تبدیل آفین:

تبدیل آفین یا **Affine Transformation** یکی از پرکاربردترین تبدیلات هندسی در پردازش تصویر است که در کتابخانه‌ی OpenCV پیاده‌سازی شده است. این تبدیل برخلاف پرسپکتیو، فقط شامل تغییرات خطی و انتقال (Translation) است و خطوط موازی در آن همچنان موازی باقی می‌مانند. در واقع، تبدیل آفین ترکیبی از عملگرهایی مانند انتقال، چرخش، تغییر مقیاس (Scaling)، برش (Shear) و بازتاب (Reflection) است. به دلیل سادگی و پایداری، از این روش در بسیاری از کاربردهای پردازش تصویر استفاده می‌شود.

در OpenCV برای محاسبه‌ی ماتریس تبدیل آفین از تابع **`cv2.getAffineTransform()`** استفاده می‌شود. این تابع با دریافت سه نقطه‌ی متناظر از تصویر اولیه و تصویر مقصد، یک ماتریس ۲×۳ ایجاد می‌کند. این ماتریس آفین سپس با استفاده از تابع `cv2.warpAffine()` بر روی تصویر اعمال می‌شود و نسخه‌ی جدید تصویر حاصل می‌گردد. انتخاب سه نقطه کافی است، زیرا تبدیل آفین فقط به اطلاعات پایه‌ای هندسی نیاز دارد و برخلاف تبدیل پرسپکتیو که به چهار نقطه احتیاج دارد، ساده‌تر و سریع‌تر محاسبه می‌شود.

کاربردهای تبدیل آفین بسیار متنوع هستند. برای مثال، در **تشخیص چهره** یا **شناسایی اشیاء**، از این تبدیل برای نرمال‌سازی تصاویر (یکسان کردن اندازه، زاویه و مکان) استفاده می‌شود تا الگوریتم‌های بعدی بهتر عمل کنند. همچنین در پردازش تصاویر ماهواره‌ای یا نقشه‌ها، از تبدیل آفین برای هم‌تراز کردن داده‌های تصویری با مقیاس‌ها و جهت‌های مختلف استفاده می‌شود. این تبدیل به دلیل حفظ روابط هندسی اصلی، یک ابزار قدرتمند برای آماده‌سازی داده‌ها در بسیاری از پروژه‌های بینایی ماشین محسوب می‌شود.

تبدیل آفین بر اساس 3 نقطه:

تبدیل آفین با سه نقطه بر پایه‌ی این ایده است که برای تعریف یک تبدیل هندسی خطی (شامل انتقال، چرخش، مقیاس و برش) به حداقل **سه جفت نقطه متناظر** نیاز داریم. در این حالت، سه نقطه‌ی انتخاب‌شده در تصویر مبدا به سه نقطه در تصویر مقصد نگاشت می‌شوند. با این اطلاعات، OpenCV می‌تواند یک ماتریس ۲×۳ بسازد که کل تصویر را بر اساس همین نگاشت تغییر دهد. دلیل کافی بودن سه نقطه این است که تبدیل آفین فضای دوبعدی را بدون خمیدگی یا تغییر پرسپکتیو توصیف می‌کند و سه نقطه می‌توانند این نگاشت را به‌طور کامل مشخص کنند.

در عمل، تابع `cv2.getAffineTransform(src_points, dst_points)` دقیقاً همین کار را انجام می‌دهد. شما باید یک آرایه شامل سه نقطه از تصویر مبدا (`src_points`) و یک آرایه شامل سه نقطه‌ی متناظر در تصویر مقصد (`dst_points`) تعریف کنید. خروجی این تابع یک ماتریس ۲×۳ است که ترکیبی از عملیات هندسی ذکرشده را

نشان می‌دهد. سپس با استفاده از `cv2.warpAffine(image, M, dsize)` این ماتریس روی کل تصویر اعمال می‌شود و تصویر جدید ساخته می‌شود.

برای مثال، فرض کنید می‌خواهید تصویری را کمی بچرخانید و در عین حال آن را به سمت راست انتقال دهید. شما کافی است سه نقطه‌ی مشخص در تصویر اصلی (مثل سه گوشه‌ی یک مثلث فرضی) را انتخاب کرده و سه نقطه‌ی مقصد متناظر را به گونه‌ای تعریف کنید که تغییرات مورد نظر شما را بازتاب دهند OpenCV. بقیه‌ی پیکسل‌ها را بر اساس این سه جفت نقطه محاسبه می‌کند و تصویر تغییر یافته به‌دست می‌آید. این روش هم ساده و سریع است و هم برای بسیاری از کاربردهای عملی مثل **هم‌تراز کردن تصاویر** یا **نرمال‌سازی داده‌ها** بسیار مناسب است.

ساده ترین کد:

یک کد ساده برای انجام این عملیان کد زیر است

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = plt.imread("images/triangle.jpg")
rows,cols,ch = image.shape

pts1 = np.float32([[100,350],[100,90],[490,350]])
pts2 = np.float32([[200,400],[500,300],[200,200]])

M = cv2.getAffineTransform(pts1,pts2)
dst = cv2.warpAffine(image,M,(cols,rows))

plt.subplot(121),plt.imshow(image),plt.title('Image')
plt.subplot(122),plt.imshow(dst),plt.title('Affine Transformed')
plt.show()
```