計算工学講演会論文集　Vol.30 (2025年6月)　　日本計算工学会

ハードウェアへの実装に向けた TransUNetの高効率化

Optimizing TransUNet Architecture for Implementing in Hardware

高澤健太[1)]，小林伸彰[2)]
Kenta Takasawa and Nobuaki Kobayashi

1) 日本大学 精密機械工学専攻 修士（〒274-0063 千葉県船橋市習志野台7-24-1, E-mail: cske21060@g.nihon-u.ac.jp)
2) 博(工) 日本大学 准教授（〒274-0063 千葉県船橋市習志野台7-24-1, E-mail: kobayashi.nobuaki@nihon-u.ac.jp）

Research about image recognition for medical care has been popular in recent years. UNet has high performance in medical image segmentation. However, it is limited in its ability to capture long-term characteristics. To solve this problem, TransUNet has been proposed by combining Vision Transformer with UNet. Because processing for TransUNet requires memory and complexity, it is challenging to implement the architecture to limited hardware. In this paper, we propose a modified TransUNet architecture for hardware implementation. In this model, we applied Dynamic-Window to reduce memory and calculation. Moreover, the window is independent so that the model can process more suitably in parallel. In this experiment, we compared the model with the conventional one and evaluated both models. We use the Synapse multi-organ segmentation dataset. As a result, we found that the model was recognized as accurately as the conventional one and reduced used memory size.

Key Words : *machine learning, TransUNet, dynamic window, hardware*

1.　緒言

画像認識モデルの一つとしてTransUNetが注目されている．本モデルは，医療画像の認識に長けたUNetと呼ばれるアーキテクチャに，Vision Transformer（ViT）が組み込まれた．UNetなど従来のCNN（畳み込みニューラルネットワーク）ベースの手法では，局所的な情報の抽出に長けており，大域的な情報を捉えにくい．一方で，ViTは大域的な情報を効率よく捉えることが可能だが，高解像度の画像に適用すると計算量が増大してしまう．そこで，これら2つを組み合わせ，大域的情報と局所的情報をバランス良く考慮できるようにしたものがTransUNetである．

近年，画像認識モデルをハードウェアに実装する研究が盛んである．ハードウェアへの実装により，処理速度の向上や低消費電力化などが見込める．しかし，現状ではTransUNetをハードウェアに実装することに関連した研究は非常に少ない．これは，TransUNetは計算量が多く，そのままハードウェアに実装することが困難であることに起因する．そこで，本研究ではハードウェアへの実装に向けた第１歩として，ハードウェア実装に特化したTransUNetの高効率化に取り組む．実験では，元のモデルと動的ウィンドウを導入したモデルとを比較することにより，本研究で提案するモデルの性能を評価する．

2.　TransUNetについて

(1) 関連する研究

a)　UNet

UNetは，Ronneberger氏らが2015年に提案した，主に医療画像セグメンテーションを目的としたCNNモデルの1種である．このモデルはU字型の構造を持ち，エンコーダとデコーダ部分で構成される．エンコーダ部分は，畳み込み層とプーリング層で構成され，入力された画像から特徴量を抽出する役割を持つ．ここでは，解像度を段階的に落とし特徴マップを小さくしていくことで，大域的な特徴を捉える．デコーダ部分は，アップサンプリングと畳み込み層で構成され，抽出した特徴量を使ってセグメンテーションマップを生成する役割を持つ．この部分では，エンコーダとは逆に解像度を段階的に上げ特徴マップを復元していくことで，局所的な予測を実行する．さらに特徴的な要素として，UNetにはスキップ接続が存在する。これにより，エンコーダの各層からデコーダの層に特徴マップを複製し連結することで，解像度を落とす際にエンコーダで失われた空間情報をデコーダに伝え，セグメンテーション精度を向上させている．図1にUNetの構造を示す．

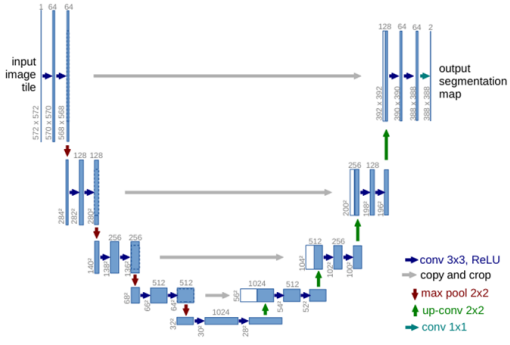

図1 UNetの構造[1]

なお，図1において，青い四角形の上の数字はチャンネル数，横の数字は縦×横のサイズを，白い四角形はスキップ接続により複製された特徴マップを示す．

UNetの長所として，精度の高さや構造の単純さ、モデルの柔軟性などが挙げられる．その一方で，セグメンテーションを行なう上で，大域的な特徴を捉えにくいという弱点がある．

b) Vision Transformer

Vision Transformer（ViT）は，Transformerを画像認識タスク向けに適用したアーキテクチャであり，Google researchのDosovitskiy氏らによって2020年に提案された．ViTはパッチ分割，パッチ埋め込み，位置エンコーディング，Transformerエンコーダ，出力と分類というステップから構成される．始めに，入力画像を固定サイズのパッチに分解し，各パッチをフラット化して線形埋め込みを適用する．次に，分割されたパッチを埋め込みベクトルに変換し，さらに各パッチに線形変換を適用してトークン化する．そこへ，画像全体の特徴を表すのに使うCLSトークンを追加する．続いて，各トークンに位置エンコーディングを加え，これらのトークンをTransformerエンコーダに入力する．そして，エンコーダの出力からCLSトークンを抽出し，それを分類ヘッドに渡し，それをもとに予測を行う．また，Transformerエンコーダは Multi-Head Self -Attention（MHSA），多層パーセプトロン（MLP），正則化（Layer Normalization），残差接続（Residual Connection），で構成される．このうちMHSAは，Self-Attentionの計算を複数回，並列に実行する仕組みである．Transformerベースのアーキテクチャには Self-Attentionという構造が提案されており，これにより画像全体の広範囲の依存関係を効率よく捉えることが可能である．MHSAは，入力トークンをヘッドと呼ばれる集まり複数個に分割し，ヘッドの数だけそれぞれ独立して計算を行う．図2にVision Transformerの構造を示す．

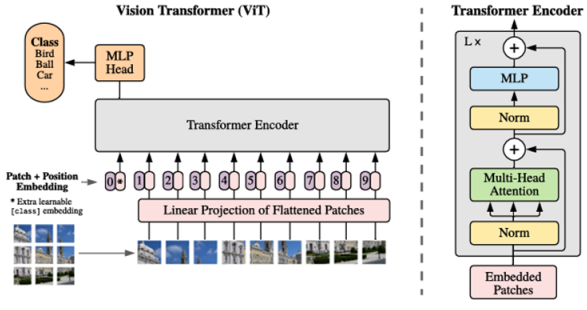

図2 Vision Transformerの構造[2]

なお，図2の右図では正則化をNorm，残差接続を反時計回りの矢印，多層パーセプトロンをMLPで表している．

ViTには，画像全体を一度に処理することが可能，大量のデータを扱う際はCNNを超える性能を発揮するなどの利点がある．一方，少量のデータを扱う場合は性能が低下する，計算コストが高いといった課題もある．

(2) TransUNet

TransUNetとは，UNetのエンコーダ部分にVision Transformer（ViT）を組み込むことで，CNNの課題の克服を目的としたハイブリッドなモデルである．このモデルは，医療画像セグメンテーション用途に向け，Chen氏らに2021年に提案された．一般に，CNNは局所的な特徴を捉えやすい分，大域的な特徴を捉えにくい．また，Transformerベースのアーキテクチャには，入力データが高解像度だと計算コストが膨大になるという課題があり，医療画像解析では高解像度データがよく使われるため不向きだとされる．これらの課題を解決するアプローチとして，TransUNetが提案された．TransUNetでは，ViTの大域的な表現能力とUNetの局所的な特徴抽出能力を融合させることで，セグメンテーション精度の向上に加え，CNNを用いた次元の削減によって，通常のViT以上の効率化が期待できる．

TransUNetは、UNetと同じようにエンコーダとデコーダ、スキップ接続で構成される．このうち，エンコーダ部分はVision Transformerが統合された，ハイブリッドエンコーダである．エンコーダ部分において，まずCNN（ResNet-50）を使用して画像をパッチに分割し特徴ベクトルに変換，初期段階の局所的特徴を抽出する．続いて，Vision Transformerでパッチを受け取り，Attention機構（Multi-Head Self-Attention）によってパッチ間の関係をモデル化し，大域的情報を捉える．そして，デコーダ部分で，先ほど抽出した特徴マップを段階的にアップサンプリングして解像度を上げていき，最終層でセグメンテーションマップを出力する．また，UNetと同様にスキップ接続が存在し，エンコーダにおける中間特徴をデコーダに直接渡す機能を持つ．そのため，CNNで抽出した局所的な特徴をデコーダ部分に引き継ぐことが可能である．これにより，局所的な特徴とVision Transformerで捉えた大域的な特徴をバランス良く組み合わせることが可能となる．図3にTransUNetの構造を示す．

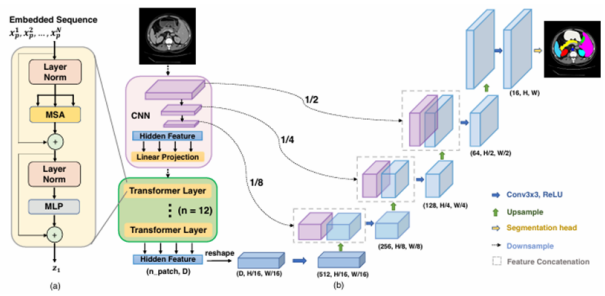

図3 TransUNetの構造[3]

図3（a）において，Layer Normは正則化，MSAはMulti-Head Self-Attention，MLPは多層パーセプトロン，反時計回りの矢印は残差接続を示す．図3（b）において，Linear projectionは線形変換，Hidden Featureはモデル内で処理中

の特徴，CNNから伸びている矢印はスキップ接続，紫の四角形は複製された中間特徴，括弧内の数字はチャンネル数，縦のサイズ，横のサイズを示す．また，図3の全体から分かるようにTransUNetをU字型の構造を持っている．

3. Attentionの効率化

(1) 関連する研究

a) Attention 機構

Transformer ベースのアーキテクチャを語る上で欠かせないのが，Attention 機構の存在であり，最も基本的なのが Self-attention である．Attention 機構においては，アテンションスコアによって、トークン間の関連性をモデル化することができる．アテンションスコア Attention(Q,K,V)は，クエリQ，キーK，バリューV，埋め込み次元d_kを用い，式(1)のように表すことができる[4].

$$Attention(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

なお，クエリは「注目すべきポイント」，キーは「関連性を計算する際の指標」，バリューは「実際の出力」を表している．また，Q^Tは「Qの転置」を意味する．Self-Attentionを始めとするAttention機構は，アテンションスコアの算出により，入力データのすべての位置情報を他の位置情報と関連付けられるため，広範囲の依存関係を効率的に捉えることを得意としている．Self-Attentionの応用として，Vision Transformer や TransUNet では Multi-Head Self-Attentionが使用されている．これは，入力トークン間の関係性を学習するための仕組みであり，ヘッドの数だけ並列に実行することができる．なお，ヘッドとは入力トークンを分割した際の，トークンのまとまりのことである．Multi-Head Self-Attentionの出力MHA(Q,K,V)は，式(2)のように表せる[4].

$$\mathrm{MHA}(Q,K,V) = \mathrm{Contact}(head_1, head_2, \cdots, head_h)W_0 \quad (2)$$

式(2)において，W_0は学習可能な重み，hはヘッドの個数を表す．Multi-Head Self-Attentionでは，各ヘッドが小さな次元でアテンションスコアを算出するため，計算時の負荷をある程度分散することができる．

b) Sliding Window Attention

Sliding Window Attentionは，Self-Attentionと比較して，さらに計算を効率的に行えるようにするため考案されたアプローチである．Transformerベースのアーキテクチャの基板となっているSelf-Attentionは，CNNと比較して広範囲における依存関係の考慮に優れるという特徴がある．その一方で，計算コストやメモリ消費が大きくなりやすく，それがハードウェアへの実装を阻む一因となっていると考えられる．この課題を解決する要素の1つとして，Sliding Window Attentionに注目した．

Sliding Window Attentionでは，「ウィンドウ」という一定の大きさを持った枠を用いて，計算する範囲を局所的なものに制限することで計算量を減らしている．また，ウィンドウは画像やシーケンスなどの全体を覆うようにスライドしていく．Vision Transformerでは各トークンが画像全体のトークンと関連付けられるのに対し，このアプローチではウィンドウ内のトークンのみと関連付けられるため計算量が削減される．Self-Attentionにおいて，トークン数をNとすると，その計算量は$O(N^2)$と表現することができる．そのため，データの規模が大きくなる(トークン数Nが大きくなる)と計算量も増大し，それに伴いメモリの消費量も増えるという課題がある．一方で，Sliding Window Attentionでは，ウィンドウの大きさをwとすると，その計算量は$O(N \cdot w)$で表すことができ，wを適切に設定することで，メモリ消費量の削減や計算効率の向上を見込める．特に，Sliding Window Attentionは高解像度画像や長いシーケンスで効果を発揮するため，高解像度画像を用いることが多い医療画像解析では有効だと考えられる．なお，このアプローチでは計算の範囲を制限することによって効率化を図るため，メモリ消費量が減る一方でモデルのパラメータ数が変わらない特徴がある．Sliding Window Attentionの代表的な応用例として，Liu氏らが提案したSwin Transformerが挙げられる．今回のアプローチでのウィンドウの使い方は，Swin Transformerのそれに近い．図4にSwin TransformerとVision Transformer (ViT)の入力画像に対する計算範囲の比較を示す．

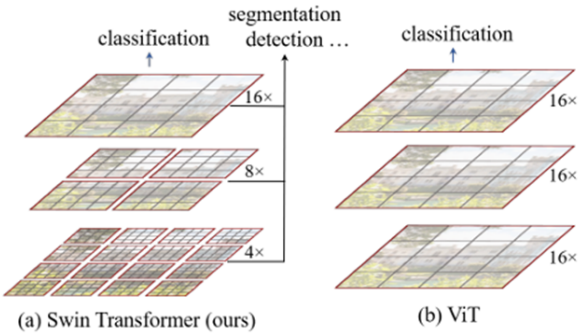

図4 計算範囲の比較[5]

図4を見ると，Swin Transformer(a)では画像を細かく分割し，赤枠の内部で計算を行うため，計算量が削減されていることがわかる．この時，ウィンドウサイズにより図4(a)における分割数が変化する．一方で，Vision Transformer(b)では大きなサイズの画像全体をそのまま計算しており，計算量も多くなっている．今回の実験で導入したSliding Window Attentionは，図4(a)の手法に近い．

c) 動的ウィンドウ

Sliding Window Attentionなどのウィンドウを用いたアプローチでは，通常はウィンドウの大きさは固定されている．ウィンドウサイズが固定の場合，画像内の性質の異なる領域が均等に処理される．これにより計算コストが無駄に増加し，さらに重要な情報を逃してしまう可能性がある．加えて，動的な依存関係を捉えるのが不得意という課題もある．それらを解決するためのアプローチの１

つが，動的ウィンドウである．

動的ウィンドウは，ウィンドウの大きさや形状，計算対象を，入力されたデータや使用されるタスクに応じて適切に調整または選択するというアプローチである[6]．画像タスクでは，重要度の低い領域ではウィンドウを大きくし，重要度の高い領域ではウィンドウを小さくするなどの調整が可能になり，局所的な情報や広範囲の依存関係を，効率良く取得することができる．また，ウィンドウを適切な大きさにすることで，計算コストや消費電力などを抑えることが可能となる[6]．例えば，小さすぎるウィンドウではウィンドウ間の重複が増加し無駄な計算が生じ，大きすぎるウィンドウではトークン数が増加しメモリ消費が増えるが，動的ウィンドウの活用によりこれらを解消できる．特に，画像セグメンテーションでは，入力されるデータ(画像)の構造が不均一なことが多いため，比較的相性が良いと考えられる．なお，ウィンドウの大きさや形状などが変動する際には計算が行われる．そのため，動的ウィンドウを導入したアーキテクチャをハードウェアに実装する場合は，並列処理に対応させるなどの工夫によって，ウィンドウの変化による計算コストの増加を防ぐ必要がある．

(2) 動的ウィンドウ搭載型TransUNet

本研究では，動的ウィンドウを採用することで効率化を実現した，ハードウェア実装向きのTransUNetを提案する．これらのアーキテクチャでは，Attention機構の内部にSliding Window Attentionと動的ウィンドウを基にしたアプローチを導入し，精度を大きく落とさずに計算量とメモリの消費量の削減を実現している．このアプローチでは，ウィンドウサイズとして，「事前に設定された値」と「シーケンス長の半分の値」の2種類が用意されている．事前に設定された値が入力された画像のシーケンス長の半分の値よりも小さければ「事前に設定された値」が，大きければ「シーケンス長の半分の値」がウィンドウのサイズとして設定される．この工程によって，ウィンドウを適切なサイズに調整し，余分な分割を防ぐことができる．そして，入力されたシーケンスをウィンドウサイズに基づき分割することでAttentionの計算がウィンドウ内に限定され計算量が削減される．sをシーケンス長，wをウィンドウサイズとして図5に提案した動的ウィンドウを示す．

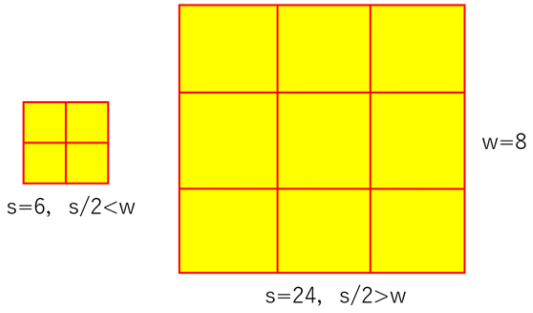

図5 今回用いた動的ウィンドウ

図5では，左図のシーケンス長を6，右図のシーケンス長を24，事前に設定されたウィンドウサイズを8と仮定している．右図では入力画像のシーケンス長の半分の値がウィンドウサイズよりも大きいので，事前に設定した値のまま分割されている．一方で，左図では入力画像のシーケンス長の半分の値がウィンドウサイズよりも小さいので，ウィンドウサイズをシーケンス長の半分の値に設定した後に分割されている．また，今回提案したアプローチには，計算は各ウィンドウ内の情報のみに依存しているという特徴がある．複数のウィンドウ間での情報共有を行っておらず，ウィンドウごとの計算が互いに影響を与えるような設計もされていない．したがって，各ウィンドウの独立性が高くなっているため，ハードウェアでの並列処理により適していると考えられる．

4. 実験方法

以下に実験手順を示す．また，今回の実験はmax_epoch＝300，batch_size＝24で行った．また，提案したモデルの事前学習モデルが完成していないため，実験で用いたものは，どちらも事前学習なしとした．

1. データセットを用いてモデルの学習を行う．
2. パラメータ数，演算回数，メモリの消費量を記録する．
3. 学習したモデルを用いて推論を行う．
4. 手順 2 の要素に加え，推論精度を記録する．
5. 手順 1~4 までを各モデルでそれぞれ 5 回繰り返す．
6. 各モデルのデータごとの平均を求め比較する．

5. 実験結果と評価

以下に，モデルの学習時と推論時の実験結果を示す．なお，結果では既存のモデルをTransUNet，動的ウィンドウを導入した後の改良モデルをDW_TransUNet(DW = Dynamic Window)として記載した．(1)では学習時，(2)では推論時における測定結果とその平均をそれぞれ示した．

実験結果では，estimated_memoryはモデル全体で必要と予想されるメモリ量，allocated_memoryは実際に使用中のメモリ量、cached_memoryは計算のために確保されているメモリの量を示す．また，表5と表6において，DSC(Dice Similarity Score)は数値が大きいほど，mean_hd(Mean Hausdorff Distance)は数値が小さいほど評価が高い．DSCは対象の内側の領域を，mean_hdは対象の外側の境界線をどれだけ正確に認識しているかを示したものである．

(1) 学習の結果

学習時のパラメータ数

TransUNet：105,277,081[個]

DW_TransUNet：105,277,081[個]

学習時の積和演算回数

TransUNet：286.95[106回]

DW_TransUNet：286.95[回]

学習時に必要になるメモリの見込み

TransUNet：10,427.17[MB]

DW_TransUNet：10,427.17[MB]

表1 モデル学習時のメモリ使用量

	学習時allocated_memory[MB]					
モデル名	1回目	2回目	3回目	4回目	5回目	**平均**
TransUNet	1295.76	1295.76	1295.86	1295.76	1295.76	**1295.78**
DW_TransUNet	1294.92	1294.92	1294.92	1294.92	1294.92	**1294.92**

表2 モデル学習時のメモリ占有量

	学習時cached_memory[MB]					
モデル名	1回目	2回目	3回目	4回目	5回目	**平均**
TransUNet	9711.70	9709.87	9711.30	9711.35	9711.58	**9711.16**
DW_TransUNet	9036.53	9036.53	9035.95	9035.95	9037.05	**9036.40**

表1と表2より，動的ウィンドウをTransUNetに導入した前後で，学習時のallocated_memoryが約0.066%，cached_memoryが約7.0%削減されたことがわかる．allocated_memoryは計算効率に直結し，cached_memoryの増大は計算速度の低下や消費電力の増加を引き起こす可能性があるので，ハードウェアへの実装を目的とする本研究において，これらの結果は有効だと考えられる．

(2) 推論の結果

推論時のパラメータ数

TransUNet：105,277,081[個]

DW_TransUNet：105,277,081[個]

推論時の積和演算回数

TransUNet：11.96[回]

DW_TransUNet：11.96[回]

推論時に必要になるメモリの見込み

TransUNet：837.85[MB]

DW_TransUNet：837.85[MB]

表3 モデル推論時のメモリ使用量

	推論時allocated_memory[MB]					
モデル名	1回目	2回目	3回目	4回目	5回目	**平均**
TransUNet	409.75	409.75	409.75	409.75	409.75	**409.75**
DW_TransUNet	409.75	409.75	409.75	409.75	409.75	**409.75**

表4 モデル推論時のメモリ占有量

	推論時cached_memory[MB]					
モデル名	1回目	2回目	3回目	4回目	5回目	**平均**
TransUNet	1014.00	1014.00	1014.00	1014.00	1014.00	**1014.00**
DW_TransUNet	1014.00	1014.00	1014.00	1014.00	1014.00	**1014.00**

一方で，表3と表4より，推論時のallocated_memoryとcached_memoryは動的ウィンドウの導入前後で変化が無かったので，こちらも削減できるアプローチを模索する必要があると考えられる．

表5 モデルのDSC

	推論精度[%]					
モデル名	1回目	2回目	3回目	4回目	5回目	**平均**
TransUNet	76.29	76.69	76.30	76.07	76.92	**76.45**
DW_TransUNet	76.21	77.04	76.01	77.09	76.85	**76.64**

表6 モデルのmean_hd

	推論精度					
モデル名	1回目	2回目	3回目	4回目	5回目	**平均**
TransUNet	23.14	22.97	26.03	26.32	25.36	**24.77**
DW_TransUNet	29.02	30.70	30.57	28.41	27.44	**29.23**

表5と表6より，動的ウィンドウの導入前後でDSCは約0.2%向上し，mean_hdは約5%増大した．DSCが向上しているため，対象の領域を捉える性能は元のTransUNetと同等以上であると考えられる．しかし，mean_hdが増大しているので，元のTransUNetと比較して境界付近の誤差が大きくなっており，対象の正確な形状を捉えられていないと考えられる．モデルを医療分野に活用する場合，境界付近の正確さが重要になるため，より正確に対象の境界を捉え，mean_hdを低減するべきだと考えられる．

課題として，モデルの性能向上とハードウェア実装が挙げられる．前者に関しては，メモリ消費量のさらなる削減とmean_hdの低減に取り組んでいく．メモリ消費量を削減するアプローチとしては，固定小数点の採用が挙げられる．このモデルでは浮動小数点を用いているので，固定小数点を導入することで，学習時に加え推論時のメモリ消費量を削減できる可能性がある．mean_hdの低減に関しては，損失関数の変更が挙げられる．モデルに適したものを選ぶことで，境界の誤差を小さくすることができると考えられる．後者に関しては，実装に用いるハードウェアを容量などの要素を考慮しながら選択，設計する必要がある．モデル全体を一度にハードウェアへ実装するのは難易度が高いと考えられるため，最初はモデルの一部をハードウェアに実装することを目指し，徐々に実装する範囲を広げていく予定である．

6. 結言

本研究では，ハードウェアへの実装に向けたTransUNetの高効率化のために動的ウィンドウを導入し，導入前のモデルとの比較を行った．その結果，学習時のメモリ消費量の削減に加え，推論時のDSCの僅かな向上が見られた．よって，精度の低下を抑えてメモリの消費量を削減するという目的は，概ね達成することができたと言える．

参考文献

[1] Olaf ronneberger et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation” (2015)
[2] Alexey Dosovitskiy et al. “AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE” (2020)
[3] Jieneng Chen et al. “TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation” (2021)
[4] Ashish Vaswani et al. “Attention Is All You Need” (2017)
[5] Ze Liu et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows” (2021)
[6] Yulin Wang et al. “Not All Images are Worth 16x16 Words: Dynamic Transformers for Efficient Image Recognition” (2021)