令和３年度卒業論文

動画像からの物体検出のための U-Net3D の改良

宮崎大学工学部　情報システム工学科

中山　隼人

指導教員　椋木雅之

目次

１．　はじめに

物体検出とは、動画や単一画像の中に含まれる特定の物体の位置や範囲を推定する技術のことである。近年、物体検出の需要は高まっている。例えば、自動運転の分野では、車に搭載されたカメラで撮影した映像中から人や信号機を検出する際に物体検出が用いられる。また、マーケティングの分野では、顧客が興味を持った商品を分析するために、カメラ映像中の人や商品の検出に物体検出が使われている。

そのため、物体検出手法も多く研究、提案されている。例えば、Viola ら[1]は、Haar-like 特徴量と呼ばれる簡単な特徴量を使って、比較的性能の低い識別器（弱識別器）を順次大量に適用することで高精度に人の顔を検出する手法を提案した。この手法では、弱識別器の構築や選別は、人の顔を撮影した学習データを大量に与えることで自動的に行えるが、使用する特徴量は予め人間が考案したものであった。

これに対して、近年、深層学習を用いた物体検出手法が提案され、大きな成果をあげている。深層学習を用いた物体検出では、大量の学習データをもとに、対象物体の検出に有効な特徴量自体も学習の過程で獲得できるため、より高精度な物体検出が可能となっている。一方で、深層学習では対象物体のどのような特徴に着目して検出が行われるのか明確でなく、検出が失敗した場合の原因も解析が困難である。特に、物体検出を動画像に適用した場合、この問題が顕著に現れる。動画像は、連続する時刻で取得した画像（フレーム）を時間方向に並べたものであり、連続するフレームでは画像内容の変化は小さい。それにもかかわらず、連続するフレームに物体検出を適用すると、あるフレームでは適切に検出できたのに、ほとんど画像内容に変化のない次のフレームでは検出に失敗することがある。動画像内の物体を追跡するために、動画像に物体検出を連続して適用することがあるが、上記のような検出失敗で、追跡が途切れることが問題となっている。

そこで本研究では、動画像からの物体検出において、時間方向での物体検出結果を安定させることを目指す。そのために、従来のように動画像の各フレームに対して 1 枚ずつ独立に物体検出処理を適用するのではなく、時間方向に連続する数フレームを束ねて、一度に物体検出処理を適用する方法を提案する。これにより、前後のフレームの情報も物体検出に利用できるようになるため、連続して物体を検出できるようになると考えられる。具体的には、動画像からの物体検出に適用するために、深層学習による物体検出手法である U-Net3D[2]を改良した U-Net3DT を提案する。U-Net3D は、深層学習を用いた単一の画像からの物体検出手法である U-Net[3]を 3 次元ボリュームデータからの物体検出に拡張したものである。本研究では、連続する数フレームを束ねた画像群（クリップ）を 3 次元画像とみなし、U-Net3D を適用する。この際、クリップにおいて、各フレーム内（空間方向）と、フレーム間（時間方向）では性質が異なるため、U-Net3D において異なる処理を行うよう改良を加えた。

以下、2 章では深層学習による画像からの物体検出の従来手法について述べる。3 章では、

本研究の提案手法である動画像からの物体検出への U-Net3D の適用と改良（U-Net3DT）について説明する。4 章では提案した U-Net3DT による動画像からの物体検出手法を、フレーム毎に独立に物体検出する手法や動画像処理のための改良を加えていない U-Net3D による物体検出手法と比較し、提案手法の有効性を評価する。5 章では本論文の総括と今後の課題について述べる。

2. 深層学習による画像からの物体検出

2.1 単一画像からの物体検出

物体検出は、与えられた画像の中に写されている特定の物体の位置や範囲を推定する技術である。通常、物体検出は与えられた単一の入力画像に対して処理を行う。例えば、Violaら[1]は、与えられた 1 枚の画像の中から、顔の領域を推定する手法を提案している。この手法では、Haar-like 特徴量と呼ばれる簡単な特徴量を使って、比較的性能の低い識別器（弱識別器）を順次大量に適用することで人の顔を検出する。この手法を含め、従来は検出対象となる物体を表現する特徴量を人手で設計し、物体検出処理に与えていた。

これに対して、近年、深層学習を使った画像認識手法が多く提案され、物体検出でも高い性能を示している。例えば、SSD[4]は VGG16[5]と呼ばれる学習済みの深層学習ネットワークを特徴抽出器として用いて、検出対象となる物体の種類と位置を同時に求めている。また、YOLO[6]も同様に畳み込みネットワーク（CNN）と呼ばれる構造を利用して特徴抽出を行い、検出対象となる物体毎の信頼度スコアを算出して物体検出を行っている。深層学習を利用して多量のデータを学習することにより、人手で設計するよりも高性能な特徴抽出器が構築でき、高精度な物体検出が行える。

深層学習を使った物体検出手法の一つに U-Net[3]がある。U-Net は、生物医科学の画像への適用目的で開発された。前述の SSD や YOLO が、物体検出結果として物体を囲う矩形の枠（バウンディングボックス）を出力するのに対して、U-Net は、画素単位でより細かく物体の領域を出力するセマンティックセグメンテーション[7]と呼ばれる手法になっている。U-Net は特定の細胞領域の検出や臓器領域の検出に使用されている。U-Net が適用される画像は、特定分野の類似した画像となるが、特定分野の比較的少数の学習データで高い検出精度が得られる。

U-Net のネットワーク構造を図 1 に示す。U-Net は、U 字型の構造になっているのが特徴である。入力画像と出力画像はともに単一の 2 次元画像である。U-Net は元の 2 次元画像の大きさを縮小しながら分析・特徴抽出（畳み込み）を行い、復元（逆畳み込み）の時には逆に画像の大きさを拡大していく。このような構造を Encoder-Decoder 構造と呼ぶ。従来の Encoder-Decoder 構造では、畳み込みを行っていくと位置情報が曖昧になり、復元する際の精度が落ちるという問題があった。しかし、U-Net では畳み込み層で出力される特徴マップ（物体の位置情報）を逆畳み込み層に連結することでその問題を解決し、精度の高いセグメンテーションを可能にした。

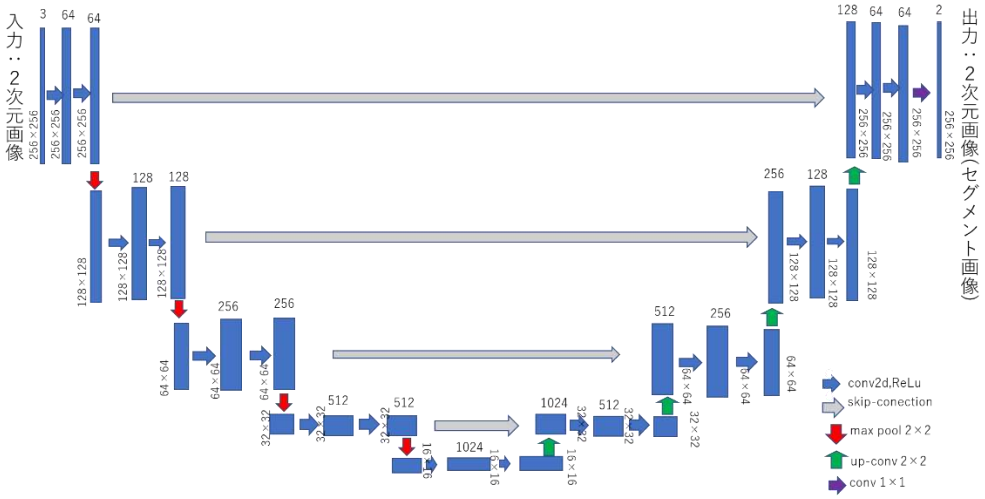


図 1：U-Net のネットワーク構造

2.2 ボリュームデータからの物体検出

物体検出は医療現場で見られる MRI や X 線 CT のボリュームデータ(3 次元画像)にも適用される。図 2 に X 線 CT で得られる腎臓のボリュームデータの例を示す。ボリュームデータは、3 次元空間を一定の大きさの小さな箱（ボクセル）に区切り、各ボクセルに値を保持させた 3 次元データである。MRI では、各空間位置にある臓器の水分量（水素原子量）を反映した値が、X 線 CT では各臓器の X 線吸収量を反映した値が格納される。これらの値の違いにより、体外から各臓器の位置や大きさを観測することができる。

Taleb ら[9]は、MRI から得られた人の脳のボリュームデータから脳腫瘍を検出するタスクや、X 線 CT から得られたボリュームデータから膵臓腫瘍を検出するタスクに、深層学習による物体検出を適用している。また、Çiçek ら[2]は、カエルの内部のボリューム画像から腎臓を正確に検出する研究を行っている。Çiçek らの研究では、3 次元画像を適用するために、2 次元画像を対象としていた従来の U-Net を 3 次元に拡張した U-Net3D を提案している。図 3 に、U-Net3D のネットワーク構造を表す。2.1 節で述べた通常の U-Net の畳み込み（分析工程）では、入力が 2 次元画像であったのに対し、U-Net3D は 3 次元画像を入力とした畳み込みになっており、画像を分析しながら縮小していく。逆畳み込み（復元工程）も同様に、通常の U-Net が 2 次元画像を復元していくのに対し、U-Net3D は 3 次元画像を復元しながら、拡大していく。逆畳み込みの際に、位置情報が曖昧になるため、畳み込み層の特徴マップを参考にする仕組みは、通常の U-Net のネットワーク構造と同様である。この従来研究[2]において、2 次元による畳み込み（U-Net）をボリュームデータの各断面の 2 次元画像（スライス）に適用するより、3 次元による畳み込み（U-Net3D）をボリュームデータに直接適用する方が精度良く腎臓領域を検出できるという実験結果が示されている。

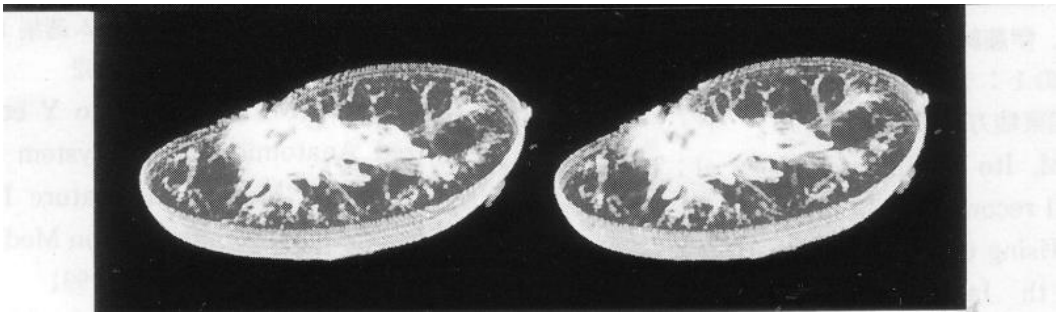


図 2：腎臓のボリュームデータ [8]

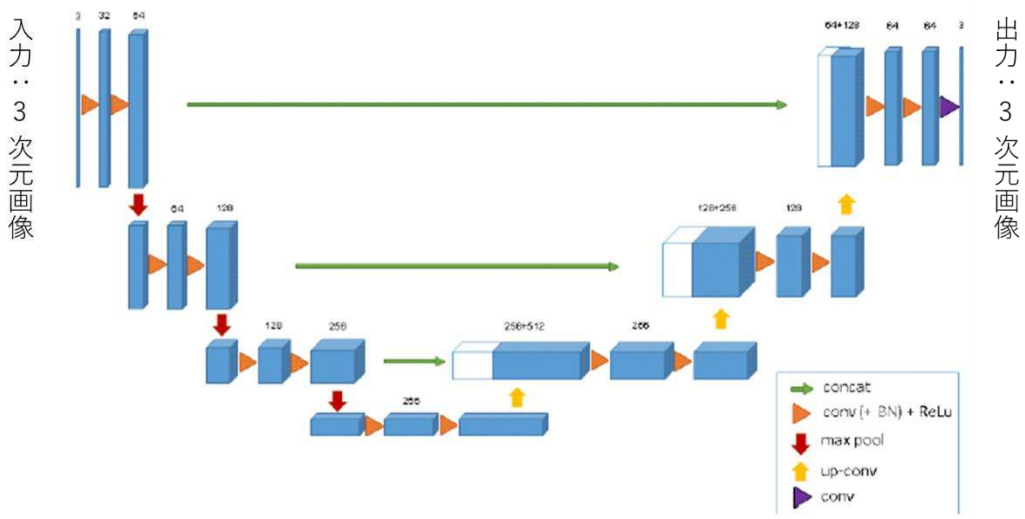


図 3:U-Net3D のネットワーク構造[2]

2.3 動画像からの物体検出

動画においても物体検出が適用されている。連続するフレームに物体検出を順次適用し、フレーム間で対応付けることで、物体を追跡することができる。このような物体追跡へのアプローチは Tracking by detection（検出による追跡）と呼ばれ、近年、多用されている。その代表例として、SORT[10]と呼ばれる手法があり、フレーム間で近い位置に検出された物体同士を対応付けることで追跡を行う。Shuai[11]らの手法は、上記の SORT を改良したものである。YOLO で検出したバウンディングボックスを使用して、フレームの前後で近い大きさと近い動きのバウンディングボックスを対応づけることで、検出対象に ID をつけて追跡を行う。DeepSORT[12]も SORT を改良したモデルであり、外観の類似度を比較する AI モデルを使用することで、対応付けに見た目の類似度の情報を利用する。これらの手法はフレームの情報を保持するが、物体検出に一定時間続けて失敗すると、そのデータが破棄され、追跡が途切れるという共通した問題点がある。

動画像は、連続する時刻で取得した画像（フレーム）を時間方向に並べたものである。連続するフレームでは画像内容の変化は小さい。それにもかかわらず、連続するフレームに物体検出を適用すると、あるフレームでは適切に検出できたのに、ほとんど画像内容に変化のない次のフレームでは検出に失敗することがある。上記のような Tracking by detection のアプローチをとると、このような検出失敗により追跡が途切れることが問題となる。

3. 動画像における U-Net３D の改良

3.1 U-Net3DT による動画像からの物体検出

従来のように動画像の各フレームに対して 1 枚ずつ独立に物体検出処理を適用するのではなく、時間方向に連続する数フレームを束ねて、一度に物体検出処理を適用する方法を提案する。この束ねた画像のことをクリップと呼ぶ。クリップ毎に処理することで、前後のフレームの情報も物体検出に利用できるようになるため、連続して物体を検出できるようになると考えられる。

具体的には、動画像からの物体検出に適用するために、深層学習による物体検出手法である U-Net3D[2]を改良した U-Net3DT を提案する。U-Net3D では 3 次元空間に対応するボリュームデータを処理対象としていた。U-Net3DT では、画像を時間方向に並べたクリップを 3 次元画像とみなし、処理対象とする。その際、分析工程で時間方向に縮小されると物体の細かい変化が無視されてしまい、物体の位置情報が曖昧になる。また、フレーム間での情報が混在してしまい、結果的に精度の悪い検出になる。従来の U-Net や U-Net3D は復元の際に分析工程で保持していた特徴マップを参照することで、物体の位置情報を補っていたが、本研究で扱うデータはそれでは不十分である。提案手法は分析工程で分析しながら画像を縮小していくが、時間方向へは縮小をさせないようにした。同様に、復元工程でも時間方向への拡大はしないようにした。時間方向への縮小をなくし、時間方向を細かく見ることで物体の小さな変化（動き）も処理結果に反映させる。

3.2 U-Net3DT のネットワーク構造

図 4 に U-Net3DT のネットワーク構造を示す。U-Net3DT のネットワーク構造は、U-Net3D とほぼ同じであるが、扱うデータの性質が異なる。U-Net3D が扱う 3 次元データは、縦×横×高さの空間的な 3 次元画像であるが、U-Net3DT が扱う 3 次元データは、縦×横×時間となっている。分析工程では、縮小の際に maxpooling を適用している。maxpooling は、処理の範囲（カーネルサイズ）内のデータの最大値を出力する処理である。処理範囲をずらしながらデータに適用することで、局所的な各部から情報を抽出する。この処理範囲をずらす幅をストライドと呼ぶ。U-Net3D の分析工程では、maxpooling における縦×横×高さのカーネルサイズとストライドを 2×2×2 としている。これにより、縦、横、高さがすべて 1/2 になる。一方、U-Net3DT では maxpooling における縦×横×時間のカーネルサイズとストライドを 2×2×1 とする。これにより、時間方向の情報が混在することを避け、同時に時間方向にデータが縮小されることを防いでいる。これに対応して、復元工程でも時間方向への拡大は行わず、復元させる。

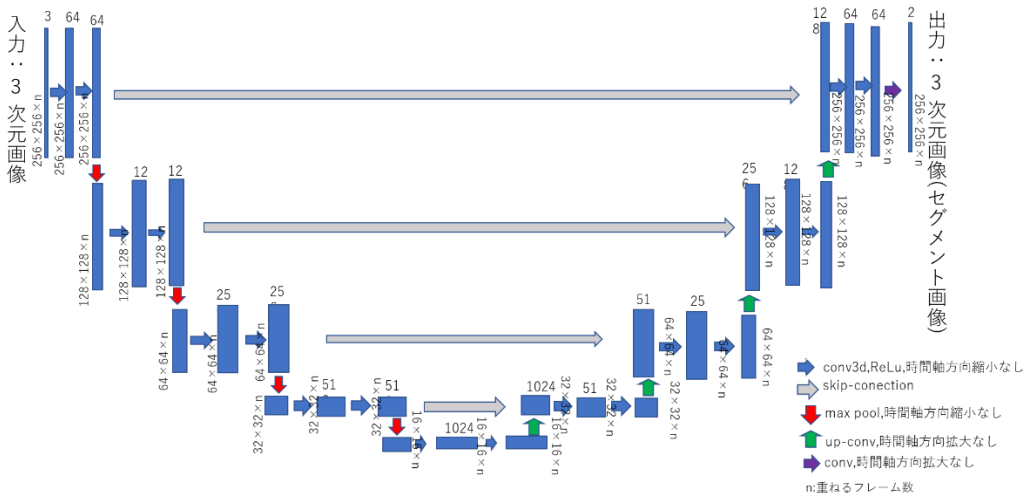


図 4：U-Net3DT のネットワーク構造

3.3 U-Net3DT の学習と結果の出力

提案した U-Net3DT の学習時には、動画像とその各フレーム内の物体位置を表す正解画像を用意する。正解画像は、物体の存在領域を 1、それ以外の背景部分を 0 にした 2 値画像である。動画像の時間方向に連続したフレームを複数枚（n 枚）束ねた動画像クリップと、動画像クリップの各フレームに対応する正解画像を束ねた正解クリップの組が学習データとなる。学習データを U-Net3DT に与えて学習することで、学習モデルを作成する。

物体検出結果を得る際にも、n 枚のフレームを時間方向に束ねた動画像クリップを作成し、入力として、学習済みの U-Net3DT に与える。出力として、n 枚のフレームに対するセマンティックセグメンテーションの結果が 2 値画像のクリップとして一気に得られる。

4. 実験

4.1 実験方法

本実験では、提案手法である U-Net3DT（3DT）が従来手法の U-Net（2D)、U-Net3D（3D）と比べて物体検出がどの程度安定するかを調査する。

実験データとして時間方向にあまり変化しない牛の動画像を用いる。表 1 に学習データの内訳、テストデータの組み合わせを示す。実験は 3 つの組み合わせで行う。test1 は 1 頭の黒牛の動画像データ(CIMG0120)であり、同じ動画像内でのデータを分割し、学習データとテストデータに分けた。学習データは動画像の前半 256 フレームを使用し、残りの 98 フレームを評価用のテストデータとした。この設定は U-Net で一般に想定される使い方である。test2 は test1 と同じ学習済みモデルを使用し、テストデータを黒牛 1 頭と白牛 1 頭が映る別の動画像データ(CIMG0123)とした。学習データに含まれない白い牛を含んでおり、学習データと違う場面への適用を想定した設定である。test3 は test2 と同様、test1 と同じ学習済みモデルを使用し、テストデータを黒牛が 2 頭映る別の動画像データ(MVI_357)とした。この場合も、学習データとテストデータは異なる場面であるが、黒い牛を対象としている点は類似している。物体検出の正解画像は labelme というツールを使って作成した。元の動画像の大きさはすべて 1920×1080 であったが、実験に用いたネットワークの入力での画像の大きさは、256×256 である。そのため、画像をリサイズをして、学習と評価に使用する画像の大きさは 2 次元画像、 3 次元画像ともに 256×256 とした。

本研究で扱う深層学習のモデルは何も学習していないものを 1 から学習させる。学習回数はすべての場合で 20 回とした。3DT ではクリップの長さを 2,4,8,16,32 フレーム、3D は 8,16 フレームとして、学習、評価を行った。

表 1 ：データセットの組み合わせ

	学習データ	テストデータ
test1	・CIMG0120(黒牛 1頭) フレーム数 256枚	・CIMG0120(黒牛 1頭) フレーム数 98枚
test2	・test1と同じ	・CIMG0123(黒牛 1頭　白牛 1頭) フレーム数 216枚
test3	・test1と同じ	・MVI_357(黒牛 2頭) フレーム数 280枚

CIMG0120

CIMG0123

MVI_357

図 5：使用した動画像データ

4.2 評価方法

本研究では物体検出結果の安定性と精度を評価するために 2 つの評価指標を用いる。 1 つ目は各学習済みモデルの出力において、時間方向で隣り合う 2 フレーム比較し、画素値が異なる画素数を集計した（隣比較)。この値が小さい程、フレーム間での検出結果の変化が少なく、安定している。2 つ目は各学習済みモデルの出力において、各フレームと正解画像を比較し、画素値が異なる画素数を集計した（正解比較)。この値が小さい程、検出結果が正解画像に近く、精度が高い。

4.3 実験結果

図 6~8 は、学習済みの 2D、3D、3DT（クリップの長さ 8）を用いて、test1,2,3 のテストデータから物体検出した結果の例である。図 6 より、学習データと同じ場面でのテストデータに対しては、どの手法でもほぼ正しく物体検出されていることが分かる。一方図 7 では、牛だけでなく柱や柵も牛として検出されており、検出精度が低い。図 8 では、主に牛の領域が検出されているが、検出できていない部分も見られる。

図 9 に test1 の結果を集計したグラフを示す。青いグラフは 2D、オレンジのグラフは 3D、灰色のグラフは 3DT を表す。横軸の数字はクリップの長さを指す。縦軸は異なる画素値をもつ画素の数の平均で図 9（a）が隣比較、図 9（b）が正解比較の結果である。図 9（a）から隣比較では、クリップの長さを長くしていくと 3D は値が大きくなっていき、安定しない。対して、3DT の場合はクリップの長さを 16 フレーム以上にすると値は小さくなっていき、2D よりも安定している。しかし、2D,3D,3DT の結果はすべて近い値である。test1 の正解画像で隣比較の平均は 1003 であった。これと比べると、図 9（a）の隣比較の値はだいぶ小さくなっている。図 9(b)の正解比較においても、3DT は 2D,3D と比べてどのクリップの長さにおいても小さな値（正解画像に近い結果）であった。しかし、これもほぼ誤差の範囲であり、特段優れているとは言えない。test1 では、学習データとテストデータが類似した画像であったため 2D,3D,3DT いずれも正確に物体検出ができていると考えられる。

図 10 に test2 の結果を示す。図 10（a）の隣比較ではクリップの長さによっては 2D,3D の両方と比べて、3DT の方が値が小さくなっている。しかし、全体的な傾向はみられない。図 10（b）の正解比較の結果ではクリップの長さ 16 の 3D が最も良い。しかし、正解画像と異なる画素数は全体的に大きな値になっている。図 7 の結果からも test2 は 2D,3D,3DT のすべてにおいて物体検出の精度が悪く、手法間の違いの傾向が読み取りにくい結果となった。

図 11 に test3 の結果を示す。図 11（a）の隣比較で 2D よりも 3D と 3DT は値が小さくなっている。クリップの長さ 8,16 を見てみると 3DT の値が一番小さい。また、図 11（b）の正解比較でもクリップの長さ 8,16 の 3DT が値が小さく、精度が高かった。test3 では学習データとテストデータが少し類似しているためクリップの長さによっては良い値が出たと考えられる。

以上のことをふまえると、3DT は物体検出の安定性においてクリップの長さによっては、2D よりも少し優れた結果を出した。また、3D より 3DT の方が全体的に良い結果を示した。

最後に、図 12 に 3DT のフレーム毎の検出結果を示す。横軸がフレーム番号、縦軸が異なる画素値をもつ画素の数を示す。折れ線グラフの青色が隣比較の結果、オレンジ色が正解比較の結果である。どの test においてもクリップの長さの倍数の時にグラフの値が大きくなっているのが分かる。これは 3DT の学習時に動画像を束ねたことに原因がある。束ねて

学習した場合、それを一つのクリップとして処理する。クリップの切れ目では、クリップの長さ分だけ異なる動きが出てくるので、物体検出の精度としては落ちてしまう。検出精度や安定性の向上のためには、このような変動を抑える必要もある。

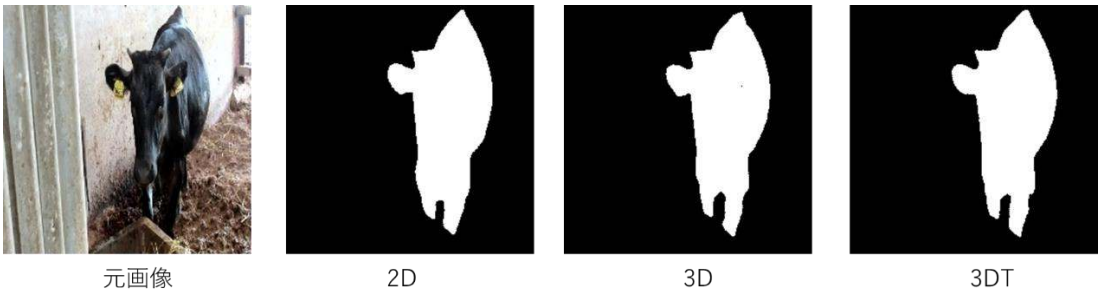


図 6： test1 の出力結果 クリップの長さ：8

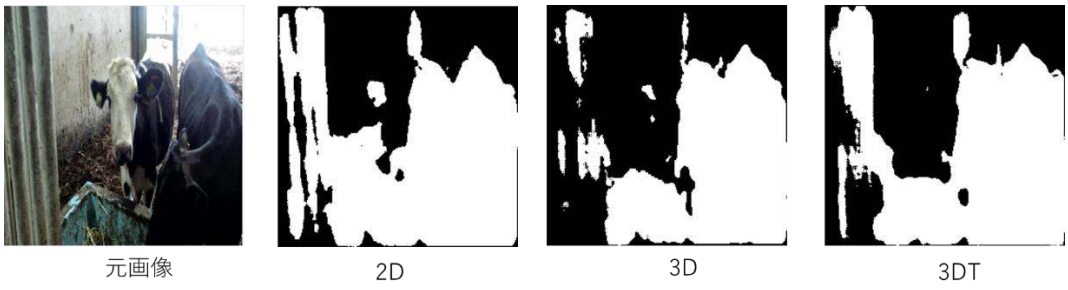


図 7：test2 の出力結果 クリップの長さ：8

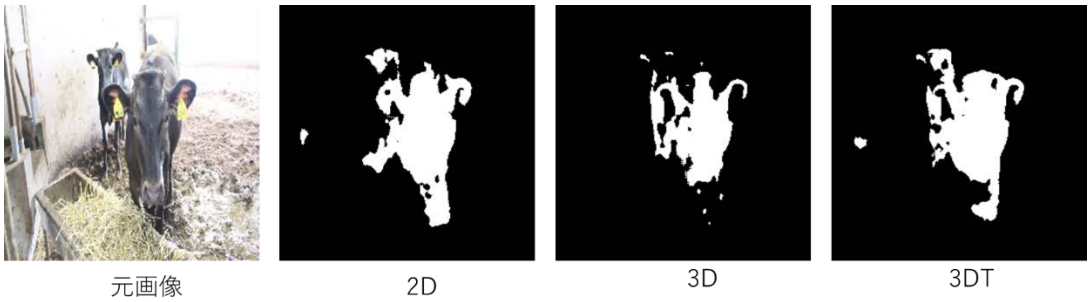
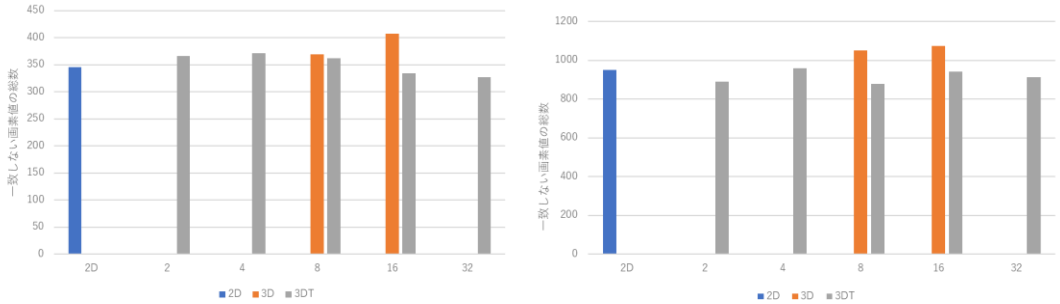
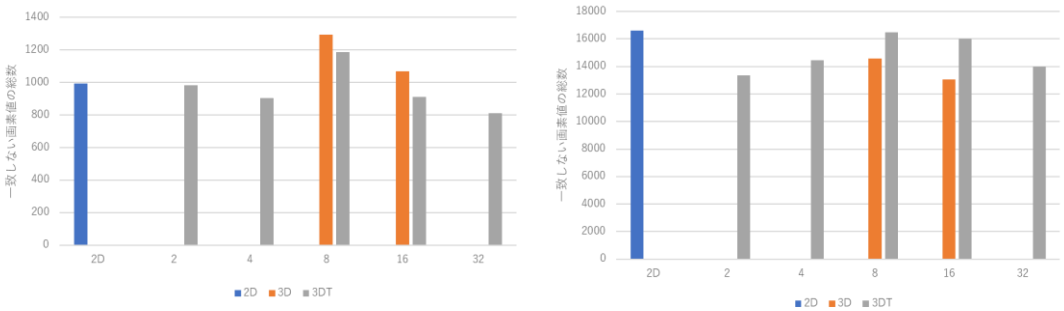


図 8 ：test3 の出力結果 クリップの長さ： 8



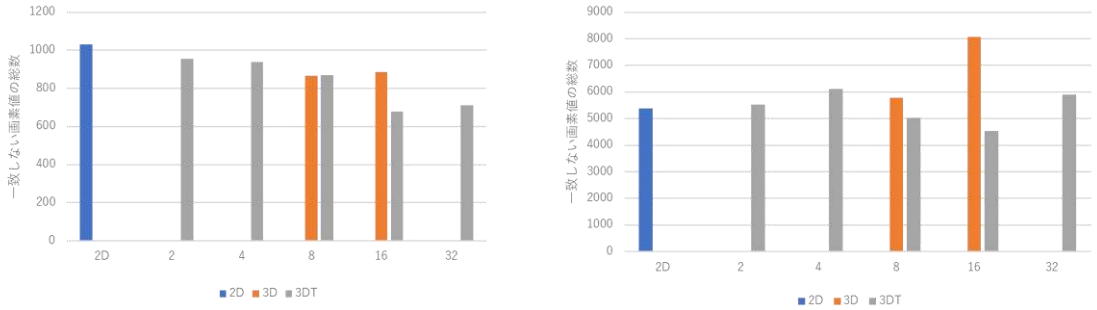
(a)test1 隣比較　　(b)test1 正解比較

図 9：test1 の結果



(a)test2 隣比較　　(b)test2 正解比較

図 10：test2 の結果



(a)test3 隣比較　　(b)test3 正解比較

図 11：test3 の結果

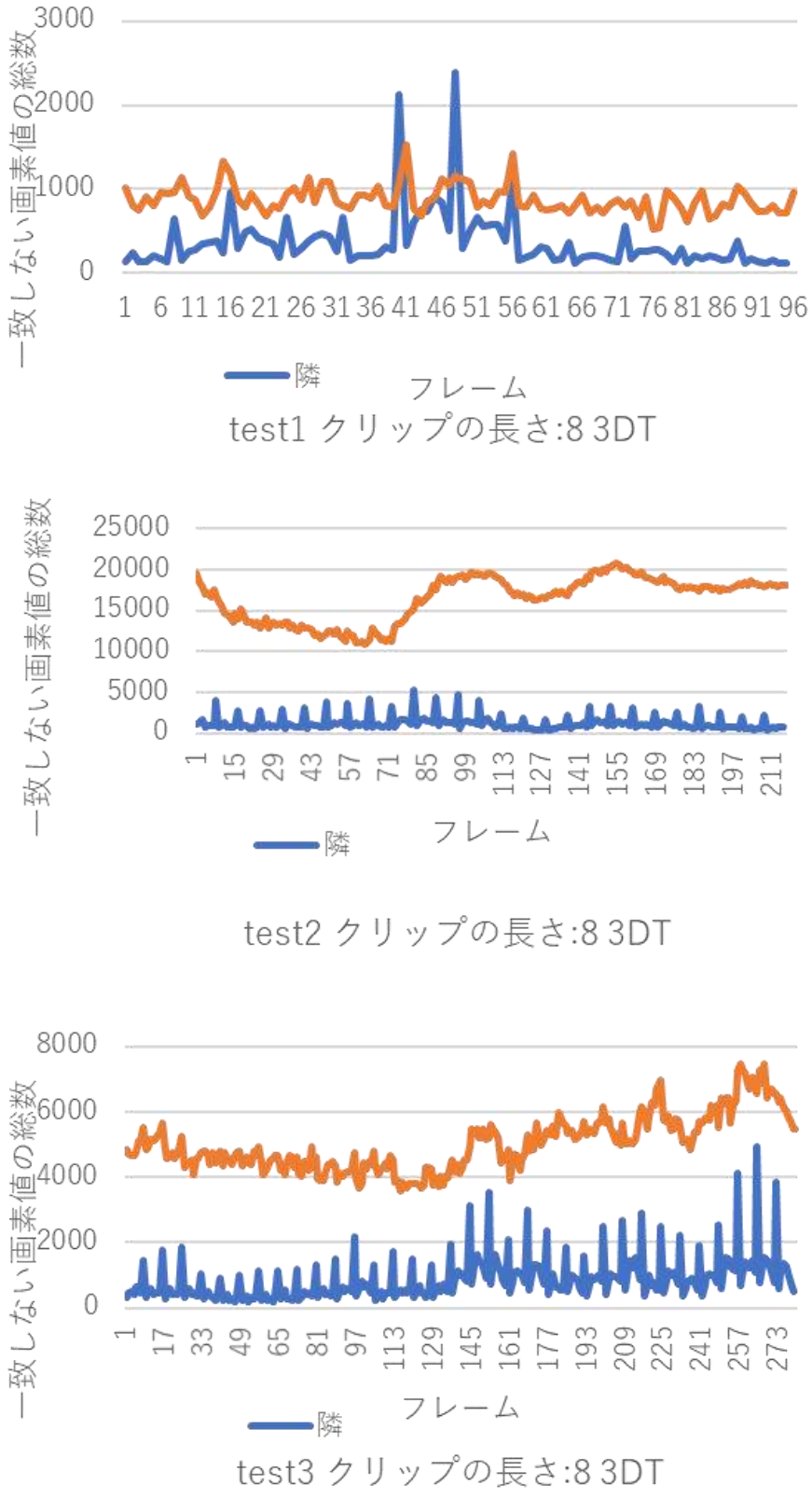


図 12：3DT のフレーム毎の結果

5. おわりに

本研究では動画像からの物体検出において、時間方向での検出結果の変動を安定させるために U-Net3DT を提案し、その評価実験を行った。

U-Net3DT は、２次元画像へ適用される U-Net を３次元画像へ拡張した U-Net3D に改良を加えたものである。U-Net3DT は、動画像の時間的に連続するフレームを束ねたクリップを３次元画像とみなして、処理を行う。クリップを処理単位とすることで、前後のフレームの情報を利用でき、物体検出が安定すると考えた。この際、時間方向に対しては、細かな情報を失わないように、通常の U-Net で行われる縮小処理を適用しないように改良した。評価実験の結果では、安定性、物体検出の精度の２点においてクリップの長さによっては U-Net よりも優れていた。また、U-Net3D よりは全体的に優れていた。しかしながら、クリップの長さによっては同程度あるいは少し劣る場合もあった。

本研究では学習データとテストデータの数が少なく、十分に検証できたとは言えない。今後の課題として、より多くの動画像データを集め、追加実験を行うことがあげられる。また、クリップを単位としたことにより、クリップの切りかわり時に検出結果の大きな変動が見られた。クリップを時間方向に重複してとるなど、さらなる安定化、高精度化に向けた検討も必要である。さらに、YOLO などの他のニューラルネットワークでも動画像を束ねることの効果があるかの検証も今後の課題とする。

謝辞

最後に、研究の結果がなかなか出なかった私を最後まで粘り強く指導してくださった椋木雅之指導教官に深く感謝の意を示します。また研究の中で切磋琢磨した研究員にも感謝をいたします。今回の卒業論文では多くのご支援・ご協力のおかげで自分の納得のいく結論を出すことができました。卒業論文制作にあたり、関係してくださった全ての方に厚く御礼を申し上げ、感謝の意を表します。

参考文献

[1] Viola, P., Jones, M.J.,“Robust Real-Time Face Detection.”, International Journal of Computer Vision 57, 137–154 (2004)
[2] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, Olaf Ronneberger, “ 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”, arXiv:1606.06650 [cs.CV](2016)
[3] Olaf Ronneberger, Philipp Fischer, Thomas Brox,“U-Net: Convolutional Networks for Biomedical Image Segmentation”, arXiv:1505.04597 [cs.CV](2015)
[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg,“SSD: Single Shot MultiBox Detector”, arXiv:1512.02325 [cs.CV] (2016)
[5] Karen Simonyan, Andrew Zisserman,“Very Deep Convolutional Networks for Large-Scale Image Recognition”, arXiv:1409.1556 [cs.CV](2015)
[6] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi,“You Only Look Once: Unified, Real-Time Object Detection”, arXiv:1506.02640 [cs.CV](2015)
[7] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla,“SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”, arXiv:1511.00561 [cs.CV](2015)
[8]鈴木雅隆,柴田昌和,周藤安造,“ボリューム・レンダリングの解剖学への応用 ラット腎臓の連続組織切片の 3 次元構築”,MEDICAL IMAGING TECHNOLOGY Vol.13 No.3 May(1995)
[9] Aiham Taleb, Winfried Loetzsch, Noel Danz, Julius Severin, Thomas Gaertner, Benjamin Bergner, Christoph Lippert, “ 3D Self-Supervised Methods for Medical Imaging”, arXiv:2006.03829v3 [cs.CV](2020)
[10] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft,“Simple Online and Realtime Tracking”, arXiv:1602.00763 [cs.CV](2017)
[11] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, Joseph Tighe,“SiamMOT: Siamese multi-object tracking”,CVPR2021(2021)
[12] Nicolai Wojke, Alex Bewley, Dietrich Paulus,“Simple Online and Realtime Tracking with a Deep Association Metric”, arXiv:1703.07402 [cs.CV](2017)