Saturday, 06-04-2024

CSE, IIT Indore

INSTRUCTOR

Dr. Puneet Gupta

PaCa-ViT: Learning Patch-to-Cluster Attention in ViT

STUDENT

Amit Kumar Makkad - 200001003
Mihir Karandikar - 200001044
Mukul Jain - 200001050
Nilay Ganvit - 200001053

Transformer

- The Transformer is a powerful deep learning architecture introduced in the paper "Attention is All You Need" [1]. It is widely used in tasks such as machine translation, text summarization, and language understanding.

- The Transformer architecture consists of an encoder and a decoder. The encoder processes the input sequence, capturing relevant information, while the decoder generates the output sequence based on the encoder's representations.

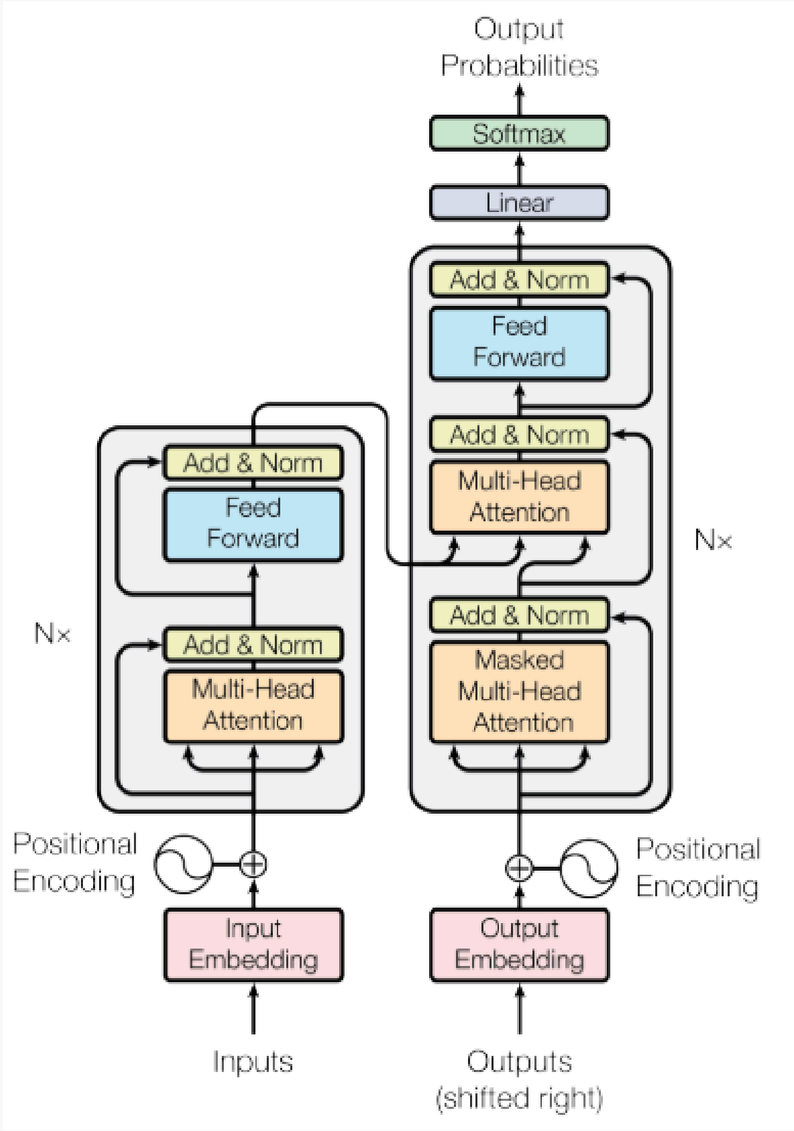


Fig: Architecture[1]

Attention

- Both components utilize self-attention mechanisms to weigh the importance of different tokens in the sequence, allowing for parallel processing and efficient learning.

- It involves three components: Query (Q), Key (K), and Value (V). The similarity scores between query-key pairs determine the attention weights, which are used to compute a weighted sum of values.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

- As Each head learns different representations of input sequence, Multi Head Attention enables model to capture diverse relationships.

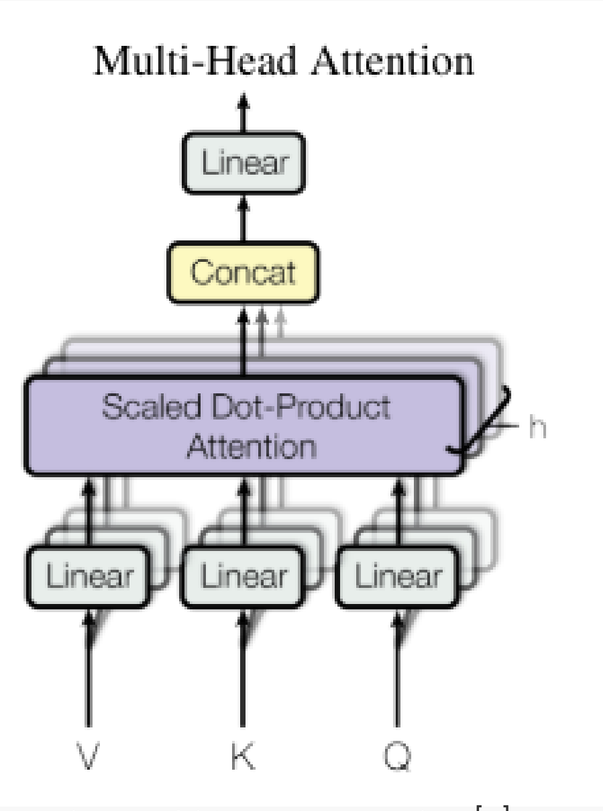


Fig: Multi head Attention[1]

ViT

- In Vision Transformers (ViTs), images are processed by treating them as sequences of "visual tokens," achieved through patch embedding.

- Self-attention in ViTs operates by relating each token to others through the concepts of query, key, and value, enabling comprehensive understanding of inter-token relationships within the image.

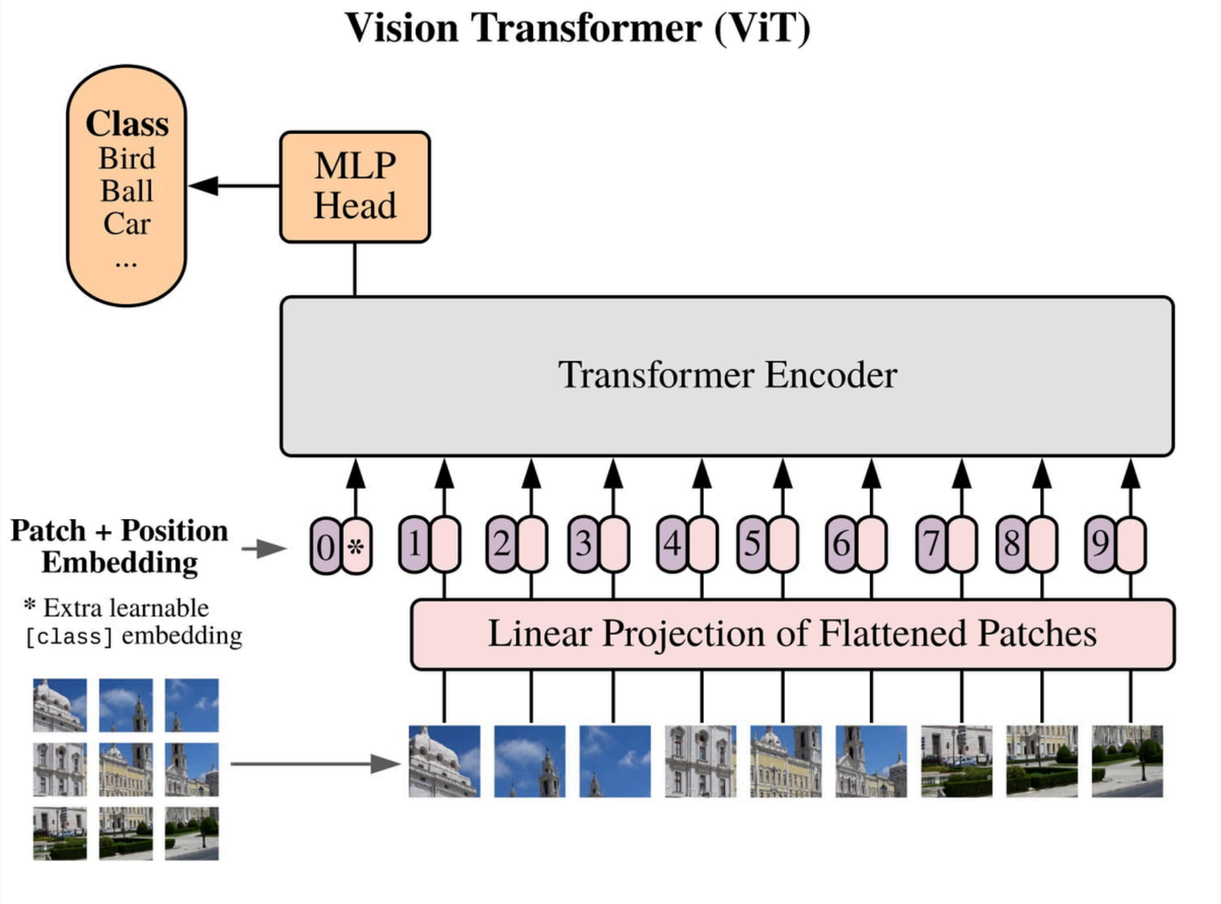
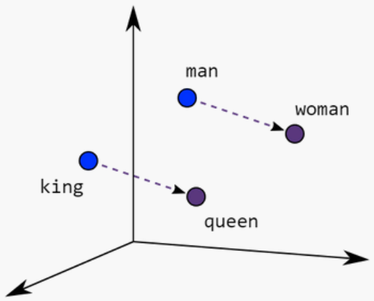


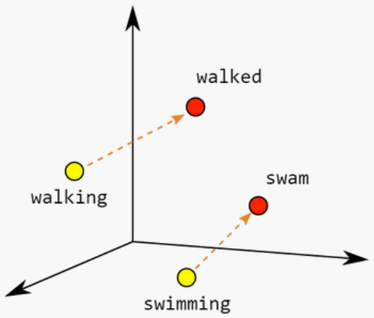
Fig: VIT Architecture[2]

Images as Embeddings in a vector space

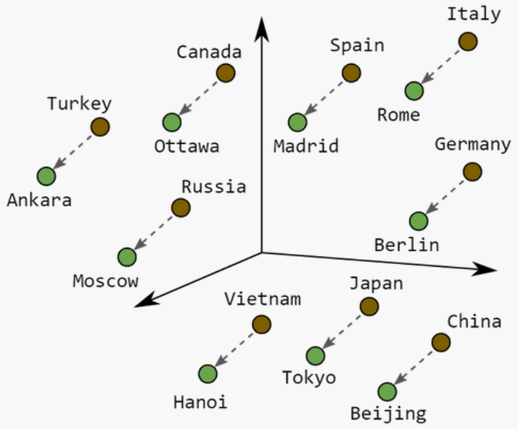
In the same way that words can be turned into vectors to represent their meaning in text, images can also be converted into meaningful vectors. This helps models understand and learn visual context accurately.



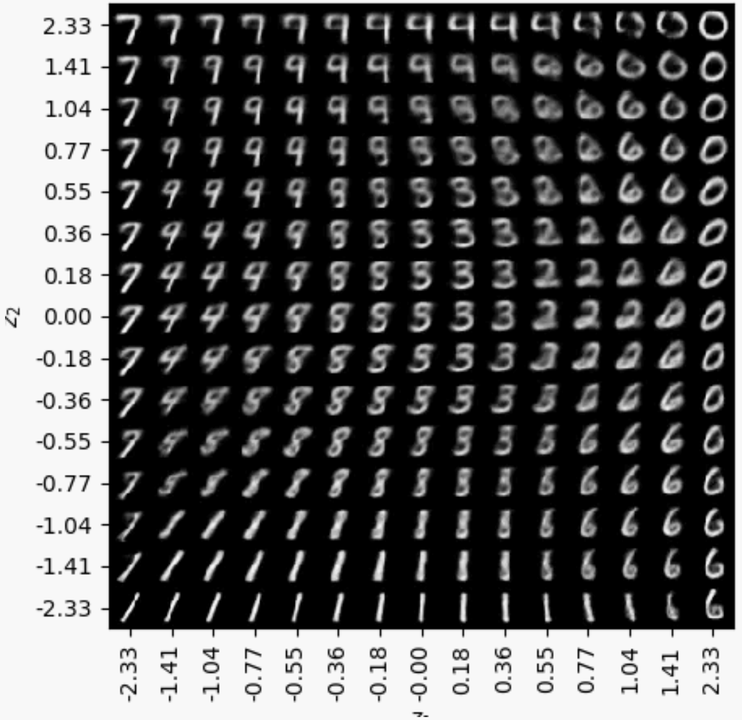
Male-Female



Verb Tense



Country-Capital



PaCa-ViT

- In Patch-to-Cluster Attention Vision Transformers (PaCa-ViT), the clusters are learned end-to-end, leading to better tokenizers and inducing joint clustering-for-attention and attention-for-clustering for better and interpretable models.

- This reduces complexity, facilitating a better visual tokenizer and enabling simple forward explainability.

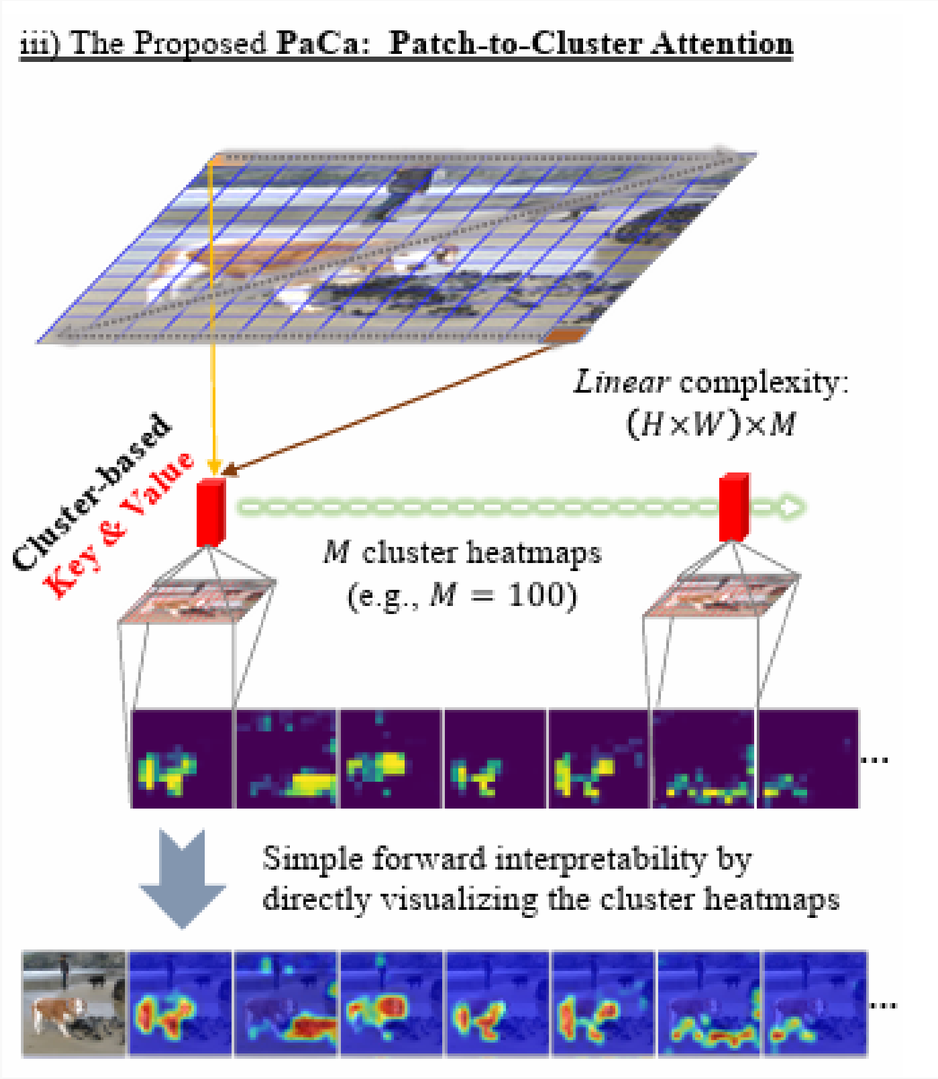


Fig: PaCa-ViT[3]

Applications

Image Classification

Dataset:

ImageNet-1k

CIFAR 10

Object Detection

Dataset:

MS-COCO

Semantic Segmentation

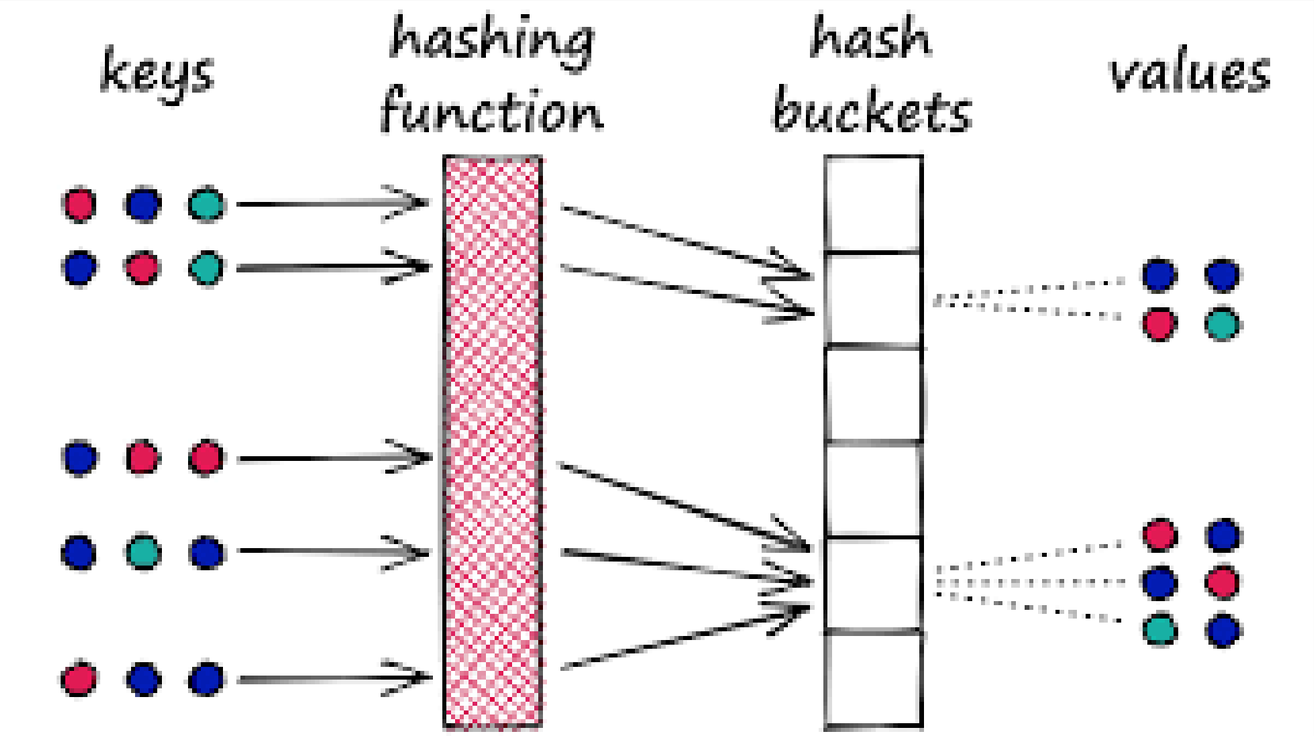
Dataset:

MIT-ADE20k

Fig: Applications of ViT[4]

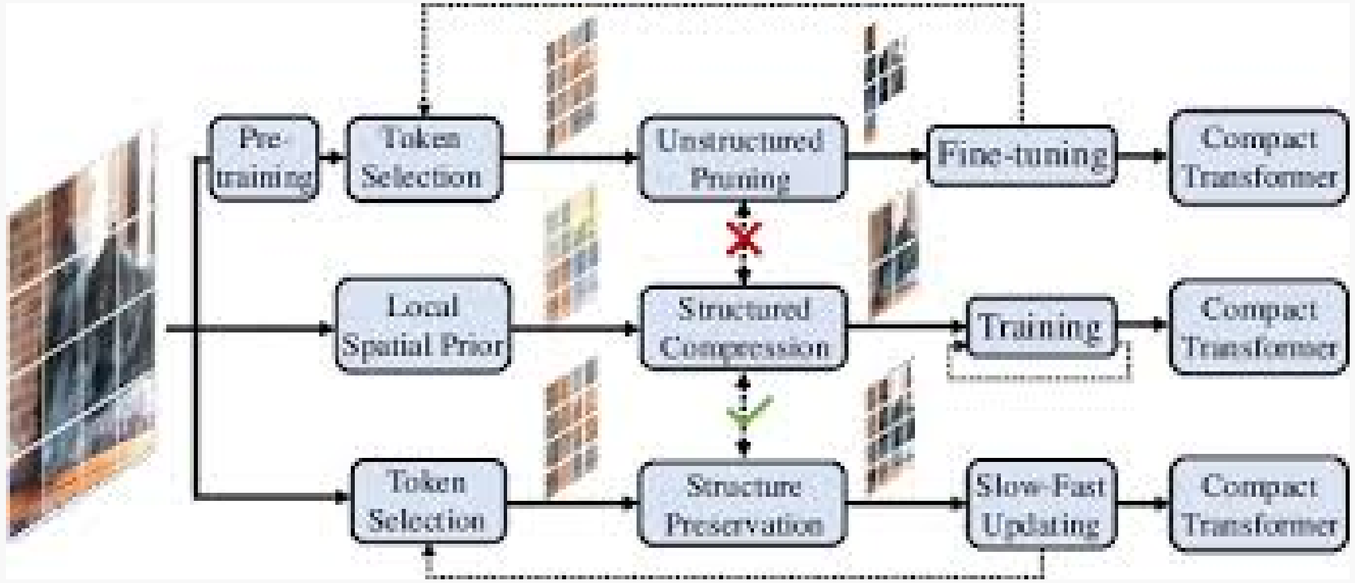
Related Work

- Some Transformer variants use inductive bias, such as local window partitioning, random sparse patterns [5], and locality-sensitive hashing (LSH) in Reformer [6], to enhance computational and model performance. However, these methods may limit the self-attention layer's capacity due to their focus on local constraints.



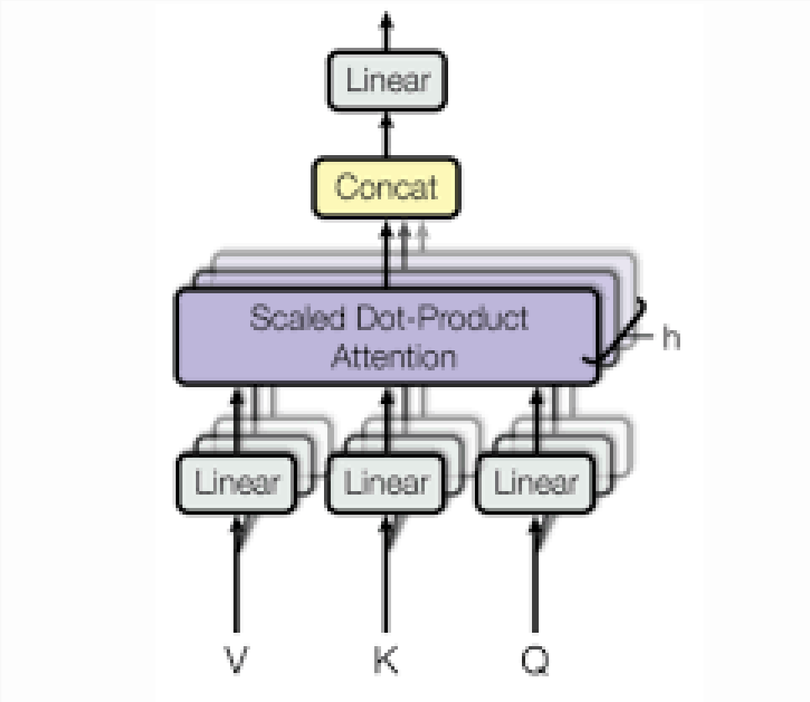
Related Work

- Evo-ViT[7] selects top-k informative tokens to reduce self-attention complexity, achieving linear complexity but primarily for image classification. Its extension to tasks like object detection and semantic segmentation remains unclear.



Related Work

- Some methods, like LinFormer[8], use low-rank projections to create a coarser representation of the input sequence, which can perform well for certain NLP tasks. However, they may lose high-fidelity token-wise information and may not perform as well on tasks requiring fine-grained local information compared to full or sparse attention approaches.



Related Work

- Recently, there has been interest in explaining vanilla isotropic ViT models using attention scores. But, As pointed out in the Improved LRP[9], reducing the explanation to only the attentions scores may be myopic since many other components are ignored.

MLP Clustering

- Identifies patterns and structures in the input data without requiring labeled data.
- It employs multilayer perceptrons to perform non-linear transformations of input features.
- It can capture complex relationships and patterns in data by learning hierarchical representations through multiple layers of neurons.

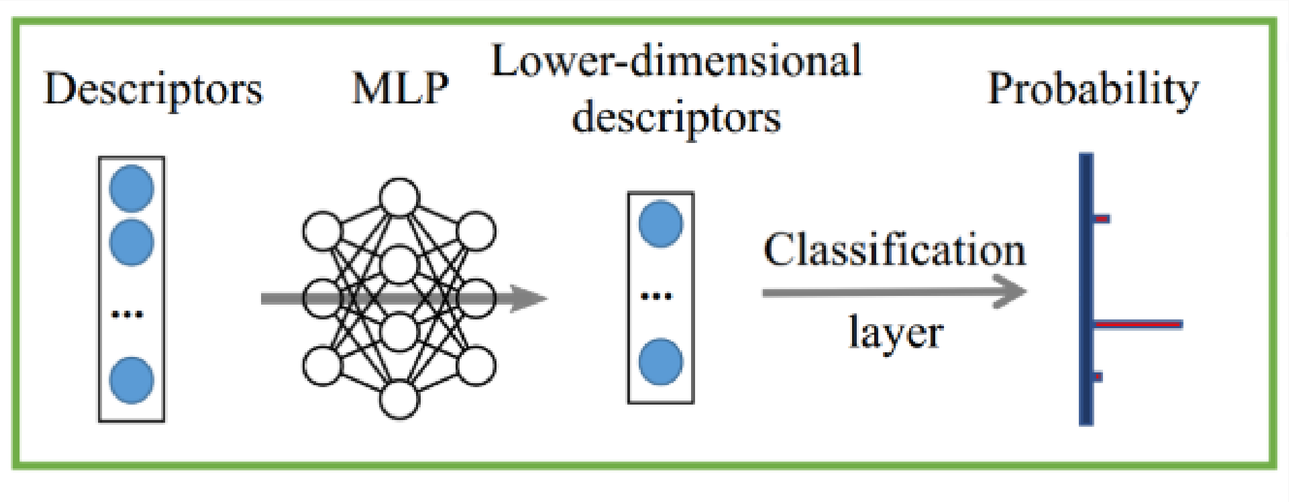


Fig: MLP Clustering[10]

KMeans Clustering

- Centroid Based

- It is an unsupervised algorithm that clusters the input data points into K different clusters groups based on patterns present in the data.

- We start with K random cluster centroids, and in every iteration, we assign the data points to the closest centroid and then recalculate the cluster centroids.

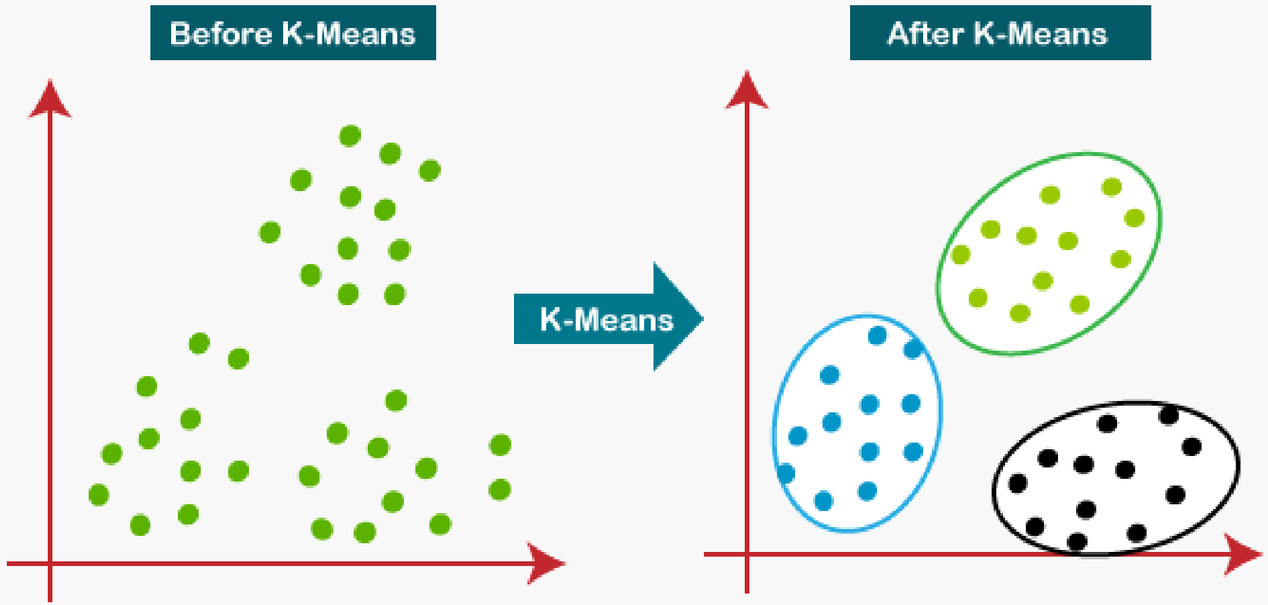


Fig: KMeans Clustering

Hierarchical Clustering

- Hierarchy Based
- It is an unsupervised machine learning technique used to group similar data points together into clusters based on a measure of similarity.
- This method starts with each data point as its own cluster and gradually merges clusters that are most similar to each other until all data points belong to a single cluster.

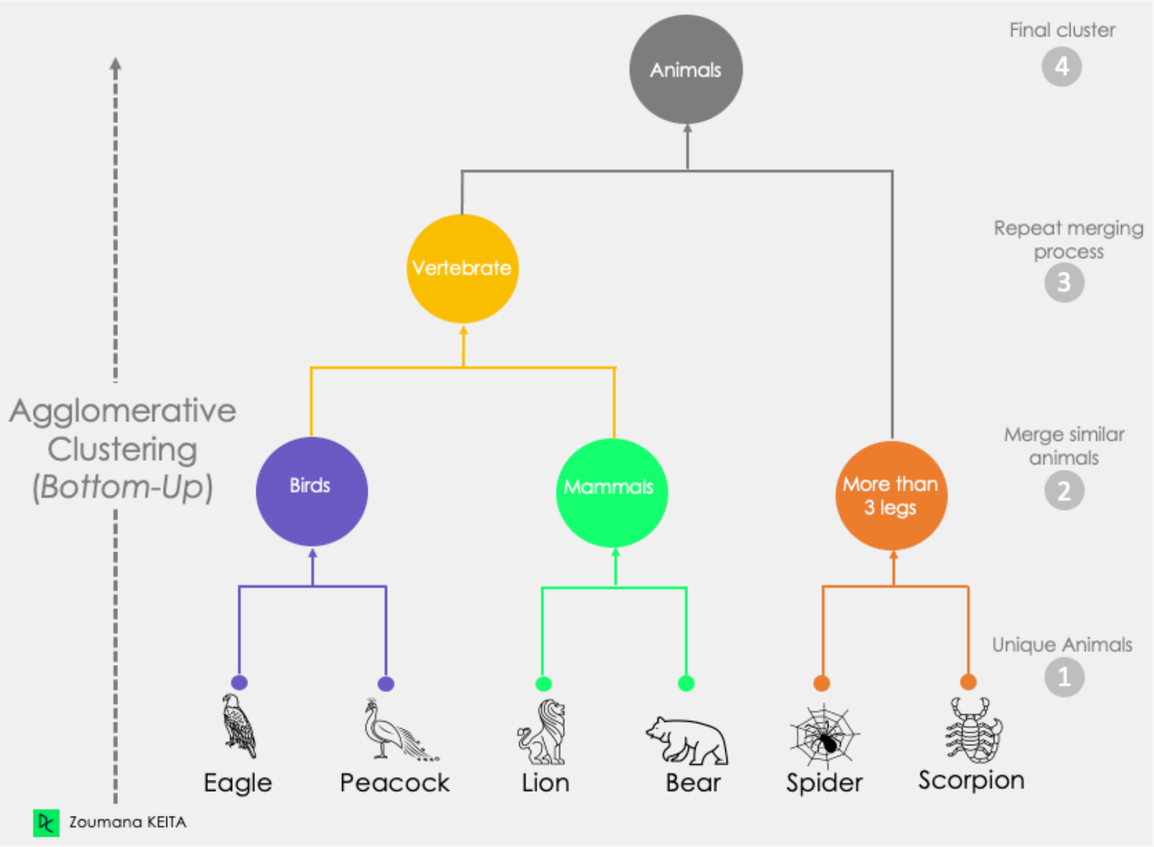


Fig: Hierarchical Clustering [11]

DBSCAN

- Density Based Spatial Clustering of Applications with Noise

- The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

- Eps - neighborhood size
- MinPts - density threshold
- Core - more than MinPts points within eps
- Border - less than MinPts but has core neighbor
- Noise - otherwise

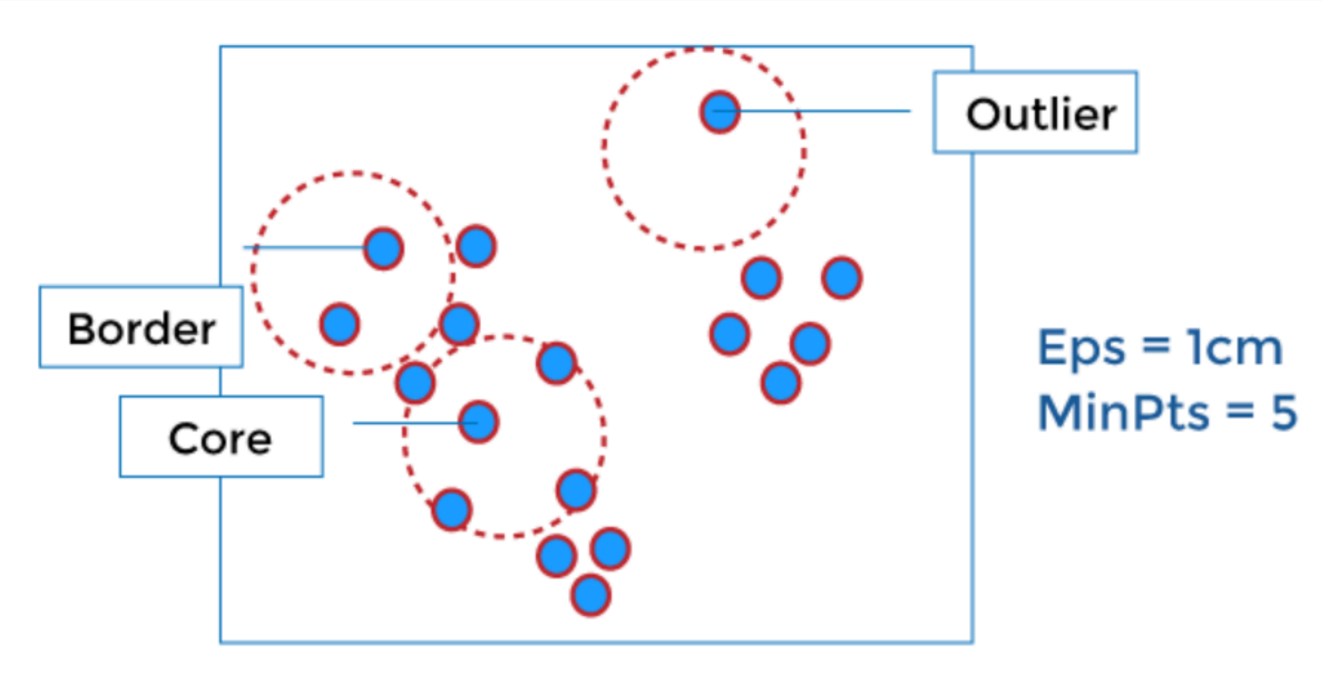


Fig: DBSCAN[12]

DBSCAN

- for core point, Recursively find all points directly density-reachable from it (there exists a sequence of core points from start point to that point).
- Assign the same cluster label to all density-reachable points.
- For each border point, assign it to the same cluster as that core point.
- Noise points will remain as they are.

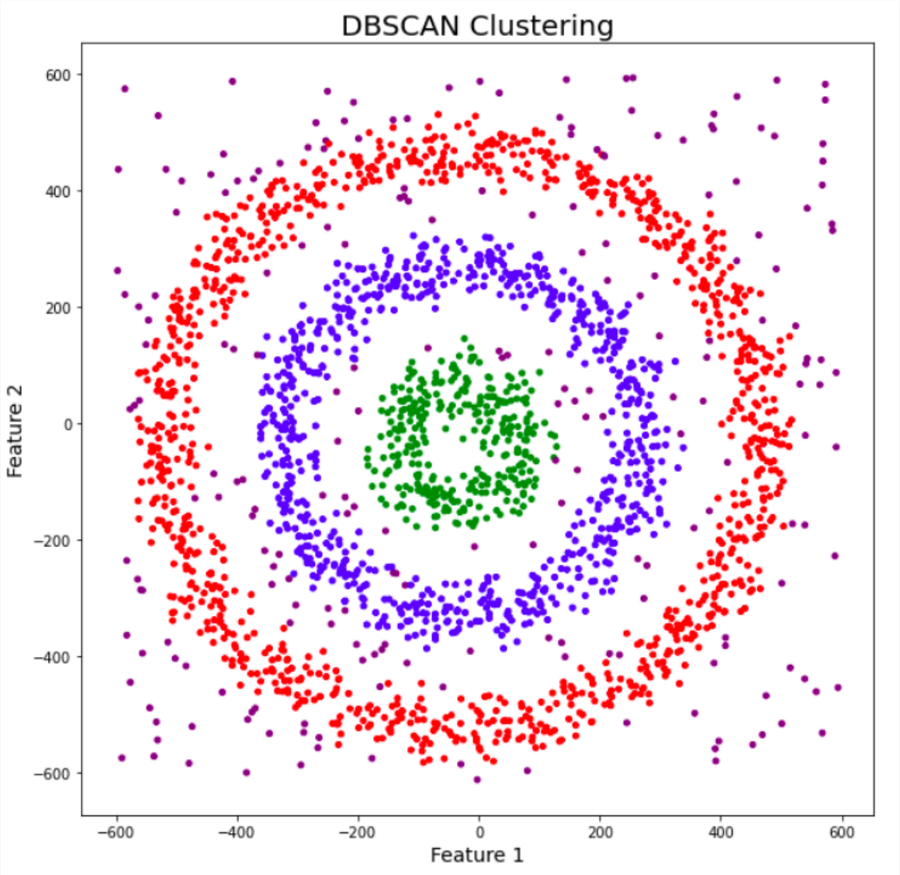


Fig: DBSCAN[12]

Gaussian Mixture Models

- Probability-Based
- It uses assumption that the clusters come from different Gaussian Distributions. So, GMM tries to model the dataset as a mixture of several Gaussian Distributions.
- For Multivariate the probability density function is given by

.

$$G(X|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(X-\mu)^T\mathbf{\Sigma}^{-1}(X-\mu)\right)$$

Gaussian Mixture Models

- Initialize the means, covariances, and mixing coefficients randomly.
- Calculate the posterior probability of each data point belonging to each component using the current parameter estimates (means, covariances, and mixing coefficients).
- Now we had to maximize the likelihood estimates.

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k N(x_i; \mu_k, \sigma_k^2)$$

- Then we will update the parameters and process is iteratively repeated until our model converges.

Architecture

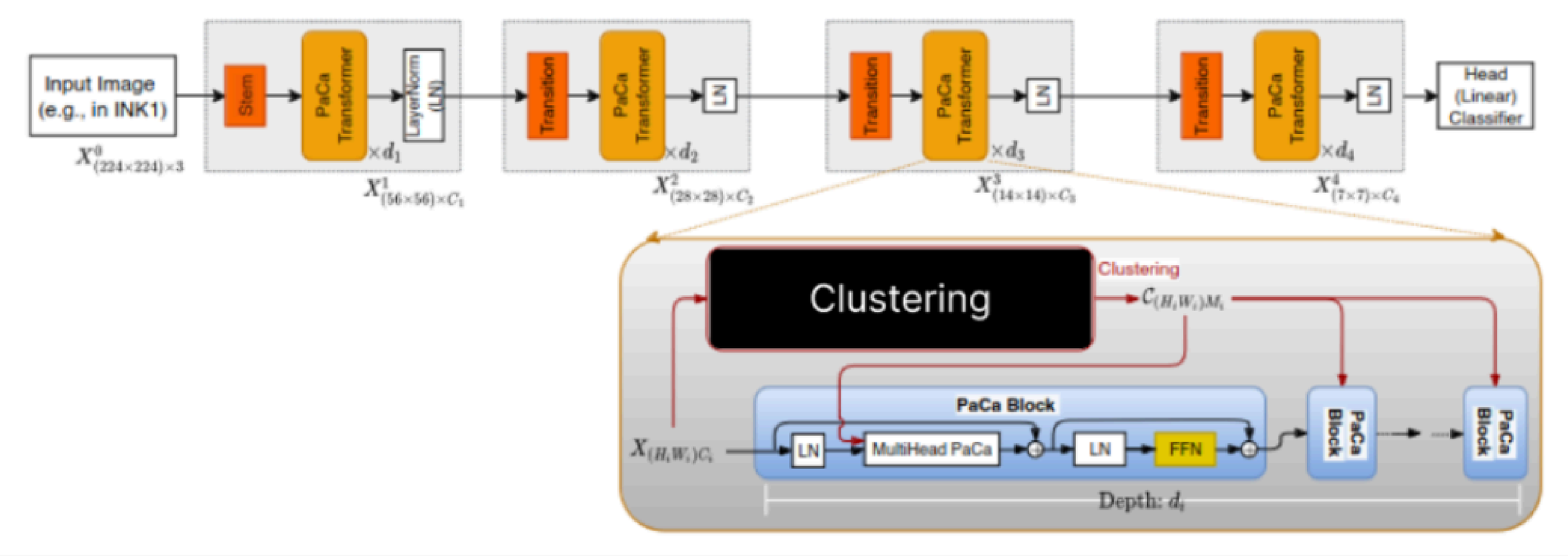


Fig: PaCa ViT Architecture[3]

Methodology

- We have N = H x W tokens, where H and W are the height and width of the patch grid formed via patch embedding.

- Lets denote input sequence by X_{nc}, embedded into C dimensional space.

- Now, first we have to compute cluster assignment C_{nm}, whose goal is to cluster the input sequence into M latent visual tokens (M is predefined number of cluster).

- The M visual tokens Z_{mc} are formed via simple matrix multiplication, can be understood as a depth-wise global weighted pooling of the input $X_{n,c}$ with learned weights, $C_{n,m}$.

$$Z_{M,C} = \text{LayerNorm}(\mathcal{C}^T_{N,M} \cdot X_{N,C})$$

Methodology

- Then, we can calculate Query Q_{nc} by X_{nc} and key K_{mc}, Value V_{mc} by Z_{mc}.

- The core of the Transformer model is to compute the scaled dot-product attention in transforming the input X_{nc} to the output Y_{nc}.

$$A_{N,M} = \text{Softmax}(\frac{Q_{N,C} \cdot K^T_{M,C}}{\sqrt{C}})_{dim=1}$$

$$Y_{N,C} = A_{N,M} \cdot V_{M,C},$$

- Then multi-head self-attention (MHSA) is used to capture the attention in different subspaces and fused by a linear projection.

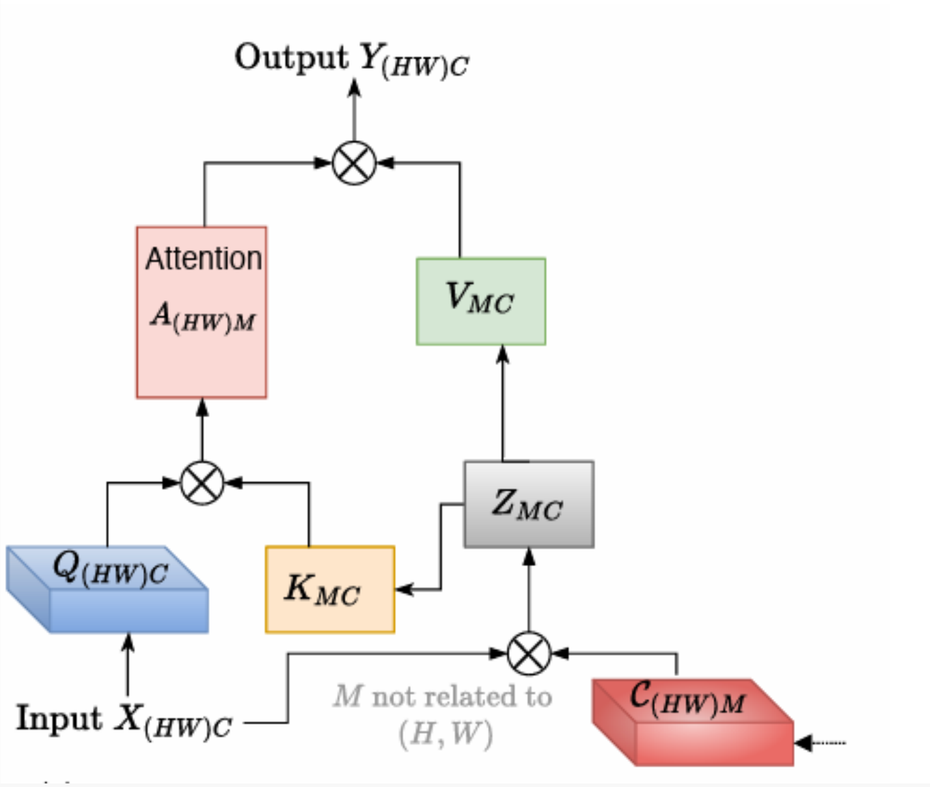


Fig: Attention in PaCa-ViT[1]

Observations

- To address the Quadratic Complexity, quadratic complexity, the key is to ensure M << N, preferably a predefined constant (e.g., M = 100) to induce the linear complexity. However, it adds the overhead of the lightweight clustering module.

- To address interpretability, In inference, we can directly visualize the clusters CN,M as heatmaps to reveal what has been captured by the trained models.

- A binary mask is generated based on clustering scores, upsampled, and applied to the input image to identify areas important for classification.

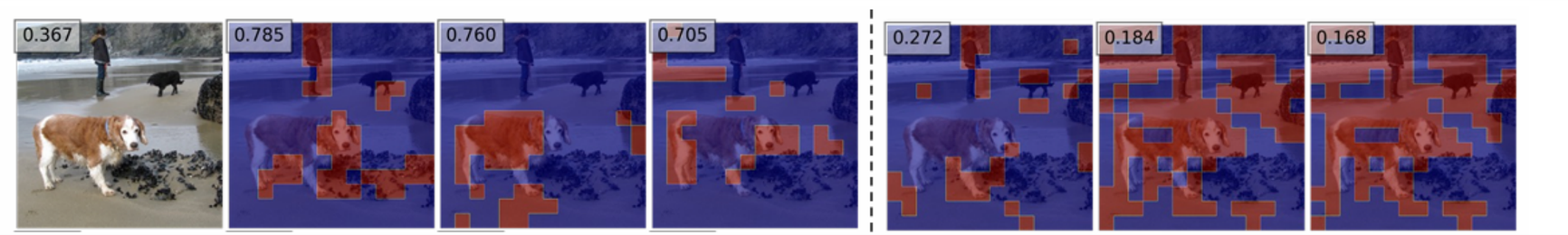


Fig: Examples of visualizing the clusters[3]

Results

Following are the results of the original methodologies (on the ImageNet dataset) presented in the paper

Method	#Params (M)↓	FLOPs (G) ↓	Top-1 Acc. (%)↑
DeiT-T/16 [43]	5.7M	1.3	72.2
PVT-T [48]	13.2	1.9	75.1
PVTv2-B1 [47]	14.0	2.1	78.7
PaCa-Tiny (ours)	12.2	3.2	**80.9** ↑2.2
DeiT-S/16 [43]	22.1	4.6	79.9
T2T-ViT$_t$-14 [58]	22.0	6.1	80.7
PVT-S [48]	24.5	3.8	79.8
TNT-S [20]	23.8	5.2	81.3
SWin-T [32]	29.0	4.5	81.3
CvT-13 [51]	20.0	4.5	81.6
Twins-SVT-S [9]	24.0	2.8	81.3
FocalAtt-Tiny [56]	28.9	4.9	82.2
PVTv2-B2 [47]	25.4	3.9	82.0
PVTv2-B2-li [47]	22.6	4.0	82.1
PaCa-Small (ours)	22.0	5.5	83.08 ↑0.98
PaCa^{mlp}-Small (ours)	22.6	5.9	83.13 ↑1.03
PaCa^{ec}-Small (ours)	21.1	5.4	**83.17** ↑1.07
T2T-ViT$_t$-19 [58]	39.0	9.8	81.4
T2T-ViT$_t$-24 [58]	64.0	15.0	82.2
PVT-M [48]	44.2	6.7	81.2
PVT-L [48]	61.4	9.8	81.7
CvT-21 [51]	32.0	7.1	82.5
TNT-B [20]	66.0	14.1	82.8
SWin-S [32]	50.0	8.7	83.0
SWin-B [32]	88.0	15.4	83.3
Twins-SVT-B [9]	56.0	8.3	83.2
Twins-SVT-L [9]	99.2	14.8	83.7
FocalAtt-Small [56]	51.1	9.4	83.5
FocalAtt-Base [56]	89.8	16.4	83.8
PVTv2-B3 [47]	45.2	6.9	83.2
PVTv2-B4 [47]	62.6	10.1	83.6
PVTv2-B5 [47]	82.0	11.8	83.8
PaCa-Base (ours)	46.9	9.5	83.96 ↑0.76
PaCa^{ec}-Base (ours)	46.7	9.7	**84.22** ↑1.02

Results

Following are the results of the model we trained, using different clustering techniques on the CIFAR 10 dataset.

Clustering Method	Accuracy	F1
MLP	0.81	0.81
KMeans	0.84	0.84
GMM	0.73	0.74
Hierarchial	0.76	0.75

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\textbf{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

$$\textbf{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Conclusion

In our presentation, we introduce the Patch-to-Cluster Attention (PaCa) module to enhance the efficiency and interpretability of Vision Transformers (ViTs). and it also addresses the quadratic complexity issue. Through various clustering methods, we've gained insights into image classification results using the CIFAR 10 dataset.

Future Scope

Apply to datasets of different domains

Further finetuning and transfer learning to improve performance

Improve the efficiency and scalability of model

Deployment in real world applications

References

[1] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
[2] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).
[3] Grainger, Ryan, et al. "PaCa-ViT: learning patch-to-cluster attention in vision transformers." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
[4] https://medium.com/analytics-vidhya/image-classification-vs-object-detection-vs-image-segmentation-f36db85fe81
[5] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In NeurIPS, 2020
[6] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In International Conference on Learning Representations.

References

[7] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 2964–2972, 2022.

[8] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and H Linformer Ma. Self-attention with linear complexity. arXiv preprint arXiv:2006.04768, 2020.

[9] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 782–791, 2021.

[10] https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

[11] https://www.datacamp.com/tutorial/introduction-hierarchical-clustering-python

[12] https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/

[13] https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

Thank you for listening!