Reading and Writing: Discriminative and Generative Modeling for Self-Supervised Text Recognition

[ACMMM, 2022]

2023. 05. 26.

김성수

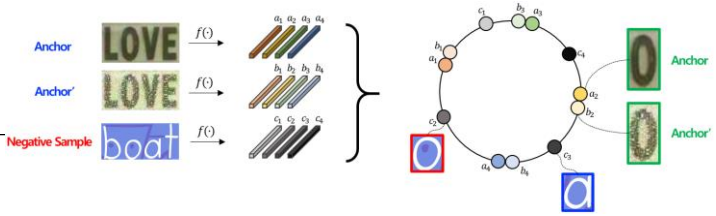
Data **M**ining and **Q**uality **A**nalytics

Data Mining Quality Analytics

고려대학교 KOREA UNIVERSITY

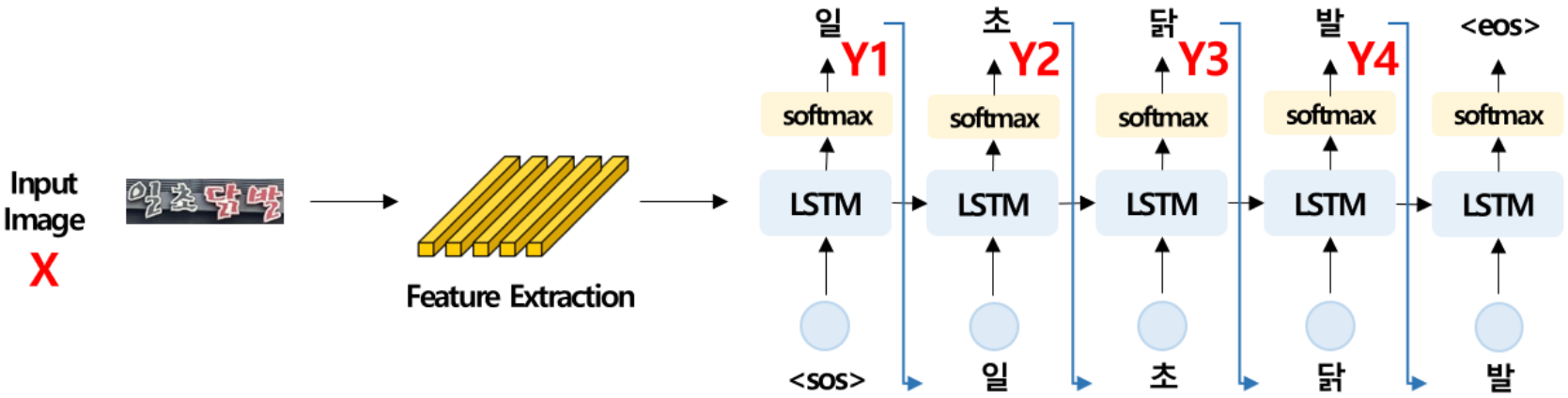
연구배경

- Task명



❖ **Task: Self-supervised Learning + Scene Text Recognition**

- SSL: Unlabeled 데이터를 활용하여 Feature Extractor의 성능 향상
- STR: 다양한 배경과 여러 형태의 글꼴이 존재하는 일상 이미지 내 문자를 인식하는 연구분야



연구배경

- 선행연구의 한계

❖ Limitation of Previous Research

- STR은 학습을 위한 Labeled 데이터가 부족
  - ✓ 이에 따라 합성 이미지를 활용하지만, 이는 Domain Gap이 존재하기에 일반화 성능 저하

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

연구배경

- 선행연구의 한계를 극복과정 (Overview of Research)

❖ Overcome the Limitation

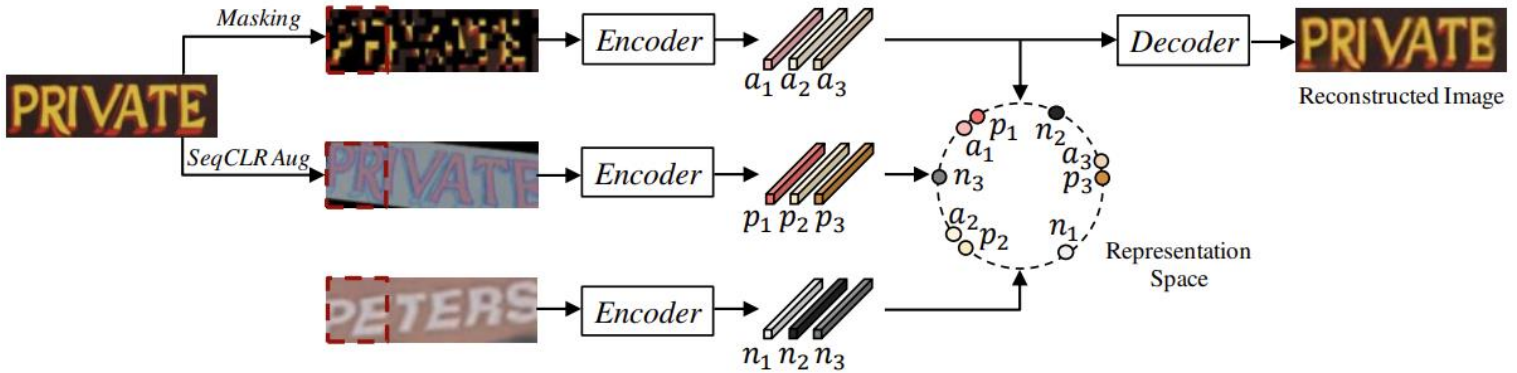
- Unlabeled 데이터를 활용하여 Labeled 데이터가 부족한 한계를 극복
- 이때, 자기지도학습 방법론 중 대조학습을 활용

연구배경

- 선행연구의 한계를 극복과정 (Overview of Research)

❖ Overcome the Limitation

- 사람은 이미지 내 문자열을 이해할 때, 읽는 행위와 쓰는 행위를 통해서 학습
- Step1. Reading (= Discriminative = Contrastive SSL)
- Step2. Writing (= Generative = Generative SSL)

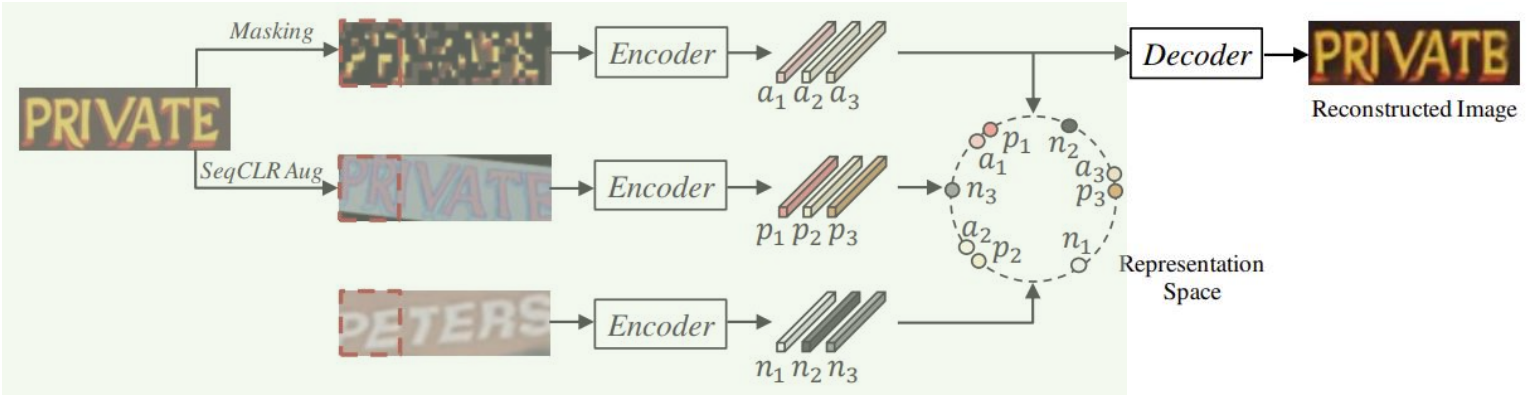


연구배경

- 선행연구의 한계를 극복과정 (Overview of Research)

❖ Overcome the Limitation

- 사람은 이미지 내 문자열을 이해할 때, 읽는 행위와 쓰는 행위를 통해서 학습
- Step1. Reading (= Discriminative = Contrastive SSL)
- Step2. Writing (= Generative = Generative SSL)



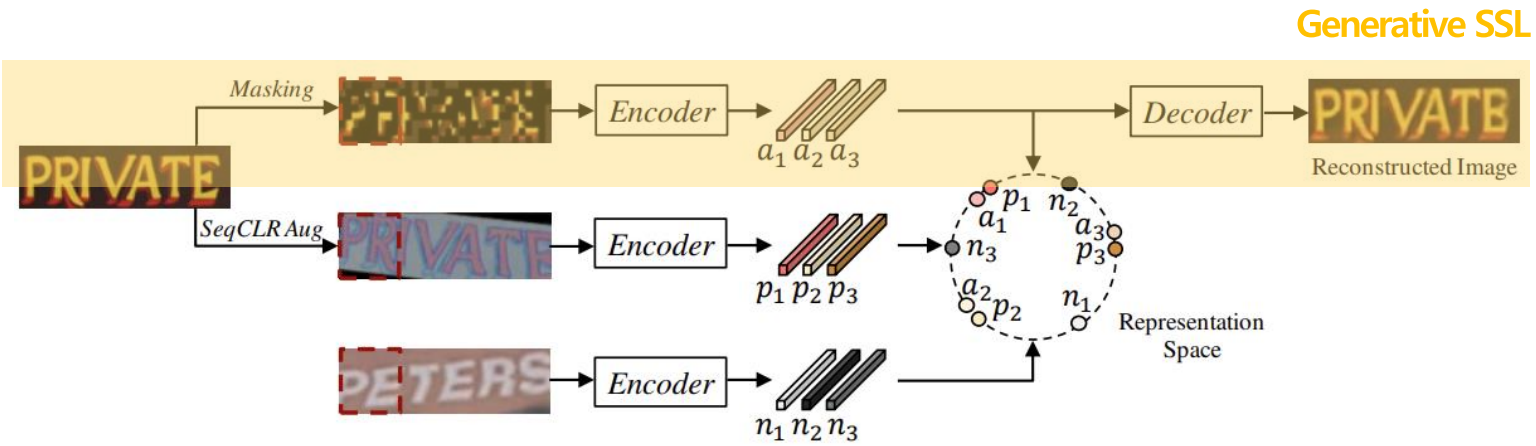
Contrastive SSL

연구배경

- 선행연구의 한계를 극복과정 (Overview of Research)

❖ Overcome the Limitation

- 사람은 이미지 내 문자열을 이해할 때, 읽는 행위와 쓰는 행위를 통해서 학습
- Step1. Reading (= Discriminative = Contrastive SSL): 다양한 각도에서 글자 간 다른 것을 식별
- Step2. Writing (= Generative = Generative SSL): 문자열을 직접 작성해보면서 학습

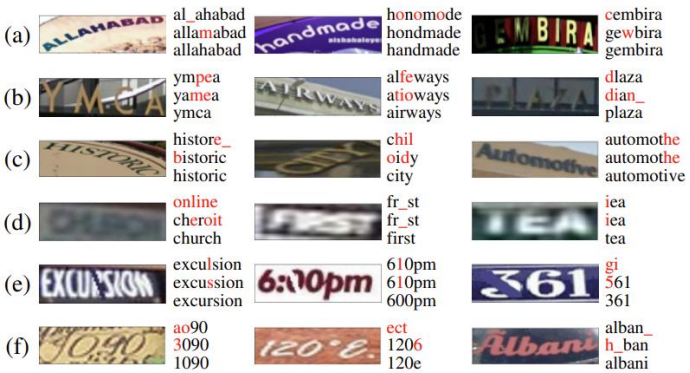


연구배경

- Contribution

❖ Contribution

- {STR+SSL}에 Generative 모델링을 처음으로 적용
- 기존 방법론들보다 성능이 크게 개선
- 다양한 Task에서 좋은 성능을 보임
  - ➢ Text Super-Resolution, Text Segmentation, Text Recognition



방법론

- 개요

❖ DiG: Discriminative and Generative Self-supervised Method

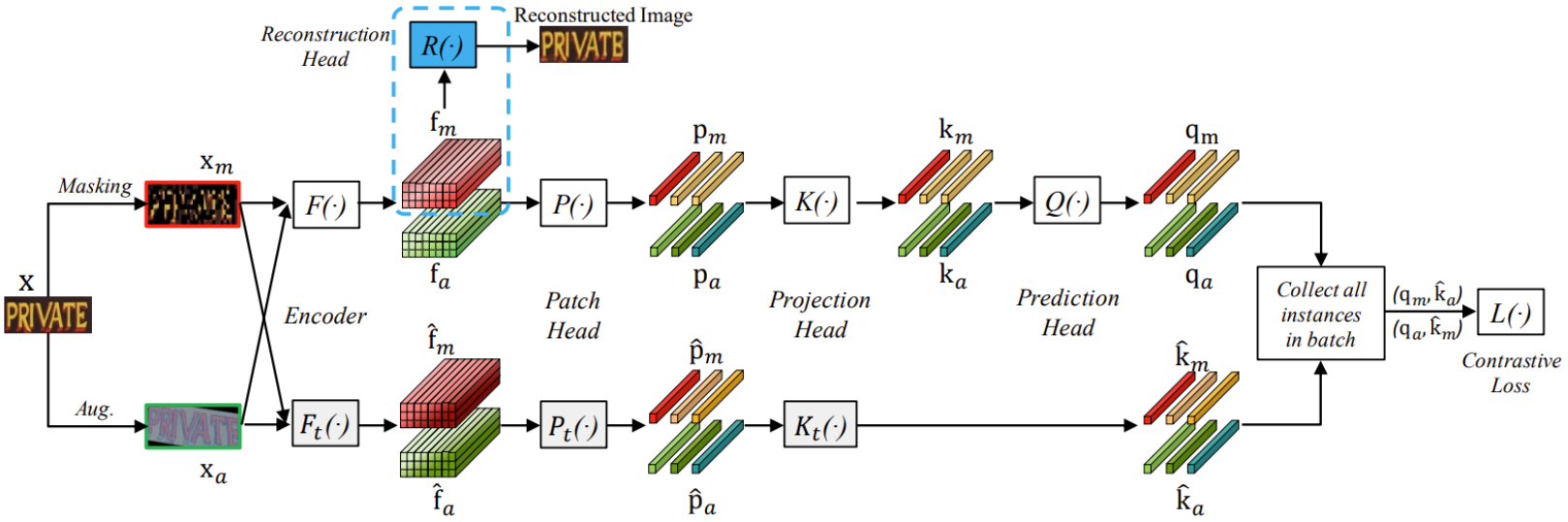
① 데이터 증강 및 Masking 수행

② Contrastive Loss 산출

③ Generative Loss 산출

④ 최종 Loss 산출

⑤ Online Network는 Backpropagation으로, Momentum Network는 EMA로 Update (BYOL style)

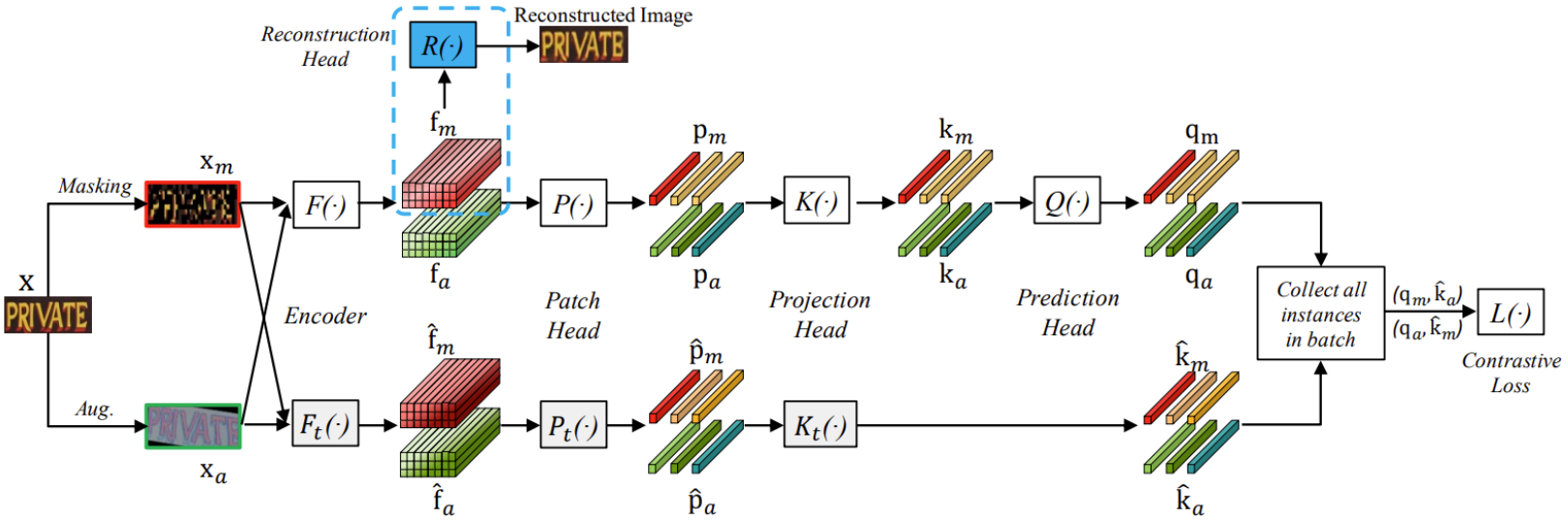


방법론

- DiG Detail

❖ Step1. 데이터 증강 및 Masking 수행

- 겹치지 않는 4x4 크기의 패치들을 활용
- 전체 이미지의 60%를 Masking
- 데이터 증강은 SeqCLR(CVPR, 2021)보다 강하게 활용

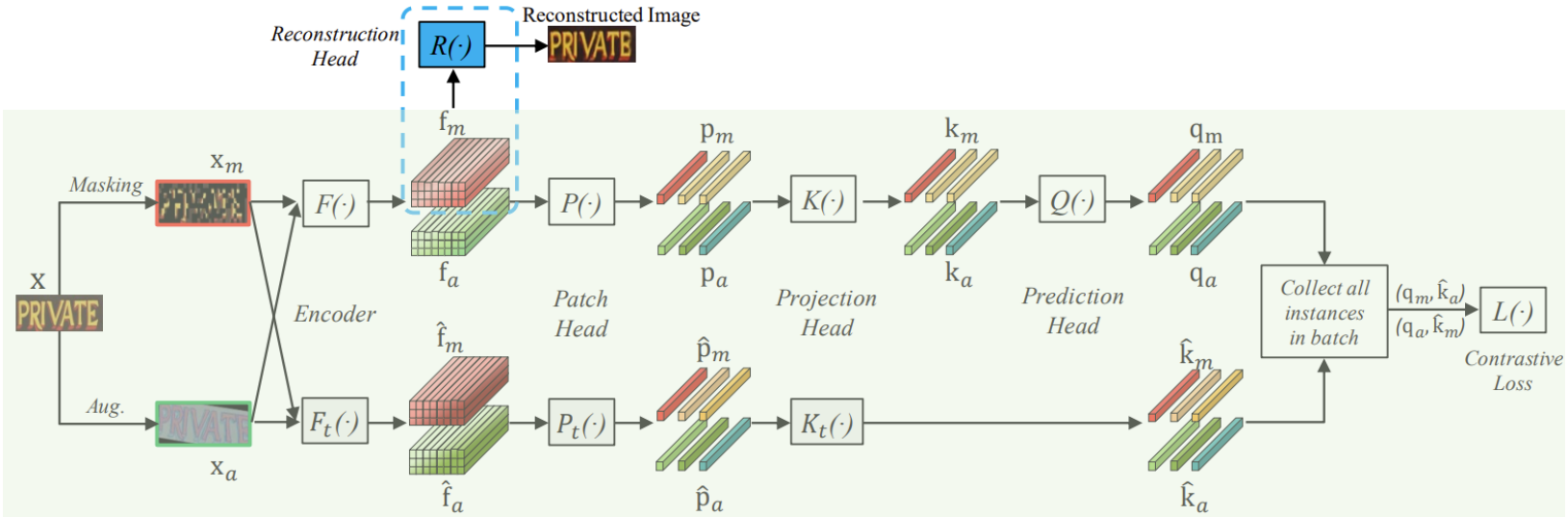


방법론

- DiG Detail

❖ Step2. Contrastive Loss 산출

- Encoder: ViT → Feature Extractor
- Patch head: Mapping Function (Adaptive Average Pooling)
- Projection Head: 3FC Layer + GELU + Layer Normalization
- Prediction Head: 2FC Layer + GELU + Layer Normalization



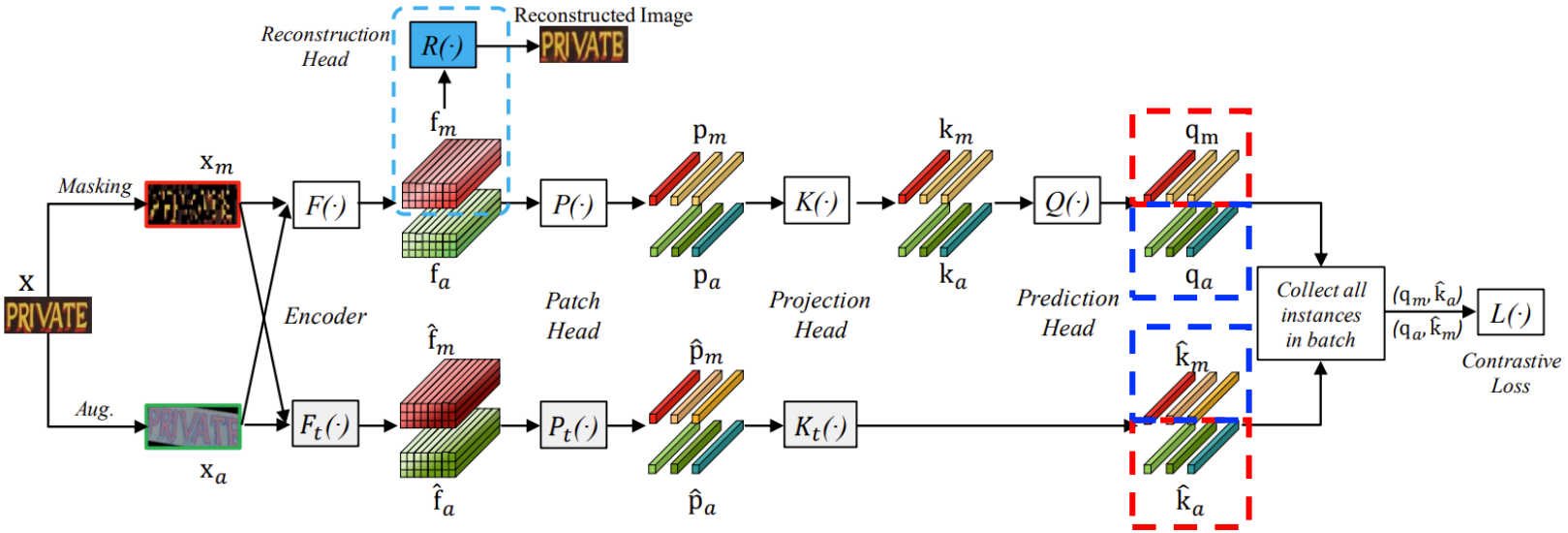
방법론

- DiG Detail

❖ Step2. Contrastive Loss 산출

$$L_c = -\log \frac{exp(q_m \cdot \hat{k}_a/\tau)}{exp(q_m \cdot \hat{k}_a/\tau) + \sum_{\hat{k}_a^-} exp(q_m \cdot \hat{k}_a^-/\tau)} - \log \frac{exp(q_a \cdot \hat{k}_m/\tau)}{exp(q_a \cdot \hat{k}_m/\tau) + \sum_{\hat{k}_m^-} exp(q_a \cdot \hat{k}_m^-/\tau)},$$

- 흔히 아는 대조학습 Loss(InfoNCE)와 유사한 형태
  - ➢ 특이점: 각 Masking과 Augmentation의 객체에서 나온 Output을 다르게 교차하여 대조학습



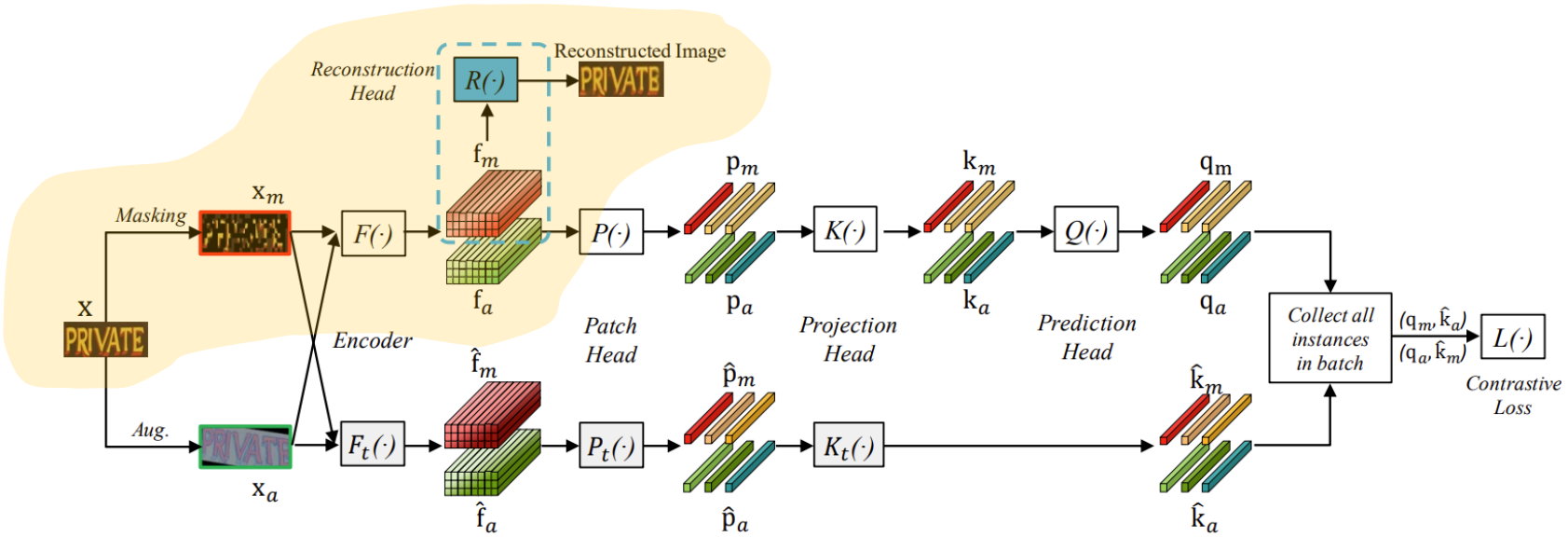
방법론

- DiG Detail

❖ Step3. Generative Loss 산출

- 이미지에서 Masking된 부분을 복원하는 방식으로 학습
- L2 Loss 활용

$$L_m = \frac{1}{N}\sum_{i \in N}(x_i - y_i)^2$$



방법론

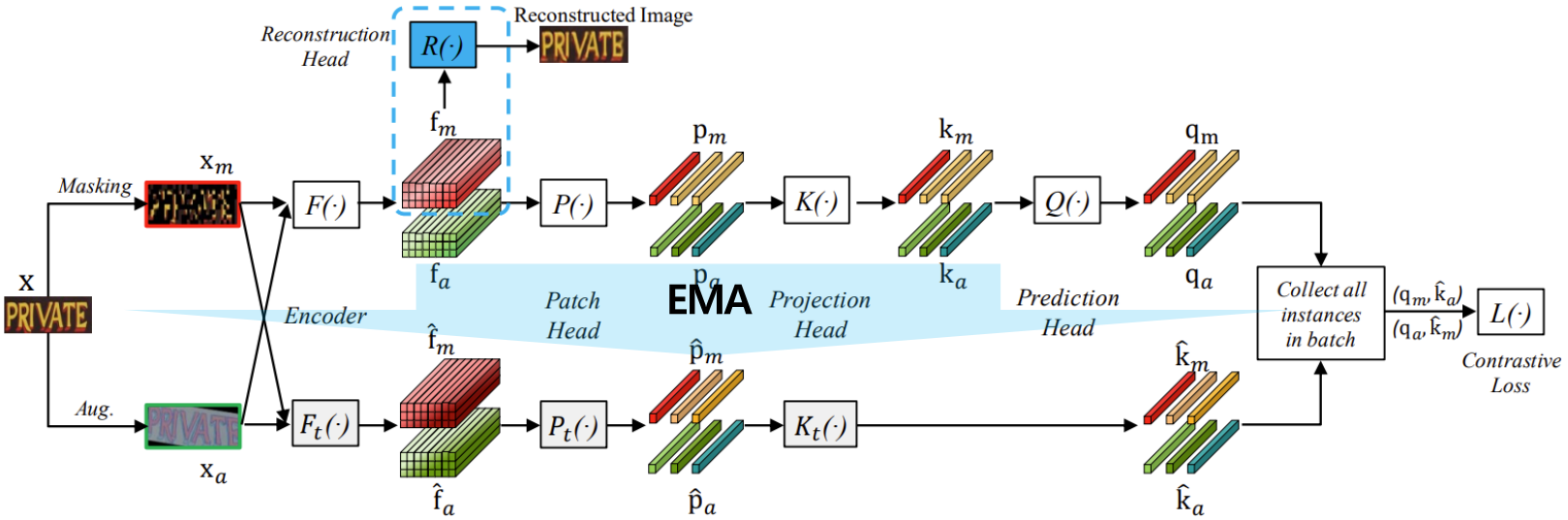
$$L_m = \frac{1}{N}\sum_{i \in N}(x_i - y_i)^2$$

$$L_c = -\log\frac{exp(q_m \cdot \hat{k}_a/\tau)}{exp(q_m \cdot \hat{k}_a/\tau) + \sum_{\hat{k}_a^-} exp(q_m \cdot \hat{k}_a^-/\tau)} - \log\frac{exp(q_a \cdot \hat{k}_m/\tau)}{exp(q_a \cdot \hat{k}_m/\tau) + \sum_{\hat{k}_m^-} exp(q_a \cdot \hat{k}_m^-/\tau)},$$

- DiG Detail

❖ Step4 & 5 Loss 가중합 및 Network Update

- Contrastive Loss와 Generative Loss를 가중합하여 최종 Loss 산출 $L = L_c + \alpha \times L_m$
- Online Network는 Backpropagation으로, Momentum Network는 EMA로 Update (BYOL style)



실험결과

- Setting

❖ 실험환경

- 사전학습에는 합성 Labeled 데이터(17M)와 실제 Unlabeled 데이터(15.77M)를 함께 활용
- DiG-ViT-Tiny / DiG-ViT-Small / DiG-ViT-Base 에 대해서 실험 수행
  - ✓ 각각은 Embedding 크기에 차이가 있음 (192, 384, 512)

실험결과

- Result

❖ Scene Text Recognition Benchmark 데이터셋 실험 결과

- 기존 방법론보다 큰 폭으로 성능 개선
- Discriminative방식과 Generative 방식 모두 효과가 있음을 입증
- 유사한 파라미터 개수의 모델과 비교 시 큰 성능 개선
  - ✓ 물론 Unlabeled 데이터를 함께 활용하니 당연한 결과 일수도„
  - ✓ ABINet같은 경우는 Language Model을 함께 활용함에도 이김

Method	Decoder	Regular			Irregular						Occluded		Handwritten	
		IIIT	SVT	IC13	IC15	SVTP	CUTE	COCO	CTW	TT	HOST	WOST	IAM	CVL
SeqCLR [1]	CTC	80.9	-	86.3	-	-	-	-	-	-	-	-	76.7	76.9
PerSec-ViT + UTI-100M [34]	CTC	85.4	86.1	92.8	70.3	73.9	69.2	-	-	-	-	-	79.9	80.5
DiG-ViT-Tiny	CTC	93.3	89.7	92.5	79.1	78.8	83.0	58.7	69.7	72.1	32.3	53.3	79.5	82.7
DiG-ViT-Small	CTC	95.5	91.8	95	84.1	83.9	86.5	64.3	76.0	76.87	48.6	67.7	82.7	86.4
DiG-ViT-Base	CTC	**95.9**	**92.6**	**95.3**	**84.2**	**85.0**	**89.2**	**66.0**	**77.3**	**78.7**	**58.0**	**73.1**	**83.2**	**87.4**
SeqCLR [1]	Attention	82.9	-	87.9	-	-	-	-	-	-	-	-	79.9	77.8
PerSec-ViT + UTI-100M [34]	Attention	88.1	86.8	94.2	73.6	77.7	72.7	-	-	-	-	-	83.7	82.9
DiG-ViT-Tiny	Attention	95.1	92.4	95.8	83.2	85.4	84.7	63.8	72.3	75.9	47.7	65.1	83.8	86.6
DiG-ViT-Small	Attention	96.4	**94.6**	**96.6**	86.0	**89.3**	88.9	68.2	76.7	80.0	65.0	77.1	84.9	89.0
DiG-ViT-Base	Attention	**96.8**	94.1	**96.6**	**86.5**	87.9	**92.4**	**68.7**	**77.7**	**81.3**	**70.1**	**80.2**	**85.6**	**90.2**
DiG-ViT-Tiny	Transformer	95.8	92.9	96.4	84.8	87.4	86.1	66.8	75.3	78.1	60.9	73.0	85.2	88.9
DiG-ViT-Small	Transformer	**96.7**	93.4	**97.1**	**87.1**	90.1	88.5	68.8	78.8	81.1	72.1	81.1	85.7	90.5
DiG-ViT-Base	Transformer	**96.7**	**94.6**	96.9	**87.1**	**91.0**	**91.3**	**69.8**	**79.3**	**81.9**	**74.9**	**82.3**	**87.0**	**91.3**

실험결과

- Result

❖ Scene Text Recognition Benchmark 데이터셋 실험 결과

- 기존 방법론보다 큰 폭으로 성능 개선
- Discriminative방식과 Generative 방식 모두 효과가 있음을 입증
- 유사한 파라미터 개수의 모델과 비교 시 큰 성능 개선
  - ✓ 물론 Unlabeled 데이터를 함께 활용하니 당연한 결과 일수도„
  - ✓ ABINet같은 경우는 Language Model을 함께 활용함에도 이김

Encoder Freeze (0)

Method	Regular			Irregular						Occluded		Avg.
	IIIT	SVT	IC13	IC15	SVTP	CUTE	COCO	CTW	TT	HOST	WOST	
Gen-ViT-Small	86.6	82.1	88.7	72.9	74.4	72.2	48.5	64.1	63.3	33.8	56.5	59.3
Dis-ViT-Small	92.6	90.4	93.4	81.2	81.7	84.0	60.0	72.8	73.1	33.3	56.1	67.0
DiG-ViT-Small	**94.2**	**93.0**	**95.3**	**84.3**	**86.1**	**87.5**	**63.4**	**77.9**	**75.8**	**41.7**	**64.0**	**71.1**

Encoder Freeze (X)

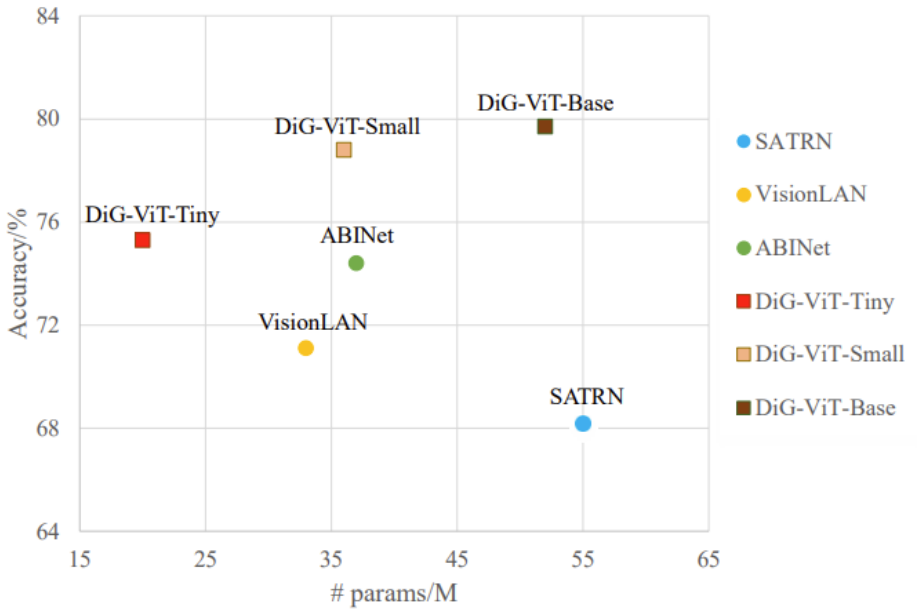
Label Fraction	Method	Regular			Irregular						Occluded		Avg.
		IIIT	SVT	IC13	IC15	SVTP	CUTE	COCO	CTW	TT	HOST	WOST	
1% (27.8K)	Scratch-ViT-Small	12.6	3.9	10.3	7.56	3.41	6.9	2.2	4.6	4.5	5.4	6.0	5.2
	Gen-ViT-Small	87.2	84.9	89.5	76.0	75.5	72.6	52.0	63.7	64.7	30.1	**54.2**	60.6
	Dis-ViT-Small	87.5	85.9	88.9	75.9	73.3	72.9	52.8	63.9	65.2	30.3	49.5	60.6
	DiG-ViT-Small	**88.4**	**86.2**	**89.9**	**79.0**	**76.6**	**77.8**	**54.8**	**67.9**	**67.2**	**33.2**	53.3	**62.9**
10% (278K)	Scratch-ViT-Small	78.4	73.6	81.8	66.8	64.8	56.6	43.2	48.9	54.4	30.7	48.4	52.3
	Gen-ViT-Small	95.0	92.3	95.1	83.7	84.7	90.6	65.1	79.3	80.6	37.8	63.9	71.9
	Dis-ViT-Small	94.6	92.3	94.6	84.5	86.2	89.9	65.7	78.2	79.8	39.0	61.3	71.9
	DiG-ViT-Small	**95.3**	**94.4**	**95.9**	**85.3**	**87.9**	**91.7**	**67.1**	**80.5**	**81.1**	**42.1**	**64.0**	**73.5**
100% (2.78M)	Scratch-ViT-Small	95.0	92.9	94.9	85.2	86.7	88.9	66.1	78.8	81.0	44.8	67.9	73.4
	Gen-ViT-Small	97.2	**97.1**	**97.6**	88.5	91.5	95.5	74.6	86.0	**89.2**	54.4	74.3	80.2
	Dis-ViT-Small	97.1	95.7	97.4	88.1	**92.1**	94.8	74.3	85.2	88.7	55.5	72.9	79.9
	DiG-ViT-Small	**97.7**	96.1	97.3	**88.6**	91.6	**96.2**	**75.0**	**86.3**	88.9	**56.0**	**75.7**	**80.7**

실험결과

- Result

❖ Scene Text Recognition Benchmark 데이터셋 실험 결과

- 기존 방법론보다 큰 폭으로 성능 개선
- Discriminative방식과 Generative 방식 모두 효과가 있음을 입증
- 유사한 파라미터 개수의 모델과 비교 시 큰 성능 개선
  - ✓ 물론 Unlabeled 데이터를 함께 활용하니 당연한 결과 일수도„
  - ✓ ABINet같은 경우는 Language Model을 함께 활용함에도 이김



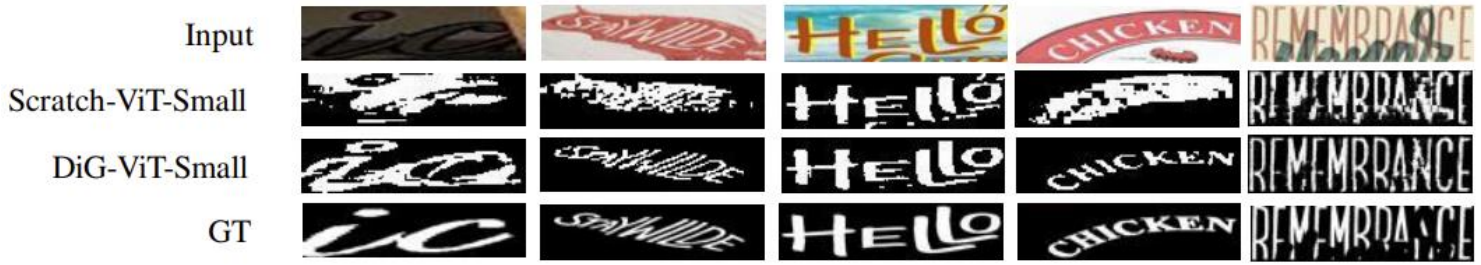
실험결과

- Result

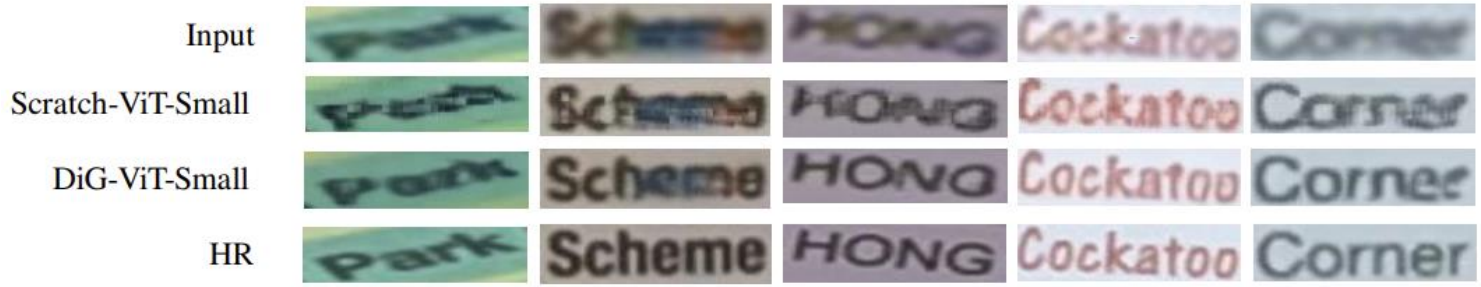
Method	SSIM			PSNR		
	Easy	Medium	Hard	Easy	Medium	Hard
SRCNN [13]	0.8152	0.6425	0.6833	23.13	19.57	19.56
SRResNet [27]	0.8176	0.6324	0.7060	20.65	18.90	19.50
TSRN [56]	0.8562	0.6596	0.7285	22.95	19.26	19.76
TBSRN [3]	0.8729	0.6455	0.7452	24.13	19.08	20.09
Scratch-ViT-Small	0.8143	0.6288	0.6845	22.90	19.65	20.45
DiG-ViT-Small	0.8613	0.6561	0.7215	23.98	19.85	20.57

❖ 다양한 Task에서 실험 결과

- 학습된 Encoder를 Text Segmentation 및 Text Super-Resolution에 적용 시 성능 향상 확인



(a)



(b)