東京大学大学院新領域創成科学研究科
社会文化環境学専攻

2017 年度
修　士　論　文

畳み込みニューラルネットワークを用いた
衛星画像における実用的高精度車両検出
Accurate Vehicle Detection using Convolutional Neural Network
in Satellite Images for Practical Applications

2018 年 1 月 22 日提出
指導教員　柴崎　亮介　教授

古 賀　　洋 平
Koga, Yohei

目次

第1章　序論

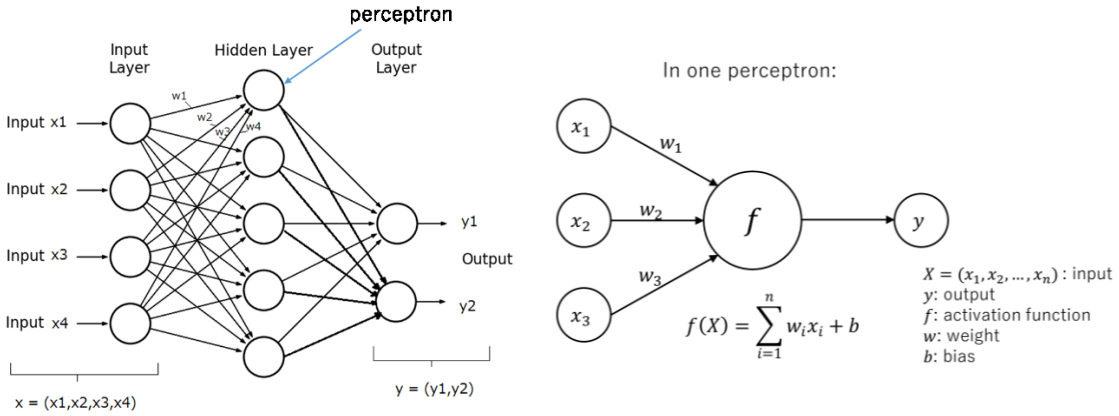
1.1　研究の背景と目的

近年、高解像度の衛星画像及び空撮画像が多く取得されるようになっている。Digital Globe [1] 社の WorldView-3 [2] が提供する衛星画像の解像度は 0.3m/pixel に達し、様々な応用への可能性を持っている。また、Planet Labs [3] 社や Black Sky [4] 社等のベンチャー企業では、解像度 1m/pixel 程度の小型衛星を多く打ち上げ、撮影頻度を高くして多くの画像を提供する試みを行っている。日本では、NTT 空間情報 [5] が日本全国の空撮画像を 1 年ごとにアップデートして提供する等、質の高い広範囲な画像が手に入りやすくなっている。このような背景から、衛星画像及び空撮画像を様々な実用的用途に用いる動きが広がっている。

従来、リモートセンシングにおいては、中像度衛星画像を用いて、NDVI 等の特徴量を用いて地表面の植生を分析するといった、比較的単純な応用が行われてきた。近年、より高解像度の画像が手に入るようになったことに加え、機械学習技術の進歩により、より複雑な特徴量を抽出し、様々な事象の理解のために用いるといったことが行われつつある。特に、深層学習（Deep Learning）と呼ばれる技術が大きく進歩し、機械学習の分野の発展に寄与している。深層学習は、画像識別や物体検出等の画像処理や、翻訳等の自然言語処理等を含めた様々な分野の精度を従来から大幅に改善した。最近では、深層学習は、私たちの身の回りの様々な分野に用いられ始めている。これはリモートセンシング分野でも同様であり、上に述べた高解像度画像を含めた様々なセンサーから得られた情報と合わせて、実用的な用途へと応用されつつある。

1.1.1　深層学習の進歩

まず、深層学習の概要と、その近年の進歩について簡潔に述べる。深層学習（Deep Learning）とは、機械学習技術の一種であり、ニューラルネットワークという技術のことを指す。ニューラルネットワークは、図 1 のように、脳の構造であるシナプスとニューロンを模した構造となっており、それぞれパーセプトロン、重み変数と呼ばれる。図 1 (a)中の左から、縦に並んでいるニューロンをそれぞれ入力層、隠れ層、出力層と呼ぶ。図 1 (b)のように、あるデータが入力されると、入力に重み変数の値が掛け合わされた後合算され、後方の層へと信号が伝達されていく仕組みである。ニューラルネットワークにおいて、（隠れ）層の数が多く（深く）なったものを深層学習と呼んでいる。



(a) ニューラルネットワークの構造。　　(b) パーセプトロンの構造。

図 1: 深層学習（ニューラルネットワーク）の概要。

(a)は http://www.astroml.org/book_figures/appendix/fig_neural_network.html から引用（一部改変）。

このニューラルネットワークに目的に応じた大量のデータを与えて、特徴等を学習させることにより、画像認識や自然言語処理等、様々なタスクをこなせるようになる。

深層学習は 2012 年、大規模な画像認識コンペティションである ILSVRC（ImageNet Large Scale Visual Recognition Challenge）[6] において、深層学習を用いた手法が従来の手法を大きく引き離して優勝したことを皮切りに、研究が大幅に進んだ。また、研究が進んだもう一つの理由として、GPU (Graphics Processing Unit; 画像処理に特化した演算装置) を用いた汎用計算技術の進歩により、深層学習に必要な膨大な計算が効率的に行えるようになったことがあげられる。近年では様々な機械学習の手法が高精度な深層学習の手法に置き換えられている。私たちの身の回りにも応用が進みつつあり、有名な例としては Facebook の顔認識や、Apple の Siri 等が挙げられる。リモートセンシング分野についても同様で、様々なセンサーで取得されたデータの深層学習手法による解析が試みられている。

1.1.2 衛星画像等解析による経済活動分析

ここで、衛星画像等を機械学習の手法により解析し、地表面の事象の理解、特に経済活動の推定等に役立てている例を紹介する。衛星画像は広範囲で取得可能なパブリックなデータであり、直接現地に赴いてデータを得ることが難しい、もしくは非常にコストがかかるようなケースにおいても利用可能である。例えば貧困度や都市の広がりなどといった状況を衛星画像から分析することにより、投資や政策といった意思決定等に役立てることができる。宮崎ら（2014）[7] は、中解像度の衛星画像と既存の GIS データを組み合わせることによって、市街地の広がりを精度よく判別した。Jean ら（2016）[8] は、深層学習を用い

て夜間光画像、日中の衛星画像及び統計調査のデータから学習を行い、日中の衛星画像から地域の貧困度を推定するシステムを開発した。これは、夜間光画像が地域の経済活動の活発さをよく表していることを利用し、夜間光画像の明るさと日中画像の特徴との関連を深層学習モデルの一種である畳み込みニューラルネットワーク(Convolutional Neural Network; CNN)に学習させることで、経済活動に関連する日中画像の特徴を学習させ、さらに学習した特徴と統計調査データの回帰分析モデルを作成することにより、精度よく貧困度推定を行ったものである。統計調査データ等の教師データは比較的少なくてすみ、広く取得できる衛星画像から精度よく地域の貧困度を推定できるため、非常に有用である。また、衛星画像とは異なる例だが、Gebru ら（2017）[9] は、膨大な Google Street View の画像から車両を検出し、それらの特徴から地域の人口統計学的特徴を推定した。まず、膨大な Google Street View 画像から従来的な画像処理手法で車両を検出し、それらの車種や製造年代といった細かい特徴を、CNN を用いて識別した。これらを地域ごとに集計したデータと、教師データとして統計調査データを用いて回帰モデルを構築することにより、精度よく収入や人種、教育水準や投票の傾向等といった地域の人口統計学的特徴を推定した。この手法も、従来ならば膨大な予算を使って調査を行っていたものが、比較的少量の教師データから精度よく調査ができるという意味で非常に有用である。このように、衛星画像等のパブリックに利用可能なデータからある特徴量を学習・抽出し、比較的少ない統計データ等と組み合わせて学習させ、それをスケールさせることでコストを抑えて地域の状況を精度よく推定する、といった手法が非常に成果を上げている。そして、それらの実現に深層学習が大きく寄与している。また、より具体的な経済活動量推定の例として、米ベンチャー企業である Orbital Insight [10] が挙げられる。彼らは、衛星画像と深層学習の手法を用いて、アメリカの小売チェーンの駐車場の車両を全国的に自動計測することにより、売り上げ指標の予測を行っているほか、衛星画像中のオイルタンクの屋根の高さを画像処理によって解析・推定することにより、中国等のオイル備蓄量の推定なども行っている。これらは民間企業による試みであるため詳細なアルゴリズムが明かされているわけではないが、衛星画像を用いた経済活動量の推定のよい例である。このように、広範囲で取得可能なパブリックなデータ、特に衛星画像を用いた経済活動の推定が非常に有望であることがわかる。

1.1.3 特徴量抽出手法としての車両検出

上で見てきたような経済活動量等の推定における、衛星画像からの特徴量の抽出手法として、大きく二つが考えられる。一つは、機械に特徴量そのものの設計をさせる方法である。[8] の手法は、経済活動量に対応する特徴量を CNN に学習させているため、自動で特徴量を抽出させる手法と考えられる。（ただし実際は、[8] の中でも触れられているように、たとえば収入によって家の素材が変わるなど、具体的なオブジェクトに対応する特徴を学習しているものと考えられる。）これは有効な手法であるが、一方で機械に解かせる問題が難

しくなり、また特徴を学習させるのにより大量のデータが必要となる。もう一つの手法は、画像からなんらかのオブジェクトを検出し、経済活動に関連する指標として利用する方法である。近年では、衛星画像の解像度も非常に高くなっていることに加え、深層学習の登場によりオブジェクト検出等の精度が大きく向上し、細かい解析も可能となっているため、よりミクロな分析を行う上ではこちらのほうが有利である。また、検出するオブジェクトとしては車両、建物、インフラ等様々なものが考えられるが、その中でも車両は経済活動とよく関連すると考えられる。例えば、商業施設が店舗出店を考えるとき、交通アクセスの良さは最も重要な点の一つと考えられる。また、上で述べたように、[9] は車の車種や年代、価格等の具体的な特徴量を基に推定を行っており、[10] の小売チェーンの売り上げ予測は、まさに売り上げに直接的な相関があると思われる車両数を計測することによって予測を行っている。また、こういった経済活動の予測のほかにも、交通量の推定等他にも様々な用途があり、高精度の車両検出手法は非常に重要性が高いと考えられる。

1.1.4 実用的精度達成への課題

近年の深層学習手法の進歩の恩恵を受け、車両検出手法もまたその精度を大きく向上させている。よって、高精度な車両検出の実現のためには、まずそれらの手法について検討を行う必要がある。ただし、実用的な精度達成という観点からは、もう一つ課題がある。それは、トレーニングに用いるデータである。一般的に、深層学習手法で高精度を達成するためには、大量の教師データが必要となる。幸い、いくつかの大規模な車両検出用公開データセットが存在する。しかし、実用途を考えたとき、利用可能なトレーニングデータの場所と、実際に車両検出を行いたい場所は異なる場合が多いと考えられる。そのような場合、たとえ深層学習が従来手法に比べ汎化能力にすぐれているとはいえ、性能は大幅に低下する。これに対して、追加でトレーニングデータを作成する、もしくは性能低下を防ぐなんらかの手法を適用するなどといったことが考えられるが、コストの観点も含め、どのようにすれば効率的に実用的な精度を達成できるか十分に検討を行う必要がある。

1.2 研究の目的

本研究では、経済活動量推定等の実用的用途に向けた、衛星画像における高精度な車両検出手法の検討を行う。まず、深層学習を用いた物体検出手法について概観し、その中で有望な手法を実際に衛星画像における車両検出に適用し、その性能を評価する。ここでは、トレーニングデータと同じ地域だけでなく、実用途を想定し、トレーニングデータと異なる地域（関心地域）での性能評価を行う。次に、関心地域で車両検出を行ったときの性能低下を回復させるための手段として、ドメインアダプテーションと呼ばれる手法を用いる。既存手法を基に、採用した車両検出手法に向けた手法を設計し、その性能の評価を行う。そして、さ

らなる精度向上のため、関心地域のトレーニングデータをいくらか得られたとき、どの程度精度を向上させることができるか検証する。最後に、実際の経済活動量推定の例として、得られた車両検出器を用いて、商業施設の駐車場の車両数推定を行う。

第2章　手法

2.1　深層学習による物体検出手法

本論文では、車両検出に深層学習の手法を用いる。それにあたり、深層学習による物体検出手法について述べる。

2.1.1　深層学習手法の概要

まず、深層学習の概要について述べる。1.1.1 で述べたように、深層学習はニューラルネットワークのことを指す。最も単純な構造は、ある層のパーセプトロンに対して、その前の層のパーセプトロンが全て結合している構造であり、これを全結合という。図 2 に深層学習の学習方法（教師有り）を示す。

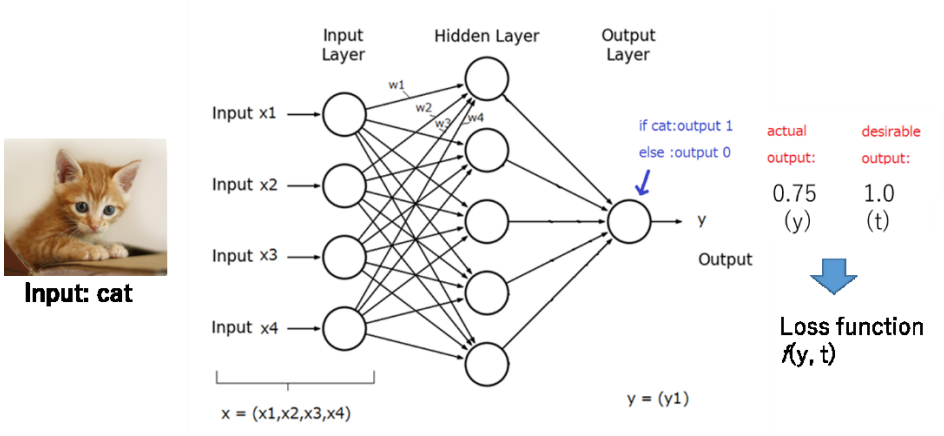


図 2: ニューラルネットワークの学習方法。
http://www.astroml.org/book_figures/appendix/fig_neural_network.html から引用（一部改変）。

図 2 において、入力として画像を与え、ネットワークは猫かそうではないかの確率を出力するものとする。実際の出力が得られた時、望ましい出力（正解ラベル）に対してどれくらい異なっているかを、誤差関数によって評価する。（今回の場合のような、確率でクラス分類を行うような場合、典型的にはソフトマックスクロスエントロピー関数が用いられる。）

そして、この誤差関数を小さくするような方向に、ネットワークの重み変数を更新する。これを多数の入力サンプルについて行うことにより、ネットワークが入力に対して正しい出力を行うよう学習する、という仕組みである。誤差関数を小さくするような重み変数の更新には、勾配法を用いる。すなわち、あるサンプルが与えられた時の誤差関数を重み変数で微分し、その傾きと逆方向に重みを更新する方法である。ニューラルネットワークは図 2 等に示されるように、計算グラフとみなすことができる。よって、各重み変数に関する誤差関数の偏微分係数は、出力層から入力層に向かって微分係数を連鎖律に従って伝搬することで計算する。これを誤差逆伝搬法という。また、深層学習では、全体のトレーニングデータを小さなデータにランダムに分割し（これをミニバッチという）、ミニバッチをひとつずつ処理していくことで学習を行う。これをミニバッチ学習という。一つのミニバッチを用いた一回のトレーニングをイテレーション(iteration)、全体のミニバッチを一通り処理し終わることをエポック(epoch)という。トレーニングは通常誤差関数が十分小さくなるまで複数エポック繰り返して行われる。

ところで、このシンプルな全結合のネットワークは、パーセプトロンの数を増やしたり、層が深くなると計算量が爆発するなどといった問題があった。そこで、畳み込みニューラルネットワーク（Convolutional Neural Network, CNN）という構造が提案された。図 3 に、CNN における計算の概要を示す。

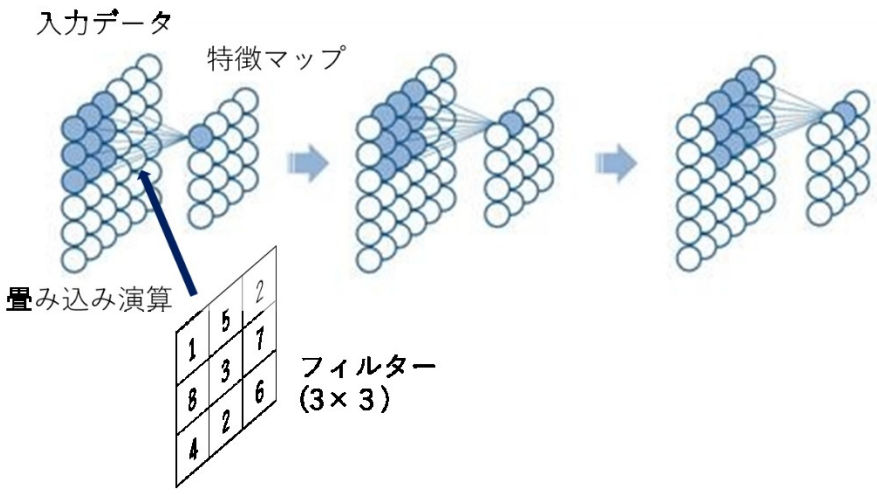


図 3: CNN における畳み込み演算の概要。
MathWorks(https://jp.mathworks.com/discovery/convolutional-neural-network.html)
から引用（一部改変）

CNN では、二次元(×チャンネル数)の入力上の局所領域に対しフィルターを用いて畳み込み演算を行う。これを入力の各ピクセル位置で行うことにより、特徴抽出を行い、特徴マップを出力する。この入力データの二次元上の各位置での畳み込み演算により、領域ベース

の特徴抽出が可能となり、CNN は画像認識等のタスクで高い性能を発揮することが可能となった。また、フィルターを入力上の各ピクセル位置において共有して用いることによって重み変数の量が削減されたほか、特徴マップを低解像度に圧縮する Pooling 処理により、計算量を削減した。1.1.1 で述べたように、この CNN ベースのネットワークが ILSVRC で従来手法を大きく引き離して優勝したことをきっかけに、深層学習手法の研究が加速し、大きく発展することとなった。現在では、CNN は画像認識にとどまらず、自然言語処理等様々な用途に応用されている。本論文で用いるような物体検出手法は、主にこの CNN を用いて実現されている。

2.1.2 物体検出手法

CNN を用いた物体検出手法について述べる。最も単純な物体検出手法は、画像中のピクセル位置を細かなステップでスライドさせていきながら、全ての局所領域を CNN で識別する、スライディングウィンドウと呼ばれる手法である。実装が簡単で、スライド幅を密にすることで高精度を達成することも可能であるが、冗長な計算が多く処理に時間がかかるという欠点がある。処理時間を改善するために、このスライディングウィンドウを、画像中の何らかの特徴量に基づいて物体が存在する可能性のある場所を事前に絞り込む手法で置き換えた、領域ベース検出器が登場した。Region-based CNN (R-CNN) [11] と呼ばれる手法は、Selective Search [12] と呼ばれる手法でまず画像中の物体が存在する可能性のある場所を計算し、それらを CNN で識別した。図 4 に手法の概要を示す。これにより、大幅に精度と処理時間を改善した。また、R-CNN の改善手法である Fast R-CNN [13] という手法も提案された。R-CNN では Selective Search によって得られた候補領域を全て個別に識別していたのに対し、Fast R-CNN ではまず画像全体の特徴マップを 1 回だけ計算し、各候補領域の識別は、RoI Pooling という機能で各候補領域を特徴マップ上に投影することで効率的に計算し、大幅に速度を向上した。

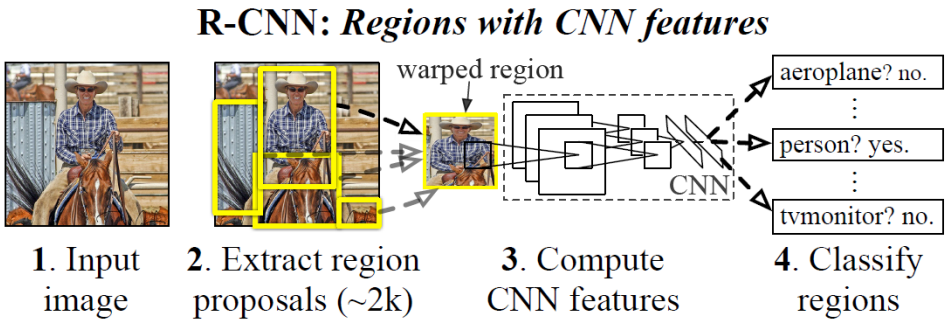


図 4: Region-based CNN (R-CNN)の概要。[11]から引用。

ただ、この Selective Search 自体がまだ比較的遅いという問題があったため、事前の物体候補領域の計算自体を Region Proposal Network (RPN)と呼ばれる深層学習の手法に置き換えて、全体の手法を全て深層学習で構成する、Faster R-CNN [14] と呼ばれる手法が提案された。Faster R-CNN の構造を図 5 に示す。Faster R-CNN により、精度と処理速度がさらに改善された。

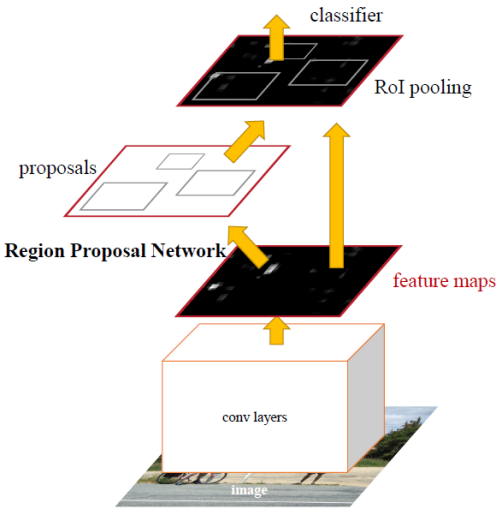


図 5: Faster R-CNN の概要。[14] から引用。Faster R-CNN は Fast R-CNN の Selective Search 部分を Region Proposal Network で置き換えた手法である。

ただ、Faster R-CNN は、まず RPN によって物体候補領域を計算し、それを Fast R-CNN で識別するという、物体候補領域の計算とそれらの識別が別のネットワークに分かれている構造である。You Look Only Once (YOLO) [15] や Single Shot MultiBox Detector (SSD) [16] はこれを一つのネットワークに統一し、物体候補領域とその識別を同時に行う手法である。これらの手法は、Faster R-CNN と同等もしくはそれ以上の性能を達成し、さらに処理速度を大幅に改善した。

2.2 車両検出手法

2.2.1 既往研究と採用手法の検討

近年の車両検出手法は、2.1.2 で紹介したような深層学習の物体検出技術を採用している。以下に既往研究を概観し、本論文での採用手法を検討する。まず、Chen ら (2014) [17] は、スライディングウィンドウによって車両検出を行った。Qu ら (2016) [18] は、スライディングウィンドウを Binarized Normed Gradients (BING) [19] という勾配画像の特徴を基

に物体候補領域を検出するアルゴリズムで置き換え、Chen らの手法と同等程度の精度を達成しつつ大幅に処理速度を向上した。Tang ら（2017）[20] は、Faster R-CNN [14] を車両検出に採用した。衛星画像中の車両は非常に小さなオブジェクトであり、Faster R-CNN のような領域ベース検出器は、物体候補領域の検出において車両のような小さなオブジェクトをとらえるのに難があったが、Tang らは RPN の物体候補領域検出において従来用いていた特徴マップに加え浅く解像度の高い特徴マップを組み合わせて用いることで、小さなオブジェクト検出の精度を向上した。また、衛星画像を小さなタイルに分割し、それぞれの検出結果を結合するという手法を導入した。さらに、Fast R-CNN で抽出した特徴の分類器に、弱分類器として決定木を使用した Real Adaboost [21] を採用した。これは一般に Hard Example Mining (HEM)と呼ばれるアルゴリズムの一種で、分類器が識別しにくいトレーニングサンプルを優先的に選んでトレーニングに用いることで、精度を向上するアルゴリズムである。これらにより、Tang らは高い検出精度を達成した。また [22] でも、類似の領域ベース検出器である YOLO [15] を用いて、高精度な車両検出を実現している。一方、Mundhenk ら [23] は、シンプルなスライディングウィンドウに深いネットワークを組み合わせるという手法を用いた。彼らは強力な分類器として GoogleNet [24] 及び ResNet [25] をベースとした深いネットワークを採用し、スライディングウィンドウのスライド幅を非常に小さくすることで密な走査を行い、非常に高い検出精度を達成した。またこれは、彼らが同時に提案して用いた大規模車両検出データセットである Cars Overhead with Context (COWC)による所も大きいと思われる。

採用手法の検討の過程で、筆者はまず Chen らの手法 [17] を簡素化したものを実装したが、(筆者自身の実装の問題がいくらかあったにせよ) 確かに検出に時間がかかった。また、その実験ではアメリカ地質調査所（United States Geological Survey, USGS）[26] の EarthExplorer [27] よりニューヨークの空撮画像を数枚ダウンロードし、車両を手動でアノテートして独自のトレーニング及びテストデータとして用いたが、トレーニングデータが少なすぎたせいで思ったように精度が出ず、誤検出が非常に多くなってしまった。Online Hard Example Mining [28] と呼ばれる手法を基に独自に設計した HEM の手法 [29] を適用していくらか改善することができたものの、十分ではなく、やはり高精度を達成するためには豊富なトレーニングデータが不可欠であることがわかった。

上で見てきたように、近年は領域ベース検出器が車両検出に多く採用され、高い精度を達成している。単純に精度だけ見れば、Mundhenk ら [23] が使用した手法も有望であるが、検出の処理時間は長くなるものと考えられる。実用的用途には処理時間も重要であると考え、本論文では領域ベース検出器を採用することとした。具体的には、SSD [16] を採用した。これは、ChainerCV [30] というライブラリを用いて容易に実装ができることと、主な領域ベース検出器である Faster R-CNN, YOLO, SSD の中では SSD が最も高い性能を報告しているからである。

2.2.2 Single Shot MultiBox Detector と車両検出向けのチューニング

本論文では、車両検出に SSD [16] を採用した。以下、SSD のアルゴリズムの概要と、車両検出向けに行ったチューニングについて述べる。

図 6 に、SSD のネットワークの概要を示す。SSD はオクスフォード大学の Visual Geometry Group が開発した VGG-16 [31] と呼ばれるネットワークをベースとしている。

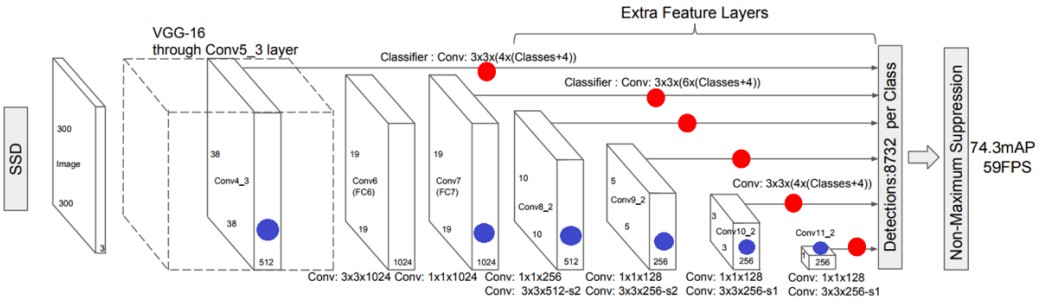
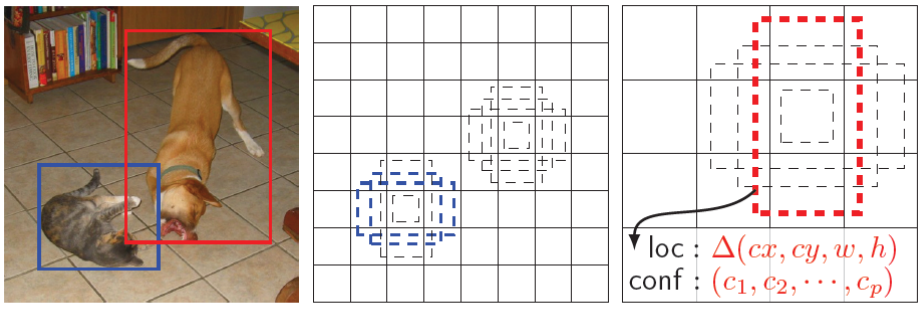


図 6: SSD のネットワーク。[16] から引用（一部改変）。青い丸印の特徴マップが物体検出に用いられる。赤い丸印は物体検出のため特徴マップ上を走査する小さな分類器である。

図 6 において、青い丸印の特徴マップが物体検出に用いられる。それぞれの特徴マップの大きさは異なっているが、左側の大きな特徴マップがより小さな物体検出に用いられ、右側の小さな特徴マップがより大きな物体検出に用いられる。これにより、SSD は異なるサイズのオブジェクトを効率的に検出できるようになっている。詳細を図 7 に示す。



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

図 7: SSD における物体検出のアルゴリズム。[16] より引用。

図 7 (b)と(c)の特徴マップの大きさは異なっているが、それぞれ入力画像全体に対応している。ここで、それぞれの特徴マップに対して、Default Box Size というパラメータが設定されている。これは、それぞれの特徴マップにおいて、検出されるべき物体の大きさのベース

を決定するパラメータである。Default Box Size は基本の正方形に加え、いくつかの縦横比の異なるバリエーションを持つ。SSD は、各特徴マップの全てのピクセル上において、この Default Box の大きさにあったオブジェクトが存在するかを検索し、Default Box からの位置及び縦横比のズレを補正する値と、物体の種類の予測値を出力することで、物体検出を行う。トレーニングにおいては、各特徴マップ上で Default Box と入力画像の Ground Truth との対応を取り、IoU（Intersection over Union；２つの領域の重なっている面積を、２つを合わせた領域の面積で割った値で、２つの領域の重なり具合を示す指標）が一定の閾値を超えた場合、正例とし、閾値以下の場合負例とする。物体検出においては、小さな分類器（図 6 における赤い丸印）が各特徴マップ上を走査し、3×3 の局所領域を畳み込み演算することで、Default Box からの位置及び縦横比のズレを補正する値と、物体の種類の確率を出力する。

本論文では、画像の入力サイズは 300×300pixel とした。また上で述べたように、SSD においては Default Box Size が検出する物体の大きさのベースを決定している。後に述べるが、本論文で使用する画像の解像度は 0.3m/pixel であり、画像上の車両の大きさはだいたい 20～25pixel と小さく、常にほぼ同じ大きさである。よって Default Box Size は、特徴マップの並びの順に 24, 30, 90,150, 210, 270 ,330 を設定した。一番大きな 38×38 の特徴マップの Default Box Size が 24 であり、車両の大きさに最適化された値となっている。他の特徴マップについても Default Box Size の設定を行っているが、実質的に車両検出に寄与しているのは 38×38 の特徴マップのみと考えられる。

2.3 ドメインアダプテーション手法

上記手法は高精度を達成できるものと期待される。しかし、1.1.4 で述べたように、実用を考えたとき、トレーニングデータの場所と車両検出を行いたい関心地域は異なることが多いと考えられる。そのような場合、検出精度は大幅に低下する。これを回復させる手段として、本論文ではドメインアダプテーションと呼ばれる手法を適用する。

2.3.1 既往研究とベース手法の検討

まず、転移学習(Transfer Learning)について説明を行う。転移学習とは、あるタスクで学習した知識を、もう一つのタスクに転用することをいう。例えば、機械学習において画像認識のためにトレーニングしたモデルを、音声認識のために転用する、といったことが例として挙げられる。転移学習は広い概念であり、知識を転用するにあたっては画像認識と音声認識のようにタスクが異なっても構わない。転移学習のうち、タスクは同じであるが、事象の分布が異なる場合をドメインアダプテーション（Domain Adaptation, 以下 DA とする）という。例えば、物体認識における RGB 画像と距離(depth)画像という例においては、物体認

識というタスクは同じであるが、RGB 画像と距離画像では見え方が大きく異なる。この場合、RGB 画像と距離画像は異なるドメインということになる。一般的な DA の問題は、一方のドメインにはラベル付きトレーニングデータが存在し、もう一方にはない（もしくは少量しか得られない）場合に、トレーニングデータのないドメインでのタスクの精度を向上させることである。この場合、ラベルがあるほうをソースドメイン、ラベルがないほうをターゲットドメインと呼ぶ。本論文で問題となる車両検出を行う地域の違いは、DA の問題にあたる。

DA の一般的な目的は、教師データを用いずに、ソースドメインとターゲットドメインの共通の特徴空間を見出し、両方の特徴を近づけることである。典型的には、ソースドメインのデータとターゲットドメインのデータの統計的な距離を算出し、その距離を最小化するようにデータをエンコードする。この距離の算出方法としてよく使われるのは、Maximum Mean Discrepancy (MMD) [32] である。MMD は、２つの異なる分布のデータがあったとき、それぞれのデータの平均を求め、平均同士の距離を求める。このとき、表現力向上のために正定値カーネルで張られる再生核ヒルベルト空間に写して計算を行う。カーネルトリックにより、高次元に写したデータそのものを陽に求めることなく、計算することができる。他の手法としては、Correlation Alignment (CORAL) [33] という、まず各分布のデータにおいてそれぞれの分散共分散行列を求め、それらの距離を求めるというシンプルな手法がある。Transfer Component Analysis (TCA) [34] は、MMD の距離を小さくするような変換行列を求め、データを変換してソース及びターゲットドメインの特徴を近づける手法である。Matasci ら（2015）[35] が、TCA のリモートセンシング分野への適用について検討を行っている。DA は有望な手法であるが、車両検出については今まで検討が行われていない。

近年、深層学習における DA 手法が活発に研究されている。上に述べたような距離ベースの DA は、距離を求める項を誤差関数に導入することにより、深層学習に拡張されている。MMD をベースとした手法には Deep Domain Confusion (DDC) [36] や Deep Adaptation Network (DAN) [37]、CORAL をベースとした手法には Deep CORAL [38] がある。一方で、最新の有望な手法として、敵対的学習とよばれるものがある。この手法においては、まずドメイン識別器を導入し、データがソース及びターゲットのどちらのドメインから来たものかを判別するようトレーニングする。一方で、特徴抽出器は、ドメイン識別器に識別されにくいような特徴を抽出するようトレーニングする。この相反する目的に基づいたトレーニングを行うことにより、特徴抽出器がソース及びターゲットに共通するような特徴を学習するというものである。[39] は、特徴抽出器のトレーニングにおいて、ドメイン識別器の識別結果が一様分布となるような誤差関数を導入した。Domain-Adversarial Neural Networks (DANN) [40] は、ドメイン識別器から伝搬される勾配を反転させる勾配逆転層を導入し、ドメイン識別器に対し識別されにくい特徴の学習を行った。Adversarial Discriminative Domain Adaptation [41] は、特徴抽出器の学習時にドメイン識別器のラベ

ルを反転させる GAN-loss を導入した。これは、Generative Adversarial Network (GAN) [42] に基づいた手法で、勾配が安定し学習が進みやすくなる。

本論文では、2.2.2 で採用した SSD に対して、2 つの DA 手法を設計し、性能の評価を行った。一つは、Deep CORAL [38]に基づいた距離ベースの手法、もう一つは、Adversarial Discriminative Domain Adaptation [41] に基づいた敵対的学習の手法である。これらを選んだ理由は、共にシンプルな手法であり、類似手法と比べたときより高い性能を報告していたからである。

2.3.2 CORAL 及び Adversarial Domain Adaptation

本論文では、Deep CORAL [38] に基づいた CORAL DA 及び Adversarial Discriminative Domain Adaptation [41] に基づいた Adversarial DA の性能評価を行う。

まず、両手法における共通事項について説明する。今回車両検出手法として採用した SSD について 2.2.2 で述べたところによれば、1 つのオブジェクトを検出するための特徴の単位は、特徴マップ上の 3×3 の局所領域である。よって、これらの局所領域を独立したサンプルとみなし、DA 手法を適用する。また、2.2.2 で述べたように、車両検出に実質的に寄与しているのは 38×38 の特徴マップのみである。よって本論文では、この特徴マップのみに DA を適用した。もしサイズの異なるオブジェクト検出を目的とする場合は、全ての特徴マップに DA を適用すべきと考えられる。

CORAL DA は Deep CORAL [38] に基づく。CORAL ロス（誤差関数）は以下のように定義される：

$$L_{CORAL} = \frac{1}{4d^2}\|C_S - C_F\|_F^2$$

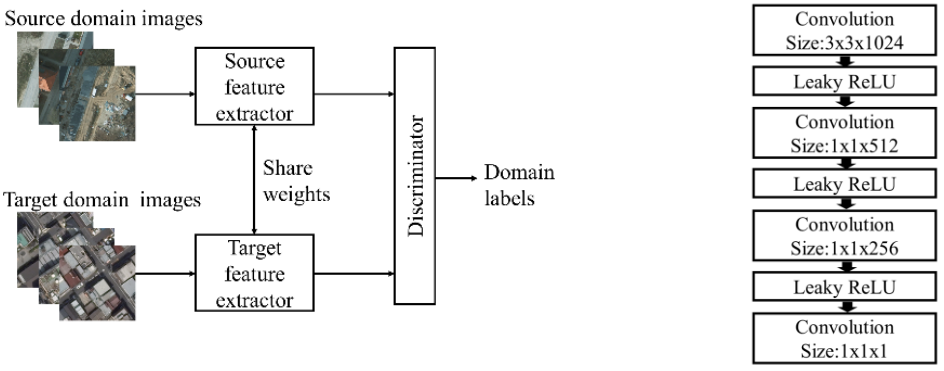
ここで、$\|\cdot\|_F^2$は 2 乗フロベニウスノルム、dはデータの次元、C_S 及び C_Fはそれぞれソース及びターゲットドメインの分散共分散行列である。各分散共分散行列は、1 イテレーション中のサンプルから計算される。公式より、各ドメインの分散共分散行列は以下のように計算できる：

$$C_S = \frac{1}{n_S - 1}\left(D_S^\top D_S - \frac{1}{n_S}(\mathbf{1}^\top D_S)^\top(\mathbf{1}^\top D_S)\right)$$

$$C_T = \frac{1}{n_T - 1}\left(D_T^\top D_T - \frac{1}{n_T}(\mathbf{1}^\top D_T)^\top(\mathbf{1}^\top D_T)\right)$$

n_S, n_Tはソース及びターゲットドメインのサンプル数、D_S, D_Tはソース及びターゲットドメインのサンプルを並べた行列である。$\mathbf{1}$ は要素が 1 の縦ベクトルである。

Adversarial DA は Adversarial Discriminative Domain Adaptation [41] に基づく。図 8 (a)は Adversarial DA のアルゴリズムの概要、図 8 (b)はドメイン識別器のネットワークを示す。



(a) (b)

図 8: Adversarial DA の概要。(a) Adversarial DA のアルゴリズム。(b) ドメイン識別器のネットワーク。Convolution:畳み込みレイヤ、Leaky ReLU: 活性化関数 Rectified Linear Unit の一種。

ドメイン識別器のネットワークにおいて、畳み込みレイヤのスライド幅、パディング幅は共に 1 である。図 8 (a)において、ドメイン識別器(discriminator)は、ソース及びターゲット特徴抽出器からの出力が、どちらのドメインから来たものかを判別するようトレーニングされる。一方で、ターゲット特徴抽出器(target feature extractor)は、ドメイン識別器に識別されないような特徴を出力し、ドメイン識別器を“騙す”ようにトレーニングされる。これら 2 つのトレーニングは交互に繰り返される。ドメイン識別器及びターゲット特徴抽出器の誤差関数は以下である：

$$L_{DIS} = -\mathbb{E}_{\mathrm{x}_s \sim \mathrm{X}_s}\left[\log D\left(M_s(\mathrm{x}_s)\right)\right] - \mathbb{E}_{\mathrm{x}_t \sim \mathrm{X}_t}\left[\log\left(1 - D\left(M_t(\mathrm{x}_t)\right)\right)\right]$$

$$L_{EXT} = -\mathbb{E}_{\mathrm{x}_t \sim \mathrm{X}_t}\left[\log D\left(M_t(\mathrm{x}_t)\right)\right]$$

X_s, X_tはそれぞれソース及びターゲットドメインからのサンプル、Dはドメイン識別器、M は特徴抽出器を表す。ドメイン識別器は、特徴抽出器の出力である特徴マップを入力とする。ドメイン識別器の誤差関数は、ドメイン識別器の出力におけるピクセル位置ごとのソフトマックスクロスエントロピーの値を合算して得られる。ソース及びターゲット特徴抽出器は、車両検出用データセットで事前訓練された SSD ネットワークの特徴抽出器で初期

化される。また、本手法においてはソース及びターゲット特徴抽出器は重みを共有しているため、実質的には２つは同一のネットワークである。トレーニングによってソース及びターゲットの特徴空間が近づけられた後は、ターゲット特徴抽出器によって抽出された特徴が、SSD の分類器に入力され、車両検出が行われる。

両 DA 手法において、まず SSD を車両検出用データセットで事前訓練した後、DA 手法で追加トレーニングする手法を採用した。このほうが、最初から DA を用いてトレーニングするよりも性能が良かったためである。また、DA の訓練中、同時に車両検出用のトレーニングも継続して行った。これは、DA の訓練のみ行った場合に、モデルが車両検出タスクのために学習した知識を失ってしまうのを防ぐためである。これにより、最終的な CORAL DA の誤差関数は以下となる：

$$L_{CORAL\_DA} = L_{SSD} + \alpha L_{CORAL}$$

αは重み係数である。また、最終的な Adversarial DA の誤差関数は以下となる：

$$L_{ADV\_DA1} = L_{DIS}$$
$$L_{ADV\_DA2} = L_{SSD} + L_{EXT}$$

Adversarial DA においては、この二つの誤差関数を用いた学習が交互に行われる。

第3章　実験

3.1　使用するデータ

3.1.1　ソースドメイン

車両検出用データセット、すなわちソースドメインのデータには、現時点で最も大きな車両検出用公開データセットである、COWC データセット [23] を採用した。COWC データセットは 32,716 台の車両を含む。ここでいう車両とは乗用車やバンなどといった比較的小型の車両であり、トラックなどの大型車両はラベルを付与されていない。解像度は 0.15m/pixel である。COWC データセットは 6 地域の画像から構成される。ただし、パンクロマティック画像と RGB 画像の両方を含み、RGB 画像の地域はカナダのトロント、ニュージーランドのセルウィン、ドイツのポツダム、アメリカのユタの 4 地域である。本論文では、RGB 画像のみ実験に用いた。図 9 に、各地域の画像の例を示す。

(a) トロントの画像。

(b)セルウィンの画像。

(c) ポツダムの画像。

(d) ユタの画像。

図 9: COWC データセットの各地域における画像の例。

本論文におけるターゲットは衛星画像であるので、まず解像度を 0.3m/pixel にダウンサンプリングした。そして、画像を 300×300pixel の格子状に分割し、４つごとのタイルをテストデータセット、残りをトレーニングデータセットとした。車両が含まれていないタイルはデータから除外した。また、全てのタイルを 90°, 180°, 270° 回転させることによって複製を行った。これは Data Augmentation といい、データに加工を行って人工的にデータの量を増やし、精度向上を図る手法である。一連の処理により、以下のデータを得た。

- 6,264 枚のソースドメイントレーニング画像（ラベル付き、300×300pixel）
- 2,088 毎のソースドメインテスト画像（ラベル付き、300×300pixel）

3.1.2　ターゲットドメイン

実際に車両検出を行いたい関心地域を想定したデータセット、すなわちターゲットドメインのデータセットには、NTT 空間情報 [5] から提供を受けた日本の空撮画像を用いた。解像度は 0.16m/pixel である。NTT 空間情報は日本全国の空撮画像を定期的にアップデートして提供している。今回使用するのは関東圏の画像で、東京の墨田区及び江東区のほとんどのエリアをカバーしている。図 10 に画像の地域及び画像の例を示す。(画像中にうすく社名のロゴが印刷されているが、実験への影響は無視できるレベルである。)

(a) 空撮画像のエリア。　　(b) 空撮画像の例。

図 10: NTT 空間情報の空撮画像。(a) 空撮画像のエリア。赤い長方形が画像の存在するエリアを示し、青い長方形が DA のトレーニングデータ用に使用したエリアを示す。(b) 空撮画像の例。

図 10 (a) において、赤い長方形が画像の存在するエリアを示す。図 10 (b) は画像の例で、非常に密集していることがわかる。実際、この画像の特徴は 3.1.1 で示したソースドメインのデータの特徴と異なっており、これが車両検出における性能低下を引き起こすと考えられる。

まず、ソースドメインのデータと同じく、解像度を 0.3m/pixel にダウンサンプリングした。そして、図 10 (a)の青い長方形のエリアをトレーニングデータとし、ソースドメインのデータと同様に、画像を 300×300 の格子状に分割した。Data augmentation も同様に適用した。これにより、1,408 枚のターゲットドメイントレーニング画像（ラベル無し）を得た。そして、赤い長方形から青い長方形を除いた残りのエリアから、1000×1000 の画像タイルをランダムに 44 枚抽出し、車両にラベルを付与した。対象としたのは、COWC データセットと同じく、乗用車やバンなどの比較的小型の車両である。(20 枚の画

像にラベルを付けるのに、おおよそ 1 日を要した。） そのうち 4 枚を、DA のトレーニングにおける性能検証（バリデーション）用、20 枚をテスト用とし、残り 20 枚は最終的な精度向上を図るためのトレーニング画像とした。

一連の処理により、以下のデータを得た。

- ターゲットドメイントレーニング画像 1,408 枚（ラベル無し、300×300pixel）
- ターゲットドメインバリデーション画像 4 枚（ラベル有り、1000×1000pixel）
- ターゲットドメインテスト画像 20 枚（ラベル有り、1000×1000pixel）
- ターゲットドメイントレーニング画像 20 枚（ラベル有り、1000×1000pixel）

最後のラベル有りトレーニング画像は、3.6 において、他のトレーニング画像と同様に 300×300pixel に加工して用いる。ここで得られた全てのターゲットドメインの画像は一切の重複を持たない。

3.2 テスト方法及び結果の評価基準

車両検出の方法、及び結果の評価基準について述べる。車両検出の方法は、採用した手法である SSD に準じる。一点変更した箇所は、元論文における物体の検出基準は、SSD が予測した矩形領域と実際の物体の矩形領域の IoU が 0.5 以上であったが、衛星画像中の車両は非常に小さいため、基準を少し低い 0.4 以上に変更した。また、画像が SSD の入力サイズである 300×300 より大きい場合、すなわち 3.1.2 のターゲットドメインバリデーション及びテスト画像（ラベル無し）を用いたテストの場合は、図 11 で示すように 50pixel 重なるように画像を 300×300 のタイルに分割し、それぞれの車両検出結果を集計する。集計時には、non-maximum-suppression (検出結果が重なり合っている場合、確率の高いもの以外を除外する処理。重なり具合の閾値として、元論文と同じ IoU0.45 を用いた。）を行う。

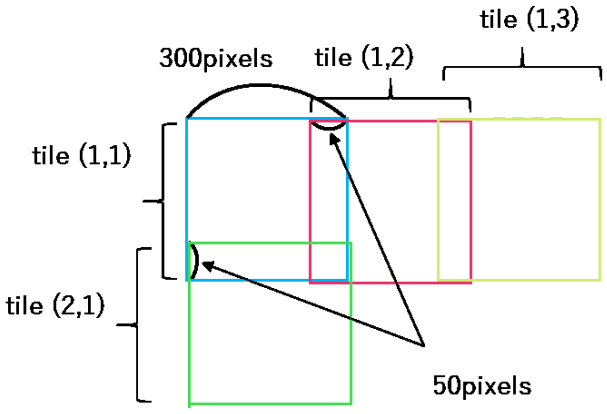


図 11: サイズの大きい画像の車両検出における、タイル分割処理。

車両検出結果の定量的評価基準として、以下の指標を用いる。

$$\mathrm{PR}\left(\text{Precision Rate; 適合率}\right) = \frac{\text{正しく検出された車両数}}{\text{SSD が検出した車両数}}$$

$$\mathrm{RR}\left(\text{Recall Rate; 再現率}\right) = \frac{\text{正しく検出された車両数}}{\text{画像中の全車両数}}$$

$$\mathrm{FAR}\left(\text{False Alarm Rate; 誤検出率}\right) = \frac{\text{誤検出数}}{\text{画像中の全車両数}}$$

$$\mathrm{F1}\left(\text{F1 measure; F1 値}\right) = \frac{2 \times \mathrm{PR} \times \mathrm{RR}}{\mathrm{PR} + \mathrm{RR}}$$

$$\mathrm{AP}\left(\text{Average Pecision; 平均適合率}\right)$$

AP は、物体検出における性能評価の基準として広く用いられている指標で、検出結果を確率の高い順に並べたとき、ランキングの上位にどれくらい正しい結果が含まれているかの指標である。すなわち、検出結果を上位から見ていき、新しく車両が検出された時点での PR の値を合計・平均した値である。

主要な性能指標としては、AP と F1 を用いる。

3.3 ソースドメインのみ用いた車両検出器の性能

3.3.1 トレーニング

まず、ソースドメインのデータのみ用いて、SSD のモデルを車両検出器としてトレーニングした。データは 3.1.1 で説明した、ソースドメイントレーニング画像（ラベル無し）である。トレーニングの設定は、いくつかのパラメータを除いて元論文 [16] と同じである。変更した点は、トレーニングの長さ（イテレーション数）を 40,000 とし、weight decay（学習率を決められたスケジュールに沿って低減させること）のスケジュールを 28,000 及び 35,000 イテレーションにした。バッチサイズは同じ 32 である。

図 12 に、誤差関数の値を記録したトレーニングカーブを示す。図が示すとおり、十分に収束している。

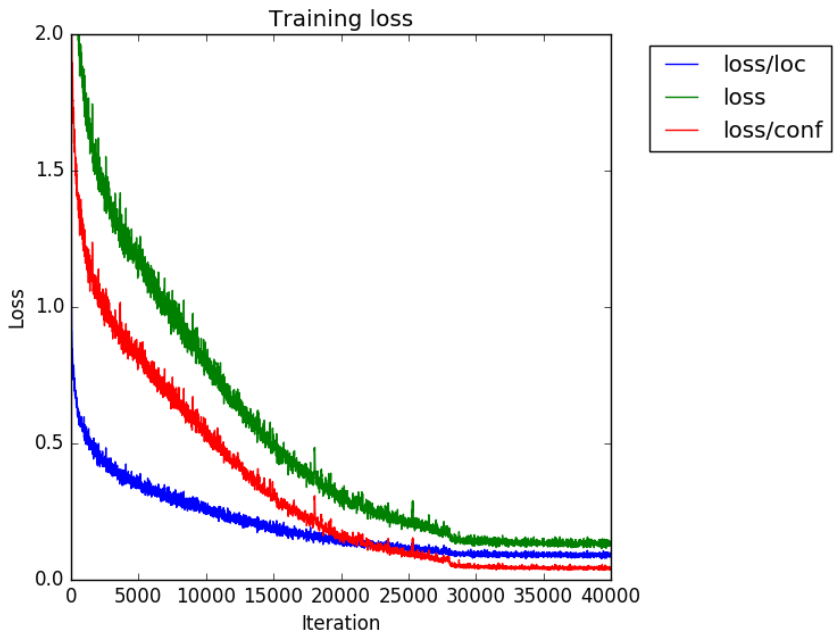


図 12: SSD のトレーニングカーブ。loss/loc は位置誤差、loss/conf は分類誤差、loss は 2 つの合計値。

3.3.2 車両検出テスト

表 1 にソースドメインテスト画像を用いた車両検出結果、表 2 にターゲットドメインテスト画像を用いた車両検出結果を示す。

表 1: ソースドメインテスト画像を用いた車両検出結果。

AP	F1	Mean of AP and F1	PR	RR	FAR
80.6%	**87.6%**	**84.1%**	90.9%	84.5%	8.5%

表 2: ターゲットドメインテスト画像を用いた車両検出結果。

AP	F1	Mean of AP and F1	PR	RR	FAR
60.2%	**72.4%**	**66.3%**	84.1%	63.5%	12.0%

表 1 において、AP は 80.6%、F1 は 87.6%に達した。使用しているデータが異なるため公正な比較とはならないが、これは、[20] と比べても同等以上の性能である。以降、これを参考値として用いる。しかし、表 2 においては、AP が 60.2、F1 が 72.4%まで低下した。他の指標について細かく見ると、特に RR の低下が著しい。すなわち、画像中の全車両のうち、検出された割合が低くなっている。以下、この性能低下を回復させるため、DA を適用した。

3.4 ドメインアダプテーションの適用

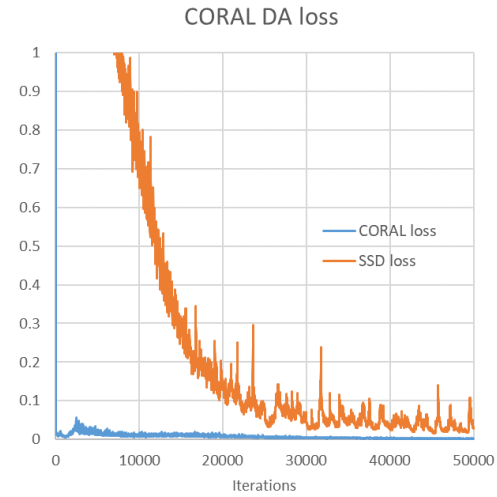
3.4.1 トレーニング

3.3.1 でトレーニングした SSD モデルを、DA 手法を用いて追加トレーニングした。2.3.2 で説明したように、DA のトレーニングと同時に、3.3.1 のトレーニングを継続して行ったが、バッチサイズは同じく 32 である。トレーニング中、ターゲットドメインバリデーション画像を用いて、10 イテレーションごとにモデルの車両検出性能評価を行った。理由は、Adversarial DA で採用したような敵対的学習においては、誤差関数をチェックするだけではトレーニングの進み具合を確認することが難しいためである。よって、ターゲットドメインバリデーション画像において車両検出の性能評価を行い、よい結果を残したモデルをスナップショットとして保存した。また、その評価の基準としては AP と F1 の平均を用いた。これは、AP は高くても F1 が非常に低いといった場合が起こりうるためで、AP のみを評価の基準とするのは不適当であったためである。

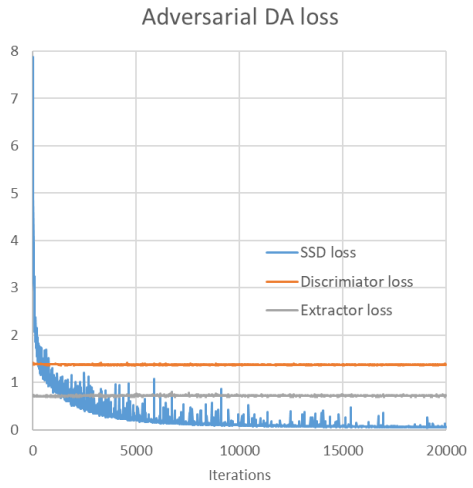
CORAL DA においては、1 イテレーション中、ソース及びターゲットドメインのそれぞれ 16 枚ずつの画像を用いて、L_{CORAL}を計算した。これは、トレーニングにおける 1 サンプルの次元数が 4,608 と大きく、分散共分散行列が非常に大きくなったため、使用した GPU のメモリの都合上 32 枚の画像を使うことが難しかったためである。トレーニングの長さは 50,000 イテレーションとした。L_{SSD} と L_{CORAL}が近いオーダーの値となるように、重み係数 αには 1e8 という大きな値を設定した。オプティマイザには Adam を使用し、パラメータα, β1, β2 はそれぞれ 0.001, 0.9, 0.999 を使用した。

Adversarial DA では、1 イテレーション中、L_{DIS} 及び L_{EXT}の計算に、ソース及びターゲットドメインのそれぞれ 32 枚ずつの画像を使用した。ドメイン識別器のトレーニングにおいては、本手法ではバッファー [43] を導入した。バッファーは、過去のトレーニング中に特徴抽出器によって抽出された特徴マップを一定枚数保存する。1 イテレーション中、ソース及びターゲットドメインについてそれぞれ 16 枚ずつのサンプルがバッファーからランダムに取り出され、ドメイン識別器のトレーニングに使用される。そして、新しく抽出された両ドメインそれぞれ 16 枚ずつの特徴マップが、バッファー内のサンプルをランダムに置き換える。この機構の意図するところは、過去に用いたトレーニングデータ

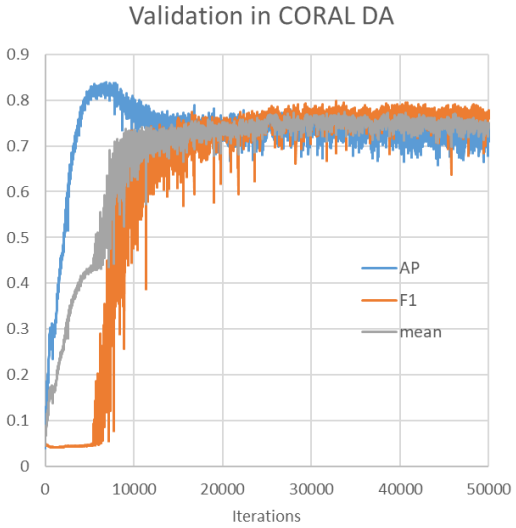
を一定数保持して現在のドメイン識別器のトレーニングに用いることで、ドメイン識別器が現在の特徴抽出器の出力に過敏に反応することを防ぎ、トレーニングを安定させることである。バッファーサイズには両ドメインそれぞれに 128 を使用した。これにより、トレーニングが若干安定した。図 13 に、各トレーニングカーブを示す。また、図 14 に、各手法のターゲットドメインバリデーション画像を用いた性能評価における、AP 及び F1 の平均値の比較を示す。



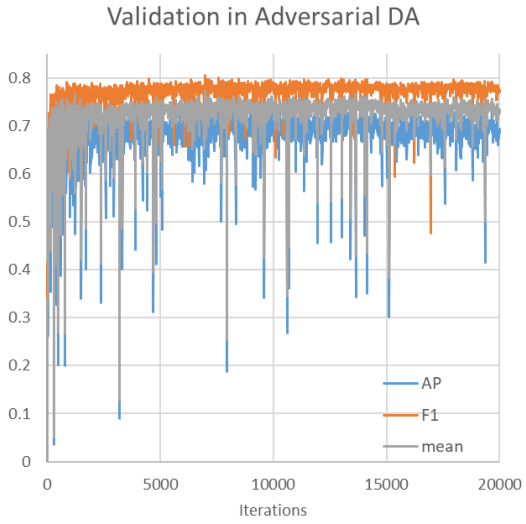
(a) CORAL DA の誤差関数。



(b) Adversarial DA の誤差関数。



(c) CORAL DA の、ターゲットドメインバリデーション画像における性能評価。



(d) Adversarial DA の、ターゲットドメインバリデーション画像における性能評価。

図 13: CORAL 及び Adversarial DA のトレーニングカーブ。

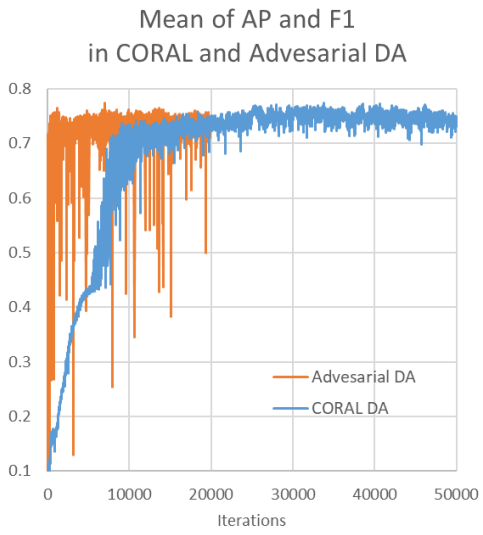


図 14: CORAL 及び Adversarial DA のターゲットドメインバリデーション画像を用いた性能評価における、AP 及び F1 の平均値の比較。

図 13 (a)において、CORAL ロス(αL_{CORAL})は、最初の値が 518 と大きかったのを除けば、非常に小さな値となった。最終的なαL_{CORAL}は、SSD ロス（L_{SSD}）に比べ、1/15 程度となった。図 13 (b)において、ドメイン識別器の誤差関数（L_{DIS}）及び特徴抽出器の誤差関数(L_{EXT})は共にほぼ一定の値となった。図 13 (d)で Adversarial DA の性能が比較的早く収束したのに対し、図 13 (c)の CORAL DA ではより多くのイテレーション数がかかっている。また、図 14 では、CORAL 及び Adversarial DA はほぼ同じ値に収束している。図 14 において最も良いスコアを記録したのは、CORAL DA が 33,750 イテレーションのとき 77.4%、Adversarial DA が 7,000 イテレーションのとき同じく 77.4%である。これらのモデルをスナップショットとして保存し、以降の車両検出テストに用いた。

3.4.2 車両検出テスト

3.4.1 で保存したスナップショットを用いて、ターゲットドメインテスト画像上で車両検出テストを行った。表 3 及び表 4 に、それぞれ CORAL 及び Adversarial DA による車両検出結果を示す。

表 3: CORAL DA によるターゲットドメインテスト画像を用いた車両検出結果。

AP	F1	Mean of AP and F1	PR	RR	FAR
72.9%	**80.7%**	**76.8%**	86.2%	75.8%	12.1%

表 4: Adversarial DA によるターゲットドメインテスト画像を用いた車両検出結果。

AP	F1	Mean of AP and F1	PR	RR	FAR
72.5%	**79.3%**	**75.9%**	84.0%	75.2%	14.3%

両手法において、大幅に性能を改善することができた。以降、各手法による車両検出結果の評価を行う。

3.5 性能評価

3.5.1 定量的評価

各手法における車両検出結果について、定量的に評価を行う。まず、全ての手法における結果を表 5 にまとめる。

表 5: 全ての手法における車両検出結果。

	AP	F1	Mean of AP and F1	PR	RR	FAR
Reference	80.6%	87.6%	84.1%	90.9%	84.5%	8.5%
Without adaptation	60.2%	72.4%	66.3%	84.1%	63.5%	12.0%
CORAL	72.9%	80.7%	76.8%	86.2%	75.8%	12.1%
Adversarial	72.5%	79.3%	75.9%	84.0%	75.2%	14.3%

図 15 に、全ての手法における AP 及び F1 の平均値をグラフで示す。DA によって、性能低下幅の半分以上を回復できたことがわかる。本実験においては、若干ながら CORAL DA のほうが Adversarial DA よりも良い結果となった。

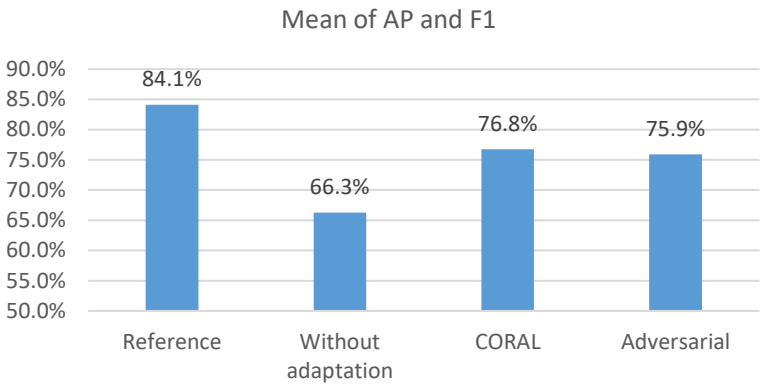


図 15: 全ての手法における、AP 及び F1 の平均値。

次に、各手法における AP, F1, PR, RR, FAR の値を折れ線グラフで図 16 に示す。

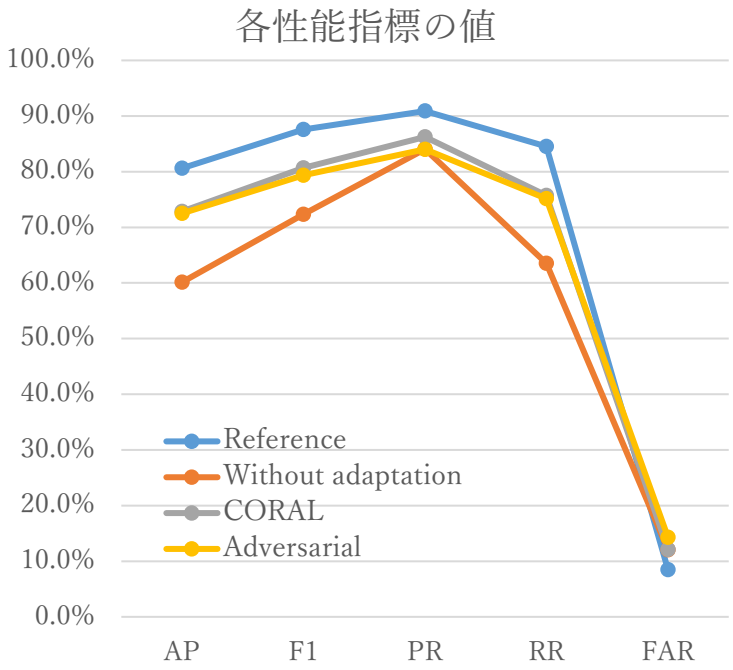


図 16: 全ての手法における、各性能指標の値。

Adversarial DA において若干 PR 及び FAR が DA 無しの場合より悪化しているが、CORAL 及び Adversarial DA でほぼ同様の傾向がみられ、ほぼ全ての指標で数値を改善している。DA 無しの場合、3.3.2 でも述べたように RR が大幅に低下していたが、表 5 及び図 16 からわかるように、RR が大幅に改善している。これが、性能改善の大きな要因となっていると考えられる。

3.5.2　車両検出結果画像の比較

ここでは、ターゲットドメインテスト画像で行った車両検出結果について、具体的な画像を見ながら比較する。まず、20 枚の個別の画像について、どの程度 F1 が改善したかを確認した。ここで F1 を使用しているのは、分類の確度によらず検出したかどうかのみを問題としたためである。結果、ほぼ全ての画像について F1 が改善していたが、1 枚だけ結果があまり変わらず CORAL DA の F1 が若干下がったものがあった。よって、F1 がおおよそ 10%以上大幅に改善した画像 1～3 の 3 枚と、あまり改善しなかった画像 4 の 1 枚を例として取り上げる。なお、画像の名前は便宜的なものである。それぞれの画像について、正解ラベル、DA 無しの結果、CORAL DA の結果、Adversarial DA の結果を示す。図中、緑の矩形は正解の車両ラベル、赤は正しい検出、青は誤検出を表す。

図 17～20 及び表 6～8 は画像 1 の結果、図 21～24 及び表 9～11 は画像 2 の結果、図 25～28 及び表 12～14 は画像 3 の結果、図 29～32 及び表 15～17 は画像 4 の結果を示す。

画像 1 及び画像 3 は、ソースドメインのトレーニング画像とは異なる密集した地域であり、使用した日本の画像に特有の特徴となっている。DA 無しだと RR が低くなっているが、CORAL 及び Adversarial DA の両手法で RR が大きく改善している。画像 2 は密集した地域ではないが、RR の改善に加え、誤検出がなくなることで大幅な検出結果の改善となっている。一方画像 4 は、比較的密集していない地域で、ソースドメインのトレーニング画像にもしばしばみられるような特徴である。CORAL DA では全ての指標がほぼ変わらず、若干結果が悪くなっている。Adversarial DA では RR が改善しているが、同時に誤検出が多くなっているため、改善は小幅となっている。

また、CORAL DA と Adversarial DA の結果を比較したときに興味深い事象として、Adversarial DA の検出結果の矩形の大きさは正解ラベルとほぼ同じになっているのに対し、CORAL DA の検出結果の矩形は明らかに小さくなっていることがある。詳しい原因は不明だが、一つの可能性として、CORAL ロスは各ドメインの特徴マップ間の分散共分散行列の距離を小さくする制約を課すが、これが特徴マップの絶対値そのものをある程度小さくするような働きをした可能性がある。検証の方法としては、特徴マップの実際の値や、モデルの重み変数の値を確認することなどがある。これは今後の課題である。

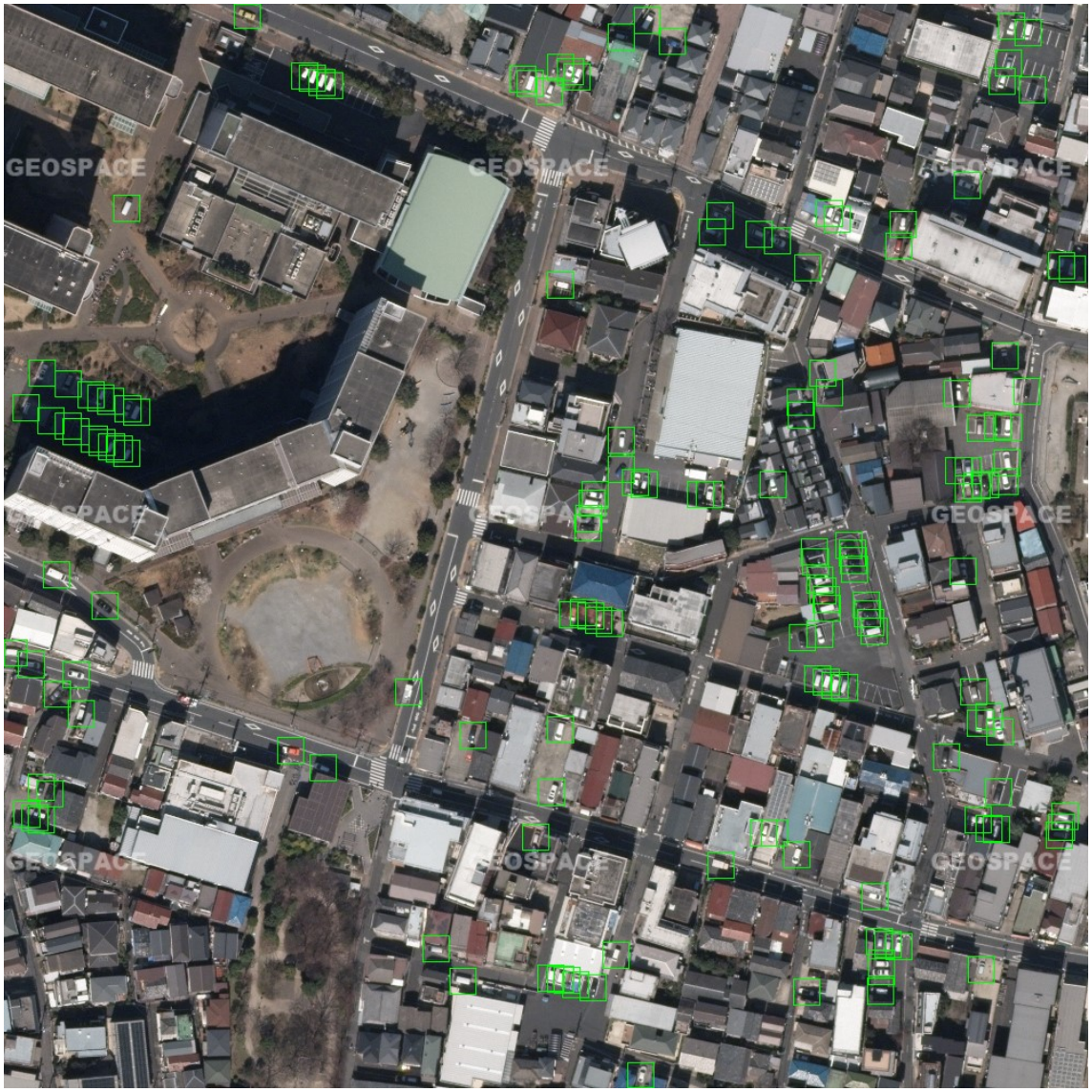


図 17: 画像 1 の正解ラベル。

図 18: 画像 1 における、DA 無しの検出結果。赤：正解、青：誤検出

表 6: 画像 1 における、DA 無しの検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
49.3%	63.9%	56.6%	77.3%	54.5%	16.0%

図 19: 画像 1 における、CORAL DA の検出結果。赤：正解、青：誤検出

表 7: 画像 1 における、CORAL DA の検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
70.4%	79.3%	74.9%	85.8%	73.7%	12.2%

図 20: 画像 1 における、Adversarial DA の検出結果。赤：正解、青：誤検出

表 8: 画像 1 における、Adversarial DA の検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
65.0%	74.1%	69.6%	79.0%	69.9%	18.6%

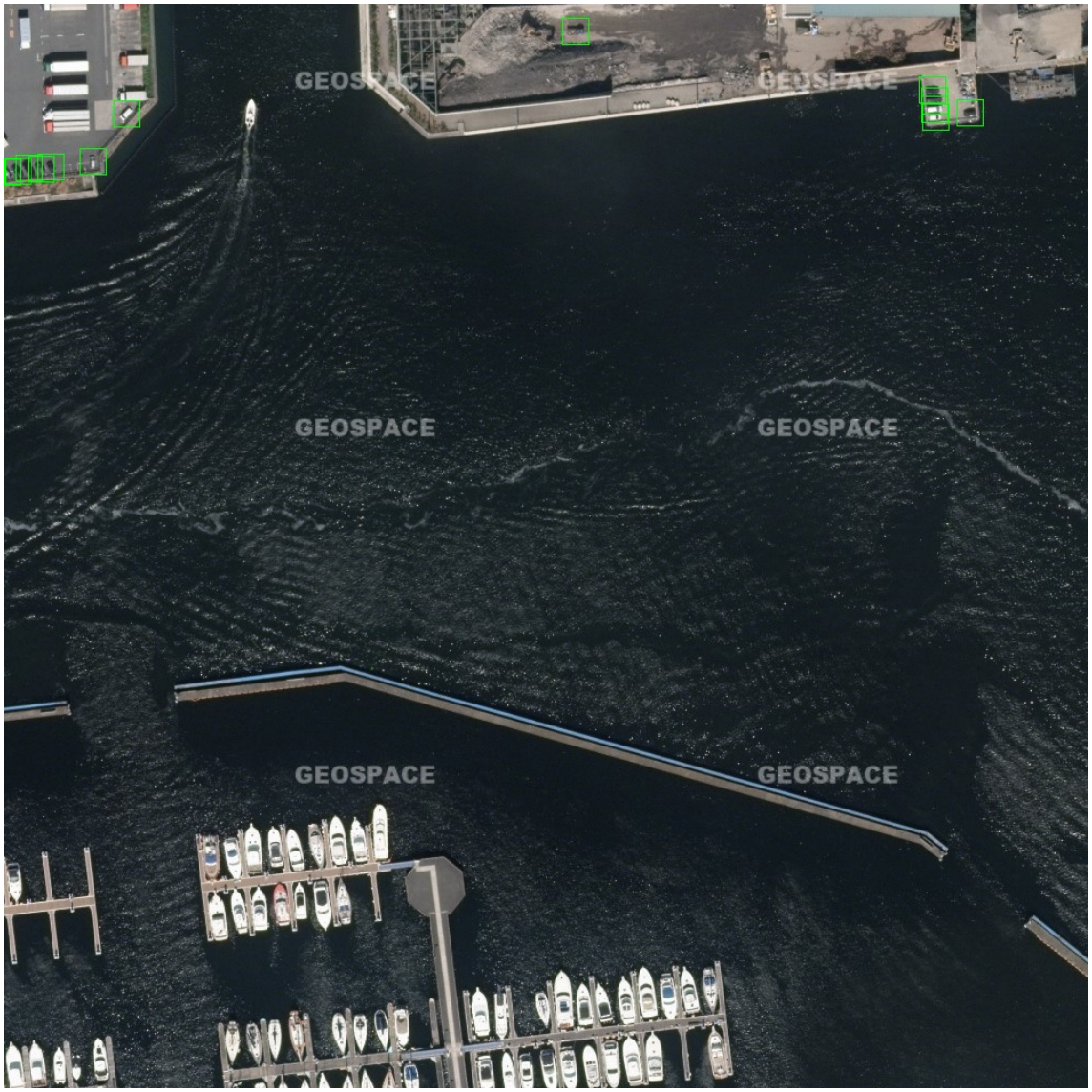


図 21: 画像 2 の正解ラベル。

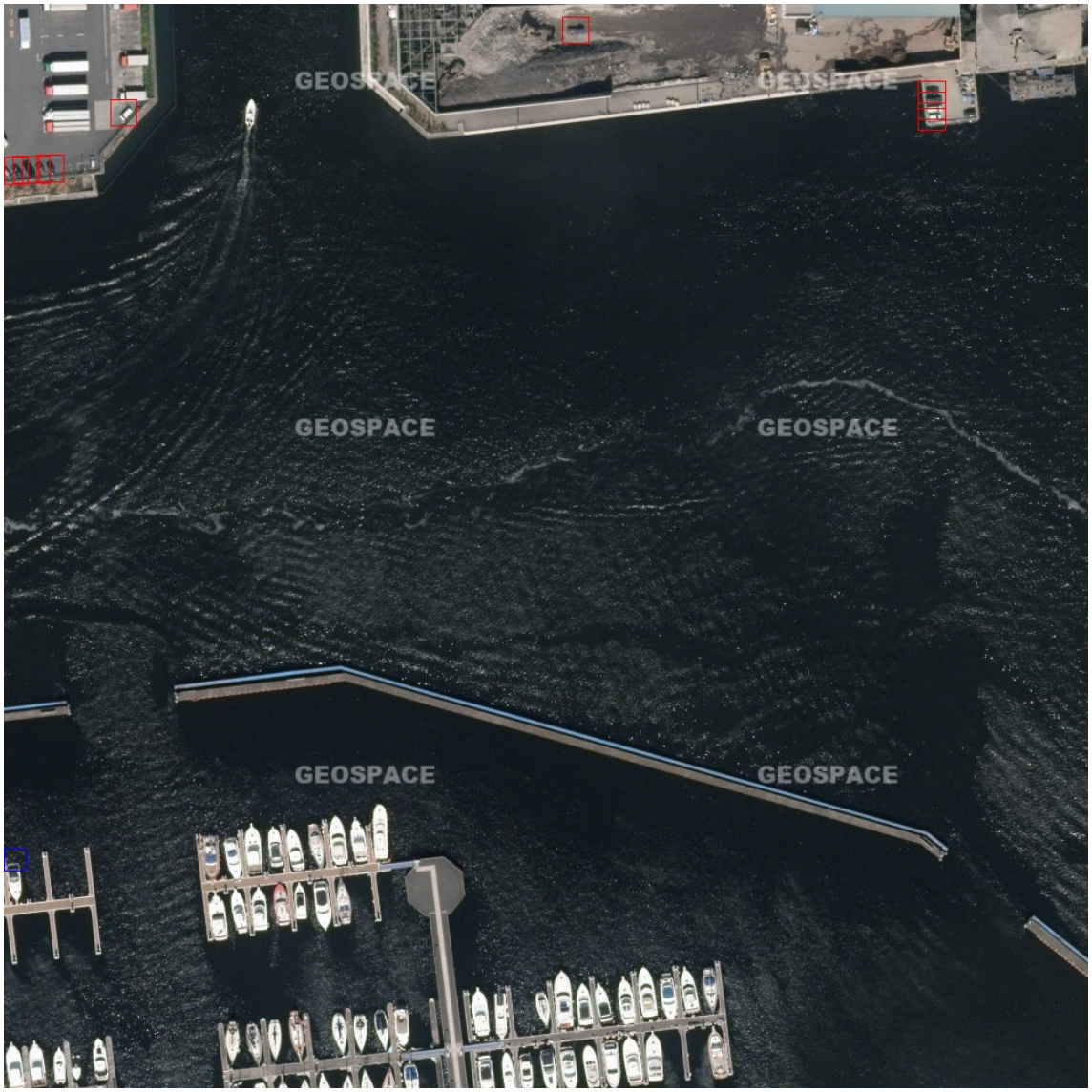


図 22: 画像 2 における、DA 無しの検出結果。赤：正解、青：誤検出

表 9: 画像 2 における、DA 無しの検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
69.2%	78.3%	73.7%	90.0%	69.2%	7.7%

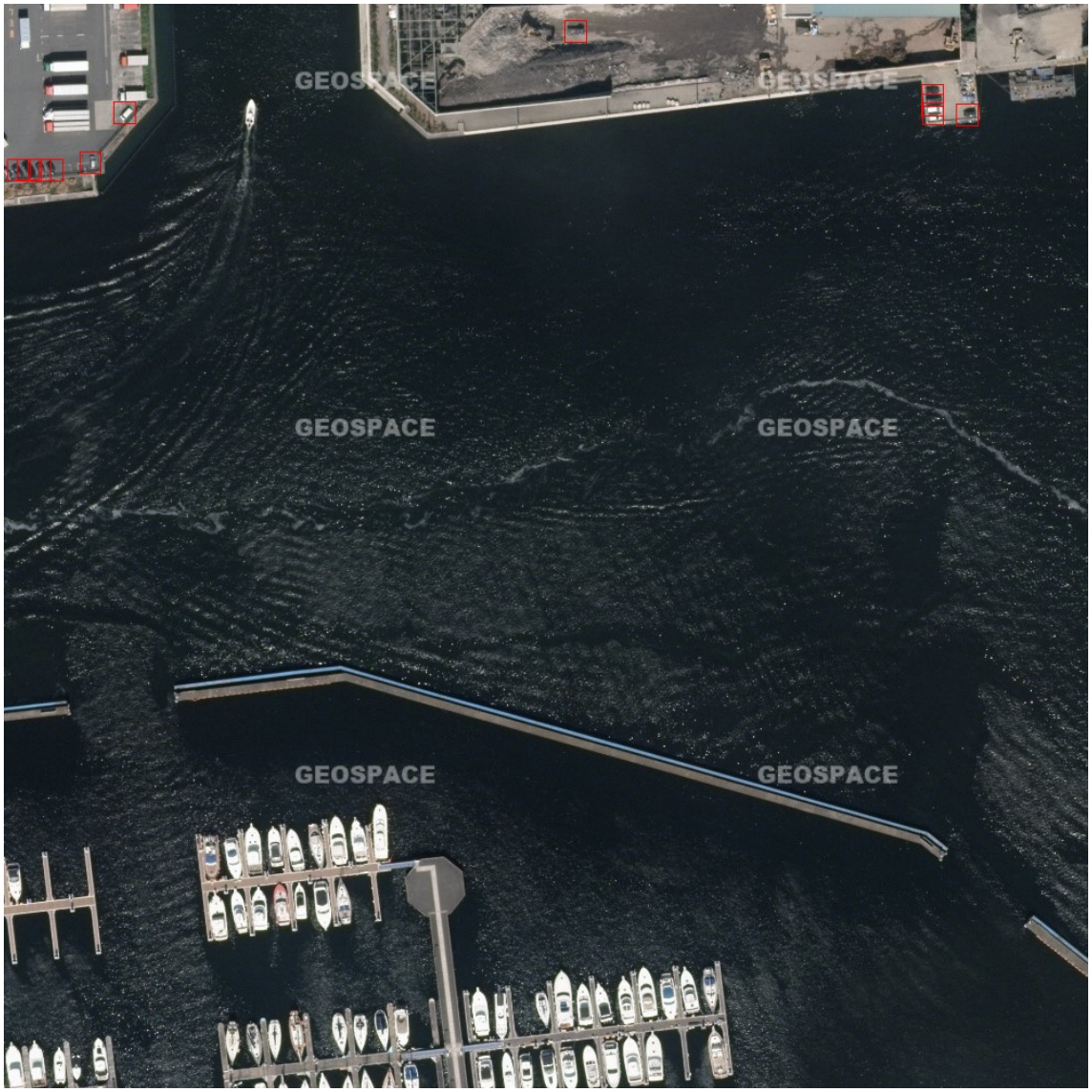


図 23: 画像 2 における、CORAL DA の検出結果。赤：正解、青：誤検出

表 10: 画像 2 における、CORAL DA の検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
84.6%	91.7%	88.1%	100.0%	84.6%	0.0%

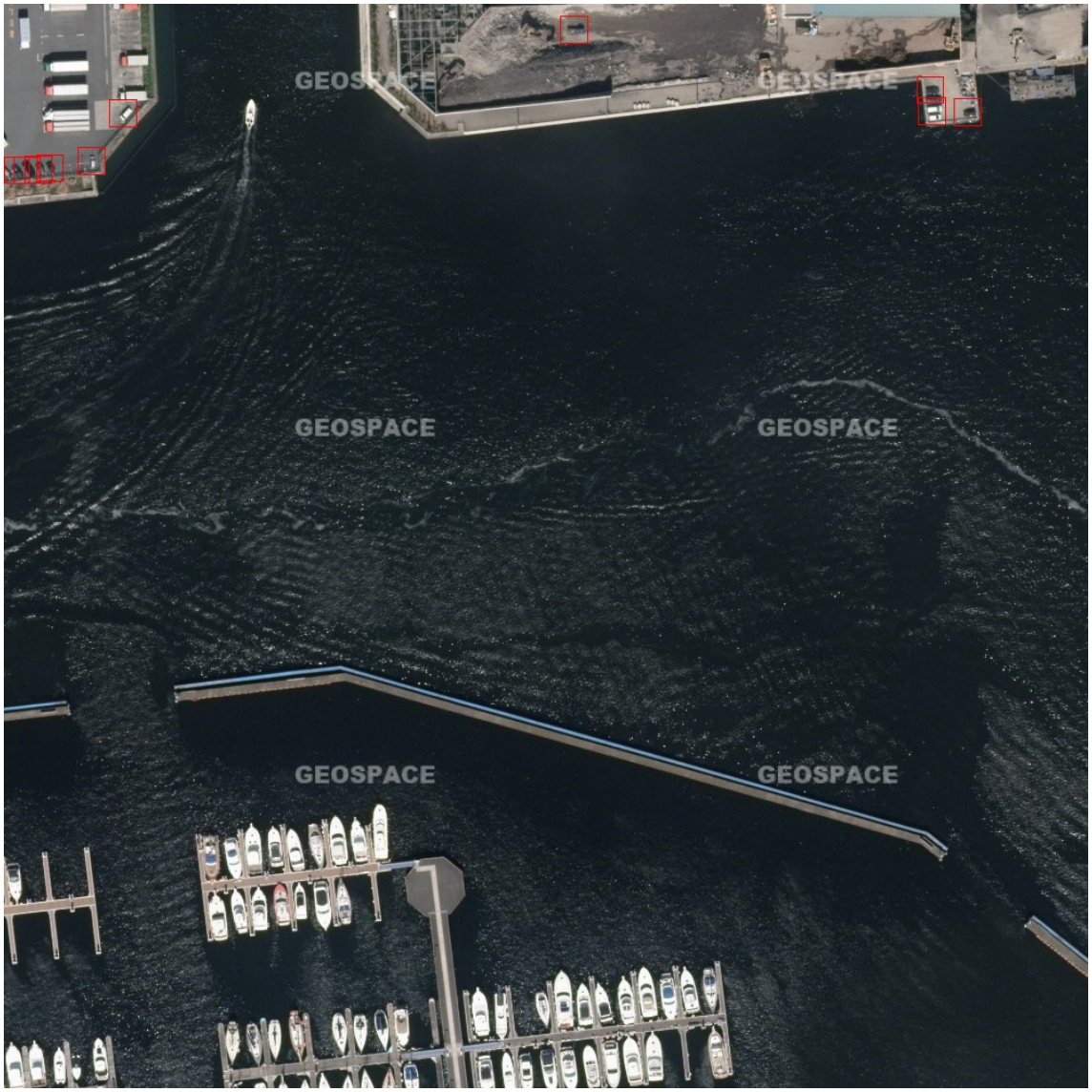


図 24: 画像 2 における、Adversarial DA の検出結果。赤：正解、青：誤検出

表 11: 画像 2 における、Adversarial DA の検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
76.9%	87.0%	81.9%	100.0%	76.9%	0.0%

図 25: 画像 3 の正解ラベル。

図 26: 画像 3 における、DA 無しの検出結果。赤：正解、青：誤検出

表 12: 画像 3 における、DA 無しの検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
54.4%	67.2%	60.8%	80.6%	57.6%	13.9%

図 27: 画像 3 における、CORAL DA の検出結果。赤：正解、青：誤検出

表 13: 画像 3 における、CORAL DA の検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
69.3%	78.1%	73.7%	87.0%	70.9%	10.6%

図 28: 画像 3 における、Adversarial DA の検出結果。赤：正解、青：誤検出

表 14: 画像 3 における、Adversarial DA の検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
71.8%	79.7%	75.7%	84.4%	75.5%	13.9%

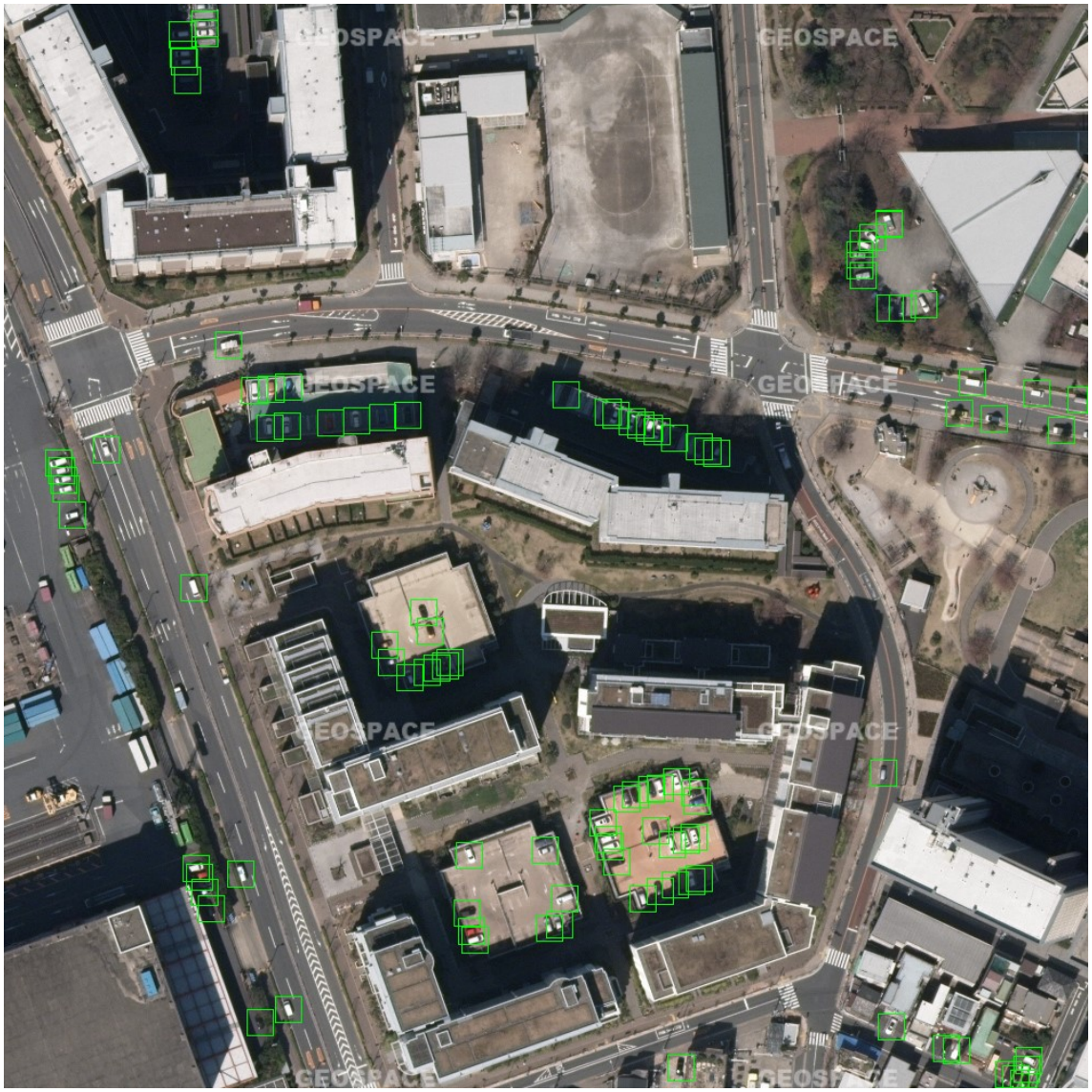


図 29: 画像 4 の正解ラベル。

図 30: 画像 4 における、DA 無しの検出結果。赤：正解、青：誤検出

表 15: 画像 4 における、DA 無しの検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
67.0%	75.6%	71.3%	78.4%	73.1%	20.2%

図 31: 画像 4 における、CORAL DA の検出結果。赤：正解、青：誤検出

表 16: 画像 4 における、CORAL DA の検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
65.5%	74.6%	70.1%	77.3%	72.1%	21.2%

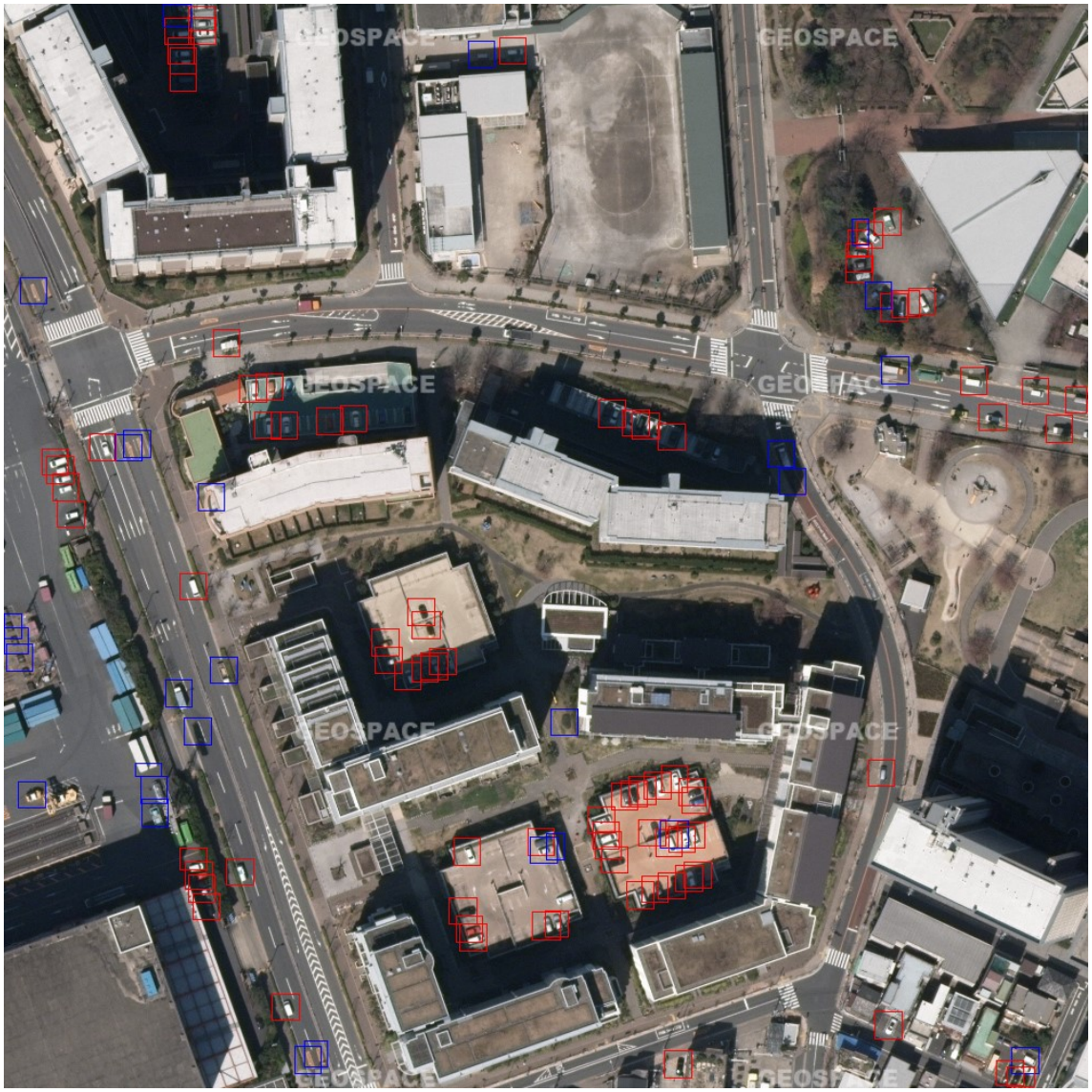


図 32: 画像 4 における、Adversarial DA の検出結果。赤：正解、青：誤検出

表 17: 画像 4 における、Adversarial DA の検出結果の各指標。

AP	F1	Mean of AP and F1	PR	RR	FAR
73.6%	77.2%	75.4%	74.8%	79.8%	26.9%

3.5.3 まとめ

SSD をトレーニングデータ（ソースドメイン）と異なる場所（ターゲットドメイン）で使用した場合、大幅に性能が低下した。これに対し、CORAL 及び Adversarial DA を適用することで、性能低下の半分以上を回復させることができた。本実験では、CORAL DA の

ほうが Adversarial DA よりわずかに良い結果となった。実際の検出結果画像を比較したところ、ターゲットドメイン特有の密集した地域で特に性能がよくなった。CORAL DA の車両検出の矩形はトレーニングデータのラベルの大きさより明らかに小さくなっているが、詳しい原因は不明であり、検証は今後の課題である。

以降では、これらの結果を踏まえ、少量のターゲットドメインにおけるラベル付きデータを使った場合に、どの程度さらなる精度向上が見込めるかについて検証する。

3.6 少量のラベル付きデータを用いた精度向上

DA の手法により、ラベル無しデータを用いて精度を大幅に改善させることができた。ここでは、少量のターゲットドメインのラベル付きトレーニングデータが得られたと仮定したとき、どの程度まで性能を向上させられるかを検証する。データとしては、3.1.2 で得たターゲットドメイントレーニングデータ（ラベル有り）を用いる。画像のサイズは 1000×1000pixel なので、ここから 300×300pixel の画像を抜き出してトレーニング画像とする。3.1.1 と同じく、画像を回転させて複製する Data Augmentation を適用する。

3.6.1 最も性能が良くなるトレーニング手順の検証

最終的な車両検出器を得る上でのトレーニング手順は様々なものがあるため、どの手順が最も性能がよくなるかを検証した。検証は、DA を含まない手法を含めて網羅的に行った。まず、トレーニングデータとして、ターゲットドメイントレーニングデータ（ラベル有り）の全ての画像(20 枚)を、300×300pixel の画像に重複が無いように分割した。これにより、564 枚のトレーニング画像を得た。ソースドメイントレーニング画像は 6,264 枚であり、4 地域の枚数が同じとすると 1 地域約 1,560 枚ほどであるから、約 1/3 の枚数ということになる。3.1.2 で述べたように、1000×1000pixel の画像 20 枚にラベルを付すのに要した時間は約 1 日である。

次に、トレーニング手順を考える。図 33 に、実験を行ったトレーニング手順の種類を示す。

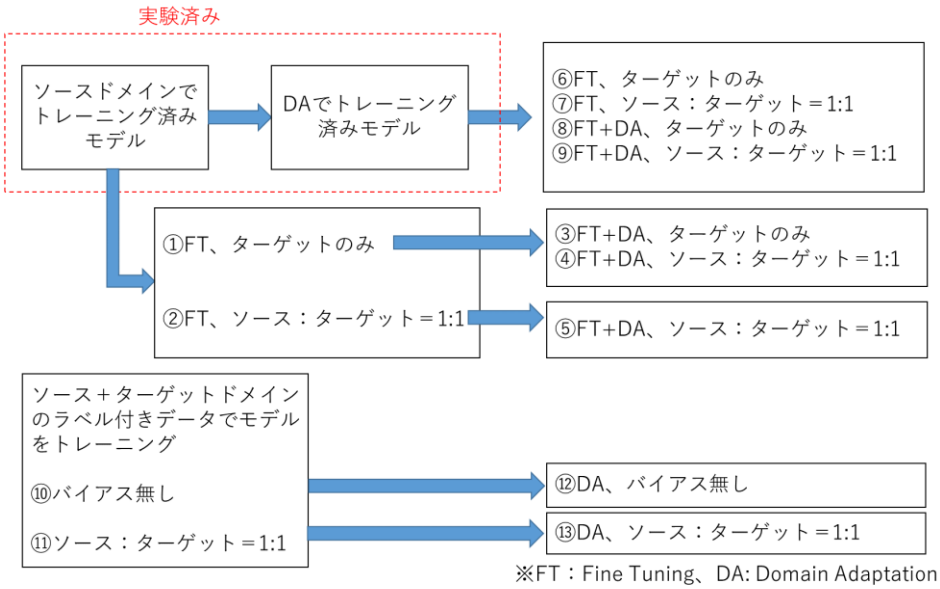


図 33: ターゲットドメインのラベル付きデータを用いた場合の、トレーニング手順の種類。

図 33 において、①～⑬が、最終的に得られるモデルの種類である。FT は Fine Tuning の略で、ターゲットドメインを含むデータを用いて、通常の手法でモデルを追加トレーニングすることを指す。⑩～⑪では、3.3.1 のトレーニングにおいて最初からターゲットドメインのデータを混ぜて SSD のトレーニングを行う。このとき、ミニバッチ内のソース及びターゲットドメインのデータの割合は性能に影響を与えると考えられる。①～⑨ではミニバッチのデータがターゲットドメインのみの場合と、ソースとターゲットドメインの量が等しい場合を実験した。DA のトレーニングにはラベル付きデータは含まれない。⑩～⑬では、ミニバッチ内のデータはソース及びターゲットドメインの割合にバイアスをかけない場合と、ソースとターゲットドメインの量が等しい場合を実験した。⑫～⑬は DA のみとなっているが、⑩～⑪のトレーニングデータにすでにターゲットドメインのデータが含まれるため、DA と同時に行う SSD のトレーニングのミニバッチ内にターゲットドメインのデータが含まれることになる。DA の手法は、これまでの実験で CORAL DA の方が僅かながらよい結果だったため、CORAL DA を用いた。ただし、これも興味深い事象であるが、⑥及び⑦の場合のみ、トレーニング開始後すぐにモデルが不正な値(NaN; Not a Number)を出力し崩壊してしまった。これは、CORAL DA の学習では数学的な制約がかかっており、この制約の上で学習したモデルに対し、条件を取り去ってトレーニングしようとしたことで、整合が取れなくなり崩壊してしまったものと思われる。Adversarial DA でトレーニングしたモデルではそのような問題は起こらなかったので、⑥及び⑦のみそれらを用いて実験を行った。

①、②、⑥、⑦、⑩、⑪の DA を用いない手順については、誤差関数の値が十分に収束したと思われるまでトレーニングを行った。他の DA を含む手順については、3.4.1 と同様に、

ターゲットドメインバリデーション画像で 10 イテレーションごとに性能評価を行い、十分に性能が上がるまでトレーニングを続け、スナップショットを保存した。表 18 に各手順におけるトレーニングの長さ、表 19 に DA を含む手順において最もよい AP と F1 の平均を記録したイテレーション数と、その数値を示す。

表 18: 各トレーニング手順におけるイテレーション数。

	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬
Training iterations	80000	120000	30000	30000	30000	150000	150000	50000	50000	50000	50000	50000	50000

※⑩及び⑪におけるweight decayのスケジュールは共に30000,40000

表 19: DA を含む各手順における、バリデーションのスコアの最高値とその時のイテレーション数。

	③	④	⑤	⑧	⑨	⑫	⑬
Iteration	29790	15210	28600	3570	47310	44610	35430
Mean of AP and F1	73.5%	79.2%	79.0%	86.7%	79.5%	80.3%	80.2%

図 34 に、各トレーニング手順による、ターゲットドメインテストデータ上のテスト結果における AP 及び F1 の平均値を示す。

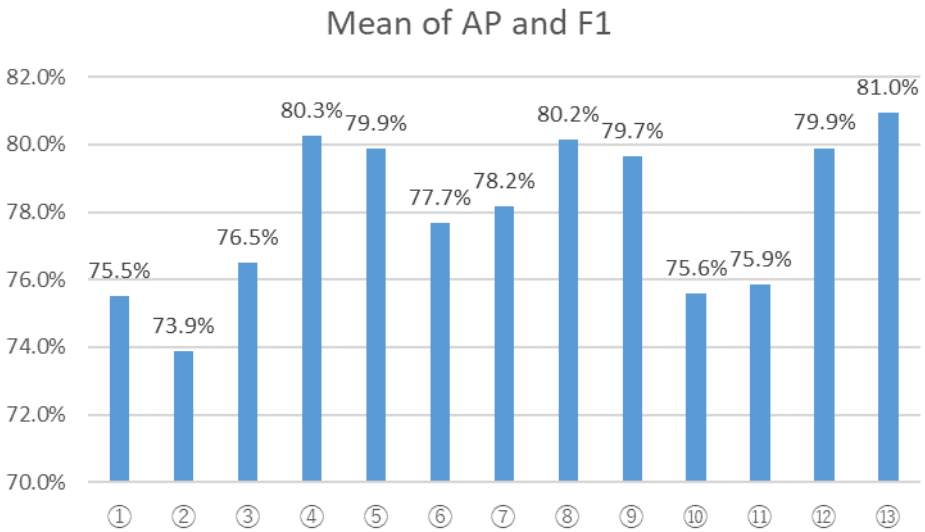


図 34: 各トレーニング手順による、ターゲットドメインテストデータ上の車両検出における AP 及び F1 の平均値。

図 34 において、DA を用いない①、②、⑩、⑪（⑥及び⑦は、FT を行ったトレーニング済みモデルが DA を使用している）は、3.4.2 の DA のみを用いたときの値とほぼ同じになった。特に性能が高くなった④、⑤、⑧、⑨、⑫、⑬については、少しの差ながら、⑬が最も高い 81.0%となった。3.4.2 における DA のみの結果から、ラベル付きデータを用いることでおよそ 4%さらに性能を改善させることができた。

3.6.2　ラベル付きデータの量と性能向上の幅

ラベル付きデータの量に応じてどのように性能が向上していくのかを検証する。手順は、3.6.1 で最も性能がよくなった⑬を用いる。3.6.1 では、ターゲットドメイントレーニング画像（ラベル有り）20 枚全てを使用したが、ここでは使用する画像の枚数を 4 枚から 4 枚ずつ増やしていき、20 枚までの 5 通りについて車両検出の性能評価を行い、どのように性能が向上するかを検証する。5 通りについて、3.6.1 と同様に重複無しで 300×300pixel の画像に分割したところ、それぞれ 132、220、348、456、564 枚のトレーニング画像を得た。これらを用いてトレーニング及びテストを行った。図 35 に、横軸をトレーニング画像の枚数、縦軸を AP 及び F1 の平均値として結果を示す。

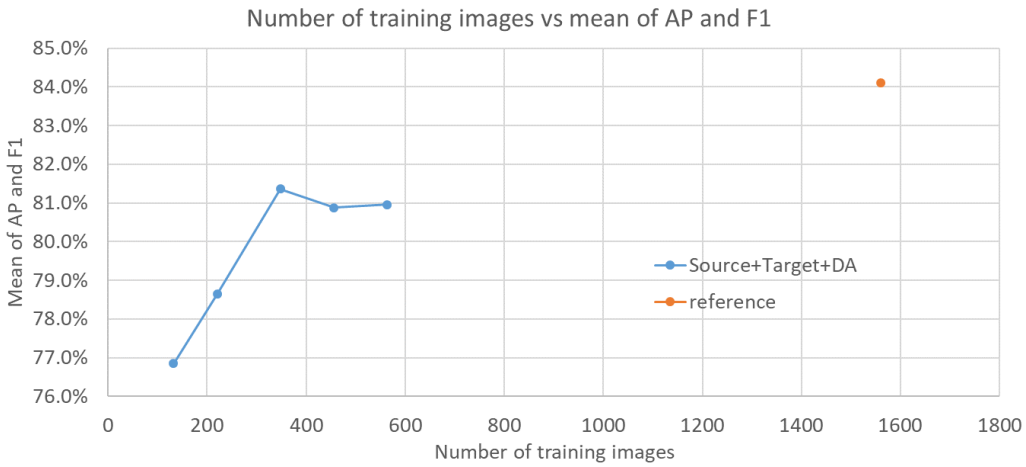


図 35: ラベル付きトレーニングデータの量を変化させた場合の、テスト結果。

トレーニング画像が 348 枚の場合が若干外れ値のようになっているが、それを除くと、データが増えるにしたがって性能向上はゆるやかになっているように見える。そのように仮定すると、トレーニング画像をそのまま増やしていっても参考値の精度には達しないと思われる。ただし、参考値は 4 地域における車両検出の平均の結果であり、全ての地域における検出精度が同じとは限らない。今回使用した日本の空撮画像は密集した地域が多く、車両検出も比較的難しいと思われるため、十分なトレーニングデータを用意したとしても、他地域に比べて性能が比較的低くなることはありうる。この仮定がもし正しければ、図 35 のト

レンドから、少量のラベル付きトレーニングデータ(400～600 毎程度)と DA 手法を組み合わせることで、十分な精度に達した可能性がある。これを確認するには、トレーニング画像をさらに 1,500 枚程度まで増やして検証を行う必要がある。これは今後の課題である。

3.6.3 Data Augmentation による精度改善

これまでは、トレーニング画像から 300×300pixel の画像を切り出すとき、重なり合う部分がないようにしてきた。3.1.1 のように画像を回転・複製させるほかにも、画像を切り出すときに重複を持たせることにより、Data Augmentation が可能である。これにより、さらに性能を改善させることができるかを検証する。

まず、有効性検証のため、トレーニング画像が少ない場合の実験を行った。ターゲットドメイントレーニング画像（ラベル有り）4 枚から、重複を上下 0, 30, 100, 200pixel として 300×300pixel の画像を抜き出した結果、それぞれ 132、136、244、992 枚の画像を得た。それぞれのトレーニング画像において、3.6.1 における⑬によってトレーニングを行い、テストを行った。結果を図 36 に示す。僅かながら、性能が改善した。

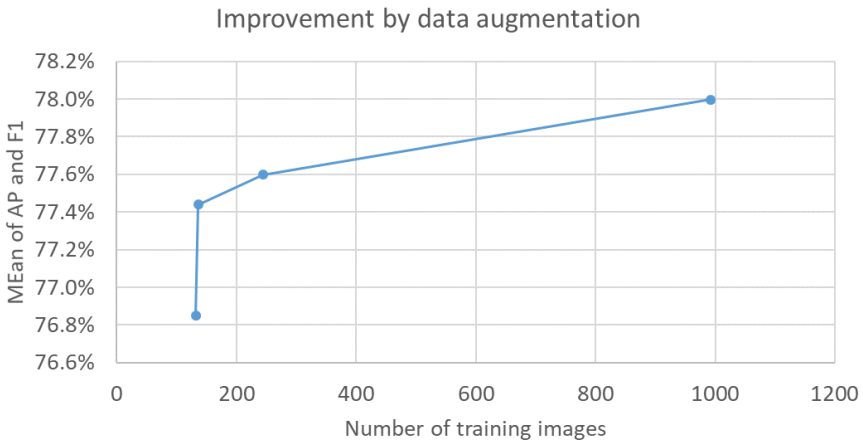


図 36: Data Augmentation による性能改善。

よって、最終的な車両検出器として、ターゲットドメイントレーニング画像（ラベル有り）を 20 枚全て用い、重複を 200pixel とし、に図 33 における⑬によってトレーニング及びテストを行った。トレーニング画像は 4,072 枚得られた。トレーニングは 50,000 イテレーション行い、バリデーションの最も良いスコアは 46,550 イテレーションのとき 80.3%となった。この時のスナップショットを用いて車両検出テストを行ったところ、AP 及び F1 の平均は 81.9%となり、3.6.1 における結果からさらに 0.9%改善した。

3.6.4　ターゲットドメインのデータのみ用いたトレーニング

比較のため、3.1.1 のデータを用いず、ターゲットドメインのラベル付きデータのみ用いたトレーニングを行った時、どの程度の性能になるか検証した。3.6.2 と同様の条件で、トレーニング画像の数を変化させて実験した。図 37 に結果を示す。

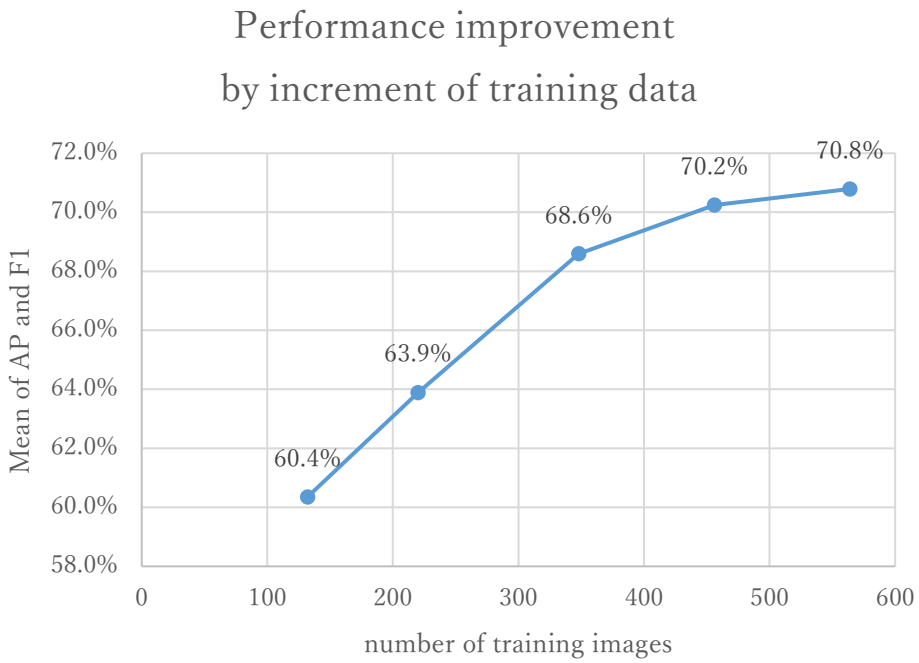


図 37: ターゲットドメインのトレーニングデータのみ用いたときの性能。

画像数を増やしていくことで、AP 及び F1 の平均は最終的に 70.8%に達した。これは、3.3.2 のソースドメインのみ用いて DA 無しの場合（66.3%）よりも高いが、3.4.2 の DA を用いた場合（76.8%及び 75.9%）よりは 5～6%低くなっている。

これに加え、3.6.3 と同様に Data Augmentation を用いたところ、性能が 75.6%と大幅に向上し、3.4.2 の DA を用いた場合に肉薄した（ただし、これまでの実験では画像重複による Data Augmentation は用いていないので正当な比較ではない）。この大幅な性能向上について考えられる理由としては、以下のようなものがある。Data Augmentation を用いない場合、トレーニング画像数は 564 枚と少なく、トレーニングデータの特徴を過剰に学習し汎化性能が低くなった(Overfitting)可能性がある。これが、Data Augmentation により画像の枚数が 4,072 枚と非常に多くなり、Overfitting を防いで汎化性能が高くなったのではないかと思われる。

3.6.5　まとめ

DA 手法に加え、少量のラベル付きトレーニングデータを用いることで、さらに性能を改善させることができ、AP 及び F1 の平均は最大で 81.9%に達した。DA と少量のラベル付きトレーニングデータのみで参考値と比べても十分な性能を達成できた可能性があるが、結論付けるにはさらなる検証が必要である。これまでトレーニングを行ってきた各モデルの性能の比較について、図 38 に示す。

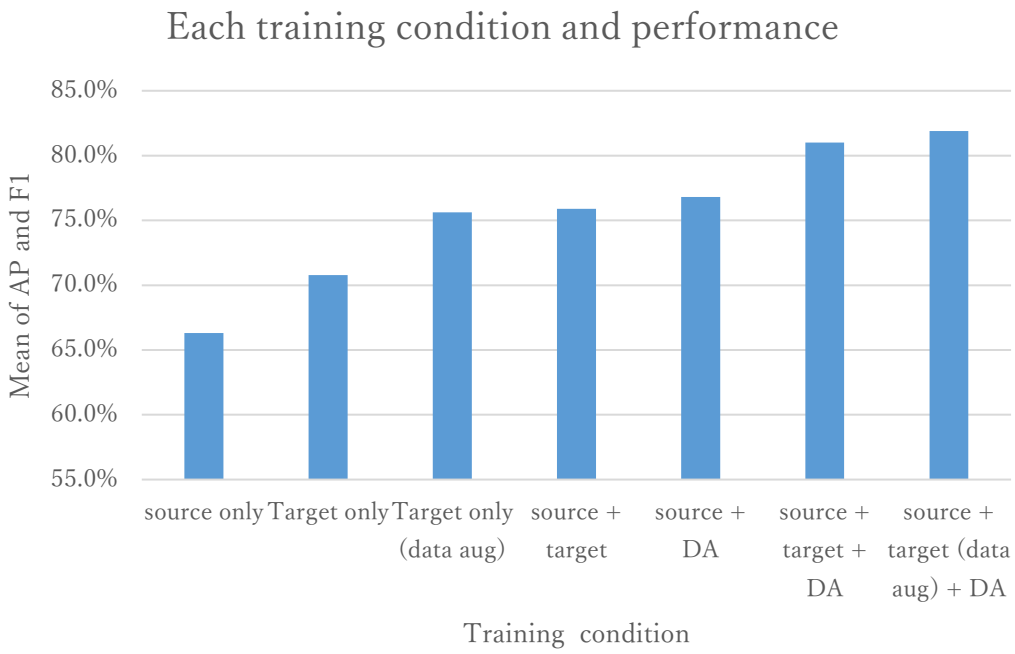


図 38: 各トレーニング条件と性能の比較。

第4章　車両検出器を用いた駐車場の車両数推定

得られた車両検出器の実用例として、衛星画像から車両数を計測することによる、商業施設の駐車場の車両数推定を行った。車両数は顧客数、ひいては売り上げ数と大きく相関があると思われるため、売り上げ推定等を行う上で有用と考えられる。本実験にあたっては、ある企業より、商業施設の現地調査やデータの提供等についてご協力頂いた。ただし、匿名性のため、企業名や施設名については伏せるほか、1 日の駐車車両数は実数ではなく、指標化した値で表す。

4.1 実験

推定手法は、衛星画像を用いて車両検出を行い、ある時間における上空から確認可能な車両数を計測し、そこから 1 日の駐車車両数の推定を行う。まず、推定を行う上でのトレーニングデータとして、ある商業施設の駐車場における、駐車車両数の調査を行った。調査を行った駐車場は立体駐車場で、屋上の駐車車両のみ上空から確認可能である。朝 10 時から夕方 6 時まで、1 時間おきに屋上の駐車車両数を実地に目視で計測した。調査日数は、時間の都合上少なくなってしまったが、2 週間の土日、計 4 日行った。調査が土日のみの理由は、平日は休日に比べて来客数が少なくなり、屋上に駐車車両があることがまれだからである。また、調査した日における、駐車場全体の駐車車両数について提供を受けた。これを基に回帰モデルを構築し、衛星画像を用いて計測した車両数から、1 日の車両数の推定を行う。1 日の駐車数は、調査を行った 4 日の平均を 1 として標準化した値をもって示す。

図 39 に、調査した時間毎の屋上駐車車両数、表 20 に各調査日の全駐車車両数を示す。

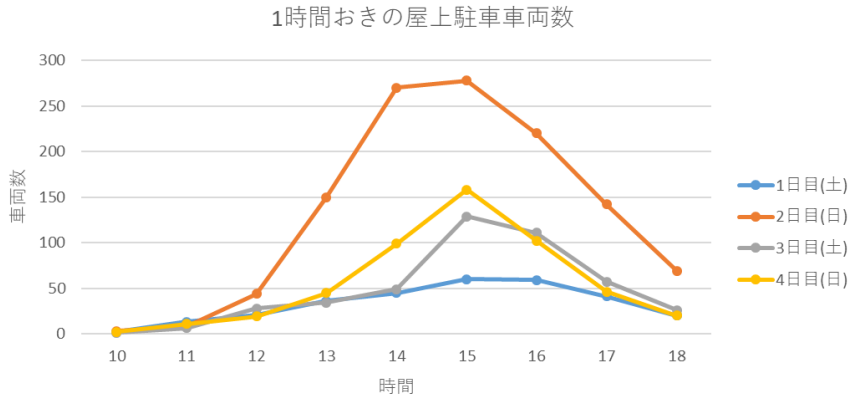


図 39: 1 時間おきの屋上駐車車両数。

表 20: 各調査日における、全駐車車両数。

1 日目 (土)	0.929
2 日目 (日)	1.035
3 日目 (土)	1.012
4 日目 (日)	1.023

図 39 及び表 20 から、屋上の車両数が多くなるほど、1 日の全駐車車両数も多くなっている。ただし、図 39 からわかるように、時間によって車両数の違いにばらつきがあり、一日の全車両数との相関にも違いがあると思われる。

回帰モデルとしては、ある時間の車両数を用いた単回帰、もしくは全ての時間の車両数を用いた重回帰分析が考えられる。ただし、衛星画像を用いて 1 日の全ての時間の車両数を

計測することは現実的には難しく、本実験においても 1 枚の画像における車両数から推定を行うため、重回帰分析を行う場合、1 つを除いた変数を何らかの手段で補完する必要がある。また、各時間における車両数同士はかなりの相関があると考えられるため、これが問題となる可能性がある（多重共線性と呼ばれる）。これらを考慮し、本実験では各時間の車両数を変数とした単回帰により推定を行う。

表 21 に、各時間を変数とする単回帰モデルを構築したときの、係数、切片及び決定係数を示す。

表 21: 各時間の単回帰モデルにおける、係数、切片及び決定係数。

	10時	11時	12時	13時	14時	15時	16時	17時	18時
係数	0.0115	-0.0107	0.0022	0.0004	0.0003	0.0004	0.0005	0.0006	0.0011
切片	0.9770	1.1015	0.9378	0.9703	0.9681	0.9320	0.9360	0.9588	0.9636
決定係数	0.04	0.48	0.28	0.27	0.37	0.68	0.55	0.32	0.28

表 21 が示すように、15～16 時の決定係数が高く、予測能力が高くなっている。これは、図 39 において 14～16 時で最も車両数が多くなっていることからも、妥当といえる。よって、15～16 時の最も駐車数が多くなる時間帯の車両数を用いて予測を行うのが、最も精度がよくなると考えられる。

これらの回帰モデルを用いて、実際に予測を行う。実験には、衛星画像の代わりとして、NTT 空間情報提供の空撮画像を、解像度を 0.3m/pixel に落として用いた。撮影時期は、2015 年 5 月 6 日 11：44 である。まず、3.6.3 で得たモデルを用いて、車両検出を行った。検出時、駐車場の範囲にマスクをし、他の領域で検出された車両は除外した。図 40 に検出結果を示す。

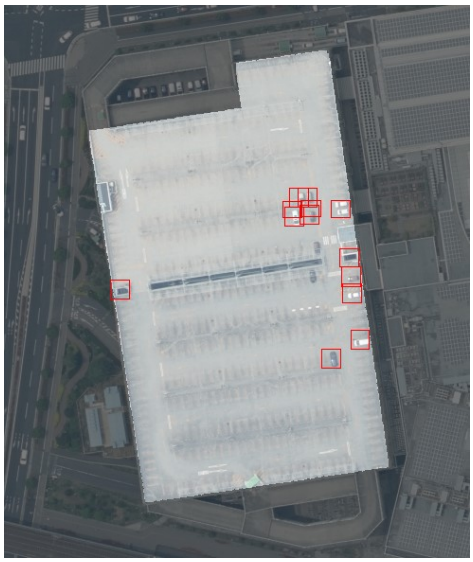


図 40: 車両検出結果。撮影時期: 2015 年 5 月 6 日 11：44。

図 40 において、画像中の実際の車両数は 13 のところ、検出結果も同じく 13 となった。内約は、未検出 3 台、誤検出 3 台（1 台は画像中左の完全な誤検出、2 台は検出の重複）である。車両検出を行った画像は 3.1.2 の画像と明るさなどの特徴がまた少し異なっており、それにより誤検出等が増えた可能性がある。

この検出数を用いて 1 日の駐車数の推定を行った。画像の撮影時刻は 11:44 なので、回帰モデルは 12 時台のものを用いた。ただし、決定係数は低くなっているため、精度のいい予測は難しい。予測結果は 0.967 となった。実際の駐車数は 0.976 であり、予測値との差は 0.009 と結果的には小さくなった。ただし、もともと屋上に駐車場が停まっているようなケースは駐車数が多くなり、予測の範囲も小さくなることに留意が必要である。

4.2 考察

今回、結果的にある程度の推定結果を得ることができたが、精度のよい推定のためには 1 日の駐車数と相関が高く、予測に適した時間帯で予測をすることが必要である。本実験のケースでいえば、15～16 時が予測に適していると考えられる。また、実験に用いたトレーニングサンプルが少ないため、より多くのサンプルを用いて回帰モデルを構築することが必要と思われる。

今回のような屋上のみ上空から駐車車両が確認なケースでは、そもそも屋上に車が停まっているのは一定以上の駐車車両数を超えた場合に限られるため、単純な回帰モデルで全体の駐車車両数を推定することは難しい。このようなケースのモデル化の可能性として、通信トラフィック理論のあふれ呼というものがある。あふれ呼とは、ある通信経路が全てふさがっているときに別経路に逃げるトラフィックのことである。このような考え方を用いることで、屋上のみの車両数から全体の駐車数をモデル化できる可能性がある。また、より推定が簡単な例として、まず地上駐車場から実験を行うことが考えられる。

第5章 結論

5.1 本論文の成果

本論文では、実用的な用途に向けた車両検出について、トレーニング地域だけでなく関心地域においても高精度が達成できるよう、主に DA 手法を用いた精度向上について検討を行った。車両検出手法には SSD を採用し、DA 手法として CORAL DA 及び Adversarial DA を提案し、性能の評価を行った。両手法とも関心地域における性能低下幅の半分以上を改善することができた。提案手法は教師無しの学習のため、少ないコストで大幅に精度を改善でき、実用的な車両検出に非常に有用と考えられる。さらに、DA 手法に加えて少量のラベル付きトレーニングデータを用いることで、さらに精度を改善させることができた。

最後に、車両検出の実用例として、得られた車両検出器を用いて、商業施設における駐車車両数の推定を試みた。小規模な実験ではあるが、今後の車両検出の実用例として、先鞭をつけることができたと考える。

5.2 今後の課題

まず、3.6.2 で挙げたように、DA 及び少量のラベル付きトレーニングデータによって十分な精度を達成できたといえるのか、さらなる検証を行う必要がある。また、今回提案 DA 手法によって大幅に精度を改善することができたが、今回用いた DA 手法は比較的シンプルな手法であり、改善の余地があると思われる。深層学習における DA 技術は近年非常に活発に研究が行われており、今後ますます高性能な手法が登場すると思われる。そのような手法を検討し、さらにラベル付きデータの使用等のコストを減らして精度を向上出来るよう、研究を進めていく必要がある。

また、車両検出を用いた経済活動量の推定は、例えば地域の活性度を把握するのに役立ち、効率的なビジネスや政策決定等に非常に有用なアプリケーションである。今回行ったような実験をより多くのデータを用いて行うことで検証の精度を高めるほか、さらに車両数だけでなくより具体的に売り上げの予測を試みるなど、研究を深めていくべきと思われる。

参考文献

[1] DigitalGlobe https://www.digitalglobe.com/ (accessed on 17 January 2018)

[2] WorldView-3 http://worldview3.digitalglobe.com/ (accessed on 17 January 2018)

[3] Planet Labs https://www.planet.com/ (accessed on 17 January 2018)

[4] Black Sky https://www.blacksky.com/ (accessed on 17 January 2018)

[5] NTT 空間情報 http://www.ntt-geospace.co.jp/ (accessed on 17 January 2018)

[6] ImageNet Large Scale Visual Recognition Challenge (ILSVRC) http://www.image-net.org/challenges/LSVRC/ (accessed on 17 January 2018)

[7] H. Miyazaki, X. Shao, K. Iwao, R. Shibasaki, “Development of a Global Built-Up Area Map Using ASTER Satellite Images and Existing GIS Data,” in Global Urban

Monitoring and Assessment through Earth Observation, London: CRC Press, 2014, p. 121.

[8] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, S. Ermon, "Combining satellite imagery and machine learning to predict poverty," in Science 2016, Vol. 353, Issue 6301, pp. 790-794, doi: 10.1126/science.aaf7894.

[9] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, E. L. Aiden, and L. Fei-Fei, "Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States," Proceedings of the National Academy of Sciences of the United States of America, vol. 114 no. 50, 13108–13113, doi: 10.1073/pnas.1700035114.

[10] Orbital Insight. https://orbitalinsight.com/ (accessed on 17 January 2018)

[11] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, Ohio, USA, 23 - 28 June 2014; page 580-587.

[12] J.R.R. Uijlings, K.E.A.v.d. Sande, T. Gevers, A.W.M. Smeulders, "Selective Search for Object Recognition," in International Journal of Computer Vision 2013, Volume 104, Issue2, page 154-171, doi:10.1007/s11263-013-0620-5.

[13] R. Girshick, "Fast R-CNN," ICCV '15 Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7-13 December 2015; page 1440-1448.

[14] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, 7-12 December 2015; page 91-99.

[15] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27-30 June 2016; doi:10.1109/CVPR.2016.91

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, "SSD: Single Shot MultiBox Detector," Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, Amsterdam, The Netherlands, 8-16 October 2016; Volume 9905, page 21-37, doi:10.1007/978-3-319-46448-0_2.

[17] X. Chen, S. Xiang, C.L. Liu, C.H. Pan, "Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks," IEEE Geoscience and Remote Sensing Letters 2014, Volume 11, Issue 10, page 1797-1801, doi:10.1109/LGRS.2014.2309695.

[18] S. Qu, Y. Wang, G. Meng, C. Pan, "Vehicle Detection in Satellite Images by Incorporating Objectness and Convolutional Neural Network," in Journal of Industrial and Intelligent Information 2016, Volume 4, Number 2, page 158-162, doi: 10.18178/jiii.4.2.158-162.

[19] M.M. Cheng, Z. Zhang, W.Y. Lin, P. Torr, "BING: Binarized Normed Gradients for Objectness Estimation at 300fps," 2014 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Columbus, OH, USA, 23-28 June 2014; page 3286-3293, doi: 10.1109/CVPR.2014.414.

[20] T. Tang, S. Zhou, Z. Deng, H. Zou, L. Lei, "Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining," in Sensors 2017, doi:10.3390/s17020336.

[21] R.E. Schapire, Y. Singer, "Improved Boosting Algorithms Using Confidence-rated Predictions," in Machine Learning 1999, Volume 37, Issue 3, page 297–336, doi: 10.1023/A:1007614523901.

[22] Car Localization and Counting with Overhead Imagery, an Interactive Exploration. https://medium.com/the-downlinq/car-localization-and-counting-with-overhead-imagery-an-interactive-exploration-9d5a029a596b (accessed on 17 January 2018)

[23] T.N. Mundhenk, G. Konjevod, W.A. Sakla, K. Boakye, "A Large Contextual Dataset for Classification, Detection and Counting of Cars with Deep Learning," in Lecture Notes in Computer Science, Proceedings of the ECCV 2016: Computer Vision—ECCV 2016,

Amsterdam, The Netherlands, 8-16 October 2016; Volume 9907, page 785-800, doi:10.1007/978-3-319-46487-9_48.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7-12 June 2015; page 1-9, doi:10.1109/CVPR.2015.7298594.

[25] K. He, X. Zhang, S. Ren J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27-30 June 2016; page 770-778, doi: 10.1109/CVPR.2016.90.

[26] U.S. Geological Survey https://www.usgs.gov/ (accessed on 17 January 2018)

[27] EarthExplorer https://earthexplorer.usgs.gov/ (accessed on 17 January 2018)

[28] A. Shrivastava, A. Gupta, R. Girshick, "Training Region-Based Object Detectors with Online Hard Example Mining," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27-30 June 2016; page 761-769, doi:10.1109/CVPR.2016.89.

[29] Y. Koga, H. Miyazaki, R. Shibasaki, "A CNN-based Method of Vehicle Detection from Aerial Images using Hard Example Mining," in Remote Sensing, 2018, 10, 124, doi; 10.3390/rs10010124.

[30] Y. Niitani, T. Ogawa, S. Saito, and M. Saito, "ChainerCV: a Library for Deep Learning in Computer Vision," ACM Multimedia, 2017.

[31] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556.

[32] K. M. Borgwardt, A. Gretton, M.J. Rasch, H.P. Kriegel, B. Schölkopf, A.J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," in Bioinformatics, vol. 22, no. 14, pp. e49–e57, Jul. 2006.

[33] B. Sun, J. Feng, K. Saenko, “Return of frustratingly easy domain adaptation,” in AAAI, 2016.

[34] S. J. Pan, I. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” in IEEE Trans. Neural Netw., vol. 22, no. 2, pp. 199–210, Feb. 2011.

[35] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia, “Semi-supervised transfer component analysis for domain adaptation in remote sensing image classification,” in Transactions on Geoscience and Remote Sensing, vol. 53, no. 7, pp. 3550–3564, 2015.

[36] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T., Darrell, “Deep domain confusion: Maximizing for domain invariance,” in CoRR abs/1412.3474, 2014.

[37] M. Long, Y. Cao, J. Wang, M.I. Jordan, “Learning transferable features with deep adaptation networks,” in ICML, 2015.

[38] B. Sun and K. Saenko, “Deep CORAL: Correlation Alignment for Deep Domain Adaptation,” ECCV 2016 Proceedings Part III, pp 443-450, 2016, doi: 10.1007/978-3-319-49409-8_35.

[39] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, “Simultaneous deep transfer across domains and tasks,” in IEEE International Conference on Computer Vision (ICCV), 2015.

[40] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain adversarial training of neural networks,” in Journal of Machine Learning Research, 2016.

[41] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in NIPS Workshop on Adversarial Training, (WAT), 2016.

[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Annual Conference on Neural Information Processing Systems (NIPS), 2014.

[43] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang and R. Webb, "Learning from Simulated and Unsupervised Images through Adversarial Training," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2242-2251, 2017, doi: 10.1109/CVPR.2017.241.

謝辞

まず、本論文では NTT 空間情報提供の空撮画像を実験に使用させていただきました。この場を借りてお礼申し上げます。ありがとうございます。

本論文執筆にあたっては、柴崎先生に多大なるご指導・ご支援を頂きました。お忙しい中、定期的にミーティングでご意見をいただき、非常に参考になりました。また、論文執筆、学会参加も積極的に薦めていただき、大変よい経験をさせていただきました。ありがとうございます。特任助教の宮崎さんには、この論文を執筆するにあたって取り組んだ各テーマにおいて、様々なご指導を頂きました。論文のアイデアの材料を頂いた他、特に論文執筆に関しては、論文構成、英文、レビューへの対応など、多大なるご支援を頂き、大変よい勉強をさせて頂きました。誠にありがとうございます。副指導教員の瀬崎先生には、主に技術的な観点から、論文の内容を深める有益なアドバイスを頂きました。ありがとうございます。Shao 先生のチームの皆さんにも、毎回のミーティングを通じて直接有益な意見を頂いただけでなく、皆さんの研究によい刺激を受けました。ありがとうございます。特に Guo 君とはテーマが近いこともあり、色々と学ばせて頂きました。修士 1 年の時には共に長崎大学で 1 か月勉強しました。長崎大の研究員の星さんと共に様々な観光地を回り、釣りをしたのはいい思い出です。ありがとう。また遊びましょう。研究室の同期である佐藤君、山本君、大崎君にもよくして頂きました。皆優秀で、お互い助け合いながら、学生生活を共に楽しく過ごせたと思います。ありがとうございます。また、自分は社会人として会社に籍を置きながら修士学生として勉強しましたが、会社の同僚の理解と助けがなければ成しえなかったと思います。ありがとうございます。最後に、いつも無条件で応援してくれる家族に感謝します。ありがとうございます。