



University of Tehran

School of Electrical and Computer Engineering



Digital Image Processing

Instructor: Hamid Soltanian-Zadeh

Assignment 10

Chapter 12 - Object Recognition

Sasan Keshavarz

810199253

Spring 2022

فهرست

چکیده	۱
سوال ۱	۲
بخش اول	۲
بخش دوم	۲
سوال ۲	۴
بخش اول	۴
بخش دوم	۴
سوال ۳	۶
سوالات تحلیلی	۷
12.6	۷
12.15	۸
12.17	۸
12.22	۹
پیوست ۱: روند اجرای برنامه	۱۱
مراجع	۱۲

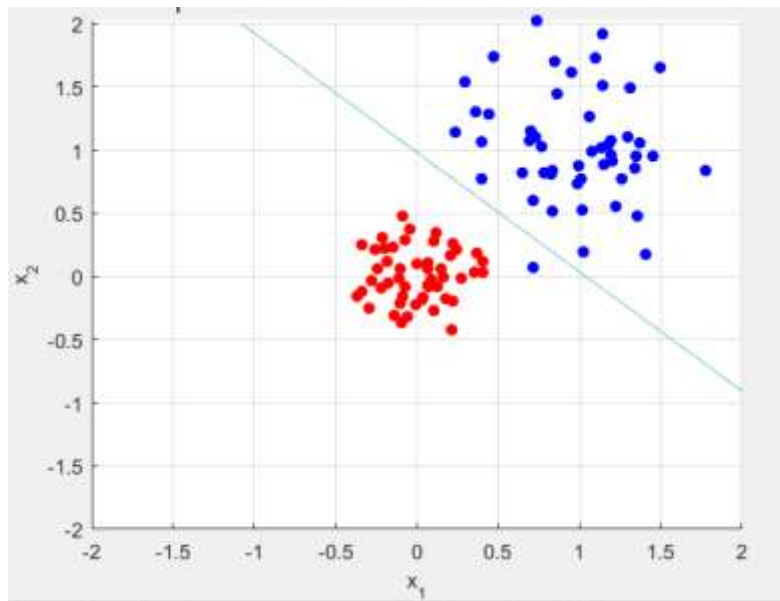
در این تمرین به بررسی مفاهیمی چون شبکه های عصبی (در حد مقدماتی) و طبقه بندی داده ها آشنا می شویم . همچنین با استفاده از استخراج یک سری ویژگی و اختصاص یک برچسب متناسب با آن یک طبقه بند را آموزش داده و سپس بر روی داده های آزمون اعمال و خطای حاصل را محاسبه می نماییم. همچنین در یکی از مثال ها طبقه بند بیز را به صورت ساده پیاده سازی و داده ها را طبقه بندی میکنیم.

سوال ۱

بخش اول

ابتدا مطابق با خواسته سوال کلاس ۱ و ۲ را ایجاد میکنیم .

سپس با استفاده از طبقه بند کمترین فاصله^۱ و روابط مربوط به آن مرز ناحیه تصمیم گیری را به شرح زیر محاسبه میکنیم :



شکل ۱ داده های دو کلاس و مرز تصمیم گیری

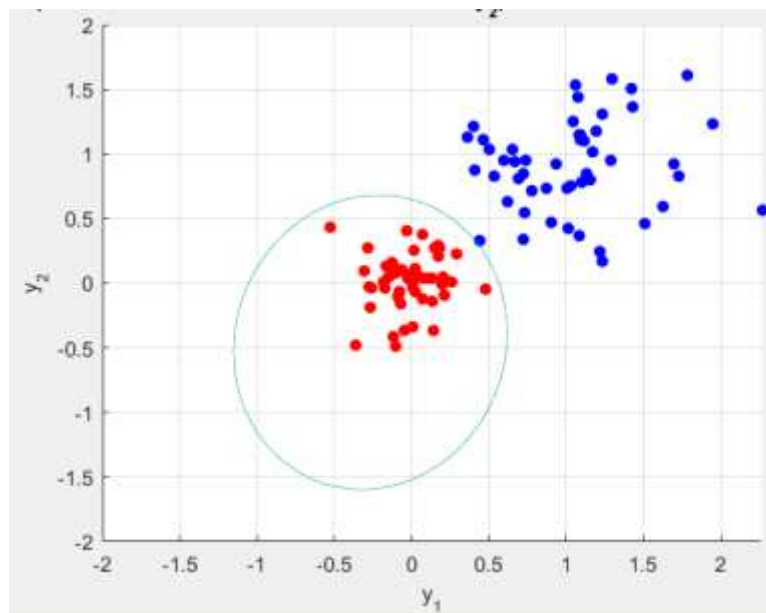
در محاسبه مرز تصمیم گیری از روابط کتاب عینا استفاده شده است و داده ها در شکل ۱ با دقت ۹۹٪ جدا شده اند. اگر ۱۰ بار کد را ران کنیم و سپس میانگین بگیریم دقت ۹۹/۲٪ شد.

بخش دوم

مطابق روابط کتاب فرمول بیز را پیاده سازی میکنیم که احتمال $P(x|w)$ یک توزیع گوسی دارد. کافیتست مختصات هر نقاط را یک بار در توزیع کلاس صفر و بار دیگر در توزیع کلاس ۲ قرار داده و آن ها را مقایسه کنیم . اگر حاصل توزیع ۱ برای مثال بزرگ تر از توزیع صفر بود آن نقطه را به کلاس ۱ نسبت می دهیم. در آخر نیز برچسب های جدید را Tbayes می نامیم و خطا را مطابق آن محاسبه میکنیم. برای محاسبه مرز تصمیم گیری نیز مشابه روابط کتاب عمل میکنیم.

¹ Minimum Distance Classifier

حاصل طبقه بندی در نهایت به مانند شکل ۲ خواهد بود.



شکل ۲ داده های اصلی و نتیجه اعمال الگوریتم بیز بر روی آن ها

برای تاکید بیشتر معادله ای که از آن برای تعیین مرز تصمیم گیری استفاده کردیم را عینا می آوریم:

$$d_j = -0.5 \ln(|C_j|) - 0.5 (x - m_j)^T C_j^{-1} (x - m_j)$$

برای هر کلاس معادله بالا را حساب و سپس از هم کم می نماییم و مرز تصمیم گیری به دست می آید.

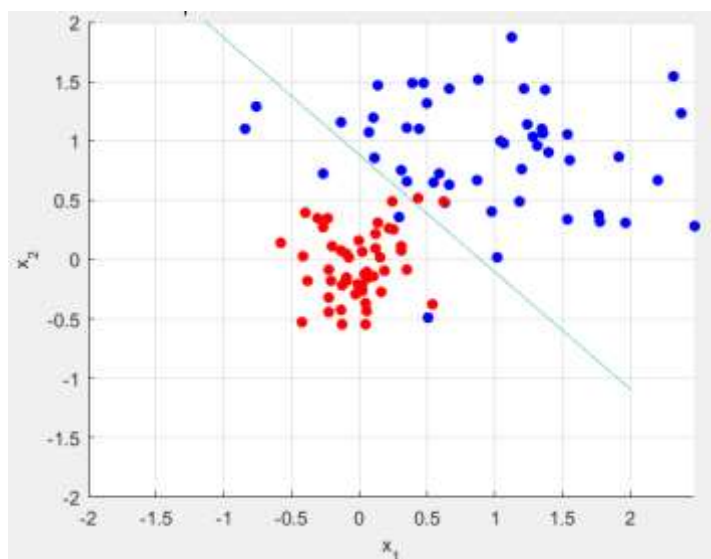
دقت ای طبقه بند بیشتر از طبقه بند حداقل فاصله است و در شکل ۲ این دقت ۱۰۰٪ بوده است. با تکرار این کد به اندازه ۱۰ بار دقت میانگین ۹۹/۷٪ به دست آمد.

نتایج طبقه بندی از طبقه بند خطی حداقل فاصله بهتر است و همینطور هم توقع میرفت چون در صورت مشخص بودن توزیع داده ها بهترین طبقه بند، طبقه بند بیزی است. ضمن اینکه مرز تصمیم بهتری هم به دست میدهد و یک خط ساده نیست.

سوال ۲

بخش اول

در این سوال الگوریتم عینا مثل سوال قبل است اما توزیه داده های هر الگو متفاوت است که منجر به کاهش دقت و عملکرد طبقه بند شده است. در شکل ۳ نتیجه اعمال الگوریتم دیده میشود.

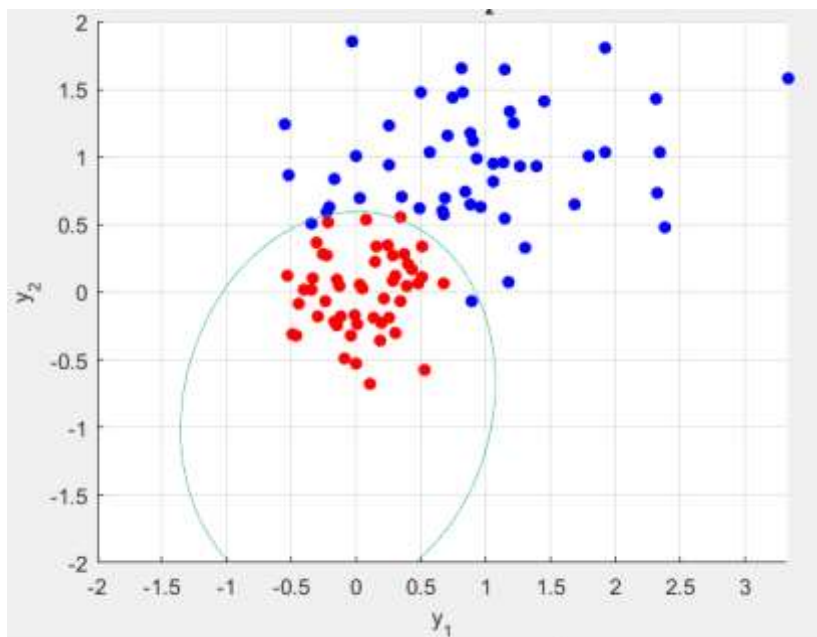


شکل ۳ داده های جدید دو کلاس و مرز تصمیم گیری جدید

دقت طبقه بند در داده های شکل ۳ ۹۳٪ است. پس از تکرار این کد به اندازه ۱۰ بار دقت میانگین حاصل شده ۹۵/۵٪ به دست می آید.

بخش دوم

برای این قسمت هم دقیقا از الگوریتم طبقه بند بیز استفاده شده در مثال قبل استفاده میکنیم. اما باز هم چون توزیع دادگان طوری است که بیشتر در هم فرو رفته اند دقت طبقه بند کاهش یافته است. با این دادگان جدید عملکرد بهتر طبقه بند بیز بیشتر مشخص است. نتایج یک بار ران کردن کد در شکل ۴ نمایش داده شده است.



شکل ۴ داده های اصلی و نتیجه اعمال الگوریتم بیز بر روی آن ها

دقت طبقه بند برای شکل ۴ ۹۷٪ است و پس از اعمال ۱۰ مرتبه کد دقت میانگین ۹۸/۵٪ به دست می آید. در قیاس با روش طبقه بند حداقل فاصله با توزیع جدید میتوان گفت که طبقه بند بیز بسیار بهتر عمل میکند و انعطاف مرز تصمیم غیر خطی بسیار به کار می آید.

پرسپترون شبکه عصبی را برای دسته‌بندی کردن سه میوه پیاده سازی کنیم. در این اینجا با استفاده از الگوریتم گفته شده در کتاب برای دسته‌بندی استفاده می‌کنیم. که این الگوریتم ساده ترین الگوریتم پرسپترون و اصلاح وزن به حالتی مشابه svm است. که هربار دسته بندی غلط بود به اندازه خود داده وزن را اصلاح میکنیم. چون سه دسته داریم و باید بین این سه دسته تصمیم‌گیری انجام دهیم، لازم است که حداقل دو لایه نورون داشته باشیم. با توجه به مقادیر داده شده در جدول ۱، مقادیر مشخصه را وارد می‌کنیم و ماتریسی مربوط به میوه‌ها میسازیم.

سپس برای هر یک از دسته‌ها، مقادیر ۱- و ۱ را اختصاص می‌دهیم. برای سیب، ۱- و ۱-، برای موز ۱- و ۱- و برای کیوی ۱ و ۱ داریم. چون که دو نورون داریم، برای بهبود سرعت، هریک از نورون‌ها را جدا جدا آموزش می‌دهیم. باید دقت کرد که به جز وزن متناظی با هر ویژگی (که ۳ ویژگی داریم) یک مقدار هم باید برای بایاس در نظر گرفت. خروجی مطلوب نورون به ازای هر ورودی معادل مقادیر متناظر مشخص شده با آن در ماتریس findex است. برای نورون دوم هم مشابه این کار را انجام می‌دهیم و الگوریتم را اجرا می‌کنیم.

پس از اصلاح وزن‌ها و به اصطلاح آموزش شبکه عصبی، از شبکه برای دسته بندی دو میوه نامعلوم استفاده میکنیم. تابع مناسب activation برای لایه آخر و تصمیم گیری شبکه hardlims است.

نتیجه اجرای کد این است که میوه نامعلوم اول کیوی است و میوه نامعلوم دوم موز است.

12.6

راه حل بخش اول این مشکل بر این اساس است که بتوان اجزای متصل را استخراج کرد و سپس تعیین محدب بودن یا نبودن یک جزء متصل را بررسی کرد. هنگامی که همه اجزای متصل استخراج شدند، بررسی تحدب را روی هر یک انجام می دهیم و آنهایی را که محدب نیستند رد می کنیم. تنها چیزی که پس از این باقی می ماند این است که مشخص شود آیا حباب های باقی مانده کامل هستند یا ناقص. برای انجام این کار، ناحیه ی قاب تصویر تشکیل شده است، منطقه الف اعلام می شود. سپس اگر پیکسل به پیکسل AND این ناحیه با یک جز خاص حداقل یک نتیجه ۱ را موجب شد، نتیجه آن این است که مرز تصویر آن جز را در برمیگیرد و آن جز ناقص نامیده می شود. وقتی فقط یک پیکسل در یک لکه AND برابر ۱ را به دست می دهد، نتیجه ای حاشیه ای داریم که در آن فقط یک پیکسل در یک لکه مرز را لمس می کند. می توانیم ساده گیری کنیم و لکه را کامل اعلام کنیم. از نقطه نظر پیاده سازی، بسیار ساده تر است که رویه ای داشته باشیم که هر زمانی که عملیات AND یک یا چند نتیجه با ارزش ۱ به دست آورد، یک حباب را ناقص می خواند. بعد از اینکه حباب ها با استفاده از روشی که اخیراً مورد بحث قرار گرفت غربال شدند، باید به یکی از سه کلاس ارائه شده در بیان مسئله طبقه بندی شوند.

ما مسئله طبقه بندی را بر اساس بردارهایی به شکل $x = (x_1, x_2)$ انجام می دهیم، که در آن x_1 و x_2 به ترتیب طول محور اصلی و فرعی یک حباب بیضوی باقیمانده پس از غربالگری هستند. میانگین بردار هر کلاس برای اجرای یک طبقه بندی کننده حداقل فاصله، در مسئله به عنوان میانگین طول هر یک از دو محور برای هر کلاس از حباب ها آورده شده است. اگر آنها داده نمی شد، می توان آنها را با اندازه گیری طول محورها برای بیضی های کامل که به طور پیشینی متعلق به هر یک از سه کلاس طبقه بندی شده اند، به دست آورد. بنابراین مجموعه بیضی های داده شده یک مجموعه آموزشی را تشکیل می دهد و یادگیری شامل محاسبه محورهای اصلی برای همه بیضی های یک کلاس و سپس به دست آوردن میانگین است. این برای هر کلاس تکرار می شود.

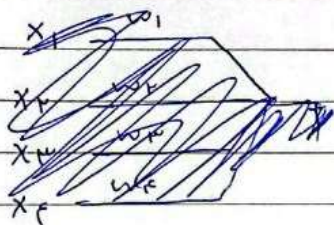
12.15

$$d_i(x) = \ln(p(w_i)) + x^T c^{-1} m_j - \frac{1}{r} m_i^T c^{-1} m_i \quad 12.15$$

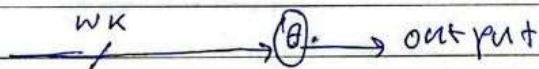
$$d_j(x) = \ln(p(w_j)) + x^T c^{-1} m_j - \frac{1}{r} m_j^T c^{-1} m_j$$

$$d_{ij}(x) = d_i(x) - d_j(x) = \underbrace{\ln(p(w_i)) - \ln(p(w_j))}_{\text{عدد ثابت}} + x^T c^{-1} (m_i - m_j) - \underbrace{\frac{1}{r} (m_i - m_j)^T c^{-1} (m_i - m_j)}_{\text{عدد ثابت}}$$

$$d(x) = \frac{w^T x}{\|w\|} - w_{n+1} \rightarrow \text{برای وزن ها: } w_k = v_k \rightarrow v = c^{-1} (m_i - m_j)$$

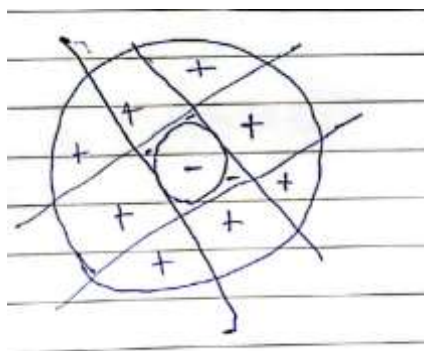


$$\text{برای } \theta = w_{n+1} = (\ln(p(w_i)) - \ln(p(w_j)) + x^T c^{-1} (m_i - m_j))$$



12.17

مرزهای مورد نیاز برای جداسازی آن دو الگو در شکل زیر نشان داده شده است. مرز حداقل پیچیدگی، در این مورد، یک مثلث است، اما در این آرایش آنقدر تنگ است که حتی اغتشاشات کوچک در موقعیت الگوها می تواند منجر به خطاهای طبقه بندی شود. بنابراین از شبکه ای با قابلیت پیاده سازی ۴ خط به صورت دو بعدی استفاده می کنیم. در این حالت گره خروجی به عنوان یک گیت AND با ۴ ورودی عمل می کند. گره خروجی یک خروجی ۱ می دهد زمانی که خروجی های ۴ گره قبلی همه به طور همزمان بالا باشند. این مربوط به الگویی است که در سمت مثبت هر ۴ خط قرار دارد و بنابراین به کلاس w_1 تعلق دارد. هر ترکیب دیگری خروجی ۰ به دست می دهد که نشان دهنده کلاس w_2 است.



12.22

از آنجایی که سرعت و هزینه برای کارفرما مهم هستند، ما یک رویکرد باینری را تصور می کنیم که در آن اخذ تصویر، پیش پردازش، و بخش بندی در یک عملیات پایه ای انجام میشوند. این رویکرد ما را به آستانه گذاری کلی به عنوان روش انتخاب میرساند.

اولین قدم تعیین اندازه ناحیه مشاهده و وضوح مورد نیاز برای تشخیص کوچکترین اجزای مورد نظر، در این مورد، ستاره ها است. از آنجایی که تصاویر در حال حرکت هستند و مکان دقیق هر پرچم مشخص نیست، لازم است میدان دیدی مشخص شود که تضمین کند هر تصویر حداقل یک پرچم کامل را شامل می شود. علاوه بر این، نرخ فریم باید به اندازه کافی سریع باشد تا هیچ پرچمی از قلم نیفتد. میدان دید باید به اندازه ای گسترده باشد که ناحیه ای کمی بزرگتر از دو پرچم به اضافه حداکثر فاصله بین آنها را در بر گیرد. بنابراین، عرض، W ، ناحیه مشاهده باید حداقل $W = 2(5) + 2.05 = 12.1$ باشد.

بازرسی بصری یک پرچم معمولی نشان می دهد که قسمت آبی پرچم حدود 0.4 برابر طول پرچم را اشغال می کند، که در این مورد حدود 100 پیکسل در هر خط در ناحیه آبی به ما می دهد. حداکثر شش ستاره در هر خط وجود دارد، و فضای آبی بین آنها تقریباً 1.5 برابر عرض یک ستاره است، بنابراین تعداد پیکسل های یک ستاره حدود 6 پیکسل است. از آنجایی که تعداد پیکسل ها در هر شی مورد نظر تنها 6 است، ما تاری را باید در کمتر از یک پیکسل ثابت می کنیم. بنابراین، $T \leq 9.52 \times 10^{-4}$ ثانیه سرعت شاتر مورد نیاز است. نرخ فریم باید به اندازه کافی سریع باشد تا از هر پرچمی که از نقطه بازرسی عبور می کند تصویری بگیرد. ما باید هر 0.3 ثانیه یک فریم بگیریم تا تضمین کنیم که هر تصویر حاوی یک پرچم کامل است و هیچ پرچمی از دست نخواهد رفت.

چهار حالت کلی تصویر باینری انتظار می رود: (۱) بخشی از یک پرچم در سمت چپ تصویر، به دنبال آن یک پرچم کامل، و به دنبال آن یک پرچم جزئی دیگر. (۲) یک پرچم کامل که مرز چپ را لمس می کند،

به دنبال آن یک پرچم کامل دوم، و سپس یک شکاف قبل از مرز سمت راست. (۳) برعکس حالت (۲)؛ و (۴) دو پرچم کامل، بدون اینکه هیچ یک از پرچم ها به مرز تصویر رسیده باشند. برای رسیدگی به مورد (۱) باید یک پرچم کامل را که بین دو پرچم جزئی قرار دارد شناسایی کنیم. پنجره مناسبی برای جستجوی شکاف قابل توجهی بین تعداد بالای حالت ۱ استفاده می شود که برای تشخیص مورد (۴) به اندازه کافی باریک است. برای مورد (۱)، این رویکرد تعداد زیادی را تولید می کند که از سمت چپ تصویر شروع می شود، سپس به تعداد بسیار کمی (مرتبط با پس زمینه) برای حدود دو اینچ کاهش می یابد، دوباره وقتی مرکز (کل پرچم) مواجه می شود، انتخاب می شود. ۱های بین دو شکاف داخلی با یک پرچم کامل مطابقت دارند و با روش های مورد بحث در زیر پردازش می شوند. ۱های دیگر نادیده گرفته می شوند. موارد (۲) و (۳) به روشی مشابه با منطق کمی متفاوت بررسی میشوند. باید دقت کرد که داده های مربوط به یک پرچم کامل (یعنی پرچم با شکاف در هر طرف) جدا شود. این رویکرد برای استخراج ۱ بر این فرض استوار است که پس زمینه بیش از حد نویز ندارد. به عبارت دیگر، سیستم تصویربرداری باید به گونه ای باشد که پس زمینه به طور مطمئنی به صورت سیاه و با نویز قابل قبول تقسیم شود.

با توجه به پایپلاین کلی شکل ۱،۲۳ کتاب گونزالس، ما از مرحله تقسیم بندی عبور کرده ایم. برای ستاره ها، ما از تجزیه و تحلیل اجزای متصل استفاده می کنیم. اما برای تشخیص نوارها، از تجزیه و تحلیل امضا استفاده می کنیم. این سیستم محورهای دو خط عمودی را تعیین که شامل کل پرچم بین آنهاست. ابتدا، تجزیه و تحلیل اجزای متصل را در نیمه سمت چپ منطقه انجام می دهیم و همه اجزای کوچکتر و بزرگتر از اندازه مورد انتظار ستاره ها را فیلتر می کنیم، مثلاً همه اجزای کمتر از 9×3 پیکسل و بزرگتر از 64×8 پیکسل. ساده ترین آزمایش در این مرحله، شمارش تعداد اجزای متصل باقی مانده (که فرض می کنیم ستاره هستند) است. اگر عدد ۵۰ باشد، آزمایش بعدی را روی نوارها ادامه می دهیم. در غیر این صورت ما پرچم را رد می کنیم.

برای تجزیه و تحلیل نوارها، فرض می کنیم که پرچم ها روی مواد سفید چاپ شده اند. بنابراین، "افتادن یک نوار" به معنای ایجاد یک نوار سفید دو برابر عرض معمول است. این یک نقص ساده است که با اجرای یک خط اسکن عمودی در ناحیه ای که تضمین شده است نوارها وجود داشته باشد و سپس به علامت شدت برای تعداد پالس ها در ارتفاع و مدت زمان مناسب قابل تشخیص است. خط اسکن باید قبل از تجزیه و تحلیل، برای پر کردن شکاف های کوچک به دلیل نویز، از قبل پردازش شود. علیرغم تغییر جهت $\pm 15^\circ$ ، یک ناحیه، مثلاً ۱ اینچ در سمت راست ناحیه آبی به اندازه کافی مستقل از تغییرات چرخشی از نظر نشان دادن تنها نوارهایی در امتداد یک خط اسکن که به صورت عمودی در آن ناحیه اجرا می شود، است. پس الگوریتم درست اجرا خواهد شد.

پیوست ۱: روند اجرای برنامه

پوشه تصاویر در فایل کدها قرار داده شده است و با انجام `set path` کدها اجرا خواهند شد. بخش‌های مختلف هر کد با %% از هم تفکیک شده‌اند. در صورت نیاز توضیحاتی در خود کد نوشته شده است.

- [1] R. C. Gonzalez and R. E. Woods, "Digital image processing." Prentice hall Upper Saddle River, NJ, 2002.
- [2] MATLAB help