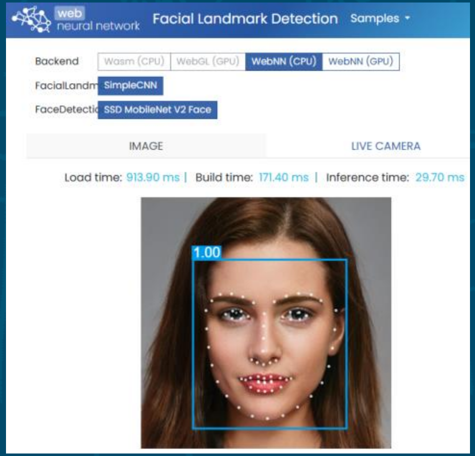
# WebNN Operator Update
# Wave 3

web machinelearning

DWAYNE ROBINSON
2024-09-22

# Wave 1 🌊, 60 ops – 2020..2024

- MobileNet, ResNet (image object classification)
- Face Landmark (facial recognition)
- Tiny YOLO (image object detection)
- MNIST (number classification)
- Fast Style Transfer
- NSNet2 (noise suppression)



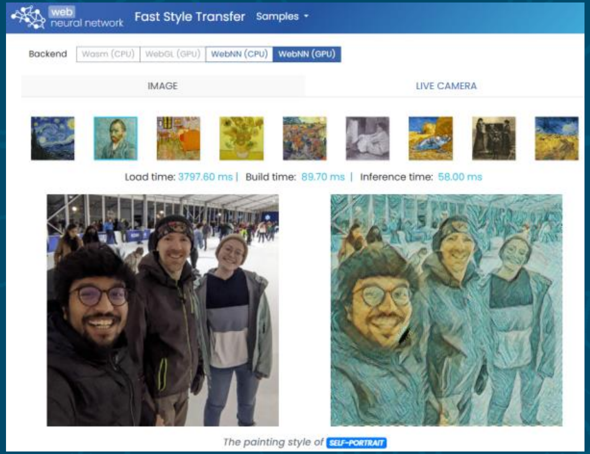
Facial Landmark Detection

Backend Wasm (CPU) WebGL (GPU) WebNN (CPU) WebNN (GPU)

FacialLandm SimpleCNN

FaceDetecti SSD MobileNet V2 Face

Load time: 913.90 ms | Build time: 171.40 ms | Inference time: 29.70 ms

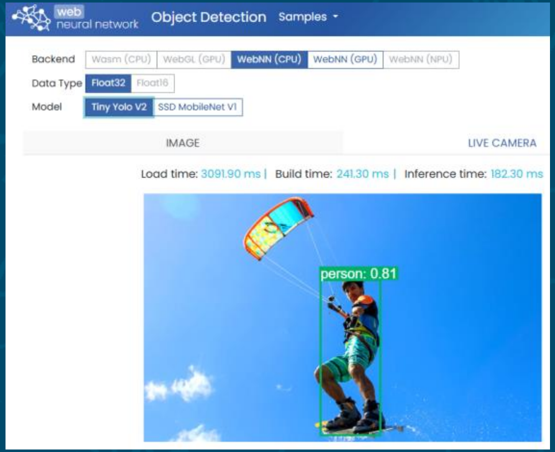


Fast Style Transfer

Backend Wasm (CPU) WebGL (GPU) WebNN (CPU) WebNN (GPU)

Load time: 3797.60 ms | Build time: 89.70 ms | Inference time: 58.00 ms

The painting style of SELF-PORTRAIT



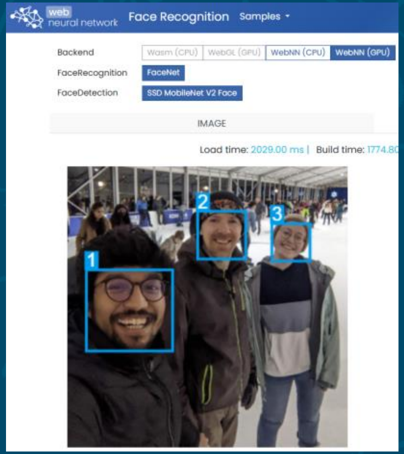
Object Detection

Backend Wasm (CPU) WebGL (GPU) WebNN (CPU) WebNN (GPU) WebNN (NPU)

Data Type Float32 Float16

Model Tiny Yolo V2 SSD MobileNet V1

Load time: 3091.90 ms | Build time: 241.30 ms | Inference time: 182.30 ms



Face Recognition

Backend Wasm (CPU) WebGL (GPU) WebNN (CPU) WebNN (GPU)

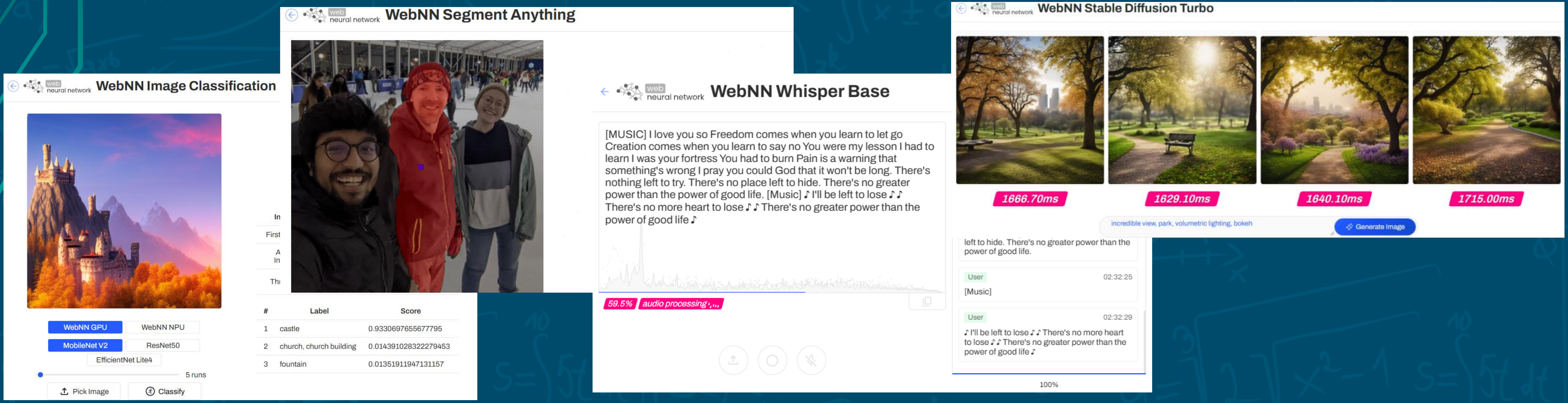
FaceRecognition FaceNet

FaceDetection SSD MobileNet V2 Face

Load time: 2029.00 ms | Build time: 1774.80

# Wave 2 – Transformers 2023-08-10+

- Segment Anything (image segmentation)
- Stable Diffusion 1.5, SD Turbo (image generation)
- Whisper base (audio to text)



WebNN Image Classification

| # | Label | Score |
|---|---|---|
| 1 | castle | 0.9330697655677795 |
| 2 | church, church building | 0.014391028322279453 |
| 3 | fountain | 0.01351911947131157 |

WebNN Whisper Base

[MUSIC] I love you so Freedom comes when you learn to let go Creation comes when you learn to say no You were my lesson I had to learn I was your fortress You had to burn Pain is a warning that something's wrong I pray you could God that it won't be long. There's nothing left to try. There's no place left to hide. There's no greater power than the power of good life. [Music] ♪ I'll be left to lose ♪ ♪ There's no more heart to lose ♪ ♪ There's no greater power than the power of good life ♪

59.5% audio processing ,,,,

left to hide. There's no greater power than the power of good life.

User 02:32:25

[Music]

User 02:32:29

♪ I'll be left to lose ♪ ♪ There's no more heart to lose ♪ ♪ There's no greater power than the power of good life ♪

100%

WebNN Stable Diffusion Turbo

1666.70ms 1629.10ms 1640.10ms 1715.00ms

incredible view, park, volumetric lighting, bokeh

# Wave 2 🌊 – Transformers 2023-08-10+

- +21 ops
- argMax / argMin – find value in tensor, return index
- cast – change data type (essential gap)
- equal / greater / lesser / .. – compare elementwise
- logicalNot – invert boolean elementwise
- erf – Gauss error function
- expand – broadcast up to new size (used by ConstantOfShape)
- ~~unsqueeze – missing corollary to Squeeze~~ (just resolve to reshape)
- ~~flattenTo2d – kin to squeeze/unsqueeze~~ (just resolve to reshape)
- gather – collect values from indices
- identity – placeholder and direct mapping for callers
- ~~fillSequence – fill numeric sequence from/to/step~~
- triangular – fill upper or lower triangular part of matrix
- where – elementwise source selection (select)
- ~~meanVarianceNormalization – encompasses instance/norm/layer/groupedChannel…~~ (added batchNormalization, layerNormalization, InstanceNormalization)
- shape getter – essential for sanity debugging and printing results
- reciprocal – dedicated to op (avoid 1/div work-around)
- sqrt – dedicated op (avoid 0.5 extra tensor)
- gelu – notable perf win for Whisper model, vs x * 0.5 * (1.0 + erf(x / sqrt(2)))

# Wave 2 🌊 – Transformers 2023-08-10+

- Also added:
  - 0D scalars
  - shape() and dataType() getters
  - int64 data type, used very frequently in ONNX models for all things indexish (avoiding lots of casts and copies by caller)

# Wave 2 🌊 – Transformers 2023-08-10+

- Backend maturity:
  - CoreML – 65/78
  - DirectML – 78/78
  - TFLite – 65/78

# Wave 3 – Transformers 2024-08-15+

- Popular Hugging Face models, including top 20 downloaded

| Model | Category | Link |
|---|---|---|
| Tiny-llama | Small Language Model | https://huggingface.co/Xenova/TinyLLama-v0 https://huggingface.co/Xenova/TinyLlama-1.1B-Chat-v1.0 |
| Phi 3 mini | Small Language Model | https://huggingface.co/microsoft/Phi-3-mini-4k-instruct-onnx-web |
| Yolov8 | Object detection | https://github.com/ultralytics/ultralytics |
| DETR | Object detection | https://huggingface.co/Xenova/detr-resnet-50, https://huggingface.co/Xenova/detr-resnet-101 |
| Llama3 | Large Language Model | https://huggingface.co/aless2212/Meta-Llama-3-8B-Instruct-onnx-fp16 |
| nomic-ai/nomic-embed-text-v1.5 | Sentence similarity | https://huggingface.co/nomic-ai/nomic-embed-text-v1.5 |
| Supabase/gte-small | Feature Extraction | https://huggingface.co/Supabase/gte-small |
| mixedbread-ai/mxbai-embed-large-v1 | Feature Extraction | https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1 |
| nomic-ai/nomic-embed-text-v1 | Sentence similarity | https://huggingface.co/nomic-ai/nomic-embed-text-v1 |
| WhereIsAI/UAE-Large-V1 | Feature Extraction | https://huggingface.co/WhereIsAI/UAE-Large-V1 |
| distil-whisper/distil-medium.en | Speech Recognition | https://huggingface.co/distil-whisper/distil-medium.en |
| Alibaba-NLP/gte-base-en-v1.5 | Sentence similarity | https://huggingface.co/Alibaba-NLP/gte-base-en-v1.5 |
| jonathandinu/face-parsing | Image segmentation | https://huggingface.co/jonathandinu/face-parsing |
| jinaai/jina-clip-v1 | Feature extraction | https://huggingface.co/jinaai/jina-clip-v1 |
| mixedbread-ai/mxbai-rerank-base-v1 | Text classification | https://huggingface.co/mixedbread-ai/mxbai-rerank-base-v1 |
| Snowflake/snowflake-arctic-embed-m | Sentence similarity | https://huggingface.co/Snowflake/snowflake-arctic-embed-m |
| jinaai/jina-embeddings-v2-base-code | Feature extraction | https://huggingface.co/jinaai/jina-embeddings-v2-base-code |
| Xenova/llama2.c-stories15M | Text generation | https://huggingface.co/Xenova/llama2.c-stories15M |
| corto-ai/nomic-embed-text-v1 | Sentence similarity | https://huggingface.co/corto-ai/nomic-embed-text-v1 |
| jinaai/jina-reranker-v1-turbo-en | Text classification | https://huggingface.co/jinaai/jina-reranker-v1-turbo-en |
| Xenova/bge-reranker-base | Text classification | https://huggingface.co/Xenova/bge-reranker-base |
| Xenova/bge-large-en-v1.5 | Feature extraction | https://huggingface.co/Xenova/bge-large-en-v1.5 |
| Xenova/distiluse-base-multilingual-cased-v2 | Feature extraction | https://huggingface.co/Xenova/distiluse-base-multilingual-cased-v2 |
| Xenova/paraphrase-multilingual-mpnet-base-v2 | Feature extraction | https://huggingface.co/Xenova/paraphrase-multilingual-mpnet-base-v2 |
| CAiRE/UniVaR-lambda-1 | Sentence similarity | https://huggingface.co/CAiRE/UniVaR-lambda-1 |

# Wave 3 – Transformers 2024-08-15+

- +12 ops (smaller delta)
- Data reorganization:
  - gatherElements (gatherAlongAxis) – gather inputs from indices
  - scatterElements (scatterAlongAxis) – scatter updates to indices
  - gatherND – gather inputs from coordinates
  - scatterND – scatter inputs from coordinates
  - tile – repeat a tensor the given times along each dimension
- Elementwise unary
  - sign – return -1,0,1 depending on <0,==0,>0.
- Elementwise binary
  - logicalAnd – a & b
  - logicalOr – a | b
  - logicalXor – a ^ b
  - notEqual – a != b. Concise not(equal(a, b)) to complete eq/lt/gt/ge/le set
- Elementwise trinary
  - dequantizeLinear - (input - zeroPoint) * scale
  - quantizeLinear - clamp(roundToNearestEvens(input / scale) + zeroPoint, 0, 255)

- Excluded
  - dropout – just map to identity for inference.
  - einSum – decompose in caller. Maps to existing operators – avoid string parsing at runtime for WebNN low-level layer.
  - localResponseNormalization – decompose in caller {averagePool, add, mul, div, pow) because of various implementation inconsistencies (odd size support, edge treatment, which axes).

# Wave 3 🌊 – Transformers 2024-08-15+

```
partial interface MLGraphBuilder
{
   ...
   MLOperand cumulativeSum(MLOperand input, unsigned long axis, optional MLCumulativeSumOptions options = {});
   MLOperand sign(MLOperand input, optional MLOperatorOptions options = {});
   MLOperand tile(MLOperand input, sequence<unsigned long> repetitions, optional MLOperatorOptions options = {});

   MLOperand gatherElements(MLOperand input, MLOperand indices, optional MLGatherOptions options = {});
   MLOperand scatterElements(MLOperand input, MLOperand indices, MLOperand updates, optional MLScatterOptions options = {});
   MLOperand gatherND(MLOperand input, MLOperand indices, optional MLOperatorOptions options = {});
   MLOperand scatterND(MLOperand input, MLOperand indices, MLOperand updates, optional MLOperatorOptions options = {});

   MLOperand dequantizeLinear(MLOperand input, MLOperand scale, MLOperand zeroPoint, optional MLOperatorOptions options = {});
   MLOperand quantizeLinear(MLOperand input, MLOperand scale, MLOperand zeroPoint, optional MLOperatorOptions options = {});

   MLOperand logicalAnd(MLOperand a, MLOperand b, optional MLOperatorOptions options = {});
   MLOperand logicalOr(MLOperand a, MLOperand b, optional MLOperatorOptions options = {});
   MLOperand logicalXor(MLOperand a, MLOperand b, optional MLOperatorOptions options = {});
   MLOperand notEqual(MLOperand a, MLOperand b, optional MLOperatorOptions options = {});
}
```

# Wave 3 🌊 – Transformers 2024-08-15+

- Smaller data types
  - Models growing very large, even with uint8
  - 32-bit WebAssembly 4GB address space limits (WASM64 pending…)
  - uint4/int4 not computable type, just for expansion (DQ, Q)

```
enum MLOperandDataType {
  "float32",
  "float16",
  "int32",
  "uint32",
  "int64",
  "uint64",
  "int8",
  "uint8",
  "uint4",
  "int4",
};
```

# What’s next? 📅

- Fill in primitive breadth 🚛 📦
  - Being model based is good to show viability and demos, but for breadth…
  - Adding increasingly more operators untenable / hard to validate
  - Useful for composition of custom operators 🪨 🪆 👉 🧱
    - Bitwise operators (and, or, xor, left shift, right shift)
    - Modulus/remainder, flooring divide
    - Rounding (nearest even, toward zero, toward infinity)
    - Random number generation
    - sumPool/minPool
    - …
- Some more specialized ops: FFT, MHA 🐆
- Polish/relax some awkward aspects (e.g. dimension limitations)
- Decide minimal data type set in opSupportLimits and const input issues
- Backend parity maturity, finishing lingering WPT’s, prototyping results
- Origin trial 🏁 🚩 ⏳