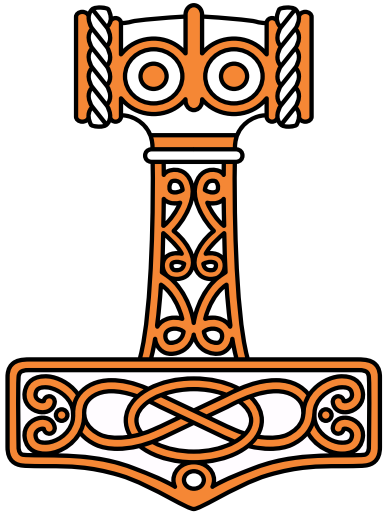
DYALOG

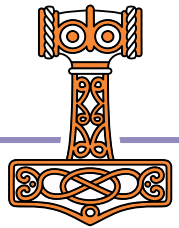
Olhão 2022

Implementing the U-Net CNN in APL



Rodrigo Girão Serrão

A Portuguese and an American walk into a bar...



U-net CNN in APL

Exploring zero-framework, zero-library machine learning

Aaron W. Hsu

Bloom States

Rodrigo Girão Serrão
rodrigo@dyalog.com
Consultant
Dyalog, Ltd.
Bramley, United Kingdom

Abstract

The APL notation wou ear match for convolutional neural networks, but traditional implementations of APL have lagged behind the performance of highly tuned, specialized frameworks designed to execute CNNs on the GPU. Moreover, most demonstrations of APL for neural net-

1 Introduction

Specialized machine learning frameworks dominate the present industrial and educational spaces for deep learning applications. A wide number of higly specialized and highly optimized libraries exist, often built on top of one another, to support the modern wave of machine learning architec-

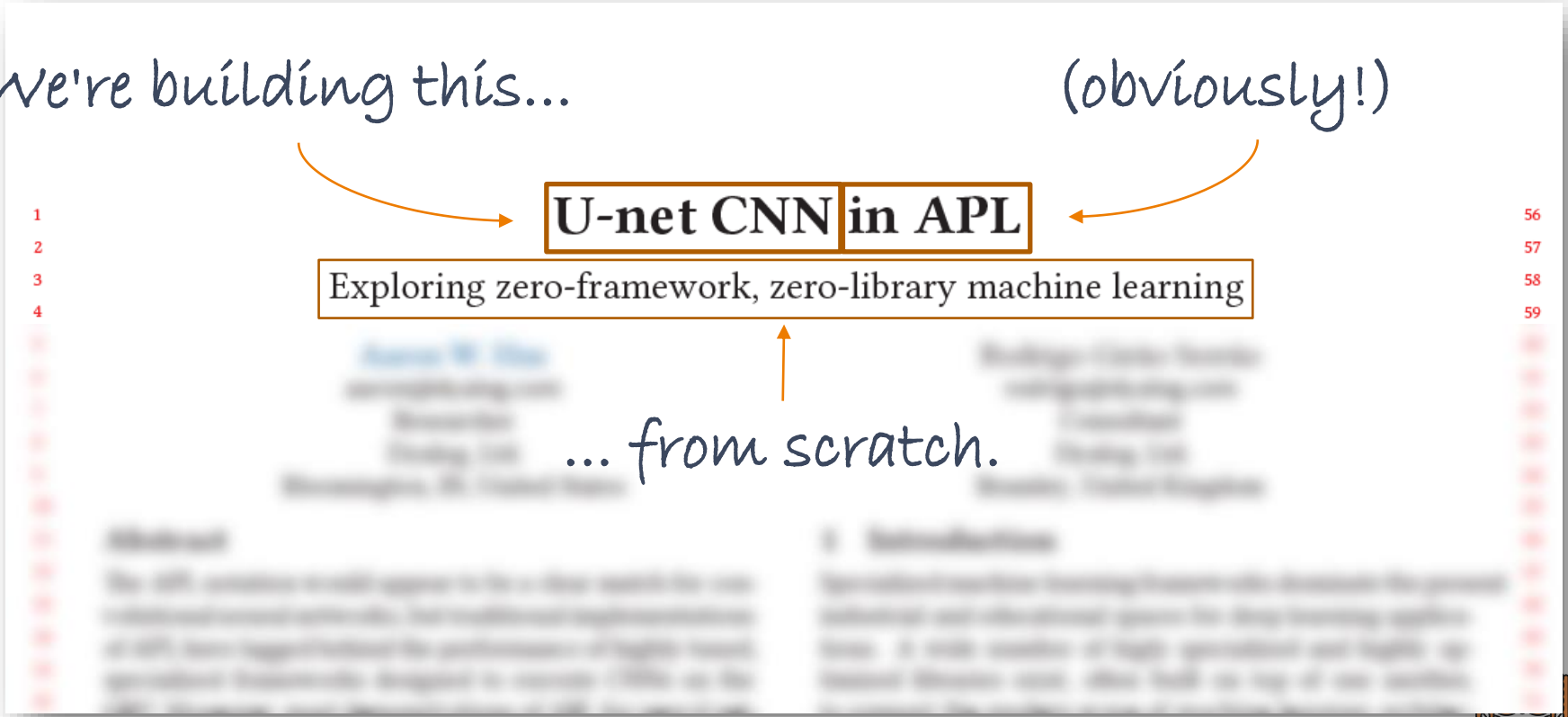
We're building this...

(obviously!)

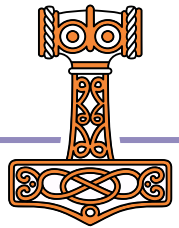
U-net CNN in APL

Exploring zero-framework, zero-library machine learning

... from scratch.



CNN?

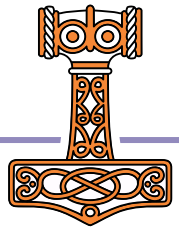


CNN

Convolutional

Neural

Network



CNN

Convolutional

Neural

Network

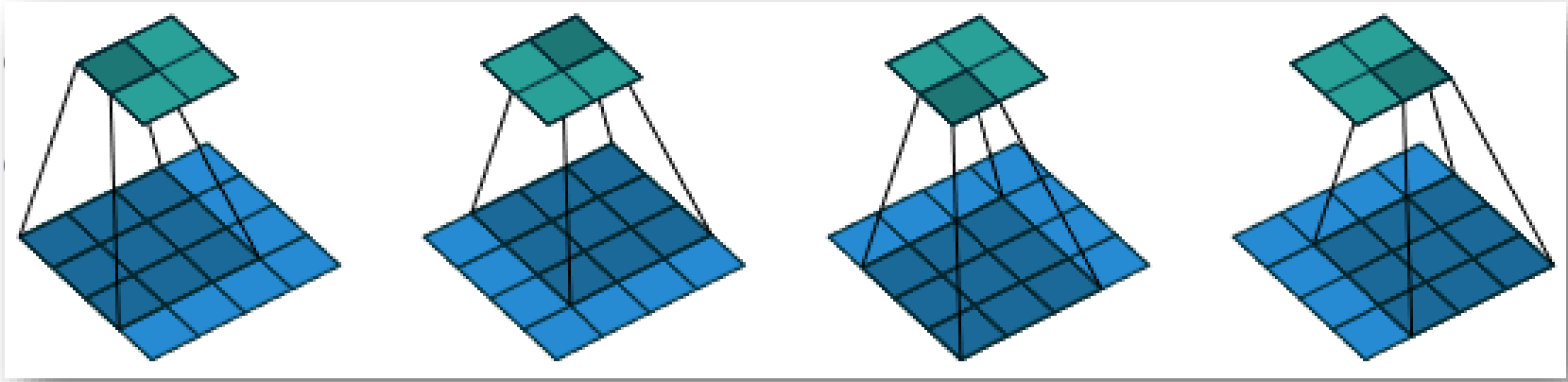
CNN

```
{+/,K×ω}⌺3 3⊢
```

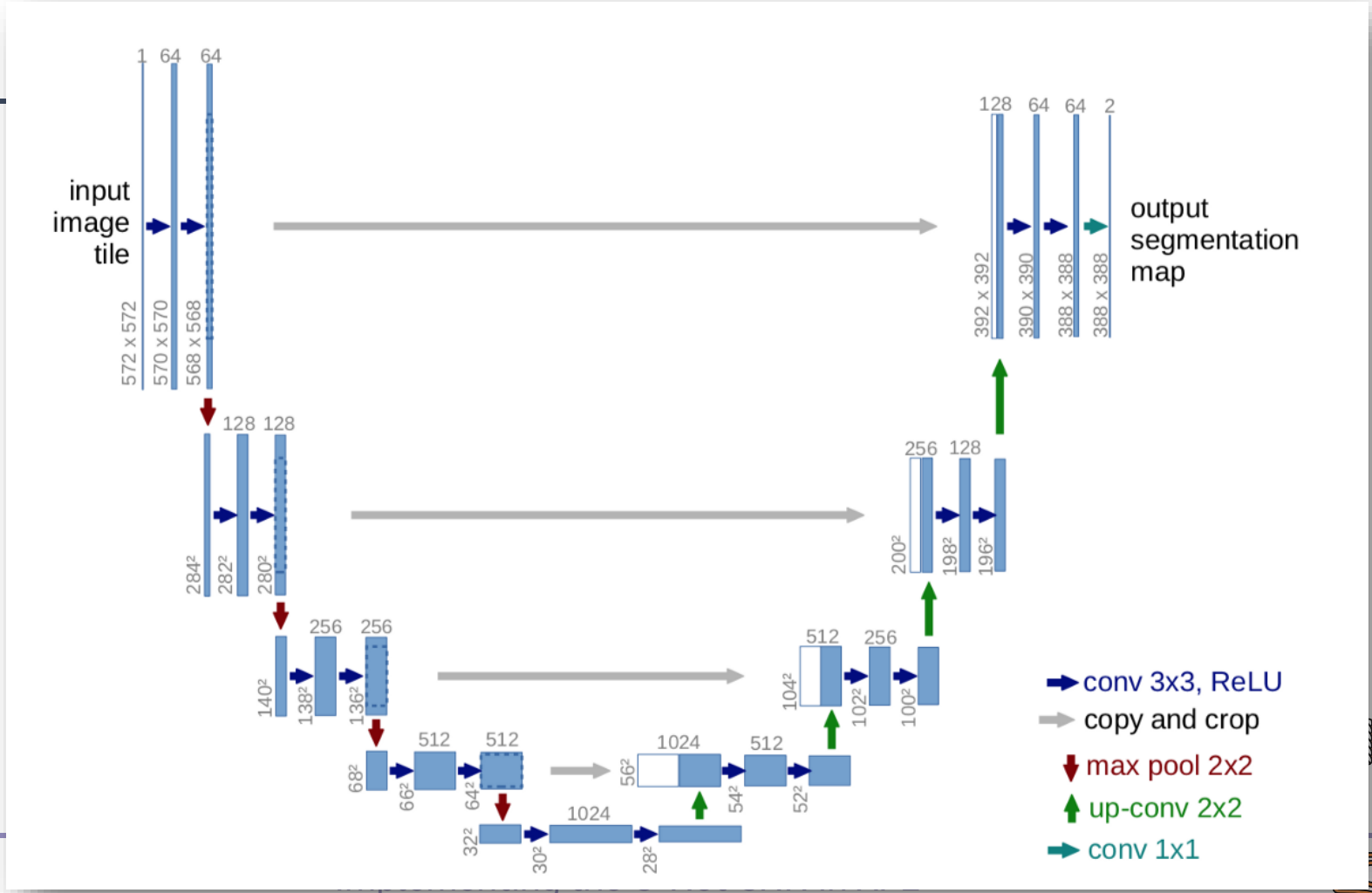
Convolutional

N

N

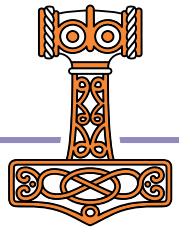


U-


input image tile
1 64 64
572 x 572
570 x 570
568 x 568
128 128
284²
282²
280²
256 256
140²
138²
136²
512 512
68²
66²
64²
1024
32²
30²
28²
512
56²
54²
256
104²
102²
100²
128
200²
198²
196²
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

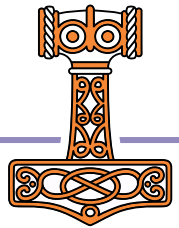
The paper – why?

- APL looks good for CNNs
  - Is it?
- Co-dfns looks like a good compiler
  - Will it run fast?



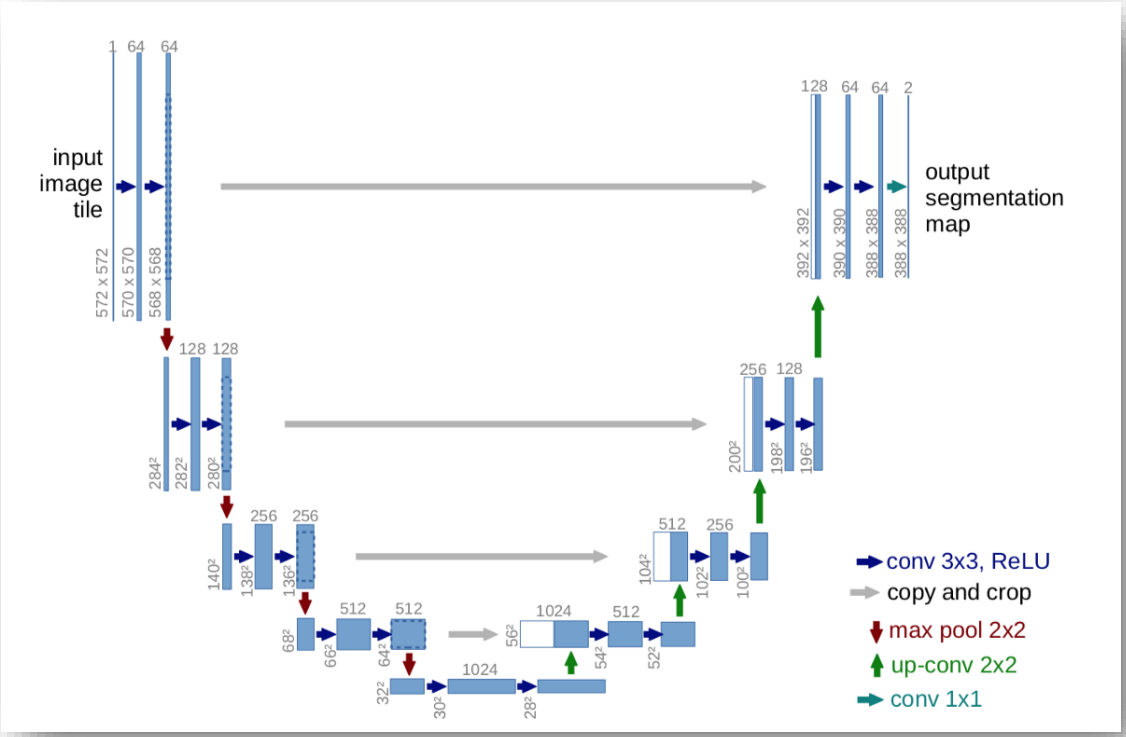
The paper – what we did

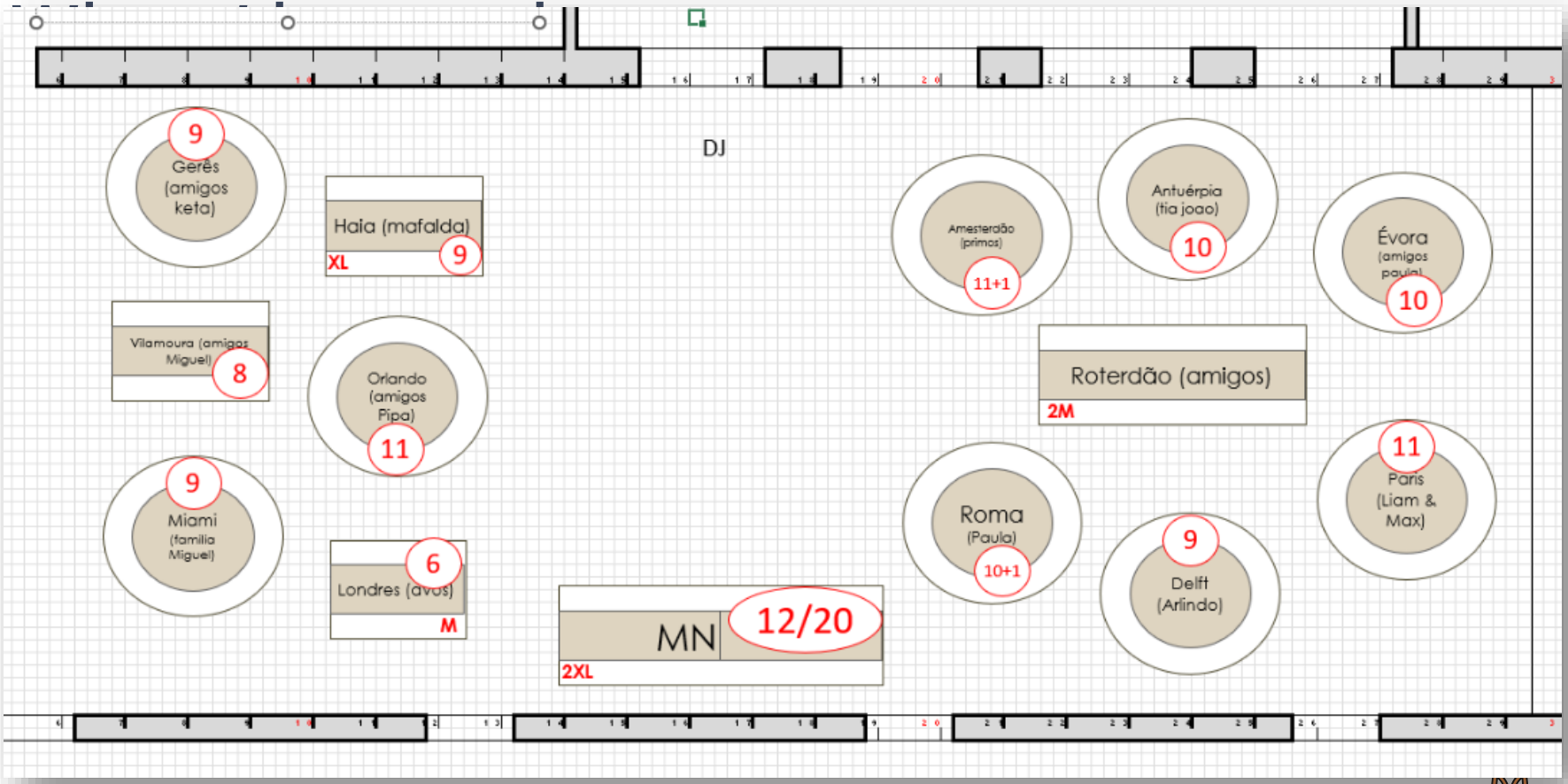
- Implemented the “U-net” in APL
- Compared it to industry-standard framework:
  - code size; and
  - runtime performance.
- Discussed advantages of our approach



What I learned

- How do

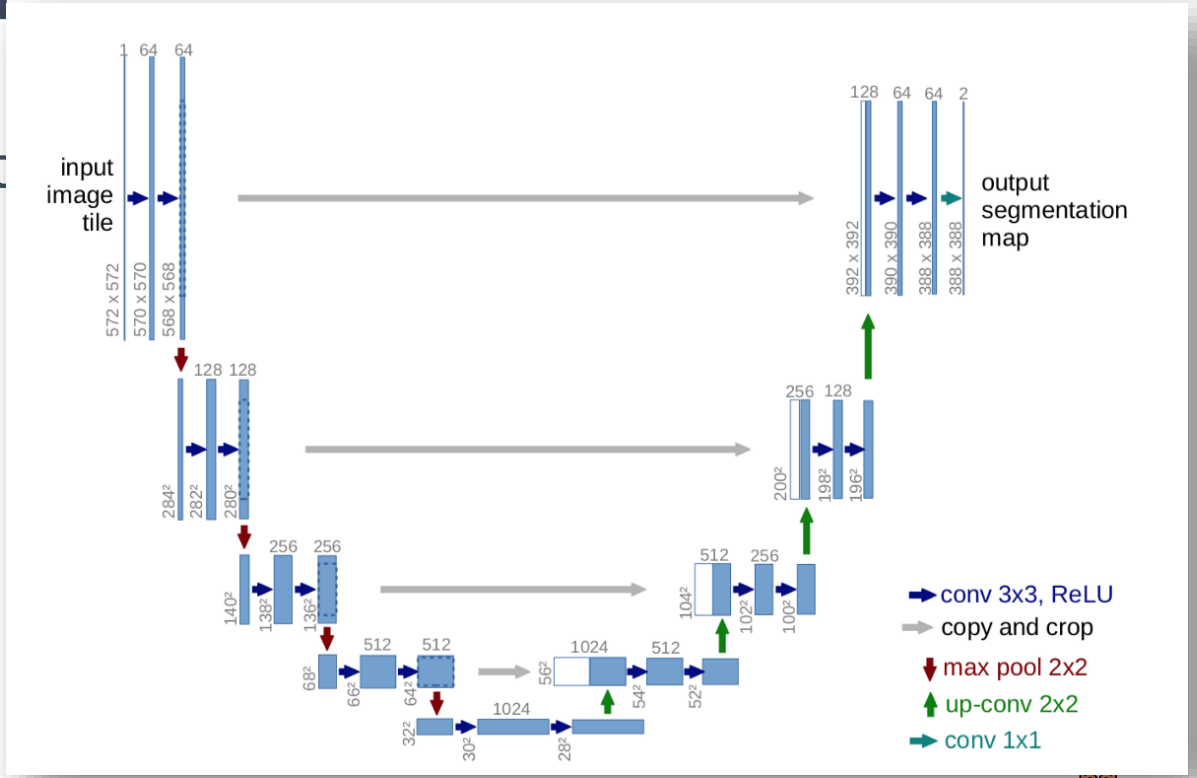



9
Gerês (amigos keta)
Haia (mafalda)
XL
9
DJ
Amesterão (primos)
11+1
Antuérpia (tia joao)
10
Évora (amigos
10
Vilamoura (amigos Miguel)
8
Orlando (amigos Pipa)
11
Roterdão (amigos)
2M
11
Paris (Liam & Max)
9
Miami (familia Miguel)
Roma (Paula)
10+1
9
Delft (Arlindo)
6
Londres (avos)
M
MN
12/20
2XL

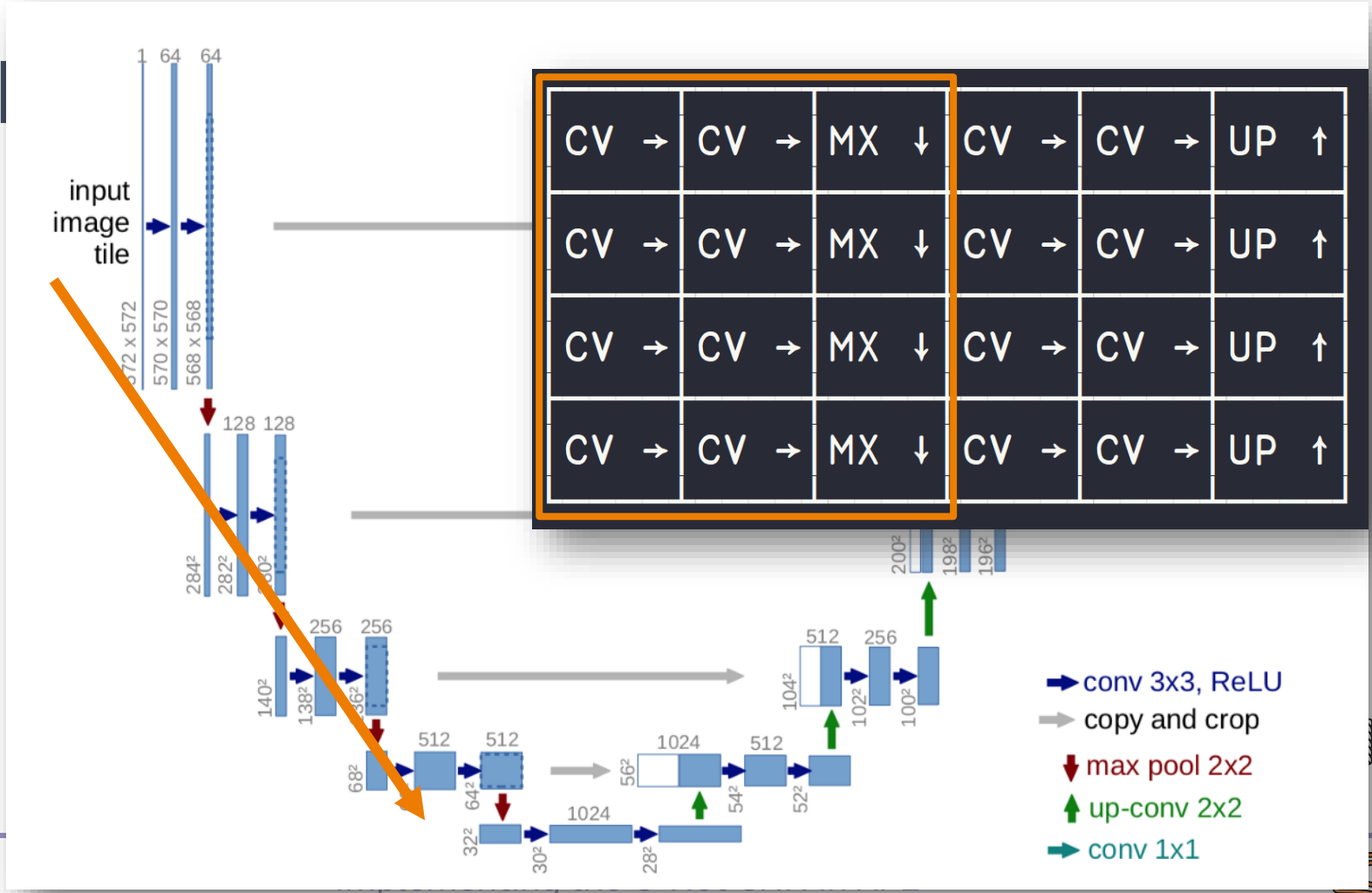
What I learned

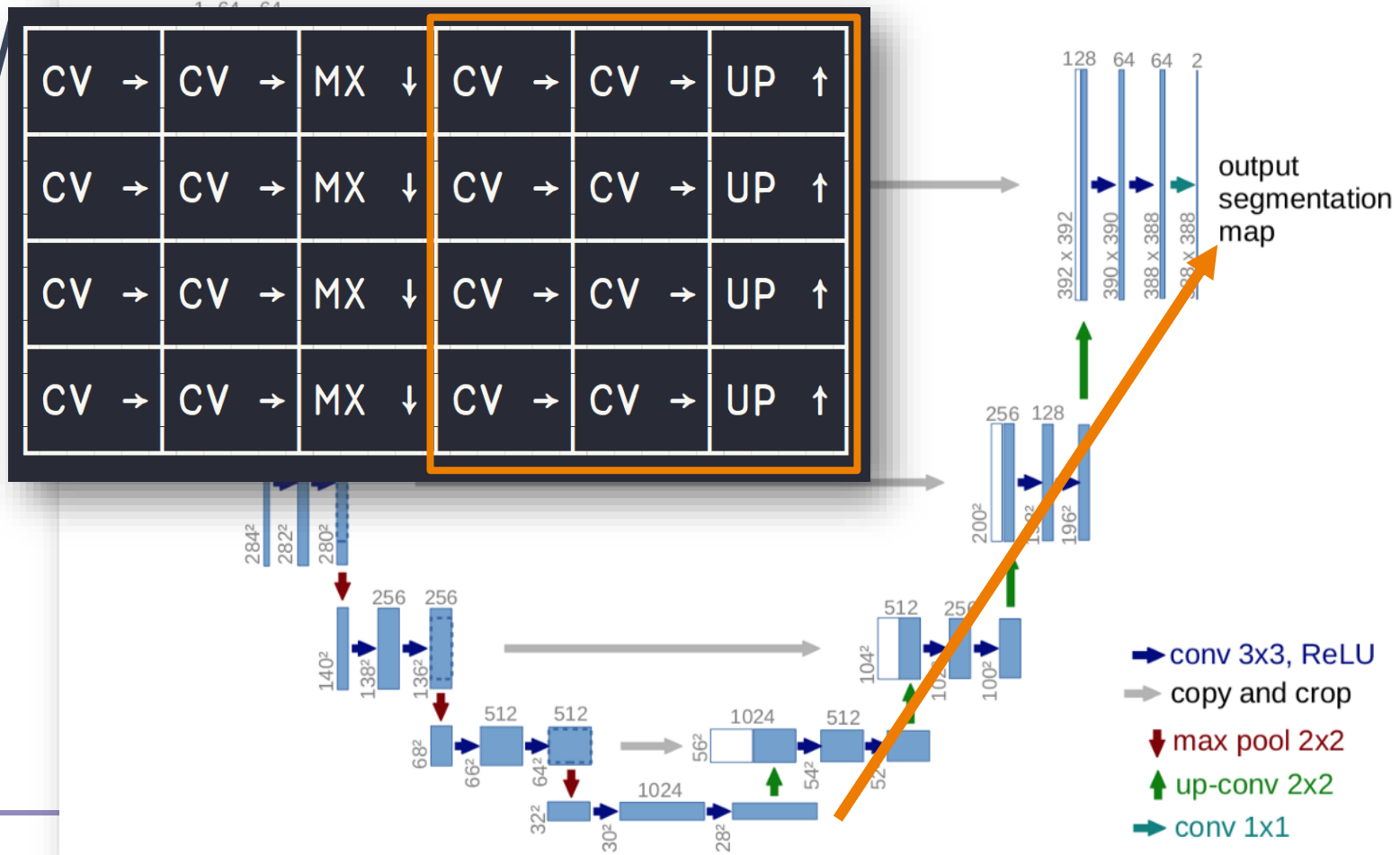
- How do I arrange t

CV →	CV →	MX ↓	CV →	CV →	UP ↑
CV →	CV →	MX ↓	CV →	CV →	UP ↑
CV →	CV →	MX ↓	CV →	CV →	UP ↑
CV →	CV →	MX ↓	CV →	CV →	UP ↑



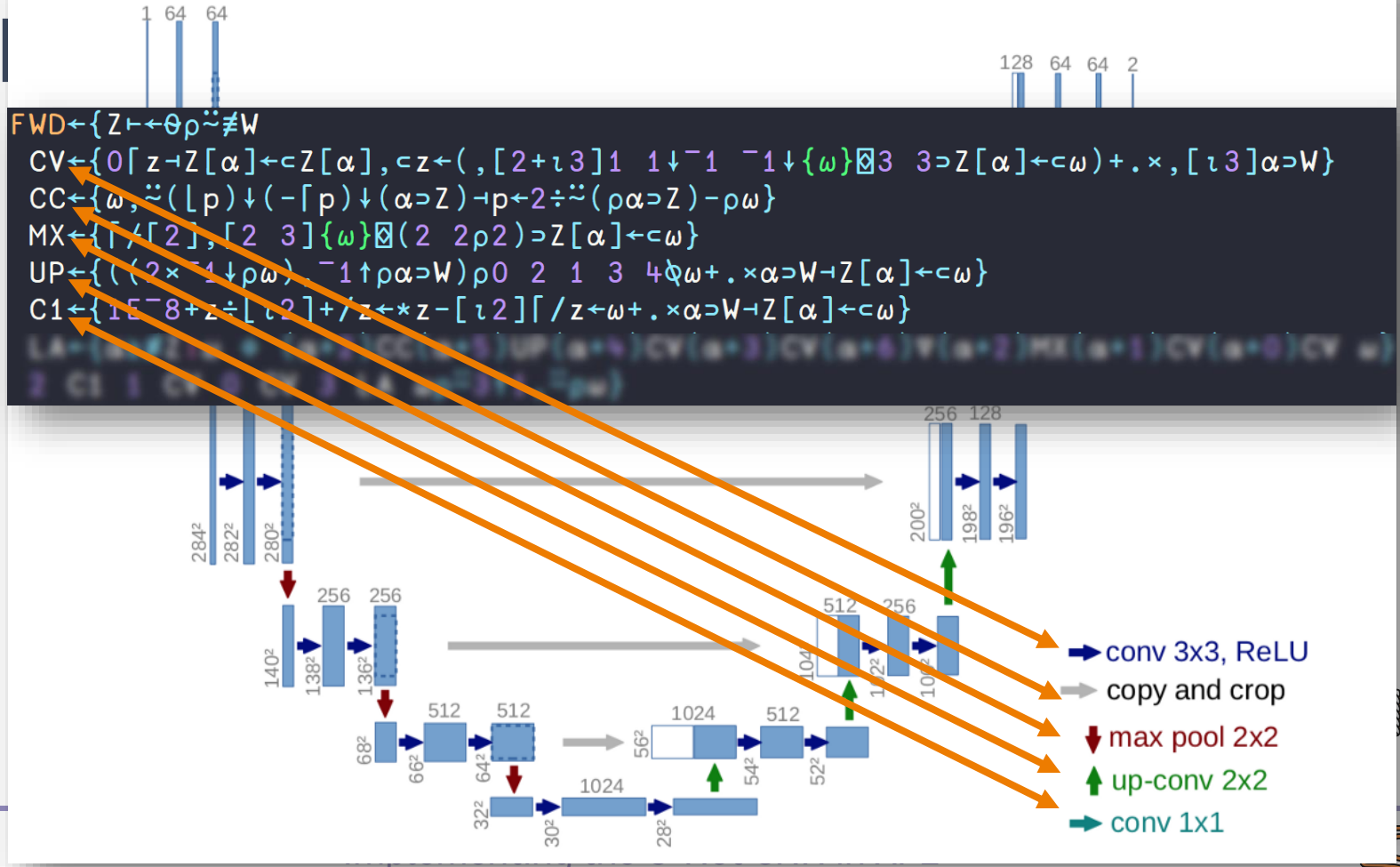
Wh


input image tile
1 64 64
572 x 572
570 x 570
568 x 568
128 128
284²
282²
256 256
140²
138²
512 512
68²
64²
1024
32²
30²
1024
56²
512
54²
512 256
104²
102²
100²
200²
198²
196²
CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

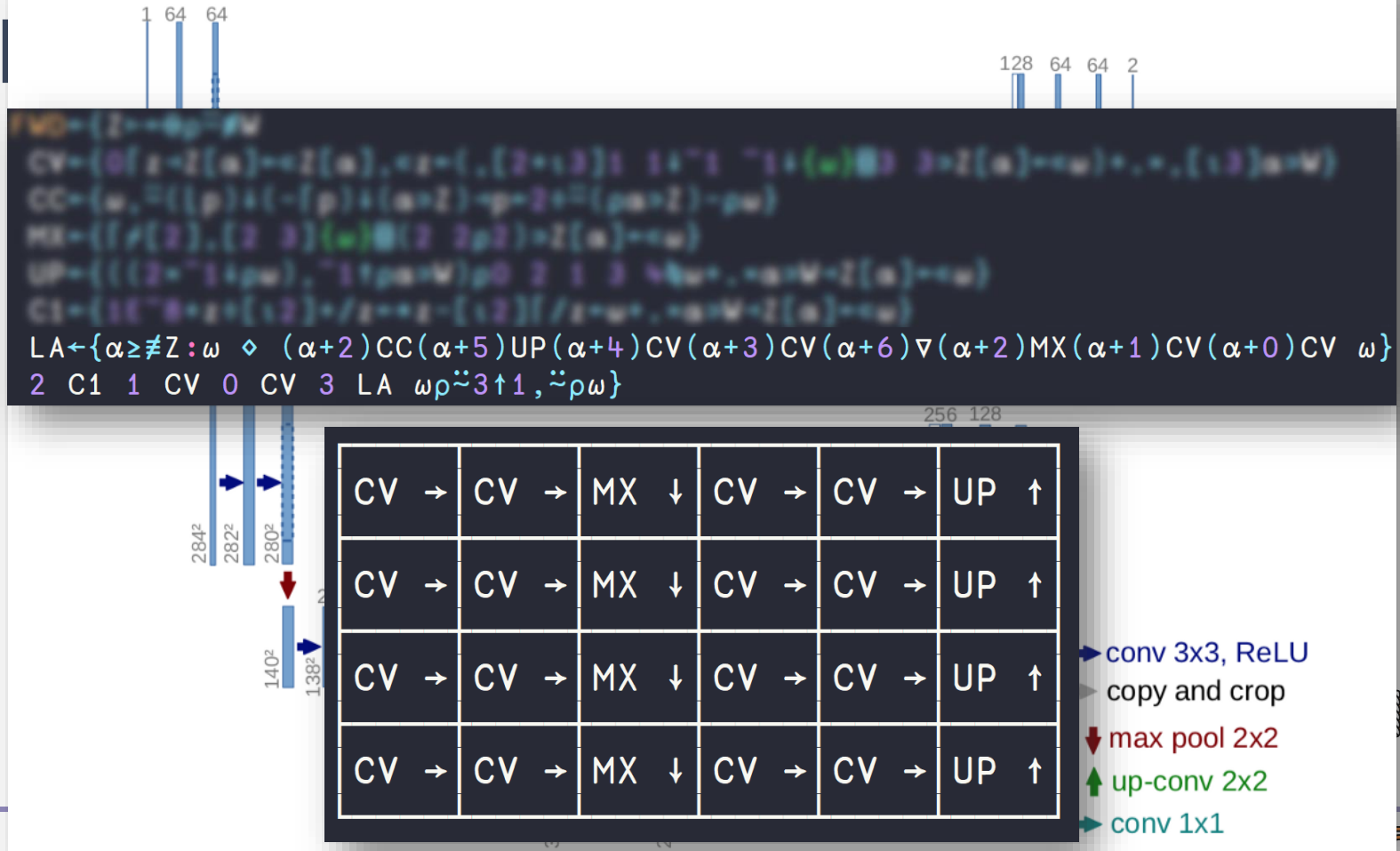

CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
128 64 64 2
392 x 392
390 x 390
388 x 388
output segmentation map
256 128
512 256
1024 512
1024
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Wl

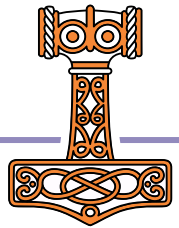
```
FWD←{Z⊢←⍬ρ⍤¯≠W
 CV←{0⌈z⊣Z[α]←⊂Z[α],⊂z←(,[2+⍳3]1 1↓⍤¯1 ¯1↓{ω}⌺3 3⊃Z[α]←⊂ω)+.×,[⍳3]α⊃W}
 CC←{ω,⍤¯(⌊p)↓(-⌈p)↓(α⊃Z)⊣p←2÷⍤¯(ρα⊃Z)-ρω}
 MX←{⌈/[2],[2 3]{ω}⌺(2 2ρ2)⊃Z[α]←⊂ω}
 UP←{((2×¯1↓ρω),¯1↑ρα⊃W)ρ0 2 1 3 4⍉ω+.×α⊃W⊣Z[α]←⊂ω}
 C1←{1⌿¯8+z÷[⍳2]+/z←*z-[⍳2]⌈/z←ω+.×α⊃W⊣Z[α]←⊂ω}
```



Wh


1 64 64
128 64 64 2
LA←{α≥≢Z:ω ⋄ (α+2)CC(α+5)UP(α+4)CV(α+3)CV(α+6)∇(α+2)MX(α+1)CV(α+0)CV ω}
2 C1 1 CV 0 CV 3 LA ωρ¨3↑1,¨ρω}
256 128
284²
282²
280²
140²
138²
CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
CV → CV → MX ↓ CV → CV → UP ↑
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

What I learned

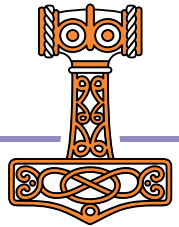



and out of nowhere
they bring out a secret menu.

What I learned

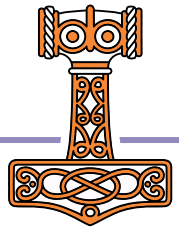
- Inner products are ***super*** useful

```
FWD←{Z⊢←⍬⍴⍤¯1≠W
 CV←{0⌈z⊣Z[α]←⊂Z[α],⊂z←(,[2+⍳3]1 1↓¯1 ¯1↓{ω}⌺3 3⊃Z[α]←⊂ω +.× [⍳3]α⊃W}
 CC←{ω,⍤¯1(⌊p)↓(-⌈p)↓(α⊃Z)⊣p←2÷⍤¯1(⍴α⊃Z)-⍴ω}
 MX←{⌈⌿[2],[2 3]{ω}⌺(2 2⍴2)⊃Z[α]←⊂ω}
 UP←{((2×¯1↓⍴ω),¯1↑⍴α⊃W)⍴0 2 1 3 4⍉ω+.×α⊃W⊣Z[α]←⊂ω}
 C1←{1E¯8+z÷[⍳2]+/z←*z-[⍳2]⌈/z←ω+.×α⊃W⊣Z[α]←⊂ω}
 LA←{α≥≢Z:ω ⋄ (α+2)CC(α+5)UP(α+4)CV(α+3)CV(α+6)∇(α+2)MX(α+1)CV(α+0)CV ω}
 2 C1 1 CV 0 CV 3 LA ω⍴⍤¯3↑1,⍤¯1⍴ω}
```



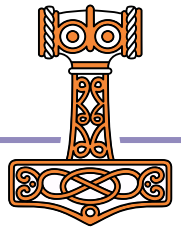
What I learned

- Inner products are super useful
- Arrange axis carefully
- Stencil has optimised left operands



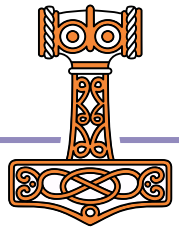
What I learned

```
MX←{                    {⌈⌿⌈⌿ω}⌺(2 2⍴2)ω}
```



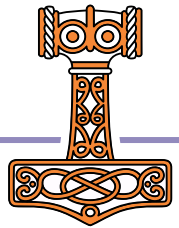
What I learned

```
MX←{          {⌈⌿⌈⌿ω}⌺(2 2⍴2)ω}
MX←{⌈⌿[2],[2 3]{ω}⌺(2 2⍴2)ω}
```



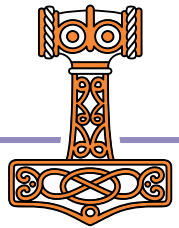
What I learned

- Inner products are super useful
- Arrange axis carefully
- Stencil has optimised left operands
- Optimising idioms is a combinatorial problem



What I learned

- It takes a bit to digest...
- Once you do, you have superpowers!
- Go study the paper :)



Questions?