

ここに題目を書く

You write your title here in English

自分野名前

Namae JIBUNNO

(20xx 年度入学, 00000000)



指導教員: 清水郁子 教授

東京農工大学 工学部 知能情報システム工学科

20xx 年度卒業論文

(20xx 年 x 月 x 日 提出)

東京農工大学 工学部 知能情報システム工学科 20xx 年度 卒業論文 要旨

題目 ここに題目を書く

You write your title here in English

If your title is long, you write it here too.

学籍番号 00000000 氏名 自分野名前 (Namae JIBUNNO)

提出日 20xx 年 x 月 x 日

ここに概要を書く。2000 文字くらいで 1 ページ分。概要には「背景」、「先行研究」、「目的」、「提案手法の内容」、「結果」を書く。図は含めず、文章のみで説明する。

目次

第 1 章	はじめに	1
1.1	研究背景	1
1.2	本論文の構成	6
第 2 章	先行研究	7
2.1	Vision Transformer	7
第 3 章	提案手法	9
3.1	提案手法の概要	9
3.2	入出力	9
3.3	ネットワーク構造	9
3.4	損失関数	9
3.5	データセット	9
第 4 章	実験	10
4.1	予備実験	10
4.2	実験	10
第 5 章	おわりに	12
5.1	まとめ	12
5.2	今後の展望	12
付録 A	補足	13
参考文献		14

図目次

1.1	研究室の様子	4
1.2	研究室リフレッシュコーナーの様子	4
2.1	Vision Transformer の構成	7
4.1	損失	11
4.2	入力画像と出力の様子	11

表目次

1.1	研究室メンバーの内訳	3
1.2	研究室メンバーの内訳	4
2.1	ViT(BERT) のネットワーク規模	8
4.1	実験環境	10

はじめに

1.1 研究背景

ここには研究の背景を書くが、せっかくなので \LaTeX の基本的な使い方について説明する。実際に論文を提出するときは、この部分はすべて消去してあることを確認してほしい。

1.1.0 コンパイルできないときは

ここでは、Docker コンテナには正しく入れていると仮定して、 \LaTeX の問題でコンパイルできないとき、コンパイルしたのに pdf に反映されていない場合を考える。その場合、以下の原因が考えられる。

- 変更を保存していない。デフォルトでは自動保存されないため、ファイルを変更した際には Ctrl+S で上書き保存を行う。変更したすべてのファイルで行うこと。
- 図や表が無い。指定したファイルが無い場合、コンパイルエラーが発生する。ファイルが存在するか、ファイル名(拡張子を含む)が間違っていないかを確認する。
- その他エラーが発生している。画面下部の「PROBLEMS」を確認し、対処する。また、まれに「PROBLEMS」には詳細なエラー内容が表示されないことがある。その場合は隣の「OUTPUT」を確認する。

1.1.1 改行

改行したい場合は、ただ改行するのではなく、一行空ける。日本語の文章の場合は、段落の最初の 1 文字分が字下げされる。

もし字下げせずに改行したい場合は、バックスラッシュ(円マーク)をふたつ挿入する。

空行を入れたい場合は行送りのコマンドを使う。

改行や空白も含めてそのまま出力させたい場合には、 verbatim*を使う。第 1.1.2 項で説明する特殊文字もそのまま出力できる。

verbatim* ブロック内では、改行や空白がそのまま出力される
改行をいくつ入れても

反映

記号も出力 \verb*! " \$" % \$ & |

1.1.2 文字の大きさ・色・フォント・特殊文字

部分的に文字サイズを変更したい場合は、小さい字や大きい字のように使う。あまり使わない。

また、文字色を変更したい場合は、定義されている色はシアンの文字、定義されていない色は青色の文字のように使う。

文字のフォントを変更したい場合は、Roman, TypeWriter のように使う。

% (percent), { (left brace), } (right brace), & (ampersand), #(hash), \$(dollar), ^(caret), ~(tilde), \(backslash), -(underscore) は特殊文字のため、そのまま入力できないことに注意する。

1.1.3 ラベルと参照

章(chaper), 節(section), 項(subsection), 目(subsubsection)には自動的に番号がつけられる。別の章を引用する場合は、第2章、第3章、・・・と直接書くのではなく、ラベルを使って第1章、第2章、・・・と引用する。ラベルを使う場合は、一般には2回コンパイル(緑の矢印を押して build すること)しないと正しく反映されないので注意する。

また、参考文献を引用したい場合は、references.bibに引用元が記述されていることを確認し、次[1]のように引用する。

1.1.4 箇条書き

番号付き箇条書きをする場合には、以下のように使う。

1. ひとつめ
 2. ふたつめ
 3. みつめ
 - (a) 入れ子のひとつめ
 - (b) 入れ子のふたつめ
- 11 見出しを変えることもできる
4. その他の項目は見出しを変える前のものに依存する

番号なし箇条書きをする場合には、以下のように使う。

- ひとつめ
- ふたつめ
 - 入れ子のひとつめ

1.1.5 数式の記述

数式を挿入する場合、インライン数式モード（文章中に数式を埋め込むモード）では $y = ax^2 + bx + c$ のように数式をドルマークで囲むことで挿入できる。文章中で変数の定義（ここで、 $0 < x < 5$ である。とか）をする場合によく用いられる。

ディスプレイ数式モード（数式をセンタリングして挿入するモード）ではいくつかの方法で挿入できる。単純な挿入の場合、

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

のようにドルマーク 2 つで囲むことで挿入できる。

しかし、可読性や余白、等号の位置などを踏まえて、一般には align 環境を用いる。式変形を書き下す場合は、列を揃えてほしい記号の前にアンパサンドを入れる。

$$T = T_1 + T_2 = \frac{1}{2}m_1(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}m_2(\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2}J_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (1.1)$$

$$U = U_1 + U_2 = 0 \quad (1.2)$$

$$F = F_1 + F_2 = \frac{1}{2}\mu_1\dot{\theta}_1^2 + \frac{1}{2}\mu_2\dot{\theta}_2^2 \quad (1.3)$$

式が長すぎて 1 行に収まらない場合は、nonumber と quad を用いる。

$$\begin{aligned} T &= \frac{1}{2}m_1(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}m_2(\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2}J_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ &= \frac{1}{2}m_1\left\{(-l_1\dot{\theta}_1 \sin \theta_1)^2 + (l_1\dot{\theta}_1 \cos \theta_1)^2\right\} + \frac{1}{2}J_1\dot{\theta}_1^2 \\ &\quad + \frac{1}{2}m_2\left[\left\{-L_1\dot{\theta}_1 \sin \theta_1 - l_2(\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2)\right\}^2 + \left\{L_1\dot{\theta}_1 \cos \theta_1 + l_2(\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2)\right\}^2\right] \\ &\quad + \frac{1}{2}J_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \end{aligned} \quad (1.4)$$

$$= \frac{1}{2}(m_1l_1^2\dot{\theta}_1^2 + J_1\dot{\theta}_1^2) + \frac{1}{2}\left[m_2\left\{L_1^2\dot{\theta}_1^2 + l_2^2(\dot{\theta}_1 + \dot{\theta}_2)^2 + 2L_1l_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) \cos \theta_2\right\} + J_2(\dot{\theta}_1 + \dot{\theta}_2)^2\right] \quad (1.5)$$

1.1.6 図表の挿入

図表を挿入する場合、figure フォルダと table フォルダのなかに図表を格納しておく。ただし、表はこのファイル内に直接記述することができるため、好きな方法を選んでほしい。

表を挿入する場合は、表 1.1 のように使う。

	Docter, etc.		Master		Bachelor
Post	Asst. Prof.	D1	M2	M1	B4
Number of people	1	2	4	3	6

表 1.1: 研究室メンバーの内訳

直接記述することもできる。

	Docter, etc.		Master		Bachelor
Post	Asst. Prof.	D1	M2	M1	B4
Number of people	1	2	4	3	6

表 1.2: 研究室メンバーの内訳



図 1.1: 研究室の様子

図を挿入する場合は、図 1.1 のように使う。

図や表を並べて挿入する場合は、minipage 環境を使う。また、[htbp] のところを [H] にすることで、強制的にここ (=here) に図を挿入できる。



(a) ゲーム



(b) ソファー

図 1.2: 研究室リフレッシュコーナーの様子

章番号と同様に、ラベルを使うことで図 1.1、表 1.1 のように引用できる。ラベルは fig:fig1, fig:fig2, … のように通し番号にするのではなく、図そのものを説明するような名づけをする。

1.1.7 プログラムの挿入

実際のプログラムの一部を(タブなども含めて)挿入したい場合は、リスト 1.1 のように挿入する。

プログラム 1.1: C 言語で”Hello world!”を出力するプログラム

```
1 #include<stdio.h>
```

```
2     int main() {
3         printf("Hello world!");
4     }
```

1.1.8改ページ

改ページしたい場合は、`clearpage` を使う。

1.2 本論文の構成

ここに論文の構成を書く。

第2章では, ···

第3章では, ···

第4章では, ···

第5章では, ···

先行研究

本章では、先行研究で広く用いられる Vision Transformer(ViT)について示したのち、それを用いたいくつかの先行研究を示す。

2.1 Vision Transformer

研究の背景でもある Vision Transformer (ViT) [3] は、画像のクラス分類問題解決のために実装されたモデルである。モデルの設計は図 2.1 に示すとおりであり、Transformer Encoder [3] に BERT [2] を適用した“Transformer”に加えて、画像のトークン化を行う“Tokenizer”を構成要素に持つ。

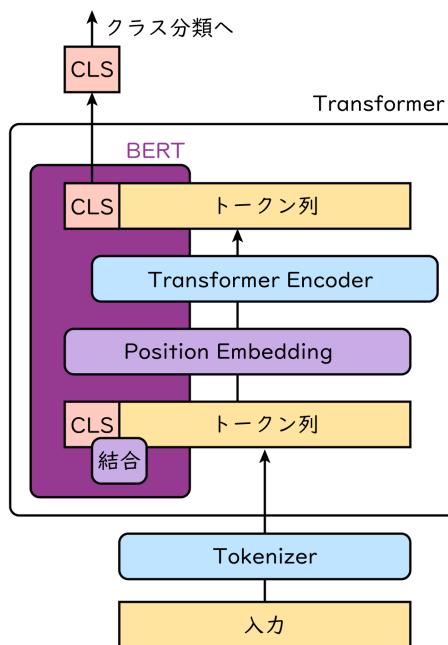


図 2.1: Vision Transformer の構成

中略

2.1.1BERT

BERT [2] とは Bidirectional Encoder Representations from Transformer の短縮表現であり、自然言語処理領域にて発表されたモデルである。ViT と同様、Transformer Encoder を基にしたモデルであり、ViT の設計においても、BERT の要素技術が反映されている。

まず、ViT モデルのネットワーク規模は BERT の表現を踏まえたものとなっている。具体的には、トークンの次元と Encoder Block の数、次節で述べる MSA のヘッド数が表 2.1 のとおり定められている。ViT では Base(B) のネットワーク規模であることを、ViT-B と表記することで明示している。

表 2.1: ViT(BERT) のネットワーク規模

ネットワーク規模	ViT の表記	トークンの次元 (d_t)	Encoder Block の数	MSA のヘッド数
Base(B)	ViT-B	768	12	12
Large(L)	ViT-L	1024	24	16

中略

2.1.2計算量

トークン数 l とその次元 d_t による影響を明確にするため、 $T(d_{vec})$ は変数を用いた表現へと改める。内積における計算処理が、要素積に対して総和を求めていることを考慮すると時間計算量 $T(d_{vec})$ は d_{vec} の比例関数とみなせるため、 $\alpha d_{vec} + \beta$ によって近似する。加えて、ViT-B16 においては、 $d_t = hd_m = 768$ かつ $l = 256$ であるため、 ld_t および l^2h は ld_t^2 や l^2d_t に対して、十分小さいものとする。以上から、全体の時間計算量 T_{msa} は式 (2.1) のとおり推定する。時間計算量の推定値において、トークン数 l による時間計算量のオーダーは $O(n^2)$ であり、トークン数の増加は実行時間に大きく影響することが分かる。

$$\begin{aligned} T_{msa} &= 3ld_t(\alpha d_t + \beta) + ld_t(\alpha d_t + \beta) + l^2h(\alpha d_m + \beta) + ld_t(\alpha l + \beta) \\ &= 2ld_t(2d_t + l)\alpha \end{aligned} \tag{2.1}$$

提案手法

3.1 提案手法の概要

3.2 入出力

3.2.1 入力

3.2.2 出力

3.3 ネットワーク構造

3.3.1 エンコーダ

3.3.2 デコーダ

3.4 損失関数

3.5 データセット

実験

ここでは、いくつかの予備実験の結果を示したのちに、提案手法を先行研究と比較した結果を示し、考察を行う。

4.1 予備実験

提案手法の妥当性を確認するため、2点の改良点をもとにアプリケーション実験を実施した。

4.1.1 データセットによる比較

4.1.2 バッチサイズによる比較

4.1.3 予備実験の考察

4.2 実験

ここでは、先行研究として～～の手法 [3] および～～の手法 [4] を取り上げ、提案手法との比較及び考察を実施する。実験環境を表 4.1 に示す。

CPU	Intel core i7 2.8[GHz]
OS	Windows 10 Pro
メモリ	32[GB]

表 4.1: 実験環境

損失をグラフ化したものを図 4.1 に示す。図 4.1 より、～～

また、図 4.2 に 2 種類のデータについて入力画像と出力画像を並べた結果を示す。図 4.2a では～～であるのに対して、図 4.2b では～～



図 4.1: 損失



(a) 入力画像 (1) と出力の様子



(b) 入力画像 (2) と出力の様子

図 4.2: 入力画像と出力の様子

4.2.1 考察

おわりに

| 5.1まとめ

| 5.2今後の展望

補足

参考文献

- [1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [2] Daniel Ugarte. Curling and closure of graphitic networks under electron-beam irradiation. *Nature*, Vol. 359, No. 6397, pp. 707–709, 1992.
- [3] Leehter Yao, Yeong-Wei Andy Wu, Lei Yao, and Zhe Zheng Liao. An integrated imu and uwb sensor based indoor positioning system. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8. IEEE, 2017.
- [4] 野村篤史, 須ヶ崎聖人, 坪内孝太, 西尾信彦, 下坂正倫ほか. UWB の測定距離と直接波の減衰度を利用したデバイスフリー複数人屋内測位. 研究報告ユビキタスコンピューティングシステム (UBI), Vol. 2022, No. 1, pp. 1–8, 2022.