令和２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

CNNによる高速かつ省メモリなドライバの状態推定に関する研究

西行　健太

論　文　要　旨

交通事故による死傷者数は減少傾向にあるものの，令和元年度の死傷者数は 3,215 人であり，現在も大きな社会問題となっている．令和元年度の原付以上運転者の法令違反別交通事故件数の内，安全運転義務違反に該当する事故の約 75%は漫然運転，脇見運転，安全不確認が占める．これらの要因による交通事故はドライバの脇見や居眠りなど注意を周辺環境から逸らしたことにより発生している．これらはドライバの姿勢推定や動作認識，眠気推定などのドライバモニタリング技術の実現により防ぐことが期待されている．

本研究では，はじめにドライバの姿勢推定について取り組む．Deep Convolutional Neural Network(CNN) を用いた高精度な人物姿勢推定が提案されているが，CNN は演算量などの消費リソースが多く，自動車内の組込機器にそのまま搭載することが難しい．本研究では Shuffle Net V2(SNV2) と Integral Regression(IR) を用いた高速かつ省メモリなドライバ姿勢推定を提案する．また，自動車内では被写体とカメラ間の距離が近く，ドライバの関節点が画角内に映らないことが多い．そのため，本研究ではドライバの関節点座標と同時にドライバの関節点が画角内に映っているかどうか（関節点有無）も推定する．SNV2 と IR を組み合わせることで，組込機器に搭載可能な演算量でも高精度にドライバの姿勢を推定できる．

次に，ドライバの動作認識について取り組む．動作認識も消費リソースの多さが組込機器への搭載の課題となる．本研究ではドライバの関節点座標，関節点有無，関節点状態の 3 つの姿勢情報と動作のマルチタスク学習を用いた高速かつ高精度なドライバ動作認識を提案する．既存のドライバ動作認識では意識を失うような深刻な状態や自動運転車での動作に対応していない．本研究では軽度な状態から深刻な状態及び自動運転車での動作まで幅広くカバーする動作認識を提案する．本研究のマルチタスク学習を用いることで，組込機器に搭載可能な演算量で高精度にドライバ動作を推定できる．

次に，ドライバの眠気推定について取り組む．ドライバ眠気推定の多くはドライバの強い眠気を検知する 2 値の眠気推定である．そのため，既存手法の特徴量やネットワークモデルは強い眠気を捉えるためのものであり，弱い眠気を含むマルチレベルの眠気推定には適さない．本研究では Average Eye Closure Time(AECT) と Soft PERCLOS の 2 つの時間特徴量と，複数の時間解像度の特徴を抽出可能な Parallel Linked Time-domain CNN を用いたマルチレベル眠気推定を提案する．本研究で提案する特徴量とネットワークモデルを用いることで，ドライバの弱い眠気も高精度に推定できる．

最後に，ネットワークモデルのコンパクト化について取り組む．ネットワークモデルの消費リソースを減らす手法として，大きな教師モデルの情報を学習に活用しコンパクトな生徒モデルを生成する Knowledge Distillation が提案されている．Knowledge Distillation の既存手法は適用できるモデルの制限や，教師モデルの中間層の情報を十分に学習に活かせないなどの課題がある．本研究では適用するモデルに制限がなく，教師モデルの中間層の情報を活かせる Sequential Layer-wise Knowledge Distillation(SLKD) を提案する．SLKD を用いることで精度の低下を抑制しつつ，モデルサイズを大幅に減らすことができる．

目 次

図 目 次

表 目 次

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

ドライバの人為的ミスによる自動車事故を防ぐため，ドライバモニタリングシステム（DMS）の導入が必要とされている．DMS に用いられるセンサには，脈波や心拍などの生体センサ，ブレーキやタイヤの動きなどの車体センサ，カメラを用いた画像センサがある．中でもドライバをカメラで撮影する画像センサはドライバの身体的な負担が少なく，車種や運転技能などの環境に影響を受けにくい．そのため，画像センサによる DMS は様々な環境で使用できる．

カメラで撮影した画像から人物の動きや状態を理解する画像認識技術では，深層学習の発展により，その認識精度が大幅に向上している．画像認識分野の深層学習として，Deep Convolutional Neural Network(CNN) が精度の高さから注目されている．しかし，CNN は演算量やメモリ消費量などの消費リソースが多く，自動車の組込機器にそのまま搭載することが難しい．そのため，画像センサを用いた DMS に CNN を導入するためには，消費リソースを減らす工夫が必要である．

画像センサを用いた DMS では，ドライバの姿勢推定，動作認識，眠気推定の 3 つの方法でドライバの状態を把握することで事故を抑制することが期待されている．

1 つ目のドライバ姿勢推定では，ドライバが正常な姿勢を保っているかを確認することで事故を防ぐのに役立つ．人物の姿勢推定は画像認識分野で活発に研究されており，OpenPose[1] など CNN を用いた高精度な人物姿勢推定が提案されている．CNN を用いた人物姿勢推定は高精度であるが，消費リソースが多く，組込機器にそのまま搭載することは難しい．また，人物姿勢推定で使用されるデータセットは可視光カメラで撮影されているが，DMS では夜間でもドライバを撮影する必要があるため，近赤外カメラを用いることが多い．更に人物姿勢推定で使用されるデータセットは被写体の全身が映ることが多いが，ドライバ姿勢推定では被写体とカメラの距離が近いため，ドライバの関節点がカメラに映らないことが多い．

2 つ目のドライバ動作認識では，ドライバがスマートフォンの操作や飲食など危険な動作をしていないかどうかを認識することで事故を防ぐ．人物動作認識も画像認識分野で活発に研究されており，CNN を用いた高精度な手法が提案されている．しかし，人物姿勢推定と同様に，CNN を用いた人物動作認識も消費リソースの多さや，近赤外線カメラに対応していない，などの課題がある．CNN を用いた人物動作認識では姿勢と動作を同時に学習するマルチタスク学習により動作認識精度を向上させることができる [2]．しかし，ドライバ動作認識では，一般的なカメラの画角と比べて，カメラに映る関節点が限定されるため，関節点座標のみを用いた既存のマルチタスク学習では効果が限定される．既存のドライバ動作認識データセット（SEU dataset[3]）では，電話，食事，ブレーキ，運転，スマートフォン操作，タバコなどの軽度な状態のみが含まれており，居眠りや発作などの意識を失うような深刻な状態や自動運転車での動作は含まれていない．

3 つ目のドライバ眠気推定では，ドライバが眠気を帯びた漫然運転を行っていないかを把握することで事故を防ぐ．ドライバ眠気推定の多くは，ドライバの強い眠気を検知する 2 値の眠気推定である．2 値の眠気推定では，ドライバの強い眠気を検知できたとしても，検知から事故が起こるまでの時間が短くなる．そのため，システムはドライバを早急に起こす必要があり，大音量の警報などドライバにとって不快な覚醒手段しか取ることができない．一方，弱い眠気も含んだマルチレベルの眠

気推定は，システムが取りうる選択肢を増やし，ドライバにとって快適な覚醒を可能にする．しかし，既存のドライバ眠気推定の多くが 2 値の眠気推定であるため，既存の特徴量やネットワークモデルは強い眠気を捉えるために設計されている．また，ドライバ眠気推定の評価に使用されるデータセットは，ドライビングシミュレータを用いて撮影されたものであり，実車環境の背景や照明，自動車の振動などの影響を評価できていない．

1.2 研究目的

本研究では，以下の 4 つの項目について取り組む．

1. 高速かつ省メモリなドライバ姿勢推定
2. 高速かつ省メモリなドライバ動作認識
3. 弱い眠気レベルを含んだマルチレベルのドライバ眠気推定
4. ネットワークモデルのコンパクト化

1 つ目は，自動車内の組込機器への搭載を想定し，高速かつ省メモリなドライバ姿勢推定を提案する．2 つ目は，同じく自動車内の組み込み機器への搭載を想定した高速かつ省メモリなドライバ動作認識を提案する．3 つ目は，眠気を早期に検知するため，弱い眠気レベルを含んだマルチレベルのドライバ眠気推定を提案する．4 つ目は，組込機器への搭載に役立つネットワークモデルのコンパクト化を提案する．以下に，4 項目における本研究の目的について述べる．

ドライバの姿勢推定　画像認識分野で人物姿勢推定は活発に研究されており，OpenPose[1] など CNN を用いた高精度な人物姿勢推定が提案されている．しかし，それらの人物姿勢推定とは異なり，ドライバ姿勢推定には消費リソースの制限や関節点が画角外に映る，近赤外線カメラでの撮影などの課題がある．本研究では，ShuffleNet V2 と Integral Regression を用いることで，高速かつ省メモリなドライバ推定を提案する．ShuffleNet V2 は畳み込みを適用するチャンネルを入れ替えながら，一部のチャンネルのみに畳み込みを適用するモジュールであり，畳み込み処理の演算量を大幅に減らす．Integral Regression は CNN が出力するヒートマップの重心位置を関節点座標として出力する手法であり，消費リソース削減のためにヒートマップの解像度を小さくした際の量子化誤差を抑制できる．また，本研究では関節点座標と同時に関節点有無も推定するドライバ姿勢推定を提案する．関節点有無を推定することで，関節点が画角外に映る多様なドライバ姿勢を把握するのに役立つ．更に近赤外線カメラを用いて撮影したドライバ姿勢推定用のデータセットを用いて提案手法を評価する．

ドライバの動作認識　人物動作認識でも CNN を用いた高精度な手法が数多く提案されているが，ドライバ動作認識はドライバ姿勢推定と同じく，消費リソースの制限，近赤外線カメラでの撮影，被写体とカメラの距離が近い，などの課題がある．本研究では，ドライバの姿勢と動作のマルチタスク学習を用いた高速かつ高精度なドライバ動作認識を提案する．ドライバ姿勢として関節点座標や関節点有無に加えて，各関節点の詳細な状態を示す”関節点状態”を用いる．関節点座標，関節点有無，関節点状態の 3 つのドライバ姿勢と動作のマルチタスク学習により，消費リソースを制限した

モデルでも高精度なドライバ動作認識が可能となる．また，本研究ではドライバの軽度な状態から深刻な状態及び自動運転中の動作なども幅広くカバーする 7 つのドライバ動作を認識する手法を提案する．提案手法の動作認識を用いることで多様なドライバ動作への対応が可能となる．

ドライバの眠気推定　ドライバ眠気推定の多くはドライバの強い眠気を検出する 2 値の眠気推定であるため，特徴量やネットワークモデルはドライバの強い眠気を検知するために設計されている．本研究では，弱い眠気を含んだマルチレベルの眠気推定に役立つ Average Eye Closure Time(AECT) と Soft PERCLOS の 2 つの時間特徴量を提案する．AECT は瞬きの際の平均閉眼フレーム数を示し，長時間目を閉じる傾向にある強い眠気と短時間の閉眼を伴う頻繁な瞬きを行う弱い眠気を区別するのに役立つ．Soft PERCLOS は完全に目が開眼していないフレームの割合を示し，閉眼フレームの割合を示す時間特徴量である PERCLOS では，捉えられないような弱い眠気を捉えるのに役立つ．また，本研究ではマルチレベルの眠気推定に有効なネットワークモデルとして，Parallel Linked Time-domain CNN を提案する．Parallel Linked Time-domain CNN は，複数の時間解像度に着目した特徴量を抽出可能であるため，目の状態の時間変化を捉えるのに役立つ．更に，本研究では実車で撮影したデータセットを構築し，実車特有の照明や振動などの影響を考慮した上で提案手法を評価する．

ネットワークモデルのコンパクト化　CNN は消費リソースが多いため，不要な重みを削減する枝刈り法 [4, 5, 6, 7, 8] や，ノードの重みの共有化法 [8, 9] など様々な工夫が提案されている．Hinton らが提案した Knowledge Distillation は大きな教師ネットワークの出力を用いて，小さな生徒ネットワークを学習することで，生徒ネットワークの精度を向上させる．Knowledge Distillation はいくつかの手法が提案されているが，教師ネットワークの中間層の情報を上手く生徒ネットワークに活かせていない，適用するネットワークの構造に制限がある，などの課題がある．本研究では，適用するネットワークの構造に制限がなく，教師ネットワークの中間層の情報を生徒ネットワークの学習に活かすことのできる Sequential Layer-wise Knowledge Distillation によるネットワークモデルのコンパクト化を提案する．

1.3 本論文の構成

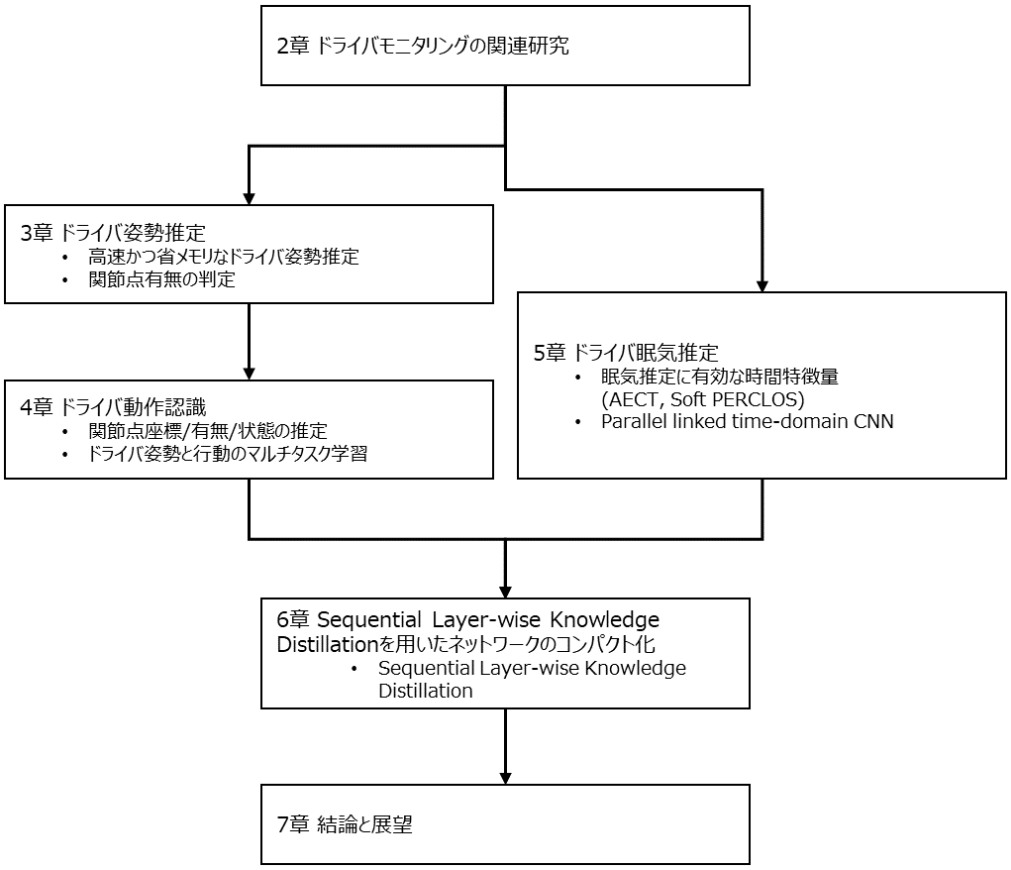

図 1.1: 本論文の構成

本論文は，図 1.1 のような章構成となる．2 章では，ドライバモニタリングシステムの関連研究についてまとめる．3 章では，ShuffleNet V2 と Integral Regression を用いた高速かつ省メモリなドライバ姿勢推定を提案する．4 章では，ドライバの関節点座標，関節点有無，関節点状態の 3 つのドライバ姿勢と動作のマルチタスク学習を用いたドライバ行動認識を提案する．5 章では，弱い眠気を含んだマルチレベル眠気推定に有効な時間特徴量とネットワークモデルを提案する．6 章では，Sequential Layer-wise Knowledge Distillation を用いたネットワークモデルのコンパクト化について提案する．7 章では，本論文の結論と展望について述べる．

第2章

交通事故とドライバモニタリング

自動車は移動手段として人々の生活に欠かせないものであるが，自動車による交通事故は大きな社会問題となっている．交通事故を減らすため，安全運転管理者制度の導入や自動運転車の開発，ドライバモニタリング技術の開発など民間，国，地方自治体が様々な取り組みが行われている．本章では，交通事故を減らすための取り組みやドライバモニタリングに使用されるセンシング方法について述べる．

2.1　自動車の事故に関する社会情勢について

自動車は移動手段として人々の生活に欠かせないものとなっているが，その便利さに反して，自動車による死傷者数は多く，大きな社会問題となっている．交通事故の死傷者数は昭和 45 年を過去最多として，年々減少傾向にあり令和元年は過去最少の 3,215 人となった [10]．交通事故は減少傾向にあるものの，高齢ドライバによる運転操作ミスによる事故や職業ドライバの超過勤務などによる居眠り事故などが社会問題となっており，以前として交通事故による死傷者をなくす取り組みは重要である．交通事故は死傷者がいること自体が大きな問題であるが，経済的な面でも大きな損失となる．内閣府による平成 21 年度を対象とした交通事故による経済的損失は約 6 兆 3,340 億円であり，GDP 比 1.3%と算定されている [11]．交通事故の損失額の詳細を図 2.1 に示す．

単位: 十億円

内訳項目			死亡	後遺障害	傷害	物損	合計
金銭的損失	人的損失	逸失利益・治療関係費・葬祭費	114	428	290	—	832
		慰謝料 [A]	87	100	340	—	527
		小計	201	528	630	—	1,359
	物的損失		3	26	433	1,249	1,711
	事業主体の損失		6	14	61	—	81
	各種公的機関等の損失		14	82	712	20	828
	金銭的損失合計 [B]		223	649	1,837	1,269	3,979
非金銭的損失	死傷損失 [C]		1,509	577	269	—	2,355
総計（慰謝料分除外）[B]-[A]+[C]			1,646	1,126	1,766	1,269	5,807
総計（慰謝料分除外せず）[B]+[C]			1,733	1,226	2,106	1,269	6,334

注1）死傷損失の値は「表 5-17　非金銭的損失額の全容」の死傷損失額による。なお、「後遺障害」は負傷QからOの合計、「傷害」は負傷Aの値である。
注2）四捨五入のため、各集計欄の値は必ずしも各欄の集計結果と一致しない。

図 2.1: 平成 23 年度交通事故の損失額．文献 [11] から引用．

令和元年度のドライバの事故の主な原因を図 2.2 に示す [12]．図 2.2 は，令和元年度の原付以上運転者 (自動車，自動二輪車及び原動機付き自転車の運転者) の法令違反別交通事故件数の内，安全運転義務違反に該当するものである．運転操作不適はハンドル誤りやブレーキとアクセルの踏み間違えなどを示す．漫然運転は眠気や考え事などが原因で，ぼんやりと運転している状態を示す．脇見運転はスマートフォンやカーナビの操作など，前方を注視していない状態を示す．動静不注視は横断歩道の歩行者への中止が足りなかった状態など，周辺環境に注意を配っていない状態を示す．安全不確認は左右の確認などの安全確認を行っていない状態を示す．安全速度は速度違反を示す．

図 2.2 に示す事故はブレーキやアクセルなどの車体情報の取得により一部で対応済みのもの，車内外のモニタリングが必要なもの，ドライバモニタリングで対応可能なものに分かれる．車体情報の取得により対応可能なものは運転操作不適，安全速度の 2 つとなる．車内外のモニタリングが必要なものは動静不注視となる．ドライバモニタリングで対応可能なものは漫然運転，脇見運転，安全

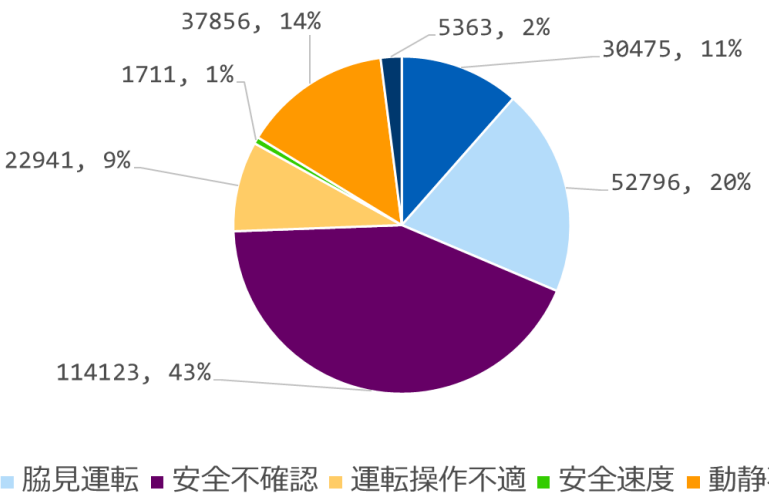

図 2.2: ドライバ事故の主な原因：令和元年度の原付以上運転者 (自動車，自動二輪車及び原動機付き自転車の運転者) の法令違反別交通事故件数の内，安全運転義務違反に該当するもの

不確認の 3 つであり，これらの要因による事故が約 75%を占めており，ドライバモニタリングの導入により多くの事故を抑制できる．本論文では，漫然運転，脇見運転，安全不確認の 3 つを抑制するドライバモニタリング技術を提案する．3 章で提案するドライバ姿勢推定と 4 章で提案するドライバ動作認識は脇見運転や安全不確認の抑制につながる．5 章で提案するドライバ眠気推定技術は漫然運転の抑制につながる．

2.1.1 安全運転に対する政府・自治体の取り組み

バスやトラックなどの職業ドライバの居眠りなどによる事故が起きており，社会問題となっている．ドライバが少ないことによる長時間労働が原因により，居眠りなどの事故が起こる．それらの事故を防ぐため，安全運転管理者制度が導入されている．安全運転管理者制度では，事業主などの使用者は安全運転管理者を選任し，各自治体に届けることを義務付けている．安全運転管理者は，運航計画や運転日誌の作成，安全運転の指導を行う．運転日誌の作成は負荷もあるため，ドライバの運転挙動を計測し，自動的に日誌を作る取り組みもなされている [13]．このようにドライバの運転挙動を正確に計測することは，安全運転への取り組みに欠かせない技術となっている．

2.1.2 自動運転を巡る社会動向

「官民 ITS 構想・ロードマップ 2020」[14] では，自動運転車により様々な社会課題の解決や産業へ良い影響を与えることが期待されている．自動運転車による人間よりも安全かつ円滑な運転で交通事故の削減，交通渋滞の緩和，環境負荷の低減などが期待される．また，自動運転による運転者の負担軽減により運転の快適性の工場，高齢者等の移動支援なども期待されている．産業について

は自動運転の産業規模・波及性が高い汎用的な技術により，自動車関連産業の競争力向上や，運輸・物流・農業などの関連産業の生産性向上に役立つことが期待されている．

自動運転は自動化される範囲でレベルが定義されている．表 2.1 は米 Society of Automotive Engineers (SAE) が提唱する自動運転レベルである．レベル 0 は自動化が全く行われない通常の自動車である．レベル 1〜3 までは一部が自動運転になっており，緊急時などには手動運転の必要がある．レベル 4〜5 はドライバによる手動運転を義務付けておらず，ドライバを必要としない．

表 2.1: 米 SAE が提唱する自動運転レベル

レベル	種類	操縦の主体
レベル 0	自動化なし	運転者
レベル 1	運転支援 (ADAS)	運転者
レベル 2	部分的な運転自動化	運転者
レベル 3	条件付き運転自動化	システム (作業継続が困難な場合は運転者)
レベル 4	高度な運転自動化	システム
レベル 5	完全運転自動化	システム

自動運転は開発が進んでおり，2020 年 4 月には日本でもレベル 3 の条件付き運転自動化が解禁された．内閣府が示した指針「官民 ITS 構想・ロードマップ 2020」[14] によると，高速道路におけるレベル 3 の市場化を 2020 年，レベル 4 の市場化を 2025 年を目指している．図 2.3 に示す通り，レベル 5 の完全運転自動化は難しく，2023 年まではレベル 3 以下の自動運転が主流になると考えられている．

レベル 3 以下の自動運転では，緊急時などに自動運転から手動運転に切り替わるため，ドライバが運転できる状態かどうかを確認するドライバモニタリングシステム (DMS) が必要となる．

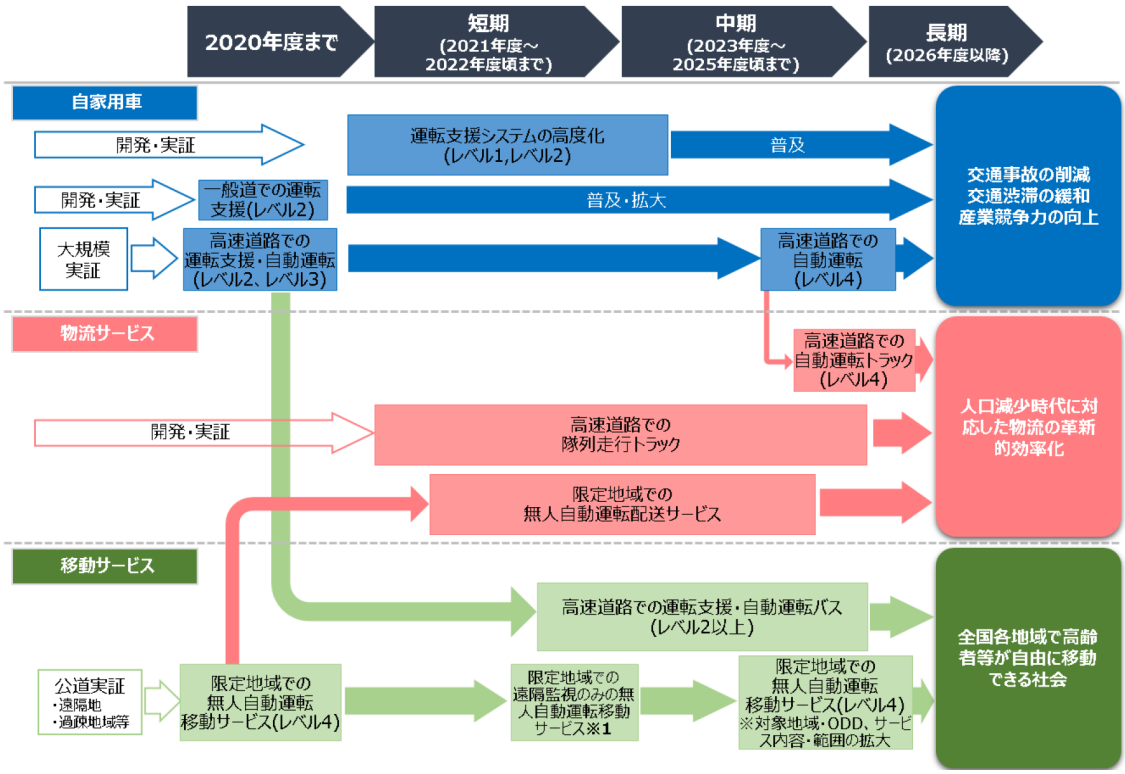

図 2.3: 自動運転のロードマップ．文献 [14] から引用．

2.2 ドライバモニタリングの手法

ドライバモニタリング技術はドライバの状態を取得するセンサにより，生体センシング，車体センシング，画像センシングの 3 つに分けられる．表 2.2 に各センシング方法の特徴を示す．

生体センシングの代表的なものとして，Electroencephalograms (EEG)，Electrocardiograms (ECG)，Electrooculograms (EOG)，などがある．生体センシングは脳波や心電などの外面には表れないような内面の情報を取得できるため，眠気推定に使用される．生体センシングで得られる内面の情報は居眠りや病気などの兆候を捉えることに役立つものの，センサをドライバの皮膚に取り付けるため，ドライバへの身体的な負担が大きく，日常的に使用することは難しい．

車体センシングの代表的なものとして，ホイールの動き，ブレーキの動き，レーンの逸脱，などがある．車体センシングは車体情報により，ドライバ動作の認識に役立つ．車体センシングはドライバへの身体的な負担はないものの，車種やドライバの運転技能，道路などの環境の影響を受けやすい．そのため，決まった車種への導入は問題ないが，幅広い車種への導入は難しい．

画像センシングではカメラを用いてドライバを撮影することで，ドライバの状態を把握する．画像センシングはドライバへの身体的な負担がなく，車種や道路などの環境の影響も受けにくい．更に画像センシングにより，ドライバの姿勢推定，動作認識，眠気推定が可能となる．従って，本研究では画像センシングを用いて，ドライバの姿勢推定，動作認識，眠気推定を行う手法を提案する．

表 2.2: センシングの種類

センシング	目的	例	Pros.	Cons.
生体	眠気推定	EEG（脳波）， ECG（心電）， EOG（眼電位），など	内面情報を取得可	身体的な負担
車体	動作認識	ホイールの動き ブレーキの動き， レーンの逸脱，など	身体的負担なし	環境に左右
画像	姿勢推定 動作認識 眠気推定	カメラ	身体的負担なし， 環境に左右されない	内面情報の取得不可

2.3 まとめ

自動車による交通事故と，交通事故を減らすための取り組みについて解説した．交通事故の主な原因のうち，ドライバモニタリングで対応可能なものは漫然運転，脇見運転，安全不確認の 3 つであり，これらが原因の事故が 75%を占めている．3 章と 4 章では脇見運転や安全不確認の抑制につながるドライバ姿勢推定とドライバ動作認識を提案する．5 章では漫然運転の抑制につながるドライバ眠気推定を提案する．また，ドライバモニタリング技術は生体センシング，車体センシング，画像センシングの 3 つのセンシング方法により実現される．本研究では，画像センシングによるドライバモニタリング技術を提案する．

第3章

高速かつ省メモリなドライバ姿勢推定

自動車の安全な運転を実現するために，ドライバモニタリングシステム (DMS) の導入が必要とされている．DMS は眠気や姿勢などのドライバ状態を推定することで，ドライバの異常状態を検知するシステムである．国土交通省が作成したドライバ異常自動検知システム基本設計書 [15] で示されている通り，ドライバ姿勢の崩れを検知することはドライバの異常検知につながる．

[15] では，ドライバ姿勢情報は物理量（角度や関節位置）への置き換えが比較的行いやすく，数値でしきい値を定義することが可能であると言及されている．そのため，ドライバ異常を直接推定する方式と比べて，ドライバ姿勢推定を用いた異常検知では，誤識別の要因が姿勢推定結果にあるのか，しきい値などの異常判定方法にあるのかを解析しやすい．本章では，ドライバの異常検知を目的として，高速かつ省メモリなドライバ姿勢推定を提案する．

従来の DMS の多くはドライバの顔部分のみに着目し，脇見などを検知する．しかし，飲食や通話など多様なドライバの状態を検知するためには，頭部の姿勢だけではなく，手や首元などのより多くの関節点を捉えることが重要となる．本章では，頭部だけではなく，首元，両手といった上半身全体のドライバ姿勢を推定する．

人物の関節点座標を推定する人物姿勢推定は，コンピュータビジョンの分野で活発に研究されている．近年，OpenPose[1] など Deep Convolutional Neural Network(CNN) を用いた高精度な人物姿勢推定が提案されている．これらの関連研究では，MS-COCO[16]，MPII[17]，LSP[18]，FLIC[19] などのデータセットが使用されている．これらのデータセットを用いて評価していた関連研究の人物姿勢推定と，ドライバ姿勢推定では消費リソースやデータセットの条件などが異なる．

デスクトップ PC やサーバーなどと比べて，DMS は電力量が制限されるため，消費リソースが限られた組込み機器での動作が求められる．CNN を用いた姿勢推定は高精度であるが，消費リソースが多くなる傾向にある．CNN を用いた姿勢推定を DMS に搭載するためには，ネットワークモデルを軽量化し，高速かつ省メモリなモデルを構築する必要がある．本研究では，ShuffleNet V2 と Integral Regression をベースとし，高速かつ省メモリなドライバ姿勢推定を提案する．

CNN を用いた姿勢推定に用いられる MS-COCO などのデータセットは，被写体とカメラ間の距離が遠いため，被写体の全身が映るシーンが多く，ほとんどの関節点が画像中に映る．そのため，人物姿勢推定で一般的に行われる評価では，推定した関節点座標と正解の誤差のみを評価し，関節点が画像中に映っているかどうかは評価しない．一方，DMS では車内にカメラを設置するため，被写体とカメラ間の距離が近く，上半身など一部の関節点しか映らないことが多い．更に，ドライバが急病などで倒れた場合には，頭部が画像に映らないため，画像中に関節点が映っているかどうか（関節点

有無）の判定が，ドライバの異常検知に役立つ．しかし，従来の人物姿勢推定は関節点有無を評価しないため，ドライバ姿勢推定に適さない．本章では，関節点座標だけでなく関節点有無も推定するモデルを提案する．

人物姿勢推定のデータセットは，可視光カメラで撮影されたものが多い．一方，DMS では夜間でもドライバを撮影する必要があるため，近赤外カメラで撮影されることが多く，色情報が失われる．本章では，人物が上半身など一部の関節点のみしか映らない，かつ，色情報がないデータセットを作成し，DMS での利用を想定した評価実験を行い，提案手法の効果を示す．

3.1 関連研究

人物姿勢推定はトップダウン型とボトムアップ型に分けられる．トップダウン型では，人検出や顔検出を用いて人物領域を特定し，特定した人物領域に対して姿勢推定を行う．一方，ボトムアップ型では，画像から人物の関節候補点を抽出し，それらの関節候補点を関節間の関係性などに基づき，つないでいくことで複数人物の姿勢推定を一度に行う．

3.1.1 トップダウン型姿勢推定

CNN を用いた姿勢推定に関する初期の研究では，画像から関節点座標を直接推定する回帰を用いた手法が提案されていた．Toshev らは，段階的に関節点座標を推定する DeepPose を提案した [20]．DeepPose では入力画像から関節点座標を推定した後，推定した座標付近の画像を切り出し，その画像から再度関節点座標を推定する．DeepPose では，このような処理を繰り返すことで関節点座標の推定精度を高める．

人物の関節点座標を一意に決定することは難しく，回帰を用いたネットワークの学習は難しい．そこで，Tompson らは関節点座標をヒートマップとして表現することで DeepPose よりも高精度な姿勢推定を実現した [21]．ヒートマップでは学習に使用する関節点座標の正解を，正解の関節点座標を中心として広がる 2 次元の正規分布として表現する．ヒートマップでは人物の関節点座標を一意に決定しないため，学習が安定し，精度が高くなる．以降，ヒートマップを用いた姿勢推定が主流となり，ヒートマップを出力するネットワーク構造を工夫した手法が提案された．

Wei らは，段階的に姿勢推定の精度を向上させていく Convolutional Pose Machines (CPMs) を提案した [22]．最初のステージでは画像を入力として，ヒートマップを出力する．以降のステージでは，前ステージの出力結果であるヒートマップと，画像に畳み込みとプーリングを適用した特徴マップを入力として，関節点座標を表すヒートマップを出力する．各ステージでヒートマップと正解の誤差を計算して学習することで，勾配消失問題を回避している．また，各ステージにはプーリング層があるため，ステージを経るごとに受容野が大きくなり，より大局的な特徴を考慮した姿勢推定が行える．しかし，CPMs では，ステージを経るごとに，高解像度の情報が失われる．

Newell らは，高解像度から低解像度にするダウンサンプリング処理と，低解像度から高解像度にす

るアップサンプリング処理を行う Hourglass Module を提案した [23]．ダウンサンプリング処理とアップサンプリング処理を行うことで様々な空間解像度の情報を抽出できる．また，Hourglass Module には skip connection があり，ダウンサンプリング処理でも高解像度の情報が保持される．また，CPMs と同様に，各 Hourglass module で正解の関節点座標を用いて学習するため，Hourglass module を重ねても勾配消失を抑制できる．

Yan らは，空間解像度の変化に頑健な Pyramid Residual Module を提案した [24]．Hourglass module では，Skip connection を一つの空間解像度情報を保持するために使用する．一方，Pyramid Residual Module では，異なる空間解像度間で Bottom-up と Top-down 処理を並列に行うことで多様な空間解像度の特徴を捉えられる．

Hourglass や Pyramid Residual Module などは高精度である一方，ネットワーク構造が複雑である．そのため，Xiao らは，ResNet と少数の Deconvolutional 層を用いた単純なネットワーク構造を用いてベースラインのモデルを提案した [25]．提案手法であるベースラインモデルは，単純な構造で既存手法と同等以上の精度を達成した．

Sun らは [25] をベースとしてモデル構造を改良し，空間解像度ごとにネットワークを分岐させて処理する HRNet を提案した [26]．HRNet では，高解像度の情報は解像度を落とすことなく，下位層から上位層へ伝えられる．そのため，Pyramid Residual Module と同様に，高解像度の特徴を抽出することができ，高精度な姿勢推定を実現している．

3.1.2 ボトムアップ型姿勢推定

Pishchulin らは，ヒートマップで推定した関節候補点同士をつなぎ合わせてグラフとみなし，各人物の正しい関節点の対応付けを整数線形計画問題として最適化する DeepCut を提案した [27, 28]．DeepCut は整数線形計画問題による最適化の処理時間が多く，リアルタイムの姿勢推定が難しかった．

Cao らは関節点間の関係をベクトル場で表現した Part Affinity Field (PAF) を用いて関節候補点をつなげることで，複数人の姿勢をリアルタイムに推定する OpenPose を提案した [1]. DeepCut に比べて，PAF を用いた関節候補点の対応付けは，画像中に映る人物数が増えた場合でも，一定の処理時間で複数人物の姿勢を推定できる．

Kreiss らは，遠方に映る低解像度の人物や，混雑しているシーンであっても高精度な姿勢推定を行うため，高解像度なヒートマップを生成する Part Intensity Fields (PIF) と，混雑しているシーンでも正確に関節同士のつながりを表現できる Part Association Fields (PAF) を提案した [29].

3.1.3 関連研究のドライバ姿勢推定への適用

ボトムアップ型の姿勢推定は，関節点間の関係を用いて関節候補点をつなげることで各人物の関節点の対応付けを行うが，画像中に一部の関節点しか映ることのないドライバ姿勢推定では，多くのシーンで関節点間の関係を利用することができない．そのため，関節点間の関係を表すヒートマップを出力する畳み込み層は，不要な演算量の増大につながる．従って，本研究ではトップダウン型

の姿勢推定を対象とする．トップダウン型では顔検出や人物検出を行い，人物領域を抽出したのち，姿勢を推定するが，本研究で扱うドライバ姿勢推定ではドライバのみが映るシーンを想定しているため，入力画像から直接ドライバの姿勢を推定する．

3.1.4　人物姿勢推定のデータセット

人物姿勢推定に用いるデータセットには，Leads Sports Pose Dataset (LSP)[18]，Frames Labeled In Cinema (FLIC)[19], MPII Human Pose Dataset (MPII)[17], Microsoft Common Objects in Context (COCO)[16] がある．各データセットの画像例を図 3.1 に示す．これらのデータセットは，被写体とカメラ間の距離が比較的遠く，多くのシーンで被写体の全身が映る．更にこれらのデータセットは，可視光カメラで撮影されたものであり，色情報が利用できる．一方，ドライバ姿勢推定では，被写体とカメラ間の距離が近いため，多くのシーンで上半身など一部の関節点しか映らない．また，夜間でもドライバを監視する必要があるため，近赤外カメラで撮影されたグレイスケール映像となる．そのため，人物姿勢推定で用いられるこれらのデータセットは，ドライバ姿勢推定には適さない．

図 3.1: 姿勢推定データセットの画像例．

3.1.5　人物姿勢推定の評価方法

人物姿勢推定の評価指標では，Percentage of Correct Parts (PCP)，Percentage of Correct KeyPoints (PCK)，Percentage of Detected Joints (PDJ)，Average Precision (AP)，Average Recall (AR) が用いられる．

PCP，PCK，PDJ は，単一の人物姿勢推定に用いられる指標であり，推定した関節点座標と正解の誤差がしきい値よりも小さいときに推定結果が正しく行われたとし，その正解率を評価指標とする．

PCP, PCK, PDJ は式 (3.1) を用いて算出され，推定座標と正解座標が一致するときに 1 となる．δ は推定座標と正解座標の距離がしきい値以下の場合に 1 となる関数を示す．th はしきい値，i は各関節点，N は関節点の数を示す．PCP では隣接する 2 つの関節点間の距離，PCK では任意のしきい値（人物頭部のサイズから決定されることが多い），PDJ では胴の直径からしきい値 th を設定する．

$$PCK = \frac{\sum_i^N \delta(d_i > th)}{N} \tag{3.1}$$

AP と AR は，COCO で用いられる評価指標であり，推定した関節点と正解の類似度を示す尺度である Object Keypoint Similarity (OKS) を用いて算出される．OKS は式 (3.2) を用いて算出され，推定座標と正解座標が一致するときに 1 となる．d_i は関節点 i の推定座標と正解座標の誤差，s は人物領域の面積，k_i は関節点毎に設定される定数 (推定が難しいほど関節点ほど大きい値)，v_i は関節点のアノテーションの有無 (関節点のアノテーションがある場合は，δ が 1 となる) を示す．

$$OKS = \frac{\sum_i \exp(\frac{-d_i^2}{2s^2k_i^2})\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{3.2}$$

Precision と Recall は式 (3.3) を用いて算出される．TP は正しく予測できた数，FP は誤って検出した数，FN は誤って検出できなかった数を示す．AP と AR は OKS のしきい値を変化させたときの Precision と Recall の平均となる．

$$Precision = \frac{TP}{TP + FP} Recall = \frac{TP}{TP + FN} \tag{3.3}$$

これらの評価尺度は，画像中に対象の関節点が映っていない場合には，評価尺度に反映されない．従って，人物姿勢推定の評価指標は関節点有無の評価には適さない．

3.1.6 ドライバ姿勢推定

DMS に関する研究の多くが，ドライバの顔部分のみに着目したドライバ姿勢推定を行っていた [30, 31]．しかし，飲食や通話など多様なドライバの状態を検知するためには，頭部の姿勢だけではなく，手や首元などの関節点座標を推定する必要がある．Eshed らは 2 台のカメラを用いて，手と顔の姿勢を捉えてドライバがどの部分に注目しているかを推定した [32]．しかし，Eshed らは可視光カメラで撮影した画像を用いており，夜間のシーンには対応していない．また，手の姿勢はハンドル，ギア，車載機器のどの領域にあるかのみを判定しているため，飲食や通話などの多様なドライバの状態を推定することは難しい．

3.2 提案手法

3.2.1 概要

ドライバ姿勢推定では組込機器で動作可能な，高速かつ省メモリなネットワークモデルが求められる．消費リソースを削減するためには，精度を低下させることなく，ヒートマップの解像度を落としたり，演算量の多い畳み込み処理を減らす必要がある．本研究では，これらを実現するために，Integral Regression[33] と ShuffleNet V2[34] をベースとしたネットワークモデルを提案する．また，関節点座標の推定と同時に，画像中に各関節点が映っているかどうか（関節点有無）を判定することで，一部の関節点しか映らないような DMS 特有のシーンに対応した手法を提案する．

提案手法のネットワークモデルを図 3.2 に示す．HRNet[26] は高精度であるが，複数の解像度を処理するため畳み込み処理の演算量が増大する傾向にあるため，提案手法は Hourglass[23] をベースとしたモデルとする．入力画像から 2 つの Hourglass Module により特徴抽出を行った後，各関節点に対して座標と有無を推定する．関節点座標は特徴マップに対して畳み込みと Softmax を適用してヒートマップを生成し，ヒートマップに対して Integral Regression を適用して座標を推定する．関節点有無は特徴マップをヒートマップで重み付けし，Global Average Pooling を適用する．その後，全結合層を適用して関節点有無を表す信頼度を出力する．

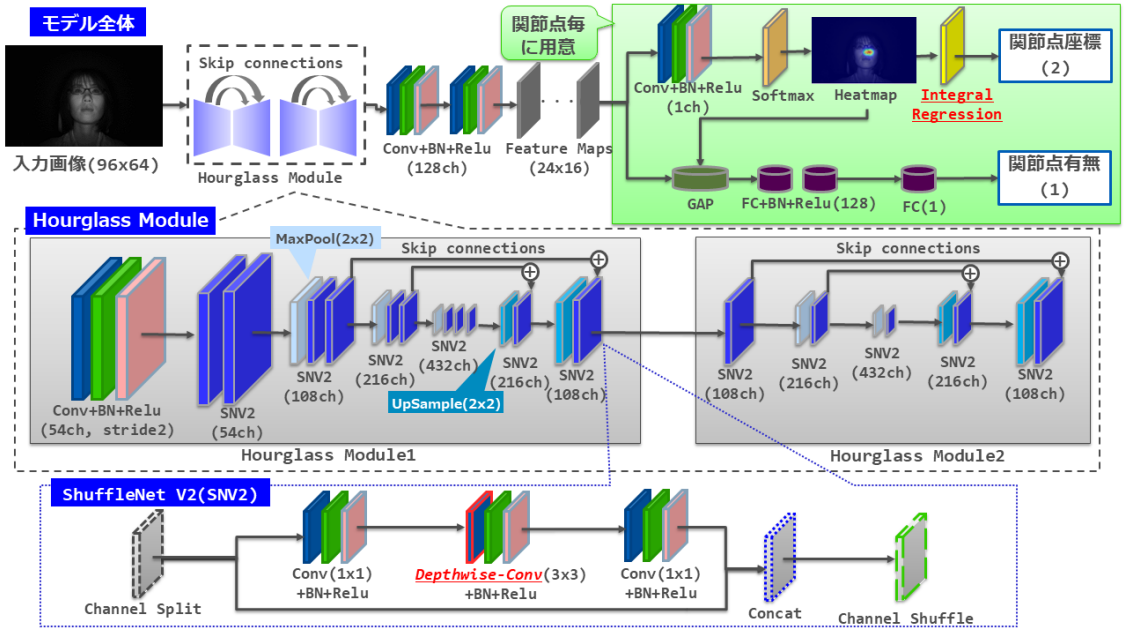

図 3.2: ネットワークモデルの概要 (提案手法, 入力画像 96×64，54ch)

3.2.2 高速かつ省メモリなドライバ姿勢推定

高速かつ省メモリなドライバ姿勢推定を実現するため，精度低下を抑制しつつ，入力画像の低解像度化と畳み込み処理の高速化を行う．

入力画像の低解像度化は，消費リソースの削減につながるが，関節点座標を表すヒートマップも低解像度となるため，量子化誤差を招く．そのため，提案手法では Sun らが提案した Integral Regression[33] を用いて，量子化誤差を抑制しつつ，入力画像を低解像度にする．また，人物姿勢推定では Convolutional Neural Network (CNN) が一般的に良く使用されるが，CNN は畳み込みの演算量が多く，処理時間が増大する傾向にある．本研究では，Ma らが提案した ShuffleNet V2[34] を用いて，畳み込みの消費リソースを削減する．

■ Integral Regression

ヒートマップを用いた人物姿勢推定では，ヒートマップ上の最大値を取る座標を関節点座標とする．そのため，ヒートマップの解像度が低くなるほど，関節点座標の量子化誤差が大きくなる．一方，ヒートマップの解像度を高くすると消費リソースが増加する．そこで，Sun らは Integral Regression を用いてヒートマップ分布の重心位置を関節点座標として出力することで，サブピクセル単位での座標推定を可能にした [33]．ヒートマップの解像度が入力解像度より低い場合でも，Integral Regression では高精度な関節点座標の推定が可能であり，消費リソースの削減につながる．また，Integral regression は最大値に基づく関節点座標の推定と異なり，ヒートマップ分布全体に関して微分可能であるため End-to-end で学習できる．

提案手法では Integral Regression を用いて，量子化誤差を抑制しながら，入力画像とヒートマップの解像度を小さくする．従来のヒートマップを用いた関節点座標の推定では，式 (3.4) の argmax を用いてヒートマップの最大値から関節点座標を推定する．H_k は k 番目の関節点のヒートマップ，p はヒートマップ上の座標，J_k は k 番目の関節点の座標を示す．Integral Regression では式 (3.5) のように，ヒートマップの期待値から関節点座標を算出する．Ω は座標の集合を示す．なお，Integral Regression で使用されるヒートマップ $\hat{H}_k$ は，式 (3.6) のような softmax を用いて，全ての要素が非負であり，全ての要素を足すと 1 になるように正規化される．式 (3.5) は Soft Argmax としても知られる．

$$J_k = arg\max_p H_k(p) \tag{3.4}$$

$$J_k = \int_{p\in\Omega} p \cdot \hat{H}_k\ (p) \tag{3.5}$$

$$\hat{H}_k(p) = \frac{e^{H_k(p)}}{\int_{q\in\Omega} e^{H_k(q)}} \tag{3.6}$$

■ **ShuffleNet V2**

畳み込み処理の演算量を減らすため，提案手法では通常の畳み込み層の代わりに ShuffleNet V2 で提案されたモジュールを用いる．ShuffleNet V2 モジュールの構成を図 3.2 の最下部に示す．このモジュールでは，初めにチャンネルを分割し，一方では畳み込み処理を行い，もう一方では畳み込み処理を行わない．これらのチャンネルを結合した後，チャンネルの順序を入れ替えることで畳み込み処理の回数を削減しながら，全てのチャンネルに畳み込み処理を適用する．また，ShuffleNet V2 は 3 つの畳み込み層から構成される．1 つ目は，1×1 のフィルタを用いた畳み込み処理，2 つ目は Depthwise の畳み込み処理，3 つ目は 1 つ目と同じく 1×1 のフィルタを用いた畳み込み処理である．2 つ目の Depthwise の畳み込み処理ではチャンネル方向の畳み込み処理を行わず，空間方向に対してのみ畳み込み処理を行うため，通常の畳み込み処理よりも計算量が少ない．

3.2.3 関節点有無の判定

人物姿勢推定では，多くの関節点が画像中に映るシーンで評価を行っていた．そのため，関節点座標と正解座標の誤差のみを評価し，関節点有無は評価していなかった．一方，ドライバ姿勢推定では，上半身など人物の一部分しか映らないため，画角内に関節点が入らないシーンが多い．例えば，意識を失っており，頭部が映っていない場合には関節点有無がドライバの異常検知に役立つ．また，人物姿勢推定では関節点座標をヒートマップから推定しているが，ヒートマップ上の最大値がしきい値以上かどうかで，関節点有無を推定できる．しかし，そのような場合は関節点有無の学習を行っていないため，関節点毎にヒートマップの最大値が大きく異なり，適切なしきい値の設定が難しい．提案手法では関節点座標の推定と同時に関節点有無を推定するため，しきい値の設定が不要となる．

3.2.4 学習

提案手法では入力画像 1 枚に対して，ヒートマップ，関節点座標，関節点有無の 3 つの損失を計算し，一度にネットワークモデルを学習する．ヒートマップの損失関数は L2-loss を用いる．ヒートマップの Ground Truth (GT) は正解の関節点座標を中心として広がる正規分布を用いて作成する．ヒートマップの損失 L_H は，式 (3.7) となる．H_k は k 番目の関節点のヒートマップ，p はヒートマップ上の座標，H_k^* はヒートマップの GT，K は関節点の数となる．

$$L_H = \sum_{k=1}^{K} \sum_{p} \|H_k(p) - H_k^*(p)\|_2^2 \tag{3.7}$$

提案手法では Integral Regression を用いるため，関節点座標についても損失を計算できる．関節点座標の GT は入力画像サイズを基準とした座標とする．関節点座標の損失関数 L_C は，L1-loss を用い

て，式 (3.8) となる．C は関節点座標，C^* は関節点座標の GT，j は j 番目の関節点を示す．

$$L_C = \sum_{j=1}^{J} \|C(j) - C^*(j)\| \tag{3.8}$$

関節点有無の損失関数 L_D は，クラス間のデータの偏りに対応するため，Focal loss[35] を用いて，式 (3.11) となる．x は関節点有無を示す 2 値の出力値，t は関節点有無の GT，CE はクロスエントロピー，γ は損失を調整するパラメータを示す．本研究では，γ を 2 とする．t は関節点が画像中に映る場合に 1，画像に映らない場合に 0 とする．

$$CE(x_t) = -\log(x_t) \tag{3.9}$$

$$x_t = \begin{cases} x & t = 1 \\ 1 - x & otherwise \end{cases} \tag{3.10}$$

$$L_D = -(1 - x_t)^{\gamma} \log(x_t) \tag{3.11}$$

関節点有無の GT は関節点が画像中に映る場合に 1，映らない場合に 0 とする．

提案手法のネットワークモデルは，これら 3 つの損失をあわせた式 (3.12) を用いて学習する．

$$L = L_H + L_C + L_D \tag{3.12}$$

3.3 実験

提案手法の有効性を確認するため，既存手法との精度比較を行う．DMS では一般的な人物姿勢推定と比較して，消費リソースが限られるため，提案手法と既存手法の計算量が同等になるようパラメータ数を揃えたネットワークモデルを用いる．また，本研究では Ablation Study として，提案手法を構成する Integral Regression，ShuffleNet V2，関節点有無の判定の 3 つの要素を除いた際の精度比較を行う．最後にドライバの行動パターン毎の評価を行い，行動パターンの傾向を確認する．

3.3.1 実験データ

本研究ではドライビングシミュレータを用いて，16 パターンの動作を 100 人の被験者で撮影し，ドライバ姿勢推定の評価データセットを作成した．16 パターンの動作は，前方注視，脇見，もたれ，ストレッチ，振り返り，眠気，スマホ操作，通話，カメラ撮影，読書，飲食，着替え，突っ伏し，パニック，居眠り，赤ちゃんを抱く，となる．100 人の被験者のデータのうち，50 人分を学習に，残りの 50 人分を評価用に用いる．外乱光の影響を軽減するため，我々が試作した近赤外線カメラを用いて撮影した．撮影した動画の解像度は 752×480 であり，それを 96×64 に縮小して使用する．ま

図 3.3: データセットの画像例（同意書により画像利用許諾確認済み）

た，計算量を減らすため，10fps で撮影した動画を 1fps に間引いて使用する．540,434 枚を学習用，528,124 枚を評価用として用いる．撮影した映像にはドライバの上半身のみが映るため，関節点は頭部中心，首元，左手中心，右手中心の計 4 点とする．図 3.3 に撮影した画像の例を示す．

3.3.2 評価実験のパラメータ

■ 入力画像，出力ヒートマップ

本実験では入力画像を 96×64 にリサイズしたのち，Global Contrast Normalization[36] を用いて正規化を行う．また，本実験で用いるヒートマップのサイズは 24×16 とする．評価は入力画像と同じ 96×64 の解像度に対して行う．

■ ネットワーク

本実験では提案手法と既存研究の HRNet[26]，Hourglass[23]，OpenPose[1] を比較する．また，消費リソースが制限された組込み機器への搭載を想定して，1 回の姿勢推定にかかる演算量が 100, 50, 25MFLOPs 程度になるモデルを用いて実験を行う．各モデルの消費リソースを表 3.1 に示す．畳み込み層の出力チャンネル数を減らすことで演算量を調整する．提案手法のネットワーク構成を，図 3.2 に示す．本実験で用いる提案手法の 3 つのモデルは，Hourglass Module 内の畳み込み層での出力チャンネル数が異なる．図 3.2 に示すモデルは，54ch モデルであり，Hourglass Module に含まれる畳み込み層の出力チャンネル数は 54ch，108ch，216ch，432ch となる．32ch モデルの出力チャンネル数は 32ch，64ch，128ch，256ch，24ch モデルは 24ch，48ch，96ch，192ch となる．本実験に用いる HRNet は，Sun らが提案したモデルと同様に 4 つのステージから構成される．演算量を調整するため，HRNet でも畳み込み層の出力チャンネル数を削減する．本実験では，1 つ目のステージの出力チャンネル数を 16ch，12ch，8ch に減らした HRNet のモデルを用いる．2 つ目以降のステージも，出力チャンネル数は 1 つ目のチャンネル数に合わせて削減する．本実験に用いる Hourglass は，提案手法で用いる Hourglass Module と同じものを使用する．ただし，ShuffleNet V2 は通常の畳み込み層に置き換え，Integral Regression や関節点有無の出力は行わない．Hourglass でも畳み込み層の出力チャンネル数を削減して演算量を調整する．本実験に用いる OpenPose では，学習を安定させるため，

Cao らが提案したモデルに対して，Batch Normalization(BN) 層 [37] を追加する．また，Cao らのモデルでは複数ステージから構成されているが，処理量削減のため，本実験では 1 つ目のステージのみ使用する．更に 1 層目の畳み込み層の出力チャンネル数を，4ch，6ch，8ch に減らし，以降の畳み込み層の出力チャンネル数も，1 層目に合わせて削減する．

表 3.1: ネットワークの演算量及びパラメータ数．OpenPose の演算量には PAF による関節点同士の関連付け処理部は含まない．

手法	MFLOPs	パラメータ数 [M]
100 MFLOPs		
HRNet [26], 16ch	113.0	3.90
Hourglass [23], 20ch	109.2	1.43
OpenPose [1], 8ch	111.2*	0.13
提案手法, 54ch	102.16	1.14
50 MFLOPs		
HRNet [26], 12ch	65.3	2.18
Hourglass [23], 15ch	61.6	0.81
OpenPose [1], 6ch	62.7*	0.07
提案手法, 32ch	43.0	0.51
25 MFLOPs		
HRNet [26], 8ch	28.3	0.97
Hourglass [23], 10ch	27.6	0.36
OpenPose [1], 4ch	28.0*	0.03
提案手法, 24ch	28.0	0.36

■ ハイパーパラメータ

初期の学習率は 0.001 とする．300iteration の間に 0.004 まで増加させた後，5000iteration ごとに学習率を半減し，合計で 30000iteration の学習を行う．重み減衰は 0.0001，勾配のクリッピングは 5.0 とする．バッチサイズは 1GPU あたり 96 に設定し，4 つの GPU を用いる．

■ 学習の詳細

Data Augumentation　手が映っているデータと映っていないデータが同等程度になるよう over sampling を行う．また，Data Augmentation として，下記 4 つの処理を行う．

- 画像の高さと幅の ±25%の範囲でランダムに並行移動
- 0.8〜1.75 倍の範囲でランダムに拡大縮小
- 50%の確率で左右反転
- 0.5〜1.5 倍の範囲でランダムに明度調整

Ground Truth　ヒートマップの学習に用いる Ground Truth (GT) は，正解の関節点座標から σ が 2 ピクセルの正規分布に従って作成したヒートマップを用いる．また，関節点有無の GT は関節点が画像に映る場合に 1，画像に映らない場合に 0 とする．

■ 評価方法

評価指標　本実験では関節点座標の評価指標は Probability of Correct Keypoints (PCK) を用いる．本実験では解像度が 96×64 の入力画像に対して，PCK のしきい値を 6 ピクセルとする．関節点有無の評価指標は，mean Average Precision (mAP) を用いる．

関節点座標，関節点有無の推定　提案手法では Integral Regression を用いるため，関節点座標はモデルの出力をそのまま用いる．一方，既存研究ではヒートマップ上で最大値を取る座標を関節点座標とする．また，提案手法では関節点有無の判定もモデルの出力を用いる．既存研究ではヒートマップの最大値がしきい値以上かどうかで関節点有無を判定する．

3.3.3　精度比較

関節点座標の精度比較結果を表 3.2 に示す．なお，OpenPose では PAF を用いて，関節点の関連付けを行うが，本実験では単一関節点のみ出現するシーンが多いため，関節点の出力のみを用いて，評価する．全ての演算量において，既存研究よりも提案手法の方が精度が高い．特に両手の精度では，提案手法は既存研究よりも 5〜10%程度改善している．従って，提案手法は消費リソースを削減した時の精度低下を抑制できる．OpenPose はパラメータ数が少ないが，50MFLOPs，25MFLOPs のように演算量を大幅に削減した場合に，提案手法よりも精度が大きく低下する．そのため，OpenPose では省メモリ化と高速化を両立しようとした場合に精度低下が起こる．

関節点有無の精度比較結果を表 3.3 に示す．100MFLOPs と 50MFLOPs のモデルでは，モデルによる精度の差がほとんどない．一方，25MFLOPs のモデルでは，HRNet の精度が Hourglass，OpenPose，提案手法と比べて低くなっている．Hourglass は提案手法よりも mAP が 1%程度高い．これより，関

表 3.2: 精度比較 (関節点座標)

手法	PCK-6px (頭部)	PCK-6px (首元)	PCK-6px (右手)	PCK-6px (左手)
100 MFLOPs				
HRNet [26], 16ch	90.4%	91.5%	82.9%	72.2%
Hourglass [23], 20ch	89.9%	91.43%	82.5%	74.3%
OpenPose [1], 8ch	89.6%	88.2%	83.5%	74.9%
提案手法, 54ch	**91.9%**	**94.5%**	**90.5%**	**84.0%**
50 MFLOPs				
HRNet [26], 12ch	89.7%	91.1%	82.3%	71.7%
Hourglass [23], 15ch	89.2%	91.1%	83.0%	73.1%
OpenPose [1], 6ch	88.8%	89.5%	73.9%	72.7%
提案手法, 32ch	**91.8%**	**94.0%**	**89.8%**	**82.3%**
25 MFLOPs				
HRNet [26], 8ch	88.6%	90.3%	78.6%	64.3%
Hourglass [23], 10ch	89.5%	90.0%	80.6%	69.1%
OpenPose [1], 4ch	88.0%	87.4%	79.5%	67.2%
提案手法, 24ch	**90.9%**	**93.5%**	**88.1%**	**79.1%**

節点有無の精度について，提案手法は既存研究の Hourglass と同等程度に，消費リソースを大幅に削減した時の精度低下を抑制できる．

表 3.3: 精度比較 (関節点有無)

手法	AP (頭部)	AP (首元)	AP (右手)	AP (左手)	mAP
100 MFLOPs					
HRNet [26], 16ch	**99.9%**	98.4%	**92.1%**	83.9%	93.6%
Hourglass [23], 20ch	**99.9%**	98.3%	91.8%	**84.7%**	**93.7%**
OpenPose [1], 8ch	99.9%	**98.7%**	89.4%	81.2%	92.3%
提案手法, 54ch	**99.9%**	98.4%	91.6%	84.2%	93.5%
50 MFLOPs					
HRNet [26], 12ch	**99.9%**	98.3%	**92.0%**	84.2%	**93.6%**
Hourglass [23], 15ch	99.9%	98.5%	91.6%	83.9%	93.5%
OpenPose [1], 6ch	99.9%	98.4%	89.6%	**86.1%**	93.5%
提案手法, 32ch	99.9%	**98.6%**	90.6%	84.5%	93.4%
25 MFLOPs					
HRNet [26], 8ch	99.9%	98.3%	88.5%	74.7%	90.4%
Hourglass [23], 10ch	**99.9%**	**98.4%**	**91.9%**	**83.1%**	**93.4%**
OpenPose [1], 4ch	99.9%	98.3%	90.9%	81.4%	92.6%
提案手法, 24ch	**99.9%**	98.2%	89.3%	81.7%	92.3%

■ 実験結果画像

提案手法と既存研究の姿勢推定結果画像を図 3.4 に示す．既存手法の関節点有無のしきい値は 0.5 とした．図 3.4 より，提案手法は既存研究よりも関節点座標の推定結果が正解座標に近く，未検出も少ないことがわかる．飲み物を把持している左手のように遮蔽があるような場合でも，提案手法では手の中心座標を正確に推定できる．

パニック動作時のヒートマップを図 3.5 に示す．ヒートマップは出力値が 0.5 の場合は赤色，0.0 は青色とした．図 3.5 より，既存研究の Hourglass では左手のヒートマップに出力があるが，関節点によりヒートマップの最大値が大きく異なるため，ヒートマップの最大値に対して同一のしきい値を用いると，未検出や誤検出が起きる．既存手法では，関節点座標のみを正解値として学習するが，提案手法では関節点座標と同時に関節点有無も学習する．そのため，提案手法ではしきい値を調整することなく，関節点有無を推定できる．

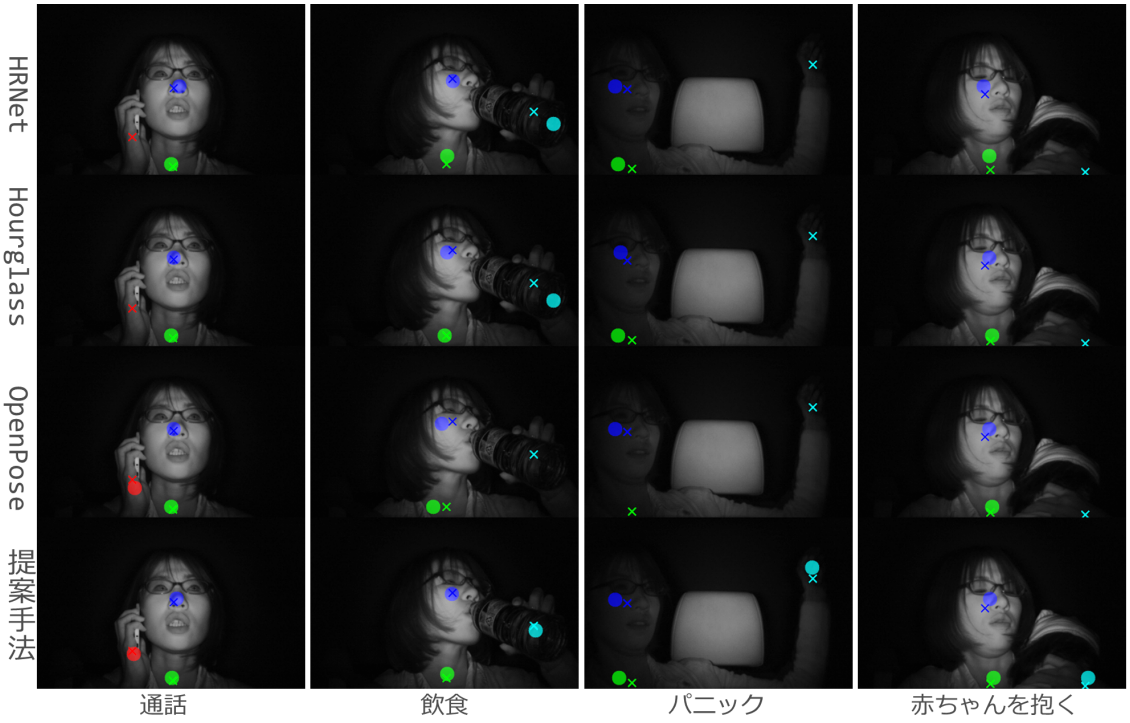

図 3.4: 姿勢推定結果．青色が頭部中心，緑色が首元，赤色が右手中心，水色が左手中心の座標を示す．○は推定結果，×は正解座標を示す．

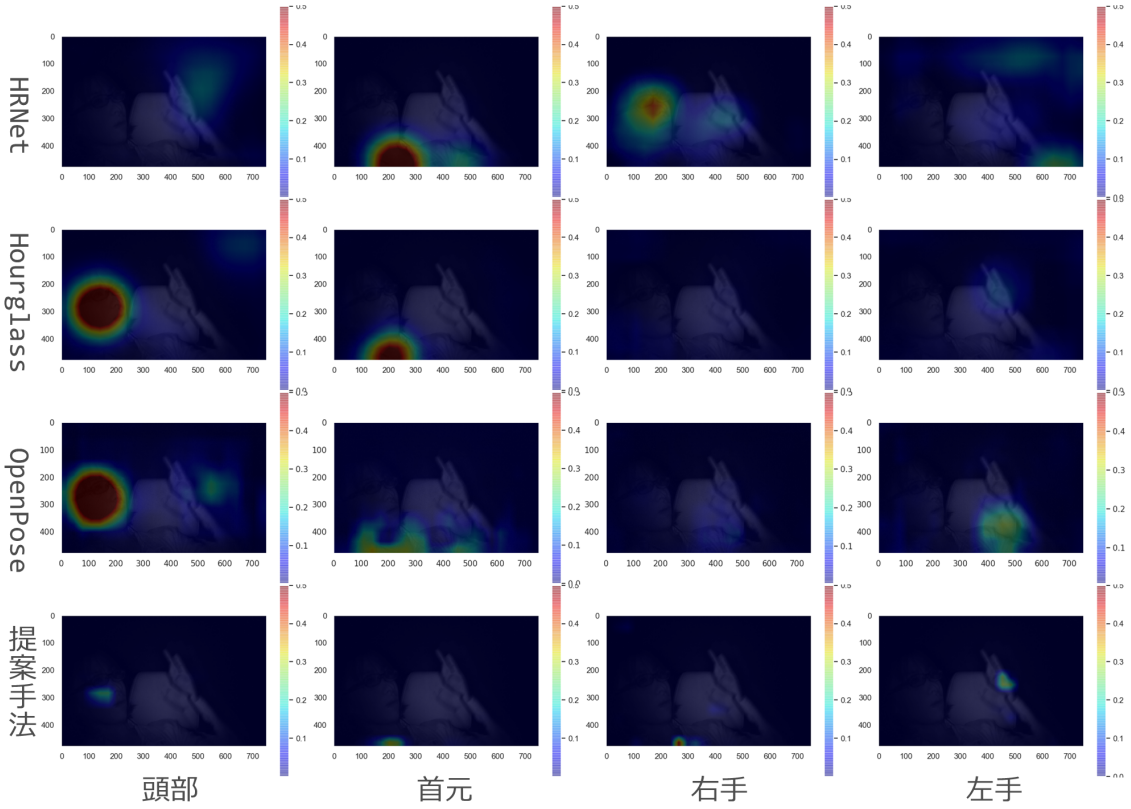

図 3.5: ヒートマップ (パニック)

3.3.4 Ablation Study

提案手法は ShuffleNet V2，Integral Regression，関節点有無の判定，の 3 つの要素から構成される．本実験では Ablation Study として，提案手法から各要素を除いた実験を行い，どの要素が演算量低下時の精度低下抑制に寄与するかを確認する．ShuffleNet V2 は通常の畳み込み層に置き換える．ただし，畳み込み層は ShuffleNet V2 に比べて演算量が多いため，出力チャンネル数を 20ch に減らして演算量が同程度になるよう調整する．Integral Regression は argmax に置き換え，ヒートマップ上で最大値を取る座標を関節点座標とする．関節点有無の判定は単純なしきい値処理に置き換え，推定した関節点座標のヒートマップの値がしきい値以上かどうかで関節点有無を判定する．

Ablation Study の関節点座標精度を表 3.4 に示す．上から提案手法 (Proposed)，提案手法から ShuffleNet V2 を除いたモデル (No-SNV2)，Integral Regression を除いたモデル (No-IR)，関節点有無の判定を除いたモデル (No-Det) の結果である．ShuffleNet V2 を除いたモデルでは提案手法に比べて 1% 程度精度が低下している．Integral Regression を除いたモデルでは，関節点座標の推定精度が大幅に低下している．関節点有無の判定は関節点座標の推定精度にはほとんど影響しない．

表 3.4: Ablation Study(関節点座標)

手法	PCK-6px (頭部)	PCK-6px (首元)	PCK-6px (右手)	PCK-6px (左手)
Proposed, 54ch	**91.9%**	94.5%	90.5%	**84.0%**
No-SNV2, 20ch	91.8%	94.4%	89.7%	81.3%
No-IR, 54ch	86.8%	82.4%	71.9%	60.9%
No-Det, 54ch	91.8%	**94.7%**	**90.7%**	83.6%

Ablation Study の関節点有無の精度を表 3.5 に示す．関節点有無については，ShuffleNet V2 と Integral Regression を除いた場合でも精度はあまり変わらない．一方，関節点有無の判定を除いたモデルでは精度が低下する．図 3.5 にて示す通り，既存研究の Hourglass では関節点によりヒートマップの最大値が大きく異なるため，適切なしきい値を設定することが難しい．一方，関節点有無の判定ではしきい値を調整することなく，関節点有無を判定できるため，精度に影響を与えたと考えられる．

表 3.5: Ablation Study(関節点有無)

手法	AP (頭部)	AP (首元)	AP (右手)	AP (左手)	mAP
Proposed, 54ch	99.9%	98.4%	91.6%	84.2%	93.5%
No-SNV2, 20ch	99.9%	98.3%	90.8%	82.7%	92.9%
No-IR, 54ch	**99.9%**	**98.5%**	**92.5%**	**85.9%**	**94.2%**
No-Det, 54ch	99.9%	91.6%	90.2%	84.8%	91.6%

3.3.5 動作パターン毎の評価

動作パターン毎の評価結果を表 3.6，表 3.7 に示す．動作によっては部位が全く映らないことがあるため，関節点有無の評価指標は正解率 (正解フレーム数 / 全フレーム数) を用いる．関節点座標では，多くの動作が高精度であるが，振り返り動作の両手，読書動作の左手，突っ伏し動作の頭部及び両手については精度が低い．これらの動作は手が映るシーンが少ないため，精度が低下したと考えられる．しかし，これらの動作は，手の座標よりも頭部座標が行動の推定に重要だと考えられるため，ドライバの異常検知には大きく影響しない．一方，通話，カメラ撮影，飲食といった動作は物体を把持することが行動の推定に重要であるため，手の位置の推定精度が重要となる．これらの動作パターンでは，多くのシーンで手が映るため，手の推定精度が 90%以上と精度が高くなったと考えられる．また，突っ伏し動作では，ドライバがハンドル方向に突っ伏す．そのため，頭部が大きく映り，頭部の正確な中心座標は人目でも判別が難しい．突っ伏し動作は，画面下部に頭部位置が移動したことにより，ドライバ異常の検知が可能であるため，大まかな頭部中心が推定できれば問題ない．

関節点有無については，スマホ操作の首元と読書時の首元の精度が低い．スマホ操作と読書時の見え方は同じであり，どちらの動作もドライバが下を向いている．そのため，首元の多くが隠れてしまい，誤判定が多くなったと考えられる．しかし，どちらの動作も頭部位置に特徴があるため，首元の有無はドライバ異常の検知には影響しない．

表 3.6: 行動パターン別評価 (関節点座標)

手法	PCK-6px(頭部)	PCK-6px(首元)	PCK-6px(右手)	PCK-6px(左手)
前方注視	99.9%	100.0%	96.6%	96.6%
脇見	99.2%	99.3%	94.7%	93.8%
もたれ	99.3%	96.3%	97.7%	92.6%
ストレッチ	94.5%	92.1%	75.9%	75.9%
振り返り	83.6%	81.2%	54.4%	32.5%
眠気	99.1%	99.3%	86.7%	99.3%
スマホ操作	97.5%	99.2%	100.0%	100.0%
通話	99.9%	99.8%	98.7%	97.6%
カメラ撮影	92.7%	90.8%	94.4%	93.6%
読書	97.8%	98.4%	100.0%	0.0%
飲食	98.9%	97.7%	93.3%	91.5%
着替え	85.8%	88.2%	74.5%	78.3%
突っ伏し	43.9%	98.0%	44.2%	49.4%
パニック	89.1%	85.4%	83.1%	80.8%
居眠り	98.6%	96.3%	88.9%	100.0%
赤ちゃんを抱く	92.7%	93.6%	75.3%	73.4%

表 3.7: 行動パターン別評価 (関節点有無)

手法	正解率 (頭部)	正解率 (首元)	正解率 (右手)	正解率 (左手)
前方注視	100.0%	96.3%	99.9%	99.9%
脇見	100.0%	91.4%	99.5%	99.5%
もたれ	98.1%	85.9%	91.5%	99.9%
ストレッチ	99.3%	87.6%	82.4%	82.3%
振り返り	95.9%	86.7%	99.5%	98.6%
眠気	100.0%	87.3%	99.5%	99.8%
スマホ操作	100.0%	61.3%	99.9%	99.9%
通話	99.9%	85.7%	91.4%	89.1%
カメラ撮影	99.1%	82.4%	89.0%	89.1%
読書	99.9%	63.8%	99.6%	99.7%
飲食	99.9%	81.3%	88.5%	95.0%
着替え	96.3%	82.2%	91.4%	91.5%
突っ伏し	85.1%	97.5%	91.1%	92.2%
パニック	96.1%	82.0%	90.3%	90.7%
居眠り	99.2%	85.9%	99.9%	99.9%
赤ちゃんを抱く	98.8%	70.1%	97.2%	95.0%

3.4 まとめ

本研究では，DMS のためのドライバ姿勢推定手法を提案した．提案手法では，ShuffleNet V2 と Integral Regression を用いることで高速で省メモリな姿勢推定を実現できることを示した．DMS 特有の関節点が見えないシーンに対応するため，関節点有無の判定を提案した．また，ドライビングシミュレータで撮影した評価データを用いて，提案手法が既存手法に比べて，モデルサイズ低減時の精度低下を抑制できることを示した．本研究ではドライビングシミュレータのデータを用いて評価を行ったが，更なる実用性向上のため，今後は実写映像での評価を実施していく必要がある．本研究では精度低下の抑制に注力したが，今後は実用性向上のため更なる精度向上に取り組む．

第4章

ドライバ姿勢と動作のマルチタスク学習による高速かつ省メモリなドライバ動作認識

保有台数 1 万台当たりの人身事故発生件数を車種別にみると，バスやタクシーなどの事業用乗用車が最も多い [38]．事業用乗用車の事故を減らすため，営業車 5 台以上を保有する事業者は安全運転管理者を選任する義務が生じる．営業車が増えるほど安全運転管理者の管理負荷は増大するため，運転動作評価用ドライバモニタリングシステム（DMS）を活用し，管理負荷を軽減することが期待されている．

従来の実用化されている DMS ではドライバの顔向きや目開閉度からドライバが運転に集中しているかどうかを推定しているが，運転動作評価用 DMS では，ドライバの飲食など，運転中にドライバが取りうる多様な動作を認識することでより詳細な評価を行うことが求められる．本研究では，運転動作評価用 DMS のためのドライバ動作認識を提案する．

DMS では自動車の組込機器での動作が求められるため，サーバやデスクトップ PC と比べて，消費リソースが限られる．近年提案されている Deep Convolutional Neural Network (CNN) を用いた人物動作認識は高精度であるが，演算量などの消費リソースが大きいため，DMS にそのまま搭載することが難しい．本研究では，軽量な姿勢推定モデルをベースとした高精度で演算量の少ないドライバ動作認識を提案する．

CNN を用いた動作認識で用いられる Kinetics[39, 40] などのデータセットは，様々な種類の動作が含まれているが，前方注視や余所見などの運転動作の評価に役立つ動作は含まれていない．本研究では，ドライバの運転への復帰時間を基に，運転中に取りうるドライバ動作を 7 つの動作として定義し，それらの動作を含んだデータセットを構築する．また，そのデータセットを用いて提案手法の評価実験を行う．

CNN を用いた従来の人物動作認識の多くは，入力動画から直接人物の動作を推定する．これらの手法は高精度であるが，本研究ではドライバの動作だけでなく，ドライバの姿勢も同時に学習するマルチタスク学習を行うことで，演算量が制限された条件で従来の人物動作認識よりも高い精度を達成できることを示す．本研究では，ドライバ動作だけでなく，ドライバの関節がどこに映っているか（関節点座標），ドライバの関節が画像中に映っているかどうか（関節点有無），ドライバの関節がどのような状態にあるか（関節点状態），の 3 つのドライバ姿勢を同時に学習するマルチタスク学習を提案する．

4.1 関連研究

4.1.1 動作認識

人物動作認識はコンピュータビジョンの分野で活発に研究されており，近年は CNN を用いた高精度な手法が数多く提案されている．

Karpathy らは，動画に対して時系列データを結合する 3 つの Fusion model (Early-fusion，Late-fusion，Slow-fusion) を提案した [41]．Early-fusion では，複数フレームの画像を深さ方向に結合し，複数チャンネルの入力画像として扱う．Late-fusion では，複数フレームの画像を別々に CNN に与え，それらの出力である特徴マップを結合する．Slow-fusion は，Early-fusion と Late-fusion の中間に相当し，入力から出力に向かって，徐々に時間方向の情報を結合する．

画像から特徴抽出を行う CNN と時系列データを扱う Recurrent Neural Network (RNN) を組み合わせた動作認識が提案されている．Donahue らは，Long Short Term Memory (LSTM)[42] を CNN と組み合わせた Long-term Recurrent Convolutional Networks (LRCN) を提案した [43]．LSTM は長期的な時間依存を学習できる RNN であり，CNN と組み合わせることで，長時間の動作を認識できる．

Simonyan らは，時間情報と空間情報を並列に処理する 2-stream CNN を提案した [44]．2-stream CNN では，時間情報のオプティカルフロー画像と，空間情報の RGB カラー画像が入力となる．時間情報と空間情報を並列に扱うことで，高精度な動作認識を実現した．

空間方向に対して畳み込みを行う CNN を時間方向に拡張した 3D-CNN[45, 46] を動作認識に適用した手法が提案されている．Feichtenhofer らは，3D の畳み込み層で，時間方向のストライドを小さくした Slow Pathway とストライドを大きくした Fast Pathway を組み合わせた SlowFast Networks を提案した [47]．SlowFast Networks は，異なるストライドを利用することで，様々な時間解像度の特徴を抽出できる．

4.1.2 マルチタスク学習

4.1.1 節で言及した動作認識モデルは人物動作を高精度に認識できるが，マルチタスク学習を用いて人物姿勢も同時に学習することで，更なる高精度化が期待できる．マルチタスク学習 [48] は，複数のタスクを学習することで，各タスクの精度を向上させる手法である．[48] では，複数のタスクを同時に学習することで，各タスクで共通して有効な特徴が選ばれやすくなり，精度が向上すると言及されている．人物動作認識でも同様に，人物姿勢と共通した有用な特徴を抽出することで精度が向上することが期待できる．

Gkioxari らは物体検出やセマンティックセグメンテーションなどで使用される R-CNN[49] を用いて，姿勢推定と動作認識のマルチタスク学習を提案した [2]．[2] では，人物動作だけでなく，人物の関節点座標も学習することで，動作認識の精度が向上することを示した．

本研究では演算量が制限された条件でドライバ動作認識の精度を向上させるため，ドライバの関節点座標だけでなく，関節点有無や関節点状態も同時に学習するマルチタスク学習を提案する．

4.1.3　ドライバ動作認識用データセット

動作認識用の評価データセットとして，Kinetics[39][40]，Charades[50]，AVA dataset[51] がある．動作認識データセットの画像例を図 4.1 に示す．これらのデータセットは，可視光カメラで撮影され

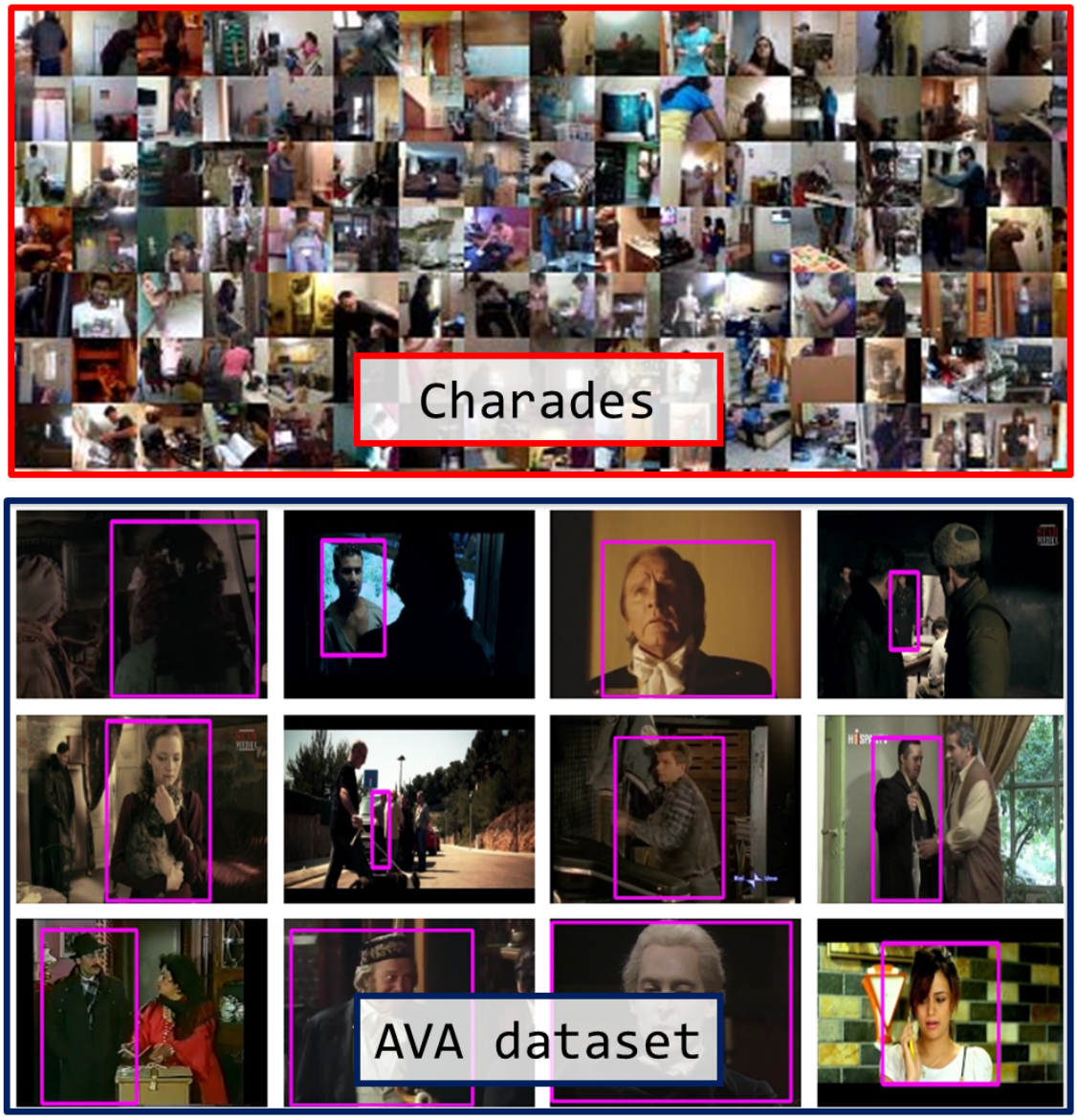

図 4.1: 動作認識データセットの画像例．

た映像である．DMS では昼だけでなく，夜間もドライバを撮影する必要があるため，可視光カメラではなく，近赤外線カメラを用いることが多い．また，データセットに含まれる動作も DMS での利用を想定していない．そのため，本研究では，DMS での利用を想定した近赤外カメラで撮影したドライバ動作認識用に構築したデータセットの例を示す．

4.2　提案手法

本研究では，ドライバの姿勢と動作のマルチタスク学習を用いた高精度かつ演算量の少ないドライバ動作認識を提案する．提案手法のネットワークモデルの概要を図 4.2 に示す．

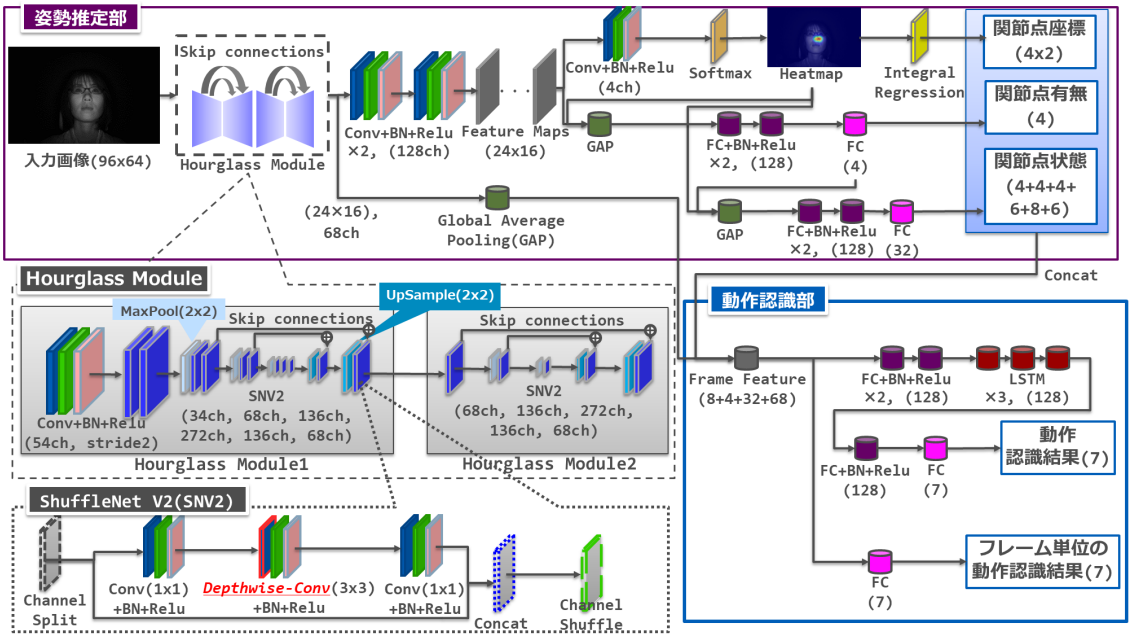

図 4.2: ネットワークモデル全体 (提案手法, 入力画像 96×64，34ch) : () 内は畳み込み層では出力チャンネル数，FC 層では出力ノード数，Feature Map ではマップの解像度を示す．SNV2 の Conv(1×1) はカーネルサイズが 1×1 の畳み込みを示す．各出力結果では，出力結果の次元数を示す．FC は全結合層，BN は Batch Normalization 層を示す．

提案手法のネットワークモデルは，姿勢推定部と動作認識部に分かれる．姿勢推定部では CNN を用いて，ドライバの関節点がどこにあるか（関節点座標），関節点が画像中に映っているか（関節点有無），関節点がどのような状態にあるか（関節点状態），の 3 つのドライバ姿勢を出力する．動作認識部では姿勢推定部が出力したドライバ姿勢と特徴マップを入力として，RNN を用いてドライバ動作を認識する．

4.2.1　運転動作の定義

Zhao らが提案したドライバ動作認識のデータセット Southeast University Driving-posture Dataset (SEU dataset)[3] では，電話，食事，ブレーキ，運転，スマートフォンの操作，タバコの 6 動作が含まれる．SEU dataset の例を図 4.3 に示す．

SEU dataset の動作は，比較的容易に正常な運転に戻れる動作である．しかし，運転時は居眠りや発作など，意識を失うような深刻な状態が起こりうるため，これらの動作の認識が深刻な事故の抑

図 4.3: SEU dataset[3] の画像例.

制につながる．また，SEU dataset は手動運転時のドライバ動作を対象としているが，今後は自動運転車の急速な普及が想定されており [14]，自動運転中のドライバ動作の認識も重要な課題である．

本研究では，SEU dataset で取り扱っている軽度な不注意状態に加えて，意識を失うような深刻な状態，さらに自動運転中にドライバが取りうる動作も検討の対象とする．そのため，本研究では安全性に問題がない状態から運転復帰が困難な状態までを幅広くカバーする 7 種類のドライバ動作（前方注視，余所見，眠気，物体把持，下向き，意識不明，パニック）を代表的な動作として取り上げる．

正常な運転状態は，前方を注視しており，手はハンドルを握っているか，何も持っていない状態である．本研究では，そのような正常な運転状態を示す動作として，“ 前方注視 ” を認識対象とする．SEU dataset のブレーキと運転の 2 動作は，本研究の ” 前方注視 ” に該当する．

正常な運転状態への復帰に少し時間がかかる危険な状態は，余所見や短時間の閉眼など前方を向いていない状態や，スマートフォンや飲食物など手で物体を掴んでいる状態である．本研究では，そのような危険な状態として，横や後ろを向いている“ 余所見 ”，うつらうつらしている“ 眠気 ”，手で物体を持っている“ 物体把持 ”，スマートフォンや本などを見ている“ 下向き ” の 4 つの動作を認識対象とする．SEU dataset の電話，食事，スマートフォンの操作，タバコの 4 動作は，本研究の“ 物体把持 ” に含まれる．自動運転時にはカメラなど多様な物体を持っている可能性が高いため，本研究では特定の物体を対象としない“ 物体把持 ” を認識対象動作とする．

正常な運転状態への復帰が困難な状態は意識を失っていたり，パニックになっているような状態である．本研究では，そのような復帰が困難な状態として，眠っていたり，急病で意識を失っていたりする“意識不明”，蜂が入ってきており運転に集中できないような状態である“パニック”の 2 つの動作を認識対象とする．現在導入が進められている自動運転車では，一般的には起こりにくい居眠りなどの運転状態への復帰が困難な動作が起こる可能性が高い．また，完全な自動運転車の開発は難しく，今後も緊急時には手動運転に切り替わるような半自動運転が主流になると考えられるため，上記のような運転動作の評価は必要となると考え，本研究では動作認識の対象とする．

以上より，本研究では，前方注視，余所見，眠気，物体把持，下向き，意識不明，パニック，の 7 動作を認識するネットワークモデルを提案する．

4.2.2 姿勢推定部

姿勢推定部では関節点座標，関節点有無，関節点状態の 3 つのドライバ姿勢を出力する．本研究で用いる姿勢推定部を図 4.2 の上部に示す．姿勢推定部には，人物姿勢推定用ネットワークモデルである Hourglass Module[23] を用いる．Hourglass Module は高解像度から低解像度に情報を圧縮していくダウンサンプリング処理を行った後，低解像度から高解像度に変換していくアップサンプリング処理を行うことで，様々な解像度の人物姿勢を推定するネットワークモデルである．本研究では Hourglass Module に，ShuffleNet V2[34] と Integral Regression[52] を導入することで，演算量の少ない姿勢推定を提案する．

■ ShuffleNet V2

本研究では通常の畳み込み層の代わりに Ma らが提案した ShuffleNet V2[34] を用いて，畳み込みの演算量を減らす．ShuffleNet V2 の構成を図 4.2 の最下部に示す．ShuffleNet V2 ではチャンネルを分割したのち，一方のみに畳み込みを行う．分割したチャンネルを結合する際に，チャンネルの順序を入れ替えることで畳み込みの回数を減らしつつ，全てのチャンネルに対して畳み込みを行う．ShuffleNet V2 では 1×1 のフィルタを用いた畳み込み，Depth-wise の畳み込み，再び 1×1 のフィルタを用いた畳み込みを順に行う．Depth-wise の畳み込みはチャンネル方向の畳み込みを行わず，空間方向に対してのみ畳み込みを行うため，通常の畳み込みよりも演算量が少ない．

■ Integral Regression

多くの人物姿勢推定では，ヒートマップを出力し，ヒートマップ上の最大値を取る座標を関節点座標とする．ヒートマップの解像度を小さくすると演算量が減るが，関節点座標の量子化誤差が増える．Sun らはヒートマップの重心位置を関節点座標として出力する Integral Regression[52] を提案した．Integral Regression は，サブピクセル単位で関節点座標を推定するため，ヒートマップの解像度を小さくした際の量子化誤差を抑制できる．本研究では Integral Regression を用いて，量子化誤差

を抑制しつつ，ヒートマップの解像度を小さくすることで，演算量を減らす．従来のヒートマップの最大値を用いた関節点座標の推定は，式 (4.1) となる．H_k は k 番目の関節点のヒートマップ，p はヒートマップ上の座標，I_k と J_k は k 番目の関節点の座標を示す．Integral Regression を用いたヒートマップの重心位置の推定は，式 (4.2) となる．Ω は座標の集合を示す．Integral Regression で使用されるヒートマップ $\hat{H}_k$ は，式 (4.3) を用いて，全ての要素が非負であり，全ての要素を足すと 1 になるように正規化される．

$$I_k = arg\max_p H_k(p) \tag{4.1}$$

$$J_k = \int_{p\in\Omega} p \cdot \hat{H}_k\ (p) \tag{4.2}$$

$$\hat{H}_k(p) = \frac{e^{H_k(p)}}{\int_{q\in\Omega} e^{H_k(q)}} \tag{4.3}$$

■ 関節点座標及び関節点有無の推定

本研究の姿勢推定部では，頭部，首元，右手，左手の 4 つの関節点座標を出力する．各関節点の座標は 2 次元，関節点座標の次元数は 8 となる．また，DMS ではカメラとドライバ間の距離が狭く，関節点が画像中に映らないことが頻繁にあるため，提案手法では関節点が画像中に映るかどうか（関節点有無）も推定する．関節点有無はドライバが急病で倒れ，頭部が画面中に映らないような動作の認識に役立つ．関節点有無は画像中に映る場合に 1，映らない場合に 0 となる 2 値の出力とする．関節点有無の次元数は 4 となる．

■ 関節点状態の推定

高精度なドライバ動作認識には，ドライバの関節点座標や関節点有無に加えて，各関節点の詳細な状態が役立つ．飲み物を顔に近づける動作と，手で顔を掻く動作ではいずれの動作も頭部付近に手があるため，関節点座標と関節点有無だけでは区別が難しい．それらの動作を区別するためには，手で物体を把持しているかどうかといった関節点の詳細な状態を把握することが必要である．本研究では，[15] で示されるドライバ観測指標を基に，ドライバ動作認識に必要なドライバ状態として，右手状態，左手状態，目状態，顔向きのピッチ，ヨー，ロールの 6 つの関節点状態を定義し，それらを推定するモデルを提案する．各関節点状態のクラスを表 4.1 に示す．右手・左手状態は上から順に，手が画面に映っていない，物体を持っている，何も持っていない，ハンドルを握っている，の 4 つとする．目状態は，目が画面に映っていない，前方を注視している，余所見をしている，目を閉じている，の 4 つとする．顔向きピッチは，顔が映っていない，下を向いている，少し下を向いている，前を向いている，少し上を向いている，上を向いている，の 6 つとする．顔向きヨーは，顔が映っていない，右後ろを向いている，右を向いている，右前を向いている，正面を向いている，左前を向いている，左を向いている，左後ろを向いている，の 8 つとする．顔向きロールは，顔が映って

表 4.1: 関節点状態

右手/左手	目	顔向きピッチ	顔向きヨー	顔向きロール
• Unknown • Hand-on • Hand-off • Handle	• Unknown • Open • Side • Close	• Unknown • Down • DownFront • Front • UpFront • Up	• Unknown • RightBack • Right • RightFront • Front • LeftFront • Left • LeftBack	• Unknown • -90 ° • -45 ° • 0 ° • 45 ° • 90 °

いない，-90°， -45°， 0°， 45°， 90°の 6 つとする．顔向きロールの角度は時計回りの方向が正の値，反時計回りの方向が負の値とする．

4.2.3 動作認識部

提案手法では，動作認識部のネットワークモデルとして，LSTM[42] を用いた RNN を提案する．動作認識部を図 4.2 の右下部に示す．動作認識部では本研究で定義した 7 動作の認識結果を出力する．動作認識部の入力は，姿勢推定部が出力する関節点座標，関節点有無，関節点状態の 3 つのドライバ姿勢と，Hourglass Module が出力する特徴マップとなる．関節点座標，関節点有無，関節点状態はそれぞれ，8，4，32 次元の特徴ベクトルとなる．また，Hourglass Module から出力される特徴マップは Global Average Pooling(GAP) を経て，68 次元の特徴ベクトルとなる．それらを結合した 112 次元の特徴ベクトルが，動作認識部の入力となる．動作認識部は全結合層により特徴抽出を行った後，LSTM 層で時系列の情報を処理する．最後に全結合層を用いて，7 動作の認識結果を出力する．また，学習を安定させるため，フレーム単位での動作認識結果も出力する．姿勢推定部で得られた特徴ベクトルを入力として，1 層の全結合層を用いてフレーム単位の動作認識結果を出力する．フレーム単位の動作認識結果は，学習にのみ使用され，テスト時の最終的な出力結果は LSTM の出力結果とする．

4.2.4 学習

提案手法では入力画像 1 枚に対して，姿勢推定部の関節点座標，関節点有無，関節点状態の 3 つのドライバ姿勢と，動作認識部のドライバ動作の計 4 つの損失を計算し，ネットワークモデルを学習する．

関節点座標を示すヒートマップの損失関数は L2-loss を用いる．ヒートマップの学習に用いる Ground Truth (GT) は正解の関節点座標から σ が 2 ピクセルの正規分布に従って作成したヒートマップを用いる．ヒートマップの損失 L_H は，式 (4.4) となる．H_k は k 番目の関節点のヒートマップ，p はヒートマップ上の座標，H_k^* はヒートマップの GT，K は関節点の数となる．

$$L_H = \sum_{k=1}^{K} \sum_{p} \|H_k(p) - H_k^*(p)\|_2^2 \tag{4.4}$$

提案手法では微分可能な Integral Regression を用いるため，関節点座標に対しても損失を計算する．関節点座標の損失関数 L_C は，L1-loss を用いて，式 (4.5) となる．C は関節点座標，C^* は関節点座標の GT，j は j 番目の関節点を示す．

$$L_C = \sum_{j=1}^{J} \|C(j) - C^*(j)\| \tag{4.5}$$

関節点有無の損失関数 L_D は，式 (4.8) となる．クラス間のデータの偏りに対応するため，Focal loss[35] を用いる．x は関節点有無の 2 値出力，t は関節点有無の GT，CE はクロスエントロピー，γ は損失を調整するパラメータを示す．本研究では，γ を 2 とする．t は関節点が画像中に映る場合に 1，画像に映らない場合に 0 とする．

$$CE(x_t) = -\log(x_t) \tag{4.6}$$

$$x_t = \begin{cases} x & t = 1 \\ 1 - x & otherwise \end{cases} \tag{4.7}$$

$$L_D = -(1 - x_t)^{\gamma} \log(x_t) \tag{4.8}$$

関節点状態の損失関数 L_S も，クラス間のデータの偏りに対応するため，Focal loss を用いた式 (4.8) を用いる．

動作認識結果はフレーム単位の認識結果と，LSTM による時間的な成分を考慮した認識結果の 2 つとなる．フレーム単位の動作認識結果の損失関数 L_F，時間的な成分を考慮した認識結果の損失関数 L_A は式 (4.10) となる．いずれの損失関数も Softmax Cross Entropy を用いる．Softmax は式 (4.11) を用いて算出される．

(4.9)

$$L_A = \sum_{c=1}^{C} t_c \dot{\log}(y_c) \tag{4.10}$$

$$y_c = \frac{e^{z_c}}{\sum_{d=1}^{C} e^{z_d}} \tag{4.11}$$

提案手法のネットワークモデルは，これらの損失をあわせた式 (4.12) を用いて学習する．w は各損失を調整するための重みである．本研究では，全ての損失を同等に扱うため，重み w を全て 1.0 とする．

$$\begin{aligned} L = w_H L_H + w_C L_C + w_D L_D \\ + w_S L_S + w_F L_F + w_A L_A \end{aligned} \tag{4.12}$$

4.3 実験

提案手法の有効性を確認するため，既存手法と精度を比較する．DMS では一般的な人物動作認識よりも，消費リソースが限られるため，提案手法と既存手法の演算量が同程度になるようパラメータ数を調整したネットワークモデルを用いる．また，本研究では Ablation Study として，提案手法の 3 つのドライバ姿勢を除いたモデルで精度を比較する．最後に提案手法で推定したドライバ姿勢と動作を示し，ドライバ姿勢が動作認識の誤認識の要因を解析するのに役立つことを示す．

4.3.1 実験データ

本研究では運転中にドライバが取りうる 7 つの動作を近赤外線カメラで撮影した．100 人の被験者を撮影し，50 人分を学習用，残り 50 人分を評価用として用いる．7 つの動作は，前方注視，余所見，眠気，物体把持，下向き，意識不明，パニック，である．撮影した動画のフレームレートは 10fps，解像度は 752×480 である．演算量を減らすため，撮影した動画を 96×64 に縮小し，データセットを構築した．1 動画当たりの撮影時間は 30 秒程度であり，各動作を被験者一人当たり 4〜12 回程度撮影した．動画は合計で 6353 本となる．

提案手法はドライバ姿勢と動作のマルチタスク学習を行うため，ドライバ姿勢用の学習及び評価用データも構築した．ドライバ動作用の撮影動画と同じ映像を使用し，頭部，首元，右手，左手の 4 つの関節点座標と関節点有無のアノテーションを行った．更に，関節点状態として，右手状態，左手状態，目状態，顔向きのピッチ，ヨー，ロールの 6 つの状態のアノテーションを行った．

4.3.2 評価実験のパラメータ

■ 入力画像，出力ヒートマップ

入力画像は 96×64 に縮小したのち，Global Contrast Normalization[36] による正規化を行う．提案手法の姿勢推定部で出力するヒートマップのサイズは 24×16 とする．本実験で用いる 3D-CNN と

Fusion-model では，複数フレームの画像が入力として必要なため，全て 32 フレームを入力として用いる．また，2-stream CNN のオプティカルフロー画像は 31 フレームを入力として用いる．

■ ネットワーク

本研究では，比較対象の既存手法を，CNN と LSTM を用いた手法として LRCN[43]，CNN と時間方向の結合を用いた手法として Fusion model(Early-fusion，Late-fusion，Slow-fusion)[41]，3D-CNN として SlowFast[47]，2-stream CNN として [44] とする．本研究で組込を想定している車載情報端末向け統合 SoC には CPU に Arm Cortex-A9 533MHz を搭載している．該当 CPU の演算性能が 8.5GFLOPs となるため，その他画像処理や OS 等の基本機能を考慮し，全演算性能の 20%を提案手法のリソースとして割くことができると仮定し，フレームレートが 30FPS の場合，$8500 \times 0.2/30 = 56$MFLOPs 以下にする必要がある．従って，各手法は演算量が 50MFLOPs，25MFLOPs 程度になるようパラメータ数を調整する．各モデルの消費リソースを表 4.2 に示す．

先行研究で提案されている LRCN や Fusion model，2-stream CNN では，学習を安定させるため，カラー画像の ImageNet[53] を用いて事前学習を行っている．しかし，本実験では近赤外カメラで撮影したグレースケール画像のドライバ動作認識を対象としているため，ImageNet による事前学習は行わない．本実験では，学習を安定させるため，Batch Normalization 層 [37] を導入する．

LRCN，Fusion model，2-stream CNN は演算量を調整するため，3 層の畳み込み層を用いる．LRCN は畳み込み層の出力チャンネル数を 14，30，48 に減らした 50MFLOPs のモデルと，12，20，30 に減らした 25MFLOPs のモデルを用いる．Early-fusion モデルは畳み込み層の出力チャンネル数を 16，16，32 に減らした 50MFLOPs のモデルと，8，14，18 に減らした 25MFLOPs のモデルを用いる．Late-fusion モデルは畳み込み層の出力チャンネル数を 12，20，32 に減らした 50MFLOPs のモデルと，8，16，18 に減らした 25MFLOPs のモデルを用いる．Slow-fusion モデルは畳み込み層の出力チャンネル数を 8，12，20 に減らした 50MFLOPs のモデルと，4，10，16 に減らした 25MFLOPs のモデルを用いる．2-stream CNN は画像を入力とする CNN とオプティカルフロー画像を入力とする CNN のいずれも畳み込み層の出力チャンネル数を 8，12，22 に減らした 50MFLOPs のモデルと，4，8，18 に減らした 25MFLOPs のモデルを用いる．2 層目以降の畳み込み層では，畳み込み層の後に 2×2 の Average Pooling 層を用いる．また，これらのモデルでは 3 層の畳み込み層の後に，2 層の全結合層をつなげる．全結合層の出力ノード数は 256，7 とする．SlowFast は [47] で提案されているモデルと同様に，4 つの residual stage を用いる．各 stage の層数は 3，4，6，3 とする．また，本実験では入力画像の解像度が小さいため，畳み込みのカーネルサイズを 3 以下とする．各ステージの出力チャンネル数は 4，8，16，32 の 50MFLOPs のモデルと，1，2，4，8 の 25MFLOPs のモデルを用いる．SlowFast は大きな stride や，1×1 のカーネルを用いた畳み込み，下位層に Global Average Pooling があるため，層数や出力チャンネル数が比較的多い場合でも，消費リソースが少ない．提案手法の 50MFLOPs モデルのパラメータは図 4.2 で示す通りである．25MFLOPs で用いるモデルでは，Hourglass Module の ShuffleNet V2 の最初の出力チャンネル数を 34ch から 22ch に減らし，以降の層のチャンネル数も 44ch，88ch，176ch に減らす．

表 4.2: ネットワークの演算量及びパラメータ数

手法	MFLOPs	パラメータ数 [M]
50 MFLOPs		
LRCN[43] 14-30-48ch	49.5	5.73
Early-fusion[41] 16-16-32ch	52.7	3.16
Late-fusion[41] 12-20-32ch	51.9	6.30
Slow-fusion[41] 8-12-20ch	51.1	1.98
SlowFast[47] 4-8-16-32ch	55.9	0.05
2-stream[44] 8-12-22ch	50.1	4.34
提案手法, 34ch	47.20	0.98
25 MFLOPs		
LRCN[43] 12-20-30ch	25.7	3.62
Early-fusion[41] 8-14-18ch	25.6	1.78
Late-fusion[41] 8-16-18ch	26.5	3.55
Slow-fusion[41] 4-10-16ch	25.5	1.58
SlowFast[47] 1-2-4-8ch	28.1	0.02
2-stream[44] 4-8-18ch	25.0	3.55
提案手法, 22ch	25.3	0.75

■ ハイパーパラメータ

学習率は 0.001 とし，300iteration までに徐々に 0.004 まで上昇させる．その後 5000iteration 毎に学習率を半減させ，30000iteration で学習終了とする．学習率を徐々に上昇させた後に，減少させる方法は Goyal らの Gradual Warmup[54] を参考にした．重み減衰は 0.0001 とし，重みのクリッピングは 5.0 とする．また，バッチサイズは 32 とし，4 つの GPU を用いて学習する．学習時には RNN の状態を 1iteration 毎に 25%の確率でリセットする．

■ 学習・評価方法の詳細

各動作の動画数が同程度になるように over sampling を行う．また，Data Augmentation として，下記 4 つの処理を実施する．

- ランダムにデータ長の調整（動画のフレーム数を 1〜500 フレームの範囲でランダムにカット）
- ランダムに平行移動（画像の高さ・幅の ±25%）
- ランダムに拡大縮小（0.8〜1.75 倍）
- 50%の確率で左右反転

本実験では動作認識の評価指標として，正解率 (Accuracy) を用いる．正解率は正解したフレーム数を合計フレーム数で除算した値となる．また，提案手法では関節点座標，関節点有無，関節点状態を出力するため，それらも評価する．関節点座標の評価指標は Probability of Correct Keypoints (PCK) を用いる．本実験では 96×64 の入力画像に対して，PCK のしきい値を 6 ピクセルとする．関節点有無の評価指標は，mean Average Precision (mAP) を用いる．関節点状態はクラス数に偏りがあるため，評価指標に正解率のマクロ平均 (マクロ正解率) を用いる．マクロ正解率はクラス毎に正解率を算出したのち，全クラスの正解率を平均した値となる．

4.3.3 精度比較

提案手法と既存手法について，動作認識の精度を比較した結果を表 4.3 に示す．また，提案手法の各ドライバ姿勢の精度を表 4.4 に示す．50MFLOPs，25MFLOPs のいずれのモデルも，既存手法よりも提案手法の方が精度が高い．また，25MFLOPs のモデルでは，既存手法で最も精度の高い SlowFast よりも提案手法の精度が 5%以上高い．従って，提案手法は，演算量を大幅に削減した時のドライバ動作認識の精度低下を抑制できる．2-stream CNN は入力画像の解像度が小さく，動きの特徴を上手くとらえられなかったため，その他の手法よりも低い精度になったと考えられる．

4.3.4 Ablation Study

本実験では Ablation Study として，提案手法から 3 つのドライバ姿勢を除いた実験を行い，各ドライバ姿勢が動作認識精度にどの程度寄与しているかを確認する．表 4.5 に Ablation Study の実験結果を示す．提案手法と，動作認識と関節点座標の推定を行うモデル (Only-Position)，動作認識と関節点有無の推定を行うモデル (Only-Detection)，動作認識と関節点状態の推定を行うモデル (Only-State)，提案手法から姿勢推定部（関節点座標，関節点有無，関節点状態の全て）を除いたモデル (No-Pose) を比較する．各モデルは演算量が 50MFLOPs，25MFLOPs 程度になるようパラメータ数を調整する．No-Pose では，動作認識部の入力は Hourglass Module が出力した特徴マップのみとなる．また，No-Pose では関節点座標，関節点有無，関節点状態を出力するために必要な畳み込み層や全結合層などを取り除く．

提案手法から姿勢推定部を除いたモデルでは，精度が 4%以上低下する．従って，ドライバの姿勢と動作のマルチタスク学習は，演算量を減らした際のドライバ動作認識の精度低下を抑制する．提案手法を除いた比較では，関節点状態の推定を行う Only-State が最も高精度である．ドライバ動作

表 4.3: 精度比較 (動作推定)

手法	正解率
50 MFLOPs	
LRCN[43],14-30-48ch	68.3%
Early-fusion[41],16-16-32ch	70.6%
Late-fusion[41],12-20-32ch	69.1%
Slow-fusion[41],8-12-20ch	70.7%
2-stream[44],8-12-22ch	42.3%
SlowFast[47],4-8-16-32ch	83.52%
提案手法, 34ch	**83.84%**
25 MFLOPs	
LRCN[43],12-20-30ch	67.9%
Early-fusion[41],8-14-18ch	66.8%
Late-fusion[41],8-16-18ch	66.2%
Slow-fusion[41],4-10-16ch	65.6%
2-stream[44],4-8-18ch	41.8%
SlowFast[47],1-2-4-8ch	76.25%
提案手法, 22ch	**82.17%**

認識では，既存手法の [2] のような関節点座標と動作のマルチタスク学習よりも関節点状態と動作のマルチタスク学習の方が高精度であることを示している．

表 4.4: 姿勢推定精度 (関節点座標，関節点有無，関節点状態)

	関節点座標 PCK-6px				関節点有無 mAP				関節点状態 マクロ正解率					
手法	頭部	首元	右手	左手	頭部	首元	右手	左手	右手	左手	目	ピッチ	ヨー	ロール
提案手法, 34ch	91.8%	94.0%	89.1%	76.6%	98.3%	73.9%	92.4%	91.1%	85.4%	83.6%	71.6%	74.7%	79.6%	81.3%
提案手法, 22ch	90.8%	93.2%	85.7%	73.2%	98.2%	73.1%	91.7%	90.3%	82.6%	82.0%	70.5%	72.7%	79.2%	78.0%

表 4.5: Ablation Study

手法	正解率	MFLOPs	パラメータ数 [M]
50 MFLOPs			
提案手法, 34ch	**83.84%**	47.20	0.98
Only-Position, 34ch	81.91%	46.86	0.64
Only-Detection, 34ch	80.30%	46.99	0.77
Only-State, 38ch	81.99%	49.76	0.93
No-Pose, 40ch	76.42%	50.53	0.73
25 MFLOPs			
提案手法, 22ch	**82.17%**	25.3	0.75
Only-Position, 22ch	80.82%	24.92	0.41
Only-Detection, 22ch	78.17%	25.05	0.54
Only-State, 26ch	81.70%	25.01	0.67
No-Pose, 28ch	77.23%	25.6	0.45

4.3.5 混同行列

動作パターン別の評価として，提案手法の 34ch のモデルの混同行列を図 4.4，22ch のモデルの混同行列を図 4.5 に示す．どちらのモデルも眠気のシーンが最も精度が低くなっている．眠気のシーンは前方注視や下向き，意識不明と誤判定している割合が多い．眠気のシーンでは開眼と閉眼を繰り返すため，前方注視や意識不明と誤判定してしまう．また，眠気のシーンでうつらうつらする場合には，下向きと誤判定されることが多い．ただし，いずれのシーンでも局所的に誤るだけのため，時間方向の平滑化などの後処理で対応可能である．意識不明やパニックも局所的に眠気や前方注視に誤るだけのため，同じく後処理で対応可能である．

正解/予測結果	前方注視	余所見	眠気	物体把持	下向き	意識不明	パニック
前方注視	81.8%	0.7%	11.0%	0.1%	2.5%	2.6%	1.3%
余所見	1.9%	91.2%	0.4%	2.3%	2.6%	1.2%	0.5%
眠気	9.9%	0.4%	69.0%	0.3%	11.8%	8.5%	0.0%
物体把持	2.6%	1.3%	1.8%	91.9%	1.7%	0.2%	0.6%
下向き	2.6%	0.1%	6.4%	0.3%	88.7%	1.5%	0.4%
意識不明	2.8%	0.5%	13.5%	0.0%	8.0%	75.0%	0.1%
パニック	10.1%	0.9%	2.2%	1.4%	7.9%	0.7%	76.8%

図 4.4: 動作パターン別の混同行列 (34ch)

正解/予測結果	前方注視	余所見	眠気	物体把持	下向き	意識不明	パニック
前方注視	82.9%	0.9%	11.7%	0.2%	2.0%	1.5%	0.9%
余所見	2.6%	94.5%	0.3%	0.1%	1.0%	1.3%	0.2%
眠気	16.3%	0.6%	66.4%	0.3%	9.0%	7.4%	0.1%
物体把持	3.2%	1.7%	1.8%	90.9%	1.4%	0.3%	0.9%
下向き	4.4%	1.0%	7.8%	0.5%	84.5%	1.4%	0.3%
意識不明	4.6%	1.2%	16.8%	0.0%	6.7%	70.6%	0.1%
パニック	18.4%	1.1%	2.3%	1.9%	5.6%	0.8%	69.9%

図 4.5: 動作パターン別の混同行列 (22ch)

4.3.6 実験結果画像

提案手法のドライバ姿勢と動作の出力結果例を表 4.6 に示す．上段，中段が正しく動作を認識できた結果，下段が誤った認識結果となる．

例 3（手元のスマートフォンを操作しているシーン），例 5（カメラで撮影しているシーン）では，関節点座標は正解座標とのズレが少しあるが，ほとんどのシーンで正確に推定している．動作や関節点状態についても，正確に推定している．例 6（急病で意識を失っているシーン）では，関節点状態の目状態が Side となっている．また，顔向きヨーも RightFront と誤認識が起きている．

例 9（前方を注視しているシーン）では少し上を向いているため，顔向きロールが 45 °と誤認識しており，目状態も Close と誤認識している．例 10（パニックのシーン）では，瞬間的に前方を向いたシーンがあり，一時的に前方注視と誤認識している．例 11（余所見のシーン）では，目状態を Close，顔向きピッチを UpFront と誤認識している．例 12（下向きのシーン）では，右下に映っている人形の一部をドライバの手と誤認識したため，関節点座標や左手状態を誤認識している．

4.3.7 考察

図 4.6 下段の誤認識結果を確認すると，誤認識の要因を推測することができる．例 9（前方を注視しているシーン）では，ドライバが上を向いており，人目でも目を開けているかどうかがわからない状態であるため，目状態を Close と誤認識したと考えられる．目を閉じている状態は眠気の特徴であるため，行動を眠気と誤認識したと考えられる．例 11（パニックのシーン）では，メガネのフレームにより目の状態を確認することが難しく，目状態を Close と誤認識し，結果として行動もパニックと誤認識したと考えられる．例 12（下向きのシーン）では，右下の人形をドライバの手と誤認識しているため，手を振り回しているパニックのシーンと誤認識したと考えられる．このようにドライバ動作の誤認識が生じた場合，中間出力結果であるドライバ姿勢が誤認識の要因の解析に役立つ．そのため，提案手法のドライバ姿勢と動作のマルチタスク学習は精度向上だけでなく，誤認識時の解析にも有用である．

種類	正解	推定
行動	前方注視	前方注視
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Open	Open
ピッチ	Front	Front
ヨー	Front	Front
ロール	0°	0°

種類	正解	推定
行動	余所見	余所見
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Side	Side
ピッチ	Front	Front
ヨー	Right	Right
ロール	0°	0°

種類	正解	推定
行動	下向き	下向き
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Side	Side
ピッチ	DownFront	DownFront
ヨー	Front	Front
ロール	0°	0°

種類	正解	推定
行動	眠気	眠気
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Close	Close
ピッチ	Front	Front
ヨー	Front	Front
ロール	0°	0°

種類	正解	推定
行動	物体把持	物体把持
右手状態	Hand-on	Hand-on
左手状態	Hand-on	Hand-on
目状態	Unknown	Unknown
ピッチ	Front	Front
ヨー	Front	Front
ロール	0°	0°

種類	正解	推定
行動	意識不明	意識不明
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Unknown	Side
ピッチ	Up	UpFront
ヨー	LeftBack	RightFront
ロール	90°	Unknown

種類	正解	推定
行動	パニック	パニック
右手状態	Unknown	Unknown
左手状態	Hand-off	Hand-off
目状態	Side	Side
ピッチ	Front	Front
ヨー	LeftFront	LeftFront
ロール	0°	45°

種類	正解	推定
行動	意識不明	意識不明
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Close	Close
ピッチ	Front	Front
ヨー	Front	RightFront
ロール	-45°	-45°

種類	正解	推定
行動	前方注視	眠気
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Open	Close
ピッチ	UpFront	UpFront
ヨー	Front	Front
ロール	0°	45°

種類	正解	推定
行動	パニック	前方注視
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Open	Open
ピッチ	Front	Front
ヨー	Front	Front
ロール	0°	0°

種類	正解	推定
行動	余所見	パニック
右手状態	Unknown	Unknown
左手状態	Unknown	Unknown
目状態	Side	Close
ピッチ	Front	UpFront
ヨー	Right	Right
ロール	0°	0°

種類	正解	推定
行動	下向き	パニック
右手状態	Unknown	Unknown
左手状態	Unknown	Hand-off
目状態	Side	Close
ピッチ	Front	DownFront
ヨー	LeftFront	LeftFront
ロール	0°	-45°

図 4.6: 実験結果画像:青色が頭部中心，緑色が首元，赤色が右手中心，水色が左手中心の関節点座標を示す．◯ は推定結果，× は正解座標を示す．誤った推定結果を赤字で示す．上段と中段は正しく動作を認識できたシーン，下段は誤って認識したシーンとなる．上段左から順に，“ 正面を向いている ”，“ 余所見 ”，“ 手元のスマートフォンを操作 ”，“ うつらうつら ” となる．中段左から順に，“ カメラで写真撮影 ”，“ 急病で意識を失い，前方に倒れている ”，“ 蜂が車内に入り，パニック ”，“ 眠っている ” のシーンとなる．下段左から順に，“ 少し上を向きながら前方を注視 ”，“ 蜂が車内に入り，パニック ”，“ 余所見 ”，“ 下を向いている ” のシーンとなる．

4.4 まとめ

本研究では運転動作評価のためのドライバ認識を提案した．提案手法ではドライバ姿勢と動作のマルチタスク学習を行うことで既存手法と比べて演算量を減らした際の精度低下を抑制できることを示した．本研究ではドライビングシミュレータのデータを用いて評価したが，今後は実車での評価や，より多くの被験者で交差検定を実施することで，更なる実用性向上に取り組む．また，目の細い人などを眠気や意識消失と誤判定しやすいなどの個人差があるため，個人認証やオンライン学習などを用いた個人適応に取り組む．

第5章

Parallel Linked Time-Domain CNNと目に関する時間特徴量によるドライバ眠気推定

ドライバの居眠り運転は深刻な自動車事故につながるため，居眠り運転を減らすことは重要な社会課題である．そのため，ドライバモニタリングシステム (DMS) にドライバの眠気推定を導入することが期待されている．居眠り運転を減らすため，様々なドライバの眠気推定の研究が行われている．

ドライバ眠気推定の研究の多くは，ドライバの強い眠気を検出する 2 値の眠気推定である．2 値の眠気推定は事故を防ぐのに役立つが，システムがドライバの眠気を検知してから事故までの時間が短くなる．検知から事故までの時間が短い場合，システムが取りうる選択肢は大音量の警報などドライバにとって快適ではない手法に限られる．一方，ドライバの強い眠気だけでなく，弱い眠気まで検知するようなマルチレベルの眠気推定では，眠気の検知から事故までの時間を長くすることが可能となる．そのため，マルチレベルの眠気推定は，システムがドライバに干渉する選択肢を増やす．例えば，ドライバが”眠そう”な状態の場合には，冷風を送ることでドライバに不快感を与えることなく，自然に眠気を抑えられる．

北島らは表情評定を用いた眠気の定義を行い，その定義を基に 5 段階の眠気推定 [55] を提案した．以降，北島らの眠気定義は広く使用されている [56, 57, 58, 59]．本研究では北島らが提案した 5 段階の眠気定義を基にしたマルチレベルの眠気推定を提案する．

Percentage of Eyelid Closure(PERCLOS) や瞬き頻度などの時間特徴量は眠気推定に有効である．これらの時間特徴量は 2 値の眠気推定のために設計されているため，強い眠気を捉えるには有効だが，弱い眠気を捉えるようなマルチレベルの眠気推定には適していない．本研究ではドライバの弱い眠気を検出するのに役立つ 2 つの時間特徴量として，Average Eye Closed Time(AECT)，Soft Percentage of Eyelid Closure(Soft PERCLOS) を提案する．AECT は瞬き間隔の平均フレーム数である．AECT は弱い眠気のドライバが頻繁に瞬きをする状態と，強い眠気のドライバが一定時間目を閉じている状態を判別するのに役立つ．Soft PERCLOS はドライバの目が完全に開いていないフレーム数の割合を示す．Soft PERCLOS は目が完全に開いていない弱い眠気を検知することに役立つ．

Time-domain Convolutional Neural Network(CNN) を用いた眠気推定が提案されている．Time-domain CNN は時間的な特徴の抽出に役立ち，その特徴は眠気推定に役立つ．Time-domain CNN ではプーリング層により，特徴マップのサイズを段階的に小さくする．特徴マップの縮小により，特徴マップの時間解像度を小さくなる．Time-domain CNN では各層が直列に繋がっているため，最上位層の全結

合層では，単一の時間解像度をもった特徴マップを用いて，眠気推定を行う．Shih らは畳み込み層を並列に結合した Multistage Spatial-Temporal Network(MSTN) を提案した [60]．MSTN では最上位層の全結合層で，複数の畳み込み層の特徴マップを用いて眠気を推定する．複数の畳み込み層の特徴マップを並列に結合されるため，MSTN では複数の空間解像度の特徴マップを用いて眠気を推定できる．複数の空間解像度の特徴マップは眠気推定に役立つが，眠気推定では空間解像度よりも時間解像度の方が重要である．そのため，本研究では複数の時間解像度の特徴マップを用いて眠気推定を行う Parallel Linked Time-domain CNN を提案する．また，本研究では Parallel Linked Time-domain CNN が複数の時間解像度の特徴を捉えていることを示すため，感度マップを用いた入力特徴量の重要度を可視化する．

本研究では AECT と Soft PERCLOS の 2 つの時系列特徴と，Parallel Linked Time-domain CNN を組み合わせた高精度なマルチレベル眠気推定を提案する．提案手法では，初めにドライバの顔と目を検出し，各フレームのドライバの画像から目の幅，高さ，開眼度などの目に関する特徴量を抽出する．次に提案手法では目に関する特徴量の時系列データから，眠気推定に役立つ 4 つの時間特徴量を抽出する．本研究では提案手法の AECT，Soft PERCLOS だけでなく，既存手法の瞬き頻度と PERCLOS も使用することで高精度なマルチレベル眠気推定に役立つ特徴量を得る．最後に目に関する特徴量と時間特徴量を入力として，Parallel Linked Time-domain CNN を用いてマルチレベルの眠気を推定する．

眠気推定に用いられるデータセットはドライビングシミュレータで撮影されている．ドライビングシミュレータは，背景や照明などの環境が実車の環境とは大きく異なる．また，実車では車の振動により目の位置が不安定となるため，正確な眠気推定は難しい．そのため，本研究では実車で撮影したデータセットを作成し，実車での利用を想定した評価実験を行い，提案手法の効果を示す．

5.1 関連研究

5.1.1 ドライバの眠気推定手法

ドライバの眠気推定はセンシングの方法により，生体センシング，車体センシング，画像センシングの 3 つに分けられる．

生体センシングは Electroencephalograms (EEG) [61, 62, 63, 64, 65, 66], Electrocardiograms (ECG) [56, 67], Electrooculograms (EOG) [56, 67] を用いた手法である．これらの方法はドライバに生体センサを取り付ける必要があるため，ドライバに身体的な負荷がかかる．

車体センシングは車のホイールやブレーキ，レーンの逸脱などの情報を基にした手法である [68, 69, 70]．これらの手法はドライバの身体的な負担はないが，眠気とは無関係の運転技術，道路環境，車の個体差などに影響を受ける．

画像センシングはドライバの画像から，目の開眼度，頭部の動き，あくび，などの顔の外観的特徴を抽出して，眠気を推定する [71, 72, 73, 60, 74, 75]．画像センシングはドライバへの負荷も少なく，運転のパターンとは異なり眠気以外の要因の影響を受けにくい．本研究ではカメラでドライバの顔

を撮影する画像センシングによる眠気推定を提案する．

5.1.2 目に関する時間特徴量

ドライバ眠気推定に役立つ目に関する時間特徴量が提案されている．眠気推定では最も使用されている特徴量は閉眼時間に関するものである．閉眼時間に関する代表的な時間特徴量として，Percentage of Eyelid Closure(PERCLOS)[76] と瞬き頻度 [77] がある．

Wierwille らは PERCLOS がドライバの眠気と強い相関を持っていることを示した [76]. PERCLOS は式 (5.1) を用いて，一定時間における閉眼状態の割合から計算される．n^t_{close} と N^t_{total} は一定時間 t における閉眼状態のフレーム数，閉眼および開眼状態のフレーム数を示す．

$$PERCLOS^t = \frac{n^t_{close}}{N^t_{total}}, \tag{5.1}$$

Zhang らは瞬き頻度もドライバ眠気推定に重要な特徴量であると述べている [77]. 瞬き頻度は式 (5.2) を用いて計算される．$n^t_{blinking}$ は時間 t における瞬きにかかるフレーム数を示す．

$$f^t_{blink} = \frac{n^t_{blinking}}{N^t_{total}}, \tag{5.2}$$

PERCLOS と瞬き頻度の 2 つの時間特徴量は，ドライバの強い眠気を捉える 2 値の眠気推定のために提案されたものである．本研究では，ドライバの弱い眠気も捉えるための時間特徴量を提案する．

5.1.3 CNN を用いたドライバ眠気推定

Convolutional Neural Network(CNN) はドライバの顔画像から直接特徴を抽出するために用いられる．Lyu らは CNN と Long Short-term Memory(LSTM)[78, 79] を用いて 2 値の眠気推定を提案した [80]．Reddy らは CNN を用いて ”眠そう”，”あくび”，”通常 ” の 3 つの状態を推定する眠気推定を提案した [75]．Huynh らは空間方向だけでなく時間方向にも畳み込みを行う 3D-CNN[81] を用いて，ドライバの動画から眠気を推定する手法を提案した [74] ．Shih らは VGG-16[82] と LSTM を用いて 2 値の眠気推定を行う MSTN を提案した [60]．MSTN は異なる畳み込み層の複数の特徴マップを結合し，その特徴マップを用いて LSTM で眠気推定を行う．そのため，MSTN は複数の空間解像度を持った特徴量を用いた眠気推定が可能となる．

これらの CNN を用いた眠気推定は全てドライバの顔画像を入力としており，CNN を用いて顔画像から眠気推定に有効な特徴量を抽出する．CNN を用いて特徴を抽出するには多くの学習データが必要となる．しかし，顔検出用の学習データなどに比べて，ドライバの眠気に関するデータは大量に収集することは難しい．

5.1.4 マルチレベルの眠気推定

5.1.3 節の CNN を用いた眠気推定は，強い眠気を推定する 2 値または 3 値の眠気推定である．そのため，それらの手法をドライバの弱い眠気を捉えるような眠気の早期検知に利用することは難しい．弱い眠気を含んだマルチレベルの眠気推定が提案されている．

中村らは顔画像列から瞼の動きや皺の変化など手動で設計した特徴量を抽出し，k 近傍法を用いて 5 段階の眠気推定を提案した [57]．瞼の動きなどの眠気推定のために設計された特徴量は眠気推定に有効であるが，k 近傍法による推定は CNN に置き換えるなど，中村らの手法は識別器に工夫の余地がある．

Sun らは時系列の瞬き情報を入力情報として，時間方向への畳み込みを行う Time-domain CNN または LSTM を用いたネットワークモデルによる眠気推定を提案した [59]．時系列の瞬きに対する Time-domain CNN は眠気推定に有効だが，単純な瞬きのみを入力としているため，Sun らの眠気推定は入力特徴量に工夫の余地がある．また，Time-domain CNN は上位層の全結合層で単一の時間解像度の特徴マップしか用いていない．

マルチレベルの眠気推定を行う関連研究には，入力特徴量や眠気推定を行うネットワークモデルに工夫の余地がある．

5.1.5 眠気推定用データセット

眠気推定のデータセットには，National Tsing Hua University Drowsy Driver Detection (NTHU-DDD) video dataset [83] や ULg Multimodality Drowsiness Database (DROZY)[84] がある．NTHU-DDD と DROZY の画像例を図 5.1 に示す．

Weng らが提案した NTHU-DDD[83] は 36 人の被験者を撮影した近赤外カメラの動画と RGB カメラの動画から構成される．NTHU-DDD には，ドライバが強い眠気を持っているかどうかの 2 値ラベルが付与されている．そのため，本研究で扱う弱い眠気を含んだマルチレベルの眠気推定には利用できない．

Massoz らが提案した DROZY は 14 人の被験者に対して，electroencephalograms(EEG), electrocardiograms(ECG), electrooculograms(EOG), electromyograms(EMG), 近赤外カメラを用いてデータを作成している．データセットには Karolinska Sleepiness Scale(KSS) を用いて 9 段階に評定されたアノテーションが付与されている．DROZY では生体信号を取得するためのセンサがドライバに多数取り付けられているため，装着による不快感が生じてしまう．

NTHU-DDD と DROZY はいずれも，ドライビングシミュレータで撮影されたデータセットである．ドライビングシミュレータでは，実車環境で起こる照明変動や車の振動などの影響を評価できない．車の振動はドライバの目の位置が不安定になるため，眠気推定への影響が大きい．

以上より，既存の眠気推定用データセットは，弱い眠気レベルを含んでいない，生体センサによるドライバへの身体的負担がある，ドライビングシミュレータを使用している，などの課題があり，本研究で使用することが難しい．従って，本研究では近赤外線カメラを用いて実車環境で撮影した眠

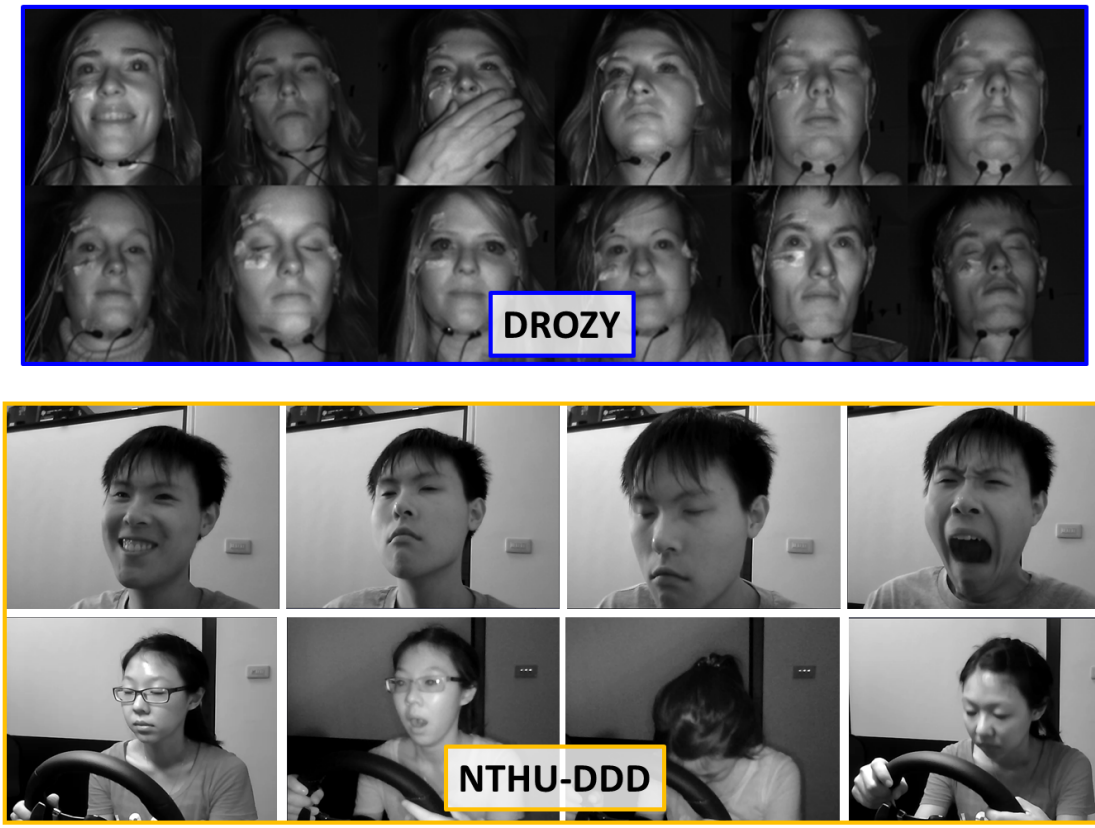

図 5.1: 関連研究の眠気推定データセット

気推定のデータセットを構築する．また，撮影したデータセットに対して，北島らが提案した 5 段階の表情評定を用いて弱い眠気レベルを含んだマルチレベルのラベルを付与する．

5.2 提案手法

本研究で提案するドライバの眠気推定は，下記 3 つの要素で構成される．

- 目に関するフレーム単位の特徴量抽出
- 目に関する時間特徴量の抽出
- Parallel Linked Time-domain CNN を用いた眠気レベルの推定

各要素の詳細を図 5.2 に示す．

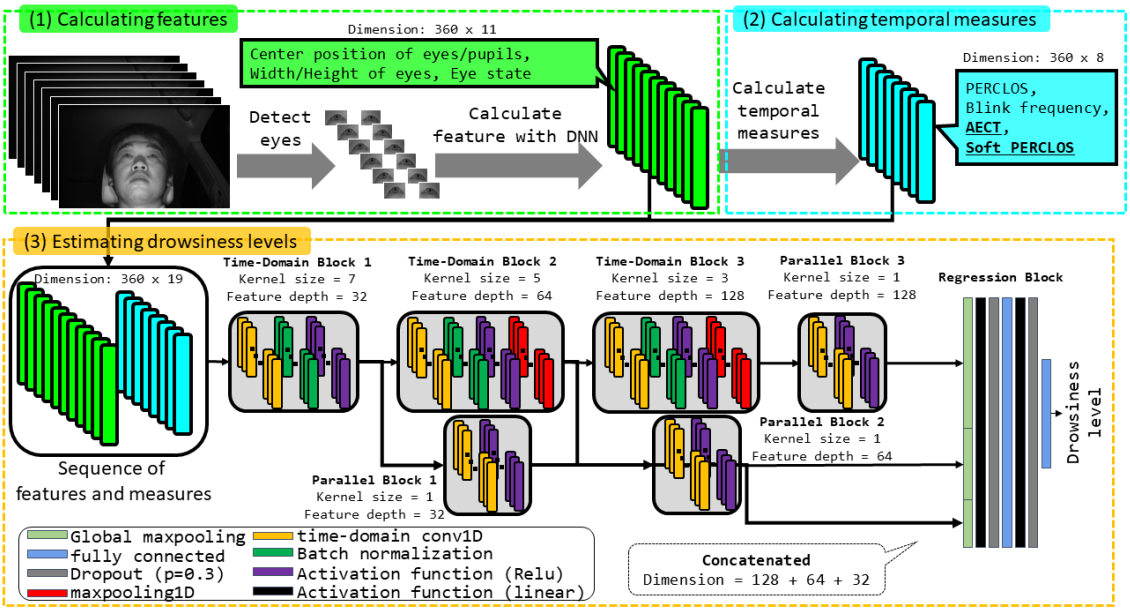

図 5.2: 提案手法の概要：(1) 目に関するフレーム単位の特徴量抽出 (2) 目に関する時間特徴量（PERCLOS, 瞬き頻度, AECT, Soft PERCLOS）の抽出 (3) Parallel Linked Time-domain CNN を用いたの眠気レベルの推定

5.2.1 眠気レベル

本研究で推定する眠気レベルとして，北島らが提案した表情評定による 5 段階の眠気定義 [55] を用いる．北島らが提案した眠気レベルは，(1) 全く眠くなさそう (alert), (2) やや眠そう (slightly drowsy), (3) 眠そう (moderately drowsy), (4) かなり眠そう (significantly drowsy), (5) 非常に眠そう (extremely drowsy) の 5 段階である．北島らが提案した眠気レベルと各レベルの基準を表 5.1 に示す．表情評定では評定者が数秒単位に区切ったドライバの画像列を目視して，基準に従い，眠気レベルを付与する．

本研究では，5.4 章で示す実験より眠気レベル 1 と 2 は人目でも区別が難しいと判断し，これらを 1 つにまとめた 4 段階の眠気レベルを用いる．DMS では，眠気レベル 3 を検知できると，十分に早い段階でドライバの眠気を検知できるため，ドライバの眠気を抑制するための対策を取るための十分な時間を得ることができる．従って，本研究ではレベル 1 とレベル 2 を区別しない．本研究で用いる眠気レベルは，(1) alert, (2) moderate drowsy, (3) significantly drowsy, (4) extremely drowsy の 4 段階とする．

5.2.2 目に関するフレーム単位の特徴量抽出

提案手法では初めに，各フレームのドライバ画像から，(1) 目と瞳孔の中心座標（x, y の 2 次元），(2) 目の幅と高さ（両目の平均），(3) 目状態（両目の平均），の 3 つの特徴量を抽出する．目の検出

表 5.1: 表情評定法の定義

評定値 (眠気レベル)	カテゴリ	基準
1	全く眠くなさそう (alert)	視線の移動が速く，頻繁である． 瞬きの周期が安定している． 動きが活発で身体の動きを伴う．
2	やや眠そう (slightly drowsy)	視線移動の動きが遅い． 唇が開いている．
3	眠そう (moderately drowsy)	瞬きはゆっくりと頻発． 口の動きがある． 座り直しがある． 顔に手をやる．
4	かなり眠そう (significantly drowsy)	意識的と思われる瞬きがある． 頭を振る． 肩の上下動など無用な体全体の動きあり． あくびは頻発し，深呼吸も見られる． 瞬きも視線の動きも遅い．
5	非常に眠そう (extremely drowsy)	瞼を閉じる． 頭が前に傾く． 頭が後ろに倒れる．

は OKAO Vision[85] を用いて行う．ドライバ画像から目画像を切り出した後，64 × 64 の解像度にリサイズする．リサイズされた目画像は，ResNet[86] ベースの CNN に入力し，上記 3 つの目に関するフレーム単位の特徴量を抽出する．目状態は-1.0〜1.0 の値をとり，0.0 以上の場合に開眼状態となる．目と瞳孔の中心座標は 2 次元とするため，合計で 8 次元となる．目の幅と高さは両目の平均値とするため，合計で 2 次元となる．目状態も両目の平均値とするため，合計で 1 次元となる．従って，目に関するフレーム単位の特徴量は合計で 11 次元となる．

5.2.3 目に関する時間特徴量の抽出

ResNet ベースの CNN を用いて推定したフレーム単位の特徴量を用いて，目に関する時間特徴量を抽出する．本研究で抽出する時間特徴量は PERCLOS，瞬き頻度，AECT，Soft PERCLOS の 4 つである．PERCLOS と瞬き頻度は眠気推定に用いられる時間特徴量であり，強い眠気かどうかを判定する 2 値の眠気推定に有効である．本研究では弱い眠気を含んだマルチレベルの眠気推定に有効な時間特徴量として，AECT，Soft PERCLOS を提案する．

■ AECT (Average Eye Closed Time)

本研究で提案する AECT は，瞬きの際の平均閉眼フレーム数である．強い眠気のドライバは長時間目を閉じる傾向がある．一方，弱い眠気のドライバは頻繁に瞬きを行うため，目を閉じる時間が短い．それらの状態を明確に区別するため，本研究では AECT を提案する．AECT は式 (5.3) にて計

算される．

$$AECT^t = \frac{PERCLOS^t}{f^t_{blink}}. \tag{5.3}$$

AECT は眠気推定で用いられる Average Eyes Closed Speed (AECS) [71] と似ているが，AECS は瞬きの速度であるため，高フレームレートのカメラを使用する必要がある．一方，AECT は本実験で使用するような 30FPS 程度の低フレームレートでも使用できる．AECT と PERCLOS はいずれも閉眼フレーム数に着目した時間特徴量であるが，その性質は大きく異なる．AECT と PERCLOS の違いが明確となる例を図 5.3 に示す．パターン 1 は閉眼が続くシーン，パターン 2 が短い閉眼が頻繁に起こるシーンである．いずれも PERCLOS は同じ値になるが，AECT は異なる値になる．強い眠気ではパターン 1 のような一定時間閉眼となるシーンが起こるが，弱い眠気ではパターン 2 のような頻繁に瞬きを行うシーンが起こる．そのため，AECT は弱い眠気を捉えるのに役立つ．

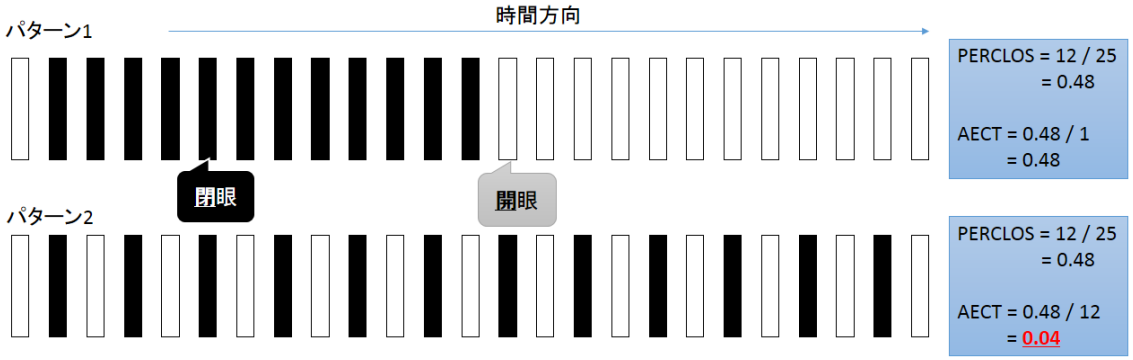

図 5.3: AECT と PERCLOS の違い

■ Soft PERCLOS

本研究では提案する Soft PERCLOS は目が完全な開眼状態でないフレームの割合である．Soft PERCLOS は式 (5.4) で計算される．

$$PERCLOS^t_{soft} = \frac{n^t_{soft\_close}}{N^t_{total}}, \tag{5.4}$$

$n^t_{soft\_close}$ は一定時間 t において，CNN の出力である目状態が $S_{eye} < 0.8$ となっているフレーム数を示す．

Soft PERCLOS と PERCLOS 及び AECT の違いが明確となる例を図 5.4 に示す．左から右に向かって，眠気レベルが低いシーンから強いシーンとなる．眠気レベル 2 の弱い眠気ではドライバの目状態は完全な開眼と閉眼を口語に繰り返す．一方，眠気レベル 3 のシーンでは開眼時の目状態がレベル 2 に比べて低い値となっている．従って，眠気レベル 2 と 3 のシーンではどちらも PERCLOS は同じ値となるが，Soft PERCLOS は異なる値となる．そのため，Soft PERCLOS は弱い眠気を捉えるのに役立つ時間特徴となる．

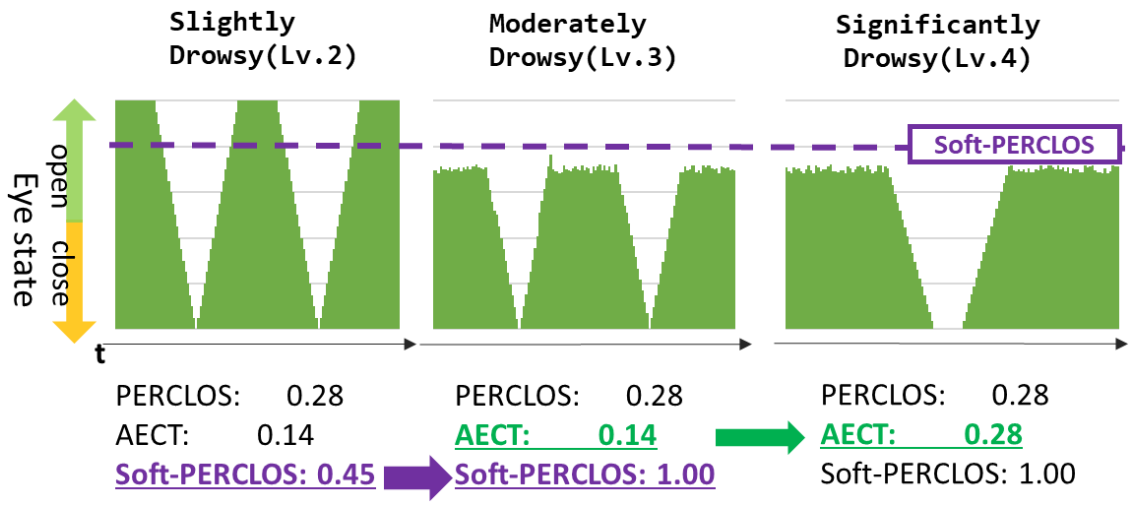

図 5.4: Soft PERCLOS

■ 複数の時間パラメータ

本研究では瞬き頻度，PERCLOS，AECT，Soft PERCLOS の 4 つの時間特徴量を使用する．これらの時間特徴量は，一定時間 t より計算される．時間 t は精度に影響するため，本研究では複数の異なる時間 t を用いて，複数の時間特徴量を抽出する．本研究では複数の時間 t として，10 秒と 20 秒を用いる．時間特徴量の次元数は 4×2=8 となる．

5.2.4 Parallel Linked Time-domain CNN を用いた眠気レベルの推定

本研究で提案するネットワークモデル Parallel Linked Time-domain CNN は，下記 3 つの要素で構成される．モデルの概要を図 5.2 に示す．

1. Time-domain Convolution Block
2. 並列平滑化ブロック
3. 回帰ブロック

ネットワークモデルは 5.2.2 節と 5.2.3 節で計算するフレーム単位の特徴量と時間特徴量を入力とする．フレーム単位の特徴量は目と瞳孔の中心座標，目の幅と高さ，目状態，時間特徴量は PERCLOS, blink frequency, AECT，Soft-PERCLOS の 4 つである．これらの特徴量は過去数フレーム分をまとめて計算され，毎フレーム同じ処理を行い，ネットワークモデルへの入力とする．なお，入力に用いるフレーム数は入力する時間範囲 T と FPS fps により決定する．本研究では T を 30，fps を 12 とするため，合計のフレーム数は 360 フレームとなる．従って，ネットワークモデルの入力は 360×19 となる．

■ **Time-domain Convolution Block**

本研究ではフレーム単位の特徴量と時間特徴量に対して，時間方向の畳み込みを行うことで眠気推定に有効な特徴量を抽出する．本研究では 3 つの Time-domain Convolution Block を用いる．各ブロックのカーネルサイズは下位層から順に 7，5，3 とする．3 つの Time-domain Convolution Block は直列に結合し，2 番目と 3 番目のブロックは Max Pooling 層を含む．

■ 並列平滑化ブロック

並列平滑化ブロックは，Time-domain Convolution Block で抽出した複数の時間解像度の特徴マップを平滑化するために使用される．並列平滑化ブロックはカーネルサイズが 1 の時間方向の畳み込み層であり，出力チャンネル数と入力チャンネル数は同じ数とする．

畳み込み層の並列化は MSTN[60] を参考にしている．MSTN の並列平滑化ブロックは，画像に対して空間方向の畳み込みの後に，複数の空間解像度の特徴マップに平滑化を行う．一方，図 5.5 に示す通り，本研究の並列平滑化ブロックは時間方向の畳み込みの後に，複数の時間解像度の特徴マップに平滑化を行う．眠気推定では空間方向の変化よりも，開眼度の変化のように時間方向の変化を捉えることが重要なため，複数の時間解像度の特徴マップが精度向上に重要である．

Parallel Linked Time-domain CNN は複数の時間解像度の特徴量を抽出するため，5.2.3 節で述べた複数の時間 t を用いた時間特徴量と似た特徴量を重複する可能性がある．しかし，本研究では両方を組み合わせることで精度が向上することを示す．

■ 回帰ブロック

並列平滑化ブロックで処理した後，本研究では時間解像度の異なる特徴マップを結合するため Global Max Pooling (GMP) 層を用いる．各並列平滑化ブロックの出力は，GMP を経て 1 次元の特徴量となる．複数の時間解像度の特徴マップは，GMP の後に結合され，回帰ブロックに入力される．回帰ブロックは 2 層の全結合層で構成され，線形の活性化関数を用いて，眠気レベルを出力する．ただし，各活性化関数の後には，0.3 の確率で値を 0 にするドロップアウト層を用いる．GMP 層を除くと回帰ブロックは単純な射影となるが，学習時にはドロップアウト層を用いるため，単純な射影に置き換えることはできない．

5.3 実験

本研究では入力特徴量の精度比較と，ネットワークモデルの精度比較を行い，提案手法の有効性を示す．また，SmoothGrad[87] を用いた感度マップを用いて，提案手法の Parallel Linked Time-domain CNN が複数の時間解像度の特徴量を抽出できることを示す．最後に眠気の早期検出に関する実験結果を示す．

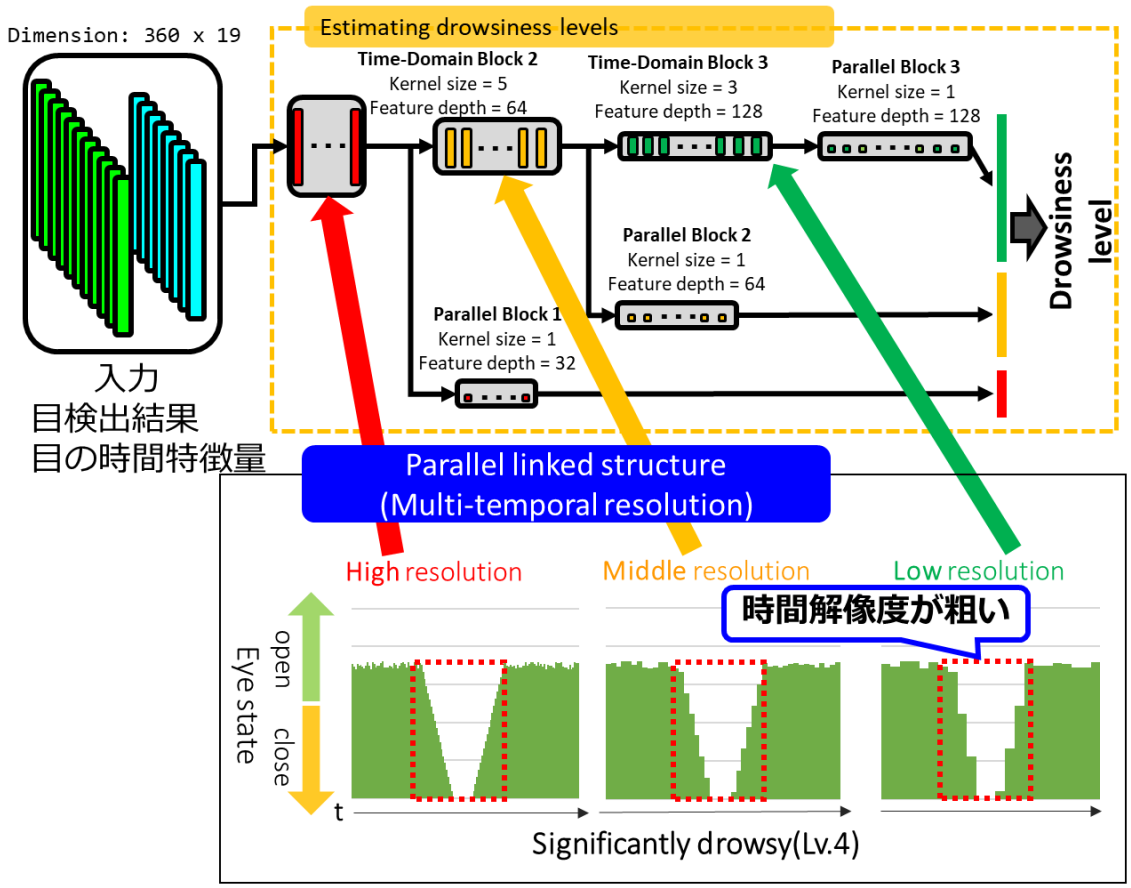

図 5.5: 複数の時間解像度の特徴抽出

5.3.1 実験データ

本研究では実車で撮影したデータセットを用いる．安全のため，助手席の前方に近赤外線カメラを取り付けて，助手席の被験者を撮影した．37 名の被験者を撮影し，数名の被験者はメガネやマスクを着けている．動画のフレームレートは 60FPS，各動画の撮影時間は 30 分程度となる．被験者の動画を 5 秒の短いクリップに分割し，各クリップに対して 3 人の評定者が表情評定により眠気レベルを付与する．実験に用いる眠気レベルは，3 人の評定者が付けた眠気レベルの平均値を使用する．また，眠気レベルの平均値は時間方向に線形補間される．本研究で用いるデータセットの例を図 5.6 に示す．弱い眠気の画像でも瞬きにより目を閉じている瞬間があり，眠気推定には時間変化を捉える必要があることを示している．

図 5.6: データセットの画像例：上段から順に眠気レベル 1 ”全く眠くなさそう”，レベル 3 の ”眠そう”，レベル 4 の ”非常に眠そう”．各レベルの画像列から 0.1 秒間隔で抜粋．

5.3.2 実験の詳細

■ 入力

計算量を減らすため，本研究では 60FPS の動画を 12FPS に間引いて評価する．各フレームの入力は，30 秒分の計 360 フレームとする．提案手法では，各フレームで計算する目に関する特徴量として，目と瞳孔の中心座標，両目の幅と高さの平均，目の状態，10 秒と 20 秒で抽出する時間特徴量として，$PERCLOS$, f_{blink}, $AECT$, $PERCLOS_{soft}$ を用いる．目と瞳孔の中心座標は 8 次元，両目の幅と高さの平均が 2 次元，目の状態が 1 次元となる．各時間特徴量は 1 次元となるため，10 秒と 20 秒の 2 種類で，計 8 次元となる．以上より，ネットワークモデルの入力は 360×19 の特徴量となる．360 は入力フレーム数，19 は入力特徴量の次元数である．また，本研究の比較実験に用いるネットワークモデルの内，画像を入力とするモデルは，OKAO Vision[85] を用いて検出した目画像を入力とする．目画像は 64×32 にリサイズするため，それらのネットワークモデルの入力のサイズは 360×2048 となる．

■ ハイパーパラメータ

本実験で用いる最適化手法 Adam[88] のハイパーパラメータは，$lr = 0.001$, $betas = (0.9, 0.999)$, $eps = 1e^{-8}$, $weight\ decay = 0.0005$ となる．損失関数は L1 損失を用いる．

■ 学習・評価方法

本研究では 5 分割交差検証を用いて，提案手法を評価する．5 分割交差検証では，データセットを被験者毎に 5 つのグループに分け，1 つのグループをテスト用に，残りのグループを学習用に用いる．テストに使用するグループを変えながら，計 5 回の評価を行う．

学習時には各エポックで学習用のデータから，ランダムに 1,024 個のクリップを抜き出して，ネットワークモデルの学習を行う．各クリップは 30 秒の動画となる．本研究では学習エポック数を 100 とするため，学習に使用するクリップは計 102,400 個となる．バッチサイズは 128 である．画像を入力とするネットワークモデルでは，GPU メモリの制限により，バッチサイズを 4 とする．評価ではノイズ除去のため，正解データと予測結果の両方に対して，Exponential Moving Average (EMA) による平滑化を行う．EMA のウィンドウ幅は 30 フレームとする．

■ 評価指標

評価指標には精度と Mean Absolute Error(MAE) を用いる．精度は下式より計算される．Y_i は i フレーム目のモデルが出力した眠気レベル, $\hat{Y}_i$ は正解の眠気レベルである．M は誤差のしきい値であり，誤差が M 未満の場合を正解とする．

$$Correct = \begin{cases} 1,\ if\ |Y_i - \hat{Y}_i| < M, \\ 0,\ otherwise. \end{cases}$$

■ 精度比較対象のネットワークモデル

提案手法の比較対象として，LSTM，VGG-LSTM，並列構造の VGG-LSTM，Time-domain Pooling を用いた VGG-LSTM，3D-CNN，1 層の Time-domain CNN，3 層の Time-domain CNN の 7 つのモデルを用いる．

実験で用いる各モデルの詳細を表 5.2 と表 5.3 に示す．表 5.2 は画像を入力とするネットワークモデル，表 5.3 は特徴量を入力とするネットワークモデルである．”LSTM”と”FC”の引数は出力ノード数を示す．”DO”はドロップアウト層，引数はドロップアウトにより値を 0 にする確率を示す．”BN”はバッチ正規化層，”Conv”は 2 次元の畳み込み層，”Conv3”は 3 次元の畳み込み層を示す．畳み込み層の引数は 1 つ目が出力チャンネル数，2 つ目がカーネルサイズを示す．”MaxPool”は最大プーリング層，引数はカーネルサイズを示す．”Block”は VGG ブロックを示し，ブロックの構成は [82] と同じである．ブロックの引数は 1 つ目が出力チャンネル数，2 つ目が畳み込み層の数を示す．

Time-domain Pooling を用いたモデルについては，Ng らによって提案されたモデル [89] を参考としており，VGG で各目画像に対して畳み込み層による特徴抽出を行った後，時間方向の Max Pooling を用いて，時間方向の解像度を小さくする．VGG-LSTM と，Time-domain Pooling を用いる VGG-LSTM は全て同じネットワーク構成となる．1 層の Time-domain CNN は時間方向の畳み込み層が 1 層となり，その後に全結合層を用いて眠気レベルを出力する．3 層の Time-domain CNN は時間方向の畳み

込み層が直列に 3 層繋がっており，その後全結合層を用いて眠気レベルを出力する．

表 5.2: 目の画像列を入力とするモデル

<table>
<tr><td>VGG-LSTM</td><td colspan="3">VGG-LSTM
(parallel)</td><td>3D-CNN</td></tr>
<tr><td>Block(64,2),
MaxPool(2)</td><td colspan="3">Block(64,2),
MaxPool(2)</td><td>Conv3(16,7),
BN, ReLU</td></tr>
<tr><td>Block(128,2),
MaxPool(2)</td><td colspan="3">Block(128,2),
MaxPool(2)</td><td>Conv3(32,5),
BN, ReLU</td></tr>
<tr><td>Block(256,2),
MaxPool(2)</td><td colspan="2">Block(256,2),
MaxPool(2)</td><td>Conv(128,1),
ReLU</td><td>MaxPool(2)</td></tr>
<tr><td>Block(512,2),
MaxPool(2)</td><td>Block(512,2),
MaxPool(2)</td><td>Conv(256,1),
ReLU</td><td>-</td><td>Conv3(64,3),
BN, ReLU</td></tr>
<tr><td>Block(512,2),
MaxPool(2)</td><td>Block(512,2),
MaxPool(2)</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>-</td><td>Conv(512,1),
ReLU</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="5">Global MaxPool</td></tr>
<tr><td>-</td><td colspan="3">concat</td><td>-</td></tr>
<tr><td colspan="4">BN, DO(0.3), FC(128),
BN, ReLU, DO(0.4), FC(128), ReLU</td><td>BN,
DO(0.3),
FC(64)</td></tr>
<tr><td colspan="4">LSTM(64), BN, FC(1)</td><td>DO(0.3),
FC(1)</td></tr>
</table>

5.3.3 精度比較

本研究ではネットワークモデルの精度比較と，入力特徴量の精度比較を行う．ネットワークモデルの精度比較では，提案手法の Parallel Linked Time-domain CNN の精度を検証する．入力特徴量の精度比較では，提案手法の時間特徴量である AECT と Soft PERCLOS の精度を検証する．

■ ネットワークモデルの精度比較

入力に使用する特徴量を同じにした条件で，ネットワークモデルのみを変更して交差検証を行う．ネットワークモデルの精度比較結果を表 5.4 に示す．上から 4 つのモデルは目画像を入力とするモデルの結果，下の 4 つのモデルは本研究の特徴量を入力とするモデルの結果である．目画像を入力とするモデルよりも，特徴量を入力とすもモデルの方が精度が高い．従って，CNN を用いて目画像から特徴抽出を行うよりも，眠気を捉えるために設計された特徴量の方が有効である．CNN を用いて画像から識別に有効な特徴を自動で抽出するには，大量のデータセットが必要となる．実車環境でドライバが眠くなるデータを大量に集めることは難しいため，ドライバの眠気推定には，眠気を捉えるために設計された特徴量の方が適している．また，特徴量を入力とするモデルでは，LSTM よりも時間方向の畳み込みを行う Time-domain CNN の精度が高い．そのため，眠気推定には時間変化を捉

表 5.3: 特徴量を入力とするモデル

<table>
<tr><td>LSTM</td><td>1 time domain conv block</td><td>3 time domain conv blocks</td><td colspan="3">Parallel Linked Time-domain CNN (3 time-domain convolution blocks, ours)</td></tr>
<tr><td>LSTM(256)
DO(0.05)</td><td>Conv(16,3)
BN, ReLU</td><td>Conv(32,7)
BN, ReLU</td><td colspan="3">Conv(32,7)
BN, ReLU</td></tr>
<tr><td>LSTM(256)
DO(0.05)</td><td>-</td><td>Conv(64,5)
BN, ReLU
MaxPool(2)</td><td colspan="2">Conv(64,5)
BN, ReLU
MaxPool(2)</td><td>Conv(32,1),
ReLU</td></tr>
<tr><td>LSTM(256)
DO(0.05)</td><td>-</td><td>Conv(128,3)
BN, ReLU
MaxPool(2)</td><td>Conv(128,3)
BN, ReLU
MaxPool(2)</td><td>Conv(64,1),
ReLU</td><td>-</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>Conv(128,1),
ReLU</td><td>-</td><td>-</td></tr>
<tr><td>-</td><td colspan="5">Global MaxPool</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td colspan="3">concat</td></tr>
<tr><td colspan="6">BN, DO(0.3), FC(64), DO(0.3), FC(1)</td></tr>
</table>

えることが可能な Time-domain CNN が有効である．また，Time-domain CNN と提案手法の Parallel Linked Time-domain CNN では提案手法の方が精度が高い．従って，提案手法が用いる複数の時間解像度の特徴量がドライバの眠気推定に最も有効である．

表 5.4: ネットワークモデルの精度比較：評価指標は精度と MAE の 2 種類，精度のしきい値 M は，1.0・上から 4 つは画像を入力とするモデル，下の 4 つは特徴量を入力とするモデルの結果を示す．

Model	精度 (M=1.0)	MAE
VGG-LSTM	68.98%	0.7589
VGG-LSTM(parallel)	51.92%	0.9831
VGG-LSTM(time-domain pool)	55.07%	1.0453
3D-CNN	63.28%	0.8282
LSTM	73.73%	0.7391
Time-domain CNN 1 block	79.77%	0.6374
Time-domain CNN 3 blocks	93.85%	0.4525
Parallel Linked Time-domain CNN (3 blocks, ours)	**95.86%**	**0.4007**

■ 入力特徴量の精度比較

異なる入力特徴量を用いて精度を比較した結果を表 5.5 に示す．なお，評価に用いるモデルは，提案手法の Parallel Linked Time-domain CNN とする．上から順に特徴量を追加した場合の精度を示す．

上の 2 つは目に関するフレーム単位の特徴量を用いた結果を示す．目・瞳孔の座標は，左右の目と瞳孔両方の座標を示し，合計で 8 次元の特徴量となる．上から 2 つ目は，目の幅と高さの平均，目状態を追加した結果となり，合計が 11 次元となる．3 つ目は，既存手法の時間特徴量である PERCLOS と瞬き頻度を追加した場合の結果となり，合計が 13 次元となる．4 つ目は提案手法の時間特徴量である AECT と Soft PERCLOS を追加した場合の結果となり，合計が 15 次元となる．3 つ目と 4 つ目の時間特徴量は t を 20 秒として計算する．5 つ目は 5.2.3 節で述べた複数の時間 t から計算した時間特徴量を追加した結果となり，t を 10 秒として計算した 4 つの時間特徴量を追加している．特徴量の合計は 19 次元となる．提案手法の Parallel Linked Time-domain CNN では，入力を目と瞳孔の座標のみとした場合でも高い精度である．提案手法の AECT や Soft PERCLOS を用いた場合では，更に精度が高くなっており，複数の時間パラメータを用いた提案手法の精度が最も高い．従って，提案手法の AECT, Soft PERCLOS はマルチレベルの眠気推定に有効である．

表 5.5: 入力特徴量の精度比較

特徴量 (次元数)	精度 (M=1.0)	MAE
目・瞳孔の座標 (8)	88.37%	0.5090
＋目の幅と高さの平均，目状態 (11)	92.20%	0.4389
＋既存手法の時間特徴量: PERCLOS，瞬き頻度 (13)	94.79%	0.4191
＋提案手法の時間特徴量: AECT，Soft PERCLOS (15)	94.91%	0.4096
＋複数の時間パラメータ (19)	**95.86%**	**0.4007**

5.3.4 解析

本研究では提案手法のネットワークモデルを解析するため，予測結果を時系列グラフにした可視化，感度マップを用いた可視化を行う．時系列グラフによる可視化では，予測した眠気レベルと正解の眠気レベルに相関があるかどうかを視覚的に確認する．感度マップでは入力に対する感度を確認し，提案手法の Parallel Linked Time-domain CNN が複数の時間解像度の特徴を抽出できることを示す．

■ 予測結果の時系列グラフ

提案手法が予測した眠気レベルと正解の眠気レベルをプロットした結果を図 5.7，図 5.8 に示す．赤線が予測した眠気レベル，青線が正解の眠気レベルを示す．提案手法が予測した眠気レベル，正解の眠気レベルのどちらも Exponential Moving Average (EMA) を用いた平滑化を行う．予測した眠気レベルと正解の差が小さく，提案手法がドライバの眠気の変化を捉えている．

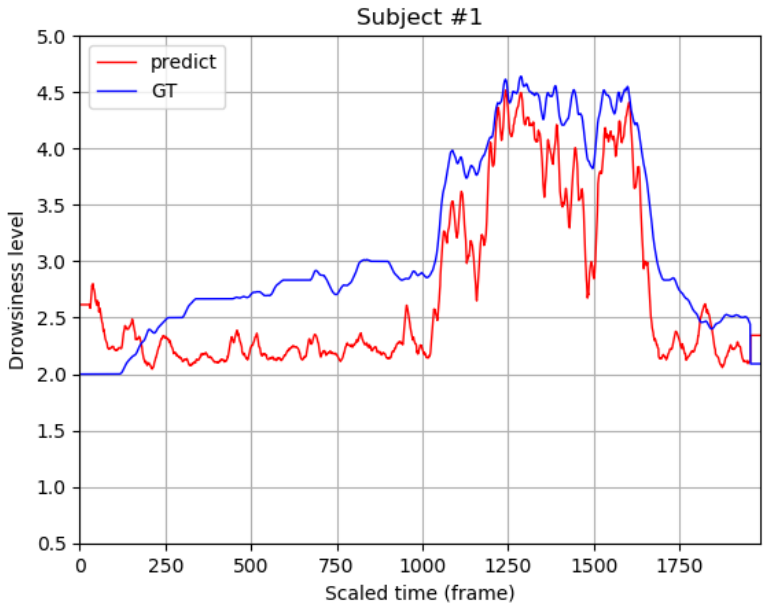

図 5.7: 予測結果の時系列グラフ：横軸は時刻，縦軸は眠気レベルを示す．青は正解の眠気レベル，赤は提案手法が推定した眠気レベルを示す．

■ 感度マップによる可視化

SmoothGrad[87] を用いて作成した入力特徴量に関する感度マップを図 5.9 に示す．提案手法と Time-domain CNN の 2 種類の感度マップを示す．寒色系は低い感度，暖色系は高い感度を示し，感度が高いほど該当する特徴量が出力の眠気レベルに影響を与えることを示す．横軸は入力特徴量を示し，左から順に目と瞳孔の中心座標，目の高さ，目の幅，t が 10 秒の時間特徴量，t が 20 秒の時間特徴量となる．縦軸は時刻を示し，上から (t - 1740) 番目のフレーム，(t - 1680) 番目のフレーム，$\cdots$，t 番目のフレームを示す．上段は弱い眠気，下段は強い眠気の感度マップを示す．左は提案手法の Parallel Linked Time-domain CNN，右は Time-domain CNN の感度マップを示す．

目と瞳孔の中心座標の感度は少しあるが，目状態は感度が低い．一方，目の幅と高さは最も感度が高い．ドライバの目の幅は実際には変化しないが，本研究で推定した目の幅は目の高さと強い相関がある．そのため，目の幅の感度が高くなっている．時間特徴量では，瞬き頻度よりも AECT の方が感度が高い．これらの時間特徴量は似た性質を持っているが，本研究では 12FPS の低フレームレートを用いているため，低フレームレートでも特徴抽出が可能な AECT の感度が高くなったと考えられる．PERCLOS と Soft PERCLOS はどちらも感度はそれほど高くない．また，提案手法では 10 秒と 20 秒の 2 種類の t を用いて時間特徴量を計算しているが，どちらも同程度の感度となっている．既存手法の Time-domain CNN では t-870 付近の局所的な時刻のみ感度が高い．一方，提案手法の Parallel Linked Time-domain CNN では，広範囲の時刻で感度が高い．従って，提案手法の Parallel Linked Time-domain CNN は，複数の時間解像度の特徴量を抽出できる．

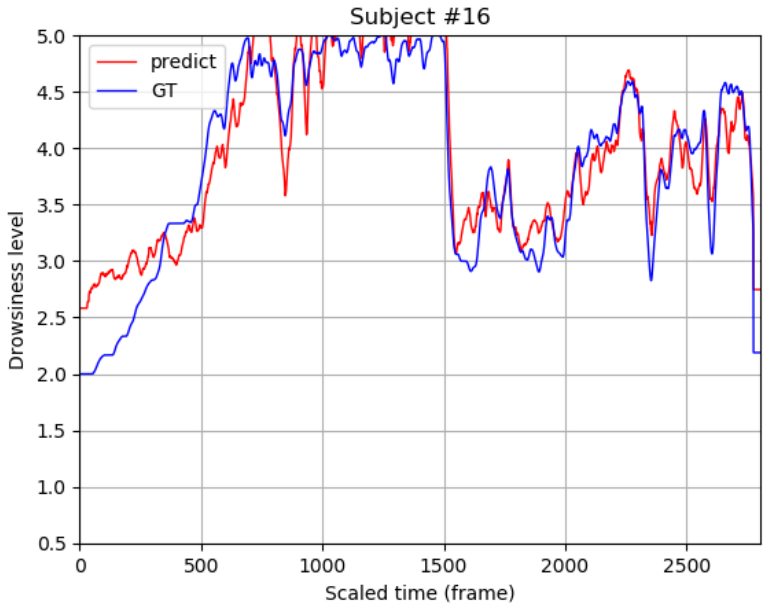

図 5.8: 予測結果の時系列グラフ

5.3.5 眠気の早期検知

ドライバの強い眠気は，深刻な自動車事故につながる．DMS がドライバの弱い眠気を検知でき，弱い眠気から強い眠気に移行するまでの時間が長い場合，検知から事故が起こるまでの時間が長くなるため，システムは事故を回避するために様々な手段を取ることができる．

本研究では，眠気の早期検知に関する 2 つの実験を示す．1 つ目は眠気レベル毎の評価，2 つ目は低い眠気レベルから高い眠気レベルに移行するまでの時間に関する統計情報である．

■ 眠気レベル毎の評価

眠気レベルの毎の評価結果を表 5.6 に示す．実験には提案手法の入力特徴量とネットワークモデルを使用する．提案手法では，眠気レベル 5 や 4 だけでなく，眠気レベル 3 も高精度に推定できている．従って，提案手法では弱い眠気を検知することができ，眠気の早期検知も可能である．その他の眠気レベルに比べて，眠気レベル 1-2 の精度が低い．これは眠気レベル 1-2 のドライバは，会話をしている最中に笑ったりすることで，眠気レベルが高い状態に誤判定される．これらの誤判定は，目以外の手や口などの特徴も活用することで，改善できる可能性がある．

■ 眠気レベルの遷移時間

低い眠気レベルから高い眠気レベルへの遷移時間の統計を表 5.7 に示す．表より，低い眠気レベルから高い眠気レベルに遷移するまで，少なくとも 110 秒以上かかることがわかる．従って，弱い眠

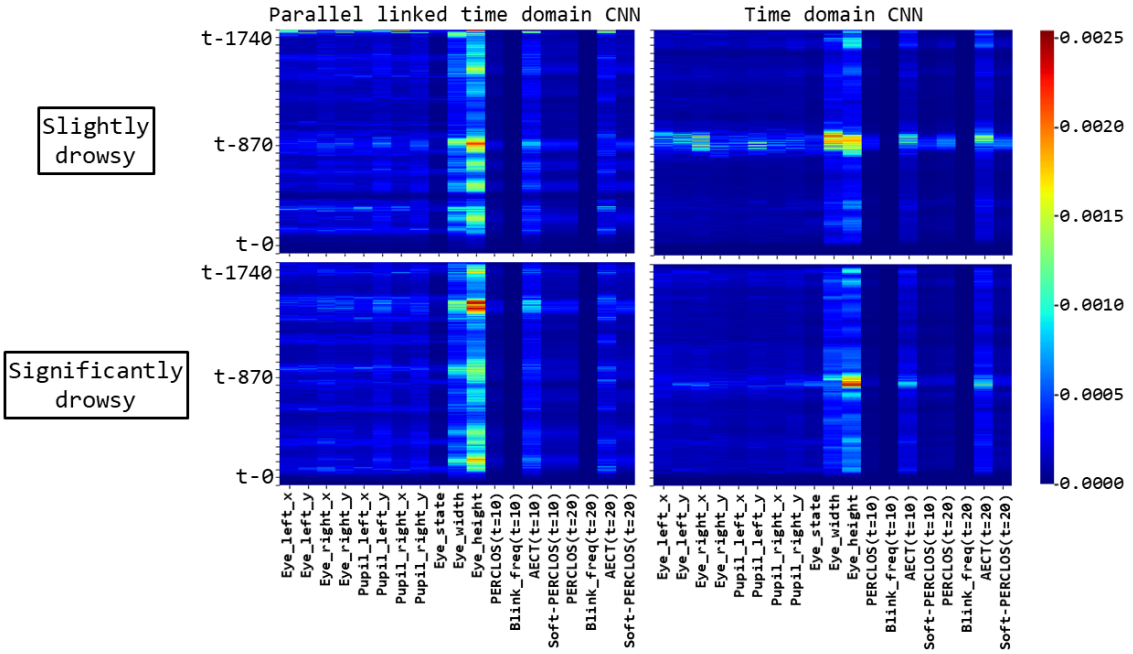

図 5.9: 入力特徴量に対する感度マップ

表 5.6: 眠気レベル毎の評価結果

Drowsiness level	精度（M=1.0）	MAE
レベル 1-2：alert, slightly drowsy	86.16%	0.6112
レベル 3：moderately drowsy	**97.32%**	**0.3159**
レベル 4：significantly drowsy	94.50%	0.4470
レベル 5：extremely drowsy	96.01%	0.3936

気を検知したのち，弱い眠気のドライバを冷風などの弱い刺激で快適に起こせる可能性がある．

■ 未来時刻の強い眠気レベルの検出

未来時刻の強い眠気レベルを検出することは，ドライバの眠気の早期検知につながる．本節では表情評定により付与された正解の眠気レベルを 4 未満と 4 以上で 2 値化したのち，現在の入力データから，2 分後の 2 値ラベルを推定する．本実験では Parallel Linked Time-domain CNN を用いて，2 分後の 2 値ラベルを推定する．また，これまでの実験では 360 フレームを入力としていたが，本実験では精度向上のため，480 フレームを入力とする．未来時刻の強い眠気レベルを推定した場合の精度は 92.12%となった．精度はしきい値 M を 1.0 とした場合の結果となる．本実験でモデルが推定した 2 値ラベルと正解の 2 値ラベルをプロットした結果を図 5.10，図 5.11 に示す．横軸がフレーム番号，縦軸が眠気レベルの 2 値ラベルとなる．一部の箇所で誤検出があるものの，多くの箇所で未来時刻の強い眠気を推定できている．

表 5.7: 眠気レベルの遷移時間

眠気レベル	平均の遷移時間 $[sec]$	最大遷移時間 $[sec]$	最小遷移時間 $[sec]$
From レベル 3：moderately drowsy to レベル 4：significantly drowsy	**375.93**	**768.06**	**113.3**
From レベル 4：significantly drowsy to レベル 5：extremely drowsy	1299.34	4830.10	135.65

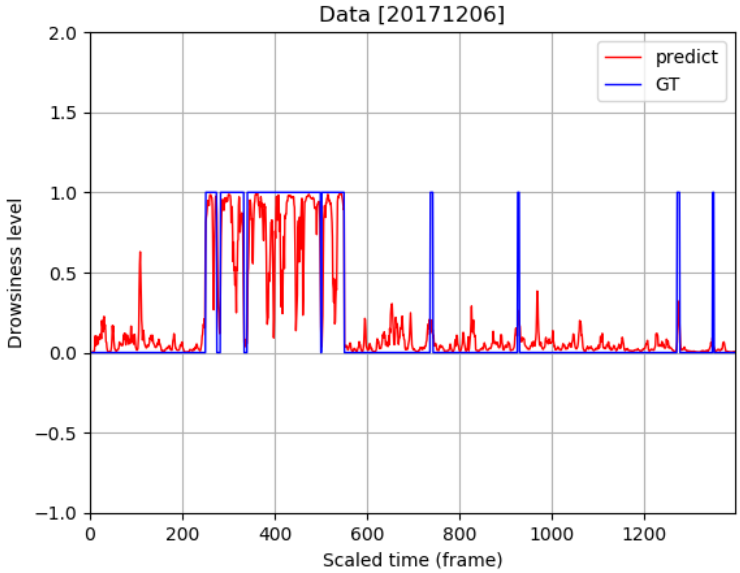

図 5.10: 未来時刻の強い眠気レベルの推定結果

5.4 表情評定方法の検証

本研究では北島らの表情評定法 [55] で付与した眠気レベルを正解データとして用いている．表情評定を用いることで眠気レベルを定量的に扱えるが，主観的な評定であるため，評定結果が安定しないことがある．

そのため，本研究では表情評定を下記 3 つの観点で検証する．

- 評定者による評定結果のバラつき
- 評定結果の再現性.（同じデータを同一の評定者が評定した場合に，評定結果が一致するか.）
- 評定結果の時間的なバイアス.（時系列に評定した結果と順序をランダムに入れ替えて評定した結果が一致するか.）

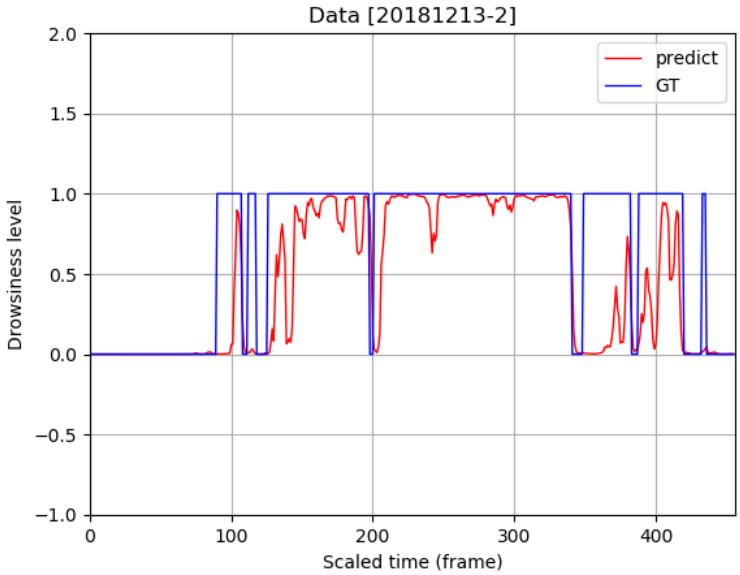

図 5.11: 未来時刻の強い眠気レベルの推定結果

5.4.1 評定者による評定結果のバラつき

本研究では 3 人の評定者により表情評定を行う．評定者による結果のバラつきがあるかどうかを確かめるため，各評定者がつけた評定結果と，3 人の評定結果の平均を混同行列とする．混同行列を可視化したものを図 5.12 に示す．図より，評定結果の平均が 1，3，4，5 付近の結果ではバラつきが小さい．一方，評定結果の平均が 2 付近の場合にはバラつきが大きく，信頼性が低い評定結果であることがわかる．そのため，本研究では，眠気レベル 1 と 2 を区別しない 4 段階の眠気レベルを用いる．

5.4.2 評定結果の再現性

評定結果に再現性があるかどうかを検証するため，同じデータに対して表情評定を 2 回行う．初回と 2 回目の評定結果を混同行列にしたものを，図 5.13 に示す．混同行列の各セルは，上段が該当のデータ数，下段がその割合を示す．また，各評定結果を時系列にプロットしたグラフを図 5.14 に示す．横軸がフレーム番号，縦軸が各評定結果の眠気レベルを示す．各色は各評定者の評定結果を示す．混同行列より，評定者のバラつきと同様に眠気レベルが 2 付近の時に最も再現性が低い．

5.4.3 評定結果の時間的バイアス

表情評定では動画を 5 秒間のクリップに分割し，各クリップに対して眠気レベルを付与する．表情評定では時系列順に評定するため，過去に評定した動画の情報が評定結果に影響する可能性が高い．そこで本節では，時系列順に評定した結果と，評定対象のクリップをランダムに入れ替えた評定

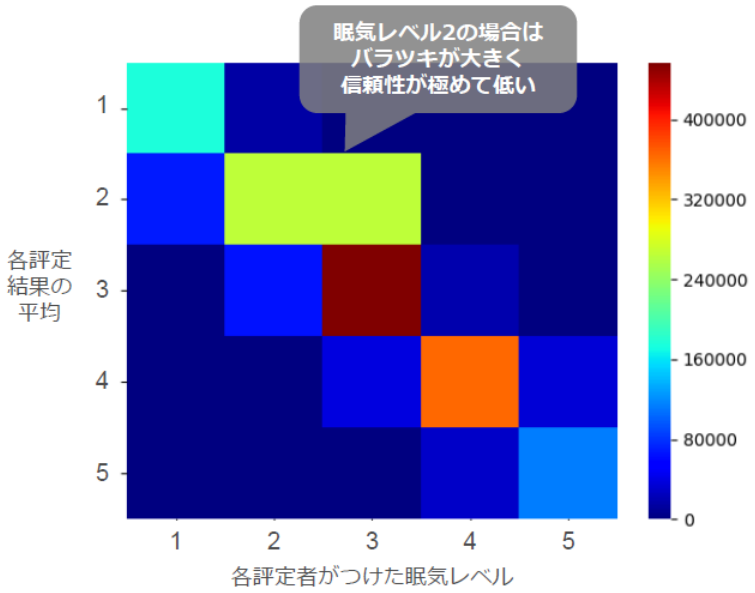

図 5.12: 評定結果の混同行列

結果を比較する．比較結果を図 5.15 に示す．横軸がフレーム番号，縦軸が評定結果の眠気レベルを示す．青色が時系列順の評定結果，黄色がランダムに入れ替えた評定結果を示す．時系列順の評定結果と比べて，ランダムに入れ替えた評定結果は眠気レベルが不安定である．従って，表情評定は時間的なバイアスの影響が強い．提案手法では，30 秒分の入力データに対して，時間方向の畳み込みによる時間変化を捉える．そのため，表情評定の時間的バイアスも含めて，眠気レベルの特徴を学習し，高精度な推定結果を得られたと考えられる．

5.5 まとめ

本研究では画像から弱い眠気を含んだマルチレベルの眠気推定を提案した．本研究の提案手法として，弱い眠気レベルを検出するのに役立つ時間特徴量である AECT 及び Soft PERCLOS と，複数の時間解像度の特徴を抽出するネットワークモデル Parallel Linked Time-domain CNN を提案した．また，提案手法の時間特徴量とネットワークモデルが既存手法よりも精度が高いことを実験により示した．提案手法では しきい値を 1.0 とした場合の精度が 95.86%，MAE が 0.4007% となり，高い精度で眠気レベルを推定できることを示した．感度マップを用いて，提案モデルの Parallel Linked Time-domain CNN が複数の時間解像度の特徴量を抽出できることを示した．最後に実験より，提案手法が眠気の早期検知に役立つことを示した．本研究では目に関する特徴量のみを用いて眠気推定を行ったが，今後は口や顔の皺など目以外の特徴量を用いて更に高精度な眠気推定に取り組む．

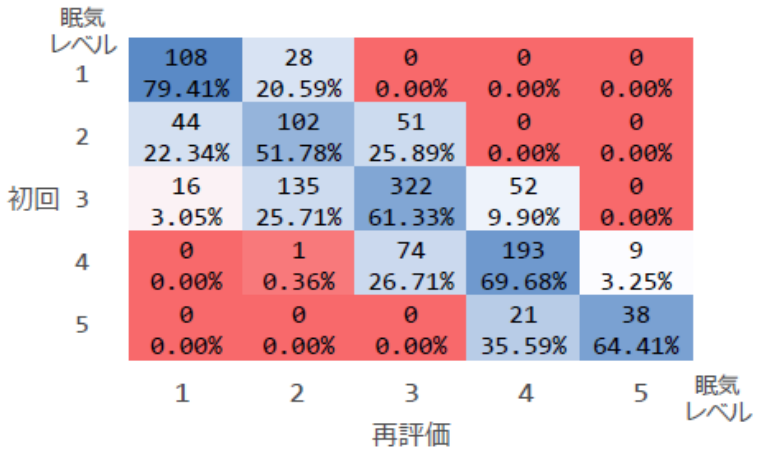

図 5.13: 再現性の検証 (混同行列)

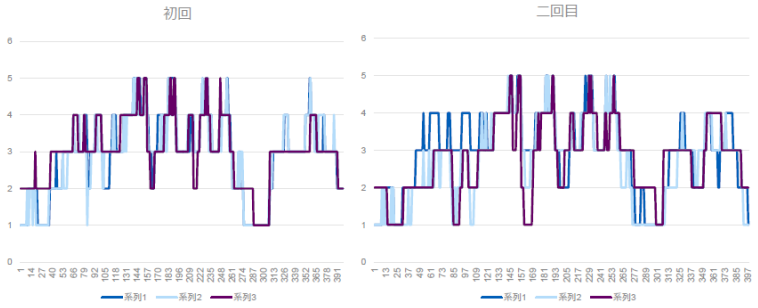

図 5.14: 再現性の検証 (グラフ)

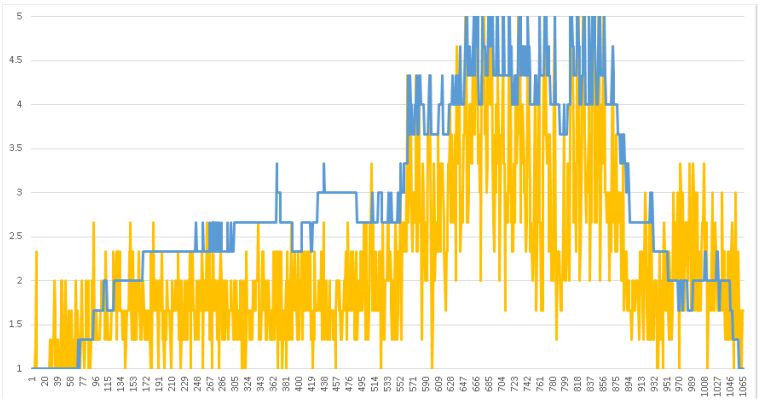

図 5.15: 時間的バイアスの検証

第6章

Sequential Layer-wise Knowledge Distillationを用いたネットワークのコンパクト化

Deep Neural Network(DNN) は画像，音声，化合物など様々な分野のコンペティション [90][91] でこれまでの記録を塗り替えるほどの高精度化を実現し，注目されている．しかし，DNN は計算量が多く，メモリ消費量が大きいため，組み込み機器に実装することが難しい．そのため，精度を下げることなく，モデルサイズを小さくする高速化・省メモリ化手法が提案されている [4]-[92]．DNN の高速化・省メモリ化手法は下記 2 つに分類できる．

1. 再学習なしの高速化・省メモリ化
   (パラメータ削減)
2. 再学習ありの高速化・省メモリ化
   (ネットワーク再構築)

再学習なしの高速化･省メモリ化は，重要度の低いノードを削除する枝刈り法 [4, 5, 6, 7, 8] や，ノードの重みの共有化法 [8, 9] などがある．再学習なしの高速化・省メモリ化は，既存のネットワークに対して，縮小やメモリ消費の効率化を図る手法であるため，ネットワークの構成自体を変更することが難しい．一方，再学習ありの高速化･省メモリ化は，ネットワークの構成を柔軟に変更することが可能である．本研究では，ネットワークの構成を柔軟に変更することが可能な再学習ありの高速化・省メモリ化を対象とする．

再学習ありの高速化･省メモリ化は，Model Compression[93]，Soft Target による Knowledge Distillation[94] 以降，近年盛んに研究されている [93, 95, 94, 96, 97, 92]．Soft Target を用いた Knowledge Distillation では，大きな教師ネットワークの出力値をまねるように小さな生徒ネットワークを学習することで，精度を下げることなく，小さな生徒ネットワークを構築する．一方で，従来の Knowledge Distillation は中間層の特徴表現が似ているかは考慮していない．

そこで本研究では，教師ネットワークと生徒ネットワークの中間層の特徴についても着目し，より高精度な Knowledge Distillation を行うために，下位層から順に Knowledge Distillation を行う Sequential Layer-wise Knowledge Distillation を提案する．

6.1 関連研究

Knowledge Distillation の関連研究は，下記 2 つに分類することが出来る．

1. 生徒ネットワークの構造を任意に設定可能な柔軟性のある手法
2. 中間層の情報に着目した手法

1 つ目は，Knowledge Distillation を用いてコンパクト化する生徒ネットワークの構造を任意に設定可能である．1 つ目に該当する関連研究のうち，いくつかの関連研究では，CNN にしか適用できない．また，教師ネットワークと生徒ネットワークの特徴マップの数を一部同じ数にする必要がある．このように生徒ネットワークの構造に制限がある．

2 つ目は，教師ネットワークの出力値だけではなく中間層の情報も生徒ネットワークの学習に使用している．Hinton らの手法 [94] では，教師ネットワークの出力値のみ用いて生徒ネットワークの学習を行っていたが，これは，中間層の情報を生徒ネットワークの学習に使用することで，より多くの情報を生徒ネットワークの学習に活かすことができる．

6.1.1 柔軟性のある手法

Caruana らは，アンサンブル学習により生成された強識別器の出力をヒントとして，ネットワークを学習することで，処理量やメモリ消費量を削減する Model Compression を提案している [93]．Model Compression にて構築するネットワークは，中間層が 1 つだけの比較的浅いネットワークに限定されている．さらに，Caruana らは，Model Compression を DNN に適用する手法も提案している [95]．これは，高精度で層の深い教師ネットワークと層の浅い生徒ネットワークの出力値の L2 ノルムを最小化するように生徒ネットワークを学習することで，層の深い教師ネットワークと同等精度の生徒ネットワークを構築している．

Hinton らは，教師ネットワークと生徒ネットワークの出力を滑らかにした Soft Target[94] を用いて，Knowledge Distillation を効果的に行い，高精度かつ小さな生徒ネットワークを学習している．教師ネットワークの出力を滑らかにすることで，教師ネットワークの不正解クラスの情報を生徒ネットワークに伝えている．Soft Target を用いた Knowledge Distillation のように，教師ネットワークの情報をヒントとして生徒ネットワークを如何に上手く学習させるかに着目した研究が提案されている [13]-[16]．

6.1.2 中間層に着目した手法

Romero らは，Soft Target では教師ネットワークの中間層の情報を生徒ネットワークの学習に活かせていないと指摘している [96]．しかし，教師ネットワークの中間層と，生徒ネットワークの中間層は幅（パラメータ数）が異なるため，中間層に対して，Knowledge Distillation を行うことは出来ない．Romero らは，生徒ネットワークの 1 つの中間層に，教師ネットワークの中間層と同じチャネル数（ユニッ

ト数）の Regressor と呼ばれる層を新たに付け足すことで，中間層に対しても Knowledge Distillation を行う FitNets[96] を提案している．FitNets を用いることで，Hinton らの Knowledge Distillation よりも高精度で小さな生徒ネットワークの学習に成功している．

Junho らは，2 つの中間層間の関係を学習する Gift を提案している [97]．これは，2 つの中間層の出力 (特徴マップ) を FSP matrix と呼ばれる行列に変換することで，2 つの中間層間の関係を 1 つの行列として表現している．教師ネットワークと生徒ネットワークのそれぞれで，2 つの層の特徴マップの内積から計算される FSP matrix を生成し，FSP matrix 間の L2 ロスを計算することで，中間層の情報を活かした生徒ネットワークの学習を行っている．

Sergey らは，畳み込み層の出力 (特徴マップ) を attention map とし，教師ネットワークと生徒ネットワークの attention map に対して中間層の Knowledge Distillation を行う Attention Transfer[92] を提案している．この手法は，特徴マップから attention map を抽出するため，適用対象は CNN のみに限定されている．

6.1.3 関連研究の課題

本稿にて引用した関連研究の特徴を，表 6.1 にまとめる．

表 6.1: 関連研究の特徴

手法	柔軟性	中間層
Model Compression[93]	✓	✗
Do deep nets...?[95]	✓	✗
Soft Target[94]	✓	✗
FitNets[96]	✓	✓
A Gift from ...[97]	✗	✓
Attention Transfer[92]	✗	✓

関連研究には，2 つの課題がある．一つ目は，[93]-[95] では，生徒ネットワークを学習する際に教師ネットワークの中間層の情報を使用していないことである．これらの関連研究では中間層の情報を使用していないため，生徒ネットワークの学習に教師ネットワークの情報を活かしきれていない．2 つ目は，生徒ネットワークの構造に制約があることである．Junho らの手法 [97] は，FSP matrix を計算するため，生徒ネットワークは任意のサイズにすることが出来ない．また，Junho らの手法 [97] と Attention Transfer[92] は CNN に対してのみ適用可能である．FitNets は中間層と柔軟性のどちらの課題も克服しているが，構築した生徒ネットワークの精度が高いとは言えない．

6.2 提案手法

本研究では, Sequential Layer-wise Knowledge Distillation による精度低下を抑えたネットワークのコンパクト化手法を提案する．

6.2.1 Soft Target を用いた Knowledge Distillation

提案手法で行う Knowledge Distillation は，Soft Target[94] をベースに行う．Soft Target では，下式のように生徒ネットワークの出力値と教師ネットワークの出力値を滑らかにした Soft Target から計算される誤差 $H(P_T^\tau, P_S^\tau)$ と，生徒ネットワークの出力値と教師信号 (Hard Target) から計算される誤差 $H(y_{true}, P_S)$ を混合する．その際，パラメータ λ によって混合する割合を調整する．P_S, P_T はそれぞれ，生徒ネットワーク，教師ネットワークの出力を滑らかにした Soft Target である．また，y_{true} は Hard Target である．

$$L_{KD}(W_S) \quad = \quad H(y_{true}, P_S) + \lambda H(P_T^\tau, P_S^\tau) \tag{6.1}$$

Soft Target を計算する際，生徒ネットワークの出力値を滑らかにするために，温度パラメータ T を使用する．また，出力層でソフトマックス関数を用いる代わりに，下式を用いてネットワークの出力値を計算する．

$$p_i = \frac{exp(\frac{z_i}{T})}{\sum_j exp(\frac{z_j}{T})} \tag{6.2}$$

提案手法では，複数の層に対して Knowledge Distillation を行うため，各層に対して個別の λ と温度 T を設定する．また，提案手法では，中間層と出力層の両方に対して，Knowledge Distillation を行うが，中間層では誤差関数 H として L2 誤差，出力層では Cross Entropy を用いる．

6.2.2 Sequential Layer-wise Knowledge Distillation

提案手法の概要を図 6.1 に示す．I，i は教師ネットワークと生徒ネットワークへの入力，L^1〜L^n，l^1〜l^n は中間層，O，o は出力層を示す．Knowledge Distillation を行うために，あらかじめ教師ネットワークを学習する．次に，教師ネットワークの中間層を利用して Knowledge Distillation を行うために，教師ネットワークの中間層に対して出力層を追加する．そして，追加した出力層の重みを学習する．この際，重み更新に使用する学習データは，教師ネットワークの学習に使用したものと同じデータとする．その後，中間層の Knowledge Distillation により，生徒ネットワークの中間層の重みを学習する．最後に，出力層の Knowledge Distillation を行い，生徒ネットワークの全層の重みをファインチューニングする．以降で，各手順の詳細について，説明する．

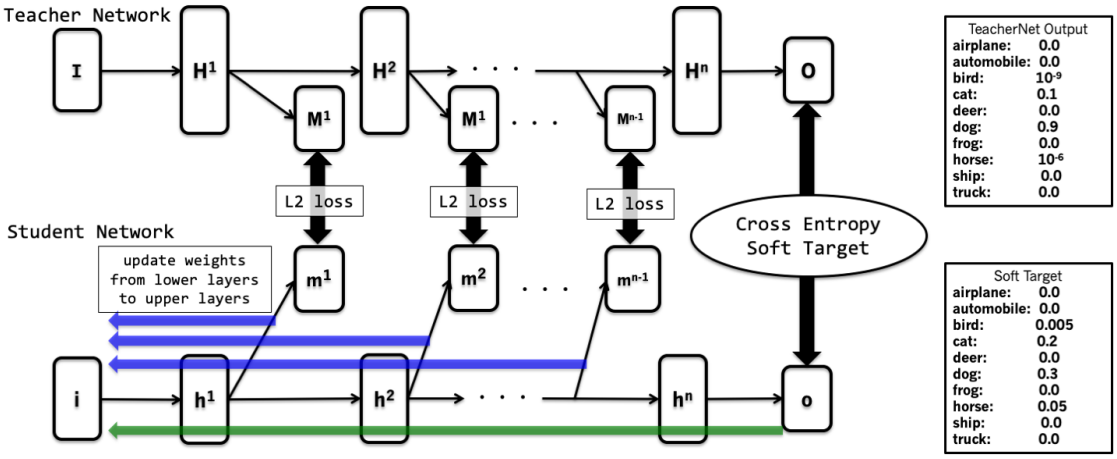

図 6.1: Sequential Layer-wise Knowledge Distillation の概要：新たな出力層 (M^1～M^{n-1}, m^1～m^{n-1}) を教師・生徒ネットワークの中間層に追加する．I，i は教師ネットワークと生徒ネットワークへの入力を，H^1～H^n，h^1～h^n は中間層を，O，o は出力層を示す．追加した出力層を用いて教師ネットワークの中間層の情報を生徒ネットワークの学習に用いる．最後に，SoftTarget による Knowledge Distillation を行い，全層の重みをファインチューニングする．

■ 教師ネットワークの学習

はじめに教師ネットワークの学習を行う．教師ネットワークは生徒ネットワークよりもパラメータが多いネットワークであり，教師ネットワークと生徒ネットワークは同じ層数である必要はない．また，教師ネットワークは学習済みのネットワークを用いても良い．

■ 教師ネットワークの出力層追加

教師ネットワークの各中間層 (L^1～L^{n-1}) に，出力層 (M^1～M^{n-1}) を追加した後，教師ネットワークの学習に使用したデータセットを用いて再学習を行い，新たに追加した出力層の重みを学習する．この再学習の際，教師ネットワークの中間層 (L^1～L^{n-1}) の重みは更新しない．また，追加した出力層 (M^1～M^{n-1}) の重みだけを学習するため，学習エポック数は，教師ネットワークを学習した際のエポック数の半分程度としている．これは，実験により教師ネットワークの学習に用いたエポック数の半分程度で，追加した出力層の重みを学習できることを確認したため，半分程度としている．

■ 生徒ネットワークの中間層の学習

次に，Sequential Layer-wise Knowledge Distillation による生徒ネットワークの学習を行う．提案手法では，生徒ネットワークの中間層の Knowledge Distillation を下位層から順に行う．ただし，FitNets のように，中間層の出力に対して Knowledge Distillation を行うのではなく，教師ネットワークの各

中間層に追加した出力層の値を用いて, SoftTarget による Knowledge Distillation を行う．図 6.1 に示すように，生徒ネットワークの中間層 (L^1〜L^{n-1}) に対しても出力層 (m^1〜m^{n-1}) を追加する．教師ネットワークの出力層 (M^1〜M^{n-1}) は既に学習済みのため，教師ネットワークの出力層 (M^1〜M^{n-1}) の出力値を用いて，Soft Target による Knowledge Distillation が可能となる．この処理を中間層の 1 層，またはブロックなどの特定の単位に対して下位層より順に行う．

FitNets では，1 つの中間層に対してのみ Knowledge Distillation を行う．これは，中間層の各層に対して Knowledge Distillation を行うと，正則化が強すぎるために，精度がかえって低下してしまう [96]．この問題を避けるため，提案手法では，中間層に追加した出力層の出力値を用いて中間層の Knowledge Distillation を行う．提案手法である Sequential Layer-wise Knowledge Distillation にて中間層の Knowledge Distillation を行うことで，上記の問題を回避することができる．

Zeiler らは，ネットワークの各層が階層的な特徴表現を担っていることを示した [98]．例えば，顔検出を行うためのネットワークの場合，下位層ほど，エッジなどの単純な特徴を抽出する役割を担っており，上位層は目や口などのより複雑な特徴を抽出する役割を担っていることになる．また，Bengio らは，単純なものを最初に学習し，その後徐々に複雑なものを学習する Curriculum Learning と呼ばれる学習手法が精度を向上させることを示した [99]．我々の提案手法である Sequential Layer-wise Knowledge Distillation も，下位層から順に上位層に向かって中間層の Knowledge Distillation を行うため，単純な特徴を抽出する下位層から学習していき，その後徐々に複雑な特徴を抽出する上位層を学習していくため，Curriculum Learning と同様の学習であると考えられる．

■ 生徒ネットワークの学習

提案手法である Sequential Layer-wise Knowledge Distillation により，生徒ネットワークの中間層の重みを逐次学習する．そして，全ての層の重みの最適化を行うため，最後に出力層 (O，o) の値（Hard Target, Soft Target の両方）を用いた Knowledge Distillation を行い，全ての層の重みをファインチューニングする．

6.3 評価実験

提案手法の有効性を確認するため，既存手法との精度の比較を行う．また，中間層の Knowledge Distillation を行うブロック数を最上位ブロックのみにした場合，精度がどのように変化するか評価を行う．

6.3.1 実験データ

実験データには，CIFAR-10/100[100] の 2 種類のデータセットを用いて評価を行う．CIFAR-10/100 は，解像度 32×32 ピクセル, RGB チャンネルの 60,000 枚の画像から構成されている．そのうち，50,000 枚を学習用，10,000 枚を評価用として用いる．また，CIFAR-10 は 10 種類のカテゴリ，CIFAR-100

は 100 種類のカテゴリを分類するタスクである．

表 6.2: モデルのパラメータとサイズ [MB]

モデル	Teacher	Student (VGG1/2)	Student (VGG1/4)	Student (VGG1/8)	Student (VGG1/16)	Student (VGG-7)
モデル構成	B(2, 64) B(2, 128) B(3, 256) B(3, 512) B(3, 512) FC(512) BN OUT	B(2, 32) B(2, 64) B(3, 128) B(3, 256) B(3, 256) FC(512) BN OUT	B(2, 16) B(2, 32) B(3, 64) B(3, 128) B(3, 128) FC(512) BN OUT	B(2, 8) B(2, 16) B(3, 32) B(3, 64) B(3, 64) FC(512) BN OUT	B(2, 4) B(2, 8) B(3, 16) B(3, 32) B(3, 32) FC(512) BN OUT	B(1, 64) B(1, 128) B(1, 256) B(1, 512) B(1, 512) FC(512) BN OUT
モデルサイズ	47.67	12.20	3.19	0.88	0.26	16.7

6.3.2 評価実験のパラメータ

本実験ではベースネットワークとして，VGGNet[101] を用いる．実験に使用した教師ネットワークと生徒ネットワークのパラメータを表 6.2 に示す．

表 6.3: CIFAR-10 の精度

手法	VGG 1/2	VGG 1/4	VGG 1/8	VGG 1/16	VGG -7
Teacher	93.77%	93.77%	93.77%	93.77%	93.77%
Student	92.41%	89.94%	85.85%	74.51%	91.74%
Student, Soft Target [94]	92.64%	90.31%	86.42%	76.68%	91.78%
Student, FitNets [96]	92.17%	90.08%	85.11%	75.83%	91.25%
Student, Gift[97]	-	-	-	-	90.24%
Student, Attention [92]	92.96%	90.77%	85.01%	76.36%	91.86%
Student, 提案手法	**93.11%**	**90.85%**	**86.78%**	**78.61%**	**92.64%**

B は VGG-block を示し，block は畳み込み層，Batch Normalization 層より構成される．活性化関数には，ReLU を使用する．B の引数は (層の数, 畳み込み層のカーネル数) を示している．本実験で使用した畳み込み層のカーネルサイズは 3×3 を使用した．なお，各 block の最後には，2×2 の Max Pooling を使用する．FC は全結合層，引数はユニット数を示している．BN は Batch Normalization[37]

層，OUT は出力層を示している．実験に使用した生徒ネットワークはそれぞれ，VGG-block のカーネル数を 1/2, 1/4, 1/8, 1/16，に減らした 4 通りとする．また，それ以外の生徒ネットワークとして，層の数を 7 に減らしたネットワークも使用する．表 6.2 で示しているモデルサイズは，重みを 32bit の float 型変数として計算している．

全ての実験において，最適化手法に Momentum SGD を使用する．学習率は，0.1 から開始し，150 エポック目，215 エポック目で，0.01, 0.001 に減らし，300 エポックで学習を終了する．中間層に追加した出力層の重みを学習する際のエポック数は，半分の 150 エポックとする．重み減数は 0.0001，バッチサイズは 128 とする．また，全ての実験において，中間層の Knowledge Distillation に使用する λ は 1.0，温度 T は 1.0 を使用する．出力層の Knowledge Distillation に使用する λ は 1.0，温度は CIFAR-10 を用いた実験 では 5，CIFAR-100 を用いた実験では 6 を使用する．中間層に追加した出力

表 6.4: CIFAR-100 の精度

手法	VGG 1/2	VGG 1/4	VGG 1/8	VGG 1/16	VGG -7
Teacher	73.90%	73.90%	73.90%	73.90%	73.90%
Student	70.28%	65.01%	55.43%	35.97%	70.17%
Student, Soft Target [94]	70.61%	64.73%	**56.42%**	36.24%	70.57%
Student, FitNets [96]	67.96%	62.85%	55.55%	37.40%	67.95%
Student, Gift[97]	-	-	-	-	67.07%
Student, Attention [92]	70.36%	**65.57%**	55.25%	33.44%	68.66%
Student, 提案手法	**71.08%**	65.48%	55.83%	**38.57%**	**71.09%**

値にはノイズが混じるため，中間層の Knowledge Distillation をする際，温度 T を高くするとノイズの影響が大きくなってしまう．従って，中間層の Knowledge Distillation に使用する温度 T は低い値を使用する．

6.3.3 生徒ネットワークの精度比較

既存研究として，Soft Target[94]，FitNets[96]，Attention transfer[92] の 3 種類を実験した．Gift[97] は適用できる生徒ネットワークの構造に制限があるため，教師ネットワークから深さを変更した VGG-7 に対してのみ実験を行った．

■ **CIFAR-10**

CIFAR-10 を用いた評価結果を表 6.3 に示す．

表 6.3 より，Soft Target[94] は全ての生徒ネットワークにおいて精度が向上している．FitNets[96] は，VGG 1/16 のような小さな生徒ネットワークに対しては Soft Target よりも高い精度になっているが，それ以外の生徒ネットワークでは精度が低下している．これより，FitNets は比較的小さな生徒ネットワークのみに有効であることがわかる．提案手法では，全てのネットワークに対して，既存研究よりも高い精度となっている．また，VGG 1/2 では教師ネットワークの精度が 93.77%であるのに対して，提案手法を適用した場合の生徒ネットワークの精度が 93.11%となっている．従って，提案手法では精度低下を約 0.6%に抑えつつ，モデルサイズを約 1/4 まで減らすことができる．

■ **CIFAR-100**

CIFAR-100 を用いた評価結果を表 6.4 に示す．

提案手法では，VGG 1/2, VGG-7 のような比較的大きなネットワークに対しては，既存研究よりも高い精度となっている．VGG 1/16 に対しても既存研究より高い精度となっているが，VGG 1/4, 1/8 のような比較的小さなネットワークに対しては，既存研究の方が精度が高くなることがある．提案手法では中間層の Knowledge Distillation を行うが，実験に使用するデータセットの難易度に対して生徒ネットワークのモデルサイズが小さい場合は，中間層に追加した出力層の情報が信頼性の低いものとなるため，誤った情報を生徒ネットワークに繰り返し伝えてしまう．そのため，比較的小さなネットワークでは既存研究よりも提案手法の精度が低下したと考えられる．VGG 1/2 では教師ネットワークの精度が 73.90%であるのに対して，提案手法を適用した場合の生徒ネットワークの精度が 71.08%となっている．従って，提案手法では精度低下を約 2%に抑えつつ，モデルサイズを約 1/4 まで減らすことができる．

6.3.4 中間層の Knowledge Distillation のブロック数の比較

CIFAR-100 を用いた評価実験において，比較的小さなネットワークでは中間層に追加した出力層の情報が信頼性の低いものとなるため，提案手法の精度が低下すると述べた．そこで，本実験では FitNets[96] のように，1 つのブロック（本実験では下から 2 つ目のブロックとした）に対してのみ提案手法を適用した場合，精度がどのように変化するかを確認する．

1 つのブロックのみに中間層の Knowledge Distillation を行った場合と，全てのブロックに対して行った場合の比較結果を表 6.5，表 6.6 に示す．”FitNets 5-loss”は，通常の FitNets が 1 つの中間層のみに Knowledge Distillation を行っているのに対して，提案手法と同じように VGGNet の全てのブロックに FitNets を用いたものである．提案手法は，”提案手法 5-loss”, ”提案手法 1-loss”の 2 種類の実験を行っており，前者が提案手法の Sequential Layer-wise Knowledge Distillation を全てのブロックに対して行った結果，後者が最上位のブロックのみ提案手法を行った結果である．

表 6.5: ブロック数の精度比較（CIFAR-10）

手法	VGG 1/2	VGG 1/4	VGG 1/8	VGG 1/16	VGG -7
Teacher	93.77%	93.77%	93.77%	93.77%	93.77%
Student	92.41%	89.94%	85.85%	74.51%	91.74%
Student, FitNets [96], 1-loss	92.17%	90.08%	85.11%	75.83%	91.25%
Student, FitNets [96], 5-loss	92.13%	89.69%	85.21%	75.47%	90.30%
Student, 提案手法, 1-loss	93.06%	90.76%	**87.38%**	**79.71%**	92.47%
Student, 提案手法, 5-loss	**93.11%**	**90.85%**	86.78%	78.61%	**92.64%**

中間層の Knowledge Distillation を行うブロック数を増やすと正則化が強すぎるため，精度が低下すると記載されているが [96]，その指摘通り，FitNets では精度が高かった VGG 1/16 において，精度が低下している．一方，提案手法では全ての生徒ネットワークにおいて，既存研究よりも高い精度となっている．これは，提案手法では VGG の各ブロックに対して中間層の Knowledge Distillation を行うことで中間層の情報を生徒ネットワークの学習に活かすことができるためであると考えられる．CIFAR-10 を用いた評価実験では，VGG1/2, 1/4, 7 のような比較的大きなネットワークに対しては，中間層の Knowledge Distillation を行うブロック数が多いほうが高い精度となり，VGG1/8, 1/16 のような小さいネットワークに対しては，FitNets と同じように最上位のブロックに対してのみ中間層の Knowledge Distillation を行うほうが精度が高くなる．CIFAR-100 を用いた評価実験でも同様に，VGG1/4, 1/8 のような比較的小さなネットワークに対しては，最上位ブロックに対して中間層の Knowledge Distillation を行うほうが精度が高くなる．

CIFAR-10 を用いた実験では，教師ネットワークの精度が 93.77%になっている．全てのブロックに対して提案手法を用いた VGG 1/2 の精度が 93.11%，最上位のブロックに対して提案手法を用いた場合の精度が 93.06%となっている．どちらの場合も教師ネットワークと比べて，1%未満の僅かな精度低下でモデルサイズを約 1/4 まで減らすことができる．CIFAR-100 を用いた実験では，教師ネットワークの精度が 73.90%となっている．全てのブロックに対して提案手法を用いた VGG 1/2 の精度が 71.08%，最上位のブロックに対して提案手法を用いた場合の精度が 69.57%となっている．どちらの場合も教師ネットワークと比べて，5%未満の精度低下でモデルサイズを約 1/4 まで減らすことができる．

本実験結果より，提案手法は使用するデータセットやネットワークにより，中間層の Knowledge Distillation を行うブロック数が多いほうが高い精度になる場合と，少ない方が高い精度になる場合

表 6.6: ブロック数の精度比較（CIFAR-100）

手法	VGG 1/2	VGG 1/4	VGG 1/8	VGG 1/16	VGG -7
Teacher	73.90%	73.90%	73.90%	73.90%	73.90%
Student	70.28%	65.01%	55.43%	35.97%	70.17%
Student, FitNets [96], 1-loss	67.96%	62.85%	55.55%	37.40%	67.95%
Student, FitNets [96], 5-loss	68.69%	62.65%	53.74%	32.97%	66.55%
Student, 提案手法, 1-loss	69.57%	**65.85%**	**57.16%**	38.56%	**71.16%**
Student, 提案手法, 5-loss	**71.08%**	65.48%	55.83%	**38.57%**	71.09%

があることがわかる．提案手法は中間層の情報を Knowledge Distillation により教師ネットワークから生徒ネットワークに伝える．実験に使用するデータセットの難易度に対して，生徒ネットワークの大きさが小さい場合は，中間層に追加した出力層の情報が信頼性の低いものとなるため，誤った情報を生徒ネットワークに繰り返し伝えてしまう．そのため，提案手法では小さなネットワークに対しては最上位ブロックのみ中間層の Knowledge Distillation を行うほうが精度が高くなる．したがって，小さな生徒ネットワークに対しては中間層の Knowledge Distillation を行うブロック数を減らし，大きな生徒ネットワークに対してはブロック数を増やすことで，高精度な生徒ネットワークを構築できると考えられる．

6.3.5 学習誤差と精度の関係

CIFAR-10, CIFAR-100 を用いた評価実験における各ネットワークの最下位ブロックの誤差を図 6.2 および図 6.3 に示す．横軸はエポック数，縦軸は誤差である．図 6.2 と図 6.3 では 3 種類の誤差の遷移を記載している．loss_hard が Hard Target を用いた場合の誤差，loss_soft が Soft Target を用いた場合の誤差，loss がそれらの誤差の平均となる．CIFAR-10 を用いた実験では，VGG 1/2, 1/4, VGG-7 のような大きなネットワークに対しては，提案手法の 5-loss のほうが 1-loss よりも精度が高かった．図 6.2 より，そのような大きなネットワークでは，Hard Target の誤差が Soft Target の誤差を大きく上回っていることがわかる．CIFAR-100 を用いた実験では，VGG 1/2 のような大きなネットワークに対しては，提案手法の 5-loss のほうが 1-loss よりも明らかに精度が高かった．図 6.3 より，CIFAR-10 の結果と同様に，VGG 1/2 では，Hard Target の誤差が Soft Target の誤差を大きく上回っているがわ

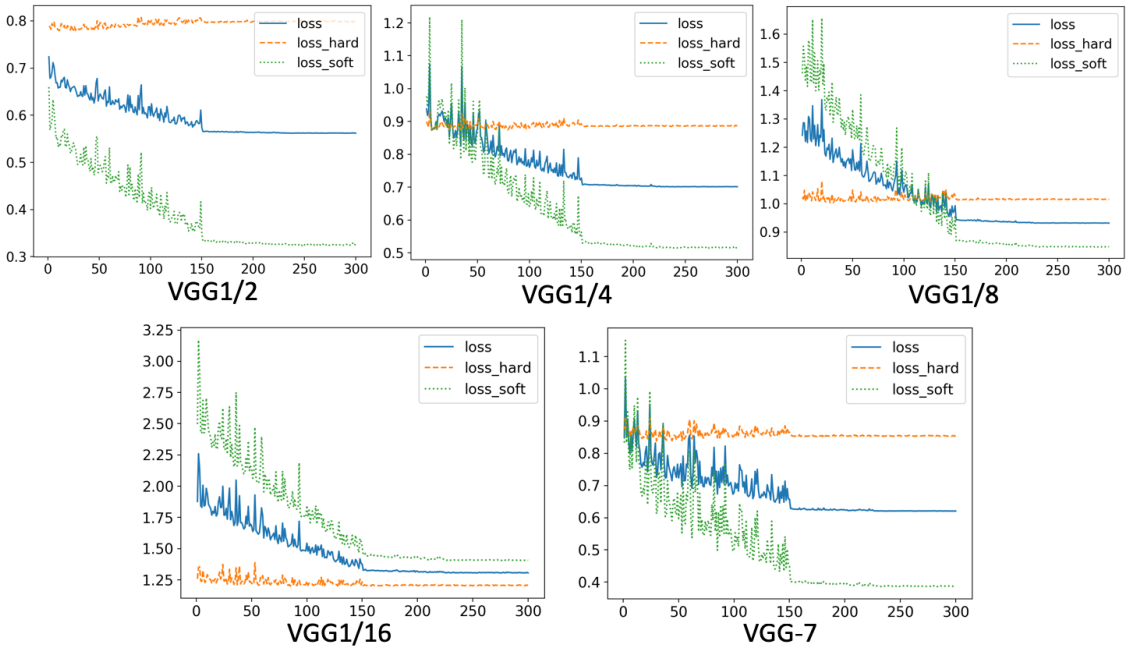

図 6.2: CIFAR-10 を用いた場合の最下位ブロックの誤差：縦軸が誤差の大きさ，横軸がエポック数を示す．誤差は Hard Target を用いた場合の誤差，Soft Target を用いた場合の誤差，それらの誤差の平均，計 3 種類．全てのブロックで中間層の Knowledge Distillation を行った方が精度が高くなる場合には，Hard Target の誤差が Soft Target の誤差よりも大きくなる傾向にある．

かる．

以上の結果より，Sequential Layer-wise Knowledge Distillation を行う際に，最下位ブロックの誤差を確認し，Hard Target の誤差が Soft Target の誤差を大きく上回っている場合には，全てのブロックで中間層の Knowledge Distillation を実施し，そうでない場合には，最上位ブロックのみで中間層の Knowledge Distillation を実施することで，常に精度の高い生徒ネットワークを構築できる．

6.4 結論

本稿では，CIFAR-10/100 を用いた実験にて，提案する Sequential Layer-wise Knowledge Distillation が既存研究よりも高い精度を達成できることを示した．また，提案手法を用いることで，精度低下を 0.6%に抑えたまま，ネットワークのモデルサイズを約 1/4 まで削減できることを確認した．今後は ImageNet などの大規模なデータセットを用いた実験により，提案手法の有効性を示す．更に，提案手法では極端に小さなネットワークに対しては効果が低いため，中間層の Knowledge Distillation を行う際に Soft Target の割合を動的に変化させることで，提案手法をより多様なネットワークに対して，有効になるよう拡張していく．

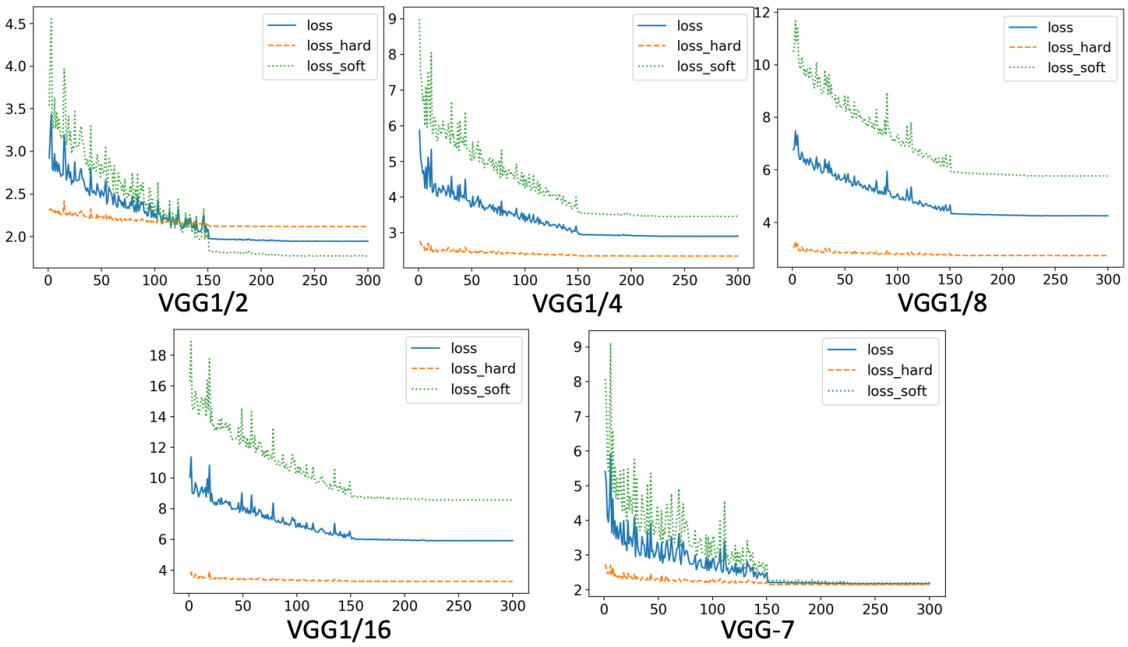

図 6.3: CIFAR-100 を用いた場合の最下位ブロックの誤差

第7章

結論と展望

本研究では，安全で快適な自動車の実現に役立つドライバモニタリング技術として，高速かつ省メモリなドライバ姿勢推定，高速かつ省メモリなドライバ動作認識，弱い眠気を含んだマルチレベルのドライバ眠気推定を提案した．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 章では，自動車の交通事故と交通事故を減らすための国，自治体，民間の取り組みについて述べた後に，ドライバモニタリング技術に関するセンシング方法についてまとめた．

3 章では，自動車内の組込機器に搭載可能な高速かつ省メモリなドライバ姿勢推定に取り組んだ．演算量やメモリ消費量などの消費リソースを削減するため，畳み込み処理の演算量を減らすことが可能な ShuffleNet V2 と，出力ヒートマップのサイズを小さくした際の量子化誤差による精度低下を抑制することが可能な Integral Regression を用いた手法を提案した．また，ドライバモニタリングでは被写体とカメラの距離が近く，ドライバの関節点が画角内に映らないことが多いため，関節点の座標と同時に関節点有無を推定する手法を提案した．ドライビングシミュレータと近赤外線カメラを用いて撮影したデータセットにおいて，演算量を制限した条件で既存手法を 1〜10%程度上回る精度を達成した．

4 章では，自動車内の組込機器に搭載可能な高速かつ省メモリなドライバ動作認識に取り組んだ．消費リソースを削減するため，4 で提案した ShuffleNet V2 と Integral Regression を用いた姿勢推定を活用した動作認識を提案した．また，ドライバの関節点座標，関節点有無，関節点状態の 3 つの姿勢情報と動作のマルチタスク学習を用いて，演算量を制限したネットワークモデルの精度向上に取り組んだ．従来のドライバ動作認識データセットは，電話，食事などの居眠りや発作などの意識を失うような深刻な状態を含んでいない．更に，従来のデータセットは手動運転時のドライバ動作のみを対象としている．そのため，電話や食事などの軽度な状態から居眠りなどの意識を失うような深刻な状態までを幅広くカバーするドライバ動作認識データセットを構築した．更に自動運転車でのドライバ動作もカバーするデータセットとした．このデータセットを用いて演算量を制限した条件で，既存手法を 5%程度上回る精度を達成した．

5 章では，眠気を早期に検知し，快適な運転を実現するため，弱い眠気を含んだマルチレベルのドライバ眠気推定に取り組んだ．弱い眠気を検知するため，Average Eye Closure Time(AECT) と Soft PERCLOS の 2 つの時間特徴量と，複数の時間解像度に着目した特徴量を抽出可能なネットワークモデルである Parallel Linked Time-domain CNN を提案した．表情評定を用いて 4 段階の眠気レベルの GT を付与した実車撮影のデータセットを構築し，提案手法を評価した．単純な目や瞳孔の中心座標のみを入力とした場合と比べて，提案手法は 7%程度高い精度を達成した．また，VGG-LSTM，3D-CNN などの動画像を入力とする既存のネットワークモデルと比べて，30%程度高い精度を達成した．提案したネットワークモデルの Parallel Linked Time-domain CNN が複数の時間解像度に着目した特徴量を抽出できることを，SmoothGrad を用いて作成した感度マップを用いて検証した．

6 章では，自動車内での組込機器に搭載するため，様々なネットワークモデルに対して適用可能な Sequential Layer-wise Knowledge Distillation を用いたネットワークモデルのコンパクト化に取り組んだ．CIFAR-10/100 を用いた評価実験で，提案手法が既存手法よりも高い精度を達成できることを示した．また，提案手法を用いることで，精度低下を 0.6%に抑制したまま，ネットワークのモデルサイズを 1/4 程度にまで削減できることを示した．

7.2 展望

本研究では，深層学習の一つである Deep Convolutional Neural Network を用いて，ドライバの姿勢推定，動作認識，眠気推定を高精度に行えることを示した．これらのドライバモニタリング技術は，漫然運転，脇見運転，安全不確認などによる交通事故を減らすことに役立つ．しかし，動静不注視は歩行者の動きなど外部環境のモニタリング技術が必要となる．また，運転操作不適や安全速度は車体情報の取得が必要となる．自動車の事故を無くすためには，本研究で提案したドライバモニタリングによる車内状況の把握だけでなく，車外や車体の状況まで把握する必要がある．

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，工学部情報工学科 山下隆義准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藤吉弘亘教授，山下准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 山内康一朗教授，慶應義塾大学理工学部電子工学科青木義満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オムロン株式会社技術・知財本部　研究開発センタの皆様に心から厚く御礼申し上げます．また，本研究の一部において，貴重なご意見を頂きました株式会社メガチップスの皆様に心から厚く御礼申し上げます．最後に，研究生活を支え，常に応援してくれた妻に心から感謝します．

参考文献

[1] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields”, CVPR, 2017.

[2] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, “R-cnns for pose estimation and action detection”, arXiv preprint arXiv:1406.5212, 2014.

[3] C. Zhao, B. Zhang, J. He, and J. Lian, “Recognition of driving postures by contourlet transform and random forests”, T-ITS, vol.6, no.2, pp.161–168, 2012.

[4] Y. LeCun, J. S. Denker, S. A. Solla, R. E. Howard, and L. D. Jackel, “Optimal brain damage.”, NIPS, vol.2, pp.598–605, 1989.

[5] B. Hassibi, and D. G. Stork, “Second order derivatives for network pruning: Optimal brain surgeon”, NIPS, vol.5, pp.161–171, 1993.

[6] T. He, Y. Fan, Y. Qian, T. Tan, and K. Yu, “Reshaping deep neural network for fast decoding by node-pruning”, ICASSP, pp.245–249, 2014.

[7] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network”, NIPS, pp.1135–1143, 2015.

[8] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding”, , 2016.

[9] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen, “Compressing neural networks with the hashing trick”, ICML, pp.2285–2294, 2015.

[10] 交通交通企画課，“令和元年中の交通事故死者数について”, ，2020.

[11] 内閣府政策統括官（共生社会政策担当），“交通事故の被害・損失の経済的分析に関する調査報告書”, https://www8.cao.go.jp/koutu/chou-ken/h23/houkoku.html.

[12] 警察庁交通局，“令和元年中の交通死亡事故の発生状況及び道路交通法違反取締り状況等について”, ，2020.

[13] O. Corporation，“ドライブカルテ”，https://socialsolution.omron.com/jp/ja/products_service/transportation/drivekarte.

[14] 高度情報通信ネットワーク社会推進戦略本部・官民データ活用推進戦略会議，“官民 its 構想・ロードマップ 2020”, ，2020.

[15] 国土交通省自動車局先進安全自動車推進検討会，“ドライバー異常自動検知システム基本設計書”, ，2018.

[16] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context”, ECCV, pp.740–755Springer, 2014.

[17] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2d human pose estimation: New benchmark and state of the art analysis”, CVPR, pp.3686–3693, 2014.

[18] S. Johnson, and M. Everingham, “Clustered pose and nonlinear appearance models for human pose estimation”, BMVC, p.4, 2010.

[19] B. Sapp, and B. Taskar, “Modec: Multimodal decomposable models for human pose estimation”, CVPR, 2013.

[20] A. Toshev, and C. Szegedy, “Deeppose: Human pose estimation via deep neural networks”, CVPR, pp.1653–1660, 2014.

[21] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation”, NIPS, pp.1799–1807, 2014.

[22] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines”, CVPR, pp.4724–4732, 2016.

[23] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation”, ECCV, pp.483–499Springer, 2016.

[24] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, “Learning feature pyramids for human pose estimation”, ICCV, pp.1281–1290, 2017.

[25] B. Xiao, H. Wu, and Y. Wei, “Simple baselines for human pose estimation and tracking”, ECCV, pp.466–481, 2018.

[26] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation”, CVPR, 2019.

[27] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, “Deepercut: A deeper, stronger, and faster multi-person pose estimation model”, ECCV, pp.34–50, 2016.

[28] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, “Deepcut: Joint subset partition and labeling for multi person pose estimation”, CVPR, pp.4929–4937, 2016.

[29] S. Kreiss, L. Bertoni, and A. Alahi, “Pifpaf: Composite fields for human pose estimation”, CVPR, pp.11977–11986, 2019.

[30] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, “Poseidon: Face-from-depth for driver pose estimation”, CVPR, pp.4661–4670, 2017.

[31] S. Jha, and C. Busso, “Challenges in head pose estimation of drivers in naturalistic recordings using existing tools”, ITSC, pp.1–6, 2017.

[32] E. Ohn-Bar, S. Martin, A. Tawari, and M. M. Trivedi, “Head, eye, and hand patterns for driver activity recognition”, ICPR, pp.660–665, 2014.

[33] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, “Integral human pose regression”, ECCV, pp.529–545, 2018.

[34] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, “Shufflenet v2: Practical guidelines for efficient cnn architecture design”, ECCV, pp.116–131, 2018.

[35] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection”, ICCV, pp.2980–2988, 2017.

[36] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?”, ICCV, pp.2146–2153IEEE, 2009.

[37] S. Ioffe, and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, ICML, pp.448–456, 2015.

[38] 交通事故総合分析センター，“交通統計　平成 30 年版”, ，2020.

[39] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al., “The kinetics human action video dataset”, arXiv preprint arXiv:1705.06950, 2017.

[40] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, “A short note about kinetics-600”, arXiv preprint arXiv:1808.01340, 2018.

[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks”, CVPR, pp.1725–1732, 2014.

[42] S. Hochreiter, and J. Schmidhuber, “Long short-term memory”, Neural Computation, vol.9, no.8, pp.1735–1780, 1997.

[43] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description”, CVPR, pp.2625–2634, 2015.

[44] K. Simonyan, and A. Zisserman, “Two-stream convolutional networks for action recognition in videos”, NIPS, pp.568–576, 2014.

[45] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, “Convolutional learning of spatio-temporal features”, ECCV, pp.140–153Springer, 2010.

[46] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks”, ICCV, pp.4489–4497, 2015.

[47] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “Slowfast networks for video recognition”, ICCV, pp.6202–6211, 2019.

[48] R. Caruana, “Multitask learning”, Machine learning, vol.28, no.1, pp.41–75, 1997.

[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR, pp.580–587, 2014.

[50] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, “Hollywood in homes: Crowdsourcing data collection for activity understanding”, ECCV, pp.510–526Springer, 2016.

[51] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al., “Ava: A video dataset of spatio-temporally localized atomic visual actions”, CVPR, pp.6047–6056, 2018.

[52] X. Sun, B. Xiao, S. Liang, and Y. Wei, “Integral human pose regression”, arXiv preprint arXiv:1711.08229, 2017.

[53] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database”, CVPR, pp.248–255IEEE, 2009.

[54] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch sgd: Training imagenet in 1 hour”, arXiv preprint arXiv:1706.02677, 2017.

[55] 北島洋樹，沼田仲穂，山本恵一，五井美搏，“自動車運転時の眠気の予測手法についての研究: 第 1 報, 眠気表情の評定法と眠気変動の予測に有効な指標について”, 日本機械学会論文集 C 編，vol.63，no.613，pp.3059–3066，1997.

[56] A. Tsuchida, M. S. Bhuiyan, and K. Oguri, “Estimation of drowsiness level based on eyelid closure and heart rate variability”, EMBC, pp.2543–2546, 2009.

[57] T. Nakamura, A. Maejima, and S. Morishima, “Driver drowsiness estimation from facial expression features computer vision feature investigation using a cg model”, VISAPP, vol.2, pp.207–214, 2014.

[58] M. Tsujikawa, Y. Onishi, Y. Kiuchi, T. Ogatsu, A. Nishino, and S. Hashimoto, “Drowsiness estimation from low-frame-rate facial videos using eyelid variability features”, EMBC, pp.5203–5206, 2018.

[59] M. Sun, M. Tsujikawa, Y. Onishi, X. Ma, A. Nishino, and S. Hashimoto, “A neural-network-based investigation of eye-related movements for accurate drowsiness estimation”, EMBC, pp.5207–5210, 2018.

[60] T. Shih, and C. Hsu, “MSTN: multistage spatial-temporal network for driver drowsiness detection”, ACCV Workshops, pp.146–153, 2016.

[61] Z. Mardi, S. N. M. Ashtiani, and M. Mikaili, “Eeg-based drowsiness detection for safe driving using chaotic features and statistical tests”, JMSS, vol.1, no.2, pp.130–137, 2011.

[62] M. V. M. Yeo, X. Li, K. Shen, and E. P. V. Wilder-Smith, “Can SVM be used for automatic EEG detection of drowsiness during car driving?”, Safety Science, vol.47, no.1, pp.115–124, 2009.

[63] C. Lin, C. Chang, B. Lin, S. Hung, C. Chao, and I. Wang, “A real-time wireless brain–computer interface system for drowsiness detection”, TBioCAS, vol.4, no.4, pp.214–222, 2010.

[64] C. Lin, L. Ko, I. Chung, T. Huang, Y. Chen, T. Jung, and S. Liang, “Adaptive eeg-based alertness estimation system by using ica-based fuzzy neural networks”, IEEE Transactions on Circuits and Systems I: Regular Papers, vol.53, no.11, pp.2469–2476, 2006.

[65] A. Picot, S. Charbonnier, and A. Caplier, “Drowsiness detection based on visual signs: blinking analysis based on high frame rate video”, I2MTC, pp.801–804, 2010.

[66] H. Albalawi, and X. Li, “Single-channel real-time drowsiness detection based on electroencephalography”, EMBC, pp.98–101, 2018.

[67] A. Tsuchida, M. S. Bhuiyan, and K. Oguri, “Estimation of drivers drowsiness level using a neural network based error correcting output coding method”, ITSC, pp.1887–1892, 2010.

[68] J. Krajewski, D. Sommer, U. Trutschel, D. Edwards, and M. Golz, “Steering wheel behavior based estimation of fatigue”, Driving Assessment, pp.118–124, 2009.

[69] H. Malik, F. Naeem, Z. Zuberi, and R. ul Haq, “Vision based driving simulation”, CW, pp.255–259, 2004.

[70] R. F. Knipling, and W. W. Wierwille, “Vehicle-based drowsy driver detection: Current status and future prospects”, IVHS America, 1994.

[71] L. Lang, and H. Qi, “The study of driver fatigue monitor algorithm combined PERCLOS and AECS”, CASCON, vol.1, pp.349–352, 2008.

[72] M. Omidyeganeh, A. Javadtalab, and S. Shirmohammadi, “Intelligent driver drowsiness detection through fusion of yawning and eye closure”, VECIMS, pp.1–6, 2011.

[73] F. Zhang, J. Su, L. Geng, and Z. Xiao, “Driver fatigue detection based on eye state recognition”, CMVIT, pp.105–110, 2017.

[74] X. Huynh, S. Park, and Y. Kim, “Detection of driver drowsiness using 3D deep neural network and semi-supervised gradient boosting machine”, ACCV Workshops, pp.134–145, 2016.

[75] B. Reddy, Y. Kim, S. Yun, C. Seo, and J. Jang, “Real-time driver drowsiness detection for embedded system using model compression of deep neural networks”, CVPRW, pp.438–445, 2017.

[76] W. W. Wierwille, S. S. Wreggit, C. Kirn, L. A. Ellsworth, and R. J. Fairbanks, “Research on vehicle-based driver status/performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness. final report”, National Highway Traffic Safety Administration, no.DOT HS 808 247, 1994.

[77] W. Zhang, B. Cheng, and Y. Lin, “Driver drowsiness recognition based on computer vision technology”, Tsinghua Science and Technology, vol.17, no.3, pp.354–362, 2012.

[78] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM”, Neural Computation, vol.12, no.10, pp.2451–2471, 2000.

[79] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization”, arXiv preprint arXiv:1409.2329, 2014.

[80] J. Lyu, Z. Yuan, and D. Chen, “Long-term multi-granularity deep framework for driver drowsiness detection”, arXiv preprint arXiv:1801.02325, 2018.

[81] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition”, TPAMI, vol.35, no.1, pp.221–231, 2013.

[82] K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, arXiv preprint arXiv:1409.1556, 2014.

[83] C. Weng, Y. Lai, and S. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network", ACCV Workshops, pp.117–133, 2016.

[84] Q. Massoz, T. Langohr, C. François, and J. G. Verly, "The ULg multimodality drowsiness database (called DROZY) and examples of use", WACV, pp.1–7, 2016.

[85] "OKAO Vision", https://plus-sensing.omron.com/.

[86] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", CVPR, pp.770–778, 2016.

[87] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise", arXiv preprint arXiv:1706.03825, 2017.

[88] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization", ICLR, eds. Y. Bengio, and Y. LeCun, 2015.

[89] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification", CVPR, pp.4694–4702, 2015.

[90] "ImageNet Large Scale Visual Recognition Challenge", `http://www.image-net.org/challenges/LSVRC/`.

[91] "Merck Molecular Activity Challenge", `https://www.kaggle.com/c/MerckActivity`.

[92] S. Zagoruyko, and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer", , 2017.

[93] C. Bucilu, R. Caruana, and A. Niculescu-Mizil, "Model compression", SIGKDD, pp.535–541ACM, 2006.

[94] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network", arXiv preprint arXiv:1503.02531, 2015.

[95] J. Ba, and R. Caruana, "Do deep nets really need to be deep?", NIPS, pp.2654–2662, 2014.

[96] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets", , 2015.

[97] J. H. B. Junho Yim, Donggyu Joo, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning", CVPR, pp.7130-7138, 2017.

[98] M. D. Zeiler, and R. Fergus, "Visualizing and understanding convolutional networks", ECCV, pp.818–833, 2014.

[99] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning”, ICML, pp.41–48ACM, 2009.

[100] A. Krizhevsky, and G. Hinton, “Learning multiple layers of features from tiny images”, Technical report, University of Toronto, 2009.

[101] K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, , 2014.

研究業績一覧

学術論文

[1] K. Nishiyuki, J.Y. Shiau, S. Nagae, T. Yabuuchi, K. Kinoshita, Y. Hasegawa, T. Yamashita, H. Fujiyoshi, “Driver Drowsiness Estimation by Parallel Linked Time-Domain CNN with Novel Temporal Measures on Eye States.”, IEICE TRANSACTIONS on Information and Systems, Vol.103, No.6, pp.1276-1286, 2020.

[2] 西行健太, 日向匡史, 田博, 木下航一, 長谷川友紀, 山下隆義, 藤吉弘亘, “高速かつ省メモリなドライバ姿勢推定”, 人工知能学会論文誌, Vol.35, No.6, 2020.

[3] 西行健太, 日向匡史, 田博, 木下航一, 長谷川友紀, 山下隆義, 藤吉弘亘, “ドライバ姿勢と動作のマルチタスク学習による高速かつ高精度なドライバ動作認識”, 人工知能学会論文誌, Vol.36, No.2, 2021.

国際会議発表論文 (査読あり)

[1] J.Y. Shiau, K. Nishiyuki, S. Nagae, T. Yabuuchi, K. Kinoshita, Y. Hasegawa, T. Yamashita, H. Fujiyoshi, “Driver Drowsiness Estimation by Parallel Linked Time-Domain CNN with Novel Temporal Measures on Eye States.”, International Conference of the IEEE Engineering in Medicine and Biology Society, 2019

学会口頭発表 (査読なし)

[1] 西行健太, 藤吉弘亘 “木構造に着目した負の転移を回避する転移学習アルゴリズム“, 電子情報通信学会技術研究報告, 2015.

[2] 西行健太, 山下隆義, 藤吉弘亘 “階層型 Knowledge Distillation による DNN のコンパクト化“, 電子情報通信学会技術研究報告, 2017.

解説記事 (技報)

[1] 日向匡史, 木下航一, 西行健太, 長谷川友紀 “自動運転時代におけるドライバモニタリング技術: 時系列 Deep Learning によるドライバ状態の推定について“ , Omron technics, 2018.