Evolving Deep Architectures: A New Blend of CNNs and Transformers Without Pre-Training Dependencies

By: Manu Kiiskilä

Motivation (1/2)

- Object segmentation is one of the fundamental computer vision tasks, partitioning an image by class
- The recently developed transformer architecture, implementing attention, played crucial role in the fast emergence of generative AI technologies (Wu et al., 2023).
- Transformer layers require significantly more data to generalize compared to convolutional layers but require less computational resources and less training time (Shi et.al., 2023).
- Concept of transformers extended to create a relatively new field – Vision Transformers (Dosovitskiy et al., 2023).
- Limited research available on hybrid architectures with transformers such as Swin and CNNs (Liu et al., 2023) and uses pre-trained models for CNNs and/or transformers (Zhang et al., 2023).

Motivation (2/2)

- When research for this paper started in 2023, limited studies available on hybrid architecture with no-pretraining done on both CNNs and transformers. Closest research was C-C-T-T architecture using transformers with no pre-training and pre-trained CNN models. (Hong et. Al., 2022).
- All my computer vision projects so far required long training time and additional GPU and computing units to train a model. Using pre-trained models have limitations such as not suitable for all domains and less flexibility in adjusting network structures.
- Goal is to investigate if a hybrid architecture consisting of CNNs and transformers with no pre-training be built and train with smaller dataset.

Related Work

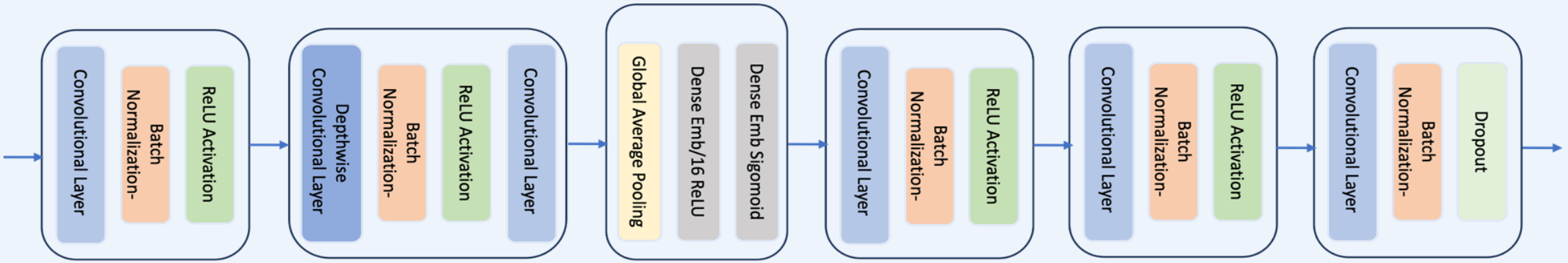
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems
- Weixiang H., Wang R., Jiangwei L., Lele X., Liheng Z., Jian W., Jingdong C., Honghai L., Wei C. (2022). Training Object Detectors from Scratch: An Empirical Study in the Era of Vision Transformer
- Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., ... & Zhang, L. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6881- 6890).
- Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems, 34, 12077-12090
- Zhang, Y., Liu, H., & Hu, Q. (2021). Transfuse: Fusing transformers and cnns for medical image segmentation. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24 (pp. 14-24). Springer International Publishing.

Objectives

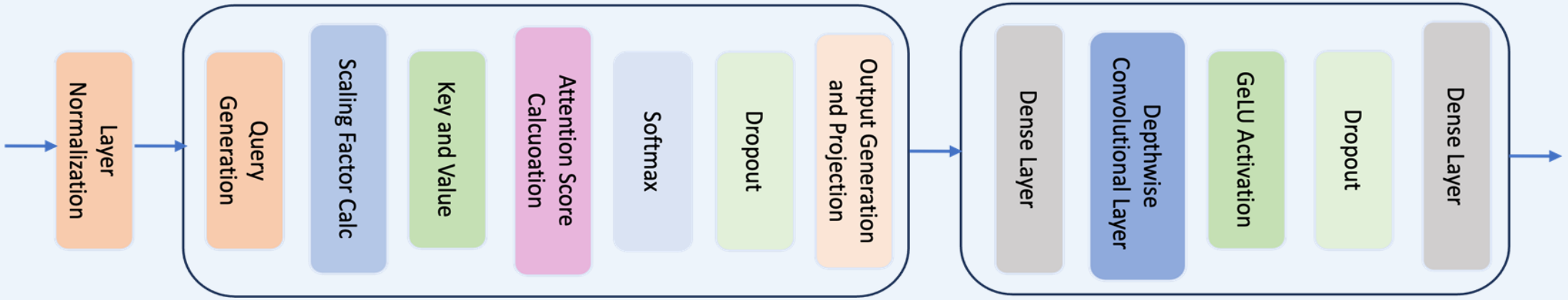
(1) Design and develop the modified C-C-T-T architecture, trained and tested on the MS COCO dataset

(2) Evaluate results from the T-T-T-T architecture on the same dataset, comparing results with the proposed model

Convolutional and Transformer Blocks for C-C-T-T architecture

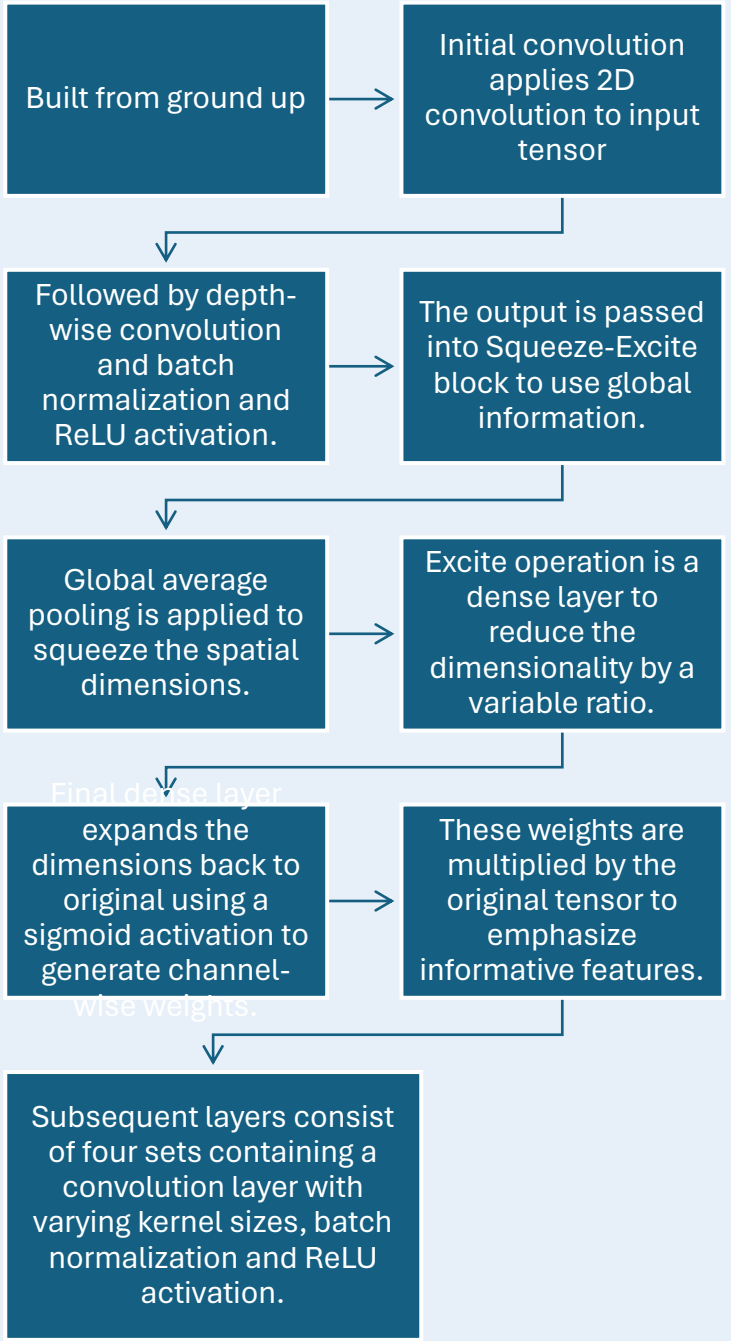


1. **Convolutional Module**



2. **Transformer Block**

Convolution Blocks



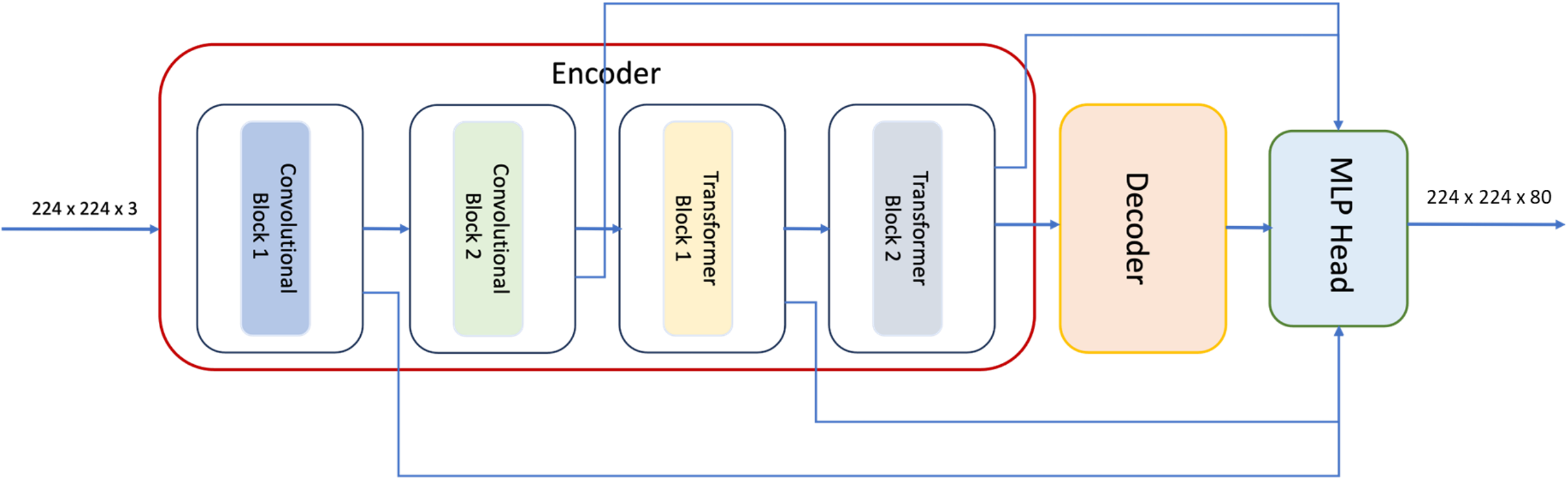
Transformer Blocks

- Based on SegFormer framework
- SegFormer chosen because it doesn't require positional embedding
- Overlap Patch merging is used to generate patches
- Layer normalization is applied to standardize the output across the embedding dimensions.
- Efficient self-attention method is used for attention.
- Feature maps are passed through the Mix-MLP to better capture broad contextual details and finer features.

MLP Decoder Head

- The architecture leverages a series of Multi-Layer Perceptrons (MLPs) to process and combine feature maps through a dense layer.
- The multi-scale feature maps are projected into a shared embedding dimension.
- The output tensor resolution is downscaled to the lowest shared resolution, unifying the spatial dimensions of the processed feature maps.
- The combined feature maps are passed through a convolutional layer to mix features across the channel dimensions.
- Batch normalization and ReLU activation are applied before the tensor is reshaped to the original image input resolution.

Proposed C-C-T-T Architecture



C-C-T-T Architecture

Two CNN modules

Two SegFormer-B5 blocks

No pre-training done on CNN or SegFormer

Hyperparameters such as learning rate changed

Experimental set up

- Google Colab
- Single NVIDIA-A100 GPU with 48GB VRAM
- TensorFlow 2.5
- MS COCO 2017 “Thing” maps dataset

Hyperparameters used in C-C-T-T architecture

Test	64	128	320	512
Strides	4	2	2	2
Expansions	4	4	4	4
Depths	NaN	NaN	20	8
Kernel size	7	3	3	3
No. of Heads	1	2	5	8
Scale factors	NaN	NaN	2	1

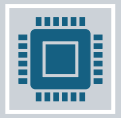
- “Scale factors” are applicable only during attention mechanisms.
- “Depths” third parameter is halved due to environmental constraints in Colab environment.
- For T-T-T-T architecture, hyperparameters from SegFormer specification are used.

Train and Test Metrics

Model	C-C-T-T		T-T-T-T	
Metrics	Pixel accuracy (PA)	Mean – IoU (mIoU)	Pixel Accuracy (PA)	Mean-IoU (mIoU)
Train	0.6213	0.4945	0.5417	0.4944
Test	0.6956	0.4944	0.4783	0.4944

- Pixel accuracy significantly better using C-C-T-T architecture compared to All-transformer architecture when trained with smaller data sets such as MS-COCO.
- Poorer Mean-IoU might be due to under-representation or missing of certain classes in the dataset

Results and analysis (1/2)



Both architectures were tested on identical test splits of the MS COCO dataset, measured metrics were Pixel Accuracy and Mean-IoU

The C-C-T-T Pixel Accuracy performed significantly better than the T-T-T-T architecture

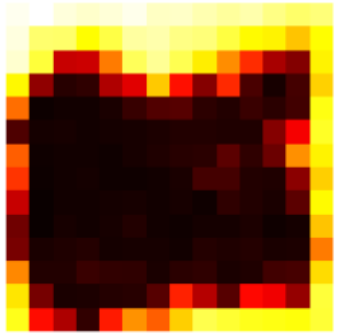
The relatively poor M-IoU results in both architectures is likely due to the inherent underrepresentation of certain classes present in the MS COCO dataset

45% improvement in Pixel Accuracy

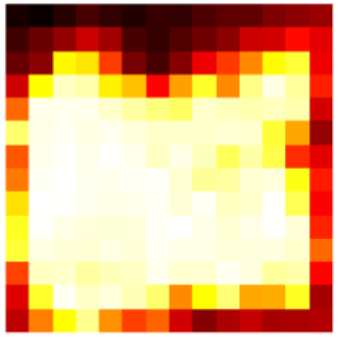
Results and analysis (2/2)

Original Image

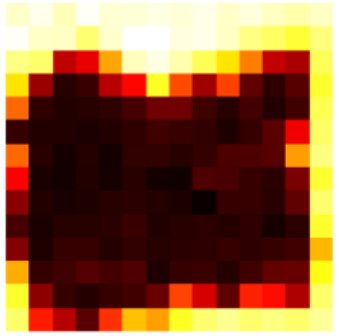
Head 1



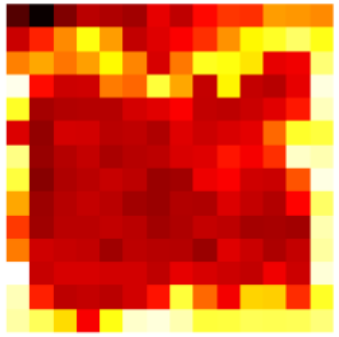
Head 2



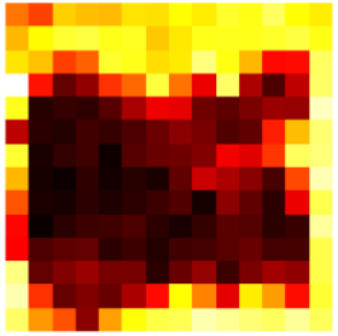
Head 3



Head 4



Head 5



Original Image

Head 1

Head 2

Head 3

Head 4

Head 5

Attention maps display the attention score before the SoftMax operation, allowing visualization of the focus of the model at different layers.

The top image displays attention maps from the first transformer block in the C-C-T-T architecture, while the second displays attention maps from the third transformer in the T-T-T-T architecture

Clear distinction in the transformers ability to capture details between the architectures, however both are capable of contextualizing global features

Discussion and limitations

- Studies proposing new hybrid architectures such as HTC-net (Tang et al., 2024) and real-life application of hybrid models such as Focal liver lesion classification (Zhao et al., 2024), musculoskeletal images (Bi et al., 2024) continue to extend the research in using CNN and transformer together.
- However, most of the latest research also focus on using pre-trained networks and transformers which can affect the generalization of a particular domain.
- Using smaller but customized dataset that can be curated to not have gaps or underrepresentation of certain classes will yield better results in training a domain specific model.
- Experimentation with hyperparameters and other hybrid architectures such as C-C-C-T and other combinations is still needed to find the perfect balance between CNNs and transformers to yield high performance and still reduce the training time and computational power required to train the model.
- Experimentation with other emerging transformers is also needed.

Contributions

This study contributes to the new field of Vision Transformers specifically for fields where large datasets are not available and require domain specific training.

This study contributes to further research into hybrid architectures leveraging inductive biases present in CNNs and powerful capabilities of transformers.

AI involvement

- No AI help is taken for this study and/or writing the paper.
- For SegFormer, code from “https://github.com/NVlabs/SegFormer” is used as base and modified as needed for the architecture including the hyperparameters
- Tensorflow is used as the framework to facilitate coding of the architectures

Acknowledgements

I would like to thank my advisor for Idea sparring, discussions & encouragement during the design and development phase and providing valuable feedback during the writing phase.