Vision Transformer

Vision Transformerは言語処理用に開発されたTransformer を画像分類タスクに応用したモデルです。

■ 画像特徴量の入力方法･･･ 画像の”トークン”系列化

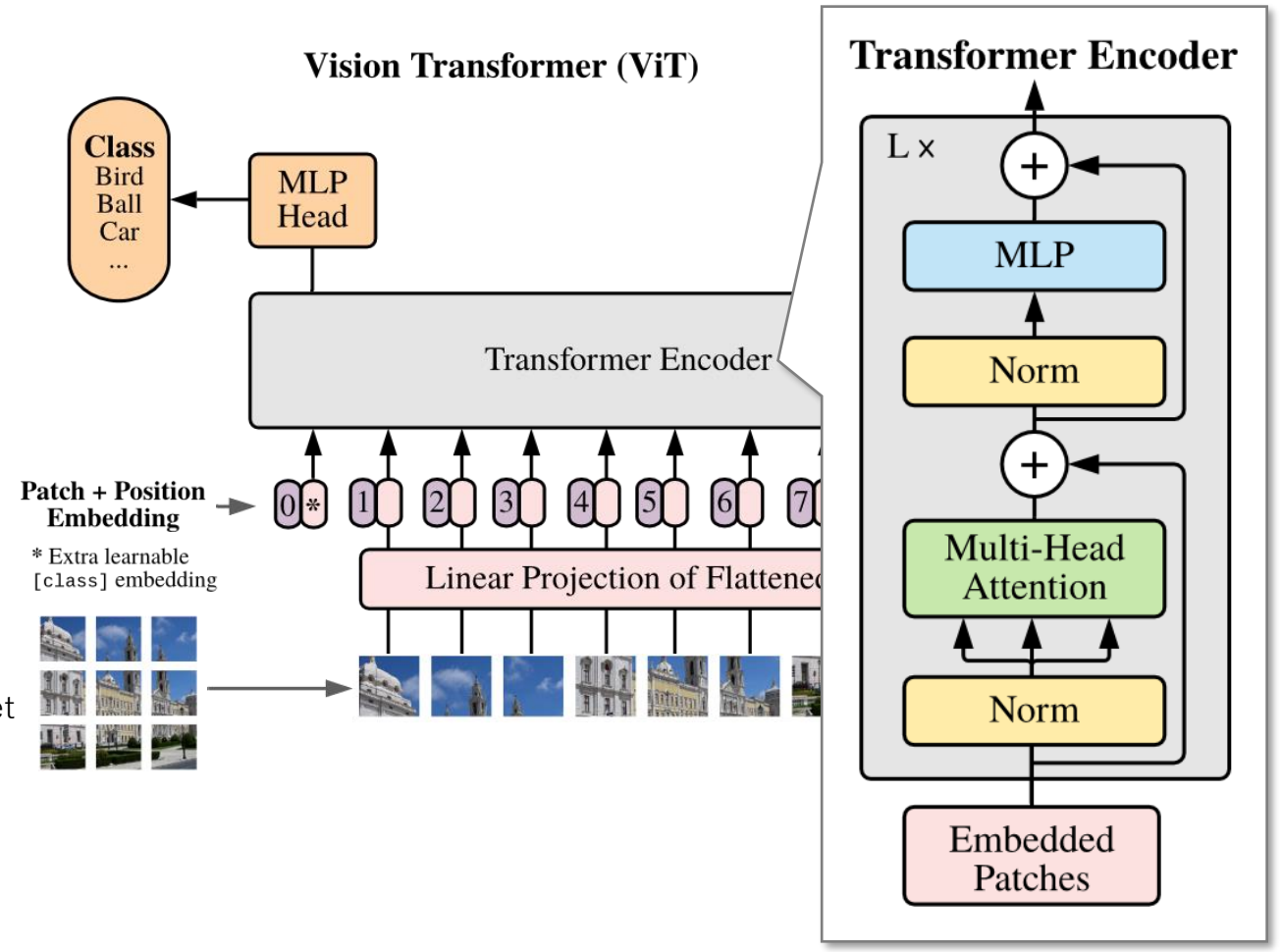
1. 画像をパッチに分割し系列化 ･･･ N個の画像パッチで構成される系列
2. パッチごとにFlatten化 ･･･ “トークン(単語)”化 → これを入力値に使用

■ ViTのアーキテクチャ ･･･ Transformer Encoderの使用

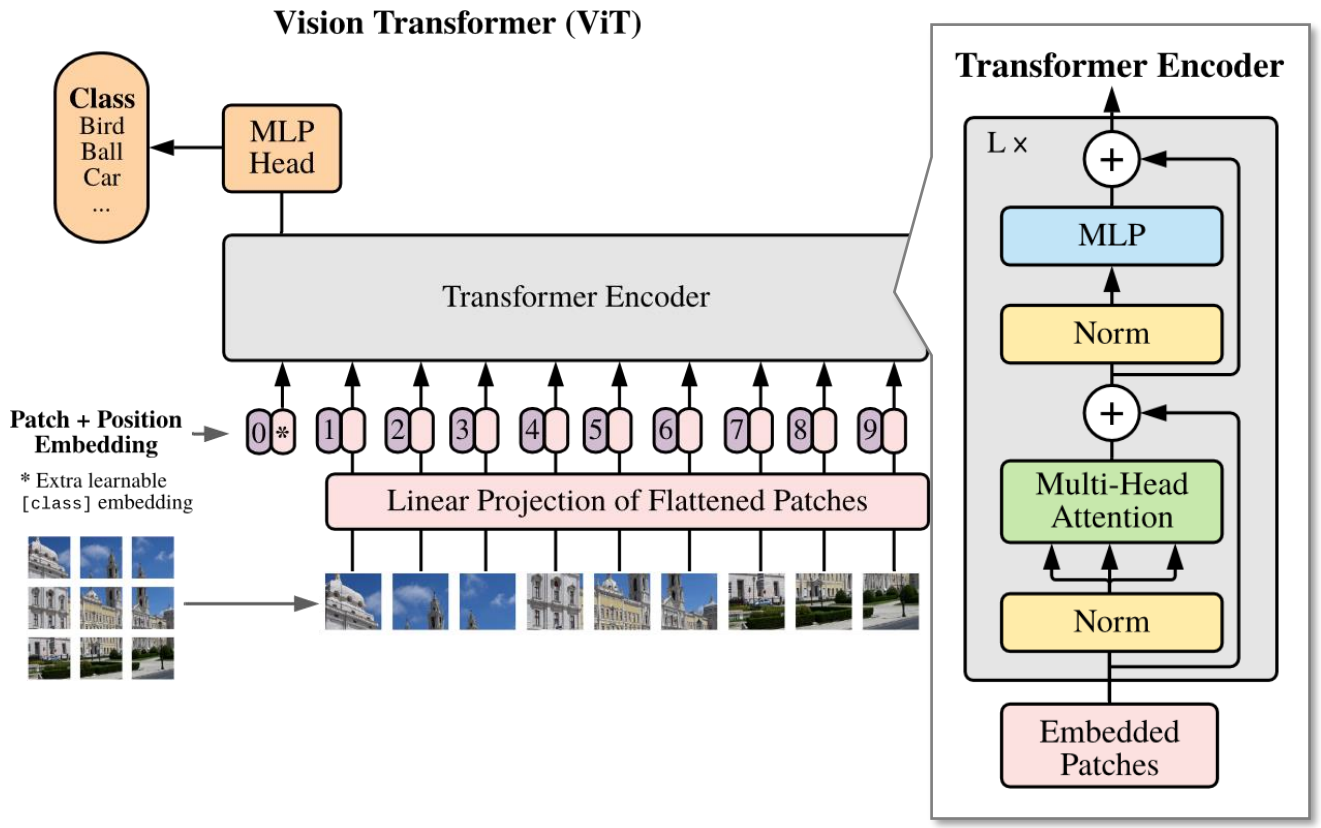
- Transformer Encoderへの入力値の準備
  1. 画像データから計算したEmbedding表現(埋込表現)を計算
  2. 系列の最初に[CLS] Tokenという特別な系列値を付加
  3. パッチの位置関係を示すPosition Embeddingの付加
- Transformer Encoder ･･･ 言語処理向けのオリジナルと同等の構造
- MLP Head ･･･ [CLS] Token系列値の出力特徴量から分類結果を出力

■ 事前学習とファインチューニング

1. 大規模なデータセットでの事前学習
   - ➢ 実験されたデータセット JFT-300M > ImageNet-21k > ImageNet
2. ファインチューニング ･･･ 事前学習より高解像度な画像を入力
   - ➢ 目的タスクに応じたMLP Headの変更
   - ➢ Position Embeddingの差し替え



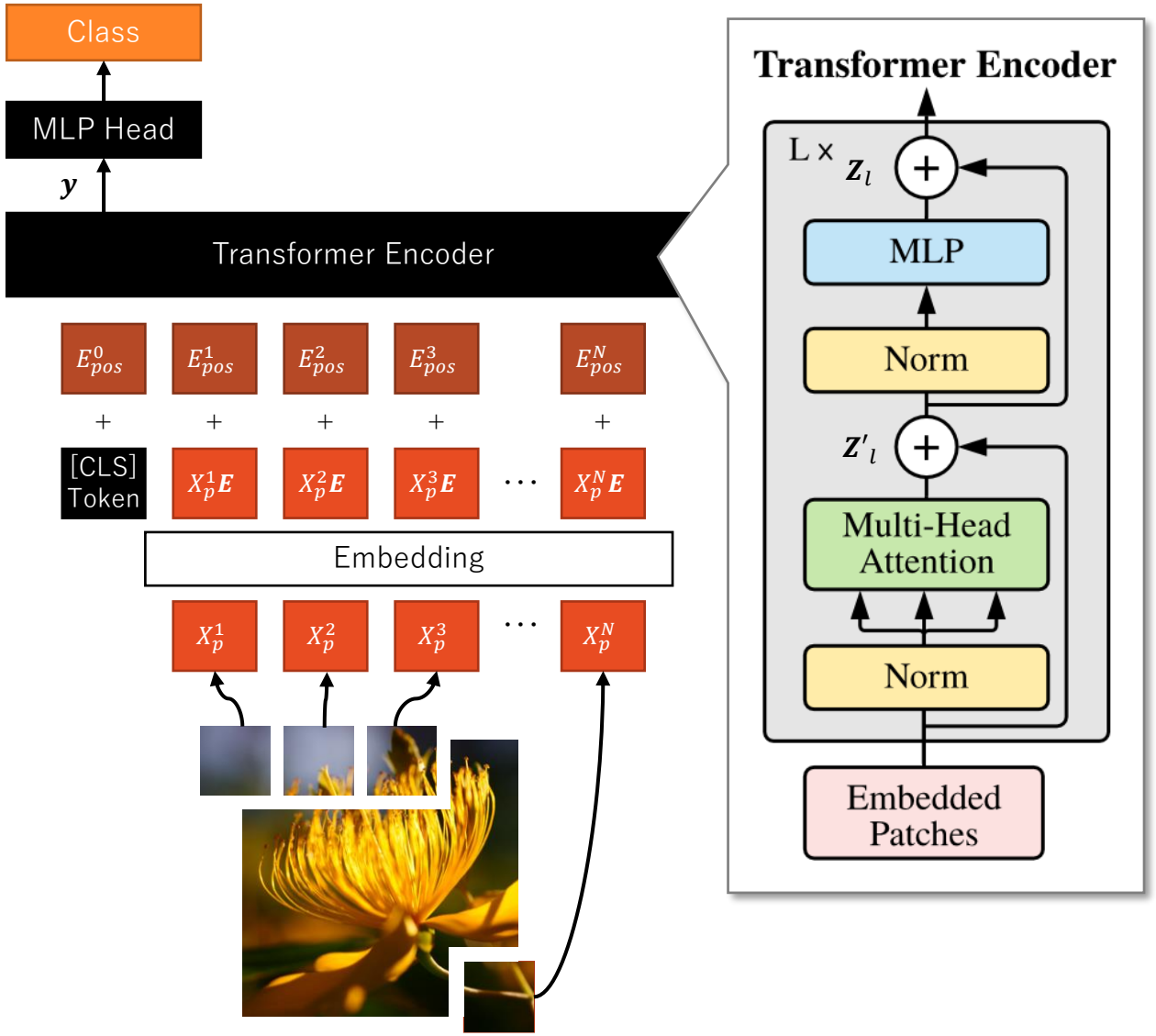
ViTでは、画像をパッチに分割し、系列データとして利用できるよう加工し、特徴量として使用します。



■ ViTにおける画像特徴量の処理

1. 入力画像をパッチに分割する
2. パッチごとにFlatten処理を行い、”トークン”系列得る
3. Embedding表現(埋込表現)に変換する
   - Inductive bias: Embedding表現に線形変換を使用
   - Hybrid Architecture: Embedding表現にCNNを使用
4. CLS Tokenを系列データの最初に付加する
   - Transformer Encoderの出力で、このトークンに対する出力をMLP Headで利用し、分類結果を得る
5. Position Embedding(パッチの位置)を付加する
   - この情報はパラメータであり、学習により自動獲得される

ViTにおける計算処理モデルは下記です。



■ ViTの計算過程

1. 画像 $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ (高さH 幅W チャンネル C)を、縦横がPのパッチで分割
   分割した画像は $\boldsymbol{X_p} \in \mathbb{R}^{N \times (P^2 \cdot C)}$ ($N = HW/P^2$はパッチ数)
   ※ NがTransformerの入力における系列数になる
2. パッチ画像をD次元の特徴量Z_0に変換しTransformer Encoderに入力
   $\boldsymbol{Z}_0 = [X_{class}; X^1_p\boldsymbol{E}; X^2_p\boldsymbol{E}; \ldots; X^N_p\boldsymbol{E}] + \boldsymbol{E}_{pos}$ $\quad \boldsymbol{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \boldsymbol{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$
   $\boldsymbol{E}$は画像の埋め込み表現($\boldsymbol{D}$次元)への変換
   X_{class} はCLS Tokenで、パッチ系列の先頭に付加される
   $\boldsymbol{E}_{pos}$はPosition Embeddingsで、パッチ画像の位置関係を学習する
3. Transformer Encoderは$\boldsymbol{L}$回($\boldsymbol{L}$層)重ね、 $\boldsymbol{l}$層目について、
   Layer NormをLN Multi-Head self attentionをMSAとし、
   $\boldsymbol{Z'}_l = MSA\big(LN(\boldsymbol{Z}_{l-1})\big) + \boldsymbol{Z}_{l-1} \quad (l = 1 \ldots L)$
   $\boldsymbol{Z}_l = MLP\big(LN(\boldsymbol{Z'}_l)\big) + \boldsymbol{Z'}_l \quad (l = 1 \ldots L)$
4. Transformer Encoderの最終層では、[CLS]トークに当たる特徴量を出力
   $\boldsymbol{y} = LN(\boldsymbol{Z}^0_L)$
5. この$\boldsymbol{y}$をMLP Headに入力し、最終的な分類結果を得る

ViTは、大規模なデータセットで事前学習が行われ、ファインチューニングでは出力クラス数・入力解像度を変更することができます

論文中の実験において事前学習で用いられたデータセット

データセット	クラス数	画像枚数
ImageNet	1,000	130万枚
ImageNet-21k	21,000	1,400万枚
JFT-300M	18,000	3億枚

論文中の実験において用いられたモデルの種類

Model	Layers	Hidden size	MLP size	Heads	Params
ViT-Base	12	768	3,072	12	86M
ViT-Large	24	1,024	4,096	16	307M
ViT-Huge	32	1280	5,120	16	632M

論文中の実験において事前学習とファインチューニングの構成の違い

	事前学習	ファインチューニング
バッチサイズ	4,096	512
オプティマイザー	Adam	SGD with momentum

■ 事前学習

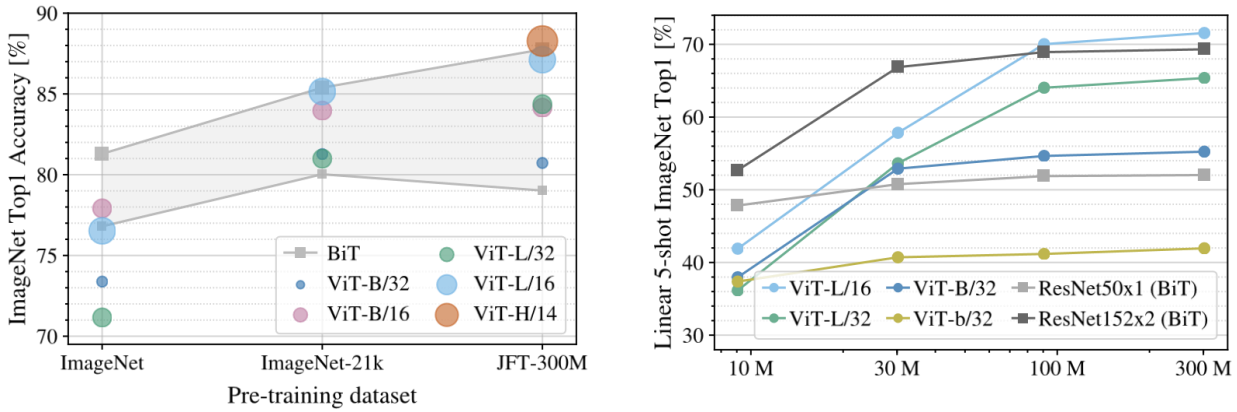
- ViTの事前学習は教師ラベル付きの大規模なデータセットで行われる
  - 論文では、ImageNet/ImageNet-21k/JFT-300M
- 事前学習の手順
  - 事前学習時にはMLP Headを出力層に使用
  - ファインチューニングより低解像度の画像を使用

■ ファインチューニング

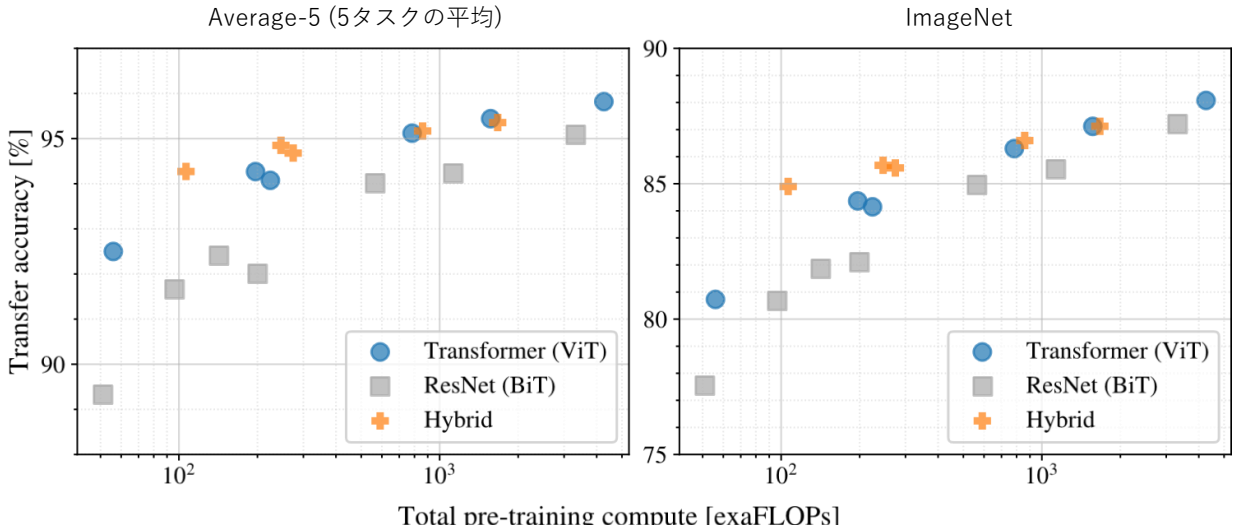
- 分類クラスの変更
  - Transformer EncoderはD次元の特徴量出力を行う
  - MLP HeadでD次元をKクラスへ変換する
  - ファインチューニング時にはMLP HeadをLinear層に取り換え
- 入力解像度の変更
  - パッチサイズは変更せず、入力時のPosition Embeddingを付替え
  - 事前学習よりファインチューニングでは高解像度の画像を使用

ViTは大規模データセットを使った事前学習で対BiTで性能の向上が見られる。同等の計算量であれば、BiTより高性能。

データセットの規模とImageNetタスクの精度



事前学習量とファインチューニング性能の関係



■ 事前学習におけるデータセット規模と性能の関係

- ImageNet < ImageNet-21k < JFT-300Mでの事前学習 (上段左)
  - ➢ Weight decay・Dropout・Label sommthingを最適化
  - ➢ 大規模データセットでViTがBiTより有利
- JFT-300Mのサブセット(9M < 30M < 90M < 300M)での事前学習 (上段右)
  - ➢ 大規模データセットでViTがResNetより有利

■ 事前学習における計算量と性能の関係 (下段)

- ViTは同計算量においてBiTより高性能
- ViTは大計算量域でさらなる性能向上が見られる
- Hybrid architectureは小計算量域で高性能 (大計算量では違いがない)

ViTのファインチューニング性能はCNNモデルより高く、学習に必要な計算量も少ないです

	Ours-JFT (ViT-H/14)	**Ours-JFT** (ViT-L/16)	**Ours-I21k** (ViT-L/16)	**BiT-L** (ResNet152x4)	**Noisy Student** (EfficientNet-L2)
ImageNet	**88.55 ± 0.04**	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4 ~ 88.5
ImageNet ReaL	**90.72 ± 0.05**	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50 ± 0.06**	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55 ± 0.04**	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56 ± 0.03**	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74 ± 0.00**	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63 ± 0.23**	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Vision Transformer (ViT)は、言語処理に用いられるTransformerを用いた、画像分類タスク用のモデルです。

■ Vision Transformerのデータ表現と入力

- 画像をパッチに分割し、パッチごとの特徴量を系列データとしてTransformerに入力する。
- パッチごとの特徴量は、ピクセル値をFlatten処理・Embedding処理(埋め込み)を行い、Position Embedding情報を加えたもの。
- 入力系列の1番目に、分類タスク用に特別な”[CLS]”トークンを連結するして入力する。従って、入力系列の長さは、パッチ数＋1 となる。

■ Vision Transformerのアーキテクチャ

- 言語処理におけるTransformerのエンコーダーとほぼ同様の構造である。(系列データを入力する。)
- エンコーダー部分からの出力における1トークン目の特徴量を、MLP Headに入力、最終的な分類結果を出力する。
- モデルサイズの違いにより、Base/Large/Hugeの3つが存在する。

■ Vision Transformerにおける事前学習(Pre-training)とファインチューニング(Fine-tuning)

- ViTの事前学習は、教師ラベル付きの巨大データセットで行われ、性能がデータセット規模の影響を受ける
- ファインチューニングでは、MLP Head部分を取り換えることで、分類タスクにおけるクラス数の違いに対応する。
- ファインチューニングで、事前学習より高い解像度の画像に、Position Embeddingの変更のみで対応できる。

■ 性能とその評価

- 事前学習におけるデータセットが大規模な場合において、既存手法より高性能。
- 事前学習のデータセットが小規模な場合、既存手法(CNN)に比べ低性能。
- 同計算量において、既存手法より高性能。大計算量域でさらなる性能向上の余地がある。