

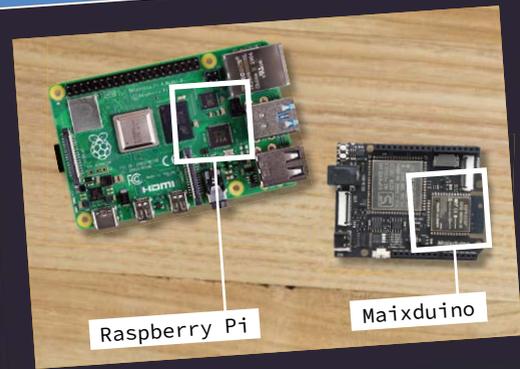
Machine Learning mit Python für PC, Raspberry Pi und Maixduino



```

t sensor
t image
t lcd
t KPU as kpu

init()
or.reset()
or.set_pixformat(sensor.RGB565)
or.set_framesize(sensor.QVGA)
or.run(1)
k = kpu.load(0x300000)
hor = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437)
kpu.init_yolo2(task, 0.5, 0.3, 5, anchor)
le(True):
img = sensor.snapshot()
code = kpu.run_yolo2(task, img)
if code:
for i in code:
print(i)
a = img.draw_rectangle(i.rect())
a = lcd.display(img)
= kpu.deinit(task)
    
```



Raspberry Pi

Maixduino



```

w":135, "h":181, "value":0.949942, "classid":0, "index":0, "objnum":1}
w":135, "h":181, "value":0.938012, "classid":0, "index":0, "objnum":1}
w":135, "h":181, "value":0.923467, "classid":0, "index":0, "objnum":1}
w":135, "h":181, "value":0.923467, "classid":0, "index":0, "objnum":1}
w":135, "h":181, "value":0.938012, "classid":0, "index":0, "objnum":1}
w":135, "h":181, "value":0.938012, "classid":0, "index":0, "objnum":1}
w":135, "h":181, "value":0.938012, "classid":0, "index":0, "objnum":1}
w":135, "h":181, "value":0.923467, "classid":0, "index":0, "objnum":1}
    
```

Günter Spanner

Machine Learning mit Python für PC, Raspberry Pi und Maixduino



Dr. Günter Spinner

● © 2021: Elektor Verlag GmbH, Aachen.

1. Auflage 2021

● Alle Rechte vorbehalten.

Die in diesem Buch veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen und Illustrationen sind urheberrechtlich geschützt. Ihre auch auszugsweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet.

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Die in diesem Buch erwähnten Soft- und Hardwarebezeichnungen können auch dann eingetragene Warenzeichen sein, wenn darauf nicht besonders hingewiesen wird. Sie gehören dem jeweiligen Warenzeicheninhaber und unterliegen gesetzlichen Bestimmungen.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autor können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für die Mitteilung eventueller Fehler sind Verlag und Autor dankbar.

● Erklärung

Autor, Übersetzer und Herausgeber haben sich nach besten Kräften bemüht, die Richtigkeit der in diesem Buch enthaltenen Informationen sicherzustellen. Sie übernehmen keine Haftung für Verluste oder Schäden, die durch Fehler oder Auslassungen in diesem Buch verursacht werden, unabhängig davon, ob diese Fehler oder Auslassungen auf Fahrlässigkeit, ein Versehen oder eine andere Ursache zurückzuführen sind und lehnen hiermit jegliche Haftung gegenüber Dritten ab.

Umschlaggestaltung: Elektor, Aachen

Satz und Aufmachung: D-Vision, Julian van den Berg | Oss (NL)

Druck: Ipskamp Printing, Enschede, Niederlande

● **ISBN 978-3-89576-475-2** Print
ISBN 978-3-89576-476-9 eBook

Elektor-Verlag GmbH, Aachen

www.elektor.de

Elektor ist Teil der Unternehmensgruppe Elektor International Media (EIM), der weltweit wichtigsten Quelle für technische Informationen und Elektronik-Produkte für Ingenieure und Elektronik-Entwickler und für Firmen, die diese Fachleute beschäftigen. Das internationale Team von Elektor entwickelt Tag für Tag hochwertige Inhalte für Entwickler und DIY-Elektroniker, die über verschiedene Medien (Magazine, Videos, digitale Medien sowie Social Media) in zahlreichen Sprachen verbreitet werden. www.elektor.de

Warnhinweise	9
Programmdownload	10
Kapitel 1 • Einführung	11
1.1 In Drei Stufen zur "Superintelligenz"?	12
1.2 Wie Maschinen lernen können	13
Kapitel 2 • Eine kleine Geschichte der KI	16
Kapitel 3 • Lernen aus großen Datenmengen	19
3.1 Maschinelles Lernen und Künstliche Intelligenz	19
Kapitel 4 • Hardwarebasis	23
Kapitel 5 • Der PC als universelle KI-Maschine	24
5.1 Der Computer als Programmierzentrale	24
Kapitel 6 • Raspberry Pi	26
6.1 Remote Desktop	27
6.2 Smartphones und Tablets als Bildschirme	29
6.3 FileZilla	30
6.4 Pimp my Pi	31
Kapitel 7 • Sipeed Maix: Der "Maixduino"	33
7.1 Klein aber fein: Die Leistungsmerkmale des "Maixduino"	33
7.2 Anwendungsbereiche	36
7.3 Inbetriebnahme und Funktionstest	38
7.4 Stromversorgung und stand-alone Betrieb	39
Kapitel 8 • Programmier- und Entwicklungs-umgebungen	41
8.1 Thonny - eine Python IDE für Ein- und Aufsteiger	41
8.2 Universalgenie: Thonny für RasPi und Maixduino	44
8.3 Umgang mit Dateien	46
8.4 Thonny auf dem Raspberry Pi	48
8.5 Tipps zur Fehlerbehebung in der Thonny IDE	51
8.6 Die MaixPy IDE	53
8.7 MicroPython-Interpreter auf dem Maixduino	56
8.8 Flash-Tool im Einsatz	58
8.9 Machine Learning und Interaktives Python	58
8.10 Anaconda	60
8.11 Jupyter	62
8.12 Installation und Start	63

8.13 Jupyter mit MicroPython-Kernel	66
8.14 Kommunikationsaufbau zum Maixduino	67
8.15 Des Pudels Kern: Kernels	68
8.16 Arbeiten mit Notizbüchern	69
8.17 Alle Libraries an Board?	70
8.18 Python mit Spyder	71
8.19 Wer programmiert wen?	73
Kapitel 9 • Python: Ein Kompendium	74
9.1 Kommentare erleichtern das Leben	76
9.2 Die print()-Anweisung	78
9.3 Ausgaben auf das Display	79
9.4 Einrückungen und Blöcke	80
9.5 Zeitsteuerung und sleep	81
9.6 Die Hardware im Griff: Digitale Ein- und Ausgänge	83
9.7 Für wichtige Werte: Variablen und Konstanten	85
9.8 Zahlen und Variablentypen	86
9.9 Konvertieren von Zahlentypen	87
9.10 Arrays als Basis Neuronaler Netze	88
9.11 Operatoren	89
9.12 Bedingungen, Verzweigungen und Schleifen	91
9.13 Versuch und Irrtum: try und except	92
Kapitel 10 • Unentbehrliche Helfer: Libraries	94
10.1 Matplotlib als Grafikkünstler	95
10.2 Das Rechengenie: NumPy	100
10.3 Die Datenkrake: pandas	103
10.4 Lernen und Visualisieren: SciKit, SciPy, SciKit-image und Co.	106
10.5 Maschinen lernen Sehen - mit OpenCV	107
10.6 Intelligenzbestien: KERAS und TensorFlow	113
10.7 Wissenstransfer: Übertragung von Lernleistungen	117
10.8 Grafische Darstellung der Netzstruktur	118
10.9 Lösung des XOR-Problems in KERAS	119
10.10 Virtuelle Umgebungen	120
Kapitel 11 • Machine Learning in der Praxis	123
11.1 Transferfunktionen und vielschichtige Netze	123

11.2 Blüten und Daten.	124
11.3 Grafische Darstellungen von Datensätzen.	126
11.4 Ein Netz für Iris-Blüten.	128
11.5 Zwei Paar Stiefel: Trainieren und Testen.	130
11.6 Welche Blüte ist das?	133
11.7 Test und Lernverhalten.	134
Kapitel 12 • Erkennung von handschriftlichen Zahlen.	137
12.1 "Hello ML" - MNIST-Datensatz	138
12.2 Ein Neuronales Netzes liest Ziffern.	140
12.3 Training, Tests und Prognosen.	141
12.4 Erweiterung auf Online-Video	143
12.5 KERAS kann es noch besser!	145
12.6 "Gefaltete" Netzwerke	146
12.7 Power-Training	151
12.8 Niemals ohne Qualitätskontrolle!	152
12.9 Livebilder erkennen	153
12.10 Chargengrößen und Epochen.	156
12.11 Auch der Maixduino liest Ziffern.	157
Kapitel 13 • Maschinen lernen sehen: Objekterkennung.	161
13.1 TensorFlow für den Raspberry	161
13.2 Virtuelle Umgebungen	163
13.3 Ein universelles TFLite-Modell im Einsatz	164
13.4 Ideal für "Messies": Klamotten sortieren.	167
13.5 Aufbau und Training des Modells	170
13.6 Maixduino erkennt 20 Objekte.	173
13.7 Gegenstände erkennen, zählen und sortieren	176
Kapitel 14 • Maschinen lernen hören und sprechen	180
14.1 Sprich mit mir!	180
14.2 RasPi lernt sprechen.	182
14.3 Messgeräte mit Sprachausgabe	185
14.4 Ich habe Sie (nicht) verstanden...	188
14.5 RasPi als "ChatBot"	193
14.6 "PlauderBots"	196
14.7 Das "sprechende Auge"	197

14.8 Eine "KI-Fledermaus"	199
Kapitel 15 • Gesichtserkennung und -identifizierung	201
15.1 Das Recht am eigenen Bild	203
15.2 Maschinen erkennen Menschen und Gesichter	204
15.3 Maixduino als Türspion	208
15.4 Wie viele Personen waren auf der Party?	210
15.5 Personalarms	212
15.6 Sozialer Sprengstoff? - Gesichtsidifizierung	213
15.7 "Big Brother" RasPi: Gesichtsidifizierung in der Praxis	214
15.8 Bitte recht freundlich ;-).	216
15.9 Foto-Training	223
15.10 Erkenne dich selbst! (und andere...).	225
15.11 Ein Biometricscanner als Türöffner	225
15.12 Geschlecht und Alter erkennen	227
Kapitel 16 • Trainieren eigener Modelle.	230
16.1 Erstellung eines Modells für den Maixduino	230
16.2 Maixduino erkennt Elektronik-Komponenten	233
16.3 Performance des trainierten Netzes	236
16.4 Praxistest	237
16.5 Ausblick: Multi-Objekt-Detektoren	238
Kapitel 17 • Zukunftsmusik: Von der KPU zum Neuromorphen Chip.	241
Kapitel 18 • Elektronische Bauelemente	247
18.1 Breadboards	248
18.2 Drahtbrücken und Jumper-Kabel	249
18.3 Widerstände	250
18.4 Leuchtdioden (LEDs)	251
18.5 Transistoren	252
18.6 Sensoren	253
18.7 Ultraschall-Entfernungsmesser	254
Kapitel 19 • Fehlersuche	255
Kapitel 20 • Bezugsquellen	256
Kapitel 21 • Literatur	257
Kapitel 22 • Abbildungsverzeichnis	258
Stichwortverzeichnis	262