CLIP

OpenAI, Alec Radford, Jong Wook Kim et al.

Presenter : Taeu Kim

Contents

- Problem & Goal
- Background & Related work
- Approach
- Key
- Limitations

Problem

- Typical vision dataset are labor intensive and costly to create
- Moreover, teach only a narrow set of visual concepts
- Require significant effort to adapt to a new task

Goal

- Typical vision dataset are labor intensive and costly to create
- Moreover, teach only a narrow set of visual concepts
- Require significant effort to adapt to a new task

→ 1. Collect dataset without labor to annotate

→ 2. Learn Abundant visual concepts with natural language supervision

→ 3. Good performance on zero-shot classification

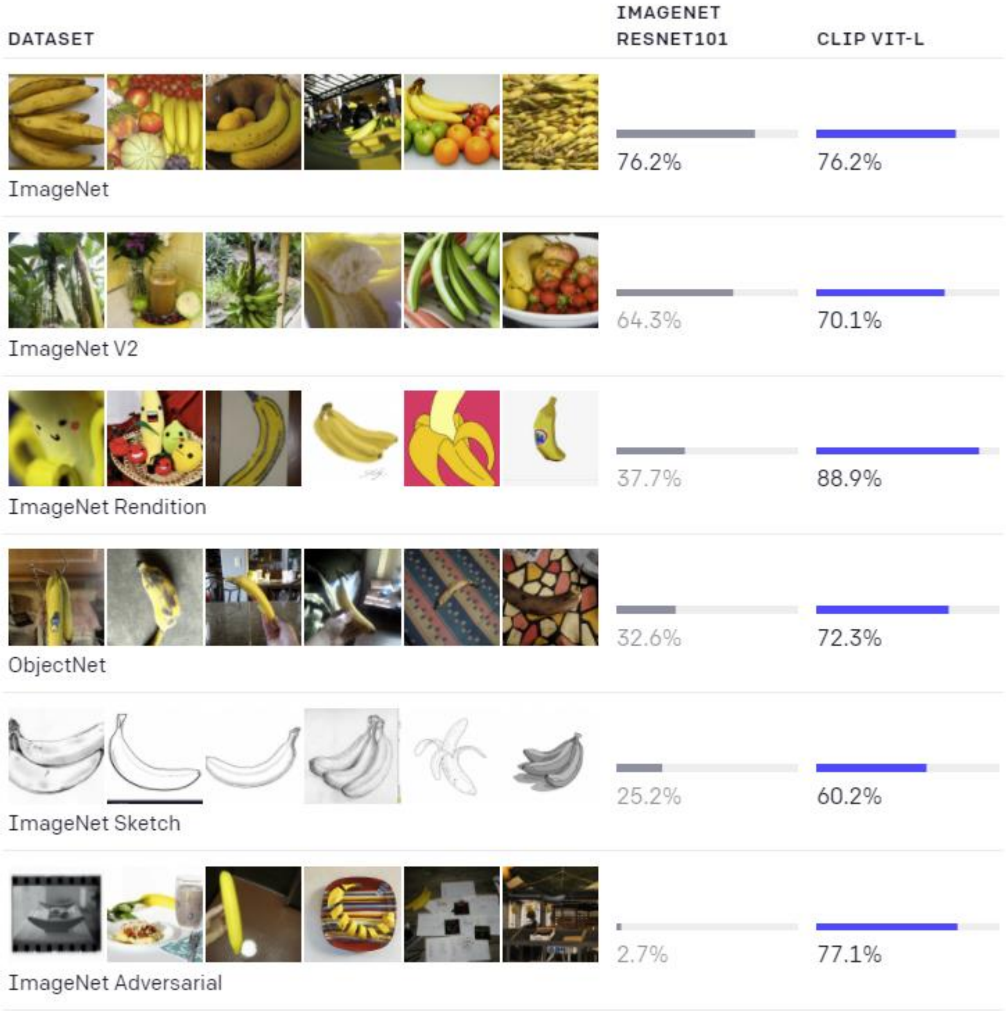
Goal

- Typical vision dataset are labor intensive
- Moreover, teach only a narrow set of vis
- Require significant effort to adapt to a ne

→ Collect dataset without labor to annotat

→ Learn Abundant visual concepts with natural language supervision

→ Good performance on zero-shot classification



DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%
ImageNet V2	64.3%	70.1%
ImageNet Rendition	37.7%	88.9%
ObjectNet	32.6%	72.3%
ImageNet Sketch	25.2%	60.2%
ImageNet Adversarial	2.7%	77.1%

Background & Related Work

- CLIP (*Contrastive Language–Image Pre-training*) builds on a large body of work on zero-shot transfer, natural language supervision, and multimodal learning.
- Most inspirational for CLIP is the work* of Ang Li and his co-authors at FAIR who in 2016 demonstrated using natural language supervision to enable zero-shot transfer to several existing computer vision classification datasets, such as the canonical ImageNet dataset. (*Learning Visual N-Grams from Web Data)

https://openaccess.thecvf.com/content_ICCV_2017/papers/Li_Learning_Visual_N-Grams_ICCV_2017_paper.pdf

Background & Related Work

- Transformer[32]
- VirTex[33]
- ConVIRT[35] (contrastive objective)

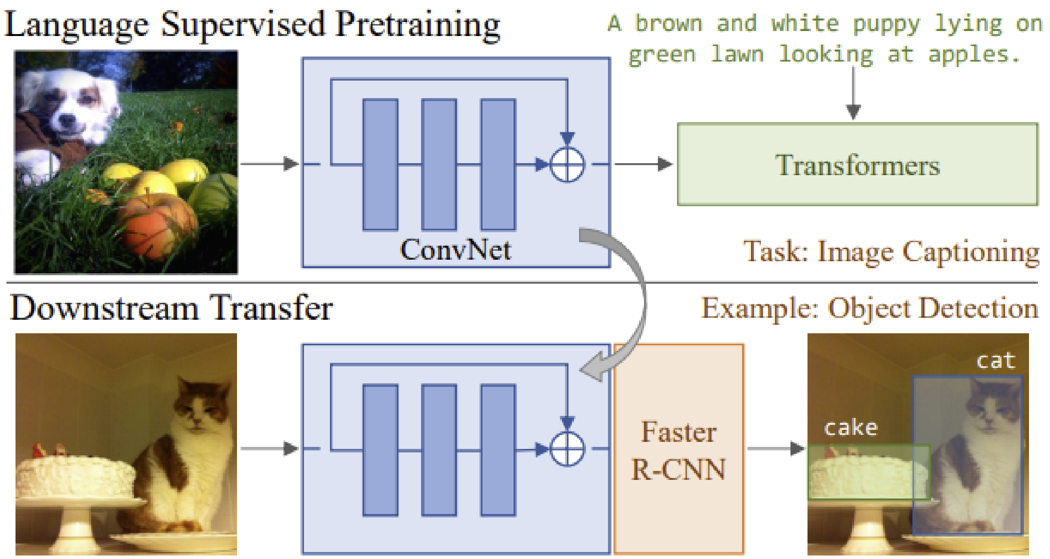
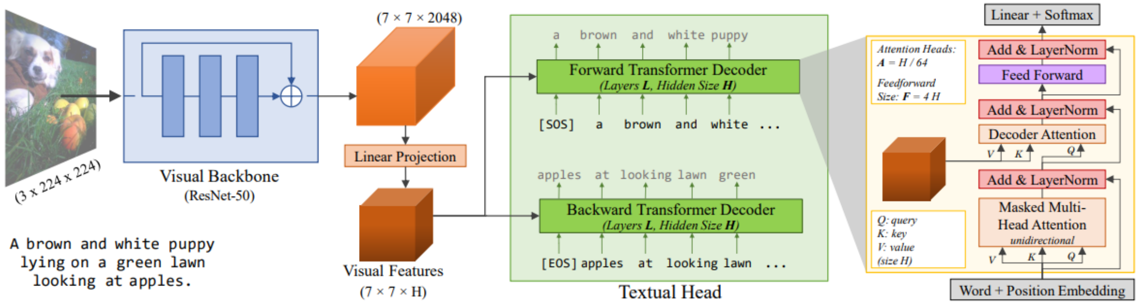

Language Supervised Pretraining
A brown and white puppy lying on green lawn looking at apples.
ConvNet
Transformers
Task: Image Captioning
Downstream Transfer
Example: Object Detection
Faster R-CNN
cat
cake

Figure 1: **Learning visual features from language:** First, we jointly train a ConvNet and Transformers using image-caption pairs, for the task of image captioning (top). Then, we transfer the learned ConvNet to several downstream vision tasks, for example object detection (bottom).

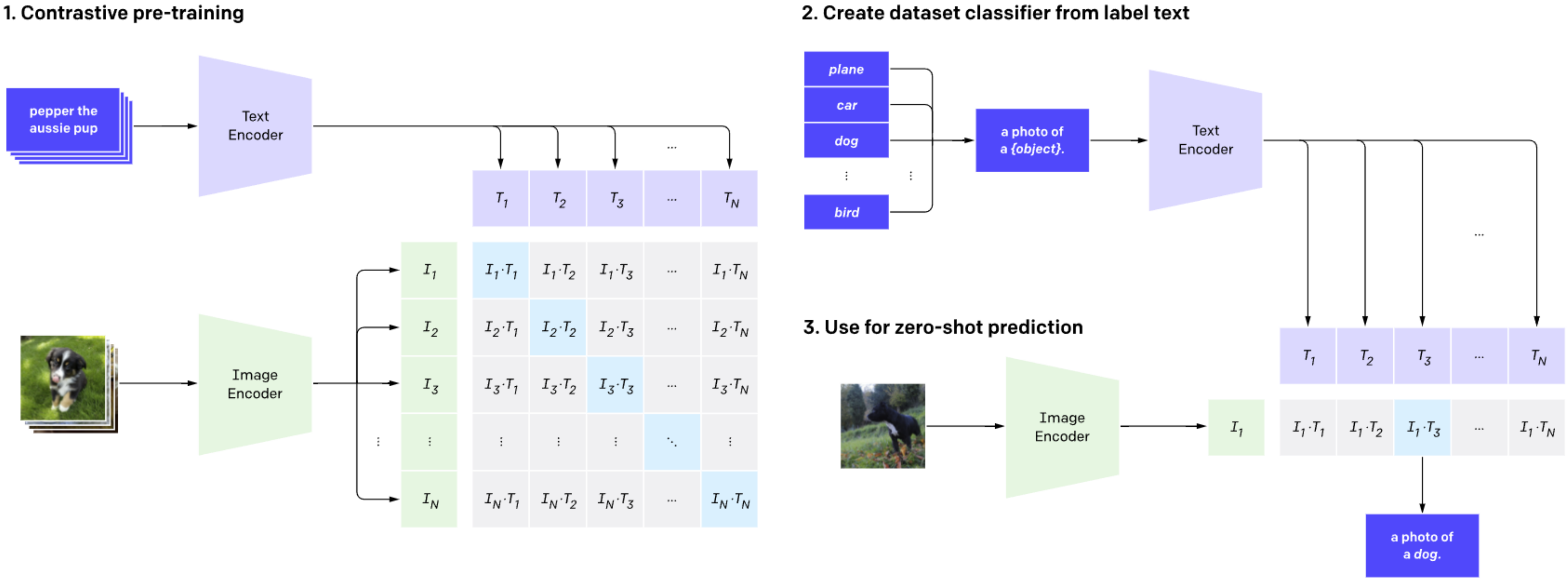


[32] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). "Attention is all you need." In NeurIPS 2017

[33] Desai, K., & Johnson, J. (2020). "VirTex: Learning Visual Representations from Textual Annotations." arXiv preprint

[35] Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., & Langlotz, C. P. (2020). "Contrastive Learning of Medical Visual Representations from Paired Images and Text." arXiv preprint

Approach



1. Contrastive pre-training

	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

2. Create dataset classifier from label text

3. Use for zero-shot prediction

	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$

Text Encoder : Transformer (63M params, 12layers 512-wide model with 8 attention heads)

Image Encoder : Vision Transformer

Result

https://openai.com/blog/clip/

To mitigate 3 major problems

- 1. Costly datasets

manually labeled datasets

The ImageNet dataset, one of the largest efforts in this space, required over 25,000 workers to annotate 14 million images for 22,000 object categories.

Without extra label

400 million (image, text) pairs collected form a variety of publicly available sources on the Internet.

WIT | WebImageText.

We search for (image, text) pairs as part of the construction process whose text includes one of a set of 500,000 queries. The base query list is all words occurring at least 100 times in the English version of Wikipedia This is augmented with bi-grams with high pointwise mutual information as well as the names of all Wikipedia articles above a certain search volume. Finally all WordNet synsets not already in the query list are added.

To mitigate 3 major problems

- 2. Narrow

Need to construct new dataset and fine-tune

An ImageNet model is good at predicting the 1000 ImageNet categories, but that's all it can do "out of the box."

Zeroshot classificaiton

To apply CLIP to a new task, all we need to do is "tell" CLIP's text-encoder the names of the task's visual concepts, and it will output a linear classifier of CLIP's visual representations.

To mitigate 3 major problems

- 2. Narrow

We show random, non-cherry picked, predictions of zero-shot CLIP classifiers on examples from various datasets below.

https://openai.com/blog/clip/

To mitigate 3 major problems

- 3. Poor real-world performance

there is a gap between “benchmark performance” and “real performance.”

We conjecture that this gap occurs because the models “cheat” by only optimizing for performance on the benchmark

In contrast, the CLIP model can be evaluated on benchmarks without having to train on their data, so it can’t “cheat” in this manner.

Key Takeaways

- 1. CLIP is highly efficient

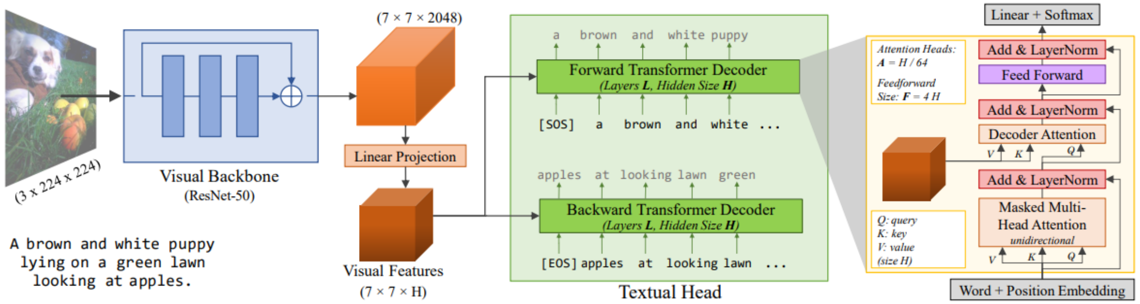
Training efficiency

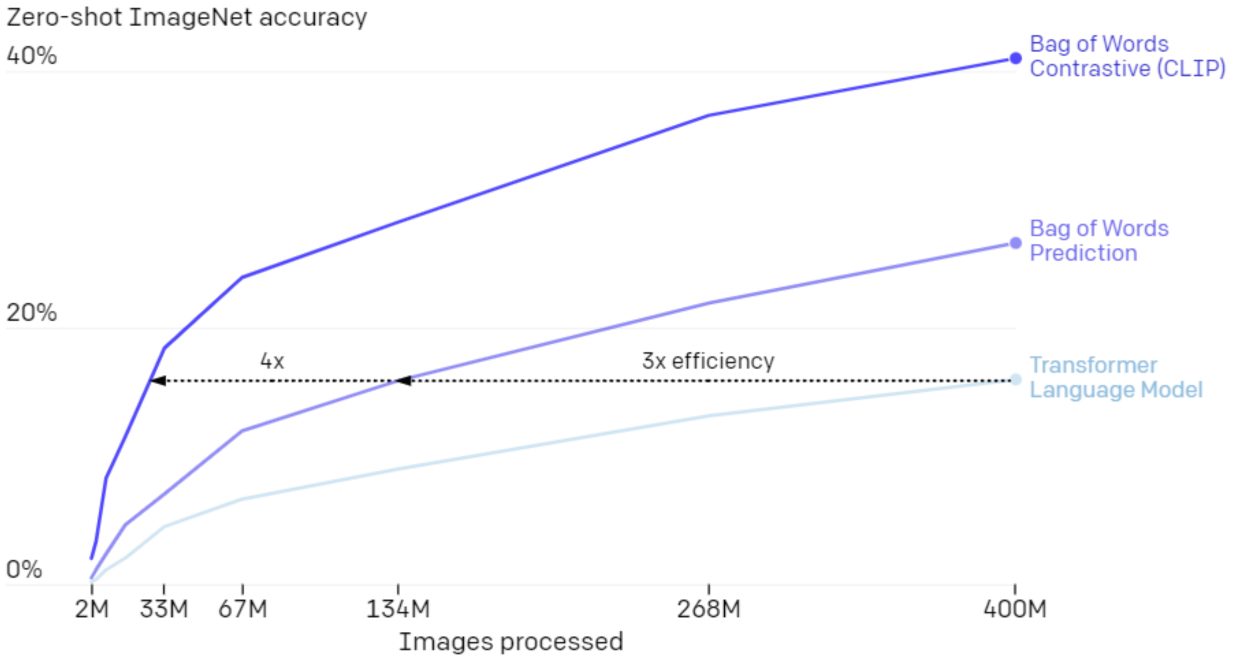
1. The adoption of a contrastive objective for connecting text with images →

2. The adoption of the Vision Transformer which gave us a further 3x gain in compute efficiency over a standard ResNet.

In the end, our best performing CLIP model trains on 256 GPUs for 2 weeks which is similar to existing large scale image models

VirTex


(3 x 224 x 224)
A brown and white puppy lying on a green lawn looking at apples.
Visual Backbone
(ResNet-50)
(7 × 7 × 2048)
Linear Projection
Visual Features
(7 × 7 × H)
a brown and white puppy
Forward Transformer Decoder
(Layers L, Hidden Size H)
[SOS] a brown and white ...
apples at looking lawn green
Backward Transformer Decoder
(Layers L, Hidden Size H)
[EOS] apples at looking lawn ...
Textual Head
Attention Heads: A = H / 64
Feedforward Size: F = 4 H
Q: query
K: key
V: value
(size H)
Linear + Softmax
Add & LayerNorm
Feed Forward
Add & LayerNorm
Decoder Attention
V K Q
Add & LayerNorm
Masked Multi-Head Attention
unidirectional
V K Q
Word + Position Embedding

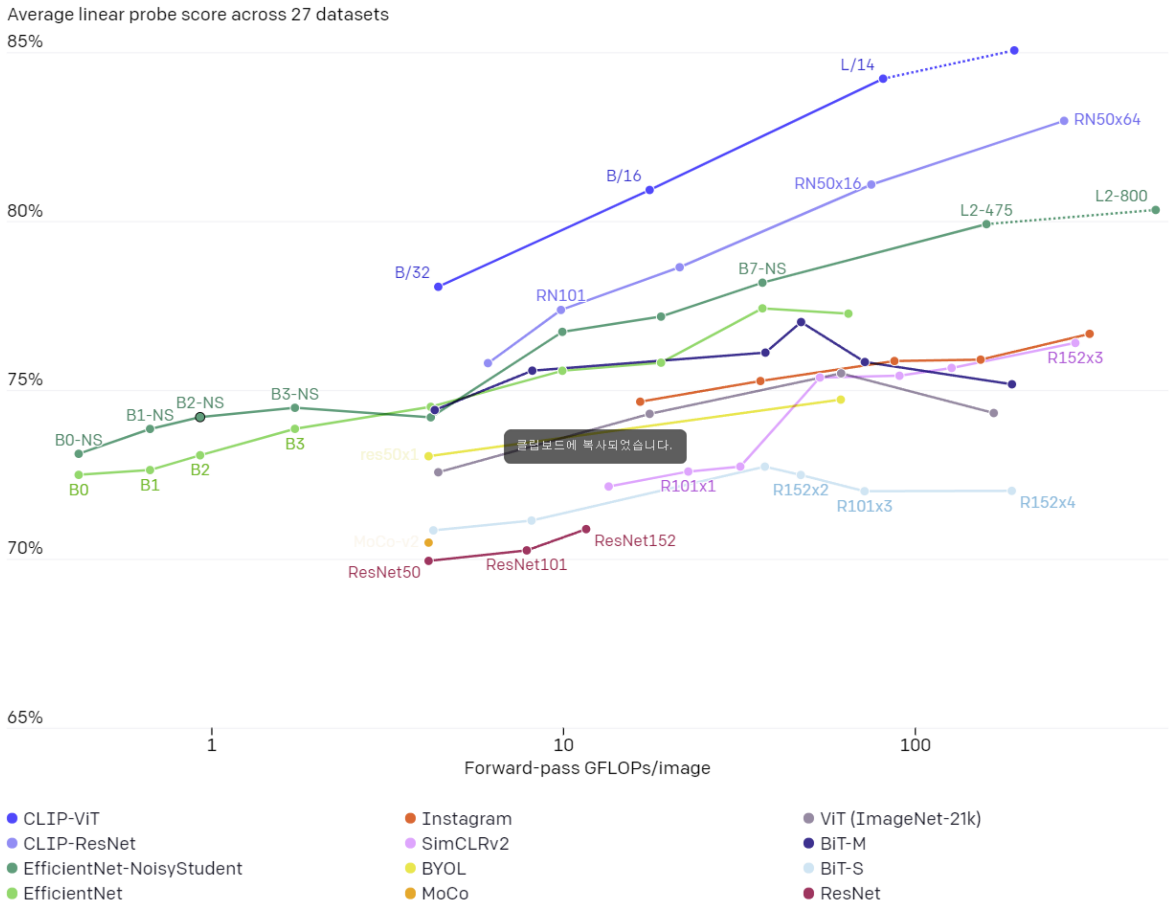


We originally explored training image-to-caption language models but found this approach struggled at zero-shot transfer. In this 16 GPU day experiment, a language model only achieves 16% accuracy on ImageNet after training for 400 million images. CLIP is much more efficient and achieves the same accuracy roughly 10x faster.

Key Takeaways

- 2. CLIP is flexible and general

Across a suite of 27 datasets measuring tasks such as fine-grained object classification, OCR, activity recognition in videos, and geo-localization, we find that CLIP models learn more widely useful image representations. CLIP models are also more compute efficient than the models from 10 prior approaches that we compare with.



Limitations

1. It struggles on more abstract or systematic tasks such as counting the number of objects in an image and on more complex tasks such as predicting how close the nearest car is in a photo. On these two datasets, zero-shot CLIP is only slightly better than random guessing.

2. Zero-shot CLIP also struggles compared to task specific models on very fine-grained classification.

3. CLIP also still has poor generalization to images not covered in its pre-training dataset. For instance, although CLIP learns a capable OCR system, when evaluated on handwritten digits from the MNIST dataset, zero-shot CLIP only achieves 88% accuracy, well below the 99.75% of humans on the dataset.

4. Finally, we’ve observed that CLIP’s zero-shot classifiers can be sensitive to wording or phrasing and sometimes require trial and error “prompt engineering” to perform well.

StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery

Or Patashnik, Zongze Wu et al.

Tel-Aviv Univ, Hebrew Univ, Adobe Research

arXiv 31, Mar 2021

Presentor : Taeu

Task : Text-Driven Image Manipulation

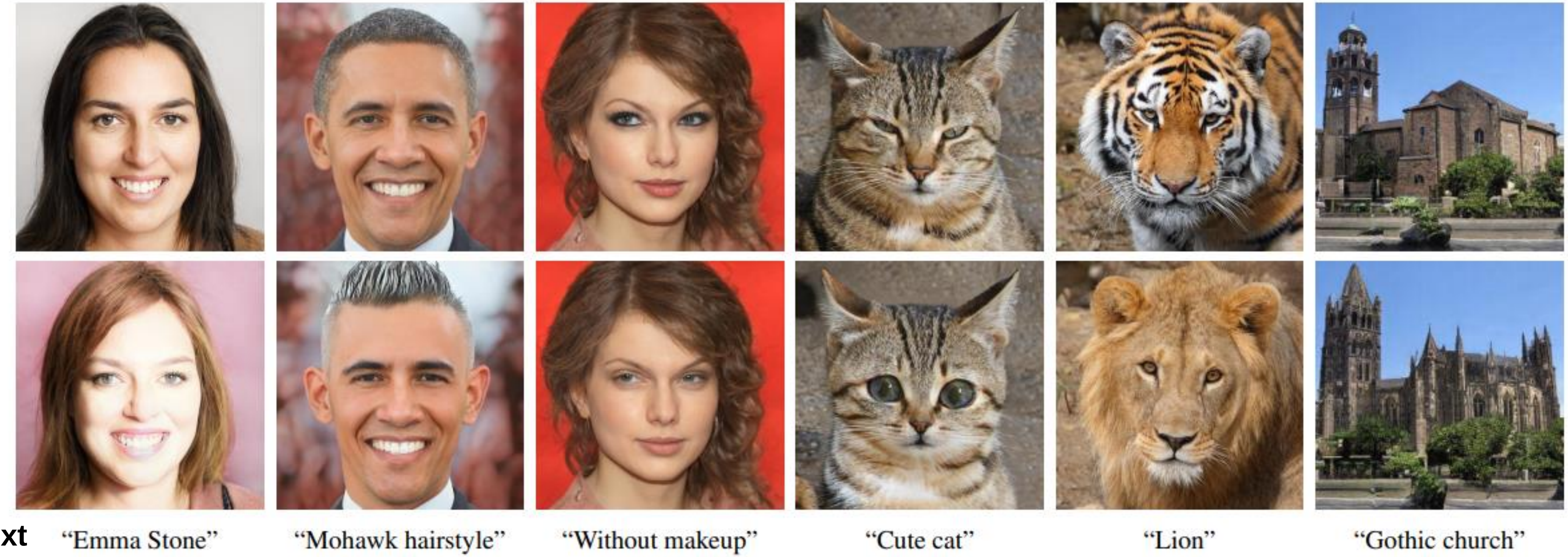


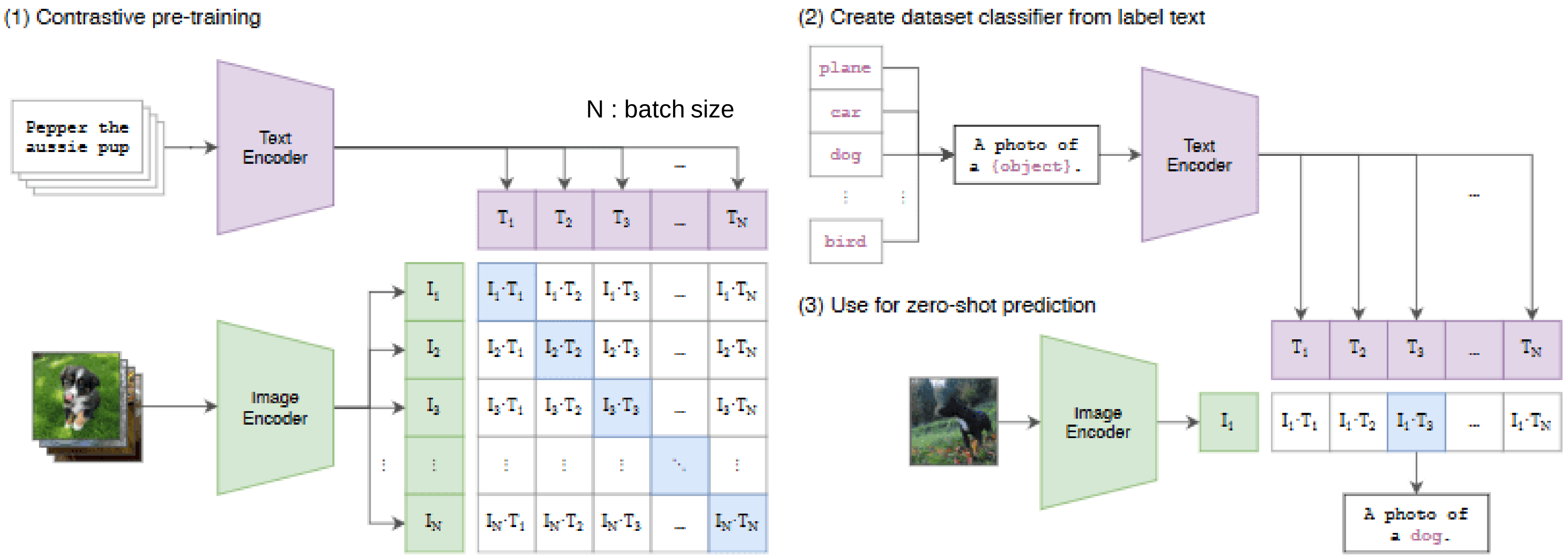
Figure 1. Examples of text-driven manipulations using StyleCLIP. Top row: input images; Bottom row: our manipulated results. The text prompt used to drive each manipulation appears under each column.

https://www.youtube.com/watch?v=5icI0NgALnQ

StyleCLIP : Text-Driven Image Manipulation

- **Contribution1** : Text-guided latent optimization, where a **CLIP** model is used as a loss network. This is the most versatile approach, but it requires a **few minutes of optimization** to apply a manipulation to an image.
- **Contribution2** : A **latent residual mapper**, trained for a specific text prompt. Given a starting point in latent space (the input image to be manipulated), the mapper yields a local step in latent space.
- **Contribution3** : A method for mapping a text prompt into an **input-agnostic (global) direction** in StyleGAN's style space, providing control over the manipulation strength as well as the degree of disentanglement.

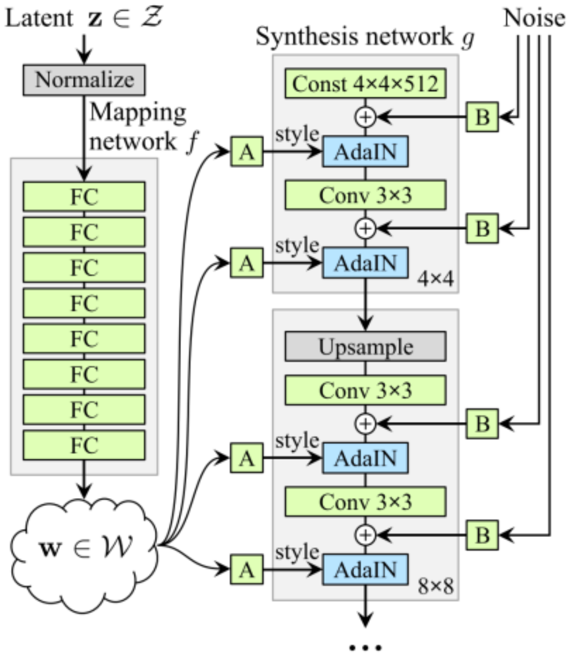
Background 1. CLIP



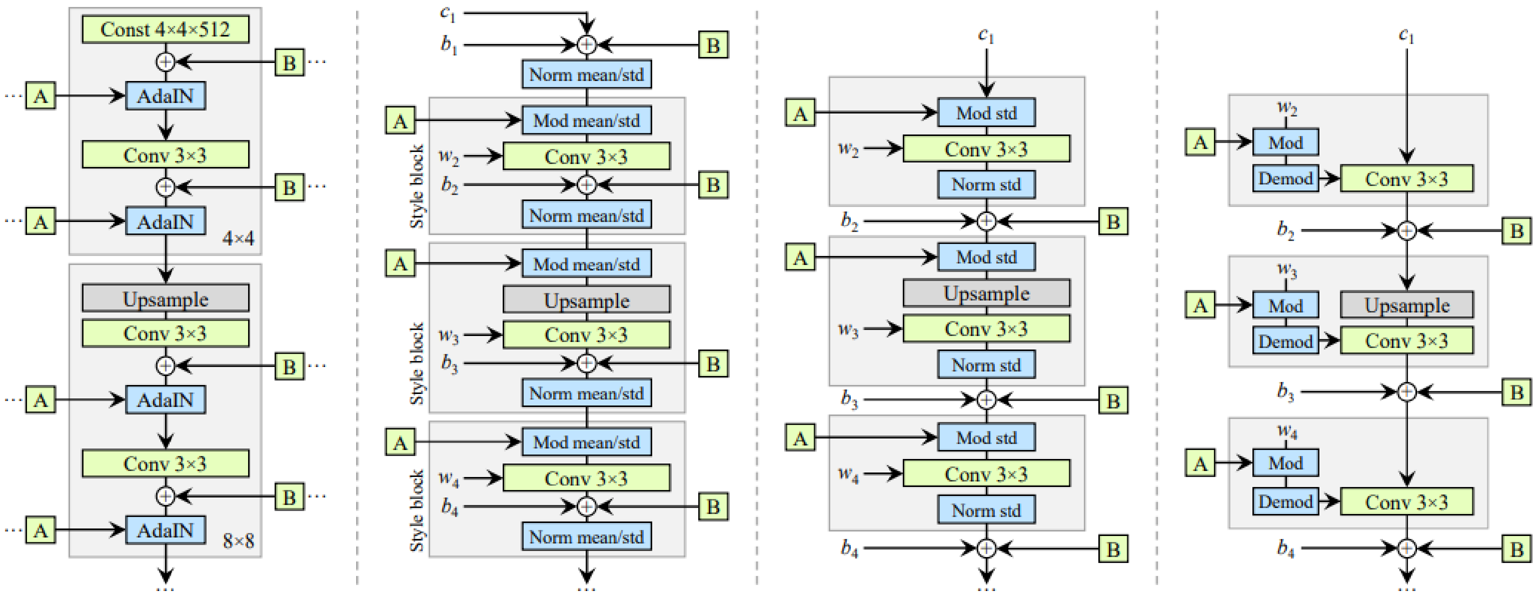
(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
N : batch size
T_1 T_2 T_3 … T_N
Image Encoder
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ … $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ … $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ … $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ … $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ … $I_1 \cdot T_N$
A photo of a dog.

- Contrastive Language-Image Pre-Training (CLIP) is a learning method developed by OpenAI that enables models to learn visual concepts from natural language supervision.
- In the multi-modal embedding space, the image encoder and text encoder are jointly trained to maximize the cosine similarity of the image and text embeddings of the real pairs in the batch.

Background 2. StyleGAN



(b) Style-based generator



(a) StyleGAN

(b) StyleGAN (detailed)

(c) Revised architecture

(d) Weight demodulation

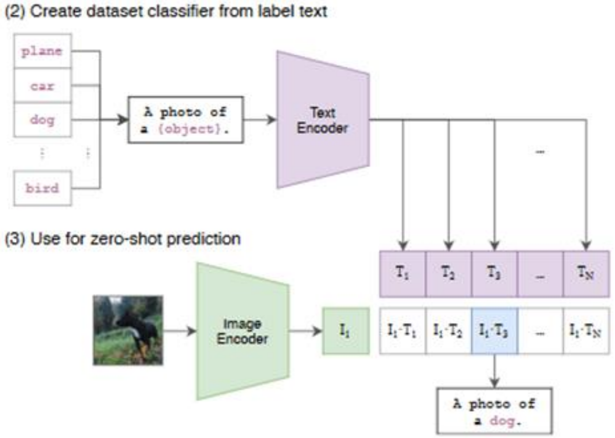
- Recently, style-based designs have become increasingly popular
  - StyleGAN uses AdaIN to modulate channel-wise means and variances
  - StyleGAN2 controls channel-wise variances by modulating the weights of the convolution kernels

* Recommend to Refer to **2021/05/24 DAVIAN Paper Seminar presented by Seunghwan Choi**

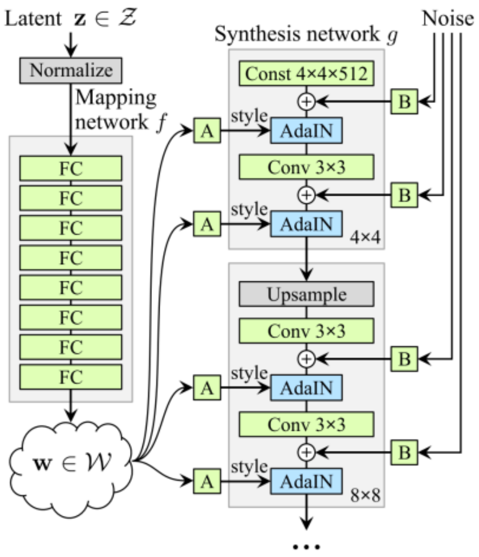
Overview of the Method

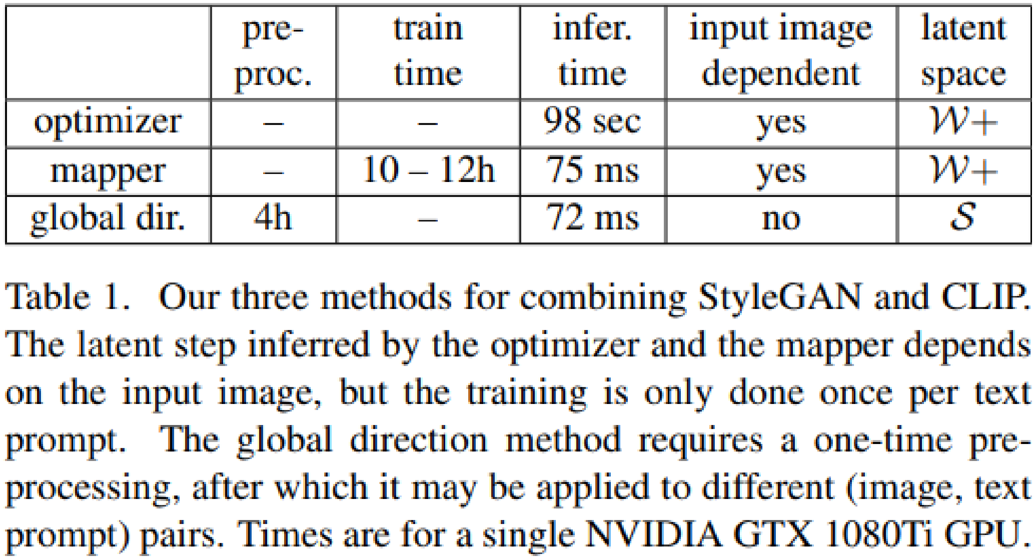
- **[1] Latent Optimization** :
  - A simple approach for leveraging CLIP to guide image manipulation is through direct latent code optimization.
- **[2] Latent Mapper** :
  - A dedicated optimization for each (source image, text prompt) pair.
- **[3] Global directions** :
  - Mapping a text prompt into a single, global direction in StyleGAN’s style space S, which has been shown to be more disentangled than other latent spaces.

CLIP



StyleGAN





	pre-proc.	train time	infer. time	input image dependent	latent space
optimizer	–	–	98 sec	yes	$\mathcal{W}+$
mapper	–	10 – 12h	75 ms	yes	$\mathcal{W}+$
global dir.	4h	–	72 ms	no	$\mathcal{S}$

Table 1. Our three methods for combining StyleGAN and CLIP. The latent step inferred by the optimizer and the mapper depends on the input image, but the training is only done once per text prompt. The global direction method requires a one-time pre-processing, after which it may be applied to different (image, text prompt) pairs. Times are for a single NVIDIA GTX 1080Ti GPU.

1. Latent Optimization

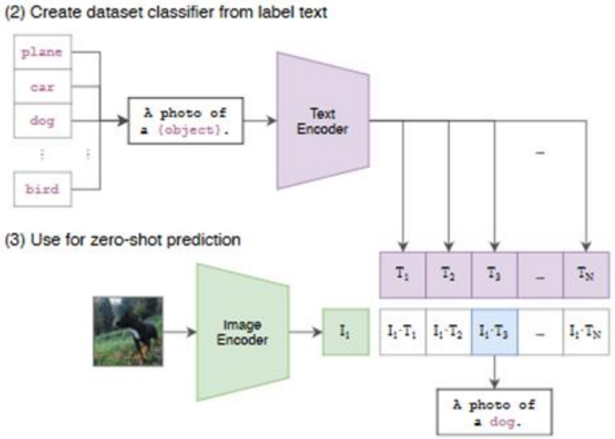
	pre-proc.	train time	infer. time	input image dependent	latent space
optimizer	–	–	98 sec	yes	$\mathcal{W}+$
mapper	–	10 – 12h	75 ms	yes	$\mathcal{W}+$
global dir.	4h	–	72 ms	no	$\mathcal{S}$

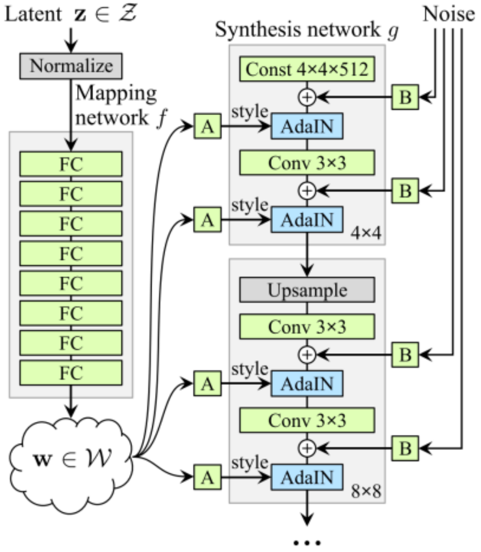
- A source latent code **ws** ∈ W+, latent code **w** which will be updated
- Text prompt **t**
- **G** is a pretrained StyleGAN generator
- **D_CLIP** is the cosine distance between the CLIP embeddings of its two arguments
- Similarity to the input image is controlled by the L2 distance in latent space, and by the identity loss where R is a pretrained ArcFace netework.
- <·, ·> computes the cosine similarity between it's arguments

$$\arg\min_{w\in\mathcal{W}+} D_{\text{CLIP}}(G(w),t) + \lambda_{\text{L2}}\left\|w-w_s\right\|_2 + \lambda_{\text{ID}}\mathcal{L}_{\text{ID}}(w),$$

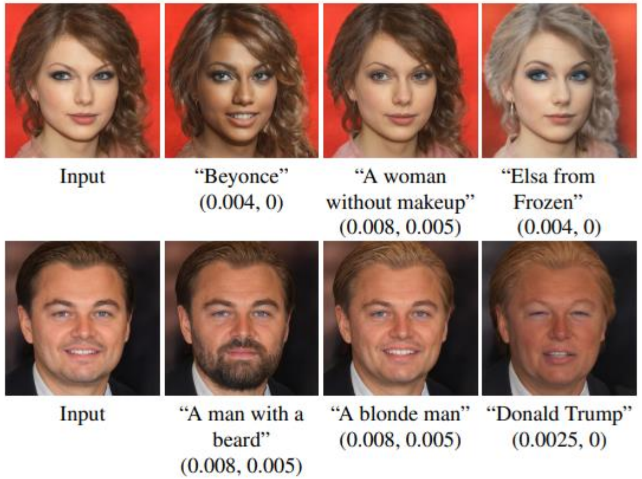
$$\mathcal{L}_{\text{ID}}(w) = 1 - \left\langle R(G(w_s)), R(G(w))\right\rangle,$$

- 200-300 iterations for the optimization. → **several minutes** to edit a single image.


(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I_1
T_1 T_2 T_3 ... T_N
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ ... $I_1{\cdot}T_N$
A photo of a dog.



(b) Style-based generator


Input

"Beyonce" (0.004, 0)

"A woman without makeup" (0.008, 0.005)

"Elsa from Frozen" (0.004, 0)

Input

"A man with a beard" (0.008, 0.005)

"A blonde man" (0.008, 0.005)

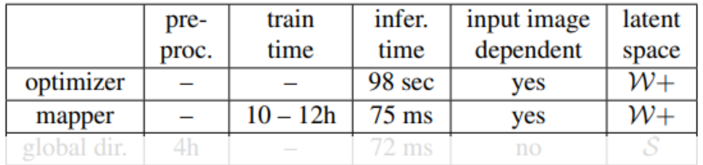
"Donald Trump" (0.0025, 0)

Figure 3. Edits of real celebrity portraits obtained by latent optimization. The driving text prompt and the $(\lambda_{\text{L2}}, \lambda_{\text{ID}})$ parameters for each edit are indicated under the corresponding result.

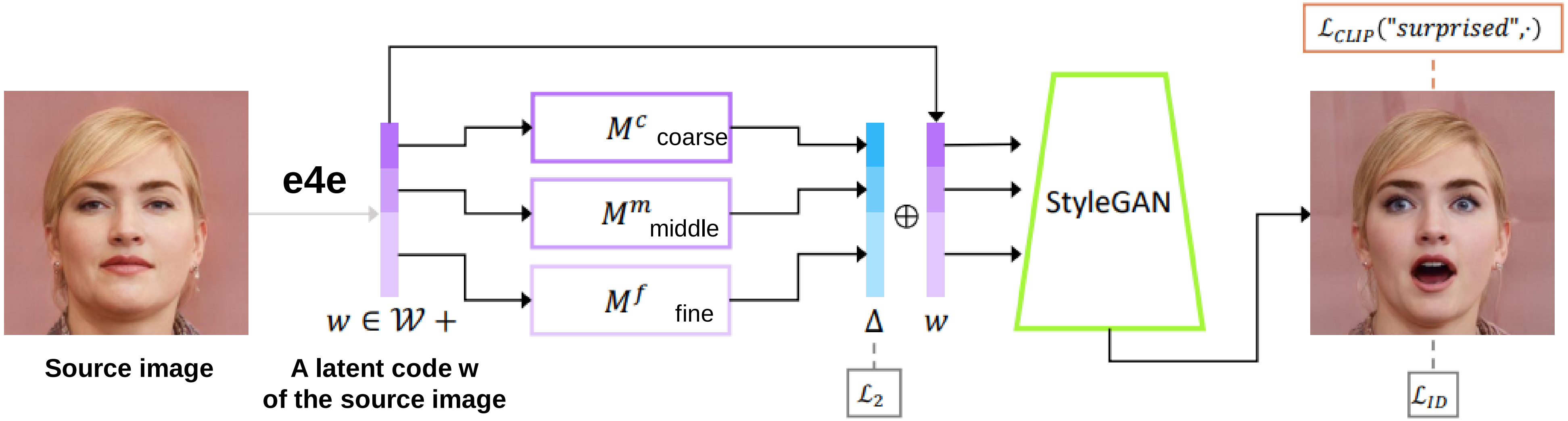
*The input images were inverted by **e4e**

***e4e** : Designing an Encoder for StyleGAN Image Manipulation

2. Latent Mapper

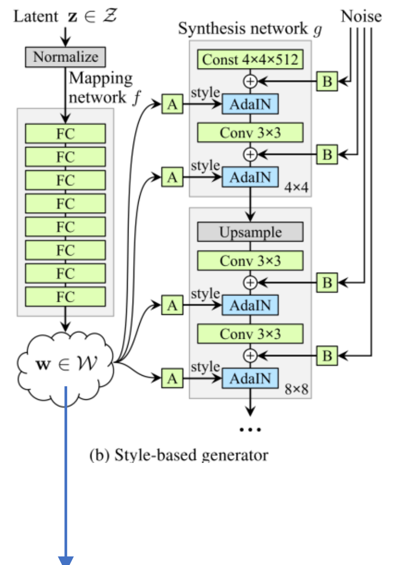


	pre-proc.	train time	infer. time	input image dependent	latent space
optimizer	–	–	98 sec	yes	$\mathcal{W}+$
mapper	–	10 – 12h	75 ms	yes	$\mathcal{W}+$
global dir.	4h	–	72 ms	no	$\mathcal{S}$



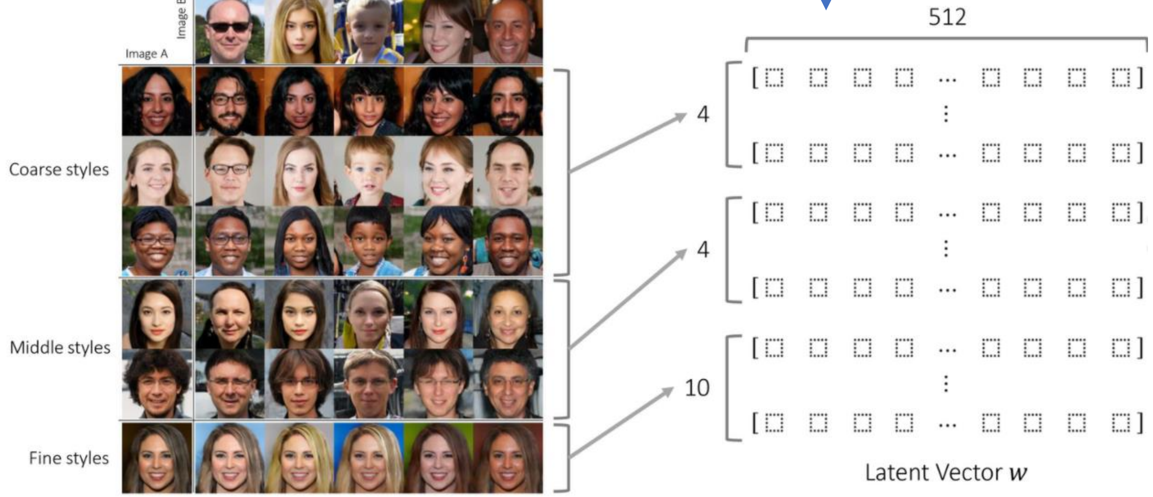
Source image

A latent code w of the source image

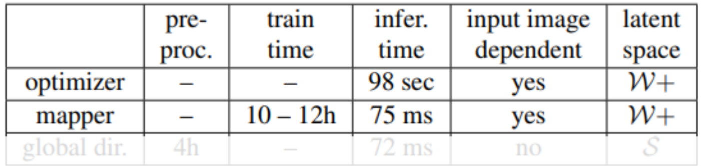


(b) Style-based generator

- Text-guided mapper(using the text prompt “surprised”, in this example) with 4 FC layers. $M_t(w) = (M_t^c(w_c), M_t^m(w_m), M_t^f(w_f)).$
- Loss $\mathcal{L}(w) = \mathcal{L}_{\text{CLIP}}(w) + \lambda_{L2} \|M_t(w)\|_2 + \lambda_{\text{ID}}\mathcal{L}_{\text{ID}}(w).$
- The CLIP Loss : $\mathcal{L}_{\text{CLIP}}(w) = D_{\text{CLIP}}(G(w + M_t(w)), t)$
- Lambda : $\lambda_{\text{L2}} = 0.8, \lambda_{\text{ID}} = 0.1$
- Need **n Text promt specific models** to synthesize the image (surprised model, angry model, Beyonce model etc.)



2. Latent Mapper



	pre-proc.	train time	infer. time	input image dependent	latent space
optimizer	–	–	98 sec	yes	$\mathcal{W}+$
mapper	–	10 – 12h	75 ms	yes	$\mathcal{W}+$
global dir.	4h	–	72 ms	no	$\mathcal{S}$

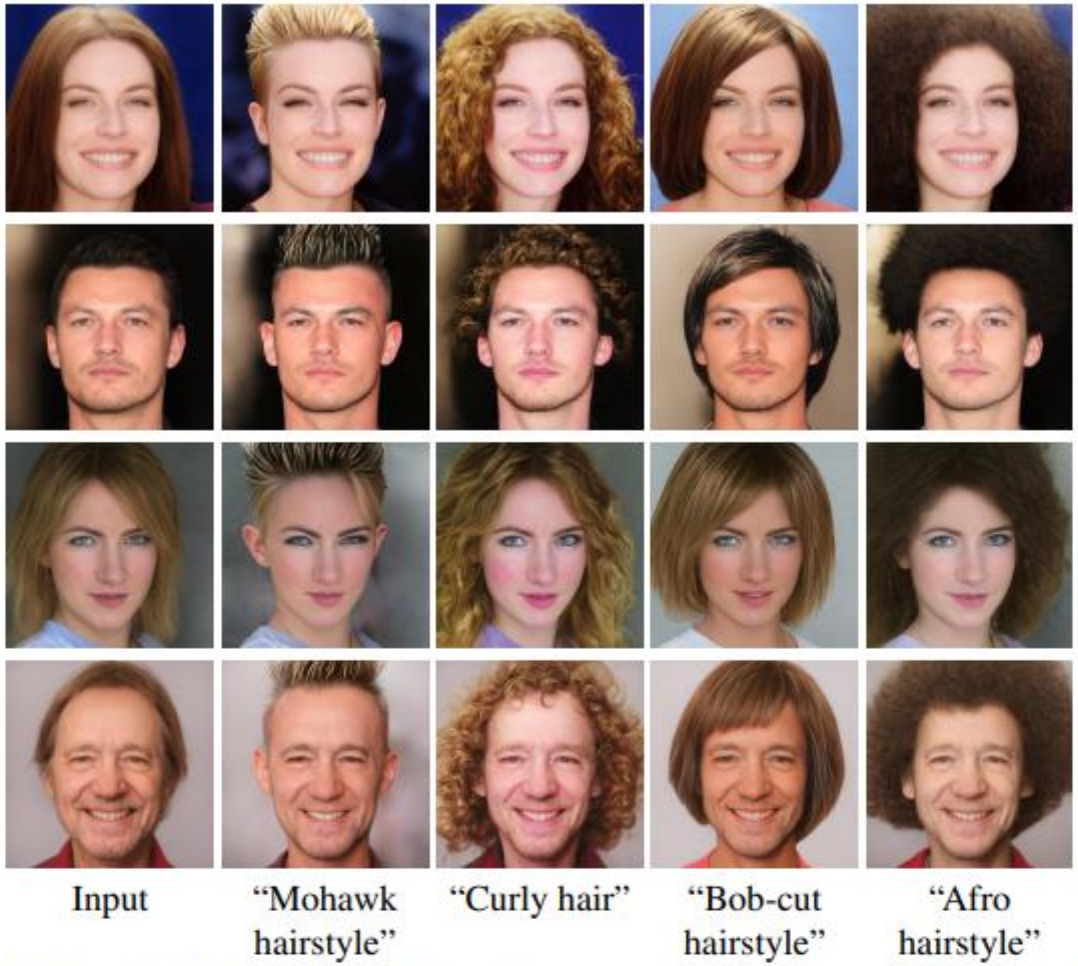


Figure 4. Hair style edits using our mapper. The driving text prompts are indicated below each column. All input images are inversions of real images.

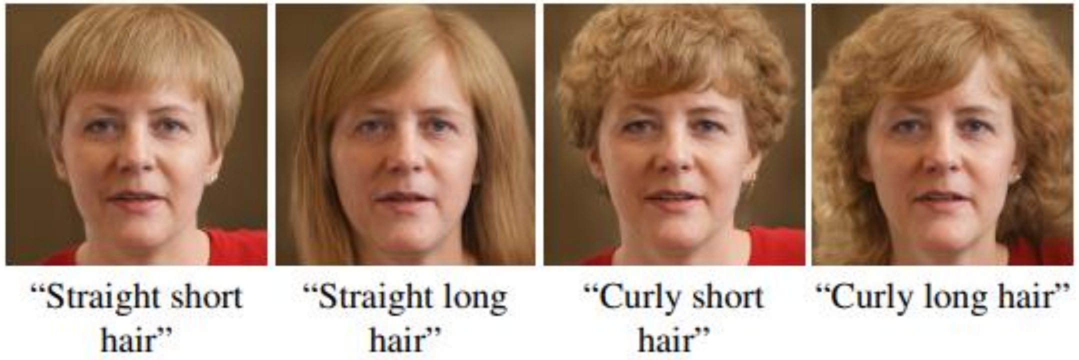
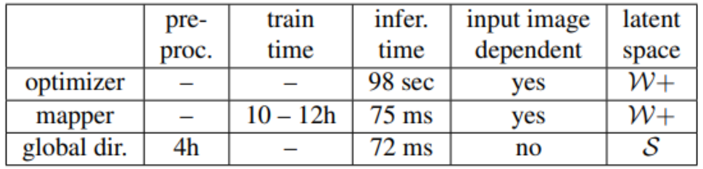


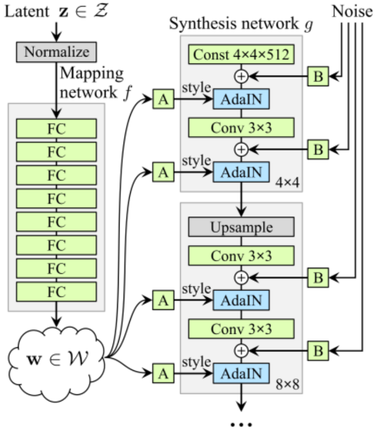
Figure 5. Controlling more than one attribute with a single mapper. The driving text for each mapper is indicated below each column.

3. Global Directions

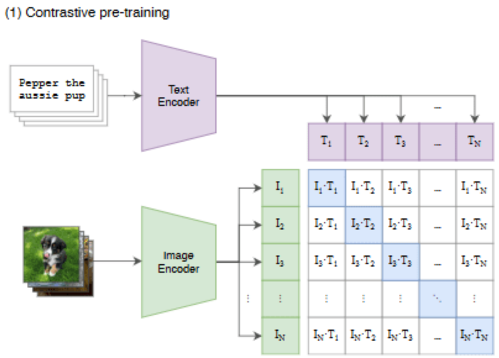


	pre-proc.	train time	infer. time	input image dependent	latent space
optimizer	–	–	98 sec	yes	$\mathcal{W}+$
mapper	–	10 – 12h	75 ms	yes	$\mathcal{W}+$
global dir.	4h	–	72 ms	no	$\mathcal{S}$

- While the latent mapper allows fast inference time, we find that it sometimes falls short when a fine-grained disentangled manipulation is desired.
- A method for mapping a text prompt into a single, global direction in StyleGAN's *style space S, which has been shown to be more disentangled than other latent spaces.
- Let **s** ∈ S denote a style code, and **G(s)** the corresponding generated image.
- We seek a manipulation direction ∆**s**, such that G(s + **α**∆s) yields an image where that attribute is introduced or amplified, without significantly affecting other attributes.
- The manipulation strength is controlled by **α**.
- Our high-level idea is to first use the CLIP text encoder to obtain a vector ∆**t** in CLIP's joint language-image embedding and then map this vector into a manipulation direction ∆**s** in S. (A stable ∆t is obtained from natural language)
- We distinguish between two manifolds(the manifold of image embeddings space and the manifold of text embeddings space), because there is no one-to-one mapping between them: an image may contain a large number of visual attributes, which can hardly be comprehensively described by a single text sentence; conversely, a given sentence may describe many different images.



(b) Style-based generator


(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 … T_N
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ … $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ … $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ … $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ … $I_N \cdot T_N$

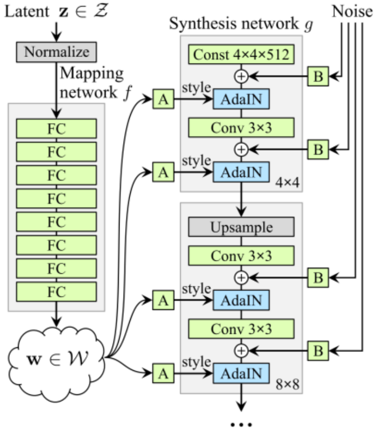
* Recommend to Refer to **2021/05/24 DAVIAN Paper Seminar presented by Seunghwan Choi**

3. Global Directions

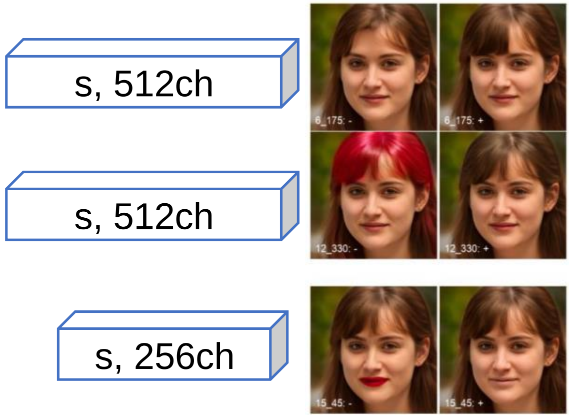
	pre-proc.	train time	infer. time	input image dependent	latent space
optimizer	–	–	98 sec	yes	$\mathcal{W}+$
mapper	–	10 – 12h	75 ms	yes	$\mathcal{W}+$
global dir.	4h	–	72 ms	no	$\mathcal{S}$

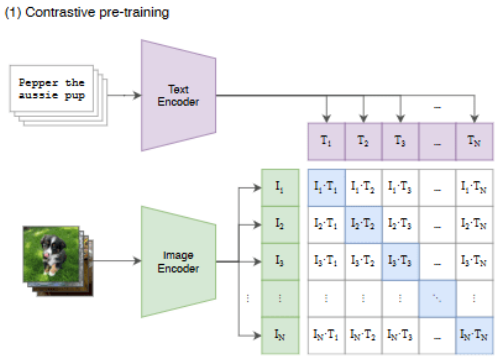
- Given a pair of images, G(s) and G(s+αΔs), we denote their I(Image) embeddings by i and i + Δi, respectively.
- Given a natural language instruction encoded as Δt, and assuming collinearity between Δt and Δi, we can determine a manipulation direction Δs by assessing the relevance of each channel in S to the direction Δi.
- From natural language to Δt
  - **Prompt engineering** (a photo of a { } , a cropped photo of the { }, a painting of a { } … → average) to reduce text embedding noise.
  - Prompt engineering for neutral class and target attirbutes (a photo of {car}, a photo of {sports car}), get normalized difference between them.
- Channelwise relevance
  - Goal : **Δs** → **Δi** → Δt assuming that collinear between Δt and Δi,
  - ΔI_c : CLIP space direction between G(s ±**αΔs_c). (α = 5)**
  - **Δs_c :** is a zero vector, except its c coordinate, which is set to the standard deviation of the ch.
  - We generate a collection of style codes s ∈ S, and perturb only

    the c channel of each style code by adding a negative and a positive value.



(b) Style-based generator





$$R_c(\Delta i) = \mathbb{E}_{s\in\mathcal{S}}\{\Delta i_c \cdot \Delta i\} \longrightarrow \Delta s = \begin{cases} \Delta i_c \cdot \Delta i & \text{if } |\Delta i_c \cdot \Delta i| \geq \beta \\ 0 & \text{otherwise} \end{cases}$$

100 image pairs

The layer index, channel index, and the direction change

The degree of disentanglement

* Recommend to Refer to **2021/05/24 DAVIAN Paper Seminar presented by Seunghwan Choi**

3. Global Directions

	pre-proc.	train time	infer. time	input image dependent	latent space
optimizer	–	–	98 sec	yes	$\mathcal{W}+$
mapper	–	10 – 12h	75 ms	yes	$\mathcal{W}+$
global dir.	4h	–	72 ms	no	$\mathcal{S}$

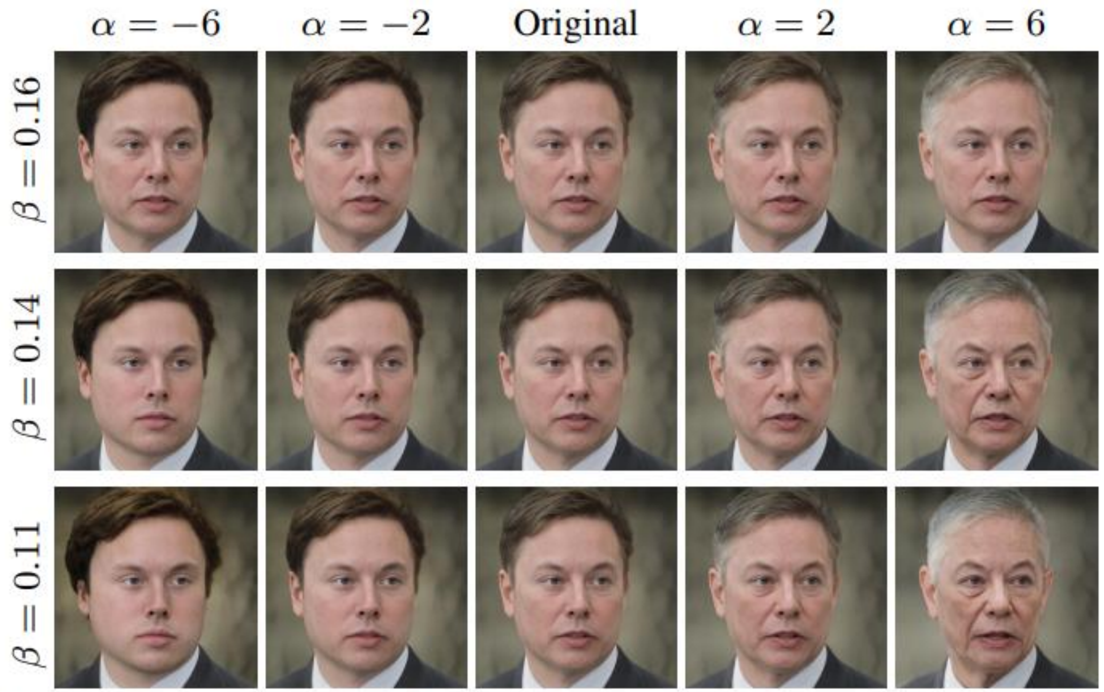


Figure 6. Image manipulation driven by the prompt “grey hair” for different manipulation strengths and disentanglement thresholds. Moving along the Δs direction, causes the hair color to become more grey, while steps in the $-\Delta s$ direction yields darker hair. The effect becomes stronger as the strength α increases. When the disentanglement threshold β is high, only the hair color is affected, and as β is lowered, additional correlated attributes, such as wrinkles and the shape of the face are affected as well.

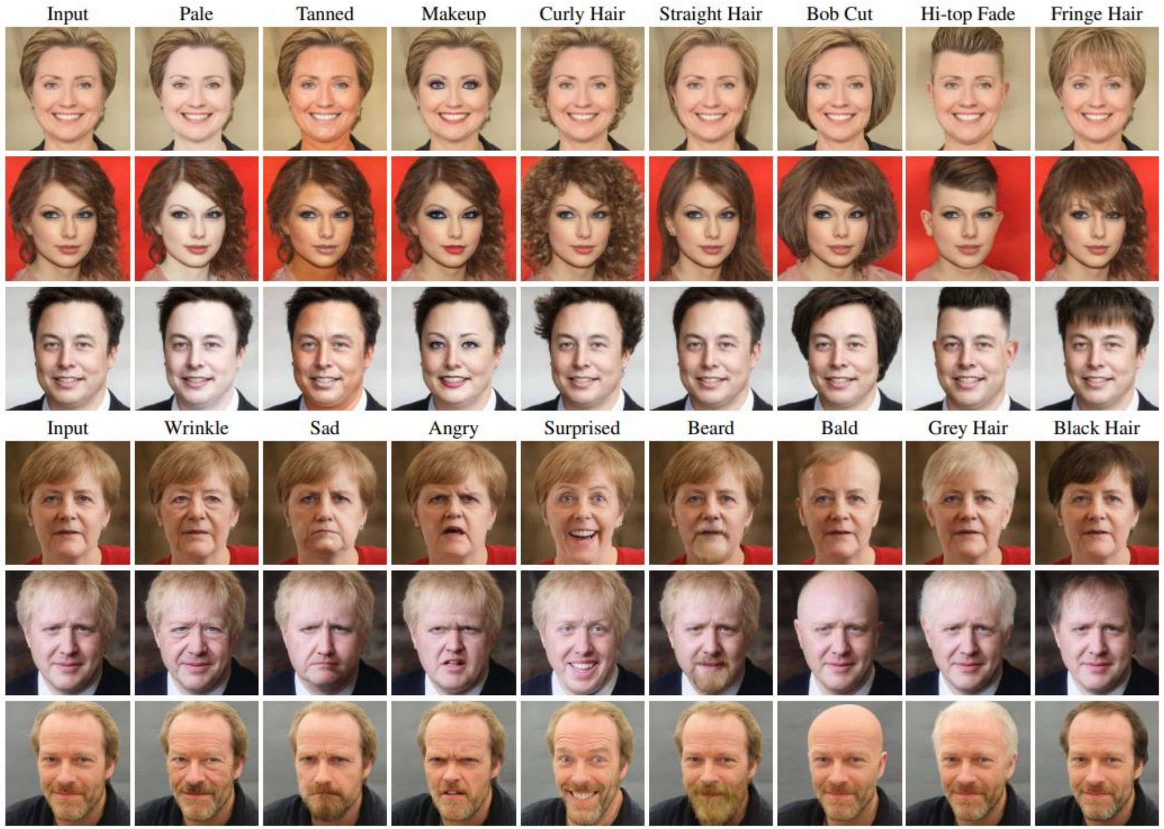


Figure 7. A variety of edits along global text-driven manipulation directions, demonstrated on portraits of celebrities. Edits are performed using StyleGAN2 pretrained on FFHQ [18]. The inputs are real images, embedded in $\mathcal{W}+$ space using the e4e encoder [46]. The target attribute used in the text prompt is indicated above each column.

G(s+αΔs)

$$\Delta s = \begin{cases} \Delta i_c \cdot \Delta i & \text{if } |\Delta i_c \cdot \Delta i| \geq \beta \\ 0 & \text{otherwise} \end{cases}$$

Results

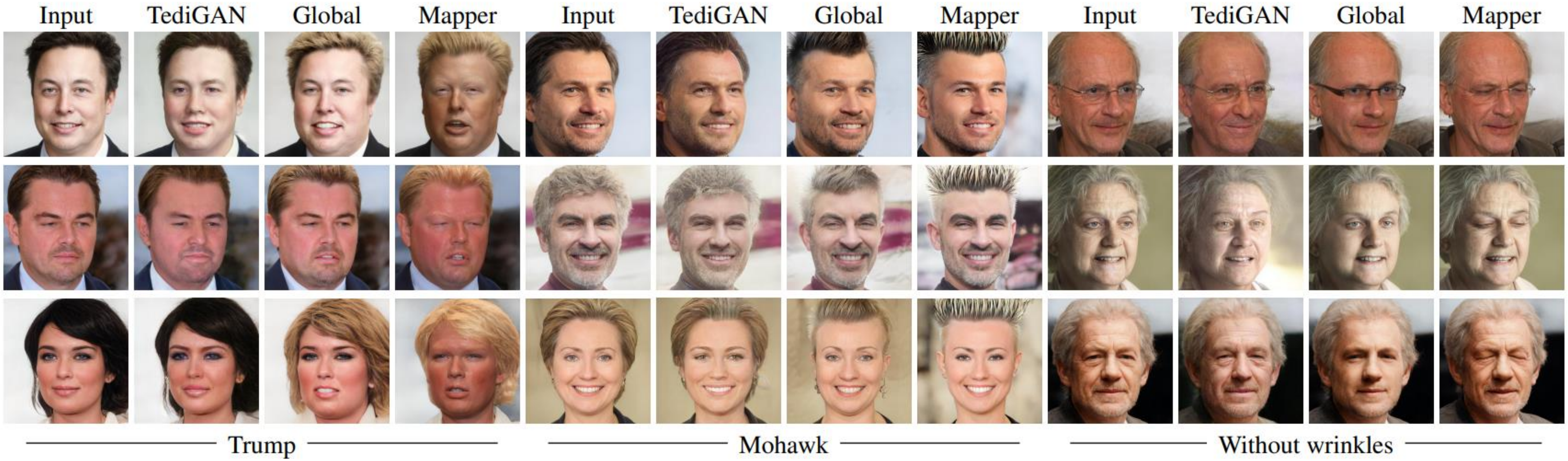


Figure 9. We compare three methods that utilize StyleGAN and CLIP using three different kinds of attributes.

Results

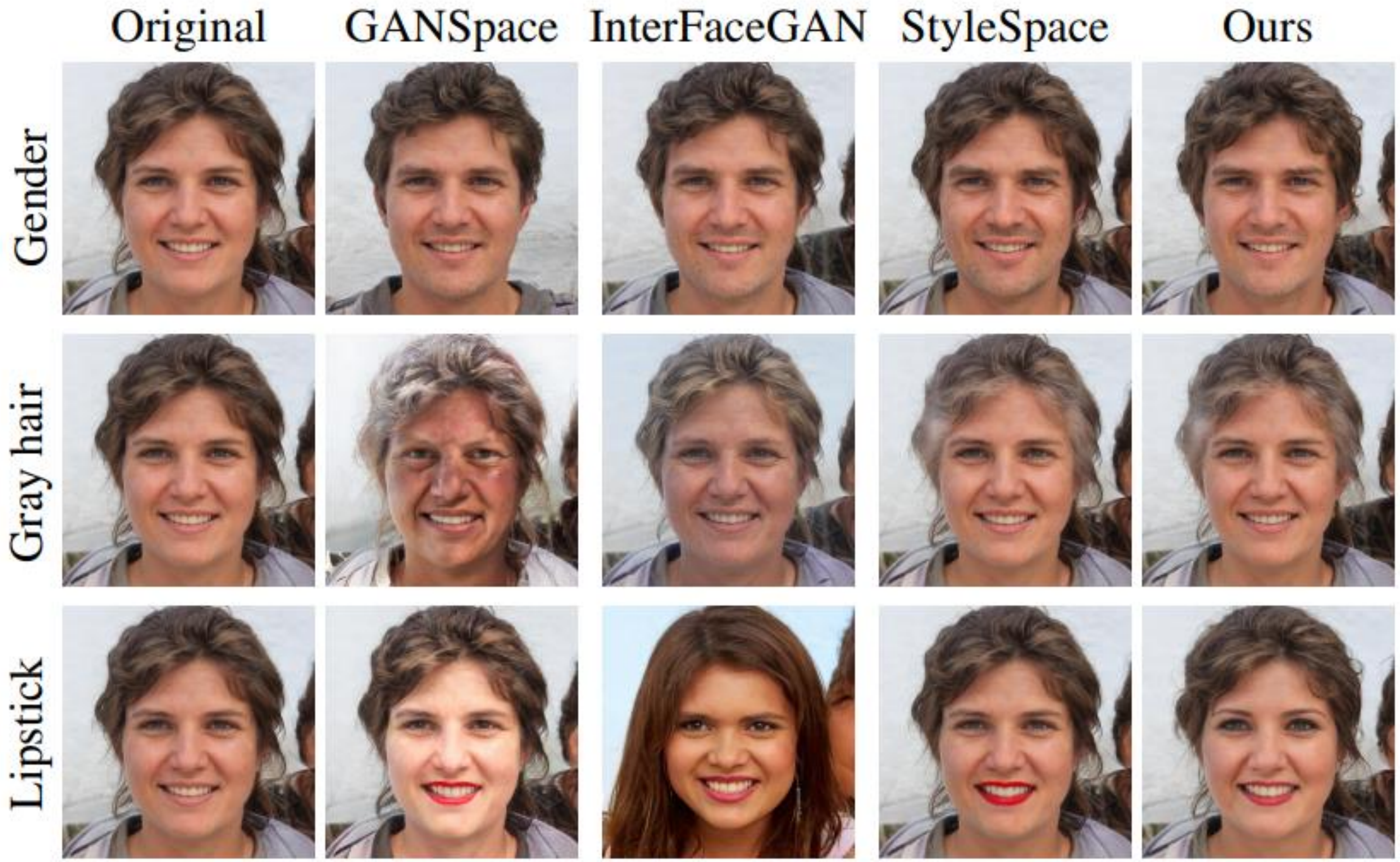


Figure 10. Comparison with state-of-the-art methods using the same amount of manipulation according to a pretrained attribute classifier.

* The comparison only examines the attributes which all of the compared methods are able to manipulate (Gender, Grey hair, and Lipstick), and thus it does not include the many novel manipulations enabled by our approach.

Results

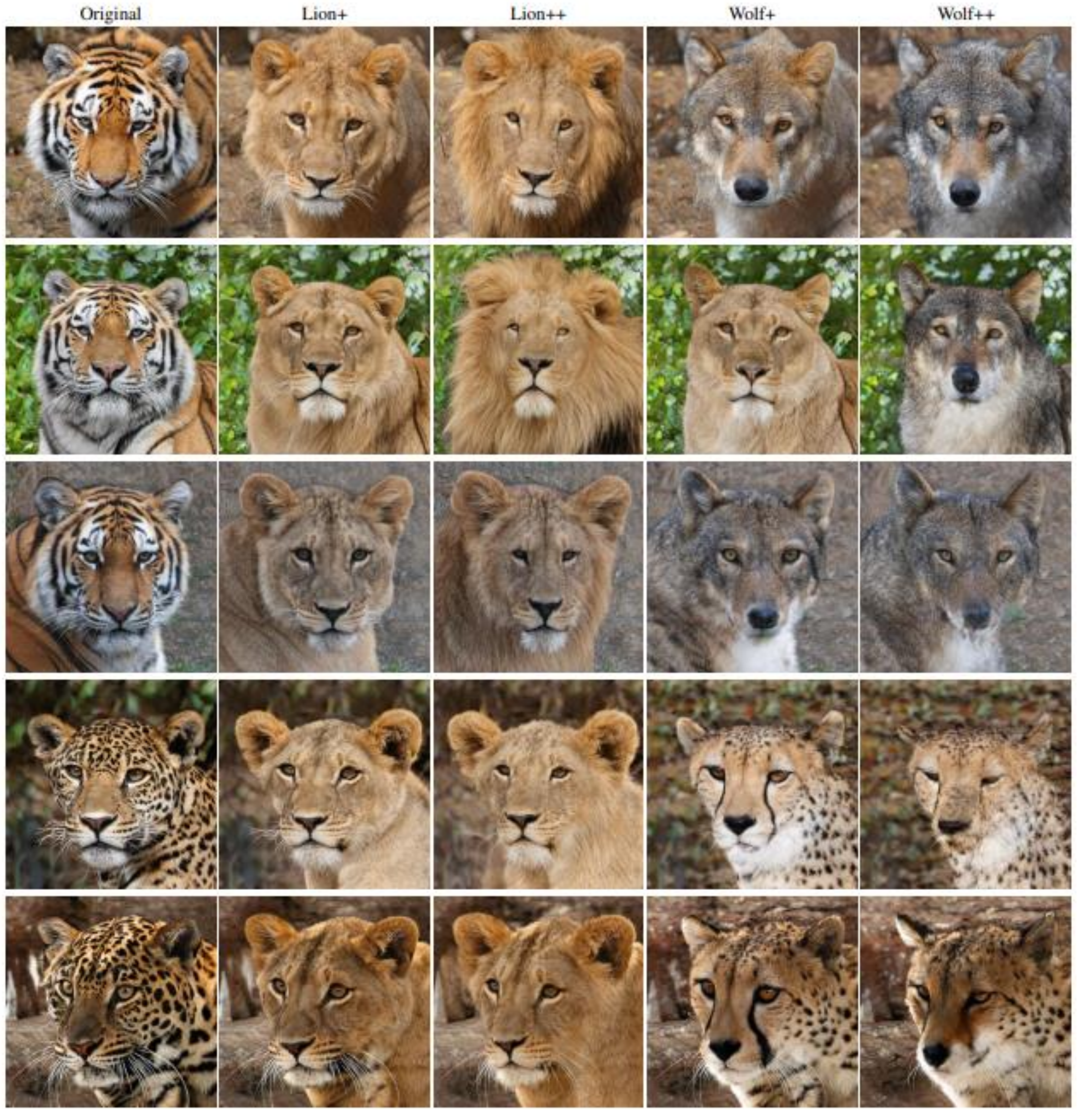


Figure 24. Drastic manipulations in visually diverse datasets are sometimes difficult to achieve using our global directions. Here we use StyleGAN-ada pretrained on AFHQ wild, which contains wolves, lions, tigers and foxes. There is a smaller domain gap between tigers and lions, which mainly involves color and texture transformations. However, there is a larger domain gap between tigers and wolves, which, in addition to color and texture transformations, also involves more drastic shape deformations. This figure demonstrates that our global directions method is more successful in transforming tigers into lions, while failing in some cases to transform tigers to wolves. The "+" and "++" indicate medium and strong manipulation strength, respectively.

End

- 논문 : https://arxiv.org/abs/2103.17249
- Code : https://github.com/orpatashnik/StyleCLIP
- Related work
  - StyleGAN : https://arxiv.org/abs/1812.04948
  - StyleGANv2 : https://arxiv.org/abs/1912.04958
  - CLIP : https://arxiv.org/abs/2103.00020
  - StyleSpace Analysis : https://arxiv.org/abs/2011.12799