機械知覚&ロボティクスグループ
中部大学　工学研究科 情報工学専攻

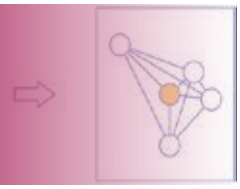
MPRG
MACHINE PERCEPTION AND ROBOTICS GROUP

2022年度　山下研究室　修士論文発表　アブストラクト

Deep Learning, Human like Guidance, Graph Convolutional Networks

視線と時系列情報を導入したHuman Like Guidanceに関する研究

宮田 昌樹



Deep Learning, Sports, Gaze Prediction

サッカーのトレーニングにおける注視点比較と視線推定に関する研究

伊佐 稜

Deep Learning, Federated Learning, Knowledge Distillation

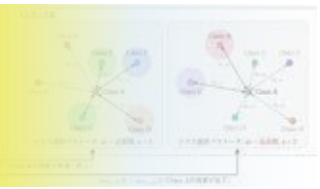
連合学習における知識転移グラフに関する研究

岩田 幸

Deep Learning, Data Augmentation, Long-tailed Recognition

クラス間距離を考慮したデータ拡張に関する研究

藤井 駿伍



Deep Learning, Object Detection, Model Compression

Knowledge DistillationとPruning を用いた物体検出モデルの圧縮に関する研究

LIU LYUZHUANG

視線と時系列情報を導入した Human-like Guidance に関する研究

TP21014 宮田 昌樹　　指導教授：山下 隆義

1. はじめに

自動車の案内として，機械による Turn-by-Turn ナビゲーションと人間によるナビゲーションの 2 つが考えられる．Turn-by-Turn ナビゲーションとは既存のカーナビゲーションシステムのことであり，デジタル地図を用いた道路に基づいた案内文によりナビゲーションをする．そのため，「100m 先を右方向です」といった定型文による案内であり，直感的な理解が難しいことがある．一方，人間によるナビゲーションでは，同乗者が視認した周辺環境の情報を利用して，建物や周辺車等を利用して案内を行う．そのため，「前の黄色い車に付いて行ってください」と言った状況に合わせた案内になり，直感的に理解しやすい．よって，Turn-by-Turn ナビゲーションによる問題を防ぐことが可能であると考えられるが，同乗者がいなければ成り立たない．この問題点を解決するために，本研究では人間のナビゲーションをシステムで再現することを目標とした Human-like Guidance (HℓG) [1] の実現を目的とする．HℓG では，状況に合わせたわかりやすい案内文でのナビゲーションがシステムのみで可能となる．本研究では，人間の運転中の視線情報を考慮したシーングラフによる HℓG の実現を提案する．

2. 提案手法

本研究では，時系列を考慮した周囲の環境情報を表すテンポラルシーングラフを提案し，視線情報とテンポラルシーングラフによる HℓG の実現手法について述べる．提案手法の概略を図 3 に示す．提案手法ではまず，周囲の状況をシーングラフとして表現する．同時に視線推定モデルを用いて，着目対象を推定し，この視線情報をシーングラフに追加する．次に，視線情報付きシーングラフを任意の時刻分連結した，テンポラルシーングラフを作成する．作成したテンポラルシーングラフを Spatial Temporal Graph Convolutional Network に入力し，テンポラルシーングラフの特徴量を獲得する．最後に，得られた特徴量から，Transformer によって案内文を生成する．

2.1 運転者の視線の推定

人間の運転中の視線を再現するために，視線推定モデルである DINet [2] を用いて視線推定を行う．DINet は RGB 画像を入力とし，視線推定結果をヒートマップとして出力する．ヒートマップから注視物体特定するために，閾値以上の領域の外接矩形を求める．学習では，出力したヒートマップと真値の視線座標を基にしたマップとの誤差を求める．ここで，視線情報には，通常運転時の視線情報ではなく，ナビゲーションを行う際に着目しやすい対称に意図的に誘導を行ったデータを用いる．これによって，HℓG のナビゲーションに適した視線の推定が可能となる．

2.2 視線情報を付与したシーングラフ

時刻 t の周囲の状況のシーングラフ G を定義する．ノードは $v_i =$ { オブジェクトクラス，座標，車両名，色 } を割り当て，エッジにはノード間の相対距離を用いる．加えて，視線推定により得られた矩形と位置が一致するオブジェクトを着目対象とする．よって，ノード集合を $vg_i =$ { オブジェクトクラス，座標，車両名，色，　視線情報の有無 } とする．図 1 に示すように，視線情報を付与したシーングラフを定義する．

2.3 ST-GCN による特徴抽出

各オブジェクトの位置関係について時系列情報を考慮するために，ST-GCN [3] を用いる．現在から未来 T 時刻分について，各時刻のシーングラフのノード集合を定義する．また，隣接した他時刻における同じオブジェクトを接続するエッジを追加することで，テンポラルシーングラフを定義する．図 2 に示すように，グラフ畳み込み処理の際に，各時刻で共通するノードを同時に畳み込む．これにより，オブジェクト毎に移動を考慮した特徴量を獲得する．

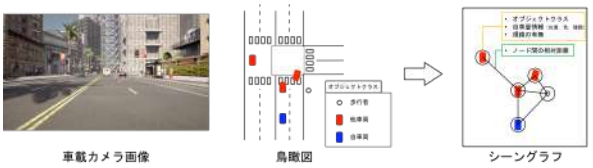


図 1：シーングラフの作成

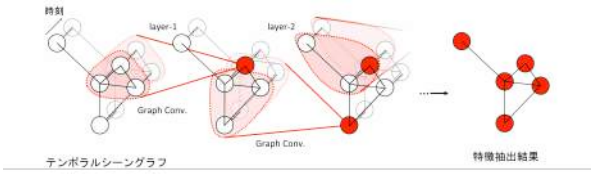


図 2：ST-GCN によるテンポラルシーングラフの特徴抽出

2.4 Transformer による案内文生成

図 4 に示すように，ST-GCN で得られたグラフ特徴量から，着目対象を基にした案内文の生成を行う．文章生成のモデルとして Transformer [4] を用いる．Trandformer Encorder では，ST-GCN で得られた各ノードの特徴量をトークンとして入力する．Transformer Decoder では，Encorder から獲得したトークンを Decorder の入力として用い，単語を出力する．この時，教師データとして該当シーンにおける正解となる案内文を用いる．正解の案内文は，該当シーンにおける真値の視線情報を用いてルールベースで作成する．これにより，各オブジェクトに対応したナビゲーション文を生成できる．

3. 評価実験

本実験では，提案した手法における時系列情報の有無による影響を評価する．

3.1 データセット

データセットには，自動運転開発シミュレータである CARLA Simulater で作成した動画像を用いる．全 14 シーン，合計 1672 フレームの運転シーンの動画像と，各フレームに存在する全ての各オブジェクトに対する案内文をデータセットとする．ここで，案内文はルールベースでオブジェクトに対する文章を自動生成した．オブジェクトが人の場合，2 名以上が近距離に位置している時は集団とし，集団に対する案内文とした．案内文作成のルールは，進行方向，対象オブジェクトの名前，進行方向に対する自車両の行動を含む文とする．対象オブジェクトは，該当シーンにおける視線情報を用いて選択する．自車両の行動に関する指示は，対象オブジェクトの進行方向と自車量の進行方向が同じであれば追従，異なる場合は対象オブジェクトの位置を用いる．

3.2 時系列導入による定性的評価

図 5 に示すように，時系列情報ありの場合は「following kawasaki ninja」と移動を伴う案内文が生成されている．したがって，動的なオブジェクトに着目したわかりやすいナビゲーションの生成できていることが確認できる．一方で，時系列情報なしの場合，「kawasaki ninja is located」と移動を伴わない案内文が生成されている．時系列情報がない場合でも，動的なオブジェクトに着目はできているが，「following」が「located」となっており，動きの情報が欠落していることが確認できる．このことから，時系列情報を導入することは Human-like Guidance において有効であることが確認できた．

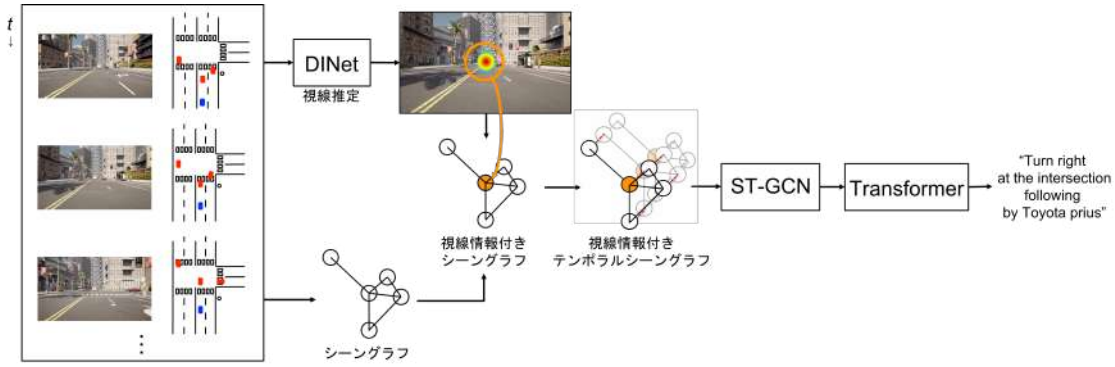


図 3：提案手法の流れ

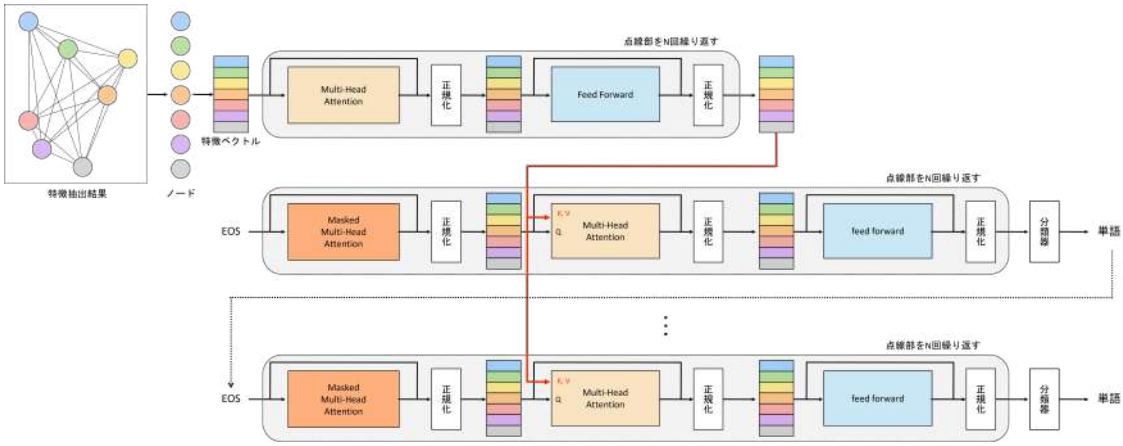


図 4：Transformer による文章生成

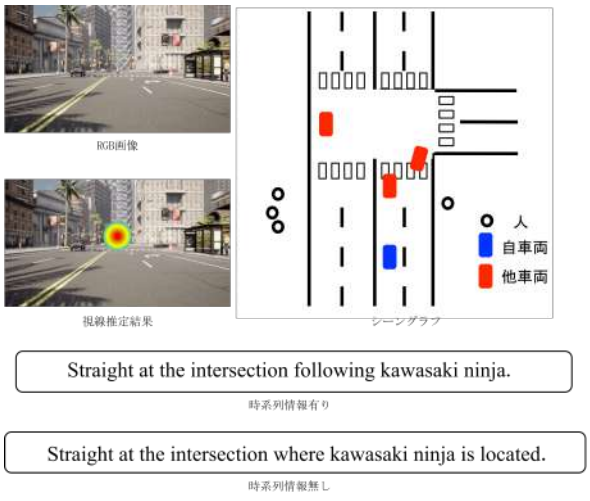


図 5：案内文生成例

3.3 定量的評価

自然言語生成の指標における定量的評価を表 1 に示す．表 1 より，時系列情報を導入することによって各評価指標での精度が向上しており，目的である Human-like Guidance に重要となる自然な案内文が生成されている．したがって，時系列情報の導入が有効な手段であることが確認できる．

4. おわりに

本研究では，HℓG のための，テンポラルシーングラフを用いたナビゲーション生成手法を提案した．視線情報と時系列情報を用いることで，人間のような状況に合わせたナビゲーションの生成が実現できた．視線推定の評価実験では，ナビゲーションに適した視線が HℓG において有効であると確認できた．時系列情報の有無による評価実験では，時系列情報を用いることが HℓG において有効であることが確認できた．今後の課題としては，視線推定モデルの変更や HℓG に適切な評価指標の検討などが挙げられる．

表 1：時系列の有無による定量的評価比較

	時系列なし	時系列あり
Bleu_4	0.291	**0.325**
METEOR	0.199	**0.223**
ROUGE_L	0.450	**0.520**
CIDEr	2.649	**2.945**
SkipThoughtCS	0.458	**0.487**

参考文献

[1] M. Tsutomu, *et al.*, “Human-Like Local Navigation System Inspired by a Hippocampal Memory Mechanism”, Brain-Inspired Information Technology,pp.29 – 32, 2010.

[2] S. Yang, *et al.*, “A Dilated Inception Network for Visual Saliency Prediction”, IEEE, 2019

[3] S. Yan, *et al.*, “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition”, AAAI, 2018.

[4] S. Vaswani, *et al.*, “Attention Is All You Need”, CoRR, 2017.

研究業績

[1] 宮田昌樹　等, “関係性部分グラフを用いた Graph Convolutional Network による経路予測”, MIRU, 2021.

[2] M. Miyata, *et al.*, “Relational Subgraph for Graph-based Path Prediction”, MVA, 2021.

サッカーのトレーニングにおける注視点比較と視線推定に関する研究

TP21001 伊佐稜　　指導教授：山下隆義

1. はじめに

スポーツ選手の視線情報を取得して解析することは，状況判断能力等の選手の能力改善への利用が期待されている．とりわけ，サッカーではスキャニングなどの周囲を確認する動作や，パスを受ける際の注視位置などに関する研究が行われている [2, 3]．また先行研究では能力改善として，視野の狭さに着目した育成事例が報告されている．しかしながら，これまでの研究ではプロサッカー選手を対象とした一人称視点の動画像撮影，及びその解析は充分になされていない．そこで，本研究ではサッカーにおけるトレーニング時の視線行動についてプロサッカー選手とアマチュアサッカー選手の比較を定量的に分析することを目的とする．合わせて，プロサッカー選手の視線推定モデルを作成し，プロサッカー選手の視線と比較した際にアマチュアサッカー選手に気づきを促すことを目的とする．

2. 関連研究

2.1. サッカーにおける視線情報

サッカー選手を対象とした視線の研究はスキャニング回数の比較 [2] と注視位置比較 [3] がある．Phantak ら [2] はサッカー選手権ユーロ 2016 において，セントラルミッドフィルダーを対象にスキャニング回数が多い選手はパスの成功率が高いことを報告した．ここで，スキャニングとは周囲確認のために頭部を振る動きである．Aksum らの研究 [3] ではエリートサッカー選手の試合において，スキャニング中のフレーム内にチームメイトより対戦相手が多いことを報告した．

2.2. 視線推定

視線推定は，一人称視点の動画像から注視点座標を推定する手法である．代表的な視線推定手法に DINet[1] がある．この手法では，Dilated ResNet と Inception 構造の畳み込み層を用いることで特徴マップの解像度を落とさずに視線推定が行える．しかし，DINet は 2 次元視線推定であるため，時系列が考慮できないという問題点がある．

3. 提案手法

本研究では，プロサッカー選手とアマチュアとして大学生サッカー選手の視線の差異を確認し，プロサッカー選手の優れている点を定量的に評価する．次に，モデル化したプロサッカー選手の視線推定モデルを作成することで，プロサッカー選手と大学生サッカー選手の視線を定性的に評価する．

3.1. 測定概要

視線計測の際に，同条件下で 2 種類のトレーニングを行う．測定対象者はセントラルミッドフィルダーのプロサッカー選手 (Pro) と大学生サッカー選手 2 名 (Uni1)，(Uni2) である．

測定項目は，ボールポゼッショントレーニング (BPT) と Small-sided games(SSG) である．ここで，SSG は攻撃フェーズ (Att) と守備フェーズ (Def) に分けて分析を行う．それぞれのトレーニングの条件を表 1 に示す．また，Uni2 は BPT のみを測定する．

測定変数は，スキャニング回数とスキャニング時の平均時間，スキャニング時の注視位置とする．また，スキャニング時の注視位置はフレーム内のボールがなくなったタイミングからボールが映るフレームまでとし，100msec 以上の時にフレームを目視で確認し，人手でクラスラベリングを行う．

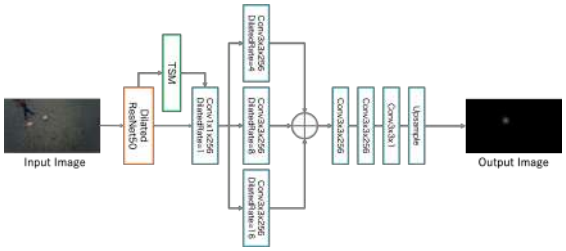


図 1: 提案手法のネットワーク構造

表 1: 各トレーニングの条件

	BPT	SSG
人数	7(攻)vs4(守)	11vs11
範囲	14m × 14m	52.5m × 40m
時間	10min	6min
条件	2 タッチ以下	フリータッチ

クラスは，Teammate(味方)， Opponent(敵)，Space(スペース)，T/O(味方と敵)，T/S(味 方とスペース)，O/S(敵とスペース)，T/O/S(味方と敵とスペース) である．

3.2. 測定結果

プロサッカー選手と大学生サッカー選手の各トレーニングにおける測定変数の結果を表 2 に示す．

表 2: 各トレーニングにおける測定結果

		回数 (回)	注視時間 (sec)	注視位置 (%)						
				Teammate	Opponent	Space	T/O	T/S	O/S	T/O/S
BPT	Pro	**117**	0.187	35.1	15.8	30.7	**12.3**	4.4	1.8	0.0
	Uni1	52	0.160	44.2	15.4	30.8	0.0	7.7	1.9	0.0
	Uni2	31	0.162	29.0	16.2	32.3	9.7	6.5	6.5	0.0
SSG	Att_Pro	**93**	**0.353**	21.7	**24.6**	14.5	**23.2**	5.8	2.9	**7.2**
	Def_Pro			**25.0**	**37.5**	25.0	4.2	4.2	0.0	**4.2**
	Att_Uni1	49	0.216	21.6	13.5	**37.8**	16.2	2.7	8.1	0.0
	Def_Uni1			25.0	33.3	16.7	0.0	8.3	**16.7**	0.0

表 2 より，各トレーニングで Pro は Uni1，Uni2 と比較し，スキャニング回数が多く，スキャニング時の平均時間が長いことがわかる．先行研究 [2] から，プロサッカー選手の方が優れた選手であると言える．また，スキャニング時の注視点は Pro が BPT では T/O，SSG では Opponent の割合が高く，視線の特徴として挙げられる．

視線分析結果から，各トレーニングにおいてプロサッカー選手と大学生サッカー選手で注視位置が異なり，プロサッカー選手の視線を模倣することが，パフォーマンスの向上を促す要素であることが確認できた．

3.3. 視線推定モデルの構築

次に，視線推定モデルを構築する．従来の DINet は画像ベースの視線推定モデルであるため，時系列の考慮ができない問題がある．そこで本研究では，時系列を考慮した視線推定モデルである，TS-DINet を提案する．提案手法のネットワーク構造を図 1 に示す．提案手法では，時系列考慮をするために Dilated ResNet50 の後に TSM[4] を用いる．TSM は，特徴量の一部を前後のフレームとシフトし，次の畳み込み層へ伝達することで，時系列の特徴を獲得している．

視線推定の出力結果は画像全体における人間の視野の確率分布と見なすことができる．予測された顕著性マップと真値を確率分布に変換する場合，これらの値を正規化する必要がある．一般的に用いられる L1 ノルムを用いて，線形正規化を行う．損失関数を以下の式 (1) に示す．ここで，$x = (x_1, x_i, x_N)$ は正規化前の値である．

$$L(p, g) = \sum_i |p_i - g_i| \tag{1}$$

$$\text{where, } p_i = \frac{x_i^p}{\sum_{i=1}^N x_i^p}, \quad g_i = \frac{x_i^g}{\sum_{i=1}^N x_i^g}$$

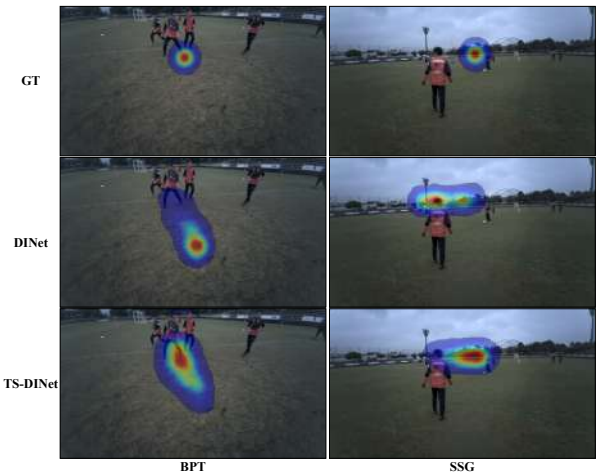
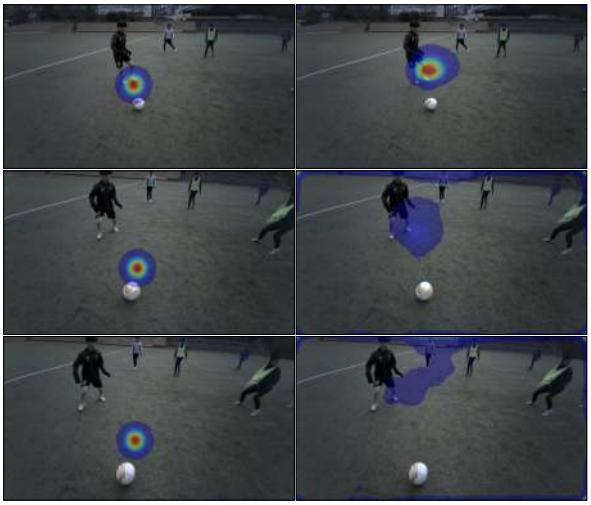
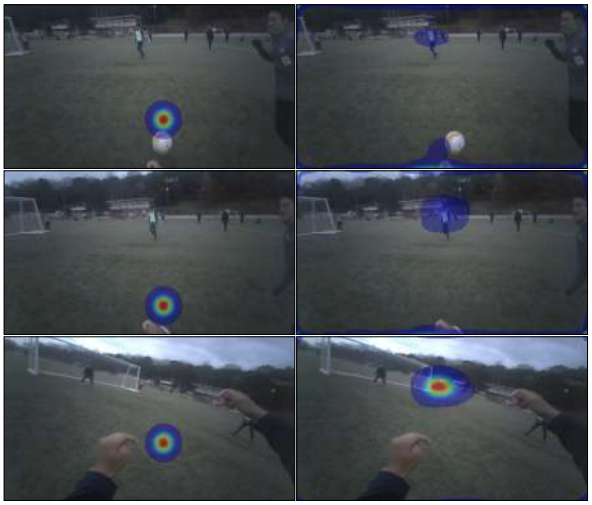


図 2: 定性的評価



(a) BPT



(b) SSG

図 3: 各トレーニングにおける出力結果比較

4. 評価実験

本章では，時系列考慮の有無でプロサッカー選手の視線の模倣できたかを検証する．比較対象として，DINet を用いる．

4.1. 実験概要

TSM で考慮するフレーム数 f は短い時系列考慮 ($f = 2, 4$)，スキャニングの平均時間に合わせた時系列考慮 ($f = 6$)，長期的な時系列考慮 ($f = 12, 24, 36, 48$) で行う．使用するデータセットはプロサッカー選手の各トレーニングの動画像フレームとする．BPT は学習に 8,884 枚/評価に 1,121 枚，SSG は学習に 6,944 枚/評価に 1,029 枚用いる．評価指標は Normalized Scanpath Saliency (NSS)，TheLinearCorrelationCoefficient (CC)，Kullback-Leibler (KL) を用いる．

4.2. 定量的評価

各トレーニングにおける定量的評価を表 3 に示す．表 3 より，従来手法である DINet と比較し，提案手法の方が精度が高いことが確認できる．

表 3: 定量的評価

		DINet	提案手法 (TS-DINet)						
			f=2	f=4	f=6	f=12	f=24	f=36	f=48
BPT	NSS ↑	7.19	7.17	7.17	**7.30**	7.19	7.18	7.11	7.16
	CC ↑	0.44	0.51	0.51	**0.52**	0.51	0.51	0.51	0.51
	KL ↓	1.99	1.97	1.98	1.98	1.97	1.97	1.97	2.00
SSG	NSS ↑	5.66	6.02	5.99	6.01	5.77	5.73	5.95	**6.10**
	CC ↑	0.42	0.43	0.43	0.43	0.48	0.51	0.51	**0.53**
	KL ↓	2.30	2.42	2.43	2.44	2.42	2.29	**2.16**	2.17

BPT では，$f = 6$ で最も精度が高いことが確認できる．BPT は限られた正方形の空間内でパス回しを中心に行うトレーニングであるため，味方の位置を確認するスキャニングが最も重要である．よって，スキャニングの平均時間に合わせた $f = 6$ の結果が最も優れているのではないかと考える．

SSG では，$f = 48$ で最も精度が高いことが確認できる．これは，SSG が実際のゲームを模したトレーニングとなっており，スキャニング時間も長いことから長期的な時系列考慮が有効であったためと考える．$f = 12, 24, 36, 48$ では，いずれの評価指標もフレーム数が上がるにつれて評価が高くなる傾向が見られる．このことから，SSG において長期的な視線を考慮することは，実際のゲームで視線を模倣するモデルを作成する際にも重要であることが考えられる．

4.3. 定性的結果

各トレーニングにおける定性的評価を図 2 に示す．また，表 3 より BPT は $f = 6$，SSG は $f = 48$ を用いて評価を行った．

BPT では，DINet と比較して提案手法が真値に近い出力であることが確認できる．また，SSG も BPT と同様に DINet より提案手法が真値に近い出力であることが確認できる．DINet は手前の人物に注視する推定結果に対して，提案手法では真値と同様に，奥のボール保持者に注視するよう推定できている．これは，提案手法は，時系列考慮によりボールを保持している情報を捉えることができ，奥の人物を注視できたためと考えられる．これらの定量的，定性的結果から，提案手法で BPT ではスキャニングの平均時間に合わせた時系列考慮，SSG では長期的な時系列考慮によりプロサッカー選手の視線を模倣する結果が確認できた．

5. 大学生サッカー選手との視線比較

プロサッカー選手の視線を学習した TS-DINet を用いて，大学生サッカー選手の視線を評価する．大学生サッカー選手とプロサッカー選手の各トレーニングにおける視線比較を図 3 に示す．また，視線推定の信頼度が理由は，学習に使用したデータ数の問題であると考える．

BPT では，大学生サッカー選手は，ボールを追うような視線の動きをしているのに対して，プロサッカー選手の視線マップはフレーム上部に集中している．プロサッカー選手は，パスを出した後にスキャニングや，周りの味方や人を確認する視線の動きをする．そのため，プロサッカー選手の視線では遠方の味方及び，パスを出した味方を注視する推定結果が出力されたと考えられる．

SSG では，大学生サッカー選手はゴール前に BPT と同様にボールに注視が寄っている．BPT と同様に，プロサッカー選手はボールより目の前の味方を注視する推定結果が出力された．

これらの結果から，プロサッカー選手の視線推定モデルは各トレーニングにおいて有効な結果を示しており，大学生に気づきを促せるレベルであると考える．

6. おわりに

本研究では，サッカーのトレーニングにおける注視点比較と視線推定を行った．データ分析の結果，プロサッカー選手の方が大学生サッカー選手より優れていることが定量的に判明した．視線推定は DINet と比較し，提案手法を用いることでプロサッカー選手の視線を模倣することができた．大学生サッカー選手との視線比較では，今後は，大学生サッカー選手に気づきを促すことでプレーの変化を確認する．

参考文献

[1] S. Yang, *et al.*, “A Dilated Inception Network for Visual Saliency Prediction”, TMM, 2019.
[2] A. Phantak, *et al.*, “Keep Your Head Up : Correlation between Visual Exploration Frequency, Passing Percentage and Turnover Rate in Elite Football Midfielders ”, MDPI Sports, 2019.
[3] K. Aksum, *et al.*, “What Do Football Players Look at? An Eye-Tracking Analysis of the Visual Fixations of Players in 11 v 11 Elite Football Match Play”, IJSSC, 2021.
[4] J. Lin, *et al.*, “TSM: Temporal Shift Module for Efficient Video Understanding”, ICCV, 2019.

研究業績

[1] 伊佐 稜　等, “物体追跡による経路データセットの自動生成”, 電気・電子・情報関係学会 東海支部連合大会, 2021.
(他 2 件)

連合学習における知識転移グラフに関する研究

TP21002 岩田幸　　指導教授：山下隆義

1. はじめに

連合学習 [1] は複数のクライアントで学習する際に，データセットを共有せず，重みパラメータのみを共有して学習する手法である．データを所有する側をクライアント，クライアントが持つモデルの重みパラメータをまとめる役割を持つ側をサーバと呼ぶ．このとき，クライアント間でデータセットを共有しないため，各クライアントが持つデータのプライバシーを保護した状態で学習できる．各クライアントが保有するデータセットを共有しないことにより，全体のデータ分布を考慮した学習が困難となる問題がある．そのため，各クライアントが保有するデータのクラス分布やデータ量が異なる場合に 1 つのサーバモデルを学習することは難しい．

本研究では，サーバが配布した Meme モデルとクライアントが持つモデルの共同学習を行う．ここで，共同学習とはネットワークが知識を転移しながら学習することを指す．このとき，各クライアントで共同学習を行うことで獲得したサーバモデルを最適化することにより，データ分布の影響を低減した学習が期待できる．

2. 連合学習

連合学習 [1] は，複数のクライアントそれぞれで学習した重みを 1 つのサーバに集約して更新する手法である．学習結果を 1 箇所に集約するため，データセットが非同一分布である際に安定した学習が困難である．そのため，モデルの重みを直接集めるのではなく，クライアント独自のモデルを用意することにより，非同一分布のデータに対応する Federated Mutual Learning (FML) [2] が提案されている．FML は，クライアント独自のモデルと Meme モデルを相互学習 [3] する．クライアント独自のモデルをつくることにより，従来の連合学習と比べてクライアントごとに特化した学習が期待できる．しかし，サーバが配布したモデルとクライアントが保有するモデルの学習方法は限定的であり，データ分布が異なるクライアントを用いた学習に適しているとは限らない．また，クライアントごとのモデル構造を人手で選択する必要がある．

3. 提案手法

本研究では，FML のサーバが配布した Meme モデルとクライアントが保有するモデルの共同学習に知識転移を導入し，最適化することを提案する．損失関数に 3 種のゲート関数 [4] を組み込むことにより，各クライアント最適な学習方法の獲得を目指す．

3.1 知識転移を導入した FML

図 1 に提案手法の概念図を示す．サーバが保有するモデルはサーバモデルのみであり，Meme モデルとクライアントモデルはクライアントが保有している．Meme モデルとクライアントモデルの学習で知識転移する際の学習方法を多様な学習表現から最適化することにより，非同一分布に適した学習の獲得をする．以下に学習の流れを示す．

Step 1. Meme モデルとクライアントモデル間で知識転移する際のゲート関数やクライアントモデルをランダムに設定

Step 2. サーバモデルの重みをクライアントの Meme モデルに配布

Step 3. 各クライアントが保有するデータを用いて，Meme モデルとクライアントモデルを共同学習

Step 4. Meme モデルの重みをサーバに集約してサーバモデルを更新

Step 5. 学習が収束するまで Step 2 に戻る

Step 6. 事前に指定した探索回数になるまで Step 1 に戻る

Step 7. サーバモデルの検証精度が最大となる学習方法を選択

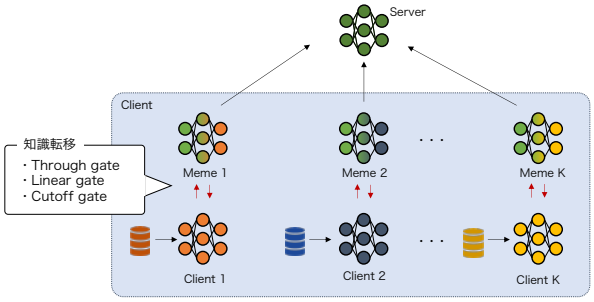


図 1：提案手法の概念図

3.2 クライアントモデルの共同学習

サーバが配布した Meme モデルとクライアントが保有するモデルの共同学習を行う．このとき，出力確率分布の損失は，Kullback-Leibler (KL) divergence を使用して算出する．クライアントモデルと Meme モデルの相互学習に使用する損失関数 $L_{s,t}$ は式 (1) のように求める．

$$L_{s,t} = \sum_{n}^{B} G_{s,t}(KL(\boldsymbol{p}_s(\boldsymbol{x}_n)||\boldsymbol{p}_t(\boldsymbol{x}_n))) + L_{C_t} \quad (1)$$

ここで，$\boldsymbol{p}_s$ は知識転移元のモデルの確率分布，$\boldsymbol{p}_t$ は知識転移先のモデルの確率分布，B はバッチサイズ，$G(\cdot)$ は，ゲート関数，L_{C_t} はモデル t と教師ラベルのクロスエントロピー誤差である．得られた損失から勾配を求め，クライアントモデルと Meme モデルを更新する．

3.3 ゲート関数

損失関数をゲート関数で制御することにより，それぞれのネットワークでの知識に多様性を持たせる．ゲート関数として Through Gate， Cutoff Gate， Linear Gate の 3 種類の関数を定義する．

Through Gate は入力されたサンプルごとの損失に変化を加えないゲート関数である．式 (2) に Through Gate を示す．Through Gate により，FML が表現可能である．

$$G_{s,t}^{\mathrm{Through}}(a) = a \quad (2)$$

Cutoff Gate は損失計算を行わないゲート関数である．式 (3) に Cutoff Gate を示す．Cutoff Gate を用いることにより，知識の伝達を切断することができる．

$$G_{s,t}^{\mathrm{Cutoff}}(a) = 0 \quad (3)$$

Linear Gate は学習時間によって損失の重みを線形に変化させるゲート関数である．式 (4) に Linear Gate を示す．学習初期には重みを小さくし，学習が進むにつれて重みを大きくする．そのため，学習初期で精度が低いモデルの知識が伝達することを抑えることができる．

$$G_{s,t}^{\mathrm{Linear}}(a) = \frac{c}{c_{end}} \cdot a \quad (4)$$

ここで，c は累積更新回数であり，c_{end} は学習終了時における更新回数である．

3.4 サーバモデルの更新

更新したクライアントの Meme モデルをサーバに集約し，サーバモデルを更新する．サーバモデルの更新後の重み w_{t+1} はクライアント k に配布して更新した Meme モデル w_{t+1}^k を用いて式 (5) で求める．

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \quad (5)$$

ここで，K はクライアント数，n はクライアント全体のデータ数，n_k はクライアント k が保有するデータ数である．各

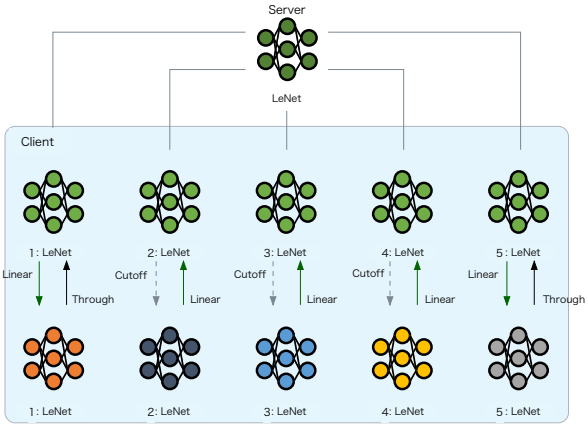


図 2： MNIST での最適化結果（Top1）

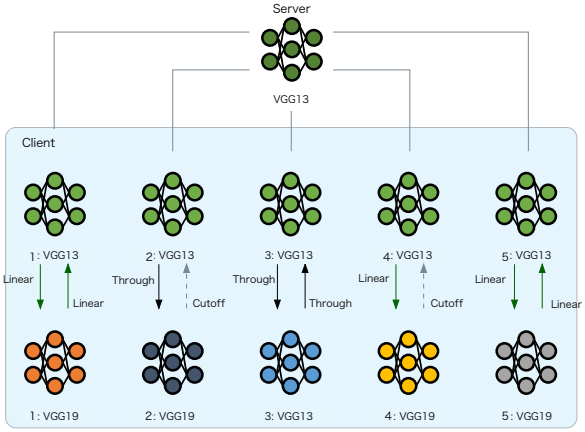


図 3： CIFAR-10 での最適化結果（Top1）

クライアントの重みは各クライアントが保有するデータ量によってサーバに対する影響が決定できる．各 Meme モデルを用いて更新したサーバモデルの重みを各クライアントに配布して Meme モデルを更新する．

3.5 学習の最適化

最適化対象のモデルをサーバモデルとする．サーバモデルの検証精度が最大となるゲート関数とモデルを選択する．使用する最適化手法はランダムサーチである．ゲート関数とモデルをランダムに選択して学習し，検証精度が最大となる学習方法を選択する．これにより，各クライアントのデータを用いて 1 つのモデルを学習する際に効果的な学習方法を獲得できる．

4. 評価実験

提案手法の有効性を調査するために評価実験を行う．従来手法には，連合学習の代表的な手法である Federated Averaging (FedAvg) [1] とゲート関数を全て Through Gate として学習に多様性を失った学習方法である FML を用いる．

4.1 実験条件

実験には 10 クラスの手書き数字の画像 MNIST と一般物体認識データセットである CIFAR-10 データセットを使用する．MNIST の訓練用 60,000 枚を学習用 50,000 枚，検証用 10,000 枚に分割，CIFAR-10 の場合は訓練用 50,000 枚を学習用 40,000 枚，検証用 10,000 枚に分割し，各クライアントで同程度の枚数となるよう分配して使用する．学習は各クライアントにおいてクラスバランスが崩れたデータセットを用い，評価時は評価用データセットを使用して，サーバモデルを評価する．非同一分布のデータセットはディリクレ分布 $\mathrm{Dir}(x|\alpha)$ を使用して作成する．α が小さいほど非同一分布となり，大きいほど同一分布に近くなる．

ネットワークモデルには，MNIST を使用する場合は LeNet，CIFAR-10 を使用する場合は VGG13 と VGG19 を用いる．各クライアントで 50 エポック学習した後にサーバへ集約し，集約する際の通信回数は 200 step である．また，サーバモデルの更新時に使用する使用するクライアント数は 5 であり，常時全てのクライアントを学習に使用する．探索は 1,000 回行い，Top1 の精度を用いて従来手法と比較する．

4.2 獲得した学習方法

ディリクレ分布のハイパーパラメータ α を 1000 に設定した際の MNIST を用いて探索した結果を図 2，CIFAR-10 を用いて探索した結果を図 3 に示す．MNIST と CIFAR-10 どちらにおいても 3 種類のゲート関数が選択されていることから，ゲート関数を用いてクライアントとサーバの学習方法に多様性を持たせることに有効性があるとわかる．

MNIST を用いて探索した結果から，クライアント 2, 3, 4 において Meme モデルは教師ラベルとクライアントモデルからの知識を使用するのに対し，クライアントモデルは教師ラベルのみを用いて学習すると良いことがわかる．このことから，教師ラベルを用いて学習した知識を徐々に学習に取り入れることが有効である．

表 1： 非同一分布データセットにおける精度 [%]

データセット	手法名	$\alpha = 1000$	$\alpha = 10$	$\alpha = 0.1$
MINST	FedAvg	88.04	99.22	94.47
	FML	99.22	99.26	96.92
	Ours	**99.30**	**99.33**	**98.50**
CIFAR-10	FedAvg	37.33	35.88	35.22
	FML	40.37	38.76	37.23
	Ours	**42.69**	**39.35**	**38.65**

CIFAR-10 を用いて探索した結果から，クライアント 1, 3, 5 の知識を取り入れることによって学習していることがわかる．このことから，全てのクライアントの知識を使用する必要はないことがわかる．Cutoff Gate の導入により，不要な知識の伝達を防ぐことができた．また，VGG13 と 19 どちらも選択されていることから，両者ともに知識も学習に有効である．

4.3 非同一分布データセットにおける精度比較

非同一分布データセットにおける精度を従来手法と比較する．表 1 に非同一分布データセットにおける精度を示す．提案手法は，従来法である FML と比較して有効であることがわかる．また，MNIST において，非同一分布（α が小さい）に近づくにつれ，FML は 3.30 pt 精度が低下した一方，提案手法は 0.80 pt の精度低下に抑えることができた．このことから，提案手法により非同一分布に有効な学習ができたといえる．

5. おわりに

本研究では連合学習の学習方法を自動最適化することを提案した．評価実験により，提案手法を用いることで，ゲート関数を用いることにより効果的な学習ができることを確認した．また，非同一分布データセットを用いて従来手法と比較して有効性を示した．今後は，探索方法を改良することにより，一度の学習で適切な学習方法を得ることができるよう検討する．

参考文献

[1] H. B. Mcmahan, *et al.*, “Communication-efficient learning of deep networks from decentralized data”, AISTATS, 2017.

[2] T. Shen, *et al.*, “Federated Mutual Learning”, arXiv:2006.16765, 2020.

[3] Y. Zhang, *et al.*, “Deep Mutual Learning”, CVPR, 2018.

[4] S. Minami, *et al.*, “Knowledge Transfer Graph for Deep Collaborative Learning”, ACCV, 2020.

研究業績

[1] S. Iwata, *et al.*, “Refining Design Spaces in Knowledge Distillation for Deep Collaborative Learning”, ICPR, 2022.

(他 1 件)

クラス間距離を考慮したデータ拡張に関する研究

TP21013 藤井駿伍　　指導教授：山下隆義

1. はじめに

深層学習モデルの学習において，データ拡張は認識精度向上に不可欠なテクニックである．mixup などの複数データを混合して生成するデータ拡張手法は，学習データに含まれない新たな多様性を獲得できる．このとき，混合するデータの選択は，ランダムサンプリングのため，学習経過において適切なクラスやデータが選択されない場合があり，学習に有効とは限らない．本研究では，特徴ベクトルに基づいてクラス間距離を算出し，学習過程で混合対象として適したクラスからデータを選択して混合するデータ拡張法を提案する．評価実験により，提案手法が一般的な物体認識データセット及び Long-tailed 物体認識データセットで認識性能を改善することを示す．

2. 複数データの混合によるデータ拡張

複数データを混合するデータ拡張手法は，単一データに対する幾何変化と比べ，学習データの見えの多様性を高めることができる．mixup[1] は，ランダムにサンプリングした画像と対応するラベルを混合することで，新たに混合データを生成する手法である．混合データは，2 枚の画像と正解ラベルを β 分布に基づいてサンプリングした比率を用いて線形補間して求める．CutMix は，画像の一部を別の画像に置き換える手法である．

これらの複数データを混合する手法は，クラス毎のサンプル数に偏りのあるデータセットを対象とする Long-tailed 物体認識にも用いられている．代表的な手法である Uniform-mixup (UniMix)[2] は，クラス毎のサンプル数を考慮して，ミニバッチ生成時のデータのサンプリング傾向と混合する比率を調整する．

これらの従来手法は，データ混合処理の改良やデータバランスを考慮した事前調整によって性能を向上させている．しかしながら，クラス間の関係性を考慮して混合データを生成するには至っていない．

3. 提案手法

本研究では，特徴ベクトルに基づくクラス間距離を考慮して，学習過程で混合するクラスを動的に選択する新たなデータ拡張法を提案する．提案手法は，学習中にネットワークが出力する特徴ベクトルを用いて，クラス間のコサイン距離を算出する．このクラス間距離を考慮し，学習過程のネットワークの出力傾向に適した混合クラスを選択する．ここで，混合クラスは，混合対象データをサンプリングするクラスを指す．図 1 に提案手法によるデータサンプリングの流れを示す．提案手法は，特徴ベクトル空間におけるクラス間距離に基づいて，混合クラスを選定する．

3.1 クラス間のコサイン距離の算出

クラス間のコサイン距離は，エポック毎に更新する．コサイン距離の算出手順を以下に示す．

1. $t-1$ エポックまで学習したネットワーク $\boldsymbol{f}_\theta$ に，学習サンプル $\boldsymbol{x}_i$ を入力し，特徴ベクトル $\boldsymbol{v}_i$ を出力する．これを全ての学習サンプルに対して行う．
2. 同一クラス（図 1 では $ClassA$）に属する学習サンプルに対するクラス確率をもとに，平均特徴ベクトル $\overline{\boldsymbol{V}_A}$ を求める．これを全てのクラスに対して行う．
3. 基準クラス（$ClassA$）と他クラス（$ClassB, ClassC$, ...）とのコサイン距離 $d_{A\to B}, d_{A\to C}, \ldots$ を算出する．これを，基準クラスを変更し，全てのクラスについて行う．$ClassA$ と $ClassB$ 間のコサイン類似度を式 (3)，コサイン距離を式 (2) に示す．

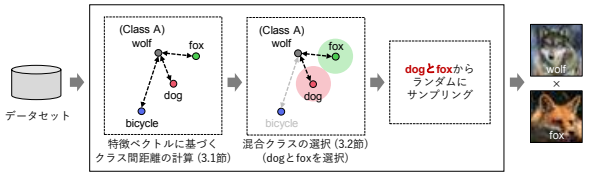


図 1：混合対象データのサンプリング処理

$$cos(\overline{\boldsymbol{V}_A}, \overline{\boldsymbol{V}_B}) = \frac{\overline{\boldsymbol{V}_A} \cdot \overline{\boldsymbol{V}_B}}{\left\|\overline{\boldsymbol{V}_A}\right\| \left\|\overline{\boldsymbol{V}_B}\right\|} = \frac{\sum_{k=1}^{dim} \overline{\boldsymbol{V}_{A_k}}\ \overline{\boldsymbol{V}_{B_k}}}{\sqrt{\sum_{k=1}^{dim} \overline{\boldsymbol{V}_{A_k}}^2} \sqrt{\sum_{k=1}^{dim} \overline{\boldsymbol{V}_{B_k}}^2}} \quad (1)$$

$$d_{A\to B} = 1 - cos(\overline{\boldsymbol{V}_A}, \overline{\boldsymbol{V}_B}) \quad (2)$$

dim は特徴ベクトルの次元数である．

3.2 クラス間距離を考慮した混合クラスの選定

算出したクラス間距離を用いて，混合クラスを選定する．処理の概要を図 2 に示す．t エポックにおいて学習サンプルを混合する際に，1 枚目のサンプルが属するクラス（図 2 では $ClassA$）とのコサイン距離に応じて，2 枚目のサンプルを選出する混合クラスを決定する．このとき，$t-2$ と $t-1$ エポック間における認識精度の変化で，選出するための距離関係 dr と混合クラス数 n を，以下の流れで基準とするクラス毎に決める．距離関係 dr は，混合クラスを選ぶ際にクラス間距離を近距離または遠距離のいずれで並び替えるかを表す．

- 1 枚目サンプルのクラスの精度が停滞・向上した場合
  - (a) 距離関係 dr は変更しない
  - (b) 混合クラス数 n は l クラス増加
- 1 枚目サンプルのクラスの精度が低下した場合
  - (a) 距離関係 dr は $t-1$ エポックが近距離の場合は遠距離，遠距離の場合は近距離に変更
  - (b) 混合クラス数 n は l クラス減少

混合クラス数 n は初期値を下限とし，1 エポックにおける増減数 l はあらかじめ定義しておく．

3.3 混合クラスの選定による学習傾向の変化

遠距離クラス混合時に精度が低下する場合，近距離クラスによって偽陽性が生じていると考えられる．そのため，t エポックでは近距離クラスを混合対象にしてクラス間距離を遠ざけるように学習する．一方，近距離クラス混合時に精度が低下する場合，距離が遠いことにより識別境界付近のデータが少ないことが要因と考えられる．そのため，t エポックでは遠距離クラスを混合対象にして識別境界を学習しやすくする．また，精度が低下した場合は精度低下の原因となるクラスに絞るため，混合クラス数を減らす．一方，精度が向上した場合は見えの多様性を向上させるために，混合クラス数を増やす．

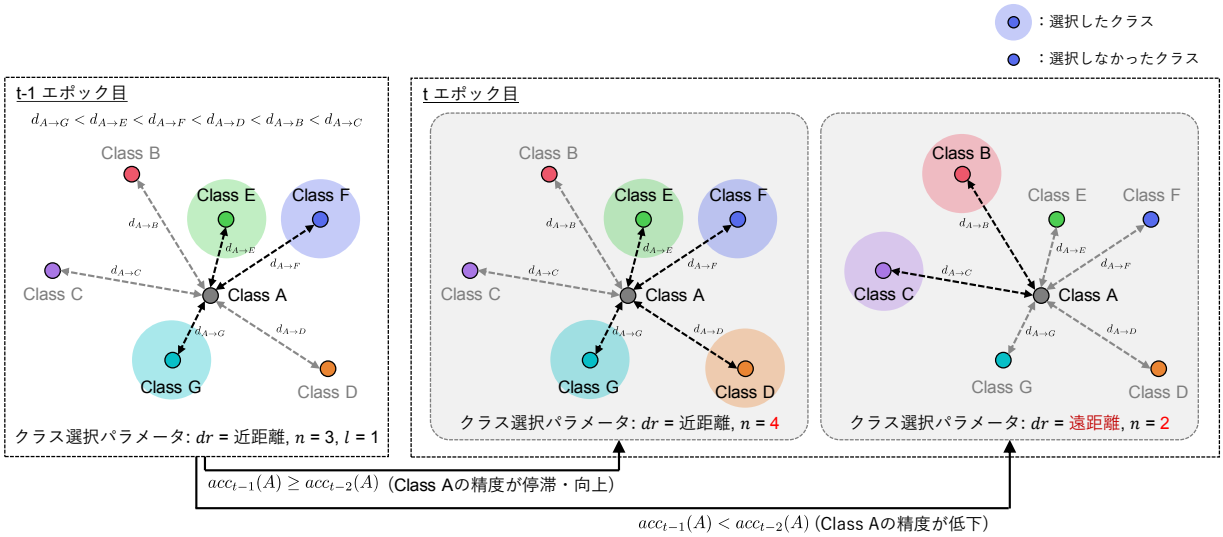


図 2：クラス間距離を考慮した混合クラスの選定処理

表 1：CIFAR-10 における精度 [%]

手法	クラス間距離	Res-32	PreAct Res-18
データ拡張無し		93.23	95.11
mixup		95.53	96.07
	✓	**95.86**	**96.20**
CutMix		95.35	96.19
	✓	**95.74**	**96.69**

表 2：CIFAR-100 における精度 [%]

手法	クラス間距離	Res-32	PreAct Res-18
データ拡張無し		74.64	75.50
mixup		**77.37**	78.38
	✓	77.26	**79.05**
CutMix		78.27	**80.38**
	✓	**80.02**	79.61

表 3：CIFAR-10-LT における精度 [%]

手法	クラス間距離	Res-32	PreAct Res-18
データ拡張無し		74.94	75.77
mixup		81.60	83.21
	✓	**83.80**	**83.94**
UniMix		86.17	87.61
	✓	**87.25**	**88.19**

表 4：CIFAR-100-LT における精度 [%]

手法	クラス間距離	Res-32	PreAct Res-18
データ拡張無し		44.02	45.53
mixup		**48.75**	49.27
	✓	48.50	**49.96**
UniMix		**53.32**	54.18
	✓	53.24	**54.40**

4. 評価実験

本章では，一般物体認識および Long-tailed 物体認識データセットを用いて，提案手法の有効性を評価する．

4.1 一般物体認識における精度比較

一般物体認識において，mixup，CutMix に提案手法を導入した際の性能を比較する．データセットは，CIFAR-10, CIFAR-100 を用いる．ネットワークモデルは，双方のデータセットにおいて ResNet-32，PreAct ResNet-18 を用いる．ミニバッチサイズは 128，エポック数は 200 とする．クラス間距離計算に用いる特徴ベクトルは，各ネットワークモデルにおける全結合層の一つ前の層から取得する．混合クラス選定の初期パラメータは，距離関係 dr を遠距離，混合クラス数 n を 5，エポック毎の n の増減数 l を 5 とする．

CIFAR-10，CIFAR-100 における各手法の認識精度をそれぞれ表 1，表 2 に示す．表より，各条件において，提案手法を導入することによって概ね精度が向上している．よって，一般物体認識において，クラス間距離を考慮した混合クラスの選定は有効であるといえる．一部の精度低下の要因は，該当の条件でモデルが多様的な混合データを十分に学習できるためであると推察する．

4.2 Long-tailed 物体認識における精度比較

Long-tailed 物体認識において，mixup，UniMix に提案手法を導入した際の精度を比較する．データセットは，CIFAR-10-LT，CIFAR-100-LT ($\rho = 50$) を用いる．ρ は，各クラスのサンプル数の比率を調整する不均衡係数である．一般的には $\rho = 10 \sim 100$ が用いられ，ρ が大きいほどサンプル数が偏った構成となる．ネットワークモデルおよび学習設定は一般物体認識の実験と同様である．

CIFAR-10-LT，CIFAR-100-LT における各手法の認識精度をそれぞれ表 3，表 4 に示す．表より，各条件において，提案手法により概ね精度が向上している．よって，Long-tailed 物体認識においても，提案手法は有効であるといえる．一部の精度低下の要因として，認識が困難なクラスの混合に制約が課されたことにより，認識が容易なクラスの混合パターンが減少したためであると推察する．

5. おわりに

本研究では，特徴ベクトルに基づくクラス間距離に応じて混合クラスを選択する手法を提案した．提案手法により，学習過程で性能向上に有効な混合クラスを選択することが可能となった．また，提案手法は，混合データを生成する従来手法に容易に導入でき，一般画像および Long-tailed 物体認識で有効であることを確認した．今後は，設計分析を継続し，データセットの構成による影響を更に抑制する設計を検討する．

参考文献

[1] H. Zhang, *et al.*, “mixup: Beyond Empirical Risk Minimization”, In ICLR, 2018.

[2] Z. Xu, *et al.*, “Towards Calibrated Model for Long-Tailed Visual Recognition from Prior Perspective”, In NeurIPS, 2021.

研究業績

[1] S. Fujii, *et al.*, “Super-class Mixup for Adjusting Training Data”, In ACPR, 2021.

(他 2 件)

Knowledge Distillation と Pruing を用いた物体検出モデルの圧縮に関する研究

TP21020 劉 履壮　　指導教授：山下 隆義

1. はじめに

物体検出を組込みデバイスで用いるには，高精度かつ高速度なモデルが必要である．Grishick らは深層学習を用いた Two-stage による物体検出モデルである R-CNN[6] を，Redmon らは One-stage の物体検出モデルである YOLO[5] を提案した．さらに, Alexey らは YOLO の特徴抽出層に，計算量を大幅削減できる CSPNet 構造と, 解像度が異なる空間的な特徴量を集約する PANet 構造を用いた YOLOv4[1] を提案した．これらの物体検出モデルのネットワーク構造は複雑かつ大規模であるため, 組込みデバイスに実装することは難しい．そこで，モデルを軽量化する必要があり，精度を維持しつつモデル圧縮が可能な枝刈り手法が注目されている．

Frankle らは枝刈り手法として，宝くじ仮説 [4] を提案した．しかし，宝くじ仮説により枝刈りしたモデルは精度が低下することが指摘されている [7]．そこで，本研究では，大規模なネットワーク出力をもとに小規模なネットワークを学習する知識蒸留と宝くじ仮説を組み合わせることで，宝くじ仮説より高精度となる学習法を提案する．

2. 関連研究

モデルの軽量化には，知識蒸留と枝刈りの 2 つのアプローチが用いられる．

2.1 知識蒸留

物体認識の知識蒸留として，Hinton らは大規模なネットワーク (教師モデル) を学習した後，教師モデルの出力情報を Soft Target として小規模なネットワーク (生徒モデル) を学習する手法 [3] を提案した．物体検出の知識蒸留として Chen らは Two-Stage の Faster R-CNN の特徴抽出層, 分類損失, 回帰損失を同時に知識蒸留する手法 [2] を提案した．知識蒸留を One-Stage 物体検出モデルに適用する際，教師モデルの出力に誤検出した検出結果や物体を重複して検出したか検出の結果も含まれるため，生徒モデル精度低下につながる．

2.2 枝刈り

深層学習モデルには，推論結果に影響しないパラメータが存在する．モデルを圧縮するには，このようなパラメータを枝刈りすることが重要である．Zhuang[8] らは BN 層の γ 係数を L1 正則化によるスパース制約項として損失関数に追加し, 出力が 0 に近い重要でないニューロンを枝刈りする手法を提案した．しかし, この手法は枝刈りに時間を要する．

Frankle ら [4] は Iterative Magnitude Pruing(IMP) 方法を用いて大規模なネットワークを枝刈りし，元のネットワークの精度と近くなる小規模なネットワーク (サブネットワーク) を見つける宝くじ仮説 [4] を提案した．この方法は, IMP 方法によって優れたスパースサブネットワークを見つけることを目的としている．IMP 法はパラメータの絶対値の大きさで，重要度を計算し，枝刈りを行う手法である．一方で，宝くじ仮説により枝刈りしたモデルはパラメータの削除率に比例して精度が低下することが指摘されている [7].

3. 提案手法

本研究では, 宝くじ仮説に知識蒸留を組み合わせて，精度低下を抑制する手法を提案する．

3.1 知識蒸留付き宝くじ仮説学習

本研究の流れを図 1 に示す，まず Frankle らが提案した宝くじ仮説により物体検出モデルを枝刈りする．次に，枝刈りする前の大規模なモデルを教師モデルとし，教師モデルから蒸留した知識を用いて枝刈りした小規模なモデルを学習する．これらの流れの詳細を以下に示す．

Step1. 物体検出モデルのパラメータを初期化する．

Step2. k エポック学習したパラメータ $\boldsymbol{w_k}$ を保存する．

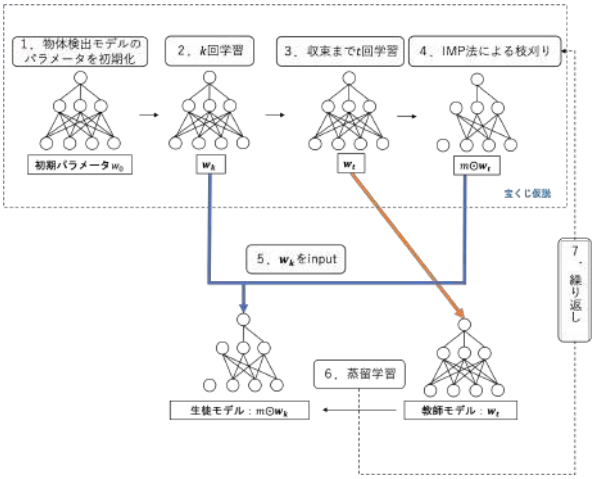


図 1: 提案手法の学習流れ

Step3. 元ネットワークを収束するまで学習し, 重みを $\boldsymbol{w_t}$ を保存する．

Step4. $\boldsymbol{w_t}$ 中で各値の絶対値が低いものから枝刈りする．重みマスク $\boldsymbol{m}$ を保存する．

Step5. $\boldsymbol{m} \odot \boldsymbol{w_k}$ をネットワークに適用し，生徒モデルとする．

Step6. Step3. で保存した $\boldsymbol{w_t}$ をもつ教師モデルとし，生徒モデルが収束するまで蒸留学習をする．

Step7. 目標パラメータに達成するまで，Step4. から Step6. まで繰り返す．

3.2 損失関数

知識蒸留を物体検出に用いる場合，誤検出した検出した結果も蒸留してしまう．本研究では教師モデルの検出結果に信頼度 (Confidence) を利用して, 蒸留の損失関数を設計する．誤検出の信頼度は目標物体の信頼度出力より小さいことが多いため, 教師モデルと生徒モデルの分類 (Classification) の損失計算式の係数とする．これにより，蒸留した知識から誤検出の影響を抑制できる．損失関数を式 (1) に示す．

$$L_{loss} = L_{hard}(s, G) + \alpha L_{soft}(s, t) \tag{1}$$

s は生徒モデル $m \odot w_k$ の出力，G は正解ラベル，t は教師モデル w_t の出力，L_{hard} は生徒モデル出力と正解ラベルの損失, L_{soft} は生徒モデル出力と教師モデル出力の損失である．L_{hard} と L_{soft} を混合する割合をパラメータ α によって調整する．本研究では α を 1 とする.L_{hard} 及び L_{soft} は以下の式となる．

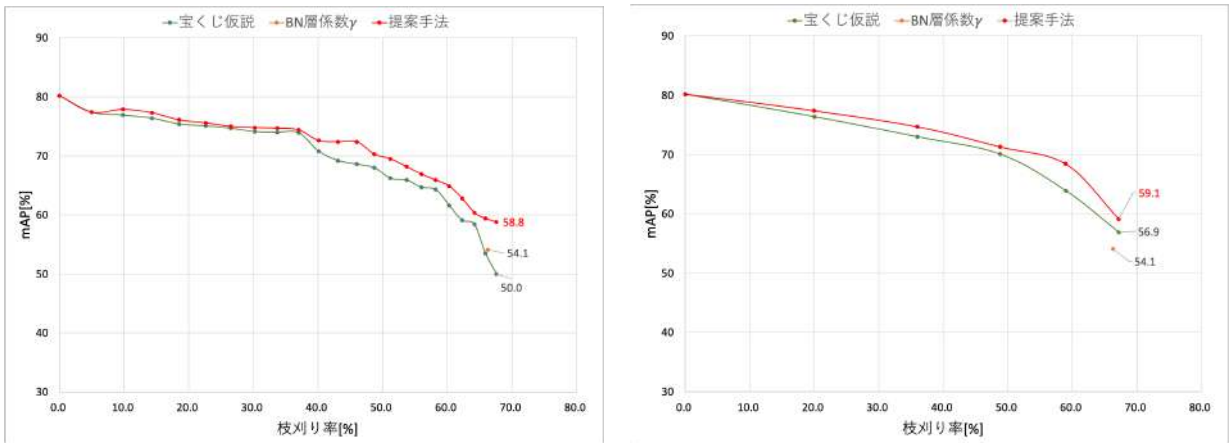
$$L_{hard}\,(s, G) = L_{conf}(s, G) + L_{cls}(s, G) + L_{reg}(s, G) \tag{2}$$

$$L_{soft}\,(s, t) = L_{con_{kd}}(s, t) + conf_t L_{cls_{kd}}(s, t) \tag{3}$$

L_{conf} は生徒モデルと正解ラベルの信頼度損失, L_{cls} は生徒モデルと正解ラベルの分類損失, L_{reg} は生徒モデルと正解ラベルのバウンディングボックス回帰の損失である．$L_{conf_{kd}}$ は生徒モデルと教師モデルの信頼度損失, $L_{cls_{kd}}$ は生徒モデルと教師モデルの分類損失である．また, $conf_t$ は教師モデルの信頼度出力である．ここで $L_{conf_{kd}}$, $L_{cls_{kd}}$ は全て平均二乗誤差 (MSELoss) で算出する．

4. 評価実験

提案手法の有効性を確認するため，VOC データセット用いて従来の宝くじ仮説による学習と比較実験を行う．



(a) 1 回の刈り率を 5%とした時　　(b) 1 回の刈り率を 20%とした時

図 2: mAP 変化の比較

表 1: モデルのパラメータ数と mAP の比較

手法	枝刈り回数 (枝刈り率/回)	残り パラメータ数	mAP [%]
YOLOv4		6.404×10^6	80.4
BN 層 係数 γ	1 (67%)	2.152×10^6	54.1
宝くじ仮説 提案手法	22 (5%)	2.088×10^6	50.0 **58.8**
宝くじ仮説 提案手法	5 (20%)	2.101×10^6	56.9 **59.1**

4.1 実験概要

VOC2007 及び VOC2012 を合わせて本実験の学習用データセットとして使用する．物体検出モデルには YOLOv4[1] を用いる．学習用及びテスト用データセットの画像枚数はそれぞれ 16,551 枚と 4,9952 枚である．精度の評価指標には, mean Average Precision(mAP) を用いる．

4.2 定量的評価

1 回の枝刈り率を 5%または 20%にして，枝刈りを繰り返した時の mAP を図 2 に示す．1 回の枝刈り率を 5%にした時，提案手法及び宝くじ仮説と共に枝刈り率が 40%までは精度低下が緩やかである．しかし，40%超えると宝くじ仮説は大きく精度が低下する．一方で提案手法は宝くじ仮説よりも精度低下を抑制することができている．1 回に枝刈り率を 20%にした時も同様に提案手法の精度低下は宝くじ仮説より緩やかである．また，表 1 より提案手法のモデルは Zhuang らの手法よりパラメータ削減でき，精度が向上したことが分かる．この結果から, 本提案手法で用いる知識蒸留は精度向上に貢献したといえる．

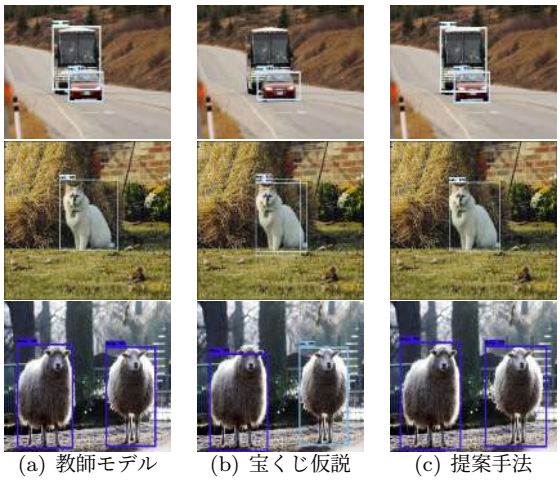
4.3 定性的評価

図 3 に検出結果例を示す．図 3 の 1 行目より，提案手法は教師モデルに近い位置精度であることが確認できる．図 3 の 2 行目より，宝くじ仮説の場合，1 つの犬を重複して検出している．一方，提案手法のモデルは 1 つの物体として検出している．図 3 の 3 行目より，宝くじ仮説では誤検出しているケースに対して提案手法は正確に検出していることが確認できる．

5. おわりに

本研究では, 知識蒸留と宝くじ仮説手法を用いた One-stage 物体検出モデルの圧縮学習法を提案した．提案手法を用いて学習したモデルは，従来手法により同程度枝刈りしたモデルと比べ, 精度低下が緩やかであり，モデルの精度を向上させることができた．

今後は, 他のデータセットによる検証と異なる枝刈り率で枝刈りしたモデルの間の比較実験を行う予定である．



(a) 教師モデル　(b) 宝くじ仮説　(c) 提案手法

図 3: 出力結果の比較

参考文献

[1] A.Bochkovskiy *et al.* Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[2] G.Chen *et al.* Learning efficient object detection models with knowledge distillation. *NeaurlIPS*, 2017.

[3] G.Hinton *et al.* Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[4] J.Frankle *et al.* The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[5] J.Redmon *et al.* You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[6] R.Girshick *et al.* Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[7] X.Ma *et al.* Sanity checks for lottery tickets: Does your winning ticket really win the jackpot? *NeaurlIPS*, 2021.

[8] Z.Liu *et al.* Learning efficient convolutional networks through network slimming. In *ICCV*, 2017.

研究業績

[1] L.Liu, *et al.*, “CLASS-WISE FM-NMS FOR KNOWLEDGE DISTILLATION OF OBJECT DETECTION”, IEEE ICIP, 2022.

(他 1 件)