

東京大学
情報理工学系研究科 電子情報学専攻
修士論文

Wireless Ad Hoc Federated Learning による物体検出と
半教師あり学習への拡張

Object Detection Using Wireless Ad Hoc Federated Learning and the
Extension to Semi-Supervised Learning

48-236447
山口 竜平
Ryuhei Yamaguchi

指導教員 落合秀也 准教授

2025 年 1 月

概要

物体検出は、自動運転や監視など、さまざまなアプリケーションにとって不可欠になっている。農業、ヘルスケア、ロボット工学などの現場特有の環境に適用する場合、プライバシー保護が求められるデータを含むことが多く、クラウドストレージへのデータ保存が不適切となるケースがある。またこうした特定のターゲット画像に対しては追加的な学習が必要となるが、近年エッジデバイス向けの AI チップの登場により、デバイス上で完結する物体検出モデルの学習が可能になりつつある。そして Device-to-Device 通信により分散協調的な学習を行うアプローチが現実味を帯びてきている。こうした背景を踏まえ、本研究では、Wireless Ad Hoc Federated Learning(WAFL) による物体検出を提案し、DETR と YOLO の二つのモデルにより、学習量・通信量も考慮した 3 つの手法によって、その性能を検証した。加えて、実際のエッジ環境による学習を再現するために、データセットを構築し、それによる性能検証も行った。また半教師あり物体検出を WAFL に拡張した手法を提案し、ラベル付きデータの割合ごとに性能比較を行った。

Abstract

Object detection has become indispensable in a wide range of applications, including autonomous driving and surveillance. In particular, when adapting to environment-specific conditions in fields such as agriculture, healthcare, and robotics, data often contain privacy-sensitive information that is unsuitable for storage on the cloud. Consequently, additional training is required for these specific target images. Recent advances in AI chips designed for edge devices have made it increasingly feasible to train object detection models entirely on-device. Furthermore, approaches leveraging device-to-device communication for distributed, collaborative learning are becoming more practical. Against this backdrop, the present study proposes object detection using Wireless Ad Hoc Federated Learning (WAFL). We employ two models, DETR and YOLO, and evaluate their performance through three methods that take both training load and communication overhead into account. In addition, we construct a dataset to replicate real-world edge environments and conduct performance evaluations using this dataset. We also propose a semi-supervised object detection method by extending WAFL, and compare performance across different proportions of labeled data.

目次

第 1 章	序論	1
1.1	本研究の背景	1
1.2	本研究の目的	2
1.3	本論文の構成	2
第 2 章	関連研究と課題	3
2.1	Federated Learning	3
2.1.1	学習形態の分類	3
2.1.2	メリットと課題	5
2.1.3	WAFL	6
2.2	物体検出	6
2.2.1	物体検出の評価指標	7
2.2.2	物体検出のモデル	7
2.3	Federated Learning による物体検出	8
第 3 章	提案手法	10
3.1	WAFL-DETR	10
3.1.1	Full Parameter Exchange (FPE)	10
3.1.2	Transformer Layer Exchange (TLE)	10
3.1.3	Head Exchange (HE)	10
3.2	WAFL-YOLO	11
3.2.1	Full Parameter Exchange (FPE)	11
3.2.2	Detection Head Exchange (DHE)	11
3.2.3	Head Last Exchange (HLE)	12
3.3	WAFL-SSOD	12
第 4 章	評価	15
4.1	WAFL-DETR のベンチマークデータセットによる評価	15
4.1.1	設定	15
4.1.2	評価結果	17

4.2	WAFL-YOLO のベンチマークデータセットによる評価	21
4.2.1	設定	21
4.2.2	評価結果	21
4.3	WAFL-DETR のエッジ環境における性能評価	24
4.3.1	設定	24
4.3.2	評価結果	26
4.4	WAFL-YOLO のエッジ環境における性能評価	29
4.4.1	設定	29
4.4.2	評価結果	29
4.5	WAFL-SSOD の評価	31
4.5.1	設定	31
4.5.2	評価結果	31
第 5 章	考察	33
5.1	ベンチマークデータセットによる評価	33
5.2	エッジ環境を再現したデータセットによる評価	33
5.3	WAFL-SSOD	34
第 6 章	結論	35
6.1	まとめ	35
6.2	今後の課題	35
	発表文献と研究活動	37
	参考文献	38
付録 A	ソースコード	42

目次

2.1	分散的な機械学習手法の分類	4
2.2	WAFL の概要	6
2.3	DETR のアーキテクチャ	8
3.1	DETR による Wireless Ad Hoc Federated Learning (WAFL-DETR) の概要	11
3.2	YOLOv9 のアーキテクチャ	12
3.3	Semi-Supervised Object Detection の概要	13
4.1	評価に用いる WAFL のトポロジー	16
4.2	ベンチマークデータセットによる WAFL-DETR-FPE, TLE, HE の mAP の 推移	19
4.3	ベンチマークデータセットによる WAFL-YOLO-FPE, DHE, HLE の mAP の推移	23
4.4	各居室におけるアノテーション付きデータの一例	25
4.5	エッジ環境を想定して作成したデータセットによる WAFL-DETR-FPE, TLE, HE の mAP の推移	27
4.6	エッジ環境を想定して作成したデータセットによる WAFL-YOLO-FPE, DHE, HLE の mAP の推移	30
4.7	WAFL-SSOD の第二段階の擬似ラベルを含む協調学習における各デバイス の mAP の推移	32

表目次

4.1	ベンチマークデータセットの分布	16
4.2	ベンチマークデータセットにおけるカテゴリごとのアノテーションの分布 . .	16
4.3	ベンチマークデータセットによる WAFL-DETR の mAP と通信量 (モデル 交換・学習手法による比較)	18
4.4	ベンチマークデータセットによる WAFL-DETR の mAP (トポロジーによる 比較)	18
4.5	学習初期段階のモデルと学習終了時のモデルにより推論されたバウンディング ボックスの可視化と比較 (ベンチマークデータセット)	20
4.6	ベンチマークデータセットによる WAFL-YOLO の mAP と通信量 (モデル 交換・学習手法による比較)	22
4.7	ベンチマークデータセットによる WAFL-YOLO の mAP (トポロジーによ る比較)	22
4.8	エッジ環境を想定して作成したデータセットのカテゴリ	24
4.9	エッジ環境を想定して作成したデータセットのアノテーションの分布	25
4.10	エッジ環境を想定して作成したデータセットによる WAFL-DETR の mAP と通信量 (モデル交換・学習手法による比較)	26
4.11	学習初期段階のモデルと学習終了時のモデルにより推論されたバウンディング ボックスの可視化と比較 (エッジ環境を想定して作成したデータセット) . .	28
4.12	エッジ環境を想定して作成したデータセットによる WAFL-YOLO の mAP と通信量 (モデル交換・学習手法による比較)	30
4.13	ラベル付きデータの各割合に対する WAFL-SSOD の mAP の比較	32

第 1 章

序論

1.1 本研究の背景

従来の機械学習では、多くの場合ユーザのデバイスや企業内で保持されているデータをクラウドなどのサーバ上にアップロードし、大規模な計算資源を用いて学習を行っていた。この手法は学習効率が高く、大規模データセットを一括して扱えるという利点があるものの、データが第三者の管理下に置かれるため、外部へのデータ流出や不正アクセスによる情報漏えいなどのリスクが常につきまとっていた。さらにユーザが自分のデータをどのように利用されるかわからないまま、データを提供することに対する不安も根強く、特にセンシティブなデータを取り扱う分野では、プライバシー保護の観点から従来の中央集約型アプローチには課題があると指摘されてきた。こうしたプライバシーやセキュリティ上の懸念は、2018 年の EU 一般データ保護規則 (GDPR)[1] の施行を契機に世界的に高まってきた。GDPR は個人データの取扱いに関する厳格なルールを定めており、違反時には高額な制裁金が科される可能性がある。そのため、企業や研究機関はデータの取り扱い方針を見直し、不要なデータの収集や長期保存を避けるなど、個人情報保護する仕組みを整備する必要性に迫られている。機械学習の分野においても、モデルの開発時におけるデータ収集の方法や管理体制の見直しが進められ、データを扱う際にはプライバシー侵害を回避するための技術的な対策の導入が強く求められている。

さらに近年では、AI チップの登場によって、エッジ環境における計算能力が向上してきている。これまで学習プロセスは複雑かつ膨大な計算リソースを必要とするため、主に高性能なサーバや GPU クラスタ上で実行されるのが一般的であった。しかし、エッジデバイスに搭載されるチップの性能向上や、省電力化技術の進歩により、IoT デバイスやスマートフォンなどの端末でも推論のみならず学習さえ行うことが可能になりつつある [2]。IoT デバイスの急速な普及も、このエッジ AI 環境の重要性を一層高めている。あらゆるセンサがインターネットに接続され、多種多様なデータがリアルタイムに生成される現状では、中央サーバにすべてのデータを集約すること自体が通信面・コスト面で大きな負担となる。また、膨大なセンサデータをすべてクラウド上で処理するのは、リアルタイム性や耐障害性といった観点からも必ずしも最適とはいえない。こうした状況を踏まえ、エッジデバイスのローカル計算資源を活用し、必要最低限の情報だけを集約・共有する分散型の手法が期待される [3]。

このような背景のもと注目を浴びているのが、Federated Learning (FL) [4] をはじめとする分散協調的な機械学習の手法である。FL では、学習に必要な生データを中央サーバに送る代わりに、各デバイス上でローカルにモデルを学習し、その学習結果として得られたパラメータのみをサーバ側に送信する。サーバは各デバイスから受け取ったパラメータを集約・更新し、再度各デバイスに新しいグローバルモデルを配信するというサイクルを繰り返す。これにより、プライバシー保護を図りつつ、大規模データの分散学習を実施できるだけでなく、エッジデバイスの計算リソースを活用し、ネットワーク帯域の効率化にも寄与するとされている。さらに、各デバイス上で学習を行うことで、個々のユースケースに即したモデルのカスタマイズが容易になるなど、新たな応用シナリオの拡大が期待されている。

総じて、クラウド中心の機械学習からエッジ分散型の機械学習へと移行する潮流は、データプライバシーやセキュリティの要請、高度化するエッジデバイスの性能、そして膨大な IoT データをリアルタイムかつ効率的に活用したいという需要の高まりによって加速している。これらの課題に応えるための技術として、Federated Learning に代表される新しい分散協調的な学習手法は今後さらに重要度を増していくと考えられる。

基本的な AI タスクの一つである物体検出においても、こうした潮流は例外ではなく、実際に FL を用いた物体検出の検証が行われてきた。一方で、これら先行研究における学習手法は完全分散型のアーキテクチャになっておらず、依然として、クラウドのような中央集権的なサーバを前提としたものになっている。

1.2 本研究の目的

本研究では、中央集権的なサーバを排した状態で、複数のエッジデバイスに分散している物体検出モデルを協調的に学習させることで、学習をエッジデバイス上で完結させたまま、高精度で汎用的な物体検出モデルを獲得することを目的とする。そのために Wireless Ad Hoc Federated Learning (WAFL) [5] によって物体検出モデルのパラメータの一部を交換しつつ、学習を進める枠組みを提案した。また協調的な学習におけるボトルネックの一つである、通信量に関して、3つの手法で比較を行い、推論精度とのトレードオフを検証した。また画像認識などのタスクと比べてユーザのアノテーションコストが高くなる物体検出においては、ユーザの保有するラベル付きデータが少数であっても、ユーザが学習に参加できるようにすることが望ましい。そのため、ユーザの保持しているラベル付きデータが全体のデータの一部になることを許容する、分散協調的な半教師あり物体検出の手法を提案し、その有効性を検証した。

1.3 本論文の構成

本論文は以下のような構成になっている。第2章では本研究に関連する技術や先行研究について説明を行う。第3章では提案手法について述べる。第4章では評価を行い、第5章でその考察を行う。最後に第6章で結論と今後の課題について述べる。

第 2 章

関連研究と課題

2.1 Federated Learning

Federated Learning は中央サーバに学習データを送信せずに、エッジデバイス上で学習を行い、学習済みモデルを中央サーバに送り、サーバ上でモデルを統合する学習手法である。Federated Learning の代表的なアルゴリズムとして Federated Averaging [4] が挙げられる。具体的な手順は **Algorithm 1** のようになっている。ここで K が全クライアント数、 n_k がクライアントの持つデータ数である。ハイパーパラメータとして学習に参加するクライアントの割合は C 、クライアントにおける学習のバッチサイズは B 、そのエポック数は E で表されている。まず複数のクライアントと中央のサーバが機械学習のモデルを持っている。これらのモデルをそれぞれローカルモデル、グローバルモデルと呼ぶ。一つのラウンドではランダムに参加するクライアントが選ばれ、中央サーバはグローバルモデルを、選んだクライアントに送る。選択されたクライアントは自分のローカルモデルを自分の持つデータで学習する。その後、各クライアントは自身のローカルモデルのパラメータを中央サーバに送る。中央サーバは各クライアントから送られたパラメータに対し、各クライアントの持つデータ数によって加重平均をとったものを新たなグローバルモデルとする。以上の手順を複数のラウンドにわたって行うことで、各クライアントのデータを他のクライアントやサーバと共有することなく、汎化性能の高いモデルを獲得することを目指す。

2.1.1 学習形態の分類

分散的な機械学習の手法は、図 2.1 に示すようにネットワークポロジやクライアント間での共有情報において複数の手法に分類される。それらの分類について紹介する。

分散機械学習 (Distributed Machine Learning)

これは中央サーバが、全ての学習データを保持しており、モデルパラメータと分割したデータセットを各クライアントに配布する形態を指す。データを中央サーバにアップロードしているので Federated Learning の枠には含まれないが、分散する計算資源を活用して学習を高速

Algorithm 1 Federated Averaging [4]

```

1: Server executes:
2: initialize  $w_1$ 
3: for each round  $t = 1, 2, \dots$  do
4:    $m \leftarrow \max(C \cdot K, 1)$ 
5:    $S_t \leftarrow$  (random set of  $m$  clients)
6:   for each client  $k \in S_t$  in parallel do
7:      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$ 
8:   end for
9:    $w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} w_{t+1}^k$  ( $n = \sum_{k \in S_t} n_k$ )
10: end for
11: ClientUpdate( $k, w$ ):
12:  $\mathcal{B} \leftarrow$  (split  $\mathcal{P}_k$  into batches of size  $B$ )
13: for each local epoch  $i = 1, 2, \dots, E$  do
14:   for batch  $b \in \mathcal{B}$  do
15:      $w \leftarrow w - \eta \nabla l(w; b)$ 
16:   end for
17: end for
18: return  $w$ 

```

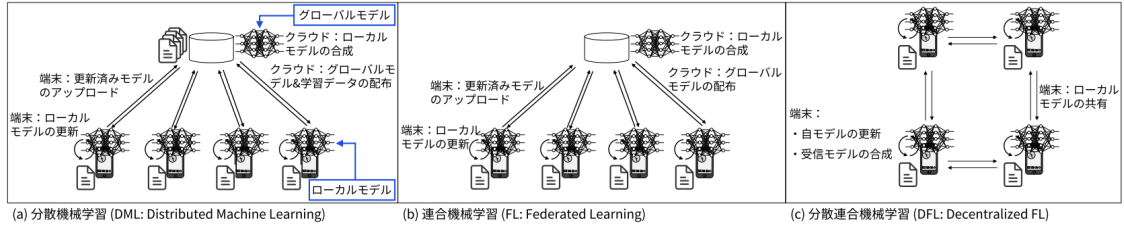


図 2.1. 分散的な機械学習手法の分類 [6]

化する場合などに用いられる方式である。

連合機械学習 (Federated Learning)

ここまで説明してきた Federated Learning のことであり、クライアントは自身の持つ学習データを開示せずに、ネットワーク上に分散した無数のデータから単一の精度の高いモデルの構築を目指す。モデルの統合は中央のサーバで行われる。

分散連合機械学習 (Decentralized Federated Learning)

Federated Learning の派生形であり、クラウドや中央サーバに一切頼らない学習形態である。クライアント上で学習を行うとともに、他のクライアントとモデルの交換を行い、モデルの

統合もクライアント上で行う。更新された全てのクライアント間でモデルの交換を行う [5] や隣接したデバイス間での無線通信によるモデル交換で学習を進める [6] のような手法がある。

2.1.2 メリットと課題

Federated Learning のメリット

Federated Learning のメリットとしては主に次の 3 つが挙げられる。

1. プライバシーの問題の解消

Federated Learning ではデータがクライアントから開示されないので、プライバシーを保護したまま学習を行うことができる。この問題は特に医療系のデータを利用したモデルの学習において、患者のプライバシーを保護したまま、実質的に大量のデータを学習に用いることを可能にする点が注目に値する。

2. 計算資源の分散

解析データが肥大化するほど、クラウド側の計算資源に対する負荷も増大する。特に IoT デバイスにより取得される大量のデータを処理する必要がある環境では、この重要性が増す。

3. 通信コスト

学習対象が複雑であるほど、多くの学習データが要求される。一般にモデルのパラメータの方が、利用されるデータよりもサイズが小さく、その分の通信負荷を軽減できる。

Decentralized Federated Learning のメリット

Decentralized Federated Learning のメリットとしては上記に加えてさらに以下の 2 つが考えられる。

4. 単一障害点の解消

中央サーバに依存しないため、サーバの故障により学習プロセスが中断されることがない。また中央サーバを介さないために、サーバを管理するサードパーティによる運営方針の変更などで全体のサービスが影響を受けることがなくなる。

5. マルチベンダ環境の実現

システムの構造上、中央サーバの管理者を必要としないため、複数の異なる事業者のエッジデバイスが学習に参加しやすい。学習に参加する事業者を増やすことは、実質的に学習に使用できるデータの増加につながるため、最終的に得られるモデルの性能向上に寄与しうる。

課題

課題の一つとして、クライアントの持つデータの異質性が挙げられる。各クライアントの持つデータは一般にサイズが異なるとともに、分布も異なっている。各データは独立同一分布に

従うことがなく、この特性は Non-IID と呼ばれる。このようなクライアントデータの Non-IID 特性が学習に悪影響を与え、サーバで集中的な学習を行った場合と比較して、精度が下がってしまうという問題がある。またモデル性能と通信量のトレードオフも課題として挙げられる。交換するパラメータサイズを増やすことで、より汎化性能の高いモデルの獲得が期待できるが、その分通信量が増加し、学習のレイテンシにつながりうる。

2.1.3 WAFL

Wireless Ad Hoc Federated Learning (WAFL) [5] は DFL の一種として提案されている。従来の DFL がインターネットを介した P2P による通信でモデル交換を行うことを想定しているのに対して、WAFL は物理的に近接したノード同士で無線アドホック通信を行い、Device-to-Device 通信によるモデル交換を行うことで、自律分散的かつ協調的に学習を進める新しい枠組みである。WAFL による学習を想定した手法として、WAFL-ViT [7], WAFL-GAN [8], WAFL-AutoEncoder [9] などが挙げられる。またモデルのパーソナライゼーションを考慮した WAFL-Personalization [10, 11], 自己位置推定を行う WAFL-Localization [12], マルチタスク学習の MT-WAFL [13] など提案されている。セキュリティ面からのアプローチとして、モデル汚染攻撃に対する耐性 [14] も研究されている。

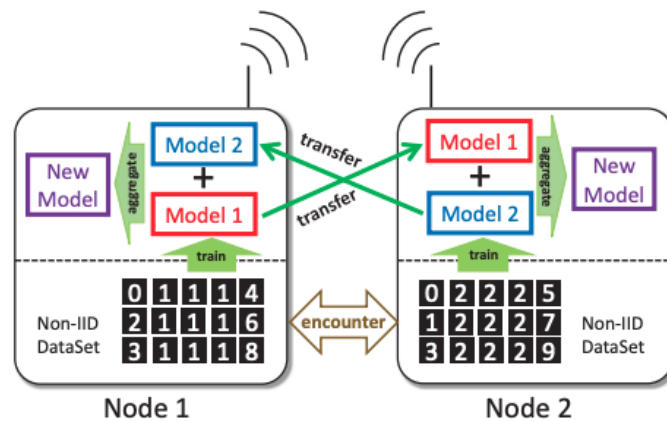


図 2.2. WAFL の概要 [5]

2.2 物体検出

物体検出とは、画像の中に含まれる物体を取り囲む矩形領域を特定するとともに、その物体の所属するクラスを分類することを目標としたコンピュータビジョンのタスクの一つである。物体を取り囲む矩形領域をバウンディングボックスと呼ぶ。

2.2.1 物体検出の評価指標

IoU(Intersection over Union)

物体検出の結果を評価するためには, 予測したバウンディングボックスと真のバウンディングボックスの一致度合いを定量化する必要がある. この評価指標を IoU(Intersection over Union) と呼び, 次式のように定義される. なお B_p は予測したバウンディングボックス, B_{gt} は正解ラベルのバウンディングボックスを表す.

$$\text{IoU} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})}$$

mAP(Mean Average Precision)

物体検出モデルの評価指標として, mAP(Mean Average Precision) が使われる. mAP はクラスごとに AP(Average Precision) の平均をとった値である. 以下では AP について説明する. あるクラスの物体を検出することを考えた場合, 検出の閾値として IoU を利用する. 例えば $\text{IoU} = 0.5$ を閾値として設定した場合を考える. この時, 予測したバウンディングボックスが正解ラベルのバウンディングボックスと結びついていて, $\text{IoU} \geq 0.5$ で十分に重なっている場合は TP(True Positive), $\text{IoU} < 0.5$ で十分に重なっていない場合は FP(False Positive) となる. また正解ラベルのバウンディングボックスが予測したバウンディングボックスと結びついていない場合, FN(False Negative) となる. 各バウンディングボックスは信頼度スコアを持っているので, このスコアに基づいて予測を降順に並べ, TP, FP, FN の値から Precision と Recall を順に求めることで, PR 曲線を描くことができる. Recall を r , その Recall に対する Precision を $p(r)$ とすると, $p(r)$ はより高い Recall \tilde{r} で発生する Precision で置き換えられる. この PR 曲線の下部の面積が AP である. AP は等間隔の Recall でのサンプリングによって近似的に計算される場合がある.

$$\text{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} p(r)$$

クラス k の平均適合率を AP_k , クラス数を N_c とすると,

$$\text{mAP} = \frac{1}{N_c} \sum_{k=1}^{N_c} \text{AP}_k$$

2.2.2 物体検出のモデル

YOLO [15]

多くの物体検出手法のように物体検出をクラスの分類とバウンディングボックスの回帰に分けるのではなく, それらをまとめて回帰問題と捉える手法として YOLO[14] が提案された. 領域提案やスライディングウィンドウの手法と異なり画像全体を見る点が特徴的で

ある。まず画像全体を $S \times S$ のグリッドセルに分割し、グリッドセルごとに B 個のバウンディングボックスを予測する。各バウンディングボックスは 4 つの座標と信頼度を値としてもち、信頼度は $Pr(Object) * IoU$ で定義される。それとは別に各グリッドは C 個の条件付きクラス確率 $Pr(Class_i|Object)$ を予測する。テスト時にはこれらを掛け合わせた値である $Pr(Class_i) * IoU$ をバウンディングボックスごとに信頼度として使用する。これは各バウンディングボックスについてそのクラスが出現している確率とバウンディングボックスが物体にフィットしている度合いの両方を意味する。単一回帰問題として推論を行うことで計算を高速化したとともに、モデル全体で End-to-End で一度に学習可能することが可能となった。

DETR [16]

直接集合予測で物体検出のタスクを解く、Transformer を利用したモデルとして DETR が提案された。モデル全体は図 2.3 のようになっており、backbone の CNN により特徴量の抽出を行った後、Transformer による演算を行うパイプラインとなっている。モデルの出力部分は 2 つの Feed Forward Network で構成され、1 つは物体のクラスを出力し、もう一方はバウンディングボックスの座標を出力する。これらの出力は各クエリに対して行われる。Non Maximum Suppression のようなヒューリスティックな手法を必要としない点が特徴的である。あらかじめ決めた数 n 個のクエリによる推論と Ground Truth の最小二部マッチングを損失とする。

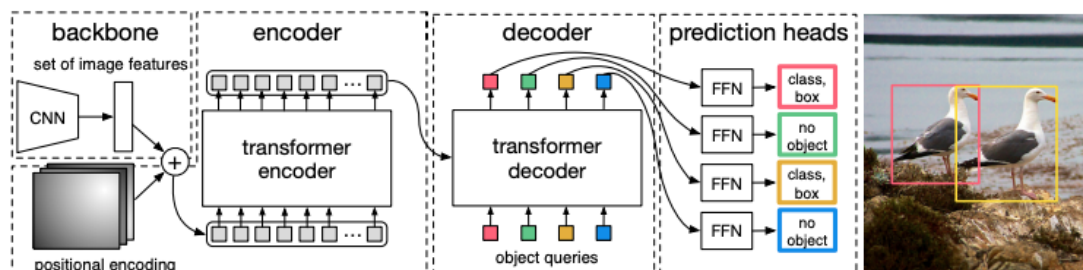


図 2.3. DETR のアーキテクチャ [16]

2.3 Federated Learning による物体検出

Luo らはストリートカメラによる撮影で得た画像から Federated Learning による物体検出のためのデータセットを作成した。また同データセットを用いて、Federated Averaging に基づいて、Faster R-CNN, YOLOv3, YOLOv3(pretrained) の 3 種類のモデルを使った Federated Learning の性能検証を行っている [17]。結果として、物体検出のような複雑なモデルを必要とするタスクにおいても、素朴に平均をとってモデルを統合する手法が有効であることがわかった。一方でデータの Non-IID 特性によって、モデルの性能低下が引き起こされるという課題は残っている。

Yu らは Pascal VOC データセットを用いて、物体検出のモデルである SSD300 の Federated

Learning による性能を検証した。また Kullback-Leibler Divergence に基づき、学習時の重みの分布間の距離を測定した。加えて重みの異常値をクリップする手法 (Abnormal Weights Supression) を提案し、Non-IID 特性による精度低下の改善を試みている [18]。結果として、Federated Averaging が SSD300 の学習に有効であることがわかった。一方で Non-IID のデータ環境ではこの手法は精度で劣るが、Abnormal Weights Supresison は極端な方向への統合モデルのシフトを緩和し、精度向上に有効であることがわかった。

いずれの先行研究においても、クライアント数の少ない条件下では Federated Averaging により、集中型の学習に近い mAP を達成することができている。一方でクライアント数が多い場合や、Non-IID 特性を強めた場合は mAP が著しく低下してしまうため、改善の余地が残されている。またこれらの先行研究はいずれも通常の Federated Learning の環境を想定しており、Decentralized Federated Learning のような派生形での性能検証はなされていない。加えてモデル交換の際にすべてのパラメータを利用しており、モデルの一部のパラメータのみを使用するような手法やそれに関連する通信量の議論は行われていない。そして Transformer を用いた物体検出モデルである DETR などの新しいモデルによる検証もなされていない。以上の点が現状での研究の課題といえる。

第 3 章

提案手法

3.1 WAFL-DETR

複数の DETR のパラメータの一部を Wireless Ad Hoc Federated Learning によって、交換、統合をしながら学習を進めることで、全体で汎化性能の高い DETR を獲得する枠組みとして、WAFL-DETR を提案する。図 3.1 に WAFL-DETR の全体像を示す。WAFL-DETR ではそれぞれのデバイスが自身の DETR を用いて、自身の環境で作成されたデータセットを使い学習を行う。ここで検出対象となる物体とそのアノテーションは特定のカテゴリに偏る場合がある。最終的に獲得するモデルでは、自身のデータでアノテーションがなされていないような物体や、アノテーションの数が少ない物体も検出できるようにすることを目指す。またモデル性能と通信量のトレードオフを考慮して、以下の 3 つの手法によってパラメータの交換と学習を行う。

3.1.1 Full Parameter Exchange (FPE)

この手法では DETR の全てのパラメータの交換及び学習を行う。

3.1.2 Transformer Layer Exchange (TLE)

この手法では DETR の中でバックボーンネットワークのパラメータは固定し、残りの Transformer 層を中心としたパラメータのみを更新する。更新したパラメータのみ交換を行う。この手法は、画像の特徴抽出を担うバックボーンネットワークの事前学習済みパラメータが下流タスクによらず、汎用的であるという仮定に基づく。

3.1.3 Head Exchange (HE)

この手法では DETR の最終層のフィードフォワードネットワークのみ更新を行い、その他のパラメータは固定する。更新したパラメータのみ交換を行う。通信量・学習量を抑えることで、現実での利用を念頭に置いている。

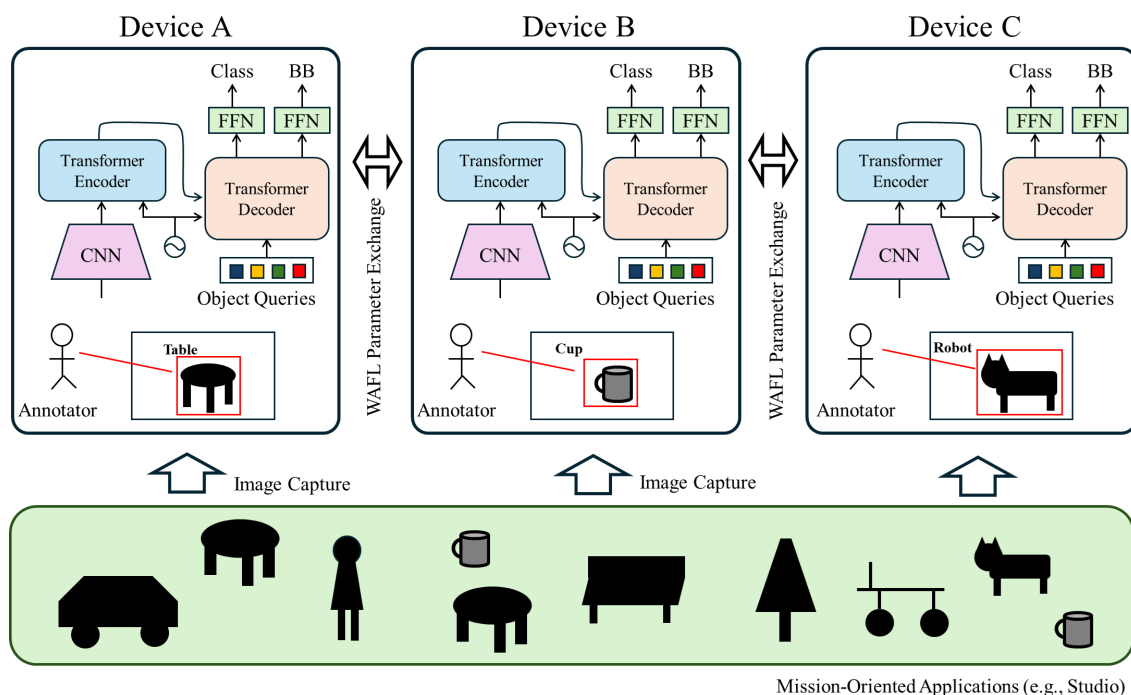


図 3.1. DETR による Wireless Ad Hoc Federated Learning (WAFL-DETR) の概要

3.2 WAFL-YOLO

複数の YOLOv9 [19] のパラメータの一部を Wireless Ad Hoc Federated Learning によって、交換、統合しながら学習を進めることで、全体で汎化性能の高い YOLOv9 を獲得する枠組みとして、WAFL-YOLO を提案する。WAFL-DETR と同様に、モデル性能と通信量のトレードオフを考慮して、以下の 3 つの手法によってパラメータの交換と学習を行う。YOLOv9 のアーキテクチャについては、図 3.2 に示している。

3.2.1 Full Parameter Exchange (FPE)

この手法では YOLOv9 の全てのパラメータの交換及び学習を行う。

3.2.2 Detection Head Exchange (DHE)

この手法では YOLOv9 の Detection Head の部分の全てのパラメータの交換及び学習を行う。なおここでの Detection Head は 2 つのヘッド両方を指している。

3.2.3 Head Last Exchange (HLE)

この手法では YOLOv9 の Detection Head 内部の最後の畳み込み演算のパラメータのみ交換及び学習を行う。なおここでの Detection Head は 2 つのヘッド両方を指している。

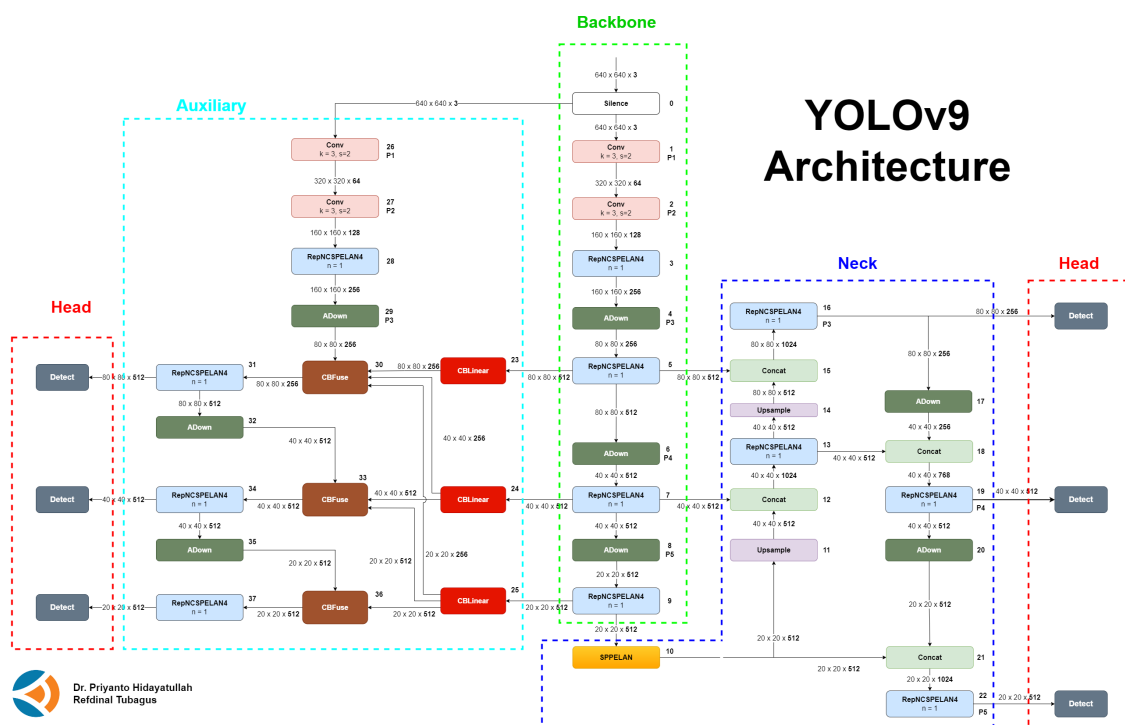


図 3.2. YOLOv9 のアーキテクチャ [20]

3.3 WAFL-SSOD

物体検出モデルの学習に必要なデータセットのアノテーションのコストを抑える手法として Semi-Supervised Object Detection(SSOD) が知られている [21]. SSOD ではラベル付きのデータを用いて学習した教師モデルによって、ラベルなしデータについての推論を行い、その推論結果に基づいて、ラベルなしデータに擬似ラベルを付与する。次にラベル付きデータと擬似ラベルを付与したラベルなしデータの両方を利用して再び学習を行う。本研究ではこの SSOD を WAFL に拡張した WAFL-SSOD を提案する。WAFL-SSOD では、初期状態として、それぞれのクライアントがラベル付きデータとラベルなしデータの両方をもつ。まずクライアントは、WAFL によってモデルの交換と統合を行いつつ、自身の持つラベル付きデータのみを使って学習を進める。次に学習が収束したそれぞれのローカルモデルを用いて、ラベルなしデータについての推論をローカルで行い、この結果を用いて、自身の持つラベルなしデータに擬似ラベルを付与する。最後にラベル付きデータと擬似ラベルを付与したラベルなしデータの両方を用いて、再び WAFL により、モデルの交換と統合を行いつつ、学習を進める。

WAFL-SSOD の疑似コードを **Algorithm2** に示す. ここで N は全デバイス数, P, T_1, T_2 はそれぞれ pre-self training, 第一段階の WAFL, 第二段階の WAFL のラウンド数を示している. また $\mathcal{D}_L^n, \mathcal{D}_{UL}^n, \mathcal{D}_{Pse}^n$ はそれぞれデバイス n の持つラベル付きデータセット, ラベルなしデータセット, 疑似ラベルの付与されたラベルなしデータセットを示している. τ は疑似ラベル生成のための信頼度スコアの閾値を示している.

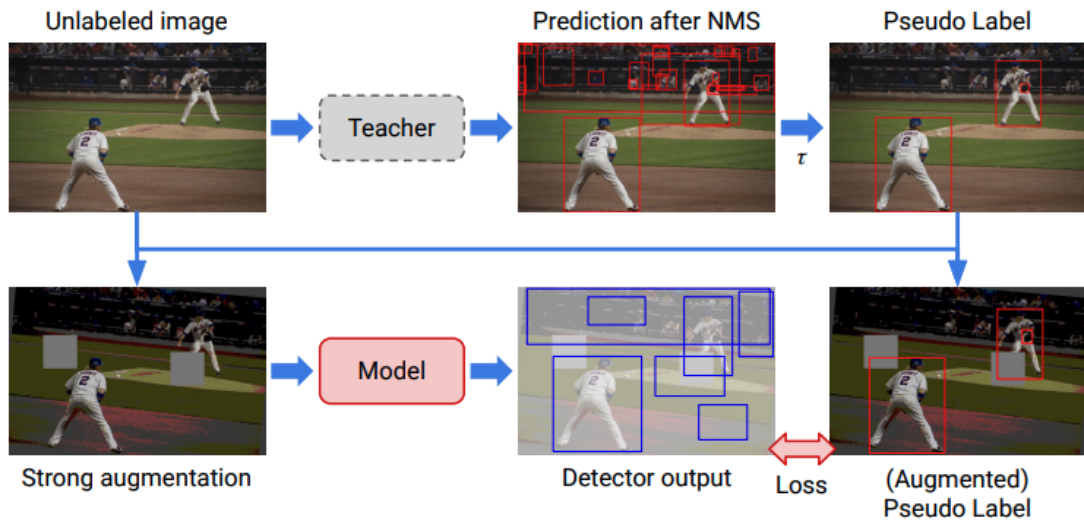


図 3.3. Semi-Supervised Object Detection の概要 [21]

Algorithm 2 WAFL-SSOD

```

1: initialize  $w_0^0, w_0^1, \dots, w_0^{N-1}$ 
   //pre-self training
2: for each local epoch  $p = 0, 1, \dots, P - 1$  do
3:   for each device  $n = 0, 1, \dots, N - 1$  in parallel do
4:      $w_0^n \leftarrow \text{DeviceUpdate}(n, w_0^n, \mathcal{D}_L^n)$ 
5:   end for
6: end for
   //WAFL aggregation and Local update (step1)
7: for each round  $t = 0, 1, \dots, T_1 - 1$  do
8:   for each device  $n = 0, 1, \dots, N - 1$  in parallel do
9:      $w_t^n \leftarrow w_t^n + \lambda \frac{\sum_{k \in \text{nbr}(n)} (w_t^k - w_t^n)}{|\text{nbr}(n)| + 1}$ 
10:     $w_{t+1}^n \leftarrow \text{DeviceUpdate}(n, w_t^n, \mathcal{D}_L^n)$ 
11:   end for
12: end for
13: for each device  $n = 0, 1, \dots, N - 1$  in parallel do
14:   Make pseudo labels  $\mathcal{D}_{Pse}^n$  from  $(\mathcal{D}_{UL}^n, w^n, \tau)$ 
15: end for
   //WAFL aggregation and Local update (step2)
16: for each round  $t = 0, 1, \dots, T_2 - 1$  do
17:   for each device  $n = 0, 1, \dots, N - 1$  in parallel do
18:      $w_t^n \leftarrow w_t^n + \lambda \frac{\sum_{k \in \text{nbr}(n)} (w_t^k - w_t^n)}{|\text{nbr}(n)| + 1}$ 
19:      $w_{t+1}^n \leftarrow \text{DeviceUpdate}(n, w_t^n, \mathcal{D}_L^n \cup \mathcal{D}_{Pse}^n)$ 
20:   end for
21: end for
22: DeviceUpdate( $n, w, \mathcal{D}$ ):
23:  $\mathcal{B} \leftarrow (\text{split } \mathcal{D} \text{ into batches of size } B)$ 
24: for each local epoch  $i = 1, 2, \dots, E$  do
25:   for batch  $b \in \mathcal{B}$  do
26:      $w \leftarrow w - \eta \nabla l(w; b)$ 
27:   end for
28: end for
29: return  $w$ 

```

第 4 章

評価

4.1 WAFL-DETR のベンチマークデータセットによる評価

4.1.1 設定

モデルと学習設定

モデルは MS COCO データセット [22] で事前学習した重みをロードした DETR を使用する。クライアント数は 10 とし、各クライアントの位置は固定されているものとする。ネットワークのトポロジーは line, tree, ringstar の 3 種類とする。各トポロジーについて、図 4.1 に示す。各クライアントは自身の持つデータセットで 100 エポックの pre-self training を行なった後、WAFL によって隣接するクライアントとのモデル交換を同期的に行いながら、200 エポックの学習を行う。学習率は 10^{-5} 、Backbone の学習率は 10^{-6} とし、最適化関数には AdamW を使用する。アグリゲーションの係数 λ は 1.0 で設定する。

データセット

学習に使用するデータセットは Open Image Dataset [23] から 10 カテゴリを選んで作成したものとする。IID と Non-IID の両方のシナリオで評価を行う。Non-IID のシナリオでは、一般にクライアントの持つデータセットはサイズが異なるとともに、その分布も異なる。このような各クライアントの持つデータセットの Non-IID 特性を再現するために、1 つのクライアントが 1 つのカテゴリのデータの 90% を保有するように設定する。それぞれのシナリオについて、各クライアントの持つデータセットの分布を表 4.1 に示す。なお複数のアノテーションを持つデータに関しては、最もアノテーション数の多いカテゴリをそのデータのカテゴリとして割り当てる。全データのアノテーションの分布は表 4.2 のようになっている。

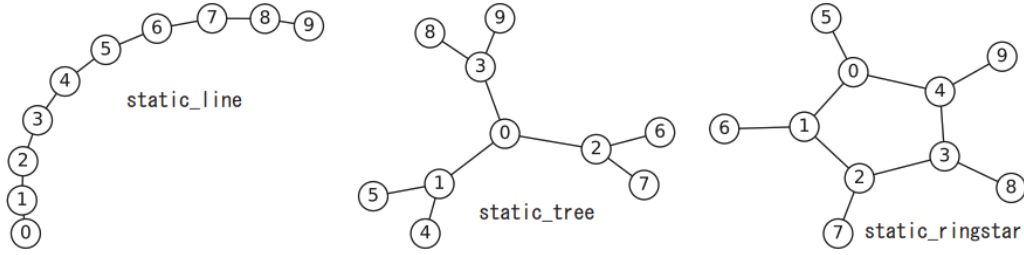


図 4.1. 評価に用いる WAFL のトポロジー

表 4.1. ベンチマークデータセットの分布

Device ID	IID Dataset											Non-IID Dataset										
	Person	Bicycle	Car	Table	Door	Fire hydrant	Waste container	Ball	Cat	Drink	Sum	Person	Bicycle	Car	Table	Door	Fire hydrant	Waste container	Ball	Cat	Drink	Sum
0	43	46	50	46	40	34	45	43	43	55	445	436	4	6	1	8	4	6	4	3	3	475
1	47	46	51	34	56	37	52	66	59	60	508	8	424	9	1	5	3	4	5	5	7	471
2	49	51	49	55	54	38	53	39	46	44	478	9	8	449	7	4	5	6	2	9	9	508
3	56	50	63	47	37	33	45	45	48	42	466	5	5	6	433	12	4	4	7	10	4	490
4	49	39	53	39	50	41	58	51	52	52	484	4	7	4	2	445	6	8	5	9	6	496
5	50	56	41	47	55	50	44	54	43	47	487	6	3	2	6	3	368	3	1	15	7	414
6	55	44	46	51	48	41	52	56	42	52	487	4	8	6	2	1	4	449	7	3	8	492
7	37	59	50	53	55	44	53	51	42	47	491	4	5	6	3	4	5	3	451	5	6	492
8	49	45	50	55	45	47	48	38	69	46	492	6	4	4	11	3	2	11	3	434	5	483
9	52	39	47	45	51	42	49	52	51	53	481	5	7	8	6	6	6	5	10	2	443	498
Sum	487	475	500	472	491	407	499	495	495	498	4819	487	475	500	472	491	407	499	495	495	498	4819

表 4.2. ベンチマークデータセットにおけるカテゴリごとのアノテーションの分布

Annotation Category	Person	Bicycle	Car	Table	Door	Fire hydrant	Waste container	Ball	Cat	Drink
Person	1922	18	24	20	0	0	0	3	0	1
Bicycle	7	1083	42	4	3	1	0	0	0	0
Car	9	20	1459	1	2	0	1	0	0	1
Table	2	0	0	915	2	0	0	2	2	10
Door	0	3	0	0	720	0	0	0	3	0
Fire hydrant	0	0	0	0	0	431	0	0	0	0
Waste container	0	0	0	0	1	0	906	0	2	0
Ball	0	0	0	1	0	0	0	912	1	0
Cat	0	0	0	0	1	0	0	0	597	2
Drink	4	0	0	21	0	0	0	0	0	1327

4.1.2 評価結果

表 4.3 に各条件下での mAP_{50} の値と通信量の比較を示している。mAP はクライアント数を 10 と仮定しているため、各デバイスの最終 10 エポックの平均 mAP について、デバイス数 10 でさらに平均をとったものを採用している。表中の s.d. はデバイスごとの mAP についての標準偏差を示している。また表中の traffic は 1 つのデバイスが、1 エポックに交換するパラメータのサイズを MByte で示している。この比較ではトポロジーは line で固定している。

IID と Non-IID の両方のケースにおいて、WAFL-DETR-TLE がそれぞれ 58.99, 54.31 の mAP を達成しており、他の手法よりも高い結果となった。この性質は単一モデルの学習 (DETR-TL) においても見られるが、WAFL の場合の方がその差がより顕著となっている。WAFL-DETR-HE は WAFL-DETR-FPE と比較して、通信量を約 $1/260$ に抑えながらも IID で 47.86, Non-IID で 46.67 の mAP を達成している。またいずれの WAFL による手法も、モデル単独での学習の場合と比較して高いスコアとなった。Centralized のケースと比較しても、IID と Non-IID のそれぞれの場合で、その差が約 1 ポイントと約 6 ポイントという結果となり、特に IID の場合は Centralized のケースにも比肩する結果となった。

各手法についての学習曲線を図 4.2 に示す。FPE と TLE は、学習開始直後は同程度の性能であったが、その後は TLE の方が性能が向上している。HE は低いスコアから始まっているものの、200 エポック時点では他の手法と比肩するスコアに推移している。

トポロジーごとに mAP_{50} の値を比較した結果を表 4.4 に示す。この評価は手法ごとの比較において最も精度が高かった WAFL-DETR-TLE の条件下で行った。IID, Non-IID の両方の場合について、ringstar が最も良い結果となったものの、line の mAP と比較して、精度向上はそれぞれ約 0.4, 1.0 ポイントであり、トポロジーによる大きな精度差は見られなかった。

表 4.5 に WAFL による学習の初期段階でのモデルを使用した場合の推論結果と 200 エポックの WAFL による学習終了後のモデルを使用した場合の推論結果を示している。なお学習手法は WAFL-DETR-TLE で、Non-IID データセットにより学習したモデルを使用している。元々自分の持つデータセットに多くを占めるクラスしか検出できていないモデルが、WAFL による学習を通じて、最終的にそうでないクラスの物体も検出できていることがわかる。

表 4.3. ベンチマークデータセットによる WAFL-DETR の mAP と通信量 (モデル交換・学習手法による比較)

		mAP ₅₀	s.d.	epochs	traffic
Decentralized (IID)	WAFL-DETR-FPE	54.70	0.67	200	157.48
	WAFL-DETR-TLE	58.99	0.44	200	68.85
	WAFL-DETR-HE	47.86	0.13	200	0.61
	DETR-FP	49.52	1.17	100	n/a
	DETR-TL	50.33	1.07	100	n/a
	DETR-H	39.67	1.69	100	n/a
Decentralized (Non-IID)	WAFL-DETR-FPE	51.37	1.99	200	157.48
	WAFL-DETR-TLE	54.31	1.81	200	68.85
	WAFL-DETR-HE	46.67	0.77	200	0.61
	DETR-FP	30.97	3.59	100	n/a
	DETR-TL	31.07	4.24	100	n/a
	DETR-H	7.73	4.26	100	n/a
Centralized	DETR-FP	59.54	n/a	30	n/a
	DETR-TL	60.87	n/a	30	n/a
	DETR-H	46.10	n/a	30	n/a

表 4.4. ベンチマークデータセットによる WAFL-DETR の mAP (トポロジーによる比較)

	topology	mAP ₅₀	s.d.	epochs
Decentralized (IID)	line	58.99	0.44	200
	tree	59.03	0.30	200
	ringstar	59.38	0.30	200
Decentralized (Non-IID)	line	54.31	1.81	200
	tree	53.85	1.34	200
	ringstar	55.20	1.39	200

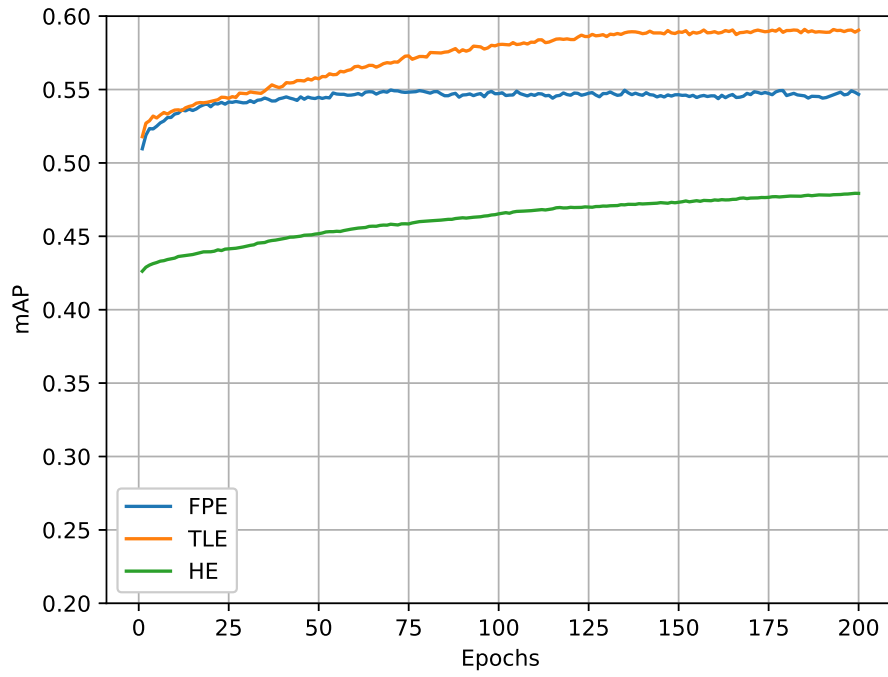
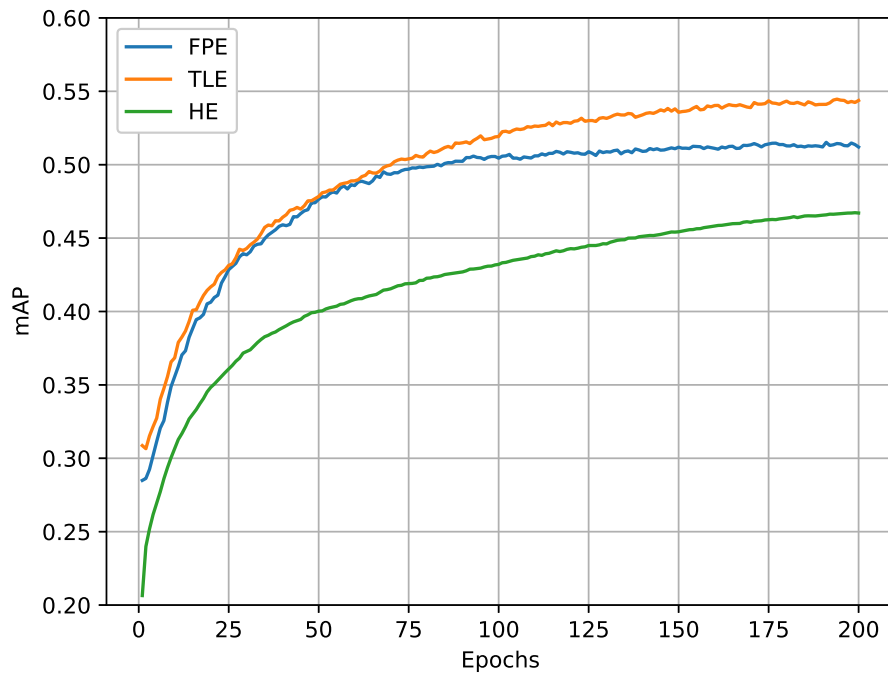






(a) mAP₅₀ on IID Dataset(b) mAP₅₀ on Non-IID Dataset

図 4.2. ベンチマークデータセットによる WAFL-DETR-FPE, TLE, HE の mAP の推移

表 4.5. 学習初期段階のモデルと学習終了時のモデルにより推論されたバウンディングボックスの可視化と比較 (ベンチマークデータセット)

Training in Progress		Trained at Epoch 200
Device A's Model	Device B's Model	Final Model
		
		
		

4.2 WAFL-YOLO のベンチマークデータセットによる評価

4.2.1 設定

モデルと学習設定

モデルは MS COCO データセット [22] で事前学習した重みをロードした YOLOv9-c を使用する。クライアント数は 10 とし、各クライアントの位置は固定されているものとする。ネットワークのトポロジーは line, tree, ringstar の 3 種類とする。各クライアントは自分の持つデータセットで 30 エポックの pre-self training を行なった後、WAFL によって隣接するクライアントとのモデル交換を同期的に行いながら、100 エポックの学習を行う。学習率は 10^{-3} とし、最適化関数には SGD を使用する。アグリゲーションの係数 λ は 1.0 で設定する。

データセット

学習に使用するデータセットは Open Image Dataset [23] から 10 カテゴリを選んで作成したものとする。IID と Non-IID の両方のシナリオで評価を行う。Non-IID のシナリオでは、各クライアントの持つデータセットの Non-IID 特性を再現するために、1 つのクライアントが 1 つのカテゴリのデータの 90% を保有するように設定する。なお複数のアノテーションを持つデータに関しては、最もアノテーション数の多いカテゴリをそのデータのカテゴリとして割り当てる。

4.2.2 評価結果

表 4.6 に各条件下での mAP_{50} の値と通信量の比較を示している。 mAP はクライアント数を 10 と仮定しているため、各デバイスの最終 10 エポックの平均 mAP について、デバイス数 10 でさらに平均をとったものを採用している。なおいずれの手法でも 90 エポック付近から mAP の低下が見られるため、90 エポックで早期終了とした場合の mAP を算出している。表中の s.d. はデバイスごとの mAP についての標準偏差を示している。また表中の traffic は 1 つのデバイスが、1 エポックに交換するパラメータのサイズを MByte で示している。この比較ではトポロジーは line で固定している。

IID と Non-IID の両方のケースにおいて、WAFL-YOLO-FPE がそれぞれ 62.80, 64.10 の mAP を達成しており、他の手法よりも高い結果となった。WAFL-YOLO-HLE は WAFL-YOLO-FPE と比較して、通信量を約 $1/4800$ に抑えながらも IID で 40.82, Non-IID で 40.81 の mAP を達成している。またいずれの WAFL による手法も、モデル単独での学習の場合と比較して高いスコアとなった。加えて FPE と DHE は Centralized のケースの mAP も上回る結果となった。また Non-IID 特性による明らかな精度低下が見受けられなかった。WAFL が Centralized のケースを上回った点については、高い学習率によって単独のモデルだと不安定になっていた学習が、モデルの平均化によって安定した可能性がある。

各手法についての学習曲線を図 4.3 に示す。FPE と TLE は、学習開始直後は同程度の性能

表 4.6. ベンチマークデータセットによる WAFL-YOLO の mAP と通信量 (モデル交換・学習手法による比較)

		mAP ₅₀	s.d.	epochs	traffic
Decentralized (IID)	WAFL-YOLO-FPE	62.80	0.17	90	194.62
	WAFL-YOLO-DHE	58.28	0.13	90	82.26
	WAFL-YOLO-HLE	40.82	0.87	90	0.04
	YOLO-FP	52.66	1.15	30	n/a
	YOLO-DH	51.96	1.14	30	n/a
	YOLO-HL	35.69	2.25	30	n/a
Decentralized (Non-IID)	WAFL-YOLO-FPE	64.10	0.40	90	194.62
	WAFL-YOLO-DHE	58.01	0.33	90	82.26
	WAFL-YOLO-HLE	40.81	0.89	90	0.04
	YOLO-FP	45.89	2.58	30	n/a
	YOLO-DH	48.83	1.94	30	n/a
	YOLO-HL	29.10	4.60	30	n/a
Centralized	YOLO-FP	62.49	n/a	30	n/a
	YOLO-DH	57.67	n/a	30	n/a
	YOLO-HL	40.89	n/a	30	n/a

表 4.7. ベンチマークデータセットによる WAFL-YOLO の mAP (トポロジーによる比較)

	topology	mAP ₅₀	s.d.	epochs
Decentralized (IID)	line	62.80	0.17	90
	tree	63.65	0.17	90
	ringstar	63.63	0.26	90
Decentralized (Non-IID)	line	64.10	0.40	90
	tree	63.99	0.37	90
	ringstar	63.98	0.29	90

であったが、その後は FPE の方が性能が向上している。

トポロジーごとに mAP₅₀ の値を比較した結果を表 4.7 に示す。この評価は手法ごとの比較において最も精度が高かった WAFL-YOLO-FPE の条件下で行った。IID, Non-IID の両方の場合について、トポロジーによる大きな精度差は見られなかった。

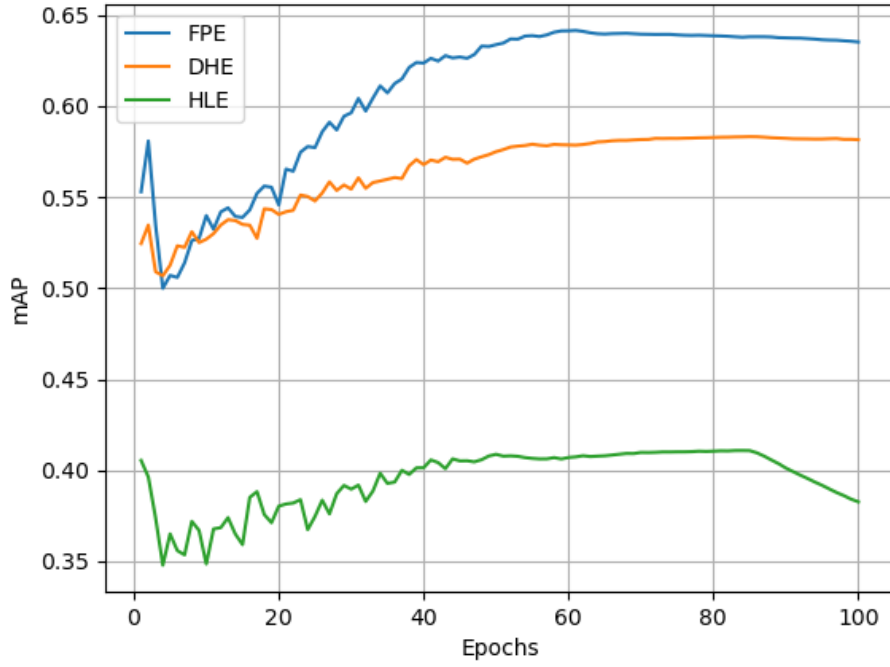
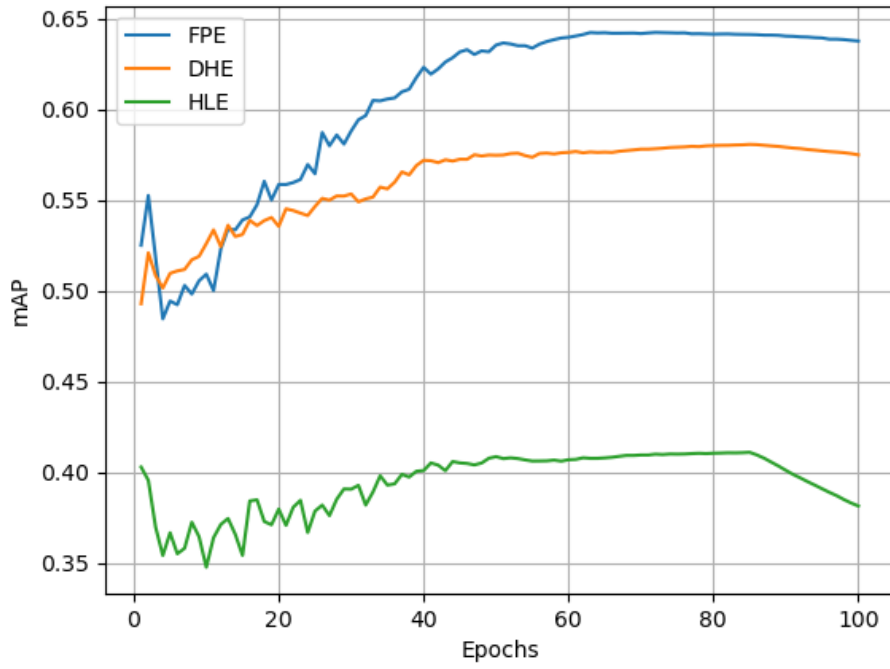
(a) mAP₅₀ on IID Dataset(b) mAP₅₀ on Non-IID Dataset

図 4.3. ベンチマークデータセットによる WAFL-YOLO-FPE, DHE, HLE の mAP の推移

4.3 WAFL-DETR のエッジ環境における性能評価

Wireless Ad Hoc Federated Learning による物体検出が実際に応用される際には、前項で評価に使用した多様な背景や一般的な物体を含むオープンな環境ではなく、その環境にしかない物体を検出するような、特殊な環境におけるタスクに対処する必要がある。このような実際のエッジ環境に近い環境で、より実践的な評価を行うために、複数の部屋における画像の撮影を行い、新たにデータセットを構築し、性能評価を行った。

4.3.1 設定

モデルと学習設定

モデルは MS COCO データセット [22] で事前学習した重みをロードした DETR を使用する。クライアント数は 6 とし、各クライアントの位置は固定されているものとする。ネットワークのトポロジーは line とする。各クライアントは自分の持つデータセットで 100 エポックの pre-self training を行なった後、WAFL によって隣接するクライアントとのモデル交換を同期的に行いながら、600 エポックの学習を行う。学習率は 10^{-5} 、Backbone の学習率は 10^{-6} とし、最適化関数には AdamW を使用する。アグリゲーションの係数 λ は 1.0 で設定する。

データセット

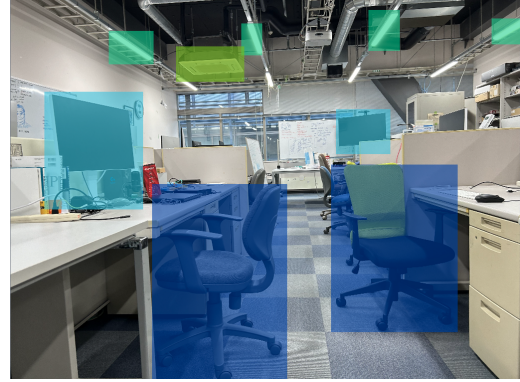
学習に使用するデータセットは東京大学工学部 2 号館 10 階の江崎・落合研究室の 4 つの居室において撮影した画像から構築したものを使用する。4 つの居室を便宜的に e-room, o-room, a-room, b-room と呼ぶ。データセットの一部を図 4.4 に示す。また検知対象となる 9 つの物体カテゴリを表 4.8 に示す。6 つのクライアントのうち、2 つのクライアントが e-room, 2 つのクライアントが o-room, 残りそれぞれ 1 つのクライアントが a-room, b-room を担当するようにデータセットの分割を行う。各クライアントの持つデータセットのアノテーションの分布を表 4.9 に示す。

category	name
0	switch
1	monitor
2	keyboard
3	light
4	books
5	airconditioner
6	cardboardbox
7	chair
8	server

表 4.8. エッジ環境を想定して作成したデータセットのカテゴリ



e-room



o-room



a-room



b-room

図 4.4. 各居室におけるアノテーション付きデータの一例

device	room	switch	monitor	keyboard	light	books	airconditioner	cardboardbox	chair	server
0	e-room	30	246	29	154	188	40	219	88	51
1	e-room	39	198	25	132	155	19	232	68	54
2	o-room	22	371	94	127	76	16	43	201	5
3	o-room	15	439	74	97	72	17	66	309	3
4	a-room	0	48	23	29	0	15	56	102	0
5	b-room	38	126	19	30	0	8	372	39	19

表 4.9. エッジ環境を想定して作成したデータセットのアノテーションの分布

4.3.2 評価結果

表 4.10 に各条件下での mAP_{50} の値と通信量の比較を示している。mAP はクライアント数を 6 と仮定しているため、各デバイスの最終 10 エポックの平均 mAP について、デバイス数 6 でさらに平均をとったものを採用している。表中の s.d. はデバイスごとの mAP についての標準偏差を示している。また表中の traffic は 1 つのデバイスが、1 エポックに交換するパラメータのサイズを MByte で示している。

WAFL-DETR-FPE が 62.58 の mAP を達成しており、他の手法よりも高い結果となった。一方で WAFL-DETR-TLE の mAP は 61.80 となり、FPE に比肩する精度となっている。またいずれの WAFL による手法も、モデル単独での学習の場合と比較して高いスコアとなった。

各手法についての学習曲線を図 4.5 に示す。FPE の収束が早いものの、600 エポック付近で TLE も同等の mAP に収束している。HE は 200 エポック付近で収束しており、そこから mAP の向上が見られない。

表 4.11 に WAFL による学習の 100 エポック時点でのモデルを使用した場合の推論結果と WAFL による 600 エポックの学習終了後のモデルを使用した場合の推論結果を示している。なお学習手法は WAFL-DETR-TLE を使用している。一行目の e-room の例では、WAFL によって新たに Light と Cardboardbox の検出に成功している。また Monitor の誤検出が解消されている。二行目の o-room の例では Cardboardbox や Light を新たに検出している。また複数の Cardboardbox が集まっている箇所のボックスの重なりが改善されている。三行目の a-room の例では Light を新たに検出している。四行目の b-room の例では Switch と Cardboardbox を新たに検出している。

表 4.10. エッジ環境を想定して作成したデータセットによる WAFL-DETR の mAP と通信量
(モデル交換・学習手法による比較)

		mAP ₅₀	s.d.	epochs	traffic
Decentralized (Non-IID)	WAFL-DETR-FPE	62.58	0.66	600	157.48
	WAFL-DETR-TLE	61.80	0.63	600	68.85
	WAFL-DETR-HE	24.58	0.07	600	0.61
	DETR-FP	34.49	8.84	100	n/a
	DETR-TL	31.96	6.49	100	n/a
	DETR-H	9.71	3.78	100	n/a
Centralized	DETR-FP	68.69	n/a	100	n/a
	DETR-TL	64.08	n/a	100	n/a
	DETR-H	26.35	n/a	100	n/a

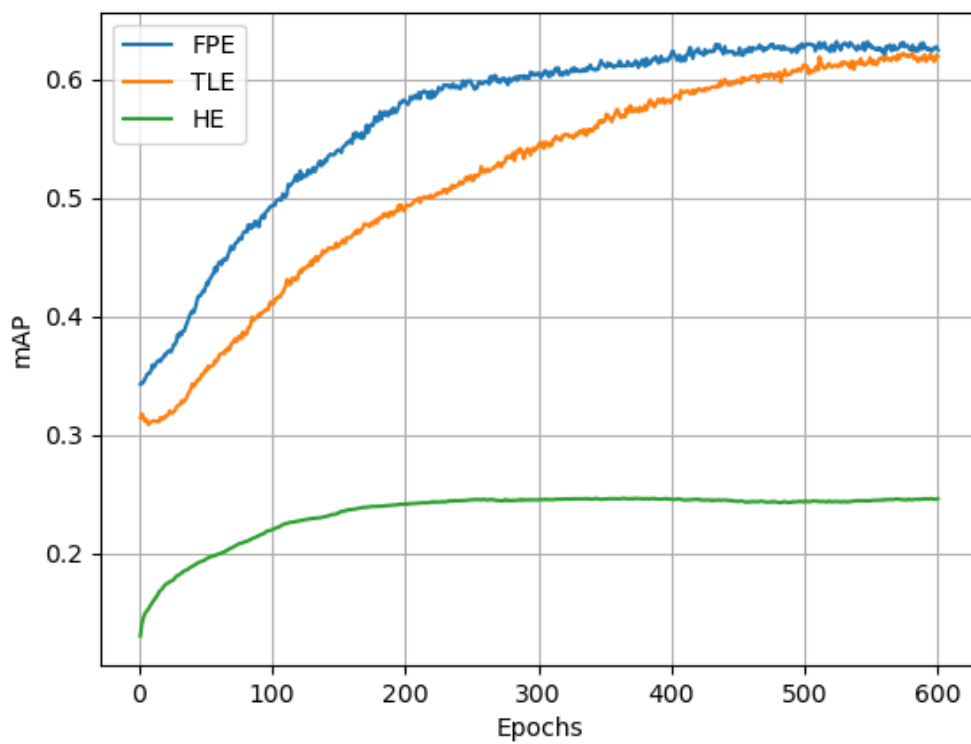

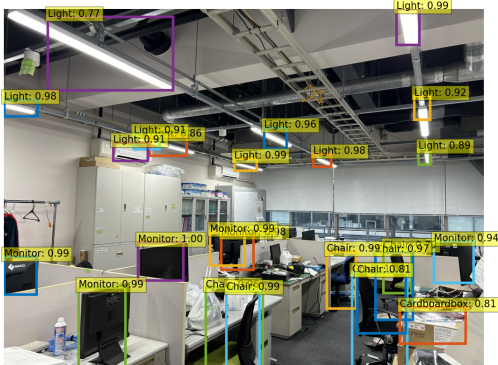
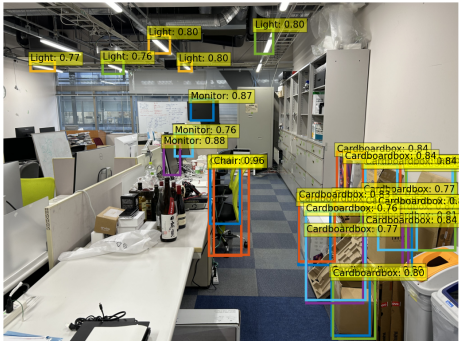
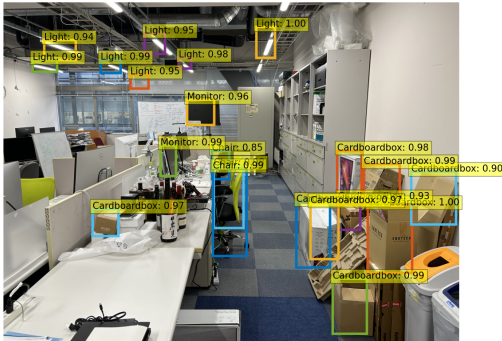
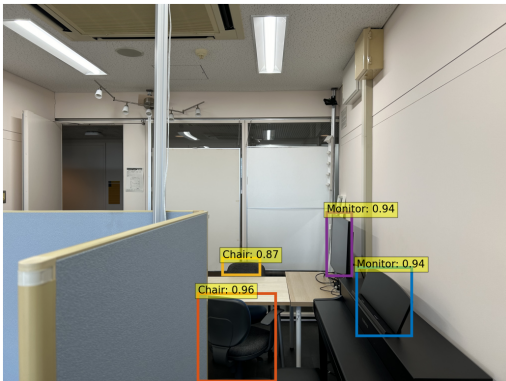
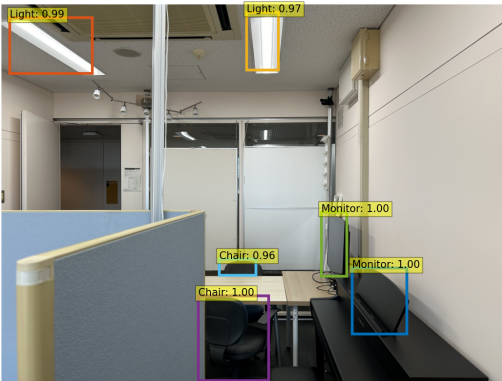
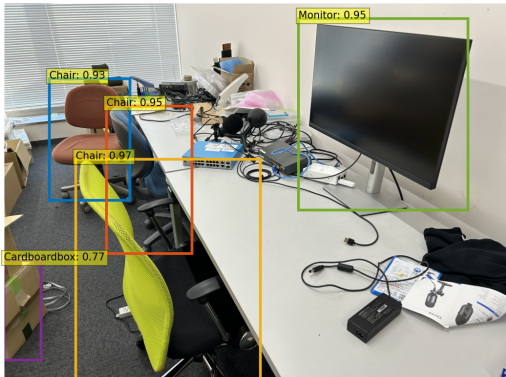
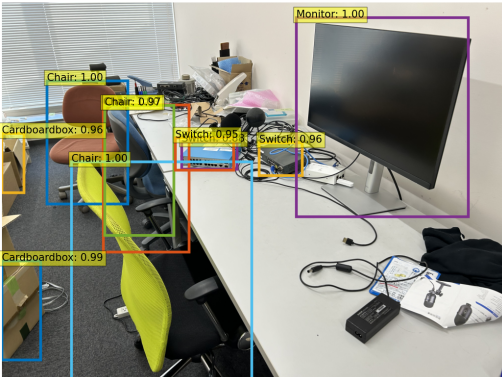


図 4.5. エッジ環境を想定して作成したデータセットによる WAFL-DETR-FPE, TLE, HE の mAP の推移

表 4.11. 学習初期段階のモデルと学習終了時のモデルにより推論されたバウンディングボックスの可視化と比較 (エッジ環境を想定して作成したデータセット)

At Epoch 100	Final Model
	
	
	
	

4.4 WAFL-YOLOのエッジ環境における性能評価

4.4.1 設定

モデルと学習設定

モデルはMS COCO データセット [22] で事前学習した重みをロードしたYOLOv9-cを使用する。クライアント数は6とし、各クライアントの位置は固定されているものとする。ネットワークのトポロジはlineとする。各クライアントは自分の持つデータセットで50エポックのpre-self trainingを行なった後、WAFLによって隣接するクライアントとのモデル交換を同期的に行いながら、200エポックの学習を行う。学習率は 10^{-3} とし、最適化関数にはSGDを使用する。アグリゲーションの係数 λ は1.0で設定する。

データセット

学習に使用するデータセットは東京大学工学部2号館10階の江崎・落合研究室の4つの居室において撮影した画像から構築したものを使用する。データセットの詳細は4.3で述べている。

4.4.2 評価結果

表4.12に各条件下での mAP_{50} の値と通信量の比較を示している。 mAP はクライアント数を6と仮定しているため、各デバイスの最終10エポックの平均 mAP について、デバイス数6でさらに平均をとったものを採用している。なおDHEは180エポック付近で mAP が低下し、他の手法もそこから mAP の向上が見られなかったため、180エポックで早期終了した場合の mAP を算出している。表中のs.d.はデバイスごとの mAP についての標準偏差を示している。また表中のtrafficは1つのデバイスが、1エポックに交換するパラメータのサイズをMByteで示している。

WAFL-YOLO-FPEが87.63の mAP を達成しており、他の手法よりも高い結果となった。一方でWAFL-YOLO-DHEの mAP は74.04となり、FPEとは10ポイント程差はあるものの、Centralizedのケースを上回る高精度を達成した。WAFL-YOLO-HEに関しては mAP の向上が見られなかった。またいずれのWAFLによる手法も、モデル単独での学習の場合と比較して高いスコアとなった。

各手法についての学習曲線を図4.6に示す。ただし、HEに関しては200エポックを通して mAP が10付近からほとんど向上しなかったため、省略した。FPEは30エポック付近ですでに mAP が収束している。一方でDHEでは180エポック付近まで mAP が漸増する結果になっている。またDHEでは180エポック付近で顕著な mAP 低下が見られた。

表 4.12. エッジ環境を想定して作成したデータセットによる WAFL-YOLO の mAP と通信量 (モデル交換・学習手法による比較)

		mAP ₅₀	s.d.	epochs	traffic
Decentralized (Non-IID)	WAFL-YOLO-FPE	87.63	0.17	180	194.62
	WAFL-YOLO-DHE	74.04	0.28	180	82.26
	WAFL-YOLO-HLE	10.00	1.20	180	0.04
	YOLO-FP	83.93	1.40	50	n/a
	YOLO-DH	57.47	5.64	50	n/a
	YOLO-HL	8.87	1.29	50	n/a
Centralized	YOLO-FP	89.64	n/a	90	n/a
	YOLO-DH	71.51	n/a	90	n/a
	YOLO-HL	10.57	n/a	90	n/a

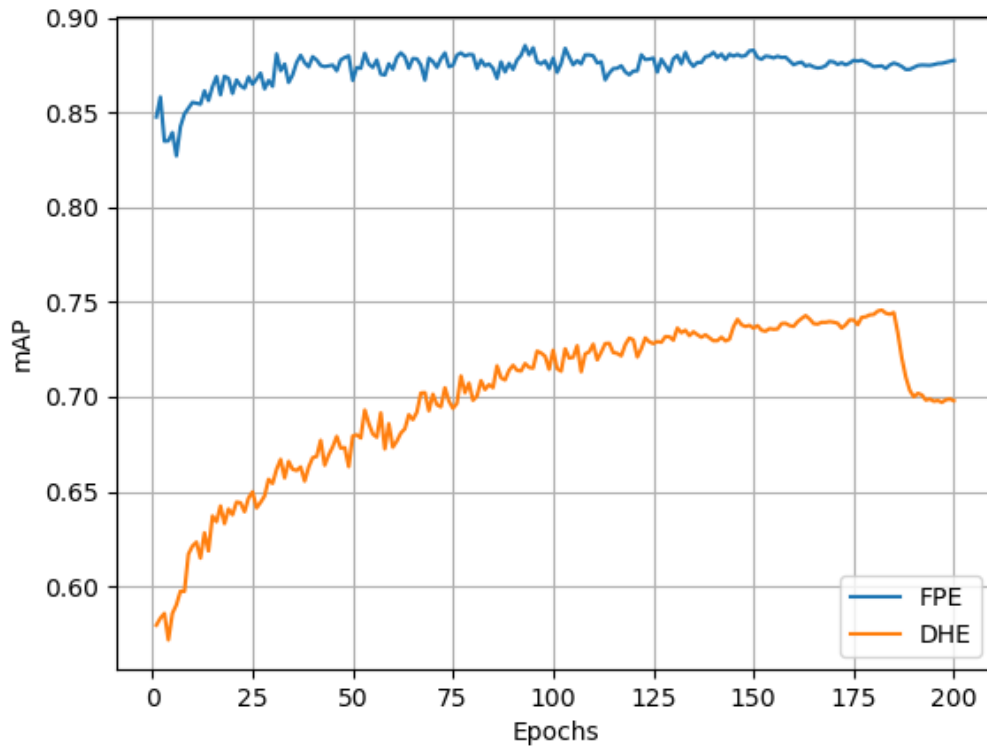


図 4.6. エッジ環境を想定して作成したデータセットによる WAFL-YOLO-FPE, DHE, HLE の mAP の推移

4.5 WAFL-SSOD の評価

4.5.1 設定

モデルと学習設定

モデルは MS COCO [22] データセットで事前学習した重みをロードした DETR を使用する。クライアント数は 6 とし、各クライアントの位置は固定されているものとする。ネットワークのトポロジーは line とする。各クライアントは自身の持つラベル付きデータで 100 エポックの pre-self training を行なった後、WAFL-DETR-TLE によって隣接するクライアントとのモデル交換を同期的に行いながら、600 エポックの学習を行う。その後、各クライアントごとにそこまで学習したモデルにより、自身の持つラベルなしデータから、擬似ラベルを生成する。擬似ラベル付きのデータを追加した新たなデータセットによって、再び WAFL-DETR-TLE によって隣接するクライアントとのモデル交換を同期的に行いながら、300 エポックの追加学習を行う。学習率は 10^{-5} 、Backbone の学習率は 10^{-6} とし、最適化関数には AdamW を使用する。アグリゲーションの係数 λ は 1.0 で設定する。擬似ラベル生成のための信頼度スコアの閾値 τ は 0.3 とする。各クライアントの持つラベル付きデータの割合は 10%, 20%, 50% でそれぞれ評価を行う。

データセット

学習に使用するデータセットは東京大学工学部 2 号館 10 階の江崎・落合研究室の 4 つの居室において撮影した画像から構築したものを使用する。データセットの詳細は 4.3 で述べている。

4.5.2 評価結果

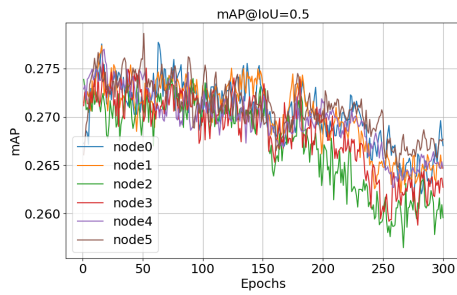
ラベル付きデータの各割合に対する通常の教師あり WAFL と WAFL-SSOD の mAP_{50} の比較を表 4.13 に示している。mAP はクライアント数を 6 と仮定しているため、各デバイスの最終 10 エポックの平均 mAP について、デバイス数 6 でさらに平均をとったものを採用している。なおラベル付きデータが 10% の場合と 20% の場合それぞれで、100 エポック、150 エポック付近から精度の低下が見られるため、それぞれ 100 エポック、150 エポックで早期終了とし、mAP を算出している。いずれのラベル付きデータの割合においても、WAFL-SSOD による擬似ラベルを加えた追加学習で mAP の向上が見られた。ラベル付きデータの割合が 20% および 50% の場合は通常の WAFL と比較して約 7% mAP が改善している。一方でラベル付きデータの割合が 10% の場合の mAP の改善は約 1% に留まっている。

また WAFL-SSOD の第二段階の擬似ラベルを含む協調学習における各デバイスの mAP の推移を図 4.7 に示している。ラベル付きデータの割合が 10%, 20% の場合ではそれぞれ 100 エポック、150 エポック付近を境にして、mAP が低下しているとともに、デバイスごとの mAP のばらつきが大きくなっている。一方でラベル付きデータの割合が 50% の場合は mAP の低

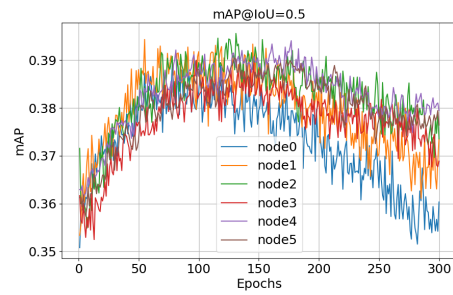
下が見られず、デバイスごとのばらつきも縮小している。

表 4.13. ラベル付きデータの各割合に対する WAFL-SSOD の mAP の比較

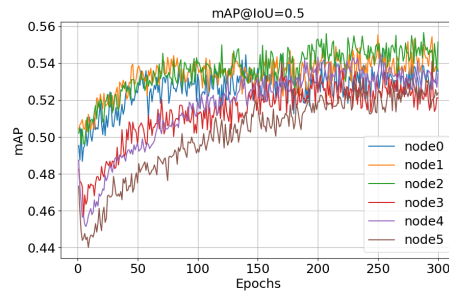
	10%	20%	50%
WAFL (supervised)	26.86 ± 0.18	36.09 ± 0.39	49.80 ± 0.53
WAFL-SSOD	27.12 ± 0.13	38.64 ± 0.38	53.27 ± 0.95



(a) mAP@10% labeled data



(b) mAP@20% labeled data



(c) mAP@50% labeled data

図 4.7. WAFL-SSOD の第二段階の擬似ラベルを含む協調学習における各デバイスの mAP の推移

第 5 章

考察

5.1 ベンチマークデータセットによる評価

ベンチマークデータセットによる WAFL-DETR と WAFL-YOLO のそれぞれの評価結果から, WAFL-DETR に関しては, そのバックボーンネットワークである ResNet の事前学習パラメータが下流タスクにも有効であり, WAFL の際にこの部分を交換・学習しなくても精度を維持できるということがわかった. 一方で WAFL-YOLO では FPE と DHE の間で, mAP に差が生じている. ただしこの mAP 低下は Centralized のケースでも見られており, WAFL による mAP の低下は見られない. そのため通信量を抑えてパラメータの一部のみのアグリゲーションを行うことによる精度低下の影響はないということがいえる. また WAFL-DETR については IID と Non-IID データセットの間で, mAP に差が生じており, Non-IID 特性による精度低下が顕著に見られる. 一方で WAFL-YOLO では学習の初期段階では Non-IID 特性による精度低下が見られるものの, 学習収束時には IID と Non-IID データセットの間で, ほとんど mAP に差が生じておらず, WAFL-YOLO の方がデータの異質性に対する耐性が強いといえる.

5.2 エッジ環境を再現したデータセットによる評価

実際のエッジ環境を想定したデータセットによる WAFL-DETR と WAFL-YOLO のそれぞれの評価結果から, そのような環境においても WAFL による物体検出の有効性が検証された. WAFL-DETR ではベンチマークデータセット (Non-IID) においてモデル単体で学習した場合と比較した時の精度改善が FPE の場合は約 1.66 倍, TLE の場合は約 1.75 倍であるのに対し, このデータセットではそれぞれ 1.81 倍, 1.93 倍であり, より大きな改善が見られる. WAFL-YOLO については同様の指標について, ベンチマークデータセットでは, FPE の場合は 1.40 倍, DHE の場合は 1.19 倍であるのに対し, このデータセットではそれぞれ 1.04 倍, 1.29 倍となっている. このことから少なくとも WAFL-DETR-TLE や WAFL-YOLO-DHE など一部のパラメータを交換・学習する場合に関しては, WAFL が本来利用を想定している, 閉じた環境においての性能が, オープンなデータセットよりも有効である可能性がある. 一方

で WAFL-DETR-HE でのベンチマークデータセットにおいて単体モデルからの精度改善が約 6.03 倍であるのに対し、このデータセットでは約 1.40 倍となっている。また WAFL-YOLO-HLE でのベンチマークデータセットにおいて単体モデルからの精度改善が約 2.53 倍であるのに対し、このデータセットでは約 1.13 倍になっている。そのため交換・学習するパラメータが極端に少ない場合には、むしろオープンなデータセットを使用した場合よりも、WAFL による性能の向上が制限される可能性がある。今回の評価では、ベンチマークデータセットに比較して、エッジ環境を想定したデータセットの方が、画像に含まれるオブジェクト数が多く、またデータセットの大半をマルチクラス画像が占めており、より多くのモデルの表現力が要求される点も原因かもしれない。

5.3 WAFL-SSOD

WAFL-SSOD の評価結果から、ラベル付きデータが少数の場合に WAFL-SSOD が mAP 向上に有効であることがいえる。一方でラベル付きデータの割合が低い場合は、追加学習による精度の改善が少ないこともわかった。これは第一段階の WAFL で使用するラベル付きデータが不十分であることにより、高い精度のモデルを獲得できず、それによって設定した閾値を超える質の高いアノテーションを生成できなかったためだと考えられる。図 4.7 において、ラベル付きデータが 10%、20% の時に見られる、学習途中からの mAP の低下とデバイスごとのばらつきの広がりも、質の低い擬似ラベルによる各デバイスの自身のデータセットへの過学習を示していると考えられる。

第 6 章

結論

6.1 まとめ

本研究では、Wireless Ad Hoc Federated Learning(WAFL) による物体検出を提案し、DETR と YOLO の二つのモデルにより、通信量も考慮した 3 つの手法によって、その性能を検証した。DETR に関しては、TLE によって精度を維持しつつ、通信量が抑えられることを確認した。加えて、実際のエッジ環境による学習を再現するために、データセットを構築し、それによる性能検証も行った。DETR と YOLO の両モデルで、実際のエッジ環境によるデータセットで、ベンチマークデータセットよりも高い性能を達成できることを確認した.. また半教師あり物体検出を WAFL に拡張した WAFL-SSOD を提案し、ラベル付きデータの割合ごとに性能比較を行った。いずれのラベル付きデータの割合でも、mAP の改善が見られたが、より大きな mAP 改善には、20% 以上のラベル付きデータがある方が望ましいことがわかった。

6.2 今後の課題

本研究では、エッジデバイス上での物体検出モデルの協調学習を再現するために、DETR や YOLO のモデルを使用した。が、現実のエッジデバイス上で実行するモデルとしては、これらは非常にサイズの大きいモデルになっている。一方でこれらのモデルにもバリエーションが存在し、ある程度は推論精度の低下を許容しつつ、よりサイズを抑えたコンパクトなモデルも登場している。このようなモデルで追加の性能検証を行うことは実用化に向けて必要だと考えられる。

また本研究では、エポックごとに全てのデバイスが同期的にモデルを交換し、協調学習を進めることを前提として評価を行った。一方で実応用を考えた際には、デバイスごとの異質性も考慮すべきであり、全てのデバイスが同期的にモデルの交換を行えるとは限らない。また同期的に行ったとしても、一番性能の低いデバイスがボトルネックになり、全体の学習が遅延するという事象が発生する。そのため、モデルの交換を非同期に行うような学習手法を取り入れて、それによる性能検証を行うことが今後の課題となる。

加えて本研究では、全てのデバイスが協調学習の全エポックに参加できていることを前提と

して, 評価を行っている. 一方で実際にはセキュリティ上の問題やデバイスの故障などによって, 一時的にデバイスが協調学習から離脱するといったシナリオが考えられる. 実応用を考えた際には, このようなケースでもシミュレーションを行う必要がある. またこうしたケースに対応できるように, 動的なトポロジーの導入も考慮する必要がある.

発表文献と研究活動

- (1) Ryuhei Yamaguchi, Hideya Ochiai, "Tuning Detection Transformer with Device-to-Device Communication for Mission-Oriented Object Detection", IEEE WiMob CWN workshop, 2024

参考文献

- [1] Lu Yu He Li and Wu He. The impact of gdpr on global technology development. *Journal of Global Information Technology Management*, Vol. 22, No. 1, pp. 1–6, 2019.
- [2] Raghubir Singh and Sukhpal Singh Gill. Edge ai: A survey. *Internet of Things and Cyber-Physical Systems*, Vol. 3, pp. 71–92, 2023.
- [3] Latif U. Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, Vol. 23, No. 3, pp. 1759–1799, 2021.
- [4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017.
- [5] Hideya Ochiai, Yuwei Sun, Qingzhe Jin, Nattanon Wongwiwatchai, and Hiroshi Esaki. Wireless ad hoc federated learning: A fully distributed cooperative machine learning, 2022.
- [6] 佐藤光哉. 無線設計の問題として見る分散連合機械学習. 電子情報通信学会 基礎・境界サイエティ Fundamentals Review, Vol. 16, No. 1, pp. 7–16, 2022.
- [7] H Ochiai, A Muramatsu, Y Ueda, R Yamaguchi, K Katoh, and H Esaki. Tuning vision transformer with device-to-device communication for targeted image recognition. In *IEEE World Forum on Internet of Things (WF-IoT)*, 2023.
- [8] Eisuke Tomiyama, Hiroshi Esaki, and Hideya Ochiai. Waf-gan: Wireless ad hoc federated learning for distributed generative adversarial networks. In *2023 15th International Conference on Knowledge and Smart Technology (KST)*, pp. 1–6. IEEE, 2023.
- [9] Hideya Ochiai, Riku Nishihata, Eisuke Tomiyama, Yuwei Sun, and Hiroshi Esaki. Detection of global anomalies on distributed iot edges with device-to-device communication. In *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Com-*

- puting, pp. 388–393, 2023.
- [10] Atsuya Muramatsu, Hideya Ochiai, and Hiroshi Esaki. Tuning personalized models by two-phase parameter decoupling with device-to-device communication. In *2024 16th International Conference on Knowledge and Smart Technology (KST)*, pp. 57–61. IEEE, 2024.
 - [11] Higuchi Ryusei, Esaki Hiroshi, and Ochiai Hideya. Personalized wireless ad hoc federated learning for label preference skew. In *IEEE World Forum on Internet of Things (WF-IoT)*, 2023.
 - [12] Yudai Ueda, Hideya Ochiai, and Hiroshi Esaki. Device-to-device collaborative learning for self-localization with previous model utilization. In *2024 16th International Conference on Knowledge and Smart Technology (KST)*, pp. 97–102. IEEE, 2024.
 - [13] Higuchi Ryusei, Esaki Hiroshi, and Ochiai Hideya. Collaborative multi-task learning across internet edges with device-to-device communications. In *IEEE Cybermatics Congress (SmartData)*. IEEE, 2023.
 - [14] Naoya Tezuka, Hideya Ochiai, Yuwei Sun, and Hiroshi Esaki. Resilience of wireless ad hoc federated learning against model poisoning attacks. In *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, pp. 168–177. IEEE, 2022.
 - [15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
 - [16] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.
 - [17] Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. Real-world image datasets for federated learning, 2021.
 - [18] Peihua Yu and Yunfeng Liu. Federated object detection: Optimizing object detection model with federated learning. In *Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, ICVISIP 2019, New York, NY, USA, 2020*. Association for Computing Machinery.
 - [19] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information, 2024.
 - [20] Priyanto Hidayatullah and Refdinal Tubagus. Yolov9 architecture explained.
 - [21] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection, 2020.
 - [22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich,*

Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755. Springer, 2014.

- [23] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Mallocci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, Vol. 128, No. 7, p. 1956–1981, March 2020.

謝辞

本論文を執筆するにあたり，研究の着想から論文の執筆まで多くの場面でご指導いただいた落合秀也准教授に深く感謝いたします。そして研究を進める上で助言をくださった江崎・落合研究室の同期・後輩を含む皆様に御礼申し上げます。また事務を通じて研究生活を支えていただいた高橋富美秘書，岩井愛映子秘書に感謝いたします。

付録 A

ソースコード

WAFL-DETR のソースコードは GitHub 上に公開している。
(<https://github.com/jo2lxq/wafl/tree/main/WAFL-DETR>)