

توضیح فایل helper.py

در تمامی تمرین ها، کنار فایل اصلی کد، یک فایل helper.py هم قرار داده شده است. این فایل به منظور تمیزتر شدن کد نوشته شده است. تابع هایی که استفاده میکنم را در این فایل پیاده سازی می کنم تا کد اصلی تمیز تر بشود.

مختصات عکس ها را به صورت دستی بدست آورده ام که در کد مشخص است.

در فایل helper.py یک کلاس Book پیاده سازی کرده ام تا ساختار کد تمیزتر و بهتری داشته باشم. این کلاس هنگام ساختن یک instance از آن، مختصات x,y چهار نقطه گوشه کتاب را میگیرد. نقاطی که قرار است کتاب در نهایت به آن مپ شود، f_x و f_y ها را هم خودش با توجه به ارتفاع و طول کتاب که بامختصات آن بدست می آید می سازد. کلاس Book یک تابع به اسم get_book_image دارد که تصویر کتاب را با استفاده از تبدیل هوموگرافی طوری تغییر میدهد که عکس کتاب خواسته شده، به مختصات (0,0) آن برود و سپس با استفاده از کراپ کردن عکس، عکس کتاب را خروجی میدهد.

تابع فوق، از تابع wrap_perspective از پکیج cv2 استفاده میکند. این تابع با گرفتن ایمج اصلی و یک ماتریس تبدیل، عکس حاصل از آن تبدیل را به ما خروجی می دهد. بدین ترتیب اگر ماتریس تبدیل مناسب داشته باشیم، عکس ما تبدیل به عکسی می شود، که کتابی که ما میخواهیم به مختصات (0,0) آن میرود (باجهت درست)

بدین ترتیب تنها مسئله باقیمانده، بدست آوردن ماتریس تبدیل است. برای آنکه کتاب ها را تبدیل کنیم، میشد از ماتریس تبدیل دوران استفاده کرد، اما چون کمی عکس کاملاً عمود گرفته نشده است، بهتر است از تبدیل هوموگرافی استفاده کنیم، بدین ترتیب با ۸ درجه آزادی، بهترین تبدیل ممکنه را خواهیم داشت.

برای ساختن ماتریس تبدیل، نیاز به حل یک دستگاه ۸ معادله ۸ مجهولی داریم، بدین ترتیب ماتریس تبدیل که یک چنین ماتریسی است را بدست خواهیم آورد:

$$\begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

برای حل این دستگاه، در اینترنت کمی سرچ کردم که به تابع np.linalg.solve رسیدم. این تابع معادله ای همچون

$$AX = B$$

را با دریافت A و B حل میکند (X را خروجی میدهد)

حال برای آنکه معادله را با کمک ماتریس ها حل کنم از معادله ماترسی زیر استفاده کردم:

$$PH = \begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1 x'_1 & y_1 x'_1 & x'_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1 y'_1 & y_1 y'_1 & y'_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2 x'_2 & y_2 x'_2 & x'_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2 y'_2 & y_2 y'_2 & y'_2 \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3 x'_3 & y_3 x'_3 & x'_3 \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3 y'_3 & y_3 y'_3 & y'_3 \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4 x'_4 & y_4 x'_4 & x'_4 \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4 y'_4 & y_4 y'_4 & y'_4 \end{bmatrix} \begin{bmatrix} h1 \\ h2 \\ h3 \\ h4 \\ h5 \\ h6 \\ h7 \\ h8 \\ h9 \end{bmatrix} = 0$$

بدین ترتیب می توان یک دستگاه ۸ معادله ۸ مجهولی را حل کرد (که h ها مجهولات ما هستند) که در مورد ما، h9 برابر ۱ است. همچنین تابع np.linalg.solve نیاز به یک ماتریس مربعی دارد که بدین ترتیب یک سطر کامل صفر به جز آخرین درایه آن که ۱ است به ماتریس فوق اضافه میکنیم. بدین ترتیب قادر به بدست آوردن ضرایب تبدیل هستیم. برای محاسبه این ماتریس هم یک تابع دیگر به اسم get_transform_matrix به کلاس Book اضافه کردیم که این ماتریس را بدست می آورد.

ماتریس های تبدیل حین اجرای کد پرینت می شوند که نتایج آن بدین صورت است:

```
Transform matrix for book1:
[[-3.50924936e-01  9.58295018e-01  2.86408736e+01]
 [-9.58273467e-01 -3.33312510e-01  7.09539006e+02]
 [ 2.67620283e-05  2.14991890e-05  1.00000000e+00]]
=====
Transform matrix for book2:
[[-1.00725730e+00 -1.69101590e-01  4.92115037e+02]
 [ 1.50237917e-01 -9.85936329e-01  6.76878154e+02]
 [-1.28984085e-04  6.07650556e-05  1.00000000e+00]]
=====
Transform matrix for book3:
[[-9.24844545e-01  5.87980638e-01  1.82482241e+02]
 [-5.86328192e-01 -8.92619038e-01  1.34311161e+03]
 [ 6.03908916e-05  5.37134426e-05  1.00000000e+00]]
=====
```