで、提案するネットワークは画像認識において重要度の高い領域を注視しながらネットワークを学習できる。視覚的説明をベースに作成された Attention 機構は、CNN の解析と性能向上を同時に行うことが可能である。

本研究では、視覚的説明を Attention 機構へ応用した Attention Branch Network (ABN) を提案する。ABN は、図 5.1 (b) のように Attention 機構をブランチ構造へ取り入れることで、Top-down の手法を Attention 機構としてネットワークに組み込んでいる。提案する Attention Branch Network は、図 5.1(b) のように Feature extractor と Attention branch, Perception branch の3つのモジュールから構成される。Feature extractor は、複数の畳み込み層を用いて入力画像から特徴マップを抽出する。Attention branch は、Top-down の手法が出力する Attention map を Attention 機構へ適応させる構造となってい



(a) Class Activation Mapping



(b) Attention Branch Network

図 5.1: Class Activation Mapping と Attention Branch Network のネットワーク構造

る。Attention branch は Attention 機構と視覚的説明の 2 つの役割を担う Attention map を作成するため、ABN において重要な役割を担っている。Perception branch は、抽出した特徴マップと Attention map を用いて、最終的な各クラスにおける確率を出力する。ABN のネットワーク構造は非常にシンプルであり、Attention branch と Perception branch から出力される学習誤差を用いることで End-to-End に学習できる。また、Attention branch をベースとなる ResNet や ResNeXt, Faster R-CNN, Multi-task Learning に導入することで、容易に応用できる。本研究の貢献は、以下の 3 つである。

- ABN は視覚的説明における Top-down の手法に Attention 機構を取り入れたブランチ構造を導入することで、ネットワークを構築する。ABN は、Top-down の手法をネットワークに取り入れることで性能を向上させる初めての手法である。
- ABN は、ベースとなる VGGNet や ResNet, Faster R-CNN, Multi-task Learning に対して、ネットワークを分割して Attention map を出力する Attention branch を導入することで様々なネットワークへ応用できる。
- ABN は性能向上を実現しつつ、CNN の判断根拠の理解を同時に行うことができる。

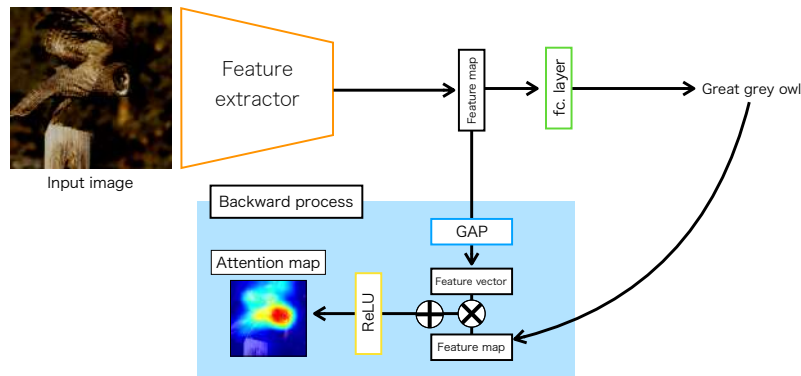


図 5.2: Gradients weight-CAM の構造

5.1 関連研究

5.1.1 画像認識分野における可視化

深層学習における画像認識分野では、推論時の Attention map を獲得し、深層学習の判断根拠を解析する視覚的説明の研究が活発に取り組まれている [27, 13, 122, 37, 123]. 注視領域の獲得には、勾配を用いた Bottom-up の手法とネットワークの応用値を用いる Top-down の手法の 2 種類がある. Bottom-up の手法の例として, SmoothGrad [37] がある. SmoothGrad は, 入力画像にノイズを付与して Sensitively map を獲得する処理を繰り返し行い, それぞれ獲得した Sensitively map を平均することで CNN の注視領域を獲得する. 他の視覚的説明における Bottom-up な手法として, Guided Backpropagation と Gradient-weighted Class Activation Mapping (Grad-CAM) [122, 123] が提案されている. Guided Backpropagation と Grad-CAM は, 逆伝播の特定のクラスにおける正值の勾配のみを用いることで, CNN の注視領域を獲得する. Guided Backpropagation と Grad-CAM は特定のクラスにおける Attention map を様々なプレトレーニングモデルから獲得できるため, CNN の解析手法として広く一般的に用いられている. Grad-CAM の構造を, 図 5.2 に示す. Grad-CAM は, 特定のクラスの出力層のユニットから勾配を発生させ, 特徴マップを獲得する. そして, 順伝播時における最後の畳み込み層の特徴マップに対して, GAP を施す. GAP により獲得した特徴ベクトルは重みとして使用し, 勾配ベースの特徴マップに対して重み付き和を求める. この重み付き和で求めた特徴マップに ReLU を施すことで, 特定のクラスに対する注視領域を獲得することができる.

視覚的説明における Top-down の手法は, ネットワークが出力した応答値を用いることで Attention map を獲得できる. Top-down の手法は Attention map を獲得するためにネットワークを再構築して再学習する必要があるが, 順伝播の過程で各クラスにおける注視領域を獲得できる. Top-down の手法の代表的な手法である CAM [13] は, 畳み込み層の応答値と全結合層の結合重みを用いることで, 各クラスにおける Attention map を獲得できる. CAM は弱教師あり学習における物体位置推定で高い性能を達成しているが, 全結合層を畳み込み層に入れ替える等の処理が必要なため画像分類においては性能低下を引き起こしやすい.

提案する ABN は、Top-down に Attention map を獲得できる CAM をベースにネットワークを構築する。CAM は順伝播の過程で Attention map を出力して特徴マップに重み付けできるため、Attention 機構との相性が非常に良い。対照的に、Grad-CAM 等の Bottom-up な手法は誤差逆伝播を用いて勾配を求める必要があるため、Attention 機構との相性が良くない。これらの理由から、提案する ABN は Top-down の手法をベースに拡張を行う。

5.1.2 Attention 機構

Attention 機構は、画像認識分野をはじめとした自然言語処理等の様々な分野で活用されている [124, 125, 126, 36]。特に、Attention 機構は Recurrent Neural Network や Long Short Term Memory (LSTM) [127] 等を用いた時系列データを用いた手法で広く扱われている [125, 128, 129, 130]。画像と時系列に Attention 機構を導入した代表的な手法として、Xu らの研究 [125] がある。Xu らの研究で用いられる Attention 機構では、Soft Attention と Hard Attention の 2 つの Attention map が用いられている。特に、Xu らの Soft Attention は LSTM のゲートを利用しており、Image captioning や Visual Question Answering の分野で活発に用いられている [128, 129]。また、特徴量全体を加味した self-attention を CNN に一般化した Non-local Neural Network [131] や、強化学習により注視点を時系列に沿って移動させる手法も提案されている [132]。

また、1 枚の画像を対象に Attention 機構を応用した手法も提案されている [126, 36, 133]。代表的な手法として、Residual Attention Network [126] や Squeeze-and-Excitation Networks (SENet) [36] がある。Residual Attention Network は、2 次元の Attention map を用いた複数の Attention 機構を導入したモデルである。Residual Attention Network では、複数の Attention 機構を積み込む Stacked Network Structure や、Residual Module による Attention map の反映方法である Attention Residual Learning を取り入れている。SENet の Attention 機構は各 Residual Module に導入され、1 次元の Attention map を特徴マップのチャンネルに対して施すことで、ネットワークの性能を向上させている。

提案する ABN は、視覚的説明をベースに、画像認識に有効な領域を強調した Attention map を用いて特徴マップに重み付けする。従来の Attention 機構を取り入れたネットワークでは、順伝播の応答値のみ用いて特徴マップに重み付けをしていた。しかし、提案する ABN は視覚的説明により獲得した Attention map を用いるため、画像認識に有効な領域を強調しやすい。

5.2 Attention Branch Network

提案する ABN は、図 5.1(a) のように Feature extractor, Attention branch, Perception branch の 3 つのモジュールから構成されている。Feature extractor は、入力画像から複数の畳み込み層により特徴マップを抽出する。Attention branch は、CAM の構造を用いることで、視覚的説明ベースの Attention map を出力する。Perception branch では獲得した Attention map と特徴マップを Attention 機構へ応用して新たな特徴マップを出力し、最終的な各クラスにおけるクラス確率を出力する。

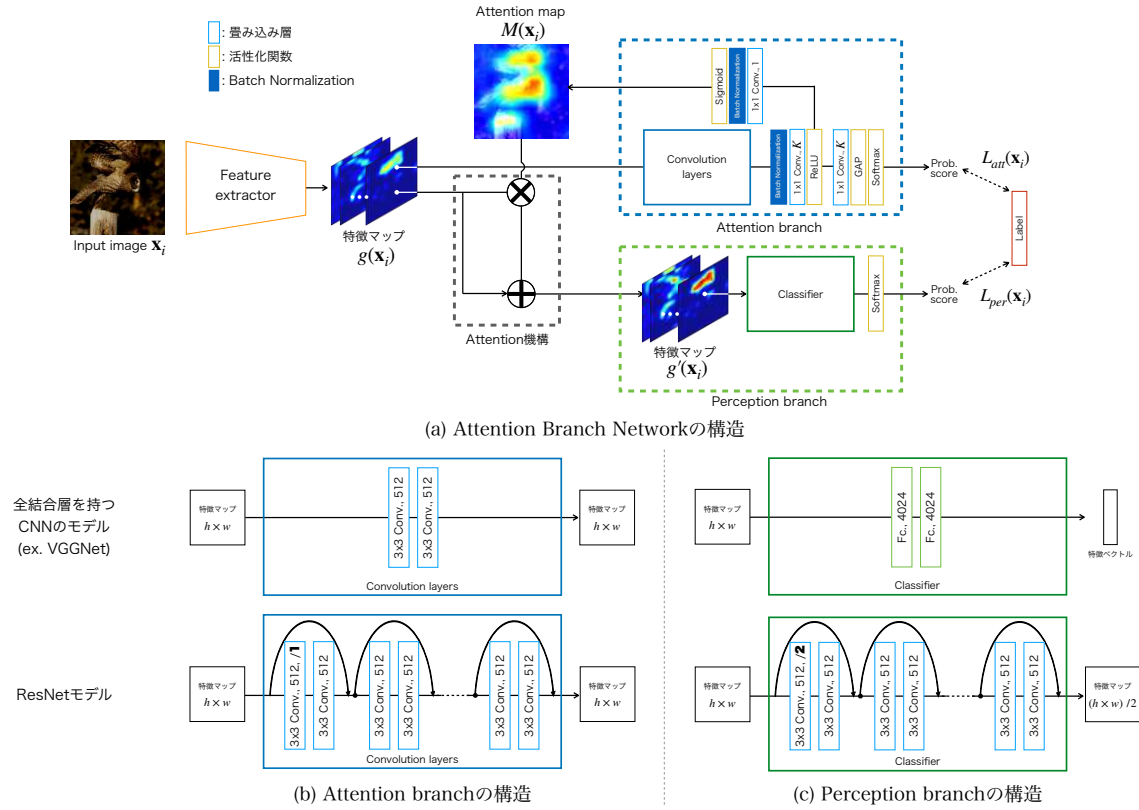


図 5.3: Attention Branch Network の構造

ABN は、VGGNet [28] や ResNet [10] 等のベースとなるネットワークに対してブランチ構造を取り入れている。Feature extractor と Perception branch は、ベースラインのネットワークを特定の層で分割することで構築する。そして、Feature extractor の後層に、CAM の上位層をベースに構築した Attention branch を導入する。ABN は、Attention branch をベースラインのネットワークに追加することで、様々な画像認識タスクへ応用できる。本稿では、ABN を画像分類、詳細識別、物体検出、顔属性認識へ応用し、有効性を調査する。

5.2.1 Attention branch

CAM は、図 5.1 のように $K \times 3 \times 3$ の畳み込み層と GAP、全結合層を配置する。ここで、 K はクラス数を示しており、“ $K \times 3 \times 3$ ”は“ 3×3 ”のカーネルを K チャンネル持つ畳み込み層を示している。 $K \times 3 \times 3$ の畳み込み層は特定のクラスにおける注視領域を示した $K \times h \times w$ の特徴マップを出力する。 $K \times h \times w$ の特徴マップは GAP により $K \times 1 \times 1$ ヘダウンサンプリングされ、全結合層を施することで各クラスにおける確率を出力する。各クラスにおける Attention map を可視化する際は、全結合層の結合重みで重み付けすることで可視化を行う。

全結合層の代わりに、CAM では畳み込み層を導入する。この制約は、Attention branch においても

生じる。次の層のユニットと密に結合する全結合層は、畳み込み層から得ら獲得できる注視領域の位置関係を打ち消してしまう。そのため、VGGNetのような全結合層を保持するネットワークをベースラインとして採用する場合、図 5.5(b) の上部のように CAM と同様に全結合層の代わりに 3×3 の畳み込み層を導入する必要がある。また、ResNet をベースにする場合、図 5.5(b) の下部のように Perception branch と同様の構造を持つ Residual block を配置する。ここで、特徴マップの解像度を下げないために、配置する Residual block の 1 層目の畳み込み層のストライドを 1 に設定する。

Attention map を出力するために、Attention map と各クラスの確率を出力する上位層を構築する。しかし、CAM は学習済みの全結合層の重みを用いて Attention map を作成するため、学習の過程で Attention map を作成することができない。この問題を解決するために、 $K \times 1 \times 1$ の畳み込み層を全結合層の代わりに構築する。この $K \times 1 \times 1$ の畳み込み層は CAM の全結合層を模しており、同じ役割を担っている。 $K \times 1 \times 1$ の畳み込み層の後に GAP により特徴マップをダウンサンプリングし、Softmax 関数を施すことで各クラスの確率値を出力する。最後に、 $K \times h \times w$ の特徴マップを用いて Attention map を出力する。K 枚の $h \times w$ の特徴マップから Attention map を作成するために、 $1 \times 1 \times 1$ の畳み込み層を施す。 $1 \times 1 \times 1$ の畳み込み層を施すことで、 $1 \times h \times w$ の特徴マップを出力する。 $1 \times h \times w$ の特徴マップに Sigmoid 関数を施すことで、Attention 機構で用いる Attention map を作成する。

5.2.2 Perception branch

Perception brach では、Feature extractor から抽出した特徴マップと Attention map を用いて最終的な認識結果を出力する。Perception branch の構造は、図 5.5(c) のように VGGNet や ResNet 等のベースラインモデルの上位層と同様の構造となっている。はじめに、Attention 機構により Attention map を特徴マップへ反映させる。Attention map は、式 (5.1) と式 (5.2) の反映方法を用いる。ここで、 $g(\mathbf{x}_i)$ が Feature extractor の特徴マップ、 $M(\mathbf{x}_i)$ が Attention map、 $g'(\mathbf{x}_i)$ が Attention 機構により出力された特徴マップ、 $\{c|1, \dots, C\}$ がチャンネル数を示している。

$$g'_c(\mathbf{x}_i) = M(\mathbf{x}_i) \cdot g_c(\mathbf{x}_i) \quad (5.1)$$

$$g'_c(\mathbf{x}_i) = (1 + M(\mathbf{x}_i)) \cdot g_c(\mathbf{x}_i) \quad (5.2)$$

式 (5.1) は単純な特定のチャンネルの特徴マップ $g_c(\mathbf{x}_i)$ と Attention map の乗算である。式 (5.2) は Attention map が強く反応した領域の特徴マップを強調することができ、Attention map の 0 領域に対しても特徴マップが消失するのを防ぐことができる。

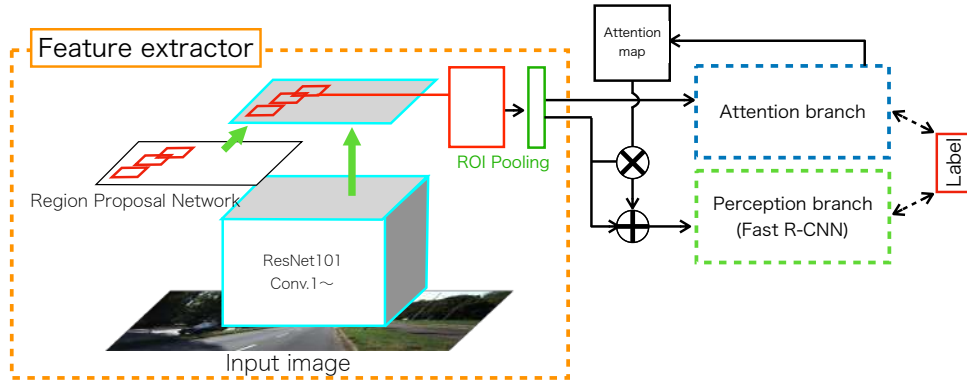


図 5.4: Attention Branch Network を導入した Faster R-CNN の構造

5.2.3 ネットワークの学習方法

ABN は、上位層に Attention branch と Perception branch の 2 つのブランチを配置している。ABN の学習誤差 $L(\mathbf{x}_i)$ は、式 (5.3) のような 2 つのブランチの学習誤差の単純な加算により求める。

$$L(\mathbf{x}_i) = L_{att}(\mathbf{x}_i) + L_{per}(\mathbf{x}_i) \quad (5.3)$$

ここで、 $L_{att}(\mathbf{x}_i)$ を Attention branch の学習誤差、 $L_{per}(\mathbf{x}_i)$ を Perception branch の学習誤差である。各ブランチの学習誤差は、Softmax 関数と Cross entropy 誤差を用いて算出される。ABN の Feature extractor は Attention branch と Perception branch の勾配を受け取ることで最適化される。他の画像認識タスクへ応用する場合、ベースとするネットワークモデルの誤差関数に応じて適応的に変更できる。

5.2.4 その他の画像認識タスクへの応用

画像分類モデルの ABN は、ネットワークの上位層が Attention map を出力するブランチと各クラスの確率値を出力するブランチを用いてネットワークを構築している。ABN は、Attention branch を導入することで物体検出や顔属性認識に応用することができる。本章では、ABN を自動車制御と歩行者検出、Multi-task Learning へ応用する際のネットワーク構造について述べる。

自動車制御 自動車制御では、村瀬らの研究をベースに ABN を導入する [134]。村瀬らの研究では、Bojarski ら [135] の研究を基に自動車制御の実験を行なっている。まず、Bojarski らは自動車と長距離の走行により自動車制御の実験を行なっているが、村瀬らは Grand Theft Auto (GTA5) のゲームのシュミレータ環境を用いて実験を行なっている。Bojarski らの実験環境は非常に高コストかつ人身事故が発生する可能性があり、実験環境を構築するのは非常に困難である。ゲームのシュミレータを用いることで、事故を発生させずに低コストで実験を行うことができる。自動車の制御では、自動車のステアリング値とスロットル値を回帰により出力する。Bojarski らの CNN では、ステアリング値のみ CNN で出力していたが、スロットル値は人により制御している。ステアリング値とスロ

トル値の両方を CNN で出力することで、完全な自動車制御を行う。ここで、使用する CNN は 1 フレームのみ入力されるため、自車のスピードを考慮した特徴抽出が困難である。この問題を解決するために、CNN の全結合層に対して車速を入力する。これにより、自車のスピードを考慮してステアリング値とスロットル値を出力することができる。

ABN に応用する場合、畳み込み層と全結合層で切り分け、Attention branch を構築する。しかし、局所的な領域をカーネルで畳み込む畳み込み層は、密に結合する全結合層に比べて回帰推定に不向きである。この問題を解決するために、Attention branch が出力した特徴マップに対して、特徴マップと同サイズのカーネルを配置して特徴マップ全体をプーリングする Weighted Global Pooling (WGP) を提案する。WGP により、入力される特徴マップ全体を重み付けしながらプーリングできるため、特徴マップ全体の特徴を欠落することなく回帰値を出力できる。最終的な回帰値は、WGP の出力値に $\tanh$ 関数を施すことで出力する。

歩行者検出 歩行者検出では、Faster R-CNN [14] に対して ABN を導入できる。Faster R-CNN は、物体候補領域を検出する RPN と、検出した物体候補領域からカテゴリ分類を行う Fast R-CNN [49] から構成されている。Faster R-CNN のような R-CNN ベースの検出法は、物体候補領域検出とカテゴリ分類の処理が分離しているため、後段の Fast R-CNN によるカテゴリ分類に対して Attention branch を導入できる。Faster R-CNN に Attention branch を導入する場合、Attention branch はクラス分類のみ行う。Attention map の作成方法と、Attention map の特徴マップへの反映方法は画像分類モデルの ABN と同様である。

Multi-task Learning 一般的な Multi-task Learning は、出力層の各ユニットが特定のタスクのスコアを出力する構造となっている [136]。学習時は出力層のそれぞれのユニットに対して学習誤差関数を設計することで、複数のタスクを 1 つのネットワークで学習する。しかし、ABN を Multi-task Learning に応用する場合に一つ問題点が発生する。画像分類や物体検出では、1 つの入力に対して 1 つのタスクを出力していた。しかし、Multi-task Learning は 1 つの入力に対して複数のタスクを出力している。入力と出力の関係が 1 対 1 の場合は 1 枚の Attention map で表現できるが、入力と出力の関係が 1 対多の場合は 1 枚の Attention map で出力できない。この問題を解決するために、Multi-task Learning を導入した ABN は各タスクに対応した複数の Attention map を出力する。ここで、Multi-task Learning を導入した ABN は ResNet101 をベースに構築する。

各タスクの Attention map を出力するために、はじめに Attention branch に Multi-task Learning を導入する。はじめに、Residual Block 4 が出力した特徴マップに対して $T \times 1 \times 1$ の畳み込み層を施すことで、 $T \times 14 \times 14$ の特徴マップを出力する。特定のタスク $\{t|1, \dots, T\}$ の確率値は、 $T \times 1 \times 1$ の畳み込み層から獲得した t 番目の 14×14 の特徴マップに対して、GAP と Sigmoid 関数を施すことで求める。学習時は、Sigmoid 関数と Binary Cross Entropy 誤差関数を用いることで学習する。そして、 14×14 の特徴マップを Attention map として使用する。

Perception branch に対して Multi-task Learning を導入する。Attention map を施した特徴マップ $g_c^t(\mathbf{x})$ は、図 5.5 のように特定のタスク t における Attention map $M^t(\mathbf{x})$ と Feature extractor の特徴マップ $g(\mathbf{x})$ を用いて作成する。特徴マップ $g_c^t(\mathbf{x})$ を作成した後に、特定のタスク t の確率値を Perception

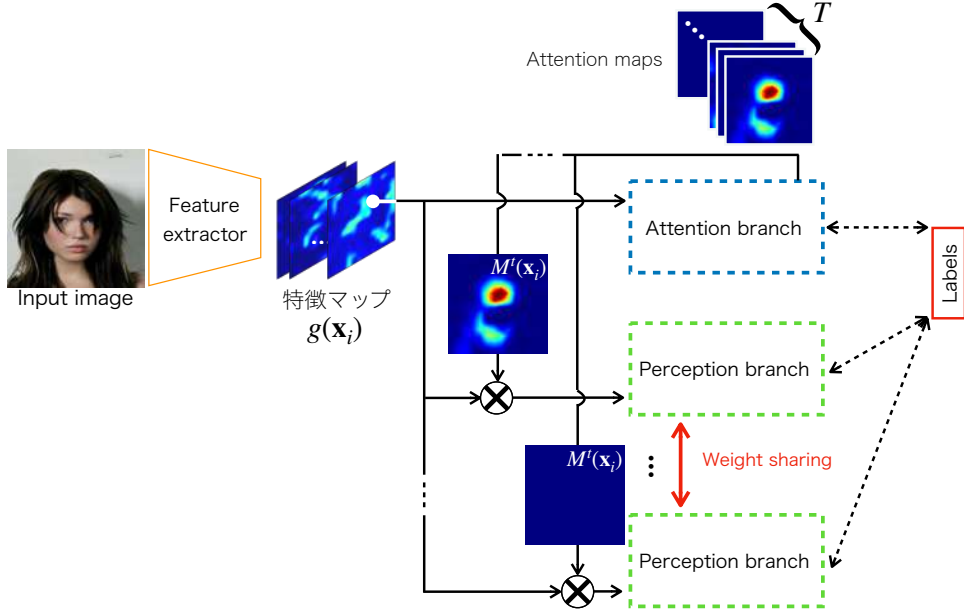


図 5.5: Multi-task Learning を導入した Attention Branch Network の構造

branch $p_{per}(\cdot)$ を用いて出力する。

$$g^{tt}(\mathbf{x}_i) = (1 + g(\mathbf{x}_i)) \cdot M^t(\mathbf{x}_i) \quad (5.4)$$

$$\mathbf{O} = p_{per}(g^{tt}(\mathbf{x}_i); \theta) \quad (5.5)$$

$\mathbf{O}(g^{tt}(\mathbf{x}_i))$ は $T \times 2$ の各タスクにおける確率値を格納した行列である。ここで、本稿で取り扱う Multi-task Learning は、特定の属性属しているか否かを複数認識する。特定のタスク t の特徴マップ $g^{tt}(\mathbf{x}_i)$ を Perception branch $p_{per}(\cdot)$ に入力した時、図 5.5 のように特定のタスク t の確率値には $\mathbf{O}(g^{tt}(\mathbf{x}_i))$ を用いる。この処理を全てのタスクに対して繰り返す。

5.3 評価実験

5.3.1 画像分類の実験概要

提案手法の有効性を調査するために、はじめに CIFAR10, CIFAR100, Street View Home Number (SVHN) [137], ImageNet [138] Dataset を用いて画像分類の評価を行う。入力の画像サイズは、CIFAR10/100, SVHN Dataset では 32×32 pixel, ImageNet Dataset では 224×224 pixels である。クラス数は、CIFAR10 と SVHN Dataset が 10 クラス, CIFAR100 Dataset が 100 クラス, ImageNet Dataset が 1000 クラスである。学習時は、一般的な Data Augmentation を施す。CIFAR10/100 と SVHN Dataset では、4 方向に 4pixel のゼロパディングを施した後に 32×32 のランダムクロッピングする。ここで、

表 5.1: Attention 機構による精度の変化

	$g(\mathbf{x})$	$g(\mathbf{x}) \cdot M(\mathbf{x})$	$g(\mathbf{x}) \cdot (1 + M(\mathbf{x}))$
ResNet20	31.47	30.61	30.46
ResNet32	30.13	28.34	27.91
ResNet44	25.90	24.83	25.59
ResNet56	25.61	24.22	24.07
ResNet110	24.14	23.28	22.82

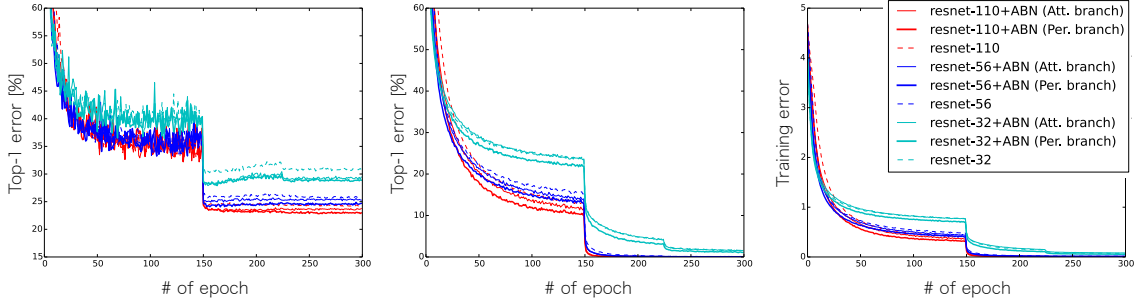


図 5.6: CIFAR100 における学習と評価の top-1 error と学習誤差. **Left** 評価サンプルに対する Top-1 error. **Middle** 学習サンプルに対する Top-1 error. **Right** 学習誤差の推移.

左右反転をランダムに施す. ImageNet Dataset では, 256×256 にリサイズした後に, 224×224 にランダムクロッピングする. 同様に, 左右反転をランダムに施す. 学習, 検証, 評価サンプルの枚数は, CIFAR10/100 の学習サンプルが 60,000 枚と評価サンプルが 10,000 枚, SVHN の学習サンプルが 604,388 枚 (train:73,252, extra : 531,131 枚) と評価サンプルが 26,032 枚, ImageNet の学習サンプルが 1,281,167 枚で 50,000 枚の検証サンプルを用いる.

評価実験では, Stochastic Gradient Descent (SGD) に Momentum 項を追加することでネットワークを最適化する. CIFAR10/100 では, 学習の更新回数を 300 epoch, バッチサイズが 256 で学習する. SVHN Dataset では, 学習の更新回数を 40 epoch, バッチサイズが 256 で学習する. ImageNet Dataset では, 学習の更新回数を 90 epoch, バッチサイズが 256 で学習する. 学習率は初めが 0.1 であり, 50% と 75% の更新時に 0.1 倍する.

5.3.2 画像分類

Attention 機構の解析 式 (5.1) と式 (5.2) の Attention 機構を用いた際の ABN の top-1 error を比較する. 比較に使用するネットワークは, CIFAR100 Dataset で学習した ResNet の {20, 33, 44, 56, 110} 層モデルである.

表 5.1 に, 式 (5.1) と式 (5.2) の Attention 機構を用いた際の ABN の精度を示す. ここで, $g(\mathbf{x}_i)$ は

表 5.2: CIFAR10, CIFAR100, SVHN, ImageNet Dataset における Top-1 error の比較

Dataset	CIFAR10	CIFAR100	SVHN [137]	ImageNet [138]
VGGNet [28]	–	–	–	31.2
VGGNet+BN	–	–	–	26.24*
ResNet [10]	6.43	24.14*	2.18*	23.0
VGGNet+CAM [13]	–	–	–	33.4
VGGNet+BN+CAM	–	–	–	27.42* _(+1.18)
ResNet+CAM	–	–	–	22.11* _(-0.08)
WideResNet [32]	4.00	19.25	2.42*	21.9
DenseNet [33]	4.51	22.27	2.07*	22.2
ResNeXt [34]	3.84*	18.32*	2.16*	22.4
Attention [126]	3.90	20.45	–	21.76
AttentionNeXt [126]	–	–	–	21.20
SENet [36]	–	–	–	21.57
VGGNet+BN+ABN	–	–	–	25.55 _(-0.69)
ResNet+ABN	4.91 _(-1.52)	22.82 _(-1.32)	1.86 _(-0.32)	21.37 _(-1.63)
WideResNet+ABN	3.78 _(-0.22)	18.12 _(-1.13)	2.24 _(-0.18)	–
DenseNet+ABN	4.17 _(-0.34)	21.63 _(-0.64)	2.01 _(-0.06)	–
ResNeXt+ABN	3.80 _(-0.04)	17.70 _(-0.62)	2.01 _(-0.15)	–
SENet+ABN	–	–	–	20.77 _(-0.80)

* は再現実験した際の精度を示している。

従来の ResNet の精度を示している。はじめに、従来の ResNet と $M(\mathbf{x}) \cdot g(\mathbf{x})$ の Attention 機構の精度を比較する。実験結果より、 $M(\mathbf{x}) \cdot g(\mathbf{x})$ の Attention 機構の方が top-1 error が低いことがわかる。次に、2 つの Attention 機構の $M(\mathbf{x}) \cdot g(\mathbf{x})$ と $(1 + M(\mathbf{x})) \cdot g(\mathbf{x})$ の精度を比較する。2 つの Attention 機構を比較した時、 $(1 + M(\mathbf{x})) \cdot g(\mathbf{x})$ の Attention 機構の方が若干精度が良いことが確認できる。ここで、同じ Attention 機構を導入している Residual Attention Network は $M(\mathbf{x}) \cdot g(\mathbf{x})$ の Attention 機構を用いた際に性能低下を引き起こしていた。そのため、提案する ABN は画像分類に有効な領域に反応できているためである。本実験では、 $(1 + M(\mathbf{x})) \cdot g(\mathbf{x})$ の Attention 機構をデフォルトとして使用する。

次に、各更新回数における Top-1 error と学習誤差の推移について比較する。比較で使用するネットワークは、ResNet の 32 層、56 層、110 層モデルである。表示している曲線は、ABN の Attention branch と Perception branch、従来の ResNet である。図 5.6(a) に評価サンプルに対する Top-1 error の推移を示す。ABN の両ブランチの精度と従来の ResNet で比較した時、従来の ResNet より Top-1 error が低いことが確認できる。そして、ABN の Perception branch の精度は Attention branch の精度より

表 5.3: ImageNet Dataset における SENet との比較

	Conventional		+ SENet		+ABN		+ SENet, ABN	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ResNet50	24.11	7.21	23.12 (-0.99)	6.60 (-0.61)	23.12 (-0.99)	6.65 (-0.56)	22.20 (-1.91)	6.30 (-0.91)
ResNet101	22.51	6.42	22.24 (-0.27)	6.13 (-0.29)	21.82 (-0.69)	6.01 (-0.41)	21.15 (-1.36)	5.70 (-0.72)
ResNet152	22.19	6.20	21.90 (-0.29)	6.08 (-0.12)	21.37 (-0.82)	5.72 (-0.48)	20.77 (-1.42)	5.57 (-0.63)

高いことがわかる．図 5.6(b) の学習サンプルに対する Top-1 error を比較した際も，同様の傾向が生じている．図 5.6(c) の学習誤差曲線を確認した時，ABN の両ブランチの学習誤差は従来の ResNet より学習誤差が低い．上位層が同様の構造を持つ Perception branch と従来の ResNet の学習誤差を比較した時，Perception branch の学習誤差の方が低いため，提案手法の Attention 機構がネットワークの学習に有効であることが確認できる．

CIFAR10/100, SVHN の精度 表 5.2 に CIFAR10/100, SVHN, ImageNet Dataset における Top-1 error を示す．CIFAR10/100, SVHN Dataset の評価では，一般的な ResNet モデル，CAM を導入したモデル，提案する ABN を導入したモデルを比較する．表 5.2 の精度は，文献に記されている精度と提案手法の精度を示している．ここで，“*”は再現実装した際の精度を示している．(・) の数値は，ベースラインモデルとの Top-1 error の差を示している．比較に使用するネットワークモデルは，ResNet (depth=110)，DenseNet (depth=100, growth rate=12)，Wide ResNet (depth=29, widen factor=4, drop ratio=0.3)，ResNeXt (depth=29, cardinality=8, widen factor=4) である．ここで，これらの ResNet モデルは Residual block 3 で分割することで ABN を構築する．

表 5.2 より，ResNet, Wide ResNet, DenseNet, ResNeXt において ABN の導入により性能が向上している．CIFAR10 では，特に ResNet と DenseNet の Top-1 error が 6.43% から 4.91%，4.51% から 4.17% に Top-1 error が削減された．加えて，CIFAR100 では全ての ResNet モデルにおいて 0.6% 以上の Top-1 error を削減できている．SVHN Dataset においても Top-1 error を削減できた．

ImageNet の精度 次に，CIFAR10/100, SVHN と同様に画像分類を表 5.2 で ImageNet を評価する．ImageNet の評価では，VGGNet (depth=16)，ResNet (depth=152)，SENet (ResNet152 モデル) を用いて評価する．はじめに，CAM の性能を比較する．CAM の精度は，全結合層を取り除いて GAP を導入する必要があるため，認識性能が低下しやすい問題がある．同様に，VGGNet に Batch Normalization (BN) [30] を導入したモデルにおいても，VGGNet の性能が少し低下している．対照的に，ResNet に CAM を導入した場合，精度がほぼ変化しなかった．ResNet の上位層は，CAM と同様に GAP と最終層に全結合層を配置した構造となっている．CAM は，最後の Residual Block の stride を 1 に設定し， $K \times 1 \times 1$ の畳み込み層を導入するだけで導入できる．そのため，CAM を導入した ResNet は GAP と全結合層の悪影響が少なく，性能低下が起こらなかったと考えられる．一方で，提案する ABN は VGGNet と ResNet の認識精度をどちらも向上させることができる．

次に，従来の Attention 機構を取り入れた Residual Attention Network と SENet の比較をする．SENet の導入により，従来の ResNet の精度を 23.00% から 21.90% まで Top-1 error まで削減できた．しかし，提案する ABN は 23.00% から 21.30% まで性能を削減でき，SENet より精度が向上している．加

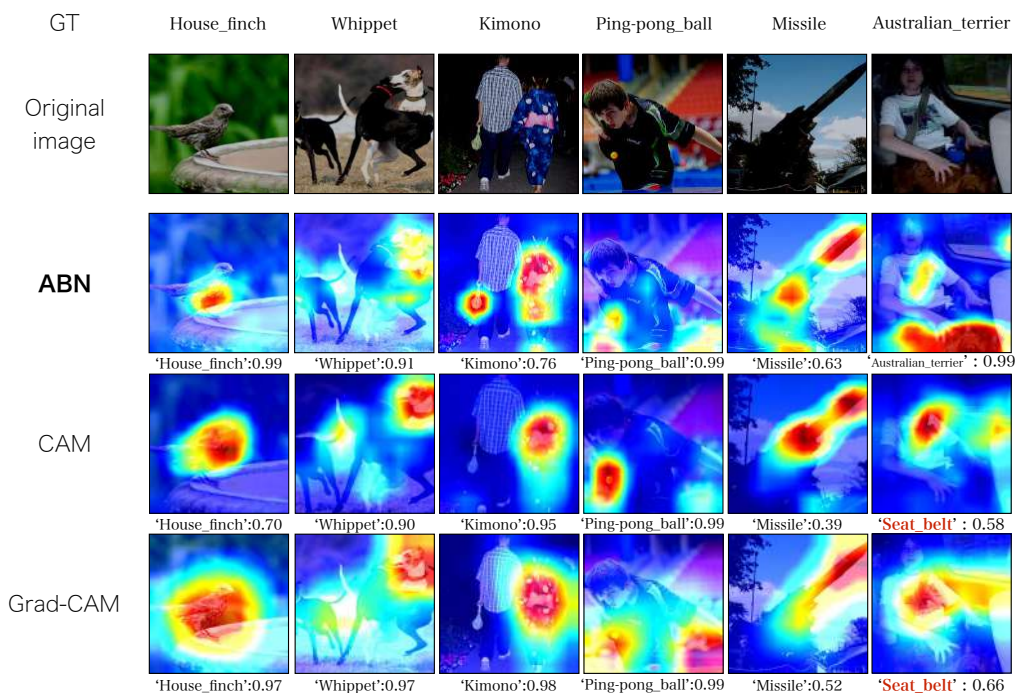


図 5.7: CAM, Grad-CAM, Attention Branch Network による Attention map の比較結果

えて、ABN は SENet と併用することができる。SENet と ABN を併用することで、従来の ResNet の精度を 23.00% から 20.77% まで Top-1 error を削減できる。この傾向は、表 5.3 のように、様々な層数の ResNet においても精度を向上させることができる。また、Residual Attention Network は ResNet ベースの場合に 21.76%、ResNeXt ベースの場合に 21.20% の誤認識率であった。しかし、今回の実験で比較した手法の中では ResNet152+SE, ABN が 20.77% であり、Residual Attention Network より精度が高いことが確認できる。

Attention map の可視化 ImageNet Dataset における注視領域の比較結果を図 5.7 に示す。ここで、CAM は ResNet152 をベースにネットワークを構築し、Grad-CAM は従来の ResNet152 を用いて Attention map を獲得する。

図 5.7 より、CAM, Grad-CAM, ABN の注視領域はほぼ同様であることが確認できる。図 5.7 の 1 番目の場合、3 つのモデルは “Violin” を認識し、“Violin” の位置に強く Attention map が反応している。2 番目の例も同様に、“Cliff” の領域に対して Attention map が強く反応している。一方で、3 番目の画像には “Australlian_terrier” と “Seat_belt” が含まれている。この例では、CAM と Grad-CAM (ResNet152) では “Seat_belt” と認識している。しかし、ABN は “Australlian_terrier” と認識しており、かつ “Australlian_terrier” と “Seat_belt” に Attention map が反応している。そのため、ABN は画像中に複数物体が含まれている場合でも、複数の物体に Attention map を反応することができる。

Attention branch は、CAM をベースに Attention map を出力する。そのため、Attention branch が認識したクラスに応じてどのような Attention map が出力されるのかを把握する必要がある。Attention

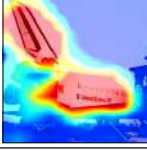
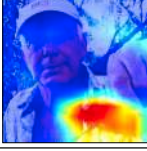
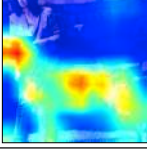
GT	Space shuttle	Rhinoceros beetle	Kingsnake	Irish wolfhound
Original image				
Attention map				
Attention branch	○ Space shuttle	○ Rhinoceros beetle	✗ Sea snake	✗ Scottish deerhound
Perception branch	○ Space shuttle	○ Rhinoceros beetle	○ Kingsnake	○ Irish wolfhound

図 5.8: Attention branch の出力結果に対する Attention map の例

表 5.4: CompCars Dataset における識別率の比較

task	車種 [%]	メーカー [%]
VGG16	85.9	90.4
ResNet101	90.2	90.1
VGG16+ABN	90.7	92.9
ResNet101+ABN	97.1	98.1

branch の出力した Attention map と認識したクラスの例を図 5.8 に示す。図 5.8 のように、Attention map が認識に成功した場合と失敗した場合で関係なく、特定の物体に Attention map が強く反応していることがわかる。ABN は、各クラスにおける注視領域の特徴マップを $1 \times 1 \times 1$ の畳み込み層で統合するため、Attention map を出力する際に Attention branch の認識結果に依存しない。そのため、Attention branch で誤認識をした場合でも、図 5.8 のように注視領域を獲得し、Perception branch で最終的な認識結果を得ることができる。

最後に、図 5.9 に ImageNet Dataset における ABN の Attention map の例を示す。図 5.9 のように、特定の物体に対して Attention map が強く反応しており、背景等に対しては Attention map が強く反応しない。

5.3.3 他の画像認識タスクへの応用

詳細識別 詳細識別では、Comprehensive Cars (CompCars) Dataset [139] を用いて評価する。実験では、文献 [139] と同様に 136,726 枚の自動車画像を用いて 432 種類の車種と 75 種類のメーカーを認識する。入力する画像は、学習サンプルの bbox の平均からアスペクト比の平均を算出し、 323×224 pixel

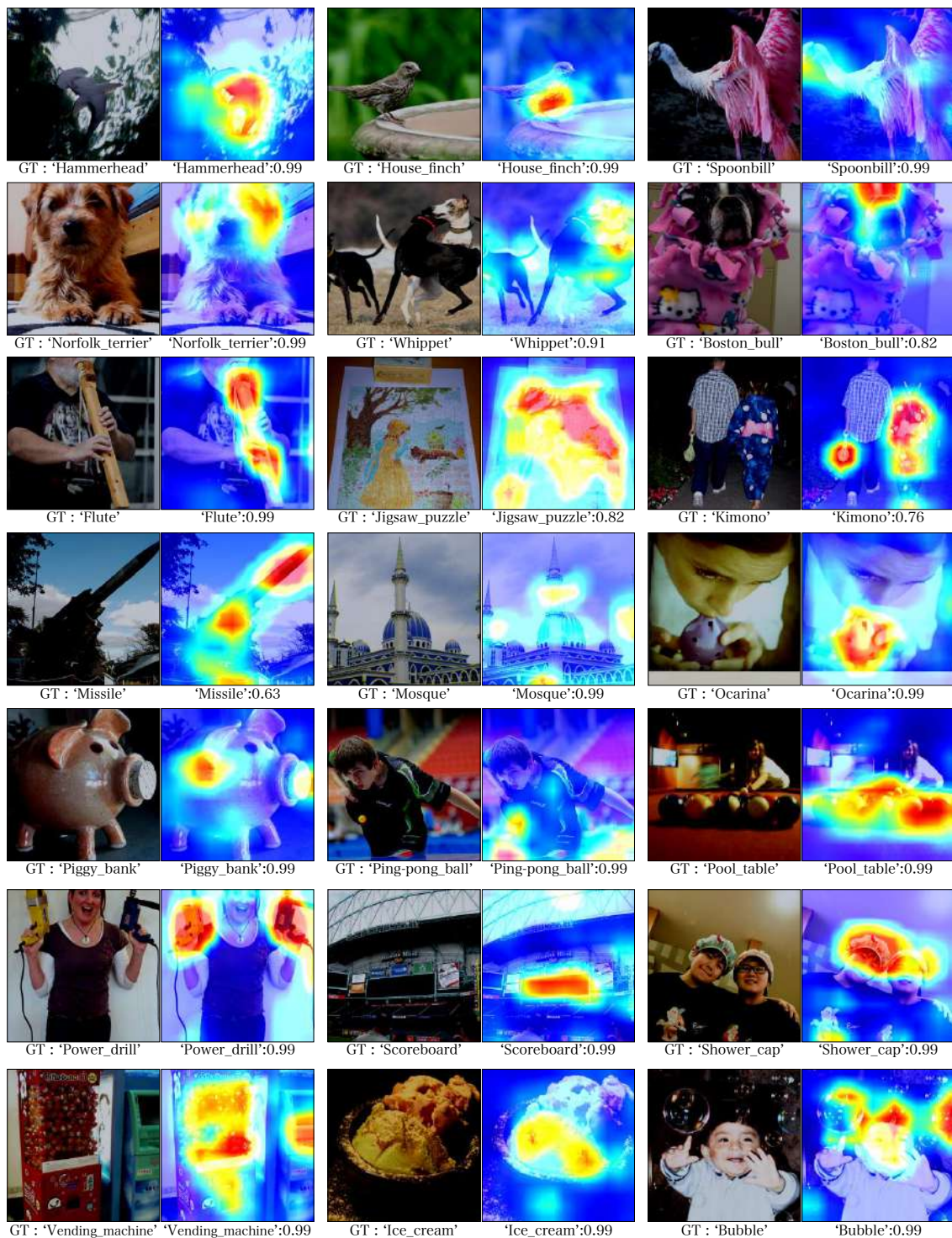


図 5.9: ImageNet Dataset における ABN の Attention map の例

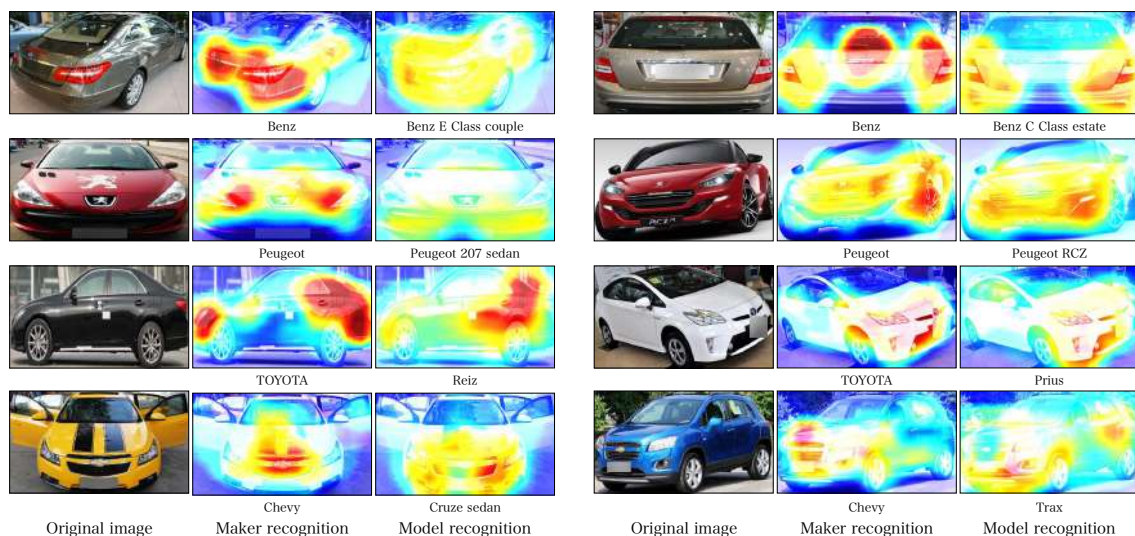


図 5.10: CompCars Dataset における ABN の Attention map の例

表 5.5: CityPerson Dataset における検出精度の比較

	Faster R-CNN	Faster R-CNN+ ABN
Reasonable	24.08	23.45
Reasonable small	58.43	55.61
Reasonable occlusion	56.15	54.58
All	48.49	48.64

にリサイズする．使用するモデルは VGG16 と ResNet101 であり，SGD+Momentum で学習する．学習の更新回数は 50epoch であり，Batch size は 32 である．

CompCar Dataset による車種認識とメーカー認識の認識結果を，表 5.4 に示す．車種認識の場合，従来の VGG16 と ResNet101 の認識精度は，それぞれ 85.9% と 90.2% であった．しかし，ABN を導入することで，VGG16 と ResNet101 の認識精度がそれぞれ約 5% と約 6% 向上した．メーカー認識においても同様の傾向が見られ，提案手法の導入により VGG16 と ResNet101 の認識精度が最大で約 2% と約 7% 向上した．これらの結果から，詳細識別においても ABN が有効であることが確認できる．獲得した Attention map を図 5.10 に示す．車種認識とメーカー認識で使用している学習，評価画像は同一であるが，獲得できた Attention map は認識するタスクに応じて変化している．車種認識では自動車の特徴的な領域に強く反応するが，メーカー認識ではメーカー認識では広い範囲で自動車全体に強く反応している．

自動車制御 従来の回帰モデルと ABN を導入した回帰モデルの推定精度の比較結果を，表 5.6 に示す．回帰問題においては，従来の回帰モデルと ABN を導入した回帰モデルの推定誤差は，ステアリングとスロットルで Mean Squared Error が 0.001 未満であり，大きな性能差が見られなかった．次

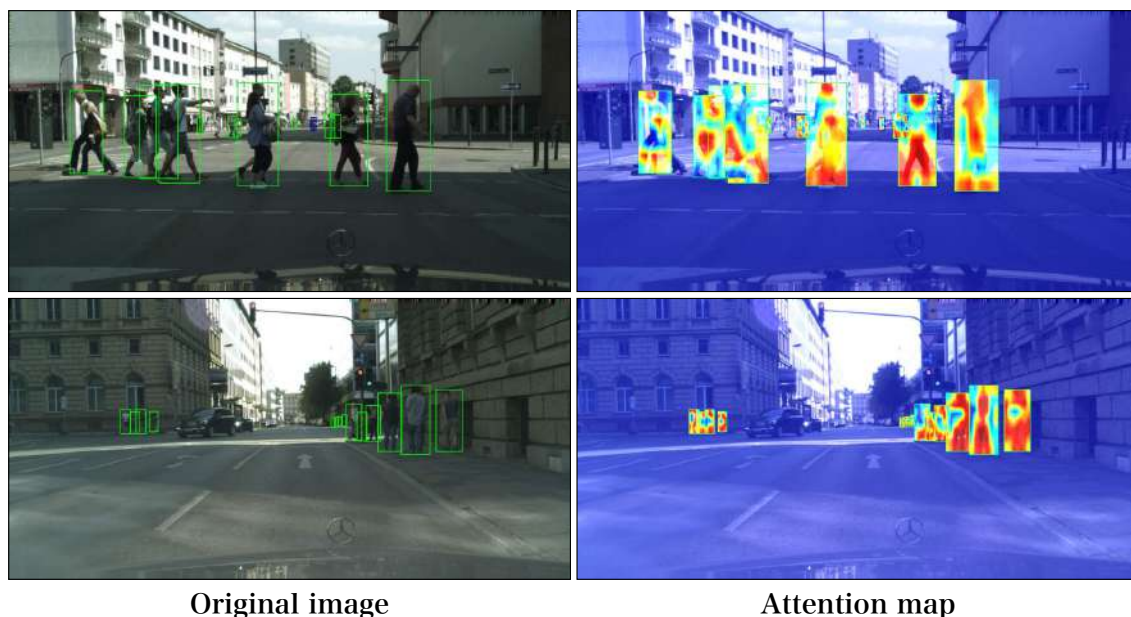


図 5.11: CityPerson Dataset における ABN の Attention map の例

表 5.6: Mean squared error on vehicle control

	Steering	Throttle
Reg. Model	0.014810	0.019124
Reg. Model + ABN	0.015480	0.019331

に，獲得した Attention map を図 5.9 に示す．図 5.9 の左の例では，前方にトラックが停車しているため，前方のトラックのテイルランプ (ブレーキランプ) に対して Attention map が強く反応している．また，図 5.9 の中心と右の例では，右左折する際の Attention map を可視化している．左折する場合は右側の白線に Attention map が強く反応しており，右折する場合は左側の白線の Attention map が強く反応している．

歩行者検出 歩行者検出では，CityPerson Dataset [78] を用いて評価する．学習サンプルでは 2,975 枚，検証サンプルでは 500 枚使用する．また，実験では VGG16 をベースにした Faster R-CNN を用いる．ここで，使用する Faster R-CNN の Fast R-CNN では“pedestrian”，“cyclist”，“sitting person”の 3 種類の歩行者を識別する．

Cityperson Dataset の評価結果を，表 5.5 に示す．表 5.5 より，Faster R-CNN に提案手法にを導入することで，全ての評価項目の Miss rate が削減できた．特に，“Reasonable small”と“Reasonable occlusion”においてはそれぞれ約 3% と約 2% の Miss rate を削減している．次に，獲得した Attention map を図 5.12 に示す．歩行者検出では，歩行者の頭部と両足に対して Attention map が強く反応した．頭部は歩行者の姿勢の変化に頑健な箇所であり，かつ両足は person, cyclist, sitting person を識別する際に変化が大きい箇所である．そのため，頭部と両足に対して Attention map が強く反応した

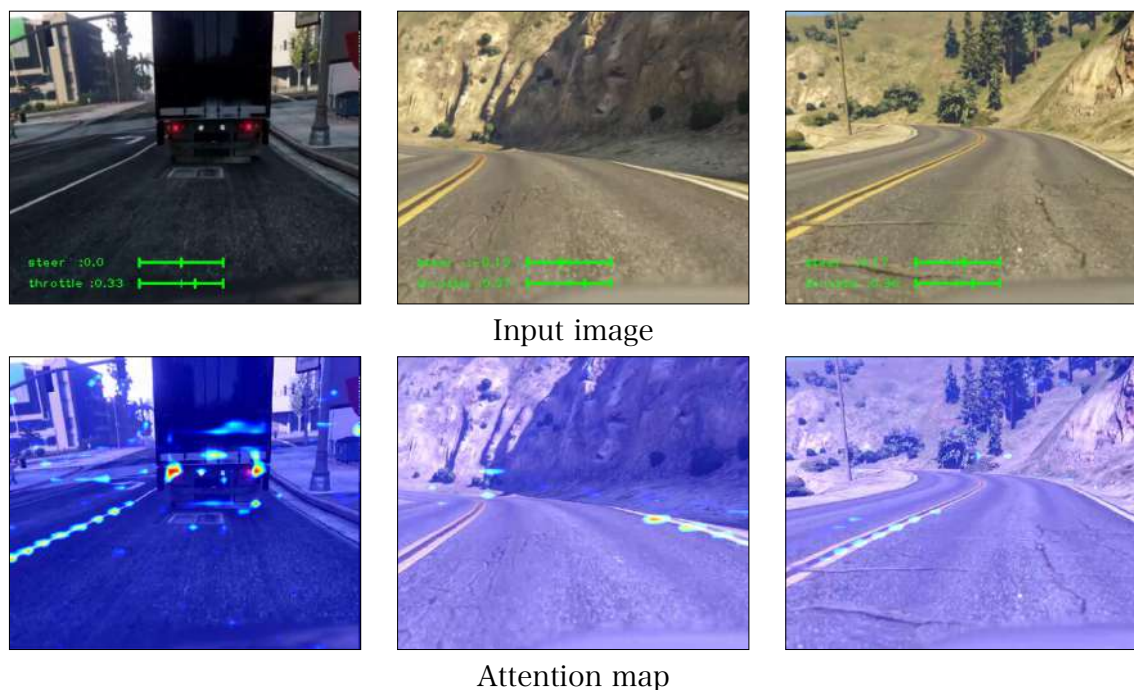


図 5.12: 自動車制御における Attention map の例

と考えられる。

Multi-task Learning Multi-task Learning における評価では, CelebA Dataset を用いて 40 種類の顔属性認識の評価を行う。CelebA Dataset は, 40 種類の顔属性のラベルが付与されたデータセットであり, 顔属性認識の評価に一般的に用いられるデータセットである。評価に使用するネットワークは ResNet50 に Multi-task Learning を導入したモデルであり, ミニバッチサイズを 16, 学習率を 0.01, Momentum+SGD で 10epoch 学習する。比較に使用する手法は, ResNet101, FaceTracer [140], PANDA-1 [142], LNet+ANet [141], Mixed Objective Optimization Network (MOON) [22] である。

CelebA Dataset による評価結果を表 5.7 に示す。表 5.7 より, 提案する ABN の導入により従来の顔属性認識手法より認識精度が向上している。FaceTracer, PANDA-1, LNet+ANet の精度はほとんどの顔属性認識において精度が上回っている。ResNet と MOON で比較した時, 特に “Arched Eyebrows”, “Big Lips”, “Pointy Nose”, “Wavy Hair”, “Wearing Earrings”, “Wearing Necklace” の顔属性認識の精度が向上している。次に, 獲得した Attention map を図 5.13 に示す。例に示す顔画像では, それぞれ特定の顔属性に該当する場合に対して Attention map が強く反応している。

5.4 まとめ

本稿では, 深層学習における注視領域を Top-down に獲得できる ABN を提案した。ABN は, CAM をベースに獲得した注視領域を Attention 機構へ応用することで, 認識性能を改善しつつ, 様々な画像

表 5.7: CelebA Dataset における認識精度の比較

Task	5.o.Clock_Shadow	Arched_Eyebrows	Attractive	Bags_Under_Eyes	Bald	Bangs	Big_Lips	Big_Nose	Black_Hair	Blond_Hair	Blurry	Brown_Hair	Bushy_Eyebrows	Chubby	Double_Chin	Eyeglasses	Goatee	Gray_Hair	Heavy_Makeup	High_Cheekbones
FaceTracer [140]	85	76	78	76	89	88	64	74	70	80	81	60	80	86	88	98	93	90	85	84
LNets+ANet [141]	91	79	81	79	98	95	68	78	88	95	84	80	90	91	92	99	95	97	90	87
PANDA-I [142]	88	78	81	79	96	92	67	75	85	93	86	77	86	86	88	98	93	94	90	86
MOON [22]	94.03	82.26	81.67	84.92	98.77	95.80	71.48	84.00	89.40	95.86	95.67	89.38	92.62	95.44	96.32	99.47	97.04	98.10	90.99	87.01
ResNet+MTL	94.39	82.66	81.00	84.03	98.98	95.75	70.57	83.29	88.68	95.23	95.82	86.42	91.99	95.41	96.24	99.68	97.37	98.14	90.86	86.22
ABN+MTL	94.53	84.26	81.36	85.07	98.70	95.68	72.46	81.96	89.56	96.03	96.24	88.55	92.38	95.17	96.51	99.54	97.33	98.11	91.20	86.80

Task	Male	Mouth_Slightly_Open	Mustache	Narrow_Eyes	No_Beard	Oval_Face	Pale_Skin	Pointy_Nose	Receding_Hairline	Rosy_Cheeks	Sideburns	Smiling	Straight_Hair	Wavy_Hair	Wearing_Earrings	Wearing_Hat	Wearing_Lipstick	Wearing_Necklace	Wearing_Necktie	Young
FaceTracer [140]	91	87	91	82	90	64	83	68	76	84	94	89	63	73	73	89	89	68	86	80
LNets+ANet [141]	98	92	95	81	95	66	91	72	89	90	96	92	73	80	82	99	93	71	93	87
PANDA-I [142]	97	93	93	84	93	65	91	71	85	87	93	92	69	77	78	96	93	67	91	84
MOON [22]	98.10	93.54	96.82	86.52	95.58	75.73	97.00	76.46	93.56	94.82	97.59	92.6	82.26	82.47	89.60	98.95	93.93	87.04	96.63	88.08
ResNet+MTL	98.33	93.64	96.88	86.94	96.24	73.27	96.69	74.62	93.22	94.52	97.87	92.49	83.17	83.16	89.88	99.08	93.69	86.93	96.84	87.78
ABN+MTL	98.19	93.72	96.92	87.22	95.98	75.13	96.70	77.16	93.57	95.01	97.77	91.82	83.38	84.13	90.70	99.01	94.06	87.98	96.70	86.18



図 5.13: CelebA Dataset における ABN の Attention map の例

認識タスクへ応用可能である。実験では、画像分類において従来の Attention 機構モデルである SENet を上回る性能を達成している。また、詳細認識においても高い性能を実現し、従来のネットワークモデルより最大で約 8% の性能向上を実現している。そして、分類問題だけでなく、回帰問題や物体検出、Multi-task Learning にも応用できることを評価実験により示した。

6 章では、教師ラベルを学習で使わない強化学習に ABN を応用する。強化学習への応用により、エージェントを制御する際の注視領域を可視化する。

第6章

Attention機構による深層強化学習の視覚的解析

深層強化学習は、ビデオゲームやロボットの自律動作獲得等の様々なタスクにおいて高い性能を達成している [143, 144, 145]. 深層強化学習では膨大な数の最適なパラメータを学習により獲得するために、膨大なパターンの行動を試行錯誤しながら学習する. このように膨大な数のパラメータ学習により獲得することで、複雑なタスクの攻略を可能としている. しかし、内部の演算は複雑なために出力された行動に対する判断根拠を理解することが困難である.

Computer Vision 分野における深層学習の解析では、5.1.1 章で述べたネットワークが推論時に注視した領域をマップで表現した Attention map により解析する視覚的説明が一般的に用いられている. 深層強化学習の分野では、Bottom-up な視覚的説明を用いることで深層強化学習を解析する手法が提案されている. しかし、Bottom-up な視覚的解析は勾配情報を計算するために、順伝播により推論結果を獲得した後に逆伝播を行う必要がある. 一方、Top-down な視覚的解析は、ネットワークの内部の応答値を用いて Attention map を獲得する. そのため、Top-down な視覚的説明は順伝播の過程のみで Attention map を獲得できる.

そこで本研究では、深層強化学習における Top-down な Attention map を獲得するために、Attention Branch Network (ABN) を深層強化学習へと応用する. ABN は、教師あり学習の画像認識タスクに用いられている. そのため、教師ラベルが存在しない強化学習に直接適用するのが困難である. この問題を解決するために、本稿では Actor-Critic 法と ABN のネットワーク構造に着目する. Actor-Critic 法は 1 つのネットワークから方策と状態価値を出力する構造となっており、学習時に状態価値を用いてエージェントが出力した行動を改善する. 提案手法では、状態価値を ABN における教師ラベルの代わりの基準値として用いることで、強化学習における Attention map を Top-down に獲得できる. Atari 2600 ゲームを用いた実験により、提案手法では出力される行動に対する判断根拠を Attention map から解析できることを示す. また、ABN と同様に Attention 機構を導入し、行動を決定することでエージェントの制御における精度が向上することを示す.

本研究の貢献は以下のようになる.

- 深層強化学習における Top-down な視覚的解析を提案する. 従来の強化学習における視覚的解析では Bottom-up な手法が用いられており、逆伝播により勾配情報を計算する必要があった. 一方で、提案手法は順伝播の過程で Attention map を獲得できる.
- 獲得した Attention map により、エージェントの行動決定の要因を解析することを可能とする.

Atari 2600 ゲームを用いた実験において、どのような情報が行動決定に影響を与えているかを議論する。

- 獲得した Attention map は判断根拠の解析だけでなく、Attention 機構によりエージェントの行動決定に活用する。高報酬な領域に強く反応した Attention map を導入することで性能向上を実現する。

6.1 関連研究

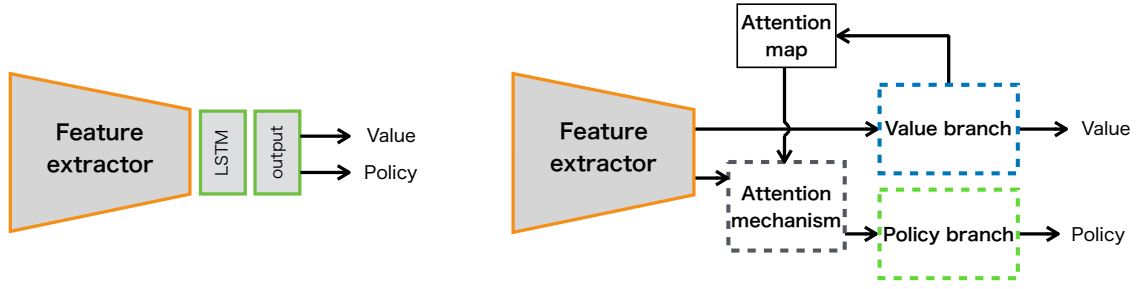
6.1.1 深層強化学習

エージェントの自律動作獲得では、Q-Learning [146] や Actor-Critic 法 [147] 等をベースにした深層強化学習法が、様々なエージェントの動作獲得において高い性能を達成している [143, 144, 145]。深層強化学習で用いられる学習アルゴリズムは、主に Value-base, Policy-base, Actor-Critic 法の3つが一般的に用いられる。これらの学習法で用いられるモデルは、ネットワークの出力値に違いがある。Q-Network [148] や Deep Q-Network (DQN) [149], Double DQN [150] 等の Value-base のネットワークは、状態 s_t で起こした行動 a_t の価値を示した行動価値 $Q(a_t|s_t; \theta)$ が出力される。Trust Region Policy Optimization (TRPO) [151] や Proximal Policy Optimization (PPO) [152] 等の Policy-base のネットワークは、状態 s_t における方策 $\pi(a_t|s_t; \theta)$ を直接出力し、Policy Gradient [153] によりネットワークを最適化する。そして、A3C [154] や Unsupervised Reinforcement and Auxiliary Learning (UNREAL) [155] 等の Actor-Critic 法におけるネットワークは、方策 $\pi(a_t|s_t; \theta)$ と状態 s_t における状態価値 $V(s_t; \theta)$ を出力する。特に、Actor-Critic 法における2つの出力は独立しており、出力される方策 $\pi(a_t|s_t; \theta)$ と状態価値 $V(s_t; \theta)$ を干渉させないことで、効率よくネットワークを学習することができる。

Value-base の強化学習法 Value-base のネットワークは、行動価値 $Q(a_t|s_t; \theta)$ を出力する。Value-base の強化学習では、式 (6.1) によりネットワークを更新する。ここで、時刻 t に対する行動を a_t 、状態を s_t 、即時報酬を r 、ネットワークのパラメータを θ 、報酬の割引率を γ とする。式 (6.1) のように、Value-base の強化学習法では最大となる行動価値 $Q(a_t|s_t; \theta)$ を用いることで、ネットワークを更新する。

$$\nabla \theta = \alpha \left(r + \gamma \max_a Q(a_t|s_{t+1}; \theta) - Q(a_t|s_t; \theta) \right) \quad (6.1)$$

Actor-Critic の強化学習法 Actor-Critic 法は、方策 $\pi(a_t|s_t; \theta)$ と状態価値 $V(s_t; \theta)$ を出力する。方策関数では、状態 s_t における行動を示している。状態価値関数は、状態 s_t において獲得できる報酬を、ネットワーク自身で評価する関数である。そのため、獲得できる報酬が高そうな状態 s_t の場合に、高い値が出力される。Actor-Critic 法のネットワークは、式 (6.2) によりパラメータを更新する。Actor-Critic 法では、式 (6.2) のように k ステップ先の状態の報酬を、状態価値関数により学習できる。この学習方法は Advantage と呼ばれており、Value-base の強化学習法とは異なり数ステップ先の



(a) Actor-Critic法

(b) 提案手法のActor-Critic法

図 6.1: 従来の Actor-Critic 法と提案手法の Actor-Critic 法の構造

状態を加味しながらネットワークを学習できる。そして、式 (6.2) のように方策 $\pi(\mathbf{a}_t|\mathbf{s}_t; \theta)$ は log を施すことで、更新値を出力する。

$$\nabla \theta = \log \pi(\mathbf{a}_t|\mathbf{s}_t; \theta) \cdot \left(\sum_{k=1}^{i=0} \gamma^i r_{t+i} + \gamma^k V(\mathbf{s}_{t+k}; \theta) - V(\mathbf{s}_t; \theta) \right) \quad (6.2)$$

本研究では、Actor-Critic 法が出力する方策 $\pi(a_t|s_t; \theta)$ と状態価値 $V(s_t; \theta)$ に対して着目する。特に、状態価値は提案手法において視覚的説明における Attention map を出力する際に使用する。これは、ABN のような視覚的説明における Attention map を Attention 機構に応用した学習が可能となり、高報酬な領域を教師ラベル無しで強調して提案手法のネットワークを学習できる。加えて、提案手法を A3C へ応用することが可能である。

6.1.2 強化学習における可視化の関連研究

強化学習の分野においても、強化学習がどのような判断根拠でエージェントを制御したのかを把握する研究が行われている [156, 157, 4]。強化学習の分野では、Saliency map や Attention 機構を用いる手法が数多く提案されている。Samuel らの研究では、マスクとガウシアンフィルタを施した摂動画像を用いることで、強化学習における Saliency map を獲得している [4]。畳み込み層をベースに構築された強化学習法は、CNN で用いられる Bottom-up の解析手法を適応できる手法がある。しかし、Bottom-up な解析手法は勾配を獲得するために逆伝播の処理を行う必要がある。Samuel らの手法は、これらの Bottom-up の解析手法より高精細に Saliency map を獲得できるが、同様に勾配を獲得する必要がある。

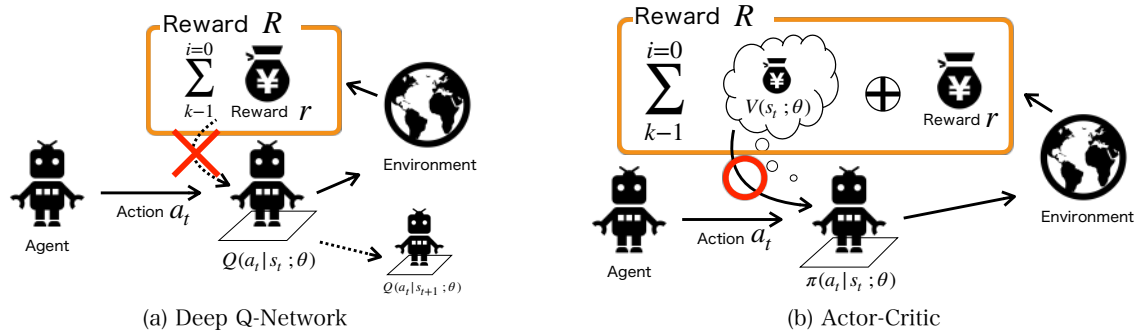


図 6.2: Deep Q-Network と Actor-Critic 法の関係

6.2 提案手法

Top-down に深層強化学習における注視領域の獲得を行い、高性能に自律動作を獲得できるネットワークを提案するために、ABN を深層強化学習へ応用する。ABN を深層強化学習に応用する際に、教師あり学習と深層強化学習の学習方法の違いが生じる。ABN は、ネットワークの出力と教師ラベルを用いて教師あり学習によりネットワークを最適化する。しかし、強化学習は環境から得られた報酬からネットワークを最適化するため、教師ラベルが存在しない。そのため、ABN を強化学習へ直接応用するのが困難である。ここで、強化学習法である Actor-Critic 法のネットワーク構造と学習方法について着目する。Actor-Critic 法の学習は、エージェントの方策 $\pi(\mathbf{a}_t|\mathbf{s}_t; \theta)$ に加えて状態価値 $V(\mathbf{s}_t; \theta)$ を出力し、ネットワークを最適化する。Actor-Critic 法のネットワークが出力する状態価値 $V(\mathbf{s}_t; \theta)$ は、状態 $\mathbf{s}_t$ が報酬にどの程度貢献しているかを数値的に示しており、この状態価値 $V(\mathbf{s}_t; \theta)$ から収益報酬 R を求めて式 (6.2) の更新式によりネットワークを更新する。式 (6.2) における $\sum_{k=1}^{i=0} \gamma^i r_{t+i} + \gamma^k V(\mathbf{s}_{t+k}; \theta)$ は獲得した収益報酬 R を示しており、式 (6.2) と図 6.2(b) のように Actor-Critic 法で出力した状態価値 $V(\mathbf{s}_t; \theta)$ が直接ネットワークの学習に干渉している。ここで、 γ は報酬の割引率、 k が報酬を考慮するステップ数、 θ がネットワークのパラメータを示している。そのため、Actor-Critic 法のネットワークが出力する状態価値 $V(\mathbf{s}_t; \theta)$ は、教師あり学習における教師ラベルの代わりの指標として考えることができる。一方で、DQN 等の Value-base の強化学習法は、出力した行動から行動価値 $Q(\mathbf{a}_t|\mathbf{s}_t; \theta)$ を出力して報酬を獲得する。そのため、図 6.2(a) のように獲得した報酬が直接ネットワークの出力に干渉していない。

Actor-Critic 法のネットワークは、図 6.1(a) のように方策 $\pi(\mathbf{a}_t|\mathbf{s}_t; \theta)$ と状態価値 $V(\mathbf{s}_t; \theta)$ を Multi-task Learning のように出力する。そのため、図 6.1(b) のような ABN のブランチ構造を取り入れることで、Attention branch から状態価値 $V(\mathbf{s}_t; \theta)$ を出力し、Perception branch から方策 $\pi(\mathbf{a}_t|\mathbf{s}_t; \theta)$ を出力する。これにより、高報酬な領域を Attention map で表現させた上で Attention 機構に応用できる。これらの事を踏まえ、本研究では ABN ベースの Attention 機構を導入した Actor-Critic 法を提案する。本章では、提案手法における学習方法やネットワーク構造について詳細に述べる。

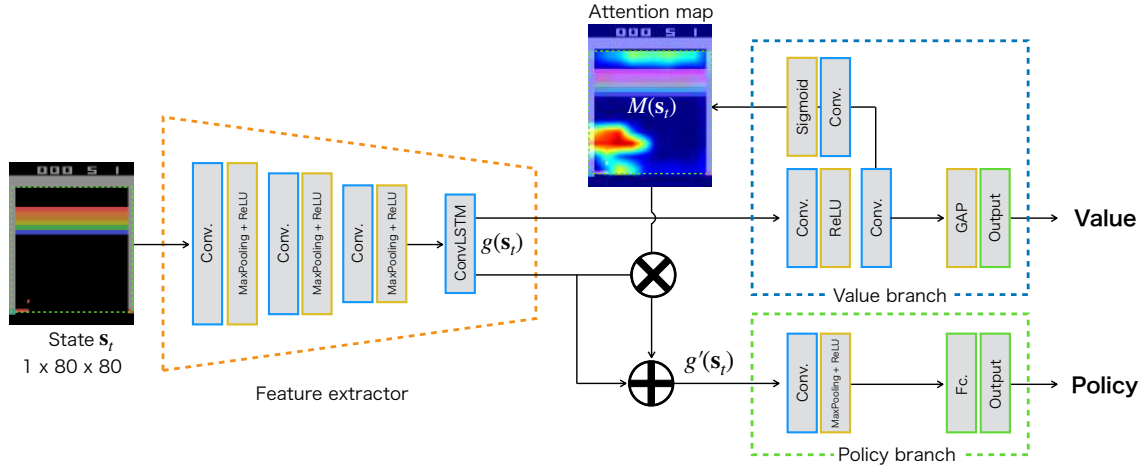


図 6.3: 提案手法のネットワーク構造

6.2.1 提案手法のネットワーク構造

Actor-Critic 法に ABN を導入するために、ベースとなる 3 つのモジュールを構築する。Actor-Critic 法に ABN を導入する際に、図 6.1 のようなネットワークを構築する。図 6.1(b) の構造は、ABN を模した構造になっており、Attention branch に位置するブランチは状態価値 $V(s_t; \theta)$ を出力する。そして、Perception branch に位置するブランチは方策 $\pi(a_t|s_t; \theta)$ を出力する。ここで、提案手法の Attention branch は状態価値 $V(s_t; \theta)$ を出力するため“Value branch”と呼称し、Perception branch は方策 $\pi(a_t|s_t; \theta)$ のみ出力するため“Policy branch”と呼称する。Value branch は、状態価値 $V(s_t; \theta)$ を出力する過程で高報酬な領域に強く反応する Attention map を獲得できる。そして、獲得した Attention map は Attention 機構に応用され、Attention map を施した特徴マップを Policy branch へ入力することで、方策 $\pi(a_t|s_t; \theta)$ を出力する。

6.2.2 Feature extractor

Feature extractor は、図 6.3(a) のように 3 層の畳み込み層をベースに構築する。Feature extractor の役割は従来の ABN と同様であり、入力画像から畳み込み層を用いて特徴マップを抽出する。抽出した特徴マップは、後層の 2 つのブランチに入力される。

Actor-Critic 法では、ネットワークの上位層に Long Short Term Memory (LSTM) [127] を設置する。LSTM により時系列を考慮した特徴抽出が可能となり、エージェントの自律動作獲得の精度を向上できる [154]。しかし、Feature extractor の後層の Value branch は畳み込み層をベースにブランチを構築するため、全結合層をベースに構築された LSTM を導入することができない。この問題を解決するため、LSTM の代わりに Convolutional LSTM (convLSTM) [158] を Feature extractor の最終層へ導入する。convLSTM の導入により、時系列を考慮した特徴マップを後層の Value branch へ入力できる。

6.2.3 Value branch

Value branch は、ABN の Attention branch をベースにブランチを構築する。ABN の Attention branch は、畳み込み層から抽出した特徴マップを使用し、GAP の応答値から Attention map を獲得する。同様に、Value branch も畳み込み層をベースにブランチを構築する。Attention branch をベースに Value branch を構築することで、状態価値 $V(s_t; \theta)$ を出力する過程で Attention map を生成できる。

Value branch の構造を、図 6.3(b) に示す。Feature extractor から抽出した特徴マップを受け取り、2 層の畳み込み層を施し、GAP で特徴マップをダウンサンプリングして全結合層へ入力することで、状態価値 $V(s_t; \theta)$ を出力する。ABN の Attention branch では、GAP を施す前の畳み込み層のチャンネル数は認識するクラスの数に設定する必要がある。対して、Value branch は状態価値 $V(s_t; \theta)$ のみ出力する。ABN のようにクラス数枚の特徴マップを出力する場合、1 チャンネルの特徴マップを出力する畳み込み層を複数設置する必要がある、余分な畳み込み層を積み重ねることになる。また、この余分な畳み込み層は出力する状態価値 $V(s_t; \theta)$ と Attention map の挙動を不安定にさせる。そのため、ResNet の上位層のように GAP と全結合層を用いることで、よりシンプルにブランチを構築することができる。ここで、GAP の直前の畳み込み層のチャンネル数は任意の数である 128 に設定する。Attention map は、従来の ABN と同様に $1 \times 1 \times 1$ の畳み込み層を施した後に、Sigmoid 関数を施すことで生成する。

6.2.4 Policy branch

Policy branch では、Attention 機構により Value branch の Attention map を Feature extractor の特徴マップに反映し、方策を出力する。Policy branch は、畳み込み層と全結合層をベースに図 6.3(c) のようにブランチを構築する。Policy branch における Attention 機構は、ABN の Perception branch と同様の処理を行っており、出力する対象が各クラスにおける確率から方策 $\pi(a_t | s_t; \theta)$ に変更している。

Attention 機構の構造は、ABN と同様に式 (6.3) の方法を用いて、Attention map $M(s_t)$ と特徴マップ $g(s_t)$ から新たな特徴マップ $g'(s_t)$ を出力する。式 (6.3) の Attention 機構は、Attention map を施した特徴マップ $M(s_t) \cdot g(s_t)$ と、Attention map を施していない特徴マップ $g(s_t)$ を加算している。単純な Attention map と特徴マップの乗算のみの場合、Attention map に含まれるゼロ成分で特徴マップが消失する問題が発生する。そのため、Attention map を乗算した特徴マップを加算することで、特徴マップの消失を防ぐと共に、Attention map が強く反応した領域の特徴を強調できる。

$$g'(s_t) = (1 + M(s_t)) \cdot g(s_t) \quad (6.3)$$

6.2.5 A3C への応用

本研究では、提案手法の Attention 機構を、Actor-Critic 法の代表的な手法である A3C へ導入する。A3C は Actor-Critic 法をベースに、数ステップ先の報酬を考慮した Advantage training と、マルチエー

ジェントな分散並列学習である Asynchronous を取り入れた強化学習法である。提案手法の ABN を導入した Actor-Critic 法を A3C へ導入する場合、A3C の Global Network と worker のネットワークを図 6.3 に置き換える。提案手法は、A3C のネットワークを変更するだけで応用できる。そのため、A3C の他の要素である Asynchronous と Advantage training の処理に変更を加える必要がない。

6.3 評価実験

提案手法の有効性を示すために、評価実験を行う。本実験では、OpenAI gym [159] における 47 種類の Atari ゲームの環境を用いて評価する。画像入力時は、入力画像に対して特定の領域を切り出し、 80×80 のグレースケール画像に変換し、 $[0, 1]$ に正規化してネットワークに入力する。ここで、切り出し位置はゲーム毎によって異なる。学習パラメータは、学習率を 0.0001, worker 数を 32, γ を 0.99, 最大エピソード数を 10,000 に設定する。学習済みモデルを比較する場合、最大エピソード数を 10,000 回に設定し、100 回評価した際の平均スコアを用いて評価する。

実験に使用する従来の A3C のネットワーク構造を、表 6.1 に示す。従来の A3C は、表 6.1 のように 4 層の畳み込み層と LSTM, 2 つの出力層からネットワークを構築する。提案手法の A3C は、表 6.1 をベースに表 6.1 のようにネットワークを構築する。ここで、FE は Feature extractor, VB が Value branch, PB が Policy branch を示す。しかしながら、ベースで使用する表 6.1 の 4 層目の畳み込み層から出力される特徴マップは、 4×4 pixel と非常に低解像度である。提案手法が出力する Attention map は注視領域を表現できる程度の解像度を必要とするため、特徴マップの解像度を拡大する処理を導入する。1 つは、3 層目と 4 層目の Max pooling を除去する。もう一つは、2 層目と 3 層目の畳み込み層のカーネルサイズを 3×3 に変更する。2 層目と 3 層目の畳み込み層は、パディングを 1 で設定しているため、 5×5 と 4×4 のカーネルを畳み込んだ際に特徴マップが縮小される。この 2 つの処理を施すことで、表 6.1 のように Attention map を 10×10 pixel まで拡大できる。

6.3.1 性能比較

従来の A3C と提案手法の A3C の性能を比較し、提案手法の A3C の有効性を評価する。本実験では、Tesla V100 を 3 枚使用し、4 時間学習させた際の性能を比較する。本稿では 47 種類の Atari ゲームの環境を用いて評価を行うが、スペースの関係から “BeamRider (BR)”, “Breakout (BO)”, “SpaceInvaders (SI)”, “MsPacman (MP)”, “SeaQuest (SQ)”, “DemonAttack (DA)” の 6 つのゲームを対象に述べる。その他のゲームの結果に関しては、添付資料を参照して頂きたい。

表 6.3 に、各手法が獲得したスコアを示す。比較対象は、従来の A3C と、LSTM の代わりに convLSTM を導入した A3C (+convLSTM), ABN を導入した提案手法の A3C (+ABN), 提案手法の Value branch から 1 チャンネルの畳み込み層を取り除いた A3C (-1Ch. Conv.) の 4 つである。はじめに、従来の A3C と +convLSTM を比較する。convLSTM の導入では、BeamRider と Breakout 以外のゲームは性能向上が見られなかった。次に、+ABN との性能を比較する。ABN を導入した場合、“MsPacman”

表 6.1: 従来法のネットワーク構造

Layer	Detail
1st Conv.	kernel : $32 \times 5 \times 5$ padding : 2 activation func. : ReLU Max pooling
2nd Conv.	kernel : $32 \times 5 \times 5$ padding : 1 activation func. : ReLU Max pooling
3rd Conv.	kernel : $64 \times 4 \times 4$ padding : 1 activation func. : ReLU Max pooling
4th Conv.	kernel : $64 \times 3 \times 3$ padding : 1 activation func. : ReLU Max pooling
1st LSTM	# of unit : 512
Output1 (Value)	# of unit : 1 activation func. : None
Output2 (Policy)	# of unit : number of action activation func. : Softmax

表 6.2: 提案手法のネットワーク構造

Layer	Detail
FE 1st Conv.	kernel : $32 \times 5 \times 5$ padding : 2 activation func. : ReLU Max pooling
FE 2nd Conv.	kernel : $32 \times 3 \times 3$ padding : 1 activation func. : ReLU Max pooling
FE 3rd Conv.	kernel : $64 \times 3 \times 3$ padding : 1 activation func. : ReLU
FE 4th Conv.LSTM	kernel : $64 \times 3 \times 3$
VB 1st Conv.	kernel : $64 \times 3 \times 3$ padding : 1 activation func. : ReLU
VB 2nd Conv.	kernel : $128 \times 1 \times 1$ padding : 1 activation func. : ReLU Global Average Pooling
VB Output (Value)	# of unit : 1 activation func. : None
PB 1st	kernel : $64 \times 3 \times 3$ padding : 1 activation func. : ReLU Max pooling
PB Output (Policy)	# of unit : number of action activation func. : Softmax

以外のスコアが従来の A3C より性能が低下した。これは、Value branch を構築する際に、1 チャンネルの畳み込み層を複数導入しているのが原因である。-1Ch. Conv. の性能を比較する。不要となる複数の 1 チャンネルの畳み込み層を取り除くことで、+ABN よりスコアを多く獲得した。また、“SeaQuest”と“DemonAttack”以外は従来の A3C より獲得できたスコアが多い。

図 6.4 に、提案手法の Attention map と Samuel らの Saliency map [4] の例を示す。Samuel らの Saliency map は、従来の A3C から Actor と Critic に対する Saliency map をそれぞれ出力する。図 6.4 の Actor に対する Saliency map は、エージェント以外の Enemy 等の動作物に対して反応しやすい。そのため、SpaceInvaders のような 1 つシーンに大量の Enemy が存在する場合、画面全体に Saliency map が強く反応する。Critic の Saliency map は、エージェントや破壊する直前の Enemy に対して反応しやすい。提案手法の Attention map は、ConvLSTM が抽出した特徴マップを用いて Attention map を出力するため、時系列を考慮した注視領域を獲得できる。図 6.4 における MsPacman の例では、

表 6.3: 評価時におけるゲームスコアの比較

A3C	+convLSTM	+ABN	-1Ch. Conv.	BR	BO	SI	MP	SQ	DA
✓				16,570.24	402.16	1,695.70	2,348.80	1,874.00	38858.90
✓	✓			18,065.04	432.21	1,356.72	2,183.30	1,841.60	35680.30
✓	✓	✓		11,442.88	365.93	573.35	2595.50	1,837.40	29884.90
✓	✓	✓	✓	18,083.42	428.87	2,075.70	2,945.00	1,860.60	16971.60

エージェントが左側の通路から右側の通路にワープする際の例を示している。Samuel らの手法では、Actor と Critic の両方の Saliency map がワープ先の地点に反応を示さなかった。一方、提案手法の Attention map ではワープ先の地点に対して Attention map が強く反応している。また、BeamRider や SeaQuest, DemonAttack では、Samuel らの Critic における Saliency map と同様に特定の敵に対して Attention map が強く反応している。

6.3.2 Attention map の可視化

提案手法が獲得した Attention map を可視化し、A3C の判断根拠を解析する。本稿では、6.3.1 章で述べた 6 つのゲームを対象に Attention map の解析を行う。

シューティングゲーム BeamRider と SeaQuest, SpaceInvaders はシューティングゲームの一つであり、エージェントの近くに出現した敵を撃ち落とすゲームである。3 つのシューティングゲームにおける Attention map を図 6.5 に示す。BeamRider の Attention map は、図 6.5(a) のようにエージェントと敵までの射線上に強く反応している。SeaQuest の Attention map は、図 6.5(b) のように画面の右側から出現する敵に強く反応している。また、画面の下側にある“OXYGEN”のステータスバーと左側の領域にも Attention map が強く反応した。画面の左側は敵が発生するため、Attention map が強く反応したと考えられる。OXYGEN のステータスバーは、エージェントが水中で活動できる時間を示しており、ステータスバーが無くなる前に画面上部に移動して OXYGEN を補給する必要がある。そのため、より高得点を獲得するために OXYGEN のステータスバーに対して Attention map が強く反応したと考えられる。SpaceInvaders の図 6.5 は、エージェントが攻撃を防壁で防ぎつつ、攻撃しているシーンである。Attention map を確認した際に、エージェント付近の防壁と前方の敵に対して Attention map が強く反応している。

ここで、獲得したスコアと出力した状態価値 $V(s_t; \theta)$ の推移を、図 6.5 のグラフで確認する。BeamRider 等のシューティングゲームは、敵の出現と消滅を繰り返し行う。そのため、出力される状態価値 $V(s_t; \theta)$ は周期的に上昇と減少を繰り返している。

Breakout と MsPacman Breakout は、エージェントであるラケットを操作してボールを跳ね返し、画面上部の壁を破壊してスコアを獲得するゲームである。MsPacman は、エージェントを操作して迷路内のドットを全て回収するゲームである。MsPacman の迷路には 4 匹のモンスターが配置されており、迷路の四隅に配置されたクッキーを回収することで捕食できる。また、連続してモンスターを

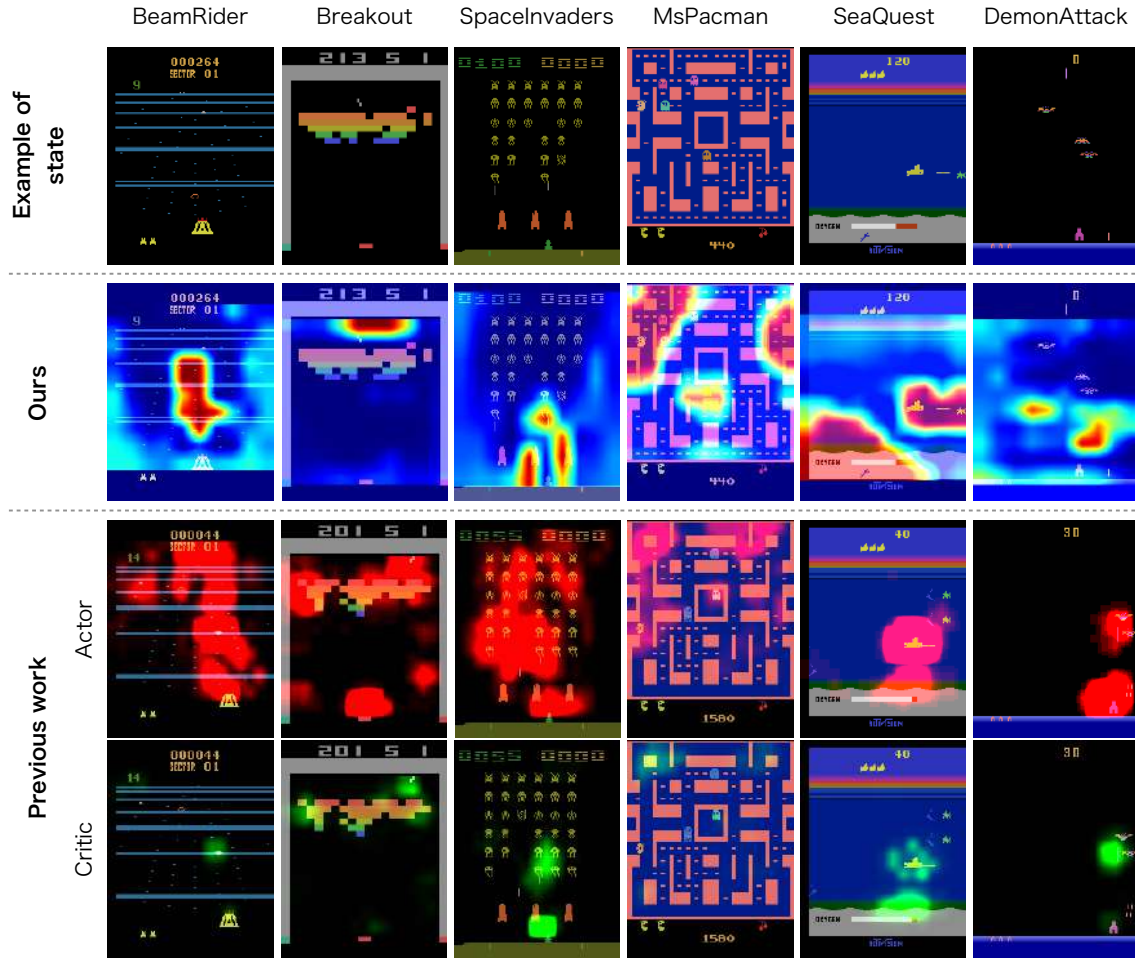


図 6.4: 提案手法の Attention map と Saliency map [4] の比較例

捕食することで、獲得できるスコアが2倍となる。この2つのゲームは BeamRider のようなシューティングゲームと異なり、敵やアイテムが初めから配置された状態でゲームを進める。そのため、行動を起こした際にアイテムや敵等の位置関係を保存したまま次の状態 s_{t+1} を更新する。

Breakout の Attention map の例を図 6.6(a) に示す。図の上部にあるグラフは、獲得したスコアと出力した状態価値 $V(s_t; \theta)$ の推移を示している。状態価値 $V(s_t; \theta)$ が最大になった Point1 は、ボールが壁の上部に初めて侵入したシーンである。Breakout の場合、壁の上部にボールが侵入することで、壁の上部内で繰り返しボールが跳ね返り、多くの壁を破壊してスコアを大量に獲得できる。状態価値 $V(s_t; \theta)$ は、状態 s_t における収益報酬 R の期待値を示している。ボールが侵入した Point1 において大量のスコア獲得を見込めたため、状態価値 $V(s_t; \theta)$ が大幅に上昇したと考えられる。状態価値 $V(s_t; \theta)$ が最大になった後、出力される状態価値 $V(s_t; \theta)$ は次のステージが開始されるまで減少し続けている。Point1 で獲得した Attention map は強く反応しなかったが、約 80 フレーム後の Point2 で壁の上部に強く反応した。そして、多くの壁が破壊された Point3 では、画面上部でボールが跳ね返る位置に Attention map が強く反応した。状態価値 $V(s_t; \theta)$ が最大となった Point1 では Attention map

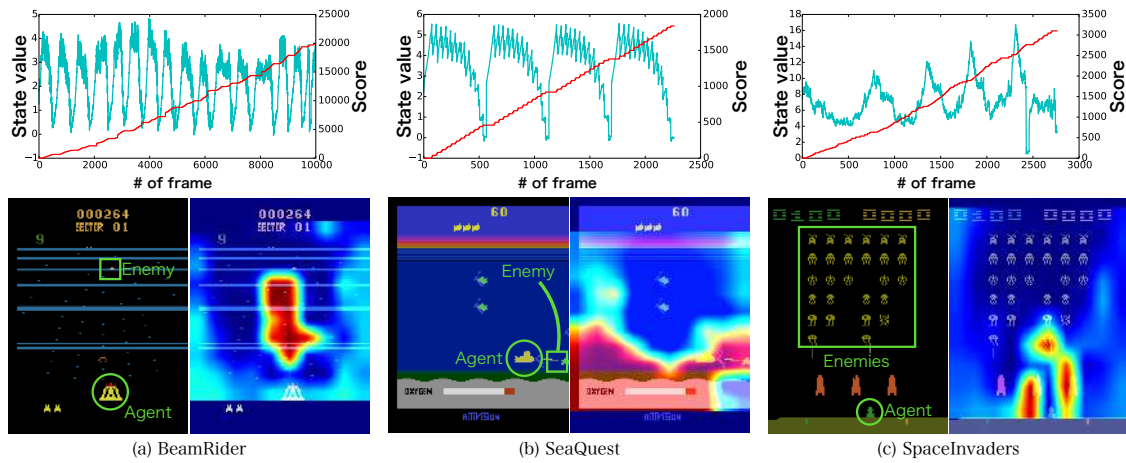


図 6.5: シューティングゲームにおける Attention map の例

が強く反応しなかったことから、収益報酬 R との相関が低いと言える。対して、ラケットでボールを打ち返す直前や、壁の上部で繰り返しボールが反射した際に Attention map が強く反応したため、即時報酬 r を獲得する際に注視した領域を Attention map が示していると考えられる。

MsPacman の Attention map の例を図 6.6(b) に示す。図 6.6(b) のグラフより、MsPacman における状態価値 $V(s_t; \theta)$ はゲームが開始してエージェントが動作可能になった際に最大となった。図 6.6(b) の Point1 と Point3 では、高得点なモンスターを倒す際の Attention map を示している。Point1 と Point3 の Attention map は、エージェント付近に対して強く反応している。また、他の例として MsPacman において左側から右側へワープする際に、画面の両サイドに対して Attention map が強く反応した。一方で、Point2 では Attention map が強く反応した領域が消失する現象が発生した。MsPacman においても、エージェントが位置する領域やワープする位置、モンスターを捕食する際に Attention map が強く反応しており、Breakout と同様に即時報酬 r を獲得する際に重要となる位置に Attention map が強く反応している。Point2 では、迷路内に存在するクッキーを全て回収したことで、高スコアを獲得できるモンスターを捕食する手段が無くなり、Attention map が反応しなくなったと考えられる。

DemonAttack DemonAttack は、BeamRider と同様にシューティングゲームに分類される Atari ゲームである。DemonAttack は、BeamRider と同様にエージェントを左右に移動させることで敵を破壊し、スコアを獲得する。ゲームの仕様は BeamRider と同様であるが、DemonAttack の敵の動作がより複雑になっている。また、表 6.3 より従来の A3C と比べて提案手法の獲得できたスコアが大幅に低下している。

図 6.7 に、DemonAttack におけるスコアと状態価値 $V(s_t; \theta)$ の推移を示す。図 6.7(a) は、従来の A3C におけるスコアと状態価値 $V(s_t; \theta)$ の推移を示している。従来の A3C は、Point1 において状態価値 $V(s_t; \theta)$ が上昇している。Point1 のシーンは、出現する敵が変化し、獲得できるスコアが上昇するシーンである (Fig. 6.7(c) の左側)。一方で、図 6.7(b) の提案手法の状態価値 $V(s_t; \theta)$ は、従来の A3C の状態価値 $V(s_t; \theta)$ のような値の上昇が見られなかった。これは、状態価値 $V(s_t; \theta)$ の推定精度が低下しているためと考えられる。状態価値 $V(s_t; \theta)$ の精度低下が原因により、従来の A3C より

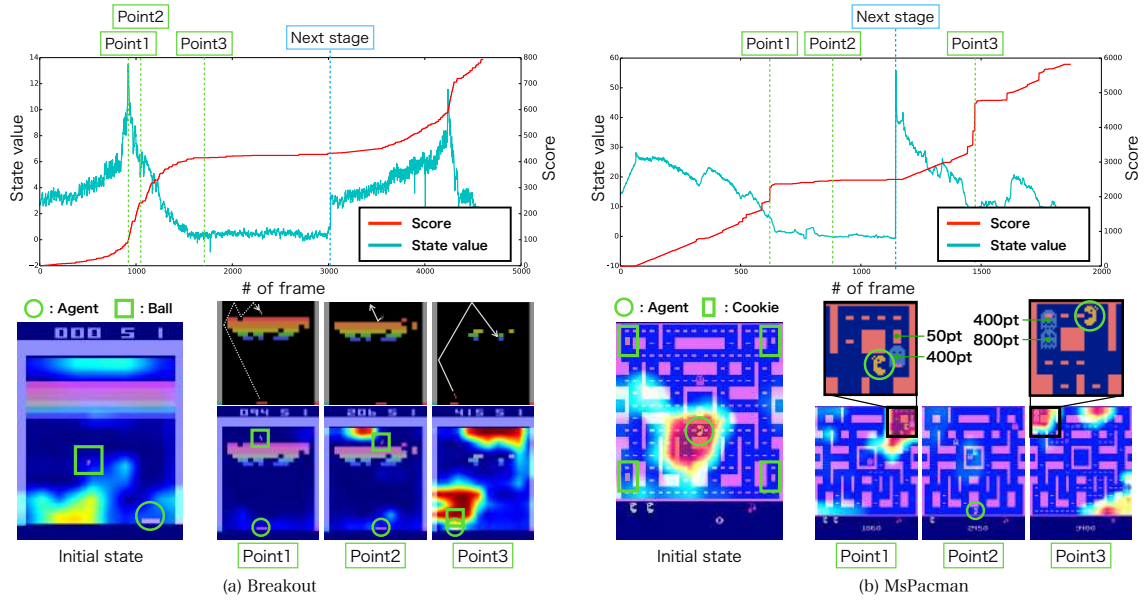


図 6.6: Breakout と MsPacman の Attention map の例

獲得できたスコアが大幅に低下したと考えられる．また，Attention map を確認した時，BeamRider のようにエージェントから敵までの射線に対して強く反応している．

従来の ABN は Perception branch の出力結果を最終的な認識結果としているため精度低下が起きない．一方で，提案手法は状態価値 $V(s_t; \theta)$ を畳み込み層をベースに構築した Value branch により出力している．この構造は，特定の認識タスクにおいて性能を低下させる可能性がある [13]．DemonAttack は，この構造が原因で状態価値 $V(s_t; \theta)$ の精度が低下したと考えられる．

6.4 まとめ

本稿では，ABN を Actor-Critic 法へ応用することで，深層強化学習における視覚的説明を行なった．提案手法は Value branch から状態価値 $V(s_t; \theta)$ を出力する過程で Attention map を出力し，Policy branch で方策を出力する．Value branch が出力する状態価値 $V(s_t; \theta)$ は状態 s_t における収益報酬 R の期待値を示しているのに対し，獲得した Attention map は即時報酬 r を獲得する際に重要となる位置を表現している．OpenAI gym における Atari ゲームの環境を用いて実験した時，Breakout や MsPacman 等のゲームにおいてスコアを獲得する直前の動作等に，Attention map が強く反応していることを確認した．また，特定の Atari ゲームに対して獲得できたスコアが向上した．

一方で，従来法より獲得できたスコアが大幅に低下した Atari ゲームが存在した．出力した状態価値 $V(s_t; \theta)$ を確認した時，状態価値 $V(s_t; \theta)$ の推定精度の低下が見られた．状態価値 $V(s_t; \theta)$ の精度低下は，獲得できるスコアの低下の原因となる．これは，Value branch を畳み込み層をベースに構築したことが原因と考えられる．今後は，提案手法において状態価値 $V(s_t; \theta)$ を安定して出力できるモデルの開発を行う．

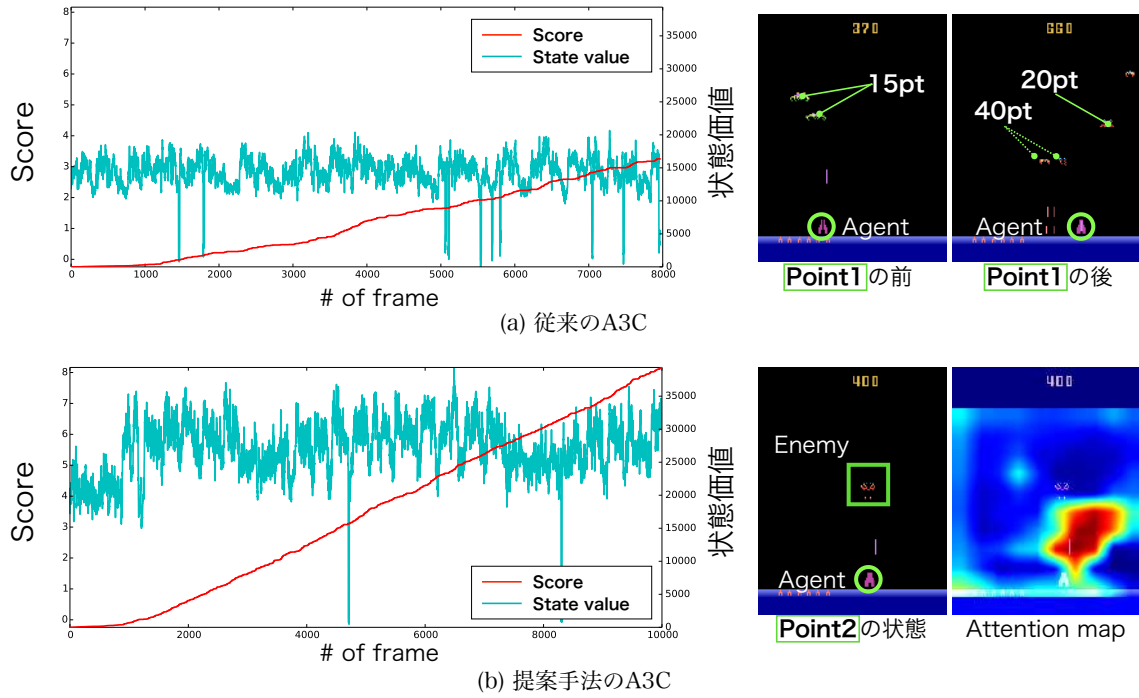


図 6.7: DemonAttack における従来の A3C と提案手法の比較

一方で、従来法より獲得できたスコアが大幅に低下した Atari ゲームが存在した．出力した状態価値 $V(\mathbf{s}_t; \theta)$ を確認した時，状態価値 $V(\mathbf{s}_t; \theta)$ の推定精度の低下が見られた．状態価値 $V(\mathbf{s}_t; \theta)$ の精度低下は，獲得できるスコアの低下の原因となる．これは，Value branch を畳み込み層をベースに構築したことが原因と考えられる．今後は，提案手法において状態価値 $V(\mathbf{s}_t; \theta)$ を安定して出力できるモデルの開発を行う．

第7章

結論と展望

本稿では、CNN の判断根拠を理解するために、視覚的説明モデルを Attention 機構へ応用した Attention Branch Network を提案した。以下に、本論文の結論と今後の展望について述べる。

7.1 結論

各章のまとめは以下の通りである。2.1 章では、画像認識分野で用いられる CNN の基礎的な構造や学習法を述べた後に、強化学習や歩行者検出法、視覚的説明等の関連研究についてまとめた。畳み込み層を主体に構築される CNN の構造は自由度が非常に高く、様々な識別対象に応じてネットワークを自由に構築できる。そのため、CNN は、画像分類をはじめとした物体 (歩行者) 検出や属性認識、エージェントの制御等の様々な分野に応用されている。画像分類モデルは、Residual Learning の提案により、より深いネットワークを構築することで画像分類問題の性能が大幅に向上した。Residual Learning の提案は、画像認識の分野で大きな貢献となっており、ベースラインとなる画像分類モデルの進展により物体検出等の性能も大幅に向上している。物体検出においても、Faster R-CNN 等のモデルが提案されたことで精度が大幅に向上し、リアルタイムに近い処理速度で検出が可能となった。Faster R-CNN が提案された後、SSD のようなより高速に物体検出可能な手法も提案されている。

3 章では、Random Dropout と Ensemble Inference Networks (EIN) による歩行者検出と標識認識に取り組んだ。Random Dropout と EIN は、Dropout をベースとした手法である。Dropout は、学習時に一定の割合でランダムにユニットを選択して応答値を 0 にする手法である。しかし、Dropout は学習処理において、ユニットの応答値を 0 にする割合を固定している。また、学習時のみユニットの応答値を 0 にする処理が行われ、推論時には行われない。提案する Random Dropout は、各更新回数で応答値を 0 にする割合をランダムに指定する。1 つ目の Random Dropout は、歩行者や標識の隠れや照明変化に対して頑健なネットワークを学習し、汎化性能を向上させる。Dropout を推論時に応用した EIN は、学習した 1 つのネットワークからランダムに選択したユニットの応答値を 0 にする全結合層を複数生成し、最終的な出力は特定の統合方法を用いることで識別する。構造の異なる複数の全結合層のネットワークを用いて検出、識別をすることで、誤検出と誤識別を削減する。Random Dropout と EIN の導入により、Caltech Pedestrian Dataset における歩行者検出では従来の CNN より Miss rate が約 8% 削減できた。標識認識においても、人の認識性能を超えた精度を実現できた。

4 章では、Multi-task Learning に関する問題について取り組んだ。4.2 章では、重み付き誤差関数を導入した Heterogeneous Learning による顔属性認識に取り組んだ。重み付き誤差関数によるアプロー

チでは、単一の CNN の学習誤差から Heterogeneous Learning の重み値を算出することで、各タスクにおける学習誤差のばらつきを抑制できた。提案手法では、特定の 1 タスクだけでなく、全てのタスクにおける精度を向上できた。特に、単一の CNN と比べて精度が著しく低下していた回帰タスクの精度が、大幅に向上できた。4.3 章では、Heterogeneous Learning の学習に対して稀少度を導入することで、少数サンプルのクラスの精度を向上させた。稀少度を導入した学習では、各学習サンプルに付与した稀少度から CNN に入力する学習サンプルを決定する。これにより、少量サンプルと過剰サンプルを平等に CNN へ入力できる。実験では、歩行者属性認識を対象とし、識別が困難であった“傘をさしている歩行者”の認識精度が向上した。また、他の少量サンプルに属すクラスの認識精度を向上できた。

5 章では、CNN の判断根拠を理解するために Attention Branch Network を提案した。Attention Branch Network は、視覚的説明モデルを Attention 機構へ応用した手法であり、視覚的説明モデルから獲得した Attention map を Attention 機構へ応用している。視覚的説明モデルを Attention 機構へ応用することで、従来の視覚的説明モデルの性能低下の問題を解決しつつ、Attention map による注視領域の可視化を同時に行うことができる。また、Attention Branch Network は歩行者検出や顔属性認識に応用が可能であり、それぞれの画像認識タスクにおいても高精度化を実現できた。

6 章では、5 章で提案した Attention Branch Network を深層強化学習へ応用した。強化学習における Attention Branch Network では、Attention branch で状態価値を出力し、Perception branch で方策を出力する。これにより、高報酬な領域に Attention map が強く反応し、Attention map を方策関数へ反映することができる。

7.2 展望

本稿では、深層学習の一つである Deep Convolutional Neural Network の判断根拠の獲得するために、視覚的説明手法の Attention map を Attention 機構へ応用することで性能向上と判断根拠の理解を同時に行った。また、強化学習への応用のために、提案した Attention Branch Network を強化学習法の一つである Actor-Critic 法へと応用した。これにより、画像認識分野だけでなく、エージェントの制御においても視覚的説明を可能にした。しかし、深層学習の判断根拠の理解における実用化には、まだ大きな課題がある。この課題は、判断根拠の言語化である。現状では、CNN が注視した領域を視覚的に強調させることで、ユーザーのような第三者に判断根拠の伝えている。しかし、これは正確な判断根拠を第三者へ伝えることができない。より正確な判断根拠を得るために、認識に対する判断根拠を言語によりユーザーへ伝える必要がある。

謝 辞

本研究は、著者が中部大学大学院工学研究科情報工学専攻博士後期課程在学中に、同大学工学部ロボット理工学科 藤吉弘亘教授の指導のもとに行ったものである。研究の遂行にあたり、常日頃ご指導を賜りました中部大学工学部ロボット理工学科 藤吉弘亘教授に深く感謝の意を表します。本論文をまとめるにあたり、有益なご討論、ご助言を賜りました中部大学工学部情報工学科 岩堀祐之教授、中部大学工学部情報工学科 山内康一郎教授、名古屋大学 大学院情報学研究科 知能システム学専攻 村瀬洋教授に謹んで感謝いたします。本研究において、貴重なご意見、ご指導を頂きました中部大学工学部情報工学科 山下隆義准教授、中部大学 平川翼研究員、中部大学工学部ロボット理工学科 山内悠嗣講師に心から厚く御礼申し上げます。最後に、本研究にご協力して頂いた藤吉研究室と山下研究室の皆様には感謝致します。

参考文献

- [1] W. Yuxin, and H. Kaiming, “Group normalization”, European Conference on Computer Vision, pp.3–19, 2018.
- [2] P. Dollár, Z. Tu, P. Perona, and S. Belongie, “Integral channel features”, British Machine Vision Conference, 2009.
- [3] R. Benenson, M. Omran, J. Hosang, and B. Schiele, “Ten years of pedestrian detection, what have we learned?”, European Conference on Computer Vision, 2014.
- [4] S. Greydanus, A. Koul, J. Dodge, and A. Fern, “Visualizing and understanding Atari agents”, International Conference on Machine Learning, vol.80, pp.1792–1801, 2018.
- [5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition”, Neural Computation, vol.1, no.4, pp.541–551, 1989.
- [6] K. Alex, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, Neural Information Processing Systems, pp.1097–1105, 2012.
- [7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, vol.60, no.2, pp.91–110, 2004.
- [8] N. Dalal, and B. Triggs, “Histograms of oriented gradients for human detection”, Computer Vision and Pattern Recognition, vol.1, pp.886–893, 2005.
- [9] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training very deep networks”, Neural Information Processing Systems 28, pp.2377–2385, 2015.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, Computer Vision and Pattern Recognition, pp.770–778, 2016.
- [11] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, “Multi-column deep neural network for traffic sign classification.”, Neural Networks, vol.32, pp.333–338, 2012.

- [12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting”, *Journal of Machine Learning Research*, vol.15, no.1, pp.1929–1958, 2014.
- [13] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization”, *Computer Vision and Pattern Recognition*, 2016.
- [14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks”, *Neural Information Processing Systems*, pp.91–99, 2015.
- [15] J. Cao, Y. Li, and Z. Zhang, “Partially shared multi-task convolutional neural network with local constraint for face attribute learning”, *Computer Vision and Pattern Recognition*, 2018.
- [16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network”, *Computer Vision and Pattern Recognition*, 2016.
- [17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn”, *International Conference on Computer Vision*, 2017.
- [18] D. H. Hubel, and T. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cats visual cortex”, *The Journal of physiology*, vol.160, pp.106–154, 1962.
- [19] K. Fukushima, and S. Miyake, “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position”, *Pattern Recognition*, vol.15, no.6, pp.455–469, 1982.
- [20] D. Rumelhart, G. Hintont, and R. Williams, “Learning representations by back-propagating errors”, *Nature*, pp.533-536, 1986.
- [21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. y. Fu, and A. C. Berg, “Ssd : Single shot multibox detector”, *European Conference on Computer Vision*, pp.1–15, 2016.
- [22] E. Rudd, M. Gunther, and T. Boulton, “Moon: A mixed objective optimization network for the recognition of facial attributes”, *European Conference on Computer Vision*, 2016.
- [23] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling”, *arXiv preprint arXiv:1505.07293*, 2015.
- [24] S. Evan, L. Jonathan, and D. Trevor, “Fully convolutional networks for semantic segmentation”, *Transactions on Pattern Analysis and Machine Intelligence*, vol.39, no.4, pp.640–651, 2017.
- [25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge”, *International Journal of Computer Vision*, vol.115, no.3, pp.211-252, 2015.

- [26] J. Sánchez, and F. Perronnin, “High-dimensional signature compression for large-scale image classification”, *Computer Vision and Pattern Recognition*, pp.1665-1672, 2011.
- [27] M. D. Zeiler, and R. Fergus, “Visualizing and understanding convolutional networks”, *European Conference on Computer Vision*, pp.818–833, 2014.
- [28] S. Karen, and Z. Andrew, “Very deep convolutional networks for large-scale image recognition”, *International Conference on Learning Representations*, 2015.
- [29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions”, *Computer Vision and Pattern Recognition*, pp.1–9, 2015.
- [30] I. Sergey, and S. Christian, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, *International Conference on Machine Learning*, pp.448-456, 2015.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”, *International Conference on Computer Vision*, pp.1026–1034, 2015.
- [32] S. Zagoruyko, and N. Komodakis, “Wide residual networks”, *British Machine Vision Conference*, 2016.
- [33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks”, *Conference on Computer Vision and Pattern Recognition*, 2017.
- [34] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks”, *Computer Vision and Pattern Recognition*, pp.5987-5995, 2017.
- [35] M. Lin, Q. Chen, and S. Yan, “Network in network”, *arXiv*, no.1312.4400, 2013.
- [36] H. Jie, S. Li, and S. Gang, “Squeeze-and-excitation networks”, *Computer Vision and Pattern Recognition*, 2017.
- [37] S. Ramprasaath, R., C. Michael, D. Abhishek, V. Ramakrishna, P. Devi, and B. Dhruv, “Grad-cam: Visual explanations from deep networks via gradient-based localization”, *International Conference on Computer Vision*, pp.618–626, 2017.
- [38] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network.”, *arXiv*, 2015.
- [39] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus).”, *arXiv*, 2015.

- [40] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, “Maxout networks”, International Conference on Machine Learning, 2013.
- [41] G. Xavier, and B. Yoshua, “Understanding the difficulty of training deep feedforward neural networks”, International Conference on Artificial Intelligence and Statistics, vol.9, pp.249–256, 2010.
- [42] J. B. Lei, K. Ryan, and H. Geoffrey E., “Layer normalization”, arXiv, vol.abs/1607.06450, 2016.
- [43] H. Nam, and H. E. Kim, “Batch-instance normalization for adaptively style-invariant neural networks”, arXiv preprint arXiv:1805.07925, 2018.
- [44] M. D. Zeiler, “Adadelta: An adaptive learning rate method”, 1212.5701, 2012.
- [45] T. Tieleman, and G. Hinton, “Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude”, COURSERA: Neural Networks for Machine Learning, 2012.
- [46] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization”, Journal of Machine Learning Research, vol.12, pp.2121–2159, 2011.
- [47] D. P. Kingma, and J. Ba, “Adam: A method for stochastic optimization”, 1412.6980, 2014.
- [48] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, Computer Vision and Pattern Recognition, pp.580–587, 2014.
- [49] R. Girshick, “Fast r-cnn”, International Conference on Computer Vision, 2015.
- [50] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition”, International Journal of Computer Vision, vol.104, no.2, pp.154–171, 2013.
- [51] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A Discriminatively Trained, Multi-scaled, Deformable Part Model”, Computer Vision and Pattern Recognition, pp.1–8, 2008.
- [52] W. Stefan, M. Nikodem, S. Konrad, and B. Schiele, “New features and insights for pedestrian detection”, Computer Vision and Pattern Recognition, 2010.
- [53] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, “Pedestrian detection at 100 frames per second”, Computer Vision and Pattern Recognition, pp.2903–2910, 2012.
- [54] W. Nam, P. Dollár, and J. H. Han, “Local decorrelation for improved pedestrian detection”, Neural Information Processing Systems, pp.1–9, 2014.
- [55] P. Dollar, R. Appel, S. Belongie, and P. Perona, “Fast feature pyramids for object detection”, Pattern Analysis and Machine Intelligence, vol.36, no.8, pp.1532–1545, 2014.

- [56] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning”, *Computer Vision and Pattern Recognition*, pp.3626–3633, 2013.
- [57] W. Ouyang, and X. Wang, “Joint deep learning for pedestrian detection”, *International Conference on Computer Vision*, pp.2056–2063, 2013.
- [58] P. Luo, Y. Tian, X. Wang, and X. Tang, “Switchable deep network for pedestrian detection”, *Computer Vision and Pattern Recognition*, pp.899–906, 2014.
- [59] J. Hosang, M. Omran, R. Benenson, and B. Schiele, “Taking a deeper look at pedestrians”, *Computer Vision and Pattern Recognition*, 2015.
- [60] Y. Tian, P. Luo, X. Wang, and X. Tang, “Pedestrian detection aided by deep learning semantic tasks”, *Computer Vision and Pattern Recognition*, 2015.
- [61] B. Yang, J. Yan, Z. Lei, and S. Z. Li, “Convolutional channel features: Tailoring cnn to diverse tasks”, *International Conference on Computer Vision*, 2015.
- [62] A. Angelova, A. Krizhevsky, M. View, M. View, V. Vanhoucke, A. Ogale, and D. Ferguson, “Real-Time Pedestrian Detection With Deep Network Cascades”, *British Machine Vision Conference*, pp.1–12, 2015.
- [63] Y. Tian, P. Luo, X. Wang, and X. Tang, “Deep learning strong parts for pedestrian detection”, *International Conference on Computer Vision*, pp.1904–1912, 2015.
- [64] Z. Cai, M. Saberian, and N. Vasconcelos, “Learning complexity-aware cascades for deep pedestrian detection”, *International Conference on Computer Vision*, pp.3361–3369, 2015.
- [65] J. Li, X. Liang, S. Shen, T. Xu, and S. Yan, “Scale-aware fast r-cnn for pedestrian detection”, *European Conference on Computer Vision*, 2015.
- [66] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection”, *European Conference on Computer Vision*, 2016.
- [67] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, “How far are we from solving pedestrian detection?”, *Computer Vision and Pattern Recognition*, 2016.
- [68] X. Du, M. El-Khamy, J. Lee, and L. S. Davis, “Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection”, *Winter Conference on Applications of Computer Vision*, 2016.
- [69] L. Zhang, L. Lin, X. Liang, and K. He, “Is Faster R-CNN Doing Well for Pedestrian Detection ?”, *European Conference on Computer Vision*, pp.1–15, 2016.

- [70] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets”, *Neural Information Processing Systems*, pp.2672–2680, 2014.
- [71] C. Papageorgiou, and T. Poggio, “A trainable system for object detection”, *International Journal of Computer Vision*, vol.38, no.1, pp.15–33, 2000.
- [72] M. Jones, and P. Viola, “Fast multi-view face detection”, *Mitsubishi Electric Research Lab Technical Report*, 2003.
- [73] S. Zhang, R. Benenson, and B. Schiele, “Filtered channel features for pedestrian detection”, *Computer Vision and Pattern Recognition*, pp.1751–1760, 2015.
- [74] P. Dollar, and C. Wojek, “Pedestrian detection: An evaluation of the state of the art”, *Pattern Analysis and Machine Intelligence*, pp.1–20, 2012.
- [75] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite”, *Computer Vision and Pattern Recognition*, 2012.
- [76] A. Ess, B. Leibe, K. Schindler, and L. van Gool, “Robust multiperson tracking from a mobile platform”, *Pattern Analysis and Machine Intelligence*, vol.31, no.10, pp.1831–1846, 2009.
- [77] M. Enzweiler, and D. Gavrilu, “Monocular pedestrian detection: Survey and experiments”, *Pattern Analysis and Machine Intelligence*, vol.31, no.12, pp.2179–2195, 2009.
- [78] Z. Shanshan, B. Rodrigo, and S. Bernt, “Citypersons: A diverse dataset for pedestrian detection”, *Computer Vision and Pattern Recognition*, 2017.
- [79] B. Hariharan, J. Malik, and D. Ramanan, “Discriminative decorrelation for clustering and classification”, *European Conference on Computer Vision*, pp.459-472, 2012.
- [80] L. Jianan, L. Xiaodan, W. Yunchao, X. Tingfa, F. Jiashi, and Y. Shuicheng, “Perceptual generative adversarial networks for small object detection”, *Computer Vision and Pattern Recognition*, pp.1951–1959, 2017.
- [81] J. Mao, Y. Jiang, T. Xiao, and Z. Cao, “Perceptual generative adversarial networks for small object detection”, *Computer Vision and Pattern Recognition*, 2017.
- [82] G. Brazil, X. Yin, and X. Liu, “Illuminating pedestrians via simultaneous detection and segmentation”, *International Conference on Computer Vision*, 2017.
- [83] W. Xinlong, X. Tete, J. Yuning, S. Shuai, S. Jian, and S. Chunhua, “Repulsion loss: Detecting pedestrians in a crowd”, *Computer Vision and Pattern Recognition*, 2018.

- [84] S. Eli, and I. Michal, “Matching local self-similarities across images and videos”, *Computer Vision and Pattern Recognition*, 2007.
- [85] Z. Zhanpeng, L. Ping, L. Chen, Change, and T. Xiaoou, “Facial landmark detection by deep multi-task learning”, *European Conference on Computer Vision*, 2014.
- [86] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation”, *Computer Vision and Pattern Recognition*, 2015.
- [87] Y. Sun, X. Wang, and X. Tang, “Deep convolutional network cascade for facial point detection”, *Computer Vision and Pattern Recognition*, pp.3476–3483, 2013.
- [88] C. Yan, C. Lang, T. Wang, X. Du, and C. Zhang, “Age estimation based on convolutional neural network”, *Pacific-Rim Conference on Advances in Multimedia Information Processing*, pp.211–220, 2014.
- [89] F. H. C. Tivive, and A. Bouzerdoum, “Age estimation based on convolutional neural network”, *Neural Networks*, pp.5336-5341, 2006.
- [90] E. Ricci, J. Varadarajan, R. Subramanian, S. R. Bulu, N. Ahuja, and O. Lanz, “Uncovering interactions and interactors: Joint estimation of head, body orientation and f-formations from surveillance videos”, *International Conference on Computer Vision*, pp.4660–4668, 2015.
- [91] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, “Regularization of neural networks using dropconnect”, vol.28, no.3, pp.1058-1066, 2013.
- [92] X. Frazao, and L. Alexandre, “Dropall: Generalization of two convolutional neural network regularization methods”, *International Conf. on Image Analysis and Recognition*, vol.LNCS 8814, pp.282 - 289, 2014.
- [93] J. Ba, and B. Frey, “Adaptive dropout for training deep neural networks”, *Neural Information Processing Systems*, pp.3084–3092, 2013.
- [94] H. Sebastian, S. Johannes, S. Jan, S. Marc, and I. Christian, “Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark”, *International Joint Conference on Neural Networks*, no.1288, 2013.
- [95] X. Wang, T. X. Han, and S. Yan, “An hog-lbp human detector with partial occlusion handling”, *International Conference on Computer Vision*, pp.32-39, 2009.
- [96] P. Viola, and M. J. Jones, “Robust real-time face detection”, *International Journal of Computer Vision*, vol.57, no.2, pp.137–154, 2004.

- [97] W. Ouyang, and X. Wang, “A discriminative deep model for pedestrian detection with occlusion handling.”, *Computer Vision and Pattern Recognition*, pp.3258-3265, 2012.
- [98] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis, “Human detection using partial least squares analysis”, *Proceedings of the International Conference on Computer Vision*, 2009.
- [99] P. Dollar, S. Belongie, and P. Perona, “The fastest pedestrian detector in the west”, *British Machine Vision Conference*, pp.68.1–68.11, 2010.
- [100] D. Piotr, A. Ron, and K. Wolf, “Crosstalk cascades for frame-rate pedestrian detection”, *European Conference on Computer Vision*, 2012.
- [101] J. Marín, D. Vázquez, A. M. López, J. Amores, and B. Leibe, “Random forests of local experts for pedestrian detection”, *International Conference on Computer Vision*, pp.2592-2599, 2013.
- [102] P. Dennis, R. Deva, and F. Charless, “Multi resolution models for object detection”, *European Conference on Computer Vision*, 2010.
- [103] B. Rodrigo, M. Markus, T. Tinne, and V. Luc, “Seeking the strongest rigid detector”, *Computer Vision and Pattern Recognition*, 2013.
- [104] C. Guang, D. Yuanyuan, X. Jing, and X. H. Tony, “Detection evolution with multi-order contextual co-occurrence”, *Computer Vision and Pattern Recognition*, pp.1798–1805, 2013.
- [105] P. Sabzmeydani, and G. Mori, “Detecting pedestrians by learning shapelet features”, *Computer Vision and Pattern Recognition*, 2007.
- [106] C. Wojek, and B. Schiele, “A performance evaluation of single and multi-feature people detection”, *Proceedings of the 30th DAGM Symposium on Pattern Recognition*, pp.82–91, 2008.
- [107] W. Nam, B. Han, and J. H. Han, “Improving object localization using macrofeature layout selection”, *International Conference on Computer Vision Workshops*, pp.1801-1808, 2011.
- [108] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition”, *Neural Networks*, no.0, 2012.
- [109] P. Sermanet, and Y. Lecun, “Traffic sign recognition with multi-scale convolutional networks”, *International Joint Conference on Neural Network*, 2011.
- [110] S. Li, Z. Q. Liu, and A. B. Chan, “Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network”, *International Journal of Computer Vision*, vol.113, no.1, pp.19–36, 2015.

- [111] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics”, *Computer Vision and Pattern Recognition*, 2018.
- [112] C. Huang, Y. Li, C. C. Loy, and X. Tang, “Learning deep representation for imbalanced classification”, *Computer Vision and Pattern Recognition*, 2016.
- [113] J. Zhu, S. Liao, D. Yi, Z. Lei, and S. Z. Li, “Multi-label cnn based pedestrian attribute learning for soft biometrics”, *International Conference on Biometrics*, pp.535-540, 2015.
- [114] L. Dangwei, C. Xiaotang, and H. Kaiqi, “Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios”, *Asian Conference on Pattern Recognition*, pp.111-115, 2015.
- [115] E. Bekele, C. Narber, and W. Lawson, “Multi-attribute residual network (maresnet) for soft-biometrics recognition in surveillance scenarios”, *Automatic Face and Gesture Recognition*, vol.00, pp.386-393, 2017.
- [116] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models”, *Pattern Analysis and Machine Intelligence*, pp.484–498, 1998.
- [117] M. Dantone, J. Gall, G. Fanelli, and L. van Gool, “Real-time facial feature detection using conditional regression forests”, *Computer Vision and Pattern Recognition*, pp.2578-2585, 2012.
- [118] E. Eiding, R. Enbar, and T. Hassner, “Age and gender estimation of unfiltered faces”, *Information Forensics and Security*, vol.9, no.12, pp.2170–2179, 2014.
- [119] L. Hui-Cheng, and L. Bao-Liang, “Multi-view gender classification using local binary patterns and support vector machines”, *Third International Symposium on Neural Networks*, 2006.
- [120] Y. Xiaolin, S. Kim, and X. Eric P., “Heterogeneous multitask learning with joint sparsity constraints”, *Neural Information Processing Systems*, pp.2151–2159, 2009.
- [121] H. Emily M., and C. Rama, “Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification”, *Association for the Advancement of Artificial Intelligence*, 2017.
- [122] S. Daniel, T. Nikhil, K. Been, B. V. Fernanda, and W. Martin, “Smoothgrad: removing noise by adding noise”, , 2017.
- [123] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks”, *arXiv preprint arXiv:1710.11063*, 2017.

- [124] T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation”, *Empirical Methods in Natural Language Processing*, pp.1412–1421, 2015.
- [125] X. Kelvin, B. Jimmy, K. Ryan, C. Kyunghyun, C. Aaron, S. Ruslan, Z. Rich, and B. Yoshua, “Show, attend and tell: Neural image caption generation with visual attention”, *International Conference on Machine Learning*, pp.2048–2057, 2015.
- [126] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification”, *Computer Vision and Pattern Recognition*, 2017.
- [127] S. Hochreiter, and J. Schmidhuber, “Long short-term memory”, *Neural Comput*, vol.9, no.8, pp.1735–1780, 1997.
- [128] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, “Stacked attention networks for image question answering”, *Computer Vision and Pattern Recognition*, pp.21-29, 2016.
- [129] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention”, *Computer Vision and Pattern Recognition*, pp.4651-4659, 2016.
- [130] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate”, *International Conference on Learning Representations*, 2016.
- [131] W. Xiaolong, G. Ross, G. Abhinav, and H. Kaiming, “Non-local neural networks”, *Computer Vision and Pattern Recognition*, 2018.
- [132] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu, “Recurrent models of visual attention”, *Neural Information Processing Systems*, pp.2204–2212, 2014.
- [133] L. Drew, S. Dan, E. Sven, and S. Thomas, “Global-and-local attention networks for visual recognition”, *arXiv*, vol.abs/1805.08819, 2018.
- [134] 村瀬卓也, 平川翼, 山下隆義, 藤吉弘亘, “自己状態を付与した cnn による自動運転制御の高精度化”, *パターン認識・メディア理解研究会*, 2017.
- [135] B. Mariusz, T. Davide Del, D. Daniel, F. Bernhard, F. Beat, G. Prasoon, J. Lawrence D., M. Mathew, M. Urs, Z. Jiakai, Z. Xin, Z. Jake, and Z. Karol, “End to end learning for self-driving cars”, *arXiv*, 2016.
- [136] R. Caruana, “Multitask learning: A knowledge-based source of inductive bias”, *International Conference on Machine Learning*, pp.41–48, 1993.
- [137] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning”, *Neural Information Processing Systems*, 2011.

- [138] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database”, *Computer Vision and Pattern Recognition*, 2009.
- [139] L. Yang, P. Luo, C. C. Loy, and X. Tang, “A large-scale car dataset for fine-grained categorization and verification”, *Computer Vision and Pattern Recognition*, pp.3973-3981, 2015.
- [140] K. Neeraj, B. Peter N., and N. Shree K., “Facetracer: A search engine for large collections of images with faces”, *European Conference on Computer Vision*, pp.340–353, 2008.
- [141] L. Ziwei, L. Ping, W. Xiaogang, and T. Xiaoou, “Deep learning face attributes in the wild”, *International Conference on Computer Vision*, 2015.
- [142] N. Zhang, M. Paluri, M. Rantazo, T. Darrell, and L. Bourdev, “Panda: Pose aligned networks for deep attribute modeling”, *Computer Vision and Pattern Recognition*, pp.1637–1644, 2014.
- [143] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, “Mastering the game of go without human knowledge”, *Nature*, vol.550, pp.354–, 2017.
- [144] F. C. Yu, E. Michael, L. Miao, and P. H. Jonathan, “Socially aware motion planning with deep reinforcement learning”, *Intelligent Robots and Systems*, pp.1343-1350, 2017.
- [145] L. Sergey, P. Peter, K. Alex, I. Julian, and Q. Deirdre, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection”, *The International Journal of Robotics Research*, vol.37, no.4-5, pp.421–436, 2018.
- [146] C. J. C. H. Watkins, and P. Dayan, “Q-learning”, *Machine Learning*, vol.8, no.3, pp.279–292, 1992.
- [147] K. Vijay, and T. John, “Actor-critic algorithms”, *SIAM Journal on Control and Optimization*, pp.1008–1014, 2000.
- [148] G. Tesauro, “Temporal difference learning and td-gammon”, *Association for Computing Machinery*, vol.38, no.3, pp.58–68, 1995.
- [149] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning”, *Nature*, vol.518, pp.529–533, 2015.
- [150] H. v. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning”, *Association for the Advancement of Artificial Intelligence*, pp.2094–2100, 2016.

- [151] S. John, L. Sergey, A. Pieter, J. Michael, and M. Philipp, “Trust region policy optimization”, International Conference on Machine Learning, vol.37, pp.1889–1897, 2015.
- [152] S. John, W. Filip, D. Prafulla, R. Alec, and K. Oleg, “Proximal policy optimization algorithms”, arXiv:1611.05397, 2017.
- [153] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation”, Neural Information Processing Systems, pp.1057–1063, 1999.
- [154] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning”, International Conference on Machine Learning, pp.1928–1937, 2016.
- [155] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, “Reinforcement learning with unsupervised auxiliary tasks”, arXiv:1611.05397, 2016.
- [156] S. Ivan, S. Alexey, P. Mikhail, F. Aleksandr, and I. Anastasiia, “Deep attention recurrent q-network”, Neural Information Processing Systems Workshop, 2015.
- [157] P. Fabio, L. Vitaly, and P. Kormushev, “Q-map: a convolutional approach for goal-oriented reinforcement learning”, arXiv, no.1810.02927, 2018.
- [158] X. SHI, Z. Chen, H. Wang, D. Y. Yeung, W. k. Wong, and W. c. Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting”, Neural Information Processing Systems, pp.802–810, 2015.
- [159] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym”, , 2016.

研究業績一覧

学術論文

- [1] H Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, H. Murase, “Training of CNN with Heterogeneous Learning for Multiple Pedestrian Attributes Recognition Using Rarity Rate”, IEICE TRANSACTIONS on Information and Systems, Vol.E101-D, No.5, pp.1222-1231, 2018.
- [2] 奥野薫子, 山下隆義, 福井宏, 山内悠嗣, 藤吉弘亘, 乗富修蔵, 新浩治, “Heterogeneous Learning を導入した Deep Convolutional Neural Network による運転手の状態のモニタリング”, 精密工学会誌, vol. 83, No. 12, pp. 1101–1108, 2017.
- [3] 福井 宏, 山下 隆義, 山内 悠嗣, 藤吉 弘亘, 村瀬 洋, “Random Dropout と Ensemble Inference Networks による歩行者検出と標識認識”, 情報処理学会 論文誌, Vol. 57, No. 3, pp. 910-921, 2016.
- [4] T. Yamashita, T. Nakamura, H. Fukui, Y. Yamauchi, and H. Fujiyoshi, “Cost-Alleviative Learning for Deep Convolutional Neural Network-based Facial Part Labeling”, Information Processing Society of Japan Transactions on Computer Vision and Application, 2015.

国際会議発表論文 (査読あり)

- [1] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita and Hironobu Fujiyoshi, “Attention Branch Network: Learning of Attention Mechanism for Visual Explanation ”, Computer Vision and Pattern Recognition, 2019.
- [2] Hiroki Adachi, Hiroshi Fukui, Takayoshi Yamashita and Hironobu Fujiyoshi, “Facial Image Generation by Weighted Conditional Generative Adversarial Networks ”, Computer Vision, Imaging and Computer Graphics Theory and Applications, 2019.
- [3] K. Okuno, T. Yamashita, H. Fukui, S. Noridomi, K. Arata, Y. Yamauchi and H. Fujiyoshi, “Body Posture and Face Orientation Estimation by Convolutional Network with Heterogeneous Learning ”, International Workshop on Advanced Image Technology, 2018.

- [4] K. Enomoto, K. Sakurada, W. Wang, H. Fukui, M. Matsuoka, R. Nakamura, N. Kawaguchi, “Filmy Cloud Removal on Satellite Imagery with Multispectral Conditional Generative Adversarial Nets”, EARTHVISION Workshop on Computer Vision and Pattern Recognition, 2017.
- [5] H. Fukui, T. Yamashita, Y. Kato, R. Matsui, Y. Yamauchi, H. Fujiyoshi, “Facial Image Analysis by CNN with Heterogeneous Learning”, International Workshop on Advanced Image Technology, 2017.
- [6] T. Yamashita, H. Fukui, Y. Yamauchi, H. Fujiyoshi, “Pedestrian and Part Detection using Regression based Multiple Task Deep Convolutional Neural Network”, International Conference on Pattern Recognition, 2016.
- [7] H. Fukui, T. Yamashita, R. Matsui, Y. Kato, T. Ogata, Y. Yamauchi, H. Fujiyoshi, “Multiple Facial Attributes Estimation based on Weighted Heterogeneous Learning”, Workshop on Facial Informatics on Asian Conference on Computer Vision, 2016.
- [8] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, and H. Murase, “Robust Pedestrian Attribute Recognition for an Unbalanced Dataset using Mini-batch Training with Rarity Rate”, IEEE Intelligent Vehicle Symposium, 2016.
- [9] T. Yamashita, Y. Kato, H. Fukui, Y. Yamauchi, H. Fujiyoshi, “Heterogeneous Learning for Multi-task Facial Analysis Using Single Deep Convolutional Neural Network”, GPU Technology Conference, 2016.
- [10] T. Usami, H. Fukui, Y. Yamauchi, T. Yamashita, and H. Fujiyoshi, “DPM Score Regression for Detecting Occluded Humans from Depth Images”, Korea-Japan Joint Workshop on Frontiers of Computer Vision, 2016.
- [11] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, and H. Murase, “Pedestrian Detection Based on Deep Convolutional Neural Network with Ensemble Inference Networks”, IEEE Intelligent Vehicle Symposium, 2015.

学会口頭発表 (査読あり)

- [1] 阿知破千浩, 福井宏, 山下降義, 藤吉弘亘, “フローヒストグラムを用いた Recurrent Neural Network による自己移動量推定”, ビジョン技術の実利用ワークショップ, 2018.
- [2] 福井宏, 平川翼, 山下降義, 藤吉弘亘, “Attention 機構を導入した A3C の提案”, 日本ロボット学会学術講演会, 2018.
- [3] 森啓介, 福井宏, 丸山祐矢, 平川翼, 山下降義, 藤吉弘亘, “Attention Branch Network による一貫学習・強化学習におけるアテンションの獲得”, 日本ロボット学会学術講演会, 2018.

- [4] 福井宏, 平川翼, 山下隆義, 藤吉弘亘, “Global Average Pooling の特性を用いた Attention Branch Network”, GPU Technology Conference Japan, 2018.
- [5] 福井宏, 西山乗, 山下隆義, 藤吉弘亘, “Global Average Pooling の特性を用いた Attention Branch Network”, 画像の認識・理解のシンポジウム, 2018.
- [6] 大角耕介, 福井宏, 山下隆義, 藤吉弘亘, “Instance Weighting を用いた Gradient Reversal Layer によるドメイン適応の高精度化”, 画像の認識・理解のシンポジウム, 2018.
- [7] 足立浩規, 福井宏, 山下隆義, 藤吉弘亘, “重みを導入した Conditional Generative Adversarial Network による顔画像生成の高品質化”, 画像の認識・理解のシンポジウム, 2018.
- [8] 荒木諒介, 中瀬架, 福井宏, 山下隆義, 藤吉弘亘, “マルチタスク学習を導入した Deconvolutional Single Shot Detector による物体検出とセグメンテーションの高精度化”, 画像の認識・理解のシンポジウム, 2018.
- [9] 榎本憲二, 櫻田健, 王維民, 福井宏, 松岡昌志, 中村良介, 河口信夫, “マルチスペクトル cGANs による衛星画像の薄雲除去”, 画像の認識・理解のシンポジウム, 2017.
- [10] 福井 宏, 山下 隆義, 山内 悠嗣, 藤吉 弘亘, “[サーベイ論文] Deep Learning を用いた歩行者検出の研究動向”, パターン認識・メディア理解研究会, 2016.
- [11] 奥野 薫子, 山下 隆義, 福井 宏, 山内 悠嗣, 藤吉 弘亘, 乗富 修蔵, 新 浩治, “Heterogeneous Learning を導入した Deep Convolutional Neural Network による運転手の骨格検出と顔向き推定”, ビジョン技術の実利用ワークショップ, 2016.
- [12] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, and H. Murase, “Training of Deep Convolutional Neural Network for Pedestrian Attribute Recognition using Rarity rate”, GPU Technology Conference, 2016.
- [13] H. Fukui, Y. Kato, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, “Weighted Heterogeneous Learning for Deep Convolutional Neural Network Based Facial Image Analysis”, 画像の認識・理解のシンポジウム, 2016.
- [14] 福井 宏, 加藤 優, 神谷 龍司, 山下 隆義, 山内 悠嗣, 藤吉 弘亘, “プライバシーに配慮した Deep Convolutional Neural Network によるクラウド型顔照合システム”, 画像センシングシンポジウム, 2016.
- [15] 神谷 龍司, 川口 俊樹, 福井 宏, 石井 育規, 小塚 和紀, 羽川 令子, 築澤 宗太郎, 山下 隆義, 山内 悠嗣, 藤吉 弘亘, “プライバシーに配慮した Deep Convolutional Neural Network によるクラウド型顔照合システム”, 画像センシングシンポジウム, 2016.

- [16] 加藤 優, 福井 宏, 山下 隆義, 山内 悠嗣, 藤吉 弘亘, “Deep Convolutional Neural Network によるクラウド型顔照合システムの分散処理における負荷検証”, バイオメトリクスと認識・認証シンポジウム, 2015.
- [17] 加藤 優, 福井 宏, 山下 隆義, 山内 悠嗣, 藤吉 弘亘, “Heterogeneous Learning と重み付き誤差関数の導入による顔画像解析”, 画像の認識・理解シンポジウム, 2015.
- [18] 山下 隆義, 福井 宏, 村瀬 将之, 山内 悠嗣, 藤吉 弘亘, “回帰型 Deep Convolutional Neural Network による人検出と部位の位置推定”, 画像センシングシンポジウム, 2015.
- [19] 山下 隆義, 木村 真稔, 福井 宏, 山内 悠嗣, 藤吉 弘亘, “Deep Convolutional Neural Network による顔器官点検出における最適なミニバッチ作成方法”, 画像センシングシンポジウム, 2015.
- [20] 木村 真稔, 福井 宏, 山下 隆義, 山内 悠嗣, 藤吉 弘亘, “Convolutional Neural Network による顔の向きに頑健な顔器官点検出”, パターン認識・メディア理解研究会, 2015.
- [21] 福井 宏, 山下 隆義, 綿末 太郎, 山内 悠嗣, 藤吉 弘亘, 村瀬 洋, “Drop and Median Inference による歩行者検出の高精度化”, ビジョン技術の実利用ワークショップ, 2014.

学術表彰

- [1] 2018 年 GPU Technology Conference Japan Poster Award.
題目: Global Average Pooling の特性を用いた Attention Branch Network
- [2] 2018 年 MIRU 学生奨励賞.
題目: Global Average Pooling の特性を用いた Attention Branch Network
- [3] 2017 年 International Workshop on Advanced Image Technology Best Paper Award.
題目: Facial Image Analysis by CNN with Heterogeneous Learning
- [4] 2016 年 情報処理学会 論文誌 / JIP 特選論文 受賞.
題目: Random Dropout と Ensemble Inference Networks による歩行者検出と標識認識
- [5] 2016 年 画像センシングシンポジウム 最優秀学術論文賞.
題目: 回帰型 Deep Convolutional Neural Network による人検出と部位の位置推定
- [6] 2016 年 画像センシングシンポジウム 優秀学術論文賞.
題目: Convolutional Neural Network による顔の向きに頑健な顔器官点検出
- [7] 2015 年 画像センシングシンポジウム オーディエンス賞.
題目: 回帰型 Deep Convolutional Neural Network による人検出と部位の位置推定