Challenges in Architecting Vision Inference Systems for Transformer Models

Cheng Wang

CTO & SVP, Software & Architecture

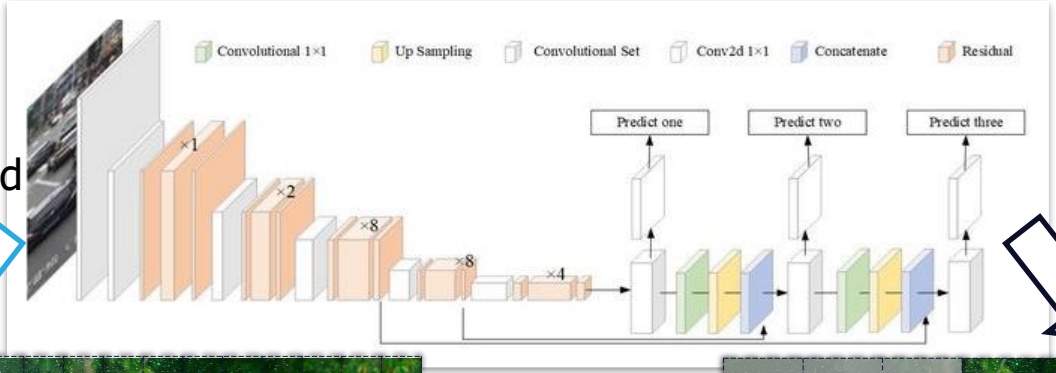
Flex Logix

flexlogix reconfigurable

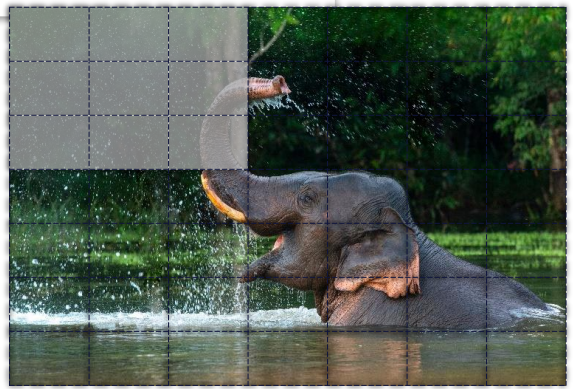
Blind person and the elephant – Receptive field is a big limitation

CNNs are limited by receptive field

Smaller receptive field
Smaller features & objects



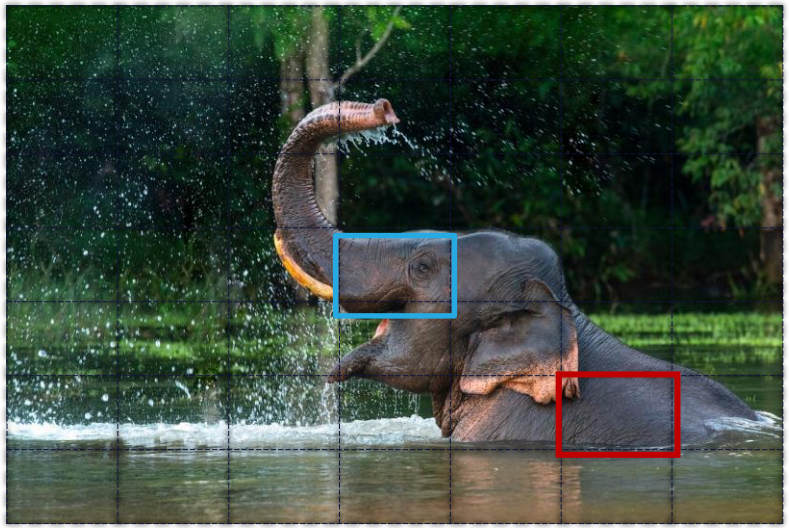
Larger receptive field
Larger features & objects



Transformers use context from whole image

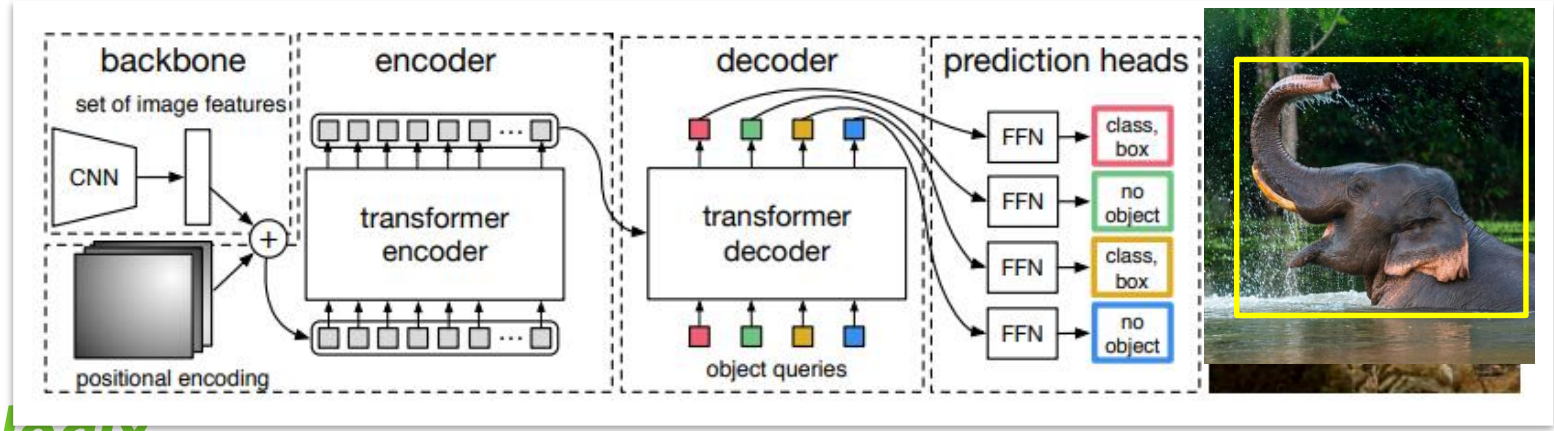
CNN: Elephant?

Transformers: Elephant!



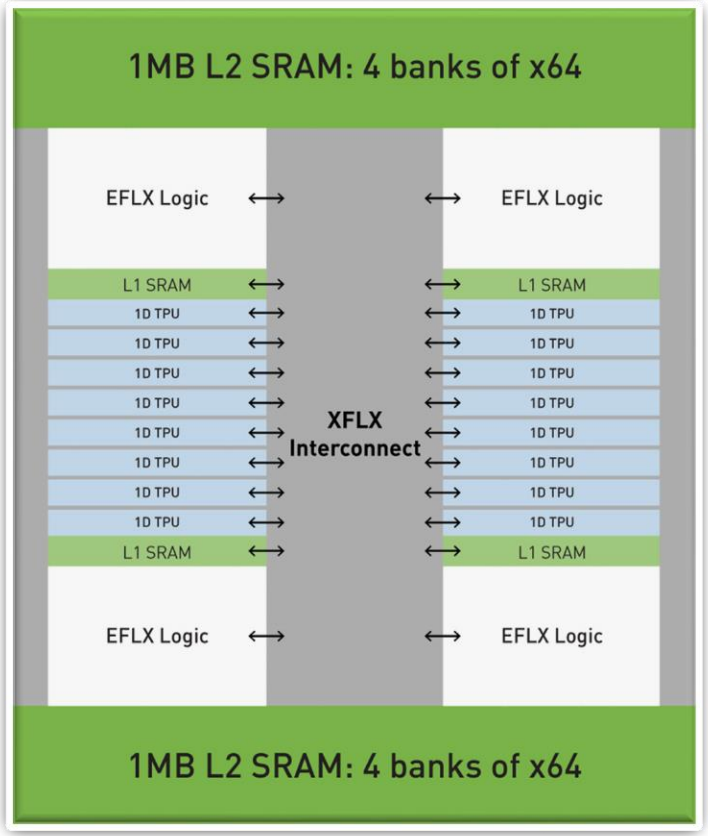
DETR 2020 – The de-facto vision transformer model

- Uses CNN backbone for feature extraction & transformer "head"
- Transformer Encoder extracts features from all patches for context
- Decoder makes predictions based on all extracted features
- Transformer Encoder/Decoder operations are very different from CNN

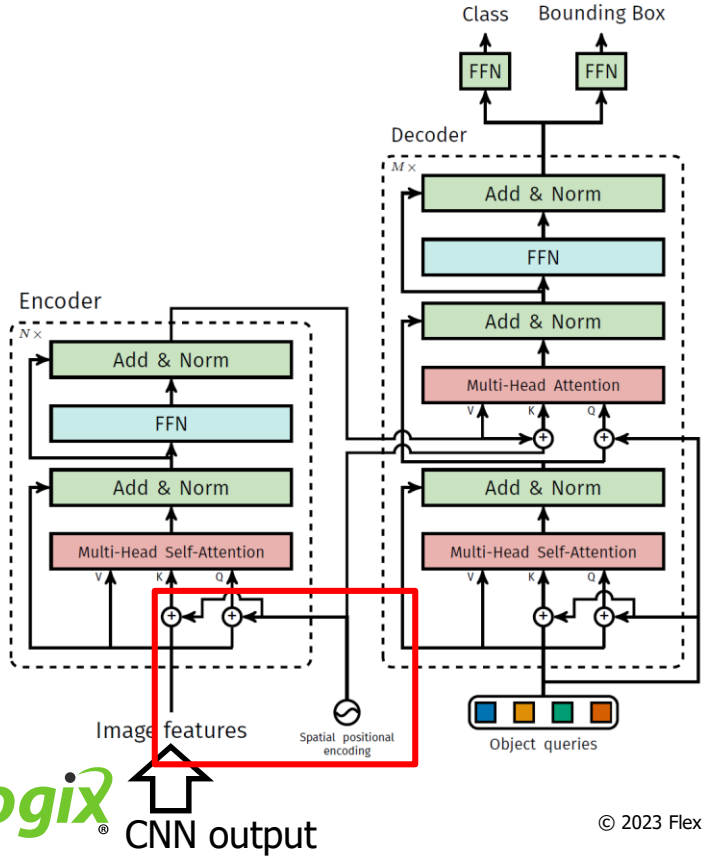


InferX dynamic TPU – Flexible, balanced & memory-efficient

- InferX provides flexibility essential for transformers:
  - Each TPU can stream data with: TPU, L1 weight mem, L2 Data mem & DDR
  - TPU natively supports mixed precision
  - Flexible activations in EFLX eFPGA
- More data bandwidth vs Network-on-Chip based AI
  - Also more flexible data manipulation

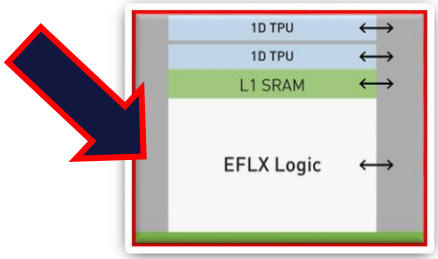


Diving into vision transformers

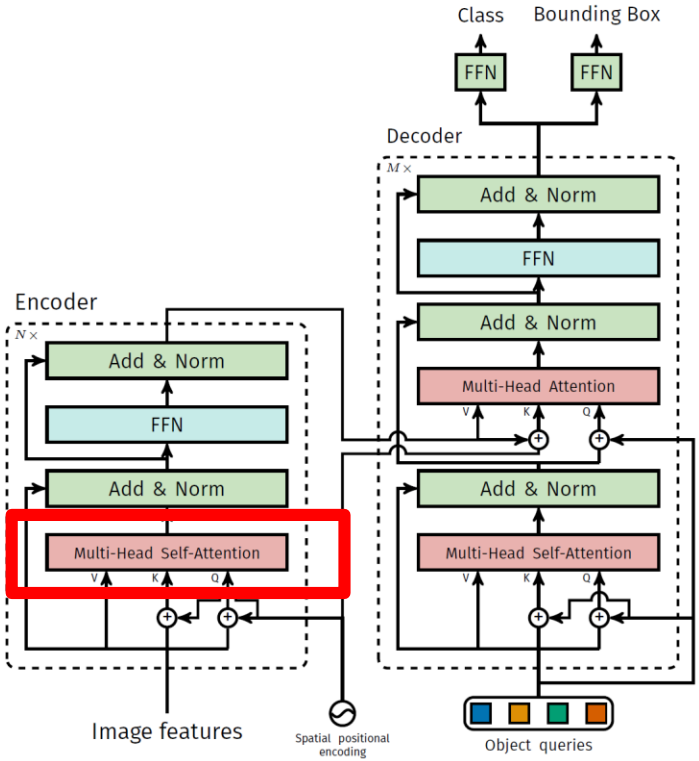


- First stage is positional encode:
  - PE values are stored eFPGA ROMs
  - EFLX lookup PE "on the fly" to add to the K/Q matrix into the attention head

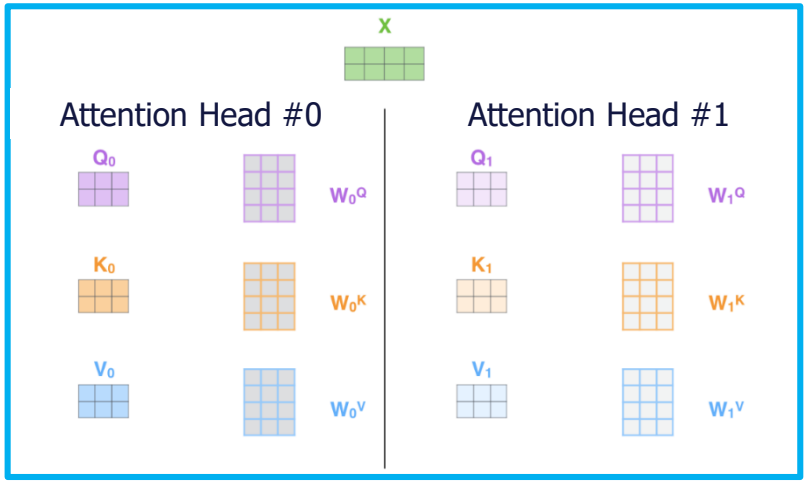
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$



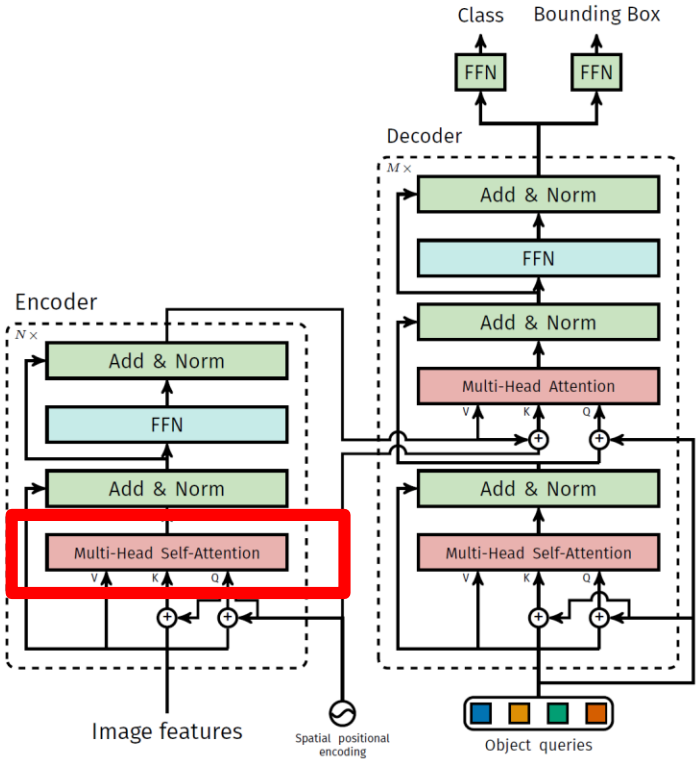
Diving into vision transformers (2)



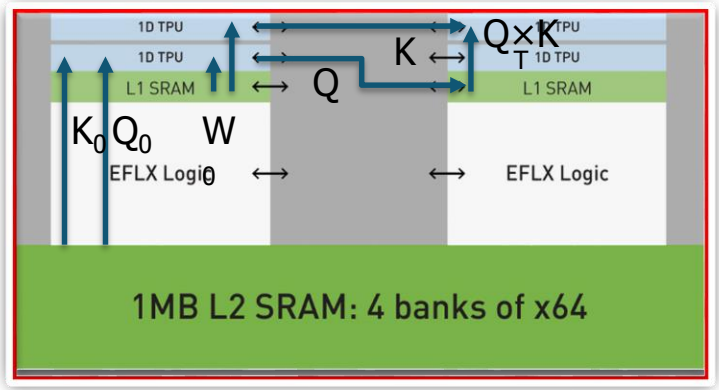
- Second stage multiplies input with 3 matrices for each head (Q/K/V):
  - Each matrix maps to TPU weights



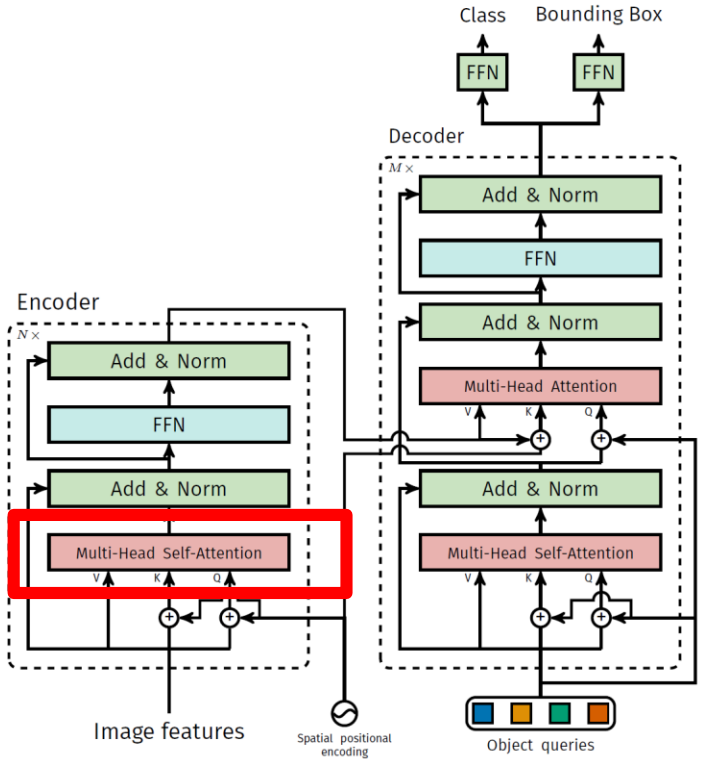
Diving into vision transformers (3)



- Main part of multi-head attention layer is a challenge on traditional edge accelerators:
  - The (Q, K, V) for each matrix is activation data
- $Q \times K^T$ multiplies 2 activation data:
  - InferX can load activation into weight memory

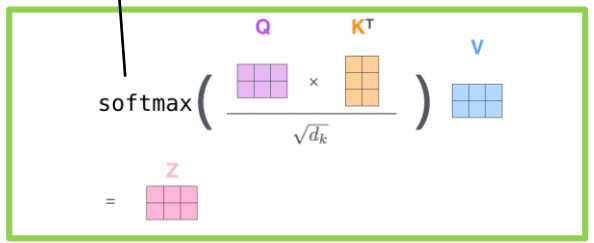


Diving into vision transformers (4)

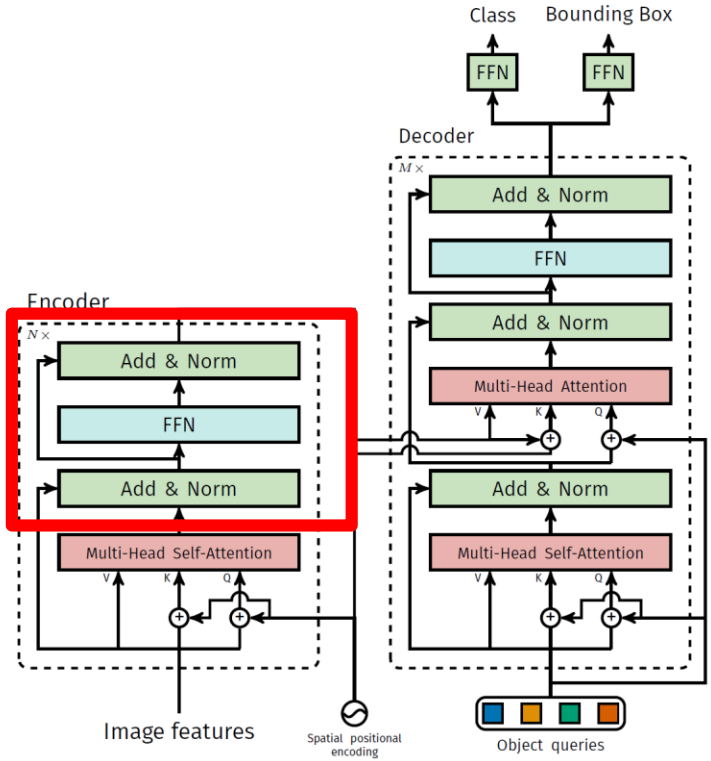


- Softmax and normalization operators are also challenging on int8 datapaths
  - InferX mixed-precision includes BFloat16 format
  - Enables softmax & normalization computation without going to a separate floating-point unit

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$



Diving into vision transformers (5)



- Normalization is executed in BF16 due to its large dynamic range.

L1 norm	$\|W\|_1 = \sum_i^n \|\omega_i\|$
Squared L2 norm	$\|W\|_2^2 = \sum_i^n \omega_i^2$

- Add and Feed-forward Network (FFN) operators are similar those in CNNs

InferX IP is linearly scalable (5nm, batch=1)

DETR 2020 (1024×1024)	12 IPS	23 IPS	70 IPS	127 IPS
YOLOv5s (640×640)	75 IPS	175 IPS	396 IPS	850 IPS
YOLOv5L6 (1280×1280)	5 IPS	10 IPS	24 IPS	48 IPS
ResNet50 (1024x1024)	18 IPS	38 IPS	102 IPS	158 IPS

How to deploy InferX models in your system?

Standard training and transfer learning workflow

Application data

Transfer learning

FP32 model

(Optional)

- Refine model for int8

Flex Logix model developer workflow with InferX MDK

Model conversion

Edge model

Int8 Model

InferX model developer kit

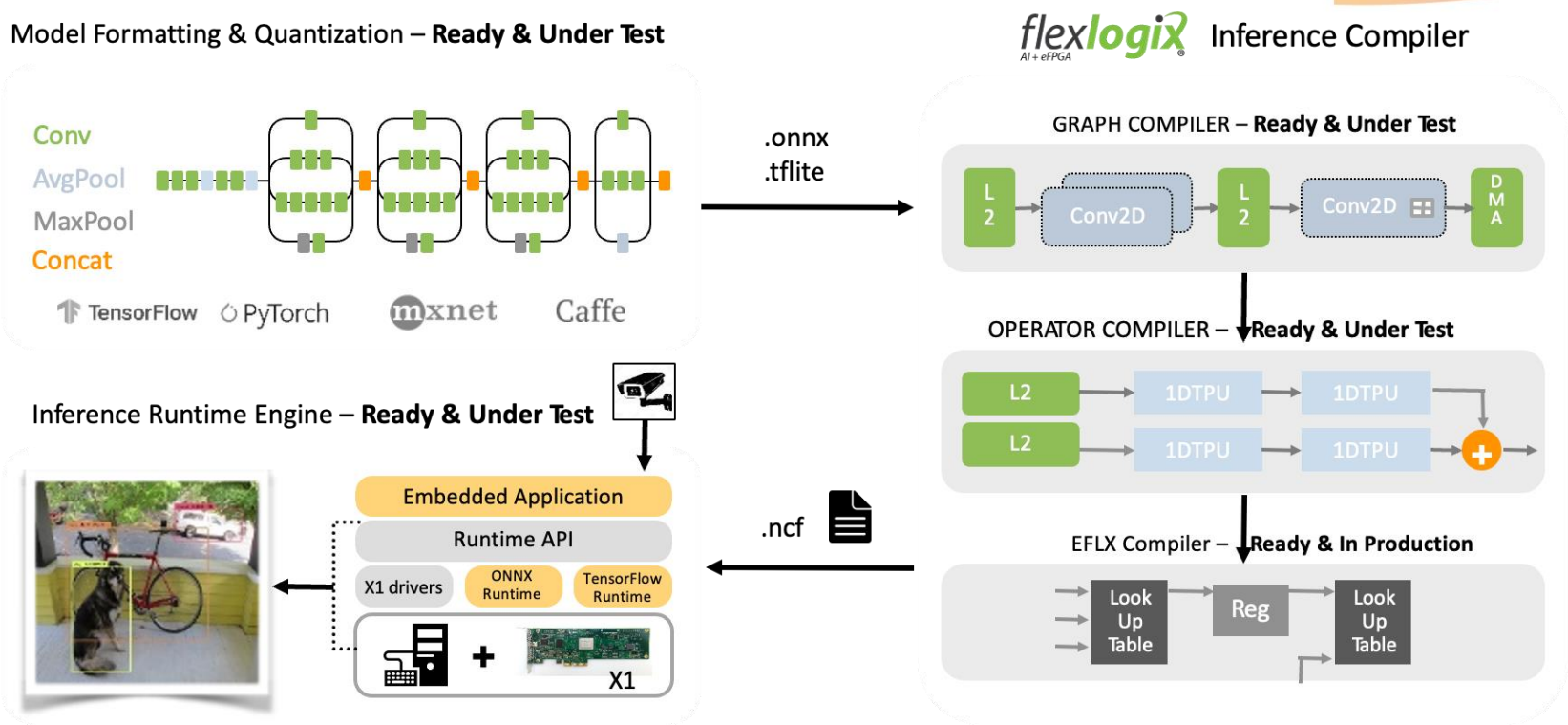
- Optimization
- Quantization
- Compilation

Your Inference Application

Your SoC with InferX accelerator

flexlogix reconfigurable

InferX compiler is available for evaluation



Come see us for more information!

- InferX IP provides flexibility to future-proof your AI solution, including state-of-the art CNN & transformers workloads
- Please come visit our booth for demos and brochures
- Please visit www.flex-logix.com for more information