**پروژه تشخیص دایره‌های قرمز با OpenCV در جاوا**

**معرفی پروژه**

این پروژه یک سرویس ساده برای تشخیص دایره‌های قرمز در تصاویر با استفاده از کتابخانه OpenCV و جاوا است. این سرویس تصاویر را از طریق یک API دریافت می‌کند، دایره‌های موجود در تصویر را شناسایی کرده و اگر دایره قرمز باشد، دور آن یک دایره سبز می‌کشد. همچنین تعداد دایره‌های قرمز شناسایی‌شده را در گوشه تصویر نمایش می‌دهد و تصویر پردازش‌شده را در روت پروژه ذخیره می‌کند.

این پروژه از الگوریتم Hough Circles برای تشخیص دایره‌ها و بررسی رنگ در فضای رنگی BGR استفاده می‌کند. مناسب برای پروژه‌های پردازش تصویر و یادگیری مفاهیم OpenCV در جاوا.

**پیش‌نیازها**

برای اجرای این پروژه، موارد زیر را آماده کنید:

* **جاوا**: نسخه 17 یا بالاتر
* **Maven**: برای مدیریت وابستگی‌ها
* **OpenCV**: نسخه 4.9.0 یا بالاتر
  + کتابخانه native OpenCV (مثل opencv\_java490.dll یا libopencv\_java490.so) باید در مسیر JVM قرار گیرد.
  + مسیر را با -Djava.library.path=/path/to/opencv/lib تنظیم کنید.
* **Spring Boot**: نسخه 3.5.0 (وابستگی‌ها در pom.xml تعریف شده‌اند)
* **ابزار تست API**: مانند Postman برای ارسال درخواست‌ها

**وابستگی‌ها (وارد شده در pom.xml)**

* org.openpnp:opencv:4.9.0-0
* org.springframework.boot:spring-boot-starter-web

**ساختار کد**

این کد در لایه سرویس (ImageProcessingService.java) پیاده‌سازی شده و شامل سه متد اصلی است که در ادامه توضیح داده می‌شوند.

**1. متد processRedCircles**

**توضیحات**

این متد تصویر ورودی را دریافت می‌کند، دایره‌های موجود را با الگوریتم Hough Circles شناسایی می‌کند، دایره‌های قرمز را فیلتر می‌کند، دور آن‌ها دایره سبز می‌کشد و تعداد دایره‌های قرمز را در گوشه تصویر نمایش می‌دهد. سپس تصویر نهایی را در روت پروژه ذخیره می‌کند.

**امضا**

public void processRedCircles(MultipartFile file) throws Exception

**پارامترها**

* file: فایل تصویر ورودی (نوع MultipartFile) که از درخواست HTTP دریافت می‌شود. فرمت‌های قابل پشتیبانی: JPG, PNG و غیره.

**خروجی**

* خروجی ندارد (void). تصویر پردازش‌شده با نام processed\_image.jpg در روت پروژه ذخیره می‌شود.

**مراحل**

1. تصویر ورودی به فرمت Mat تبدیل می‌شود.
2. تصویر به خاکستری تبدیل می‌شود.
3. دایره‌ها با الگوریتم Hough Circles شناسایی می‌شوند.
4. دایره‌های قرمز فیلتر می‌شوند.
5. دور دایره‌های قرمز یک دایره سبز رسم می‌شود.
6. تعداد دایره‌های قرمز در گوشه تصویر نمایش داده می‌شود.
7. تصویر نهایی ذخیره می‌شود.

**2. متد isRedCircle**

**توضیحات**

این متد بررسی می‌کند که آیا دایره‌ای با مرکز مشخص قرمز است یا خیر. یک ناحیه کوچک (3x3 پیکسل) اطراف مرکز دایره را بررسی کرده و میانگین مقادیر رنگی (BGR) را محاسبه می‌کند.

**امضا**

private boolean isRedCircle(Mat image, int xCent, int yCent)

**پارامترها**

* image: تصویر اصلی در فرمت Mat (فضای رنگی BGR).
* xCent: مختصات X مرکز دایره (عدد صحیح).
* yCent: مختصات Y مرکز دایره (عدد صحیح).

**خروجی**

* boolean: اگر دایره قرمز باشد true و در غیر این صورت false.

**معیار تشخیص قرمز**

* مقدار کانال قرمز (Red) باید بیشتر از 170 باشد.
* مقادیر کانال‌های سبز (Green) و آبی (Blue) باید کمتر از 60 باشند.

**3. متد bufferedImageToMat**

**توضیحات**

این متد تصویر ورودی را از فرمت BufferedImage (جاوا) به فرمت Mat (OpenCV) تبدیل می‌کند تا برای پردازش با OpenCV آماده شود.

**امضا**

private Mat bufferedImageToMat(BufferedImage bi)

**پارامترها**

* bi: تصویر ورودی در فرمت BufferedImage.

**خروجی**

* Mat: تصویر تبدیل‌شده در فرمت OpenCV با نوع CV\_8UC3 (تصویر رنگی 8 بیتی با 3 کانال).

**نحوه اجرا**

1. **تنظیم پروژه**:
   * پروژه را در یک IDE (مثل IntelliJ) باز کنید.
   * مطمئن شوید pom.xml شامل وابستگی‌های لازم (OpenCV و Spring Boot) است.
   * مسیر فایل‌های native OpenCV را در تنظیمات JVM مشخص کنید:

-Djava.library.path=/path/to/opencv/lib

1. **اجرای برنامه**:
   * برنامه Spring Boot را اجرا کنید.
   * API در آدرس /api/image/process در دسترس خواهد بود (کنترلر جداگانه باید تنظیم شود).
2. **ارسال درخواست**:
   * از Postman یا ابزار مشابه استفاده کنید.
   * یک درخواست POST به /api/image/process ارسال کنید.
   * فایل تصویر را به‌عنوان multipart/form-data با کلید file آپلود کنید.
3. **بررسی خروجی**:
   * تصویر پردازش‌شده با نام processed\_image.jpg در روت پروژه ذخیره می‌شود.
   * اگر دایره‌های قرمز شناسایی شوند، دور آن‌ها دایره سبز کشیده شده و تعداد آن‌ها در گوشه تصویر نمایش داده می‌شود.

**نکات مهم**

* **پارامترهای HoughCircles**:
  + dp=1: نسبت وضوح تصویر.
  + minDist=10: حداقل فاصله بین مراکز دایره‌ها.
  + param1=1.5, param2=0.7: پارامترهای حساسیت الگوریتم Hough.
  + minRadius=10, maxRadius=25: محدوده شعاع دایره‌ها.
  + این مقادیر را می‌توانید بر اساس نیاز تنظیم کنید.
* **تشخیص رنگ**:
  + معیار تشخیص رنگ قرمز ممکن است برای تصاویر با نورپردازی مختلف نیاز به تنظیم داشته باشد.
  + می‌توانید مقادیر آستانه (170, 60, 60) را تغییر دهید.
* **ذخیره‌سازی**:
  + تصویر خروجی در روت پروژه ذخیره می‌شود. برای تغییر مسیر، outputPath را در متد processRedCircles تغییر دهید.

**نمونه تصویر ورودی و خروجی**

**تصویر ورودی**

تصویری با دایره‌های قرمز روی پس‌زمینه سفید یا خاکستری ایده‌آل است. مثلاً یک دایره قرمز با شعاع 15 پیکسل.

**تصویر خروجی**

* دایره‌های قرمز با دایره سبز احاطه می‌شوند.
* تعداد دایره‌های قرمز در گوشه بالا-چپ تصویر نمایش داده می‌شود (مثلاً: Red circle count: 2).

**محدودیت‌ها**

* این کد فقط دایره‌ها را شناسایی می‌کند و برای اشکال دیگر کار نمی‌کند.
* در نورپردازی ضعیف یا تصاویر پیچیده ممکن است دقت کاهش یابد.
* فقط دایره‌هایی با شعاع بین 10 تا 25 پیکسل شناسایی می‌شوند.