

## **Approaches in Using Python for Image Processing**

Ass. Prof. Dr. Stoicho Stoev  
University of Economics - Varna, Varna, Bulgaria  
s.stoev@ue-varna.bg

### **Abstract**

*Image processing is an increasingly widespread field in computer science. On the other hand, it is increasingly used both in modern business and in social communication. We will look at image manipulation methods, emphasizing approaches for comparing them. We will offer ways to process images included in popular specialized applications. We emphasize the use of the popular Python programming language, which is becoming more and more popular due to its ease of use and large range of tools. We will look at some of the image processing and analysis libraries used in Python.*

*Keywords: Python , images, image processing*

*JEL Code: C88*

*DOI: 10.56065/IJUSV-ESS/2023.12.3.122*

### **Въведение**

Данните са най-ценният ресурс, с който разполагат фирмите в днешната дигитална ера и голяма част от тези данни се състоят от изображения. Обработката на изображения е процес на трансформирането им в цифрови данни преди извършване на специални операции върху тях, което позволява да се осъществят множество функционалности, полезни за бизнеса.

Изображенията заемат все по-голям дял в съвременните информационни технологии. Те са неделима част при разработката на мобилни приложения, Уеб-приложения, социални мрежи, системи за сигурност и др. Някои от областите на приложение са строго специализирани като медицински изследвания, разпознаване на образи, геосистеми и др. От друга страна интерес представляват програмите за масово ползване за обработка на изображения. Поради широкото си разпространение са създадени широка гама от приложения за тяхното манипулиране. Тези от тях с безплатно използване, често прилагат алгоритми за преобразуване на изображения, реализирани на Python .

Python е език за програмиране с нарастваща популярност. Разполага с разнообразни библиотеки и инструменти за обработка на изображения. Това го прави подходящ за интегриране в приложенията за манипулиране на изображения, компютърните игри или социални мрежи. Някои графични приложения предлагат цялостна платформа за използване, настройка и тестване на различни алгоритми, реализирани на Python .

### **Преглед на литературата**

Манипулирането на изображения се отнася до автоматична обработка, промяна, анализ и интерпретация на изображения с помощта на алгоритми и кодове на компютър. Има приложения в много дисциплини и области на науката и технологиите като телевизия, фотография, роботика, дистанционно наблюдение, медицинска диагностика и промишлена инспекция. Сайтовете за социални мрежи като Facebook и Instagram, с които сме свикнали в ежедневието си и където качваме хиляди изображения всеки ден, са типични примери за индустрии, които трябва да използват/иновират много алгоритми за обработка на изображения, за да повишат атрактивността на информацията, която качване (Sandipan Deu, 2018).

Python е добър избор при обработка на изображения (Jason M. Kinser, 2018), защото:

1. Той е свободно достъпен и с отворен код. Тъй като това е безплатен софтуер, всички имат достъп до него без никакви ограничения.

2. Осигурява предварително опаковани инсталации, налични за всички основни платформи без заплащане.

3. Това е предпочитаният език от високо ниво за учени и инженери.

4. Признат е като може би най-лесният език за научаване от не-програмисти.

Освен това, стандартният Python (Niklas Dellby, 2019), поддържа и координира събирането на данни, съхранението, обработката, анализа и визуализацията, но оставя обработката чрез по-сложни алгоритми, на оптимизирани библиотеки на Python .

За значението на различните библиотеки на Python при решаване на конкретни задачи (Samira Gholizadeh, 2022) има роля и изследванията им и областите на приложения. Това би помогнало на разработчиците да се насочат към необходимата библиотека без да влагат време и ресурси в изследването им.

При обработка на изображения посредством Python има възможност за използване на широка гама от библиотеки, насочени към различни процеси: от извличане на данни за изображения, трансформиране на данните с някои алгоритми, използване библиотечни функции за предварителна обработка, подобряване, възстановяване, представяне (с дескриптори), сегментиране, класифициране и разпознаване (обекти), за да се анализират, разбират и тълкуват по-добре данните (Sandipan Dey, 2018).

### **Начини за обработка на изображения.**

В зависимост от целите, които имаме при обработката на изображения, може да дефинираме няколко основни метода. Те зависят от това дали ще манипулираме цялото изображение или част от него. Дали ще използваме алгоритми за разпознаване на изображения или искаме просто да го анализираме. Приложението на Python функции при обработване на изображения може да се разделят в няколко направления:



Фигура 1. Цялостна обработка на изображения. (Gimp)

**Цялостно изображение.** Какъвто и алгоритъм да използваме, той касае цялата матрица на самото изображение. Тоест не се ограничаваме върху отделна област. Това позволява да се прилагат множество филтри за трансформиране оригиналното изображение към ново с желани параметри. Пример за цялостна обработка може да бъде преобразуване на цветно в сиво изображение (фиг.1)

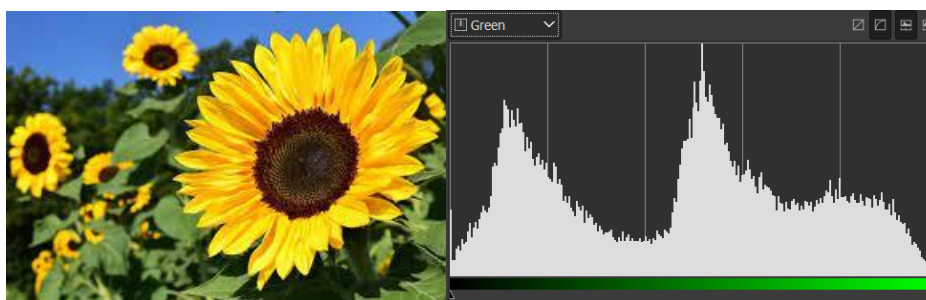
**Частично изображение.** При този метод на обработка се избира част от изображението, което вследствие се манипулира. Алгоритмите за обработка се прилагат след предварително прехвърляне на част от изображението в друга матрица. Изборът на обработваната част може да се извърши ръчно, тоест потребителят да я определи или чрез използване на специален алгоритъм за автоматично определяне на рамките на областта. Например по сходни характеристики на изображението (фиг.2).



Фигура 2. Частична обработка на изображения. (Gimp)

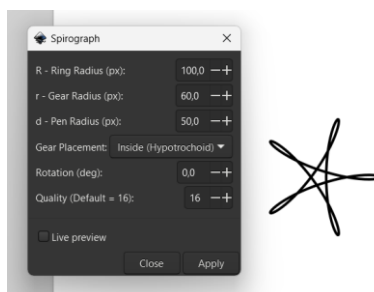
**Преобразуване.** Този метод се използва при компютърното зрение. Целта е да се преобразува визуалната информация в данни. Пример за използване на този метод е при разпознаване на текст или номера на МПС. Разчита се, алгоритъмът за обработка на изображения да определи контурите на отделните елементи, които вследствие да трансформира в данни. Този подход се използва и в обратната посока, например когато един текст се преобразува в растерно изображение.

**Характеристики.** Съществуват функционалности в различните библиотеки на Python, предназначени не за промяна на самото изображение, а получаване на информация за него. Такива са алгоритмите за анализ на изображението и представяне на неговата хистограма (фиг.3).



Фигура 3. Хистограма на разпределение на зеленият цвят. (Gimp)

**Генериране.** При този подход се използват алгоритми за създаване на нова част към изображението на базата на конкретен математичен метод. В това направление обикновено функциите на Python за генериране на определено изображение, добавя възможност за настройка на алгоритъма. Пример за този подход е показан на фиг.4.



Фигура 4. Генериране на изображение, чрез функция на Python в Inkscape

В изброените направления за обработка на изображения са обхванати най-често използваните манипулации в програмите за тяхната обработка.

**Сравнение.** Все по-популярен функционалност при обработката на изображения с цел намиране на сходни или еднакви изображения. Наличието на огромен брой снимки както в

личните албуми, социалните мрежи или изобщо в мрежата, налага все по-често да при управлението на съдържанието да се използва сравнение за еднаквост или подобност на определена информация. Сходството на изображението може да се разглежда като числено представяне на това колко еднакви са две изображения по отношение на тяхното визуално съдържание. Има няколко измерения, по които изображенията могат да бъдат подобни, като цвят, форма, текстура и композиция. Използват се различни математически и изчислителни методи за количествено определяне на тези прилики, което ни позволява да сравняваме и категоризираме изображенията ефективно.

### **Подходи за намиране на сходство на изображения**

Способността за измерване на приликата на изображението е жизненоважен компонент на множество приложения в днешния свят, управляван от визуално. Сходството на изображението може да се използва при разпознаването на обекти, за да се съпостави даден обект с известна база данни. Алгоритъм за сходство на изображения се използва за идентифициране на хора чрез сравняване на лицата им с база данни.

Най-разпространените подходи за сравняване на изображения са следните:

- Подходи, базирани на хистограма
- Индекс на структурно сходство (SSIM)
- Подходи, базирани на функции
- Подходи, базирани на дълбоко обучение

#### ***Подходи, базирани на хистограма***

Хистограмите улавят разпределението на стойностите на пикселите в изображението. Като сравнявате хистограмите на две изображения, можете да измерите тяхната прилика. Базираните на хистограма алгоритми за съпоставяне на изображения се опитват да измерят сходството в съдържанието чрез своите хистограми между моделно изображение и всякакви изображения в базата данни, т.е. целеви изображения, за да класифицират правилно или да извлекат изображенията.

OpenCV библиотеката на Python предоставя инструменти за изчисляване и сравняване на хистограми.

#### ***Индекс на структурно сходство (SSIM)***

SSIM (Structural Similarity Index Method) е широко използван показател, който оценява структурното сходство между две изображения. Той взема предвид яркостта, контраста и структурата, като дава оценка между -1 (различни) и 1 (идентични). Библиотеката scikit-image в Python предлага внедряване на SSIM.

Мярката за индекс на структурно сходство (SSIM) е метод за предсказване на възприеманото качество на цифрова телевизия и кинематографични изображения, както и други видове цифрови изображения и видео. SSIM се използва за измерване на приликата между две изображения. Индексът SSIM е пълна референтна метрика, с други думи, измерването или прогнозирането на качеството на изображението се основава на първоначално некомпесирано изображение или изображение без изкривяване.

Основният недостатък на подхода SSIM в сравнение с подхода на хистограмата е, че изображенията трябва да са с едно и също измерение. Можем да направим изваждане на фона и премахване на прозрачността от изображенията, за да подобрим резултата за сходство.

#### ***Подходи, базирани на функции***

Тези методи извличат конкретни характеристики от изображенията, като ръбове, ъгли или ключови точки. Техники като Scale-Invariant Feature Transform (SIFT) и Speeded-Up Robust Features (SURF) идентифицират отличителни точки в изображенията, които след това могат да бъдат сравнени между различни изображения.

Методът за сравняване на изображения, базиран на функции, сравнява различни изображения и цели да намерите подобни двойки изображения, като използва набор от локални характеристики за всяко тях. Наборът от данни за сравнение са по компактни от самото изображение и дават възможност за съхранението им отделна база от данни. Този подход позволява повишаване на ефективността и бързодействието в процеса на сравняване.

Базираното на функции сравнение на изображения съдържа три стъпки:

1. Откриване на характеристики,
2. Изчисляване на дескриптори
3. Сравнение на характеристики.

Библиотеката `opencv-Python` може да се използва за SIFT и SURF.

### ***Подходи, базирани на дълбоко обучение***

Дълбокото обучение революционизира задачите за сходство на изображения. Използвайки предварително обучени CNN<sup>1</sup> (convolutional neural networks) като ResNet, VGG и Inception, можете да извлечете характеристики от изображения.

CLIP (Contrastive Language-Image Pre-Training) от openAI е впечатляващ мултимодален нулев класификатор на изображения, който постига впечатляващи резултати в широк диапазон от области без фина настройка. Той прилага последните постижения в широкомащабните трансформатори като GPT-3.

За да се подобри точността, може предварително да се обработи изображението.

Изборът на подход, зависи от конкретната задача и ресурсите, необходими за нейното решаване.

### **Библиотеки, използвани при обработка на изображения на Python**

Едно от предимствата на Python е наличието на широка гама от готови библиотеки, предназначени за решаване на проблеми в различни области. В този смисъл разработчиците на софтуер могат да се съсредоточат в решаване на конкретната задача, без необходимостта от разработка на необходимите инструменти.

Поради големият брой библиотеки разработени за Python, ще се концентрираме върху най-популярните от тях, свързани с обработката на изображения:

**OpenCV** е библиотека за компютърно зрение с отворен код. Тази библиотека се състои от около 2000+ оптимизирани алгоритми, които са полезни за компютърно зрение и машинно обучение. Има няколко начина, по които можете да използвате `opencv` при обработката на изображения, някои от които са изброени по-долу:

Преобразуване на изображения от едно цветово пространство в друго, например между BGR и HSV, BGR и сиво и т.н.

Извършване на прагово определяне на изображения, като просто определяне на прагове, адаптивно определяне на прагове и др.

Изглаждане на изображения, като прилагане на персонализирани филтри към изображения и замъгляване на изображения.

Извършване на морфологични операции върху изображения.

**Scikit**. Това е библиотека с отворен код, използвана за предварителна обработка на изображения. Тя използва машинно обучение с вградени функции и може да извършва сложни операции върху изображения само с няколко функции.

Scikit-Image, който използва масиви NumPy като обекти на изображения, предлага много различни алгоритми за сегментиране, манипулиране на цветовото пространство, геометрична трансформация, анализ, морфология, откриване на характеристики и много други. Някои операции, които могат да се извършват с помощта на `scikit image` са:

---

<sup>1</sup> Конволюционна невронна мрежа (CNN): Вид изкуствена невронна мрежа, която се използва главно за анализ на изображения.

- Можете да приложите операции за прагове, използвайте метода `try_all_threshold()` на изображението. Той ще използва седем глобални алгоритъма за прагове.
- За да реализирате откриване на ръбове, използвайте метода `sobel()`.
- За да приложите гаусово изглаждане, използвайте метода `gaussian()` в модула за филтри .
- За да завъртите изображението, използвайте функцията `rotate()` под модула за трансформация .
- За да промените мащаба на изображението, използвайте функцията `rescale()` от модула за трансформация .
- За да приложите морфологични операции, използвайте функцията `binary_erosion()` и `binary_dilation()` в модула за морфология .

**PIL** (Python Imaging Library). Друга библиотека с отворен код за задачи за обработка на изображения, Pillow е разширена версия на PIL (Python Imaging Library). С Pillow можете да извършвате много процеси в обработката на изображения като точкови операции, филтриране и манипулиране.

Pillow е една от най-добрите библиотеки за работа с изображения, благодарение на поддръжката си за широк набор от формати на изображения. Библиотеката за обработка на изображения е лесна за използване, което я прави един от най-разпространените инструменти за специалисти по данни, които работят с изображения.

Тя поддържа широк набор от формати на изображения като PPM, JPEG, TIFF, GIF, PNG и BMP.

**NumPy**. С тази библиотека можете да изпълнявате прости техники за изображения, като обръщане на изображения, извличане на функции и тяхното анализиране.

Докато NumPy е библиотека на Python с отворен код, използвана за числен анализ, тя може да се използва и за задачи за обработка на изображения като изрязване на изображения, манипулиране на пиксели, маскиране на стойности на пиксели и др. NumPy съдържа матрица и многомерни масиви като структури от данни.

NumPy може да се използва и за подпомагане на намаляването на цветовете, бинаризацията, поставяне със срез, положителна или отрицателна инверсия и много други функции. Изображенията също могат да се считат за съставени от масиви, което позволява на NumPy да изпълнява различни задачи за обработка на изображения.

Ето някои от основните акценти на NumPy:

- Компактно съхранение на данни
- Високоскоростна обработка на масиви
- Помага с много функции
- Съвместимост на данните с други библиотеки

**Mahotas**. Това е една от най-добрите библиотеки за обработка на изображения в Python, която първоначално е проектирана за био-изображения. Mahotas позволява на разработчиците да се възползват от разширени функции като локални двоични модели. Той може да изчислява 2D и 3D изображения чрез своя модул `mahotas.features.haralick` и извлича информация от снимки, за да извърши разширена обработка на изображения.

Mahotas има много популярни функции като водораздел, изчисления на изпъкнали точки, морфологична обработка и съвпадение на шаблони. Има над 100 функционалности за възможности за компютърно зрение.

Ето някои от основните акценти на Mahotas:

- Над 100 функционалности за компютърно зрение
- Разширени функции за обработка
- Изчислява 2D и 3D изображения
- Постоянно добавяне на нови функционалности

Въпреки, че ползването на конкретна библиотека за манипулиране на изображения, предполага допълнително изучаване на нейната функционалност и правила, предимствата са неоспорими. Разработчикът не трябва да изучава математическите основи на отделните алгоритми, а само върху практическото им ползване.

### **Заклучение**

Използването на подходящи алгоритми и по-важното, лесната им реализация разширява възможностите на приложение на информационните технологии. Обработката на изображения ни позволява да трансформираме и манипулираме много изображения и да извличаме полезна информация от тях. Следователно тя, може да намери приложения в почти всяка област.

Благодарение на своите възможности за манипулиране на големи масиви от данни, Python намира все по-голямо приложение при графичните системи. Следователно важността на избор на подход и подходящи библиотеки, става все по актуално. Всяка задача изисква индивидуален подход за оптимално решение. Поради ниската защитеност на изходния код, засега масово се използва в некомерсиални приложения.

### **References**

1. Chen, J., Jiang, T., Yu, D., & Hu, H. (2023). Pattern-based circular reference detection in Python . *Science of Computer Programming*. 227, 102932.
2. Dey, S. (2018). *Hands-On Image Processing with Python : Expert techniques for advanced image analysis and effective interpretation of image data*. Packt Publishing Ltd., 178934185X, 9781789341850.
3. Gholizadeh, S. (2022). Top Popular Python Libraries in Research. *Journal of Robotics and Automation Research*. 3(2).
4. Kinser, J. M. (2018). *Image Operators: Image Processing in Python* , CRC Press, 0429835949, 9780429835940.
5. Meyer, C., Dellby, N., Hachtel, J. A., Lovejoy, T., Mittelberger, A., Krivanek, O. (2019). Nion Swift: Open Source Image Processing Software for Instrument Control, Data Acquisition, Organization, Visualization, and Analysis Using Python . *Microscopy and Microanalysis*. 25(S2), 122–123.
6. Pajankar, A. (2019). *Python 3 Image Processing: Learn Image Processing with Python 3, NumPy, Matplotlib, and Scikit-image*, BPB Publications, 9388511727, 9789388511728.
7. Singh, V. K., Soni, S. K., Yadav, N. D., & Chandra, P. (2023). A system for cartoonifying an image using Python . *A Jurnal of Physical Sciences, Engeneering and Tegnology*, volume 14, 84-100.