



دانشکده مهندسی کامپیوتر

## تمرین سری ۹ بینایی کامپیوتر

نام درس

مبانی بینایی کامپیوتر

نام دانشجو

زهرا انوریان

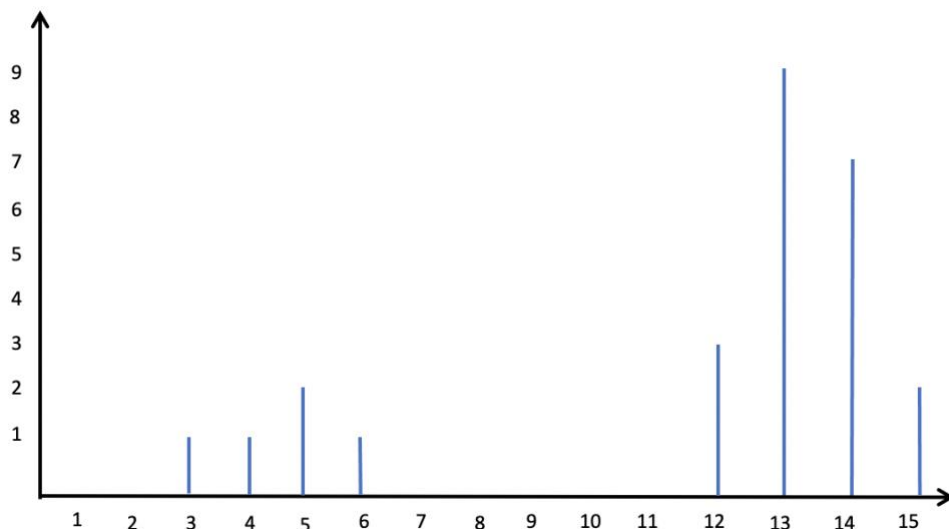
نام استاد درس

دکتر محمدی

پاییز ۱۳۹۹

## سوالات

۱. با توجه به تصویر زیر ابتدا هیستوگرام را به دست آورده سپس به سوال‌های زیر پاسخ دهید (سطوح رنگ از ۰ تا ۱۵ است) (۲۰ نمره).  
پاسخ: نمودار هیستوگرام بدست آمده:



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$n_k$	0	0	0	1	1	2	1	0	0	0	0	0	3	9	6	2

الف) میانگین، میانه، مد و واریانس شدت روشنایی را به دست آورید.

پاسخ:

میانگین:

$$\begin{aligned} \text{mean} &= 3 + 4 + (5 * 2) + 6 + (3 * 12) + (9 * 13) \\ &\quad + (6 * 14) + (2 * 15) = 290 \Rightarrow \frac{290}{25} = 11.6 \\ &\cong 12 \end{aligned}$$

میانه:

$$\text{median} = \frac{25}{2} = 12.5 \cong 13 \Rightarrow 13\text{th number is } 13$$

مد:

$$\text{mode} = 13$$

واریانس:

$$\begin{aligned} \text{variance} &= (3 - 10)^2 + (4 - 10)^2 + 2 * (5 - 10)^2 + (6 - 10)^2 + 3 * (12 - 10)^2 \\ &\quad + 9 * (13 - 10)^2 + 6 * (14 - 10)^2 + 2 * (15 - 10)^2 \\ &= 49 + 36 + 50 + 16 + 12 + 18 + 96 + 50 = \frac{327}{25} = 13.8 \end{aligned}$$

ب) اگر تصویر به دو گروه شدت روشنایی بزرگتر یا مساوی با ۱۲ و کمتر از ۱۲ تقسیم شود، مقادیر میانگین، میانه، مد و واریانس شدت روشنایی را برای هر گروه به دست آورید.

پاسخ: گروه ۱ (۱۲ کمتر)

میانگین:

$$\text{mean} = \frac{23}{5} = 4.6 \cong 5$$

میانه:

$$\text{median} = \frac{5}{2} = 2.5 \cong 3 \Rightarrow 3\text{th number is } 5$$

مد:

$$\text{mode} = 5$$

واریانس:

$$\begin{aligned} \text{variance} &= (3 - 4.6)^2 + (4 - 4.6)^2 + 2 * (5 - 4.6)^2 + (6 - 4.6)^2 \\ &= 2.56 + 0.36 + 0.32 + 1.96 = \frac{5.2}{5} = 1.04 \end{aligned}$$

گروه ۲ (۱۲ و بیشتر)

میانگین:

$$\text{mean} = 36 + 117 + 84 + 30 = \frac{267}{20} = 13.35 \cong 13$$

میانه:

$$\text{median} = \frac{20}{2} = 10 \Rightarrow 10\text{th number is } 13$$

مد:

$$\text{mode} = 13$$

واریانس:

$$\begin{aligned} \text{variance} &= 3 * (12 - 13.35)^2 + 9 * (13 - 13.35)^2 + 6 * (14 - 13.35)^2 + 2 * (15 - 13.35)^2 \\ &= 5.4675 + 1.1025 + 2.535 + 5.445 = \frac{14.55}{20} = 0.7275 \end{aligned}$$

ج) سطح آستانه بهینه را برای این تصویر به دست آورید و بگویید مقدار تابع هزینه برای چند مقدار باید محاسبه شود.

$$k = 4 \Rightarrow w_1 = \frac{1}{25} = 0.04, \quad w_2 = \frac{24}{25} = 0.96$$

$$\sigma_1^2 = (3 - 3)^2 = 0$$

$$\begin{aligned} \sigma_2^2 &= (4 - 11.95)^2 + 2 * (5 - 11.95)^2 + (6 - 11.95)^2 + 3 * (12 - 11.95)^2 \\ &\quad + 9 * (13 - 11.95)^2 + 6 * (14 - 11.95)^2 + 2 * (15 - 11.95)^2 \\ &= 63.2025 + 96.605 + 35.4025 + 0.0075 + 9.9225 + 25.215 + 18.605 \\ &= \frac{248.96}{24} = 10.37 \end{aligned}$$

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2 = 0.04 * 0 + 0.96 * 10.37 = 9.9552$$

$$k = 5 \Rightarrow w_1 = \frac{2}{25} = 0.08, \quad w_2 = \frac{23}{25} = 0.92$$

$$\sigma_1^2 = (3 - 3.5)^2 + (4 - 3.5)^2 = 0.25 + 0.25 = \frac{0.5}{2} = 0.25$$

$$\begin{aligned} \sigma_2^2 &= 2 * (5 - 12.30)^2 + (6 - 12.30)^2 + 3 * (12 - 12.30)^2 + 9 * (13 - 12.30)^2 \\ &\quad + 6 * (14 - 12.30)^2 + 2 * (15 - 12.30)^2 \\ &= 106.58 + 39.69 + 0.27 + 4.41 + 17.34 + 14.58 = \frac{182.87}{23} = 7.95 \end{aligned}$$

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2 = 0.08 * 0.25 + 0.92 * 7.95 = 7.354$$

$$k = 6 \Rightarrow w_1 = \frac{4}{25} = 0.16, \quad w_2 = \frac{21}{25} = 0.84$$

$$\begin{aligned} \sigma_1^2 &= (3 - 4.25)^2 + (4 - 4.25)^2 + 2 * (5 - 4.25)^2 = 1.5625 + 0.0625 + 1.125 = \frac{2.75}{4} \\ &= 0.6875 \end{aligned}$$

$$\begin{aligned} \sigma_2^2 &= (6 - 13)^2 + 3 * (12 - 13)^2 + 9 * (13 - 13)^2 + 6 * (14 - 13)^2 + 2 * (15 - 13)^2 \\ &= 49 + 3 + 6 + 8 = \frac{66}{21} = 3.14 \end{aligned}$$

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2 = 0.16 * 2.75 + 0.84 * 3.14 = 3.0776$$

$$k = 12 \Rightarrow w_1 = \frac{5}{25} = 0.2, \quad w_2 = \frac{20}{25} = 0.8$$

$$\sigma_1^2 = 1.04$$

$$\sigma_2^2 = 0.7275$$

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2 = 0.2 * 1.04 + 0.8 * 0.7275 = 0.208 + 0.582 = 0.79$$

$$k = 13 \Rightarrow w_1 = \frac{8}{25} = 0.32 \quad , \quad w_2 = \frac{17}{25} = 0.68$$

$$\begin{aligned}\sigma_1^2 &= (3 - 7.375)^2 + (4 - 7.375)^2 + 2 * (5 - 7.375)^2 + (6 - 7.375)^2 \\ &= 19.140 + 11.390 + 4.75 + 1.890 = \frac{37.17}{8} = 4.64\end{aligned}$$

$$\begin{aligned}\sigma_2^2 &= 9 * (13 - 13.58)^2 + 6 * (14 - 13.58)^2 + 2 * (15 - 13.58)^2 \\ &= 3.0276 + 1.0584 + 4.0328 = \frac{8.1188}{17} = 0.4775\end{aligned}$$

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2 = 0.32 * 4.64 + 0.68 * 0.4775 = 1.8095$$

$$k = 14 \Rightarrow w_1 = \frac{17}{25} = 0.68 \quad , \quad w_2 = \frac{8}{25} = 0.32$$

$$\begin{aligned}\sigma_1^2 &= (3 - 10.35)^2 + (4 - 10.35)^2 + 2 * (5 - 10.35)^2 + (6 - 10.35)^2 \\ &\quad + 3 * (12 - 10.35)^2 + 9 * (13 - 10.35)^2 \\ &= 54.0225 + 40.3225 + 57.245 + 18.9225 + 8.1675 + 63.2025 \\ &= \frac{241.8825}{17} = 14.228\end{aligned}$$

$$\sigma_2^2 = 6 * (14 - 14.25)^2 + 2 * (15 - 14.25)^2 = 0.375 + 1.125 = \frac{1.5}{8} = 0.1875$$

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2 = 0.68 * 14.228 + 0.32 * 0.1875 = 9.73504$$

$$k = 15 \Rightarrow w_1 = \frac{23}{25} = 0.92 \quad , \quad w_2 = \frac{2}{25} = 0.08$$

$$\begin{aligned}\sigma_1^2 &= (3 - 11.30)^2 + (4 - 11.30)^2 + 2 * (5 - 11.30)^2 + (6 - 11.30)^2 \\ &\quad + 3 * (12 - 11.30)^2 + 9 * (13 - 11.30)^2 + 6 * (14 - 11.30)^2 \\ &= 68.89 + 53.29 + 79.38 + 28.09 + 1.47 + 26.01 + 43.74 \\ &= \frac{300.87}{23} = 13.0813\end{aligned}$$

$$\sigma_2^2 = 2 * (15 - 15)^2 = 0$$

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2 = 0.92 * 0 + 0.08 * 13.0813 = 1.0465$$

در نهایت کمترین واریانس بدست آمده در بین تمام واریانس‌ها را انتخاب کرده و آن  $k$  مناسب‌ترین آستانه می‌باشد. پس  $k=12$  مناسب‌ترین آستانه برای باینری کردن تصویر زیر می‌باشد.  
برای بدست آوردن آن نیز در این تصویر باید برای ۷ مقدار آستانه، تابع هزینه محاسبه شود.

۱۳	۶	۱۲	۴	۳
----	---	----	---	---

۱۳	۱۳	۱۴	۱۵	۱۴
۱۴	۵	۱۳	۱۵	۱۴
۵	۱۲	۱۳	۱۳	۱۳
۱۴	۱۲	۱۳	۱۴	۱۳

۲. در این تمرین با استفاده از توابع موجود در OpenCV تصویر 1.jpg را در قاب موجود در تصویر background.jpg قرار دهید. برای این کار مختصات چهار طرف تصویر را به صورت دستی به دست آورید و در تابع AR از آن استفاده کنید. خروجی این تابع باید تصویری باشد که تصویر 1.jpg در قاب موجود در تصویر background.jpg باشد.

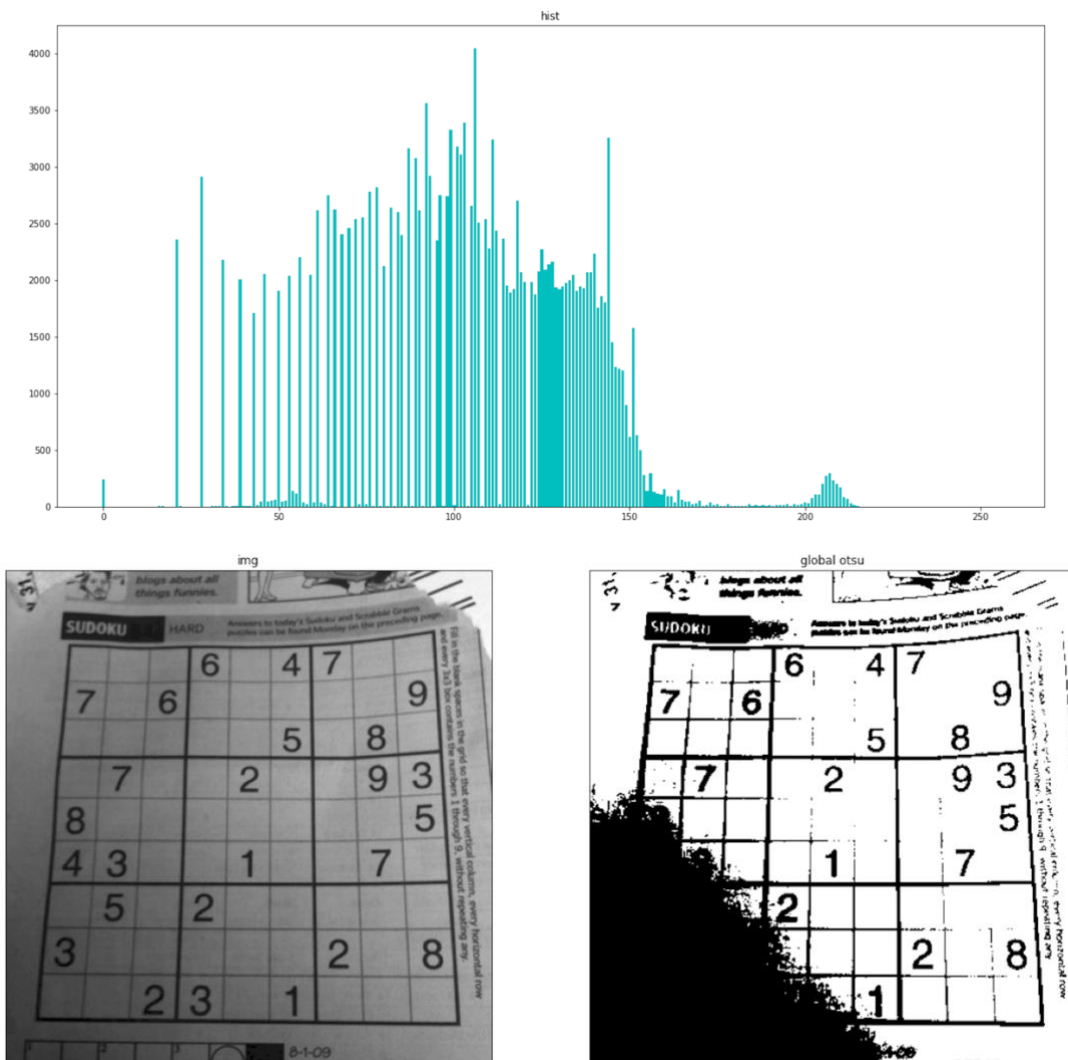
راهنمایی: توابع findHomography و warpAffine را بررسی کنید (۳۰ نمره).  
**پاسخ:** ابتدا به صورت دستی چهار گوشه‌ی داخلی قاب و تصویر گل را بدست می‌آوریم و با استفاده از تابع findHomography ماتریس تبدیل را بدست می‌آوریم و سپس با استفاده از تابع warpAffine تصویر تبدیل شده‌ی گل در جایگاه قاب را بدست می‌آوریم و در نهایت تصویر بدست آمده را بر روی تصویر background قرار می‌دهیم.



۳. بخش‌های زیر را پیاده‌سازی کنید (۳۰ نمره).

**الف) الگوریتم otsu** را بدون استفاده از توابع OpenCV پیاده‌سازی کنید و بر روی تصویر q2.png اعمال کنید.

**پاسخ:** برای پیاده‌سازی الگوریتم otsu ابتدا باید هیستوگرام تصویر را محاسبه کنیم و سپس یک آستانه‌ای را برای گروه‌بندی هیستوگرام به جهت باینری کردن عکس انتخاب می‌کنیم و حال برای ارزیابی مناسب بودن آستانه‌ی انتخاب شده، واریانس دو گروه را بدست می‌آوریم و سپس واریانس کل را با استفاده از فرمول زیر بدست می‌آوریم که هرچه مقدار واریانس بدست آمده کمتر باشد یعنی شدت تغییرات کمتری داریم پس با استفاده از آستانه‌ی انتخاب شده به درستی دو گروه را می‌توان بدست آورد و تصویر را باینری کرد.



**ب) تصویر q2.png** را به چهار بخش تقسیم کنید و الگوریتم otsu را به شکل جدا برای هر بخش اجرا کنید.

**پاسخ:** ابتدا تصویر را به چهار قسمت تقسیم می‌کنیم و سپس الگوریتم otsu توضیح داده شده را برای هر قسمت اجرا می‌کنیم..



ج) با استفاده از تابع `AdaptiveThreshold` آستانه‌گذاری افقی را روی تصویر `q2.png` انجام دهید و کاربرد پارامترهای تابع را توضیح دهید. همچنین خروجی خود را با خروجی `OpenCV` مقایسه کنید.

پاسخ: با استفاده از تابع `adaptiveThreshold` تصویر را باینری می‌کنیم. این تابع دارای پارامترهای زیر می‌باشد.

`dst = cv2.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C)`

`src`: تصویر ورودی

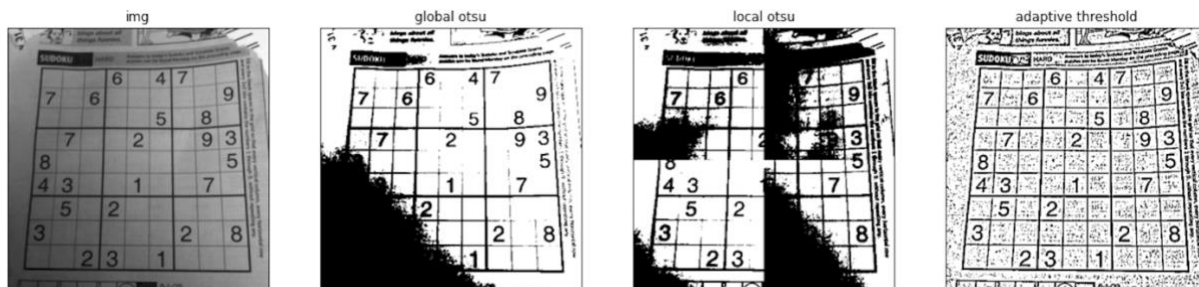
`maxValue`: بیشترین مقداری می‌توانند پیکسل‌ها داشته باشند.

`adaptiveMethod`: تعیین الگوریتم آستانه‌گذاری افقی میانگین‌گیری یا فیلتر گوسی (البته لازم به ذکر است که این دو الگوریتم با احتساب پیکسل مرکزی انجام می‌شوند اگر می‌خواهیم نقطه‌ی مرکزی در محاسبات لحاظ نشود و فقط همسایگی آن در نظر گرفته شود و یا می‌خواهیم از الگوریتم دیگری (میان) استفاده کنیم باید تابع آن را خود پیاده‌سازی کنیم).

`thresholdType`: نوع آستانه‌گذاری را مشخص می‌کند که به صورت باینری یا معکوس آن می‌تواند باشد.

`blockSize`: ابعاد فیلتری که می‌خواهیم روی پیکسل مورد نظر برای انجام الگوریتم آستانه‌گذاری افقی استفاده کنیم را مشخص می‌کند.

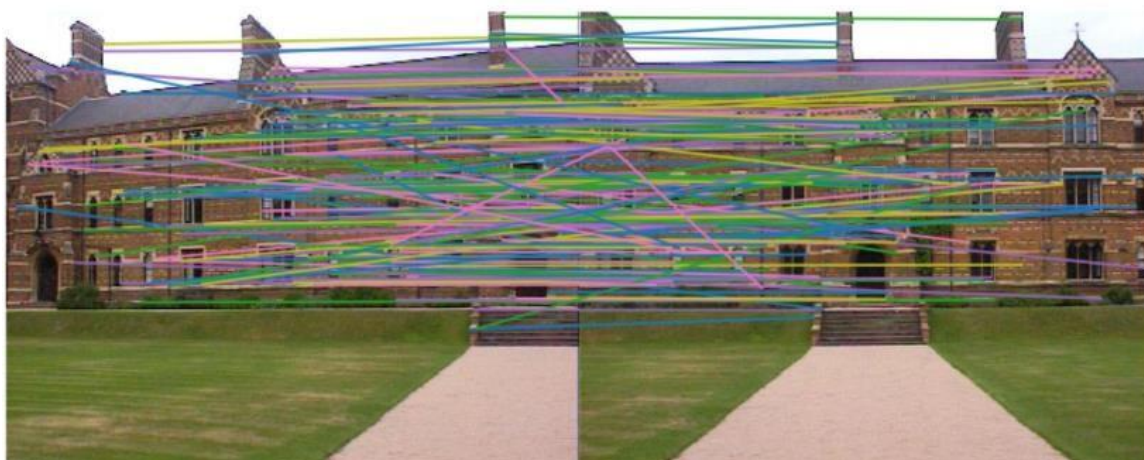
`C`: عدد ثابتی که از میانگین کسر می‌شود تا خطا را کم کند. (لزوماً هم مثبت نمی‌باشد)





۴. در تمرین‌های گذشته با normalized cross-correlation آشنا شده‌اید. در این تمرین شما ابتدا نقاط کلیدی را به کمک harris detector استخراج کرده (میتوانید از توابع آماده درون OpenCV استفاده کنید) و با استفاده از همسایگی نقاط کلیدی یک توصیفگر ساده برای این نقاط در نظر گرفته و استخراج کنید. در مرحله بعد از شاخص normalized cross-correlation برای انطباق توصیفگرها استفاده کنید. سپس با استفاده از روش گفته شده، نقاط کلیدی متناظر را در دو تصویر building1.jpg و building2.jpg را به دست آورده و با یک خط آن‌ها را به هم متصل کنید. در نهایت ویژگی‌های این روش و نقاط ضعف و قوت آن را بیان کنید. یک نمونه خروجی در تصویر زیر آمده است. شما میتوانید خروجی بهتری را بدست آورید (۴۰ نمره).

**پاسخ:** ابتدا با استفاده از تابع cornerHarris نقاط کلیدی تصاویر را پیدا می‌کنیم. حال باید برای هر نقطه یک توصیفگر تعریف کنیم تا بتوان با مقایسه‌ی دو به دوی توصیفگر نقاط، تناظر درست را پیدا کنیم و با یک خط آن‌ها را به یکدیگر متصل کنیم. یک پنجره‌ی 5x5 حول نقطه‌ی کلیدی توصیفگر در نظر گرفته شده می‌باشد و سپس با استفاده از فرمول زیر NCC را میان توصیفگرها بدست می‌آوریم و سپس نقطه‌ای از تصویر ۲ را که ماکسیمم مقدار NCC را دارد، به عنوان نقطه‌ی متناظر تصویر ۱ انتخاب می‌کنیم و سپس خطی میان آن‌ها با رنگ‌های رندوم می‌کشیم تا این نگاشت بهتر دیده شود.



• موفق باشید.