فصل 16

کاربردهای فازی در پردازش تصاویر

16-1: مقدمه

پردازش تصویر، حوزه پرکاربردی است که هر ساله تحقیقات زیادی در ارتباط با آن انجام می‌شود. بسیاری از الگوریتمهای موجود در این حوزه تلاش می‌کنند تا سیستم بینایی چشم انسان را شبیه‌سازی نمایند. اما اغلب آنها غیر قابل اعتماد هستند، زیرا بر پایه منطقی دقیق و قطعی استوار شده‌اند. لذا به‌سختی می‌توانند دانش غیرقطعی دنیای واقعی را به تصویر بکشند. به عبارت دیگر، بسیاری از مشکلات فعلی حوزه‌های تحقیقاتی، به دلیل استفاده از ریاضیات و سیستم منطق سنتی برای مدل‌سازی دنیای پیچیده می‌باشد.

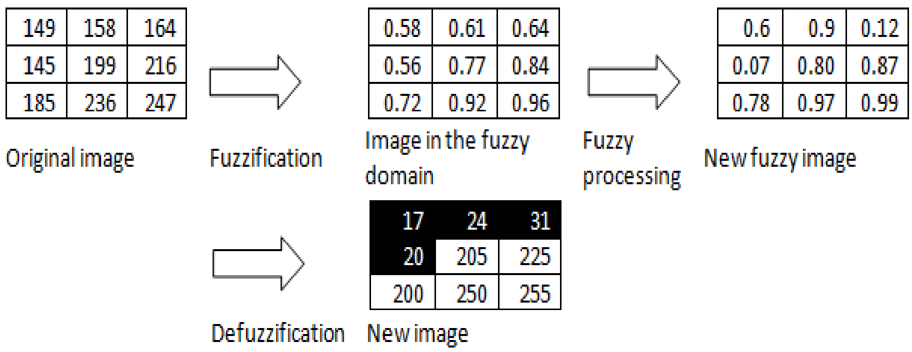
یک ساختار مناسب برای مدل‌کردن دنیای واقعی باید تا حد امکان تعریفی منطبق بر واقعیت، از آن ارائه کند. ارائه تعاریف دقیق و قطعی از مفاهیم موجود در دنیای واقعی، منشأ بسیاری از مشکلات خواهد بود.

به عنوان مثال، فرض کنید می‌خواهیم استانداردی را برای تعیین افراد قدبلند ارائه بدهیم. اولین مشکل ما بیان واضحی از مفهوم "بلندقدی" می‌باشد. در سیستم سنتی، یک مرز دقیق را مشخص می‌کنیم. مثلاً 180سانتیمتر و بیشتر را قدبلند به حساب می‌آوریم. هرچند که انتخاب همین مرز خود جای بحث دارد، اما سوالی مهمتر مطرح می‌شود: این چگونه حکمی است که برای افراد دارای قد 180 سانتیمتر و بیشتر صادق است، ولی برای افراد دارای قد 179.99یا اندکی کمتر از این مقدار، نادرست است؟ برای حل این مشکلات نیاز به سیستمهای جبری داریم تا این‌گونه عدم‌قطعیتها را پوشش بدهد. منطق فازی ابزاری است که این امکان را برای ما فراهم می‌کند. باید توجه داشت که اگرچه منطق فازی ابزار مناسبی برای بیان و مدل‌سازی این مفاهیم است، اما پیاده‌سازی آن در سیستمهای عملی به قیمت افزایش در هزینه محاسبات می‌باشد.

پردازش تصویر دارای مفاهیمی غیرقطعی مانند لبه، روشنایی و اندازه ویژگیها در تصویر می‌باشد. بسیاری از مفاهیم مطرح‌شده، تعریف مشخص و دقیقی ندارند. مثلاً نمی‌توان تعریف دقیقی برای پیکسل مرزی (پیکسل متعلق به پس‌زمینه یا شیء خاص) ارائه کرد. ارائه تعاریف فازی از مفاهیم سنتی، حوزه‌های جدیدی را برای پژوهش فراهم می‌کند.

16-2: پردازش فازی تصاویر

پردازش فازی تصاویر مشابه روشهای کلاسیک پردازش تصویر است، با این تفاوت که ابتدا تصویر را برای فازی کردن به یک دامنه جدید (دامنه فازی) نگاشت می‌دهیم و پس از انجام عملیات فازی مجددا تصویر را به دامنه ابتدایی برمی‌گردانیم. مراحل کلی کار در قالب یک مثال در شکل16-1 بیان شده است.



شکل16-1: تغییرات پیکسلهای یک تصویر با پردازش فازی

شکل16-2 یک طرح کلی را برای پردازش تصاویر در فضای فازی ارائه می‌کند. مفاهیم مطرح‌شده در این طرح شامل موارد زیر هستند:

- $I(x,y)$: تصویر اولیه
- F: مجموعه‌ای از نگاشتها برای نگاشت تصویر از فضای معمولی به فضای فازی
- $I_F(x,y)$: نگاشت تصویر اصلی به فضای فازی
- $I'_F(x,y)$: تصویر فازی پردازش‌یافته
- D: اپراتور نگاشت تصویر فازی به فضای مکانی
- $I'(x,y)$: نگاشت تصویر فازی به فضای مکانی (خروجی نهایی سیستم)

$$\boxed{I(x,y)} \rightarrow \boxed{F: I(x,y) \Rightarrow I_F(x,y)} \rightarrow \boxed{\Im[I_F(x,y)] = I_F{}'(x,y)} \rightarrow \boxed{D: I_F{}'(x,y) \Rightarrow I'(x,y)}$$

شکل16 -2: مراحل پردازش فازی تصویر

مراحل کار شامل قسمتهای زیر می‌باشد:

1. تصویر را از فضای مکانی به فضای فازی نگاشت می‌دهیم تا دامنه اعداد $[0,255]$ تصاویر خاکستری، به دامنه فازی نرمال‌شده $[0,1]$ نگاشته شود.
2. عملگر فازی روی تصویر حاصل اعمال می‌شود تا تصویر جدیدی را در فضای فازی ارائه کند. این عملگر ممکن است مشتق فازی، انتگرال فازی یا هر عملگر فازی دلخواهی باشد.
3. عمل غیرفازی‌سازی را انجام می‌دهیم. این عملگر تبدیلی است که تصویر را از فضای فازی به فضای مکانی نگاشت می‌دهد.

در ادامه به معرفی مفاهیم استفاده شده در پردازش فازی تصاویر یعنی نگاشت فازی، اپراتورهای فازی و تابع معکوس فازی می‌پردازیم.

16-2-1: نگاشت فازی

نگاشت فازی یک تصویر براساس ویژگی مطلوبی که از تصویر توقع داریم، انجام می‌شود. این ویژگیها مواردی مانند لبه‌ها، پیوستگی نواحی و پیچیدگیهای تصویر هستند که با متغیرهای زبانی بیان می‌شوند. در هنگام تعریف متغیرهای زبانی، باید اصلاحاتی را روی تعاریف اعمال کرد تا بتوان مشخصات و ویژگیهای تصویر را به فرم مناسب‌تری نمایش داد. تابع نگاشت فازی می‌تواند هر تابع عضویت دلخواهی مانند تابع مثلثی، ذوزنقه‌ای، تابع S گونه یا Zگونه (معرفی‌شده در فصل دوم)، ... باشد. اگر L بیانگر مجموعه سطوح خاکستری تصویر باشد، روشنایی یک پیکسل را می‌توان از رابطه زیر به نحوی به دست آورد که خروجی نگاشت در بازه صفر تا یک باشد:

$$F\left(l\right) = ml + b \qquad l \partial L$$

و در ادامه با استفاده از ∇ به عنوان عملگر گرادیان، می‌توان لبه‌ها را به صورت زیر در فضای فازی تعریف کرد:

$$F(\nabla)=S(\nabla;a,b,c)=\begin{cases}0 & \nabla\le a\\ 2\left(\dfrac{x-a}{c-a}\right)^2 & a\le\nabla\le b\\ 1-2\left(\dfrac{x-a}{c-a}\right)^2 & b\le\nabla\le c\\ 1 & \nabla>c\end{cases}$$

از نگاشتها می‌توان با کمک اصلاحات و اعمال ضرایب روی آن، با کارآمدی بیشتری برای نمایش ویژگیها و مشخصات استفاده کرد. مثلاً پیکسلهای نگاشت‌یافته می‌توانند روشن‌تر یا تاریک‌تر (با توجه به ضریب تعریف‌شده) از معمول باشند:

$$\alpha F(l)=\alpha(ml+b)$$

که در این رابطه اگر α بزرگتر از یک باشد پیکسل را روشن‌تر و در غیر این صورت تاریک‌تر خواهد کرد.

16-2-2: عملگرهای فازی

عملگرهای فازی را می‌توان به دو صورت شامل عبارتهای ریاضی و قوانین نوشت. در ادامه، دو عملگر ریاضی پرکاربرد معرفی می‌شوند:

16-2-2-1: عملگر تشدید وضوح[1]

$$INT(A)=\begin{cases}2\mu_A^2(l) & 0\le\mu_A(x)\le 0.5\\ 1-2(1-\mu_A^2(l)) & 0.5\le\mu_A(x)\le 1\end{cases}$$

[1] Contrast intensification

μ_A تابع عضویتی است که روی سطوح خاکستری تعریف می‌شود و میزان روشنایی هر پیکسل را می‌توان با این عملگر تصحیح نمود.

16-2-2-2: فیلتر میانگین

فیلتر میانگین در فضای فازی به صورت زیر نمایش داده می‌شود:

$$I\left(x_j\right)=\frac{\sum_{j=1}^{N}\mu_{ij}I\left(x_j\right)}{\sum_{j=1}^{N}\mu_{ij}}$$

$I\left(x_j\right)$ مقدار پیکسل در یک نقطه را نشان می‌دهد.

همچنین، می‌توان عملگرها را به صورت مجموعه‌ای از قوانین به فرم زیر نوشت:

$$Rule\ \ k: if\quad fuzzy\ \ condition\quad k\quad then\quad fuzzy\ action\ k$$

فارغ از نوع نمایش عملگرها، عملگرهای تعریف شده در فضای فازی وظیفه انجام عملیات بر روی داده‌های فازی را با استفاده از اصول و روشهای استدلال و استنتاج فازی به عهده دارند.

16-2-3: معکوس فازی

عملگر معکوس فازی (برای عمل غیرفازی‌سازی) تصویر فازی را به فضای مکانی نگاشت می‌دهد تا ویژگیهای آن، پس از پردازشهای فازی لازم، به صورت مناسبی نمایش داده شود. این نگاشت به فرم کلی زیر می‌باشد:

$$D:\left[0,1\right]\Rightarrow\left[0,L-1\right]$$

فرض می‌شود سطوح قابل مشاهده، دارای L مقدار متمایز باشد. مثلاً L برای تصاویر خاکستری برابر با 256 است. حالت کلی تابع در زیر دیده می‌شود:

$$D:\left\{\mu_A(\chi_i)\right\}\Rightarrow I'\left(x,y\right)$$

بدیهی است خروجی پردازش فازی یک تصویر که با $I'\left(x,y\right)$ نمایش داده می‌شود همانند تصویر ورودی با فرمت یک تصویر ارائه و نمایش داده می‌شود.

16-2-4: مثالهایی از پردازش فازی تصاویر

در این قسمت، به توضیح چند مثال از پردازش فازی تصاویر می‌پردازیم. هر کدام از مثالها بیانگر کاربرد جالبی از منطق فازی در پردازش تصاویر می‌باشد.

16-2-4-1: رقمی‌سازی فازی[1]

در تصاویر دودویی (باینری)، هر کدام از پیکسلهای تصویر در حالت سنتی، فقط به یکی از دو کلاس موجود (کلاس پس‌زمینه و کلاس اشیای تصویر) انتساب داده می‌شوند. مشکل اصلی در این مسئله، انتساب صحیح نقاط مرزی به کلاسها است. در منطق فازی، هر کدام از پیکسلها دارای درجه‌ای هستند که میزان انتساب آن پیکسل به هر یک از دو کلاس را تعیین می‌کند. این روش، از اطلاعات بیشتری در مقایسه با روش سنتی برای کلاس‌بندی داده‌ها سود می‌برد. تابع فازی که بیانگر میزان عضویت یک پیکسل به کلاس اشیا است، با در نظر گرفتن یک شیء روشن در یک زمینه تیره به صورت زیر تعریف می‌شود:

$$\mu_{1j} = \frac{\left(\text{gray level of } l_j\right)}{255}$$

و برای تعیین میزان عضویت به کلاس پس زمینه می‌توان از رابطه زیر استفاده کرد:

$$\mu_{2j} = 1 - \mu_{1j}$$

مرکز هر کدام از کلاسها را با فرمول مرکز ثقل می‌توان از فرمول زیر به دست آورد:

$$V_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^m x_j}{\sum_{j=1}^{n} (\mu_{ij})^m}$$

و توابع عضویت را می‌توان به کمک مراکز جدید با فرمول زیر به‌روزرسانی کرد:

[1] Fuzzy binarization

$$\mu_{ij}=\frac{\left(\dfrac{1}{d^2(x_j,V_i)}\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{c}\left(\dfrac{1}{d^2(x_j,V_i)}\right)}$$

که $d^2\left(x_j,V_i\right)$ به عنوان فاصله هر پیکسل تا مرکز ثقل کلاس مربوطه به صورت زیر تعریف شده است:

$$d^2\left(x_j,V_i\right)=\left(x_j-V_i\right)^T\left(x_j-V_i\right)$$

در پایان، تابع هدف را که می‌خواهیم بهینه شود، به صورت زیر تعریف می‌کنیم:

$$J_m=\sum_{i=1}^{c}\sum_{j=1}^{n}(\mu_{ij})^m d^2\left(x_j,V_i\right)$$

که با در نظر گرفتن c کلاس با هدف کمترین اختلاف روشنایی در درون کلاس و بیشترین اختلاف روشنایی بین کلاسها تعریف شده است.

16-2-4-2: تعریف فازی لبه

لبه‌یابی یکی از مهمترین مراحل در پردازش تصاویر می‌باشد. به همین دلیل، تحقیقات زیادی در این رابطه صورت می‌گیرد. اما با تمام تلاشهای صورت‌گرفته در این زمینه، مسئله تعیین لبه همچنان یک *مسئله باز*[1] محسوب می‌شود.

مسئله اساسی در تعیین لبه، مشخص کردن تعلق یک پیکسل به کلاس لبه‌ها می‌باشد. لبه‌ها را می‌توان به دو کلاس تقسیم کرد: در کلاس سطح یک، لبه‌ها را به‌راحتی می‌توان با چشم تشخیص داد، اما تشخیص لبه‌ها در کلاس سطح دو به دقت بیشتری نیاز دارد.

در یک دید کلی، عملیاتی که منظور تشخیص لبه انجام می‌شود دارای مراحل زیر می‌باشد:

1. به دست آوردن گرادیان تصویر اصلی:

$$\nabla : I \Rightarrow I_e$$

---

[1] Open question (problem)

2 مشخص کردن تعلق یک پیکسل به کلاس لبه‌ها به روش زیر:

$$if\ \left|p_{ed}-c_i\right|<\left|p_{ed}-c_j\right|\ \text{for i, j}=1,2,...,5\ \ (i\neq j)$$
$$\text{then } p_{ed}\in E_{li}\ \text{ and mark } p_{ed}\ \ as\,level\,i$$

که C_i مرکز کلاس لبه‌ها و p_{ed} پیکسلی که کاندیدای تعلق به یکی از کلاسهای لبه‌ها است را نشان می‌دهد.

3 خروجی مرحله سوم، تصویری است که مقدار پیکسلهای آن بیانگر میزان تعلق هر کدام از پیکسلها به کلاس لبه‌ها می‌باشد. مرکز هر کلاس را می‌توان مطابق روش بیان‌شده در مرحله قبل محاسبه نمود و برای میزان تعلق هر پیکسل به کلاس لبه‌ها می‌توان از ویژگیهای خاصی مانند میزان تغییرات روشنایی استفاده نمود. نظر به اهمیت فراوان تشخیص لبه در تصاویر، این موضوع به تفصیل در فصل 18 مورد بررسی قرار خواهد گرفت.

16-2-4-3: اندازه‌گیری ویژگیها به صورت فازی

در بسیاری از کاربردها لازم است تا ویژگی خاصی را در اشیای تصویر، مورد بررسی قرار بدهیم. مثلاً ممکن است بخواهیم محیط یا مساحت یا قطر یک شئ خاص را مشخص کنیم. چالش اساسی در این‌گونه مسائل، تعیین نواحی مرزی اشیای تصویر می‌باشد.

برای سهولت کار، بهتر است تا مسئله را به صورت دودویی درآوریم که این امر با روشهای مختلف رقمی‌سازی قابل انجام است. تابع مشخص‌کننده مساحت یک شکل، مطابق رابطه زیر تعریف می‌شود که Q مجموعه نقاط شیء مورد مطالعه و PQ مجموعه نقاط قرار گرفته روی قطر آن را نشان می‌دهد:

$$Fuzzy\ Area=\sum_{x\in Q}\mu(x)$$

تابع مشخص‌کننده اندازه قطر یک شکل، مطابق زیر است:

$$Fuzzy\,diameter=\sum_{x\in P\,Q}\mu(x)$$

همانگونه که مشاهده می‌کنید اندازه مساحت یک شکل یا اندازه قطر یک شکل هندسی در حالتی که میزان تعلق هر یک از پیکسلهای مربوط به هر یک برابر با یک باشد برابر با تعداد پیکسلها است. اما در حالتی که میزان تعلق هر پیکسل برابر با مقداری در بازه صفر تا یک است، اندازه مساحت و قطر شکل برابر با مجموع مقدار عضویتها خواهد بود که در حالت خاص می‌تواند حالت کلاسیک به صورت قطعی عمل نماید.

16-3: فیلتر کردن در تصاویر

برای استفاده از اطلاعات موجود در تصاویر، نیاز به استخراج ویژگیهای دلخواه از آنها داریم. لذا باید با کمک فیلترها، تصاویر را پردازش کنیم. در این قسمت، ابتدا تبدیل فوریه و سپس فیلترهای فرکانسی و مکانی را معرفی می‌کنیم. در ادامه مطلب، به بیان چند کاربرد جالب از روشهای فازی می‌پردازیم که با کمک آنها، جوابهای بهینه‌ای برای مسائل حاصل می‌شود.

تبدیل فوریه یکی از ابزارهای مهم در پردازش تصویر می‌باشد. در این تبدیل، تصویر را از حوزه مکان به حوزه فرکانس انتقال می‌دهیم تا انجام عملیاتی که در حوزه مکان روی تصویر، پیچیده یا غیرممکن بود بسیار کارآمدتر انجام بگیرد. با توجه به ارتباط بین فیلترها در حوزه مکان و فرکانس، می‌توان فیلترهای مکانی بزرگ را که به زمان پردازش زیادی نیاز دارند، با معادل آن در حوزه فرکانس جایگزین کرد تا زمان پردازش کاهش یابد.

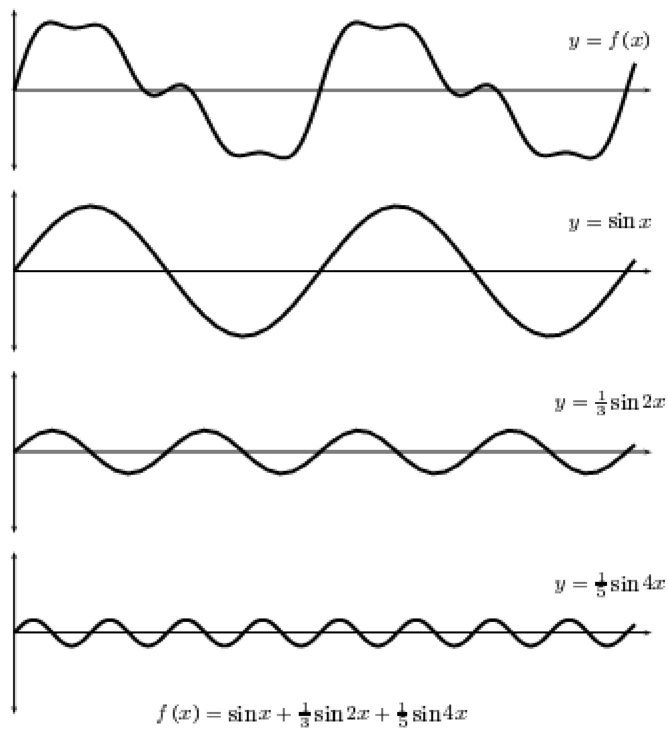
از دیگر ویژگیهای این تبدیل، می‌توان به امکان انجام اصلاحات روی یک طیف خاص فرکانسی برای طراحی فیلترهای بالاگذر و پایین‌گذر اشاره کرد.

16-3-1: تبدیل فوریه یک بعدی

بنا بر قضایای ریاضی، هر تابع متناوبی را می‌توان به صورت حاصل‌جمع توابع سینوسی و کسینوسی (با دامنه‌ها و فرکانسهای مختلف) بیان کرد. برای تقریب بعضی توابع، تعداد محدودی از جملات کفایت می‌کند، درحالی‌که برای نمایش توابع دیگر به تعداد نامحدودی از جملات نیاز داریم. واضح است که تعداد بیشتری از جملات باعث تقریب بهتری از تابع

خواهد شد. مثلاً برای نمایش دقیق تابع مربعی به بی‌نهایت جمله نیاز داریم و با تعداد محدودی جمله، فقط می‌توان (مطابق فرمول زیر) تقریبی از آن را به نمایش گذاشت.

$$f(x) = \sin(x) + \frac{1}{3}\sin(3x) + \frac{1}{5}\sin(5x) + \frac{1}{7}\sin(7x) + \frac{1}{9}\sin(9x)$$

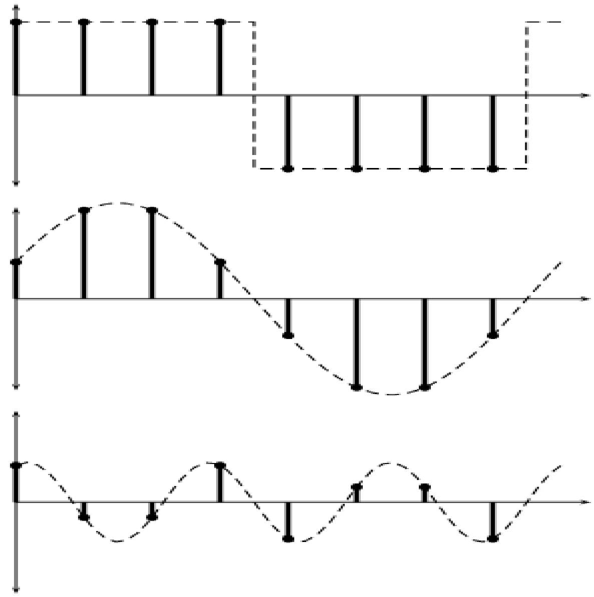


شکل16-3: تقریب یک تابع دلخواه با توابع سینوسی

شکل16-3 تقریب ساده‌ای از یک تابع را با استفاده از سه جمله اول سری فوریه نشان می‌دهد. برای پردازش یک تصویر، می‌توان آن را به عنوان یک تابع گسسته فرض کرد که دارای جملات محدودی می‌باشد. بنابراین، نیازمند ارائه یک تبدیل گسسته از توابع هستیم. به عنوان مثال، تابع گسسته و محدود زیر را (تقریبی از تابع مربعی) فرض می‌کنیم:

1 1 1 1 -1 -1 -1 -1

مطابق آنچه در شکل16-4 نشان داده شده است، می‌توان این تابع را به کمک دو تابع سینوسی گسسته تقریب زد.



شکل16 - 4: تقریب یک تابع گسسته

حال *تبدیل فوریه گسسته*[1] یک‌بعدی را تعریف می‌کنیم. برای سریهای زمانی گسسته

$$\boldsymbol{f} = [f_0, f_1, f_2, \ldots, f_{N-1}]$$

تبدیل فوریه گسسته را می‌توان به صورت دنباله زیر نمایش داد:

$$\boldsymbol{F} = [F_0, F_1, F_2, \ldots, F_{N-1}]$$

که در آن:

$$F_{\mathrm{u}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{x}=0}^{\mathrm{N}-1} \exp\left[-2\pi \mathrm{i} \frac{\mathrm{xu}}{\mathrm{N}}\right] f_{\mathrm{x}}$$

با توجه به فرمول بالا، مشخص است که DFT یک تبدیل خطی می‌باشد. برای تبدیل معکوس از رابطه زیر استفاده می‌شود:

[1] Discrete Fourier Transform (DFT)

$$x_u = \sum_{x=0}^{N-1} \exp\left[2\pi i \frac{xu}{N}\right] F_x$$

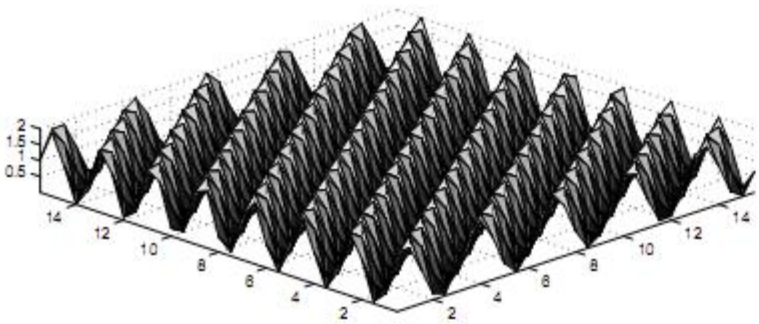
16-3-2: تبدیل فوریه دو بعدی

ورودی و خروجی DFT دوبعدی به فرم ماتریسی است. شکل کلی این تبدیل فوریه و معکوس آن مانند زیر است که f بیانگر ماتریس ورودی بوده و $\boldsymbol{F}$ تبدیل‌یافته آن می‌باشد.

$$\boldsymbol{F} = F\left(f\right)$$
$$f = F^{-1}\left(\boldsymbol{F}\right)$$

برای نمایش تابع و انجام تبدیلات در فضای دوبعدی از تابع سینوسی زیر (شکل16-5) استفاده می‌کنیم:

$$z = a \sin\left(bx + cy\right)$$



شکل16 - 5: نمودار تابع دوبعدی Z

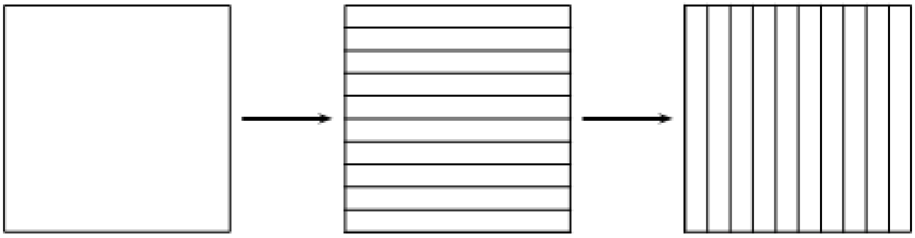
اگر تصویر به صورت یک ماتریس M×N نمایش داده شود، آنگاه می‌توان تبدیل فوریه و معکوس تبدیل فوریه را به صورت زیر نمایش داد:

$$F\left(u,v\right) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f\left(x,y\right) \exp\left[-2\pi i\left(\frac{xu}{M} + \frac{yv}{N}\right)\right]$$
$$f\left(u,v\right) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F\left(x,y\right) \exp\left[2\pi i\left(\frac{xu}{M} + \frac{yv}{N}\right)\right]$$

DFT دوبعدی علاوه بر خطی بودن، دارای خاصیت جداپذیری می‌باشد. توجه کنید که می‌توانیم تابع پایه داده‌شده را به صورت زیر نمایش دهیم:

$$\exp\left[2\pi \mathrm{i}\left(\frac{\mathrm{xu}}{\mathrm{M}}+\frac{\mathrm{yv}}{\mathrm{N}}\right)\right]=exp\left[2\pi i\ \frac{xu}{M}\right]exp\left[2\pi i\ \frac{yv}{N}\right]$$

جمله اول عبارت سمت راست فقط به x و u وابسته است و جمله دوم، فقط به y و v بستگی دارد. بنابراین می‌توان تبدیل فوریه دوبعدی را معادل با دو تبدیل فوریه یک‌بعدی دانست. برای این کار مطابق شکل16-6، ابتدا یکبار بر روی سطرها تبدیل فوریه می‌گیریم. سپس برای نتیجه حاصل، تبدیل فوریه را نسبت به ستونها به دست می‌آوریم.



شکل 16-6: DFT دوبعدی معادل با دوDFT تک‌بعدی

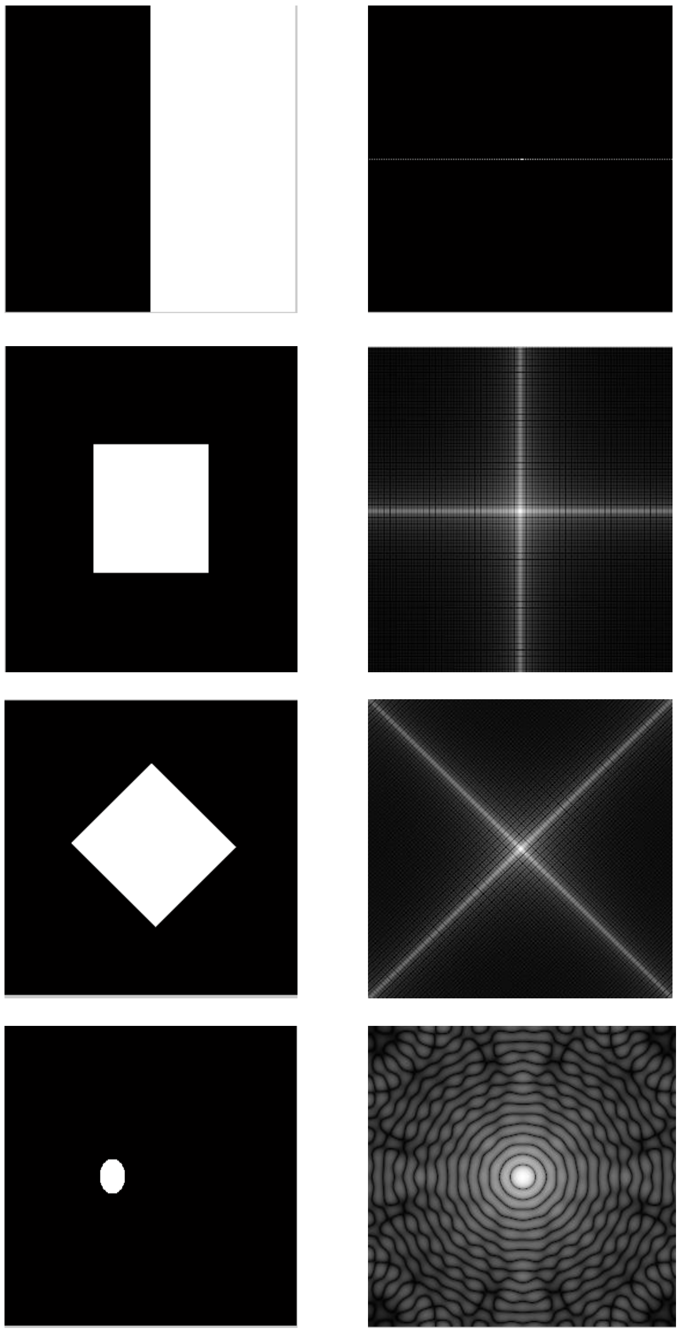
همچنین، عمل Convolution در حوزه زمان، به ضرب ساده در حوزه فرکانس تبدیل خواهد شد که کاهش فراوانی در حجم و زمان محاسبات را سبب می‌شود.

برای رسم تبدیل فوریه یک تابع، از قدرمطلق اعداد مختلط استفاده می‌شود. اگر مقدار ماکزیمم قدر مطلق برابر با m باشد، تمام اعداد ماتریس خروجی را بر m تقسیم می‌کنیم. برای نمایش اعداد حاصل، از لگاریتم (مطابق فرمول زیر) استفاده می‌شود:

$$\log\left(1+\left|F(u,v)\right|\right)$$

در شکل16-7، برای اشکال ساده‌ای نظیر خط، مربع، مربع دوران‌یافته و دایره، تبدیلات فوریه متناظر (سمت چپ شکل) رسم شده است. در بسیاری از موارد، می‌توان تغییراتی را روی فرکانسهای خاصی اعمال کرد. درحالی‌که انجام این کار به وسیله فیلترهای مکانی، ممکن است پیچیده و پرهزینه باشد. بنابراین از تبدیل فوریه برای فیلترکردن تصاویر استفاده می‌شود.

فیلترهای متعددی برای کاربرد در حوزه فرکانس وجود دارند که در مباحث مربوط به پردازش تصویر، به تفصیل بررسی می‌شوند. در ادامه، مرور کوتاهی بر انواع و ویژگیهای این دسته از فیلترها خواهیم داشت.



شکل16-7: تبدیل فوریه برای اشکال ساده هندسی

16-3-3: فیلترهای حوزه فرکانس

در این قسمت به معرفی و بررسی عملکرد چند فیلتر متداول در حوزه فرکانس، با ذکر مختصری از ویژگی و نحوه کار آنها خواهیم پرداخت.

16-3-3-1: فیلتر ایده‌آل

در این فیلتر، ابتدا ماتریس تبدیل فوریه تصویر را شیفت می‌دهیم تا مولفه DC آن به وسط تصویر منتقل شود. این فیلتر دارای دو نوع پایین‌گذر و بالاگذر می‌باشد که در حالت پایین‌گذر، تمام عناصر ماتریس از مرکز تا یک شعاع دلخواه (حد آستانه) را حفظ کرده، بقیه عناصر ماتریس را صفر می‌کنیم. به عبارت ساده‌تر، فرکانسهای پایین‌تر از یک عدد خاص را حفظ نموده، فرکانسهای بالاتر را حذف می‌کنیم. مکانیزم کار مانند فرمول ساده زیر است که در آن، D بیانگر عدد آستانه یا فرکانس قطع فیلتر می‌باشد:

$$f\left(x\right)=\begin{cases}1 & if\ \ x<D\\ 0 & if\ \ x\geq D\end{cases}$$

افزایش عدد حد آستانه، باعث افزایش شعاع تصویر و وضوح بیشتر در آن می‌شود و کاهش این عدد، بر ابهام تصویر می‌افزاید. فرکانسهای پایین‌تر سطوح خاکستری تغییرات اندکی دارند و برای نمایش پس‌زمینه و بافت تصاویر مناسب می‌باشند. بنابراین از فیلتر پایین‌گذر برای هموارسازی تصاویر استفاده می‌شود. این فیلترها باعث مبهم شدن تصویر و حذف جزئیات از آن می‌شوند. البته حذف جزئیات تصویر می‌تواند برای سیستمهای خودکار بینایی ماشین که نیاز به تشخیص سریع اشیا دارند، مناسب باشد.

فیلترهای بالاگذر دارای مکانیزم معکوس فیلترهای پایین‌گذر هستند. یعنی فرکانسهای بالا را مطابق فرمول زیر حفظ کرده، فرکانسهای پایین را حذف می‌کنند:

$$f\left(x\right)=\begin{cases}1 & if\ \ x>D\\ 0 & if\ \ x\leq D\end{cases}$$

در واقع، فرکانسهای بالا تغییرات ناگهانی سطح خاکستری در تصویر را نشان می‌دهند. تغییرات شدید و ناگهانی سطح خاکستری در تصاویر، معمولاً ناشی از لبه‌ها یا نویز می‌باشد.

به همین دلیل، از فیلتر بالاگذر برای استخراج و نگهداری لبه‌ها استفاده می‌گردد. از نتایج حاصله برای حذف نویز نیز می‌توان استفاده کرد.

16-3-3-2: فیلتر Butterworth

در فیلترهای ایده‌آل پایین‌گذر (بالاگذر)، عناصر درون (بیرون) یک دایره حفظ می‌شود و عناصر بیرون (درون) آن را حذف می‌کنیم. ایراد کلی این فیلترها، وقوع پدیده ring است که به صورت یک دایره ناخواسته در اطراف تصویر اصلی می‌باشد. وجود این دایره به علت قطع ناگهانی فرکانسها در فیلتر ایده آل است.

برای برطرف‌کردن این مشکل، در فیلترهای Butterworth تابعی تعریف می‌شود که به جای قطع ناگهانی فرکانس، به تدریج آن را تضعیف می‌کند. تعاریف ریاضی فیلتر بالاگذر و پایین‌گذر مطابق زیر بهبود می‌یابند:

فیلتر پایین‌گذر:

$$f(x) = \frac{1}{1 + (x/D)^{2n}}$$

فیلتر بالاگذر:

$$f(x) = \frac{1}{1 + (D/x)^{2n}}$$

16-3-3-3: فیلتر گوسی [1]

این فیلتر در حوزه مکانی عمل می‌کند و برای هموارسازی استفاده می‌شود. فرمول کلی آن (برای میانگین برابر با صفر) مطابق زیر است:

$$f(x) = e^{-\frac{x^2}{2\sigma^2}}$$

---

[1] Gaussian

پیاده‌سازی فیلتر گوسی در حوزه فرکانس حاوی نکته مهمی است. می‌دانیم که تبدیل فوریه یک تابع نمایی، همچنان یک تابع نمایی در حوزه فرکانس خواهد بود. با استفاده از این نکته، مراحل کار برای این فیلتر در حوزه فرکانس، مطابق زیر می‌باشد:

1. ماتریس گوسی را به کمک تابع گوسی محاسبه می‌کنیم.
2. ضرب عنصربه‌عنصر ماتریس حاصل و ماتریس تصویر را می‌یابیم.
3. از نتیجه حاصل، معکوس فرکانس می‌گیریم.

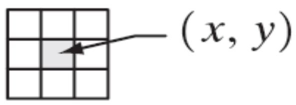
توضیح بیشتر درباره این فیلتر، در بخش مربوط به فیلترهای مکانی ارائه شده است.

16-3-4: فیلترهای حوزه مکانی

بسیاری از تغییرات اعمال‌شده روی تصاویر، در حوزه مکان انجام می‌شود، یعنی مکان پیکسل در تصویر حائز اهمیت می‌باشد. عملکرد فیلترهای مکانی براساس مقدار یک پیکسل خاص و نیز مقادیر پیکسلهای مجاور آن خواهد بود.

روش کار در این فیلترها بدین نحو است که یک ماسک (معمولاً یک مستطیل) را بر روی تصویر می‌لغزانیم و تصویر جدیدی به وجود می‌آوریم که در آن، مقدار هر پیکسل از اعمال ماسک و یک تابع روی مقادیر همسایه پیکسل حاصل می‌شود. معمولاً ابعاد ماسک، اعداد فرد می‌باشد (شکل16-8) تا یک پیکسل خاص به عنوان مرکز ماسک قرار بگیرد.

به ترکیب ماسک و تابع، فیلتر می‌گوییم. اگر تابعی که برای تولید مقادیر جدید مورد استفاده قرار می‌گیرد، تابعی خطی از مقادیر ماسک باشد، یک فیلتر خطی خواهیم داشت.



شکل16-8: یک ماسک 3×3 به مرکزیت پیکسل (x,y)

فرض کنید ماسک به صورت زیر باشد:

m(-1,-2)	m(-1,-1)	m(-1,0)	m(-1,1)	m(-1,2)
m(0,-2)	m(0,-1)	m(0,0)	m(0,1)	m(0,2)
m(1,-2)	m (1,-1)	m(1,0)	m(1,1)	m(1,2)

و مقادیر پیکسلها نیز به صورت زیر باشد:

p(i-1,j-2)	P(i-1,j-1)	P(i-1,j)	P(i-1,j+1)	P(i-1,j+2)
p(i,j-2)	P(i,j-1)	P(i,j)	P(i,j+1)	P(i,j+2)
p(i+1,j-2)	P(i+1,j-1)	P(i+1,j)	P(i+1,j+1)	P(i+1,j+2)

و تابع مورد استفاده، تابع خطی زیر باشد:

$$\sum_{s=-1}^{1} \sum_{t=-2}^{2} m(s,t)\, p(i+s, j+t)$$

در این صورت، گامهای زیر برای فیلترکردن انجام می‌شود:

1. انتقال ماسک به مکان مناسب
2. محاسبه حاصل‌ضربهای مقدار هر پیکسل در درایه متناظر آن در ماسک
3. یافتن مجموع این حاصل‌ضربها

16-3-4-1: فیلتر میانگین[1]

همان‌طور که اشاره شد، مقدار جدید هر پیکسل با توجه به تابع و مقادیر زیر ماسک به دست می‌آید. یکی از پرکاربردترین این فیلترها، فیلتر میانگین می‌باشد که مقدار جدید هر پیکسل را برابر با میانگین مقادیر موجود در زیر ماسک قرار می‌دهد. مثلاً برای ماسک به مرکزیت پیکسل با مقدار E مطابق زیر:

A	B	C
D	E	F
G	H	I

مقدار جدید این پیکسل از رابطه میانگین حاصل می‌شود:

$$E_{new} = \frac{1}{9}\left(A+B+C+D+E+F+G+H+I\right)$$

[1] Mean (Average) Filter

هر ماسک را می‌توان به فرم ماتریسی نشان داد. ماتریس متناظر با فیلتر میانگین دارای عناصر یکسان می‌باشد، بدین معنا که هر درایه این ماتریس، اهمیت و وزن یکسانی در تعیین مقادیر جدید دارد:

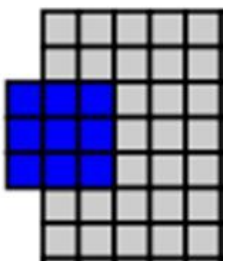
$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

ممکن است مقادیر متناظر برای پیکسلهای یک ماسک، با یکدیگر متفاوت باشند. در این صورت، پیکسلهای مجاور با ضرایب متفاوتی در یافتن مقدار جدید دخالت دارند. مثالی از این حالت برای ماسک زیر و ماتریس متناظر با آن دیده می‌شود:

$$E_{new} = \left(A - 2B + C - 2D + 4E - 2F + G - 2H + I\right)$$

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

سوالی که در اینجا مطرح می‌شود این است که محاسبه مقادیر برای پیکسلهای واقع در لبه تصویر چگونه انجام می‌شود؟ همان‌طورکه در شکل16-9 مشاهده می‌شود، قسمتی از ماسک در خارج تصویر قرار می‌گیرد و مقدار متناظری برای آن وجود ندارد.



شکل16 - 9: اعمال ماسک فیلتر روی لبه تصاویر

دو راهکار برای رفع این مشکل وجود دارد:

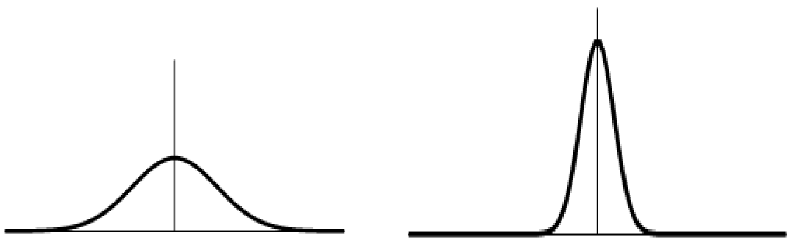
1. **نادیده‌گرفتن:** فقط مقادیری را مورد ارزیابی و پردازش قرار دهیم که ماسک با قرار گرفتن بر روی آن، کاملاً درون تصویر واقع می‌شود. اگر در این روش، اندازه ماسک بزرگ باشد مقدار زیادی از پیکسلهای حاشیه تصویر نادیده گرفته می‌شوند.
2. **توسعه حاشیه:** تصویر را تا اندازه مناسب برای پردازش توسعه دهیم و سپس آن را پردازش کنیم. به این ترتیب، هیچ قسمتی از تصویر نادیده گرفته نمی‌شود و بعد از پردازش، می‌توان تصویر اصلی را (با حذف توسعه انجام‌شده) بازیابی کرد. برای این نکته که افزایش حاشیه در تصاویر به چه صورتی باشد، موارد زیر را می‌توان پیشنهاد کرد:
   - برای پیکسلهای حاصل از توسعه، مقدار صفر را در نظر بگیریم.
   - مقدار حاشیه را تکرار کنیم.
   - تصویر را به صورت آینه‌وار تکرار کنیم.

16-3-4-2: فیلتر گوسی

یکی از مشهورترین فیلترهای مکانی، فیلتر گوسی است که متعلق به کلاس فیلترهای پایین‌گذر می‌باشد. این فیلتر بر مبنای تابع توزیع نرمال یک‌بعدی زیر عمل می‌کند:

$$f(x) = e^{-\frac{x^2}{2\sigma^2}}$$

شکل16-10 این تابع را برای دو مقدار مختلف σ (انحراف معیار) نشان می‌دهد. مقادیر کوچک انحراف معیار، تابع را فشرده‌تر و مقادیر بزرگ آن، تابع را گسترده‌تر می‌کند.

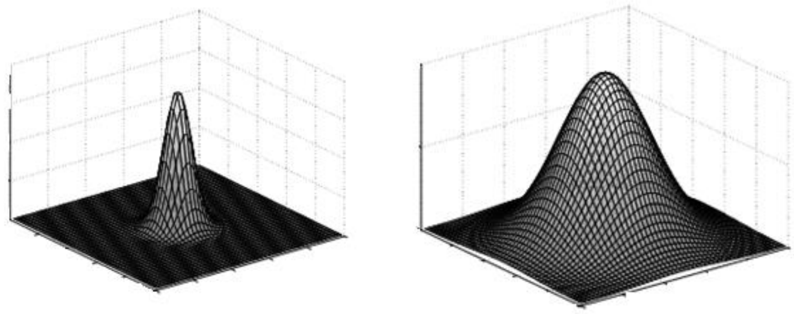


شکل16 - 10: تابع گوسی با σ کوچک (راست) و σبزرگ (چپ)

رابطه تابع فیلتر گوسی در فضای دوبعدی به صورت زیر خواهد بود:

$$f\left(x,y\right)=e^{-\frac{x^2+y^2}{2\sigma^2}}$$

این تابع در شکل16-10 برای دو مقدار مختلف σ رسم شده است:



شکل16-10: تابع گوسی دوبعدی برای $\sigma = 9$ (راست) و $\sigma = 3$ (چپ)

16-3-4-3: فیلتر مرتبه‌ای[1]

فیلتر مرتبه‌ای، یک فیلتر غیرخطی است. این فیلتر ابتدا مقادیر زیر ماسک را به ترتیب صعودی مرتب می‌کند. سپس برای n عنصر، مقداری را که در مرتبه دلخواه (از 1 تا n) قرار دارد انتخاب می‌نماید. سه مورد از این n حق انتخاب، معروف‌تر می‌باشند:

- فیلتر ماکزیمم: $r = n$
- فیلتر مینیمم: $r = 1$
- فیلتر میانه[2]: $r = (n+1)/2$ (برای n های فرد)

این فیلتر مقدار وسط لیست مرتب‌شده را انتخاب می‌کند تا داده‌ها را هموار کند و داده‌های پرت را حذف یا کم‌اثر نماید. این فیلتر کاربردهای فراوانی در حذف نویز دارد که در فصل 17 مورد بررسی بیشتر قرار خواهد گرفت.

[1] Rank order filter
[2] Median filter

16-4: کاربرد فازی در بهبود عملکرد فیلترها

در این قسمت روشهایی برای پردازش تصاویر به کمک مجموعه‌ای از قوانین فازی ارائه می‌شود. این مجموعه از قوانین، براساس یک نگاشت عمل می‌کند که با توجه به نتایج حاصل از عمل Convolution بین عملگرهای کلاسیک (غیرفازی) و تصاویر، ایجاد می‌شود. ماسک Convolution مکانی، براساس سادگی در پیاده‌سازی انتخاب می‌گردد.

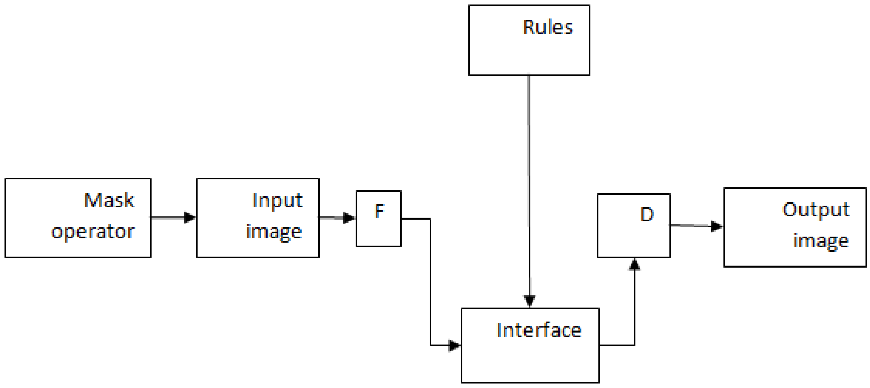
لازم به یادآوری است که فعالیتهای انجام‌شده در پردازش تصویر، با دو هدف متفاوت صورت می‌گیرد: ممکن است تغییرات تصویر در جهتی انجام شود که تفسیر آن را برای چشم و ذهن انسان آسان‌تر نماید و نیز ممکن است که هدف ما، کسب اطلاعات از تصاویر به منظور استفاده در سیستمهای خودکار (عمدتاً موارد مربوط به بینایی ماشین) باشد.

به منظور برآورده‌کردن این دو هدف، از علوم مختلف مانند ریاضی، فیزیک، روانشناسی و ... استفاده می‌شود تا کاربردهایی مانند تشخیص اشیا، فشرده‌سازی تصاویر، قطعه‌بندی انجام پذیرد.

در تکنیکهای متعدد بهبود و پردازش تصاویر، سعی بر این است که روی هر ناحیه از تصویر، مناسب‌ترین عملگر را برای رسیدن به بهترین نتیجه انتخاب نماییم. یک روش مفید برای مدل‌سازی سیستم پردازش تصاویر، به صورت دنباله‌ای از مراحل با if-then های فازی بیان می‌شود.

16-4-1: توسعه سیستم

سیستم توسعه‌یافته فازی برای پردازش تصاویر در شکل16-12نمایش داده شده است. ورودی این سیستم، نتایج ماسکهای مختلف و تصاویر ورودی می‌باشد. باید توجه داشت که مقادیر ماسکها غیرفازی هستند. پس لازم است تا با عمل فازی‌سازی (مرحله F) به مقادیر فازی تبدیل شوند. در مرحله بعدی با توجه به این مقادیر و به کمک بانک دانش موجود، پردازش و استنتاجهای لازم روی تصویر انجام می‌گیرد و با غیرفازی‌سازی (مرحله D) برای خروجی این قسمت، نتایج نهایی را نمایش می‌دهیم.



شکل16-12: شمای کلی سیستم فازی توسعه‌یافته برای پردازش تصاویر

برای پیاده‌سازی بانک عملگرها، تعدادی از ماسکهای Convolution شامل سه ماسک برای استخراج لبه، هموارسازی و ارتقاء[1] تصویر انتخاب می‌شوند.

برای تصمیم‌گیری در مورد استخراج لبه، می‌توان از اطلاعات محلی مانند گرادیان استفاده کرد. برای انتخاب و گزینش گرادیان می‌توان از اطلاعاتی که توسط ماسکهای دیگر به دست می‌آید استفاده نمود تا خروجی مطمئن‌تری حاصل شود. هموارسازی را می‌توان یک فیلتر پایین‌گذر برای گرفتن میانگین در یک ناحیه برای یک پیکسل دانست و برای عمل ارتقاء تصویر نیز از فیلتر سوبل استفاده کرد. مثالهایی از این قوانین عبارتند از:

If edges is SMALL and high-boost is SMALL and Smoothing is SMALL then p(i,j) is SMALL

If edges is SMALL and high-boost is SMALL and smoothing is BIG then p(i,j) is BIG

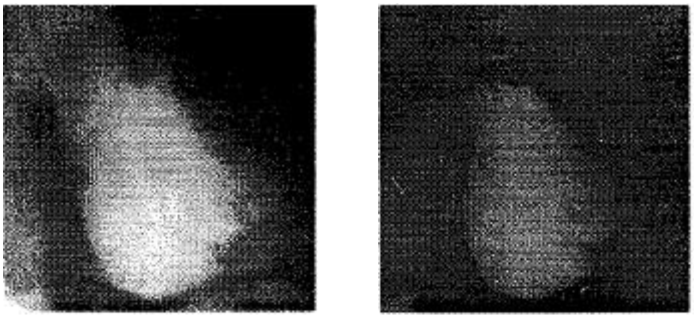
If edges is BIG and high-boost is SMALL and Smoothing is MEDIUM then p(i,j) is BIG

If edges is MEDIUM and high-boost is SMALL and Smoothing is BIG then p(i,j) is BIG

[1] Boost

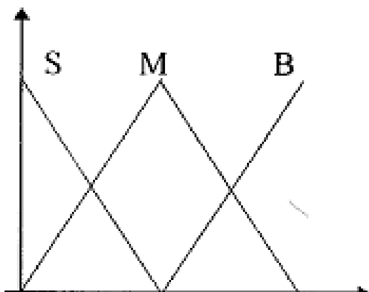
16-4-2: تصویر پزشکی

در اینجا با استفاده از این سیستم، یک تصویر پزشکی را پردازش می‌کنیم. هدف این است که ناحیه روشن‌تر درون تصویر را از ناحیه تاریک جدا کنیم. پس از انجام پردازشهای لازم روی تصویر ورودی، ناحیه روشن داخل تصویر را به وسیله سطح خاکستری مناسب از پس‌زمینه تصویر جدا می‌کنیم. نتیجه کار در شکل16-13دیده می‌شود.



شکل16-13: تصویر پزشکی (راست) و پردازش‌شده آن (چپ)

توابع عضویت (Small, Medium, Big) سیستم فازی در شکل16-14به نمایش درآمده است. به علت شکل توابع عضویت، سطوح خاکستری نگاشت‌یافته توسط این سیستم در بازه [0,255] می‌باشند. در صورتی که مقدار یک پیکسل در هیچ‌کدام از قوانین سیستم صدق نکند، مقدار صفر را به آن نسبت می‌دهیم.



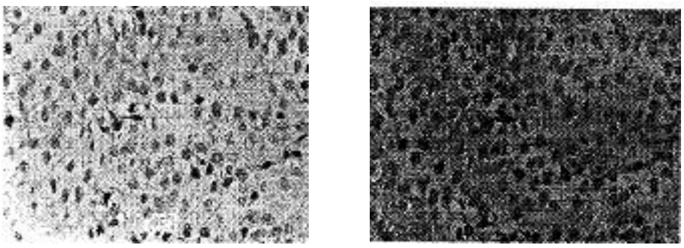
شکل16-14: توابع عضویت سیستم فازی

16-4-3: تصویر سلولی

کاربرد دیگر برای این سیستم، کلاس‌بندی داده‌های تصویر می‌باشد. شکل16-15سمت چپ، یک تصویر سلولی را نشان می‌دهد. پیکسلهای موجود در این تصویر متعلق به یکی از دو کلاس هسته یا بیرون هسته می‌باشند. هسته‌ها با سطوح خاکستری پایین‌تر (تاریک‌تر) از نواحی بیرونی که دارای سطوح بالاتری (روشن‌تر) می‌باشند، قابل تشخیص هستند.

اولین راه‌حلی که به ذهن می‌رسد، استفاده از یک حد آستانه برای جداکردن دو کلاس از یکدیگر است. اما با نگاهی دقیق‌تر، مشاهده می‌کنیم که در درون هسته‌ها، نواحی روشن وجود دارد و نواحی بیرون هسته‌ها نیز شامل پیکسلهای تاریک می‌باشد. به همین دلیل نیاز به سیستم دقیق‌تری برای تفکیک دو کلاس از هم داریم.

به کمک پردازش تصویر فازی با پایگاه دانش شامل قوانین فازی برای p(i,j)، تصویر را به دو کلاس هسته و بیرون هسته تقسیم می‌کنیم. نتیجه نهایی کار در شکل سمت راست قابل رویت است.



شکل16 - 15: تصویر اولیه سلول (چپ) و تصویر پردازش شده (راست)

16-4-4: تصویر مدار چاپی

تصویر دیگری که با این سیستم مورد پردازش قرار گرفته است، تصویر برد مدار چاپی[1] می‌باشد. در این‌گونه تصاویر، سه کلاس متمایز شامل خطوط مسی، عایق[2] و سوراخها را

---

[1] Printed Circuit Board (PCB)
[2] Isolator

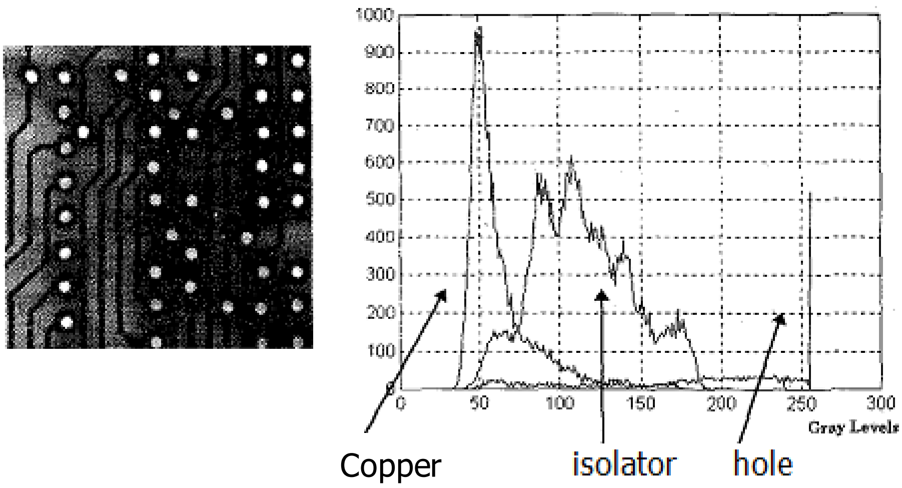
می‌توان تعریف کرد. هدف این است که در تصویر بهبودیافته نهایی، پیکسلهای متعلق به هر کدام از کلاسها دارای مقادیری به صورت زیر باشند:

$$pixel \in copper\ class \rightarrow gray\ level\ 0$$

$$pixel \in isolator\ class \rightarrow gray\ level\ 127$$

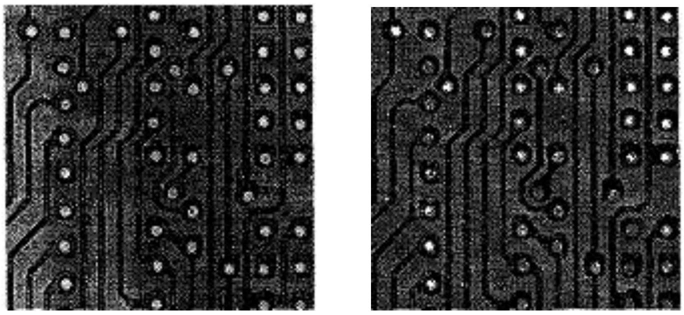
$$pixel \in hole\ class \rightarrow gray\ level\ 255$$

تصویر برد مدارچاپی و هیستوگرام نظیر آن در شکل16-16 دیده می‌شود. همان‌طورکه مشاهده می‌شود، کلاسها دارای نواحی هم‌پوشان می‌باشند که این امر، ممکن است باعث بروز خطا شود. به عنوان یک راهکار، می‌توان از اصول احتمال بیزین برای تفکیک کلاسها از یکدیگر استفاده کرد.



شکل16-16: تصویر برد مدار چاپی (چپ) وهیستوگرام نظیر (راست)

شکل16-17به مقایسه عملکرد روش حد آستانه و روش سیستم فازی برای قطعه‌بندی می‌پردازد. همان‌طور که در این شکل مشاهده می‌شود، سیستم فازی در تشخیص کلاس خطوط مسی عملکرد بهتری دارد. اما در تشخیص پیکسلهای متعلق به کلاس سوراخها، این روش حد آستانه است که بهتر عمل می‌کند.



شکل16-17: تصویر پردازش‌یافته با سیستم فازی (راست) و حد آستانه (چپ)

در این فصل، کاربردهای متعددی از پردازش فازی تصاویر مورد بررسی قرار گرفت و نتایج حاصله با روشهای کلاسیک (غیرفازی) مقایسه شد. نظر به اهمیت موضوع، در دو فصل آتی به توضیح تفصیلی دو مبحث مهم در پردازش تصویر و بینایی ماشین می‌پردازیم.

در فصل 17 روشهای فازی در رفع نویزهای مختلف را بررسی می‌کنیم و در فصل 18 به کاربرد منطق فازی در تشخیص لبه‌های موجود در تصاویر خواهیم پرداخت.