Anomaly detection

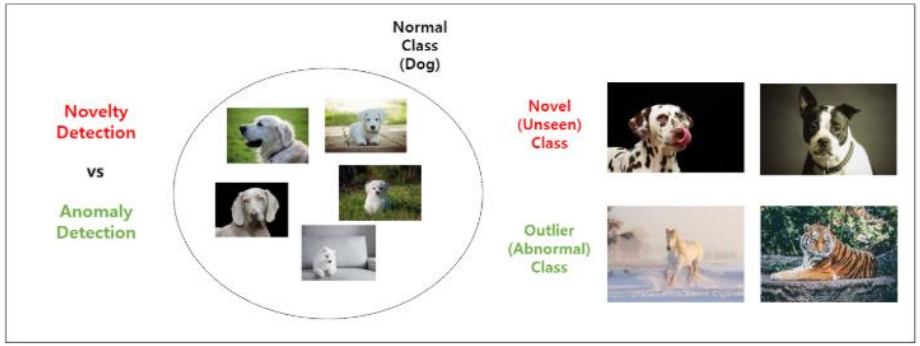
Vision and Display System Lab.
Sogang University

Outline

- Anomaly detection
  - Background
- Knowledge distillation
  - Meta Pseudo Labels
  - Uninformed Students: Student–Teacher Anomaly Detection with Discriminative Latent Embeddings
- Vision Transformer
  - An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
  - Transformer-Based Anomaly Segmentation
- Contrastive learning
  - A Simple Framework for Contrastive Learning of Visual Representations
  - ~~CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances~~
- Conclusion

Anomaly detection

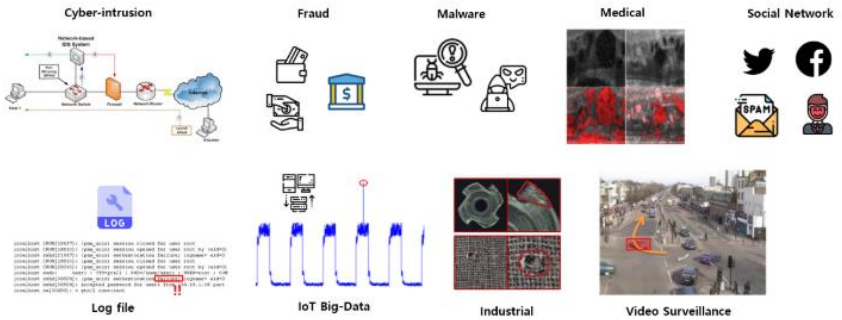
- Anomaly detection
  - Normal sample과 abnormal sample을 구별해내는 문제
  - 구분하고자 하는 대상에 따라 여러 분야로 나뉨
    - 비정상 sample 정의에 따른 분류
      - Novelty Detection(Unseen)
        - ✓지금까지 등장하지 않았지만 충분히 등장할 수 있는 sample을 찾아내는 분야
      - Outlier Detection(abnormal)
        - ✓등장할 가능성이 거의 없는, 데이터에 오염이 발생했을 가능성이 있는 sample을 찾아 내는 분야



< Sample 예시 >

Anomaly detection

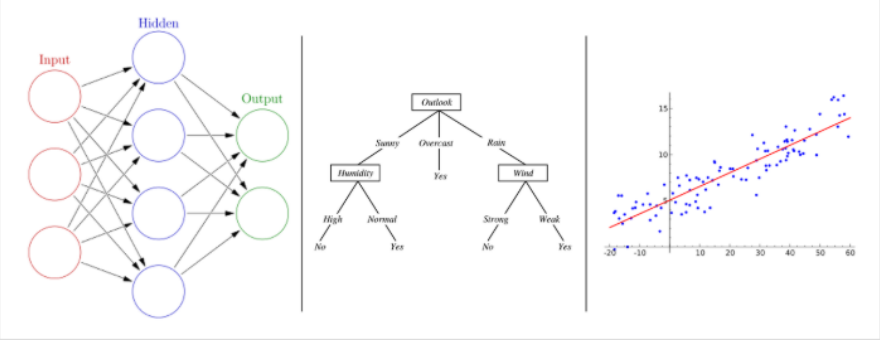
- Anomaly detection
  - Normal sample과 abnormal sample을 구별해내는 문제
  - 데이터 셋의 구성에 따라 anomaly detection의 분야가 달라짐
    - Supervised Anomaly Detection
      - 이상 데이터와 정상 데이터의 label이 존재
    - Semi-supervised (One-Class) Anomaly Detection
      - 정상 데이터에만 label이 존재
    - Unsupervised Anomaly Detection
      - 모든 데이터가 label이 존재하지 않음



< Anomaly detection 적용 분야 >

Anomaly detection

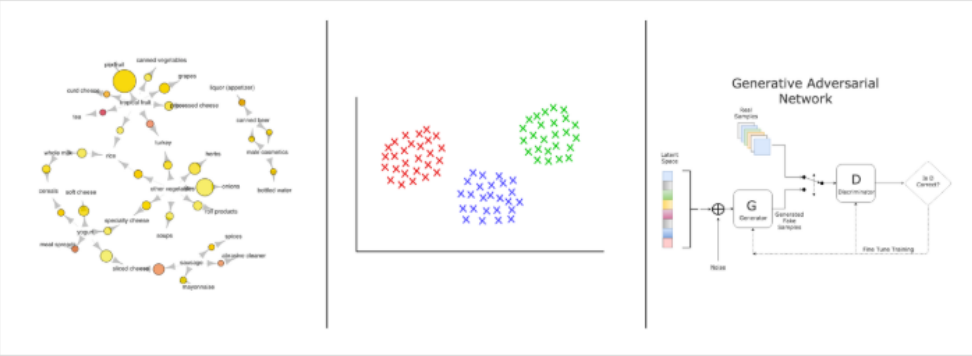
- Supervised Anomaly Detection
  - 정상 sample과 비정상 sample의 Data와 Label이 모두 존재하는 경우
  - 다른 방법 대비 정확도가 높음
    - 그러나 현실적인 상황에서는 정상 sample보다 비정상 sample의 발생 빈도가 적어 class imbalance문제를 발생
    - 또한 이상치의 기준이 명확하지 않아 이상치 label을 생성하기가 어렵고 시간과 비용이 많이 소모됨
    - 훈련된 class나 데이터 유형이 아닌 이상치가 들어올 경우 모델 전체를 다시 훈련시켜야 함
      - 다중 회귀 분석, SVM, 의사결정나무, 인공신경망 등의 기법이 존재



< Supervised 기법 예시 >

Anomaly detection

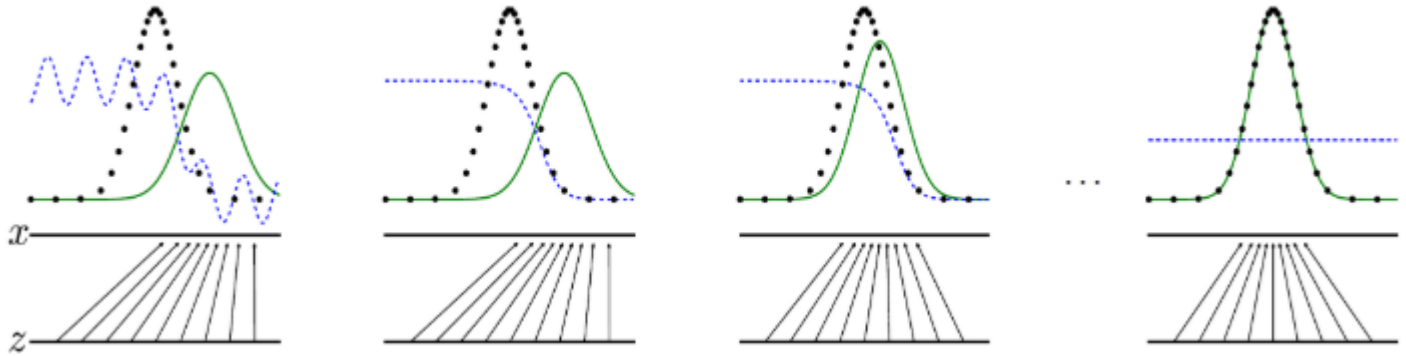
- Semi-supervised (One-Class), Unsupervised Anomaly Detection
  - Data와 Label이 존재하지 않거나 정상 데이터의 Label만 존재하는 경우
  - 정상 데이터를 이용하여 데이터가 본질적으로 가지고 있는 특징을 추출
    - 정상 sample들을 둘러싸는 discriminative boundary를 설정하고, 이 boundary를 최대한 좁혀 boundary 밖에 있는 sample들을 모두 비정상으로 간주
    - 정상 데이터의 특징을 학습하므로 학습되지 않은 모든 이상치 데이터들을 구분할 수 있음
      - ⁂ 별도의 labeling작업이 필요하지 않음
    - 모델 파라메터나 정상데이터의 구성에 따라 모델의 성능이 불안정함



< Unsupervised 기법 예시 >

Anomaly detection

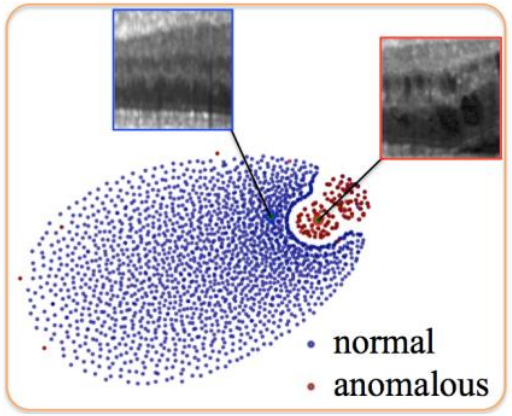
- Background
  - GAN based anomaly detection
    - 기존 딥러닝 모델은 입력에 대해 가장 높은 확률을 지닌 출력을 내도록 학습
      - 학습되지 않은 유사 데이터나 전체 데이터 셋의 분포를 고려하기 어려움
    - GAN model은 학습 데이터셋의 distribution을 학습
      - Data distribution을 학습하기때문에 배우지 않은 데이터에 대해서도 In of distribution이라 판단되면 생성할 수 있음



< GAN이 분포를 학습하는 과정 >

Anomaly detection

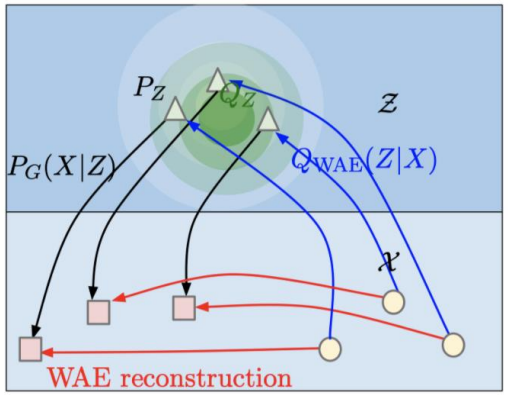
- Background
  - GAN based anomaly detection
    - 데이터의 분포를 학습하는 GAN의 성질을 이용하여 Unsupervised anomaly detection을 수행할 수 있음
      - 이미지는 WxHxC차원의 하나의 포인트라고 가정
      - 분포안에 data라고 판단하는 포인트에 대해서는 재구성이 가능
      - 분포 밖의 data에 대해서는 모델이 재구성하기 힘들기 때문에 그림이 뭉개지는 현상이 발생
      - 이때 입력 이미지와 생성된 이미지의 l1loss를 통해 anomaly detection을 수행할 수 있음



< Data의 분포 >

Anomaly detection

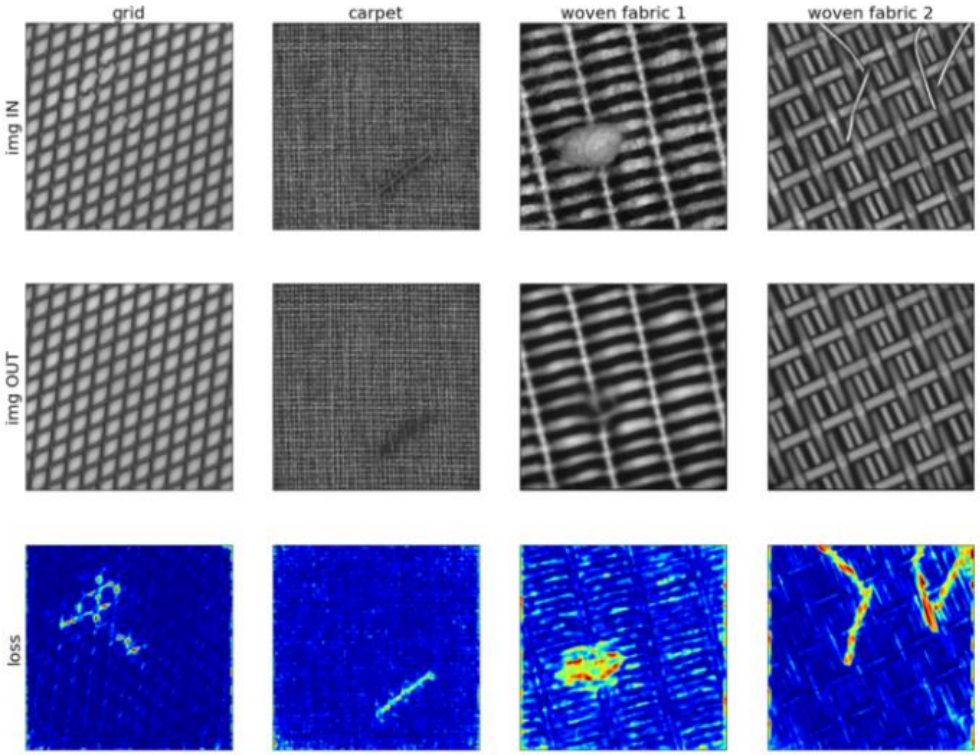
- Background
  - VAE based anomaly detection
    - Variational auto encoder (VAE)를 사용하여 데이터의 분포를 학습
      - 데이터가 가지고 있는 분포를 특정 분포로 가정
        - ✓Ex) Gaussian, Bernoulli 등
      - 가정한 분포에 데이터를 맵핑 되게 학습
        - ✓GAN과 비교해서 학습이 안정



< VAE의 분포 >

Anomaly detection

- Background
  - Anomaly detection 예시

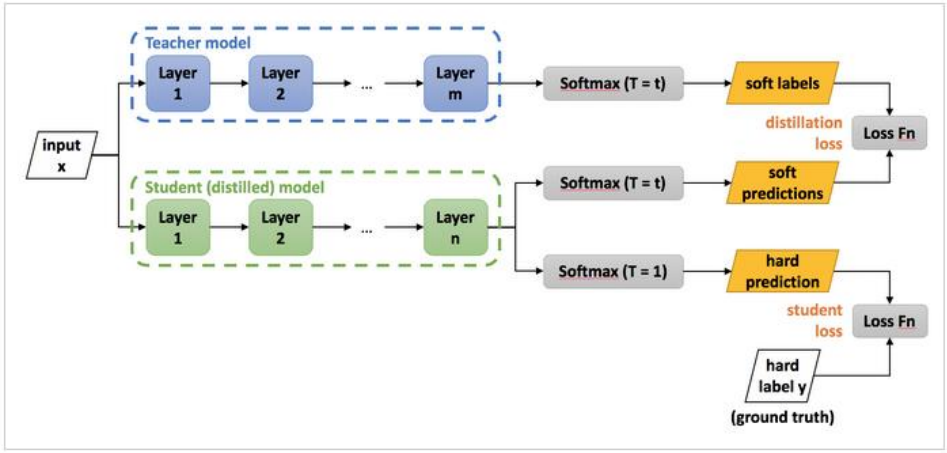


Anomaly detection

- Background
  - Anomaly detection 주요 문제점
    - Representation
      - Normal data의 분포를 얼마나 잘 표현 할 수 있는가
        - ✓Normal data가 복잡하여 분포를 예측하기 어려움
        - ✓다양한 data를 학습하기 힘들기 때문에 normal data의 분포를 표현하기 어려움
    - Localization
      - Anomaly 영역을 특정하기 쉽지 않음
      - CNN의 경우 처리하는 크기가 일정하기 때문에 다양한 크기에 대해 이상치 검출을 하기 어려움

Knowledge distillation

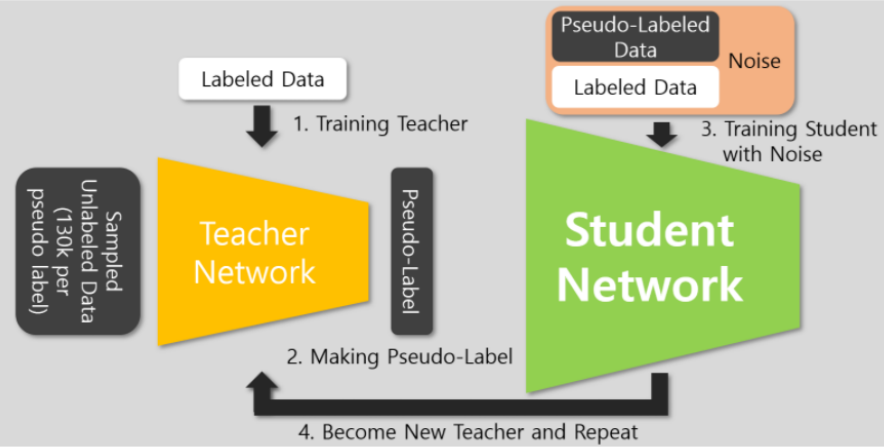
- Background
  - Teacher network의 지식을 실제로 사용하고자 하는 Student network에게 전달
  - 크게 2가지 목적을 위해 사용
    - Inference time, model parameters
      - Student network의 크기를 teacher network의 크기보다 작게 만들어 더 작은 모델에서 동일 성능을 내도록 학습시킴



< Knowledge distillation 구조도 >

Knowledge distillation

- Background
  - Teacher network의 지식을 실제로 사용하고자 하는 Student network에게 전달
  - 크게 2가지 목적을 위해 사용
    - Model performance
      - Student network의 크기를 teacher network의 크기보다 크게 구성
        - ✓Teacher network로 unlabeled data를 labeling하여 기존 데이터에 포함하여 학습



< Knowledge distillation 구조도 >

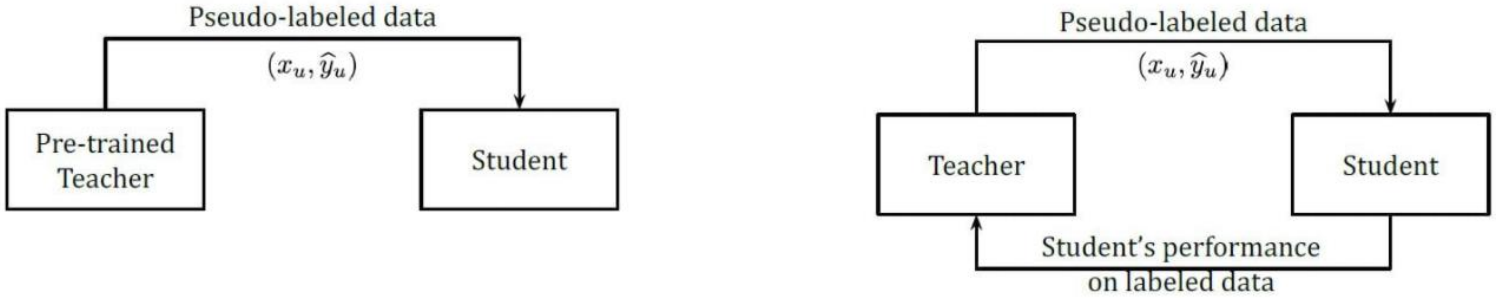
Knowledge distillation

- Meta Pseudo Labels
  - Imagenet Top 1 accuracy 90.2%

RANK	MODEL	TOP 1 ACCURACY	TOP 5 ACCURACY	NUMBER OF PARAMS	EXTRA TRAINING DATA	PAPER	CODE	RESULT
1	Meta Pseudo Labels (EfficientNet-L2)	90.2%	98.8%	480M	✓	Meta Pseudo Labels		
2	Meta Pseudo Labels (EfficientNet-B6-Wide)	90%	98.7%	390M	✓	Meta Pseudo Labels		
3	EfficientNet-L2-475 (SAM)	88.61%		480M	✓	Sharpness-Aware Minimization for Efficiently Improving Generalization		
4	ViT-H/14	88.55%		632M	✓	An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale		

Knowledge distillation

- Meta Pseudo Labels
  - 기존 Pseudo Labeling 과정은 teacher network를 고정시키고 수행됨
    - 따라서 Pseudo Labels을 사용한 student network의 학습 결과는 teacher network에 dependency함
  - 제안된 방법은 student network의 학습을 진행하면서 그 결과로 teacher network의 영향을 끼치도록 설계
    - labeled data에 대한 student의 model의 결과를 이용하여 다시 teacher model을 학습
      - student network의 loss를 사용하여 teacher network를 추가로 학습



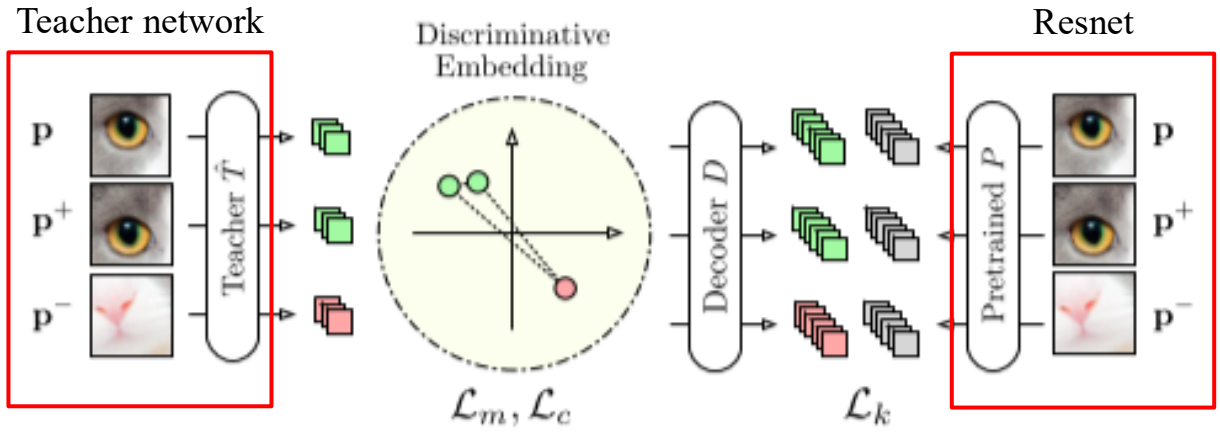
< Pseudo Labeling 방법 (기존 / proposed) >

Knowledge distillation

- Student-teacher-anomaly-detection
  - 데이터의 특징을 추출하여 다른 vector로 잘 표현하기 위해서는 많은 이미지의 특성을 배우는 것이 중요
    - 따라서 pretrain model을 사용하여 다른 task를 수행하는 경우가 많음
  - 그러나 anomaly-detection task에서는 normal data의 분포를 학습해야 함
    - Pretrain model을 사용하면 anomaly한 data의 특징도 잘 추출하게 됨
    - 그러나 pretrain model을 사용하지 않은 경우보다 normal data를 잘 표현하기 어려움
      - ∴ Pretrain model이 이미지를 더 정확히 분석 할 수 있음
  - 따라서 imagenet으로 pretrain된 resnet의 represent vector를 배우는 teacher network구성
    - Teacher network는 normal data로 학습되기 때문에 anomaly data에 대해서는 특징을 추출하기 어려움
    - 또한 pretrain된 resnet의 represent vector를 학습하기 때문에 normal data의 분포를 잘 나타낼 수 있음

Knowledge distillation

- Student-teacher-anomaly-detection
  - Proposed method
    - Teacher network를 이용하여 각 픽셀을 대표하는 latent vector를 추출
      - Teacher network는 5개의 Convolution layer와 FC layer로 구성됨
      - Teacher network의 출력은 pretrain된 resnet의 fc layer전의 512차원의 vector를 target으로 학습
      - L_k: Knowledge Distillation loss (L2-norm loss), L_c: Compactness loss
      - 256x256의 이미지를 p크기의 patch로 잘라 학습



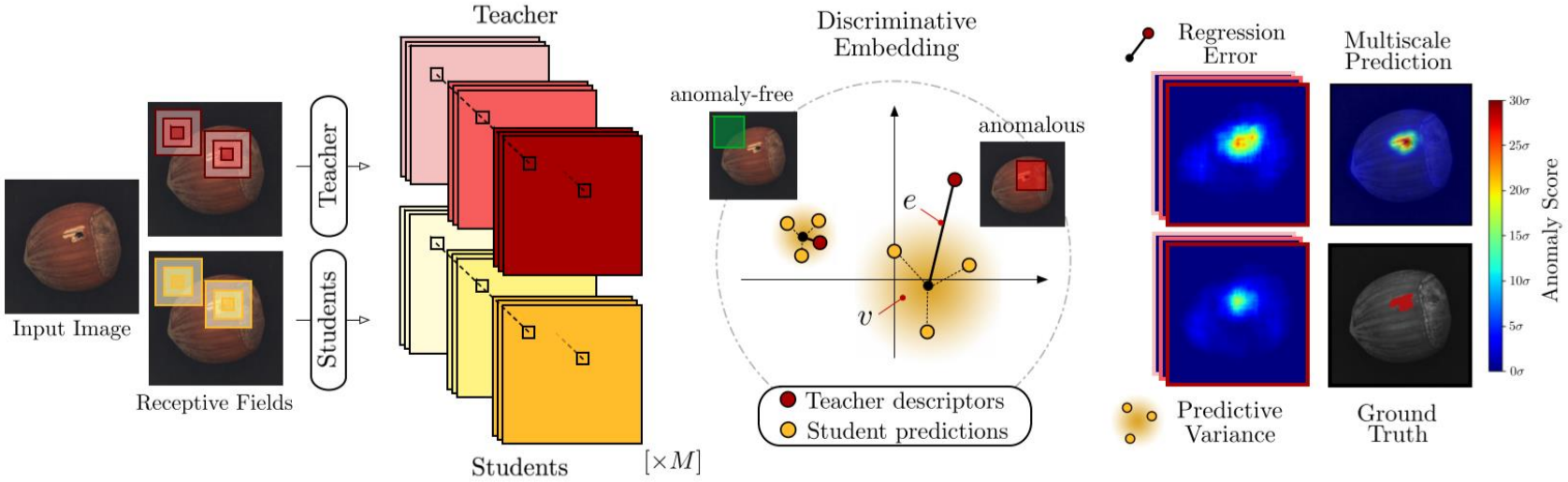
< Teacher network training >

Knowledge distillation

- Student-teacher-anomaly-detection
  - Proposed method
    - Fast_Dense_Feature_Extraction (FDFE)
      - P크기의 patch로 이미지를 잘라 사용하면 PxP 크기당 하나의 latent vector밖에 추출할 수 없음
        - ✓pixel 당 하나의 latent vector를 구해야 함
        - ✓이를 위해선 대표 pixel을 중앙으로 잡고 patch를 구성한 후 모든 pixel에 대해 연산을 하는 방법이 있음
          - 그러나 이 방법은 256x256 개의 patch에 대한 연상을 수행 해야함
      - 이 과정을 빠르게 수행하기 위해 FDFE 알고리즘이 사용됨
      - 이를 통해 patch (PxP)로 학습된 네트워크에서 전체 이미지 (256x256)의 latent vector를 추출할 수 있게 됨

Knowledge distillation

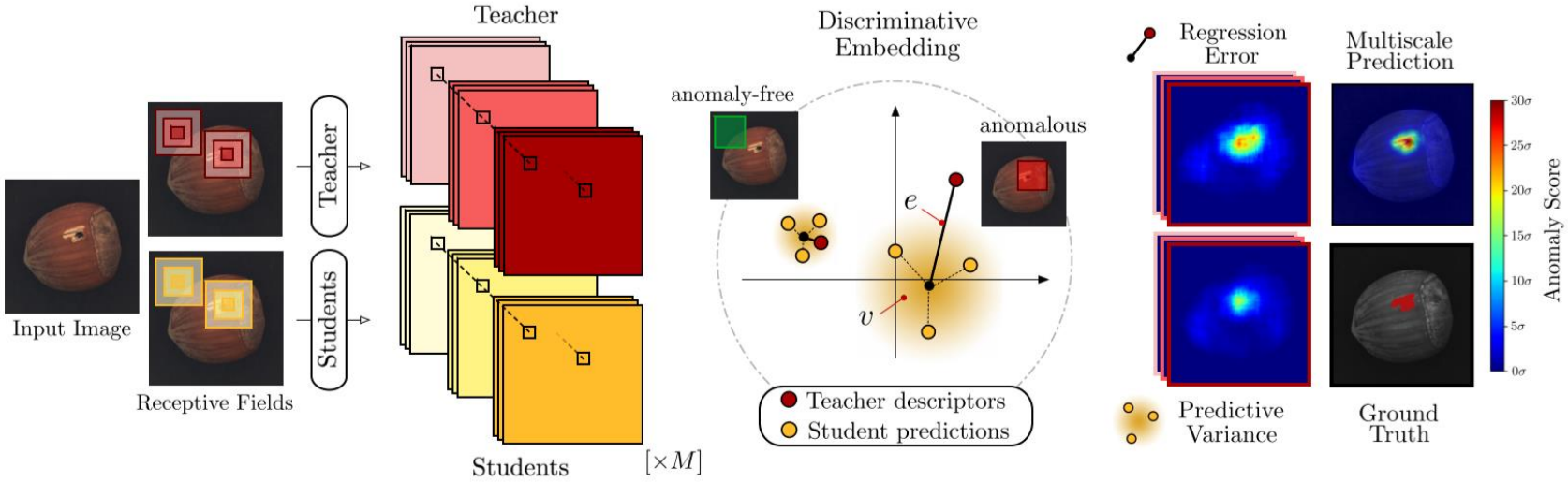
- Student-teacher-anomaly-detection
  - Proposed method
    - Student network는 앙상블 효과를 사용하기 위해 M개의 network로 구성
    - Student network들은 teacher network의 출력을 학습
      - Normal한 이미지에 대해서는 학습된 latent vector를 출력하기 때문에 같은 vector로 수렴



Knowledge distillation

- Student-teacher-anomaly-detection
  - Proposed method
    - Anomaly한 이미지에 대해서는 학습된 vector가 존재하지 않기 때문에 teacher와 큰 차이(e) 를 보이는 vector를 생성
    - 또한 학습된 student의 출력의 값에도 차이가 생기기 때문에 이 값들의 분산(v)을 anomaly score로 이용

      ⁘ Anomaly score = e + v



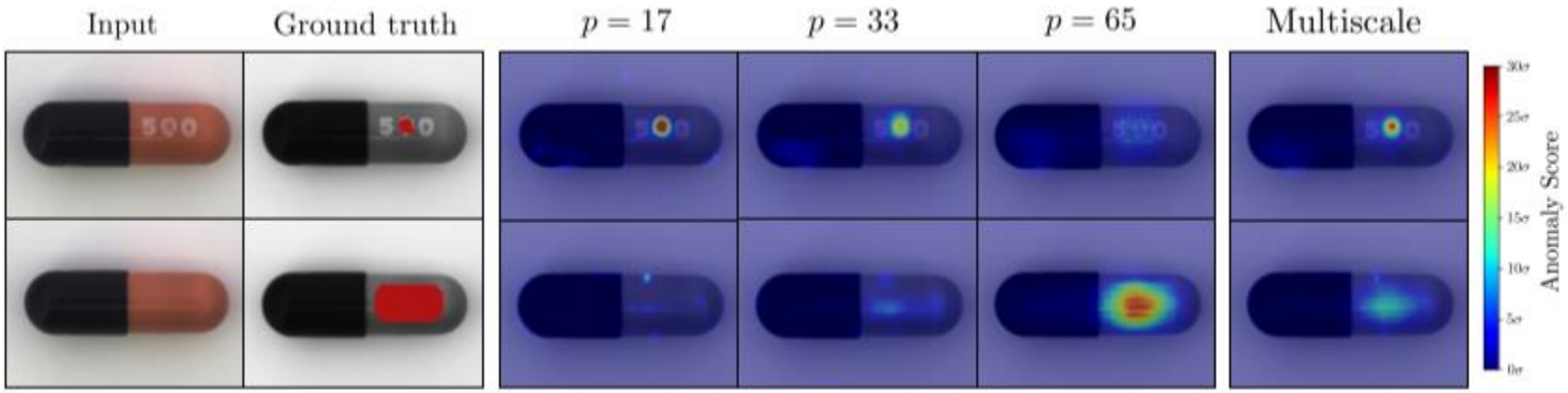
Knowledge distillation

- Student-teacher-anomaly-detection
  - Experiment
    - Metric: PRO-curve

	Category	Ours $p = 65$	1-NN	OC-SVM	K-Means	ℓ_2-AE	VAE	SSIM-AE	AnoGAN	CNN-Feature Dictionary
Textures	Carpet	**0.695**	0.512	0.355	0.253	0.456	0.501	0.647	0.204	0.469
	Grid	0.819	0.228	0.125	0.107	0.582	0.224	**0.849**	0.226	0.183
	Leather	**0.819**	0.446	0.306	0.308	**0.819**	0.635	0.561	0.378	0.641
	Tile	**0.912**	0.822	0.722	0.779	0.897	0.870	0.175	0.177	0.797
	Wood	0.725	0.502	0.336	0.411	**0.727**	0.628	0.605	0.386	0.621
Objects	Bottle	**0.918**	0.898	0.850	0.495	0.910	0.897	0.834	0.620	0.742
	Cable	**0.865**	0.806	0.431	0.513	0.825	0.654	0.478	0.383	0.558
	Capsule	**0.916**	0.631	0.554	0.387	0.862	0.526	0.860	0.306	0.306
	Hazelnut	**0.937**	0.861	0.616	0.698	0.917	0.878	0.916	0.698	0.844
	Metal nut	**0.895**	0.705	0.319	0.351	0.830	0.576	0.603	0.320	0.358
	Pill	**0.935**	0.725	0.544	0.514	0.893	0.769	0.830	0.776	0.460
	Screw	**0.928**	0.604	0.644	0.550	0.754	0.559	0.887	0.466	0.277
	Toothbrush	**0.863**	0.675	0.538	0.337	0.822	0.693	0.784	0.749	0.151
	Transistor	0.701	0.680	0.496	0.399	**0.728**	0.626	0.725	0.549	0.628
	Zipper	**0.933**	0.512	0.355	0.253	0.839	0.549	0.665	0.467	0.703
	Mean	**0.857**	0.640	0.479	0.423	0.790	0.639	0.694	0.443	0.515

Knowledge distillation

- Student-teacher-anomaly-detection
  - Experiment
    - Patch 별 성능 평가
    - Pixel의 latent vector를 patch별로 추출하기 때문에 작은 patch에서는 작은 이상치 영역을 검출하기 용이함
    - 따라서 앙상블 하는 student network 다른 patch로 구성하여 성능 향상



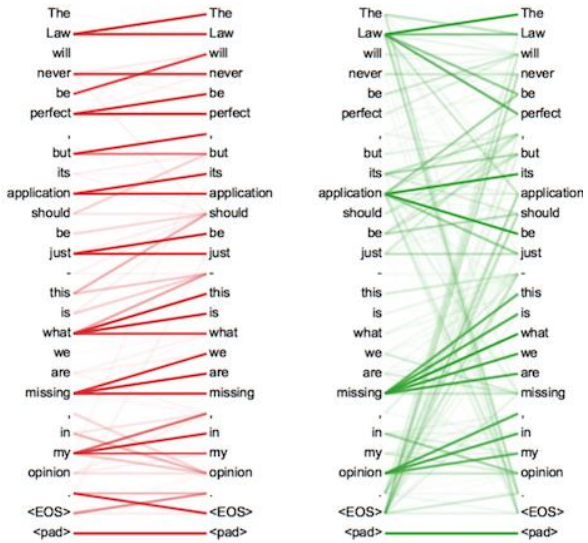
Knowledge distillation

- Student-teacher-anomaly-detection
  - Experiment

	Category	$p = 17$	$p = 33$	$p = 65$	Multiscale
Textures	Carpet	0.795	**0.893**	0.695	0.879
	Grid	0.920	0.949	0.819	**0.952**
	Leather	0.935	**0.956**	0.819	0.945
	Tile	0.936	**0.950**	0.912	0.946
	Wood	**0.943**	0.929	0.725	0.911
Objects	Bottle	0.814	0.890	0.918	**0.931**
	Cable	0.671	0.764	**0.865**	0.818
	Capsule	0.935	0.963	0.916	**0.968**
	Hazelnut	0.971	**0.965**	0.937	**0.965**
	Metal nut	0.891	0.928	0.895	**0.942**
	Pill	0.931	0.959	0.935	**0.961**
	Screw	0.915	0.937	0.928	**0.942**
	Toothbrush	**0.946**	0.944	0.863	0.933
	Transistor	0.540	0.611	**0.701**	0.666
	Zipper	0.848	0.942	0.933	**0.951**
	Mean	0.866	0.900	0.857	**0.914**

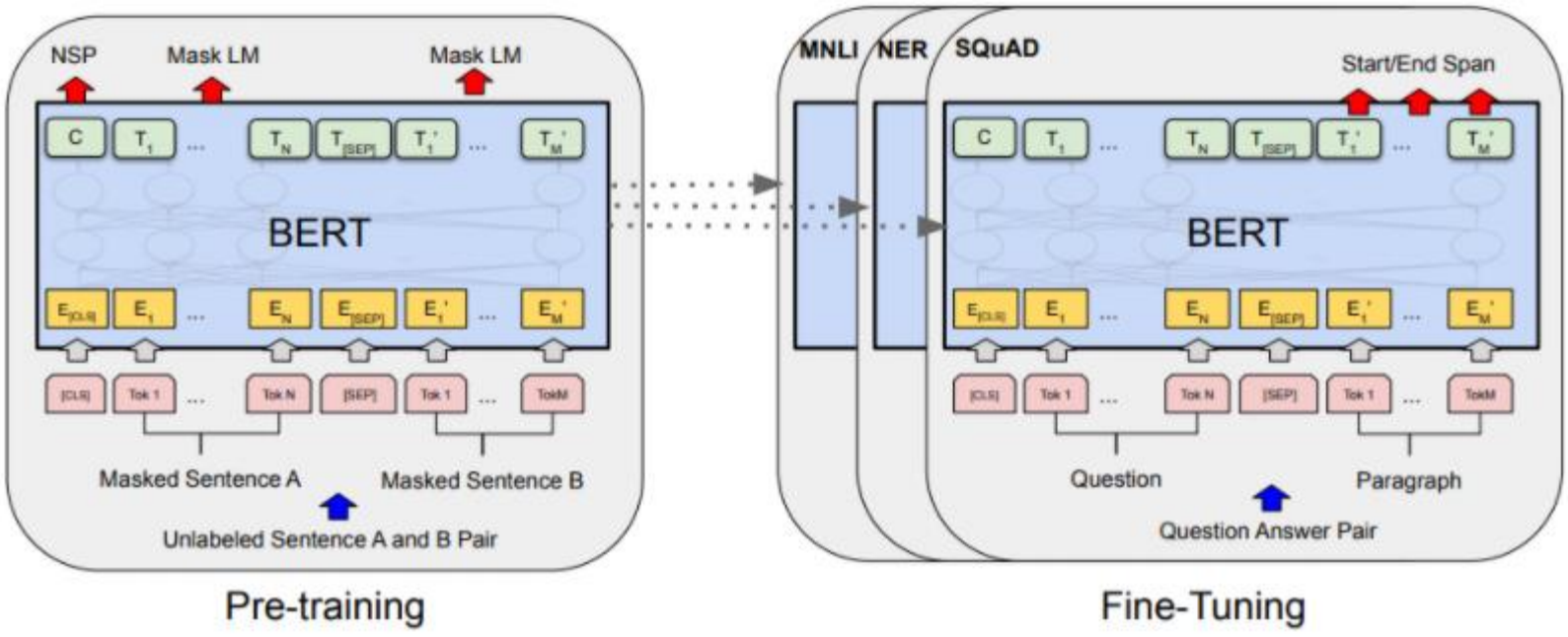
Visual transformer

- Background
  - Transformer
    - Attention is all you need에서 나온 번역모델 구조
    - 문장 같은 시계열 구조에서는 각 단어 간의 vector를 고려할 때 필요한 단어만 확인하는 것이 중요
    - 이를 구현한 알고리즘이 attention mechanism으로 transformer가 가장 많이 사용되는 구조



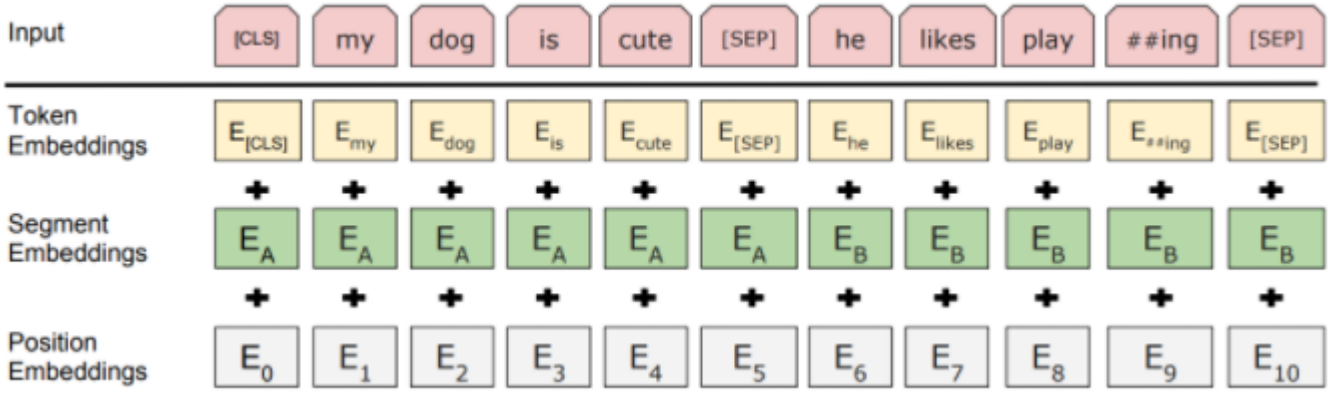
Visual transformer

- Background
  - BERT
    - Transformer encoder구조를 사용한 자연어 처리 모델
    - Pretrain task로 학습 한 후 fine-tuning



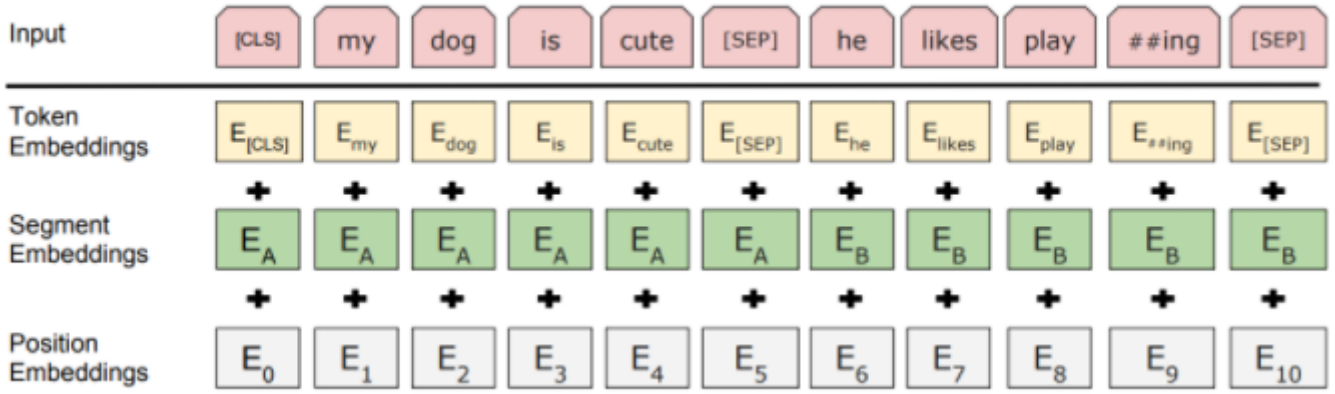
Visual transformer

- Background
  - BERT
    - CLS란 특수 토큰 (vector)를 첫 단어로 입력한 후 문장을 입력
    - CLS의 출력 vector를 이용하여 문장을 분류하는 task를 하게 됨
      - Encoder를 거치면서 문장의 중요한 정보가 결합됨
        - ✓Ex) 문장 감정 분류 등



Visual transformer

- Background
  - BERT
    - Position embedding
      - BERT는 FC layer로만 구성
        - ✓따라서 RNN이나 CNN처럼 순서, 위치 정보를 가지고 있지 않음
        - ✓이런 경우에 단어의 순서가 바뀌어도 동일한 의미를 가지게 됨
          - Ex) You are, Are you
        - ✓따라서 위치정보를 가지는 Position embedding을 더하여 입력을 구성하게 됨



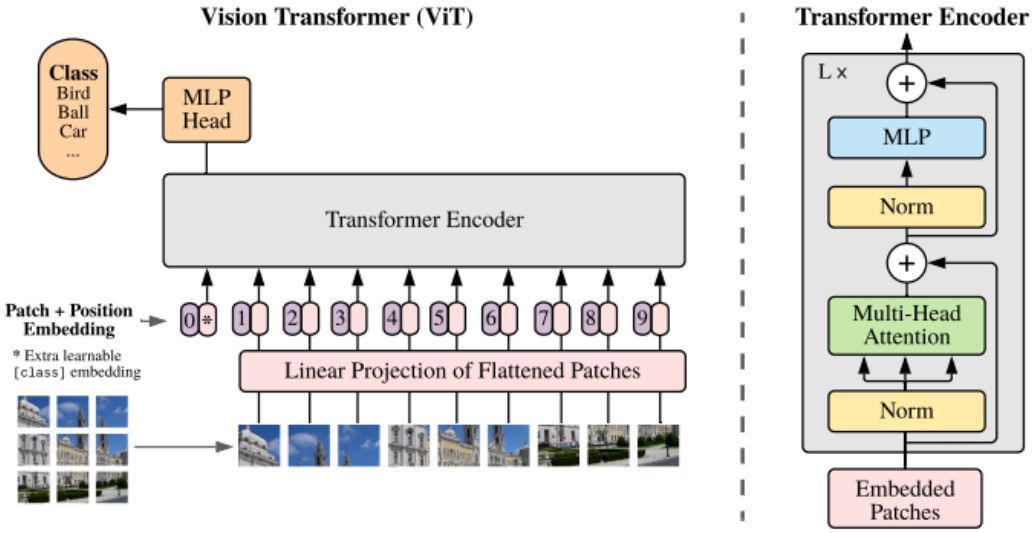
Visual transformer

- An Image is Worth 16X16 Words: Transformers for Image Recognition at Scale
  - Transformer architecture는 아직 computer vison task에서 제한적
    - CNN architecture의 특정 부분을 대체하는 쪽에만 이용
  - 본 논문에서는 이미지를 patch별로 구분하여 transformer에 적용될 때 image classification task에서 잘 수행될 수 있음을 보여줌
  - 많은 양의 데이터에 대해 사전 학습을 수행하고 transfer learning을 수행하면 Vision Transformer는 훨씬 적은 computational resource를 가짐
    - 동시에 SotA CNN과 비교하여 더 우수한 결과를 얻는 것을 보여줌

RANK	MODEL	TOP 1 ACCURACY ↑	TOP 5 ACCURACY	NUMBER OF PARAMS	EXTRA TRAINING DATA	PAPER	CODE	RESULT
1	Meta Pseudo Labels (EfficientNet-L2)	90.2%	98.8%	480M	✓	Meta Pseudo Labels		
2	Meta Pseudo Labels (EfficientNet-B6-Wide)	90%	98.7%	390M	✓	Meta Pseudo Labels		
3	EfficientNet-L2-475 (SAM)	88.61%		480M	✓	Sharpness-Aware Minimization for Efficiently Improving Generalization		
4	ViT-H/14	88.55%		632M	✓	An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale		

Visual transformer

- An Image is Worth 16X16 Words: Transformers for Image Recognition at Scale
  - Proposed method
    - 전체 이미지를 p size의 patch로 분할
    - Patch를 transformer 구조에 입력 가능하도록 Linear projection을 통해 1차원의 vecto로 변형
    - 첫번째 입력으로 bert의 CLS와 유사하게 학습 가능한 embedding을 추가



Visual transformer

- An Image is Worth 16X16 Words: Transformers for Image Recognition at Scale
  - Experiment

	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

Visual transformer

- Transformer-Based Anomaly Segmentation
  - Sub-Image Anomaly Detection with Deep Pyramid Correspondences
    - Pretrain model만을 사용하여 훈련과정을 거치지 않고 inference로만 anomaly detection을 수행
    - Transformer-Based Anomaly Segmentation은 pretrain model을 Vit로 사용
    - Detection 과정은 2단계로 구성
      - Train data를 사용하여 pretrain 된 모델에서 feature를 추출
        - ✓Ex) Average pooling 전 feature
      - Test image의 feature를 구한 train feature와 비교하여 K개의 nearest image feature를 추출 한 후 그 feature와의 평균 거리를 anomaly score로 사용

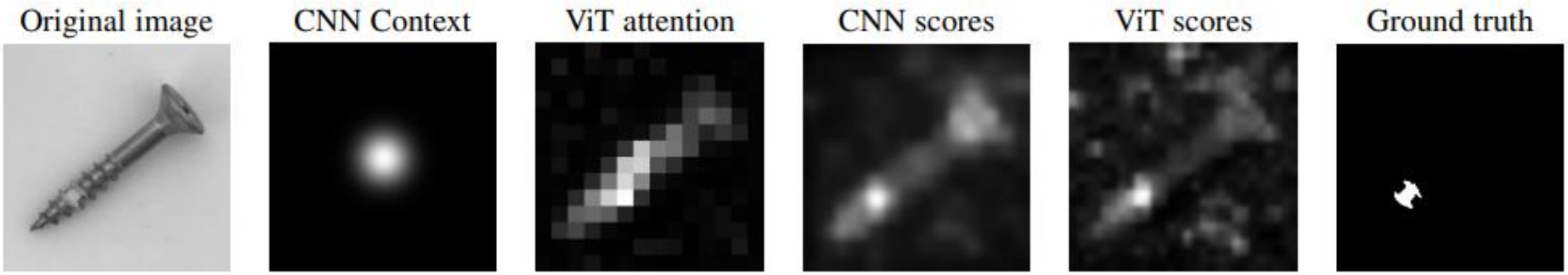
$$d(y) = \frac{1}{K} \sum_{f \in N_K(f_y)} |f - f_y|^2$$

      - 그 후 localization을 위해 K개의 nearest image feature를 사용하여 test feature와 같은 위치에 있는 k개의 pixel feature들 중 가장 가까운 pixel feature를 추출
      - 일정 범위 밖의 pixel을 anomaly로 판정

$$d(y,p) = \frac{1}{\kappa} \sum_{f \in N_\kappa(F(y,p))} |f - F(y,p)|^2$$

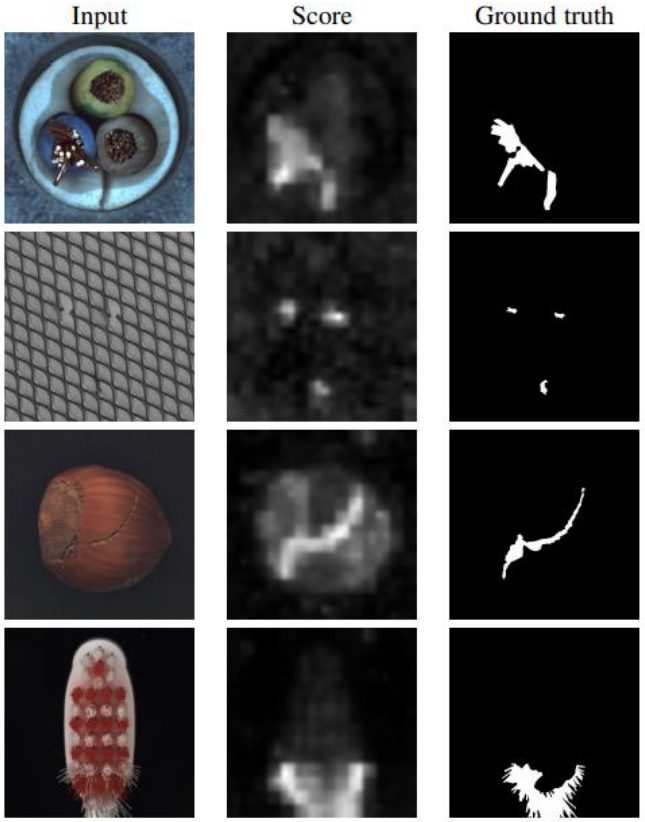
Visual transformer

- Transformer-Based Anomaly Segmentation
  - Experiment



Visual transformer

- Transformer-Based Anomaly Segmentation
  - Experiment



Visual transformer

- Transformer-Based Anomaly Segmentation
  - Experiment

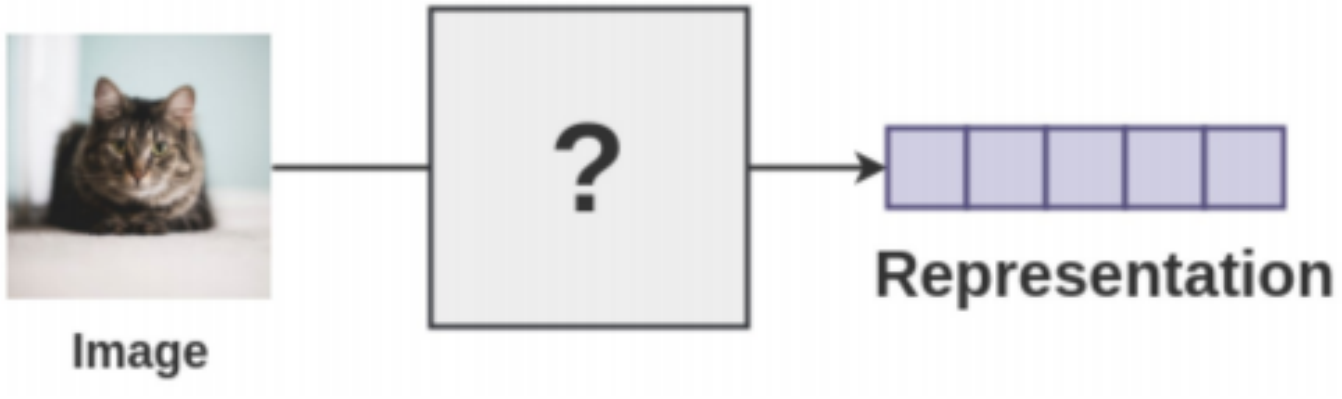
Class	Baselines							Ours	
	OC-SVM	1-NN	VAE	CNN-Dict	AE_{ℓ_2}	AE_{SSIM}	Student	SPADE	MST
Carpet	35.5	51.2	50.1	46.9	45.6	64.7	69.5	95.4	97.8
Grid	12.5	22.8	22.4	18.3	58.2	84.9	81.9	92.9	96.3
Leather	30.6	44.6	63.5	64.1	81.9	56.1	81.9	98.1	99.0
Tile	72.2	82.2	87.0	79.7	89.7	17.5	91.2	85.7	92.5
Wood	33.6	50.2	62.8	62.1	72.7	60.5	72.5	91.1	96.1
Bottle	85.0	89.8	89.7	74.2	91.0	83.4	91.8	94.7	96.4
Cable	43.1	80.6	65.4	55.8	82.5	47.8	86.5	86.1	91.1
Capsule	55.4	63.1	52.6	30.6	86.2	86.0	91.6	94.4	91.3
Hazelnut	61.6	86.1	87.8	84.4	91.7	91.6	93.7	93.1	96.4
Metal nut	31.9	70.5	57.6	35.8	83.0	60.3	89.5	94.1	95.5
Pill	54.4	72.5	76.9	46.0	89.3	83.0	93.5	95.4	95.6
Screw	64.4	60.4	55.9	27.7	75.4	88.7	92.8	96.2	95.3
Toothbrush	53.8	67.5	69.3	15.1	82.2	78.4	86.3	91.5	93.0
Transistor	49.6	68.0	62.6	62.8	72.8	72.5	70.1	85.6	85.4
Zipper	35.5	51.2	54.9	70.3	83.9	66.5	93.3	94.4	94.5
Average	47.9	64	63.9	51.5	79	69.4	85.7	92.6	**94.4**

Contrastive learning

- Background
  - Human supervision 없이 effective visual representations를 학습하는 것은 매우 어려움
  - Effective visual representations 위한 2가지 접근 방법이 많이 사용됨
    - Generative method: pixel level generation은 computationally expensive하고 재구성 하는 내용이 visual representations에 크게 필요하지 않음
    - Discriminative method: object function을 활용하여 representations을 학습
  - 기존 방법과는 다르게 Contrastive learning은 가장 간단한 task인 이미지가 유사한지를 판단하여 (binary) effective visual representations을 하는 기법

Contrastive learning

- Background
  - Contrastive learning의 목적은 학습 과정을 통해 이미지를 가장 잘 표현할 수 있는 vector를 추출하는 것이 목표



Contrastive learning

- Background
  - Contrastive learning의 목적 함수로 Noise Contrastive Estimator Loss를 사용

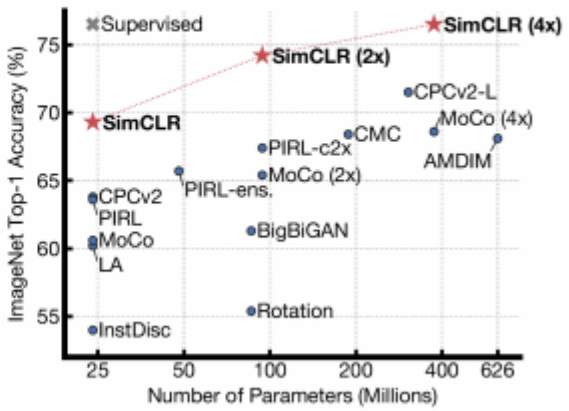
$$NCE_{Loss} = -\log \frac{\exp(\mathrm{sim}(g(\mathrm{x}), g(\mathrm{x}^{+})))}{\exp(\mathrm{sim}(g(\mathrm{x}), g(\mathrm{x}^{+}))) + \sum_{k=1}^{K} \exp(\mathrm{sim}(g(\mathrm{x}), g(\mathrm{x}_{k}^{-})))}$$

  - x+: positive sample, x-: negative sample
    - Positive sample – 입력 이미지 x와 비슷한 이미지
    - Negative sample – 입력 이미지 x와 다른 이미지
  - Sim(.): cosine similarity 함수
  - NCE를 최대화 시키기 위해서는 positive간의 유사도가 높아져야 함
    - 또한 negative간의 유사도는 동시에 멀어져야 함
  - 이를 이용하여 Contrastive learning을 수행

Contrastive learning

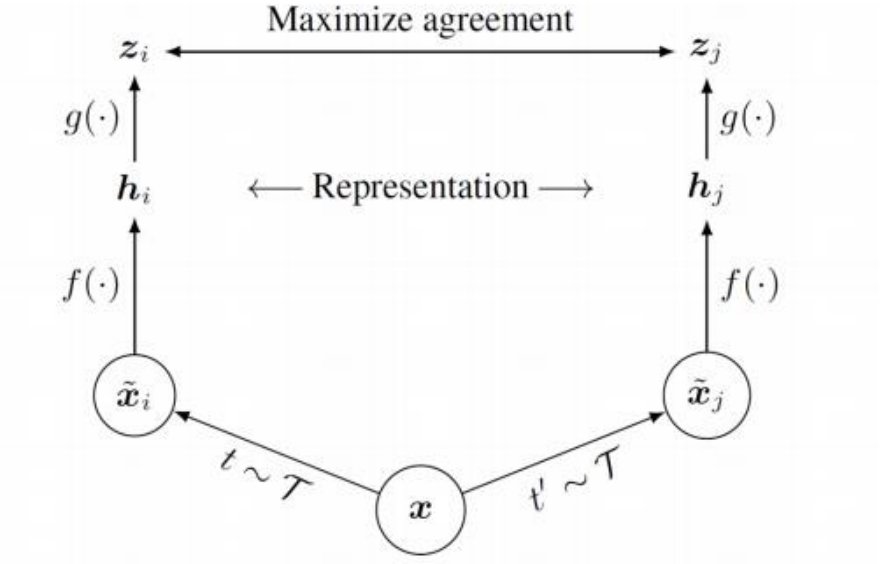
- A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)
  - 기존 NCE loss를 사용하기 위해서는 입력 데이터에 대한 positive/negative sample이 구성되어야 함
  - 따라서 unsupervised learning을 위해서는 positive/negative pair를 만드는 과정 없이 수행되야 함
    - 이를 적절한 data augmentation, learnable nonlinear transformation등을 사용하여 해결하고 기존 supervised 방법과 동일한 성능을 얻는 결과를 보임

Method	Architecture	Param.	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (*)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**



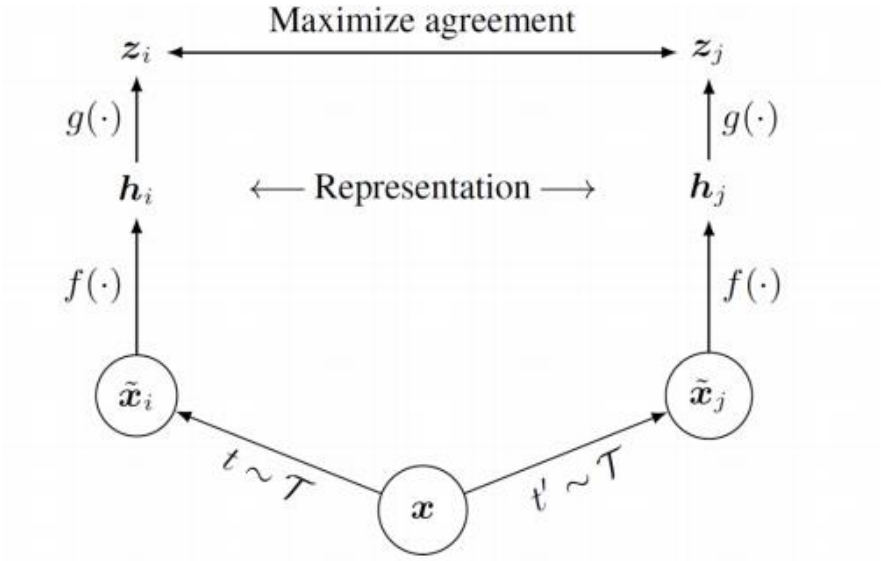
Contrastive learning

- A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)
  - Proposed method
    - 입력 이미지 x를 2가지 augmentation (t, t')을 이용하여 2개의 입력 이미지를 구성
    - Encoder f(.)를 이용하여 이미지의 representation을 추출
      - Resnet의 average pooling 전의 feature를 사용



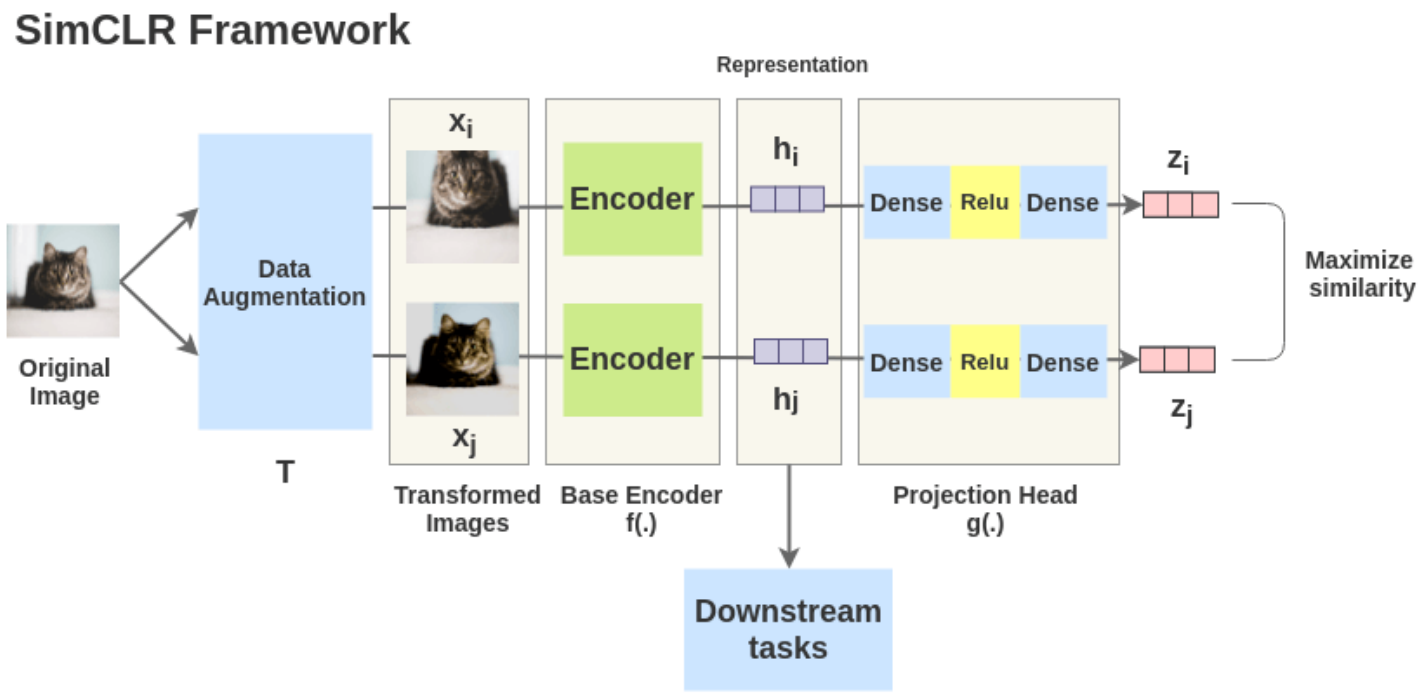
Contrastive learning

- A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)
  - Proposed method
    - 다음으로 represent vector를 non-linear fc로 유사도를 구할 vector로 변환
    - Positive pair경우 유사도를 크게 하고 negative pair일 경우 유사도를 작게 훈련



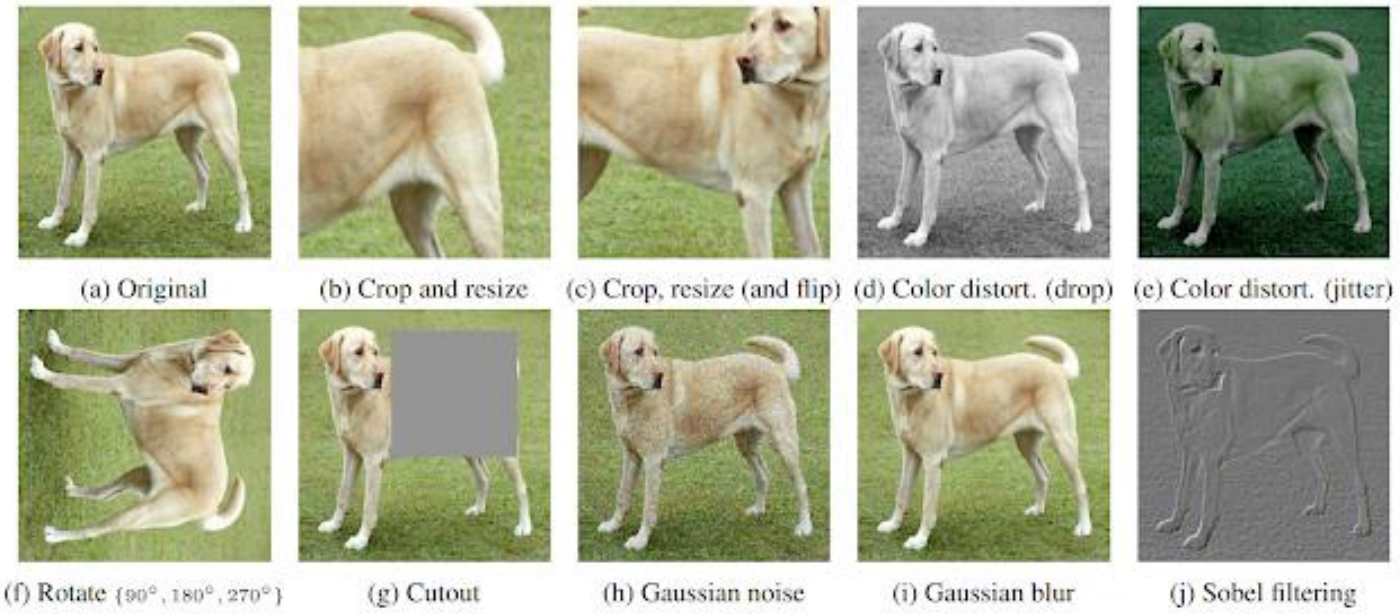
Contrastive learning

- A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)
  - Proposed method



Contrastive learning

- A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)
  - Conclusion
    - Data augmentation 조합의 중요
      - Random cropping과 color distortion두가지 조합이 성능이 가장 높음



Contrastive learning

- A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)
  - Conclusion
    - Non linear projection
      - CNN출력 후의 vector가 동일하기는 어려움
      - 따라서 Non linear projection을 통해 보정
      - 이를 통해 다른 task로 전환시에 큰 성능 향상을 보임
    - Batch size
      - 제안된 방법에서는 augmentation을 이용하여 positive pair밖에 얻을 수 없음
      - 따라서 큰 batch size를 구성하여 1 batch내에 많은 negative 이미지가 존재하도록 하는 것이 성능향상에 도움이 됨

$$NCE_{Loss} = -\log \frac{\exp(\mathrm{sim}(g(\mathrm{x}), g(\mathrm{x}^{+})))}{\exp(\mathrm{sim}(g(\mathrm{x}), g(\mathrm{x}^{+}))) + \sum_{k=1}^{K} \exp(\mathrm{sim}(g(\mathrm{x}), g(\mathrm{x}_{k}^{-})))}$$

Conclusion

- Anomaly detection의 개념과 중요 주제
- 최근 computer vision 논문
- 다른 task의 기술을 Anomaly detection의 적용하는 과정