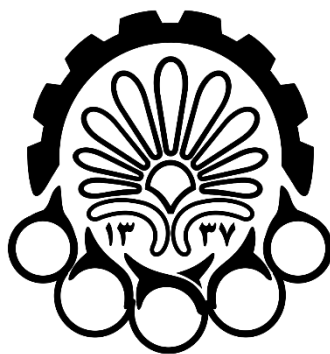


به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تمرین درس بینایی ماشین – سری دوم

فردین آیار

شماره دانشجویی: ۹۹۱۳۱۰۴۰

استاد: دکتر صفابخش

دانشکده کامپیوتر – پاییز ۱۴۰۰

(۱) کد مربوط به این قسمت در فایل 1.py قرار دارد. در شکل ۱، خروجی الگوریتم به ازای پارامترهای $\text{Rho} = 1$ ، $\text{Theta} = \text{PI}/180$ و ترشهولد ۲۰۰ رسم شده است. همانطور که مشاهده می‌کنید بسیاری از خط‌ها تشخیص داده نشده‌اند و بعضی از خط‌ها اضافه رسم شده‌اند. علت این موضوع این است که تابع `houghlinesP` برای پیدا کردن خط‌ها، از زیر مجموعه‌ای از نقاط استفاده می‌کند و ممکن است دقت آن از `houghlines` کمتر باشد. می‌توان با افزایش پارامتر Rho و Theta و یا کاهش ترشهولد، خط‌های بیشتری را تشخیص داد اما این کار باعث می‌شود بسیاری از اعداد اشتباه خط راست تشخیص داده شوند که مطلوب ما نیست. برای بهبود این مشکل می‌توان از یک پارامتر اضافه که تابع `houghlinesP` ارائه می‌کند، `maxLineGap`، استفاده کنیم. این پارامتر اجازه می‌دهد یک خط ناپیوسته باشد و ناپیوستگی خط را به صورت خودکار برطرف می‌کند. با در نظر گرفتن مقدار ۵۰ این پارامتر، خروجی مطلوب در شکل ۲ ارائه شده است.

3			8		1			2
2		1		3		6		4
			2		4			
8		9				1		6
	6						5	
7		2				4		9
			5		9			
9		4		8		7		5
6			1		7			3

شکل ۱

3			8		1			2
2		1		3		6		4
			2		4			
8		9				1		6
	6						5	
7		2				4		9
			5		9			
9		4		8		7		5
6			1		7			3

شکل ۲

به ازای مقادیر ثابت Θ و ρ ، کاهش ترشهولد باعث می‌شود خطوط بیشتری توسط الگوریتم شناسایی شود اما زمان اجرای الگوریتم تغییری نمی‌کند. به ازای مقادیر ثابت ترشهولد، با افزایش Θ و ρ ، فضای جستجوی مسئله کوچکتر می‌شود. این موضوع دو تأثیر در خروجی دارد: اول، زمان اجرا کمتر می‌شود؛ دوم، حساسیت مسئله به محل نقاط کمتر می‌شود و در نتیجه تعداد خطوط تشخیص داده شده بیشتر می‌شود.

۲) کد مربوط به این قسمت در فایل 2.py قرار دارد. خروجی با نام outpy.avi ذخیره می‌شود. پس از اعمال یک فیلتر هموارساز، از تابع HoughCircles برای تشخیص دایره‌ها استفاده می‌کنیم. به منظور کاهش اشتباه الگوریتم، برای پارامترهای minDist، minRadius و maxRadius نیز مقدار تعیین می‌کنیم. همانطور که در خروجی مشاهده می‌شود، سکه‌ها به جز در مواردی که به صورت چرخیده قرار دارند، به خوبی تشخیص داده شده‌اند. با کم کردن محدودیت‌ها احتمالاً بتوان در این حالت نیز سکه‌ها را تشخیص داد، اما این کار باعث می‌شود دایره‌های اشتباه زیادی در تصویر بوجود بیاید که مطلوب نیست. این تابع به طور متوسط برای هر فریم حدود ۰.۰۰۶ زمان نیاز دارد که نسبتاً سریع محسوب می‌شود.

۳) کد مربوط به این قسمت در فایل 3.py قرار دارد. به منظور افزایش سرعت برنامه، ابعاد تصویر ورودی کاهش و اطراف آن برش داده شده است. همچنین به منظور تحلیل بهتر نتایج، ۵ بیضی با بیشترین امتیاز را نشان می‌دهیم. به منظور افزایش بیشتر سرعت برنامه، برای اپراتور Canny ترشهولد بالایی در نظر گرفته شده است. مجموع این عوامل و این نکته که صورت ممکن است بیضی کامل نباشد، باعث شده عملکرد الگوریتم اندکی ضعیف باشد.

۴) با توجه به اینکه در تبدیل هاف همه پارامترهای بیضی در نظر گرفته شده‌اند، به صورت تئوری به نظر می‌رسد این تبدیل باید نسبت به چرخش و مقیاس مقاوم باشد. اما در عمل این مقاومت تنها نسبت به تغییرات کم در وضعیت سر مقاوم است. عامل اصلی این موضوع این است که این تغییر وضعیت‌ها می‌توانند سر را از حالت بیضی خارج کند. عامل دیگر این موضوع این است که دقت تبدیل به منظور افزایش سرعت آن کاهش داده شده است. برای حل این موضوع به نظر می‌رسد باید علاوه بر افزایش دقت تبدیل، دانش‌های پیش زمینه‌ای مانند نسبت ابعاد صورت را نیز به الگوریتم اضافه کرد.