# Visual Transformers: Token-based Image Representation and Processing for Computer Vision

School of Industrial and Management Engineering, Korea University

Young Jae Lee

KOREA UNIVERSITY

DMQA

# Contents

# Research Purpose

❖ Visual Transformers: Token-based Image Representation and Processing for Computer Vision (arXiv, 2020)

- Facebook, UC Berkeley에서 연구하였고 2021년 07월 02일 기준으로 약 29회 인용

**Visual Transformers: Token-based Image Representation and Processing for Computer Vision**

Bichen Wu[1], Chenfeng Xu[3], Xiaoliang Dai[1], Alvin Wan[3], Peizhao Zhang[1]
Zhicheng Yan[2], Masayoshi, Tomizuka[3], Joseph Gonzalez[3], Kurt Keutzer[3], Peter Vajda[1]
[1] Facebook Reality Labs, [2] Facebook AI, [3] UC Berkeley
{wbc, xiaoliangdai, stzpz, zyan3, vajdap}@fb.com
{xuchenfeng, alvinwan, tomizuka, jegonzal, keutzer}@berkeley.edu

**Abstract**

*Computer vision has achieved remarkable success by (a) representing images as uniformly-arranged pixel arrays and (b) convolving highly-localized features. However, convolutions treat all image pixels equally regardless of importance; explicitly model all concepts across all images, regardless of content; and struggle to relate spatially-distant concepts. In this work, we challenge this paradigm by (a) representing images as semantic visual tokens and (b) running transformers to densely model token relationships. Critically, our* ***Visual Transformer*** *operates in a semantic token space, judiciously attending to different image parts based on context. This is in sharp contrast to pixel-space transformers that require orders-of-magnitude more compute. Using an advanced training recipe, our VTs significantly outperform their convolutional counterparts, raising ResNet accuracy on ImageNet top-1 by* ***4.6 to 7 points*** *while using fewer FLOPs and parameters. For semantic segmentation on LIP and COCO-stuff, VT-based feature pyramid networks (FPN) achieve 0.35 points higher mIoU while reducing the FPN module's FLOPs by* ***6.5x****.*

2) **Not all images have all concepts**: Low-level features such as corners and edges exist in all natural images, so applying low-level convolutional filters to *all* images is appropriate. However, high-level features such as ear shape exist in specific images, so applying high-level filters to all images is computationally inefficient. For example, dog features may not appear in images of flowers, vehicles, aquatic animals etc. This results in rarely-used, inapplicable filters expending a significant amount of compute.

3) **Convolutions struggle to relate spatially-distant concepts**: Each convolutional filter is constrained to operate on a small region, but long-range interactions between semantic concepts is vital. To relate spatially-distant concepts, previous approaches increase kernel sizes, increase model depth, or adopt new operations like dilated convolutions, global pooling, and non-local attention layers. However, by working within the pixel-convolution paradigm, these approaches at best mitigate the problem, compensating for the convolution's weaknesses by adding model and computational complexity.

To overcome the above challenges, we address the root



DMQA

# Research Purpose

❖ Visual Transformers: Token-based Image Representation and Processing for Computer Vision (arXiv, 2020)

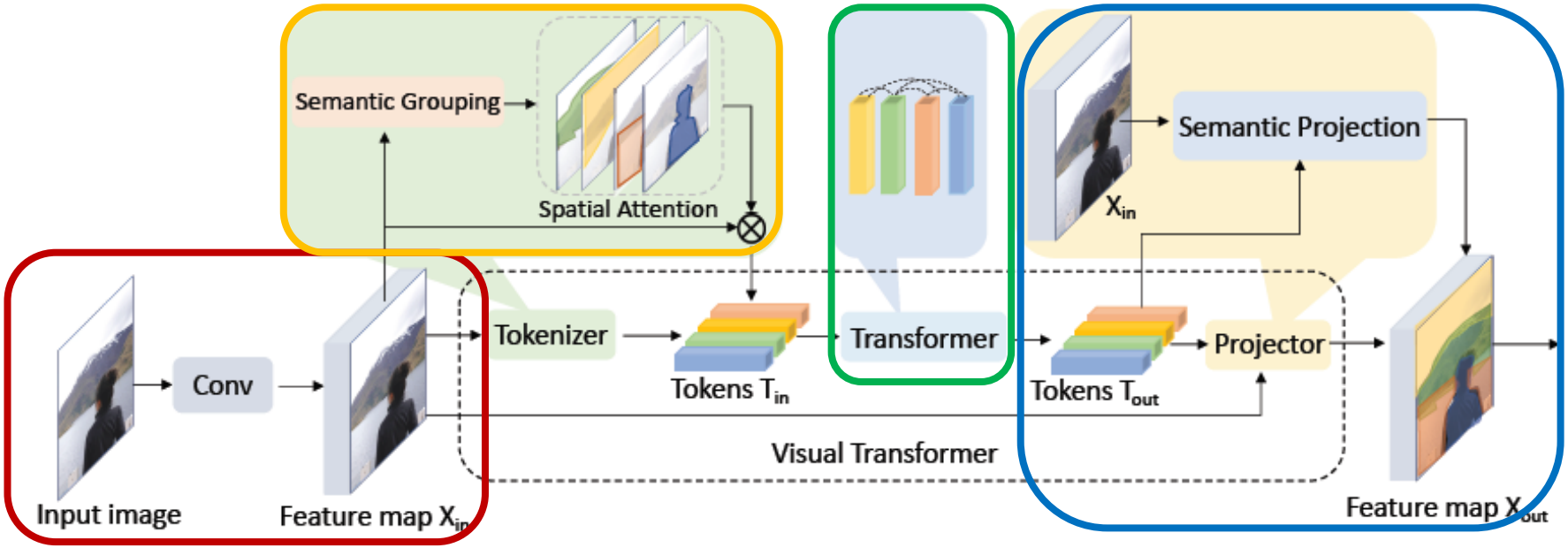
- Computer Vision에서 Convolution 연산의 문제점 제시
  - ✓ Not all pixels are created equal: 중요도와 관계없이 모든 이미지 픽셀들을 동등하게 다루는 문제
  - ✓ Not all images have all concepts: 특정 이미지에만 있는 High-Level Features (Dog Ear Shape) 필터 적용의 연산 비효율성
  - ✓ Convolutions struggle to relate spatially-distant concepts: Kernel Size, Model Depth 증가 등 계산 복잡성을 추가하여 Convolution 약점 보완

- 이미지의 High-Level Concept 처리 및 표현을 위해 Visual Transformers (VTs) 제안

# Visual Transformers (VTs)

❖ Diagram of a Visual Transformer (VT)

- **First**: Convolution 연산을 적용하여 Low-Level Feature를 추출 → Feature Map
- **Second**: Feature Map을 Tokenizer에 적용
  - ✓ 픽셀을 적은 수(16)의 Visual Tokens으로 그룹화하여 각각 이미지의 Semantic Concept을 나타냄
- **Third**: Tokens 사이의 관계 파악을 위해 Transformer 적용
- **Final**: Image Classification or Semantic Segmentation 수행



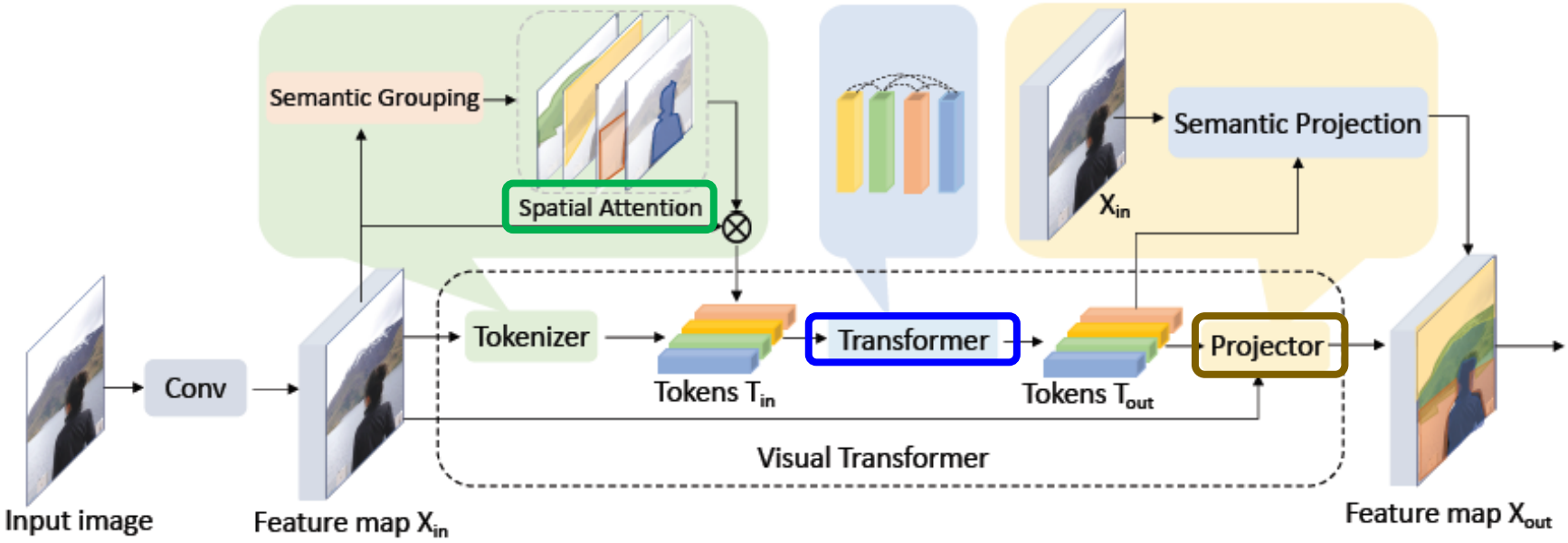



DMQA

# Visual Transformers (VTs)

❖ Diagram of a Visual Transformer (VT)

- **Spatial Attention**: Convolution 연산으로부터 출력한 Feature Map을 Compact한 Semantic Tokens 집합으로 변환
- **Transformer**: Spatial Attention으로부터 처리한 Token들 간 상호관계를 Self-Attention Module로 파악
- Image Classification을 위해 Tokens을 직접 사용 또는 Semantic Segmentation을 위해 Feature Map에 다시 **투영(Project)**하여 Task를 수행



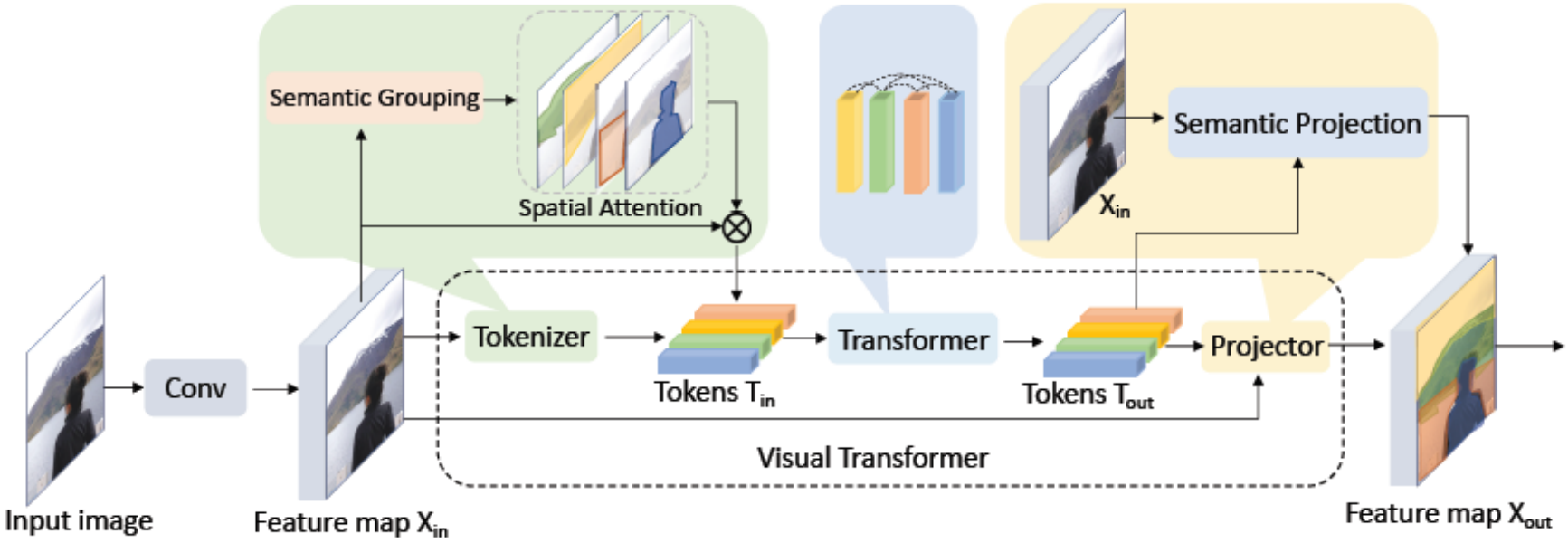



DMQA

# Visual Transformers (VTs)

❖ Diagram of a Visual Transformer (VT)

- Convolution 연산과 달리 VT는 세 가지 문제를 잘 처리할 수 있음
  - ✓ 모든 픽셀을 동등하게 처리하는 대신 중요한 지역에 주의를 기울여 계산을 신중하게 할당
  - ✓ 이미지와 관련된 Visual Tokens들로 Semantic Concept을 인코딩
  - ✓ Token-Space에서 Self-Attention을 통해 공간적으로 먼 Concept을 연관 지을 수 있음



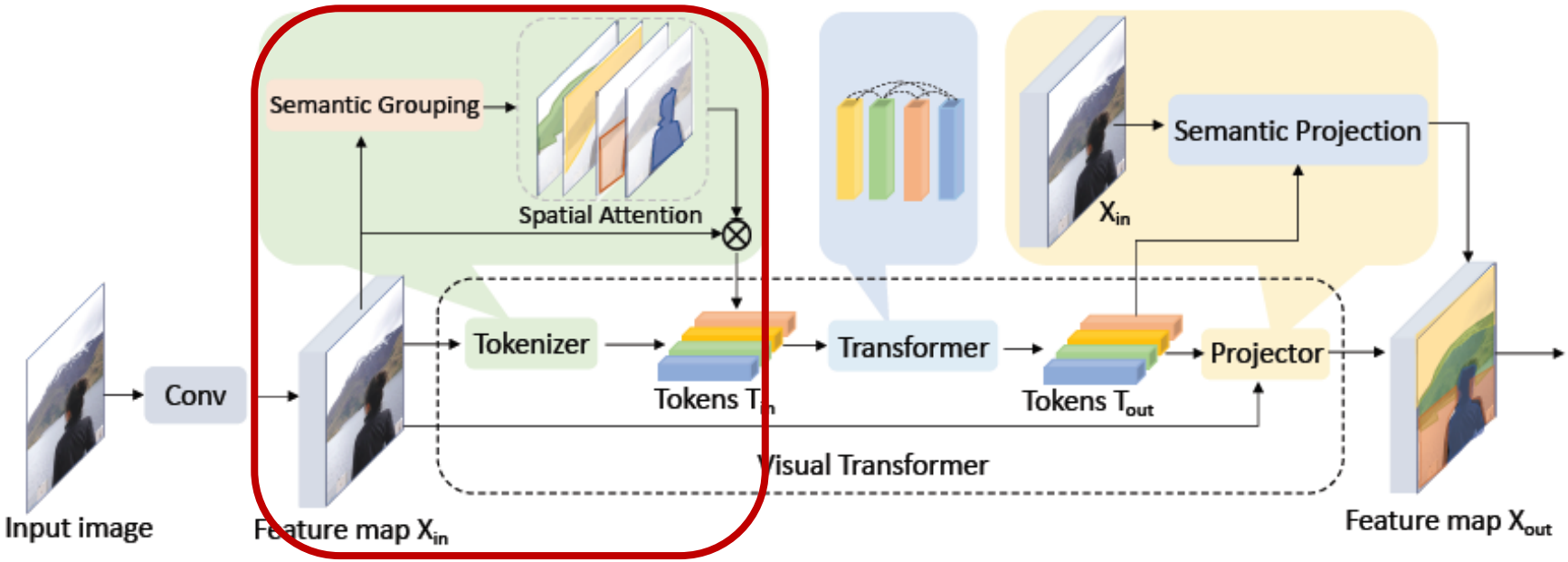



DMQA

# Visual Transformers (VTs)

❖ Tokenizer

- 이미지는 Visual Token으로 요약할 수 있음
  - ✓ Feature Map을 Compact한 Semantic Visual Tokens 집합으로 변환
- Represented as follows:
  - ✓ $\boldsymbol{X} \in \mathbb{R}^{HW \times C}$: Feature Map ($H$: height, $W$: width, $C$: channels) / $\boldsymbol{T} \in \mathbb{R}^{L \times C}$: Visual Tokens ($L$: number of tokens)
  - ✓ S.T.: $L \ll HW$





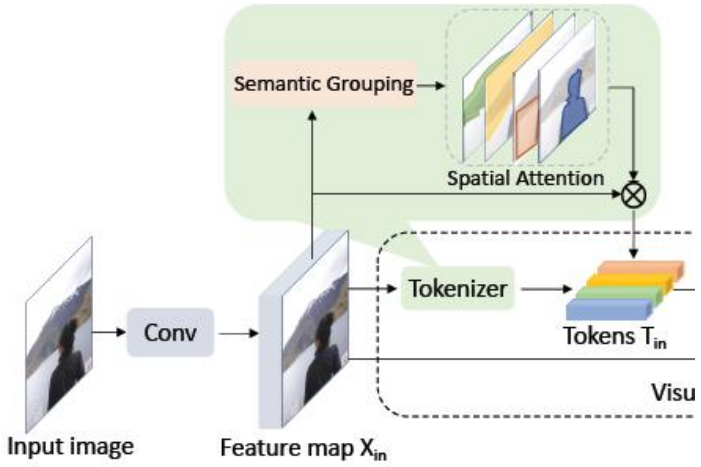

DMQA

# Visual Transformers (VTs)

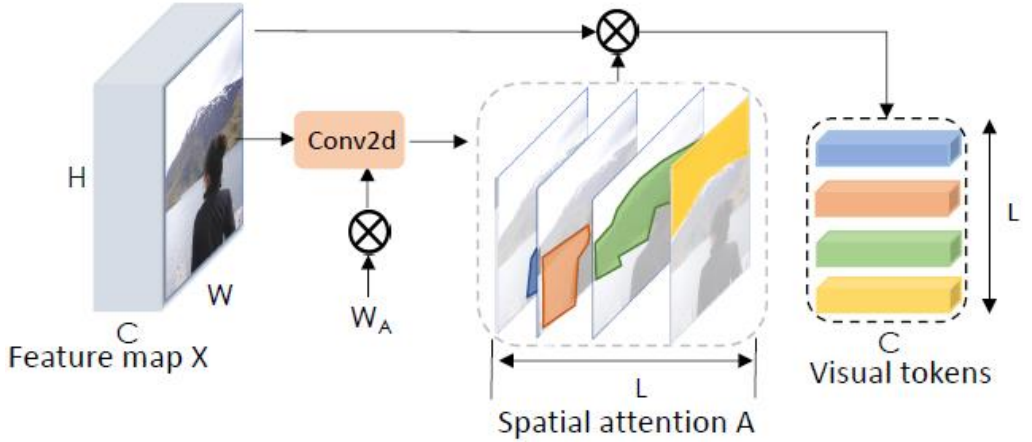
## ❖ Tokenizer 1. Filter-based Tokenizer

- Feature Map **X** 경우, Point-wise Convolution 연산으로 각 픽셀 $\boldsymbol{X}_p$를 $L$ Semantic Group 중 하나에 매핑
- 각 Semantic Group 내에서 공간적으로 픽셀을 풀링하여 Token **T**를 얻음
- Represented as follows:
  - ✓ $\mathbf{T} = \boldsymbol{softmax_{HW}(XW_A)^T}X = \boldsymbol{A}X$ $(softmax_{HW}(XW_A)^T = A \in \mathbb{R}^{HW \times L})$
  - ✓ $W_A \in \mathbb{R}^{C \times L}$: $X$에서 Semantic Group을 형성 / $softmax_{HW}(\cdot)$: Activations을 Spatial Attention으로 변환
  - ✓ $A$는 $X$와 곱하고 $X$에서 픽셀의 가중 평균을 계산하여 $L$개의 Visual Token을 생성

General Tokenizer




Filter-based Tokenizer





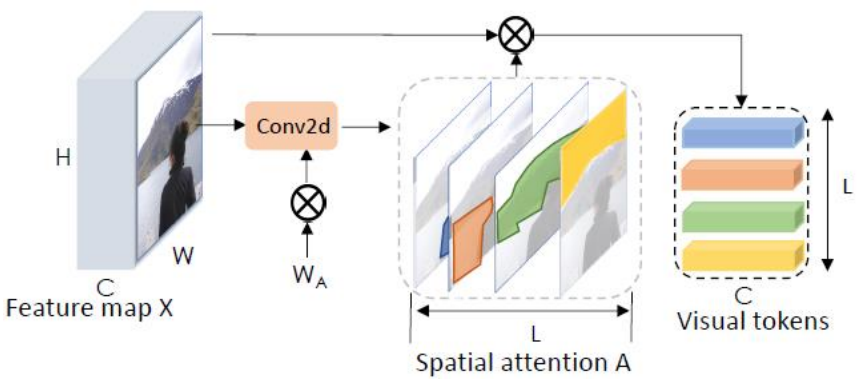

DMQA

# Visual Transformers (VTs)

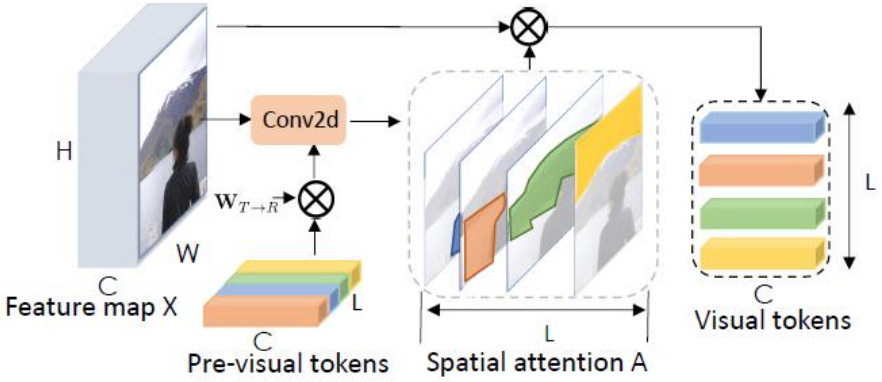
❖ Tokenizer 2. Recurrent Tokenizer

- Filter-based Tokenizer의 한계를 해결하기 위해 이전 레이어의 Visual Tokens에 의존하는 가중치를 가진 Recurrent Tokenizer를 제안(현재 Token이 이전 Token에 따라 계산된다는 점)
- 이전 레이어의 Token $\mathbf{T}_{in}$ 이 현재 레이어의 New Token 추출의 가이드 역할
- VT는 이전에 처리된 Concept에 따라 Visual Tokens 세트를 점진적으로 개선할 수 있음
- Represented as follows:
  - ✓ $\mathbf{T} = softmax_{HW}(X\boldsymbol{W_R})^T X,$ $\boldsymbol{W_R = T_{in} W_{T\to R}},$ $W_{T\to R} \in \mathbb{R}^{C\times C}$

Filter-based Tokenizer




Recurrent Tokenizer



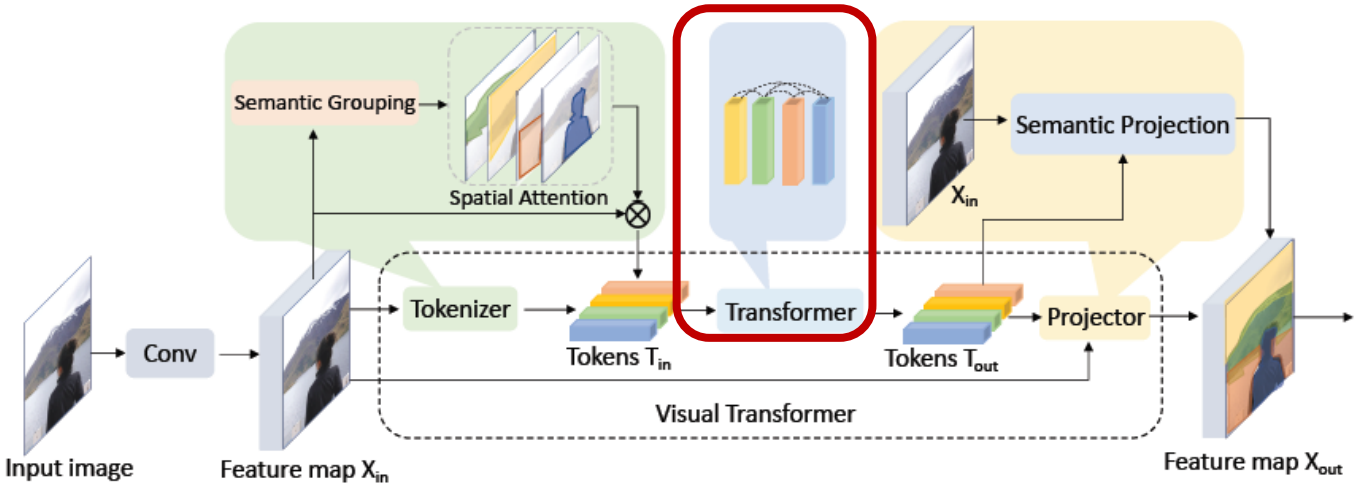



DMQA

# Visual Transformers (VTs)

❖ Transformer

- Tokenizer 적용 후 Visual Token들 간의 상호작용을 모델링하는 역할
- Transformer에서 Token 간의 가중치는 Input에 따라 다르며, Key-Query Product로 $\mathbf{T}_{in}K(\mathbf{T}_{in}Q)^T \in \mathbb{R}^{L\times L}$을 계산 → 적은 수(16)의 Visual Token 사용 가능
- Represented as follows:
  - ✓ $\mathbf{T}'_{out} = \mathbf{T}_{in} + softmax_L((\mathbf{T}_{in}K(\mathbf{T}_{in}Q)^T)\mathbf{T}_{in}$
  - ✓ $\mathbf{T}_{out} = \mathbf{T}'_{out} + \sigma(\mathbf{T}'_{out}\mathbf{F}_1)\mathbf{F}_2$ → 비선형 함수와 2개의 Point-wise Convolution 사용$(\mathbf{F}_1, \mathbf{F}_2)$
  - ✓ $\mathbf{T}_{in}, \mathbf{T}_{out}, \mathbf{T}'_{out} \in \mathbb{R}^{L\times C}$: Visual Tokens



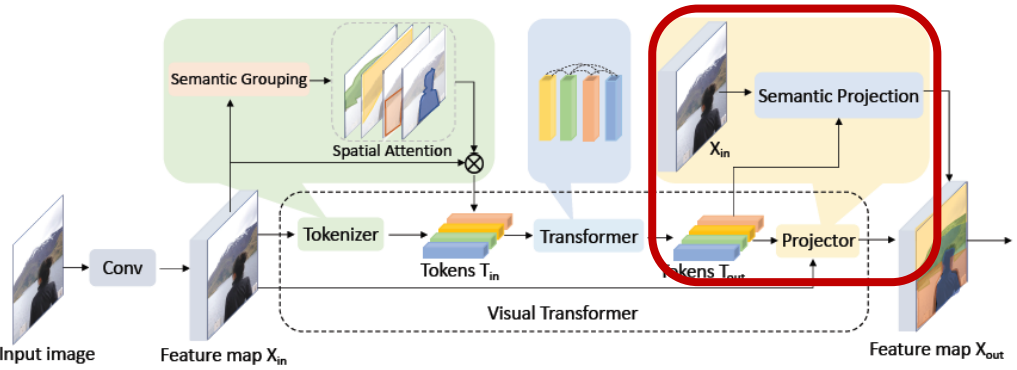



DMQA

# Visual Transformers (VTs)

## ❖ Projector

- Vision Tasks에는 pixel-level의 세부 정보가 필요하지만 Visual Tokens에는 보존되지 않음
- Transformer의 출력물을 Feature Map과 융합하여 Feature Map의 pixel-array 표현을 구체화 함
- Represented as follows:
  - ✓ $\mathbf{X}_{out} = \mathbf{X}_{in} + softmax_L\left((\mathbf{X}_{in}\mathbf{W}_Q)(\mathbf{T}\mathbf{W}_K)^T\right)$ $\mathbf{T}$: Pixel-array representation
  - ✓ $\mathbf{X}_{in}, \mathbf{X}_{out} \in \mathbb{R}^{HW\times C}$: Input and Output Feature Map
  - ✓ $\mathbf{X}_{in}\mathbf{W}_Q \in \mathbb{R}^{HW\times C}$: Input Feature Map $\mathbf{X}_{in}$에서 계산된 Query / $\mathbf{T}\mathbf{W}_K \in \mathbb{R}^{L\times C}$: Token $\mathbf{T}$에서 계산된 Key
  - ✓ $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{C\times C}$: Key, Query를 계산하는데 사용되는 학습 가능한 가중치
  - ✓ Key-Query Product는 Visual Token T로 인코딩된 정보를 원래 Feature Map에 투영(Projection)






DMQA

# Experiments

❖ Image Classification Metric

- Top-1 Accuracy: Softmax의 Output에서 제일 높은 수치를 가지는 값이 정답일 경우에 대한 지표
- Float Point Operations Per Second (FLOPs): 컴퓨터의 성능을 표현하는 지표
- Parameters: Model의 Weight 또는 Parameter 수

❖ Semantic Segmentation Metric

- Mean Intersection over Union (mIoU): 예측 및 실제 픽셀 간 교집합에 포함되는 정도에 대한 지표
- Float Point Operations Per Second (FLOPs): 컴퓨터의 성능을 표현하는 지표

# Experiments

## Image Classification Results

❖ Visual Transformer (VT) vs. ResNet with default training recipe

| | Top-1 Acc (%) (Val) | Top-1 Acc (%) (Train) | FLOPs (M) | Params (M) |
|---|---|---|---|---|
| R18 | 69.9 | 68.6 | 1814 | 11.7 |
| VT-R18 | **72.1** | 76.5 | 1570 | 11.7 |
| R34 | 73.3 | 73.9 | 3664 | 21.8 |
| VT-R34 | **75.0** | 80.8 | 3280 | 21.9 |

Table 2: VT-ResNet vs. baseline ResNets on the ImageNet dataset. By replacing the last stage of ResNets, VT-ResNet uses 224M, 384M fewer FLOPs than the baseline ResNets while achieving 1.7 points and 2.2 points higher validation accuracy. Note the training accuracy of VT-ResNets are much higher. This indicates VT-ResNets have higher model capacity and require stronger regularization (e.g., data augmentation) to fully utilize the model. See Table 8.

| Models | Top-1 Acc (%) | FLOPs (G) | Params (M) |
|---|---|---|---|
| R18[14] | 69.8 | 1.814 | 11.7 |
| R18+SE[21, 41] | 70.6 | 1.814 | 11.8 |
| R18+CBAM[41] | 70.7 | 1.815 | 11.8 |
| LR-R18[19] | 74.6 | 2.5 | 14.4 |
| R18[14](ours) | 73.8 | 1.814 | 11.7 |
| VT-R18(ours) | **76.8** | **1.569** | **11.7** |
| R34[14] | 73.3 | 3.664 | 21.8 |
| R34+SE[21, 41] | 73.9 | 3.664 | 22.0 |
| R34+CBAM[41] | 74.0 | 3.664 | 22.9 |
| AA-R34[1] | 74.7 | 3.55 | 20.7 |
| R34[14](ours) | 77.7 | 3.664 | 21.8 |
| VT-R34(ours) | **79.9** | **3.236** | **19.2** |
| R50[14] | 76.0 | 4.089 | 25.5 |
| R50+SE[21, 41] | 76.9 | 3.860* | 28.1 |
| R50+CBAM[41] | 77.3 | 3.864* | 28.1 |
| LR-R50[19] | 77.3 | 4.3 | 23.3 |
| Stand-Alone[30] | 77.6 | 3.6 | **18.0** |
| AA-R50[1] | 77.7 | 4.1 | 25.6 |
| $A^2$-R50[5] | 77.0 | - | - |
| SAN19[49] | 78.2 | **3.3** | 20.5 |
| GloRe-R50[6] | 78.4 | 5.2 | 30.5 |
| VT-R50(ours) | **80.6** | 3.412 | 21.4 |
| R101[14] | 77.4 | 7.802 | 44.4 |
| R101+SE [21, 41] | 77.7 | 7.575* | 49.3 |
| R101+CBAM[41] | 78.5 | 7.581* | 49.3 |
| LR-R101[19] | 78.5 | 7.79 | 42.0 |
| AA-R101[1] | 78.7 | 8.05 | 45.4 |
| GloRe-R200[6] | 79.9 | 16.9 | 70.6 |
| VT-R101(ours) | **82.3** | **7.129** | **41.5** |

Table 8: Comparing VT-ResNets with other attention-augmented ResNets on ImageNet. *The baseline ResNet FLOPs reported in [41] is lower than our baseline.

# Experiments

## Image Classification Results

### ❖ Tokenizer Ablation Studies

| | | Top-1 Acc (%) | FLOPs (M) | Params (M) |
|---|---|---|---|---|
| R18 | Pooling-based | 70.5 | 1549 | 11.0 |
| | Clustering-based | 71.8 | 1579 | 11.6 |
| | Filter-based | **72.1** | 1580 | 11.7 |
| R34 | Pooling-based | 73.6 | 3246 | 20.6 |
| | Clustering-based | **75.2** | 3299 | 21.8 |
| | Filter-based | 74.9 | 3280 | 21.9 |

Table 3: VT-ResNets using with different types of tokenizers. Pooling-based tokenizers spatially downsample a feature map to obtain visual tokens. Clustering-based tokenizer (Appendix C) groups pixels in the semantic space. Filter-based tokenizers (3.1.1) use convolution filters to group pixels. Both filter-based and cluster-based tokenizers work much better than pooling-based tokenizers, validating the importance of grouping pixels by their semantics.

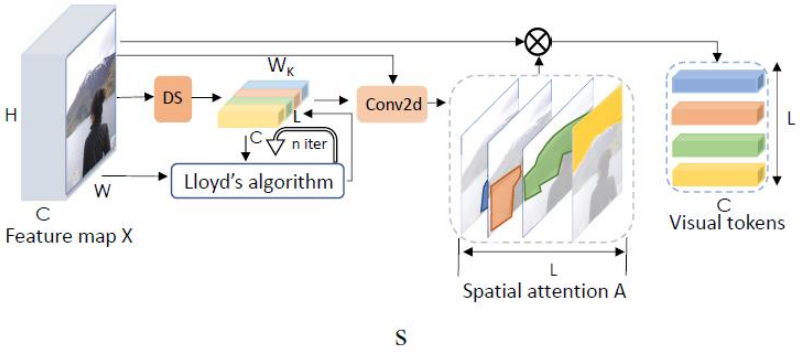



Figure 6: Cluster-based tokenizer that group pixels using the K-Means centroids of the pixels in the semantic space.

# Experiments

## Image Classification Results

❖ Modeling Token Relationships / Token Efficiency Ablation / Projection Ablation

| | | Top-1 Acc (%) | FLOPs (M) | Params (M) |
|---|---|---|---|---|
| R18 | None | 68.7 | 1528 | 8.5 |
| | GraphConv | 69.3 | 1528 | 8.5 |
| | Transformer | **71.5** | 1580 | 11.7 |
| R34 | None | 73.3 | 3222 | 17.1 |
| | GraphConv | 73.7 | 3223 | 17.1 |
| | Transformer | **75.2** | 3299 | 21.8 |

Table 5: VT-ResNets using different modules to model token relationships. Models using transformers perform better than graph-convolution or no token-space operations. This validates that it is important to model relationships between visual token (semantic concepts) and transformer work better than graph convolution in relating tokens.

| | No. Tokens | Top-1 Acc (%) | FLOPs (M) | Params (M) |
|---|---|---|---|---|
| R18 | 16 | 71.8 | 1579 | 11.6 |
| | 32 | 71.9 | 1711 | 11.6 |
| | 64 | **72.1** | 1979 | 11.6 |
| R34 | 16 | **75.1** | 3299 | 21.8 |
| | 32 | 75.0 | 3514 | 21.8 |
| | 64 | 75.0 | 3952 | 21.8 |

Table 6: Using more visual tokens do not improve the accuracy of VT, which agrees with our hypothesis that images can be described by a compact set of visual tokens.

| | | Top-1 Acc (%) | FLOPs (M) | Params (M) |
|---|---|---|---|---|
| R18 | w/ projector | **72.0** | 1569 | 11.7 |
| | w/o projector | 71.0 | 1498 | 9.4 |
| R34 | w/ projector | **74.8** | 3280 | 21.9 |
| | w/o projector | 73.9 | 3159 | 17.4 |

Table 7: VTs that projects tokens back to feature maps perform better. This may be because feature maps still encode important spatial information.

# Experiments

## Semantic Segmentation Results

❖ Visual Transformer Feature Pyramid Networks (VT-FPN)

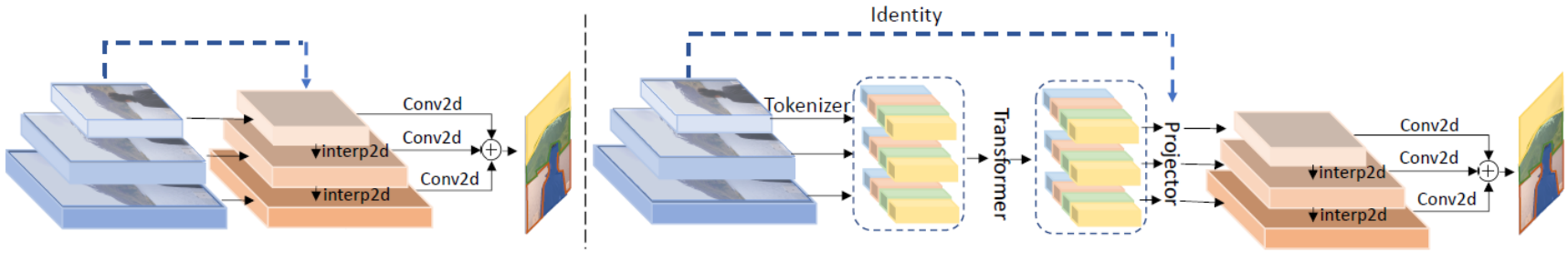



Figure 4: Feature Pyramid Networks (FPN) (left) vs visual-transformer-FPN (VT-FPN) (right) for semantic segmentation. FPN uses convolution and interpolation to merge feature maps with different resolutions. VT-FPN extraxt visual tokens from all feature maps, merge them with one transformer, and project back to the original feature maps.

| | | mIoU (%) | Total FLOPs (G) | FPN FLOPs (G) |
|---|---|---|---|---|
| R-50 | FPN | 40.78 | 159 | 55.1 |
| | VT-FPN | 41.00 | 113 (1.41x) | 8.5 (6.48x) |
| R-101 | FPN | 41.51 | 231 | 55.1 |
| | VT-FPN | 41.50 | 185 (1.25x) | 8.5 (6.48x) |

Table 9: Semantic segmentation results on the COCO-stuff validation set. The FLOPs are calculated with a typical input resolution of 800×1216.

| | | mIoU (%) | Total FLOPs (G) | FPN FLOPs (G) |
|---|---|---|---|---|
| R50 | FPN | 47.04 | 37.1 | 12.8 |
| | VT-FPN | 47.39 | 26.4 (1.41x) | 2.0 (6.40x) |
| R101 | FPN | 47.35 | 54.4 | 12.8 |
| | VT-FPN | 47.58 | 43.6 (1.25x) | 2.0 (6.40x) |

Table 10: Semantic segmentation results on the Look Into Person validation set. The FLOPs are calculated with a typical input resolution of 473×473.

# Experiments

## Semantic Segmentation Results

❖ Visual Transformer Feature Pyramid Networks (VT-FPN)

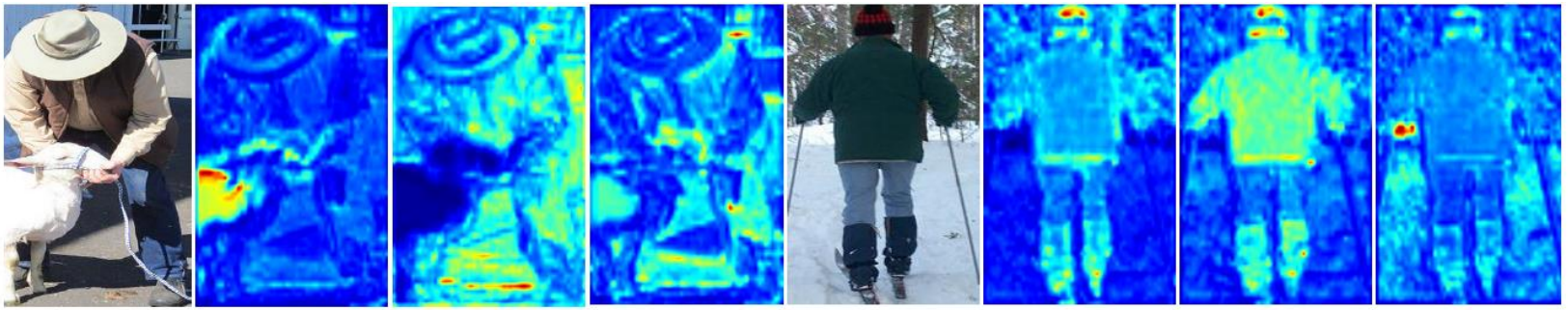


Figure 5: Visualization of the spatial attention generated by a filter-based tokenizer on images from the LIP dataset. Red denotes higher attention values and color blue denotes lower. Without any supervision, visual tokens automatically focus on different areas of the image that correspond to different semantic concepts, such as sheep, ground, clothes, woods.

# Conclusion

- Computer Vision에서 Convolution 연산의 문제점을 제기하면서 Visual Transformer를 제안
- Visual Transformer는 이미지의 High-Level Concept을 처리하고 잘 표현할 수 있음
- 특히, Visual Tokens라는 개념을 활용하여 특정 이미지에서 추출할 수 있는 High-Level Feature를 잘 포착하도록 함
- Image Classification & Semantic Segmentation에 제안 모델을 사용할 수 있으며 Top-1 Accuarcy / mIoU에서 우수한 성능을 보이지만 계산의 복잡도는 높아짐(FLOPs는 감소)
  - 후기: Convolution 연산 문제점을 제시하면서 새로운 High-Level Feature Extraction 패러다임 제안은 좋았음. Visual Tokens 활용 아이디어는 좋았으나 모델 복잡도를 해결할 수 있는 방안이 필요함. Distillation Method를 활용하여 Self-Attention의 복잡도를 줄이고 성능을 유지하는 방향도 고려해볼 법 함.

- Reference
  
  - Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., ... & Vajda, P. (2020). Visual transformers: Token-based image representation and processing for computer vision. arXiv preprint arXiv:2006.03677.
  


Thank You