## TECHNISCHE UNIVERSITÄT DRESDEN

# Fakultät für Elektrotechnik und Informationstechnik

### Institut für Biomedizinische Technik

# Versuchsanleitung Praktikum Mikrorechentechnik II Versuch GMM-3

### Bildaufnahme und -verarbeitung mit OpenCV und Python

Stand: 03.05.2024

Ansprechpartner:

Matthieu Scherpf, Dipl.-Ing. Institut für Biomedizinische Technik Fakultät Elektrotechnik und Informationstechnik TU Dresden Fetscherstraße 29, 01307 Dresden Email: Matthieu.Scherpf@tu-dresden.de Tel.: +49 (0)351-463 32118

### Inhaltsverzeichnis

1	Ein	ıführung	1
	1.1	Ziel des Versuchs	1
	1.2	Methodik	1
	1.3	Coronabedingte Onlinedurchführung	1
		1.3.1 Installation von Python	1
		1.3.2 Installation des Pakets $mrt2pkgGMM3$	2
		1.3.3 Installation eines Editors	2
	1.4	Umgang mit dem Terminal	2
<b>2</b>	Inh	alt	3
	2.1	Exkurs in die digitale Bildverarbeitung	3
	2.2	Einführung in OpenCV mit Python	3
		2.2.1 Auslesen der Kamera mit OpenCV	4
		2.2.2 Hardwareinteraktion	4
		2.2.3 Wie "sieht" der Computer ein Bild?	4
		2.2.4 Bildverarbeitung mit OpenCV	5
		2.2.5 Nutzung einer graphischen Oberfläche	5
	2.3	Hardware	5
3	Gut	te Praxis bei der Programmierung	5
4	Vor	rbereitungsaufgaben	<b>5</b>
<b>5</b>	Ver	rsuchsdurchführung	6
	5.1	Aufgabenstellung	6
	5.2	Hinweise zur Durchführung	6
6	Lite	eratur	7
7	Arb	beits- und Brandschutzhinweise	7
•			•
8	Que	ellcode	8

### 1 Einführung

#### 1.1 Ziel des Versuchs

In diesem Versuch sollen Methoden der digitalen Bildverarbeitung mit Python demonstriert werden. Das Ziel des Versuches ist es, Bilder von einer Webcam einzulesen, durch eigene Bildverarbeitungsalgorithmen zu verändern und auf dem Bildschirm darzustellen. Dabei wird mit der Verwendung externer Bibliotheken (OpenCV-Python) der Funktionsumfang von Python erweitert. Am Beispiel der Bildverarbeitung wird gezeigt, wie man komplexe Aufgaben durch Abstraktion mit wenig und gut lesbarem Quellcode bewältigen kann.

#### 1.2 Methodik

Zur Programmierung während des Versuches wird Python verwendet. Sie werden die entsprechenden Anweisungen in einzelnen Befehlssequenzen umsetzen. Dabei werden Sie Ihre Befehlssequenzen in eine bereits existierende Main-Funktion eintragen. Diese übernimmt bereits die Bildübertragung von der Kamera in den Rechner. Ihre Befehle werden nach und nach die Main-Funktion ergänzen. Nach jeder Versuchsaufgabe erfolgt eine neue Testphase. Im Kolloquium werden auch die Hardwarekomponenten des Versuches und ihre Funktionen besprochen.

### 1.3 Coronabedingte Onlinedurchführung

Sollte der Praktikumsversuch sowie das Kolloquium nicht in Präsenz stattfinden können, dann ist der Versuch am eigenen Rechner zu bearbeiten. Hierfür wird davon ausgegangen, dass in der Praktikumsgruppe mindestens ein Rechner mit einer Webcam ausgestattet ist und entsprechend für die Bearbeitung der Aufgabe genutzt werden kann. Für die Bewertung ist der finale Quellcode einzureichen. Die Benotung findet dann auf Basis des digital durchgeführten Kolloquiums sowie der Qualität des eingereichten Quellcodes statt. Haben Sie diesbezüglich Fragen, können Sie sich per E-Mail (Matthieu.Scherpf@tu-dresden.de) an den Praktikumsbetreuer wenden.

Im folgenden finden Sie Informationen zur Installation von Python sowie des Quellcode-Grundgerüsts für die Versuchsdurchführung.



#### 1.3.1 Installation von Python

Abbildung 1: Bei der Installation muss das Häckchen gesetzt werden, um Python über das Terminal starten zu können.

Laden Sie sich die folgende Windows installer Datei für Python 3 herunter, welche Sie unter benanntem Link (https://www.python.org/ftp/python/3.8.6/python-3.8.6-amd64.exe) finden können. Führen Sie anschließend den heruntergeladenen installer aus um Python 3 für Windows zu installieren (nutzen

Sie für alle Optionen die Standardvorgaben des installers - **zusätzlich ist das Kästchen "Add Python 3.8 to PATH" anzukreuzen, siehe Abbildung 1**). Testen Sie, ob die Installation erfolgreich war, indem Sie ein Terminal öffnen (unter Windows *cmd* oder *Eingabeaufforderung* genannt) und dort *python* eingeben. Startet jetzt der Python-Interpreter (siehe Abbildung 2), dann hat alles funktioniert. Weiterhin können Sie unter [4] Hilfe für die Installation zu finden.

```
Eingebeaufforderung-python × + ~ - □
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\Matthieu>python
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

Abbildung 2: Terminal nach Eingabe des python-Befehls.

Sollten Sie ein anderes Betriebssystem verwenden (macOS, Linuxdistribution), dann sind die Schritte prinzipiell die gleichen. Es muss natürlich der installer für das entsprechende Betriebssystem heruntergeladen werden, anstatt des oben angegebenen.

Bevor Sie den Praktikumsbetreuer um Hilfe bitten, sollten Sie zuerst versuchen im Internet mögliche Lösungen zu finden bzw. Ihre Gruppenmitglieder um Hilfe zu bitten. In der Regel lassen sich auf diese Weise die meisten Probleme schnell beheben.

#### 1.3.2 Installation des Pakets mrt2pkgGMM3

Nachdem Python erfolgreich installiert wurde, ist nun noch der Quellcode erforderlich, auf dem dieser Praktikumsversuch aufbauen soll. **Dieser kann als zip-Archiv unter benanntem Link (https:** //cloudstore.zih.tu-dresden.de/index.php/s/N4GCQ6pbdLCMP6c) heruntergeladen werden. Das Archiv muss vor der Installation entpackt werden. Öffnen Sie anschließend ein neues Terminal und navigieren Sie mittels folgendem Befehl zum eben entpackten Ordner:

cd /path/to/folder

Dann nutzen Sie den vorherig automatisch mitinstallierten python-package-manager (kurz pip) um alle notwendigen Bibliotheken zu installieren:

python -m pip install -r requirements.txt

Hierzu zählt unter anderem *opencv-python*, welche zum Auslesen der Webcam sowie der weiteren Verarbeitung der Bilddaten benötigt wird.

#### 1.3.3 Installation eines Editors

Um nun Pythoncode schreiben zu können, sollte noch ein entsprechender Editor heruntergeladen werden. Es wäre theoretisch auch möglich, hierfür einfach den Windows standard Texteditor zu verwenden, allerdings gilt es sprachspezifische Regeln einzuhalten, wofür ein Editor entsprechende Hilfen liefert (bspw. die korrekte Einrückung von Codezeilen). Als Editor empfiehlt sich *Visual Studio Code* (https://code.visualstudio.com/). Die Installation und Nutzung dieses Editors ist im Internet gut beschrieben und wird hier nicht weiter erläutert (siehe [6]).

#### 1.4 Umgang mit dem Terminal

Der einfachste Weg Pythoncode auszuführen ist über das Terminal (auch Kommandozeile oder Eingabeaufforderung genannt). Um ein Pythonskript, also eine Datei mit der Endung *.py* auszuführen, starten Sie entweder ein Terminal beim Ordner, wo auch die auszuführende Datei liegt oder starten Sie ein Terminal und navigieren Sie dann mit dem Befehl aus 1.3.2 zum Zielordner. Anschließend können Sie das Pythonskript mit folgendem Befehl starten. Hilfe für die Nutzung des Terminals unter Windows können Sie unter [5] finden.

```
python skriptname.py
```

Eine eigenständige Kompilierung ist in Python nicht notwendig, da der Programmcode zur Laufzeit kompiliert wird. Das bedeutet, dass der Programmcode direkt zum Start automatisch in den entsprechenden Maschinencode übersetzt wird.

#### 2 Inhalt

#### 2.1 Exkurs in die digitale Bildverarbeitung

Die Bildverarbeitung ist ein Spezialgebiet der digitalen Signalverarbeitung. Sie beschäftigt sich mit der Aufnahme, Diskretisierung, Speicherung und Verarbeitung der Bilder. Digitale Bildverarbeitung ist allgegenwärtig, in der Industrieautomatisierung und Prozesssteuerung, ebenso wie in Verbrauchergeräten (Digitalkameras, Smartphones, Webcams). Digitale Bilder können auf vielerlei Art erzeugt werden. Im Folgenden wird als Bild stets die Abbildung von sichtbarem Licht gemeint. Digitale Bilder können jedoch auch aus anderen physikalischen Größen oder Benutzereingaben errechnet werden (Ultraschall, Magnetresonanztomographie, Radar o.ä.).

In Abbildung 3 ist der Ablauf der Aufnahme eines digitalen Videos, also einer Bildfolge, schematisch dargestellt. Die Einzelelemente des Bildsensors - die Pixel - quantisieren einfallendes Licht. Dadurch kann die Bildinformation eines Objektes diskretisiert werden. Die diskreten Werte können dann digital gespeichert werden.



Abbildung 3: Ablauf der Aufnahme eines digitalen Bildes.

Im Falle eines graustufen Bildes werden Helligkeitswerte gespeichert, die bei der Wiedergabe auf einem Bildschirm der Helligkeit der einzelnen Bildpunkte entsprechen. Bei einem Farbbild wird das Licht gefiltert, bevor es den Sensor erreicht. So kann für jeden Bildpunkt das Mischungsverhältnis mehrerer Grundfarben bestimmt werden. Im Allgemeinen werden dazu rote, grüne und blaue Farbfilter verwendet. Im RGB-Farbraum werden damit für jeden Bildpunkt genau drei Helligkeiten gespeichert. Bei der Wiedergabe entsprechen diese der Helligkeit, mit der die einzelnen Subpixel eines Bildschirmes leuchten (siehe Abbildung 4). Durch den geringen Abstand zueinander können die einzelnen Subpixel vom menschlichen Auge nicht aufgelöst werden. Es entsteht ein Farbeindruck, welcher der additiven Mischung der Grundfarben entspricht.

Π
Π
Π
Π
П
Π
I
I
I
I
 I
ľ
ľ
ľ
I

Abbildung 4: Diskretisierung von Bildern - vom Pixel zum Subpixel.

Die möglichen Verarbeitungsschritte, die über die Bildakquise, Speicherung und Darstellung hinausgehen sind sehr vielfältig. Geläufige Beispiele sind das Filtern (z.B. Weichzeichnen) oder Skalieren (Ändern der Größe) der Bilder. Komplexere Algorithmen sind notwendig um den Inhalt digitaler Bilder zu analysieren. Die Fortschritte in der Miniaturisierung der Rechentechnik erlauben es, anspruchsvolle Bildverarbeitungsaufgaben, wie das Erkennen von Gesichtern, auf günstiger Hardware in Echtzeit durchzuführen.

#### 2.2 Einführung in OpenCV mit Python

Die Open Source Computer Vision Library ist eine Bibliothek von C-Funktionen für die Bildverarbeitung. Die durch OpenCV bereitsgestellten Funktionen vereinheitlichen den Zugriff auf Kamerahardware und erleichtern den Umgang mit komplexen Datentypen wir mehrkanaligen Bildern und Videos. OpenCV ist hardwareunabhängig. Dadurch lässt sich die Bibliothek unter verschiedenen Prozessorarchitekturen und Betriebssystemen nutzen (Windows, Linux, macOS, Android, iOS). Weiterhin existieren Schnittstellen zu vielen bedeutenden Programmiersprachen wie C++, C#, Java und Python.

#### 2.2.1 Auslesen der Kamera mit OpenCV

Listing 1: Aulesen der Kamera mit OpenCV in Python. Entnommen aus dem für dieses Praktikum vorgesehenen Paket mrt2pkgGMM3.

```
import cv2
if -_name__ = '__main__':
    cam_stream = cv2.VideoCapture(0)
    cv2.namedWindow('Demo', cv2.WINDOW_AUTOSIZE)
    while True:
        check, frame = cam_stream.read()
        frame = cv2.medianBlur(frame, 11)
        cv2.imshow('Demo', frame)
        if cv2.waitKey(1) == 27:
            break
        cam_stream.release()
        cv2.destroyAllWindows()
```

Listing 1 zeigt ein kurzes, aber vollständiges Python Programm, das OpenCV verwendet. Das Verständnis dieses Programmteils ist Voraussetzung zum Lösen der Aufgaben. OpenCV wird ständig weiterentwickelt. Deswegen muss darauf geachtet werden, auf welche Version sich die im Internet angebotenen Dokumentation und Beispiele beziehen. Dieser Versuch baut auf Version 4.5 auf.

#### 2.2.2 Hardwareinteraktion

Durch die große Menge an unterschiedlichen Verbraucher- und Industriekameras existiert eine nur schwer überschaubare Menge an Standards, Treibern und Softwareschnittstellen. OpenCV verbirgt die Kommunikation mit der Kamera vor dem Programmierer und bietet stattdessen ein einheitliches und einfach zu bedienendes Interface an. Damit können einzelne Bilder (Frames) sowohl von verschiedenen Kameras, als auch aus Videodateien gelesen werden. Dieses Interface präsentiert sich als Objekt vom Typ cv2.VideoCapture(). Nach erfolgreicher Initialisierung können durch Aufruf dessen Methode read() einzelne Bilder von der Kamera abgerufen werden.

#### 2.2.3 Wie "sieht" der Computer ein Bild?

Digitale Bilder können in einer Vielzahl von Formaten gespeichert werden. Im Rahmen dieses Versuches genügt es das Standardformat von OpenCV zu verstehen. Dabei handelt es sich um 24 Bit Farbbilder mit den drei Farbkanälen rot, grün und blau. Obwohl dieser Typ in der OpenCV Dokumentation als RGB-Format bezeichnet wird, liegen die Helligkeitswerte der einzelnen Kanäle in der Reihenfolge BGR im Arbeitsspeicher. Dies wird in Abbildung 5 veranschaulicht.



Abbildung 5: Ablage von RGB Bildern im Arbeitsspeicher bei Nutzung von OpenCV.

#### 2.2.4 Bildverarbeitung mit OpenCV

OpenCV stellt viele Funktionen bereit, um Bilder zu analysieren und zu manipulieren. Diese können direkt mit den von der Kamera ausgelesenen Bildern (i.e. *array*) umgehen. Ein Beispiel dafür ist die Funktion *cv2.medianBlur()*, welche in Listing 1 genutzt wird. Sie erwartet ein Bild (vom Typ *numpy.ndarray*) als Quelle. Durch direkten Zugriff auf die Bilddaten können beliebige eigene Algorithmen verwirklicht werden. Sie können Hilfe für die Nutzung von OpenCV mit Python unter [3] finden.

#### 2.2.5 Nutzung einer graphischen Oberfläche

OpenCV verfügt über Funktionen zum Erstellen einer graphischen Benutzeroberfläche (GUI), die als High-level GUI bezeichnet wird. Diese kann zur Darstellung von Bildern verwendet werden und erfasst Benutzereingaben über Maus und Tastatur. Um ein Fenster zu erstellen, genügt ein Aufruf der Funktion cv2.namedWindow(). Beim Erzeugen wird dem Fenster ein Name zugewiesen, durch den es im weiteren Programmverlauf identifiziert wird. Ein Bild kann dem Fenster durch die Funktion cv2.imshow() übergeben werden. Graphische Benutzeroberflächen werden im Allgemeinen durch Ereignisse (Events) gesteuert. Das bedeutet, dass ein Programmteil für die Ein- und Ausgabe dieser Ereignisse verantwortlich sein muss. In der High-level GUI übernimmt die Funktion cv2.waitKey() diese Rolle. Entgegen dem Namen liest diese Funktion nicht nur Tastatureingaben aus, sondern übernimmt die gesamte Ereignisverwaltung. Aus diesem Grund muss sie aufgerufen werden, wann immer sich der Inhalt eines Fensters ändern soll. Der Funktion kann eine Wartezeit (Timeout) übergeben werden. Diese bestimmt, wie lange auf eine Tastatureingabe gewartet wird, bevor das Programm weiterläuft.

#### 2.3 Hardware

Je nachdem, ob das Praktikum als Präsenz- oder Onlineveranstaltung durchgeführt wird, kommt entweder die Webcam in Ihrem Computer oder speziell für das Praktikum vorgesehene Hardware zum Einsatz. Sollte das Praktikum als Onlineveranstaltung stattfinden, bedenken Sie, dass das Praktikum als Gruppenarbeit angesetzt ist. Sollten Sie also keine Webcam zur Verfügung haben oder ähnliche Probleme auftreten, sind diese in aller Regel im Rahmen der Gruppe lösbar. Für den Fall einer Durchführung im Rahmen einer Präsenzveranstaltung, steht Ihnen eine *Logitech StreamCam* (https://www.logitech.com/dede/product/streamcam#specification-tabular) zur Verfügung. Die Kamera kann maximal mit einer Auflösung von 1920x1080 Pixeln bei 30 Bildern pro Sekunde ausgelesen werden.

### 3 Gute Praxis bei der Programmierung

Da nicht nur das Kolloquium, sondern auch das Erstellen bzw. Erweitern des Programms mit dem geforderten Funktionsumfang Teil der Bewertung ist, werden hier noch Tipps für die gute Praxis beim Programmieren gegeben. Sie sollten im Rahmen dieses Praktikums folgende Punkte beachten:

- Fügen Sie an den erforderlichen Stellen Kommentare ein, sodass Ihr vorgehen nachvollzogen werden kann.
- Versuchen Sie den Code möglichst lesbar (also einfach) zu halten.
- Achten Sie auf eine konsistente und verständliche Benennung von Variablen und ggf. Funktionen und Klassen (sollten für dieses kleine Projekt nicht notwendig sein).
- Achten Sie darauf, dass Ihr Code in der letzten Version, die Sie abgeben, funktioniert. Das heißt, dass alle drei Aufgabenteile (*aufgabe\_x.py*) erfolgreich ausgeführt werden können und die entsprechenden Aufgabenteile umgesetzt sind.

### 4 Vorbereitungsaufgaben

Beantworten Sie die folgenden Fragen (jeder Student einzeln) in Stichpunkten schriftlich. Die Antworten bilden die Grundlage für das Kolloquium und sind Bestandteil des Protokolls. Ohne diese schriftlichen Vorbereitungen findet das Praktikum nicht statt!

- Welche Bildaufnahmesensoren wurden vor Einsatz der CCD und CMOS Sensoren verwendet? Nennen Sie einige davon!
- Wie funktioniert die Bildaufnahme und -ausgabe der Bilddaten auf einem CCD Sensor?

- Wie funktioniert die Bildaufnahme und -ausgabe der Bilddaten auf einem CMOS Sensor?
- Vergleichen Sie kurz beide Bauformen (CCD, CMOS) in Bezug auf ihre praktische Anwendbarkeit.
- Wie wird mit Bildaufnahmesensoren eine Farbaufnahme ermöglicht (Grundprinzipien)?
- Welche technischen Anwendungen der elektronischen Bildaufnahme sind Ihnen bekannt?
- Nennen Sie speziell Anwendungen von Kameras in der Medizin.
- Nennen Sie die wichtigsten Baugruppen einer WebCam und erläutern deren Funktion.
- Nennen Sie den Unterschied zwischen Helligkeitswerten und Graustufen.
- Welche Größe (in Byte) hat ein Farbbild der Auflösung 128x128 mindestens?
- Wie kann man mit einem Programm aus einem Farbbild ein graustufen Bild erzeugen?

### 5 Versuchsdurchführung

#### 5.1 Aufgabenstellung

Im Verlauf des Praktikums sind folgende Aufgaben zu lösen. Lösen Sie die Aufgabe der Reihe nach und nutzen Sie die hierfür vorgesehenen Dateien/Skripte. Sofern möglich, können Sie den Code aus der vorherigen Aufgabe einfach in die neue Aufgabe übernehmen/kopieren. Bitte protokollieren Sie die Lösung jeder Aufgabe mithilfe eines Screenshots ihres Bildschirms.

Einführung: Führen Sie das Beispielprogramm example.py über das Terminal aus (siehe 1.4).

Aufgabe 1 (nutzen Sie hierfür die Datei *aufgabe\_1.py*)

- a. Verändern Sie das Programm so, dass ein zweites Fenster zur Anzeige geöffnet wird. In diesem sollen die modifizierten Bilder der nächsten Teilaufgabe dargestellt werden.
- b. Stellen Sie im zweiten Fenster nur den grünen Farbkanal des originalen Bildes dar.

Aufgabe 2 (nutzen Sie hierfür die Datei aufgabe\_2.py)

- a. Verändern Sie das Programm so, dass ein zweites Fenster zur Anzeige geöffnet wird. Nutzen Sie hierfür die Klasse *CustomWindow* und übergeben Sie ihr regelmäßig das darzustellende Bild. Diese Klasse stellt die Funktionen *imshow* sowie *namedWindow*, welche sich genauso wie über OpenCV nutzen lassen, zur Verfügung.
- b. Wenden Sie auf das in diesem Fenster dargestellte Bild den Schwellwertfilter aus thresholdfilter.py an indem Sie die Funktion apply\_threshold auf das Bild anwenden.
- c. Justieren Sie die gegebenen Schwellwerte im Fenster so, dass möglichst effektiv die Gesichtshaut (möglichst keine Haare oder Hintergrund) während der Aufnahme erkannt wird. Dokumentieren Sie die optimalen Schwellwerte.

**Aufgabe 3** (nutzen Sie hierfür die Datei *aufgabe\_3.py*): Bauen Sie auf dem letzten Stand von Aufgabe 2 auf. Instanziieren Sie ein Objekt der Klasse *PPGExtractor* bevor die Schleife ausgeführt wird. Übergeben Sie dieser Instanz regelmäßig das Originalbild nach Schwellwertanwendung. Nach ca. 10 Sekunden sollte das aus den Hautpixeln geschätzte Photoplethysmogramm der letzten 3 Sekunden angezeigt werden. Zusätzlich wird im Terminal in einem definierten Zeitintervall die geschätzte Herzrate ausgegeben. Die Bestimmung der Parameter ist hier sehr einfach umgesetzt und entsprechend Störanfällig (bspw. durch Bewegung oder Beleuchtungsschwankungen). Wie und warum das funktioniert, sollten Sie mit Ihrem Praktikumsbetreuer besprechen.

#### 5.2 Hinweise zur Durchführung

Für die bewertete Versuchsdurchführung tragen Sie sich bitte in folgende Umfrageliste ein: https://dud-poll.inf.tu-dresden.de/vyzh3BOIMQ/. Melden Sie sich mit Ihrer Praktikumsgruppe für einen Termin an. In dieser bewerteten Versuchsdurchführung werden die Vorbereitungsaufgaben (siehe 4) sowie die Lösungen zu den jeweiligen Aufgaben aus 5.1 geprüft.

### 6 Literatur

- Klaus R., Käser H.: Grundlagen der Computertechnik, Vdf Hochschulverlag AG an der ETH Zürich, 1998
- Stotz, Dieter: Computergestützte Audio- und Videotechnik, Springer Verlag, Berlin Heidelberg New York, 1995
- [4] How-to install python: https://realpython.com/installing-python/
- [3] OpenCV Python tutorials: https://opencv-python-tutroals.readthedocs.io/en/latest/py\_tutorials/py\_tutorials.html
- [5] How-to use windows command line with python: https://docs.python.org/3/faq/windows.html# how-do-i-run-a-python-program-under-windows
- [6] How-to use python with visual studio code: https://code.visualstudio.com/docs/python/ python-tutorial

### 7 Arbeits- und Brandschutzhinweise

Vorbeugende Maßnahmen:

- Die Praktikumsteilnehmer haben sich so zu verhalten, daß Gefahrensituationen und Unfälle vermieden werden.
- Die Befugnis zum Bedienen und Nutzen von Geräten ist auf den zugewiesenen Praktikumsplatz beschränkt.
- Eingriffe in die zum Praktikumsaufbau gehörenden Geräte sind nicht erlaubt.
- Der Anschluss und der Betrieb privater Geräte in den Praktikumsräumen ist verboten.
- Defekte an Geräten oder Gebäudeeinrichtungen sind unverzüglich dem Betreuer mitzuteilen. Betroffene Geräte sind außer Betrieb zu nehmen. Andere Personen sind vor Gefahren zu warnen. Den Anweisungen der Praktikumsbetreuer bzw. anderer aufsichtsführender Personen ist unbedingt Folge zu leisten.
- Betriebsfremde dürfen sich nur mit Erlaubnis des Praktikumsbetreuers in den Praktikumsräumen aufhalten.
- Rauchen und Umgang mit offenem Feuer ist nicht gestattet.
- Nach Ende des Praktikums ist der Arbeitsplatz in sauberem und aufgeräumten Zustand zu hinterlassen.
- Außergewöhnliche Ereignisse bzw. besondere Vorkommnisse sind umgehend dem Betreuer oder dem diensthabenden Assistenten zu melden.

# 8 Quellcode

虔 ex	xample.py ≻
1	<i>nnn</i>
2	
4	
5	Author: Matthieu Scherpf
6	Email: Matthieu.Scherpf@tu-dresden.de Website: https://becupiouss.github.ic/matthieuscherpf/
8	
9	Purpose: Für Praktikum Mikrorechentechnik 2 - Versuch GMM 3 - Bildaufnahme und
10	-verarbeitung mit OpenCV und Python
10	ини
12	
13	#
14 15	# Import benotigter python Pakete bzw. Module #
16	import cv2 # OpenCV Paket für Python
17	Harris Maria Chalad Theo des Datherlahansates in Tradicit and shaked with
10	dann wird der folgende Code ausgeführt
19	ifname == 'main':
20	#
21	Wenn nur eine Kamera am System angeschlossen ist, ist die ID <0>.
	Eventuell muss dieser Wert auf <1> gesetzt werden, sofern das System bspw.
22	Front- und Rückkamera besitzt -> ausprobieren.
22	<pre># cam stream = cv2.VideoCapture(0)</pre>
24	
25	#
20	welcher Auflösung soll die Kamera ausgelesen werden, mit wie vielen
	Bildern pro Sekunde, etc.); Die Parameter werden nicht zwingend von der
	verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht
27	menr korrekt ausgelesen werden kann: #
28	<pre># cam_stream.set(cv2.CAP_PROP_FRAME_WIDTH, 1920) # defaults to: 640</pre>
29	<pre># cam_stream.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080) # defaults to: 480 # com_stream_set(cv2.CAP_PROP_FRAME_HEIGHT, 1080) # defaults to: 20</pre>
31	# cam_stream.set(cv2.cAP_PROP_PPS, 50) # defaults to: 50
32	#
33	# Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster passt
	wird später über den ersten Übergabeparameter <'Demo'> referenziert
34	#
35	cv2.namedWindow('Demo', cv2.WINDOW_AUTOSIZE)
37	#
38	# Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das
39	Fenster muss hierbei angewählt (im Fokus) sein.) #
40	while True:
41	#
42	Werte zurückaeaeben: boolean check (True, wenn das frame erfolgreich
	ausgelesen wurde, was hier nicht weiter geprüft wird; False sonst),
42	frame (ein array, was die Werte des ausgelesenen frames enthält) #
44	<pre># check, frame = cam_stream.read()</pre>
45	
40	#
48	# Verarbeitung des eingelesenen Bildes
<b>49</b>	#
51	" # Anwendung eines Medianfilters mit einer Filtergröße von 11x11 Pixeln
52	<pre>frame = cv2.medianBlur(frame, 11)</pre>
53 54	#
55	# # Anzeigen des manipulierten frames
56	#
57	cv2.imshow('Demo', frame)
59	#
60	# Prüfen, ob eine Nutzereingabe getätigt wurde um das Programm zu
	<pre>beenden; 2/ entspricht der Escape-Taste (siehe ASCII Tabelle: http:// www.asciitable.com/)</pre>
61	#
62	<pre>if cv2.waitKey(1) == 27:</pre>
63 64	break
65	#
66	# Wenn das Programm via Escape-Taste beendet wurde muss die Kamera
	nierruber informiert werden und noch geöffnete Fenster sauber geschlossen werden
67	#
68	cam_stream.release()
70	CV2.uCSLFUYAIIWINUWS()

Ş	au	fgabe_1.py >
1	$\sim$	nnn
2		
3	$\sim$	Created: 08.02.2021, 13:04
5		Author: Matthieu Schernf
6		Email: Matthieu.Scherpf@tu-dresden.de
7	$\sim$	Website: https://becuriouss.github.io/matthieuscherpf/
8		
9	$\sim$	Purpose: Fur Praktikum Mikrorechentechnik 2 - Versuch GMM 3 - Bildaufnahme und
10		
11		ппп
12		
13		#
15		#
16		import cv2 # OpenCV Paket für Python
17		from thresholdfilter import apply_threshold
18		# wenn dieses Skrint über den Pythoninterpreter im Terminal gestartet wird.
10		dann wird der folgende Code ausgeführt
20	$\sim$	<pre>ifname == 'main':</pre>
21		
22		# Dem Konstruktor muss die ID (ein integer) der Kamera übergeben werden.
		Eventuell muss dieser Wert auf <1> gesetzt werden, sofern das System bspw.
		Front- und Rückkamera besitzt -> ausprobieren.
23		#
24		<pre>cam_stream = cv2.VideoCapture(0)</pre>
26		#
27		# OPTIONAL (zum experimentieren): Parameter können gesetzt werden (mit
		welcher Auflösung soll die Kamera ausgelesen werden, mit wie vielen
		Bildern pro Sekunde, etc.); Die Parameter werden nicht zwingend von der
		mehr korrekt ausgelesen werden kann!
28		#
29		<pre># cam_stream.set(cv2.CAP_PROP_FRAME_WIDTH, 1920) # defaults to: 640</pre>
30		# cam_stream.set(cv2.CAP_PROP_FRAME_HEIGHI, 1080) # defaults to: 480 # cam_stream_set(cv2.CAP_PROP_FPS_30) # defaults to: 30
32		
33		#
34		# Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster passt
		wird später über den ersten Übergabengrameter <'Demo'> referenziert
35		#
36		<pre>cv2.namedWindow('Demo', cv2.WINDOW_AUTOSIZE)</pre>
37		#
39		# # Lese die Bilder der Kamera aus, bis die "ESC" Taste betätiat wird (Das
		Fenster muss hierbei angewählt (im Fokus) sein.)
40		#
41	$\sim$	while True:
43		# lese das nächste verfügbare frame der Kamera ein; Es werden zwei
		Werte zurückgegeben: boolean check (True, wenn das frame erfolgreich
		ausgelesen wurde, was hier nicht weiter geprüft wird; False sonst),
44		#
45		<pre>check, frame = cam_stream.read()</pre>
46		
47		#
49		# # Verarbeitung des eingelesenen Bildes
50		#
51		#
52		
55		#
55		#
56		cv2.imshow('Demo', frame)
57		#
59		# # Prüfen, ob eine Nutzereingabe getätigt wurde um das Programm zu
		beenden; 27 entspricht der Escape-Taste (siehe ASCII Tabelle: http://
~~		www.asciitable.com/)
60 61	~	<pre># if cv2.waitKev(1) == 27.</pre>
62	~	break
63		
64		
65		# wenn das Programm via Escape-laste beendet wurde muss die Kamera hierrüher informiert werden und noch geöffnete Fenster sauber geschlossen
		werden
66		#
67		cam_stream.release()
69		CATINGER ONWEINTING

ę	aufgabe_2.py >
1	ппп
2	
3	Created: 08.02.2021, 13:04
4	
5	Author: Matthieu Scherpf
6	Email: Matthieu.Scherpf@tu-dresden.de
7	Website: <a href="https://becuriouss.github.io/matthieuscherpf/">https://becuriouss.github.io/matthieuscherpf/</a>
8	
9	Purpose: Für Praktikum Mikrorechentechnik 2 - Versuch GMM 3 - Bildaufnahme und
10	-verarbeitung mit OpenCV und Python
10	
12	
13	#
14	# Import benötigter python Pakete bzw. Module
15	#
16	<pre>import cv2 # OpenCV Paket für Python</pre>
17	<pre>from thresholdfilter import apply_threshold</pre>
18	from customWindow import CustomWindow as cw
19	# vann diasas Shniat ühan dan Duthanintannatan im Tanming, gastantat vind
20	# wenn uteses skript über den rythoninterpreter um Terminut gesturtet wurd, dann wird der folgende Code gusgeführt
21	if name == ' main ':
22	#
23	# Dem Konstruktor muss die ID (ein integer) der Kamera übergeben werden.
	Wenn nur eine Kamera am System angeschlossen ist, ist die ID <0>.
	Eventuell muss dieser Wert auf <1> gesetzt werden, sofern das System bspw.
	Front- und Rückkamera besitzt -> ausprobieren.
24	#
25	<pre>cam_stream = cv2.VideoCapture(0)</pre>
20	#
27	# # OPTIONAL (zum experimentieren): Parameter bönnen gesetzt werden (mit
20	welcher Auflösung soll die Kamera gusgelesen werden, mit wie vielen
	Bildern pro Sekunde, etc.); Die Parameter werden nicht zwingend von der
	verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht
	mehr korrekt ausgelesen werden kann!
29	#
30	<pre># cam_stream.set(cv2.CAP_PROP_FRAME_WIDTH, 1920) # defaults to: 640</pre>
31	<pre># cam_stream.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080) # defaults to: 480</pre>
32	# cam_stream.set(cv2.CAP_PROP_FPS, 30) # defaults to: 30
20	*
35	# # Initialisiere ein Fenster, zur anzeige der Bilder: Das Fenster nasst
	sich automatisch der Größe bzw. der Auflösung der Aufnahme an: Das Fenster
	wird später über den ersten Übergabeparameter <'Demo'> referenziert
36	#
37	cv2.namedWindow('Demo', cv2.WINDOW_AUTOSIZE)
38	
39	#
40	# Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das
41	renster muss nierbei angewanit (im Forus) sein.)
41	# while True:
43	#
44	# lese das nächste verfügbare frame der Kamera ein; Es werden zwei
	Werte zurückgegeben: boolean check (True, wenn das frame erfolgreich
	ausgelesen wurde, was hier nicht weiter geprüft wird; False sonst),
	frame (ein array, was die Werte des ausgelesenen frames enthält)
45	#
40	check, trame = cam_stream.read()
48	#
49	#
50	# Verarbeitung des eingelesenen Bildes
51	#
52	#
53	
54	#
55	# Anzeigen des manipulierten frames
57	# cv2 imchow('Dama' frama)
58	CV2. AND TOWN ( DCIND ) 11 dire /
59	#
60	# Prüfen, ob eine Nutzereingabe getätigt wurde um das Programm zu
	beenden; 27 entspricht der Escape-Taste (siehe ASCII Tabelle: http://
	www.asciitable.com/)
61	#
62	<pre>if cv2.waitKey(1) == 27:</pre>
63	break
65	#
66	" # Wenn das Proaramm via Escane-Taste beendet wurde muss die Kamera
50	hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen
	werden
67	#
68	<pre>cam_stream.release()</pre>
69 79	cv2.destroyAllWindows()

<pre>variable (% a) (% a</pre>	<u>a</u> .	
<pre>1 *** 2 ********************************</pre>		augabe_s.py 7
<pre>2 Created: 08.02.021, 13:04 3 Witho: Matthie Schergf 4 Website: https://beurieus.githuk.io/matthieuschergf/ 7 Website: https://beurieus.githuk.io/matthieuschergf/ 8 Urpose: für Posttime Withorechentechtk 2 - Versuch CM 3 - Bildaufname und -verarbeitung mit OpenCV and Python 10 ************************************</pre>	1	
<pre>1 Franker, Bender, 2014 Author: Northies Scherpfile-dreaden.de Website: https://beurieuss.github.is/wathheuscherpf/ Purpose: fin Prathieum Nikrorechentechth 2 - Versuch GW 3 - Bildaufnahme und -verarbeitung mit GenCV und Python """" """ """ """ """ """" """" """" """ """" """ """" """ """" """" """" """" """" """" """" """" """" """" """" """" """" """" """"" """"""""</pre>	2	
durbne: Marthieu Schergf     debsite: https://becuriouss.gthub.io/mathieuschergf/     dubsite: https://becuriouss.gthub.io/mathieuschergf/     dubsite: https://becuriouss.gthub.io/mathieuschergf/     dubsite: https://becuriouss.gthub.io/mathieuschergf/     dubsite: https://becuriouss.gthub.io/mathieuschergf/     dubsite: https://becuriouss.gthub.io/mathieuschergf/     dubsite: ption Postet Law Mikroacebertechtk 2 - Versuch GM 3 - Bildaufname undverarbeitung mit GeencY und Python     dim und GeencY und Python     from theodifilte: python Postet bay. Nobule     from theodifilte: python	2	Createa: 08.02.2021, 15:04
<pre>     Email: Nathies.Scherfpftu-dreaden.de     Website: https://beurious.github.is/mathieuscherff/     Purpose: Fib Prottham Mikrorechentkel.2 - Versuch GM 3 - Bildaufnahme und     -verarbeitum mit GeenCV und Python     # Import bendtigter python Pakete bzw. Nodule     # Import bendtigter import apply_threshold     from tockenHood mass die ID (ein integer) der Kamera übergeben werden.     Wen mur eine Komera am System angeschossen ist, ist die ID (00.     Eventuelt mass dies ID (ein integer) der Kamera übergeben werden.     Wen mur eine Komera am System angeschossen ist, ist die ID (00.     Eventuelt mass diese leit. J ic Augescheren.     # Jmore Import bendtigt und se kann sein, duss givten bzw.     Front- und Mickhamera besitst -&gt; ausprobleren.     # Jmore Import bendtigt und se kann sein, duss die Kamera nicht mehrende und er-     verwendeten Kamera unterstülzt und se kann sein, duss die Kamera nicht mehrende understülzt und se kann sein, duss die Kamera nicht mehrende under GuePROP_REME_KDUM, 1200 Fenter passt     sich autoantisch der Grüße Dwu. der Aufläung der Aufnähm en; Das Fenster     wird apter über der Grüße Dwu. der Aufläung der Aufnähm en; Das Fenster     wird apter die reich GuePROP_REME_KDUM, 1200 Fenter passt     sich autoantisch der Grüße Dwu. der Aufläung der Aufnähm en; Das Fenster     wird apter die reich GuePROP_REME_KDUM, 1200 Fenter passt     sich autoantisch der Grüße Dwu. der Aufläung der Aufnähme enzöl, referenziert     # Lese die Bilder der Kamera aus, bis die "ESC" Toste betätigt wird (Das     Fenster muss hierbeite gerpräft wird; Tats eistni,     fent (cim mry, num die kerte des ausgete</pre>	5	Author: Matthieu Scherpf
<pre>Mosite: http://beuriaus.gitub.io/mattheacheng// Purpose: Fur Praktium Mikrorechentechnik 2 - Versuch GMI 3 - Bildaufnahme und -verarbeitung mit QenCV und Python</pre>	6	Email: Matthieu.Scherpf@tu-dresden.de
<pre>Propose fun Praktium Mikrorechemtechnik 2 - Versuch GMI 3 - Bildaufinnhme und -verarbeitung mit OpenCV und Python # Import benötigter python Pohete bax. Nobule # Import v2 # OpenCV Pohet für Python from thresholdFilter Import apply threshold from cutesminour import Cutosminous as cu from prestructor import Pohete bax. Nobule # amort examinour import Cutosminous as cu from prestructor import Poheter terms in Terminal gestartet wird, damn wird der folgende Code ausgeführt if _ mem</pre>	7	Website: https://becuriouss.github.io/matthieuscherpf/
<pre>9 Purpose: Fur Protectiums mit Concententies 2 - Versuch GMT 3 - Bildoufnamme und -verorbeitums mit Concent und Python 1</pre>	8	
<pre>-verarbeitung mit OpenCV und Python  # Import tenntityter python Pakete bax, Module # Import tenntityter python Pakete bax, Module # Import tenntityter python Pakete bax, Module # Import tenntityter import apply threshold from turesholdefilter import apply threshold from turesholdefilter import apply threshold from turesholdefilter import apply threshold from appextractor import PoEstartator  # Jamon disess Skript über den Pythoninterpreter im Terminal gestartet wird, dam wird der Folgemet Lastowilduke as (u from presentator muss die ID (ch integer) der Kamera übergeben werden. idem nur eine Kamera am System appeschlessen ist, ist die ID (d). Eventuell muss dieser Mert auf (2) gestett werden, sofern das System bayw. Front und Aubhamer besktt -&gt; ausproblerer. com_stream = cv2.VideoCapture(0) # J</pre>	9	Purpose: Für Praktikum Mikrorechentechnik 2 - Versuch GMM 3 - Bildaufnahme und
<pre>19 19 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>		-verarbeitung mit OpenCV und Python
<pre>11 mm 12 # Import benotigter python Pakete bas. Nobule 13 # Import v2 # OpenCV Paket fur Python 14 from thresholdFilter import apply threshold 14 from customAndow as cu 15 from ppertractor import PrOEstructure 14 # used misses Shrip ibor dan Pythom interpreter in Terminal gestartet wird, 15 mme</pre>	10	
<pre>11 # Import benotigter python Pakete bax. Hodule 12 # Import V2 # OpenVY Paket für Python 13 # proft V2 # OpenVY Paket für Python 14 # wenn dieses Skript über den Pythoninterpreter im Terminal gestartet wird, 15 # ome dieses Skript über den Pythoninterpreter im Terminal gestartet wird, 16 # one dieses Skript über den Pythoninterpreter im Terminal gestartet wird, 17 # one dieses Skript über den Pythoninterpreter im Terminal gestartet wird, 18 # wenn dieses Skript über den Pythoninterpreter im Terminal gestartet wird, 19 # one Konstruktor muss del DD (eln integer) der Kamera übergeben werden. 19 # Den Konstruktor muss del DD (eln integer) der Kamera übergeben werden. 19 # one Konstruktor muss del DD (eln integer) der Kamera übergeben werden. 19 # onertel mussier werden bestet -&gt; ausprobleren. 20 # oPTTOML (um experimentieren): Parameter bönnen gesett werden (mit 21 welcher Aufläusung stul die Camera uungelesen werden, mit wielen 23 # oPTTOML (um experimentieren): Parameter bönnen gesett werden (mit 24 wendeten Kamera unterstützt und es kann sein, dass die Kamera nicht 25 # oPTTOML (um experimentieren): Parameter bönnen gesett werden (mit 26 wendeten Kamera unterstützt und es kann sein, dass die Kamera nicht 27 # origitesen werden kam! 28 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 29 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 29 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 29 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 29 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 20 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 21 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 22 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 23 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 24 cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 25 # cam_streem.set(v2. GAP_PROF_RAME_HIGHT, 1920) # defaults to: 480 25 # cam_streem.sete</pre>	11	
<pre># Import benötigter python Pakete bzw. Module import cv2 # OpenCV Paket für Python from thresholdFilter import apply threshold from custominious import Custominious as cv from prestractor import PoEstractor # wenn disess Skript über den Pythoninterpreter im Terminal gestartet wird, dam wird der folgende Code ausgefüht if none main:: # Dem Konstruktor muss die ID (ein integer) der Kamera übergeben werden. Wenn mur eine Kamera aus System angeschlassen ist, ist die ID (eb. Eventuell muss dieser Wert auf &lt;1) gestetzt werden, sofern das System bzw. Front - um Rickhamer beskitt -&gt; ausgrobierne. # OPTIONAL (zum experimentieren): Parameter kömen gesetzt werden (mit welcher Auflösung sol die Kamera ausgetesm werden, mit vie vielen Blidern pro Sekunde, etc.); Die Parameter werden nicht zwingend von der verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht # com_stremm.set(col2.CMP_POR_PKMP_KIDTN, 1920) # defaults to: 640 # com_stremm.set(col2.CMP_POR_PKP_KIDTN, 1920) # defaults to: 640 # com_stremm.set(col</pre>	12	#
<pre>is g</pre>	14	" # Import benötiater python Pakete bzw. Module
<pre>16 import cv2 = 0 QuercV Paket für Python 17 from thresholdfilter import apply threshold 18 from pastractor import PPEEtractor 24 2 # wenn dieses Stript über den Pythoninterpreter im Terminal gestartet wird, 25 damm wird der folgende Code ausgeführt 26 if promemain: 27 # Den Konstruktor muss die ID (ein integr) der Kamera übergeben werden. 28 Wenn auf eine Kamera am System angeschlossen ist, ist die ID (00. 29 # Den Konstruktor muss die ID (ein integr) der Kamera übergeben werden. 29 # Den Konstruktor muss die ID (ein integr) der Kamera übergeben werden. 20 Einstrukt muss die ider Vis gestet werden, sofern das System bepw. 29 # DerTMONL (zum experimentieren): Parameter Nännen gesetzt werden (mit ausged von der 29 # derTMONL (zum experimentieren): Parameter Nännen gesetzt werden (mit 20 werwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht 20 mer korrekt ausgelesen werden, kann! 21 # cam_stream.set(v2.CAP PROP_FRAME_MEDIGT, 1280) # defaults to: 640 23 # cam_stream.set(v2.CAP PROP_FRAME_MEDIGT, 1280) # defaults to: 480 24 # am_stream.set(v2.CAP PROP_FRAME_MEDIGT, 1280) # defaults to: 480 25 # 26 Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster 27 wird später über den ersten Übergubepanwerter ("Demo'r referentiert 28 29 # Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster 27 wird später über den ersten Übergubepangenter ("Demo'r referentiert 28 29 # Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das 29 Fenster muss Hierbei angewählt (ün Fohus) sein.) 29 # Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das 29 Fenster muss Hierbei angewählt (ün Fohus) sein.) 20 # Lese das nächste verfügbare frame der Kamera ein; Es werden zweil 29 # Lese das nächste verfügbare frame der Kamera ein; Es werden zweil 20 # Lese das nächste verfügbare frame der Kamera ein; Es werden zweil 20 # Lese das nächste verfügbare frame der Kamera ein; Es werden zweil 20 # Lese das nächste verfügbare frame der Kamera ein; Es werden zweil 20 # Lese da</pre>	15	#
<pre>17 from thresholdfilter import apply_threshold 17 from pogextractor import PREstractor 19 17 from pogextractor import PREstractor 11 21 # wenn dieses Skript über den Pythoninterpreter im Terminal gestartet wird, 22 dam wird der folgende Code ausgeführt 23 if</pre>	16	<pre>import cv2 # OpenCV Paket für Python</pre>
<pre>13 from customkindow import LyDExtractor 23 # wenn dieses Skript über den Pythoninterpreter im Terminal gestartet wird, 24 25 26 den Konstruktor muss die 10 (ein integer) der Kamera übergeben werden. 25 # Der Konstruktor muss die 10 (ein integer) der Kamera übergeben werden. 26 Konstruktor muss die 10 (ein integer) der Kamera übergeben werden. 27 # Der Konstruktor muss die 10 (ein integer) der Kamera übergeben werden. 28 # Der Konstruktor muss die 10 (ein integer) der Kamera übergeben werden. 29 # OPTIONAL (zum experimentieren): Parameter können gesetzt werden (mit 20 # OPTIONAL (zum experimentieren): Parameter können gesetzt werden (mit 29 # OPTIONAL (zum experimentieren): Parameter werden nicht zwingend von der 29 werwendeten Kamera umterstützt und es kann sein, dass die Kamera anicht 20 met korrekt ausgelesen werden kann! 21 # cam_stream.set(cv2.CAP_APOP_FRAME_HIGHT, 1920) # defaults to: 640 22 # cam_stream.set(cv2.CAP_APOP_FRAME_HIGHT, 1920) # defaults to: 480 23 # cam_stream.set(cv2.CAP_APOP_FRAME_HIGHT, 1920) # defaults to: 480 24 # cam_stream.set(cv2.CAP_APOP_FRAME_HIGHT, 1920) # defaults to: 480 25 # Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster 27 wird später über den ersten Übergabeparameter ('Den') referenziert 28 # Liste die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das 29 Fenster muss hierbei angewählt (im Fokus) sein.) 28 # Liste die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das 29 Fenster muss hierbei angewählt (im Fokus) sein.) 39 # Liste die Bilder der Kamera nicht der Gradies enthält) 40 # cam_stream.steindendw('Demo', cv2.MIDOM_AUTOSIE) 41 # Lese die Bilder der Kamera mas, bie frace ('Den') referenziert 42 # Lese das nächste verfügbare frame der Kamera ain; Es werden zwei 44 # Lese das nächste verfügbare frame der Kamera ain; Es werden zwei 45 # Lese das nächste verfügbare frame der Kamera ain; Es werden zwei 45 # Lese das nächste verfügbare frame der Kamera ain; Es werden zwei 46 # Lese die Bilder der Kamera aus, biet fes ausopies enthält</pre>	17	<pre>from thresholdfilter import apply_threshold</pre>
<pre>fram pogextractor import PPGExtractor  fram pogextractor import PPGExtractor  fram ndieses Skript über den Pythoninterpreter im Terminal gestartet wird,     dann wird der folgende Code ausgeführt []  f</pre>	18	from customWindow import CustomWindow as cw
<pre>49 49 49 49 49 49 49 40 40 40 40 40 40 40 40 40 40 40 40 40</pre>	19	from ppgextractor import PPGExtractor
<pre>21  * Meth dieses Set UL wer den rythoniter feter un (erhund: gesuntet win), 22  dan wird der folgende Code ausgefühtt 23  ifname == 'main': 24  # Dem Konstruktor muss die ID (ein integer) der Kamera übergeben werden. 25  Henn nur eine Kamera an System angeschlossen ist, ist die D &lt;0&gt;. 26  Eventuell muss dieser Wert auf (1&gt; gestit werden, sofern das System bepw. 27  Front- und Rückhamera besitt- &gt; ausprobleren. 28  # Dem Konstruktor gold die Kamera ausgelesen werden, mit wie vielen 29  # OPTIOMU (zum experimentieren): Parameter können gesetzt werden (mit 20  # OPTIOMU (zum experimentieren): Parameter können gesetzt werden (mit 29  # OPTIOMU (zum experimentieren): Parameter können gesetzt werden (mit 29  # COPTIOMU (zum experimentieren): Parameter werden nicht zwingend von der 29  werwendeten Komera unterstützt und es kann sein, dass die Kamera nicht 29  # cam_stream.set(cv2.CAP_PROP_FRAVE_HEDIT, 1920) # defaults to: 640 29  # cam_stream.set(cv2.CAP_PROP_FRAVE_HEDIT, 1920) # defaults to: 480 29  # cam_stream.set(cv2.CAP_PROP_FRAVE_HEDIT, 1920) # defaults to: 480 20  # cam_stream.set(cv2.CAP_PROP_FRAVE_HEDIT, 1920) # defaults to: 480 20  # Initialisiere ein Fenster, zum anzeige der Bilder; Das Fenster passt 20  sit automatisch der Größe zuw. der AufGaung der Aufnahm en; Das Fenster 20  wird später über den ersten Übergabeparameter c'Demo'&gt; referenziert 20  # Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das 20  Fenster muss hierbei angewählt (im Fokus) sein.) 2 # Lese das nächste verfügdare frame der Kamera ein; Es werden zwei 20  Kerte zurückgegeben: boolean check (True, wenn das frame erfolgreich 20  ausgelesen wurde, was hier nicht weiter genift wird; false sonst), 20  frame (ein arroy, was die Werte des ausgelesenen frames enthält) 20  # Lese das machate verfügdare frame der Kamera ein; Es werden zwei 21  # Anzeigen des manipulierten frames 22  # Anzeigen des manipulierten frames 23  # Anzeigen des manipulierten frames 24  # Anzeigen des manipulierten frames 25  # Anzeigen</pre>	20	# warm diagon Christ über der Dutheristerpreter im Terminal gestertet wird
<pre>is</pre>	21	# wenn aleses Skript über den Fylhoninterpreter im Terminal gestartet wird, dann wind den folgende Gode gusgeführt
<pre># Dem Konstruktor muss die ID (ein integer) der Kamera übergeben werden. Wenn nur eine Kamera am System angeschlossen ist, ist die ID (0). Eventuell muss dieser Wert auf (1) gesetzt werden, sofern das System bspw. Front- um Rückhmera besitzt -&gt; ausprobiern. #</pre>	22	if name == ' main ':
<pre>24 # Dem Konstruktor muss die ID (ein integer) der Komera übergeben werden. Wenn nur eine Kamera am System angeschlossen ist, ist die ID ‹0. Eventuell muss dieser Wert auf <l> gestatt = Angestatt = Angestatt werden, sofern das System bapw. Front- und Rückkamera besitzt = Angestatt werden, sofern das System bapw. Front- und Rückkamera besitzt = Angestatt werden, sofern das System bapw. Front- und Rückkamera besitzt = Angestatt werden, sofern das System bapw. Front- und Rückkamera besitzt = Angestatt werden, mit wie vielen Bildern por Steunde, etc.): Die Parameter werden nicht zwingend von der verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht mehn korret ausgelesen werden kann! # cam_streem.set(vu2.CAP_PROP_FRAME_HIGHT, 1920) # defaults to: 640 # cam_streem.set(vu2.CAP_PROP_FRAME_HIGHT, 1920) # defaults to: 489 # cam_streem.set(vu2.CAP_PROP_FRAME_HIGHT, 1920) # defaults # cam_streem.set(vu2.CAP_PROP_FRAME_HIGHT, 1920) #</l></pre>	23	#
<pre>Wenn nur eine Kamera am System angeschlossen ist, ist die ID «D. EventuelL muss dieser Wert auf <lop bspw.<br="" das="" gestert="" sofern="" system="" werden,="">Front- und Rückkamera besitzt -&gt; ausprobieren. #</lop></pre>	24	# Dem Konstruktor muss die ID (ein integer) der Kamera übergeben werden.
<pre>Functual muss dieser Wert auf <l> gesetzt werden, sofern das System bspw. Front um Richkamera besitzt -&gt; ausprobieren. # cam_stream = cv2.VideoCapture(0) # # OPTIONAL (zum experimentieren): Parameter können gesetzt werden (mit welcher Auflösung soli die Kamera ausgelesen werden, mit wie vielen Bildern pro Stehunde, etc.); Die Parameter werden nicht zwingend von der verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht mehr korrekt ausgelesen werden kann! # com_stream.set(cv2.CAP_PROP_FRAME_HIDTH, 1920) # defaults to: 640 # com_stream.set(cv2.CAP_PROP_FRAME_HIDTH, 1920) # defaults to: 480 # com_stream.set(cv2.CAP_PROP_FRAME_HIDTH, 1920) # defaults to: 530 # com_stream.set(cv2.CAP_PROP_FS, 30) # defaults to: 700 * referenzient wind später Wiber den ersten Ubergabeparameter </l></pre> Vond später Ward for den strent Disegabeparameter Vono's referenzient # com_stream.set com_stream.set sis die "ESC" Taste betätigt wird (Das Fenster muss hierbei angewählt (im Fokus) sein.) # terse auskingegeben: boolean check (True, wenn das frame erfolgreich ausgelesen wurde, was die Nerte des ausgelesenen frames enthält) # terse auskingegeben: boolean check (True, wenn das frame erfolgreich ausgelesen wurde, was die Nerte des ausgelesenen frames enthält) # terse auskingegeben: boolean check (True, wenn das frame erfolgreich ausgelesen wurde, was die Nerte des ausgelesenen frames enthält) # tervarbeitung des eingelesenen Bildes # verarbeitung des eingelesenen Bildes # warascitable.com/) # www.ascitable.com/) # www.ascitable.com/) # www.ascitable.com/) # www.ascitable.com/) # www.ascitable.com/) # tereak # Anzelgen des manipulierten frames # therräuber informiert werden und		Wenn nur eine Kamera am System angeschlossen ist, ist die ID <0>.
<pre>Front- und Rückhamera besitzt -&gt; ausprobieren.  Gam_stream = cv2.VideoCapture(9)  # Cam_stream = cv2.VideoCapture(9)  # Com_stream set(cv2.CAP_PROP_provement example of the set of th</pre>		Eventuell muss dieser Wert auf <1> gesetzt werden, sofern das System bspw.
<pre>20 #</pre>		Front- und Rückkamera besitzt -> ausprobieren.
<pre>20 Cam_stream = cv2.vlaeeCapture(0) 22 23 24 25 25 25 26 26 27 26 27 27 28 28 29 29 29 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	25	#
<pre># OPTIONAL (zum experimentieren): Parameter können gesetzt werden (mit # OPTIONAL (zum experimentieren): Parameter werden nicht zwingend von der verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht mehr korrekt ausgelesen werden kann! # com_stream.set(cv2.CAP_PROP_FANE_MIDIN, 1920) # defaults to: 640 # com_stream.set(cv2.CAP_PROP_FANE_MELGHT, 1980) # defaults to: 480 # com_stream.set(cv2.CAP_PROP_FANE_MELGHT, 1980) # defaults to: 480 # com_stream.set(cv2.CAP_PROP_FANE_MELGHT, 1980) # defaults to: 480 #</pre>	26	<pre>cam_stream = cv2.videocapture(0)</pre>
<pre># OPTIONAL (zum experimentieren): Parameter können gesetzt werden (mit welcher Auflösung soll die Kamera ausgelesen werden, mit wie vielen Bildern pro Sekunde, etc.); Die Parameter werden nicht zwingend von der verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht mehr korrekt ausgelesen werden kann! # com_stream.set(vo2.CAP_PROP_FANE_HIDIH, 1920) # defaults to: 640 # com_stream.set(vo2.CAP_PROP_FANE_HIDIH, 1920) # defaults to: 640 # com_stream.set(vo2.CAP_PROP_FANE_HIDIH, 1920) # defaults to: 480 # com_stream.set(vo2.CAP_PROP_CPANE_HIDIH, 1920) # defaults to: 480 # com_stream.set(vo2.CAP_PROP_CPANE_HIDIH, 1920) # defaults to: 480 # com_stream.set(vo2.CAP_PROP_CPANE_HIDIH, 1920) # defaults to: 480 # com_stream.set(vo2.CAP_PROP_FANE_HIDIH, 1920) # defaults to: 480 Programm via Escape-Taste beendet wurde muss</pre>	22	#
<pre>welcher Aufläsung soll die Kamern ausgelesen werden, mit wie vielen Bildern pro Sekunde, etc.); Die Parameter werden nicht zwingend von der verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht mehr korrekt ausgelesen werden kann! #</pre>	29	" # OPTIONAL (zum experimentieren): Parameter können aesetzt werden (mit
<pre>Bildern pro Sekunde, etc.); Die Parameter werden nicht zwingend von der verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht mehr korreit ausgelesen werden kann! #</pre>		welcher Auflösung soll die Kamera ausgelesen werden, mit wie vielen
<pre>verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht mehr korrekt ausgelesen werden kann! # cam_stream.set(cv2.CAP_PROF_RAWE_WIDTH, 1920) # defaults to: 640 # cam_stream.set(cv2.CAP_PROF_RAWE_WIDTH, 1920) # defaults to: 640 # cam_stream.set(cv2.CAP_PROF_RAWE_WIDTH, 1920) # defaults to: 480 # Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster passt sich automatisch der Größe bzw. der Auflösung der Aufnahme an; Das Fenster wird später über den ersten Übergabeparameter &lt;'Demo'&gt; referenziert # cv2.namedkindow('Demo', cv2.WINDOM_AUTOSIZE) #</pre>		Bildern pro Sekunde, etc.); Die Parameter werden nicht zwingend von der
<pre>mehr korrekt ausgelesen werden kann! #</pre>		verwendeten Kamera unterstützt und es kann sein, dass die Kamera nicht
<pre>30 #</pre>		mehr korrekt ausgelesen werden kann!
<pre>31 # com_stream.set(v2.CAP_PROP_FAME_HIGHT, 1920) # defaults to: 840 32 # com_stream.set(v2.CAP_PROP_FAME_HIGHT, 1920) # defaults to: 480 33 # com_stream.set(v2.CAP_PROP_FAME_HIGHT, 1920) # defaults to: 480 34 #</pre>	30	#
<pre>32 * Com_Stream.set(v2.CAP_PROP_FPS, 30) # defaults to: 30 33 # cam_stream.set(v2.CAP_PROP_FPS, 30) # defaults to: 30 34 35 # 36 # Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster passt 37 sich automatisch der Größe bzw. der Auflösung der Aufnahme an; Das Fenster 38 vird später über den ersten Übergabeparameter &lt;'Demo'&gt; referenziert 39 40 # 41 # Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das 39 Fenster muss hierbei angewählt (im Fokus) sein.) 42 # 44 # # 45 # Lese das nächste verfügbare frame der Kamera ein; Es werden zwei 44 # # 45 # Lese das nächste verfügbare frame der Kamera ein; Es werden zwei 44 # # 45 # Lese das nächste verfügbare frame der Kamera ein; Es werden zwei 46 # des ausgelesen wurde, was die Werte des ausgelesenen frames enthält) 47 check, frame = cam_stream.read() 48 49 #</pre>	31	# cam_stream.set(cv2.CAP_PROP_FRAME_WIDTH, 1920) # defaults to: 640
<pre>34 35 4 #</pre>	33	# cum_stream.set(cv2.CAF_FNOF_FNAHL_HEIGHT, 1000) # defaults to: 400
<pre>#</pre>	34	# cum_st cum.set(cvz.cm_nor_rrsy.so) # ucjuuces co. so
<pre>36 # Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster passt sich automatisch der Größe bzw. der Aufhösung der Aufnahme an; Das Fenster wird später über den ersten Übergabeparameter &lt;'Demo'&gt; referenziert 37 38 cv2.namedwindow('Demo', cv2.WINDOM_AUTOSIZE) 39 40 # 41 # Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das Fenster muss hierbei angewählt (im Fokus) sein.) 42 # 43 while True: 44 # 45 # Lese das nächste verfügbare frame der Kamera ein; Es werden zwei 46 Werte zwückagegeben: boolean check (True, wenn das frame erfolgreich 47 ausgelesen wurde, was hier nicht weiter geprüft wird; False sonst), 47 frame (ein array, was die Werte des ausgelesenen frames enthält) 48 49 # 50 # 51 # Verarbeitung des eingelesenen Bildes 52 # 53 # 54 # 55 # 56 # Anzeigen des manipulierten frames 57 # 58 cv2.imshow('Demo', frame) 59 59 50 # 50 # 51 # Prüfen, ob eine Nutzereingabe getätigt wurde um das Programm zu 59 beenden; 27 entspricht der Escape-Taste (siehe ASCII Tabelle: http:// 53 Wera.sciitable.com/) 54 55 # 56 # Moni das Program via Escape-Taste beendet wurde muss die Kamera 56 hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen 56 werden 57 werden 58 werden 59 cam_stream.release() 50 cov2.destroyAllWindows()</pre>	35	#
<pre>sich automatisch der Größe bzw. der Auflösung der Aufnahme an; Das Fenster wird später über den ersten Übergabeparameter &lt;'Demo'&gt; referenziert # cv2.namedWindow('Demo', cv2.WINDOM_AUTOSIZE) # #</pre>	36	# Initialisiere ein Fenster, zur anzeige der Bilder; Das Fenster passt
<pre>wird später über den ersten Übergabeparameter &lt;'Demo'&gt; referenziert #</pre>		sich automatisch der Größe bzw. der Auflösung der Aufnahme an; Das Fenster
<pre>#</pre>		wird später über den ersten Übergabeparameter <'Demo'> referenziert
<pre>33 CV2.mamedwindow('Demo', CV2.WINDOW_AUTOSIZE) 34 35 46 47 4 # Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das Fenster muss hierbei angewählt (im Fokus) sein.) 42 43 44 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4</pre>	37	#
<pre>39 #</pre>	38	cv2.namedWindow('Demo', cv2.WINDOW_AUTOSIZE)
<pre># Lese die Bilder der Kamera aus, bis die "ESC" Taste betätigt wird (Das Fenster muss hierbei angewählt (im Fokus) sein.) # while True: # #</pre>	39	#
<pre>Fenster muss hierbei angewählt (im Fokus) sein.)  Fenster muss hierbei angewählt (im Fokus) sein.)  Fenster muss hierbei angewählt (im Fokus) sein.)  fenster muss hierbei angewählt (im Fokus) sein.)  funder the set of the set</pre>	41	" # Lese die Bilder der Kamera aus, bis die "ESC" Taste betätiat wird (Das
<pre>42 #</pre>		Fenster muss hierbei angewählt (im Fokus) sein.)
<pre>43 44 45 44 45 45 46 46 47 46 46 47 47 47 47 47 48 49 49 49 49 49 49 49 49 49 49 49 49 49</pre>	42	#
44       #         45       # Lese das nächste verfügbare frame der Kamera ein; Es werden zwei         46       Werte zurückgegeben: boolean check (True, wenn das frame erfolgreich ausgelesen wurde, was hier nicht weiter geprüft wird; False sonst), frame (ein array, was die Werte des ausgelesenen frames enthält)         46       #         47       check, frame = cam_stream.read()         48       #         49       #         50       #         51       # Verarbeitung des eingelesenen Bildes         52       #         53       #         54       #         55       #         56       # Anzeigen des manipulierten frames         57       #         58       cv2.imshow('Demo', frame)         59       #         50       #         51       # Prüfen, ob eine Nutzereingabe getätigt wurde um das Programm zu beenden; 27 entspricht der Escape-Taste (siehe ASCII Tabelle: <a href="http://www.asciitable.com/">http://www.asciitable.com/</a> )         51       #         52       #         53       if cv2.weitKey(1) == 27:         64       break         65       #         66       #         67       Wenn das Program via Escape-Taste	43	while True:
<pre>45     # Lese das nächste verfügbare frame der Kamera ein; Es werden zwei     Werte zurückgegeben: boolean check (True, wenn das frame erfolgreich     ausgelesen wurde, was hier nicht weiter geprüft wird; False sonst),     frame (ein array, was die Werte des ausgelesenen frames enthält) 46     #</pre>	44	#
<pre>werte zuruckgegeben: boolean check (riue, wenn das programe erfolgreich ausgelesen wurde, was hier nicht weiter geprüft wird; False sonst), frame (ein array, was die Werte des ausgelesenen frames enthält) #</pre>	45	# lese das nächste verfügbare frame der Kamera ein; Es werden zwei
<pre>dusgetesen worde, was her netter geprof wird, have sonst;, frame (ein array, was die Werte des ausgelesenen frames enthält) # Check, frame = cam_stream.read() # Userabeitung des eingelesenen Bildes # Userabeitung des eingelesenen Bildes # Userabeitung des manipulierten frames # C2.imshow('Demo', frame) # Userabeitung des manipulierten frames # Userabeitung des des des des des des des des des des</pre>		Werte zuruckgegeben: boolean check (Irue, wenn das frame erfolgreich
<pre>46  #</pre>		frame (ein array, was die Werte des ausgelesenen frames enthält)
<pre>47 check, frame = cam_stream.read() 48 49 49 49 49 49 49 49 49 49 40 40 50 4 50 4</pre>	46	#
<pre>48 49 49 49 49 49 49 49 49 49 49 49 49 4 49 49</pre>	47	<pre>check, frame = cam_stream.read()</pre>
<pre>49 #</pre>	48	
<pre>50  # Verarbeitung des eingelesenen Bildes 51  # Verarbeitung des eingelesenen Bildes 52  # 53  # 54  # 55  # 56  # Anzeigen des manipulierten frames 57  # 58  cv2.imshow('Demo', frame) 59  # 60  # 61  # Prüfen, ob eine Nutzereingabe getätigt wurde um das Programm zu beenden; 27 entspricht der Escape-Taste (siehe ASCII Tabelle: http:// 62  # 63  if cv2.waitKey(1) == 27: 64  break 65  # 66  # 67  # Wenn das Programm via Escape-Taste beendet wurde muss die Kamera hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen werden 68  # 69  cam_stream.release() 70  cv2.destroyAllWindows()</pre>	49	#
<pre>S1 # Verarbeitung des eingelesenen Bildes 52 #</pre>	50	#
<pre>32 #</pre>	51	# Verarbeitung des eingelesenen Bildes
<pre>54 55</pre>	52	#
<pre>55  #</pre>	54	#
<pre>56  # Anzeigen des manipulierten frames 57  # 58  cv2.imshow('Demo', frame) 59 60  # 61  # Prüfen, ob eine Nutzereingabe getätigt wurde um das Programm zu beenden; 27 entspricht der Escape-Taste (siehe ASCII Tabelle: http:// www.asciitable.com/) 62  # 63  if cv2.waitKey(1) == 27: 64  break 65 66  # 67  # Wenn das Programm via Escape-Taste beendet wurde muss die Kamera hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen werden 68  # 69  cam_stream.release() 70  cv2.destroyAllWindows()</pre>	55	#
<pre>57  #</pre>	56	# Anzeigen des manipulierten frames
<pre>58 cv2.imshow('Demo', frame) 59 60 #</pre>	57	#
<pre>59 60 61 62 63 64 65 66 66 66 66 66 66 67 67 68 68 68 68 69 60 68 69 60 60 60 60 60 60 60 60 60 60 60 60 60</pre>	58	cv2.imshow('Demo', frame)
60 # # rüfen, ob eine Nutzereingabe getätigt wurde um das Programm zu beenden; 27 entspricht der Escape-Taste (siehe ASCII Tabelle: <u>http://www.asciitable.com/</u> ) 62 #	59	
<pre>beenden; 27 entermitter eingube getetigt wurde um dus Programm 2d beenden; 27 entspricht der Escape-Taste (siehe ASCII Tabelle: http:// www.asciitable.com/) f if cv2.waitKey(1) == 27: break if cv2.waitKey(1) == 27: break if eingube eingen if eingen if eingube eingen if eingube eingen if eingube eingen if eingen if</pre>	60	# # Drüfen ob eine Nutzereingebe setätist wurde um des Drosserre ev
62       www.asciitable.com/)         63       if cv2.waitKey(1) == 27:         64       break         65       ************************************	01	Herden: 27 entspricht der Escape-Taste (siehe ASCII Tabelle: http://
62 # 63 if cv2.waitKey(1) == 27: 64 break 65 66 # 67 # Wenn das Programm via Escape-Taste beendet wurde muss die Kamera hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen werden 68 # 69 cam_stream.release() 70 cv2.destroyAllWindows()		www.asciitable.com/)
63 if cv2.waitKey(1) == 27: 64 break 65 66 # 67 # Wenn das Programm via Escape-Taste beendet wurde muss die Kamera hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen werden 68 # 69 cam_stream.release() 70 cv2.destroyAllWindows()	62	#
64     break       65     *       66     #       67     # Wenn das Programm via Escape-Taste beendet wurde muss die Kamera hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen werden       68     #       69     cam_stream.release() cv2.destroyAllWindows()	63	<pre>if cv2.waitKey(1) == 27:</pre>
65 66 # 67 # Wenn das Programm via Escape-Taste beendet wurde muss die Kamera 67 hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen werden 68 # 69 cam_stream.release() 70 cv2.destroyAllWindows()	64	break
<pre>66 # 67 # Wenn das Programm via Escape-Taste beendet wurde muss die Kamera 68 hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen 68 # 69 cam_stream.release() 70 cv2.destroyAllWindows()</pre>	65	
<ul> <li># Wenn das Programm via Escape-Taste beendet wurde muss die Kamerä hierrüber informiert werden und noch geöffnete Fenster sauber geschlossen werden</li> <li>#</li></ul>	66	# //
retruer informert werden und noch geoffnete Fenster sauber geschlossen werden     #     Gam_stream.release()     cv2.destroyAllWindows()	67	# Wenn das Programm via Escape-Taste beendet wurde muss die Kamera
68         #           69         cam_stream.release()           70         cv2.destroyAllWindows()		nierruber informiert werden und noch geoffnete Fenster sauber geschlossen werden
69 cam_stream.release() 70 cv2.destroyAllWindows()	68	#
70 cv2.destroyAllWindows()	69	<pre>cam stream.release()</pre>
	70	cv2.destroyAllWindows()