



UNIVERSITY OF TWENTE.

EMBEDDED AI

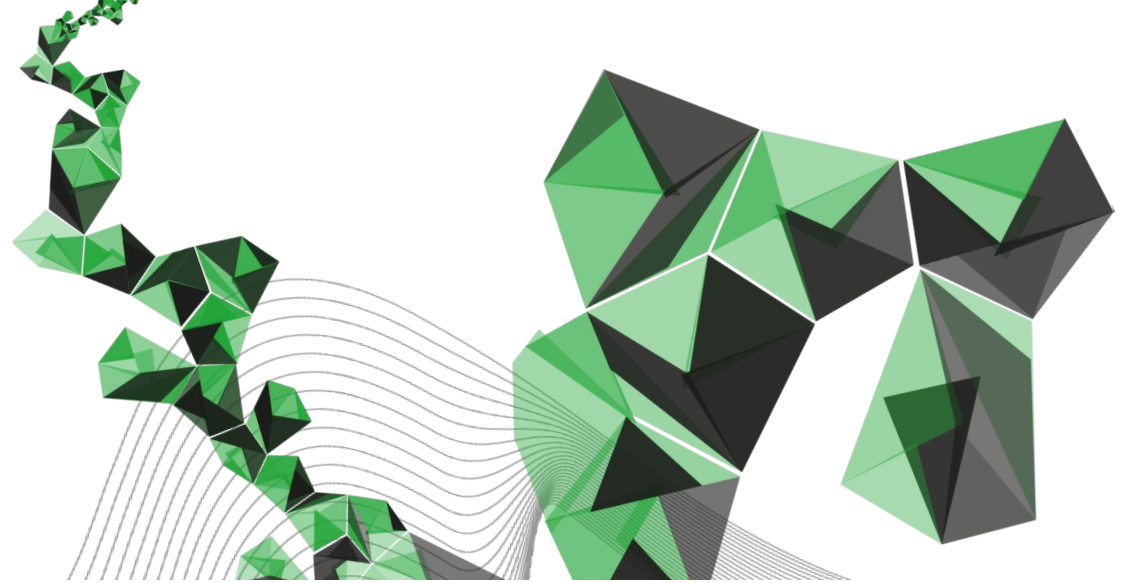
DSI SEMINAR BY SEBASTIAN BUNDA MSc



WHO AM I?

- PhD Student @ DMB and CAES
 - Subject: Embedded AI
- Background in Electrical Engineering: 2015-2022
 - Specialisation Computer Vision & Biometrics @Twente





EMBEDDED AI



UNIVERSITY
OF TWENTE.



ARTIFICIAL INTELLIGENCE

DEFINITION

Artificial Intelligence is an umbrella term for various computational strategies able to display **human-like capabilities** such as reasoning and learning

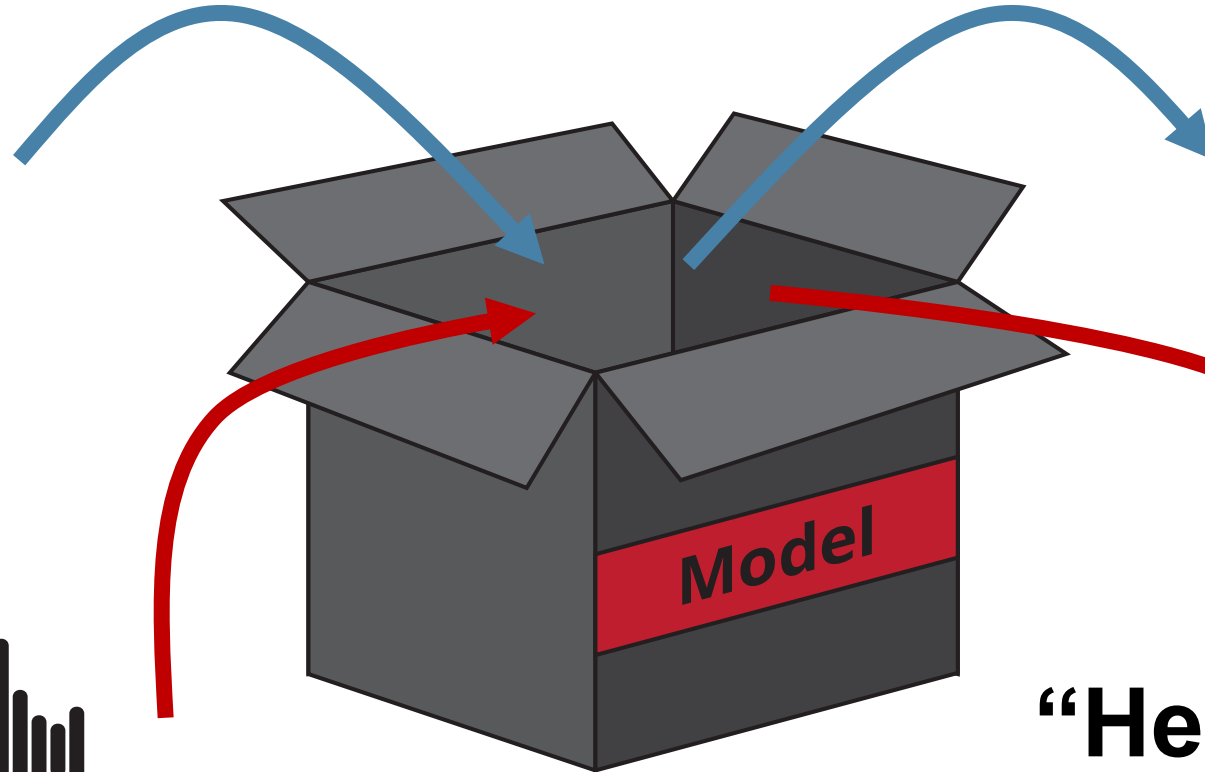
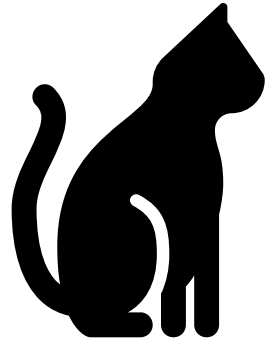
*Examples: **machine learning, robotics and natural language processing***





MACHINE LEARNING

DEEP LEARNING MODEL



CAT

“Hey Google”



Key Concepts



UNIVERSITY
OF TWENTE.



Key Concepts

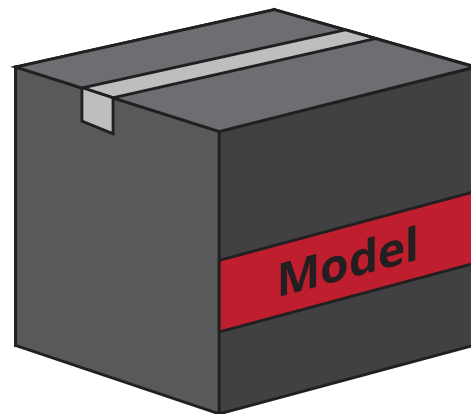
TYPES DEEP NEURAL NETWORKS

**Convolutional
Neural Network**

e.g. Image Classification

**Recurrent
Neural Network**

e.g. Speech Processing



Transformer

e.g. Natural Language Processing

**Generative
Adversarial Networks**

e.g. Synthetic Face Generation



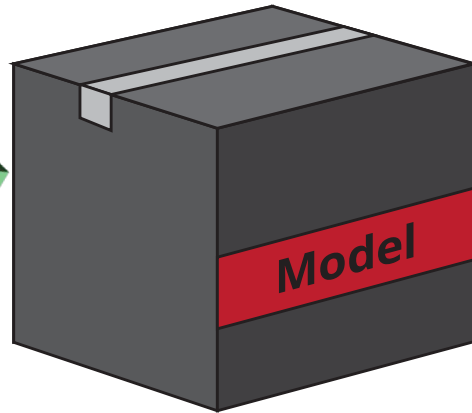
UNIVERSITY
OF TWENTE.

IMPLEMENTING AI ON EMBEDDED SYSTEMS

CHALLENGES



Key Concepts



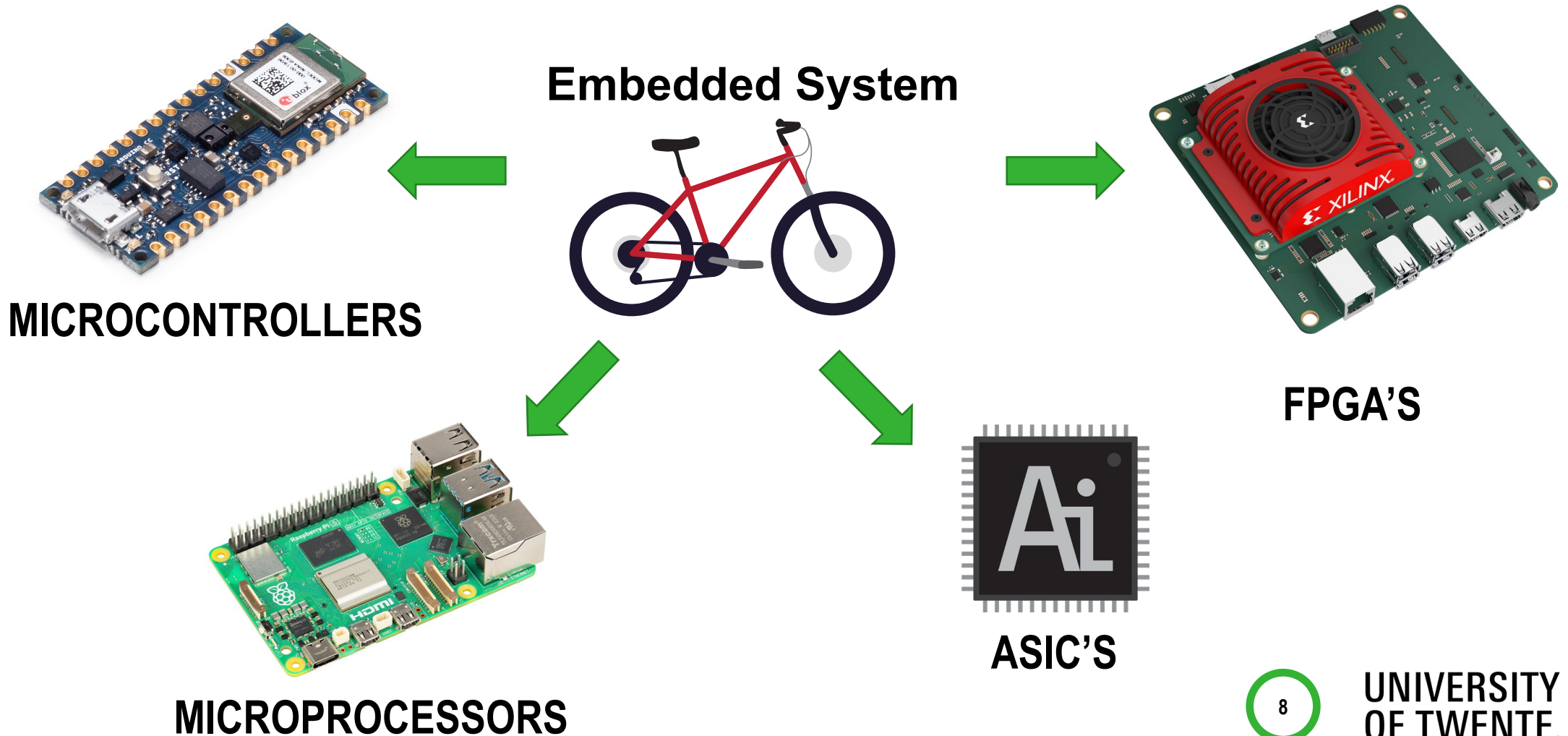
Embedded System



UNIVERSITY
OF TWENTE.

EMBEDDED SYSTEMS

A DEVICE DESIGNED FOR ONE SPECIFIC TASK WITHIN A SYSTEM



Key Concepts



UNIVERSITY
OF TWENTE.

IMPLEMENTING AI ON EMBEDDED SYSTEMS

COMPARE EMBEDDED SYSTEM WITH PC

Embedded System



- Special Purpose Hardware
- Pre-programmed firmware
- Real-time response and efficiency is key
- Little power consumption
- Cheap
- Local data processing



- Generic Hardware
- Programmable by user
- Performance and capacity is key
- High power consumption
- Expensive
- Cloud-based data processing



Key Concepts



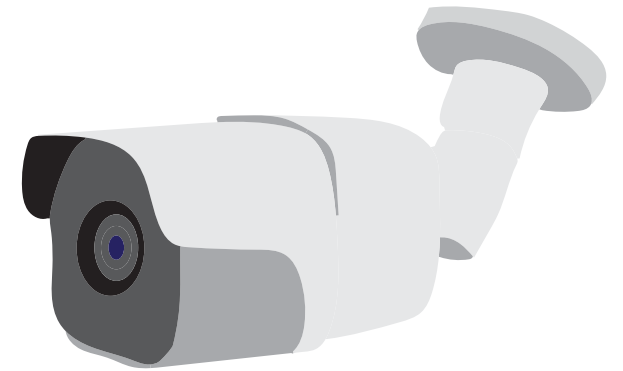
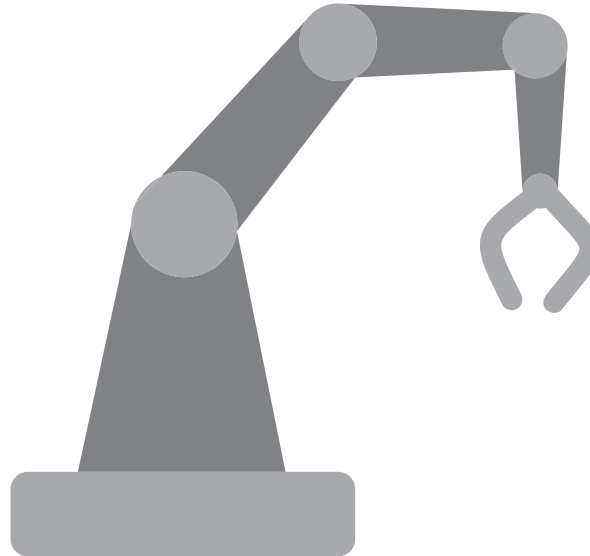
UNIVERSITY
OF TWENTE.

APPLICATIONS



Personal Healthcare

Predictive Maintenance

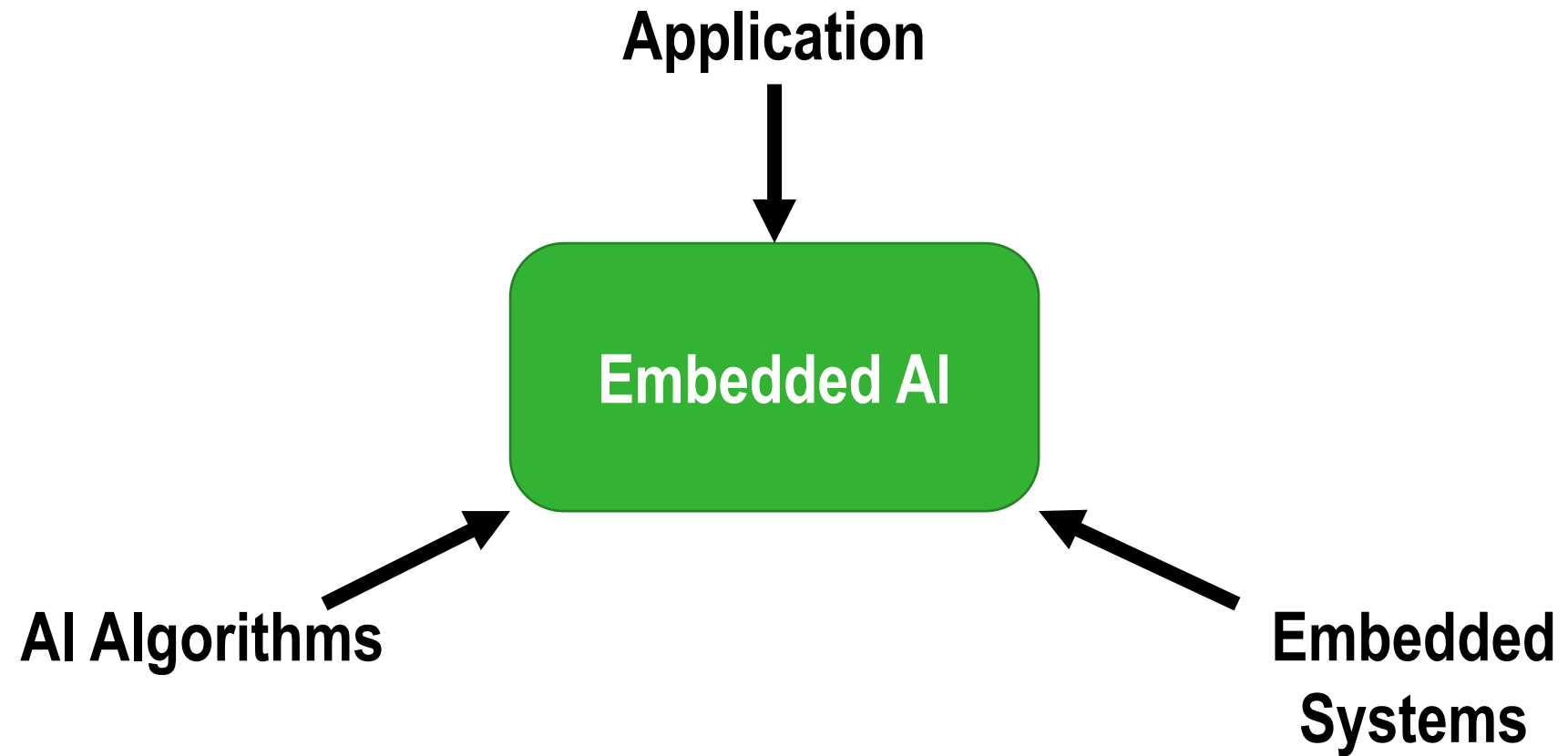


Smart Camera's



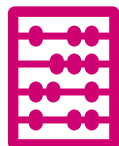


EMBEDDED AI





MY RESEARCH

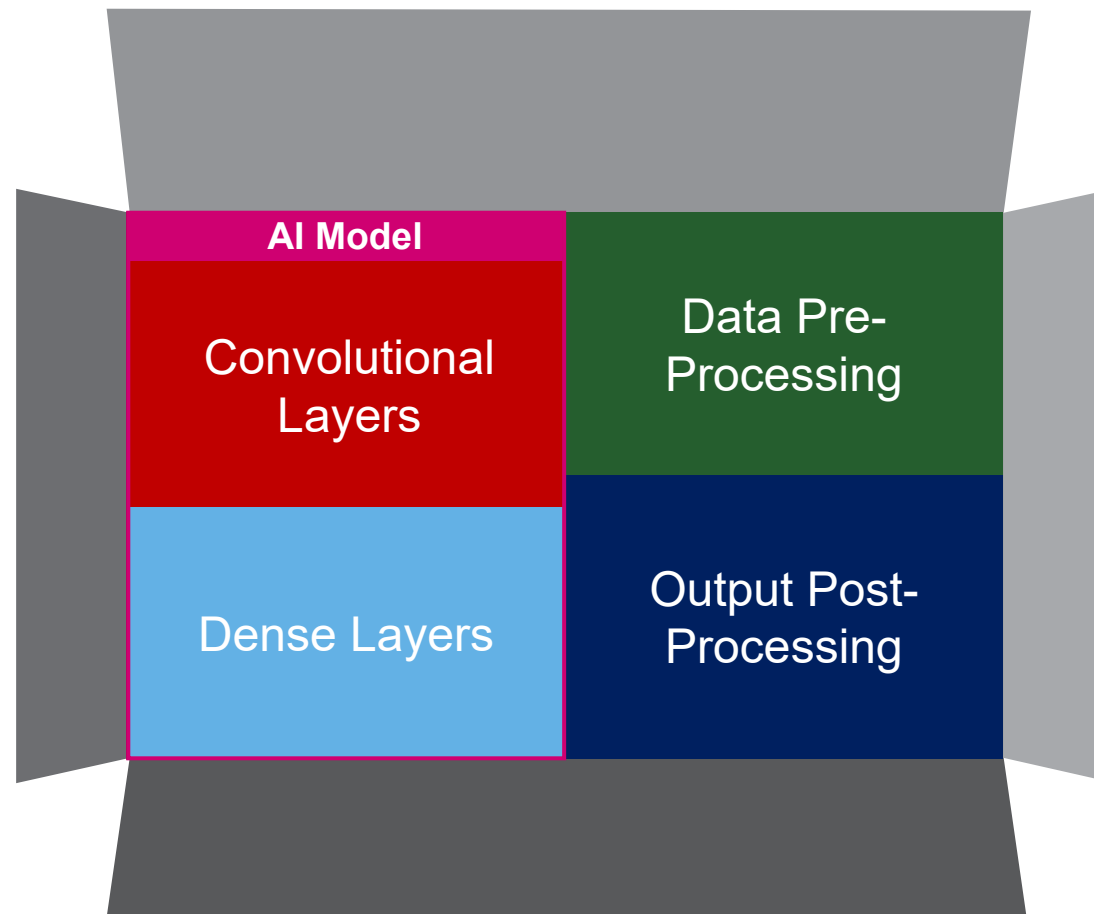


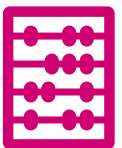
Techniques

Efficient Computation

MODEL DESIGN & PERFORMANCE

- AI model contains several different layers



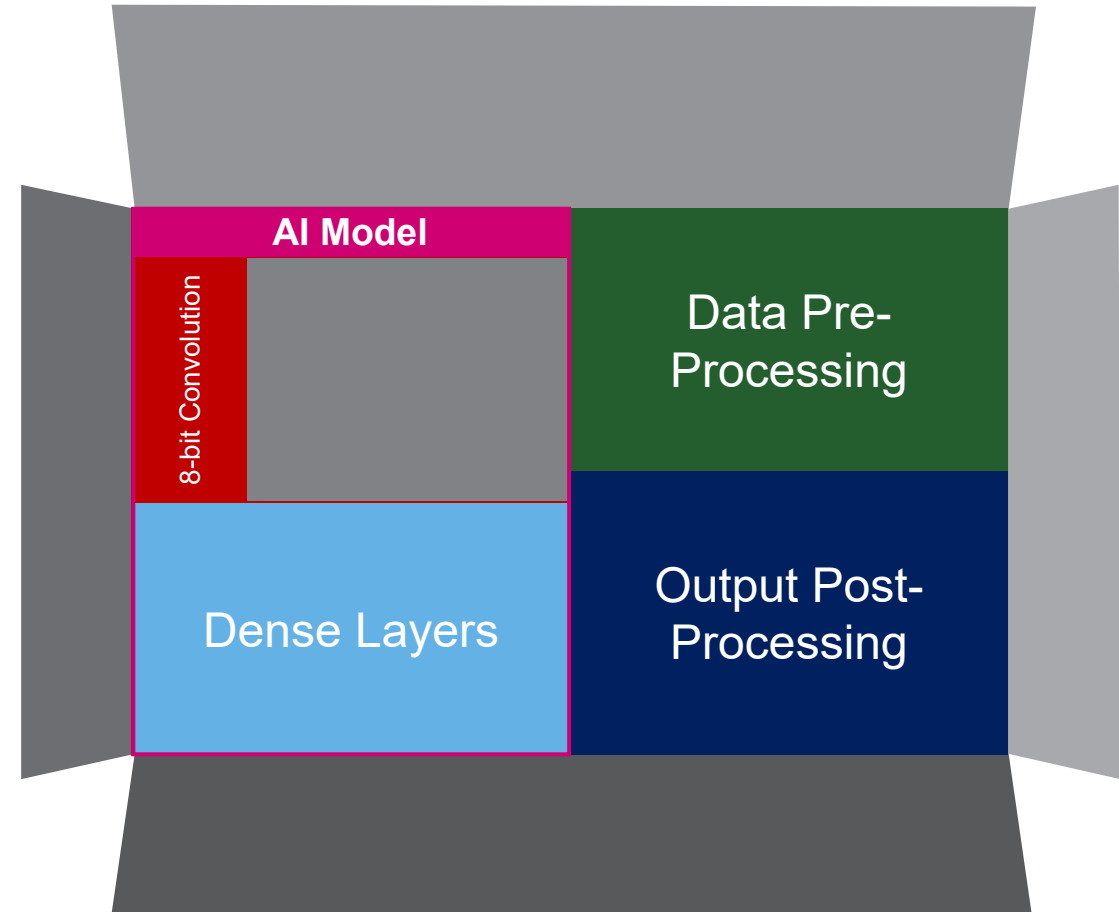


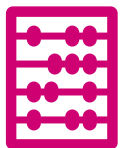
Techniques

Efficient Computation

MODEL DESIGN & PERFORMANCE

- Quantization
 - Approximating a 32-bit number with an 8-bit number
 - Can reduce footprint by a factor 4



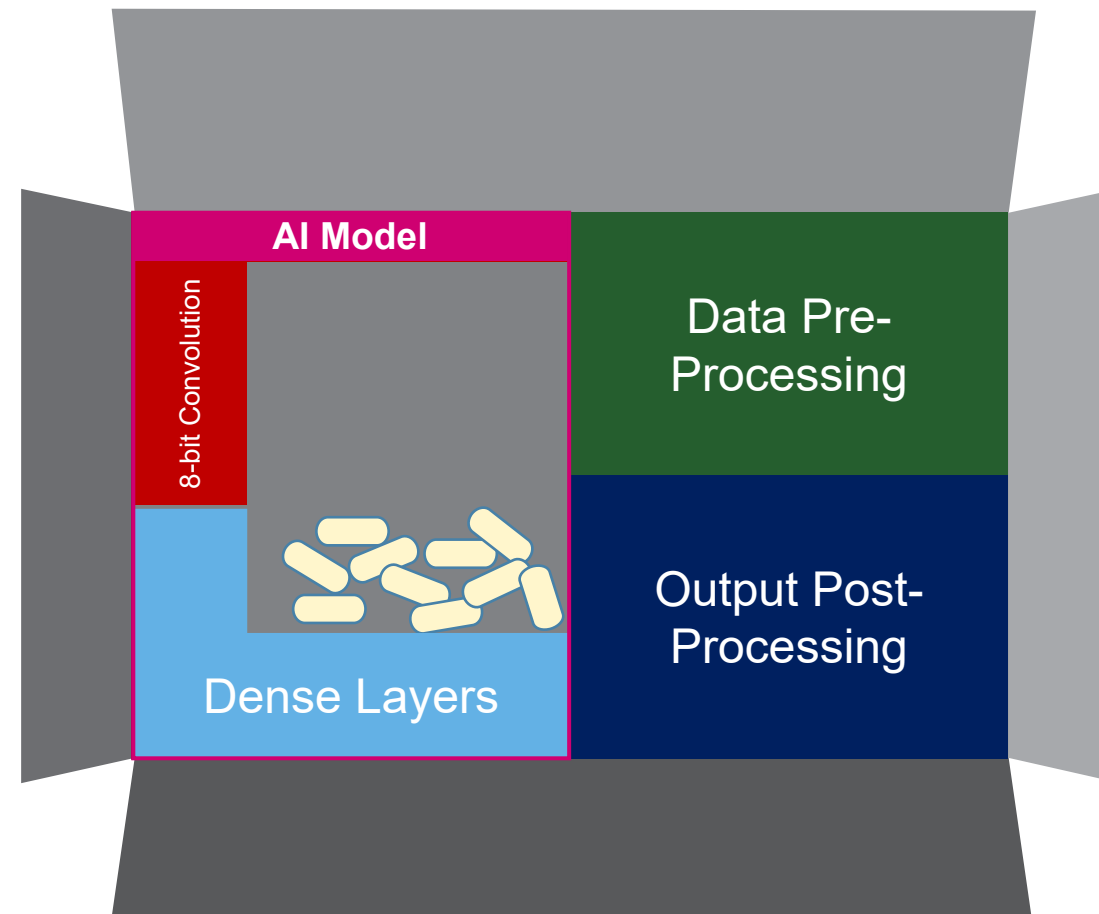


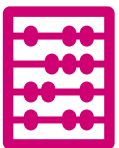
Techniques

Efficient Computation

MODEL DESIGN & PERFORMANCE

- Pruning
 - Removing connections in neural network that do not contribute
 - Like packaging peanuts that fill up space





Techniques

Efficient Computation

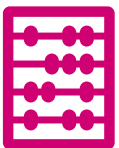
EFFECT OF QUANTIZATION ON FACE RECOGNITION

Face Recognition Dataset	32-bit	8-bit	4-bit	2-bit
Post Training Quantization	98.85%	94.65%	63.15%	51.55%
Quantize Aware Training	98.85%	98.68 ± 1.15%	98.63 ± 0.18%	93.45 ± 0.66%

S. Bunda, L. Spreeuwers and C. Zeinstra, "Sub-byte quantization of Mobile Face Recognition Convolutional Neural Networks," 2022 International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 2022, pp. 1-5, doi: 10.1109/BIOSIG55365.2022.9897025.



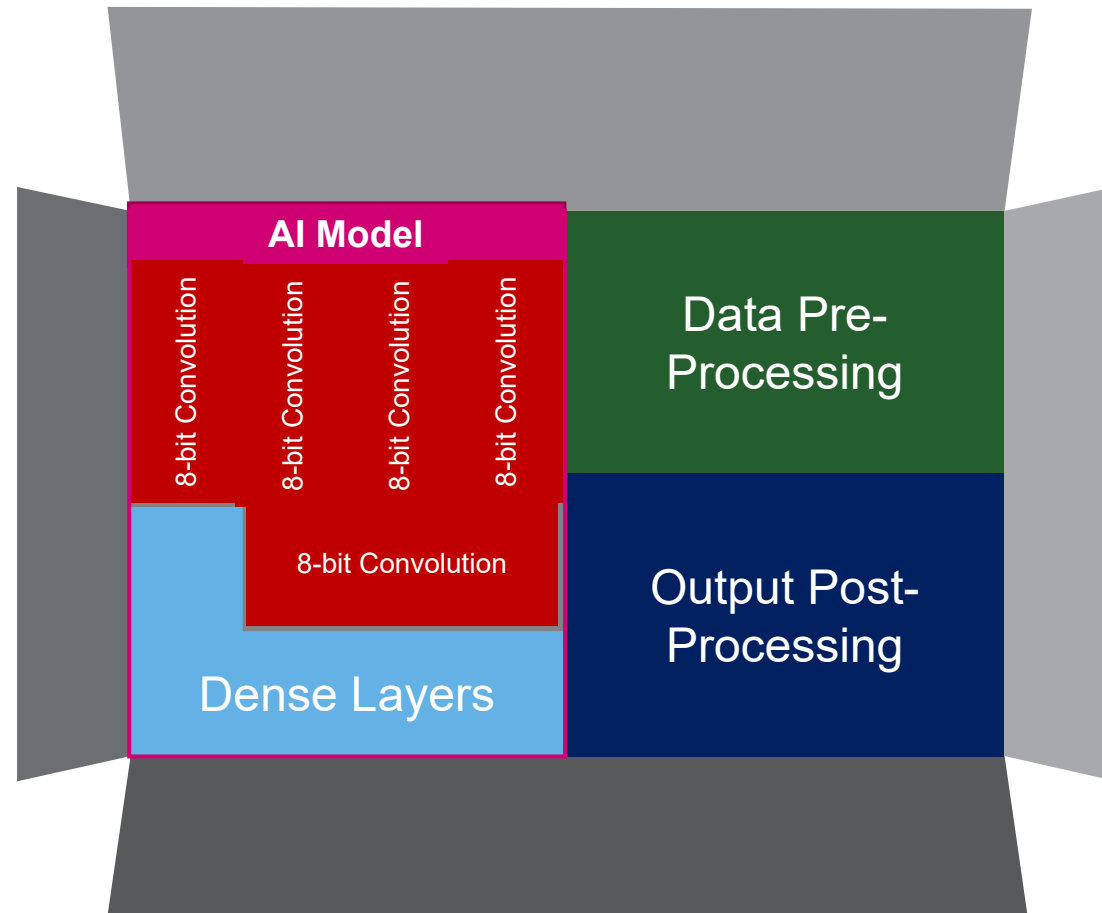
UNIVERSITY
OF TWENTE.



Techniques

Efficient Computation

OPTIMIZING HARDWARE USAGE



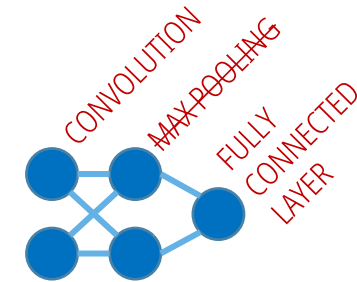
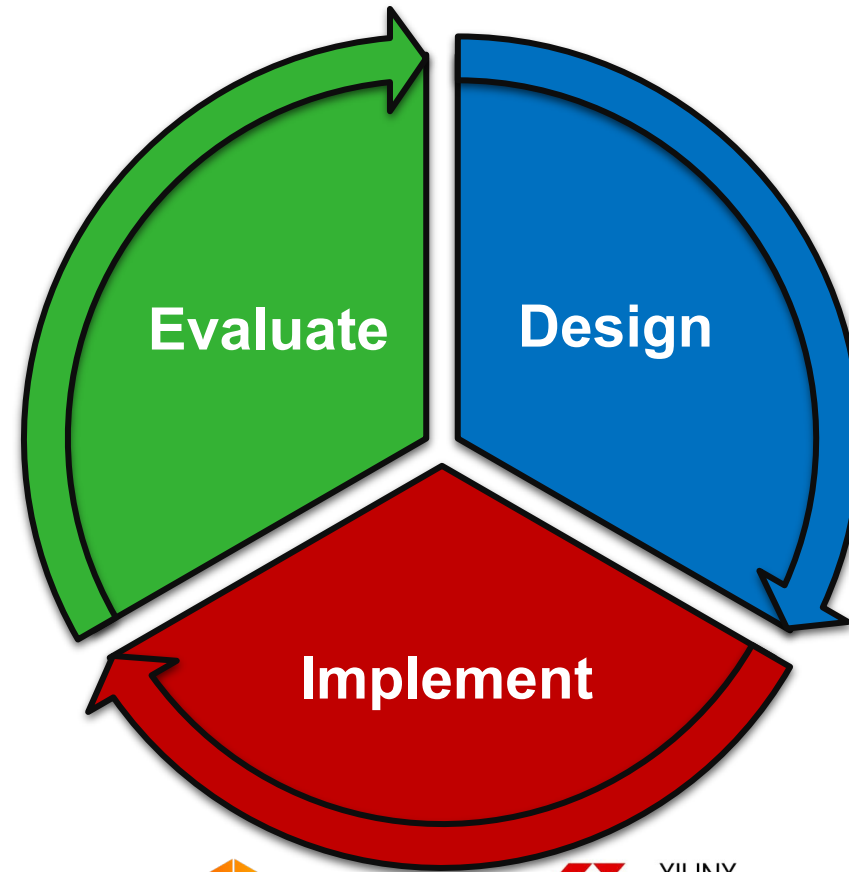
Beware power
consumption
and **latency!**

MY RESEARCH

	Required	Measured
--	----------	----------

Accuracy	> 99 %	98%
----------	--------	-----

Latency	< 100ms	89ms
---------	---------	------



Create/Adapt
Model
Architecture



TensorFlow Lite



XILINX
VITISTM
AI



Our current research
and goals

18

UNIVERSITY
OF TWENTE.



HARDWARE CO-DESIGN NEURAL ARCHITECTURE SEARCH

Optimize design by:

- Proposing models that fit within the **target hardware memory**
- Search for optimal **implementation**
- Evaluate based on e.g. **accuracy, latency and energy**

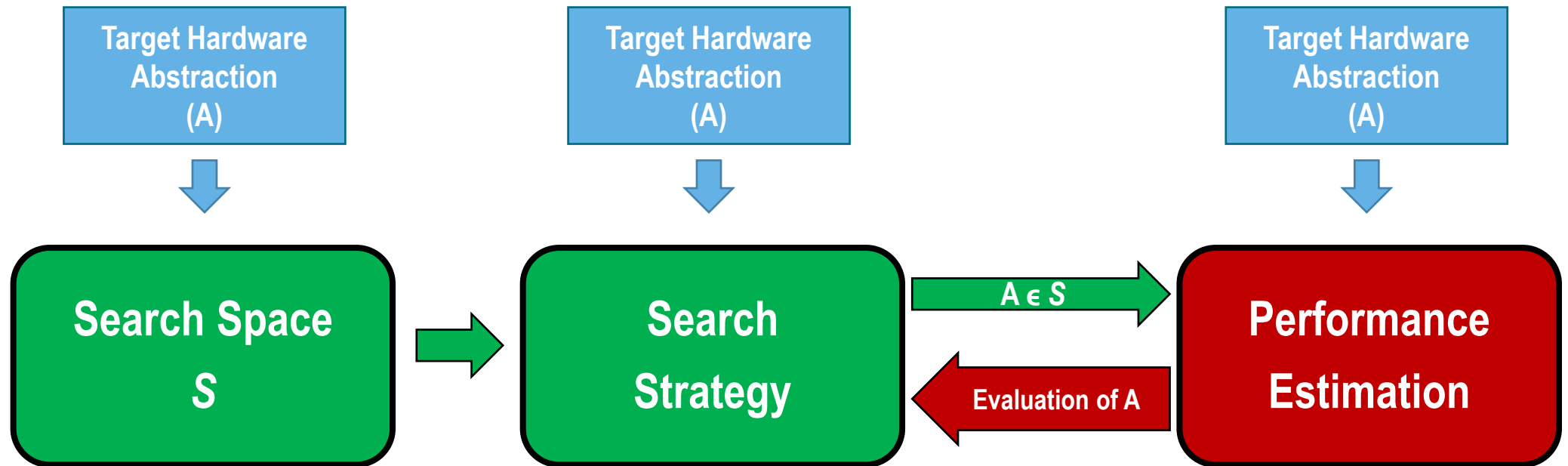


Our current research
and goals



UNIVERSITY
OF TWENTE.

HARDWARE CO-DESIGN NEURAL ARCHITECTURE SEARCH





ACTIVE PROJECTS

SOME ACTIVE PROJECTS

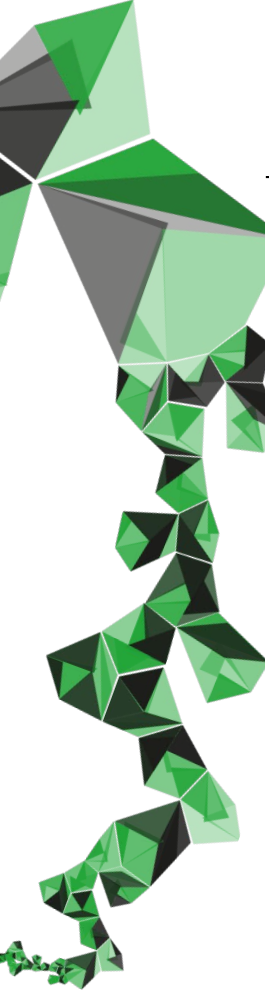
- Image classification optimization using FINN-aware neural architecture search
- Vision-based object distribution detection using YOLO and Raspberry Pi
- Efficient Transformer Networks by researching 8-bit Arithmetics for Transformers
- Hardware acceleration for Genetic Selective Sweep Detection using modern technologies



Our current research
and goals



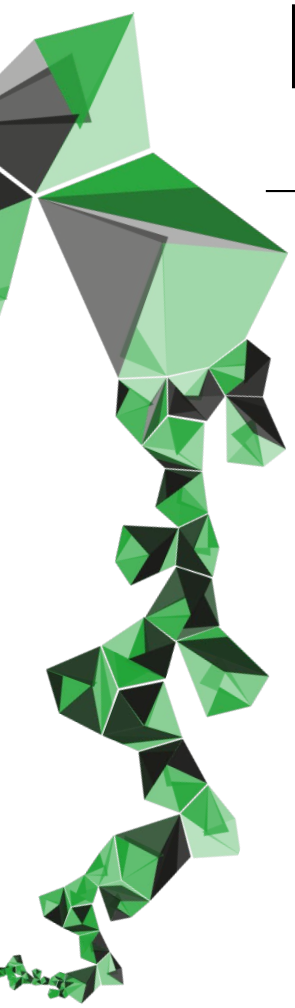
UNIVERSITY
OF TWENTE.



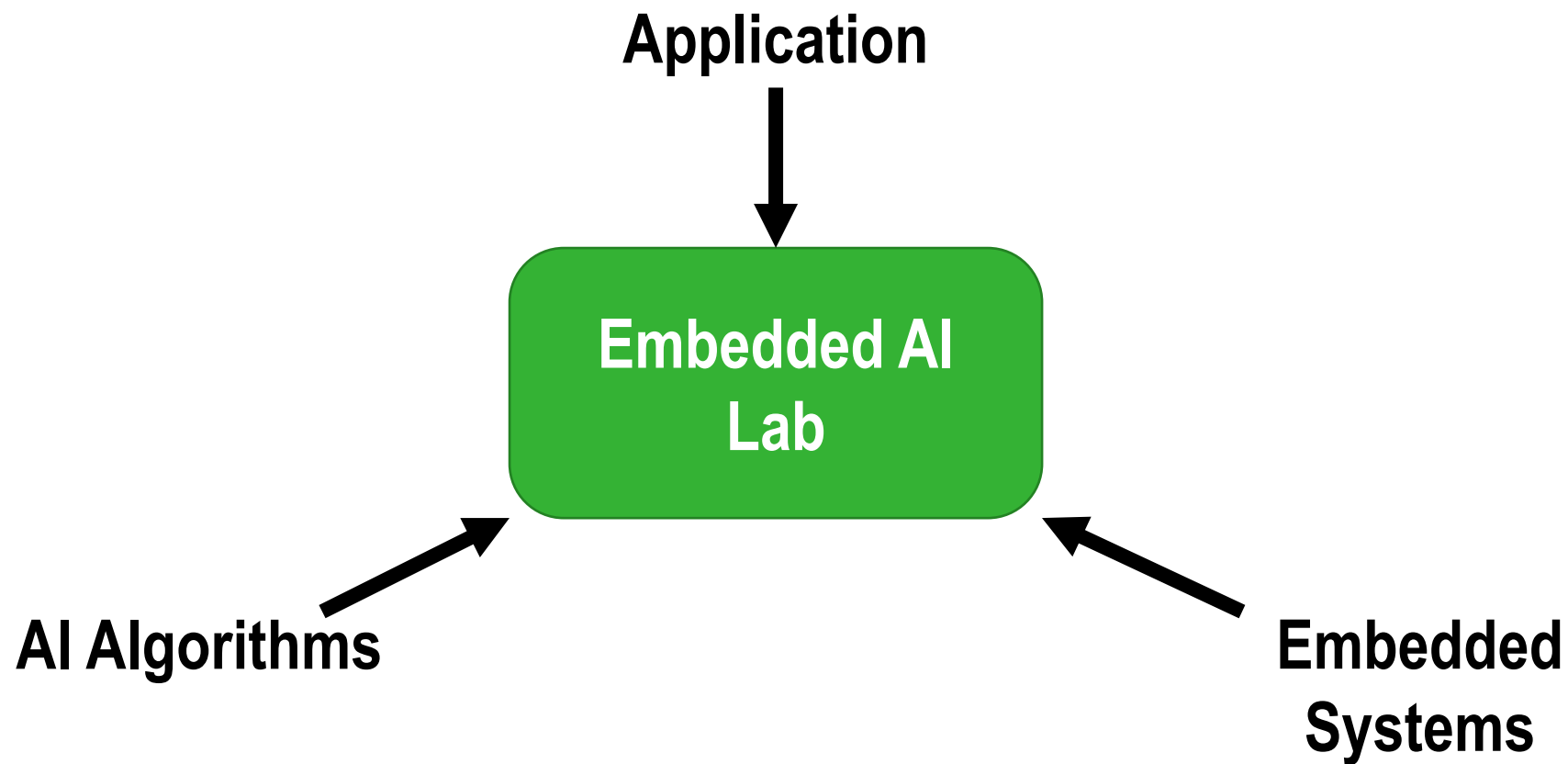
OUTLOOK OF EMBEDDED AI

- Expect new efficient AI-acceleration hardware
 - AMD Ryzen AI NPU
 - Neuromorphic Processors
- Leverage the knowledge of the domain of AI and Embedded Systems for Specific Applications





EMBEDDED AI LAB





OBJECTIVES EMBEDDED AI LAB

- **Objective 1:** Create an Embedded AI Community at UT
 - Creating new contacts and collaborations within the University of Twente
- **Objective 2:** Inspire using DSI Embedded AI Seminars
 - Everyone is welcome to present their work and start discussions
- **Objective 3:** Stimulate student projects through collaboration and teaching
 - Co-supervise students on Embedded AI topics
 - Teaching the basics through Master course

PLANNING

9th of October

Mario Ruiz from AMD on their new Neural Processing Units (NPUs)

December

Talk on Neuromorphic Computing

November

Qing Wang on his research on Embedded AI in TU Delft



EMBEDDED AI LAB

- Simulate **collaboration** on the topic of **Embedded AI** within University of Twente and with industry
- Develop **AI** methods to achieve State-of-the-Art performance with **limited resources**
- Build **Demonstrators** to show applications



SIGN UP FOR NEXT SEMINAR 9TH OF OCTOBER



An abstract graphic featuring a series of thin, grey, wavy lines that flow across the page. Overlaid on these lines are several green, three-dimensional geometric shapes, primarily triangles and polygons, some of which are shaded to create a sense of depth. The shapes are scattered across the upper right and middle sections of the image.

QUESTIONS?

The background features a series of thin, grey, wavy lines that flow across the slide. On the right side, there are several green, low-poly geometric shapes, some of which are dark green and others light green, arranged in a way that suggests a 3D structure or a map of a coastline.

EXTRA SLIDES

ARTIFICIAL INTELLIGENCE

ARTIFICIAL INTELLIGENCE

A system that is able to display human-like capabilities such as reasoning and learning

Natural Language Processing

Robotics

Automatic Programming

MACHINE LEARNING

Algorithms whose performance improves by being exposed to more data

Random Forest

DEEP NEURAL NETWORKS

ML algorithm based on multi-layered neural networks

Decision Trees

Support Vector Machines

Convolutional
Neural Networks

Transformer
Networks

Logistic Regression

Principle Component Analysis

Recurrent Neural Network

k-means Clustering

TENSORFLOW LITE & TINYML

COMPILATION TOOLING

- tensorflow.org/lite/microcontrollers
- tinyurl.com/tinymml-book



EDGE IMPULSE

DEPLOYMENT TOOLING




Overview of Available
Tools

 **EDGE IMPULSE**


 Dashboard

 Devices

 Data acquisition

 Experiments

 Impulse design


 Upload model

 Retrain model

 Live classification

 Model testing

 Deployment

 Versioning

GETTING STARTED

 Documentation

 Forums

Sebastian / m7715157-project-1 PERSONAL

Target: Arduino Nicla Vis...

S

Impulse #1

< Step 2: Process "mobilenet-v1-tflite-0-25-128-q"

Configure model settings for optimal processing.

Model input

Input shape: (128, 128, 3)

Image (RGB)

How is your input
scaled?

Pixels ranging 0..255 (not no

Input should be in RGB format (one
value per pixel). If your model uses a
different channel order, or is scaled
differently, then select "Other".

Model output

Output shape: (1001)

Classification

Output labels (1001)

Enter labels for your
model separated by ','.

class 1, class 2, class 3, class 4, c

Save model

On-device performance

Arduino Nicla Vision (...)



PROCESSING TIME

35 ms.



FLASH USAGE

485,6K

This model won't run on MCUs. Calculated arena size is >6MB

Check model behavior

Upload test data to ensure correct model settings and proper
model processing. (Optional)

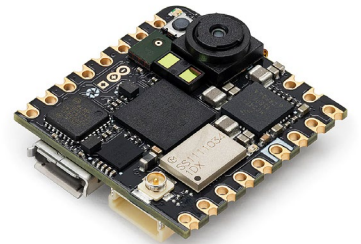
Upload an image

Upload an image to try out your model. The image will be automatically
resized to 128x128 (RGB).

Bestand kiezen

Geen bestand gekozen

Test sample



© 2024 EdgImpulse Inc. All rights reserved

33

UNIVERSITY
OF TWENTE.

MICROPYTHON IN OPENMV

DEPLOYMENT TOOLING



Overview of Available
Tools

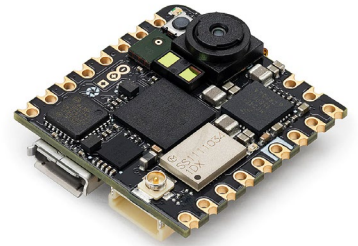
tf_object_detection_1.py - OpenMV IDE

```
File Edit Tools Window Help
helloworld_1_tf.py x face_detection_1.py x face_detection.py x helloworld_1.py x tf_object_detection_1.py x
tf_object_detection_1.py
1 # This work is licensed under the MIT license.
2 # Copyright (c) 2013-2024 OpenMV LLC. All rights reserved.
3 # https://github.com/openmv/openmv/blob/master/LICENSE
4 #
5 # TensorFlow Lite Object Detection Example
6 #
7 # This examples uses the builtin FOMO model to detect faces.
8
9 import sensor
10 import time
11 import ml
12 from ml.utils import NMS
13 import math
14 import image
15
16 sensor.reset() # Reset and initialize the sensor.
17 sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or GRAYSCALE)
18 sensor.set_framesize(sensor.QVGA) # Set frame size to QVGA (320x240)
19 sensor.set_windowing((240, 240)) # Set 240x240 window.
20 sensor.skip_frames(time=2000) # Let the camera adjust.
21
22 min_confidence = 0.4
23 threshold_list = [(math.ceil(min_confidence * 255), 255)]
24
25 # Load built-in FOMO face detection model
26 model = ml.Model("fomo_face_detection")
27 print(model)
28
29 # Alternatively, models can be loaded from the filesystem storage.
30 # model = ml.Model('object_detection_modelwork>.tflite', load_to_fb=True)
31 # labels = [line.rstrip('\n') for line in open("Labels.txt")]
32
33 colors = [ # Add more colors if you are detecting more than 7 types of classes at once.
34     (255, 0, 0),
35     (0, 255, 0),
36     (255, 255, 0),
37     (0, 0, 255),
38     (255, 0, 255),
39     (0, 255, 255),
40     (255, 255, 255),
41 ]
42
43
44 # FOMO outputs an image per class where each pixel in the image is the centroid of the train
45 # object. So, we will get those output images and then run find_blobs() on them to extract th
46 # centroids. We will also run get_stats() on the detected blobs to determine their score.
47 # The Non-Max-Suppression (NMS) object then filters out overlapping detections and maps their
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Serial Terminal

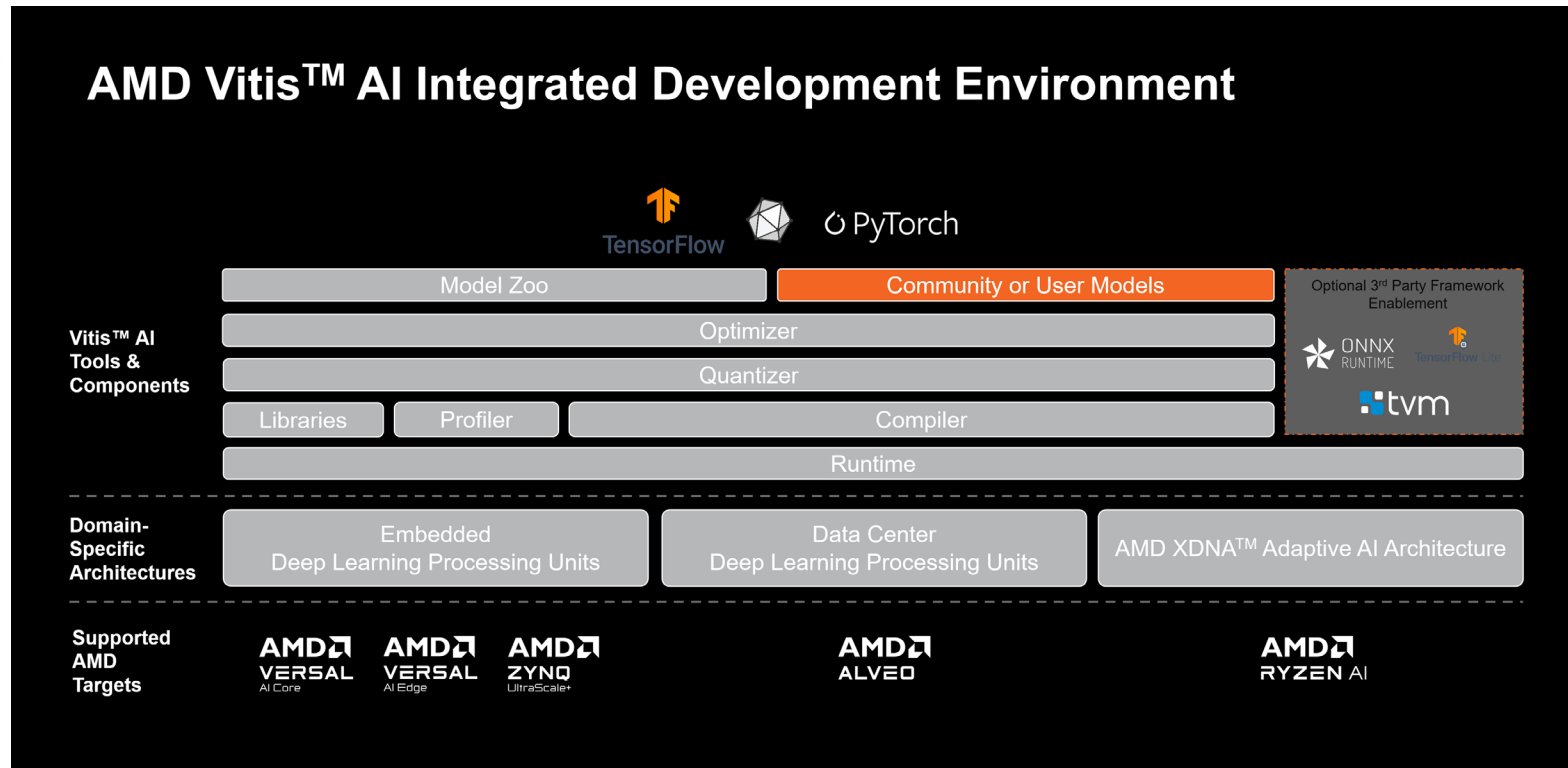
```
x 120 y 110 score 0.71235
13.5547 fps
***** face *****
x 120 y 110 score 0.649726
13.5547 fps
***** face *****
x 130 y 110 score 0.874088
13.5547 fps
13.5546 fps
13.5547 fps
```

Board: Arduino Nicla Vision Sensor: GC2145 Firmware Version: 4.5.9 - [latest] Serial Port: COM4 Drive: E:/ FPS: 13.7



VITIS AI & FINN

DEPLOYMENT TOOLING



<https://xilinx.github.io/Vitis-AI/3.5/html/index.html>
<https://xilinx.github.io/finn/>



Overview of Available
Tools

STM32CUBE.AI

- Optimize and deploy Deep Neural Networks on STM32 microcontrollers:
 - <https://stm32ai.st.com/stm32-cube-ai/>

