



User Manual of Computer Vision Tools



درس: مبانی بینایی کامپیوتر

مدرس: جناب آقای دکتر سیفی پور

تهیه کنندگان: علی عابدینی، شایان کاشفی، پارسا ناصری

ترم پاییز 1402-1403

3	مقدمه:
3	بینایی کامپیوتر چیست؟
3	اهمیت ابزارهایی مانند Pillow و OpenCV
4	سایر ابزارهای مهم در بینایی کامپیوتر و پردازش تصویر part 1 :
6	ادامه ابزار های بینایی کامپیوتر part2:
8	ادامه ابزار های بینایی کامپیوتر part3:
10	تنظیم محیط:
10	نمای کلی کتابخانه:
10	ویژگی ها و قابلیت های OpenCV و Pillow
13	مقایسه سریع
13	آموزش های کاربردی:
13	بارگذاری و نمایش تصاویر
13	اعمال تبدیل های اولیه تصویر (به عنوان مثال، تغییر اندازه، برش).
13	اصول اولیه پردازش ویدیو (به عنوان مثال، فیلمبرداری از یک وب کم).
13	موضوعات پیشرفته:
13	استفاده از OpenCV برای تشخیص اشیا.
17	استفاده از Pillow برای دستکاری تصویر پیشرفته.
19	منابع:
19	منابع عمومی یادگیری پایتون
19	مبانی بینایی کامپیوتر
20	راه اندازی محیط
20	آموزش های کاربردی
20	دوره ها و راهنماهای کامل
20	مخازن کد و نمونه ها
21	کتاب و پی دی اف

مقدمه:

بینایی کامپیوتر چیست؟

بینایی کامپیوتر یک زمینه مطالعاتی در هوش مصنوعی (AI) است که بر روی توانمند ساختن رایانه‌ها برای رهگیری و استخراج اطلاعات از تصاویر و ویدیوها، به روشی مشابه بینایی انسان تمرکز دارد. این شامل توسعه الگوریتم‌ها و تکنیک‌هایی برای استخراج اطلاعات معنادار از ورودی‌های بصری و درک دنیای بصری است.

این آموزش Computer Vision برای افراد مبتدی و حرفه‌ای با تجربه طراحی شده است و مفاهیم اولیه و پیشرفته بینایی کامپیوتر از جمله عکاسی دیجیتال، پردازش تصویر ماهواره‌ای، تبدیل پیکسل، تصحیح رنگ، لایه برداری، فیلتر کردن، تشخیص و تشخیص اشیاء و تقسیم بندی تصویر را پوشش می‌دهد.

اهمیت ابزارهایی مانند OpenCV و Pillow



اهمیت OpenCV

کتابخانه جامع: بیش از 2500 الگوریتم بهینه‌شده برای پردازش تصویر و بینایی کامپیوتر.

کاربردهای گسترده: تشخیص چهره، ردیابی حرکت، پردازش ویدئو و یادگیری ماشین.

سرعت بالا: مناسب برای برنامه‌های بلادرنگ (Real-Time).

جامعه بزرگ: مستندات غنی و پشتیبانی قوی.



اهمیت Pillow

سادگی: ابزاری کاربرپسند برای پردازش تصویر پایه (تغییر اندازه، چرخش، فیلترگذاری).

پشتیبانی از فرمت‌ها: GIF, BMP, PNG, JPEG و TIFF.

ادغام آسان: با کتابخانه‌هایی مانند NumPy و OpenCV.

سایر ابزارهای مهم در بینایی کامپیوتر و پردازش تصویر part 1 :

TensorFlow و PyTorch :



این کتابخانه‌ها برای یادگیری عمیق و ساخت مدل‌های پیشرفته بینایی کامپیوتر مانند شبکه‌های عصبی کانولوشنال (CNN) استفاده می‌شوند.

ابزارهایی مانند TensorFlow Object Detection API و Torchvision برای تشخیص اشیا و پردازش تصویر بسیار محبوب هستند.

Keras:



یک کتابخانه سطح بالا برای ساخت مدل‌های یادگیری عمیق که معمولاً با TensorFlow استفاده می‌شود.

Dlib :



یک کتابخانه قدرتمند برای تشخیص چهره، ردیابی اشیا و تحلیل تصاویر.

Scikit-Image :



یک کتابخانه پایتونی برای پردازش تصویر که بر روی NumPy ساخته شده و الگوریتم‌های متنوعی برای فیلترگذاری، تقسیم‌بندی تصویر و استخراج ویژگی ارائه می‌دهد.

YOLO (You Only Look Once) : یک چارچوب محبوب برای تشخیص اشیا در زمان واقعی با دقت و سرعت بالا.

Fast.ai : یک چارچوب یادگیری عمیق که استفاده از مدل‌های پیشرفته بینایی کامپیوتر را ساده می‌کند.



: Matplotlib



برای نمایش و تجسم تصاویر و داده‌های پردازش شده استفاده می‌شود.

: NumPy



برای انجام عملیات ریاضی و ماتریسی روی تصاویر (تصاویر به صورت آرایه‌های NumPy نمایش داده می‌شوند).

: SimpleCV



یک کتابخانه ساده برای پردازش تصویر که برای مبتدیان مناسب است.

: MediaPipe



MediaPipe

یک چارچوب متن‌باز گوگل برای تشخیص چهره، دست و اشیاء در زمان واقعی.

ادامه ابزار های بینایی کامپیوتر part2:

Caffe

Caffe

Caffe یک چارچوب یادگیری عمیق است که در ابتدا در دانشگاه کالیفرنیا، برکلی توسعه یافت. این منبع باز است، تحت مجوز BSD. به زبان ++C و با رابط پایتون نوشته شده است.

BoofCV



BoofCV یک کتابخانه بینایی کامپیوتری متن باز است که به طور کامل در جاوا نوشته شده و تحت مجوز آپاچی 2.0 منتشر شده است.

CUDA



در محاسبات، CUDA یک پلت فرم محاسبات موازی اختصاصی و رابط برنامه نویسی کاربردی است که به نرم افزار اجازه می دهد تا از انواع خاصی از واحدهای پردازش گرافیکی برای پردازش سریع همه منظوره استفاده کند، رویکردی که محاسبات همه منظوره در GPU نامیده می شود. CUDA توسط Nvidia در سال 2006 ایجاد شد.

OpenVINO



OpenVINO یک ابزار نرم افزار منبع باز برای بهینه سازی و استقرار مدل های یادگیری عمیق است. برنامه نویسان را قادر می سازد تا راه حل های مقیاس پذیر و کارآمد هوش مصنوعی را با خطوط نسبتاً کمی کد توسعه دهند.



Amazon Rekognition یک نرم افزار مبتنی بر ابر به عنوان یک پلتفرم بینایی کامپیوتری خدماتی است که در سال 2016 راه اندازی شد. این نرم افزار به تعدادی از سازمان های دولتی ایالات متحده فروخته شده و مورد استفاده قرار گرفته است، از جمله اداره مهاجرت و گمرک ایالات متحده و پلیس اورلاندو، فلوریدا، و همچنین نهادهای خصوصی.

MATLAB



پردازش تصویر، ویدئو و سیگنال، یادگیری عمیق، یادگیری ماشین و سایر برنامه ها همگی می توانند از محیط برنامه نویسی MATLAB بهره مند شوند. این شامل یک جعبه ابزار بینایی کامپیوتری با ویژگی ها، برنامه ها و الگوریتم های متعدد است که به شما در ایجاد راه حل هایی برای مشکلات مربوط به بینایی کامپیوتر کمک می کند.

Viso Suite



Viso Suite یک پلتفرم کامل برای توسعه بینایی کامپیوتری، استقرار و نظارت است. بهترین پشته نرم افزاری در کلاس برای بینایی کامپیوتر، که پایه و اساس پلتفرم بدون کد است، شامل CVAT، OpenCV، OpenVINO، TensorFlow یا PyTorch است.

Clarifai



Clarifai پیشرو در هماهنگ سازی و توسعه هوش مصنوعی است و به سازمان ها، تیم ها و توسعه دهندگان کمک می کند تا هوش مصنوعی را در مقیاس بسازند، مستقر کنند، هماهنگ کنند و عملیاتی کنند.



Detectron2



OpenAI



DeepFace

محبوب ترین کتابخانه بینایی کامپیوتری منبع باز برای یادگیری عمیق تشخیص چهره در حال حاضر DeepFace است. این کتابخانه یک روش ساده برای استفاده از پایتون برای انجام بینایی کامپیوتری مبتنی بر تشخیص چهره ارائه می کند.

Detectron2

[Detectron2](#) یک کتابخانه پیشرفته برای تشخیص و تقسیم بندی اشیا است که توسط فیس بوک AI Research توسعه یافته است. از انواع وظایف بینایی کامپیوتری از جمله تشخیص اشیا، نمونه بندی و تقسیم بندی معنایی و تقسیم بندی پانوپتیک پشتیبانی می کند. ساخته شده بر روی چارچوب PyTorch، عملکرد و انعطاف پذیری بالایی را ارائه می دهد و آن را برای تحقیق و تولید مناسب می کند. معماری مدولار Detectron2 امکان سفارشی سازی و گسترش آسان را فراهم می کند و نیازهای بینایی کامپیوتری پیشرفته را برآورده می کند.

OpenAI – در [Eden AI](#) موجود است

OpenAI طیف وسیعی از قابلیت های بینایی رایانه را از طریق API خود ارائه می کند، از جمله طبقه بندی تصویر، تشخیص اشیا و تولید تصویر. API بر روی مدل های زبان پیشرفته OpenAI ساخته شده است و می تواند برای انجام کارهایی مانند شناسایی اشیاء در تصاویر، طبقه بندی محتوای تصویر و حتی تولید تصاویر جدید بر اساس توضیحات متنی استفاده شود. اگرچه به اندازه برخی دیگر از ارائه دهندگان بینایی کامپیوتر تخصصی نیست، راه حل های OpenAI می توانند افزودنی ارزشمند برای برنامه هایی باشند که به قابلیت های پردازش تصویر انعطاف پذیر و قدرتمند نیاز دارند.

[Raster Vision](#)

همه چیزهایی که برای ساخت و استقرار مدل های بینایی کامپیوتری نیاز دارید، از ابزارهای حاشیه نویسی خودکار گرفته تا راه حل های استقرار با کارایی بالا.

Google Cloud Vision API

google-research/
vision_transformer



Google Cloud Vision API از قابلیت‌های یادگیری ماشینی Google برای ارائه تجزیه و تحلیل و طبقه‌بندی دقیق تصاویر استفاده می‌کند. پشتیبانی آن از حاشیه نویسی تصویر قدرتمند و تعدیل محتوا، آن را به گزینه‌ای ایده‌آل برای برنامه‌هایی که نیاز به بینش بصری دقیق دارند تبدیل می‌کند.

مورد استفاده: برنامه‌های تجارت الکترونیک از Google Cloud Vision API برای استفاده از قابلیت‌های یادگیری ماشینی Google برای تشخیص تصویر استفاده می‌کنند.

مزایا: حاشیه نویسی دقیق تصویر، و بینش بصری قدرتمند.

معایب: وابستگی به اکوسیستم Google Cloud.

Microsoft Azure Computer Vision



Azure Computer Vision مجموعه‌ای جامع از ابزارها را برای تجزیه و تحلیل تصویر، OCR (تشخیص کاراکتر نوری) و درک ویدیو ارائه می‌دهد. API خدمات شناختی آن ادغام یکپارچه با زیرساخت ابری Azure را تسهیل می‌کند و از صنایع از خرده فروشی گرفته تا رسانه و سرگرمی پشتیبانی می‌کند.

مورد استفاده: زنجیره‌های خرده‌فروشی می‌توانند از Microsoft Azure Computer Vision برای پرداخت خودکار و مدیریت موجودی در زمان واقعی در فروشگاه‌های خرده‌فروشی استفاده کنند.

مزایا: ادغام یکپارچه با زیرساخت Azure، API خدمات شناختی قوی.

معایب: منحنی یادگیری خاص Azure برای کاربران جدید.

IBM Watson Visual Recognition



سرویس تشخیص بصری IBM Watson که توسط هوش مصنوعی پشتیبانی می‌شود، در درک و طبقه‌بندی محتوای بصری مانند تصاویر و ویدیوها عالی است. می‌توانید از آن برای تبدیل تصاویر به داده‌های کاربردی با برنامه‌های خود استفاده کنید. ویژگی‌های درجه‌سازمانی آن، از جمله آموزش مدل‌های سفارشی و استخراج ابرداده، بخش‌های مختلفی مانند مراقبت‌های بهداشتی، کشاورزی و تولید را تامین می‌کند.

مورد استفاده: تشخیص IBM Watson می‌تواند تصاویر پزشکی مانند اشعه ایکس، ام آر آی و سی تی اسکن را تجزیه و تحلیل کند تا به متخصصان مراقبت‌های بهداشتی در تفسیر داده‌های پزشکی کمک کند.

مزایا: ویژگی‌های درجه‌سازمانی و قابلیت‌های مبتنی بر هوش مصنوعی.

معایب: پیچیدگی یکپارچه‌سازی در محیط‌های غیر IBM.

تنظیم محیط:

1. نصب پایتون

`pip install opencv – python pillow`

2. نصب کتابخانه‌ها:

3. راه‌اندازی یک IDE (به عنوان مثال، VS Code، Jupyter Notebook).

نمای کلی کتابخانه:

ویژگی‌ها و قابلیت‌های OpenCV و Pillow

1. تشخیص چهره و اشیا

موبایل‌ها: قفل گوشی با تشخیص چهره (Face ID) در آیفون.

امنیت: تشخیص چهره در سیستم‌های نظارتی و دوربین‌های امنیتی.

خودروهای خودران: تشخیص عابرین پیاده، ماشین‌ها و علائم راهنمایی.

فروشگاه‌ها: تشخیص دزدی در فروشگاه‌ها با ردیابی حرکات مشکوک.

2. پردازش تصاویر پزشکی

تشخیص بیماری‌ها: تحلیل تصاویر MRI و CT اسکن برای تشخیص تومورها یا بیماری‌های قلبی.

جراحی‌های رباتیک: کمک به جراحان با ارائه تصاویر سه‌بعدی از اندام‌های داخلی.

پایش سلامت: تشخیص بیماری‌های پوستی با تحلیل تصاویر پوست.

3. واقعیت افزوده (AR) و سرگرمی

فیلترهای چهره: فیلترهای Snapchat و Instagram.

بازی‌ها: بازی‌های AR مانند Pokémon GO.

آموزش: شبیه‌سازی‌های آموزشی با استفاده از AR مثلاً آموزش آناتومی بدن.

4. صنعت و تولید

کنترل کیفیت: تشخیص عیوب محصولات در خطوط تولید (مثلاً ترک‌های روی شیشه یا قطعات معیوب).

رباتیک: هدایت ربات‌ها در انبارها برای شناسایی و جابه‌جایی کالاها.

مونتاژ خودکار: تشخیص قطعات و کمک به ربات‌ها در مونتاژ محصولات.

5. کشاورزی

تشخیص آفات: تحلیل تصاویر گیاهان برای تشخیص بیماری‌ها یا آفات.

برداشت خودکار: استفاده از ربات‌ها برای شناسایی و برداشت محصولات رسیده.

پایش مزارع: تحلیل تصاویر ماهواره‌ای برای بررسی سلامت محصولات.

6. ترافیک و حمل‌ونقل

تشخیص پلاک خودرو: سیستم‌های کنترل ترافیک و عوارضی.

مدیریت ترافیک: تحلیل ترافیک جاده‌ها با استفاده از دوربین‌های نظارتی.

پارکینگ‌های هوشمند: تشخیص فضای خالی در پارکینگ‌ها.

7. هنر و طراحی

ویرایش تصویر: نرم‌افزارهایی مانند Photoshop از الگوریتم‌های پردازش تصویر استفاده می‌کنند.

بازسازی تصاویر قدیمی: ترمیم و رنگ‌آمیزی عکس‌های قدیمی.

خلق آثار هنری: تولید آثار هنری دیجیتال با استفاده از هوش مصنوعی.

8. نظامی و دفاعی

شناسایی هدف: تشخیص خودروها، هواپیماها یا افراد در تصاویر ماهواره‌ای.
نظارت مرزی: استفاده از دوربین‌ها و سنسورها برای شناسایی حرکات مشکوک.
ربات‌های نظامی: هدایت ربات‌ها در مناطق جنگی.

9. محیط زیست

پایش حیات وحش: شناسایی و ردیابی حیوانات با استفاده از دوربین‌های تله‌ای.
تحلیل تصاویر ماهواره‌ای: بررسی تغییرات آب‌وهوا، جنگل‌زدایی و آلودگی.
شیلات: تشخیص ماهی‌ها و تحلیل جمعیت آن‌ها.

10. آموزش و پژوهش

تشخیص دستخط: تحلیل دستخط برای تصحیح خودکار اوراق امتحانی.
آموزش زبان اشاره: ترجمه زبان اشاره به متن با استفاده از دوربین.
پژوهش‌های علمی: تحلیل تصاویر میکروسکوپی یا داده‌های بصری در آزمایشگاه‌ها.

11. فروش و بازاریابی

تحلیل رفتار مشتری: ردیابی حرکات مشتریان در فروشگاه‌ها برای بهبود چیدمان محصولات.
تبلیغات هوشمند: نمایش تبلیغات متناسب با سن یا جنسیت مشتری.
تشخیص محصولات: سیستم‌های پرداخت خودکار در فروشگاه‌های بدون صندوق.

12. ورزش

تحلیل حرکات ورزشی: بررسی تکنیک‌های ورزشکاران با استفاده از ویدئو.

داوری خودکار: تشخیص خطاها یا امتیازات در مسابقات.

پخش زنده: استفاده از دوربین‌های هوشمند برای پخش زنده مسابقات.

مقایسه سریع

ابزار	بهترین استفاده	نقاط قوت	نقاط ضعف
OpenCV	پردازش تصویر و ویدئو بلادرنگ	جامع، سریع، جامعه بزرگ	پیچیده برای مبتدیان
Pillow	پردازش تصویر پایه	ساده، کاربر پسند	محدود به کاربردهای ساده
TensorFlow	یادگیری عمیق و مدل‌های پیشرفته	قدرتمند، انعطاف‌پذیر	نیاز به دانش یادگیری ماشین
YOLO	تشخیص اشیاء در زمان واقعی	سريع، دقيق	نیاز به سخت‌افزار قدرتمند
Dlib	تشخیص چهره و ردیابی اشیاء	دقيق	کند برای کاربردهای بزرگ مقیاس
Scikit-Image	پردازش تصویر علمی	ساده، مبتنی بر پایتون	محدود به کاربردهای پایه
MediaPipe	تشخیص دست و چهره بلادرنگ	بهینه‌شده برای بلادرنگ	محدود به کاربردهای خاص
Fast.ai	یادگیری عمیق برای مبتدیان	ساده‌سازی مدل‌های پیشرفته	انعطاف‌پذیری کمتر
SimpleCV	پردازش تصویر برای مبتدیان	بسیار ساده	محدود به کاربردهای ساده
NumPy	عملیات ریاضی روی تصاویر	پایه‌ای، انعطاف‌پذیر	فاقد قابلیت‌های پیشرفته

آموزش‌های کاربردی:

* در کنار این مستند یک فایل **Jupyter notebook** تهیه شده است که به صورت عملی نحوه اجرای این ابزارها را نشان می‌دهد. همچنین ویدیوایی از توضیحات آن نیز ضمیمه شده است. *

بارگذاری و نمایش تصاویر

اعمال تبدیل‌های اولیه تصویر (به عنوان مثال، تغییر اندازه، برش).

اصول اولیه پردازش ویدیو (به عنوان مثال، فیلمبرداری از یک وب‌کم).

موضوعات پیشرفته:

استفاده از OpenCV برای تشخیص اشیاء

مرحله 1: OpenCV را نصب کنید

قبل از شروع، اطمینان حاصل کنید که OpenCV نصب شده است. می‌توانید آن را با استفاده از pip نصب کنید:

```
pip install opencv-python
```

برای عملکرد بیشتر (به عنوان مثال، مازول های یادگیری عمیق)، نصب کنید:

```
pip install opencv-python-headless opencv-contrib-python
```

مرحله 2: OpenCV و کتابخانه های دیگر را وارد کنید

با وارد کردن OpenCV و سایر کتابخانه های ضروری شروع کنید:

```
import cv2
import numpy as np
```

مرحله 3: یک مدل از پیش آموزش دیده را بارگیری کنید

OpenCV مدل های از پیش آموزش دیده ای را برای تشخیص اشیا ارائه می دهد، مانند:

Haar Cascades: برای تشخیص چهره و اشیا.

مدل های یادگیری عمیق: برای تشخیص پیشرفته تر اشیا (مانند SSD، YOLO).

گزینه 1: استفاده از Haar Cascades

یک مدل Haar Cascade را برای تشخیص اشیا بارگذاری کنید:

```
# Load a pre-trained Haar Cascade model (e.g., for face detection)
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

گزینه 2: استفاده از مدل های یادگیری عمیق

یک مدل یادگیری عمیق از پیش آموزش دیده را بارگیری کنید (به عنوان مثال MobileNet SSD):

```
# Load pre-trained model and configuration file
net = cv2.dnn.readNetFromCaffe('deploy.prototxt', 'model.caffemodel')
```

مرحله 4: یک تصویر یا ویدیو را بارگیری کنید

می توانید از یک تصویر یا ویدیو برای تشخیص اشیا استفاده کنید.

برای تصاویر:

```
# Load an image
image = cv2.imread('image.jpg')
```

برای ویدیو:

```
# Open a video file or webcam
cap = cv2.VideoCapture('video.mp4') # or use 0 for webcam
```

مرحله 5: تشخیص اشیاء را انجام دهید

گزینه 1: استفاده از Haar Cascades

تشخیص اشیاء با استفاده از مدل Haar Cascade:

```
# Convert the image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Detect objects (e.g., faces)
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

# Draw rectangles around detected objects
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 2)
```

گزینه 2: استفاده از مدل های یادگیری عمیق

تشخیص اشیاء را با استفاده از مدل یادگیری عمیق انجام دهید:

```
# Prepare the image for the model
blob = cv2.dnn.blobFromImage(image, scalefactor=1.0, size=(300, 300), mean=(104.0, 177.0, 123.0))

# Pass the blob through the network
net.setInput(blob)
detections = net.forward()

# Loop over the detections
for i in range(detections.shape[2]):
    confidence = detections[0, 0, i, 2]

    # Filter out weak detections
    if confidence > 0.5:
        # Get the bounding box coordinates
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # Draw the bounding box
        cv2.rectangle(image, (startX, startY), (endX, endY), (0, 255, 0), 2)
```

مرحله 6: نمایش نتایج

نمایش تصویر یا ویدیو با اشیاء شناسایی شده:

```

# Display the image
cv2.imshow('Object Recognition', image)
cv2.waitKey(0)
cv2.destroyAllWindows()

# For video, use a loop
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Perform object detection on the frame
    # (Repeat Step 5 for each frame)

    # Display the frame
    cv2.imshow('Object Recognition', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the video capture object
cap.release()
cv2.destroyAllWindows()

```

مرحله 7: ذخیره خروجی (اختیاری)

ذخیره تصویر یا ویدیوی پردازش شده:

```

# Save the image
cv2.imwrite('output.jpg', image)

# Save the video
out = cv2.VideoWriter('output.mp4', cv2.VideoWriter_fourcc(*'mp4v'), 30, (frame_width, frame_height))
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
    out.write(frame)
out.release()

```

References

- OpenCV Haar Cascades: [Haar Cascades Documentation](#)
- OpenCV Deep Learning Module: [DNN Module Documentation](#)
- OpenCV Video I/O: [Video I/O Documentation](#)

استفاده از Pillow برای دستکاری تصویر پیشرفته

مرحله 1: Pillow را نصب کنید

قبل از شروع، مطمئن شوید که Pillow نصب شده است. می توانید آن را با استفاده از pip نصب کنید:

```
pip install pillow
```

مرحله 2: Pillow را وارد کنید

با وارد کردن ماژول های لازم از Pillow شروع کنید:

```
from PIL import Image, ImageFilter, ImageEnhance, ImageDraw, ImageFont
```

مرحله 3: یک تصویر را بارگذاری کنید

یک تصویر را با استفاده از متد *Image.open()* بارگیری کنید:

```
# Load an image
image = Image.open('image.jpg')
```

مرحله 4: دستکاری تصویر پایه

عملیات اساسی مانند تغییر اندازه، برش و چرخش را انجام دهید:

```
# Resize the image
resized_image = image.resize((300, 300))

# Crop the image
cropped_image = image.crop((100, 100, 400, 400)) # (left, top, right, bottom)

# Rotate the image
rotated_image = image.rotate(45) # Rotate by 45 degrees
```

مرحله 5: اعمال فیلترها

Pillow فیلترهای مختلفی را برای بهبود تصویر ارائه می دهد. آنها را با استفاده از ماژول *ImageFilter* اعمال کنید:

```
# Apply a Gaussian blur
blurred_image = image.filter(ImageFilter.GaussianBlur(radius=5))

# Apply edge enhancement
edge_enhanced_image = image.filter(ImageFilter.EDGE_ENHANCE)

# Apply a sharpen filter
sharpened_image = image.filter(ImageFilter.SHARPEN)
```

مرحله 6: ویژگی های تصویر را تنظیم کنید

از ماژول *ImageEnhance* برای تنظیم روشنایی، کنتراست و سایر خصوصیات استفاده کنید:

```
# Enhance brightness
enhancer = ImageEnhance.Brightness(image)
brightened_image = enhancer.enhance(1.5) # Increase brightness by 50%

# Enhance contrast
enhancer = ImageEnhance.Contrast(image)
contrast_enhanced_image = enhancer.enhance(2.0) # Double the contrast

# Enhance sharpness
enhancer = ImageEnhance.Sharpness(image)
sharpness_enhanced_image = enhancer.enhance(2.0) # Double the sharpness
```

مرحله 7: متن را به تصاویر اضافه کنید

از ماژول *ImageDraw* برای افزودن متن به تصاویر استفاده کنید:

```
# Create a drawing object
draw = ImageDraw.Draw(image)

# Define font (you may need to provide the path to a .ttf file)
font = ImageFont.truetype('arial.ttf', size=40)

# Add text to the image
draw.text((10, 10), "Hello, Pillow!", fill="white", font=font)
```

مرحله 8: دستکاری پیشرفته با ماسک

ایجاد و اعمال ماسک برای جلوه های پیشرفته:

```
# Create a mask (e.g., a black and white image)
mask = Image.new('L', image.size, 128) # 128 = gray

# Apply the mask to the image
masked_image = Image.composite(image, Image.new('RGB', image.size, 'red'), mask)
```

مرحله 9: تصویر پردازش شده را ذخیره کنید

تصویر دستکاری شده را با استفاده از متد *save()* ذخیره کنید:

```
# Save the image
image.save('output.jpg')
```

مرحله 10: نمایش تصویر

تصویر را با استفاده از متد *show()* نمایش دهید:

```
# Display the image
image.show()
```

References

- Pillow Documentation: [Pillow Official Docs](#)
 - Image Module: [Image Module Docs](#)
 - ImageFilter Module: [ImageFilter Docs](#)
 - ImageEnhance Module: [ImageEnhance Docs](#)
 - ImageDraw Module: [ImageDraw Docs](#)
-

منابع:

منابع عمومی یادگیری پایتون

- اسناد رسمی پایتون
- <https://docs.python.org/3> - منبع رسمی برای نحو، مازول ها و توابع پایتون.
- آموزش های تعاملی پایتون
- <https://realpython.com> - آموزش های مبتدی تا پیشرفته پایتون را ارائه می دهد.

مبانی بینایی کامپیوتر

- مقدمه ای بر بینایی کامپیوتر
- کامپیوتر ویژن چیست؟ ([GeeksforGeeks](#))
- اسناد رسمی OpenCV
- [OpenCV Docs](#) - مستندات جامع برای توابع و آموزش های OpenCV.
- مستندات Pillow
- [Pillow \(PIL Fork\) Docs](#) - راهنمای رسمی برای دستکاری تصویر با استفاده از Pillow.

- ابزار های بینایی:

- [Eden AI](#)
- [align minds](#)
- [Mark tech post](#)
- [labellerr](#)

- کانال های یوتیوب
- Sentdex - عالی برای آموزش بینایی کامپیوتر مبتنی بر پایتون.

Corey Schafer - آموزش مقدماتی پایتون و OpenCV.

راه اندازی محیط

- نصب پایتون : [پایتون را دانلود کنید](#)
- نصب پکیج ها : [با استفاده از pip](#)
- نصب نوت بوک Jupyter : [Jupyter Docs](#)

آموزش های کاربردی

- پروژه های OpenCV

[تشخیص چهره در زمان واقعی](#)

[مبانی پردازش تصویر](#)

- نمونه های Pillow

[کار با تصاویر](#)

- بینایی کامپیوتر و یادگیری عمیق

[OpenCV را یاد بگیرید](#) - آموزش هایی در مورد موضوعات پیشرفته مانند تشخیص و ردیابی اشیاء.

دوره ها و راهنماهای کامل

1. کورسرا : [مقدمه ای بر بینایی کامپیوتر](#)
2. کادل : [پایتون برای کامپیوتر ویژن](#)
3. FreeCodeCamp : [بینایی کامپیوتر با پایتون \(یوتیوب\)](#)

مخازن کد و نمونه ها

- پروژه های OpenCV GitHub

[OpenCV GitHub](#)

- Pillow GitHub Repo

[Pillow GitHub](#)

- پروژه های نمونه

[مخازن awesome کامپیوتر ویژن](#)

[نمونه های پردازش تصویر](#)

کتاب و پی دی اف

- برنامه نویسی کامپیوتر ویژن با پایتون توسط جان اریک سولم

[در O'Reilly موجود است](#)

- آموزش OpenCV-Python (PDF و کتاب های الکترونیکی)

[OpenCV PDF را یاد بگیرید](#)

- پایتون و OpenCV عملی توسط آدریان روزبروک

[کتاب های PyImageSearch](#)

- [Computer Vision and Image Processing: A Practical Approach Using Cviptools with Cdrom](#)