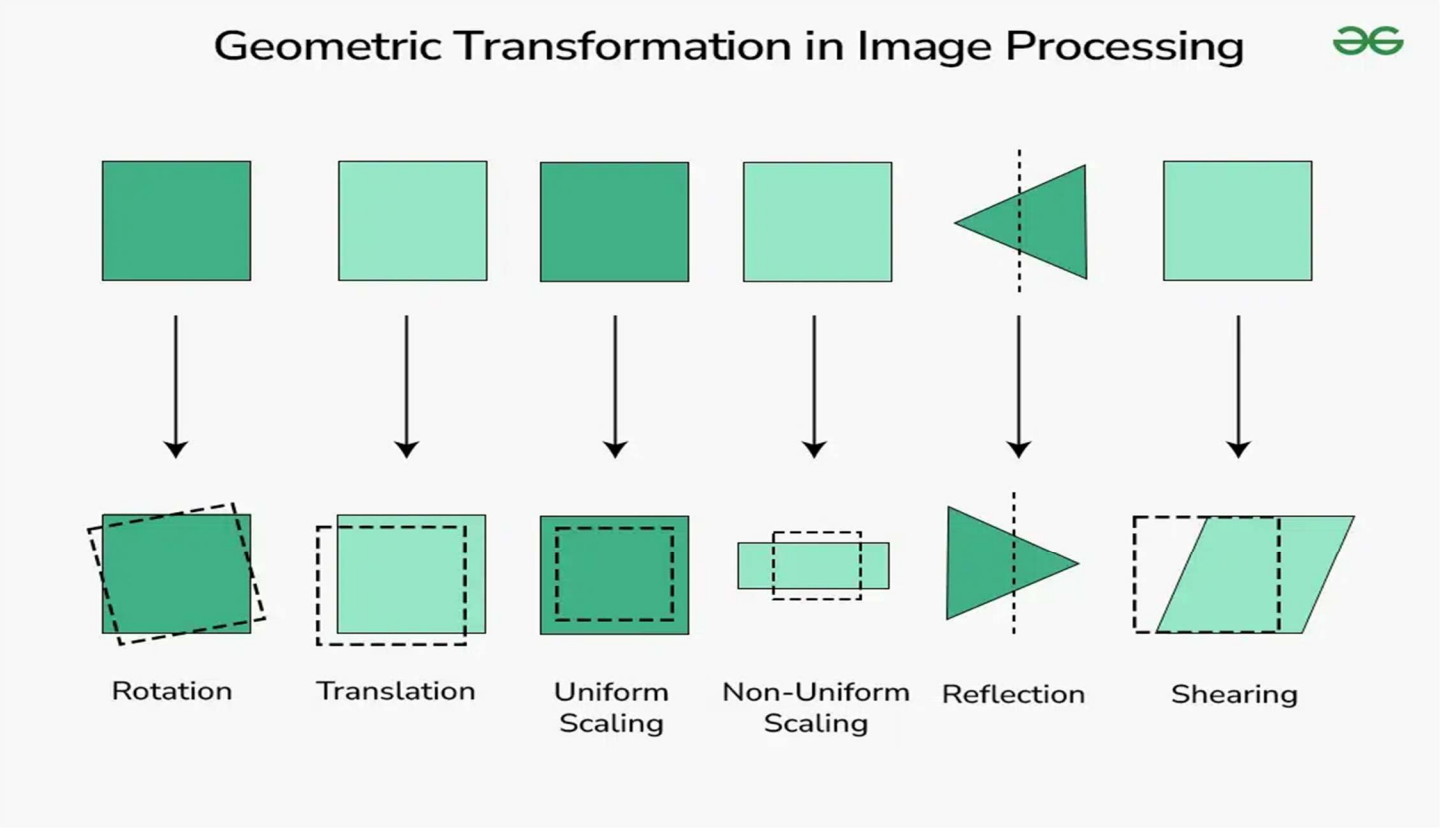
geometric transformations :

در کتابخانه **OpenCV**، تبدیلات هندسی یکی از پرکاربردترین ابزارها برای پردازش تصویر هستند. این تبدیلات شامل تغییر موقعیت، اندازه یا جهت نقاط تصویر می‌شوند و در بسیاری از کاربردها مانند تصحیح زاویه تصویر، بزرگ‌نمایی یا کوچک‌نمایی، چرخش و حتی تبدیل پرسپکتیو استفاده می‌شوند. اساس این تبدیلات بر پایه‌ی اعمال **ماتریس‌های تبدیلی** بر مختصات پیکسل‌ها است و به کمک توابعی مانند `cv2.warpAffine` یا `cv2.warpPerspective` پیاده‌سازی می‌شوند.

مهم‌ترین تبدیلات هندسی شامل **انتقال(Translation)** ، **چرخش (Rotation)** و **مقیاس‌دهی (Scaling)** هستند. انتقال باعث جابه‌جایی تصویر در راستای محورهای x و y می‌شود و با یک ماتریس ۲×۳ ساده قابل انجام است. چرخش نیز با استفاده از زاویه و نقطه‌ی مرجع مشخص می‌شود و علاوه بر تغییر جهت، می‌تواند شامل تغییر اندازه هم باشد. مقیاس‌دهی هم برای بزرگ‌کردن یا کوچک‌کردن تصویر به کار می‌رود و در این حالت پیکسل‌های جدید از طریق روش‌های **درون‌یابی (Interpolation)** مانند نزدیک‌ترین همسایه یا bilinear تعیین می‌شوند.

نوع پیشرفته‌تر این تبدیلات، **تبدیلات آفین (Affine Transformations)** و **پرسپکتیو (Perspective Transformations)** هستند. در تبدیل آفین، خطوط موازی همچنان موازی باقی می‌مانند و می‌توان تغییراتی مثل برش (Shearing) یا کشیدگی ایجاد کرد. در مقابل، تبدیل پرسپکتیو برای حالاتی استفاده می‌شود که تصویر نیاز به تغییر زاویه دید دارد (مثل تصحیح عکس یک صفحه که از زاویه گرفته شده). این تبدیل از ماتریس ۳×۳ استفاده می‌کند و انعطاف بیشتری در تغییر شکل تصویر دارد. به همین دلیل، در بینایی ماشین و کاربردهایی مثل تشخیص اسناد یا واقعیت افزوده بسیار اهمیت دارد.



Translation :

جابجایی یا **Translation**یکی از ساده‌ترین و پایه‌ای‌ترین تبدیلات هندسی در پردازش تصویر است. در این تبدیل، کل تصویر یا بخشی از آن در راستای محورهای افقی (x) و عمودی (y) به اندازه‌ی دلخواه جابه‌جا می‌شود، بدون آنکه شکل، اندازه یا زاویه اجزای تصویر تغییر کند. به عبارت دیگر، Translationفقط مکان پیکسل‌ها را تغییر می‌دهد و تمام ویژگی‌های دیگر آن‌ها ثابت باقی می‌ماند. این تبدیل زمانی اهمیت دارد که بخواهیم موقعیت یک تصویر یا شیء خاص را تغییر دهیم یا داده‌های تصویری را برای تحلیل بعدی آماده کنیم.

برای پیاده‌سازی Translation از یک **ماتریس انتقال** استفاده می‌شود که ابعاد آن ۲×۳ است. این ماتریس معمولاً به شکل زیر تعریف می‌شود:

$$\begin{bmatrix} tx & 0 & 1 \\ ty & 1 & 0 \end{bmatrix} = M$$

در اینجا، `tx` مقدار جابه‌جایی در راستای محور x و `ty` مقدار جابه‌جایی در راستای محور y را نشان می‌دهد. وقتی این ماتریس بر مختصات هر پیکسل اعمال شود، نقطه‌ی (x,y) به نقطه‌ی جدید (ty+tx,y+x) منتقل می‌شود.

در کتابخانه‌ی **OpenCV**، این کار با کمک تابع `cv2.warpAffine` انجام می‌شود. ابتدا ماتریس انتقال را با مقادیر مورد نظر تعریف می‌کنیم، سپس تصویر و ماتریس را به تابع می‌دهیم تا تصویر جدید تولید شود. نکته‌ای که باید به آن توجه کرد این است که هنگام جابه‌جایی، ممکن است بخشی از تصویر از محدوده‌ی قاب خارج شود یا در بخش‌هایی از تصویر که داده‌ی اصلی وجود ندارد، فضای خالی (معمولاً سیاه‌رنگ) ایجاد شود.

کاربرد Translation در بسیاری از زمینه‌های پردازش تصویر دیده می‌شود. برای مثال در **بینایی ماشین**، اگر بخواهیم الگوریتم تشخیص الگو نسبت به موقعیت اجسام مقاوم باشد، لازم است تصاویر آموزشی را با جابه‌جایی‌های مختلف ایجاد کنیم (Data Augmentation). همچنین در ردیابی اشیا(Object Tracking) ، هنگامی که جسمی در تصویر حرکت می‌کند، سیستم باید بتواند مکان آن را به‌درستی جابه‌جا کرده و دنبال کند. به این ترتیب، Translationبه عنوان یک ابزار ساده اما بسیار مهم در بسیاری از سیستم‌های بینایی کامپیوتری و یادگیری ماشین به کار می‌رود.

برای انجام این عملیات می توانیم از کد زیر استفاده کنیم:

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

image = cv2.imread('images/input.jpg')
height, width, _ = image.shape

#       | 1 0 Tx |
#  T  = | 0 1 Ty |

# T is our translation matrix
T = np.float32([[1, 0, 100], [0, 1,30]])
# We use warpAffine to transform the image using the matrix, T
img_translation = cv2.warpAffine(image, T, (width+100, height+30))
plt.imshow(img_translation[...,::-1])
```

در این کد ابتدا تصویر مورد نظر را می خوانیم. سپس ابعاد تصویر را پیدا میکنیم.

```
height, width, _ = image.shape
```

تابع `shape` . ابعاد تصویر را برمی‌گرداند. مقدار اول **ارتفاع(height)** ، مقدار دوم **عرض (width)** و مقدار سوم تعداد کانال‌ها برای تصویر رنگی سه کانال BGR است.

تعریف ماتریس انتقال:

```
T = np.float32([[1, 0, 100], [0, 1,30]])
```

اینجا ماتریس انتقال تعریف می‌شود. این ماتریس ۲×۳ به شکل زیر است:

$$\begin{bmatrix} 100 & 0 & 1 \\ 30 & 1 & 0 \end{bmatrix}$$

به این معنا که تصویر به اندازه‌ی **۱۰۰ پیکسل در راستای محور x افقی** و **۳۰ پیکسل در راستای محور y عمودی** جابه‌جا خواهد شد.

- **۱** یعنی مختصات همون محور رو بدون تغییر حفظ کن.
- **۰** یعنی محوری رو با محور دیگه ترکیب نکن (هیچ چرخش یا برشی اتفاق نیفتد).

OpenCVانتظار دارد که ماتریس تبدیل از نوع **float32**باشد، چون محاسبات هندسی (به‌خصوص اگر شامل چرخش یا تغییر مقیاس شوند) نیاز به دقت اعشاری دارند. اگر نوع داده‌ی ماتریس اشتباه باشد) مثلاً(int ، احتمال دارد خطا بدهد یا خروجی نادرست شود.

اعمال عملیات جابجایی:

```
img_translation = cv2.warpAffine(image, T, (width+100, height+30))
```

تابع cv2.warpAffine

تابع `warpAffine`یکی از توابع مهم OpenCV برای اعمال **تبدیلات آفین (Affine Transformations)** روی تصویر است.

تبدیلات آفین شامل Translation جابجایی، Rotationچرخش، Scaling(تغییر اندازه) و Shearing (برش) هستند.

ساختار کلی:

```
cv2.warpAffine(src, M, dsize)
```

`src` تصویر ورودی

`M` ماتریس تبدیل ۲×۳

`dsize` ابعاد تصویر خروجی (عرض و ارتفاع)