PRUNE AND TELL: VIT VS CONVNEXT IN IMAGE CAPTIONING WITH TOKEN PRUNING

Exploring Efficient Vision-Language Models through Backbone Comparison

Ghufron Wahyu Kurniawan / 413830003

What is image captioning?

"Man in black shirt is playing guitar."

"Construction worker in orange safety vest is working on road."

"Two young girls are playing with lego toy."

Source: COCO captions dataset

- Image captioning is a multimodal problem where the goal is to learn a mapping from visual data (images) to natural language (sentences).

Motivation

Why Efficient Image Captioning or VLM?

- Vision-language tasks (e.g., image captioning, VLM) are computationally expensive.
- Cross attention in image captioning are powerful, but can be redundant in token usage.
- Reducing computation without sacrificing accuracy is crucial for real-time or edge devices.

Theoretical Background

- **Vision Transformer (ViT)**
  Paper: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (Dosovitskiy et al., 2020)

- **Swin Transformer**
  Paper: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows (Liu et al., 2021)

- **ConvNeXt**
  Paper: A ConvNet for the 2020s (Liu et al., 2022)

- **MetaFormer**
  Paper: MetaFormer is Actually What You Need for Vision (Yu et al., 2022)

- **Token Pruning**
  - **Paper:** DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification

Transformer with the multihead self attention is powerfull but redundant in complexity O(n^2)

Image captioning block

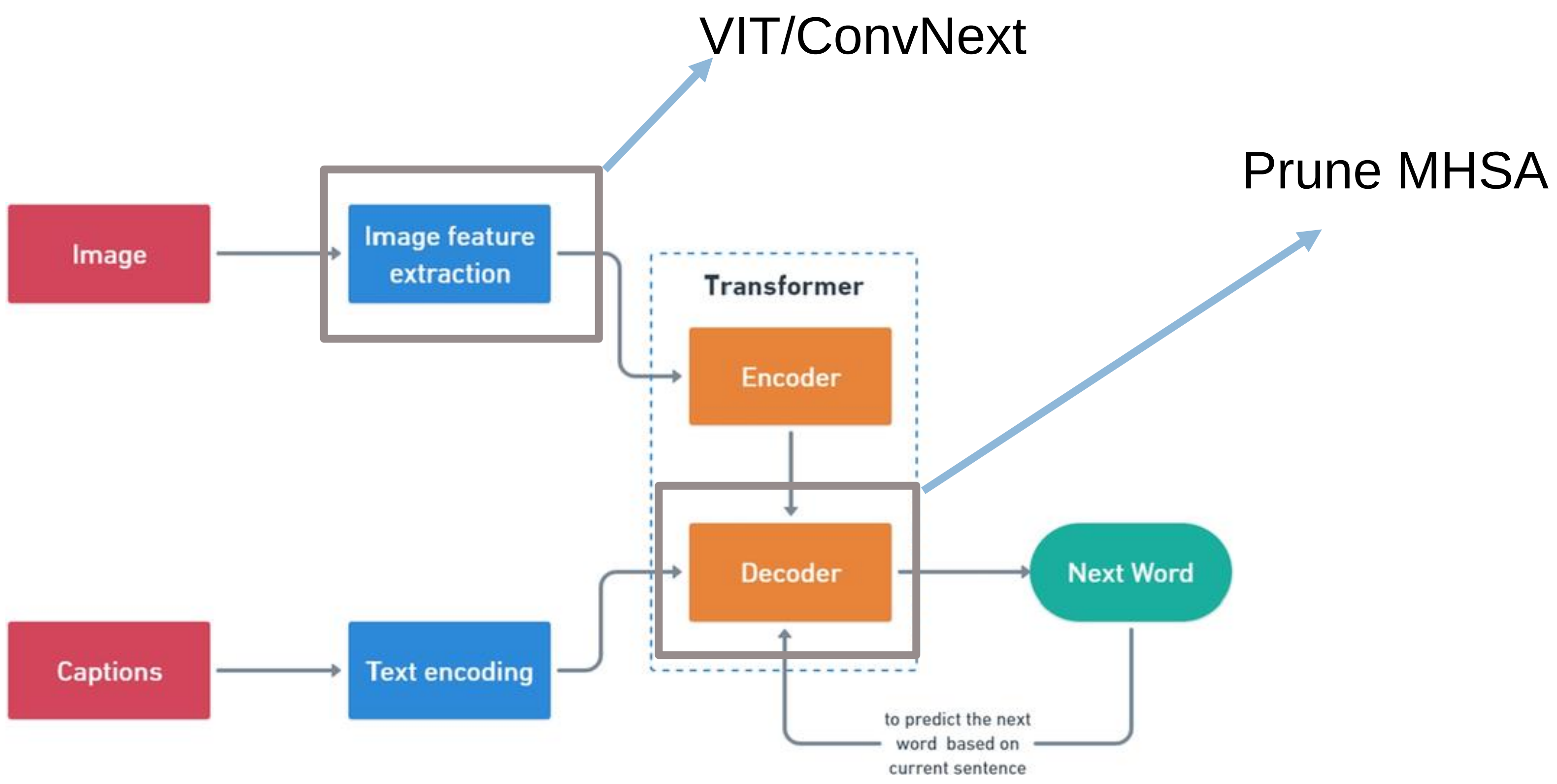
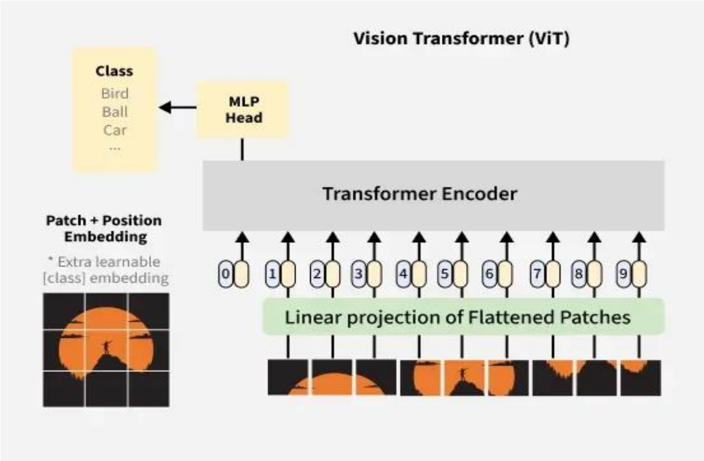
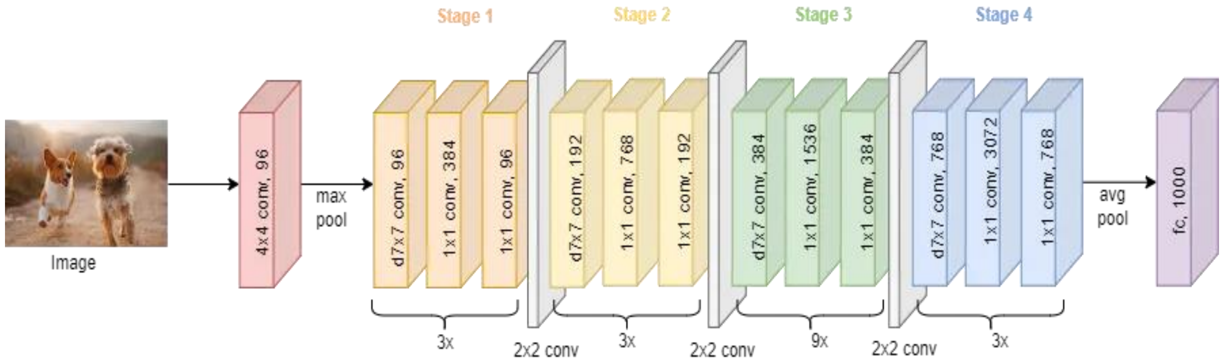
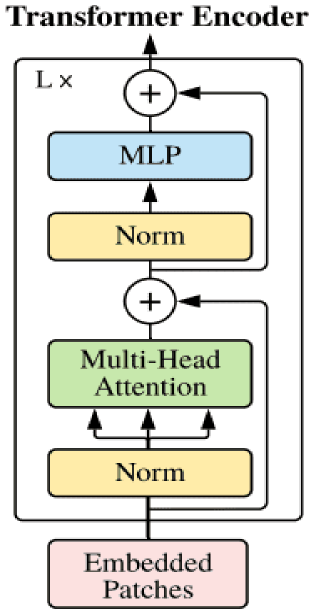
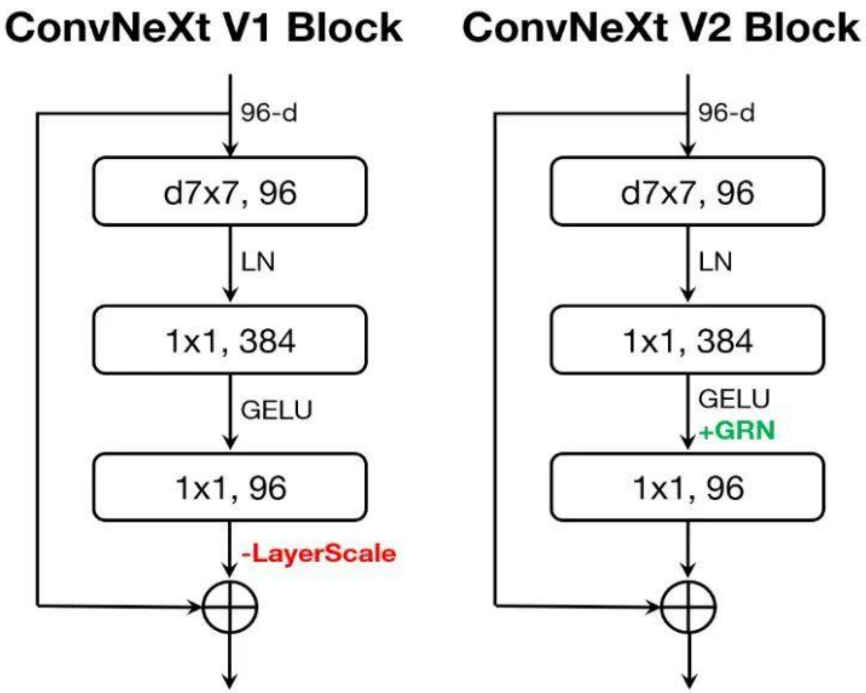


Image feature extraction (VIT and Convnext)

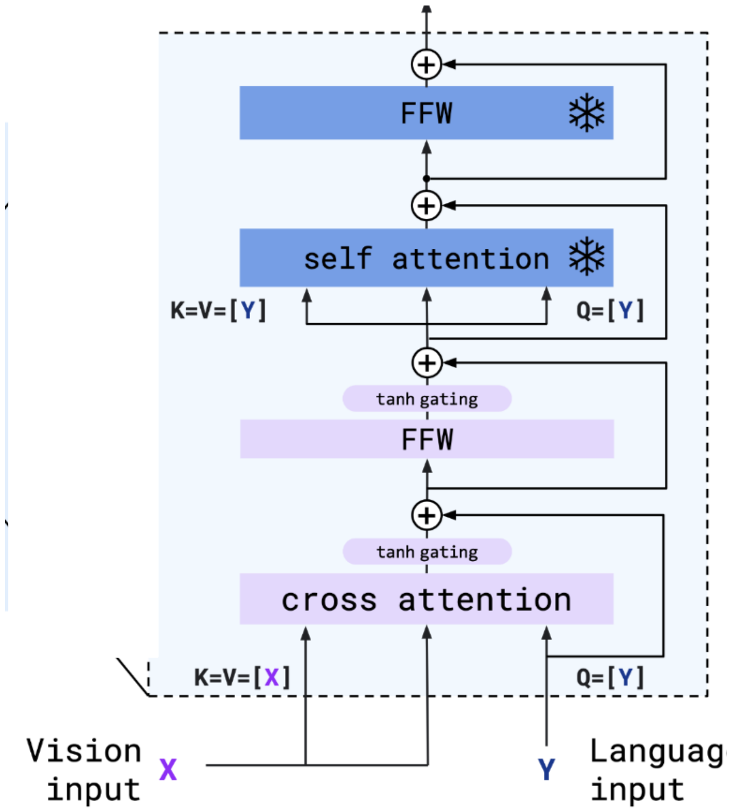






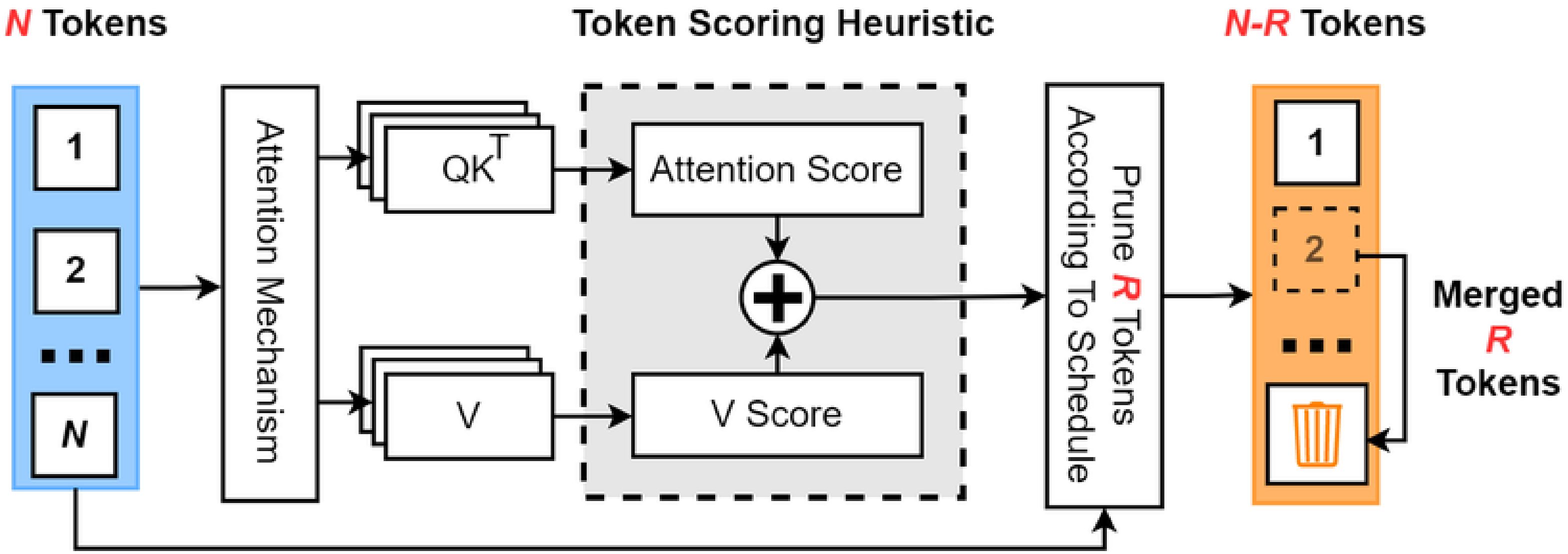


Decoder



MHSA with Cross-Attention is Key part of image captioning because find the correlation of image-text

Token Prune illustration



BEFORE PRUNE

AFTER PRUNE

HOW we BUILD image captioning

- Vision Transformer (ViT) or ConNext as Encoder
- Projection Layer (to ensure image and text have same dimention)
- Tokenization and Positional Embedding
  - Tokenizer (Converts ground truth captions into token IDs using a vocabulary)
  - PosEmbedding (Embeds token IDs into vectors ([batch, seq_len, d_model]), Adds positional encodings to preserve word order.)
- Masking Mechanisms
  - Padding Mask: Prevents attending to <PAD> tokens.
  - Subsequent Mask: Prevents a word from attending to future words during training (causal masking).
- Transformer Decoder
  - Cross-Attention with token pruning: Attend to visual features from visual encoder.
  - Feed-Forward Network (FFN): Non-linear transformation.
- Output Layer (Caption Generation)

Training Strategy and hyperparameter

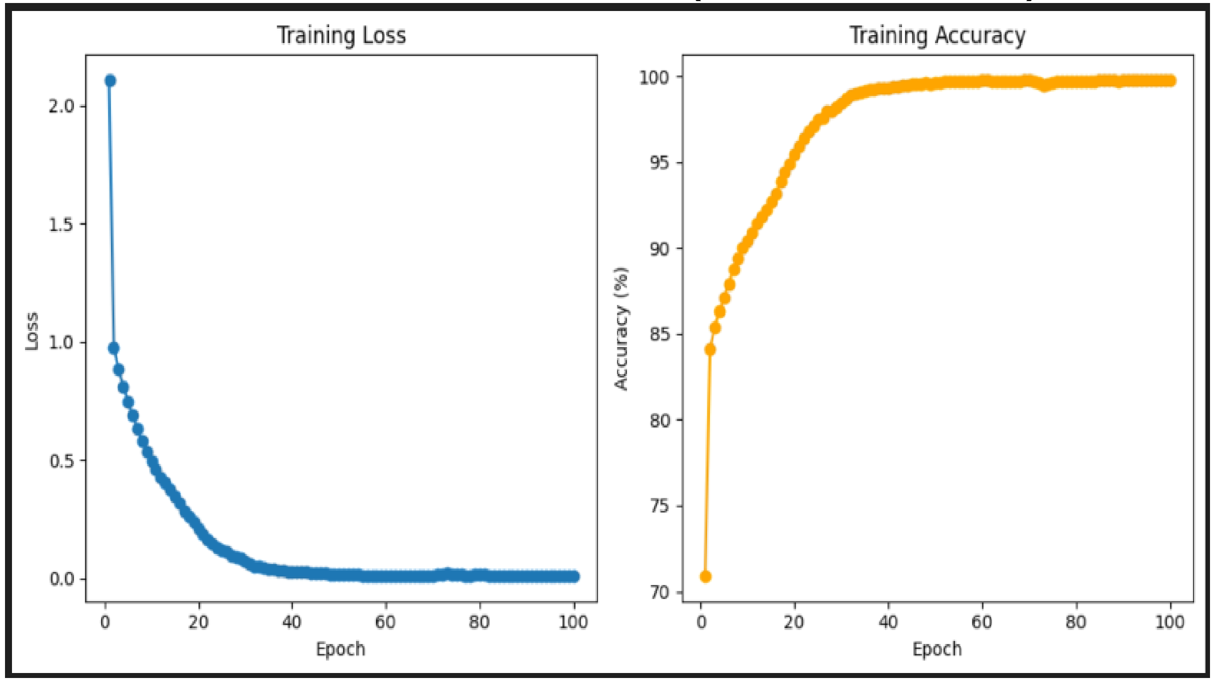
- batch_size = 128
- num_epochs = 100
- patch_dim = 768
- vocab_size = 2033
- d_model = 512
- n_layers = 2
- nhead = 4
- ff_dim = 2048
- dropout_ratio = 0.1
- learning_rate = 0.001

- Dataset = coco image captioning (14GB)
- Prune ratio = 30%

(train in single RTX4090, before use prune the batch size maximum=48)

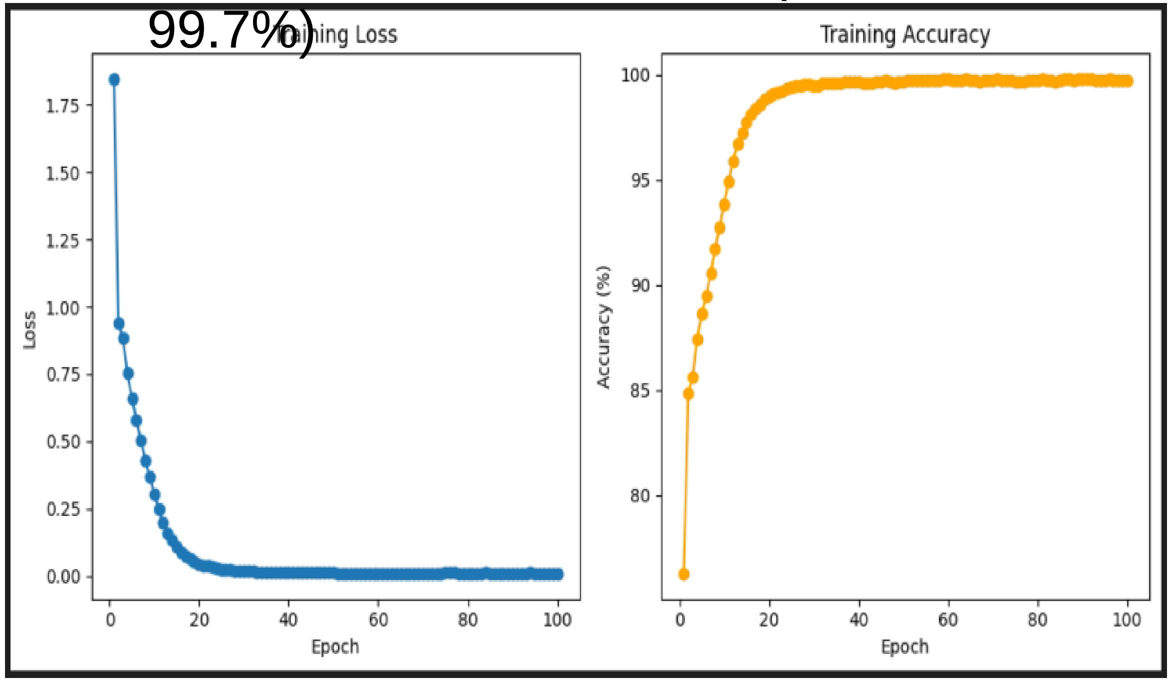
Training Result

VIT BACKBONE (ACC = 99.8%)




```
Epoch 97/100 | 100%|          | 8/8 [00:15<00:00,  1.96s/it, acc=99.78%, loss=0.0001]
Epoch 98/100 | 100%|          | 8/8 [00:15<00:00,  1.91s/it, acc=99.80%, loss=0.0001]
Epoch 99/100 | 100%|          | 8/8 [00:15<00:00,  1.95s/it, acc=99.78%, loss=0.0001]
Epoch 100/100 | 100%|          | 8/8 [00:15<00:00,  1.91s/it, acc=99.80%, loss=0.0001]
```

CONVNEXT BACKBONE (ACC = 99.7%)




```
Epoch 97/100 | 100%|          | 16/16 [00:16<00:00,  1.01s/it, acc=99.75%, loss=0.0002]
Epoch 98/100 | 100%|          | 16/16 [00:15<00:00,  1.02it/s, acc=99.72%, loss=0.0002]
Epoch 99/100 | 100%|          | 16/16 [00:16<00:00,  1.01s/it, acc=99.70%, loss=0.0002]
Epoch 100/100 | 100%|          | 16/16 [00:16<00:00,  1.02s/it, acc=99.70%, loss=0.0002]
```

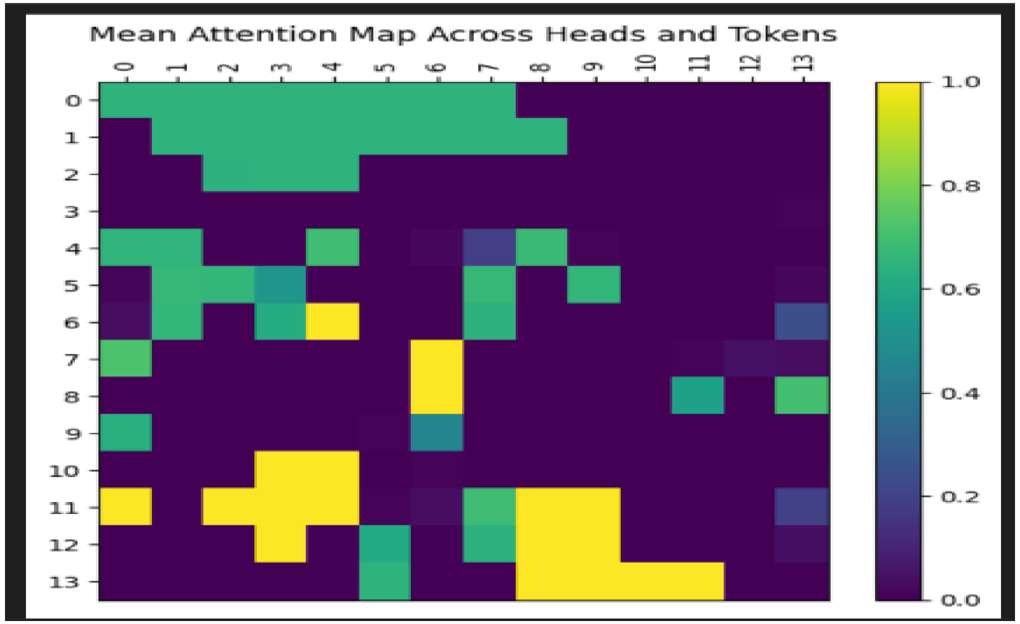
Real world testing

```
A group of people walking along a snow covered slope,
tensor([  1,  12,  89,   8,  10,  93, 301,  12, 350, 624, 983,  92,   2,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0])

<matplotlib.image.AxesImage at 0x748543dfc190>
```

Real world testing

MEAN ATTENTION MAP (SCALING)



Conclusion

- Summary
  - We compared Vision Transformer (ViT) and ConvNeXt architectures for the image captioning task, integrating token pruning to reduce computation. Both models were evaluated based on caption quality, efficiency.
- Key Findings
  - ViT with token pruning preserved semantic richness but was more sensitive to pruning rate.ConvNeXt, a modern CNN, showed greater robustness to token pruning with better speed-performance trade-off.Token pruning significantly improved efficiency with minimal loss in caption accuracy for both models.