BEIT : BERT Pre-Training of Image Transformers (ICLR 2022)

- Young Jo Choi
- Department of Digital Analytics

Abstract

- Self-supervised vision representation model
- **B**idirectional **E**ncoder representation from **I**mage **T**ransformers
- Following BERT, the paper propose *masked image modeling* task to pretrain vision Transformer.
- Tokenizing the original image into visual token → masking some image patches → backbone Transformer.
- The pre-training objective is to recover the original visual tokens based on the corrupted image patches.
- Experimental results in image classification and semantic segmentation show that BEIT achieves competitive results with previous pre-training methods.

Introduction

- In computer vision fields, empirical studies show that vision Transformers require more training data than convolutional neural networks. To cope with the lack of data, self-supervised pre-training is a promising solution, such as contrastive learning and self-distillation.

- BERT has achieved great success in natural language processing (NLP) by masking certain parts of the text and recovering the masked parts. (=MLM, masked language model)

Introduction

- It is challenging to directly apply BERT style pre-training for image data. There is no pre-exist vocabulary for vision Transformer's input unit (image patches). So, we cannot simply employ a softmax classifier to predict over all possible candidates for masked patches.

- In NLP the language vocabulary, such as words and BPE, is well-defined and eases auto-encoding prediction.

- A straightforward application of BERT is regarding the task as a regression problem, which predicts the raw pixels of masked patches. However, such pixel-level recovery task tends to waste modeling capability on pre-training shortrange dependencies and high-frequency details.

- The goal is to overcome the above issues for pre-training of vision Transformers.

Related works (BERT)

- **B**idirectional **E**ncoder **R**epresentation from **T**ransformer

- Masked language model (MLM)

- EX)
  Input: My dog is [A]. He likes playing.
  Target : 'cute'

- Words are tokenized according to predefined criteria, which BERT uses WordPiece tokenizer. (BPE for GPT)

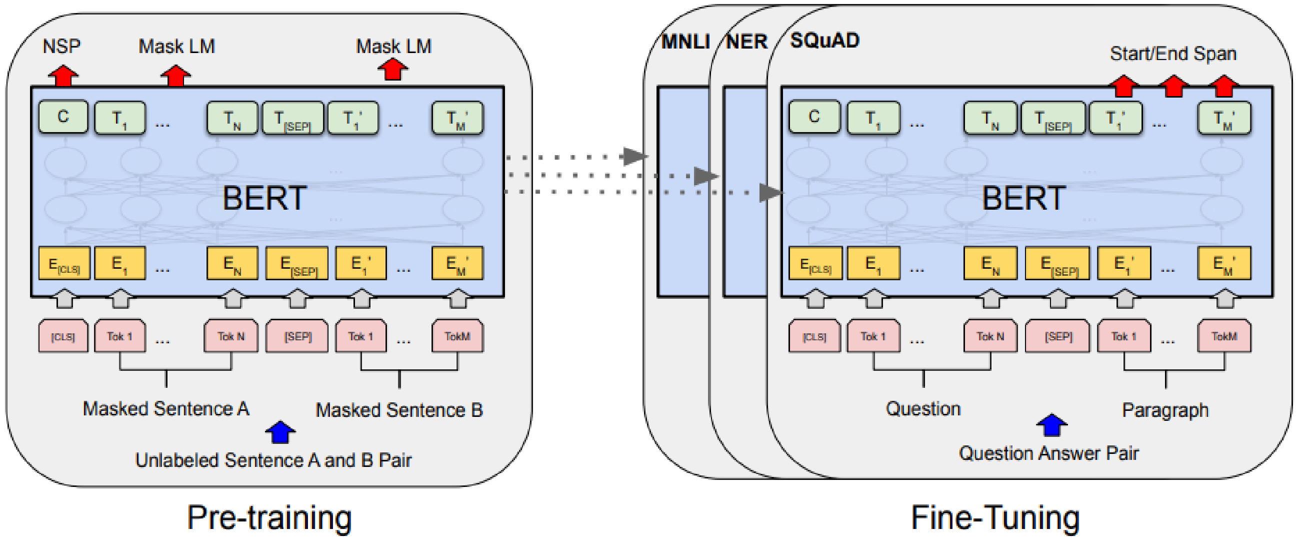
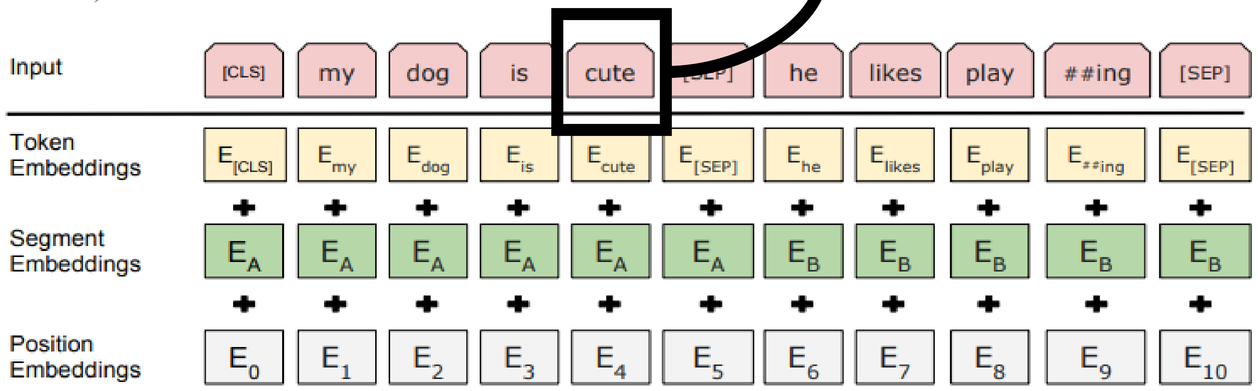


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).



Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

Related works (variational auto encoder)

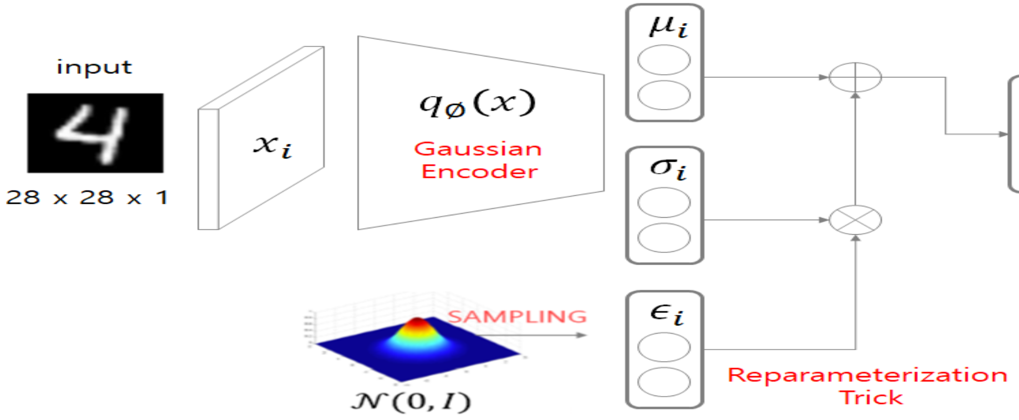


그림 출처 : https://taeu.github.io/paper/deeplearning-paper-vae/

- $z_{i,l} \sim N(\mu_i, {\sigma_i}^2 I)$ ➔ $z_i = \mu_i + \sigma_i \cdot \varepsilon,\ \ \varepsilon \sim N(0, I)$ (for backpropagation)
- Objective : maximize $p_\theta(x_i)$

$$\log p_\theta(x_i) = \mathbb{E}_{z\sim q_\emptyset(z|x_i)}[p_\theta(x_i)] = \mathbb{E}_z[\log p_\theta(x_i|z)] - \mathbb{E}_z\left[\log \frac{q_\emptyset(z|x_i)}{p_\theta(z)}\right] + \mathbb{E}_z\left[\log \frac{q_\emptyset(z|x_i)}{p_\theta(z|x_i)}\right]$$

$$= \underbrace{\mathbb{E}_z[\log p_\theta(x_i|z)] - D_{KL}\big(q_\emptyset(z|x_i)\,\big|\,p_\theta(z)\big)}_{\text{ELBO loss}} + \underbrace{D_{KL}\big(q_\emptyset(z|x_i)\,|\,p_\theta(z|x_i)\big)}_{\geq 0 \text{ (non-computable)}}$$

$$\rightarrow \underset{\theta,\emptyset}{\operatorname{argmax}} \textstyle\sum_i \underbrace{-\mathbb{E}_z[\log p_\theta(x_i|z)]}_{\text{Reconstruction error}} + \underbrace{D_{KL}\big(q_\emptyset(z|x_i)\,\big|\,p_\theta(z)\big)}_{\text{Regularization error}}$$

Related works (discrete VAE, VQ-VAE)

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." *Advances in neural information processing systems* 30 (2017).

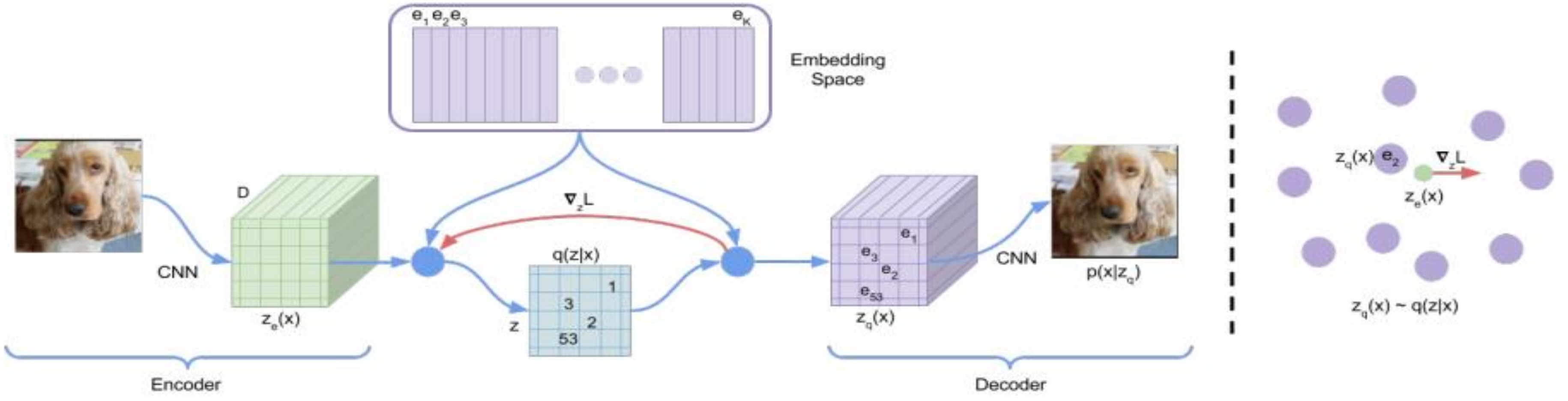


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

- Input image : x
- Output of encoder $z_e(x)$ is mapped to the nearest point in "Codebook"
- Decoder input $z_q(x) = e_k,$ where $k = \underset{j}{\operatorname{argmin}} \left\| z_e(x) - e_j \right\|_2$
- During forward computation the nearest embedding $z_q(x)$ is passed to the decoder, and during the backwards pass the gradient $\nabla_z L$ is passed unaltered to the encoder.

YONSEI UNIVERSITY

Introduction

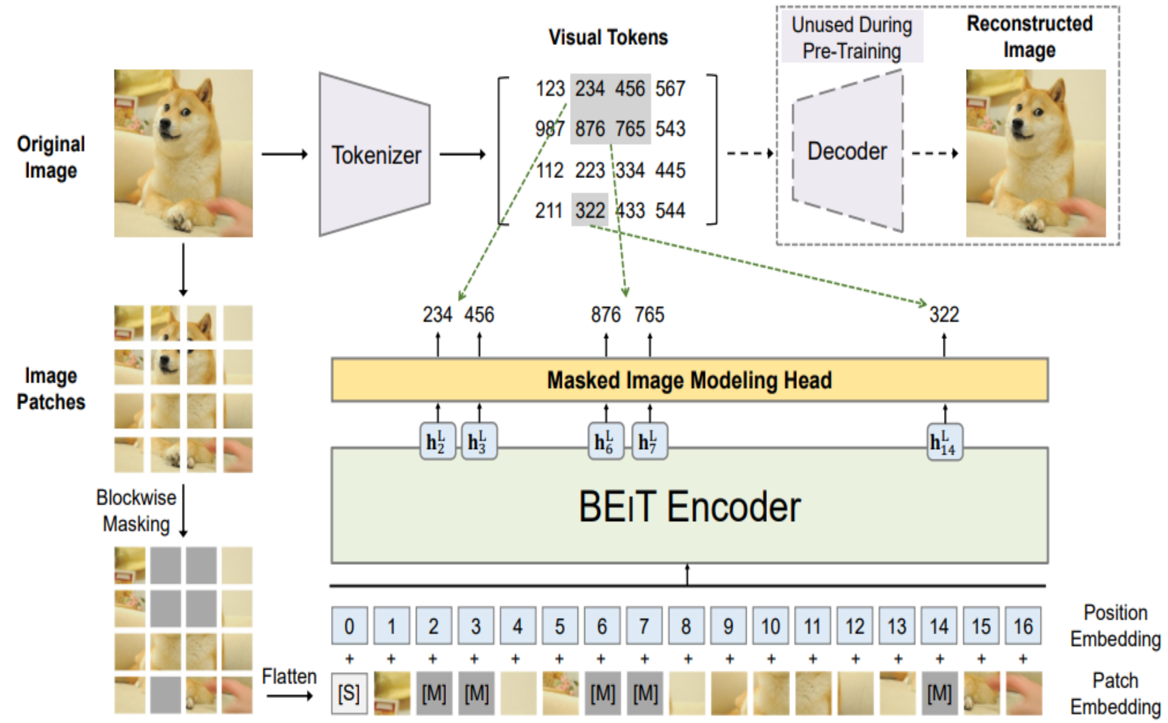


Figure 1: Overview of BEIT pre-training. Before pre-training, we learn an "image tokenizer" via autoencoding-style reconstruction, where an image is tokenized into discrete visual tokens according to the learned vocabulary. During pre-training, each image has two views, i.e., image patches, and visual tokens. We randomly mask some proportion of image patches (gray patches in the figure) and replace them with a special mask embedding [M]. Then the patches are fed to a backbone vision Transformer. The pre-training task aims at predicting the visual tokens of the *original* image based on the encoding vectors of the *corrupted* image.

- Inspired by BERT, masked image model (MIM) is proposed.
- MIM uses two views for each images, i.e., image patches, and visual tokens.
- We split the image into a grid of patches that are the input representation of backbone Transformer.

  (some proportions are randomly masked)
- Moreover, we 'tokenize' the image to discrete visual tokens, which is obtained by latent codes of discrete VAE.

Introduction

(contributions)

- We propose a masked image modeling task to pretrain vision Transformers in a self-supervised manner. We also provide a theoretical explanation from the perspective of variational autoencoder.

- We pretrain BEIT and conduct extensive fine-tuning experiments on downstream tasks, such as image classification, and semantic segmentation. Experimental results indicate that BEIT outperforms both from-scratch training and previous strong self-supervised models.

- We present that the self-attention mechanism of self-supervised BEIT learns to distinguish semantic regions and object boundaries, although without using any human annotation.

Methods (Image Representation)

- Image Patch
  - Input image $x \in \mathbb{R}^{H \times W \times C}$ → $x^p \in \mathbb{R}^{N \times (P^2 C)}$, $N = HW/P^2$ is the number of patches,
    (P, P) : resolution of each patch
  - The image patches $\{x_i^p\}_{i=1}^N$ are flattened into vectors and linearly projected, which is similar to word embeddings BERT
  - In this experiments, 224×224 images → 14×14 grid of patches (each patch is 16×16)

- Visual Token
  - $z = [z_1, \dots, z_N] \in \mathcal{V}^{h \times w}$, where the vocabulary $\mathcal{V} = \{1, \dots, |\mathcal{V}|\}$ contains discrete token indices (h, w : grid of patches, $h \times w = N$)
  - $q_\emptyset(z|x)$ maps image pixels x into discrete tokens z according to a visual codebook (i.e., vocabulary)
  - $p_\psi(x|z)$ (=decoder) learns to reconstruct the input image x based on the visual tokens z.
  - Reconstruction object can be written as $\mathbb{E}_{z \sim q_\emptyset(z|x)}[\log p_\psi(x|z)]$
  - In this experiments, each image is tokenized to a 14×14 grid of visual tokens. Set $|\mathcal{V}|$=8192.

Methods (Backbone Network : Image Transformer)

- The input of Transformer is a sequence of image patches $\{x_i^p\}_{i=1}^N$
- Ex_i^p, where $E \in \mathbb{R}^{(P^2C)\times D}$: patches are linearly projected.
- $E_{pos} \in \mathbb{R}^{N\times D}$: positional encoding
- The input vectors $H_0 = \left[e_{[s]}, Ex_1^p, \dots, Ex_N^p\right] + E_{pos}$ is fed into Transformer.
  $e_{[s]}$: special token
- Transformer blocks $H^l = \mathrm{Transformer}(H^{l-1})$, where $l = 1, \dots L$
- The output vectors of last layer $H^L = [h_{[s]}^L, h_1^L, \dots, h_N^L]$: encoded representation for image patches, where h_i^L is the vector of i−th image patch.

Methods (Pre-Training BEIT: Masked Image Modeling)

- $\mathcal{M} = \{1, \ldots, \mathrm{N}\}^{0.4\mathrm{N}}$: index of masked patches
- $e_{[M]} \in \mathbb{R}^{D}$: learnable parameter that functions as replace the masked patches.
- The corrupted image patches $x^{\mathcal{M}} = \{x_i^p : i \notin \mathcal{M}\}_{i=1}^{N} \cup \{e_{[M]} : i \in \mathcal{M}\}_{i=1}^{N}$ are fed into the L-layer Transformer.
- For each masked position $\{h_i^L : i \in \mathcal{M}\}_{i=1}^{N} \rightarrow p_{MIM}(z' | x^{\mathcal{M}}) = \mathrm{softmax}_{z'}(W_c h_i^L + b_c)$, where $W_c \in \mathbb{R}^{|\mathcal{V}| \times D}, b_c \in \mathbb{R}^{|\mathcal{V}|}$

$$\max \sum_{x \in D} \mathbb{E}_{\mathcal{M}} \left[\sum_{i \in \mathcal{M}} \log p_{MIM}(z_i | x^{\mathcal{M}}) \right]$$

Methods

- Rather than randomly choosing patches for the masked positions M, we employ blockwise masking in our work

Algorithm 1 Blockwise Masking

Input: $N(= h \times w)$ image patches
Output: Masked positions $\mathcal{M}$
$\mathcal{M} \leftarrow \{\}$
repeat
  $s \leftarrow \mathsf{Rand}(16, 0.4N - |\mathcal{M}|)$ ▷ *Block size*
  $r \leftarrow \mathsf{Rand}(0.3, \frac{1}{0.3})$ ▷ *Aspect ratio of block*
  $a \leftarrow \sqrt{s \cdot r}; b \leftarrow \sqrt{s/r}$
  $t \leftarrow \mathsf{Rand}(0, h - a)\ ; l \leftarrow \mathsf{Rand}(0, w - b)$
  $\mathcal{M} \leftarrow \mathcal{M} \bigcup \{(i, j) : i \in [t, t + a), j \in [l, l + b)\}$
until $|\mathcal{M}| > 0.4N$ ▷ *Masking ratio is 40%*
return $\mathcal{M}$

- Select at least 16 patches
- Grouping several patches that are close to each other into a block and masking them all at once
- r : aspect ratio of block
- Repeat these steps until obtaining enough masked patches.

Methods (from perspective of variational autoencoder)

The BEIT pre-training can be viewed as variational autoencoder training.

x : original image, $\tilde{x}$: masked image, z : visual tokens

- ELBO

$$\sum_{(x_i,\tilde{x}_i)\in D} \log p(x_i|\tilde{x}_i) \geq \sum_{(x_i,\tilde{x}_i)\in D} \left(\underbrace{\mathbb{E}_{z_i\sim q_\emptyset(z|x_i)}[\log p_\psi(x_i|z_i)]}_{\text{visual token reconstruction}} - D_{KL}[q_\emptyset(z|x_i), p_\theta(z|\tilde{x}_i)] \right)$$

Simplifying

- $q_\emptyset(z|x_i)$: image tokenizer that obtains visual tokens
- $p_\psi(x_i|z_i)$: decodes the original image given input visual tokens
- $p_\theta(z|\tilde{x}_i)$: recovers the visual tokens based on the masked image

$$\sum_{(x_i,\tilde{x}_i)\in D} \left(\underbrace{\mathbb{E}_{z_i\sim q_\emptyset(z|x_i)}[\log p_\psi(x_i|z_i)]}_{\text{stage 1}} + \underbrace{\log p_\theta(\hat{z}_i|\tilde{x}_i)}_{\text{stage 2}} \right)$$

- Stage 1 : visual token reconstruction
- Stage 2 : masked image modeling (BEIT pre-training objective)

Methods

- Pre-training setup
  - Network architecture : ViT-Base (12-layer transformer with 768 hidden size, 12 attention heads)
  - Training on Imagenet-1K
  - 224×224 images → 14×14 grid of patches ($N = 196$) (each patch is 16×16)
  - Randomly mask at most 75 patches (i.e, roughly 40% of total image patches)
  - (Training steps take about five days using 16 Nvidia Telsa V100 32GB GPU cards.)

- Fine-tuning BEIT on downstream tasks (end-to-end fine-tuned)
  - Image classification : a simple linear classifier is directly employed.

    $\mathrm{softmax}(\mathrm{avg}(\{h_i^L\}_{i=1}^N W_c)),\ W_c \in \mathbb{R}^{D \times C},\ C$: number of labels
  - Semantic segmentation : pretrained BEIT is used as backbone encoder, and several deconvolution layers are incorporated as decoder to produce segmentation.

Experiments (image classification)

Models	Model Size	Resolution	ImageNet
Training from scratch (i.e., random initialization)			
ViT_{384}-B [DBK+20]	86M	384^2	77.9
ViT_{384}-L [DBK+20]	307M	384^2	76.5
DeiT-B [TCD+20]	86M	224^2	81.8
DeiT_{384}-B [TCD+20]	86M	384^2	83.1
Supervised Pre-Training on ImageNet-22K (using labeled data)			
ViT_{384}-B [DBK+20]	86M	384^2	84.0
ViT_{384}-L [DBK+20]	307M	384^2	85.2
Self-Supervised Pre-Training on ImageNet-1K (without labeled data)			
iGPT-1.36B† [CRC+20]	1.36B	224^2	66.5
ViT_{384}-B-JFT300M‡ [DBK+20]	86M	384^2	79.9
MoCo v3-B [CXH21]	86M	224^2	83.2
MoCo v3-L [CXH21]	307M	224^2	84.1
DINO-B [CTM+21]	86M	224^2	82.8
BEIT-B (ours)	86M	224^2	83.2
BEIT_{384}-B (ours)	86M	384^2	84.6
BEIT-L (ours)	307M	224^2	85.2
BEIT_{384}-L (ours)	307M	384^2	**86.3**

Table 1: Top-1 accuracy on ImageNet-1K. We evaluate base- ("-B") and large-size ("-L") models at resolutions 224×224 and 384×384. †: iGPT-1.36B contains 1.36 billion parameters, while others are base-size models. ‡: ViT_{384}-B-JFT300M is pretrained with the "masked patch prediction" task on Google's in-house 300M images, while others use ImageNet.

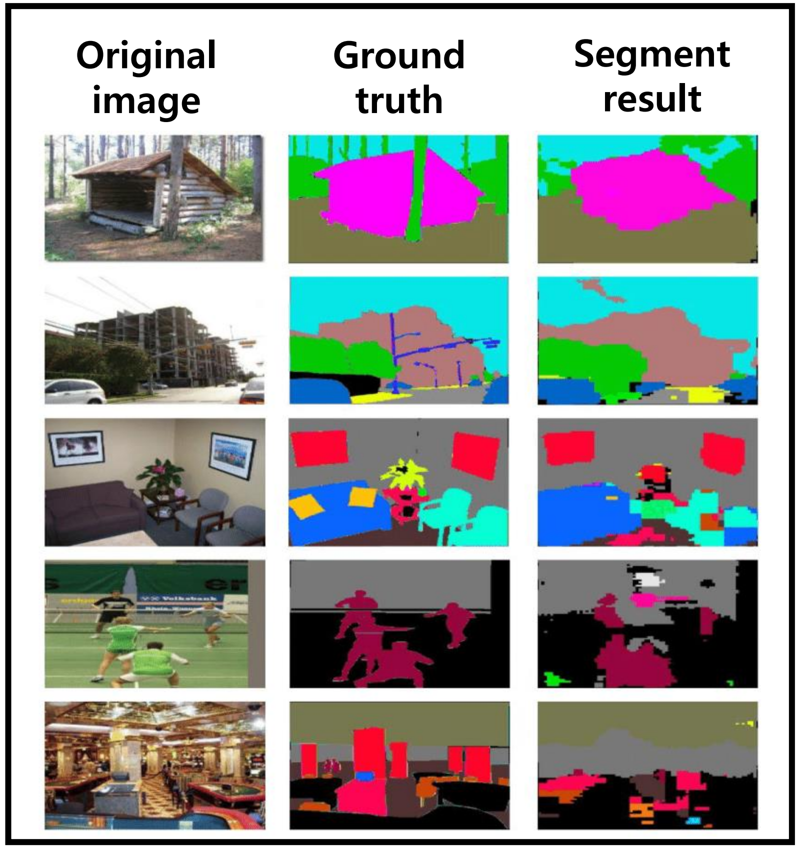
- Evaluation on ImageNet-1K.
- Comparing among Training from scratch, pre-training on larger dataset (ImageNet-22K) and previous SOTA self-supervised methods for Transformer.
- Input image resolution is fixed at 224 or 384.
- For the same model size, pre-training with the self-supervised method performs better than pre-training with a larger dataset.

Experiments (semantic segmentation)

- Evaluation on ADE20K benchmark with 25K images and semantic categories.
- Reported metric is mean Intersection of Union (mIoU) averaged over all semantic categories.
- In this experiments, task layers in SETR-PUP are used to produce segmentation.
- We find that our proposed method achieves better performance than supervised pretraining, although BEIT does not require manual annotations for pre-training.
- BEIT + Intermediate Fine-Tuning : first, pre-trained BEIT is fine-tuned on ImageNet, and then fine-tuned on ADE20K.

Models	ADE20K
Supervised Pre-Training on ImageNet	45.3
DINO [CTM+21]	44.1
BEIT (ours)	45.6
BEIT + Intermediate Fine-Tuning (ours)	**47.7**

Table 3: Results of semantic segmentation on ADE20K. We use SETR-PUP [ZLZ+20] as the task layer and report results of single-scale inference.



Experiments (Ablation studies)

Models	ImageNet	ADE20K
BEiT (300 Epochs)	82.86	44.65
− Blockwise masking	82.77	42.93
− Visual tokens (i.e., recover masked pixels)	81.04	41.38
− Visual tokens − Blockwise masking	80.50	37.09
+ Recover 100% visual tokens	82.59	40.93
− Masking + Recover 100% visual tokens	81.67	36.73
Pretrain longer (800 epochs)	83.19	45.58

Table 4: Ablation studies for BEiT pre-training on image classification and semantic segmentation.

Image Tokenizer	Reconstruction Error	ImageNet
DALL-E Tokenizer [RPG+21]	**0.0856**	**82.86**
Our reimplementation	0.0880	82.70

- Blockwise masking is better than random masking.
- Usage of visual tokens is helpful than predicting the raw pixels of masked patches
- Recovering all the visual tokens harms performance on downstream tasks
- Pre-training the model longer can further improve performance on downstream tasks.
- our reimplemented tokenizer obtains comparable reconstruction loss and ImageNet fine-tuning performance compared with the off-the-shelf DALL-E tokenizer.

Experiments (Analysis of Self-Attention Map)

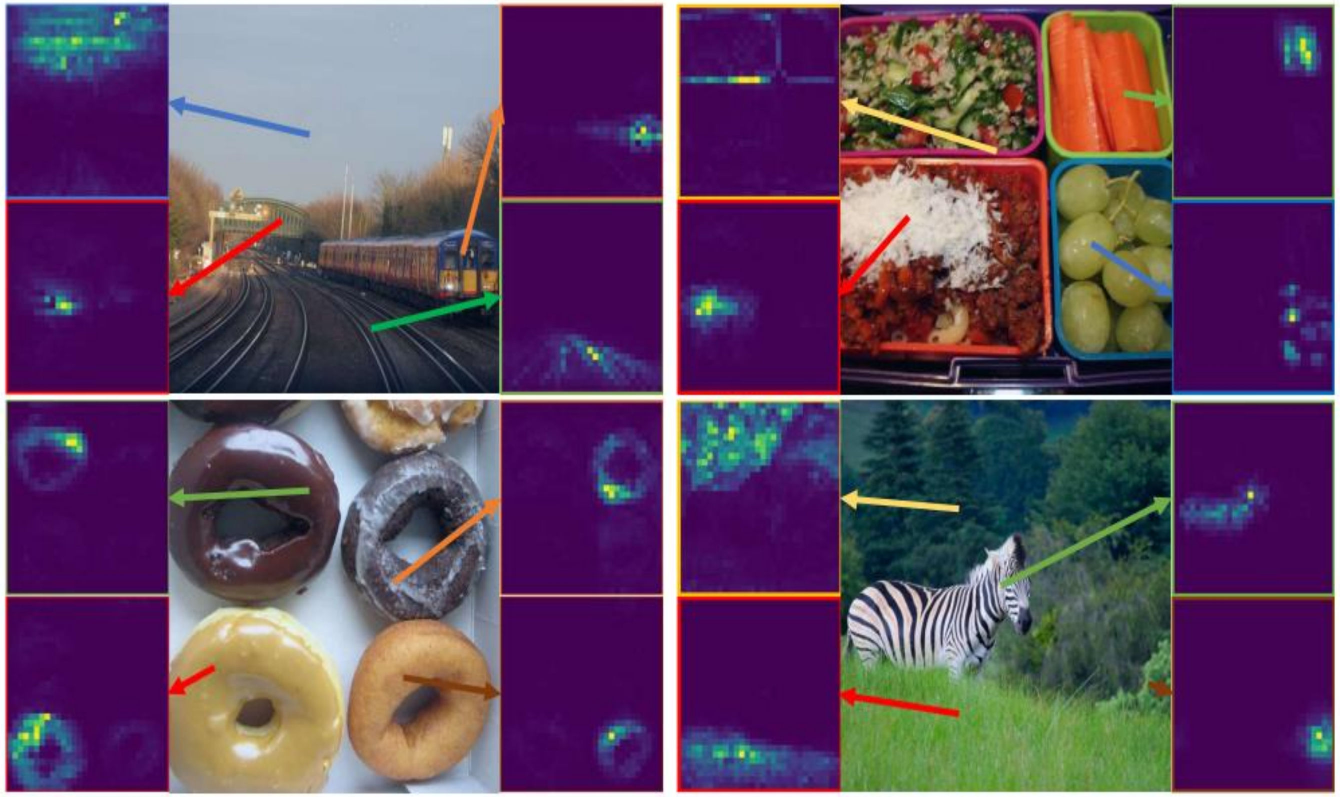


Figure 2: Self-attention map for different reference points. The self-attention mechanism in BEIT is able to separate objects, although self-supervised pre-training does not use manual annotations.

- In Fig 2, the visualizations are produced by attention scores computed via query-key product in the last layer.
- For each reference point, we use the corresponding patch as query and show which patch is attends to.
- We show that the self-attention mechanism in BEIT can separate objects, even though our pre-training does not rely on any manual annotation at all.

Conclusion

- We introduce a self-supervised pre-training framework for vision Transformers, achieving strong fine-tuning results on downstream tasks, such as image classification, and semantic segmentation.
- We show that the proposed method is critical to make BERT-like pre-training (i.e., auto-encoding with masked input) work well for image Transformers.
- We also present the intriguing property of automatically acquired knowledge about semantic regions, without using any human-annotated data.
- In the future, we would like to scale up BEIT pre-training in terms of data size and model size. Moreover, we will conduct multimodal pre-training in a more unified way, using the similar objectives and the shared architecture for texts and images.

YONSEI UNIVERSITY
COLLEGE OF MEDICINE